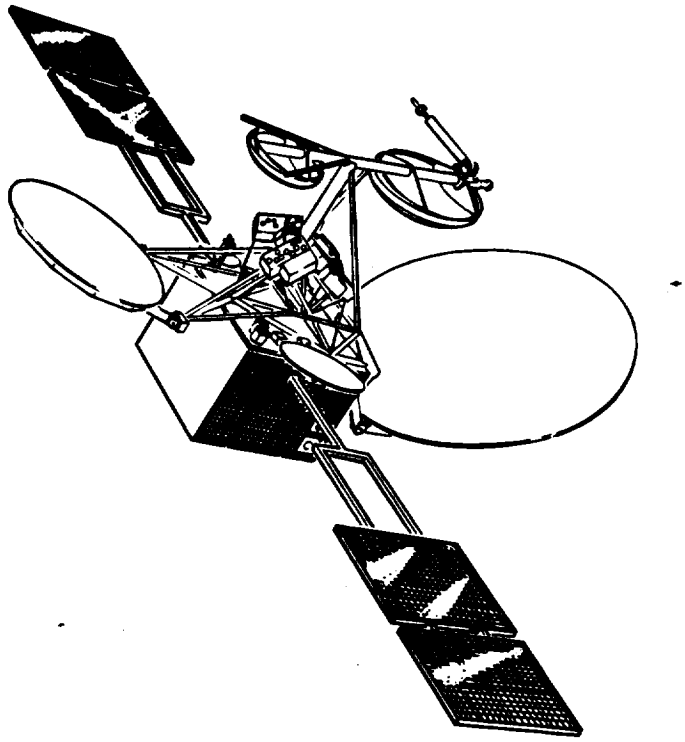
NASA Contractor Report 189161

Advanced Communications Technology Satellite High Burst Rate Link Evaluation Terminal Experiment Control and Monitor Software Maintenance Manual

Version 1.0

Richard C. Reinhart
Analex Corporation
Brook Park, Ohio



December 1992

Prepared for
Lewis Research Center
Under Contract NAS3–25776

National Aeronautics and Space Administration

High Burst Rate Link Evaluation Terminal
Experiment Control and Monitor Software Maintenance manual

Version 1.0, December 1992

Table of Contents

List of Figures

List of Tables

HIGH BURST RATE LINK EVALUATION TERMINAL

CONTROL AND PERFORMANCE MONITOR SUBSYSTEM

Experiment Control and Monitor Software Maintenance Manual

Version 1.0, December 1992

1.0 INTRODUCTION

1.1 Identification of Document

This is the LET-EC&M Software Maintenance Manual for the NASA Advanced Communications Technology Satellite (ACTS) High Burst Rate Link Evaluation Terminal (HBR-LET). This document complies with the NASA Software Management and Assurance Program (SMAP) guidelines in the Information System Life-Cycle and Documentation Standards, Release 4.3. This manual is one component of the Control and Performance Monitor (C&PM) Subsystem document series.

1.2 Scope of Document

The LET-EC&M Software Maintenance Manual was compiled for developers who know FORTRAN VII computer language. All HBR-LET software resides on a Concurrent Corporation 3205 minicomputer utilizing Concurrent's OS/32 operating system. Familiarity with the Concurrent minicomputer's capabilities is beneficial.

Familiarity with the various types of instruments and devices used by the LET, and a working knowledge of their operation is required to use this document.

1.3 Purpose of Document

This document provides information about the development, operation, and maintenance of the Experiment Control and Monitor (EC&M) Software. The LET-EC&M Software Maintenance Manual will assist programmers with future software modifications and upgrades as necessary. Refer to the HBR-LET EC&M User's Guide for information concerning the specific operation of the EC&M Software.

1.4 Document Status and Schedule

Version 1.0 is the first Contractor Report publication of the LET-EC&M Software User's Guide. A preliminary version (not formally published) was released in May, 1991 at which time the HBR-LET was fully integrated and tested.

The HBR-LET participated in ACTS System test at General Electric's Astro-Space division in July, 1992. Minor modifications were made to the software and are discussed in the program summaries.

1.5 Document Organization

This document consists of eight sections. Sections 1 and 2 introduce and describe this document and other related documents. Section 3 provides specific information concerning the implementation of the source code and the associated data files. Hardware and software dependencies are also discussed. Modification aids necessary for future revisions are discussed in Section 4. Section 5 describes the adaptations for the source code, while Sections 6 and 7 contain a list of the abbreviations and acronyms used in this document and a glossary of terms. Finally, Section 8 contains appendices describing the EC&M Software Instrument Table, program structure, procedure library and task commons in detail. A list of the instruments controlled by the EC&M Software is also provided.

2.0 RELATED DOCUMENTATION

2.1 Parent Document

None

2.2 Applicable Documents

The following instrument manuals and other related documentation are referenced herein and pertain to this document. Refer to these documents for additional operating procedures.

1. *EIP 545b Microwave Frequency Counter Users Manual,* EIP Microwave Inc., March 1980.

2. *General Microwave 60 dB Attenuator Controller User's Manual,* NASA LeRC, Ivansic, W., 1990.

3. *HBR-LET Acceptance Test Report, IF Noise Unit,* NASA LeRC, Fujikawa, G., 1990.

4. *HBR-LET Experiment Control and Monitor Software User's Manual,* NASA Contractor Report 189160, NASA LeRC, Reinhart, R.C., October, 1992.

5. *HP (Hewlett Packard) 11713A Attenuator/Switch Driver, Operating and Service Manual,* Hewlett Packard Company, 1985.

6. *HP 437B Power Meter, Operating and Service Manual,* Hewlett Packard Company, 1985.

7. *HP 59306A Relay Actuator, Operating and Service Manual,* Hewlett Packard Company, 1986.

8. *HP 6632A DC Power Supply, Operating and Service Manual,* Hewlett Packard Company, 1986.

9. *MENUP Menu Driver documentation,* NASA LeRC, Andro, M., 1987.

10. *SA (Scientific Atlantic) 930 Beacon Receiver Users Manual,* Scientific Atlanta Inc.

11. *Satellite Ground-Terminal User Simulation,* NASA Technical Memorandum 100234, NASA LeRC, Shalkhauser, M. J., January, 1988.

12. *TERMEB Terminal documentation,* NASA LeRC, Walters, J., 1984.

13. *Wavetek 8502A Peak Power Meters, Operating and Maintenance Manual,* Wavetek Microwave, Inc., 1989.

2.3 Information Documents

Refer to the following documents for additional information on computer operation and the various editors.

1. *FORTRAN VII Language System Reference and Users Guide,* Perkin Elmer Corporation, 1983.

2. *MEDIT User Guide,* Perkin Elmer corporation, 1984.

3. *MicroEMACS Reference Manual,* Lawrence, D.M. and Straight, B., 1987.

4. *MTM Primer,* Concurrent Computer, 1986.

5. *OS/32 EDIT User Guide,* Concurrent Computer Corporation, 1986.

6. *OS/32 Supervisor Call Reference Manual,* Concurrent Computer Corporation, 1986.

3.0 IMPLEMENTATION DETAILS

3.1 Specific Data Representations and Formats

The LET-EC&M software system comprises three independent programs. The Instrument Definition, Sequence Definition, and Sequence Execution programs are designed to create and to execute a wide variety of HBR-LET experiments. Multiple device control, data acquisition, instrument monitoring, and bit error rate (BER) measurement testing are among the EC&M Software functions available.

There are several data representations that must be observed when modifications are made to the EC&M Software. Each program in the EC&M Software system creates a data file which stores information about the instrument's configuration, sequence commands, or instrument data. The format type and corresponding file types are discussed. All files referenced in the following sections are located in the Control and Performance Monitor Software Account (CPMSA) or in a private user account.

3.1.1 Instrument Definition File

The first component of the EC&M Software, the Instrument Definition Software (IDS), enables a user to define the instrumentation that will be used within an experiment. The IDS also allows the user to provide the instrument's initial configuration and control parameters that are required by the EC&M Software.

This component creates the Instrument Definition File (formerly reffered to as the Test Definition File or TDF). All data input using the IDS is saved to the TDF in an array format, according to the defined instrument. Each instrument occupies one column of data within the array. Instrument parameters are saved in various rows within the column and are described in detail in Appendix I of the EC&M Software User's Guide. TDF files are saved in a binary format with a record length of 256. These files can reside in any private account but must have the filename extension TDF.

3.1.2 Sequence Definition File

The sequence of commands that will carry out the desired experiment are defined in the second component of the EC&M Software, the Sequence Definition Software (SDS). Sequence commands are available to control both the instrumentation used in the experiment and in sequence execution. Sequence commands are described in detail in Section 6 of the EC&M Software User's Guide.

Commands defined using the SDS are saved to the Sequence Definition File in an array format similar to the TDF file. Each sequence command is stored in a separate column with its required parameters in various rows within the column. The Sequence Definition Array (SDA) is also saved in a binary format with a record length of 256. SDA files can reside in any private account, but must have the filename extension SDA.

3.1.3 Data Acquisition File

The Sequence Execution Software (SES) executes the predefined sequence while performing data acquisition of all instruments included in the test. Instrument output readings and errors messages are displayed to the user's terminal during sequence execution.

Data gathered during sequence execution is stored to a file with each individual measurement time stamped for future reference and analysis. The name of the data file created by the SES has the same name as the corresponding TDF and SDA files, but with the extension DAT. The DAT file is stored in a binary format with a record length of 512. One can view sequence data files using the C&PM Post Processing Software.

3.1.4 Instrument File

An ASCII data file named **INSTRMTS.LET** stores the information for the Instrument Table used in the EC&M Software. This file contains a list of the instruments currently available to the HBR-LET, also listed in Appendix B. Each instrument, has a NASA tag number, interface type, IEEE 488 bus number, instrument type, label, identification, description, and address assigned to it at time of publication. Refer to Appendix A for information on the correct record entry and column position for each item listed in the data file. For more information on the Instrument Table, consult Section 4 of the EC&M Software User's Guide. Modifications to the Instrument File must not change its current format as required by the programs within the EC&M Software system.

3.2 Operating System Interfaces and Dependencies

All HBR-LET EC&M Software runs under Concurrent's OS/32 Operating System, Version 8-3.2 or later. The Concurrent computer also has a Multi-Terminal Monitor (MTM) which oversees the communication and sharing of system resources between the individual users on the system. Tasks (programs) are not required to run under the control of MTM. The EC&M Software system runs independently of MTM which enables multiple tasks to share a common output device.

Serial communication ports dedicated to the LET project must be removed from MTM for proper operation. Failure to remove the ports from MTM will result in an inability to initialize the EC&M Software. Communication port assignments are listed in Section 3.4. For instructions on removing a port from MTM, consult the *MTM Primer* or the *EC&M Software User's Guide*.

3.3 Support Software and Libraries

Each component of the EC&M Software has one main program and many libraries. Several libraries are also used by the HBR-LET Check-Out Software, but are documented here for completeness. A detailed description of each program, library, and task common (area of computer memory utilized by two or more programs executing simultaneously) is provided in Appendix C, Appendix D, and Appendix E, respectively.

The program technical summary headings contain the program name, author names, and short histories of development. Also included in the summary is an outline of a program's major functions, the task commons it shares, the logical units it uses and their device or file assignment. A brief explanation of each procedure and its corresponding library is provided.

The library documentation provides additional information on each procedure. The library name, author, and history are found at the beginning of each summary. A description of the variables passed in the procedure calls, their type declaration, and an explanation of each procedure function is also provided. A list of the programs that use a particular library is also included for cross referencing with the program documentation.

Appendix E describes the task commons used by the Sequence Execution control and Sequence Execution display programs. These two programs use task commons to share archived data, status flags, and the Instrument Definition and Sequence Definition arrays. The task common summaries describe the variables, their type declarations and a list of programs that use the particular task common.

3.4 Hardware Devices

All HBR-LET EC&M Software resides on a Concurrent Corporation 3205 minicomputer and is written in FORTRAN VII. The menu driver utilized by the EC&M Software requires a WYSE 50 or compatible terminal for best operation. The EC&M Software can also be executed from a PC using a modem to communicate with the Concurrent minicomputer.

A printer must be connected to the minicomputer to produce a hard copy of the data compiled by the EC&M Software, although it is not required for normal operation. It is possible to view all the data gathered by the SES using the Post Processing Software from within a private account under MTM.

Two types of interfaces are used to communicate with the instrumentation and devices used in the LET. An IEEE 488 General Purpose Interface Bus (GPIB) provides a link to all programmable radio frequency (RF) instrumentation. An RS-232 interface is used to communicate with the Digital Ground Terminal (DGT), Data Generator/Data Checker (DG/DC), and General Microwave (GM) Attenuators.

Serial communication ports of the Concurrent 3205 minicomputer have been assigned dedicated devices (listed in Table 3-1). The DGT, DG/DC, and the GM attenuators must be connected to the assigned port for the system to function properly. Failure to do so will result in the inability to install or execute the EC&M Software. The IEEE 488 bus is connected to the computer via an IEEE 488 circuit board.

Comm Port	Dedicated Device	Rate/Bits/Parity
CRT1:	Beacon Data	9600/7/even
CRT2:	Menu Display	9600/7/even
CRT3:	Data Generator/Data Checker	9600/7/even
CRT4:	Digital Ground Terminal	9600/7/even
CRT5:	General Microwave Attenuators	9600/8/none
CRT6:	Protocol Receive Link	9600/8/none
CRT7:	Protocol Transmit Link	9600/8/none

Table 3-1 Concurrent 3205 Communication Port Assignments

IEEE 488 bus extenders (IOTECH brand) are used with the IEEE 488 GPIB to communicate with instruments outside the maximum range of the standard GPIB. The maximum length of the IEEE 488 bus is about 20m depending on the number of instruments connected to the bus. IOTECH bus extenders are interconnected using an RS-232 cable which can carry a signal about 50ft.

An IOTECH bus expander is also used to allow additional instruments to be connected to the same GPIB and extend the maximum length. The standard IEEE 488 GPIB can support 14 instruments. The IOTECH expander can support an additional 14 instruments, thus a maximum of 27 instruments and 1 bus expander can be placed on a single GPIB (The IOTECH bus expander counts as an instrument when connected to the GPIB). The Concurrent 3205 currently uses one GPIB; therefore, the IOTECH bus expander must be used to accommodate the instrumentation used in the HBR-LET system.

4.0 MODIFICATION AIDS

4.1 Instrument Augmentation

The EC&M Software accesses an ASCII data file named **INSTRMTS.LET** located in the CPMSA to construct the Instrument Table used in the IDS. This file contains a list of the instruments currently available to the HBR-LET. The information required for proper execution of the EC&M Software is associated with each instrument.

Users may easily modify the Instrument File at any time to change instrument address assignments, add new instruments of an existing type, or change other instrument data. To add new types of instruments or devices to the Instrument Definition Software, follow the procedure outlined in this section.

1) Assign the instrument or device an unused type number and add it to the instrument file as described in the LET-EC&M User's Guide. Table 4-1 lists the type numbers and their corresponding instruments.

Type Number	Instrument
1	HP 436A Power Meter
2	General Microwave (GM) Attenuator
3	HP 11713A Attenuator/Switch Driver
4	Wavetek 8502 Peak Power Meter
5	Scientific Atlantic 930 Beacon Receiver
6	HP 59306A Relay Actuator
7	EIP 545B Frequency Counter
8	HP 6632A DC Power Supply
9	HP 11713A Switch Driver (Only)
10	Data Generator
11	Data Checker
12	HP Universal Counter
13	Keithley Multimeter
14	Philips Frequency Counter
15	HP Dual Sensor Power Meter

Type Number	Instrument
16	HP signal Generator
17	Eb/No Calculation
18	HP 6624A Power Supply
19	HP 437B Power Meter
20	GM Power Meter
21	Digital Ground Terminal
22	HP sweep Oscillator
23	HP Spectrum Analyzer
24	Wiltron Signal Generator
25	Compu-motor Indexers

Table 4-1 Instrument Type Numbers

2) Add the new instrument's parameters to the user default file M1:DEFAULT.LET/111. Include the additional parameters in the M1:LABELS.LET/111 file, which is used with the user default file. The default parameter files, DEFAULT.LET and LABELS.LET are described in Section 5 of this document.

3) Modify the Instrument Definition Menu Software, M1:TDFLET.MNU/111, to include a menu displaying the parameters associated with the new instrument or device.

4) Develop the necessary software to initialize and decode the menu that coincides with the new instrument. Currently, two subroutines exist for each instrument in the TDF software. One subroutine initializes the instrument menu with the default parameters and a second subroutine decodes the instrument menu into the proper locations within the Instrument Array (RINST/INST).

4.2 Sequence Command Expansion

Sequence commands added to the existing EC&M Software system will require modifications to both the Sequence Definition and Sequence Execution programs. Both the SDS and SES have modular designs with separate subroutines which aids in debugging errors and making future modifications. To add commands to the SDS and SES, follow the steps listed below:

1) Control numbers specify a group of sequence commands of the same type. Assign a sequence control number to the new command if it does not fit into an existing category listed in Table 4-2.

Control Number	Function
100000	Executes the specified sequence command. Single execution statement.
200000	Executes the SET UP A LOOP sequence command. Single execution statement within a loop. Sequence commands are repeated until an end of loop control number is encountered.
299999	Executes the END LOOP sequence command. Terminates the loop and continues the sequence.
300000	Executes the CALL A SUBSEQUENCE sequence command.
399998	Executes the RETURN TO MAIN SEQUENCE sequence command. Terminates Subsequence execution and transfers control back to the Main Sequence.
999997	Executes the END SEQUENCE sequence command. Terminates the execution of the main sequence.

Table 4-2 Sequence Execution Control Numbers

Note: User input label numbers are added to the control number for the purpose of **GOTO** and **Check a Parameter** sequence commands. To avoid overlapping control categories as a result of a label number, the maximum label number is 99997.

2) Assign an unused action number to the sequence command. Existing action numbers are listed in Table 4-3. These numbers are used to direct the software to execute the proper command.

Action Number	Action
21	Set a Parameter
60	Zero Power Meter
61	Step a Parameter
101	Wait
102	Goto Statement
110	Check a Parameter
301	Start Data Generator
302	Stop Data Generator
303	Stop Data Checker
311	Set Data Generator Errors
313	Perform BER Measurement
315	DGT Command

Table 4-3 Sequence Command Action Numbers

3) Assign the new command a one-to-four character mnemonic to assist users in recognizing commands within the sequence. The mnemonic is used to identify sequence commands in sequence display, printouts, editing, and execution. Existing mnemonics are listed in Table 4-4.

Mnemonic	Sequence Command
ATTN	Set or Step Attenuation Parameter
BCNR	Set SA Beacon Receiver Parameter
BER	Perform BER Measurement
CKBE	Check BER Parameter
CKCU	Check DC Power Supply Current Parameter
CKEB	Check Eb/No Parameter
CKFR	Check Frequency Parameter
CKPR	Check Power Level Parameter
CKVT	Check DC Power Supply Voltage Parameter
CSUB	Call Subsequence Command
DGTC	Digital Ground Terminal Command
EBN0	Set or Step Eb/No Parameter
ENDL	End Loop
FREQ	Set Frequency Parameter
GOTO	Goto Statement
LOOP	Set Up a Loop
NERR	Set Data Generator Errors
PSCR	Step DC Power Supply Current Parameter.
PSVT	Step DC Power Supply Voltage Parameter.
PWCR	Set DC Power Supply Current Parameter.

Mnemonic	Sequence Command
PWVT	Set DC Power Supply Voltage Parameter.
SPDC	Stop Data Checker
SPDG	Stop Data Generator
STDG	Start Data Generator
TGSW	Set Parameter Toggle Switch Command
WAIT	Wait Command
WVTK	Set Wavetek Parameter
ZERO	Zero Power Meter

Table 4-4 Sequence Command Action Mnemonics

4) Identify the required parameters for the new command and add them to the Instrument and Sequence Editor Documentation found in the EC&M Software User's Guide.

5) Develop software to execute the new command. Copy the software used in the CTRLLET program to the loop and subsequence subroutines in the SES. Insure that the new command can be executed from within a loop and subsequence.

4.3 Instrument and Sequence Editor

An Instrument and Sequence editor program enables users to view or edit existing instarument and sequence files. Each instrument's initial configuration parameters and sequence command parameters are described in Appendix I of the *HBR-LET EC&M Software User's Guide* (bound under separate cover). The guide also includes the location of each instrument and sequence parameter within the instraument and sequence file respectively. Use the Instrument and Sequence Editor for all interaction with the instrument or sequence files. Related instrument and sequence files share the same filename but have the appropriate extensions described in the *HBR-LET EC&M User's Guide*.

5.0 CODE ADAPTATION

5.1 Link File Modifications

Link files join together one or more object files to create an executable task file. For proper operation, several system options must be included in the link files of the main programs (TDFLET, SDALET, DSPLET, CTRLLET) used in the EC&M Software. A brief description of these options follow. Refer to the *MTM OS/32 User's Manuals* for further information on the available link options.

ACPRIV	Provides a user task with additional privileges for accessing files in private accounts. This option is used by all programs in the EC&M Software.
COMMUNICATE	Specifies that the executing task can perform the SVC6 intertask communication function. This feature is required for sending messages between the Sequence Execution and Display Software.
CONTROL	Specifies that the executing task can perform the SVC6 intertask control function. This feature is required to enable the Display program to load and start the SES.
IOB	Specifies the number of I/O blocks assigned to a task.
LU	Specifies the maximum number of logical units that can be assigned by a task.
XSVC1	Indicates that, if a task executes a SVC1 with bit 7 of the function code set, the options specified by the SVC1 extended option field are to be executed.

5.2 Default Initialization File

Two data files correspond to the default values assigned to the instrument parameters. The user file, M1:DEFAULT.LET/111, is designed to be edited by a system operator for modifying existing default parameter values.

The DEFAULT.LET file contains the various instruments and instrument parameters used in the EC&M Software. For this description, the following definitions are used. Each line of the file is called a field or record. Instrument and parameter names are called field descriptors. Programs within the IDS require that

the instrument and parameter **names** and their **locations** remain unchanged within the field and, therefore, must NOT be altered by the user. Figure 5-1 illustrates several fields within the default parameter file. Only a portion of the file introduction is shown in the figure.

```
This is the default parameter file for the High Burst Rate
Link Evaluation Terminal...

HP POWER METER 437B

RANGE                      1.0    AUTO
TRIGGER MODE               3.0    FREE RUN
LOW LIMIT               -100.0
HIGH LIMIT                10.0
CALIBRATION FACTOR        96.8
UNITS                      1.0    dBm
OUT OF LIMIT ACTION        1.0    CONTINUE TEST

GM ATTENUATOR

INITIAL ATTENUATION       25.0
LOW LIMIT                 10.0
HIGH LIMIT                60.0
UNITS                      1.0    dB
OUT OF LIMIT ACTION        2.0    HALT TEST
```

Figure 5-1 DEFAULT.LET Default File

Users can modify the value of any record in the file. Either a space or comma must appear both before and after the numerical value for it to be interpreted by the software. Control characters (tabs) are not permitted in the file. The numerical value must be the first entry following the field descriptor; its location, however, is arbitrary. Comments or any type of user identification can be included in the same field following the parameter value if there is a space or comma separating the two entries as described. Any number of records can separate the field descriptors. These records can contain descriptive text or be used to separate existing field descriptors.

Several parameters have default values that correspond to available options. Appendix E of the *HBR-LET EC&M Software User's Guide* lists the available option values for each parameter.

To ease the format restrictions of the DEFAULT.LET user data file, a second file, M1:LABELS.LET/111 was developed. LABELS.LET contains the instrument names and parameter labels associated with each type of instrument defined in the DEFAULT.LET data file.

Figure 5-2 illustrates a section of the LABELS.LET data file. Each available instruments is listed first preceded by the number of characters in its name. The parameters that correspond to the instrument are listed in order, preceded by the number of characters in the parameter name. The number of characters in the instrument and parameter names determine where to start looking for parameter values within the DEFAULT.LET data file. The number of parameters is required for the software to locate all parameters for a single instrument in the event the user inputs comment statements between the instrument field descriptors.

```
10                                  <-10 instruments in the file
19 HP POWER METER 437B              <-Inst num 1, 19 char in name
13 GM ATTENUATOR                    <-Inst num 2, 13 char in name
27 HP ATTENUATOR/SWITCH DRIVER          .
24 WAVETEK PEAK POWER METER             .
18 SA BEACON RECEIVER                   .
21 EIP FREQUENCY COUNTER
17 HP RELAY ACTUATOR
18 HP DC POWER SUPPLY
14 DATA GENERATOR
16 HP SWITCH DRIVER                 <-Inst num 10, 16 char in name
 9                                  <-9 parameters for inst num 1
 5 RANGE                            <-Parm num 1 for inst num 1
12 TRIGGER MODE                     <-Parm num 2 for inst num 1
18 CALIBRATION FACTOR                   .
 9 LOW LIMIT                            .
10 HIGH LIMIT                           .
 6 OFFSET
 5 UNITS
17 INITIAL FREQUENCY
19 OUT OF LIMIT ACTION              <-Parm num 9 for inst num 1
 5                                  <-5 parameters for inst num 2
 9 LOW LIMIT                        <-Parm num 1 for inst num 2
10 HIGH LIMIT                       <-Parm num 2 for inst num 2
 :   :
```

Figure 5-2 LABELS.LET Default File

In Figure 5-2, the first 10 indicates that there are 10 types of instruments contained in the data file. The instruments and the number of characters in their name follows. Next, the corresponding parameters are listed. The number 9 (by itself) indicates that the HP POWER METER 437B has the 9 following parameters. Each parameter lists its name and number of characters in the name. This format is repeated for every instrument. The instrument default parameter data file was developed to avoid changing existing software during development of the LET. Instrument parameters can be added and modified without recompiling existing software. LABELS.LET is the responsibility of the software developer due to its strict format restrictions and effect on the EC&M Software. If the user requires modification to this file, submit a CPMPR form to the C&PM Software manager stating the nature of the change.

6.0 ABBREVIATIONS AND ACRONYMS

A list of the abbreviations and acronyms and their definition is provided for user reference.

ABBREVIATION	DEFINITION
ACTS	Advanced Communications Technology Satellite
ASCII	American Standard Code for Information Interchange
BER	bit error rate
C&PM	Control and Performance Monitor Software Account
CAL	Calibration
CPMSA	Control and Performance Monitor Software Account.
DG/DC	data generator and data checker
DGT	digital ground terminal
EC&M	Experiment Control and Monitor
GM	General Microwave
GMAT	General Microwave Attenuator
GPIB	General Purpose Interface Bus
HBR-LET	High Burst Rate Link Evaluation Terminal
HP	Hewlett Packard
HPA	High Power Amplifier
HPPM	Hewlett Packard Power Meter
HPSD	Hewlett Packard Attenuator/Switch Driver (Used solely as a switch driver)
IDS	Instrument Definition Software

Section 6 - Abbreviations and Acronyms

IEEE	Institute of Electrical and Electronic Engineers
LB	Loopback
LeRC	Lewis Research Center
Mbps	Megabits per second
MTM	Multi-Terminal Monitor
NASA	National Aeronautics and Space Administration
PC	personal computer
RF	radio frequency
SA	Scientific Atlantic
SDA	Sequence Definition Array
SDS	Sequence Definition Software
SES	Sequence Execution Software
SMAP	Software Management and Assurance Program
TDF	Test (Instrument) Definition File
UC	Upconverter
WTPM	Wavetek Peak Power Meter

7.0 GLOSSARY

bit error rate - the number of bit errors divided by the total number of bits received. Bit errors are determined by the data checker after comparing the received bit stream with that produced by the data generator.

data generator - produces a pseudorandom data source which simulates a transmitting user.

data checker - outputs the number of bits received and the number of bit errors.

digital ground terminal - transmits and receives bursted 110.592 and 221.184 Mbps serial minimum shift keying signals..

Link Evaluation Terminal - ground station that provides a LeRC-ACTS-LeRC communication link with transmission rates of 110.592 and 221.184 Mbps. The LET will be used to demonstrate adaptive uplink power control through various experiments.

8.0 APPENDICES

Appendix A

Instrument Table Definitions and Placement

The Instrument Table contains the following entries; NASA tag number, address or CRT number, interface type, IEEE 488 interface bus number, type number, label, id, and an instrument description.

NASA TAG: NASA tag number affixed to the instrument. Entry must be six characters or less. (Numerical and alphanumeric characters are permitted).

ADDR/CRT: IEEE 488 address. For instruments using an RS-232 interface, this field contains the communication port number. There cannot exist two instruments on the same bus extender, same bus type, with the same address.

BUS: Interface type. Valid inputs are "1" for IEEE 488 or "2" for RS-232.

EXT: IEEE 488 interface bus extender number. Valid bus numbers are "0" or "1". Entry is limited to one character. Currently, only bus "0" is available on the computer system.

LABEL: The label is a three character name of the instrument. The label is taken from the suitable subsystem hardware documentation.

ID: A four-character user-defined name of the instrument. The ID is used to remind the user of the instrument name in the instrument table.

DESCRIPTION: The name of the instrument. This entry can be up to 30 characters in length.

TYPE: Type of instrument as defined in Table A-1. New type numbers can be added as necessary, provided no two different instruments share the same number. Type number can consist of up to four characters.

Type	Instrument
1	HP Power Meter 436A
2	GM Attenuator Controller
3	HP 11713A Atten/Switch Driver
4	Wavetek Peak Power Meter
5	SA 930 Beacon Receiver
6	HP 59306A Relay Actuator
7	EIP 545B Frequency Counter
8	HP 6632A DC Power Supply
9	HP Switch Driver
10	Data Generator
11	Data Checker
19	HP Power Meter 437B
21	Digital Ground Terminal

Table A-1 Instrument Type Definitions

Each entry in the table has been given an assigned location within the file for formatting purposes. Figure A-1 depicts a portion of the Instrument Table used at the time of this publication. At the top of the table is the column number where each entry begins and ends. (Leading zeros are shown as place holders only.) Column numbers are shown for this reference only and are not displayed in the actual data file.

The first entry of the data file contains the number of instruments currently in the file and is limited to four characters. The remainder of the first line contains the name of the file and a short file description. Record two contains column headers of each entry. Instrument descriptions must begin in the third row.

```
         1         2         3         4         5         6         8
12345678901234567890123456789012345678901234567890123456789012345...0

0005  M1:INSTRMTS.LET/111      Instruments available to the HBR-LET
NASA TAG ADDR BUS EXT TYPE  LABEL ID       DESCRIPTION
G88833    0001   1   0   09  B37   HPSD     HP Switch Driver
G50340    0002   1   0   19  N20   HPPM     HP Power Meter 437B
071649    0003   1   0    2   A1   GMAT     GM Attenuator
183198      10   2   0    2   B2   GMAT     GM Attenuator
183196      21   2   0    2
```

Figure A-1 Instrument File Definition

The GM Attenuator controllers are controlled using both an address and a channel. A method of encoding the address and channel has been developed. Use the address listed in Table A-2 that corresponds to the correct attenuator address and channel.

ADDR	Channel	Address
10	A	0
11	A	1
20	B	0
21	B	1

Table A-2 GM Attenuator Address Codes

Appendix B

Experiment Control and Monitor Software Instrumentation

Each instrument presently defined in the HBR-LET is listed in Table B-1. The address associated with each instrument has been predefined, but can be changed if desired. The devices that are connected to the Concurrent 3205 via RS-232 links have a dedicated communication port within the EC&M Software System. Abbreviations for the subsystem names used within the table are listed in Section 6.

Table B-1 Instrument Identification

Instrument	Tag Number	Name	Subsystem	Addr
WAVETEK 8502 PEAK POWER METER	G88833	PM1	HPA	01
HP 11713A SWITCH DRIVER	G50340	B37	HPA	02
HP 437B POWER METER	071649	N20	Receiver	03
HP 437B POWER METER	071646	L16	Calibration(CAL)	04
HP 11713A ATTEN/SWITCH DRIVER	068280	N6	Receiver	05
HP 11713A ATTEN/SWITCH DRIVER	068281	N12	Receiver	06
HP 11713A ATTEN/SWITCH DRIVER	068275	R12	Receiver	07
WAVETEK 8502 PEAK POWER METER	TBD	PM2	HPA	08
HP 6632A POWER SUPPLY	071580	PIN	HPA	09
HP 59306A RELAY ACTUATOR	071535	C7	Calibration	10
HP 437B POWER METER	003612	C26	Calibration	11
HP 437B POWER METER	G88647	C27	Calibration	12
HP 437B POWER METER	071643	U21	Calibration	13
HP 437B POWER METER	071645	R11	Loopback (LB)	14
HP 11713A ATTEN/SWITCH DRIVER	068276	U13	Loopback	15

Table B-1 Instrument Identification				
Instrument	Tag Number	Name	Subsystem	Addr
HP 11713A ATTEN/SWITCH DRIVER	068274	L6	Loopback	16
HP 437B POWER METER	071647	U16	Loopback	17
EIP 545B FREQUENCY COUNTER	071557	C9	Receiver/LB/CAL	18
EIP 545B FREQUENCY COUNTER	071556	R9	Receiver/LB/CAL	19
HP 11713A ATTEN/SWITCH DRIVER	068277	TBD	Monitor	20
HP 11713A ATTEN/SWITCH DRIVER	068278	U4	Monitor	21
HP 11713A ATTEN/SWITCH DRIVER	068279	U12	Monitor	22
SA 930 BEACON RECEIVER	G90028	SA1	Receiver	23
GM ATTENUATOR CONTROLLERS	183198	GA0	Loopback/ Upconverter	A,0
GM ATTENUATOR CONTROLLERS	183192	GA1	Loopback/ Upconverter	A,1
GM ATTENUATOR CONTROLLERS	183191	GB0	Loopback/ Upconverter	B,0
GM ATTENUATOR CONTROLLERS	183196	GB1	Loopback/ Upconverter	B,1
DATA GENERATOR	277172	DG	BER	CRT3
DATA CHECKER	277170	DC	BER	CRT3
DIGITAL GROUND TERMINAL	704905	DGT	DGT/BER	CRT4

Table B-1 Instrument Identification

Appendix C

Experiment Control and Monitor Software Program Descriptions

Source code for each program in the Experiment Control and Monitor Software System can be found in the CPMSA. The summaries provide an overview of the major functions and use of each program. The task commons and libraries referenced by each program are documented in Appendix C and Appendix D, respectively.

Appendix C - Program Descriptions

Program Name M1:CTRLLET.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December, 1990
Original Sequence Execution control program written for SITE by Ike Kramarchuk and Elaine Daugherty / NASA, Digital Technology Branch, 1984

Description This program executes the sequence commands defined in the Sequence Definition Software. Each sequence command has one or more subroutines associated with it to execute the command. The modular software design enables additional commands to be added without changing existing code.

Available commands include:

Set a Parameter	Step a Parameter
Start Data Generator	Stop Data Generator
Stop Data Checker	Wait
DGT Command	Goto
Execute Sub-Sequence	Set up a Loop
Perform BER Measurement	Check a Parameter
Set Data Generator Errors	Zero Power Meter

Task Common M1:INSTRCOM.IMG/111

Contains both the Instrument Definition and Sequence Definition arrays used throughout the Instrument Definition, Sequence Definition and Sequence Execution programs. Also includes the variable STORE used for writing the header to the data file created by the Sequence Execution Software.

M1:DSPCTRL.COM/111

Contains information about the logical units used by both the Display and Sequence Execution Programs. Other variables contained within this task common include logical variables used as flags for the Data Generator, executing Sub-Sequences, and performing BER measurements.

The logical unit assignments used in the program CTRLLET are listed in Table C-1.

Logical Unit	Assignment	Form	Access	Record Length
9	DGT Command File	Formatted	Sequential	80
10	GM Attenuator	Formatted	Sequential	80
11	M1:SEQ.LOG/111	Formatted	Sequential	80
12	DGT	Formatted	Sequential	80
13	DAT filename	Binary	Direct	512
14	M1:SEQ.LOG/111	Formatted	Sequential	80
16	DG/DC	Formatted	Sequential	80

Table C-1 Sequence Execution Control Logical Unit Assignments

Table C-2 lists the procedures called by the program CTRLLET. A short description of each procedure and the library where it resides is provided.

Procedure	Description	Library
ENABLE	establishes the necessary data structures to allow a specified task trap to be handled by a user written trap handling FORTRAN subroutine	RTL
ERR	logs command execution errors to a file and clears the IEEE interface bus	CTRLLET
EXECL	determines which sequence command to execute, calls the appropriate subroutines, and calls the data acquisition subroutine before each command is executed	EXECLET
EXIT	allows the calling task to terminate its own execution	RTL
ICLEAR	clears the IEEE interface bus	GENERLIB

Procedure	Description	Library
INIT	initializes a program's real-time data structures prior to enabling any task trap	RTL
SNDMSG	allows a foreground task to send a message to another foreground task	RTL
SUBSEQ	executes commands within a sub-sequence in a manner similar to the main control program	CTRLLIB2
SUB3	a task trap occurs whenever a message is sent from a task via a SNDMSG call, causing a user written trap handling subroutine to receive the message	RTL
SYSIO	performs I/O functions on a level comparable to assembly	RTL
WAIT	suspends task execution for a specified amount of time	RTL

Table C-2 Sequence Execution Control Program Procedures

Libraries:

M1:ATSETM2.OBJ/111
M1:BERDGT.OBJ/111
M1:CTRLLIB1.OBJ/111
M1:CTRLLIB2.OBJ/111
M1:DATALET.OBJ/111
M1:DECDATA.OBJ/111
M1:DGDCLIB.OBJ/111
M1:DGTLIB.OBJ/111
M1:DSPLIB1.OBJ/111
M1:DSPLIB2.OBJ/111
M1:EBNZERO.OBJ/111
M1:EXECLET.OBJ/111
M1:GENERLIB.OBJ/111
M1:GMLIB.OBJ/111
M1:GRTL15.OBJ/111
M1:INSETUP.OBJ/111
M1:PIN.OBJ/111
M1:TERMEB.OBJ/111
M1:TDFLIB.OBJ/111
M1:SA1SEC.OBJ/111

RTL - Concurrent FORTRAN VII Run-Time Library

Appendix C - Program Descriptions

Program Name	M1:DSPLET.FTN/111
Author	Richard C. Reinhart / Analex Corporation
History	Written December, 1990 Original TDF program written for SITE by Ike Kramarchuk and Elaine Daugherty, 1984 / NASA, Digital Technology Branch

Description — This program displays the current status or output reading of all instruments defined for a particular experiment. In addition, it loads and starts the control program of the Sequence Execution Software. All warning and error messages accrued during sequence execution are displayed to the user terminal.

Task Common — M1:INSTRCOM.IMG/111

Contains both the Instrument Definition and Sequence Definition arrays used throughout the Instrument Definition, Sequence Definition and Sequence Execution programs. Also includes the variable STORE used for writing the header to the data file created by the Sequence Execution Software.

The logical unit assignments used in the program DSPLET are listed in Table C-3.

Logical Unit	Assignment	Form	Access	Record Length
1	TDF filename	Binary	Direct	256
7	M1:CTRL.TSK/111	N/A	N/A	N/A
8	M1:CTRL.LOG/111	Formatted	Sequential	80
9	DAT filename	Binary	Direct	512
10	GM Attenuator	Formatted	Sequential	80
12	DGT	Formatted	Sequential	80
13	DAT filename	Binary	Direct	512
13	SDA filename	Binary	Direct	512
14	CRTX:	Menu	N/A	80
16	DG/DC	Formatted	Sequential	80
17	DSP filename	Formatted	Index	96

Table C-3 Sequence Execution Display Logical Unit Assignments

Note: CRTX: refers to the communication port dedicated to menu display. M1:DSPLET.MNU/111 is the DSPLET menu software.

Table C-4 lists the procedures called by the program DSPLET. A short description of each procedure and the library where it resides is provided.

Procedure	Description	Library
ABORT	allows a FORTRAN program to terminate the execution of a foreground task	RTL
ALLINIT	initializes all the instruments defined for a particular experiment	DSPLIB2

Procedure	Description	Library
ALLLOCAL	places all the instruments defined for an experiment in local mode, returning control of the instrument to the front panel keys	DSPLIB3
ALLREMOTE	places all the instruments defined for an experiment in remote mode, placing the instrument under computer control	DSPLIB1
ASSEMB	allows the DSPLET program to initialize variables from a CSS file. Used to initialize the DEBUG flags.	ASSEMB
BITDECODE	decodes the variable BIT_POS to determine if an out of limit condition exist	DSPLIB3
DISPLAY	outputs the sequence definition array to either the user terminal or the line printer at the user's discretion	DSPLIB3
ENABLE	establishes the necessary data structures to allow a specified task trap to be handled by a user written trap handling FORTRAN subroutine	RTL
EXIT	allows the calling task to terminate its own execution	RTL
HOLD	suspends execution of a task until released	RTL
ILBYTE	loads a single byte from one argument to another	RTL
INIT	initializes a program's real-time data structures prior to enabling any task trap	RTL
INQUIRE	inquires as to whether or not a file exists	RTL
IOERR	outputs error messages to the system console when an I/O error occurs as a result of a SYSIO	RTL

Procedure	Description	Library
ISBYTE	stores a single byte from one argument to another	RTL
LOAD	allows a program to load a second program into memory	RTL
MENUF	menu driver which permits user I/O from any user terminal defined in row 1, col 31 of the menu software	MENUP2
OPENW	allows a calling program to assign one of its logical units to a file or device	RTL
PCUR	positions the curser to a specified location on the user terminal	MENUP2
RELSE	releases a suspended task	RTL
SENDLU	causes the device or file assigned to the specified lu of the calling task to be reassigned to the specified lu of the directed task. The lu of the sending task is closed	RTL
SNDMSG	allows a foreground task to send a message to another foreground task	RTL
START	allows a program to start another program that has been loaded into memory	RTL
SUB3	a task trap occurs whenever a message is sent from a task via a SNDMSG call, causing a user written trap handling subroutine to receive the message	RTL
SUB9	a task trap is enabled whenever a subtask has a state change, causing a user written trap handling subroutine to be executed	RTL
SYSIO	performs I/O functions on a level comparable to assembly	RTL

Procedure	Description	Library
TERMEB	performs terminal I/O functions such as clear terminal, beep, lock and unlock keyboard, etc	TERMEB
WAIT	suspends task execution for a specified amount of time	RTL

Table C-4 Sequence Execution Display Program Procedures

Libraries:
M1:ASSEMB.OBJ/111
M1:ATSETM2.OBJ/111
M1:BERDGT.OBJ/111
M1:CTRLLIB1.OBJ/111
M1:CTRLLIB2.OBJ/111
M1:DGDCLIB.OBJ/111
M1:DGTLIB.OBJ/111
M1:DSPLIB1.OBJ/111
M1:DSPLIB2.OBJ/111
M1:DSPLIB3.OBJ/111
M1:GENERLIB.OBJ/111
M1:GMLIB.OBJ/111
M1:GRTL15.OBJ/111
M1:INSETUP.OBJ/111
M1:MENUP2.OBJ/S
M1:TDFLIB.OBJ/111
M1:TERMEB.OBJ/111

RTL - Concurrent FORTRAN VII Run-Time Library

Program Name M1:SDALET.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December, 1990
Original SDA program written for SITE by Ike Kramarchuk and Elaine Daugherty, 1984 / NASA, Digital Technology Branch

Description This program enables a user to define the sequence of commands that the Sequence Execution Software will use to carry out a computer controlled experiment. Sub-sequences as well as Main Sequences can be created using this software. The LET-EC&M Software User's Guide provides individual descriptions of each sequence command. The purpose of this program is to build the sequence definition file which is utilized by the Sequence Execution Software. The instrument definition file must be created before executing the sequence definition software.

Task Common M1:INSTRCOM.IMG/111

Contains both the Instrument Definition and Sequence Definition arrays used throughout the Instrument Definition, Sequence Definition and Sequence Execution programs. Also includes the variable STORE used for writing the header to the data file created by the Sequence Execution Software.

The logical unit assignments used in the program SDALET are listed in Table C-5.

Logical Unit	Assignment	Form	Access	Record Length
1	TDF filename	Binary	Direct	256
6	SDA filename	Binary	Direct	256
10	INSTRMTS.LET	Formatted	Direct	80
11	System Console	Formatted	Sequential	80
14	CRTX:	Menu	N/A	80

Table C-5 Sequence Definition Logical Unit Assignments

Note: CRTX: refers to the communication port dedicated to menu display. M1:SDALET.MNU/111 is the SDALET menu software.

Table C-6 lists the procedures called by the SDALET program. A short description of the procedure is given in addition to the library where it resides.

Procedure	Description	Library
BERMEASURE	defines the number of BER measurements at a particular Eb/No	SDALIB3
CALLSUB	defines the name of the sub-sequence to execute during sequence execution	SDALIB3
CHECKPARM	specifies the instrument to check, high and low limits, and label numbers to goto if reading is out of limit	SDALIB3
DISPLAY	outputs the sequence definition array to either the user terminal or the line printer at the user's discretion	SDALET

Procedure	Description	Library
ENDSEQ	verifies that all Check a Parameter and GOTO sequence command argument label numbers exist. Issues error messages when required label numbers are not defined	SDALET
EXIT	allows a calling task to terminate its own execution	RTL
GOTOST	specifies the target label number within a GOTO sequence command	SDALIB3
LABELIT	converts a user input label number to a sequence control number for the sequence execution software	SDALIB3
MENUF	menu driver which permits user I/O from any user terminal defined in row 1, col 31 of menu software	MENUP
PMZERO	specifies the power meter to receive the zero function during sequence execution	SDALIB2
SDADGTCOMM	defines the name of the DGT file to execute during sequence execution	SDALIB4
SETERROR	allows user to specify the number of errors to put in data stream output by the Data Generator	SDALIB3
SETLOOP	sets up a loop within a main or sub sequence	SDALIB3
SETPARM	sets a specified parameter to a user defined value. Instrument label and parameter value must be input	SDALIB2
STARTDG	command to start the Data Generator	SDALIB2
STEPPARM	steps a specified parameter by a user defined amount. Instrument label and step parameter value must be input	SDALIB2

Procedure	Description	Library
STOPDC	command to stop the Data Checker	SDALIB2
STOPDG	command to stop the Data Generator	SDALIB2
TERMEB	performs terminal I/O functions such as clear terminal, beep, lock and unlock keyboard, etc	TERMEB
WAIT	suspends task execution for a specified amount of time	RTL
WAITT	suspends sequence execution for a specified amount of time	SDALIB2

Table C-6 Sequence Definition Program Procedures

Libraries:
M1:BERLIB2.OBJ/111
M1:ENCDATA.OBJ/111
M1:GENERLIB.OBJ/111
M1:MENUP.OBJ/S
M1:TERMEB.OBJ/S
M1:TDFLIB.OBJ/111
M1:TDFLIB2.OBJ/111
M1:SDALIB2.OBJ/111
M1:SDALIB3.OBJ/111
M1:SDALIB4.OBJ/111
M1:SDALIB5.OBJ/111

RTL - Concurrent FORTRAN VII Run-Time Library

Program Name M1:TDFLET.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December, 1990

Description This program allows a user to define the instruments to be used in a sequence. Each instrument is selected from an Instrument Table and has required inputs which are defined in the LET-EC&M Software User's Guide. The purpose of this program is to build the instrument definition file which is utilized by both the Sequence Definition and Sequence Execution software.

Task Common M1:INSTRCOM.IMG/111

Contains both the Instrument Definition and Sequence Definition arrays used throughout the Instrument Definition, Sequence Definition and Sequence Execution programs. Also includes the variable STORE used for writing the header to the data file created by the Sequence Execution Software.

The logical unit assignments used in the program TDFLET are listed in Table C-7.

Logical Unit	Assignment	Form	Access	Record Length
1	TDF filename	Binary	Direct	256
8	LABELS.LET	Formatted	Direct	80
8	DEFAULT.LET	Formatted	Direct	80
10	INSTRMTS.LET	Formatted	Direct	80
11	System Console	Formatted	Sequential	80
14	CRTX:	Menu	N/A	80

Table C-7 Instrument Definition Logical Unit Assignments

Note: CRTX: refers to the communication port dedicated to menu display. M1:TDFLET.MNU/111 is the TDFLET menu software.

Table C-8 lists the procedures called by the program TDFLET. A short description of each procedure and the library where it resides is provided.

Procedure	Description	Library
ADDR21	initializes the switch positions of instrument U4 using the desired path from input to output	TDFLIB3
ADDR22	initializes the switch positions of instrument U12 using the desired path from input to output	TDFLIB3
CLEARM	used to reset the menu variables	GENERLIB
DECDG	decodes the DG/DC menu from the user terminal into the proper instrument array positions	TDFLIB2
DECEIPF	decodes the EIP Frequency Counter menu from the user terminal into the proper instrument array positions	TDFLIB2
DECGMAT	decodes the GM Attenuator menu from the user terminal into the proper instrument array positions	TDFLIB2
DECHPAT	decodes the HP Attenuator/Switch Driver menu from the user terminal into the proper instrument array positions	TDFLIB2
DECHPPM	decodes the HP Power Meter menu from the user terminal into the proper instrument array positions	TDFLIB2
DECHPPS	decodes the HP Power Supply menu from the user terminal into the proper instrument array positions	TDFLIB2
DECHPRA	decodes the HP Relay Actuator menu from the user terminal into the proper instrument array positions	TDFLIB2

Procedure	Description	Library
DECSABR	decodes the SA Beacon Receiver menu from the user terminal into the proper instrument array positions	TDFLIB2
DECHPSD	decodes the HP Switch Driver menu from the user terminal into the proper instrument array positions	TDFLIB2
DECWTPM	decodes the Wavetek Peak Power Meter menu from the user terminal into the proper instrument array positions	TDFLIB2
EXIT	allows the calling task to terminate its own execution	RTL
INSETUP	reads in the default values of the instrument parameters from an external ASCII file	TDFLIB
MENUF	menu driver which permits user I/O from any user terminal defined in row 1, col 31 of the menu software	MENUP
OPTION21	sets up the option field for instrument U4. Options contain the available inputs and outputs respectively for that instrument	TDFLIB3
OPTION22	sets up the option field for instrument U12. Options contain the available inputs and outputs respectively for that instrument	TDFLIB3
TAGCHEK	verifies that no two instruments on the same bus extender, and bus extender type have the same address	TDFLIB
TDFDG	initializes and displays the Data Generator/Data Checker (DG/DC) menu to the user terminal	TDFLIB
TDFEIPF	initializes and displays the EIP Frequency Counter menu to the user terminal	TDFLIB

Procedure	Description	Library
TDFGMAT	initializes and displays the General Microwave Attenuator menu to the user terminal	TDFLIB
TDFHPAT	initializes and displays the HP Attenuator/Switch Driver menu to the user terminal	TDFLIB
TDFHPPM	initializes and displays the HP Power Meter menu to the user terminal	TDFLIB
TDFHPPS	initializes and displays the HP Power Supply menu to the user terminal	TDFLIB
TDFHPRA	initializes and displays the HP Relay Actuator menu to the user terminal	TDFLIB
TDFHPSD	initializes and displays the HP Switch Driver menu to the user terminal	TDFLIB
TDFSABR	initializes and displays the Scientific Atlantic (SA) Beacon Receiver menu to the user terminal	TDFLIB
TDFWTPM	initializes and displays the Wavetek Peak Power Meter menu to the user terminal	TDFLIB
TERMEB	performs terminal I/O functions such as clear terminal, beep, lock and unlock keyboard, etc	TERMEB
WAIT	suspends task execution for a specified amount of time	RTL

Table C-8 Instrument Definition Program Procedures

Libraries:

```
M1:BERLIB2.OBJ/111
M1:GENERLIB.OBJ/111
M1:INSETUP.OBJ/111
M1:TDFLIB.OBJ/111
M1:TDFLIB2.OBJ/111
M1:TDFLIB3.OBJ/111
M1:TERMEB.OBJ/111
M1:MENUP.OBJ/S
```

RTL - Concurrent FORTRAN VII Run-Time Library

Appendix D

Experiment Control and Monitor Software Library Descriptions

Each library used in the EC&M Software System is located in the CPMSA. The library outlines provide a short description of the variables passed by the subroutines and an explanation of the functions performed by each. The programs that use the respective library are listed on the bottom of each outline. Appendix C contains a summary of the EC&M Software programs.

Library Name M1:ATSETM2.FTN/111
Author Brian A. Kachmar / Analex Corporation
History Written October, 1990
Revised November, 1990 <RCR>
Original ATSETM written by Elaine Daugherty NASA / Digital Technology Branch

Variables used by the subroutines:

ADDRESS an INTEGER*4 input variable containing the instrument address.

ATTN a REAL*4 input variable containing the desired value of attenuation. $0 \leq ATTN \leq 81$

IBUS an INTEGER*4 input variable containing the IEEE 488 interface bus number.

ITEMP an INTEGER*4 input variable temporarily used until further modifications are made to each ATSET call statement. The variable is not used within the subroutine and a value of zero is appropriate.

STATUS an INTEGER*4 output variable used to return the execution status of the subroutine or the status of the system.

SW9 an INTEGER*4 input variable depicting the position of switch 9.
0 off
1 on

SW0 an INTEGER*4 input variable depicting the position of switch 0.
0 off
1 on

SUBROUTINE ATSET (ITEMP,IBUS,ADDRESS,ATTN,SW9,SW0,STATUS)
Sets the desired attenuation value and proper switch positions on the HP 11713A Step Attenuators. Maximum attenuation is 81 dB. Switch positions are represented using the convention; 1 On, 0 Off.

Appendix D - EC&M Software Libraries

Programs:

Experiment Control and Monitor Software

M1:CTRLLET.FTN/111

Check-Out Software

M1:BER.FTN/111
M1:L6.FTN/111
M1:N6.FTN/111
M1:N12.FTN/111
M1:MAINCTRL.FTN/111
M1:R12.FTN/111
M1:U13.FTN/111

Library Name M1:BERDGT.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written November, 1989
Revised September 21, 1990 Version 1.0 <RCR>

Variables used by the subroutines:

BURSTED an INTEGER*4 input variable which contains a value of '1' if the data rate is not a multiple of the desired clock rate.

CONTINUOUS an INTEGER*4 input variable which contains a value of '1' if the data rate is an even multiple of the desired clock rate.

RATE an INTEGER*4 input variable containing the value of the desired data rate.

R110 a LOGICAL input/output variable depicting the current bursting rate of the Digital Ground Terminal.
TRUE - DGT is set for 110 Mbps communication.
FALSE - DGT is set for other than 110 Mbps comm.

R220 a LOGICAL input/output variable depicting the current bursting rate of the Digital Ground Terminal.
TRUE - DGT is set for 220 Mbps communication.
FALSE - DGT is set for other than 220 Mbps comm.

STATUS an INTEGER*4 output variable used to return the execution status of the subroutine.

STATUS1 see STATUS.

SUBROUTINE NOTRAKACQ(STATUS1)
Sends the command 'DISABLE BVERF SFVERF TRKINT' to the DGT.

SUBROUTINE TRAKACQ(STATUS1)
Sends the command 'ENABLE BVERF SFVERF TRKINT' to the DGT.

SUBROUTINE RESETMUX(STATUS1)
Sends the 'START RESET SYNC' and 'RXSYNC PCA' commands to the DGT.

SUBROUTINE NEWACQ (STATUS)
Acquires the status of the DGT.

SUBROUTINE DGTCLOCK(BURSTED,CONTINUOUS,STATUS)
Sets the appropriate clock rate of the DGT depending on the current burst rate.

SUBROUTINE MODEMRATE(RATE,R220,R110,STATUS)
Sets the user desired bursting rate of the DGT. Valid options are 220 Mbps and 110 Mbps.

Programs:

Experiment Control and Monitor Software
M1:DSPLET.FTN/111
M1:CTRLLET.FTN/111

Check-Out Software
M1:BER.FTN/111

Library Name M1:BERLIB2.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Revised September 21, 1990 Version 1.0 <RCR>

Variables used by the subroutines:

IRATE an INTEGER*4 input(SETUPBER)/output(TYPERATE) variable depicting the data rate. Data rates and data types are listed in Table D-1.

Data Rate Mbps	IRATE	ITYPE
1.25	0	0
5.00	3	0
12.50	5	0
13.824	5	8
25.00	8	0
27.648(88)	8	8
27.648(09)	9	0
27.648(89)	9	8
50.00	12	0
55.296	12	8
100.00	13	0
110.592	13	8
200.00	15	0
221.184	15	8

Table D-1 Data Rate and Type Definitions in the BERLIB2 Library

ITYPE — an INTEGER*4 input(SETUPBER)/output(TYPERATE) variable depicting the data type. Data rates and data types are listed in Table D-1.
- 0 Represents a bursted mode of operation. Data rate is not a multiple of the clock rate.
- 8 Represents continuous mode of operation. Data rate is an even multiple of the clock rate.

IN — a CHARACTER*24 input(TYPERATE)/output(SETUPBER) array containing the input fields for the current menu.

SUBROUTINE TYPERATE(IRATE,ITYPE,IN)
Converts the user selected menu inputs into the data type and data rate variables ITYPE and IRATE respectively.

SUBROUTINE SETUPBER(IRATE,ITYPE,IN)
Converts the variables ITYPE and IRATE into data type and data rate menu inputs for display to the user terminal.

Programs:

Experiment Control and Monitor Software
M1:TDFLET.FTN/111

Check-Out Software
M1:BER.FTN/111

Library Name M1:CTRLLIB1.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990

Variables used by the subroutines:

ADDR — an INTEGER*4 input variable containing the instrument address.

ADDRESS — see ADDR

COUNT1 — an INTEGER*4 output variable whose value indicates the Master Division Alarm bit in error.

CURRENT — a REAL*4 output variable containing the output current of the HP DC Power Supply.

DBUG — an INTEGER*4 input variable once used to output debug statements during the software development. This variable is not used within the subroutine and a value of zero is appropriate.

FREQ — a REAL*4 output variable containing the frequency read by the EIP Frequency Counter.

IBUS — an INTEGER*4 input variable containing the IEEE 488 interface bus number.

ID — a CHARACTER*5 input variable containing the name of the function sending a message to a particular instrument.

IUNIT — an INTEGER*4 output variable depicting the measurement units of the power measured by the Wavetek Peak Power Meter.
1 Watts
2 dBm

KBUS — see IBUS

LENGTH — an INTEGER*4 input variable containing the number of bytes contained in the message string, sent to an instrument.

MSG — an INTEGER*4 input/output array used as a message buffer containing the ASCII command sent to an instrument or device. Also contains the instrument response when one is transmitted.

OPTION an INTEGER*4 input variable representing the channel of the Wavetek Peak Power Meter to read.

RMSG a REAL*4 input array containing the instrument parameter values to send to an instrument.

SINGLE a REAL*4 output variable containing a single reading of the output power of the Wavetek Peak Power Meter.

STATUS an INTEGER*4 output variable used to return the execution status of the subroutine.

STBIT an INTEGER*4 output variable containing the error code returned by the Wavetek Peak Power Meter, if an error is present. Consult the Wavetek User's manual for a detailed description of the error codes.

STREN a REAL*4 output variable containing the signal strength of the SA Beacon Receiver.

TEST a INTEGER*4 input array containing the Master Division Alarm Status bits returned from the SA Beacon Receiver.

TRIG an INTEGER*4 input variable representing the trigger mode of the HP Power Meter.

VOLTS a REAL*4 output variable containing the output voltage of the HP DC Power Supply.

SUBROUTINE FCREAD(DBUG,IBUS,ADDRESS,FREQ,STATUS)
Similar to FCREAD2, with the addition of the hold active command. This routine was developed when errors occurred with the IEEE 488 interface bus.

SUBROUTINE FCREAD2(DBUG,IBUS,ADDRESS,FREQ,STATUS)
Author: Elaine Daugherty
Obtains the current frequency reading displayed on the EIP Frequency Counter. This subroutine is no longer used by the EC&M Software.

SUBROUTINE HPREAD(DBUG,IBUS,ADDR,STATUS,TRIG,RMSG)
Obtains the current output power of the HP 437B Power Meter.

SUBROUTINE WTREAD(ADDR,IBUS,DBUG,OPTION,STATUS,SINGLE,
STBIT,IUNIT)
Obtains the current output power of the Wavetek Peak Power Meter. The OPTION variable corresponding to the channel and mode read are listed in Table D-2.

OPTION	Channel	Mode
2	A	CW
4	B	CW
6	A	PEAK
8	B	PEAK

Table D-2 Wavetek Peak Power Meter Channel/Mode Definition

SUBROUTINE PSREAD(ADDR,IBUS,DBUG,STATUS,VOLTS,CURRENT)
Obtains the output voltage and current from the HP 6632A DC Power Supply.

SUBROUTINE ERRCHECK(ADDR,LENGTH,DBUG,KBUS,STATUS,RMSG)
Queries the status bit of the HP Power Meter and decodes the error returned if one occurs. Called from within HPREAD subroutine.

SUBROUTINE SASTATUS(DBUG,IBUS,ADDR,STATUS)
Reads the status of the MASTER DIVISION ALARMS of the SA Beacon Receiver. Correct alarm bit setting is user defined. Settings in place at time of this publication are listed in Table D-3.

Bit Number	Correct Value	Description
0	0	No hardware fault present in Tuner lock alarm
1	0	Demod slot 1 locked
2	1	Demod slot 2 not locked
3	0	Carrier above threshold
4	0	No hardware fault present in Frequency Synthesizer
5	1	Internal reference
6	0	No hardware fault present in 2nd LO Lock Alarm
7	1	Internal LO

Table D-3 Master Division Alarm Status Codes

SUBROUTINE SACHECK(DBUG,IBUS,STATUS,TEST,COUNT1)
Compares the MASTER DIVISION ALARM status returned from the SA Beacon Receiver with the user defined configuration and returns the appropriate status to the SASTATUS subroutine.

SUBROUTINE GETPOWER(DBUG,IBUS,ADDR,STATUS,STREN)
Obtains the beacon signal strength from the SA Beacon Receiver.

SUBROUTINE SASEND(ADDR,LENGTH,DBUG,KBUS,STATUS,MSG,ID)
Sends and receives ASCII messages to/from the SA Beacon Receiver.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111

Library Name M1:CTRLLIB2.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990

Variables used by the subroutines:

ADDR an INTEGER*4 input variable containing the instrument address.

CHANNEL an INTEGER*4 input variable representing the desired channel for the Wavetek Peak Power Meter command. Table D-4 lists the available modes and channels available on the Wavetek Peak Power Meter.

Option Number	Channel	Mode
1	A+B	CW
2	A	CW
3	A+B	PEAK
4	B	CW
5	A	GRAPH
6	A	PEAK
7	B	GRAPH
8	B	PEAK
9	A	MARKER
10	B	MARKER
11	A	PULSE
12	B	PULSE

Table D-4 Wavetek Peak Power Meter Channel/Mode Definitions

CRT an INTEGER*4 input variable containing the logical unit assigned to the DGT.

DBUG — an INTEGER*4 input variable once used to output debug statements during the software development. This variable is not used within the subroutine and a value of zero is appropriate.

EDGE — an INTEGER*4 input variable representing either the rising or falling edge of the Wavetek Peak Power Meter.
1 Rising Edge
2 Falling Edge

FILENAME — a CHARACTER*20 input variable containing the name of the DGT Command File.

IBUS — an INTEGER*4 input variable containing the IEEE 488 interface bus number.

LOG2 — an INTEGER*4 input variable corresponding to the logical unit assigned to an output device such as the terminal or file.

an INTEGER*4 input variable representing the p392XOP function parameter of the Wavetek Peak Power Meter or SA Beacon Receiver. Parameters for the respective instruments are listed in Table D-5.

OPTION	Wavetek	SA Beacon Receiver
0	Frequency	Frequency
1	Reference Delay	2nd LO VFO
2	Curser Delay	Video Attenuation
3	Start Delay	Sig Strength Offset
4	Window Delay	Carrier Indicator
5	Reference Delay	Sig Strength Slope
6	Pulse Rise Time Start	N/A
7	Pulse Rise time End	N/A
8	Pulse Fall Time Start	N/A
9	Pulse Fall Time End	N/A
10	Pulse Width Time Start	N/A
11	Pulse Width Time End	N/A
12	Marker 1 Placement	N/A
13	Marker 2 Placement	N/A
14	Marker 3 Placement	N/A
15	Marker 4 Placement	N/A

Table D-5 Wavetek Peak Power Meter/SA Beacon Receiver Parameter Options

R110 a LOGICAL output variable whose logical state (TRUE) represents the current modem rate of the DGT.

R220 a LOGICAL output variable whose logical state (TRUE) represents the current modem rate of the DGT.

RMSG a REAL*4 input array containing the instrument parameter values to send to an instrument.

SEQN an INTEGER*4 input variable containing the current sequence command number.

STATUS an INTEGER*4 output variable that used to return the execution status of the subroutine.

VALUE a REAL*4 input variable containing the value of the parameter specified by the OPTION variable.

SUBROUTINE CHEKRATE(STATUS,R220,R110,LOG2)
Queries the DGT to determine the current modem rate of the DGT. Sets logical flags indicating its status (220 Mbps /110 Mbps). Other routines use the modem rate flags to determine when to change the modem rate.

SUBROUTINE DGTFILE(FILENAME,CRT,STATUS)
Reads DGT commands from an ASCII text file and sequentially transmits each command to the DGT. Commands are transmitted until an end of file is reached or when an error is encountered. If an error is returned from the DGT, command transmission is halted and an error message is returned to the user terminal.

SUBROUTINE PPMSINGLE(IBUS,ADDR,OPTION,CHANNEL,EDGE,
VALUE,STATUS)
Sends a single command to the Wavetek Peak Power Meter. Instrument command is determined by the OPTION variable.

SUBROUTINE SABRSET(DBUG,IBUS,ADDR,STATUS,OPTION,RMSG)
Sends a single command to the SA Beacon Receiver. Instrument command is determined by the OPTION variable.

SUBROUTINE SUBSEQ(DBUG,SEQN,STATUS)
Executes the sequence commands from a sub-sequence. The code in this subroutine is similar to that in the CTRLLET.FTN program. Nested sub-sequences, and loops are not permitted within a sub-sequence.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111
M1:DSPLET.FTN/111

Library Name M1:DATALET.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990

Variables used by the subroutines:

ERR_SLOT an INTEGER*4 output variable containing the slot number of the instrument in error.

INDEX an INTEGER*4 input variable containing the current sequence number.

IREC an INTEGER*4 input variable specifying the current record number of the data acquisition file.

STAT an INTEGER*4 output variable used to return the execution status of the subroutine.

SUBROUTINE DATALET(ERR_SLOT,INDEX,IREC,STAT)
Acquires data from each instrument defined for a particular sequence. The first measurement is taken after the instrument initialization and before the first sequence command is executed. BER measurements are taken only during a PERFORM BER sequence command. The DGT acquisition status is also verified after each command provided it is defined for the particular sequence. Data is saved to the data file after each reading. Out of limit conditions are saved in the variable BUF2 using bit level commands.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111

Library Name M1:DECDATA.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990

Variables used by the subroutines:

IRATE an INTEGER*4 input variable specifying the user date rate as listed in Table D-6.

Data Rate Mbps	IRATE	ITYPE
1.25	0	0
5.00	3	0
12.500	5	0
13.824	5	8
25.000	8	0
27.648(88)	8	8
27.648(09)	9	0
27.648(89)	9	8
50.000	12	0
55.296	12	8
100.000	13	0
110.592	13	8
200.000	15	0
221.184	15	8

Table D-6 Data Rate and Type Definitions in the DECDATA Library

ITYPE an INTEGER*4 input variable specifying the data type, listed in Table D-6.

0 Represents bursted mode of operation. Data rate is not an integer multiple of the modem rate.

8 Represents continuous mode of operation. Data rate is an integer multiple of the modem rate.

RATE — a REAL*4 output variable whose value is the data rate in Mbps determined by the data rate and data type variables IRATE and ITYPE respectively.

SUBROUTINE DECDATA(IRATE,ITYPE,RATE)
Converts the data rate and data type variables into a numerical value equal to the data rate in Mbps specified by the user.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111

Library Name M1:DGDCLIB.FTN/111
Author Elaine Daugherty / NASA Digital Technology Branch
History Revised September 21, 1990 Version 1.0 <RCR>

Variables used by the subroutines:

ISTSP — an INTEGER*4 input variable depicting the command that is sent to the Data Generator or Data Checker.
1 - START
2 - STOP

IDGC — an INTEGER*4 input variable depicting which device is to receive the command specified by the ISTSP variable.
1 - DG
2 - DC
3 - DC BER only

IDGCN — an INTEGER*4 input variable containing the Data Generator or Data Checker number

ISTAT — an INTEGER*4 input variable containing the status returned from the Data Generator or Data Checker.
0 - ok, no error
1 - request denied (DG)
2 - time out (DG)
95 - couldn't start or stop DG
96 - not 'ACK' response from DG/DC to STOP message
97 - illegal DG/DC
98 - invalid response from DG/DC; contained in
99 - invalid request

ISTAT1 — an INTEGER*4 output variable. See error 98 above.

ISTAT	ISTAT1/START	ISTAT1/STOP
0	who(hex)	0
<> 0	error word	response

IDEST — an INTEGER*4 input variable containing the destination of the Data Generator data. The LET contains one Data Checker for data destination.

IRATE an INTEGER*4 input variable containing the data rate listed in Table D-7.

Data Rate Mbps	IRATE	IDATATYP
1.25	0	0
5.00	3	0
12.500	5	0
13.824	5	8
25.000	8	0
27.648(88)	8	8
27.648(09)	9	0
27.648(89)	9	8
50.000	12	0
55.296	12	8
100.000	13	0
110.592	13	8
200.000	15	0
221.184	15	8

Table D-7 Data Rate and Type Definitions in the DGDCLIB Library

IDATATYP an INTEGER*4 input variable containing the data type listed in Table D-7.
- 0 Represents a bursted mode of operation. Data rate is not a multiple of the clock rate.
- 8 Represents a continuous mode of operation. Data rate is an even multiple of the clock rate.

LU an INTEGER*4 input variable containing the logical unit assigned to the 6809 port.

ISTSP — an INTEGER*4 output variable depicting which command is issuing a response.
1 - START status message
2 - STOP status message
3 - STATUS check message (DG or DC)

INPUTP — an INTEGER*4 output array containing the ASCII status message received from the DG/DC.

WRDS — a DOUBLE PRECISION output variable containing the number of words read from the Data Checker.

XERR — a DOUBLE PRECISION output variable containing the number of errors read by the Data Checker.

BER — a DOUBLE PRECISION output variable containing the bit error rate read from the Data Checker.
Formula used to determine the BER:
BER = WRDS*64 / XERR

IERR — an INTEGER*4 input variable containing the number of errors to send to the Data Checker.
0 < IERR < 99

IDG — see IDGCN.

```
SUBROUTINE STSPDGC(ISTSP,IDGC,IDGCN,ISTAT,ISTAT1,IDEST,
                   IRATE,IDATATYP,LU)
```
Starts or stops the Data Generator or Data Checker.

```
SUBROUTINE STATCHCK(ISTSP,INPUTP,ISTAT,ISTAT1)
```
Decodes the status returned from the 6809 controller.

SUBROUTINE DGDCSTAT (IDGC,IDGCN,ISTAT,ISTAT1,WRDS,XERR,BER,LU)
Reads the status from the Data Generator or Data Checker, and obtains the BER data from the Data Checker.

SUBROUTINE SENDERR(IDG,IERR,ISTAT,ISTAT1,LU)
Sends a user defined number of errors to the Data Generator.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111
M1:DSPLET.FTN/111

Check-Out Software
M1:BER.FTN/111

Library Name M1:DGTLIB.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written September 21, 1990 Version 1.0

Variables used by the subroutines:

BUFF — a CHARACTER*80 output variable containing the ASCII string returned by the 68000 microprocessor.

BUFF1 — see BUFF.

COUNT — The number of characters in the ASCII string sent to the 68000 microprocessor.

DBUG — an INTEGER*4 input variable once used to output debug statements during the software development. This variable is not used within the subroutine and a value of zero is appropriate.

MSG — an INTEGER*4 input/output array used as a message buffer containing the ASCII command sent to an instrument or device. Also contains the instrument response when one is transmitted.

STATUS — an INTEGER*4 output variable used to return the execution status of the subroutine or the status of the system.

SUBROUTINE DGTMSG (DBUG,MSG,COUNT,BUFF,STATUS)
Sends an ASCII string to the 68000 microprocessor.

SUBROUTINE DGTSTAT(DBUG,TEST,BUFF1,STATUS)
Obtains the status of the Digital Ground Terminal.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111
M1:DSPLET.FTN/111

Check-out Software
M1:BER.FTN/111
M1:DGT.FTN/111
M1:MAINCTRL.FTN/111

Library Name	M1:DGTLIB2.FTN/111
Author	Richard C. Reinhart / Analex Corporation
History	Revised September 21, 1990 Version 1.0 <RCR>
	Revised December, 1990 Version 1.0a <RCR>

Variables used by the subroutines:

DAY3	an INTEGER*2 input variable that contains the day of the month that the MAINCTRL Check-Out Software program was installed. This variable is used in the DGT.TSK Check-Out Software program.
DBUG	an INTEGER*4 input variable once used to output debug statements during the software development. This variable is not used within the subroutine and a value of zero is appropriate.
FILENAME	a CHARACTER*20 input variable containing the name of the DGT command file.
FILENAME3	a CHARACTER*19 input variable that contains the name of the DGT error log file.
GBUFF	a CHARACTER*80 output variable containing the error messages generated by the DGT.
NAME3	a LOGICAL input variable that is used as a flag to signify whether the file to save DGT errors has been opened or not. If not opened, the DGT.TSK program opens a new data file.
NUMERR	an INTEGER*4 output variable containing the number of errors generated by the DGT.
RECN3	an INTEGER*4 input variable containing the current record number of the DGT error data file.
STATUS	an INTEGER*4 output variable used to return the execution status of the subroutine.
TEST	a CHARACTER*80 output variable containing the response of the DGT to the 'STAT' command.

SUBROUTINE DGTCOMFILE(DBUG,FILENAME,STATUS)
Sends commands to the DGT from a user data file. Commands are sequentially read from the file and transmitted to the DGT.

SUBROUTINE DGTERR(DBUG,DAY3,NAME3,RECN3,FILENAME3,STATUS)
Sends the RERRLOG command to the DGT, and records the errors generated by the DGT.

SUBROUTINE DGTSTATUS(DBUG,TEST,STATUS)
Decoded the status returned from the 68000 microprocessor.

SUBROUTINE SAVERR(DBUG,NUMERR,GBUFF,DAY3,NAME3,RECN3,
FILENAME3,STATUS)
Records the errors generated by the DGT when the RERRLOG command is executed.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111
M1:DSPLET.FTN/111

Check-Out Software
M1:DGT.FTN/111

Library Name	M1:DSPLIB1.FTN/111
Author	Richard C. Reinhart / Analex Corporation
History	Written December 1990

Variables used by the subroutines:

ADDR — an INTEGER*4 input variable containing the instrument address.

CHANNEL — an INTEGER*4 input variable representing the desired channel for the Wavetek Peak Power Meter command.
Valid inputs for the CHANNEL variable are:
1 Channel A
2 Channel B
3 Channel A and B

DBUG — an INTEGER*4 input variable once used to output debug statements during the software development. This variable is not used within the subroutine and a value of zero is appropriate.

IBUS — an INTEGER*4 input variable containing the IEEE 488 interface bus number.

ID — a CHARACTER*5 input variable containing the name of the function sending a message to a particular instrument.

ISTAT — an INTEGER*4 output variable used to return the execution status of the subroutine.

KBUS — see IBUS

LENGTH — an INTEGER*4 input variable containing the number of bytes contained in the message string to send to an instrument.

MSG — an INTEGER*4 input/output array used as a message buffer containing the ASCII command sent to an instrument or device. Also contains the instrument response when one is transmitted.

OPTION — an INTEGER*4 input variable specifying the condition of the Service Request Query. Valid inputs for the OPTION variable are:
0 Disable SRQ
1 Enable SRQ

STATUS see ISTAT

STATUS1 see STATUS

STBIT an INTEGER*4 output variable containing the error code returned by the Wavetek Peak Power Meter, if an error is present. Consult the Wavetek User's manual for a detailed description of the error codes.

SUBROUTINE ALLREMOTE(DBUG,STATUS1)
Places all instruments defined for a particular sequence into remote mode, placing the instrument under computer control. If the DGT or Data Generator/Data Checker are defined for the experiment, logical units are opened to the respective devices.

SUBROUTINE ALLZERO(DBUG,STATUS1)
Zeros all power meters defined in a particular sequence. This subroutine is no longer used. Power meters are zeroed individually when the Zero Power Meter command is executed.

SUBROUTINE HPZERO(DBUG,IBUS,ADDR,STATUS,ISTAT)
Zeros the HP 437B Power Meter. Disconnect all RF power from the sensor before attempting to zero the power meter.

SUBROUTINE PMZERO(DBUG,IBUS,ADDR,CHANNEL,STATUS,STBIT)
Zeros the defined channels on the Wavetek 8502 Peak power Meter.

SUBROUTINE SAZERO(DBUG,IBUS,ADDR,STATUS)
Sends the ZERO ON NOISE command to the SA Beacon Receiver.

SUBROUTINE PMSTAT(ADDR,IBUS,DBUG,MSG,STATUS)
Checks the status of the Wavetek Peak Power Meter. Returns the appropriate error code if an error exists.

SUBROUTINE SRQPOLL(ADDR,IBUS,DBUG,OPTION,STATUS)
Enables and Disables the service request poll of the Wavetek Peak Power Meter. The SRQ is used by the instrument to flag error conditions.

SUBROUTINE DSPSEND(ADDR,LENGTH,DBUG,KBUS,STATUS,MSG,ID)
Transmits/receives ASCII messages to/from instruments on the IEEE 488 interface bus. This subroutine will read the output parameter from an instrument when the ID variable is properly specified.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111
M1:DSPLET.FTN/111

Library Name M1:DSPLIB2.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990

Variables used by the subroutines:

ADDR an INTEGER*4 input variable containing the instrument address.

DBUG an INTEGER*4 input variable once used to output debug statements during the software development. This variable is not used within the subroutine and a value of zero is appropriate.

IBUS an INTEGER*4 input variable containing the IEEE 488 interface bus number.

ISLOT an INTEGER*4 input variable containing the slot number of the instrument in the instrument file array.

NSTAT an INTEGER*4 output array indicating which of the initialization parameters are in error.

OPTION an INTEGER*4 input variable indicating the desired channel for the Wavetek Peak Power Meter command.

SETTINGS an INTEGER*4 input array containing the HP Relay Actuator switch positions.
Valid inputs for the SETTINGS variable are:
0 Off
1 On

STATUS1 an INTEGER*4 output variable used to return the execution status of the subroutine.

STBIT an INTEGER*4 output variable containing the error code returned by the Wavetek Peak Power Meter, if an error is present. Consult the Wavetek User's manual for a detailed description of the error codes.

SUBROUTINE ALLINIT(DBUG,STATUS1)
Calls the appropriate subroutines to initialize all the instruments defined for a particular experiment.

SUBROUTINE CHECKERR(ADDR,DBUG,IBUS,STATUS1)
Checks the status of the HP 437B Power Meter.

SUBROUTINE CWINIT(DBUG,IBUS,ADDR,STATUS1,ISLOT,OPTION,STBIT)
Initializes the CW mode of the Wavetek Peak Power Meter. Parameters include frequency, number of averages, offset and units. The channel to receive the command is also specified using this subroutine.

SUBROUTINE DGDCINIT(DBUG,ISLOT,STATUS)
Determines the state of the DGT, and if necessary enables the proper modem burst rate. The DGT control is included in the DG/DC initialization routine because the initial modem burst rate is contained in the Data Generator slot within the Instrument File. This routine will initialize the burst rate of the DGT if not properly configured.

SUBROUTINE DGTINIT(DBUG,ISLOT,STATUS)
Queries the DGT to determines its present state. Similar to DGDCINIT, except that it does not change the state of the DGT. This is a read only command.

SUBROUTINE FCINIT(ADDR,DBUG,IBUS,ISLOT,STATUS1)
Initializes the EIP Frequency Counter. Parameters include frequency band, resolution, frequency offset, frequency high limit, and frequency low limit.

SUBROUTINE HPINIT(DBUG,IBUS,ADDR,ISLOT,STATUS)
Initializes the HP 437B Power Meter. Parameters include frequency, range, trigger mode, offset, calibration factor, units, low limit, and high limit.

SUBROUTINE PEAKINIT(DBUG,IBUS,ADDR,STATUS1,ISLOT,OPTION,STBIT)
Initializes the PEAK mode of the Wavetek Peak Power Meter. Parameters include frequency, number of averages, offset, reference delay, cursor delay, and units. The channel to receive the command is also specified using this subroutine.

SUBROUTINE PININIT(DBUG,IBUS,ADDR,ISLOT,STATUS1)
Initializes the HP 6632A Power Supply. Parameters include setting voltage, current, and high voltage.

SUBROUTINE RAINIT(ADDR,IBUS,DBUG,STATUS,SETTINGS)
Initializes the HP Relay Actuator. Each switch is initialized to either the on or off position.

SUBROUTINE SAINIT(DBUG,IBUS,ADDR,ISLOT,STATUS,NSTAT)
Initializes the SA Beacon Receiver. Parameters include frequency, 2nd IF slot, gain control, gain hold, AGC time constant, demod slot, 2nd LO mode select, output synchronous AGC, anti-sideband select, and search mode selection.

SUBROUTINE SWDRINIT(DBUG,IBUS,ADDR,ISLOT,STATUS)
Initializes the HP Attenuator Switch Drivers when used solely as a switch driver. Switch positions are determined from individual locations within the instrument array.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111
M1:DSPLET.FTN/111

Library Name M1:DSPLIB3.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990

Variables used by the subroutines:

DBUG — an INTEGER*4 input variable once used to output debug statements during the software development. This variable is not used within the subroutine and a value of zero is appropriate.

FLAG — a CHARACTER*3 output variable containing a mnemonic string indicating the out of limit condition, that is displayed to the user terminal during data acquisition.

INS_SLOT — an INTEGER*4 input variable containing the instrument file array position of the instrument being tested. Instrument slot number indicates which bit position to test for an under or over limit condition.

OF — an INTEGER*4 variable that specifies the desired output field number to be updated with data contained in 'OFDATA'.

OUT_OF_LIMIT — a LOGICAL output variable indicating if an out of limit condition exist.

POSITION — an INTEGER*4 output array indicating which data array variable is out of limit.

SDA_FILE — a CHARACTER*20 input variable containing the name of the sequence definition file to be output to either the user terminal or to the line printer.

STATUS1 — an INTEGER*4 output variable used to return the execution status of the subroutine.

TDF_FILE — a CHARACTER*20 input variable containing the name of the instrument definition file to be output to either the user terminal or to the line printer.

VIEW — a CHARACTER*1 input variable indicating the device to output the instrument definition and sequence definition files.

SUBROUTINE ALLLOCAL(DBUG,STATUS1)
Places all instruments defined for a particular sequence into local mode, returning control of the instrument to its front panel keys.

SUBROUTINE BITDECODE(INS_SLOT,FLAG,OF,OUT_OF_LIMIT,POSITION)
Determines if a saved reading is out of the user specified limits by testing the BUF2 variable. Out of limit conditions include:

Under Limit	Under Range
Over Limit	Over Range

The out of limit condition is determined by the position of the bit set in the variable BUF2, listed in Table D-8. Status bits are stored in BUF2 beginning with bit number 128, or BUF2(6). Bit positions are determined by the data field location and the instrument position within the Instrument File. This is illustrated in Table D-8 as an array with each bit position corresponding to an instrument and either its first or second data field.

Out of limit condition	Data field	Instrument Number						
		1	2	3	4	5	...	50
under limit	Data(1,*)	128	136	144	152	...		
over limit	Data(1,*)	129	137	145	...			
under range	Data(1,*)	130	138	146				
over range	Data(1,*)	131	139	147				
under limit	Data(2,*)	132	140	148				
over limit	Data(2,*)	133	141	149				
under range	Data(2,*)	134	142	150				
over range	Data(2,*)	135	143	151				

Table D-8 Out of Limit Status Bit Positions

SUBROUTINE DISPLAY(VIEW,SDA_FILE,TDF_FILE)
Outputs both the Instrument Definition and Sequence Definition files to either the line printer or to the user terminal. This subroutine was mainly used during the software development, and is not used for program execution.

Programs:

Experiment Control and Monitor Software
M1:DSPLET.FTN/111

Library Name M1:EBNZERO.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990

Variables used by the subroutines:

D_EBNO a REAL*4 input variable containing the value of Eb/No desired by the user.

SEQ_COL an INTEGER*4 input variable containing the sequence command number executed by the Sequence Execution Software.

STATUS an INTEGER*4 output variable used to return the execution status of the subroutine.

SUBROUTINE EBNZERO(SEQ_COL,D_COL,STATUS)
Sets the noise attenuator located within the IF Noise Unit to provide an Eb/No within +/- .55 dB of the user specified value. The routine will take measurements from the power meter specified in the Data Generator slot of the instrument file. This routine will not affect the signal attenuator. An initial noise attenuation value is read from the instrument file to begin the Eb/No calculation. Future calculations are made from the last value of attenuation used.

The software was modified at ACTS System test to allow an external signal input to the independent experimenter uplink port to surpress the noise floor during noise power measurements.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111

Library Name M1:ENCDATA.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written March 1991

Variables used by the subroutines:

IN a CHARACTER*24 output array containing the input fields of the current menu.

IRATE an INTEGER*4 input variable specifying the user data rate listed in Table D-9.

Data Rate Mbps	IRATE	ITYPE
1.25	0	0
5.00	3	0
12.500	5	0
13.824	5	8
25.000	8	0
27.648(88)	8	8
27.648(09)	9	0
27.648(89)	9	8
50.000	12	0
55.296	12	8
100.000	13	0
110.592	13	8
200.000	15	0
221.184	15	8

Table D-9 Data Rate and Type Definitions in the ENCDATA Library

ITYPE an INTEGER*4 input variable specifying the data type listed in Table D-9.
- 0 Represents bursted mode of operation. Data rate is not an integer multiple of the modem rate.
- 8 Represents continuous mode of operation. Data rate is an integer multiple of the modem rate.

SUBROUTINE TYPERATE(IRATE,ITYPE,IN)
Converts a user input data rate into the proper IRATE and ITYPE variables.

SUBROUTINE SETUPBER(IRATE,ITYPE,IN)
Converts the IRATE and ITYPE variables into the proper menu input data rate parameter.

Programs:

Experiment Control and Monitor Software
M1:SDALET.FTN/111

Library Name M1:EXECLET.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990

Variables used by the subroutines:

COMM an INTEGER*4 input variable containing the current sequence command number.

DBUG an INTEGER*4 input variable once used to output debug statements during the software development. The variable is not used within the subroutine and a value of zero is appropriate.

GCOMM see COMM

SEQ_NUM see COMM

STATUS1 an INTEGER*4 output variable used to return the execution status of the subroutine or the status of the system.

VALUE a REAL*4 input variable containing the output value of the instrument specified in the Check a Parameter command.

IDUM see DBUG

ICOMM see COMM

ISTAT see STATUS1

ISTATUS see STATUS1

SUBROUTINE CKLIMIT(VALUE,ICOMM,ISTAT)
Compares the output value of the instrument specified in the Check a Parameter command to the user defined high and low limits to determine the appropriate control action. There are three possible paths.
1) Output value greater than high value.
2) Output value lower than low value.
3) Output value within the specified range.
Sequence control is transferred to the proper command accordingly. Refer to the EC&M User's Guide for additional information on control transfer.

SUBROUTINE EXECL(DBUG,SEQ_NUM,STATUS1)
Executes the appropriate subroutine to carry out each command specified in the sequence. Data acquisition is performed before each command is executed. Any errors that occur are transferred to the display program via the SNDMSG FORTRAN VII RTL subroutine.

SUBROUTINE PARMCHECK(COMM,STATUS1)
Obtains the current output reading from the appropriate instrument and passes it to the CKLIMIT subroutine. Limits specified for this command are independent of those in the Instrument Definition Software.

SUBROUTINE PARMGOTO(GCOMM,ISTATUS)
Transfers sequence control from the executing GOTO command to the sequence command corresponding to the label number specified by the user. An error is returned if no label number matches the label specified in the GOTO command.

SUBROUTINE PARMSET(DBUG,COMM,ISTATUS)
Sets a specified parameter to a user defined value. The new parameter value is compared to the high and low limits specified in the IDS and appropriate action taken depending on the out of limit condition, also specified in the IDS. Potential out of limit errors are also detected by this routine and the user notified. All first occurrences are reported to the user regardless of the out of limit action. Future occurrences are dependent upon the out of limit action selected by the user.

Parameters include:

Attenuation	Eb/No
Power Supply Voltage	Power Supply Current
Frequency	Toggle Switch
Wavetek Parameters	SA Beacon Receiver Parameters

SUBROUTINE PARMSTEP(DBUG,COMM,ISTATUS)
Increments or decrements (user defined) a specified parameter by a user defined amount. The new parameter value is compared to the high and low limits specified in the IDS and appropriate action taken depending on the out of limit condition also specified in the IDS. Potential out of limit errors are also detected by this routine and the user notified. All first occurrences are reported to the user regardless of the out of limit action. Future occurrences are dependent upon the out of limit action selected by the user.
Parameters include:

Attenuation	Eb/No
Power Supply Voltage	Power Supply Current

SUBROUTINE PERFBER(DBUG,COMM,STATUS1)
Uses subroutine DATALET to obtain BER measurements from the data checker. This routine uses the number of readings and time between each reading as specified by the user to take the proper BER measurements. Control flags are used to instruct DATALET to obtain the BER readings along with its normal data acquisition. A final data acquisition (without a BER measurement) is performed so that all BER readings are displayed to the user terminal.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111

Library Name M1:GENERLIB.FTN/111
Author Elaine Daugherty / NASA Digital Technology Branch
History Revised July, 1990 Version 1.0 <RCR>

Variables used by the subroutines:

ADDRESS an INTEGER*4 input variable containing the address of the instrument.

DBUG an INTEGER*4 input variable once used to output debug statements during the software development. The variable is not used within the subroutine and a value of zero is appropriate.

FN a CHARACTER*20 input variable containing the file name of the menu used by the menu driver.

IDEV a CHARACTER*5 input variable representing the I/O device for the menu display. This variable is no longer used by the subroutine.

IBUS an INTEGER*4 input variable containing the IEEE 488 interface bus number.

OFDATA a CHARACTER*80 input variable containing the output field for the current menu.

P a CHARACTER*24 input variable containing the input fields for the current menu.

STATUS an INTEGER*4 output variable used to return the execution status of the subroutine.

SUBROUTINE CLEARM (IDEV,FN,P,OFDATA)
Clears the input and output variables for the current menu.

SUBROUTINE ICLEAR (DBUG,IBUS,STATUS)
Clears the IEEE 488 interface bus.

SUBROUTINE LOCAL (DBUG,IBUS,ADDRESS,STATUS)
Returns control of an instrument to the front panel keys.

SUBROUTINE REMOTE (DBUG,IBUS,ADDRESS,STATUS)
Places an instrument under computer control.

Programs:

Experiment Control and Monitor Software

M1:TDFLET.FTN/111
M1:SDALET.FTN/111

Check-Out Software

M1:ALL.FTN/111
M1:B37.FTN/111
M1:BER.FTN/111
M1:C9.FTN/111
M1:C26.FTN/111
M1:C27.FTN/111
M1:L6.FTN/111
M1:L16.FTN/111
M1:MAINCTRL.FTN/111
M1:N6.FTN/111
M1:N12.FTN/111
M1:N20.FTN/111
M1:PIN.FTN/111
M1:PM1.FTN/111
M1:PM2.FTN/111
M1:R9.FTN/111
M1:R11.FTN/111
M1:R12.FTN/111
M1:RFU.FTN/111
M1:SA1.FTN/111
M1:U13.FTN/111
M1:U16.FTN/111
M1:U21.FTN/111

Library Name M1:GMLIB.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written October 2, 1990 Version 1.0 <RCR>

Variables used by the subroutines:

ADDRESS an INTEGER*4 input variable containing the decimal address of the attenuator.

ADDRS see ADDRESS.

ATTN a REAL*4 input variable containing the desired value of attenuation. $0 \leq ATTN \leq 60.0$ (.05 increments)

BUFF2 an INTEGER*4 output array that contains the response from the attenuator if the echo is enabled or if an error occurs.

BUFF3 see BUFF2.

CHANN a CHARACTER*1 input variable containing the attenuator channel.
1 Channel A
2 Channel B

IN a CHARACTER*24 input array containing the input fields for the current menu.

OFON an INTEGER*4 input variable that determines whether to turn the echo off or on.
0 Off
1 On

PORT an INTEGER*4 input variable containing the logical unit assigned to the attenuator.

STATUS an INTEGER*4 output variable used to return the execution status of the subroutine.

SUBROUTINE ECHOFON(OFON,ADDRESS,PORT,STATUS,BUFF3)
Enables or disables the echo feature on the GM attenuator controller board.

SUBROUTINE GMASET(ATTN,CHANN,ADDRS,PORT,STATUS,BUFF2)
Custom initializes the GM attenuator with the ECHO DISABLED. The value of attenuation, attenuator address and channel are user specified.

SUBROUTINE GMCHECK(ATTN,IN,STATUS)
Verifies that the value of attenuation input by the user is within the valid range.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111
M1:DSPLET.FTN/111

Check-Out Software
M1:GA0.FTN/111
M1:GA1.FTN/111
M1:GB0.FTN/111
M1:GB1.FTN/111

Library Name M1:GRTL15.OBJ/S
Author BGL Technology Corporation
History Copyright 1980

Variables used by the subroutines are:

ADDRESS an INTEGER*4 input variable containing the decimal address of the device.

GRETC an INTEGER*4 output variable used to return the execution status of the subroutine.

LENGTH an INTEGER*4 input variable containing a decimal value indicating the number of characters contained in the MESSAGE string.

MESSAGE an INTEGER*4 input/output array used as a message buffer containing command characters to be sent to a device or as a buffer to receive the response from the device.

UTERMC an INTEGER*4 input variable containing the hexadecimal code for the message termination character used by the device.

SUBROUTINE GCLOSE (GRETC)
Closes the IEEE bus interface when all messages between the computer and bus devices have been sent. A GRETC value of zero indicates successful closing of the interface.

SUBROUTINE GCTRL (MESSAGE,LENGTH,GRETC)
Sends bus control messages to the bus controller. For this subroutine MESSAGE contains the hexadecimal command characters to be sent to the bus controller. A GRETC value of zero indicates successful execution of the subroutine.

SUBROUTINE GINIT (GRETC)
Initializes the IEEE 488 Interface.
NOTE: This Subroutine MUST be called first. For this subroutine GRETC must be set to zero before calling the subroutine. A GRETC value of zero indicates successful initialization of the interface.

SUBROUTINE GLSTNB (ADDRESS,MESSAGE,LENGTH,GRETC)
Receives a binary string from a device on the bus. For this subroutine MESSAGE is a buffer into which a response from a device on the bus is stored. For this subroutine LENGTH is the number of characters read from the device.

SUBROUTINE GTALKA (ADDRESS,MESSAGE,LENGTH,GRETC)
Sends an ASCII string to a device on the bus. For this subroutine MESSAGE contains the ASCII characters of the device command string. A GRETC value of zero indicates successful execution of the subroutine.

SUBROUTINE GLSTNA (ADDRESS,MESSAGE,LENGTH,UTERMC,GRETC)
Same as GLSTNB except that an ASCII string is expected from the device and the string termination character must be supplied in the UTERMC variable. GLSTNB works for most instruments and is more generic

SUBROUTINE GTALKB (ADDRESS,MESSAGE,LENGTH,GRETC)
Same as GTALKA except that a binary string is sent to the device. Most devices expect an ASCII string, therefore GTALKA is preferred.

Programs:

Experiment Control and Monitor Software
M1:DSPLET.FTN/111

Check-Out Software
M1:ALL.FTN/111
M1:B37.FTN/111
M1:BER.FTN/111
M1:C9.FTN/111
M1:C26.FTN/111
M1:C27.FTN/111
M1:L6.FTN/111
M1:L16.FTN/111
M1:MAINCTRL.FTN/111
M1:N6.FTN/111
M1:N12.FTN/111
M1:N20.FTN/111
M1:PIN.FTN/111
M1:PM1.FTN/111
M1:PM2.FTN/111
M1:R9.FTN/111
M1:R11.FTN/111
M1:R12.FTN/111
M1:SA1.FTN/111
M1:U13.FTN/111
M1:U16.FTN/111
M1:U21.FTN/111

Library Name M1:INSETUP.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990

Variables used by the subroutines:

DATA — a CHARACTER*80 input variable containing one record of data that is read from the DEFAULT.LET data file.

DEFAULT2 — a REAL*4 two-dimensional output array containing the instrument default values read from the DEFAULT.LET file.

START — an INTEGER*4 input variable specifying the starting position of the data record to decode.

STATUS2 — an INTEGER*4 output variable used to return the execution status of the subroutine.

VALUE — a REAL*4 output variable containing the value decoded from the data record.

SUBROUTINE INSETUP (DEFAULT2,STATUS2)
Reads in the default values of the instrument parameters from an external ASCII file. The name of the ASCII data file is M1:DEFAULT.LET/111. Users may edit the data file DEFAULT.LET to modify the instrument parameter default values. DEFAULT.LET is used in conjunction with a second file M1:LABELS.LET/111 to reduce the number of format restrictions within the DEFAULT.LET file. The second file LABELS.LET, is maintained by the program developer and contains a list of the instruments and instrument parameters used in the DEFAULT.LET data file. Refer to Section 5 of this manual for additional information.

SUBROUTINE DECREC(START,DATA,VALUE)
Decodes each record of data read in from the DEFAULT.LET data file. A minimum of one space or a comma must exist between the parameter name and parameter value in each record.

Programs:

Experiment Control and Monitor Software
M1:TDFLET.FTN/111

Library Name M1:MENUP.FTN/2
M1:MENUP2.FTN/111

Author Monty Andro / NASA Digital Technology Branch
History Revised 1987 Ed Petrik / NASA DTB

Variables used by the subroutines:

IDEV — a CHARACTER*5 input variable representing the I/O device for the menu display. This variable is no longer used by the subroutine.

P — a CHARACTER*24 input array which contains the input field reads.

OF — an INTEGER*4 input variable which specifies the desired output field number to be updated with data contained in 'OFDATA'

OFDATA — a CHARACTER*80 input variable which contains the data to be displayed in the output field number passed in 'OF'.

CMND — an INTEGER*4 input variable used to activate various features and options depending on the bit settings.

OP — an INTEGER*4 input variable which contains a menu number to be displayed or input field number when defining input options.

FKEY — an INTEGER*4 input/output variable which contains the function key number activated during a read.

FN — a CHARACTER*20 input variable containing the file name of the menu used by the menu driver.

STAT — an INTEGER*4 output variable which returns the status of the subroutine.

SUBROUTINE MENUP(IDEV,CMND,OP,FKEY,FN,P,OF,OFDATA,STAT)
Displays menus to the user terminal. For further details, consult the MENUP documentation.

Note: The difference between library MENUP and MENUP2 is that MENUP2 performs a write with a wait, while MENUP performs a write and continue. The write and wait or write and continue refers to the function used in the SYSIO when writing menus to the user terminal. Refer to the OS/32 User's Manual for addition information on the SYSIO subroutine.

Programs:

Experiment Control and Monitor Software

M1:SDALET.FTN/111
M1:TDFLET.FTN/111

Check-Out Software

M1:ALL.FTN/111
M1:B37.FTN/111
M1:BER.FTN/111
M1:BIT.FTN/111
M1:C9.FTN/111
M1:C26.FTN/111
M1:C27.FTN/111
M1:DAI.FTN/111
M1:DAT.FTN/111
M1:GA0.FTN/111
M1:GA1.FTN/111
M1:GB0.FTN/111
M1:GB1.FTN/111
M1:L6.FTN/111
M1:L16.FTN/111
M1:MAINCTRL.FTN/111
M1:N6.FTN/111
M1:N12.FTN/111
M1:N20.FTN/111
M1:PIN.FTN/111
M1:PM1.FTN/111
M1:PM2.FTN/111
M1:R9.FTN/111
M1:R11.FTN/111
M1:R12.FTN/111
M1:RFU.FTN/111
M1:SA1.FTN/111
M1:U13.FTN/111
M1:U16.FTN/111
M1:U21.FTN/111

Library Name M1:SA1SEC.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990

Variables used by the subroutines:

ADDR an INTEGER*4 input variable containing the instrument address.

COUNT an INTEGER*4 output variable whose value indicates the Master Division Alarm bit in error.

COUNT1 see COUNT

IBUS an INTEGER*4 input variable containing the IEEE interface bus number.

SLOPE an INTEGER*4 output variable containing the slope value read from the SA Beacon Receiver.

STATUS an INTEGER*4 output variable used to return the execution status of the subroutine.

TEST a INTEGER*4 input array containing the Master Division Alarm Status bits returned from the SA Beacon Receiver.

SUBROUTINE RSLOPE(IBUS,ADDR,SLOPE,STATUS)
Reads the SIG STR SLOPE setting of the SA Beacon Receiver. Values range from 10.00 to 40.00 db/Volt.

SUBROUTINE SASTAT(IBUS,ADDR,STATUS,COUNT)
Reads the status of the MASTER DIVISION ALARMS of the SA Beacon Receiver. The alarm bit settings are user defined. Settings in place at time of this publication are listed in Table D-10.

Bit Number	Correct Value	Description
0	0	No hardware fault present in Tuner lock alarm
1	0	Demod slot 1 locked
2	1	Demod slot 2 not locked
3	0	Carrier above threshold
4	0	No hardware fault present in Frequency Synthesizer
5	1	Internal reference
6	0	No hardware fault present in 2nd LO Lock Alarm
7	1	Internal LO

Table D-10 Master Division Alarm Status Codes

SUBROUTINE SA1ERRCK(IBUS,STATUS,TEST,COUNT1)
Compares the MASTER DIVISION ALARM status returned from the SA Beacon Receiver with the user defined configuration and returns the appropriate status to the SASTAT subroutine.

Programs:

Experiment Control and Monitor Software
M1:CTRLLET.FTN/111

Library Name M1:SDALIB2.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990 Version 1.0

Variables used by the subroutines:

COL an INTEGER*4 input variable that contains the current sequence number.

SUBROUTINE PMZERO (COL)
Defines an instrument to zero. Valid instruments include the Wavetek Peak Power Meters, HP Power Meters, or the SA Beacon Receiver.

SUBROUTINE SETPARM (COL)
Defines the parameter to set, parameter value and instrument. Valid parameters to set are; frequency, attenuation, Eb/No, power supply voltage, power supply current, toggle switch, selected Wavetek parameters, and selected SA Beacon Receiver parameters.

SUBROUTINE STARTDG (COL)
Sets up a command to start the Data Generator. User data type, user data rate, modem rate and data destination are required. Internal flags (Continuous, Bursted) are set for DGT control.

SUBROUTINE STEPPARM (COL)
Defines the parameter to step, parameter increment or decrement value, and instrument. Valid parameters to step are; attenuation, power supply voltage, power supply current and Eb/No.

SUBROUTINE STOPDC (COL)
Sets up command to stop the Data Checker. Default is Data Checker number 1.

SUBROUTINE STOPDG (COL)
Sets up command to stop the Data Generator. Default is Data Generator number 1.

SUBROUTINE WAITT (COL)
Defines the length of time to suspend sequence execution. Total time and interval time must both be entered. Interval time controls how many readings are taken within the total wait time. Bypass data acquisition by selecting the appropriate menu option.

Programs:

Experiment Control and Monitor Software
M1:SDALET.FTN/111

Library Name M1:SDALIB3.FTN/111
Author Richard C. Reinhart / Analex Corporation
History Written December 1990 Version 1.0

Variables used by the subroutines:

COL	an INTEGER*4 input variable that contains the current sequence number.
COL2	an INTEGER*4 input variable that contains the current sequence number.
LABELIN	an INTEGER*4 input variable containing a user input label number for the sequence number COL2.
LABELOUT	an INTEGER*4 output variable containing the sequence control label number generated by the LABELIT subroutine.
LAST_LABEL	an INTEGER*4 input variable that contains the label number last input by the user before entering the SETLOOP subroutine.
MENU	an INTEGER*4 input variable that contains the menu number of the current menu for the purpose of displaying error messages to the user terminal.

SUBROUTINE BERMEASURE (COL,MENU)
Defines a bit error rate measurement. Required parameters are time from starting the Data Generator until reading the Data Checker, the number of measurements at a particular Eb/No, and the data destination. Data Generator number one is automatically selected. Define both a Data Generator and a Data Checker in the IDS or errors will result.

SUBROUTINE CALLSUB (COL)
Defines the name of a sub-sequence within a main sequence. A Sub-sequence file must exist before it is called. Nested Sub-sequences are not supported.

SUBROUTINE CHECKPARM (COL)
Defines the parameter to check, the parameter high and low limits, and the respective label numbers to goto in the event the parameter is outside the defined limits. Parameters to check include power level, Eb/No, bit error rate, power supply voltage, power supply current and frequency counter reading.

SUBROUTINE GOTOST (COL)
Transfers control during a sequence to a command other than the succeeding command. Label numbers are converted to sequence control numbers only if the label currently exists. If a label does not yet exist, it is converted to a control number by the ENDSEQ subroutine in program SDALET.

SUBROUTINE LABELIT (LABELIN,LABELOUT,COL2,STATUS1)
Converts a user input label number to a sequence control label number. Sequence control label numbers can be a single execution label number, loop label number or sub-sequence label number. Label numbers are converted to sequence control numbers only if the label currently exists. If a label does not yet exist, it is converted to a control number by the ENDSEQ subroutine in program SDALET.

SUBROUTINE SETERROR (COL)
Defines the number of errors to include in user data. Data Generator number one is assumed, no other input required.

SUBROUTINE SETLOOP (COL,LAST_LABEL)
Defines a loop within a sequence. The number of commands within the loop is calculated by the subroutine. Nested loops are not supported. The label number of the final sequence command 'END SEQUENCE' is reserved for software control.

Programs:

Experiment Control and Monitor Software
M1:SDALET.FTN/111