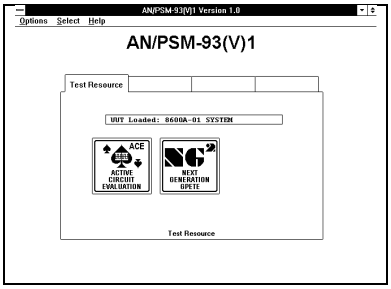
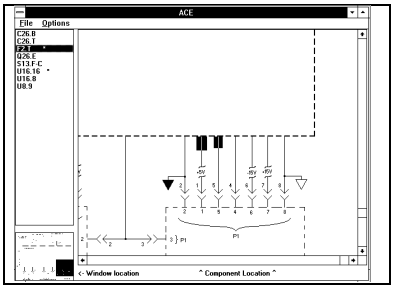
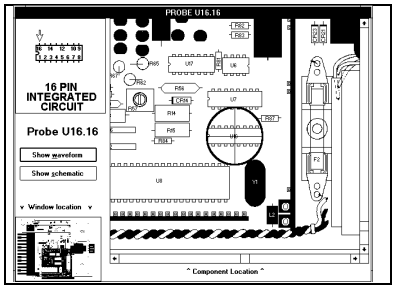
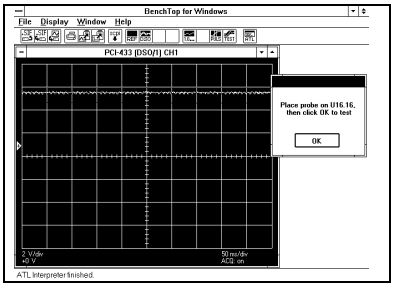
AN/PSM-93(V)
Computer Test Set
Developer's Guide









AN/PSM-93(V)
Computer Test Set
Developer's Guide
Version 1.4

TABLE OF CONTENTS

TABLE OF CONTENTS, CONTINUED

SECTION 1

INTRODUCTION

1.1 Purpose. This guide is designed for **S**ignal **E**valuation **S**et (**SES**) developers using the *AN/PSM-93(V) Computer Test Set* . It will lead the user through the various application programs and software packages needed to develop complete and useful SESs for a given **U**nit **U**nder **T**est (UUT). This guide was written for someone using the AN/PSM-93(V) who has little or no knowledge of computers. An intermediate or advanced user may not need to reference all of the topics found in this guide. This guide outlines recommended and suggested procedures and formats for the successful development of fleet deployable SESs by an end user. The guide incorporates information and programs designed specifically for the AN/PSM-93(V) Computer Test Set by the Crane Division of the Naval Surface Warfare Center (Code 6037), Crane, IN, and other commercial software from manufacturers.

1.2 AN/PSM-93(V) Description. The AN/PSM-93(V) Computer Test Set is essentially a Portable Maintenance Aid (PMA), whose main components are the **A**ctive **C**ircuit **E**valuation (**ACE**) software package, and **N**ext **G**eneration **G**PETE (**NG²**) instrumentation hardware. It was designed and developed to function as a powerful suite of integrating test software and PC based test instrumentation. Although the main focus is on PC based instruments, the test set can be configured to control and provide for the utilization of other test hardware such as VXI, IEEE, and standard bench top **G**eneral **P**urpose **E**lectronic **T**est **E**quipment (**GPETE**), as well as to be incorporated into, and work in conjunction with, other **A**utomatic **T**est **E**quipment (**ATE**). The AN/PSM-93(V) can also function as a self-contained paperless storage and retrieval system, allowing access to thousands of pages of information, drawings, and test programs stored on CD-ROM, such as Interactive Electronic Technical Manuals (ITEMs).

1.3 AN/PSM-93(V) Purpose and Test Philosophy. The purpose of the AN/PSM-93(V) is to aid test, repair, and maintenance personnel in the efficient and intelligent performance of their duties by providing them with a highly portable, yet powerful combination of test instrumentation, reference waveforms, measurement data, and technical documentation bundled together in the form of **S**ignal **E**valuation **S**ets (**SESs**). The SESs are comprised primarily of UUT Technical Information and UUT Reference Waveforms. The UUT Technical Information is specific to the UUT and is comprised of, at minimum, UUT SCHEMATIC DIAGRAMS, a UUT PARTS LAYOUT DRAWING, and a UUT PARTS LIST. The UUT Reference Waveforms can be either AC or DC waveforms collected with a PC based storage oscilloscope from user specified points or nodes in a given circuit, electrical assembly, system, or sub-system. The test philosophy employed in the AN/PSM-93(V) via the Active Circuit Evaluation (ACE) software package is as follows: UUT Reference Waveforms are to be collected from a known good UUT or system that is operating in a known, repeatable, and technically relevant, power-on state. These stored, or "golden" reference signals will then be utilized by the users as baseline test signals to compare with the signals taken from a failed or suspect UUT. By comparing these reference signals collected from a known good UUT to those of a failed or suspect UUT, the operator should be able to locate, isolate, and repair failures present on the failed UUT.

1.4 AN/PSM-93(V) Developer's Guide Overview. The AN/PSM-93(V) Developer's Guide is designed to take the user through the steps necessary to develop a SES, in the order in which they should be completed. The guide is therefore broken down into sections covering these various steps in the development process. If the user still has difficulty with the AN/PSM-93(V), or the development of SESs, Section 1.11 contains support personnel contact information.

1.5 Computer Requirements. Listed below are the **recommended minimum** computer requirements for running the AN/PSM-93(V):

Computer Configuration, *Software*

Microsoft MS-DOS 6.0.2 Microsoft Windows 3.11

Computer Configuration, *Hardware*

80486DX4 PC (100MHz)	2 serial ports
16MB RAM (1MB Video RAM)	1 parallel port
1024x768, 256 color VGA	101 key extended keyboard
500 MB hard disk drive	10” VGA display
3.5” 1.44 MB floppy drive	CD-ROM drive
mouse, or integral pointing device	external VGA port

1.6 AN/PSM-93(V) Software. The following software will be supplied and installed into each AN/PSM-93(V) assigned to fleet activities:

a. CD-ROM: M-31170, AN/PSM-93(V), Computer Test Set

b. 5 ¼” Diskette: Digital Multimeter Software Installation Disk (5 ½ Digit DMM Control Software)

1.7 Mouse. The AN/PSM-93(V) is primarily a Windows® based system. This means that to run the AN/PSM-93(V), the operator uses a mouse to point to an application icon (a picture representing the application) he or she wishes to access, and then clicks on the icon to run the application. Table I lists the definitions of terms applicable to using a mouse.

1.8 Graphical User Interface (GUI). The AN/PSM-93(V) uses graphics (pictures) to interface the user with the computer. The AN/PSM-93(V) uses small pictures, called icons, to represent applications and functions. To access these applications an AN/PSM-93(V) operator moves (points) the mouse cursor over an icon and clicks on it. The application will then run. The AN/PSM-93(V) Version 1.0 uses Microsoft® Windows® as the GUI. Double-click on the AN/PSM-93 icon from the Windows Desktop, or select from the icons found in the AN/PSM-93(V) Development Folder.

Term	Meaning
Click	To quickly press and release the mouse button.
Double-click	To click the mouse button twice in rapid succession.
Drag	To press and hold down the mouse button while you move the mouse.
Point	To move the mouse until the mouse pointer on the screen is over the item of choice.

Source: *MicrosoftÒ WindowsÒ Operating System 3.1 User's Guide*, page 56.

Table 1. Mouse Terms

1.9 HELP. The AN/PSM-93(V) contains on-line help for most applications. Much of this manual was derived from this help information. To access help, point to and click on the HELP Command on the AN/PSM-93(V)'s MENU BAR, or press the F1 key.

1.10 Conventions. Some of the procedures herein require some form of user input. To help in the use of this guide, several conventions for user input were established:

Bold	Type the command and switches, if any, *exactly* as printed, then followed by pressing the ENTER key. Example: To display a disk directory the user would be directed to type **DIR** (followed by pressing the ENTER key).
Italic	Information that the operator must provide. Example: To print a user unique file, the user would be directed to type **Print** *filename* (followed by pressing the ENTER key), in this case the user would type his or her file name, such as MYFILE.DOC.
ALL CAPITALS	Names of keys, icons, directories and files. Examples: ENTER key, PREV MENU icon, C:\DOS directory, and AUTOEXEC.BAT file.
KEY1+KEY2	This plus sign (+) means the keys must be pressed at the same time. Example: When the user is directed to press CTRL+ALT+DEL, he or she would press and hold the CTRL (control) key and ALT (alternate) key and then press the DEL (delete) key.
KEY1, KEY2	The comma (,) means the keys must be pressed in order. Example: The user may be directed to press ALT, F3, this means he or she would press then release the ALT key, immediately followed by pressing the F3 key.

1.11 Support. Any questions or modifications pertaining to the AN/PSM-93(V), or the AN/PSM-93(V) Developer's Guide may be directed to the AN/PSM-93(V) Technical Representative:

COMMANDER
Phil Hoffsetz, Code 6037, Bldg 2917
NAVSURFWARCENDIV
300 Highway 361, Crane, IN 47522-5001
Commercial: 812-854-1708; DSN: 482-1708; FAX: 812-854-1972
hoffsetz@homer.crane.navy.mil

Any changes or additions to the AN/PSM-93(V) will be accompanied by changes to the Developer's Guide. The changes will come with written instructions for the installation of any software and manual changes.

SECTION 2

CREATING SES SUPPORT FILES

2.1 Overview. Because the nature and type of UUTs for which SESs might be generated can vary dramatically, the nature, quantity, and type of SES support files can also vary. The UUT can be anything from a single, individual circuit card, to an entire system or collection of electronic assemblies. In the case of an individual circuit card, the minimum list of support files would typically be as follows:

a. the Operator Instruction Files
b. the Schematic Diagram for the UUT.
c. the Parts Layout, or assembly drawing for the UUT.
d. the Parts List for the UUT.

In the case of a collection of assemblies, or an entire system SES, the Operator Instruction Files may pertain to an entire collection of equipment. The Schematic Diagram might instead be a functional block diagram for the entire system, or a system wiring diagram. The Parts Layout for a system might be a depiction of the location of the individual assemblies within a card cage or system cabinet, and the Parts List may be inclusive of all of the parts found in all of the assemblies. **All of the files generated by a developer for an SES via the development process will be saved by the system to the C:\TESTER subdirectory.**

2.2 Start Development. The first step in a new SES Development process is to double click on the START DEVELOPMENT icon in the AN/PSM-93(V) Development Folder. This will remove all previous files from the C:\TESTER directory, allowing for the start of a fresh SES Development process.

2.3 Creating Setup Instruction Files. ***Since ACE is used to test and evaluate energized electrical circuits and assemblies, it is highly recommended that developers provide SETUP INSTRUCTION FILES to users to promote safety, insure proper setup of the test environment, and insure correct evaluation and initialization of testing parameters.*** SETUP INSTRUCTION FILES are .PCX graphics files displayed by the ACE software package that are meant to depict the UUT and any needed information concerning equipment setup, safety concerns, equipment disassembly, probing locations, sub-assembly environments, etc. These Setup Instruction Files can be drawn by hand using a graphical drawing package; imported from a CAD package or digital camera; or scanned from existing hardcopies and cleaned up. No matter which generation method is used, they must be 790 X 535 pixels or less in size, and be 16 or 256 color. It is highly recommended that the images first be generated as bitmap (.BMP) files and then converted to .PCX files. This conversion may be done by double-clicking on the CONVERT GRAPHICS FILES icon in the AN/PSM-93 Development Folder. This will initiate a graphics conversion program which will allow the developer to select the file to be converted, and enter the new filename.

Multiple Setup Instruction Files may be used with the following restrictions: They must be named **INSTR_0.PCX** through **INSTR_Z.PCX** and they must reside in the **C:\TESTER** directory. These files are to be numbered first chronologically, then alphabetically, that is, in the order in which they are to be displayed to the user. For example: the first Instruction file to be displayed to the user will be saved as **INSTR_0.PCX**, the second file to be displayed will be saved as **INSTR_1.PCX**, and so on. Upon reaching file **INSTR_9.PCX**, naming would continue with alphabetical characters with files called **INSTR_A.PCX** through **INSTR_Z.PCX** Example Setup Instruction Files are shown in Figure 1 below.

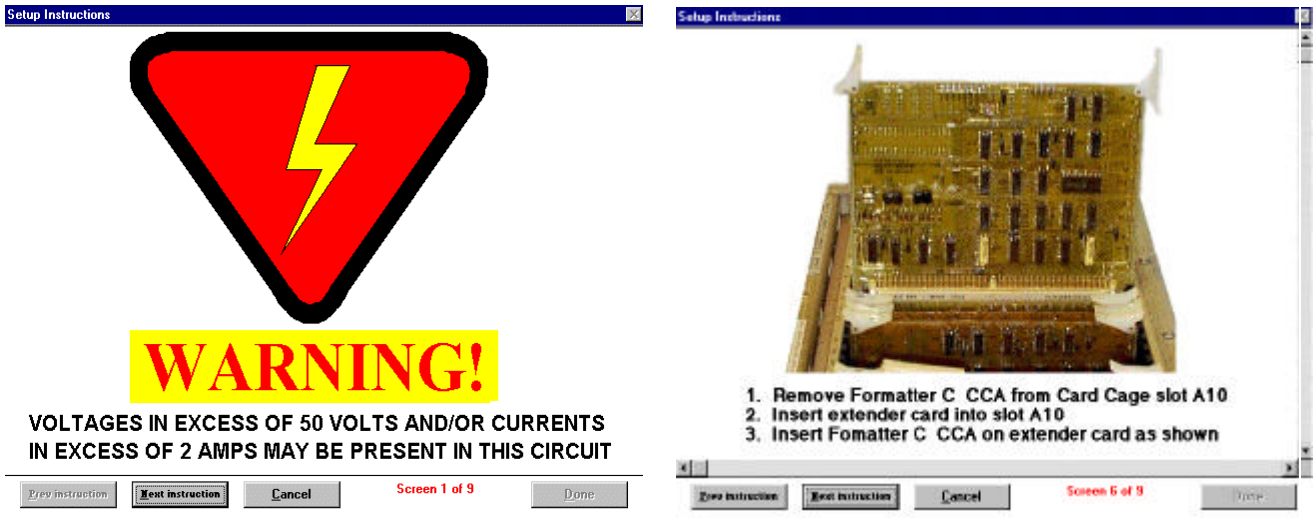


Figure 1. Example *Setup Instruction Files*

2.4 Creating/Editing Test Notes. The AN/PSM-93 system software allows the developer to generate Test Notes for the benefit of the SES user. These notes are used to convey general supply, configuration, and technical information about the UUT to the user such as : UUT Weapons System, UUT Part Number, UUT Revision, UUT Nomenclature, etc. An area for any test notes that need to be shared with user is also provided. Double-clicking on the TEST NOTES EDITOR icon in the AN/PSM-93 Development Folder will allow the developer to generate or edit these Test Notes. Windows Notepad will be initiated with the Testnote.txt filename already in place. Selecting the FILE and SAVE options will perform the saving of the C:\TESTER\TESTNOTE.TXT file.

2.5 Creating The Schematic File. The Schematic File can be generated one of several ways: it can be drawn by hand using a graphical drawing package; it can be imported from a CAD package; or it can be scanned from existing hardcopy and cleaned up.

- If the Schematic File is drawn, it should be saved as a black-and-white (monochrome) graphic file in .BMP format.
- If the Schematic File is generated by scanning an existing drawing, the steps below must be followed:

a) Scan the Schematic Drawing utilizing the scanner hardware and software. The scanned Schematic File should be saved as a 2 color (monochrome) bitmap (.BMP).

b) Open and edit the newly scanned Schematic File in a graphics software package, such as Windows Paintbrush, that is capable of saving 2 color bitmap format files. Use a compatible package of this nature to clean up any broken, missing, or illegible component or trace information, or to correct any errors found with the drawing. Save the cleaned-up version of the file(s) as 2 color bitmap files in the C:\TESTER sub-directory.

c) Use the provided Graphics Conversion file to convert the cleaned and saved file(s) from bitmap (.BMP) files to .PCX files by double-clicking on the CONVERT GRAPHICS FILES icon in the AN/PSM-93 Development Folder. This will initiate a graphics conversion program which will allow the developer to select the file to be converted, and enter the appropriate new filename (GRAPH01.PCX through GRAPH10.PCX). This graphics conversion program assumes that the files to be converted reside in the C:\TESTER sub-directory.

d) Once all of the necessary files have been converted from .BMP to .PCX and saved in the C:\TESTER sub-directory, the developer should go back and delete all of the .BMP files used to create the .PCX files.

There are no restrictions on the method of generating the Schematic Files, the graphics package used, or it's actual size, only on it's format and where the completed file is stored. The completed files must be called **GRAPH01.PCX** through **GRAPH10.PCX**, and they must reside in the **C:\TESTER** sub-directory. **It is a recommended practice for the developer to include the UUT Part Number and Revision Information in this Schematic File as a method of specifically identifying this schematic to a particular UUT for identification and record keeping purposes.** A sample directory tree for a completed SES package, and a sample Schematic File are shown in Appendix B.

2.6 Creating The Parts Layout File. The Parts Layout file is used to locate the individual points that will be probed during the ACE diagnostic process. The file is normally a .PCX file drawn using the guidelines for Huntron Gold Disk Parts Layout files (LAYOUT.PCX). This is not a requirement, as the drawing may show a block diagram of the system rather than an actual UUT parts layout. As with the Schematic File, the Layout File may also be scanned or imported.

If the Layout File is obtained by scanning a photograph or using a digital camera, the final file may be any reasonable, legible size, and be 256 color .BMP format.

If the Layout File is generated by hand drawing or scanning an existing drawing (black & white), the final file may be any reasonable, legible size, and be 16 color .PCX format. The 16 color format will ensure that the Huntron Gold Disk Layout guidelines (color coding) can be satisfied immediately or at a future date.

Open and edit the newly scanned Layout File in a graphics software package, such as Windows Paintbrush. Use a compatible package of this nature to clean up any illegible component information, or to correct any errors found with the drawing. Save the cleaned-up version of the file as a bitmap (.BMP) image in the C:\TESTER sub-directory.

Use the provided Graphics Conversion file to convert the cleaned and saved file(s) from bitmap (.BMP) files to .PCX files by double-clicking on the CONVERT GRAPHICS FILES icon in the AN/PSM-93 Development Folder. This will initiate a graphics conversion program which will allow the developer to select the file to be converted, and enter the appropriate new filename (LAYOUT.PCX). This graphics conversion program assumes that the files to be converted reside in the C:\TESTER sub-directory.

Once all of the necessary files have been converted from .BMP to .PCX and saved in the C:\TESTER sub-directory, the developer should go back and delete all of the .BMP files used to create the .PCX files.

The final version of the file must be called **LAYOUT.PCX**, and reside in the **C:\TESTER** directory. **It is a recommended practice for the developer to include the UUT Part Number and Revision Information in this LAYOUT.PCX file as a method of specifically identifying this Parts Layout file to a particular UUT for identification and record keeping purposes.** A sample directory tree for a completed SES package, and a sample Layout File are shown in Appendix B.

2.7 Creating The Parts List File. The Parts List File is used to identify components on the UUT for probing, replacement, etc. The file is normally a .DOC file created using the guidelines for Huntron Gold Disk Parts List files (BOARD.DOC files). The Parts List File should be saved as an ASCII text file generated using a common word processing package such as Windows® Notepad®, Microsoft® Word®, WordPerfect®, etc. The file must be called **BOARD.DOC**, and reside in the **C:\TESTER** directory. A sample directory tree for a completed SES package, and a sample Parts List File are shown in Appendix B.

The Part List File should possess a header section which contains the following types of information:
a). Host Equipment b). APL c). UUT Part Number d). UUT Revision e). UUT Nomenclature
f). Reference Symbol Number g). UUT NSN h). User Setup Instructions

The body of the Parts List file should also have the following minimum information for each component found on the UUT:
a). Part Number b). Reference Designator c). National Stock Number

SECTION 3

LEARNING SES SUPPORT FILES

3.1 Overview. In a completed SES running under the ACE (Active Circuit Evaluation) software package, all testing begins by selecting a component or node by clicking on points on a finished Schematic. The operator can select his or her own starting point by clicking on any "learned" spot on the displayed schematic diagram, or by clicking on the component name in the listbox on the left hand side of the screen. (See Figure 2 below).

Once a component is selected from the Schematic Diagram, the Parts Layout is then displayed with the selected component depicted under a bull's eye. This shows the user the location and pinout of the device they have selected. (See Figure 3 below).

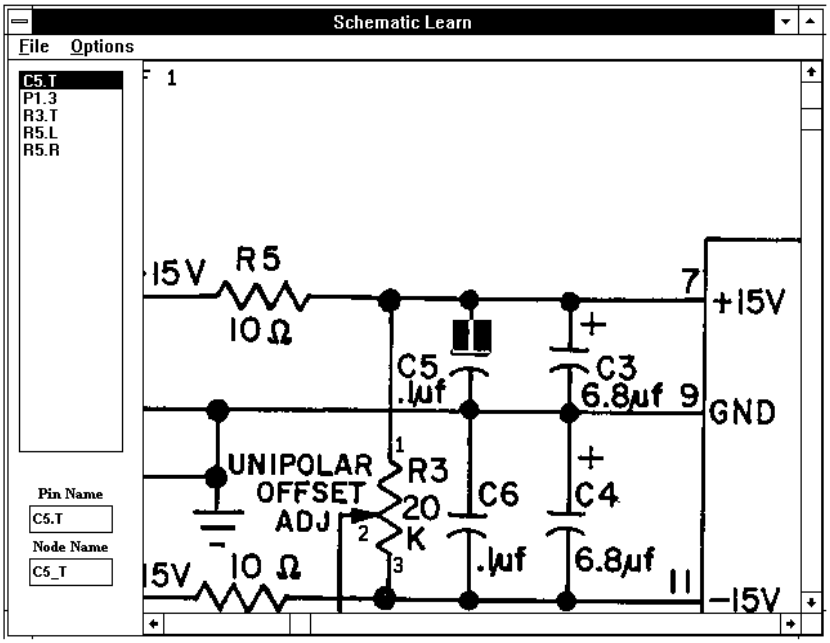


Figure 2. Example *ACE Schematic Diagram*

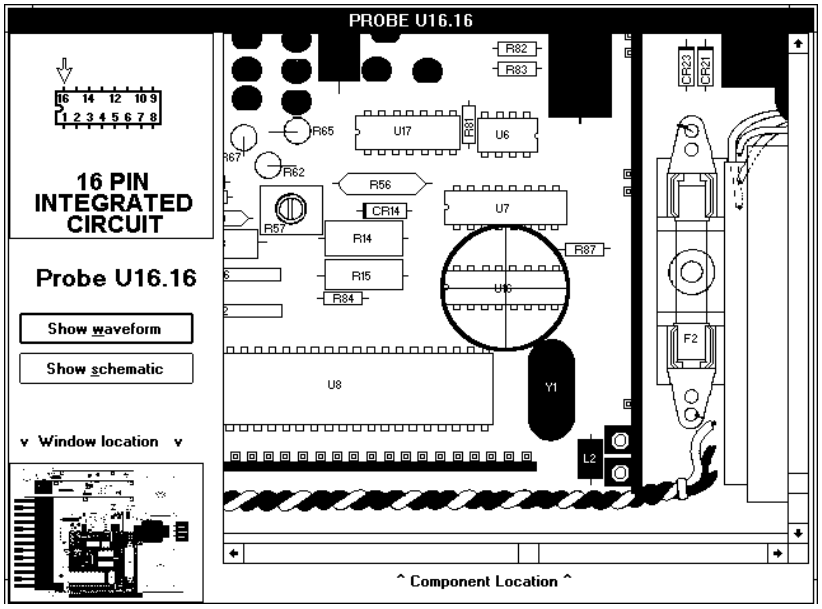


Figure 3. Example *ACE Parts Layout Diagram*

In order for ACE to transition from the selected schematic point, to its location on the parts layout, the

relationship between the schematic information and the parts layout information for the part must be established. This is relationship is determined by the developer through the Schematic and Parts Layout Learn processes.

Once ACE has shown the user the location of the part he or she is interested in, they can select either SHOW WAVEFORM or SHOW SCHEMATIC (See Figure 3 above). SHOW SCHEMATIC returns the user to the schematic diagram, while SHOW WAVEFORM loads and displays the stored waveform(s) for the selected device. In order for ACE to load and display the proper waveform for the selected component, the relationship between the component selected and its corresponding waveform file(s) must be established by the developer. This is done through the Parts Layout Learn process. ***A flow chart example of the development process is shown in Appendix B.***

NOTE: The first step in the SES development process is to double-click on the START DEVELOPMENT icon in the AN/PSM-93 (V)1 folder. This action will copy necessary development files into the C:\TESTER sub-directory, as well as allowing the developer to empty this directory prior to starting development if so desired.

NOTE: At any point in the development process the developer can save his or her work to floppy disk by double-clicking on the SES PACKAGE BACK-UP icon in the AN/PSM-93 folder, or can re-load his or her work from floppy disk by double-clicking on the LOAD DATA FROM FLOPPY icon in the AN/PSM-93 folder.

3.2 Learning The Schematic File. As described in paragraph 3.1, the points of interest on the schematic diagram must be learned by the developer to establish the relationship between the Schematic File (**GRAPH01.PCX**, see paragraph 2.5) and the Parts Layout File (**LAYOUT.PCX**, see paragraph 2.6). This learning of these points will make them selectable, or "clickable" to the operator. The following basic steps must be accomplished in the Schematic Learn process:

1) identify and define component orientation (LAYOUT vs. SCHEMATIC) (paragraph 3.2.1).
2) assign component orientation labels and generate the SCHEMATIC WORKSHEET (paragraph 3.2.1).
3) define and learn component areas (paragraphs 3.2.2 and 3.2.3) .
4) assign pin and node names to the learned areas (paragraphs 3.2.4 and 3.2.5) .

3.2.1 Identifying Component Orientation. The first step in this Schematic File learn process is to define the relationships between the location and orientation of the components as found on the UUT and their corresponding orientation as it pertains to the Schematic Diagram. For example, a two lead component such as a resistor will most likely be installed on the UUT in either a vertical (North-South) or horizontal (East-West) orientation. This same resistor may be depicted differently on the Schematic and the Parts Layout diagrams. EXAMPLE: In Figure 4 below, resistor R7 is found to be mounted vertically on the actual UUT, but is depicted horizontally on the schematic. The developer must determine which leg of R7 on the UUT corresponds to which electrical point on the schematic.

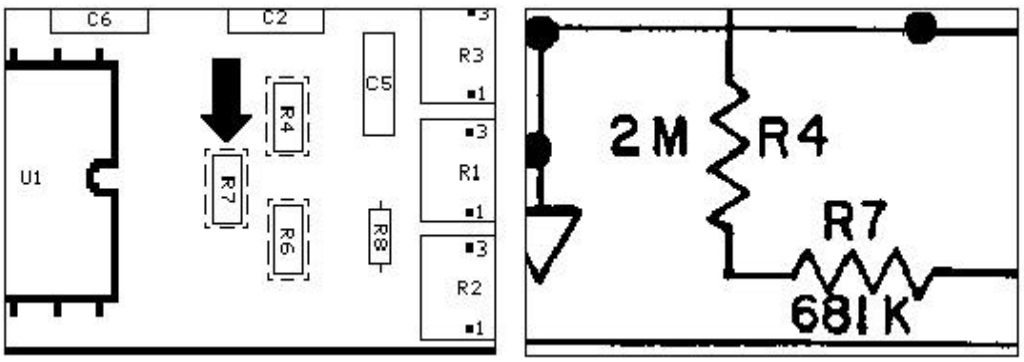


Figure 4. Example *R7 Diagrams*

***These orientation relationships must be established by the developer for every component or point on the UUT for which a waveform will ultimately be collected*.** This will be accomplished through the generation of a SCHEMATIC WORKSHEET. This Schematic Worksheet will consist of a hardcopy of the UUT schematic diagram on which the developer will record (write) the results of the component orientation investigation. An example of a completed Schematic Worksheet with component orientation labels recorded can be seen in Figure 5 below. A convention for recording these component orientation labels has been developed, and is shown in Table 2. When all of the needed points have been defined and labeled on the schematic worksheet, the actual learning of points on the **C:\TESTER\GRAPH01.PCX** file can begin.

3.2.2 Learning Schematic Pins. Once all of the component orientation relationships between the Parts Layout and the Schematic Diagram have been investigated, identified, and recorded on the Schematic Worksheet, the Schematic Learn itself is ready to proceed. This is started by double clicking on the **LEARNING THE SCHEMATIC FILE** icon in the **AN/PSM-93(V)** Development Folder in Program Manager (or in the AN/PSM-93(V) Development Folder for WIN '95). This will activate the **C:\WORK2\LRN_SCM.EXE** file which will load and display the **GRAPH01.PCX** file that the developer generated and previously saved in the **C:\TESTER** directory.

Once the **GRAPH01.PCX** file is loaded and displayed, the developer will learn all of the points on the schematic which correspond to a waveform which will be collected and stored. The order in which pins are learned is by the preference of the developer. Pins can be learned from input to output, by the functions of the parts of the circuit, by component types, in accordance with a maintenance or troubleshooting procedure, or by component numerical designation, etc. As the pins are learned, regardless of the method employed, they will be listed alphabetically by first character in the list box at the left hand side of the SCHEMATIC LEARN screen. The following steps must be accomplished for each pin learned:

1) learn the coordinate, or "hot-spot" area for the pin (paragraph 3.2.3).
2) assign a PIN NAME to the learned area (paragraph 3.2.4).
3) assign a NODE NAME to the learned area (paragraph 3.2.5).

After completing GRAPH01.PCX, if there is more than 1 page of the schematic, simply click on File and Open to retrieve and learn the next pages (GRAPH02, GRAPH03, ...).

3.2.3 Defining Pin Areas. The developer must learn, or define the area of the schematic diagram which will pertain to each pin. This is the area which will become a "hot-spot", or a spot on the schematic that a user will be able to click on when desiring to test this pin. This pin area learn is accomplished by drawing (clicking and dragging) a box at, or around the schematic location in question. (See Figure 5 below).

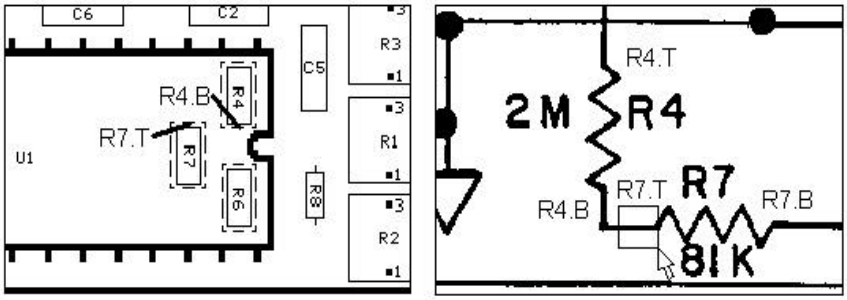


Figure 5. Example *Schematic Worksheet* and *Area Learn Box*

Component Type	Example Name	Explanation
2 Leg:	R1.T, R1.B, R2.L, R2.R	.T, .B, .L, .R, = top, bottom, left, and right.
3 Leg:	Q1.E, Q1.B, Q1.C	.E, .B, .C, = emitter, base, and collector.
	Q2.S, Q2.G, Q2.D	.S, .G, .D, = source, gate, and drain.
Multi-leg:	U1.1, U1.2, U1.3	.1, .2, .3, = component pin numbers.
Connectors:	P1.1, J1.2, etc.	.1, .2, .3, = connector pin numbers.

Table 2. Naming Conventions

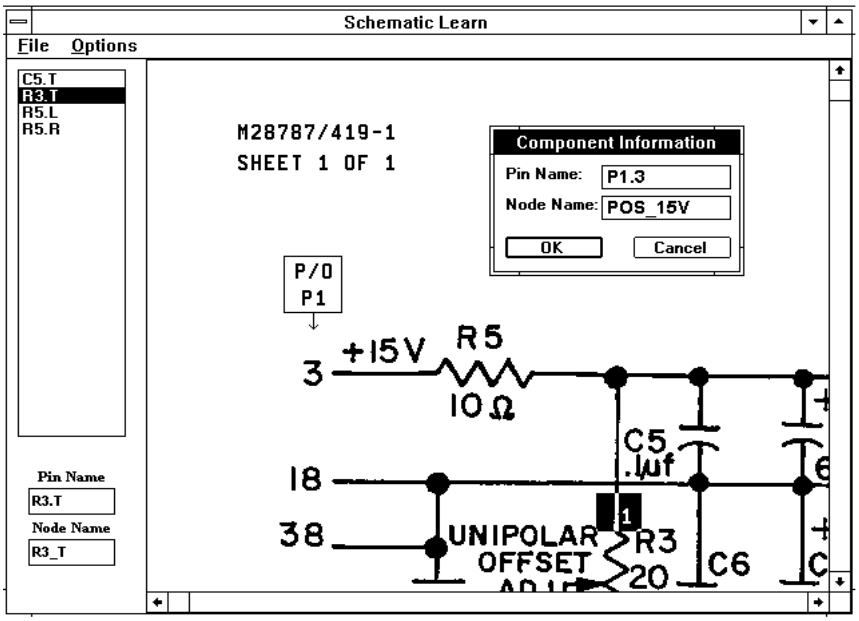
3.2.4 Assigning Pin Names. Once learned, the developer must give this newly learned area of the schematic a PIN NAME. This PIN NAME needs to be a description of the learned area or pin. For example, in Figure 5 above, the box to the left of R7 on the schematic shows the pin area just learned by the developer. This box on the schematic corresponds electrically to the top of the R7. During the establishment of the relationship between the UUT and the Schematic (see paragraph 3.2.1), the developer determined that this point on the schematic corresponds electrically to the bottom of R4, or the top of R7. The developer therefore hand recorded a working name for this pin on the Schematic Worksheet as "R7.T" at that time.

3.2.5 Assigning Node Names. After the pin has been learned and assigned a PIN NAME, the developer must next assign a NODE NAME to this point. A node is a point in a circuit where two or more component pins are connected together making them electrically equivalent. While the PIN NAME was a physically descriptive name, the NODE NAME needs to be an electrically descriptive name for the learned point. This Node Name is used to indicate which component pins are directly tied together.

Returning to Figure 5 for an example: the node or point between the bottom of R4 and the top of R7 is an electrically equivalent point. If the component R7 were learned before component R4, this circuit node would have already been assigned both a Pin Name and a Node Name. The Pin Name would be “R7.T” and the Node Name would be “R7_T”. When the bottom of resistor R4 was learned, it would be given a Pin Name of “R4.B” but the Node Name of “R7_T” would be retained. It is this node or point in the circuit, the point between R4 bottom and R7 top, that is identified by the Node Name “R7_T”, but its Pin Name depends on the component being dealt with (either R7 or R4 in this case). In a second example, if the node learned is edge pin P1.3, which is a positive 15 Volt input pin, a good descriptive PIN NAME would be "P1.3", and a good NODE NAME would be "POS_15V" (See Figure 6 below). The Node Name “POS_15V” is more descriptive of the physical circuit point at the pin P1.3 than a Node Name of “P1_3” would be. Also, each point in the circuit that it electrically equivalent to P1.3 can now also be assigned a Node Name of “POS_15V” rather than one of “P1_3”.

The Node Name is normally a derivation of the Pin Name that is the signal source ***to*** the node. In other words, the Node Name is typically labeled for the pin that is driving the node in question. The exception to this convention occurs when the node includes an edge pin. In this case, the Node Name is a derivation of the edge pin name, whether it is an input or an output. It should be noted that this convention has been adopted for clarity and ease of understanding. EXAMPLE: Refer to Figure 6 below, the right side of resistor R5 would be considered the driving or signal source for the top of R3 and the top of C5. Therefore the preferred Node Name for all three of these points would be R5_R, while the Pin Names would be

R5.R, R3.T, and C5.T.



Example PIN NAMES	**Example NODE NAMES**
R1.T, R1.B, R2.L, R2.R	R1_T, R1_B, R2_L, R2_R.
Q1.E, Q1.B, Q1.C	Q1_E, Q1_B, Q1_C.
R1.1, R1.2, R1.3	R1_1, R1_2, R1_3.
U1.1, U1.2, U1.3	U1_1, U1_2, U1_3.
P1.3	POS_15V
P1.18	P1_18

Figure 6. Example *PIN* and *NODE NAME* Depictions

Table 3. Example Pin and Node Names

There will be cases where the PIN and NODE names will be similar. For example, in Figure 6 above, the right side of R5 may have a PIN NAME of R5.R and a NODE NAME of R5_R. ***Notice the difference between "R5.R" and "R5_R"! NODE NAMES cannot contain a "dot", or "period" character*** as they will be used in naming waveform files at a later time in the development process! NODE NAMES should be no more than 8 characters in length, and contain only letters, numbers and underscores, and not include symbols such as "+" or ">". The NODE NAME "R5_R" will eventually become part of a filename for the file called "R5_R.DIF", which will be the saved waveform that was collected from the node that is the same as the right side of R5 on the schematic. Therefore, the convention that has been established for NODE NAMES is to replace what would be a "dot", or "period" character in the PIN NAME, with an

"underscore", "_" character in the NODE NAME. (See Table 3 above).

3.2.6 Completing The Schematic Learn and Software Options. Several software options are available in the Schematic Learn software package. These are usable to the developer by clicking on either the FILE, or OPTIONS choices in this window's menu bar. SAVE and EXIT are accessed under the FILE option, while SHOW ALL, HIDE ALL, SHOW ALWAYS, SHOW CURRENT NODE, NODE LIST, EDIT PIN INFO, and DELETE COMP INFO are found under the OPTIONS selection. A right mouse click will select the pin under the cursor if one has been learned, or move the area of the drawing under the cursor to the center of the window (or as close to the center as possible) if the cursor is not over a learned area. Descriptions for the various functions are listed below:

SHOW ALL: Highlights all areas (hot spots) on the Schematic Diagram which have been learned.
HIDE ALL: Undoes SHOW ALL and hides the previously highlighted areas.
SHOW ALWAYS: Each area learned will become and remain highlighted as it is learned.
SHOW CURRENT NODE: Displays and Highlights the node the developer is currently dealing with, either because it was associated with the pin selected from the left-screen list box of learned pins, or because it is associated with the pin currently being learned.
NODE LIST: Generates a **C:\TESTER\NODELIST.TXT** text file which is a listing of learned points by Node Name and Pin Name. This function can be utilized as a display of which points have, or have not been learned, thereby providing a method of checking the overall learn status.
EDIT PIN INFO: Opens an EDIT PIN INFORMATION window, and allows the developer to make changes to Pin and Node names, and screen coordinates for learned points.
DELETE COMP INFO: Deletes all learned information for the component that is selected or highlighted in the left-screen list box at the time DELETE PIN INFO is selected.

Once the developer has learned all the nodes desired, or upon wishing to exit, ***the file must be saved by exercising the SAVE option in order to save the accomplished work*** and create the **C:\TESTER\GRAPH01.MAP** file. The **C:\TESTER\GRAPH01.MAP** file is the "learned" Schematic file the developer has just created while learning the **C:\TESTER\GRAPH01.PCX** file through the Schematic Learn.

3.3 Learning The Parts Layout File. The next step in the process of generating all of the necessary files for a complete SES, is to learn the previously created Parts Layout File (LAYOUT.PCX, see paragraph 2.4). The Parts Layout File Learn function is performed by the **C:\WORK2\LRN_ASY.EXE** file, which will be started when the developer double-clicks on the LEARNING THE PARTS LAYOUT FILE icon in the AN\PSM-93(V) Development Folder. This Learning The Parts Layout process is designed to generate a **C:\TESTER\LAYOUT.MAP** file which will further define the relationships between the point that the user has selected from the schematic, the location of the selected component on the layout, and the stored waveform files for this device. The basic steps in the Layout Learn process are as follows:

1) search the standard component library for an existing bitmap for each component. Track the results of this component library search by creating a LAYOUT WORKSHEET (paragraphs 3.3.1. and 3.3.2).
2) create graphic files for components not found in the Standard Library. (paragraph 3.3.3).
3) learn and define the component area of the Layout File and assign component and library element names for each device (paragraph 3.3.4).

3.3.1 Using The Standard Component Library. A Standard Component Library exists in the **C:\COMP_LIB** directory. The contents of this library have been re-produced in Appendix C as a hardcopy catalog listing. The developer is to perform a comparison check between this Appendix C catalog listing of standard library components and the Layout drawing to identify any components on the UUT for which there is not already an existing bitmap picture. **The developer must create bitmap drawings for all components which do not already have an existing bitmap in the library.** The component library can also be

checked on the AN/PSM-93(V) by double-clicking on the LEARNING COMPONENT AREAS icon in the AN/PSM-93(V) Development Folder (or folder). A "Choose a Component" box will appear, with **C:\TESTER** highlighted as the default directory.\

To search through the Standard Library, the developer must change to the **C:\COMP_LIB** directory by first double-clicking on the **C:** folder, then scrolling down in the window and double-clicking on the **COMP_LIB** folder. This will allow the developer to search through the components found in the Standard Library visually on the PC.

3.3.2 Creating The Layout Worksheet. As the developer searches the Standard Library for each component, entries should be made on the hardcopy drawing of the UUT Layout file. This hardcopy of the Layout drawing will become the LAYOUT WORKSHEET. The developer should annotate beside each component on the layout drawing, the name of any existing library files which can be used for this SES. The developer should also make notes beside each component describing whether or not a library file exists, and if a bitmap file will need to be created. An example LAYOUT WORKSHEET is shown below in Figure 7.

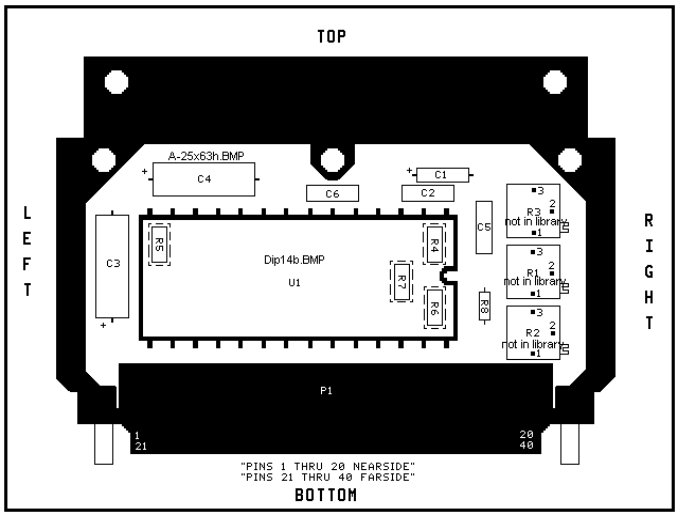


Figure 7. Example *Layout Worksheet*

In Figure 7 above, it can be seen that the developer has searched the Standard Component Library for components R1, R2, R3, U1, and C4. Resistors R1-R3 were not found in the library and the developer has labeled them "not in library". Because these resistors were not found in the Standard Library, custom drawings to depict them will have to be created by the developer and stored in the **C:\TESTER** directory. Integrated circuit U1 was found to be best depicted by the Standard Component Library bitmap called **Dip14b.BMP**, and was labeled accordingly by the developer on the worksheet. The results of the developer's library search for capacitor C4 is that it was found to be best depicted by the library bitmap called **A-25x63h.BMP**, and is so annotated on the worksheet. **This library search and Layout Worksheet annotating process must be completed for every component found on the UUT.**

3.3.3 Creating Custom Graphics For Components Not Found In The Standard Library. Custom graphics, or pictures, must be created for each component not found in the Standard Component Library. The Custom Graphic drawn for a component should depict the device from the viewpoint a user would have when holding and looking at the physical UUT. In most cases, this is a "top-down" view, or what one would see when looking down on the top of the component. These Custom Component Library graphic files can be drawn by hand using a graphical drawing package; imported from a CAD package; or scanned from

existing hardcopies and cleaned up. No matter which generation method is used, **they must be 161 X 165 pixels in size, be a 16 color .BMP (bitmap) format, and be stored in the C:\TESTER directory.** There is a template, or example, of this size bitmap that can be used by the developer as a "window" or guide template for this size bitmap. This template file is **C:\COMP_LIB\TEMPLATE.BMP**.

3.3.4 Learning Custom Graphics For Components Not Found In The Standard Library. Once custom graphics have been created for any components that were not found in the Standard Component Library, these custom files must then be learned. The Component Learn software is used to perform this learn and designate what the pin names are, and the location and direction of the arrow that indicates each pin. The Component Learn process is initiated by clicking on the LEARNING COMPONENT AREAS icon in the AN/PSM-93(V) Development Folder. There are two different methods that can be used to designate the pins:

Method 1: Drag the appropriate arrow from the grid at the lower left of the window to the desired location on the graphic. You will then be prompted for the pin name. If a pin in the pin list is highlighted, the next logical name will be filled in for you. Change the name as required, change the arrow direction if incorrect, and click OK when done.

Method 2: Click the spot on the picture where the head of the arrow will appear. You will then be prompted for the pin name and arrow direction. If a pin in the pin list is highlighted, the next logical name will be filled in for you. Change the name as required, select the arrow direction, and click OK when done.

After a pin has been defined, the developer can view, change, delete, or save the pin information. When a pin is clicked on in the pin list, the arrow for that pin is shown in the picture. The highlighted pin name can be deleted from the pin list by pressing the DELETE key on the keyboard. There are also several menu choices that affect the pin information.

3.3.4.1 File Menu Explanation.

OPEN: This menu choice allows the developer to open another component file. The file may be a new file (one without any pins defined), or one that already has pins defined. If CANCEL is chosen, the current component remains loaded. If the current data had been changed since the last save, the developer will be prompted whether to save the old data before loading a new component. CTRL+O also accesses this menu choice.

SAVE: This menu choice saves the current information back to the file that was opened with the OPEN menu choice. The .MAP file will be saved in the same directory as the .BMP file. The .BMP file is not saved, only the .MAP. CTRL+S also accesses this menu choice.

EXIT: This menu choice exits the software package. If the current data had been changed since the last save, the developer will be prompted whether to save the old data before exiting. CTRL+Q also accesses this menu choice.

3.3.4.2 Options Menu Explanation.

SHOW ALL ARROWS: This menu choice shows arrows for all the pins in the pin name list. The picture is cleared when the next pin is defined, or HIDE ALL ARROWS menu choice is clicked. CTRL+A also accesses this menu choice, or pressing the letter A on the keyboard.

HIDE ALL ARROWS: This menu choice clears all arrows from the picture. CTRL+H also accesses this menu choice, or pressing the letter H on the keyboard.

EDIT PIN INFO: This menu choice allows the editing of the currently highlighted pin in the pin list. The information that can be changed includes the pin name, the pin location and the arrow direction. CTRL+E also accesses this menu choice, or pressing the letter E on the keyboard. This menu choice can also be accessed by double-clicking the appropriate pin in the pin list.

3.3.5 Learning Component Areas. Once a picture or graphic exists for each component on the UUT Layout drawing, either from the Standard Component Library, or by developer creation, the actual learning of component areas on the **C:\TESTER\LAYOUT.PCX** file can begin. This step in the process will generate the **C:\TESTER\LAYOUT.MAP** file. To begin, double-click on the LEARNING THE PARTS LAYOUT FILE icon in the AN/PSM-93(V) Development Folder. This will initiate the **C:\WORK2\LRN_ASY.EXE** file, and will display the LAYOUT.PCX file in a window, along with a list box for learned components, COMPONENT NAME and LIB ELEMENT NAME boxes, and a window that displays the picture the developer chose to depict the component. (See Figure 8 below).

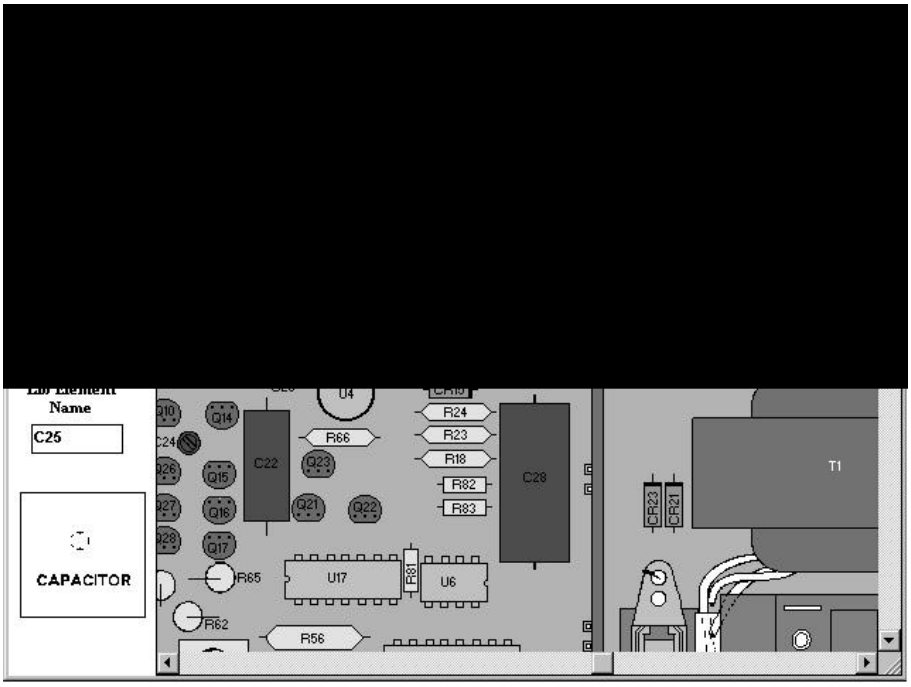


Figure 8. Example *Learn Layout* Screen

The Layout Drawing is learned by drawing (clicking and dragging) a box around each component, and then assigning both a COMPONENT NAME and a LIBRARY ELEMENT NAME (Lib Element Name) to this boxed area. The COMPONENT NAME should be descriptive of the part or component in question, such as "C1", "R4", "VR6", "P1.4", etc. The LIBRARY ELEMENT NAME should be the name of a component graphic (bitmap) file that is found in either the **C:\TESTER** or the **C:\COMP_LIB** directories. What the developer is doing is defining an area on the Layout, and then assigning both a name and a drawing to that area. This process defines where a component is located, and what graphic or picture will be displayed to the user for that component.

When the area box has been drawn, a COMPONENT INFORMATION window or box will appear on the screen. (See Figure 9 below). From this window the developer is to enter the COMPONENT and LIBRARY ELEMENT NAMES for the component in question. There are BROWSE buttons available

beside each of these functions. The BROWSE button beside the COMPONENT NAME function allows the developer to select COMPONENT NAMES from the previously generated Schematic MAP file (**C:\TESTER\GRAPH01.MAP,** paragraph 3.2.6). This BROWSE function will also notify the developer when all available names from the Schematic MAP file have been exhausted.

The BROWSE button next to the LIBRARY ELEMENT function will bring up another BROWSE COMPONENT LIBRARY window. (See Figure 10 below). This window allows the developer to view and select existing component bitmap files from either the Standard Component Library (**C:\COMP_LIB**) or from the Custom Component Library (**C:\TESTER**). (See paragraphs 3.3.1 - 3.3.3). It is from this window that the developer will select which bitmap for each component will be used in the Layout MAP file. The selected bitmap will come from either the Standard or Custom Component Libraries.

The following steps from paragraph 3.3 are repeated in greater detail as an example of the process flow for performing the learning of the Layout drawing, and creating the **C:\TESTER\LAYOUT.MAP** file:

1) Search the Standard Component Library for an existing bitmap for each component. This can be done by: a) double-clicking on the LEARN COMPONENT icon in the AN/PSM-93(V) Development Folder (or folder), and using the "Choose a Component" box to view the bitmaps in the Standard Library, or b) by viewing the bitmaps from the hardcopy catalog found as Appendix C of this developer's guide.
2) The developer is to track the results of this component library search by creating a LAYOUT WORKSHEET (paragraphs 3.3.1. and 3.3.2). The worksheet is to be filled out with the names of the bitmap files that will be used for each component, regardless of whether they are Standard or Custom.
3) The developer must create Custom bitmap files for **all** components not found in the Standard Library. (paragraph 3.3.3). **These Custom files must be saved in C:\TESTER**. These developer created bitmaps are the "Custom" library for the SES in development.
4) The developer needs to learn and define the component area of the Layout File and assign the appropriate Component and Library Element names for each device (paragraph 3.3.4).

Figure 9. Example Layout Learn *Component Information* Window

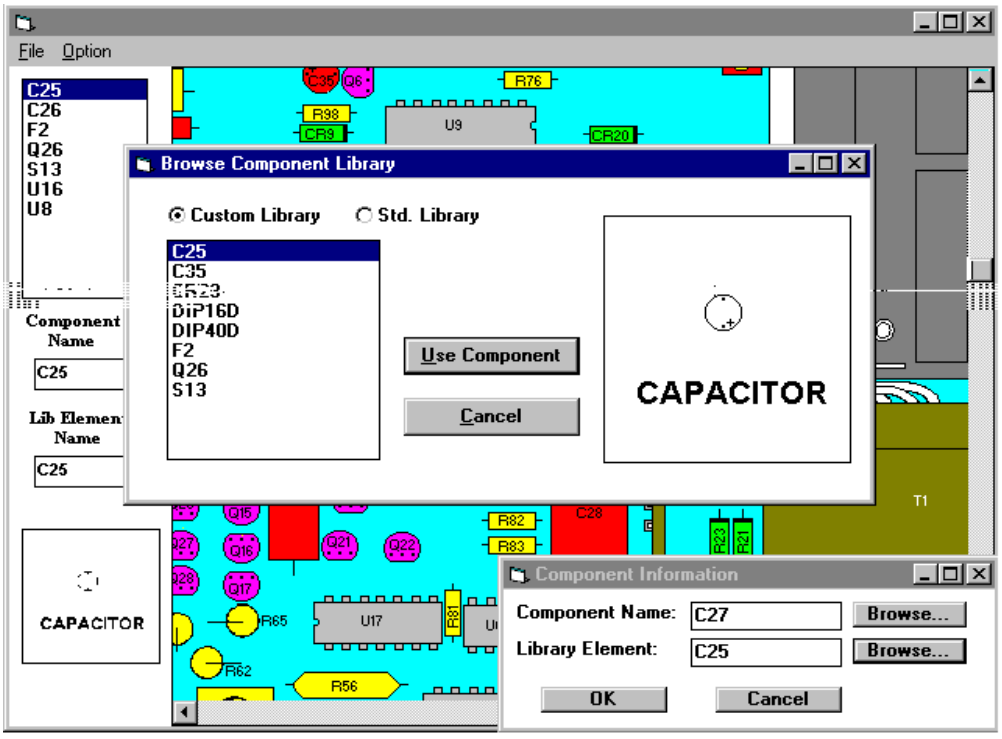


Figure 10. Example Learn Layout *Browse Component Library* Window

3.3.6 Completing The Layout Learn and Software Options. Several software options are available in the Learning The Parts Layout File software package. Clicking on a learned component on the Layout display will bring up the Component Information Window for the learned component. The other software options are usable to the developer by clicking on either the FILE, or OPTIONS choices in this window's menu bar. SAVE and EXIT are accessed under the FILE option, while SHOW ALL, HIDE ALL, SHOW ALWAYS, COMPONENT LIST, EDIT COMPONENT INFO, and DELETE COMPONENT INFO are found under the OPTIONS selection. A right mouse click will select the pin under the cursor if one has been learned, or move the area of the drawing under the cursor to as close to the center as possible if the cursor is not over a learned area. Also, SHIFT+RIGHT-CLICK "zooms" in at a scale of times 4 (4x) on the Layout Drawing. Pressing SHIFT+RIGHT-CLICK a second time zooms back to times one (1x) scale.

Highlighting a component in the component list box will show the component name in the COMPONENT NAME box, the library element name in the LIB ELEMENT NAME box, and show a "thumbnail" view of the library element graphic in the box at the lower left of the window. Clicking on the thumbnail view shows a full-size view of the library element graphic in the center of the window. The full-size view will be visible as long as the left mouse button is held down, and disappears when the mouse button is released.

Descriptions for the various software functions are listed below:

SHOW ALL: Highlights all areas (hot spots) on the Layout Diagram which have been learned.

HIDE ALL: Undoes SHOW ALL and hides the previously highlighted areas.

SHOW ALWAYS: Each area learned will become and remain highlighted as it is learned.

COMPONENT LIST: Generates a C:\TESTER\COMPLIST.TXT text file which is a listing of learned points.

EDIT COMPONENT INFO: Opens an EDIT COMPONENT INFORMATION window, and allows the developer to make changes to Component and Library Element names, and screen coordinates for learned points.

DELETE COMPONENT INFO: Deletes all learned information for the component that is selected or highlighted in the left-screen list box at the time DELETE COMPONENT INFO is selected.

Once the developer has learned all the components desired, or upon wishing to exit, ***the file must be saved by exercising the SAVE option, in order to save the accomplished work***, and create the **C:\TESTER\LAYOUT.MAP** file. The **C:\TESTER\LAYOUT.MAP** file is the "learned" Layout file the developer has just created while learning the **C:\TESTER\LAYOUT.PCX** file from the Layout Learn package.

3.4 Adding and Editing Operator Instructions. To add specific instructions for a single waveform, or to instruct the operator to choose multiple waveforms for a single node, the Operator Instruction Editor is used to insert and edit these instructions. To use the Editor, double-click on the OPERATOR INSTRUCTION EDITOR icon in the AN/PSM-93(V) Development Folder. The editing process is as follows:

1) Choose the node that instructions will be required for.
2) Choose a unique name that will distinguish this instruction from others for the same node.
3) Enter the text of the instruction.
4) Save the work.

Instructions previously entered may be modified at any time using the editor, as well as adding more instructions, or deleting existing ones.

3.4.1 Using The Instruction Editor. To use the editor, select the ADD INSTRUCTIONS under the OPTIONS menu pick (or press Ctrl A). For the first node, a "NODE DOESN'T EXIST" message may appear. Click OK. Pick the node name from the top in the "ADD NODE/INSTRUCTION" dialog window. The node name drop-down box allows the developer to choose a defined node (from the Schematic Learn software). The instruction name must be a file name that the AN/PSM-93 will use to store the waveform data for this instruction. This file name is a maximum of 8 characters in length and may not include a period or other characters that can't be used in a normal MS-DOS file name. If only one waveform for this node will exist,

the node name and the instruction name may be the same, otherwise use the remaining 8 characters to uniquely identify this waveform. Click the OK button.

The node and instruction name indicated in the dialog window will appear in boxes at the upper right of the window. The currently selected waveform is shown at the top. Click the mouse cursor in the instruction text box and type the instructions for this waveform. When complete, click the "SAVE INSTRUCTION" button below the text box.

To modify the instructions for an existing waveform, use the node and instruction name drop-down boxes to select the desired instruction, then click the mouse cursor in the instruction text box and edit as required. Again, click the "SAVE INSTRUCTION" button.

To add a second instruction for a node, select the node from the drop-down box and press Ctrl A. A unique filename must be entered (different than the previous one). Click OK. The instruction text from the previously selected node will be copied to the new instruction, and can be modified as required. Click the "SAVE INSTRUCTION" button.

To add instructions for a different node, press Ctrl A, then select the new node and proceed as before.

To delete a single instruction, select DELETE INSTRUCTION under the OPTIONS menu.

To delete all the instructions for a node, select DELETE NODE under the OPTIONS menu.

The WRITE TO DISK selection under the FILES menu option saves all edits.

SECTION 4

LEARNING WAVEFORM AND BOUNDARY FILES

4.1 Overview. At the heart of the test philosophy approach of the AN/PSM-93(V) is the collection, storage, and retrieval of AC or DC waveforms from electronic assemblies, sub-assemblies, and systems. These waveforms are to be collected from known good equipment and saved as Boundary Envelope computer files by a storage oscilloscope and the ACE and BENCHTOP™ software packages. These Reference Waveforms will be collected from developer specified pins or nodes in a given circuit, electrical assembly, system, or sub-system. The test philosophy employed in the AN/PSM-93(V) via the Active Circuit Evaluation (ACE) software package is as follows: UUT (Unit Under Test) Reference Waveforms are to be collected **from a known good UUT or system that is operating in a known, repeatable, and technically relevant, power-on state**. These stored, or "golden" reference waveforms will then be utilized by users as baseline test signals to compare with the signals taken from a failed or suspect UUT. By comparing these reference signals collected from a known good UUT, to those of a failed or suspect UUT, the operator should be able to locate, isolate, and repair failures present on the failed UUT. These stored waveforms will be in the form of WAVEFORM BOUNDARY FILES. A Boundary File is an envelope surrounding a UUT waveform which serves as a “window” or area within which a good or passing waveform must fall. (See Figure 11 below). These Boundary Files are automatically built through the ACE software package at the time that the UUT reference waveforms are probed, collected, and stored.

It is the job of the developer to identify and implement the approach for SES development, and then to collect the best quality baseline test data for implementation of this chosen approach. This section will deal with the methods and practices of acquiring, collecting, and storing this waveform data for incorporation into usable SESs.

4.2 Learning ACE Waveforms. To start the Waveform Learn process, double click on the **LEARNING ACE WAVEFORMS** icon in the **AN/PSM-93(V)** Development Folder. This will activate the **C:\WORK2\LRN_WVF.EXE** file which will load and display the **GRAPH01.PCX** file that the developer generated, learned, and previously saved in the **C:\TESTER** directory.

The first step is to choose the point or component for which a waveform is to be collected. Once the UUT Schematic Diagram is displayed, the developer can either click on the point or component to be learned directly on the schematic, or by selecting it (by double clicking) from the listbox on the left hand side of the window. Once the pin, node, or component has been selected from the UUT Schematic Diagram, the UUT Layout Drawing will then be displayed with the selected component shown under a bull's eye. (See Figures 12 and 13 below).

To proceed with the Waveform Learn Process from the Waveform Learn Layout Window, the developer must click on the LEARN WAVEFORM button found on that screen. Clicking on the SHOW SCHEMATIC button will return the developer to the Waveform Learn Schematic Window. (See Figures 12 and 13 below).

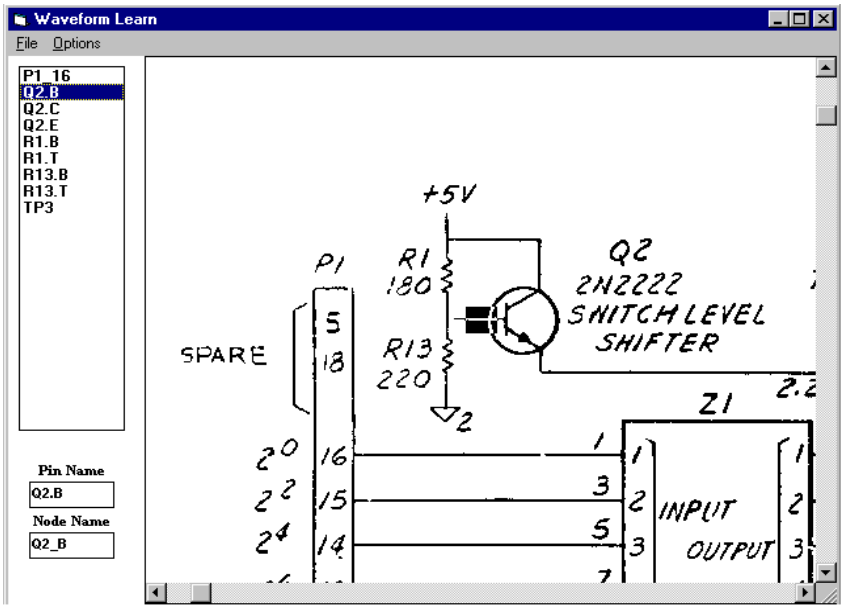


Figure 12. Example *Waveform Learn* Schematic Window

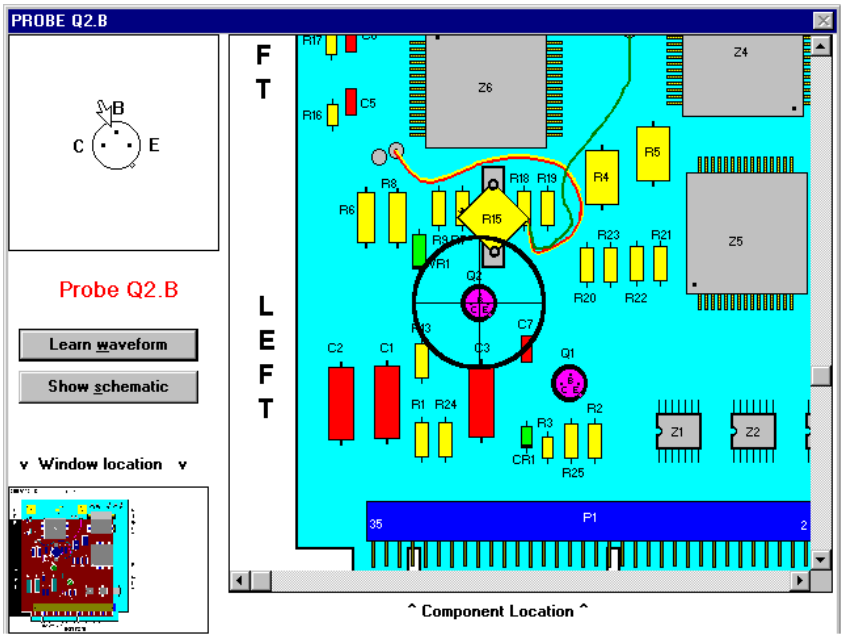


Figure 13. Example *Waveform Learn* Layout Window

Clicking on the LEARN WAVEFORM button on the Waveform Learn Layout Window will initiate the **C:\PCI\BTOP\BTOP.EXE** file which is part of the Commercial-Off-The-Shelf (COTS) software package (BENCHTOP) that controls and operates the OS-301(P)/PSM-93(V) Oscilloscope. The BENCHTOP™ FOR WINDOWS software package will come up with a standardized screen layout displaying the most widely used oscilloscope control functions, for changing or setting the scope settings for waveform capture. Overlaid on top of this BenchTop screen is an ACE package dialog box which allows the developer to click on either a SAVE WAVEFORM button or a CANCEL button. (See Figure 14 below).

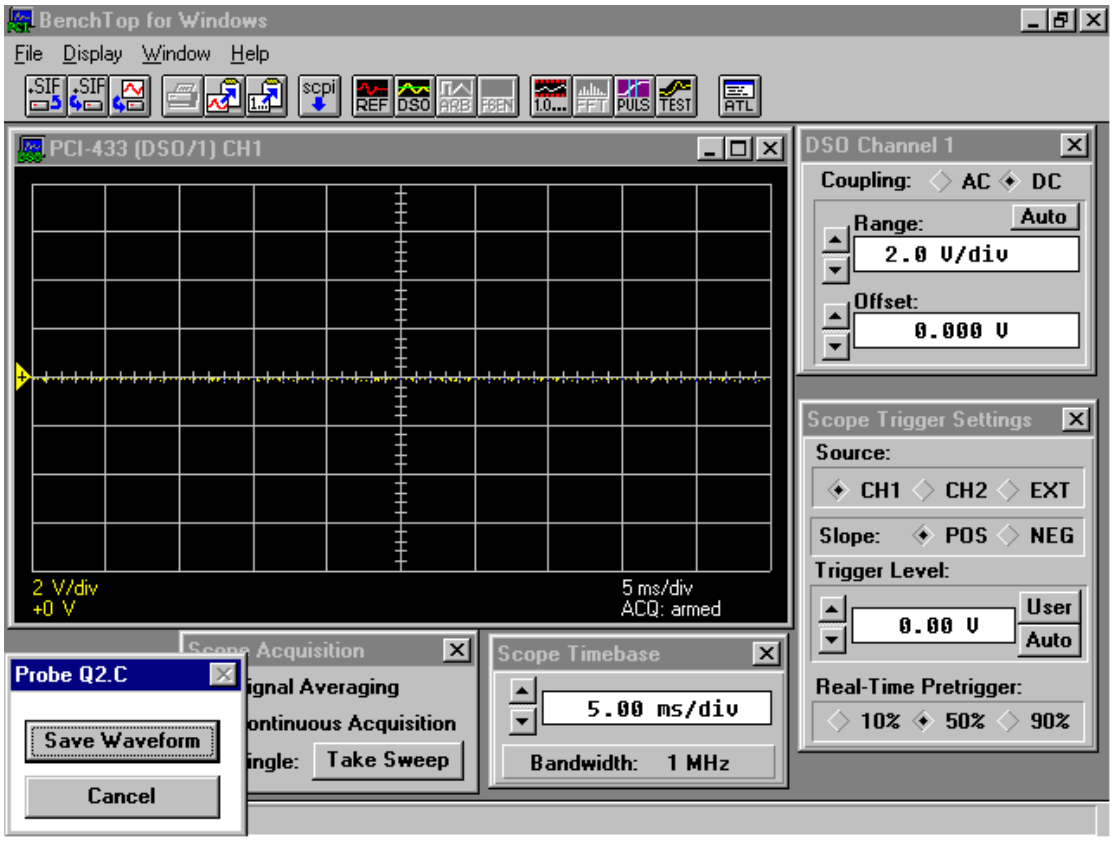


Figure 14. Example *BENCHTOPÔ FOR WINDOWS Waveform Learn* Screen

The specifics of the features and workings of the BENCHTOP software package are not discussed in this DEVELOPER'S GUIDE. Refer to the manufacturer's manual, **BENCHTOP FOR WINDOWS USER'S GUIDE**, for detailed instruction and guidance on the BENCHTOP software package, and the associated oscilloscope controls and features.

4.3 Basic Guidelines For Waveform Capture. After connecting the oscilloscope probe to the desired node and connecting the probe shield to an appropriate ground, adjust the vertical deflection of the oscilloscope to achieve a signal amplitude as large as possible without exceeding 7 divisions, and attempt to keep the most important, rapidly changing, unique, or indicative portions of the waveform away from the extreme edges of the Oscilloscope grid display, as this will allow space to display, and later, edit the boundaries. Adjust the oscilloscope timebase to display two to five cycles of the chosen signal. These are general guidelines since the types of signals encountered are so widely varied that some judgement by the developer will be required on some signals. Example: If a signal is a narrow pulse, the setup of the oscilloscope will be different depending on which aspect of the pulse is the most important: the pulse width, or the pulse period.

4.4 Making Waveform Boundary Files. The Waveform Boundaries used in the determination of acceptable signals are generated by double-clicking on the MAKE BOUNDARIES icon in the AN/PSM-93(V) Development Folder. This software asks the developer what the percent tolerance will be used to determine the boundary values. These values are entered in the appropriate text boxes on the opening window by clicking the up and down arrows next to each tolerance box. The values range from 1 to 5, with 1 being the default. This is the value normally used. All waveforms that have been learned will then have boundary files generated for them. The MAKE BOUNDARY FILE information box informs the developer what waveform file is currently being processed. This box will disappear when the boundary generation is complete.

4.5 Verifying The SES Package. The completed SES package includes several different files generated by the developer. The SES verifier checks that all of the files required to use the SES are present. The verification process is performed by double-clicking on the MAKE BOUNDARIES icon in the AN/PSM-93(V) Development Folder. This software generates a log file called VERIFY.LOG in the C:\TESTER\ subdirectory. If any inconsistencies are found, they are reported in this file. Some examples include;

1) Not learning the waveform for a pin designated in the schematic.
2) A component was called out in the LAYOUT LEARN software that does not have an associated library element.

Any inconsistencies must be resolved before the SES package can be used. Inconsistencies can be corrected by using the information in the VERIFY.LOG file to identify the problem, and generating the required files.

After the verifier process has been completed and there are no inconsistencies in the SES package, the ACTIVE CIRCUIT EVALUATION icon is double-clicked in the AN/PSM-93(V) Development Folder. This will run the ACE software directly, allowing the waveforms to be compared to the boundary information in the SES package. This also checks that the correct waveform has been learned for each pin, and that the electrical point on the schematic matches the physical location on the layout. If any waveforms are incorrect, that waveform must be checked for accuracy.

4.6 SES Quality Assurance and Distribution. In order to maximize the benefits realized from the development of SESs, it is requested that all Fleet developed SES packages, whether partial or complete, be forwarded to the following address for Quality Assurance processing and distribution (**copying the entire contents of the C:\TESTER directory will capture all of the SES files. Copying all SES files to floppy disk can be accomplished automatically by double-clicking on the SES PACKAGE BACKUP icon in the AN/PSM-93 (V)1 Development Folder**)

COMMANDER
Phil Hoffsetz, Code 6037, Bldg 2917
NAVSURFWARCENDIV
300 Highway 361
Crane, IN 47522-5000

SESs should be mailed on 3.5" floppy diskettes, or transmitted electronically as compressed (zipped) e-mail attachments to "psm93@homer.crane.navy.mil", with the following minimum information:

- Name, rank/rate, organization, address, phone, etc of the developer.
- Major system or assembly nomenclature, part number, NSN/NIIN, and revision information descriptive of the UUT used, or system from/for which the SES was developed.

SECTION 5

SYSTEM TROUBLESHOOTING TIPS

5.1 Overview. Generally, problems with a computer occur due to incompatibilities within the software, the hardware, or both. Occasionally problems with a computer are due to operator error. Rarely, there are times when computer hardware fails catastrophically, causing loss of data on the hard drive. Listed below are some steps which can be taken to bring the computer back up. If these steps don't produce a satisfactory result, the computer must be turned into local sources for repair.

5.2 Re-Boot. Sometimes a computer will get some code or a combination of codes that will lockup the CPU. Other times a printout will get lost in the I/O (Input/Output) stage. Initiating a warm boot by pressing the CTRL + ALT + DEL keys or turning off power for a few seconds will resolve most problems.

NOTE: Do not re-boot without some thought to data loss. If one of the re-boot procedures is used, all data in the current memory will be lost. If the user has performed a great deal of work, this will be lost unless it was saved periodically.

5.3 Input/Output Problems. These problems generally occur whenever a printout is requested or during keyboard operations.

5.3.1 If the user requests a printout through a program, an error message may appear or the computer may lock-up. If a printout is called while in the DOS environment, a message will say XXXX.XXX IS CURRENTLY BEING PRINTED. Accidentally pressing the PRINT SCRN key may lock up the computer also.

5.3.2 The keyboard may cause a computer lockup. Pressing the wrong series of keys, or key combination may lockup the computer. The user might want to ensure the keyboard is properly plugged into the computer.

5.3.3 A final I/O problem could occur with the monitor. If the computer appears to be running, but you can not see anything on the monitor, check the power switch first. Other things to consider: is the power plug seated properly, is the monitor plugged into the proper connector on the CPU, are the intensity and brightness controls at the proper setting, and check for monitor capabilities as set forth by the program in use.

5.4 Boot Failure. This type of failure usually occurs whenever the computer is re-booted or turned on. The failure may be due to a hard drive problem, CPU/Memory failure, keyboard not plugged in properly, or some form of shock affecting one or the other.

5.4.1 **This should be done only in extreme cases! No re-format or re-installation should be attempted before contacting the Technical Support contact specified in paragraph 1.11 of this document.** For a suspected hard disk problem, insert the MS-DOS floppy #1 into drive A: and reset the computer, first turning the power off, waiting a few seconds and turning the power on. If the computer boots and the user gets a DOS prompt on the monitor, the hard drive may have failed. At this point in time, try to get to drive C: by typing **C**: and the ENTER key. If you get the prompt C:\>, type **dir** and the ENTER key. If a directory listing is displayed on the screen, you might have access to the hard drive. The user now needs to get a backup of key files. Once the backup is obtained, the user is now ready to try to re-format and re-install the entire AN/PSM-93(V) system.

5.4.2 For a keyboard problem, turn of all power to the computer; gently remove the keyboard connector at the back of the computer; re-seat the connector; then restore power.

5.4.3 If a CPU/Memory failure occurs it is usually due to shock, over-heating, voltage failures and spikes, or corrosion on the circuit board connectors. They may appear as Parity Errors or a panic message. A qualified electronics technician, after de-energizing the unit in a static free environment, may try to re-seat all the circuit boards, connectors, and the keyboard connector.

NOTE: Removing the memory battery from the CPU will cause a loss of the system BIOS. It is recommended that the battery remain connected at all times. If the BIOS is lost, the user can call the Technical Representative to restore the data.

5.4.4 All the above procedures are for Emergency Cases Only. If the user is unsure of any of these procedures or the computer does not respond, call the Technical Representative listed in section 1.11 for help.

APPENDIX A

COMMON TERMS

AN/PSM-93(V) Terms. Listed below are some of the terms common to the associated software packages and the development of SESs via the AN/PSM-93(V):

ACE - Active Circuit Evaluation: A software package that uses graphic, text, and information files to aid an operator in determining the conditional readiness of a module or subsystem.

BOUNDARY - A boundary is an upper or lower voltage or timing limit that a passing waveform cannot exceed.

BOUNDARY ENVELOPE - The area between, or enveloped by, the voltage and timing boundaries for a given waveform.

BOUNDARY FILE - A computer file generated by the user through the oscilloscope software in which acceptable upper and lower limits have been established. Subsequent waveforms must fall within this user developed boundary envelope in order to be considered as "passing".

GPETE - General Purpose Electronic Test Equipment.

LOG FILE - A file used in the ACE software to track Development or Test data such as points tested, points learned, etc.

MAP FILE - A file used by the ACE software to establish the relationship(s) between the geographic (x,y) coordinates, pin or node labels, or bitmap pictures for the Schematic or Layout files in a Signal Evaluation Set.

NODE - a point in a circuit where two or more component pins are connected together making them electrically equivalent.

NG² - Next Generation GPETE: Instrumentation that functions as GPETE, but is designed to reside in a portable PC and can be used either as stand-alone instrumentation, or under computer control.

PCB - Printed Circuit Board.

SES - Signal Evaluation Set: A group of files residing on a PC that are used in conjuction with ACE software and NG² hardware to aid in determining the conditional readiness of a module or subsystem. The files in a SES are unique to, and descriptive of, a specific Unit Under Test, assembly, or system.

UUT - Unit Under Test: The module, assembly, or system that is currently being checked for operational condition.

APPENDIX B

SAMPLE FILES AND DOCUMENTS

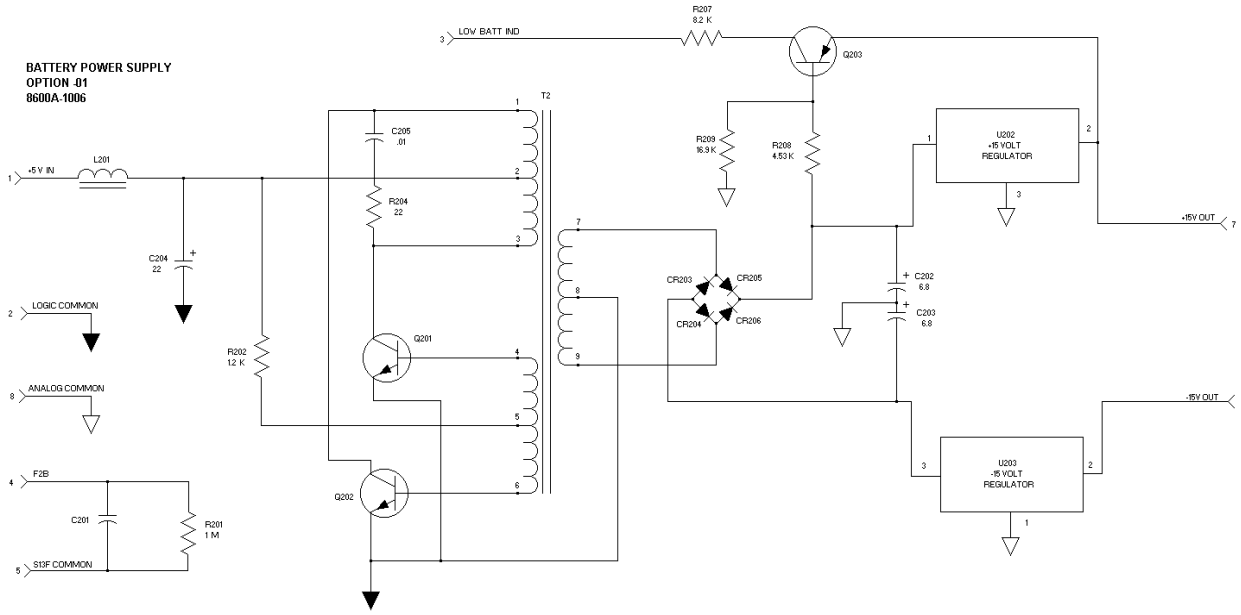


Figure 15. Example *Schematic Diagram (GRAPH01.PCX)*

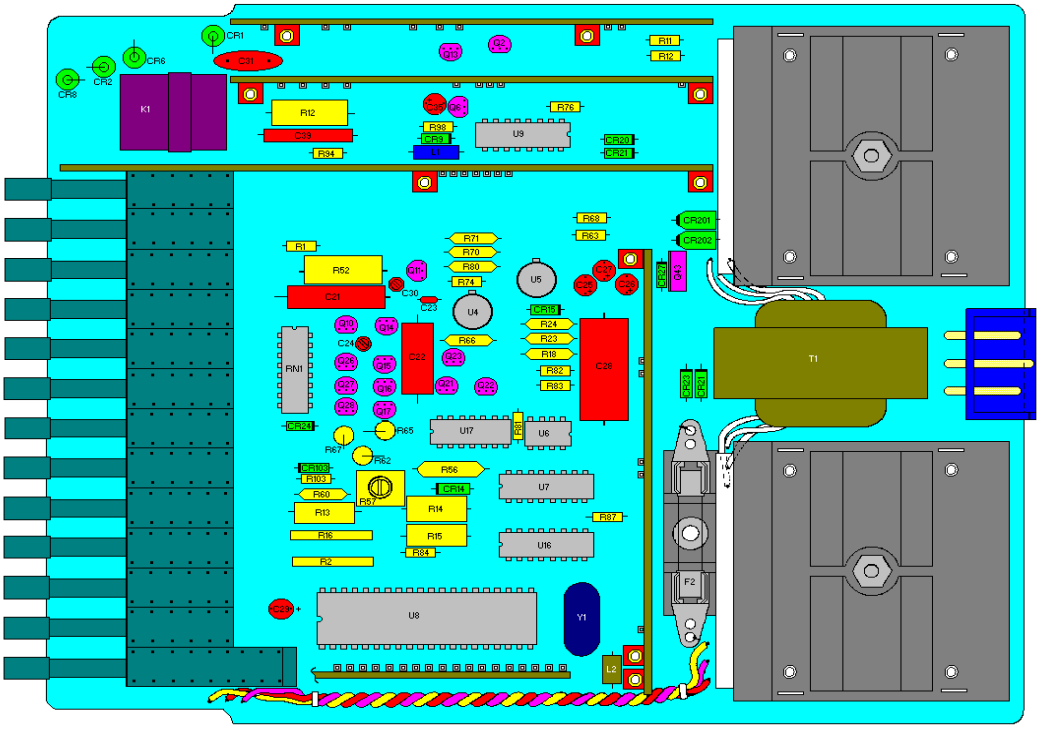


Figure 16. Example *Layout Diagram (LAYOUT.PCX)*

EQUIPMENT:	EIC:	APL:
R-2124/SLQ-32 (V) RCVR, CM	NONE	81022074

CCA Part Number:	Rev:	CCA Nomenclature:
928202-1	E	D/A CONTROL & CALIBRATION

REFERENCE DESIGNATION	CAGE/PART NUMBER	STOCK NUMBER
A1	49956/845763-1	5962-01-043-8326
A2	49956/845629-1	5962-01-237-3399
A3	49956/845629-1	5962-01-237-3399
C1	81349/M39006/09-8295	5910-01-119-4306
C2	81349/M39006/09-8318	5910-01-119-4324
C3	81349/M39006/09-8318	5910-01-119-4324
CR2	81349/JANTXV1N4148-1	5961-01-055-8655
CR3	81349/JANTXV1N4148-1	5961-01-055-8655
J1	49956/276-6067P029	5935-00-032-8399
R1	81349/RNC55H2741FR	5905-00-403-4447
R2	81349/M39015/3-006PP	5905-00-496-3601
R3	81349/RNC60H7503FS	5905-00-576-5054
U1	49956/845643-1	5962-01-372-6623
U2	49956/845526-1	5962-01-341-0544
U3	49956/845526-1	5962-01-341-0544
VR1	81349/JANTXV1N759A	5961-01-055-8655

Figure 17. Example *Parts List File (BOARD.DOC)*

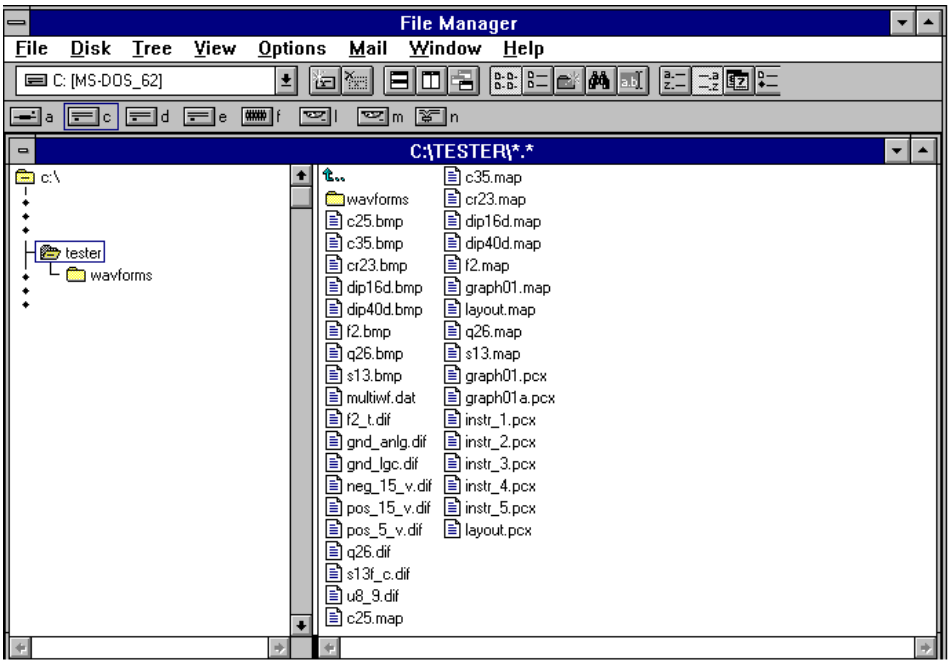


Figure 18. Example *Directory Tree (C:\TESTER*.*)*

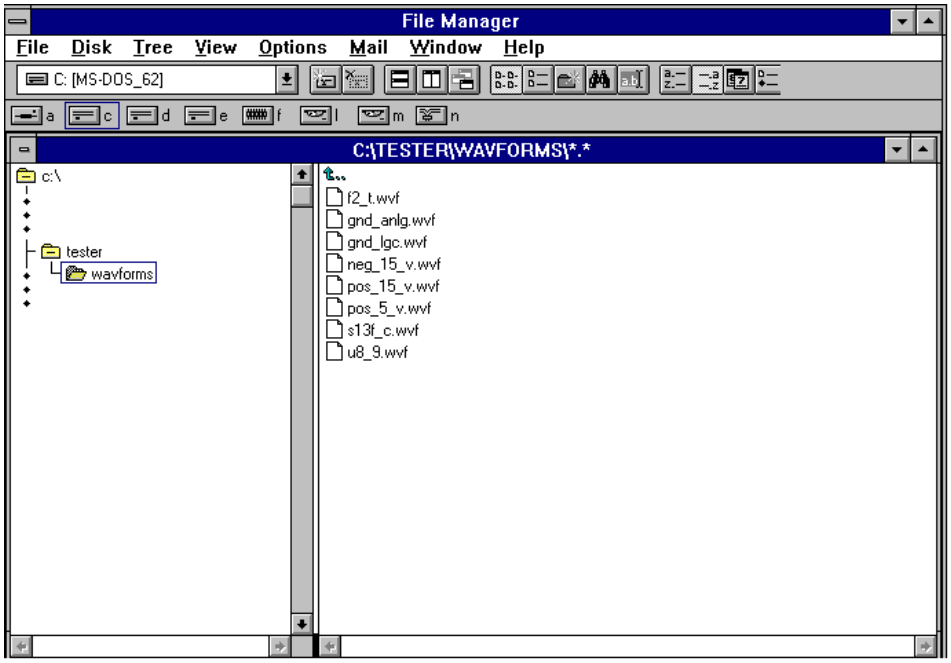


Figure 19. Example *Directory Tree (C:\TESTER\WAVFORMS*.*)*

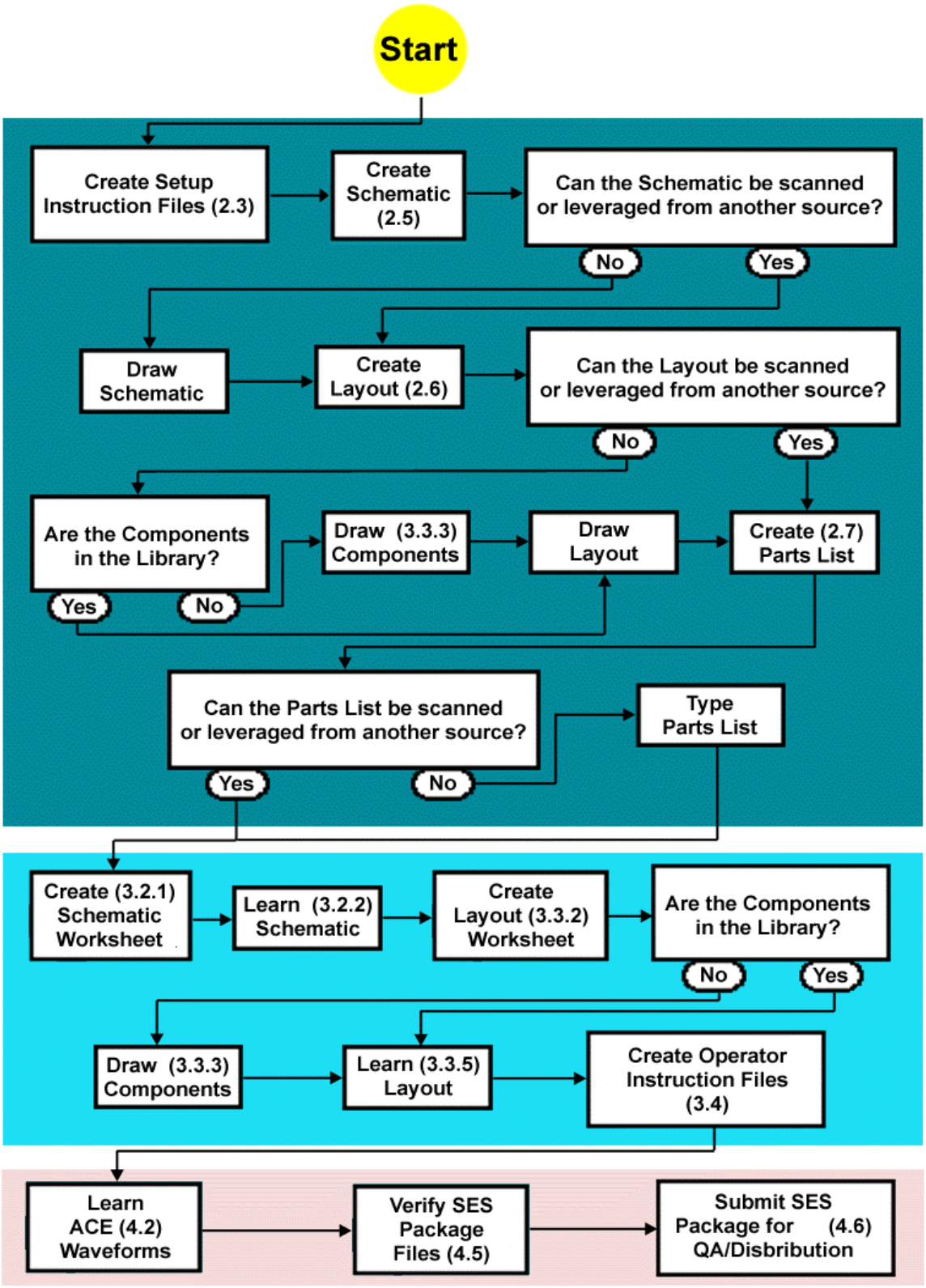
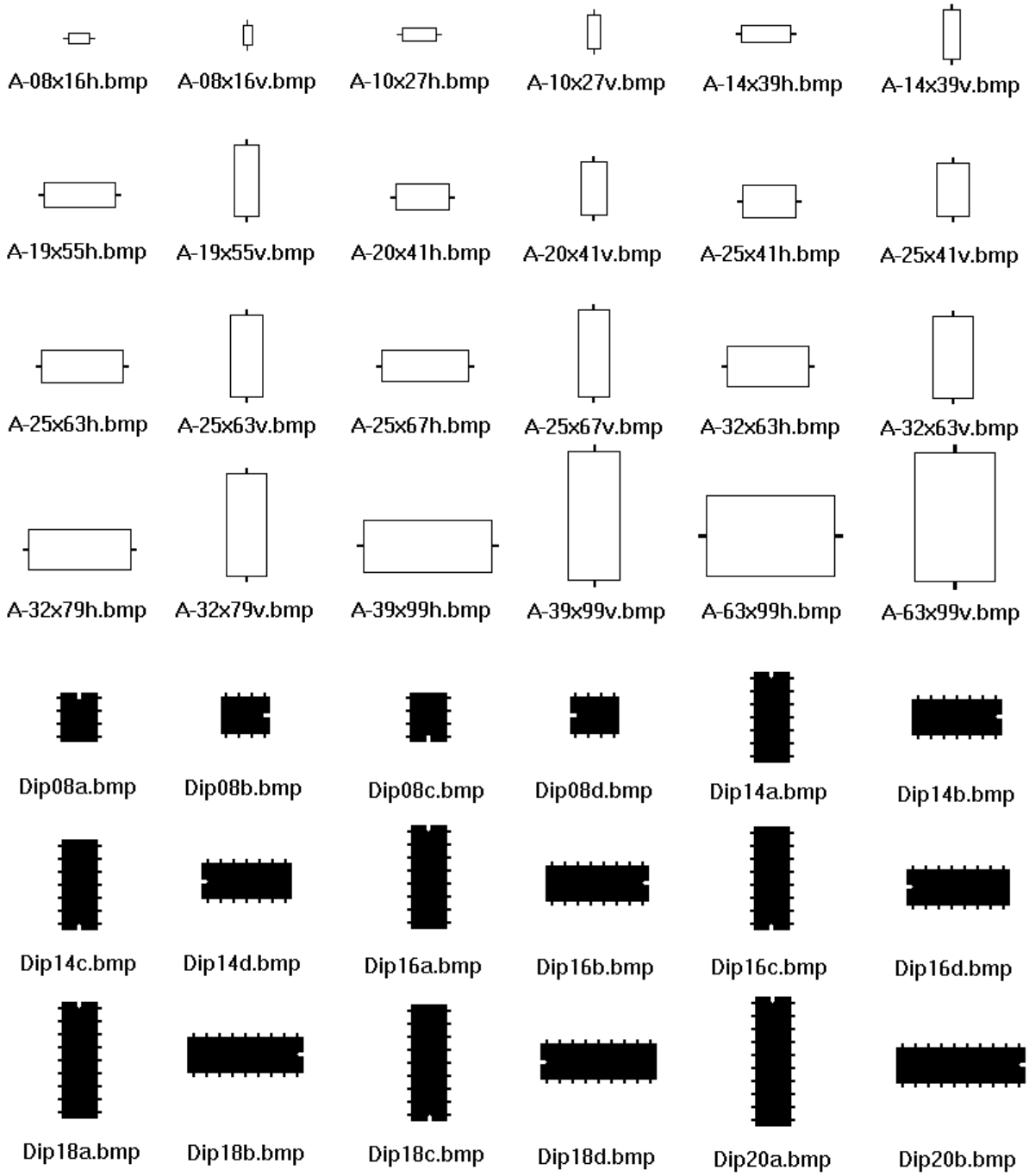


Figure 20. ***SES Development Flowchart***

APPENDIX C

COMPONENT LIBRARY CATALOG




Dip20c.bmp


Dip20d.bmp


Dip24a.bmp


Dlp24b.bmp


Dip24c.bmp

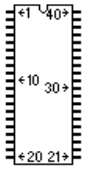

Dip24d.bmp


Dip28a.bmp

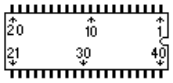

Dip28b.bmp


Dip28c.bmp

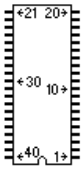

Dip28d.bmp



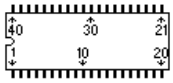
Dip40a.bmp



Dip40b.bmp



Dip40c.bmp



Dip40d.bmp


FP10a.bmp


FP10b.bmp


FP10c.bmp


FP10d.bmp


FP14a.bmp


FP14b.bmp


FP14c.bmp


FP14d.bmp


FP16a.bmp


FP16b.bmp


FP16c.bmp

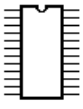

FP16d.bmp

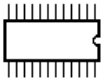

FP20a.bmp

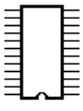

FP20b.bmp

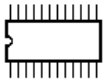

FP20c.bmp


FP20d.bmp


FP24a.bmp


FP24b.bmp


FP24c.bmp


FP24d.bmp

(Top View)
Opamp.bmp


Soic08a.bmp


Soic08b.bmp


Soic08c.bmp


Soic08d.bmp


Soic14a.bmp


Soic14b.bmp


Soic14c.bmp


Soic14d.bmp


Soic16a.bmp


Soic16b.bmp


Soic16c.bmp


Soic16d.bmp


Soic16wa.bmp


Soic16wb.bmp


Soic16wc.bmp

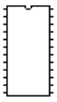

Soic16wd.bmp


Soic20a.bmp

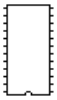

Soic20b.bmp


Soic20c.bmp

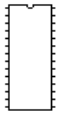

Soic20d.bmp


Soic24a.bmp

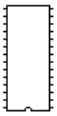

Soic24b.bmp


Soic24c.bmp


Soic24d.bmp


Soic28a.bmp


Soic28b.bmp


Soic28c.bmp

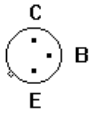

Soic28d.bmp

To-220a.bmp

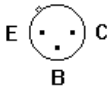
To-220b.bmp

To-220c.bmp

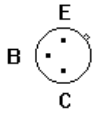
To-220d.bmp



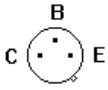
To-39a.bmp



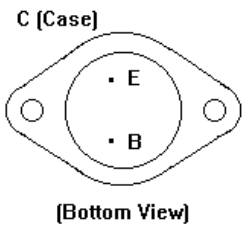
To-39b.bmp



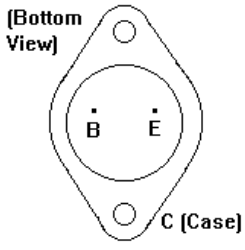
To-39c.bmp



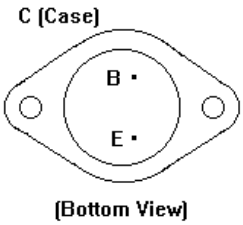
To-39d.bmp



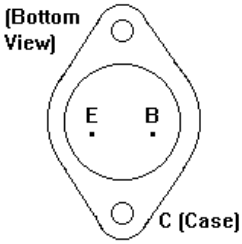
To-3a.bmp



To-3b.bmp



To-3c.bmp



To-3d.bmp

To-46a.bmp

To-46b.bmp

To-46c.bmp

To-46d.bmp

To-5.bmp

To-92a.bmp

To-92b.bmp

To-92c.bmp

To-92d.bmp

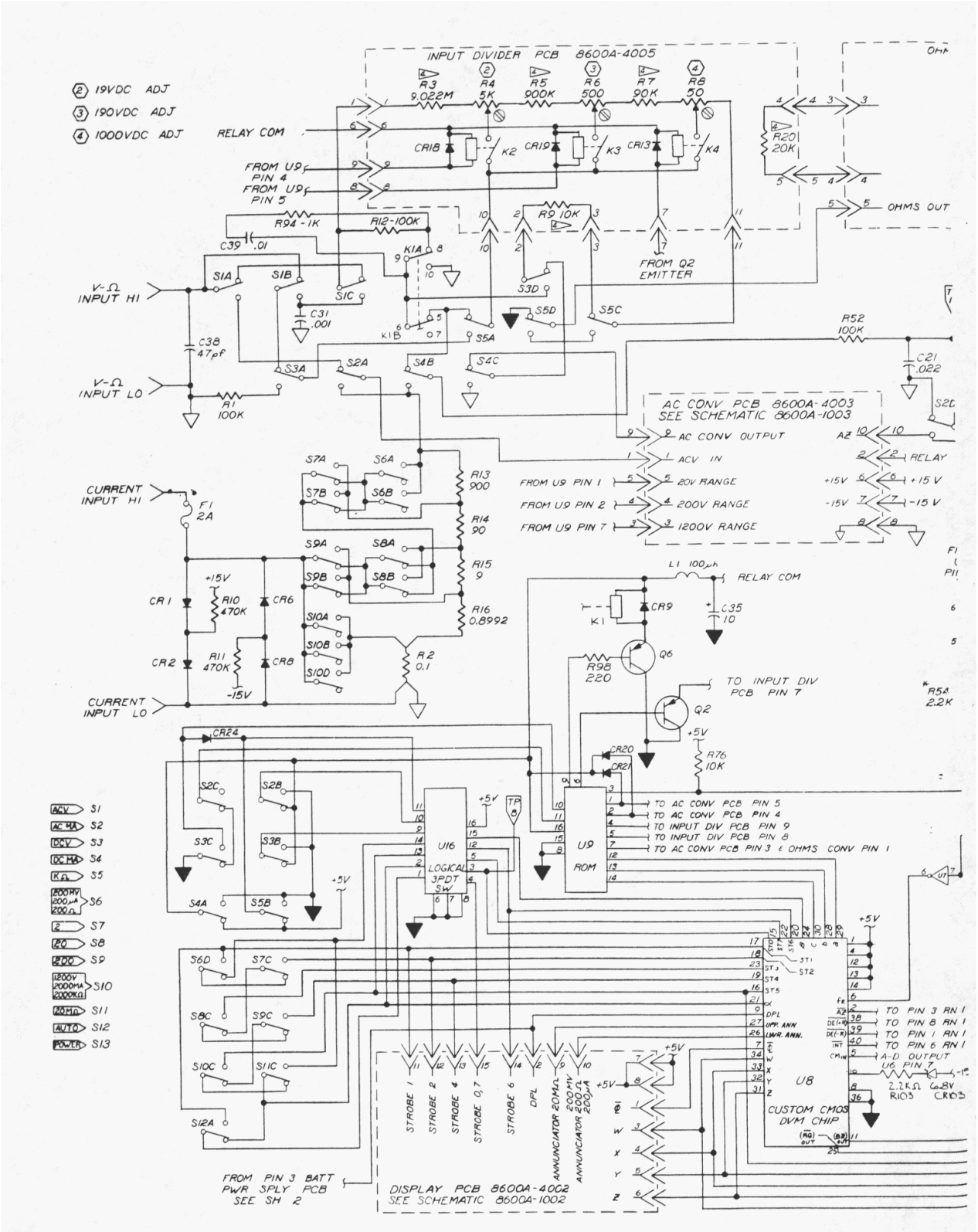


Figure 21. ***Schematic Scan Sheet A***

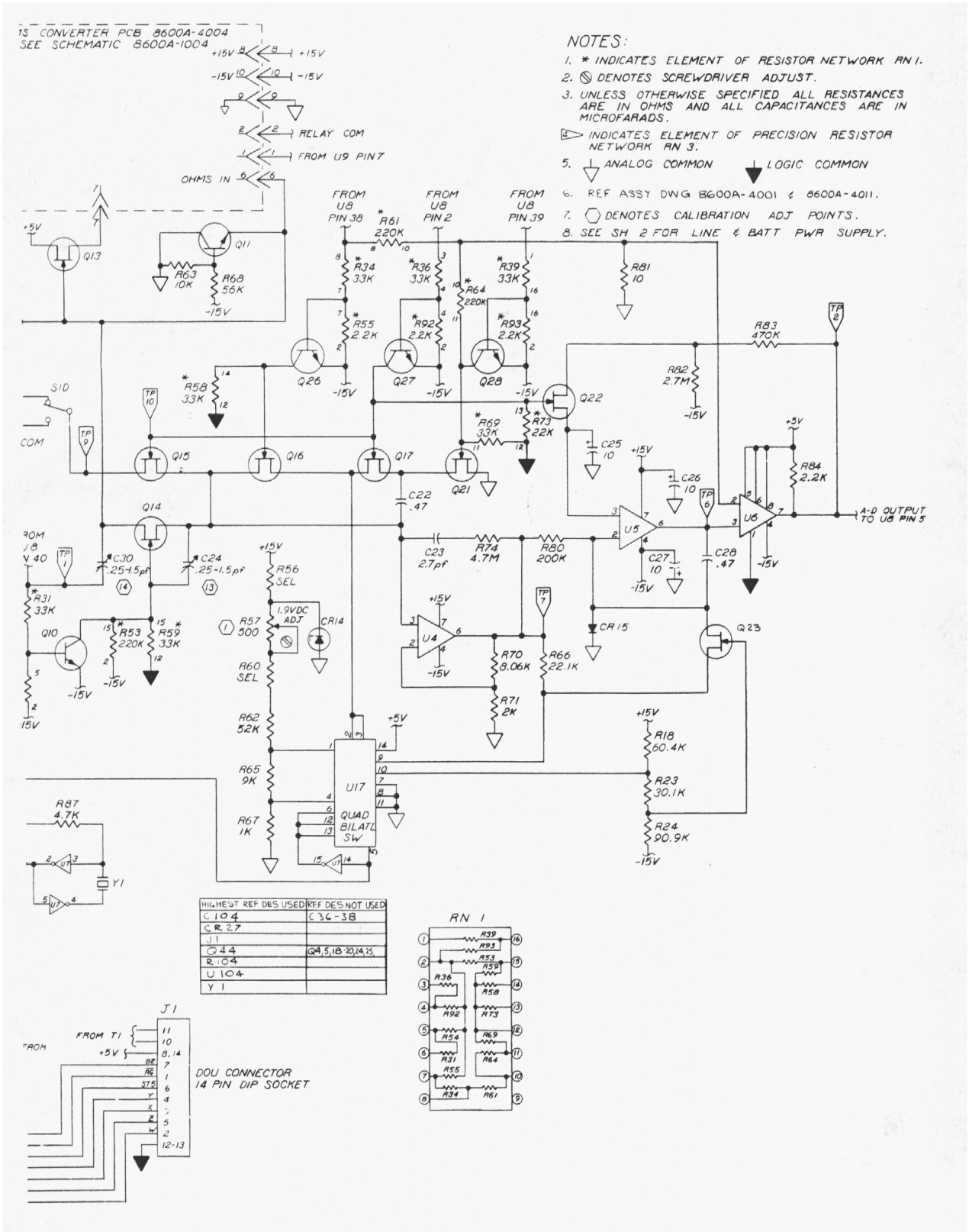


Figure 22. ***Schematic Scan Sheet B***

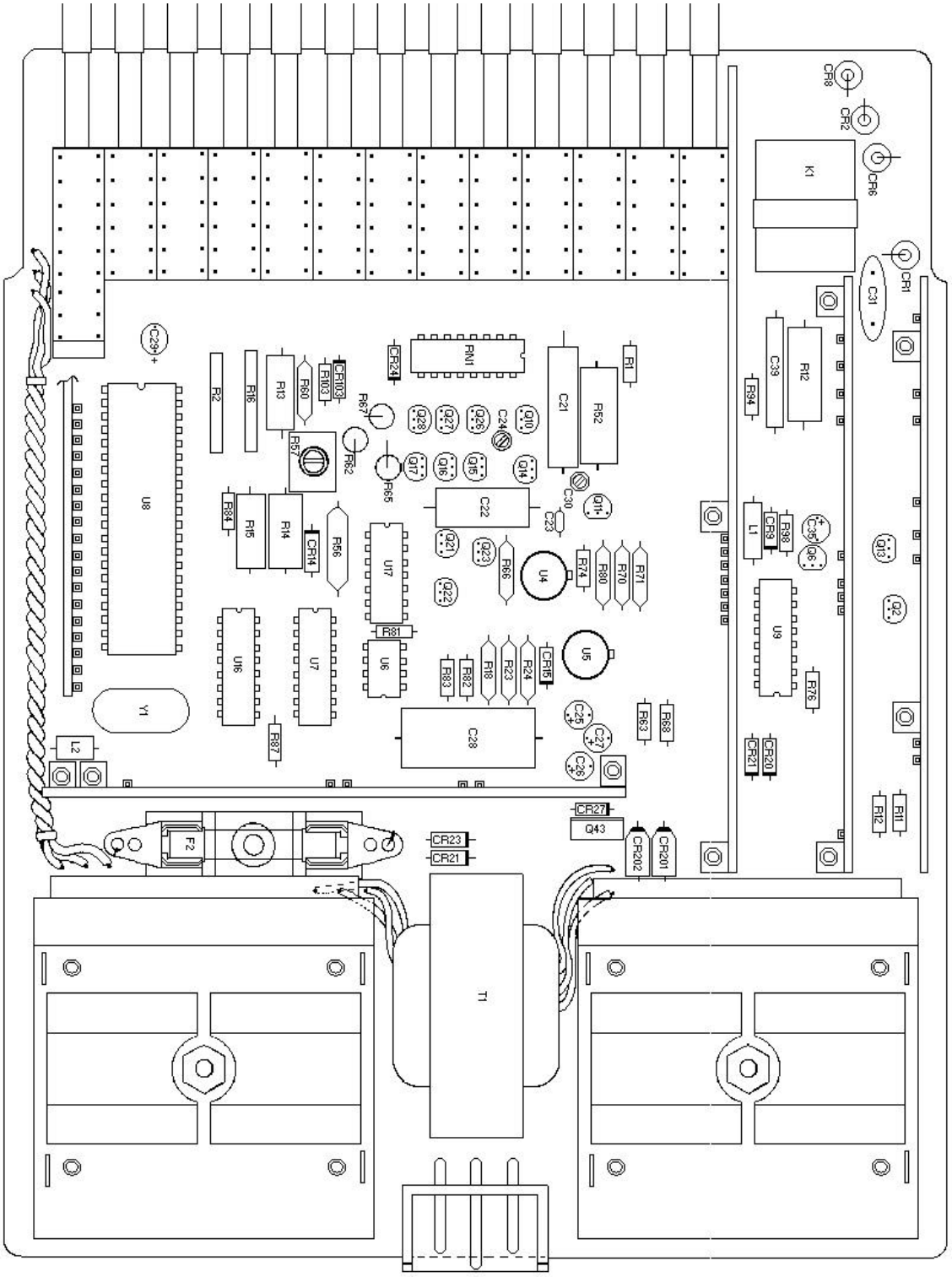


Figure 23. ***Layout Scan Sheet***