Distribution
Category UC-705

SAND93-1571
Unlimited Release
Printed January 1994

AUTOSIM USER GUIDE

EDDIE D. BAKER
SANDIA NATIONAL LABORATORIES
9321 INSTRUMENTATION DEVELOPMENT DEPARTMENT

Abstract

Autosim is a software package written to control and trigger the programmable instruments that are used to supply simulated signals to the recording devices on underground nuclear weapons effects tests at the Nevada Test Site. These instruments are located either in the tunnel or at a remote site, and may be controlled from anywhere on the Department 9320 Computer Network. Autosim incorporates commands to control the operation of the Laser Calibrator that is a fiber optic device that transmits a signal from down-hole to the uphole recorders. Autosim also supports the task of characterizing the cable links by communicating to some high bandwidth digitizers that are used to input the pulse of the downhole simulator. To minimize the learning interval, Autosim utilizes menus and offers on-line help on most of the selections in the menu options.

rb

1. Introduction

1.1 Before Autosim

Pulse generators, also called simulators in this document, are usually located in the vicinity of the gauges and are used to supply signals to the recording devices. The defined signals are sent to test the accuracy and continuity of the recording system before the actual test. The simulators need to be powered on and set up correctly before each scheduled dry run and per the operator's request. In the past, this required someone who knew how to operate the simulators to go to all the locations of the simulators in the tunnel and manually perform these tasks whenever the simulators' signals were needed.

In the late 1980s, Sandia National Laboratories Underground Nuclear Testing Department, 9320, made a commitment to move the high frequency recording systems out of the tunnel and into building 12-909 above ground. However, it was still necessary to have an experienced operator go down in the tunnel and turn on and set up all the simulations. A remote-control system was needed that could perform these operations from building 12-909.

1.2 Introducing Autosim

Autosim is a software package written to control and trigger programmable instruments that includes 3 pulse generators, a laser calibrator and 3 high bandwidth digitizers. The pulse generators provide the trigger and simulator signals. The laser calibrator controls the transmission of the simulator and gage signals from downhole to the uphole recorders. The digitizers are used to record the signals taken from the simulators in the tunnel for cable characterization. The software resides on one of the computer nodes on department 9321 Digital Equipment Corp (DEC) VAX Computer Network.

The software is written in Fortran, and utilizes the DEC licensed package Form Management System, FMS, to build menus. Autosim utilizes department 9321's Ingres database sys.em to retrieve parameters for each simulator. Autosim powers on and off all simulators and downloads the stored parameters into each device. With Autosim, an operator in building 12-909 may perform a dry run, sending signals and triggers to the system, without any need for downhole personnel.

Section 1 of this document presents an introduction to Autosim. Section 2 lists and describes the environment required to run this package. Section 3 illustrates how to start-up and run Autosim, followed by a thorough explanation of each option listed in the menus. Section 4 explains the main flow of the code and has a description of the software subroutines used in Autosim.

2. External Interface Description

2.1 Hardware Layout

Autosim is designed to communicate with instruments that may be up to two miles away from the computer or terminal server. It accomplishes this by using a device manufactured by Black Box Corporation called the 488 Controller. Figure 2.0 shows a simplified version of the hardware layout as used in our system.

The 488 Controller uses its two I/O ports to communicate to the computer and to the simulator. The interface between the computer and the 488 Controller is a standard serial RS422 port set at a baud rate of 19200, and the other interface between the 488 Controller and the simulators is a General Purpose Interface Bus, GPIB. The serial RS422 port is specified to transmit and receive over a distance of two miles. The GPIB bus conforms to IEEE 488-1978 standard, which states that up to 14 devices or simulators may be connected to it at any given time. The 488 Controller is configured as the GPIB system controller.

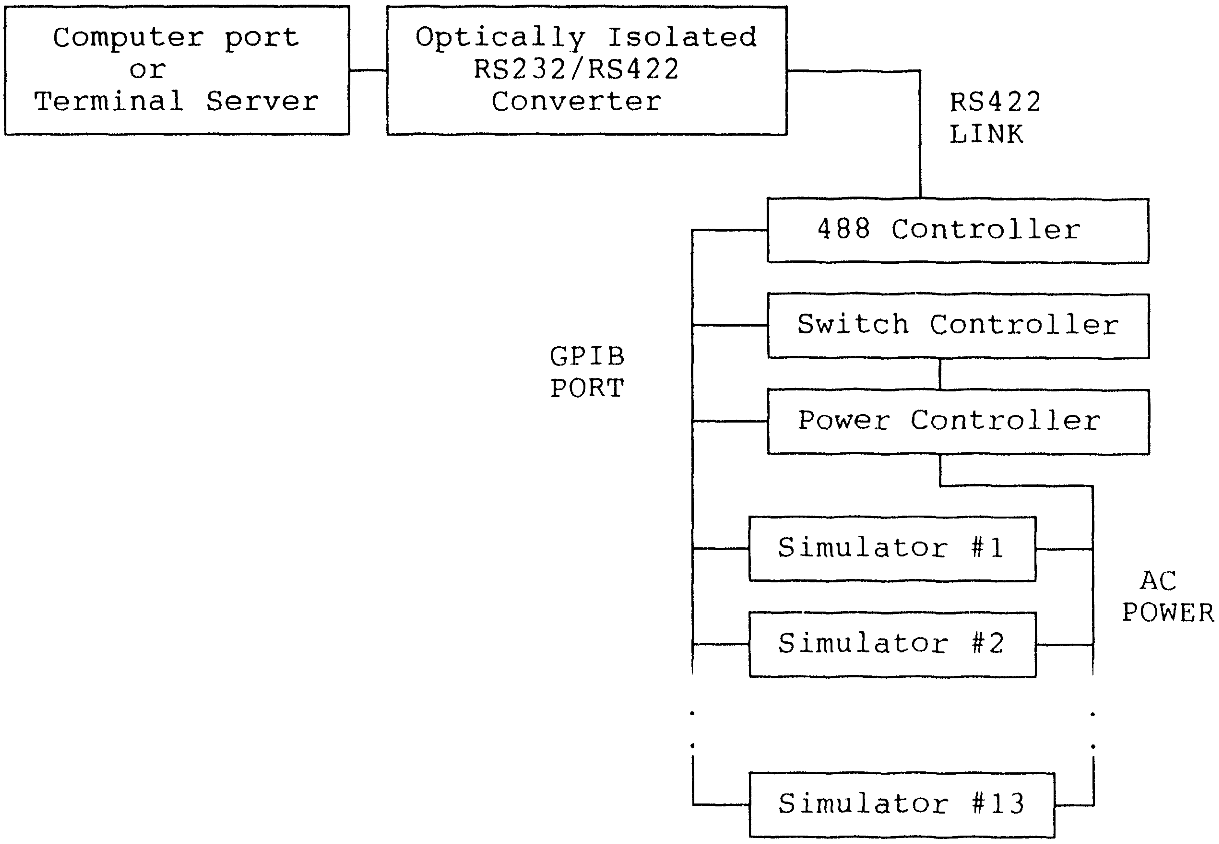


Figure 2.0 Hardware layout

2.2 Device Descriptions

The optical isolator/converter model 262 shown in Figure 2.0, is made by Telebyte Technology, inc. It is used to convert a single-ended RS232 signal to a differential RS422 signal. The conversion increases the distance that signals may be transmitted. Also, this device is used to optically isolate the Terminal Server and the 488 Controller from each other.

The Hewlett Packard Switch Controller, model 3488, and the Power Controller, designed and built by Dennis Banto of EG&G, work together to remotely control the 120 volts AC power to the devices. A schematic of the Power Controller is shown in Appendix A. Inside the Power Controller are ten individual controlled circuits. The simulators' AC power cords are plugged into the Power Controller's AC power receptacles. Each circuit in the Power Controller is designed to monitor the state of a switch, that is, whether the switch is opened or closed. A closed switch turns the AC power on to a device, and an open switch turns off the AC power to a device. The Hewlett Packard Switch Controller is used to control the switches. By being programmable, commands may be sent over the GPIB interface to control the switch's actions. The 488 Controller, Switch Controller and the Power Controller must remain on at all times to be able to control the simulators' AC power state.

Autosim has code to communicate to the following programmable pulse and trigger generators: Wavetek Pulse Generator, model 1560; Berkeley Nucleonics Corporation Universal Pulse Generator, model 6040 with model 202h and 203E plug-ins; and Stanford Research Systems Inc Pulse Generator, model DG535. Other types of pulse generators are also used in department 9320 system; however, these generators are not programmable and may only be powered on, triggered and powered off by the code.

2.3 System's Connection

This section lists the instructions for assembling the Autosim system. This section also lists and explains some of the error messages that the code displays when it detects either a bad connection or a bad device.

The System's Network Manager must set up the parameters of the computer port where Autosim hardware runs. Appendix B presents the port parameters set up by Leonard Livingston, 9321, that displays the correct parameters for a terminal server port. Also, because of the large ASCII files that are sometimes transferred, all operators running this software package must have buffer limits on the DEC computer set to a minimum of 50,000 bytes.

The RS232/RS422 converter, a 25 pin connector, plugs into this computer port. To transmit a differential signal, the converter has 4 output pins that are connected to the 488 Controller. The plus transmit, +Tx, on the Converter goes to the plus receive, pin 14, on the Controller. The minus transmit, -Tx, on the Converter goes to the minus receive, pin 2, on the Controller. The plus receive, +Rx, on the converter goes to the plus transmit, pin 16, on the Controller. The minus receive, -Rx, on the converter goes to the minus transmit, pin 3, on the Controller.

Figure 2.1 shows the configuration of 3 dip switches located inside of the 488 Controller that must be set as shown. The "*" represents the required switch position. These switch positions control things like the Baud rate and the system mode. For a complete definition of all switch settings, refer to the 488 Controller manual. Also, there is a shorting plug inside the Controller positioned at J206 that must be moved to J205 to convert the Controller from RS232 to RS422.

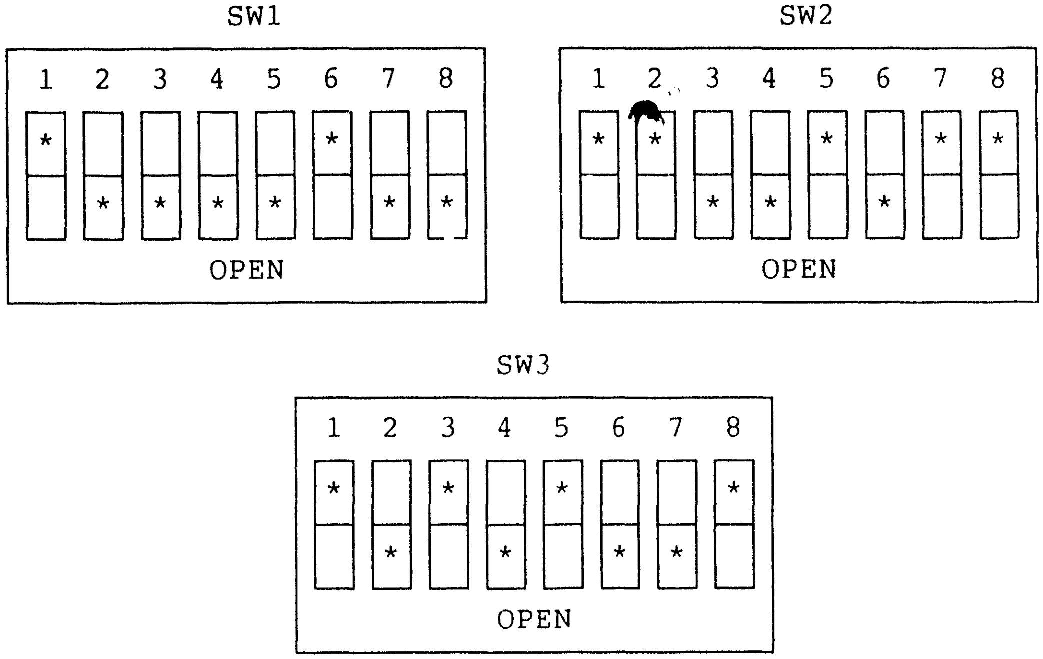


Figure 2.1 488 Dip Switches Layout

2.4 Error Circumstance

Error checking and handling are performed on all operator inputs. An error detected causes the code to either prompt the operator to re-enter the correct option or the code will run the default option as listed in the prompt.

Upon detecting an error, the code returns the identity of the device that failed and the error number the device reported. Error numbers reported by the Computer or the simulator are listed in their respective manuals. Error numbers reported by the 488 Controller are listed in its manual and Appendix C.

Incorrect connections or bad devices will be reported by the code. If the computer has problems assigning the terminal port, the code returns:

```
AUTOSIM error assigning channel to "PORT NAME"
              Return STATUS = ,"STAT",'HEX'

                        or

AUTOSIM error assigning "PORT NAME".
```

where "PORT NUMBER" is the name the code is trying to assign to the computer port, and "STAT" is the error number, in hexadecimal, return status of a DEC SYS$ASSIGN routine.

If the port is assigned correctly, but the 488 Controller is not receiving any commands then the code returns:

```
Error 488 Controller not responding to command
Press Return To Continue <CR>

                  or

AUTOSIM error setting GPIB remote line
Press Return to Continue <CR>
```

These problems could be related to bad wiring, a bad Optical Isolator, a bad 488 Controller or the internal switches of the 488 Controller could be set incorrectly.

When Autosim detects a bad connection to the simulators or a bad simulator, the error number is returned in the Input/Output Status Byte, IOSB. It uses the return status of DEC QIO routine.

```
QIO SYSTEM SERVICE ERROR
IOSB = "error status"
Press Return To Continue
```

This QIO routine is the VAX way of communicating with outside hardware. Error messages returned may be found in the DEC System Services Reference Manual. For example, the most common returned error number is 144 which is a timeout error.

2.5 Instrumentation Database

The Department 9320 Instrumentation Database contains information about the underground test, experimenters, recorders and simulators. The database is created by a Database Administrator and implemented using the Ingres relational database management software package. Access to the database is accomplished via an extensive database application software package written in Fortran and using Ingres database access subroutine libraries.

The database only resides on two networks nodes (primary and backup) and has limited write access. The information is propagated to the rest of the network via conventional VMS direct access flat files called Instrument Control Files (ICFs). ICFs in the past were called Tables. A major part of the database application software creates ICFs from the database tables.

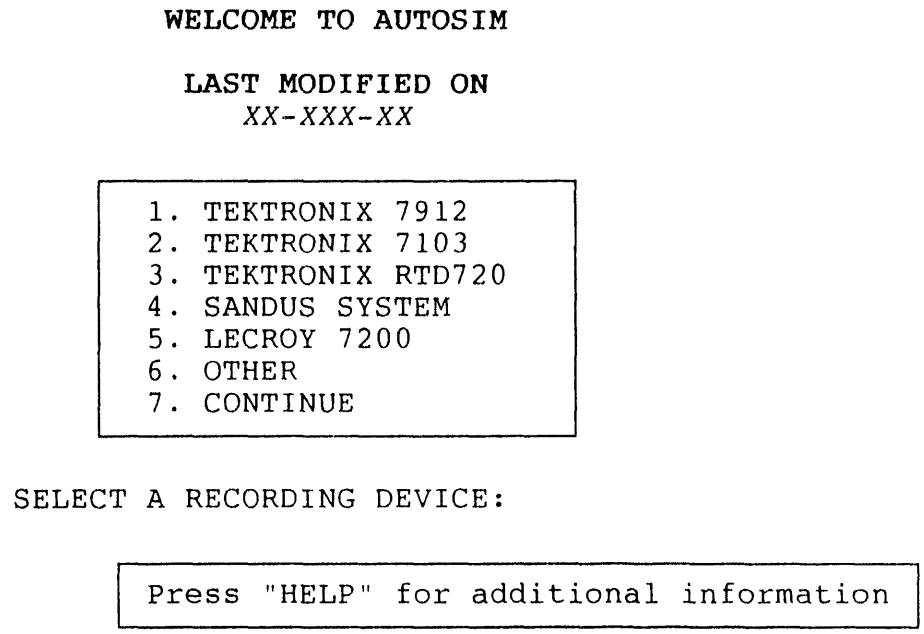
The Database Administrator creates an ICF for simulators called Simwork.tbl. Inside the ICF are records for each simulator that are laid out in a DEC Fortran record structure format. The record name for the each simulator is SIMULATOR_DESC. The complete listing of the simulator's structure with values and definitions that were used in the Diamond Fortune test is listed in Appendix D. When the operator runs Autosim to perform a certain operation on a simulator, the code reads the record of the simulator selected from the channel name entered.

3.0 Getting Started

3.1 Control Account

Autosim runs in a captive account called Control, which is a restricted account as defined by the VMS Operating System and the System Managers. Assuming you have the proper privileges, log into the Control account and then select Autosim from a list of options displayed in the menu. This activates a command file that calls up and runs Autosim. When exiting Autosim, the Control menu is once again displayed and the option "Exit" may be selected to log out from the system.

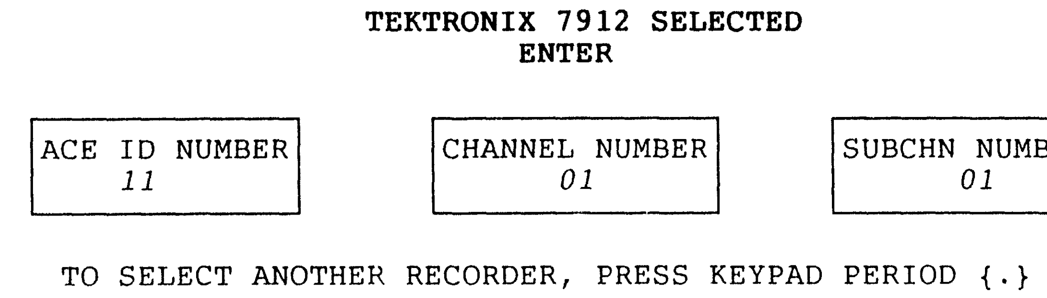
Autosim begins by displaying the Introductory menu with options and then prompts for a response. Autosim menus have a descriptive title located at the top of the display followed by a list of options to be selected. Options are selected by entering the alphanumeric characters listed beside the selection then pressing the "RETURN" key. Pressing the "HELP" key on a VT100 terminal provides additional information on many options. Figure 3.0 shows the Introductory menu that contains a title, the date the code was last updated and a list of 7 options to choose from.



```
                    WELCOME TO AUTOSIM

                     LAST MODIFIED ON
                        XX-XXX-XX

              1. TEKTRONIX 7912
              2. TEKTRONIX 7103
              3. TEKTRONIX RTD720
              4. SANDUS SYSTEM
              5. LECROY 7200
              6. OTHER
              7. CONTINUE

          SELECT A RECORDING DEVICE:

               Press "HELP" for additional information
```

Figure 3.0 Autosim introductory menu

Six of the options involve choosing a simulator by selecting the recorder that the simulator's signal goes to. By reading the channel number entered by the operator, the code will direct all future commands to this simulator until a new channel number is selected. When selecting a recorder, a sub-menu will appear prompting the operator to enter additional information about the selected recorder. Depending on the recorder, information requested could be the Computer node, port or ACE id. This information is used to build a channel name that will match the channel name listed in the Simwork.tbl. Figure 3.1 shows the menu and values that would be entered to give the channel A011-001-01. This notation signifies the recorder is on VAX node name ACE10 (A011), recorder number 1 (001), and the first signal on the trace sub-channel 1 (01).



TEKTRONIX 7912 SELECTED
ENTER

ACE ID NUMBER	CHANNEL NUMBER	SUBCHN NUMBER
11	*01*	*01*

TO SELECT ANOTHER RECORDER, PRESS KEYPAD PERIOD {.}

Figure 3.1 7912 Channel selection

The seventh option is selected when the other functions of Autosim are required such as commanding the laser calibrator.

Figure 3.2 displays the next menu brought up by Autosim, "AUTOSIM ACTION MENU". This menu lists all of Autosim's functions.

```
          AUTOSIM ACTION MENU

         1. INITIALIZE SIMULATOR
         2. MODIFY SIMULATOR PARAMETERS
         3. TRIGGER SYSTEM
         4. LASER CALIBRATOR
         5. DATA ACQUISITION
         6. SELECT ANOTHER CHANNEL
         7. EXIT AUTOSIM

              ENTER OPTION:

                   +----------------------------+
                   | For help, press HELP       |
                   | To continue, enter OPTION  |
                   +----------------------------+
```

Figure 3.2 All Autosim functions

These options may be performed by entering one of the numbers listed beside them, (1 through 7), and pressing "RETURN". The following sections explain the functions of each option.

3.2 Initialize Simulator -- 1

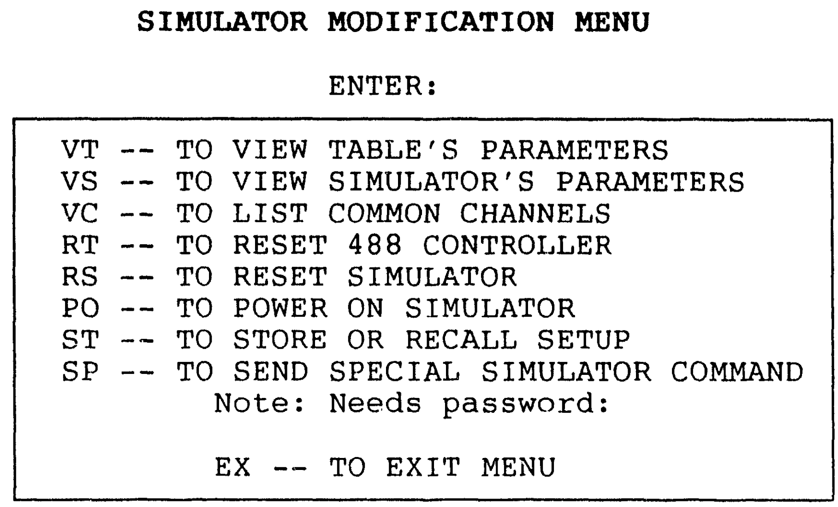
This option brings up a sub-menu as shown in Figure 3.3. Selecting option 1 out of this menu allows the operator to power up and download parameters, read from Simwork.tbl, into the simulator selected. Option 2 also performs option 1 functions but for all the simulators defined in Simwork.tbl. Before downloading parameters, some time is granted for the instrument to complete its power-up test or to stabilize. Option 3 powers off all simulators. Option 4 exits this menu and returns the operator back to the Action menu. Option 2 is the default option; if "RETURN" key is pressed with no value entered this option will be executed.

```
1 = POWER ON AND INITIALIZE SIMULATOR
2 = POWER ON AND INITIALIZE ALL SIMULATORS
3 = POWER OFF ALL SIMULATORS
4 = EXIT MENU

   ENTER OPTION -- DEFAULT = 2
```

Figure 3.3 Initialize options

3.3 Modify simulator's parameters -- 2

This Action menu selection displays the sub-menu, "Simulator Modification Menu" as shown in Figure 3.4. It lists the selections: VT, VS, VC, RT, RS, PO, ST and SP. The following sections describe each of these selections.



```
                SIMULATOR MODIFICATION MENU

                          ENTER:

    VT -- TO VIEW TABLE'S PARAMETERS
    VS -- TO VIEW SIMULATOR'S PARAMETERS
    VC -- TO LIST COMMON CHANNELS
    RT -- TO RESET 488 CONTROLLER
    RS -- TO RESET SIMULATOR
    PO -- TO POWER ON SIMULATOR
    ST -- TO STORE OR RECALL SETUP
    SP -- TO SEND SPECIAL SIMULATOR COMMAND
            Note: Needs password:

            EX -- TO EXIT MENU
```

Figure 3.4 Selections for modifying simulator

3.3.1 View Tables's Parameters -- VT

Brings up a menu showing the parameters read from the record structure and allows the operator the options of viewing the parameters, downloading the parameters or modifying them and then downloading the parameters. Changes made here are not stored in the simulator's record; thus, the next initialization done on the simulator will override value entered in this option. Also, a password is required to download any parameters to any of the simulators.

3.3.2 View Simulator's Parameters -- VS

Reads the present state of the simulator's parameters and displays those values on the terminal screen. No changes are made to the simulator or its record.

3.3.3 List Common Channels -- VC

Reads the simulator's record for the list of channels recording a simulator's signal. These channel numbers are displayed on the terminal screen. This is helpful in knowing which channels are affected by changes in simulator's setting.

3.3.4 Reset 488 Controller -- RT

Resets the 488 controller to its power on condition and sends an interface clear on the GPIB interface. This is helpful in clearing up communications problems.

3.3.5 Reset Simulator -- RS

Sends a reset to the simulator. This sets the simulator into a default or known state depending on the simulator type used. Commands are either a device dependent reset command or a Selected Device Clear, SDC, GPIB command.

3.3.6 Power On Simulator -- PO

List the options to power on, off or view the power state of the selected simulator.

3.3.7 Store or Recall Setup -- ST

List the options to store or recall the stored or front panel setting of the simulator into or out of its non-volatile memory.

3.3.8 Send Special Simulator Command -- SP

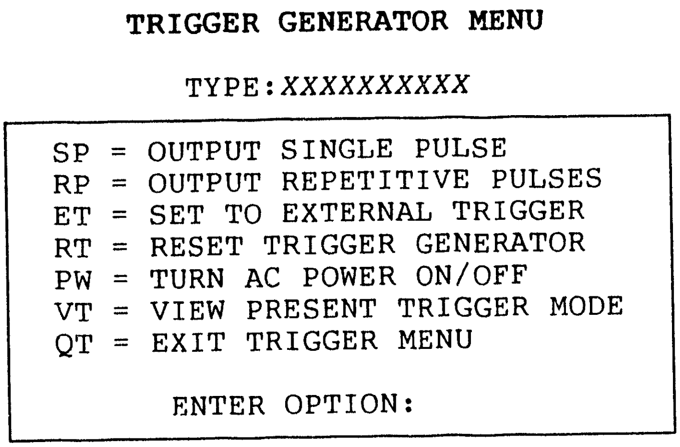
Send any device dependent commands to the selected simulator. A knowledge of the simulator's GPIB command language is required. This is useful in situations where the simulator is required to operate in a unique mode of operation, e.g. in double mode instead of single pulse mode. This option also requires a password to download parameters to the simulators.

3.3.9 Exit Menu -- EX

This option exits out of this sub-menu and returns the operator to the Action menu.

3.4 Trigger System -- 3

This option generates a list of options to command the special Trigger Pulse Generator which delivers the master trigger signal to the external trigger input of the simulators. Figure 3.5 shows the menu and its options. The pulse generator may be commanded to operate in any of the following trigger modes: single, SP; repetitive, RP; or external, ET. Powering the Generator on or reading the status of its state may also be done from this menu. When exiting this menu, Autosim automatically sets the Generator to its dry run state, that is, to its external trigger mode.



```
TRIGGER GENERATOR MENU

TYPE:XXXXXXXXXX

SP = OUTPUT SINGLE PULSE
RP = OUTPUT REPETITIVE PULSES
ET = SET TO EXTERNAL TRIGGER
RT = RESET TRIGGER GENERATOR
PW = TURN AC POWER ON/OFF
VT = VIEW PRESENT TRIGGER MODE
QT = EXIT TRIGGER MENU

     ENTER OPTION:
```

Figure 3.5 Trigger generator menu

3.5 Laser Calibrator -- 4

Figure 3.6 displays the menu to remotely control 9320 Laser Calibrator fiber optics system. This system must be powered on and set up correctly for data to pass through to the recorders.

The Calibrator options require closures to operate or change states. The Hewlett Packard Switch Controller, model 3488, provides these closures. The first column lists the calibrator's operation modes, the middle column lists the only options to choose from for each mode and the last column lists the current condition of each mode. The values now listed in this column represent the dry run state; it is also the state it would be in if the Switch Controller resets or losses AC power.

A password word is needed in order to run this routine. If accidently set up incorrectly, data could be blocked from going through the Laser Calibrator and not recorded on the above ground recorders.

LASER CALIBRATOR

ENTER PASSWORD:

POWER CAL?	Y OR N	*Y*
VIEW SETUP?	Y OR N	*N*
RST HP_CONT?	Y OR N	*N*
LOCKOUT ON?	Y OR N	*N*
MODE?	RMTE OR LCAL	*LCAL*
FREQUENCY?	2HZ,1MHZ,EXTL	*EXTL*
SWITCH POS?	SIG OR CAL	*SIG*

DIRECTIONS

```
Enter -- keypad "0" to trigger 8010a CALIBRATOR PULSER
Enter -- keypad "." to exit menu
```

Figure 3.6 Menu of Laser Calibrator

3.5.1 Power Cal?

Select this option to turn the Calibrator AC power on or off. The choices are "Y" for yes (power on) and "N" for no (power off).

3.5.2 View Setup?

Select this option to view the present state of the Calibrator. The choices are "Y" for yes (view setup) and "N" for no (not to view setup).

3.5.3 Rst Hp_Cont?

Select this option to reset the Switch Controller that supplies the closure to the Calibrator. The choices are "Y" for yes (reset Controller) and "N" for no (not to reset Controller).

3.5.4 Lockout On?

Select this option to lock out the manual controls of the Calibrator. The choices are "Y" for yes (lockout on) and "N" for no (lockout off).

3.5.5 Mode?

Select this option to put the Calibrator in a remote (RMTE) or local (LCAL) mode of operation.

3.5.6 Frequency?

Select this option to determine the signal the Calibrator will output. Inside the Calibrator is a generator that will send a pulse at a rate of 2HZ or 1Mhz if the Calibrator is in the calibrate mode. If the "EXTL" options is selected, the Calibrator is in the pass through mode and all signals at its input will pass through to its output.

3.5.7 Switch Pos?

Select this option to put the Calibrator in the calibrate (CAL) or the signal (SIG) mode. The calibrate mode blocks the signals from the gages or simulator and allows the generator to output the internal generator's signal. The signal mode is the default dry run position, and allows the simulator or gage signal to pass through the unit.

3.6 DATA ACQUISITION -- 5

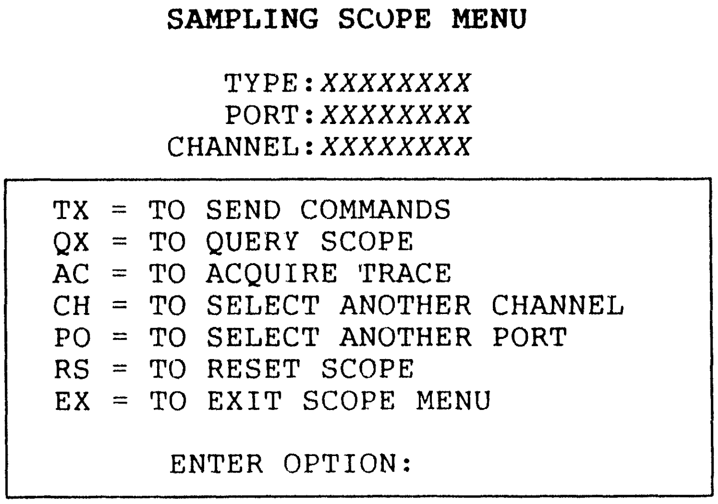
Autosim Digital Acquisition System provides the capability to control and read data from five different types of digitizers. Figure 3.7 lists the digitizers supported by Autosim.

```
            1 = 8751A NETWORK ANALYZER--GPIB=15
            2 = 8753A NETWORK ANALYZER--GPIB=16
            3 = 11801 SAMPLING SCOPE--GPIB=25
            4 = SCD5000 TRANSIENT SCOPE--GPIB=26
            5 = TDS6XX DIGITIZING SCOPE--GPIB=27

     ENTER THE NUMBER OF THE RECORDER TYPE -- DEFAULT = 1
```

Figure 3.7 Recorder Selection

Options 1 and 2 communicate to network analyzers designed by Hewlett Packard. The Tektronix 11801 sampling recorder is a high bandwidth 20GHz sampling scope and requires a repetitive input signal. The SCD5000 transient recorder bandwidth is 4.5GHz and has a fixed input sensitivity level of 4 volts full scale. The TDS620 and TDS640 digitizing recorder bandwidth is 500MHz and extremely portable. For complete specification on the above recorders please refer to the operator manual of the device.

Once the recorder is selected, Autosim brings up the menu as displayed in Figure 3.8. The following sub-sections describe each option listed in Figure 3.8. The digitizer selected appears in the in the menu beside the label "TYPE:".



```
                SAMPLING SCOPE MENU

                   TYPE:XXXXXXXX
                   PORT:XXXXXXXX
                CHANNEL:XXXXXXXX

   TX = TO SEND COMMANDS
   QX = TO QUERY SCOPE
   AC = TO ACQUIRE TRACE
   CH = TO SELECT ANOTHER CHANNEL
   PO = TO SELECT ANOTHER PORT
   RS = TO RESET SCOPE
   EX = TO EXIT SCOPE MENU

         ENTER OPTION:
```

Figure 3.8 Menu of digitizer's functions.

3.6.1 Send Commands -- TX

This option is used to send device dependent commands to the scope. Operators will need to refer to the scope's programmable user guide for a complete listing of all valid commands.

3.6.2 Query Scope -- QX

This option is used to receive information from the digitizers. The "QX" command entered must be a query device dependent command to instruct the digitizer what to send back.

3.6.3 Acquire Trace -- AC

This option is used to acquire data from one of the digitizer's channels. The data returned is stored in a file name "CHNCHANNELNAME.CHN" if the channel name is in the ICF and "INPCHANNELNAME.CHN" if not. In cases where the scope has multiple channels, the code will prompt the operator for which channel to input. When the transfer is completed, the code will then prompt for any attenuators that were used between the digitizer and the signal source. The operator must enter the number 0 for no attenuation or for the last value entered after all attenuation values have been entered.

3.6.4 Select Another Channel -- CH

Autosim uses this channel number to search the ICF for information on the channel number. This information is used as header data on plots attained with the AC option. The channel name entered here is used in the file name acquired with the "AC" option. The channel name entered appears in the menu beside the label "CHANNEL:"

3.6.5 Select Another Port -- PO

This option allows the operator to connect the 488 Controller to any terminal port set up to communicate to Autosim, and run the options listed in Figure 3.7. These ports are located close to the simulators to allow the operator to read in the simulator's signal at its location. The Port name entered appears in the menu beside the label "PORT:".

3.6.6 Reset Scope -- RS

This option sends a reset to the digitizer selected which sets the digitizer into its power-on or default state.

3.6.7 Exit Scope Menu -- EX

This option exits this menu and returns the operator to the Action Menu.

3.7 Select Another Channel -- 6

Use to select or change to another channel number. Channel numbers are used by the code to select the simulator tied to the channel or recorder number.

3.8 Exit Autosim -- 7

Use to stop program executing and exit Autosim code.

4. Software Technical Description

4.1 Introduction

The following sections deliver an in-depth look at the Autosim software package. A knowledge of the fortran language and an understanding of FMS operation are recommended to fully understand the information in these sections. The sections describe the location of the development software, how the Digital FMS package is used in Autosim and the logical flow of the software.

4.2 Software Location

Autosim is written in a modular style with the modules being placed in different subdirectories of the Library Directory, LD:, according to the functions they perform. The five subdirectories are AUTOSIM, FORM, GPIB, HARDWARE and UTIL.

The AUTOSIM section contains the main software source files, executable files and command files. The command file BUILD_ALL_LIBS.COM compiles all ".for" files in all five sections mentioned above and stores the object files in a library file called AUTOSIM.OLB. The file FORLINK.COM links the files in AUTOSIM.OLB and creates an executable file called AUTOSIM.EXE. The file "COMPILE.COM" compiles one file at a time and puts its object file in the AUTOSIM.OLB library.

The FORM section contains all FMS files, the GPIB section contains the read and write software source files for communicating with the 488 Controller, the HARDWARE section contains the software source driver files for each type of simulator Autosim supports and the UTIL section contains the Fortran Include and maintenance software source files.

4.3 Form Management System

VAX-11 Form Management System, FMS version 2.0 is a package from Digital set up to run on VAX/VMS version 3.2 or higher systems. This package creates the video forms used throughout Autosim.

Custom forms are created by calling the FMS editor. Once in the editor, forms are named and designed with information that will be displayed on the screen when called by the program. Autosim forms have a the title of the menu and a list of options to be selected at the prompt. Help is available on most options. To obtain additional information in selecting an option, press the "HELP" key on the terminal. Interfacing between the code and the forms are through fields on the form. Fields names are character variables and have special attributes assigned to them when created.

4.4 Software Flow

This section examines program flow for the main program Autosim.for. The code begins by reading the date into a variable DATE, which is changed by the software developer when any of the software source files are updated. This DATE has the value that will be shown when the Introductory menu is displayed.

Next the FMS environment is defined using a series of calls that set up the workspace, open the library which contains the forms and enables the check routines. Then the Introductory menu is called which lists the options as displayed in Figure 3.0.

The code first initializes a loop terminator IERR, and then executes a DO-WHILE loop to input the option selected from the Introductory menu by the operator. Inside the loop, a subroutine, called CHAN_DECODE, is called which takes the option entered and builds a channel number. The software flow then continues on to the section that brings up the Action menu as shown in Figure 3.2. If option 7 was selected, subroutine CHAN_DECODE is not called and program flow continues to the section that bring up the Action menu.

Once the Action menu is brought up, the code waits for the operator selection. The operator entry is checked to see if it is valid, and if not, a message is displayed and the operator is asked to reenter an option. If a correct entry is made, program flow enters a IF-THEN-ELSEIF-END loop. This loop checks to see which option was selected and then runs that option by calling the subroutine listed following the IF-THEN part of the loop. For example, when the operator selects option 1 from the action menu, the program calls the subroutine SIM_INIT. Each of the subroutines called in Autosim is explained in the following sections. Appendix E contains the Autosim tree created by a DEC package, called Fortran link. This tree shows all routine calls in the Autosim program.

4.5 Functions and Subroutine

This section lists in alphabetical order and explains all the fortran functions and subroutines called in Autosim. For the complete picture of all the calls, refer to Appendix E.

4.5.1 Acq_11801.for

This module resides in the HARDWARE directory and is used to communicate to the Tektronix 11801 Sampling Scope. This routine is called when the operator selects this recorder and then selects the "AC" option out of the SCOPEMENU menu to acquire a trace of data.

4.5.2 Acq_scd.for

This module resides in the HARDWARE directory and is used to communicate with the Tektronix SCD5000 Transient Digitizer. This routine is called when the operator selects the this recorder and then selects the AC option out of the SCOPEMENU menu to acquire a trace of data from it.

4.5.3 Acq_tds.for

This module resides in the HARDWARE directory and is used to communicate to the Tektronix TDS Digitizing Oscilloscope. This routine is called when the operator selects this recorder and then selects the AC option out of the SCOPEMENU menu to acquire a trace of data from the recorder.

4.5.4 Actmenu.frm

This video form resides in the FORM directory and is called by the code when the operator needs to perform some of Autosim functions listed in the Action menu.

4.5.5 Chan_decode.for

This module resides in the AUTOSIM directory and is called to build a channel number to any of the recording devices supported by Autosim. This code examines the variable IRECRD which indicates what type of channel number to build. It builds the channel number by prompting the operator for information about the recorder environment, for example, the Computer node or ACE ID. The code will set the variable IERR to -1 if an error occurred.

4.5.6 Check_err.for

This module resides in the HARDWARE directory and is used to check status from a Simwork.tbl read operation.

4.5.7 Check_fmsstatus.for

This module resides in the HARDWARE directory and is used to check the success or failure of FMS calls.

4.5.8 Ckerstat.for

This module resides in the HARDWARE directory and reads the error status of a simulator after a read operation. The variable TYPE identifies the simulator and the Variable IER contains the success or failure of the read operation.

4.5.9 Ck_fms.for

This module resides in the HARDWARE directory and checks status of FMS calls that set up the FMS environment.

4.5.10 Ck_trm.for

The module resides in the HARDWARE directory and reads the status of a 488 Controller after a write operation to see if it was successful. Successful operation returns 0. Anything else returned identifies an error code as listed in the 488 Controller manual.

4.5.11 Cmgpib.for

This module resides in the GPIB directory and is called by the code to send commands to the 488 Controller.

4.5.12 Comm_channel.inc

This module resides in the UTIL directory and includes declaration and definition of common variables used in routines.

4.5.13 Comm_fmsparm.inc

This module resides in the UTIL directory and includes declaration and definition of common variables used in routines using FMS functions.

4.5.14 Con_time.for

This module resides in the UTIL directory and was written by George Perkins to check frequency ranges and then convert this frequency input into a format suitable for downloading into a Wavetek simulator. For example, if the value entered is 10.0E-9, the routine returns 10N.

4.5.15 Con_volt.for

This module resides in the UTIL directory and was written by George Perkins to check voltage ranges and then convert this voltage input into a format suitable for downloading into a Wavetek simulator. For example, if the value entered is 10.0, the routine returns 10.0V.

4.5.16 Crunch.for

This module resides in the UTIL directory and was written by George Perkins to remove all imbedded blanks from a character string.

4.5.17 Fomenu.frm

This video form resides in the FORM directory and is called by the code when the operator requests the laser calibrator functions.

4.5.18 Fo_trig.for

This module resides in the AUTOSIM directory and is called to communicate to the fiber optic devices. It performs this task by setting switches on a HP Switch Controller. This code displays the Laser Calibrator menu as shown in Figure 3.6. When the operator changes a setting in the menu, the code performs that option which changed.

4.5.19 Gpbuscm.for

This module resides in the GPIB directory and is called by the code to send GPIB commands such as, Selected Device Clear SDC, to a device on the 488 Controller's GPIB bus.

4.5.20 Gpread_cr.for

This module resides in the GPIB directory and is called to read status or data from a device on the 488 Controller's GPIB bus. Code terminates read after it detects a carriage return, <CR>.

4.5.21 Gpread_ct.for

This module resides in the GPIB directory and is called to read status or data from a device on the 488 Controller's GPIB bus. Code terminates read after it reaches a specified byte count.

4.5.22 Gpread_lf.for

This module resides in the GPIB directory and is called to read status or data from a device on the 488 Controller's GPIB bus. Code terminates read after it detects a line feed, <LF>.

4.5.23 Gpwrt.for

This module resides in the GPIB directory and is called to write commands to instruments on the 488 Controller's GPIB bus.

4.5.24 Init_trm.for

This module resides in the HARDWARE directory and resets the 488 Controller to its power on condition.

4.5.25 Lecry.frm

This video form resides in the FORM directory and is called by the code when the operator needs to build a LeCroy channel number.

4.5.26 Length.for

This module resides in the UTIL directory and was written by George Perkins to count the alphanumeric characters in a string.

4.5.27 Main.frm

This video form resides in the FORM directory and is called by the code when the program starts executing.

4.5.28 Modmenu.frm

This video form resides in the FORM directory and is called by the code when the operator needs to modify or initialize a simulator.

4.5.29 Mod_sim.for

This module resides in the AUTOSIM directory and is called to communicate to the simulator selected. It displays the Simulator Modification menu as shown in Figure 3.4 and allows the operator to perform all selections listed in the menu. The code that calls this subroutine must supply the GPIB address of the simulator in the variable GPIB_addr. The code will set the variable IER to -1 if an error occurred.

4.5.30 Other.frm

This video form resides in the FORM directory and is called by the code when the operator needs to build a channel number other then the recorder channel list displayed by Autosim.

4.5.31 Pull_sim_sett.for

This module resides in the AUTOSIM directory and opens the Simwork.tbl to read the record structure in SIMULATOR_DESC of the simulator selected. The code examines each record until a match is found between "CHANNEL", a channel number entered by the operator, and "SIMUL(i).SIM_CHAN(j)", the channel number in the record. The "i" stands for each simulator and the "j" stands for each channel or recorder the simulator sends a signal to. The code will set the variable IER to -1 if an error occurred.

4.5.32 Rdbnc202.for

This module resides in the HARDWARE directory and reads and displays the current condition of the parameters of a Berkeley Nucleonics Corporation Pulse Generator, Model 6040 with a 202h plug-in.

4.5.33 Rddg535.for

This module resides in the HARDWARE directory and reads and displays the current condition of the parameters of a Stanford Research System Inc, Model DG535 Pulse Generator.

4.5.34 Rdmenu.frm

This video form resides in the FORM directory and is called by the code when it displays the parameters read back from a simulator.

4.5.35 Rdwav1560.for

This Dummy module resides in the HARDWARE directory and returns a message stating the Wavetek,model 1560 Pulse Generator is a write only instrument; that is, parameters may not be read from this device.

4.5.36 Read_rtd_chan.for

This module resides in the AUTOSIM directory and is used to communicate to the laser calibrator connected to the RTD channel number selected.

4.5.37 Right_just.for

This module resides in the UTIL directory and was written by George Perkins to right justify a character string.

4.5.38 Sandus.frm

This video form resides in the FORM directory and is called by the code when the operator needs to build a SANDUS channel number.

4.5.39 Scopemenu.frm

This video form resides in the FORM directory and is called by the code when the operator needs to communicate to the digitizers.

4.5.40 Scp_data

This module resides in the AUTOSIM directory and is used to communicate to the digitizers and sampling scopes. It displays the Sampling Scope menu as shown in Figure 3.8 and allows the operator to perform all selections listed in the menu on the digitizers selected. The code will set the variable IER to -1 if an error occurred.

4.5.41 Setup.for

This Module resides in the AUTOSIM directory and is used to set up the GPIB port after a terminal port has been assigned. It checks the link between the Computer Port and 488 Controller, then sets the time out and Remote Enable line of the 488 Controller GPIB port. The code that calls this subroutine must supply a logical unit number and the port name. The code will set the variable IER to -1 if an error occurred.

4.5.42 Sim_init.for

This module resides in the AUTOSIM directory and deals with powering and initializing one or all of the simulators. If the variable DISPLY is set true, the record of the simulator will be displayed where the operator may decide whether or not to download these parameters into the simulator.

4.5.43 Squeeze.for

This module resides in the UTIL directory and was written by George Perkins to replace multiple blanks with a single blank.

4.5.44 Status.inc

This module resides in the UTIL directory and has data type definitions for FMS call routines.

4.5.45 Tek7103.frm

This video form resides in the FORM directory and is called by the code when the operator needs to build a Tektronix 7103 channel number.

4.5.46 Tek7912.frm

This video form resides in the FORM directory and is called by the code when the operator needs to build a Tektronix 7912 channel number.

4.5.47 Tekrtd.frm

This video form resides in the FORM directory and is called by the code when the operator needs to build a Tektronix RTD 720 channel number.

4.5.48 Trig_gen.for

This module resides in the AUTOSIM directory and calls the trigger menu to list the available options that are involved in powering and triggering the Generator responsible for delivering the trigger signal to the simulator. One option sets the Generator in a repetitive mode. The rate selected will not trigger the simulator fast enough to damage the fiber optics. Upon exiting this module, the generator is automatically set to external trigger mode which is the dry run setting.

4.5.49 Trigmenu.frm

This video form resides in the FORM directory and is called by the code when the operator needs to communicate to the master trigger generator.

4.5.50 Trmread.for

This module resides in the GPIB directory and sends commands to the 488 Controller.

4.5.51 Wr1560.for

This module resides in the HARDWARE directory and reads the Simwork.tbl parameters for a Wavetek, Model 1560 Pulse Generator. Values are displayed in the menu WRMENU, where parameters may be edited and then downloaded to the simulator.

4.5.52 Wrbnc202.for

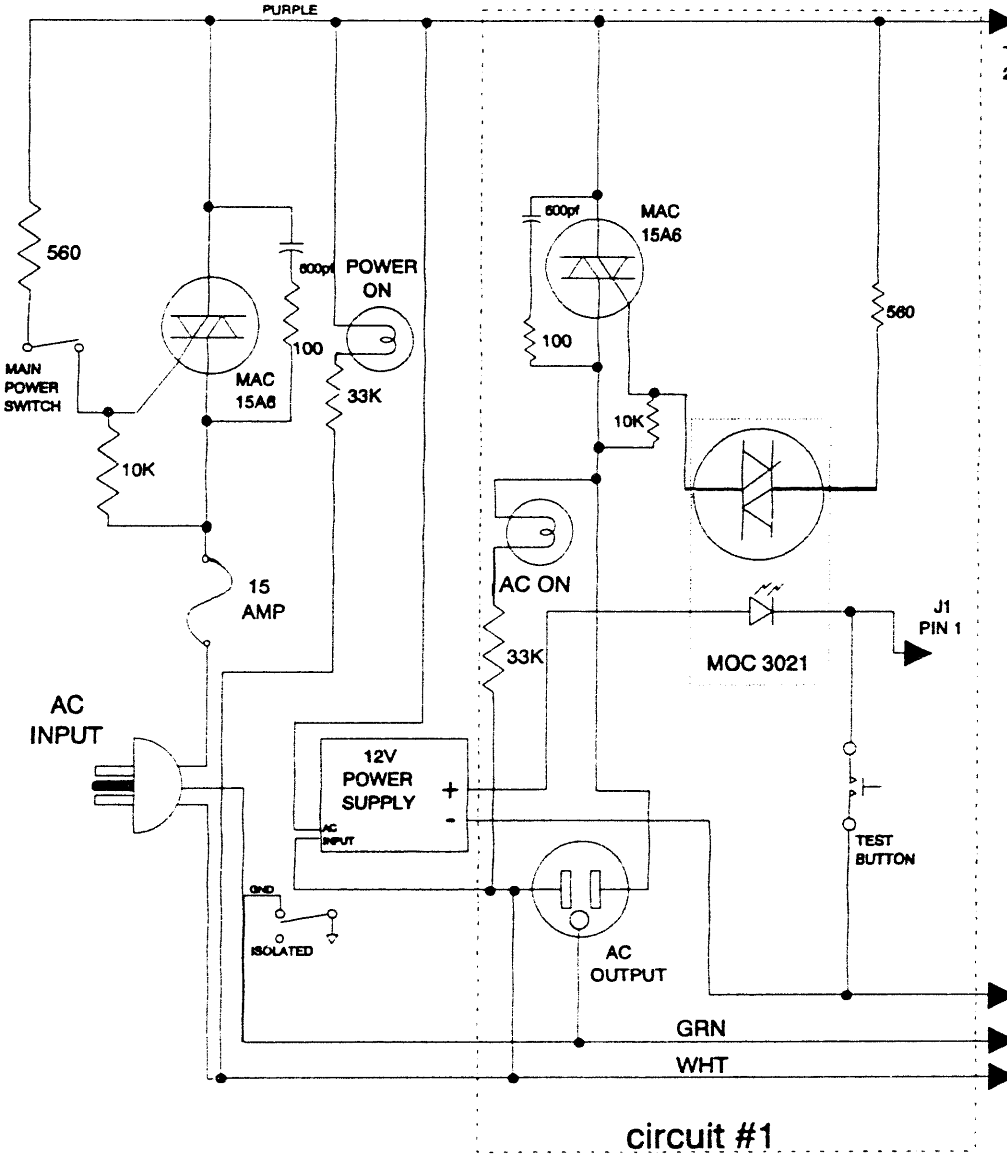
This module resides in the HARDWARE directory and reads the ICF parameters for a Berkeley Nucleonics Corporation Pulse Generator, Model 6040 with a 202h plug-in module. Values are displayed in a menu, where parameters may be edited then downloaded to the simulator.

4.5.53 Wrdg535.for

This module resides in the HARDWARE directory and reads the ICF parameters for a Stanford Research System Inc, Model DG535 Pulse Generator. Values are displayed in the menu WRMENU, where parameters may be edited and then downloaded to the simulator.

4.5.54 Wrmenu.frm

This video form resides in the FORM directory and is called by the code when it writes the parameters from Simwork.tbl to a simulator.

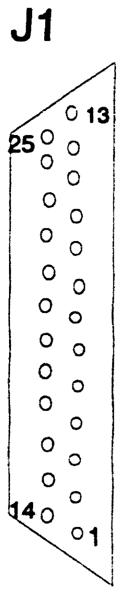


Designed by D.E. Banto
EG&G
Januuary 28, 1992

NOTE 1: CIRCUIT #1 is duplicated 10 times for individual control of 10 devices.

NOTE 2: Closure applied to J1 pin 1 and pin 14 activates AC power to device connected to circuit 1.

J1 PIN#	Circuit #
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10



APPENDIX A POWER CONTROLLER

APPENDIX B - Terminal Server Parameters

The following list of parameters are copied from a terminal server port set to run the Autosim hardware.

```
!DEFINE PORT 11 WHICH IS MAPPED TO LTA811: ON SERVER VCCDH1


DEFINE PORT 11 CHARACTER SIZE SPEED 19200 FLOW CONTROL DIS PARITY
               ODD NODEM CONTROL DISABLED
DEFINE PORT 11 ACCESS REMOTE LOCAL SWITCH NONE BACKWARDS SWITCH
               NODE NAME AUTOSIM_11
DEFINE PORT 11 BREAK DISABLED FORWARD SWITCH NONE TYPE SOFT
DEFINE PORT 11 AUTOBAUD DISABLED AUTOPROMT DISABLED BROADCAST
               DISABLED INACTIVITY LOGOUT DISABLED
DEFINE PORT 11 INPUT FLOW CONTROL ENABLED LOSS NOTIFICATION
               DISABLED MESSAGE CODED DISABLED
DEFINE PORT 11 OUTPUT FLOW CONTROL DISABLED VERIFICATION DISABLED
```

APPENDIX C - 488 Error Messages

Error No. Error Text and Description

00 OK

01 INVALID ADDRESS
Caused by an invalid address outside the allowable IEEE 488 bus range of 00 to 30 for primary addresses and 00 to 31 for secondary addresses.

02 INVALID COMMAND
Caused by an unrecognized command or invalid parameter.

03 WRONG MODE
Caused by trying to execute a command not allowed within the present state of the interface. (e.g. REMOTE16 as a peripheral)

04 Unassigned - Reserved

05 Unassigned - Reserved

06 NO MACRO
A DOMACRO or READ command was received but the MACRO buffer specified is empty.

07 MACRO OVERFLOW
No memory is available to allocate as a MACRO buffer.

08 COMMAND OVERFLOW
More than 127 characters were received and interpreted as a command.

09 ADDRESS OVERFLOW
More than 15 primary address/secondary address pairs were received.

10 MESSAGE OVERFLOW
No memory is available to buffer the received data of the OUTPUT command.

11 NOT A TALKER
As the Active Controller, an un-addressed OUTPUT, a SEND DATA, or a SEND CMD was received and the 488 Controller was not in the Talk Addressed State.

12 NOT A LISTENER
As the Active Controller, an un-addressed ENTER or a SEND ENTER was received and the 488 CONTROLLER was not in the Listen Addressed State.

13 BUS ERROR
The 488 Controller tried to output data to the bus but there was no active listener to accept it.

14 TIMEOUT - WRITE
The specified TIME OUT time has elapsed before the last command or data byte sent by the 488 Controller was accepted by an external device.

15 TIMEOUT - READ
The specified TIME OUT time has elapsed while the 488 Controller was waiting for a data byte from an external device.

16 OUT OF MEMORY
The 488 Controller was unable to perform the last command requested due to the lack of sufficient memory in the USED heap.

17 MACRO RECURSION
A DOMACRO command specified a Macro buffer that was already executing.

APPENDIX D - Autosim Record Structure

The following listing describes the record structure of a simulator. There is one record of this type for every simulator.

```
structure /SIMULATOR_DESC/
        integer*2         struct_rec_num       !Structure record number
        byte              struct_rec_type      !Structure record type(12)
        byte              dummy                !Dummy field (not used)

        character*16      sim_unique_id        !Unique identification
        character*15      sim_model            !Model description
        character*12      sim_arm_mode         !Arming/Control mode
        character*12      sim_location         !Simulator location
        character*12      sim_vax_node         !VAX Port/Node name
        character*12      sim_ts_node          !Terminal server node name
        character*12      sim_trg_source       !Trigger source
        character*12      sim_trg_mode         !Trigger mode
        character*12      sim_trg_slope        !Trigger slope
        real*4            sim_trg_level        !Trigger level
        real*4            sim_trg_delay        !Trigger delay
        real*4            sim_thruput          !Simulator thruput delay
        logical*4         sim_term_50          !Simulator termination

        real*4            sim_baseline         !Baseline voltage
        real*4            sim_peak_v           !Peak voltage
        real*4            sim_pul_width        !Pulse width
        real*4            sim_pul_rise         !Pulse risetime
        real*4            sim_pul_fall         !Pulse falltime
        real*4            sim_pul_area         !Pulse area
        real*4            sim_pul_delay        !Pulse delay
        real*4            sim_pul_atten        !Pulse attenuation
        real*4            sim_pul_period       !Pulse period
        character*12      sim_polarity         !Pulse polarity
        real*4            sim_max_volts        !Maximum voltage
        real*4            sim_max_width        !Maximum pulse width
        real*4            sim_max_delay        !Maximum pulse delay
        real*4            sim_max_offset       !Maximum offset
        real*4            sim_max_period       !Maximum period
        integer*4         sim_x_menu           !Memory recall number
        integer*4         sim_gpib_addr        !Unique GPIB number
        integer*4         sim_pwr_switch       !Turn on/off power
        integer*4         num_channels         !Number of channel
        character*12      sim_chan(16)         !Channel array
        character*12      sim_tgen_model       !Trigger generator model
        character*12      sim_tgen_loc         !Trig. gen. location
        character*12      sim_trg_node         !Trig. gen. server
        integer*4         sim_tgen_gpib        !Trig. gen. Gpib addr
        integer*4         sim_tgen_p_sw        !Trig. gen. power mode
        real*4            sim_tgen_volts       !Trig. gen. output voltage
        real*4            sim_tgen_width       !Trig. gen. pulse width
        real*4            sim_tgen_delay       !Trig. gen. delay
        real*4            sim_tgen_rate        !Trig. gen. pulse period
        character*24      last_edit            !Last edit
 end structure
```

APPENDIX E - Autosim Software Calls

The following is a listing of Autosim and all of Autosim software calls. It was generated by a DEC package called Fortran Link. This DEC package examines each routine and displays all calls made by that routine. Routines that are called more then once are assigned a reference number enclosed in parentheses.

```
This is a primary tree starting at the program 'AUTOSIM'

AUTOSIM-+-(FDV$ATERM)
        |
        +-(FDV$AWKSP)
        |
        +-(FDV$CDISP)
        |
        +-(FDV$CLEAR)
        |
        +-(FDV$CLEAR_VA)
        |
        +-(FDV$DTERM)
        |
        +-(FDV$GET)
        |
        +-(FDV$LCLOS)
        |
        +-(FDV$LOPEN)
        |
        +-(FDV$PUT)
        |
        +-(FDV$PUTL)
        |
        +-(FDV$SPADA)
        |
        +-(FDV$SSRV)
        |
        +-TRIG_GEN-+-(FDV$CDISP)
        |          |
        |          +-(FDV$CLEAR)
        |          |
        |          +-(FDV$CLEAR_VA)
        |          |
        |          +-(FDV$GET)
        |          |
        |          +-(FDV$PUT)
        |          |
        |          +-(FDV$PUTL)
        |          |
        |          +-CK_TRM (1)-+-(LIB$FREE_EF)
        |          |            |
        |          |            +-(LIB$GET_EF)
        |          |            |
        |          |            |
        |          |            +-(SYS$QIOW)
        |          |            |
        |          |            +-LENGTH
        |          |            |
        |          |            +-(FDV$CLEAR)
        |          |            |
        |          |            +-(FDV$CLEAR_VA)
        |          |
        |          +-(SYS$ASSIGN)
        |          |
        |          +-(SYS$DASSGN)
        |          |
        |          +-SETUP (2)-+-CMGPIB (3)-+-(LIB$FREE_EF)
        |          |           |            |
        |          |           |            +-(LIB$GET_EF)
        |          |           |            |
        |          |           |            +-(SYS$READEF)
        |          |           |            |
        |          |           |            +-(SYS$QIO)
        |          |           |            |
        |          |           |            +-LENGTH
        |          |           |
|           |            |            |
|           |            |            +-(LIB$WAIT)
|           |            |            |
|           |            |            +-(FDV$CLEAR)
|           |            |            |
|           |            |            +-(FDV$CLEAR_VA)
|           |            |
|           |            +-TRMREAD-+-(LIB$FREE_EF)
|           |            |         |
|           |            |         +-(LIB$GET_EF)
|           |            |         |
|           |            |         +-(SYS$QIOW)
|           |            |         |
|           |            |         +-LENGTH
|           |            |
|           |            +-(FDV$CLEAR)
|           |            |
|           |            +-(FDV$CLEAR_VA)
|           |            |
|           |            +-(LIB$WAIT)
|           |
|           +-CHECK_FMSSTATUS  (4)--(FDV$DTERM)
|           |
|           +-CRUNCH  (5)--LENGTH
|           |
|           +-GPWRT  (6)-+-(LIB$FREE_EF)
|           |            |
|           |            +-(LIB$GET_EF)
|           |            |
|           |            +-(SYS$QIOW)
|           |            |
|           |            +-LENGTH
|           |            |
|           |            +-(FDV$CLEAR)
|           |            |
|           |            +-(FDV$CLEAR_VA)
|           |
|           +-GPREAD_LF  (7)-+-(LIB$FREE_EF)
|           |                |
|           |                +-(LIB$GET_EF)
|           |                |
|           |                +-(SYS$QIOW)
|           |                |
|           |                +-LENGTH
|           |                |
|           |                +-(FDV$CLEAR)
|           |                |
|           |                +-(FDV$CLEAR_VA)
|           |
|           +-CKERSTAT  (8)-+-(LIB$FREE_EF)
|           |               |
|           |               +-(LIB$GET_EF)
|           |               |
|           |               +-(SYS$QIOW)
|           |               |
|           |               +-CK_TRM  see  1
|           |               |
|           |               +-LENGTH
|           |               |
|           |               +-GPWRT  see  6
|           |               |
|           |               +-(FDV$CLEAR)
|           |               |
|           |               +-(FDV$CLEAR_VA)
|           |
|           +-(LIB$WAIT)
|
+-CK_FMS-+-(FDV$STAT)
|        |
|        +-CHECK_FMSSTATUS  see  4
|
+-CHECK_FMSSTATUS  see  4
|
+-CHAN_DECODE  (9)-+-(FDV$CDISP)
|                  |
|                  +-(FDV$GET)
|                  |
|                  +-(FDV$PUT)
|
|                  |
|                  +-(FDV$PUTL)
|                  |
|                  +-LENGTH
|                  |
|                  +-CHECK_FMSSTATUS  see 4
|                  |
|                  +-CRUNCH  see 5
|
+-PULL_SIM_SETT  (10)-+-(LIB$FREE_EF)
|                     |
|                     +-(LIB$GET_EF)
|                     |
|                     +-CHECK_ERR--(LIB$WAIT)
|                     |
|                     +-(OPEN_TBL)
|                     |
|                     +-(CLOSE_TBL)
|                     |
|                     +-(READ_TBL_REC)
|                     |
|                     +-(BAD_STATUS)
|
+-SIM_INIT  (11)-+-(FDV$CLEAR)
|                |
|                +-(FDV$CLEAR_VA)
|                |
|                +-CK_TRM  see 1
|                |
|                +-LENGTH
|                |
|                +-(SYS$ASSIGN)
|                |
|                +-(SYS$DASSGN)
|                |
|                +-PULL_SIM_SETT  see 10
|                |
|                +-SETUP  see 2
|                |
|                +-GPWRT  see 6
|                |
|                +-(LIB$WAIT)
|                |
|                +-WRDG535-+-(FDV$CDISP)
|                |         |
|                |         +-(FDV$CLEAR)
|                |         |
|                |         +-(FDV$CLEAR_VA)
|                |         |
|                |         +-(FDV$GET)
|                |         |
|                |         +-(FDV$PUT)
|                |         |
|                |         +-(FDV$PUTL)
|                |         |
|                |         +-CK_TRM  see 1
|                |         |
|                |         +-LENGTH
|                |         |
|                |         +-(LIB$WAIT)
|                |         |
|                |         +-CHECK_FMSSTATUS  see 4
|                |         |
|                |         +-CRUNCH  see 5
|                |         |
|                |         +-GPWRT  see 6
|                |         |
|                |         +-CKERSTAT  see 8
|                |
|                +-WR1560-+-(FDV$CDISP)
|                |        |
|                |        +-(FDV$CLEAR)
|                |        |
|                |        +-(FDV$CLEAR_VA)
|                |        |
|                |        +-(FDV$GET)
|                |        |
|                |        +-(FDV$PUT)
|                |
|                |        |
|                |        +-CK_TRM  see  1
|                |        |
|                |        +-LENGTH
|                |        |
|                |        +-(LIB$WAIT)
|                |        |
|                |        +-CHECK_FMSSTATUS  see  4
|                |        |
|                |        +-CRUNCH  see  5
|                |        |
|                |        +-CON_TIME
|                |        |
|                |        +-CON_VOLT
|                |        |
|                |        +-GPWRT  see  6
|                |
|                +-WRBNC202-+-(FDV$CDISP)
|                           |
|                           +-(FDV$CLEAR)
|                           |
|                           +-(FDV$CLEAR_VA)
|                           |
|                           +-(FDV$GET)
|                           |
|                           +-(FDV$PUT)
|                           |
|                           +-(FDV$PUTL)
|                           |
|                           +-CK_TRM  see  1
|                           |
|                           +-LENGTH
|                           |
|                           +-CRUNCH  see  5
|                           |
|                           +-(LIB$WAIT)
|                           |
|                           +-CHECK_FMSSTATUS  see  4
|                           |
|                           +-SQUEEZE--LENGTH
|                           |
|                           +-GPWRT  see  6
|
+-MOD_SIM-+-(FDV$CDISP)
|         |
|         +-(FDV$CLEAR)
|         |
|         +-(FDV$CLEAR_VA)
|         |
|         +-(FDV$GET)
|         |
|         +-(FDV$PUTL)
|         |
|         +-(SYS$ASSIGN)
|         |
|         +-(SYS$DASSGN)
|         |
|         +-CK_TRM  see  1
|         |
|         +-GPBUSCM-+-(LIB$FREE_EF)
|         |         |
|         |         +-(LIB$GET_EF)
|         |         |
|         |         +-(SYS$QIOW)
|         |         |
|         |         +-LENGTH
|         |         |
|         |         +-(FDV$CLEAR)
|         |         |
|         |         +-(FDV$CLEAR_VA)
|         |
|         +-LENGTH
|         |
|         +-(LIB$WAIT)
|         |
|         +-SETUP  see  2
|         |
|         +-CHECK_FMSSTATUS  see  4
|
|         |
|         +-SIM_INIT  see  11
|         |
|         +-RDDG535-+-(FDV$CDISP)
|         |         |
|         |         +-(FDV$CLEAR)
|         |         |
|         |         +-(FDV$CLEAR_VA)
|         |         |
|         |         +-(FDV$PUT)
|         |         |
|         |         +-LENGTH
|         |         |
|         |         +-CK_TRM  see  1
|         |         |
|         |         +-GPWRT  see  6
|         |         |
|         |         +-GPREAD_CR  (12)-+-(LIB$FREE_EF)
|         |         |                 |
|         |         |                 +-(LIB$GET_EF)
|         |         |                 |
|         |         |                 +-(SYS$QIOW)
|         |         |                 |
|         |         |                 +-LENGTH
|         |         |                 |
|         |         |                 +-(FDV$CLEAR)
|         |         |                 |
|         |         |                 +-(FDV$CLEAR_VA)
|         |         |
|         |         +-CKERSTAT  see  8
|         |         |
|         |         +-CHECK_FMSSTATUS  see  4
|         |
|         +-RDWAV1560
|         |
|         +-RDBNC202-+-(FDV$CDISP)
|         |          |
|         |          +-(FDV$CLEAR)
|         |          |
|         |          +-(FDV$CLEAR_VA)
|         |          |
|         |          +-(FDV$PUT)
|         |          |
|         |          +-CK_TRM  see  1
|         |          |
|         |          +-LENGTH
|         |          |
|         |          +-GPWRT  see  6
|         |          |
|         |          +-GPREAD_CR  see  12
|         |          |
|         |          +-CKERSTAT  see  8
|         |          |
|         |          +-CHECK_FMSSTATUS  see  4
|         |
|         +-PULL_SIM_SETT  see  10
|         |
|         +-INIT_TRM-+-CMGPIB  see  3
|         |          |
|         |          +-(SYS$ASSIGN)
|         |          |
|         |          +-(SYS$DASSGN)
|         |          |
|         |          +-(LIB$WAIT)
|         |
|         +-GPWRT  see  6
|         |
|         +-GPREAD_LF  see  7
|         |
|         +-CKERSTAT  see  8
|
+-FO_TRIG-+-(FDV$CDISP)
|         |
|         +-(FDV$CLEAR)
|         |
|         +-(FDV$CLEAR_VA)
|         |
|         +-(FDV$GET)
|         |
|         +-(FDV$ILTRM)
|         |
|         +-(FDV$PUT)
|         |
|         +-(FDV$PUTL)
|         |
|         +-(SYS$ASSIGN)
|         |
|         +-(SYS$DASSGN)
|         |
|         +-CK_TRM  see  1
|         |
|         +-READ_RTD_CHAN--RIGHT_JUST--LENGTH
|         |
|         +-(LIB$WAIT)
|         |
|         +-GPWRT  see  6
|         |
|         +-GPREAD_LF  see  7
|         |
|         +-CKERSTAT  see  8
|         |
|         +-CHECK_FMSSTATUS  see  4
|
+-SCP_DATA-+-(FDV$CDISP)
           |
           +-(FDV$GET)
           |
           +-(FDV$PUT)
           |
           +-(FDV$PUTL)
           |
           +-CK_TRM  see  1
           |
           +-LENGTH
           |
           +-(SYS$ASSIGN)
           |
           +-(SYS$DASSGN)
           |
           +-CHECK_FMSSTATUS  see  4
           |
           +-(LIB$WAIT)
           |
           +-SETUP  see  2
           |
           +-GPWRT  see  6
           |
           +-GPREAD_LF  see  7
           |
           +-ACQ_SCD-+-CK_TRM  see  1
           |         |
           |         +-(FDV$CLEAR)
           |         |
           |         +-(FDV$CLEAR_VA)
           |         |
           |         +-GPWRT  see  6
           |         |
           |         +-GPREAD_LF  see  7
           |         |
           |         +-LENGTH
           |         |
           |         +-GPREAD_CT-+-(LIB$FREE_EF)
           |         |           |
           |         |           +-(LIB$GET_EF)
           |         |           |
           |         |           +-(SYS$QIOW)
           |         |           |
           |         |           +-LENGTH
           |         |           |
           |         |           +-(FDV$CLEAR)
           |         |           |
           |         |           +-(FDV$CLEAR_VA)
           |         |
           |         +-OUTPUT_ENG(13)-+-LOOKUP-+-(OPEN_TBL)
           |                          |        |
           |                          |        +-(READ_TBL_REC)
```

```
                    |                           |         |
                    |                           |         +-(CLOSE_TBL)
                    |                           |         |
                    |                           |         +-(STR$UPCASE)
                    |                           |         |
                    |                           |         +-(LIB$GETJPI)
                    |                           |         |
                    |                           |         +-(LIB$FIND_FILE)
                    |                           |         |
                    |                           |         +-LENGTH
                    |                           |
                    |                           +-(DATE)
                    |                           |
                    |                           +-(TIME)
                    |                           |
                    |                           +-LENGTH
                    |
                    +-ACQ_TDS-+-CK_TRM  see 1
                    |         |
                    |         +-(FDV$CLEAR)
                    |         |
                    |         +-(FDV$CLEAR_VA)
                    |         |
                    |         +-GPWRT  see 6
                    |         |
                    |         +-GPREAD_LF  see 7
                    |         |
                    |         +-LENGTH
                    |         |
                    |         +-OUTPUT_ENG  see 13
                    |
                    +-ACQ_11801-+-CK_TRM  see 1
                    |           |
                    |           +-(FDV$CLEAR)
                    |           |
                    |           +-(FDV$CLEAR_VA)
                    |           |
                    |           +-GPWRT  see 6
                    |           |
                    |           +-GPREAD_LF  see 7
                    |           |
                    |           +-LENGTH
                    |           |
                    |           +-OUTPUT_ENG  see 13
                    |
                    +-CHAN_DECODE  see 9
This is a detached tree starting at the subprogram 'RJOFIL'

RJOFIL-+-LEFT_JUSTIFY
       |
       +-LENGTH

This is a detached tree starting at the subprogram 'GPREAD_CT'

GPREAD_CT-+-(LIB$FREE_EF)
          |
          +-(LIB$GET_EF)
          |
          +-(SYS$QIOW)
          |
          +-LENGTH
          |
          +-(FDV$CLEAR)
          |
          +-(FDV$CLEAR_VA)
```

Distribution:

2 Defence Nuclear Agency
Attn: FCTIU (Kiran Shah)
4420 Mitchell St.
North Las Vegas, NV 89031

1	9324-1/MS1391	W.J. Kluesner
1	9324-1/MS1391	D.S. Nelson
1	9324-1/MS1391	R.L. Wilson
1	9324-1/MS1391	B. Betz(EG&G)
1	9324-1/MS1391	E.W. Marsh(EG&G)
1	9324-1/MS1391	B.J. Wohlbrandt(EG&G)

5	7141/MS0899	Technical Library
1	7151/MS0619	Technical Publications
10	7613-2/MS0100	Document Processing for DOE/OSTI
1	8523-2/MS9018	Central Technical Files
1	9305/MS1170	M.J. Navratil
1	9321/MS1168	W.B. Boyer
1	9321/MS1168	R.D. Aden
12	9321/MS1168	E.D. Baker
1	9321/MS1168	R.B. Caudell
1	9321/MS1168	T.L. Downey
1	9321/MS1168	P.C. Kaestner
1	9321/MS1168	R.L. Kinchen
1	9321/MS1168	J.W. Lee
1	9321/MS1168	L. Livingston
1	9321/MS1168	P.F. McKay
1	9321/MS1168	R.A. Mills
1	9321/MS1168	J.L. Romine
1	9321/MS1168	L. Bishop(Diversus)
1	9321/MS1168	D. Geuss(Bell Atlantic)
1	9321/MS1168	R. Salas(EG&G)
1	9321/MS1168	M. Ward(EG&G)
1	9322/MS1169	C.W. Cook
1	9322/MS1169	R.E. French
1	9322/MS1169	G.J. Hansen
1	9322/MS1169	E.S. Herrera
1	9322/MS1169	R.G. Oliveira
1	9322/MS1169	F. Salas
1	9323/MS1168	G.L. Ogle
1	9323/MS1168	B.C. Bedeaux
1	9323/MS1168	D.C. Browne
1	9323/MS1168	R. Davis
1	9323/MS1168	J.W. McKeever
1	9323/MS1168	P.L. Wehrman
1	9324/MS1391	P.L. Nelson
1	9324/MS1391	D.J. Belding
1	9324-1/MS1391	F.M. Raymond
1	9324-1/MS1391	J.A. Chael

END

DATE
FILMED

3/30/94