1991

An automated non-destructive test system

Connie Lee Lohr Bishop
University of Dayton

AN AUTOMATED NON-DESTRUCTIVE TEST SYSTEM

Thesis

Submitted to

Graduate Engineering & Research
School of Engineering

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree

Master of Science in Electrical Engineering

by

Connie Lohr Bishop

UNIVERSITY OF DAYTON

Dayton, Ohio

April 1991

92 00423

AN AUTOMATED NON-DESTRUCTIVE TEST SYSTEM

APPROVED BY:

[redacted]

Anthony J. Evers, M.S.E.E.
Advisory Committee, Chairman

[redacted]

Gary A. Thiele, Ph.D.
Associate Dean/Director
Graduate Engineering & Research
School of Engineering

[redacted]

Gordon A. Sargent, Ph.D.
Dean, School of Engineering

ABSTRACT

AN AUTOMATED NON-DESTRUCTIVE TEST SYSTEM

Name: Bishop, Connie Lohr
University of Dayton, 1991

Advisor: A. J. Evers

The Automated Non-destructive Test System (ANTS) was specifically designed to test a transducer at EG&G Mound Applied Technologies. The tests consist of verifying that the circuits are wired correctly and that critical faults do not exist. The proper function and safety of this device require that every unit built pass these tests or be rejected.

This paper will describe the hardware and software design criteria and how they were implemented in the fabrication and operation of the test system. The test system consists of commercial equipment integrated with custom designed hardware. A Hewlett Packard computer automates the system by interfacing controlling software with the hardware. A software program written to control the function and the timing of the tests also provides a means for data acquisition.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

FIGURE

ACKNOWLEDGMENTS

I would like to thank those who supported me on this project. Especially Leander Hall for fabricating the test system; my supervisor Keith Ohler and my department manager J. Fred Madden.

A special thank-you must go to my husband Todd for giving me the encouragement and understanding I needed.

CHAPTER I

INTRODUCTION

The reliability of components produced by the weapons industry is essential. There are few businesses where the survival of so many, the defense of a nation, depends as heavily on the proper function of its product. Non-destructive testing is a means by which a device can be analyzed without impairing its functional qualities. The results of these tests determine whether the device is acceptable and can be used in the next stage of assembly.

EG&G Mound Applied Technologies (The Mound), located in Miamisburg, Ohio, is a Department of Energy (DOE) installation. The Mound is a production facility of weapon system components. In addition to production, non-destructive and destructive testing of these components is also performed at this site. The scope of this paper will describe in detail the Automated Non-destructive Test System (ANTS) designed to perform various tests on a transducer that is part of a system sub-assembly.

The Automated Non-destructive Test System is designed to verify critical wiring in the transducer. Accurate measurements and assessments must be made for several different circuits in the unit to be assured of transducer reliability. For safety and operational purposes, ANTS is

also required to verify that there is sufficient insulation so that no arcing will occur between the circuitry and the case of the transducer when exposed to an input signal. This is done for nine different test points. The system incorporates a computer to control the function, timing, order, and assessment of these tests.

The controlling computer controls and interrogates other equipment via a parallel bus. Custom designed hardware in the test system is interfaced by general purpose input/output cards in the computer. Test results are first stored in memory and then transferred to storage on a floppy disk. The data is electronically transmitted to Sandia National Laboratories where it is archived. Numerous software routines execute the various test and data storage sequences. User friendly operator interface routines expedite the testing procedure since a large number of transducers are tested by the test system.

Connecting to the test sample requires special fixturing that must be durable, reliable, and reusable. The design of the fixture ensures that adequate electrical contacts for testing are made without damaging the connectors on the unit.

Since the test results can be only as good as the system that produces the results, the validity of ANTS must be verified. Calibration of the equipment and electronics in the system is performed and documented on a periodic basis. This was taken into consideration in the hardware design of the system. Separate from the testing software,

calibration software was written to aid in this task. In addition to periodic calibrations, simulators were designed to be used with the system. These simulators emulate the test component and are used to verify that the test system is functioning properly. The simulators are calibrated as part of the test system and then used by the operator at the beginning and the end of every test session.

These are the considerations that were taken into account when designing the non-destructive tester. Design considerations and decisions for fabrication and operation of ANTS will be described in the following chapters.

CHAPTER II

TRANSDUCER OVERVIEW

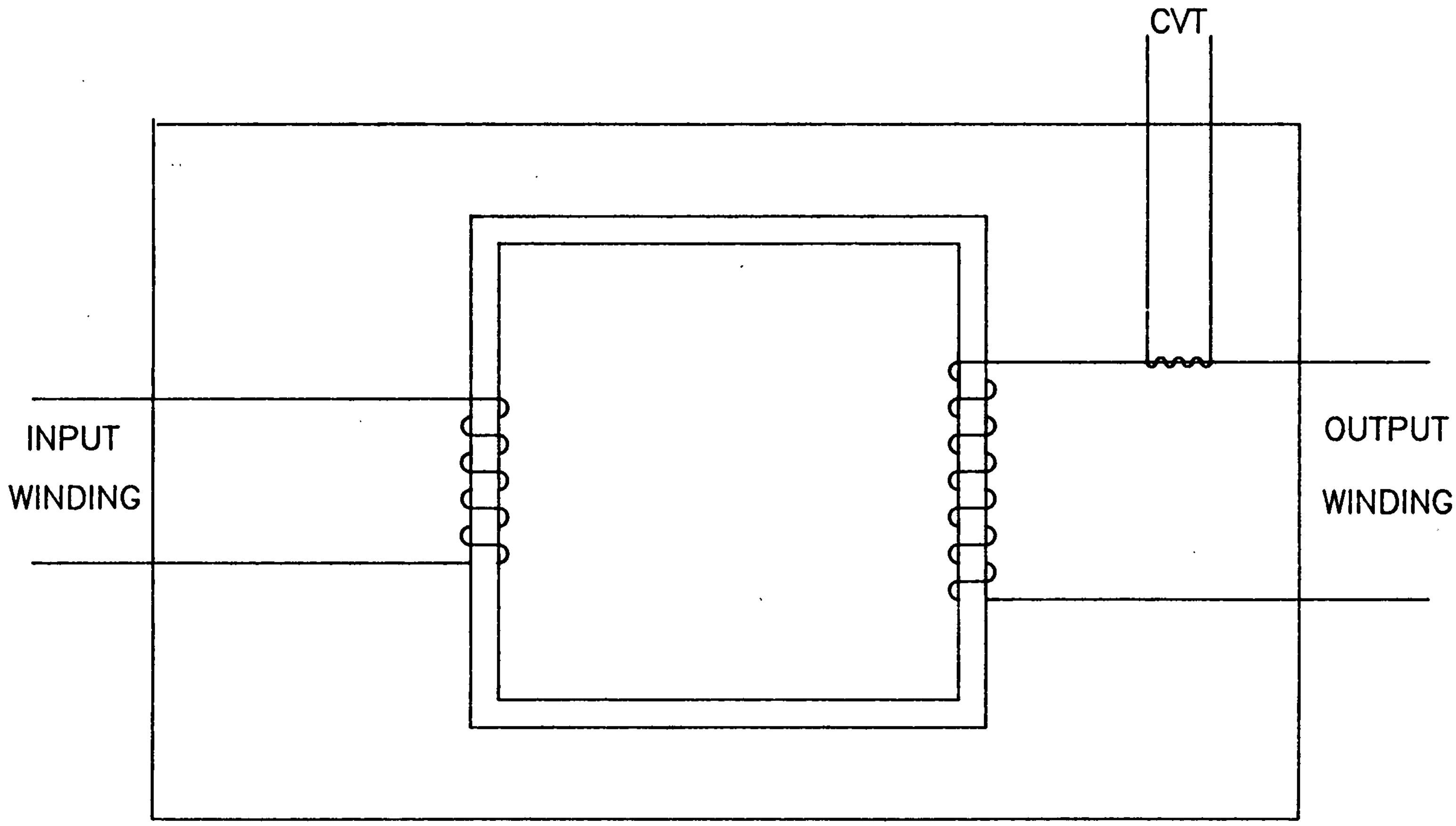
The transducer which the Automated Non-destructive Test System is designed to test is based on transformer technology. Figure 2.1 is a simplified diagram of the device. It is a transformer with a primary winding and a secondary winding.

On one leg of the secondary winding is a Current Viewing Transformer (CVT). This device is a means of monitoring the function of the transducer during destructive testing. The current is calculated from the voltage produced by the induced electromotive force (emf). The emf is the number of turns in the coil times the derivative of the magnetic flux with respect to time.[1]

$$emf=Nd\phi/dt$$

The calculated current is based on parameters supplied by the manufacturer of the CVTs. This current signal is analyzed to determine if the transducer functioned properly. The Current Viewing Transformers are critical in

[1] W. H. Hayt, Jr., Engineering Electromagnetics (New York: McGraw-Hill, 1981) ,p. 347.



FIGRUE 2.1: TRANSDUCER

predicting the performance of an entire production lot. For this reason, the CVTs are 100 percent non-destructively tested on ANTS for proper static parameters.

The transducer also contains an explosive device that has input leads which are not shown in Figure 2.1. These leads must be electrically measured for safety reasons. If there is a fault in this circuit, the unit could malfunction.

Additional non-destructive type tests are performed to determine the stability and electrical isolation of the case. The entire device is packaged in a metal case.

Function

The transducer functions when a direct current is applied to the primary, or input, winding of the transformer. This current produces a steady magnetic field. By the integral form of the Biot-Savart Law the magnetic field intensity H is the sum of the differential current elements, Idl, over a closed surface.

$$H=\int((IdlXa_R)/(4\pi R^2))$$

where the cross product with a_R defines the direction of the field and R is the distance from the differential current element to the surface of the magnetic field.[2]

[2]Ibid.,p240.

The core of the transformer becomes saturated with magnetic flux lines ϕ. Magnetic flux is defined by

$$\phi=\int_{s} B\,ds$$

where B is the magnetic flux density which is the product of the magnetic field intensity and the permeability μ of the material of the core.[3]

$$B=\mu H$$

Once a steady magnetic field is established, the explosive device is initiated by an external signal which is not related to the primary or the secondary windings. The resulting explosion causes the established magnetic field to collapse. This produces a sudden change in the magnetic flux which causes an induced current in the secondary, or as it will be referred to from now on, the output winding. The timing sequence is critical since the signal from the output winding must reach its destination before the circuit is destroyed. The abrupt burst of current in the output winding of the transformer is the desired end of the transducer's function.

---

[3] Ibid.,p319.

Tests

The non-destructive tests that ANTS performs on this transducer reinforces the reliability of the components that go into the weapon system. As described above, the proper function of the transformer type circuit is critical.

The first set of parameters that ANTS tests is the resistance of all the coils in the unit. This consists of connecting to six different points on the transducer and taking three measurements. The resistance readings are automated by the controller which interfaces to a digital ohmmeter in the system. The technique by which these resistance values are obtained is known as a four wire resistance measurement and results in negligible error due to the test leads. In resistance readings below ten ohms, the resistance of the test leads can significantly contribute to the measurements which would result in erroneous data. This is a factor which must be taken into consideration since the resistance of the coils fall within this range.

This resistance is used primarily to verify that the number of turns in each coil meet the required specification. Since the resistance is a function of the length of the wire in the coil, the number of turns in the coil defines the length, therefore specifying the required resistance:

$$R=L/\sigma S$$

where σ is the conductivity of the material, S is the cross-sectional area and L is the length of the wire.[4] Since the output current is dependent on the number of turns in the coil as described earlier, a faulty winding in the transformer could result in a device that would not function properly. The resistance of the Current Viewing Transformer is also verified since the proper calculation of current is needed to correctly indicate the function of the transducer during destructive testing. A resistance reading out of range could also indicate other types of faults in the circuit. Whenever a resistance falls out of range, the unit is rejected and is analyzed to determine the cause of improper function.

The next type of test to be addressed is referred to as Insulation Resistance (I.R.). This is executed by applying a high voltage across two terminals and verifying that the current flow, known as the leakage current, is a minimal value. This will ensure that this circuit is characterized by a very high resistance. A test of this nature provides concrete data to verify that these points are electrically isolated.

There are nine of these tests performed. Three of them are accomplished by applying high voltage across one terminal of each coil and the case of the transducer. This

[4] Ibid.,p.136.

includes the input and the output windings and the CVT. The next test is the application of the high voltage across an input signal lead to the explosive device and the case of the transducer. The remaining five tests consist of applying the high voltage across other points that must be verified to be electrically isolated. These other points are not described in this paper for security reasons. However, they are mentioned here since ANTS performs these tests. Each of these tests is performed independently of all the others. During each application of the high voltage, a current comparator circuit in the test system is monitored by the computer. If the current is too high, the comparator circuit recognizes this and sends out a signal indicating the condition. A failure of this test implies that the points of voltage application are not sufficiently insulated from each other. This can result in an unstable device and make the operation of the transducer questionable.

The establishment of continuity and insulation resistance is a critical factor when determining the reliability of the transducer. The Automated Non-destructive Test System is a vital supporting part of the production of such units.

CHAPTER III

HARDWARE

As Figure 3.1 shows the Automated Non-destructive Test System is a single bay bench top tester. The controlling computer, printer and safety chamber are all within close proximity of the tester cabinet.

The hardware of the test system consists of both commercial equipment and custom designed circuits. The selection of purchased equipment was based on specifications of accuracy, precision and safety limits required by the transducer. Figure 3.2 is a block diagram of the test system. Integrating this equipment with the other hardware components is what gives ANTS the capability of performing the required tests as described earlier.

Equipment

There are five major pieces of commercial equipment used in ANTS to take measurements, output signals, collect and store data, and control the process. They are a Hewlett Packard Controller, a Keithley Digital Micro-Ohmmeter, a Hewlett Packard Digital Multimeter, a Hewlett Packard Disk Drive, and a Hewlett Packard Printer.

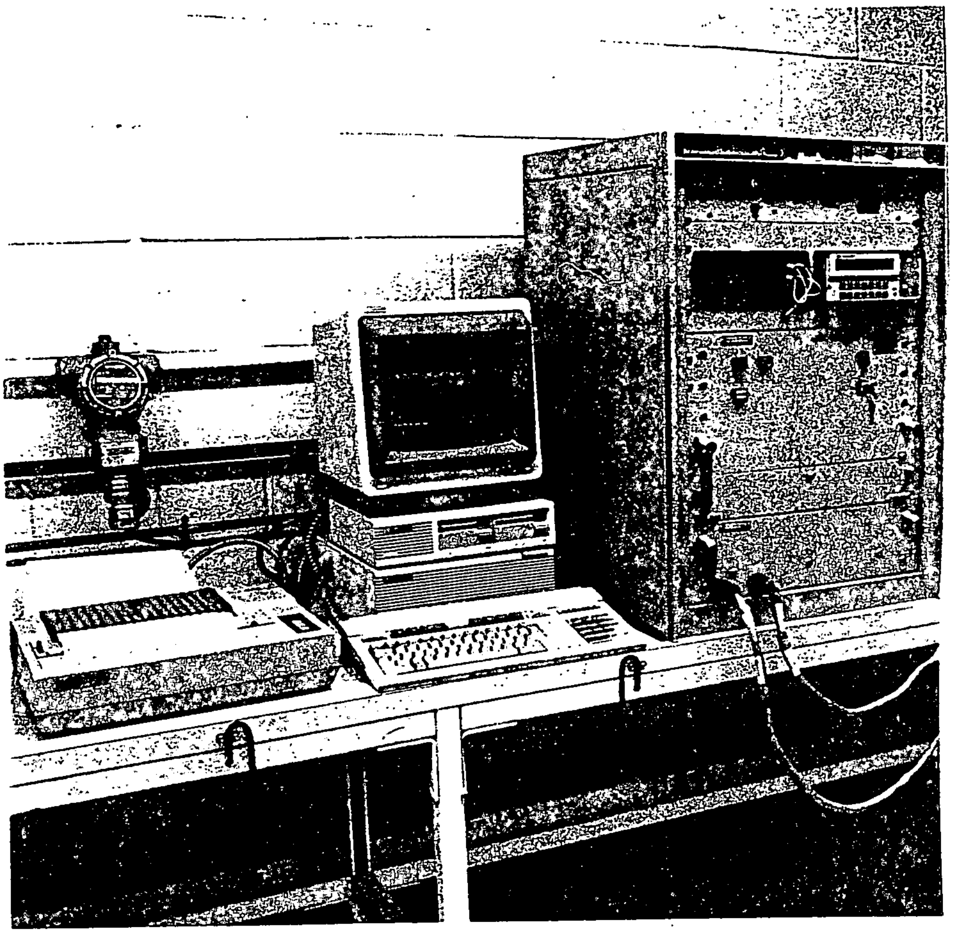


FIGURE 3.1: AUTOMATED NON-DESTRUCTIVE TEST SYSTEM

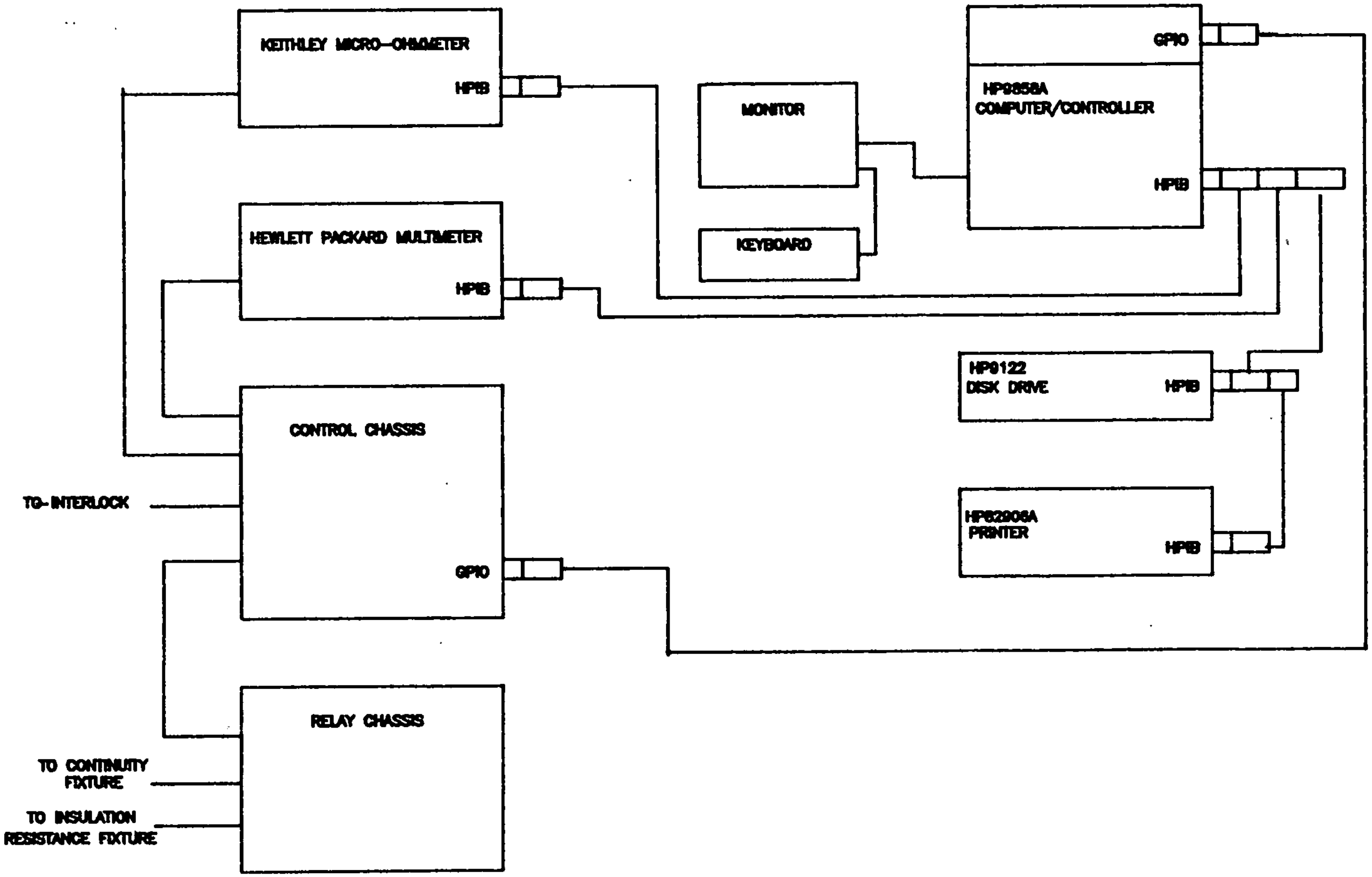


FIGURE 3.2: BLOCK DIAGRAM

Controller

The computer which controls the test system is a Hewlett Packard 9000 family 332 workstation. The Model 332 is based on a MC68030 processor and has a 16.7 megahertz clock. The processor is characterized by a 32 byte data cache and a 32 byte instruction cache. One megabyte of RAM is accessed by a 32 bit data bus and a 32 bit address bus.

The computer interfaces to other equipment via the Hewlett-Packard Interface Bus (HPIB). All electrical and timing requirements of the IEEE 488 standard parallel interface are satisfied by this bus. Each piece of equipment is assigned an address on this bus. The computer uses this addressing scheme to select a piece of equipment to listen or to talk. This address is selected by setting DIP switches on each device.

The custom designed hardware is controlled and monitored by General-Purpose Input/Output (GPIO) Interface lines of the computer. The GPIO Interface provides a 16 bit bi-directional binary data transfer technique. It has the capability of transmitting data on the output lines at the same time the computer is receiving data on the GPIO input lines. The computer system used in ANTS is configured with two sets (cards) of GPIO Interface lines. Each card has an interface select code which is selected by setting a DIP switch that provides a means for the computer to differentiate between the two.

The data on each input line is received by an exclusive OR gate on the GPIO Interface. The hardware

designed to transmit data indicating the state of various process conditions uses a double pole switch. Figure 3.3 shows this configuration. The interface requires at least three volts on its input lines to register as a binary one. A low must be indicated with no more than 0.7 volts on the input lines. As Figure 3.3 shows, five volts and ground are used to establish the desired states. Switching is accomplished by a relay or a key switch.

Each of the GPIO data-output lines is driven by an open collector buffer amplifier on the interface. This is used to drive a 4N35 optically coupled isolator in the test circuit. The isolator is used to protect the GPIO interface from electrical spikes that might be transmitted from the test circuit. Figure 3.4 shows this configuration. When a GPIO output line turns an opto isolator on, voltage is supplied to further trigger events in the testing process.

The workstation has a monochrome monitor which displays operating instructions and test results to the operator. An alpha-numeric keyboard is supplied for the user to enter test item information and to make test process selections that are displayed on the screen of the monitor.

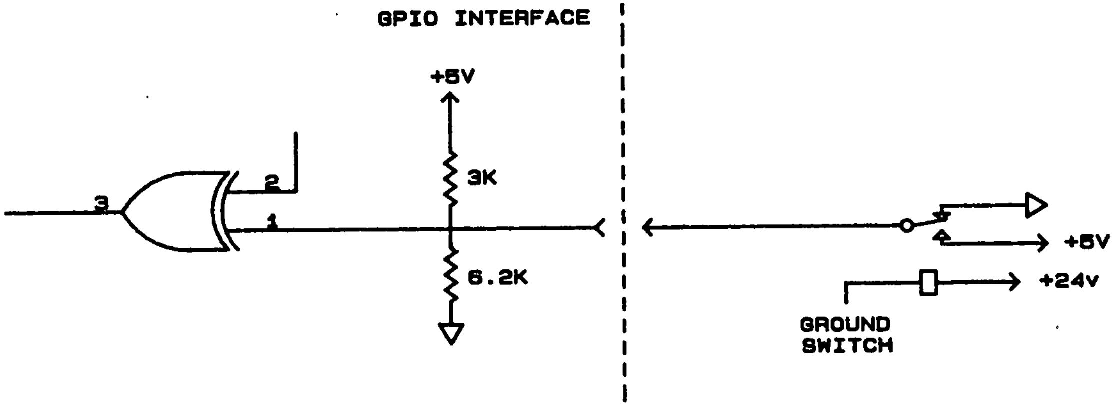


FIGURE 3.3: GPIO INPUT INTERFACE

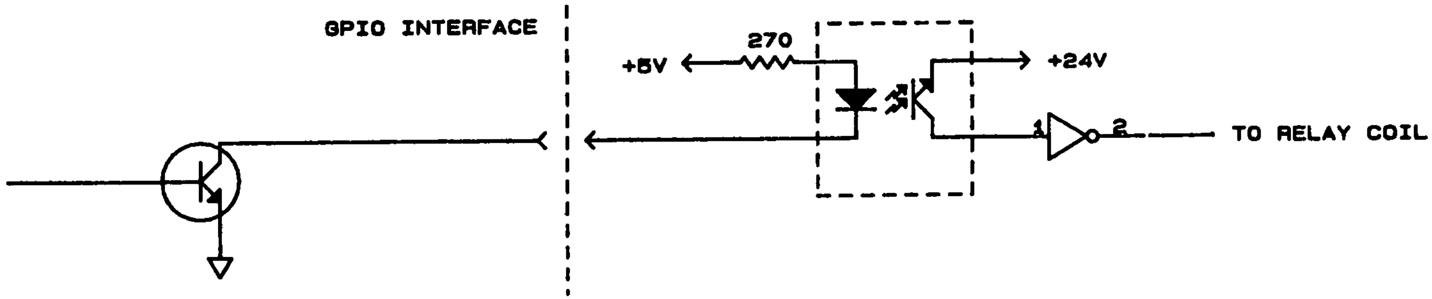


FIGURE 3.4: GPIO OUTPUT INTERFACE

Digital Micro-Ohmmeter

The first piece of commercial equipment that interfaces to the computer is a Model 580 Keithley Digital Micro-Ohmmeter. It performs the Continuity tests on the transducer by making accurate four wire resistance measurements.

This type of resistance measurement consists of supplying a known current through a resistor, or as in this case, a coil of a transformer. A source leg and a return leg for this current source comprises two of the leads in a four wire system. As the current is being supplied through the element, the voltage is measured across the coil. To make the voltage measurement, two more leads are used to attach to the device being tested. The two points at which this is accomplished is in the same proximity as the connections of the two current leads. Figure 3.5 illustrates this circuit. As a consequence no voltage induced by the current through the current leads is added to the voltage across the coil. Given the measured voltage and the known current the resistance is calculated via Ohm's Law. The current sourcing, voltage measuring and resistance calculating are all done internal to the Keithley Micro-Ohmmeter. A four wire cable connects to four terminals on the front panel of the meter and is fed back through the chassis to interface with custom designed switching circuitry.

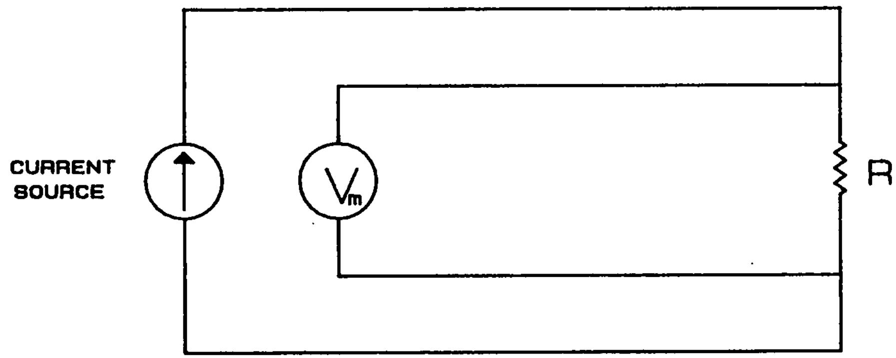


FIGURE 3.5: FOUR WIRE RESISTANCE MEASUREMENT

The computer sends instructions over the HPIB interface bus to the meter to set it up to take resistance measurements. When a test is performed, the necessary switching is done to make the required connections between the micro-ohmmeter and the device under test. The result of a test of this nature is a resistance reading which is placed in a temporary memory at the address of the meter. The software controls the timing of the switching and addresses the micro-ohmmeter at the appropriate time to send this data back over the bus to the computer.

Digital Multimeter

A Hewlett Packard Model 3478A Digital Multimeter is used to verify the voltage output of the high voltage power supplies while performing the Insulation Resistance tests. A cable from the terminals on the rear panel of the meter connects the meter across a 1000:1 voltage divider network. This divider network is switched between each of the outputs of the two high voltage power supplies. The computer sends instructions over the HPIB bus to the meter to take the voltage readings. After these values are obtained, they are transmitted to the computer over the same bus.

The HP3478A has the capability of making a four wire resistance measurement. However, the stability and accuracy of the device is insufficient in the resistance range where the Continuity tests are performed. The

initial plan to switch this meter between the two functions, resistance and voltage, for the two types of tests was discarded.

Storage Devices

A Hewlett Packard HP9122 dual disk drive is the source for program and data storage. A floppy disk with the operation or calibration program is inserted into one of the drives, specifically drive 0. There is an operation and a calibration procedure that instructs the operator to do this before turning on the test system. Even though the program is loaded into memory, the disk remains in that drive during operation. In the second drive, drive 1, a blank disk is used to store the test results.

In addition to storing the data on a disk, a print out is generated for each unit tested. A Hewlett Packard HP82906A printer is used for this purpose.

Control Chassis

The control chassis consists of two high voltage power supplies, two comparator circuits, and hardware designed to interface these to the Hewlett Packard computer. A special circuit designed for safety purposes is also incorporated into this chassis. This extra safety feature is required because of the explosives used in the transducer.

Power Supplies

There are two Venus high voltage power supplies to accommodate two different voltage level requirements. The first of the nine Insulation Resistance tests requires A +/-C volts be applied at the terminals to be tested. The remaining eight I.R. tests need B +/-D volts. A and B are defined as two different high voltage levels and C and D are respectively the associated tolerances. It follows that one of the high voltage power supplies is set at A volts while the other is set at B volts. This is verified during calibration of the tester. The tolerances are given to allow for drift that may occur between calibrations. The voltages are regulated by means of a screwdriver adjustable potentiometer which is remotely placed behind a calibration door on the front bottom of the control chassis. Figure 3.6 shows the front panel layout with the position of these terminals.

A 24 volt D.C. source is used to power the Venus Power Supplies. A single-pole-double-throw relay determines which high voltage power supply is to be turned on. In the de-energized state of this relay the B volt power supply is selected. If the computer, via its GPIO interface, energizes this relay then the A volt power supply is chosen. Corresponding to this relay state, a double-pole-double-throw high voltage relay connected to the output of each power supply determines which voltage will be applied to the device being tested. After all of

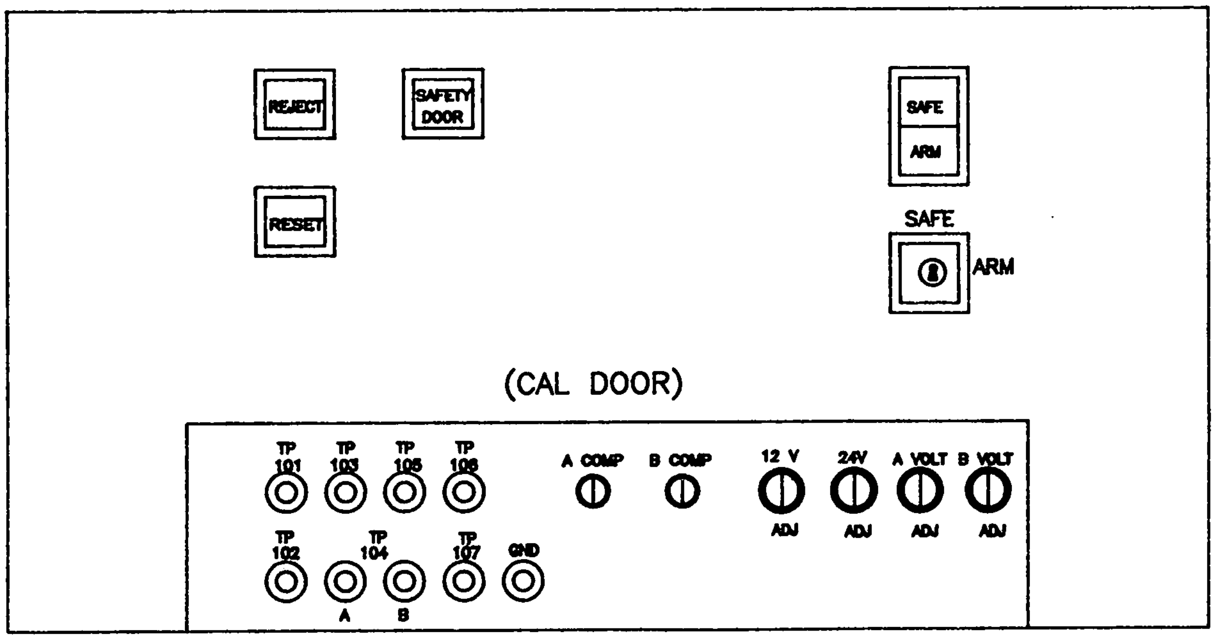


FIGURE 3.6: CONTROL CHASSIS FRONT PANEL

the necessary relay contacts between the output of the selected high voltage power supply and the unit under test have been activated, another relay is used to supply the 24 volt input to the selected high voltage power supply. The voltage level at the unit will change to its required value as a ramp function. The slope of the ramp is based on the internal rise time of the Venus power supply. If the high voltage was turned on before making the desired contacts through the relays the result would be the switching of high voltage. This step type behavior must be avoided to prevent possible damage to the hardware. A sudden change in current would also be detected by the comparator circuit. As will be described later, this would result in the power supply being shut down and the unit being recognized as a reject.

Comparator Circuit

In order for the Automated Non-destructive Test System to perform the Insulation Resistance tests, a method of detecting a condition that does not provide isolation is necessary. Two comparator circuits are used to accomplish this task. The circuits are identical except for being biased with different parameters for the two high voltage requirements of A volts and B volts. The comparators are best described in two stages, an input stage and an output stage.

Depending on which high voltage power supply is being used, a relay switches the corresponding comparator circuit into the test circuit. The input of this circuit is placed in series of the return leg of the selected high voltage test circuit. The input stage is shown in Figure 3.7. This configuration allows the leakage current to flow through a resistor connected in series with a pair of parallel connected back-to-back diodes. The leakage current is the small amount of current that trickles out of the unit under test when the high voltage is applied across the high resistance of the isolated points of the transducer. Ideally, these isolated points should be an open circuit with no current flow. The actual value specified for the Insulation Resistance tests is a resistance greater than ten megohms. Knowing the voltage applied across the device under test and the minimum resistance for a unit to be considered acceptable, the maximum leakage current is defined. For the A volts that current is aa.a microamps and for the B volt supply the maximum current is bb.b microamps. The variables aa.a and bb.b are used to represent these currents. The voltage produced by the maximum allowable leakage current flowing through a resistance in the input stage sets the reference voltage for the comparator. This resistance is actually a potentiometer with its adjust terminal behind the calibration door (Figure 3.6). It is set so that when the leakage current causes the voltage to exceed the set point a pulse is sent to the second stage of the comparator.

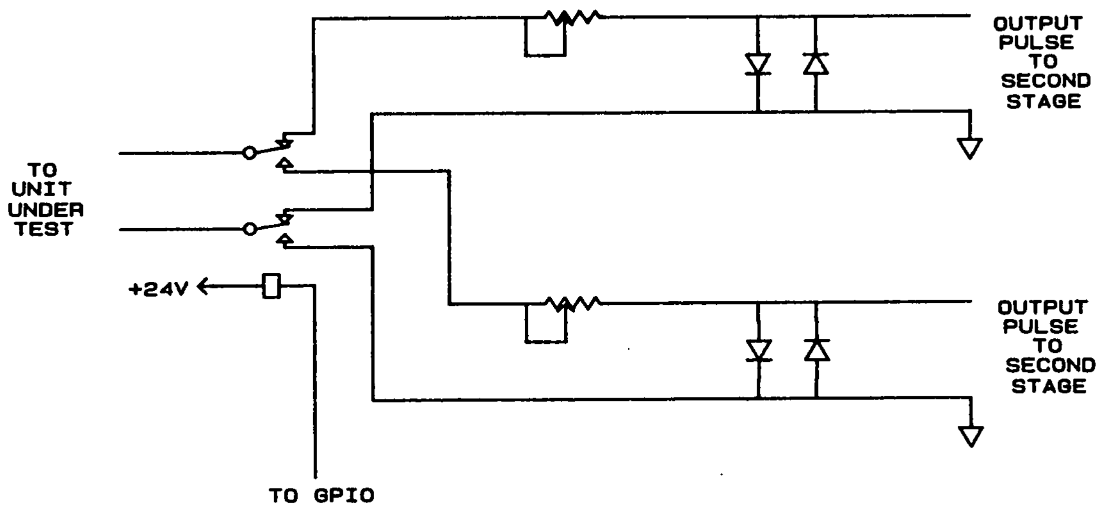


FIGURE 3.7: COMPARATOR INPUT

This will occur when the isolation resistance is too low, thus indicating a defective unit.

The input of this circuit is very sensitive to sudden changes in voltage. This is why the high voltage is ramped up to the desired value instead of being applied as a step function.

A silicon controlled rectifier(SCR) is the main element in the second stage of the comparator shown in Figure 3.8. When the gate of the SCR is triggered by the pulse generated by the first stage, the device starts to conduct, thus providing a path to ground. This ground completes a circuit needed to energize a relay and power a lamp which indicates a reject. When this relay is turned on, the path that supplies the power to the coil of the high voltage relay is broken. This eliminates the high voltage to the unit being tested. For safety reasons high voltage should be removed from a unit that has displayed characteristics of low resistance during an Insulation Resistance test. Another set of contacts on this relay is used to flag an input line on the GPIO to communicate to the computer what has happened. The software then handles this information.

Before the testing process can continue, the comparator circuit needs to be reset. This is accomplished by turning off the SCR. A momentary push button switch is placed in parallel with the anode and cathode of the silicon controlled rectifier. As Figure 3.6 shows, this push button is accessible from the front of the control

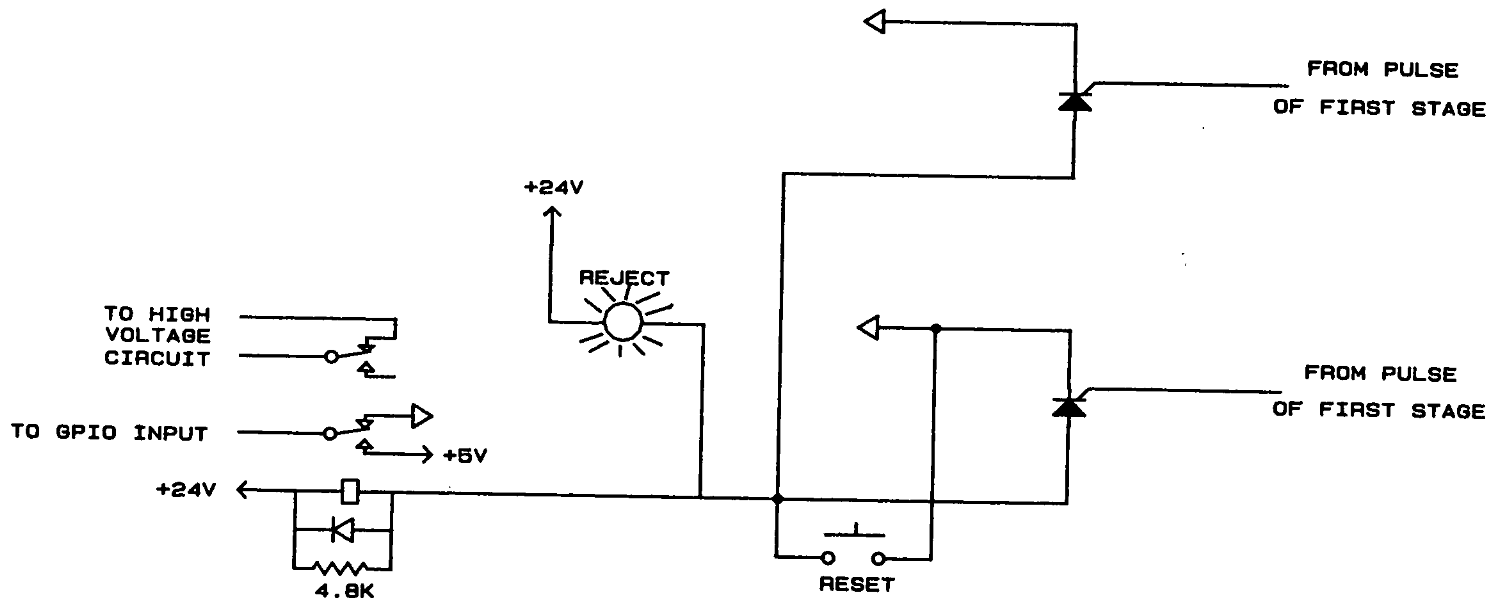


FIGURE 3.8: COMPARATOR OUTPUT

chassis. When this switch is closed a path is created which will shunt the current to ground. The moment the switch is closed, the SCR turns off. Since the high voltage has been removed there is no possible stimulus coming from the first stage to trigger the gate of the SCR at this time. After the switch unlatches there is no conducting path between the anode and cathode. Thus resetting of the comparator circuit is complete.

Control Hardware

The General-Purpose Input/Output Interface controls the testing processes by interfacing to various relays. Relays were chosen over other switching devices because of their low contact resistance.

One side of the coil of each relay is supplied with 24 volts. The relay is energized when the other side of the coil is pulled to ground. This is done via an output line on the GPIO Interface. As mentioned earlier, output lines on the GPIO are used to drive 4N35 Opto Isolators. The output of an isolator provides a path to ground when the input is driven by a GPIO line. Each relay is assigned to a GPIO output line that controls it. The software addresses the required GPIO lines for the desired process.

One set of relays is designed to make the three Continuity tests. Since these three tests have no points in common, a four-pole-double-throw relay is used to connect the Keithley Ohmmeter to a selected test. Three of these relays provide an automated method for switching the

four leads of the ohmmeter to the test points. The normally closed contacts of these relays are tied to ground. This insures that the points under test are isolated from all others. Figure 3.9 illustrates this configuration. The resistance reading obtained over the HPIB parallel bus is correlated with the selected relay and is stored by the computer for the associated test.

To perform an Insulation Resistance test, one of the two high voltage power supplies is selected as described earlier. Two test points are then selected to have the high voltage applied across. This is accomplished by closing two of the relays labeled as K202 through K209 in Figure 3.10. One of each selected relay switches its corresponding test point to the ground side of the selected high voltage power supply. The other relay switches its test point to the high side of the same selected high voltage power supply. Since two of the test points are required for more than one test, an additional relay, K201 in Figure 3.10, is used to switch the two corresponding relays, K202 and K203, between the high and the ground side of the power supply. The normally closed contact of the relays that connect to the unit are tied to ground. In addition to insulating the test points, it provides a safety feature by isolating the high voltage leads from the connectors on the exterior of the tester.

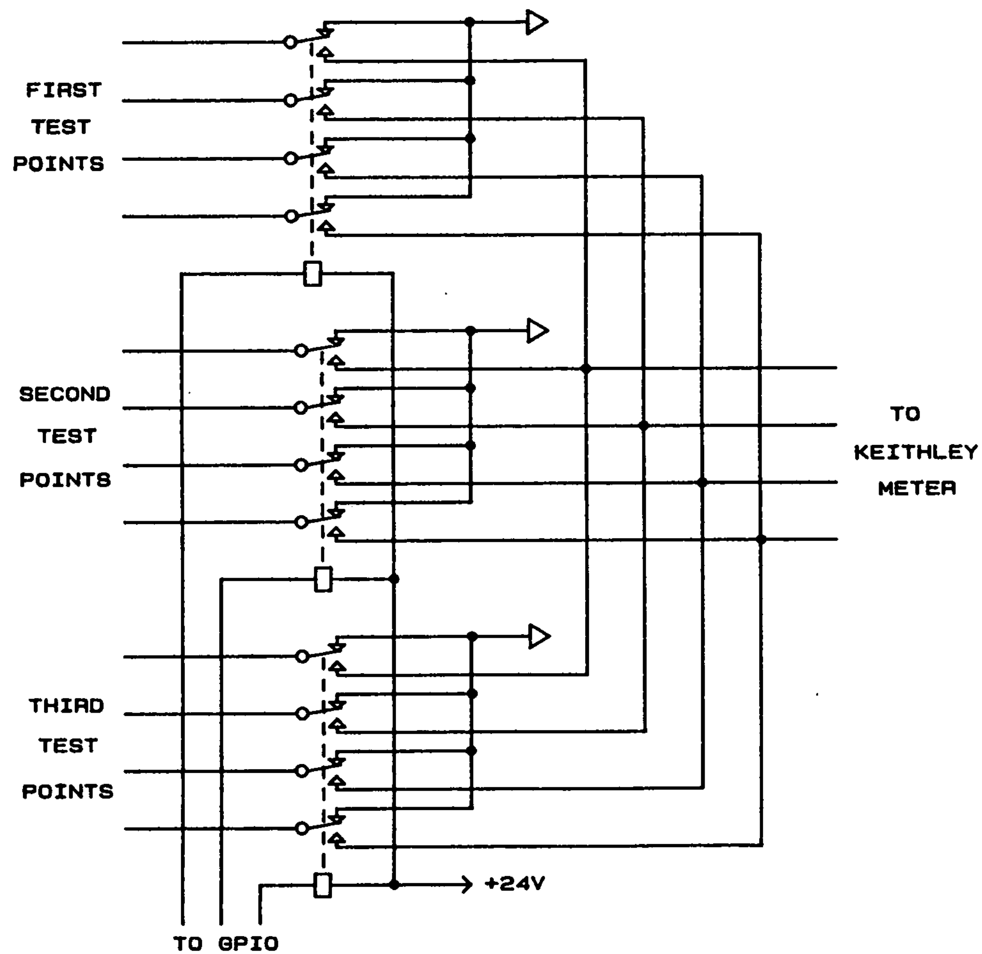


FIGURE 3.9: CONTINUITY TESTS RELAY CONFIGURATION

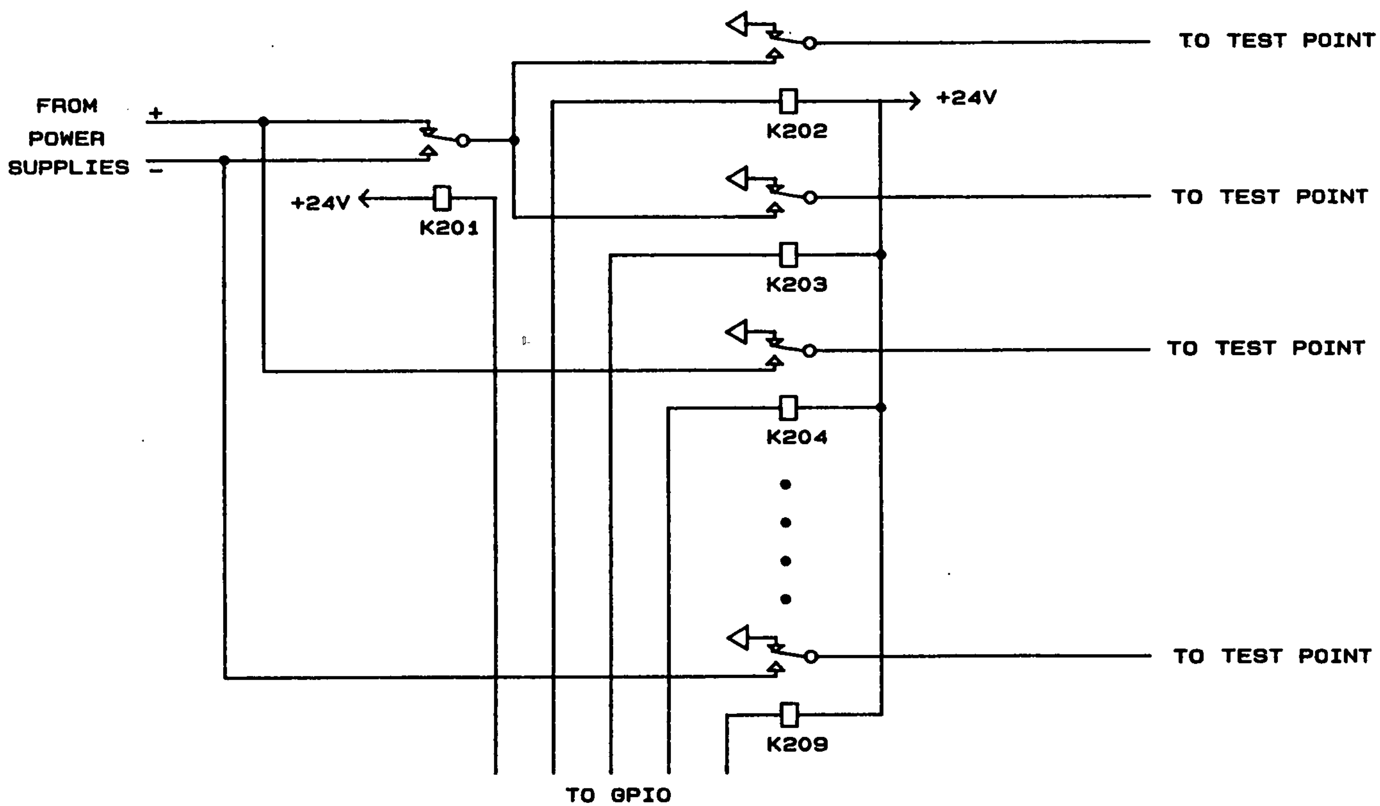


FIGURE 3.10: INSULATION RESISTANCE TESTS RELAY CONFIGURTION

Safing Circuitry

Because of the explosives in the transducer and high voltages at the test points, safety features are imperative. A unit is placed in a barricade approved for such explosives prior to testing. Figure 3.11 shows this safety barricade. The port hole on top not only allows for exhaust should the device explode, but also provides a means to get the test cables to the unit being tested. There is an interlock in the front door of this chamber that is closed only when the door is secure.

This interlock is a contact switch that interfaces to ANTS via a two lead cable. Twenty-four volts used to energize a relay in the safing circuitry is fed out of the tester by one of the leads, through the interlock, and back into the tester by the other lead. If the interlock is not closed, the relay can not be activated. This relay labeled as K101 is shown in Figure 3.12. The path to ground for the relay coil is provided when a safe/arm key switch is in the safe position. The relay is wired so that it is self energizing. Once it is activated, the ground for the coil is provided through one of its own closed contacts. As long as the 24 volts is being furnished the relay will stay on. This is done since switching the key to the arm position removes the initial path to ground. This relay also supplies power to a lamp through one of its normally opened contacts to indicate that the safety chamber door is not secure. Another set of contacts is used to send a high signal to a GPIO input line when the relay is unenergized

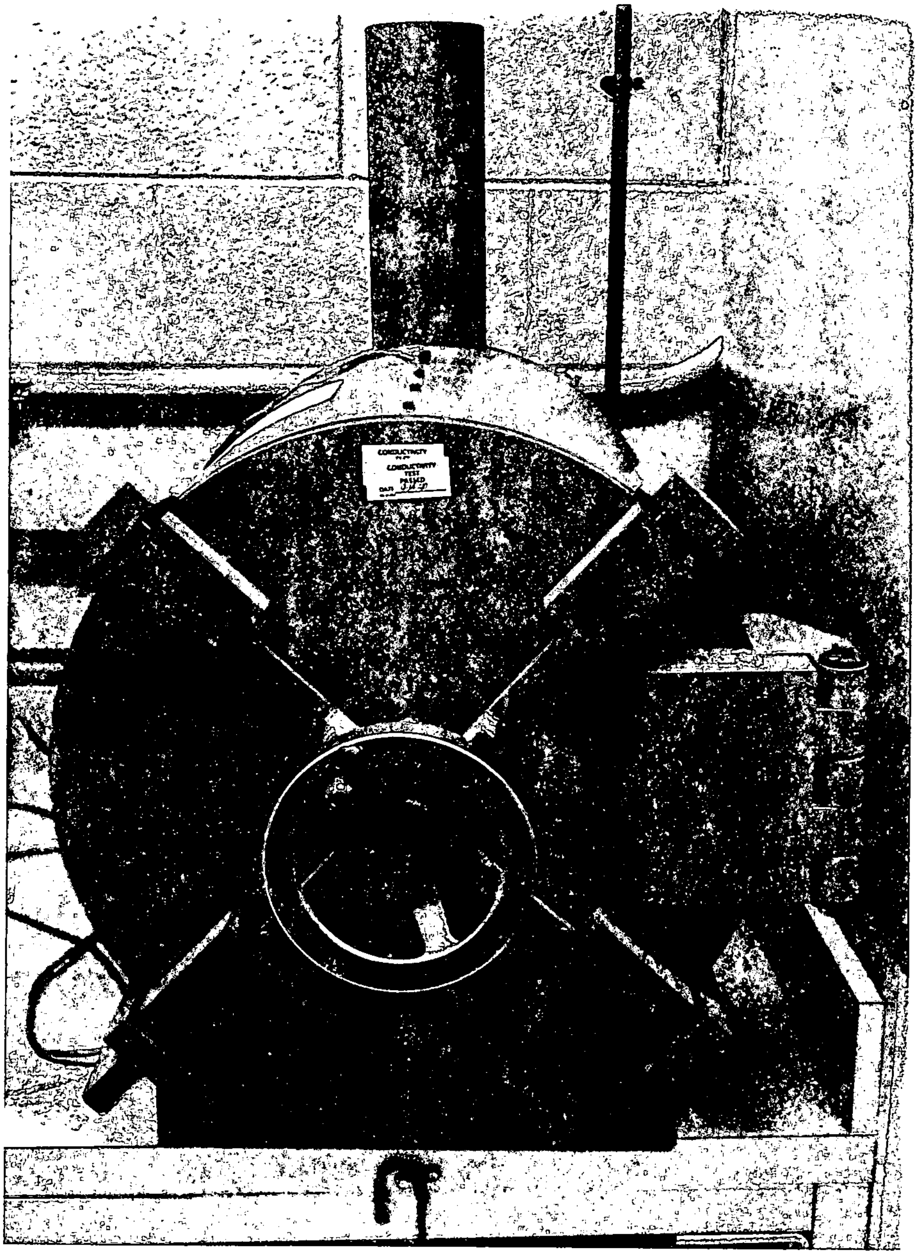


FIGURE 3.11: SAFETY BARRICADE

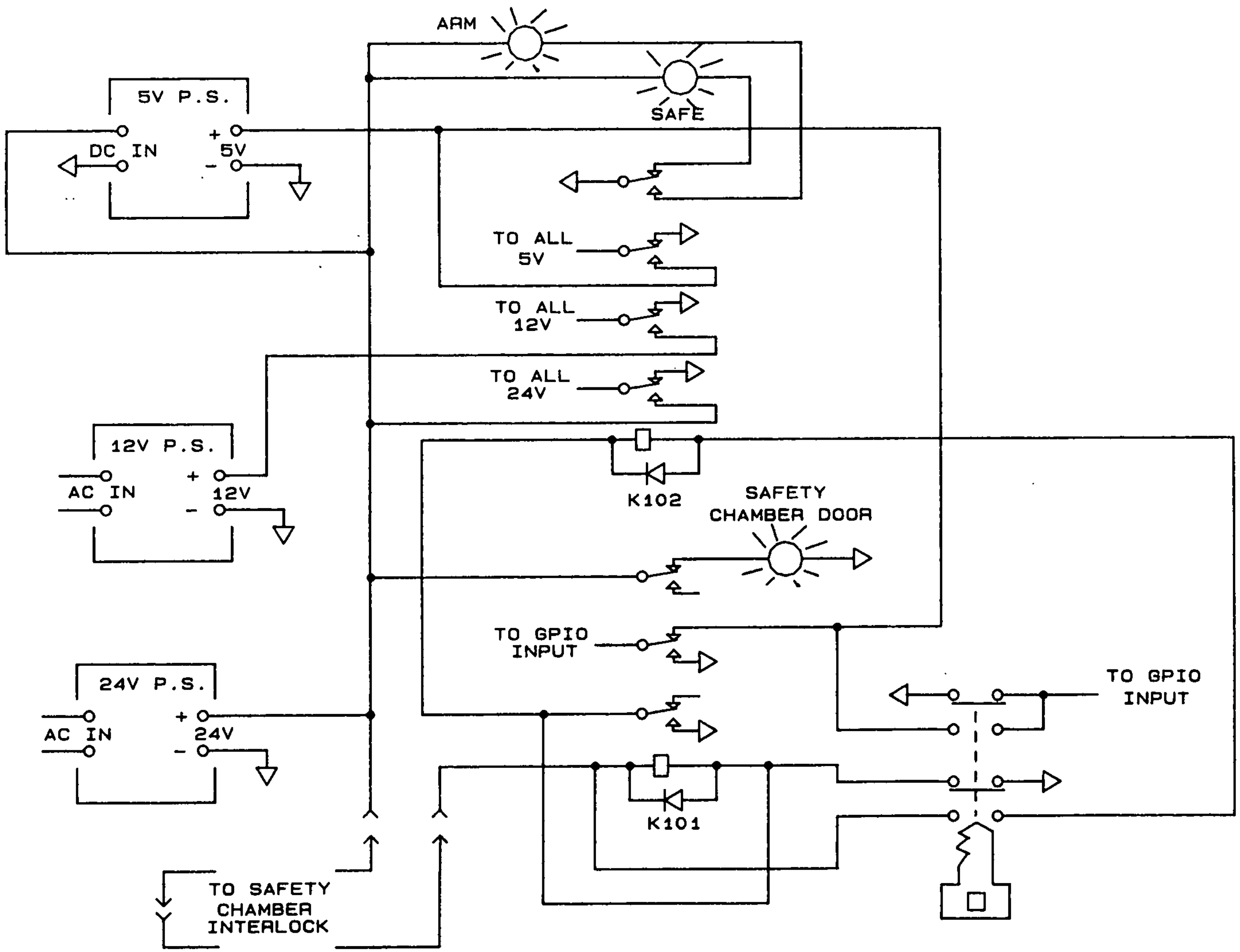


FIGURE 3.12: SAFETY CIRCUITRY

to communicate to the computer the position of the barricade door. When the relay is on, this contact is switched to ground, thus informing the controller that the barricade is secure. The relay will only become energized if the barricade door is closed and the key is in the safe position.

The purpose of this key switch is to provide a manual way to supply and remove power from the control chassis. This is done in compliance with safety procedures required at Mound. Opening the barricade door would not be a desirable way to shut down power if something critical happened to the transducer being tested. The closed contact that provides the ground for the self energizing relay is also the ground needed by another relay (K102) in the safing circuitry. To complete the circuit to energize this relay, the 24 volts is supplied through the key switch in the arm position. It is through this relay that all the hardware is powered. The five volt, 12 volt and 24 volt power supplies are fed through three closed contacts of this relay. The relays and integrated circuits get their voltages from these contacts. A fourth contact on the relay toggles a two position lamp between a green safe and a red armed indicator. This task is done by a relay instead of directly supplying all the power through contacts on the key switch because the current rating on the key switch contacts is not adequate. The coil of the one relay is the only element the key switch has to drive through one of its contacts. Another key switch contact is

used to send either five volts or ground to a GPIO input line so the computer knows the position of the key.

To summarize the function of the safing circuitry, the desired performance will be explained. The key must be in the safe position and the barricade door closed to energize the first relay. After the key is repositioned to arm the tester, the relay stays on because it attains the needed ground through one of its own contacts. If the chamber door is opened, the relay loses its power which removes the ground path for the coil. Closing the barricade door again will supply the voltage but since the key is in the arm mode no ground will be available to the coil. Therefore, the key must be returned to the safe position before the Automated Non-destructive Test System can be armed again.

Through a combination of the key switch being in the armed position and the first relay being activated, a second relay is energized. It is through this second relay that all power needed to perform the automated tests is furnished.

Fixtures

To perform all the required tests on the transducer, a method of connecting to the various tests points had to be devised. Access to these points is by specialized connectors. Since these types of connectors are precarious, extreme care in fixturing is required. Two fixtures, one for the Continuity tests and one for the

Insulation Resistance tests, were designed to be used with the test system. Both fixtures are side by side on the same base as shown in Figure 3.13.

The Continuity fixture is on the left side of the base. The transducer is placed on two poles and mating connectors are attached to the device. Each mating connector has two wires leading to it in order to maintain the four-wire resistance measuring technique. From the mating connectors, the cable carrying all these lines goes back into the test system and is integrated with the corresponding test selection relays for the Continuity tests.

The Insulation Resistance fixture, on the right in Figure 3.13, has a similar configuration except that it only needs one wire leading to each mating connector and uses higher voltage conductors. For ease in operation, some of these mating connectors are in a lid type form (not shown). This lid is placed on top of the transducer and a spring loaded arm clamps down on the lid and thus makes several of the required connections with one act. A cable from this fixture is correlated in the tester with the appropriate relays to select all of the Insulation Resistance tests.

When each transducer is tested, it is placed on one of the fixtures and then all of the appropriate connectors are attached to it. All of the tests corresponding to that fixture are performed. The unit is then moved to the other

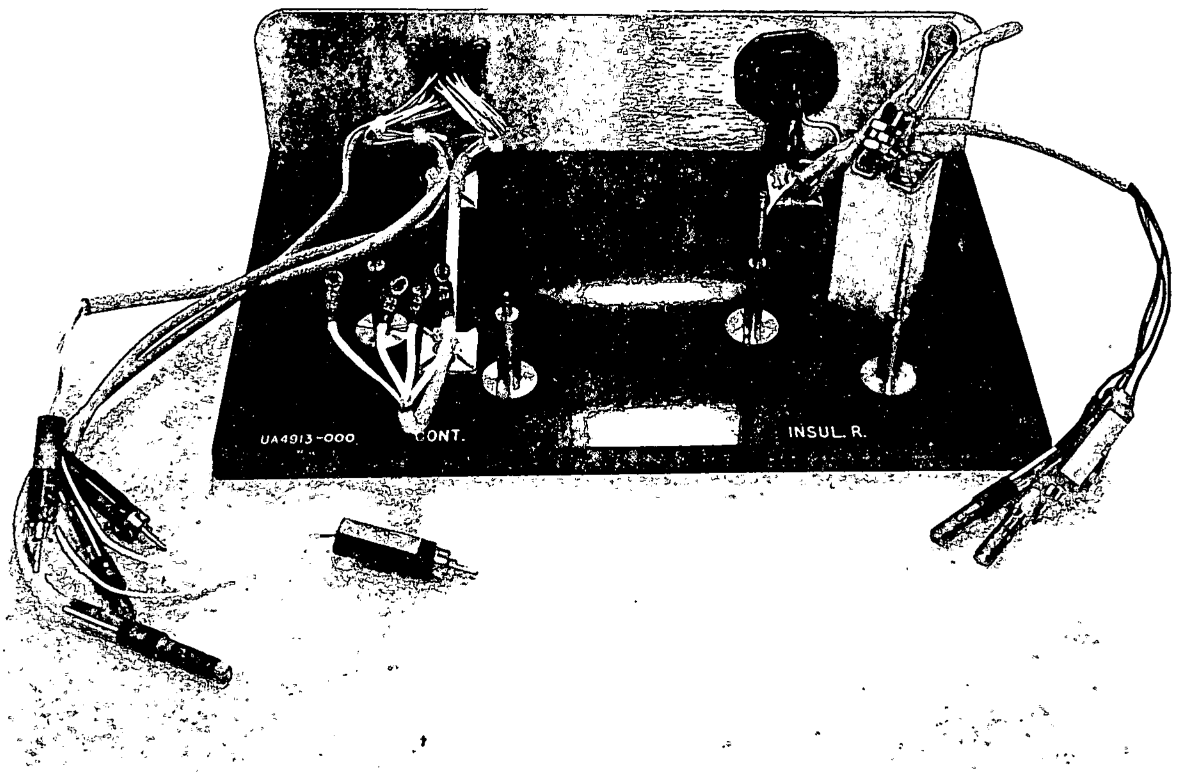


FIGURE 3.13: FIXTURES

fixture, all connections are made and all of the tests corresponding to that fixture are completed. The order of testing is determined by the operator via the software.

Simulators

A simulator is a device that is designed with some of the same characteristics as the transducer. The electrical characteristics of interest will be the same as those tested by the Automated Non-destructive Test System. The simulators contain no explosives and resemble the actual device enough to use the same fixtures. By using them at the beginning and the end of each day of testing, confidence in the test system is established. This procedure verifies that all the test results obtained that day are accurate and valid. Since the two types of tests performed by the Automated Non-destructive Test System verify different electrical conditions, two simulators are used.

The simulator for the Continuity tests is a hollow block of clear lexan with holes and connectors intricately placed to accommodate the connections for the tests. Inside of this box are precision resistors with values within the limits of the actual transducer. It is connected to the Continuity fixture the same way the device is and has the same measurements made. If the simulator fails, it is assumed that there is a fault in the test system. This problem must be corrected before transducers can be tested.

The Insulation Resistance simulator is more complex because it must be able to verify an accept and a reject condition for each test. Since an Insulation Resistance test is a method to verify the existence of a very high resistance, an open circuit would incorrectly be considered acceptable. If somewhere in the test circuit a connection is not made or a wire is broken, the result would be the same as an accepted unit. The simulator proves that none of these faults exist by defining a test to be performed on a resistance known to be below the acceptable value.

On the other hand, the simulator must also be capable of verifying that nothing is shorted together in the tester or fixture. A resistance greater than the minimum requirement is used to eliminate the possibility of false rejects.

Similar to the Continuity simulator, the Insulation Resistance simulator consists of a lexan box with the appropriate connectors to mate with the fixture. On the inside of this box, wires attached to the connectors run out of the lexan and into a box with two rotary switches. One switch isolates two pins on the connector to single out a particular test. The other switches between two resistors in the box, one just above the limit of ten megohms to simulate an accept condition and the other just below the limit to mimic a rejected unit. In the latter case the current flowing through the circuit will be too high and will cause the comparator circuit to trip. Because an acceptable Insulation Resistance test is

verifying an open circuit, reject conditions need to be simulated to confirm that the tester is functioning properly. When this simulator is tested, the software performs each test twice, once looking for an accept condition and once looking for a reject status. The software instructs the operator where to position the switches and then performs the corresponding tests. If these conditions are not met, the software instructs the operator to double check the switch positions. If the switch positions are verified to be correct it is assumed that there is something wrong with the tester. The type of error is documented to aid in finding and correcting the problem.

Both simulators are verified during the calibration of the test system. Tests on the simulators are then made prior to testing any transducer. This establishes the validity and accuracy of the test results obtained by the Automated Non-destructive Test System.

CHAPTER IV

SOFTWARE

The Automated Non-destructive Test System is controlled by a BASIC operating system. The test program consists of many routines written to perform specific tasks needed to test the transducer, store the data, and make the program user friendly.

Upon power up of the test system, the BASIC system is loaded into memory from a floppy disk which is in drive 0 of the HP9122 disk drive. This includes all the binaries and drivers needed for interfacing to the other equipment and the operator via the monitor and keyboard. After this is accomplished, the system loads and runs an auto executable program called "AUTOST". This program is simply used to call the main testing program into memory and then to run it. The flow chart for the main testing program is found in Appendix A.

Program Setup

The first thing the main program does is graphically display an introduction screen which consists of the title of the test system and further instructions to execute the program. Upon execution, the program branches to the routine which defines the memory location for the

constants. In BASIC they are given names which have significance related to their function. The resistance limits for the three Continuity tests are set up in two one by three dimensional arrays and are named "R_up" and "R_low". The tolerances for the high voltages are added and subtracted from the nominal value to define the upper and lower limits for the voltage requirements for the Insulation Resistance tests. These limits are stored in BASIC arrays named "Up_volt_lim" and "Low_volt_lim". Status bits are given names which will be used in condition testing routines.

The main program branches to a routine which initializes all the variables which are used to store the test results. This routine is executed before every test to clear the memory locations of previous test data. By doing this, the validity of the test results is assured.

Defining the addresses of all the equipment is the next task to be accomplished by the software. Each piece of equipment is configured to perform its specific function.

There are four monitoring routines which check the status of the GPIO interface bits. The first one examines the condition of the line that is tied to the safe/arm key switch in the hardware. A high state, a binary one, indicates to the software that the key is in the safe position and the program will continue. If a low or a binary zero is detected on this bit, a message to return the key to the safe position is flashed on the screen. In

this case the program will not continue until this is done. After this is accomplished, the desired high state will be detected and the routine will be exited. When performing a test, a routine that does just the opposite is executed. The key switch must be in the armed position to supply power to the test circuits.

Two more routines which operate in the same way are used to detect the state of the safety chamber door and the reset switch of the comparator circuits. The door routine is executed every time the door must be closed. A routine to ensure that the comparator circuits are in the reset state is used to verify that the system is ready to perform the Insulation Resistance tests.

These four program segments are called at various times throughout the execution of the test program to perform their designated tasks.

A display of the main menu is the next operator interfacing function performed by the software. It provides the option to test a unit, run the simulators or access the stored data for editing and printing purposes.

Testing

To verify that the tester is functioning properly, the first choice must be to run the simulators. The primary difference between the testing of the simulators and of an actual transducer is the storing of the test data. Any variation in the testing procedure will be detailed in the

explanation of each type of test.

Entering Data

After the selection to test a unit or run the simulators has been made, various information pertaining to the lot of transducers is entered by the operator. This information is used to identify when the lot was built, when it was tested, who tested it, at what stage of its development it was tested, and the reason for performing these tests on it. The different development stages of a transducer consists of being exposed to a variety of environmental conditions such as extreme temperatures and vibrational shock. A unit is tested on the Automated Non-destructive Test System after various environmental tests to verify that it was not damaged by them. Reasons for testing the transducer are for obtaining development or process data, accepting a product or testing stockpile samples. All these questions are answered by completing prompts as they appear on the screen of the monitor.

The above information is entered one time for the entire lot. However, a serial number for each unit is entered for every transducer. When testing the simulators, the serial number is labeled as such by the software when that selection is made from the main menu.

After entering this information, the operator is then instructed to choose between performing Continuity tests, performing Insulation Resistance tests, storing the test results or ending the testing session.

Continuity Tests

The routine to perform the Continuity tests is entered by making the appropriate selection from the above mentioned menu. The instruction to install the transducer in the Continuity fixture is given. If testing the simulator, the instructions refer to the simulator and all of the testing techniques are the same as for an actual device. The question is then asked whether all three or just one or two of the Continuity tests are to be made. If all are to be tested the routine makes the measurements in an automated fashion. One or more tests can be singled out for individual testing if deemed necessary for investigative purposes. This allows for the test in question to be re-tested without having to run through all of the tests. All of the tests must be made before the test will be considered complete and the data stored.

If this question is answered with the "Y" prompt then all of the resistance measurements are made. After executing the routine to instruct the operator to "ARM" the tester by switching the safe/arm key to the arm position, the program increments through all three of the tests. Each test is performed when a subroutine corresponding to that Continuity test number is called. These subroutines close the necessary relays to connect the Keithley Micro-Ohmmeter to the points under test. The relays are closed when the appropriate GPIO interface output bits are set by a software instruction. The meter is then instructed to take a four-wire resistance reading. After

waiting one second, this value is entered into the corresponding variable in the computer memory. The GPIO interface lines are cleared, thus de-energizing the relays. An instruction to "SAFE" the tester by returning the safe/arm key to the safe position is given. The routine to verify that this has been accomplished is executed. The newly obtained resistance value is then rounded to meet the precision required by the specifications of the transducer.

If the option to not test all of them is made, the operator is then instructed to enter in the test numbers to be tested. After the tester is "ARMED", the testing routine only branches to the subroutines necessary to make the selected tests.

After testing is complete the results are printed out on the monitor screen for the operator to review. If any of the results indicate that one of the connectors to the unit was not secure during the test, the tests in question can be re-tested after checking the connections. Connecting to the transducer for the tests performed by the Automated Non-destructive Test System is a very tedious task, even for a trained operator. That is why the capability to test more than once is provided.

If all of the continuity tests look valid, the software branches back to the menu that gives the type of test choices.

Insulation Resistance Tests

The Insulation Resistance testing routine is selected from this menu. The first thing this routine does is to instruct the operator to connect the transducer or the simulator to the Insulation Resistance fixture. When testing a transducer, the process is similar to that of the Continuity testing routine. The operator is prompted to select the number of tests to be performed and to "ARM" the tester. If the choice to automatically perform all nine I.R. tests is made, the routine will increment a counter corresponding to each Insulation Resistance test. Every time the counter is incremented, a subroutine will be called to execute that corresponding test. These subroutines close the appropriate relays needed to make the tests. After a set of relays is closed, the software branches to the routine that checks the state of the comparator circuit by polling the appropriate GPIO Interface input bit. This is accomplished by software instruction statements. A high state on this bit indicates that the comparator circuit was not triggered. Under this condition the program continues and the test result is considered acceptable. However, if this bit is low, the comparator circuit has been triggered thus interrupting the test circuit. After detecting this condition, the software displays instructions to "SAFE" the tester, as described earlier, and then to reset the I.R. circuit. As described in the Hardware chapter, resetting this circuit involves

depressing a push button on the front of the control chassis. The program will continuously loop through the reset I.R. routine until the circuit is reset. After the circuit is reset, the test is labeled as a reject and the next Insulation Resistance test is started.

If the operator chooses to perform only a few of the tests, the prompt to enter in the desired test numbers must be answered. Only the tests corresponding to the ones selected will be executed.

After the Insulation Resistance tests have been completed the results are displayed on the screen of the monitor. This gives the operator the opportunity to verify that the unit was connected properly and that the results are valid. There are no values associated with these tests, just an accept or reject status based on whether the comparator circuit tripped. The data is not stored until all nine of these tests are completed and verified.

If the simulator is being tested, the serial number is labeled as such and a routine for verifying the insulation resistance simulator is called. This routine confirms that an accept and reject condition for each of the Insulation Resistance tests is satisfied. This is done in conjunction with the switching box of this simulator. The software instructs the operator to position the two switches on the box for a specified condition on a particular test. One switch selects the test and the other switches between the accept and reject condition.

The software keeps track of which test is being performed. For each test on the simulator, the actual testing routine is branched to twice from the simulator routine. The first time an acceptable condition is required to proceed. The second time a reject must be detected and the comparator circuit must be reset for the software to continue to the next test. Each test of the Insulation Resistance simulator must meet the accept and reject requirement to be considered a "passed" test. If both of these conditions are not satisfied the test is labeled as a "failed" test. The operator is given the opportunity to verify that the switches are in the proper positions and to run the tests again. A failed simulator test indicates that the test system requires maintenance.

Storing Data

After all of each type of test is completed and the operator reviews the results, the program returns to the menu to choose the type of test to be performed, to store the data or to end the testing session. The option to store the data must be made. When this is done, a routine is called that displays all of the test results of both the Continuity tests and the Insulation Resistance tests. It also verifies that all of the tests have been completed. If not all of the tests have been made, the program will notify the operator of this and will not store the data. Upon acknowledgment from the operator, the software will return to the menu to choose the type of test to be

performed.

If all of the tests are completed for the given transducer the test results are printed out on the printer. Figure 4.1 is such a print out. The print out generated when the simulator is tested varies slightly because some of the header information does not apply. The print out from the simulator is shown in Figure 4.2.

The data from a transducer is formatted to be stored on a data disk. The test results are associated with identifying codes and then placed in a string variable to be stored in a random access ASCII formatted file. The characteristics of this data string are defined by Sandia National Laboratory in Albuquerque (SNLA) for all weapon components. This is done for transmitting and storing purposes. The standard ASCII format can be read by other operating systems. Any test results obtained from the simulators does not get stored on a disk.

Editing Data

From the original menu the option is given to edit the data. This selection allows the information that was entered by the operator to be changed. None of the actual test results can be accessed by the program. Due to human error, mistakes such as transposing serial numbers or entering in the wrong date can occur.

Using the print out as a guide, the record number that the mistake appears in is selected. The data string stored in the file associated with this record number is brought

AUTOMATED NON-DESTRUCTIVE TEST SYSTEM
CONTINUITY/INSULATION RESISTANCE TESTER

PROCUCT SPECIFICATION: 123456-A
PART NUMBER: 123456-01
LOT NUMBER: 0001

SOFTWARE: SW123456-A
DATE: 11-12-90
TEST CODE: 00
DATA SOURCE CODE: X

SERIAL NUMBER: 000012

DATE MANUFACTURE CODE: 290

TEST	Limits(Ohms)		VALUE(ohms)	STATUS
CONT TEST NUMBER 1	x.xxxx	y.yyyy	z.zzzz	ACCEPT
CONT TEST NUMBER 2	x.xxxx	y.yyyy	z.zzzz	ACCEPT
CONT TEST NUMBER 3	x.xxxx	y.yyyy	z.zzzz	ACCEPT
I.R. TEST NUMBER 1				ACCEPT
I.R. TEST NUMBER 2				ACCEPT
I.R. TEST NUMBER 3				ACCEPT
I.R. TEST NUMBER 4				ACCEPT
I.R. TEST NUMBER 5				ACCEPT
I.R. TEST NUMBER 6				ACCEPT
I.R. TEST NUMBER 7				ACCEPT
I.R. TEST NUMBER 8				ACCEPT
I.R. TEST NUMBER 9				ACCEPT

FIGURE 4.1: TRANSDUCER PRINTOUT

AUTOMATED NON-DESTRUCTIVE TEST SYSTEM
CONTINUITY/INSULATION RESISTANCE TESTER

PROCUCT SPECIFICATION:
PART NUMBER:
LOT NUMBER:

SOFTWARE: SW123456-A
DATE: 11-12-90
TEST CODE:
DATA SOURCE CODE:

SERIAL NUMBER: SIMULATOR

DATE MANUFACTURE CODE:

TEST	Limits(Ohms)		VALUE(ohms)	STATUS
CONT TEST NUMBER 1	x.xxxx	y.yyyy	t.tttt	ACCEPT
CONT TEST NUMBER 2	x.xxxx	y.yyyy	t.tttt	ACCEPT
CONT TEST NUMBER 3	x.xxxx	y.yyyy	t.tttt	ACCEPT
I.R. TEST NUMBER 1				ACCEPT
I.R. TEST NUMBER 2				ACCEPT
I.R. TEST NUMBER 3				ACCEPT
I.R. TEST NUMBER 4				ACCEPT
I.R. TEST NUMBER 5				ACCEPT
I.R. TEST NUMBER 6				ACCEPT
I.R. TEST NUMBER 7				ACCEPT
I.R. TEST NUMBER 8				ACCEPT
I.R. TEST NUMBER 9				ACCEPT

FIGURE 4.2: SIMULATOR PRINTOUT

into the computer's memory. The string is searched for the identification codes of the operator entered data that can be changed. Variables allocated in memory are filled with this information. The operator selects from a menu which parameter needs to be edited. The data in the temporary variable is displayed on the screen and is changed by being typed over. After the change is verified by the operator, the variable strings are concatenated back into the single data string to be stored on the disk. This string overwrites the original string stored in that file.

After this is accomplished, the program returns to the introduction screen.

CHAPTER V

CALIBRATION

Calibration of the Automated Non-destructive Test System is required to be assured of the validity of the test data. The tester is calibrated periodically by the EG&G Mound metrology laboratory using a procedure and software written for this purpose. The calibration is traceable to the National Institute of Standards and Technology (NIST), formerly the National Bureau of Standards(NBS).

Simulator Verification

As part of the calibration, the electrical characteristics of the simulators are verified. The three precision resistors in the Continuity simulator are measured with a calibrated ohmmeter from the electrical standards laboratory. This measurement employs the four wire resistance measuring technique described earlier. Each set of leads from the standard ohmmeter is attached to pins from mating connectors. These pins are inserted into two desired points in the simulator connector. The reading obtained from the two desired points is compared to the original design value. This is done for all three of the resistance values that constitutes the Continuity

simulator. These values are recorded as part of the history of the test system.

The standard ohmmeter is connected to the Insulation Resistance simulator the same way as the Continuity simulator. For each of the two points in the connector the meter is used to take two measurements. A 12 megohm resistor and a ten megohm resistor in the simulator switching box are checked. The two different resistor values are used to simulate an accept and a reject condition for each of the nine Insulation Resistance tests. These values are verified and recorded.

Equipment Calibration

Each of the digital meters is calibrated using references from the electrical standards laboratory. Two ranges on the Keithley Micro-Ohm Meter are calibrated with standard resistors. Each of the standard resistors is plugged directly into the meter. The meter must read within 0.01 percent of the standard resistors. If one does not, the discrepancy must be noted and adjustments made to the micro-ohm meter.

The Hewlett Packard meter is calibrated for voltage measurements since that is its only parameter used by the Automated Non-destructive Test System. A standard voltage power supply is used to provide an A/1000 volt signal and a B/1000 volt signal to the meter. These represent the voltages seen by the meter across the voltage divider network. The meter must read these values within the

acceptable tolerances. The same procedure for noting discrepancies and making adjustment is followed as for the Keithley Micro-Ohm Meter.

At the bottom of the front panel is the calibration door shown in Figure 3.6. Test points for calibrating the five volt, 12 volt and 24 volt power supplies are accessed by opening the door. After connecting a standard voltmeter across TP105 and ground the 24 volt adjust terminal is regulated until the standard meter reads within ten percent of the nominal 24 volts. The 12 volt and five volt power supplies are calibrated the same way by connecting the standard voltmeter across TP106 and ground, and TP107 and ground and making the needed adjustments to the appropriate terminals.

Software Aided Calibration

The remainder of the calibration is dependent on software control. A program separate from the operating program is written for this task. This program is used by technicians from the electrical metrology laboratory only.

Similar to the operating program, the calibration software is written in the BASIC language. Memory allocation is configured the same way for variables and test parameters. The program consists of many routines to perform various tasks. Most of the routines interface to the technician by giving instructions and relaying what the routine is doing. The flow chart of this program is found in Appendix B.

Comparator Circuits

After the program is loaded into memory and an introduction screen is displayed the calibration process begins. The first task to be accomplished must be to verify that the A volt and the B volt power supplies are performing at the expected levels. The standard voltmeter is connected across TP101 and TP102 in the calibration panel. This connection places the standard meter directly across the output of the high voltage power supplies. A software routine is then called to close the needed relays to turn on the B voltage power supply. The voltage value is verified on the standard meter. A second software routine is used to verify the A voltage power supply. The voltage being supplied by both sources is noted as "as found" values. The reading on the Hewlett Packard digital meter in the Automated Non-destructive Test System must be equal to the readings on the standard meter divided by 1000. This verifies the voltage divider network in the test circuit. This data is used for historical purposes since calibrating the comparator circuits involves taking the high voltage power supplies out of limits.

The comparator circuits are calibrated by setting the leakage current values that will cause the comparator circuit to be triggered. The calibrated Insulation Resistance simulator provides the needed resistance at the device under test terminals to cause an unacceptable leakage current. This is achieved by placing the accept/reject switch on the simulator switching box in the

reject position. The test selection switch is positioned according to which comparator circuit is being calibrated. The first test must be selected when calibrating the circuit that corresponds to the A volt test. Since the remaining tests all use the same circuit in conjunction with the B volt test, the selection is arbitrarily made for the second test. The variables A and B were defined previously as two different high voltage levels. The software gives the instructions and keeps track of which test is being selected. The leakage current is measured by removing a shorting plug, TP104, in the calibration panel, Figure 3.6, and placing a standard ammeter in series in the return leg of the comparator circuit. The software activates the circuit as would be done during an actual test except there is no time limit. After the technician "arms" the tester by placing the safe/arm key in the arm position, the circuit functions until the key is returned to the safe position. While the circuit is functioning, the potentiometer which determines the reference voltage for the input of the comparator is adjusted until the leakage current no longer causes a reject. The terminal for this potentiometer is located behind the calibration door. The reset button on the front panel is used to continuously reset the tripped comparator until the potentiometer adjustment no longer causes a reject.

The high voltage power supply is adjusted until the standard ammeter reads a value just below the acceptable maximum leakage current. For the A volt tests this value

is aa.a microamps and for the B volt test the value is bb.b microamps. As defined earlier, aa.a and bb.b represent the values of the two maximum leakage currents. After the leakage current is established, the potentiometer of the comparator circuit is adjusted until it is just sensitive enough to cause a reject at the defined current. The standard ammeter is removed from the circuit and the shorting plug is placed back in the circuit.

The above procedure is followed for calibrating both of the comparator circuits one at a time.

The two high voltage power supplies must now be calibrated since they were adjusted out of limits earlier. The switching box of the Insulation Resistance simulator is switched to the accept position. A standard voltmeter is connected across TP101 and TP102 in the calibration panel, Figure 3.6, as was done when verifying the "as found" voltage values. The power supplies are calibrated by making the needed adjustments to the appropriate terminals on the calibration panel until the standard voltmeter reads the desired voltage. The voltage must be A +/- C volts for the A volt power supply and B +/-D volts for the B volt power supply, where C and D are the respective tolerances. The voltage read by the Hewlett Packard Digital Multimeter in the test system must be equal to that read by the standard meter divided by 1000 +/-0.5 percent. The one half percent is the maximum allowable error of the voltage divider network. These values are recorded as the "as left" values.

The values that are recorded as the leakage currents are calculated by dividing the full scale voltages by the acceptable resistances measured on the Insulation Resistance simulator. This is done for both the "as found" case and the "as left" case. This provides traceable records for how the test system was functioning before and after calibration.

Timing Verification

Since the specification of the transducer requires that the device be exposed to the high voltages for a given time, the timing of the system clock must be calibrated. This is achieved by connecting a standard storage scope across TP102 and TP103 in the calibration panel, Figure 3.6. This places the scope across the voltage divider network. A selected test on the Insulation Resistance simulator is placed in the accept state. The software runs through the testing procedure after the safe/arm key is placed in the arm position. A trace on the scope is generated. This reading is recorded from the 50 percent points on the trace and must fall between two and nine seconds.

System Check

Both the Continuity simulator and the Insulation Resistance simulator are connected to their appropriate fixture. The entire testing process is executed for all tests.

The results of the Continuity tests are compared to those values obtained from measuring the Continuity simulator with the standard meter. All values must fall within the acceptable range.

By performing all of these tests, it is verified that everything is configured and functioning properly.

CHAPTER VI

SUMMARY AND CONCLUSIONS

Before fabrication of the Automated Non-destructive Test System was started drawings were generated and a general software development plan was devised. As fabrication proceeded it became apparent that there were better ways to achieve the desired effect. This included some minor alterations to the hardware design.

One such alteration consisted of the addition of the Keithley Digital Micro-ohmmeter. The original design used the Hewlett Packard Digital Multimeter to make the three Continuity measurements in addition to measuring the voltages. Even though the Hewlett Packard meter has the capability of making a four-wire resistance measurement, it proved not to be stable in the milli-ohm range. This left the only measurement made by the Hewlett Packard to be the voltage readings. The addition of the Keithley meter resulted in the changing of some of the switching circuitry. The Hewlett Packard meter had to no longer be switched between taking resistance readings and measuring the high voltages. By having two meters dedicated to specific tasks some relays were eliminated.

Software development evolved around the fabrication of the system. The control and timing of the tests were

established at this time. During this development it was determined that the comparator circuits were too sensitive. They had a tendency to trigger when the high voltage was first applied even though the reference voltage was never exceeded. After investigating this problem, it was realized that it was the sudden change in voltage that was causing this false trigger. The solution was to let the voltage ramp up to the desired value using the internal rise time of the Venus high voltage power supply. This was accomplished in the software by controlling the order in which the relays were closed.

During the first stages of development of the system a simulator was used in testing the hardware. Everything functioned satisfactory. The first transducers to be tested on the system were mock inert units. When testing these a failure status on one specific Insulation Resistance test kept recurring. This raised the concern over whether all the transducers built had the same fault or whether ANTS was malfunctioning. The integrity of the tester was verified by use of the simulator. After scrutinizing the units, it was determined that they met all the qualifications. It became apparent that there was some form of interface problem between the fixture and the actual transducer that did not occur between the fixture and the simulator. After further investigation it was discovered that a thin clear coating of epoxy type material existed on the case of the device. This coating acted as a dielectric between the two points of high voltage

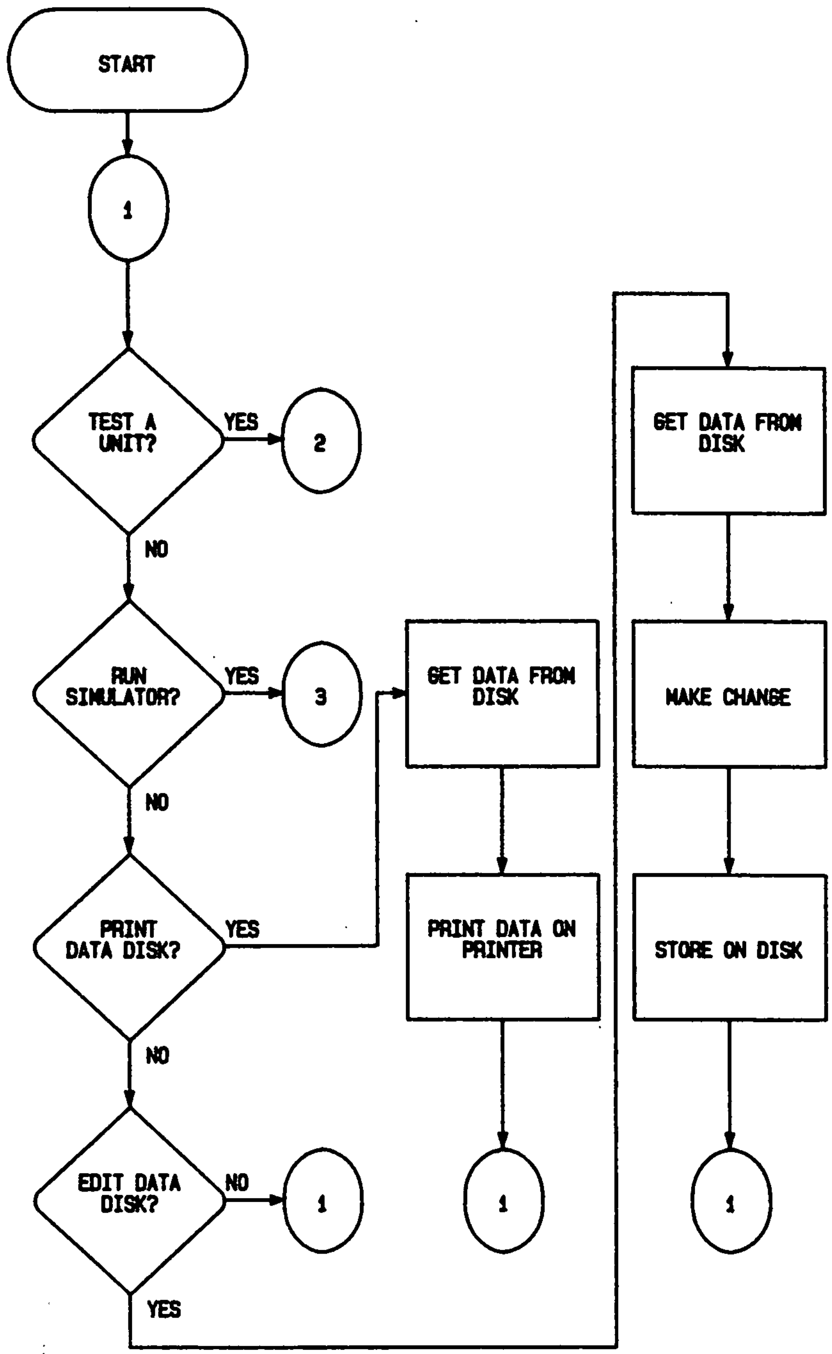
application. A charge was building up on these two "plates", thus resulting in a capacitor type behavior. This problem was eliminated by modifying the fixture so the contact to the case would be in a location where none of this coating would be present.

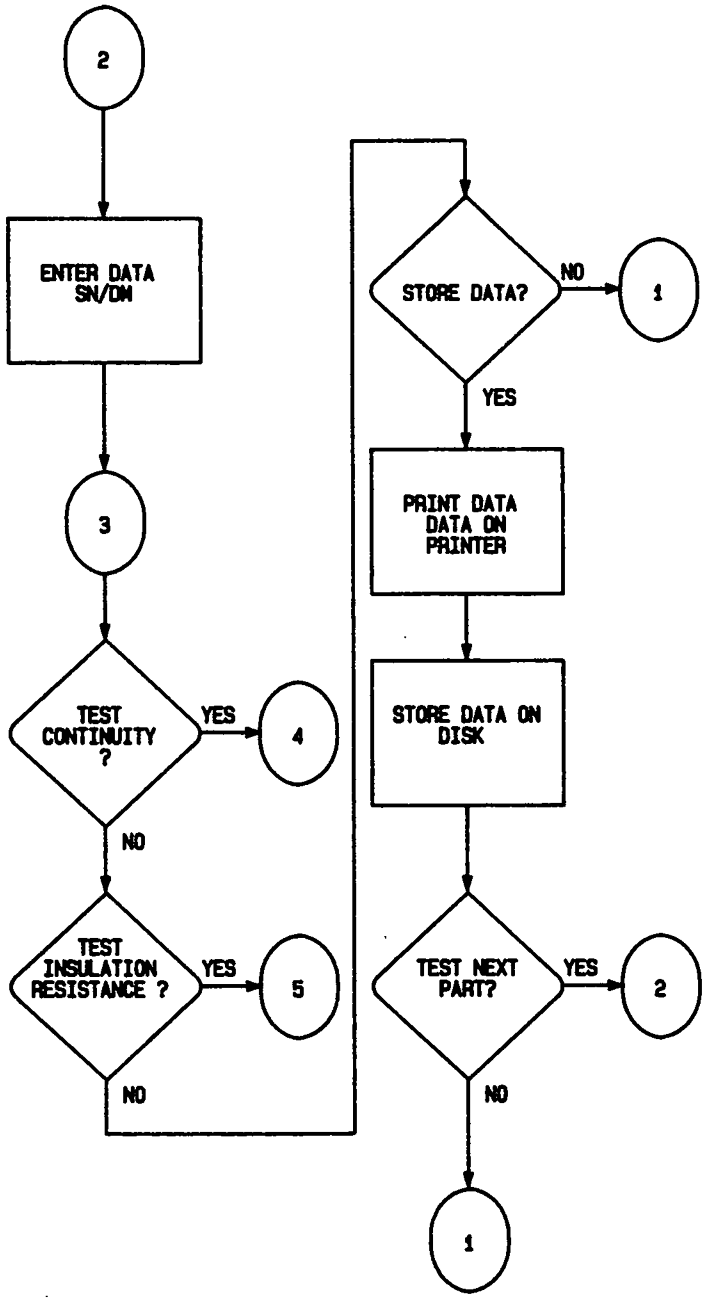
A general calibration plan was outlined during the design of ANTS. This included test points to all the voltages and adjust terminals to the various potentiometers to be located behind the calibration panel on the front of the control chassis. The specifics were developed after the fabrication was complete.

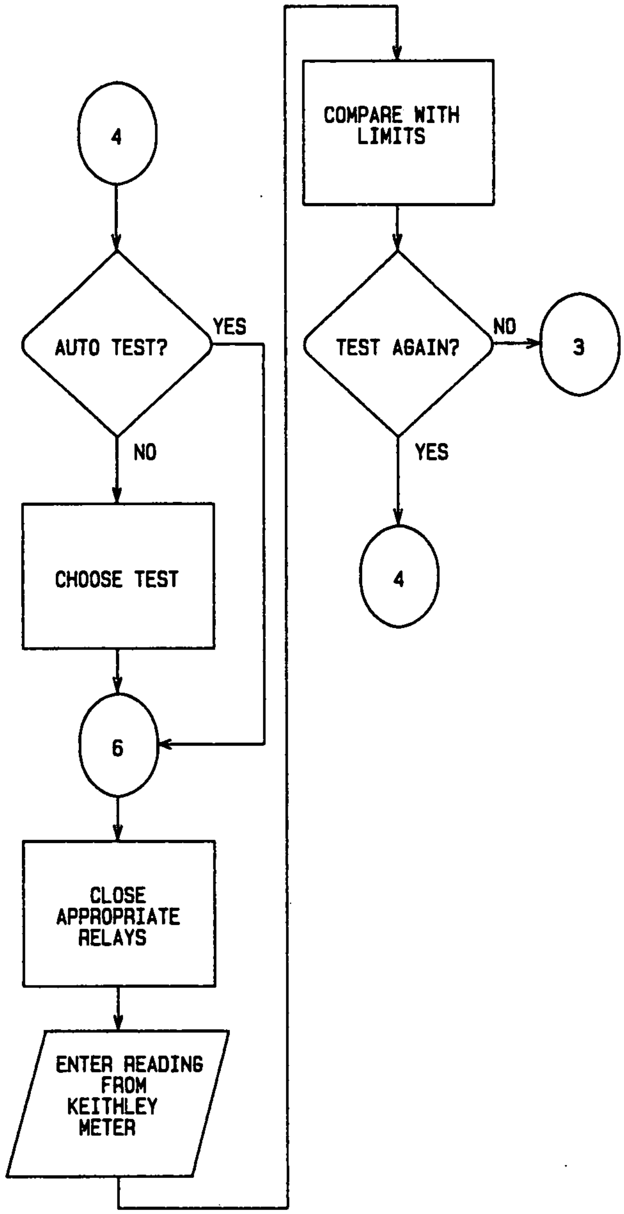
The Automated Non-destructive Test System was completed in time to support development testing of the transducers at the Mound. This provided an automated means of testing the units and storing the data for later comparisons with production units. The test system has proved to be very reliable in detecting errors in Continuity and poor Insulation Resistances.

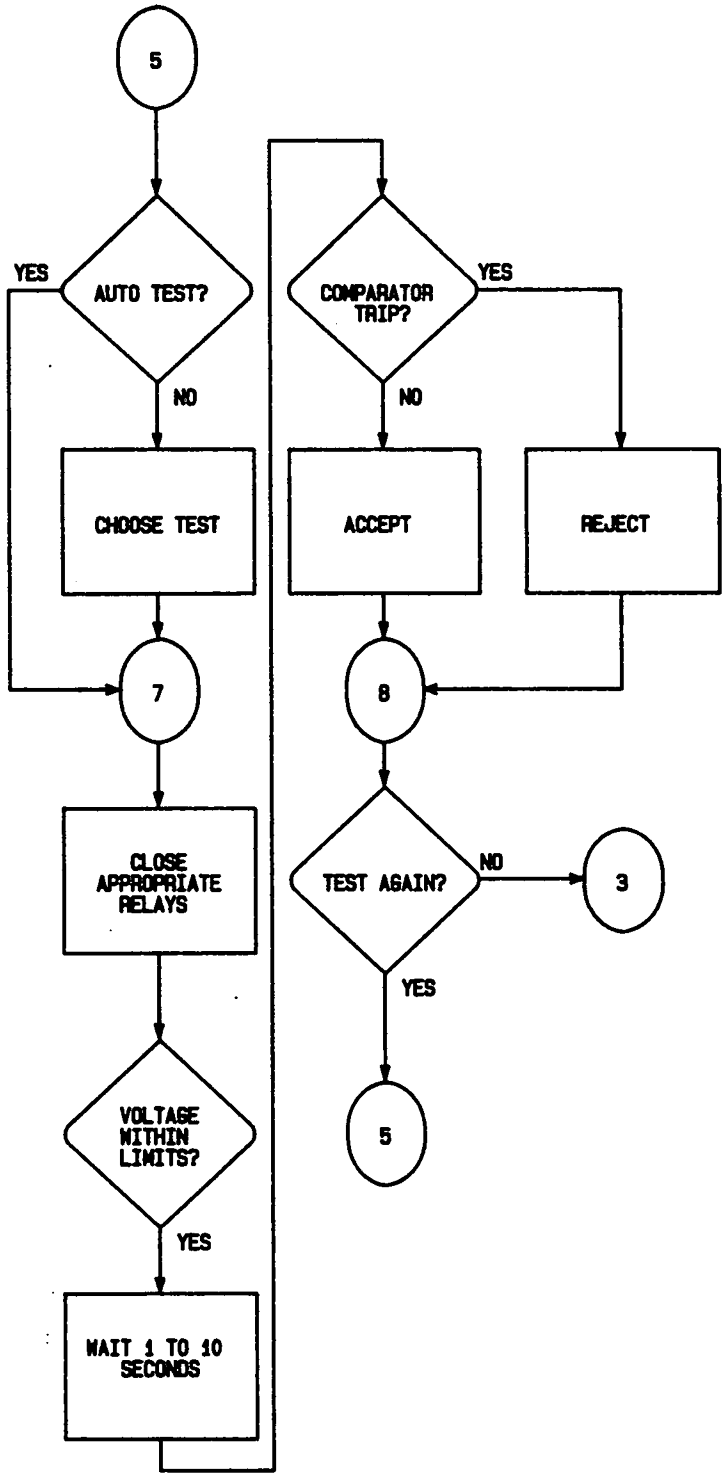
APPENDIX A

OPERATION FLOWCHART


START
1
TEST A UNIT?
YES
2
NO
RUN SIMULATOR?
YES
3
NO
PRINT DATA DISK?
YES
NO
EDIT DATA DISK?
NO
1
YES
GET DATA FROM DISK
PRINT DATA ON PRINTER
1
GET DATA FROM DISK
MAKE CHANGE
STORE ON DISK
1

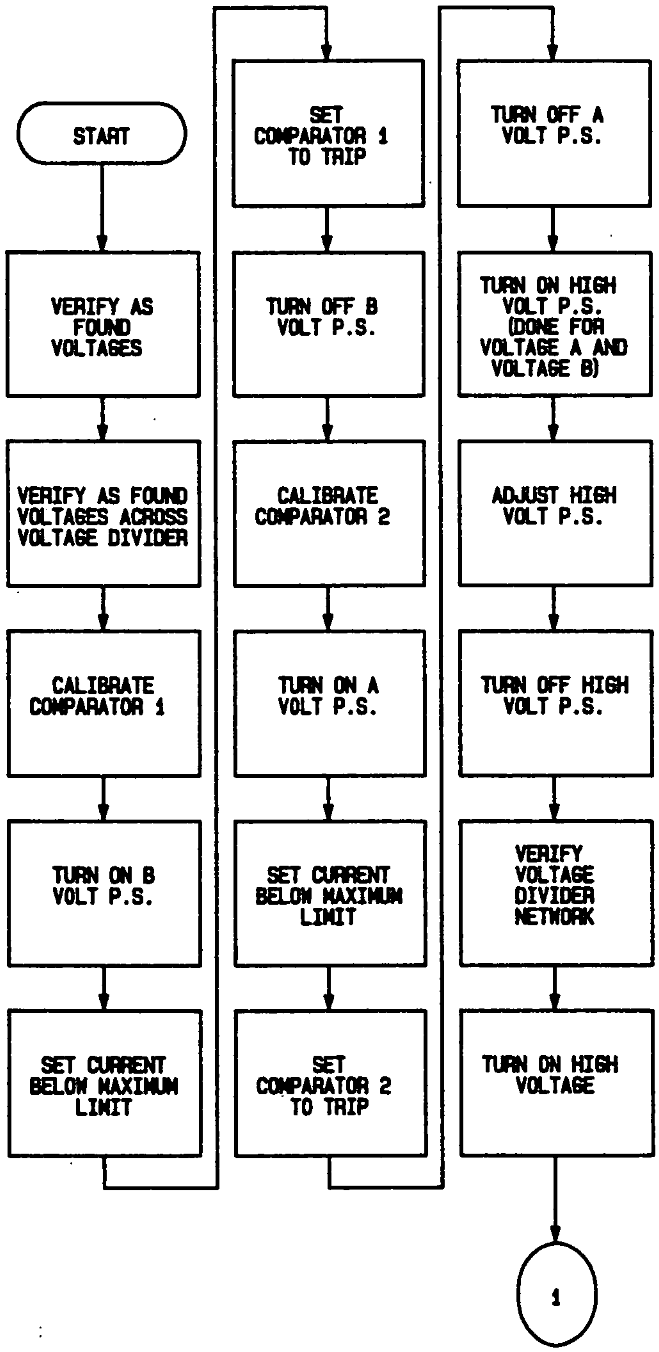

2
ENTER DATA
SN/DM
3
TEST
CONTINUITY
?
YES
4
NO
TEST
INSULATION
RESISTANCE ?
YES
5
NO
STORE DATA?
NO
1
YES
PRINT DATA
DATA ON
PRINTER
STORE DATA ON
DISK
TEST NEXT
PART?
YES
2
NO
1

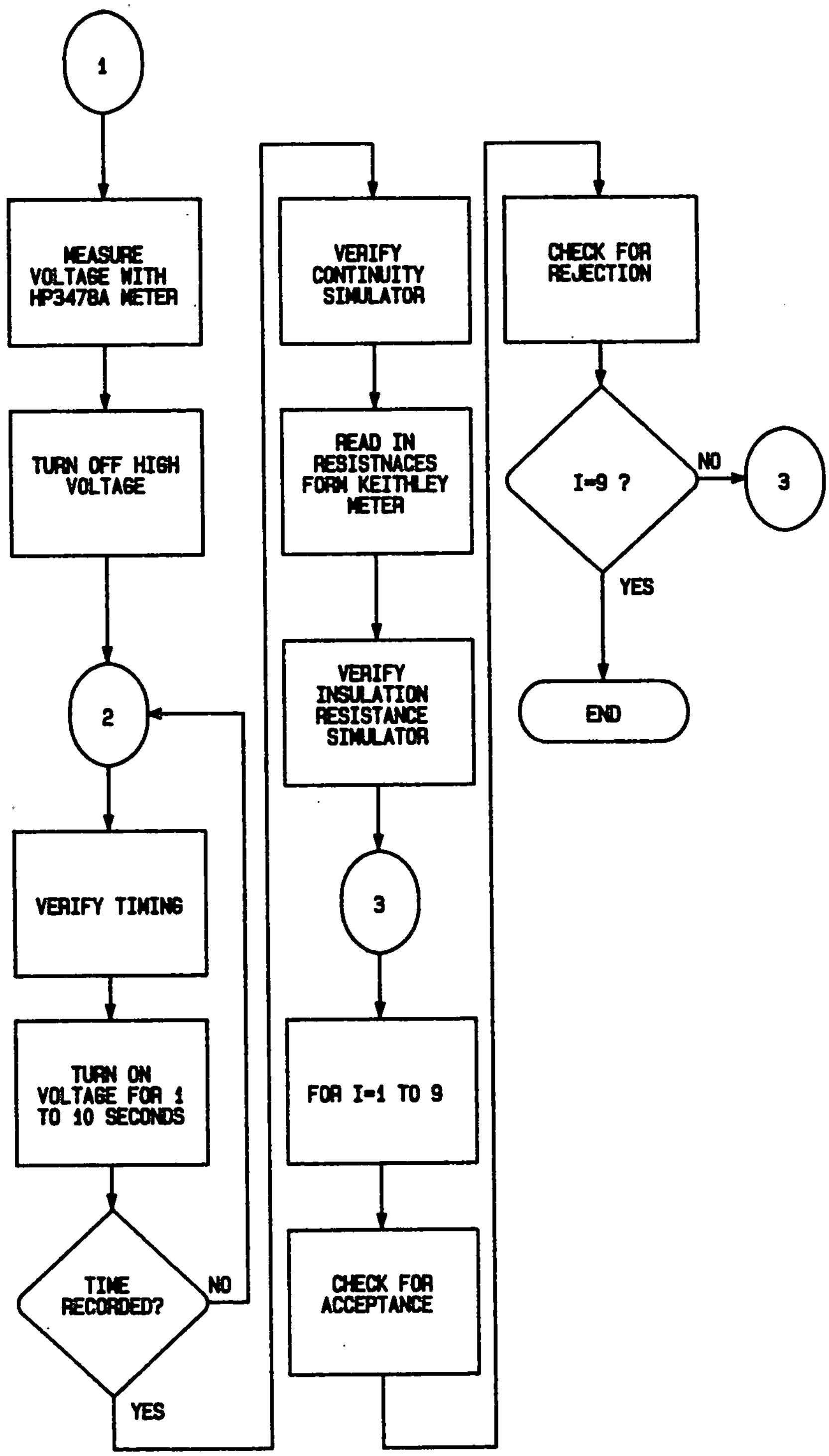

4
AUTO TEST?
YES
NO
CHOOSE TEST
6
CLOSE APPROPRIATE RELAYS
ENTER READING FROM KEITHLEY METER
COMPARE WITH LIMITS
TEST AGAIN?
NO
3
YES
4


5
AUTO TEST?
YES
NO
CHOOSE TEST
7
CLOSE APPROPRIATE RELAYS
VOLTAGE WITHIN LIMITS?
YES
WAIT 1 TO 10 SECONDS
COMPARATOR TRIP?
YES
NO
ACCEPT
REJECT
8
TEST AGAIN?
NO
3
YES
5

APPENDIX B

CALIBRATION FLOWCHART


START
VERIFY AS FOUND VOLTAGES
VERIFY AS FOUND VOLTAGES ACROSS VOLTAGE DIVIDER
CALIBRATE COMPARATOR 1
TURN ON B VOLT P.S.
SET CURRENT BELOW MAXIMUM LIMIT
SET COMPARATOR 1 TO TRIP
TURN OFF B VOLT P.S.
CALIBRATE COMPARATOR 2
TURN ON A VOLT P.S.
SET CURRENT BELOW MAXIMUM LIMIT
SET COMPARATOR 2 TO TRIP
TURN OFF A VOLT P.S.
TURN ON HIGH VOLT P.S. (DONE FOR VOLTAGE A AND VOLTAGE B)
ADJUST HIGH VOLT P.S.
TURN OFF HIGH VOLT P.S.
VERIFY VOLTAGE DIVIDER NETWORK
TURN ON HIGH VOLTAGE
1


1
MEASURE VOLTAGE WITH HP3478A METER
TURN OFF HIGH VOLTAGE
2
VERIFY TIMING
TURN ON VOLTAGE FOR 1 TO 10 SECONDS
TIME RECORDED?
NO
YES
VERIFY CONTINUITY SIMULATOR
READ IN RESISTNACES FORM KEITHLEY METER
VERIFY INSULATION RESISTANCE SIMULATOR
3
FOR I=1 TO 9
CHECK FOR ACCEPTANCE
CHECK FOR REJECTION
I=9 ?
NO
3
YES
END

APPENDIX C

SELECTED OPERATION SOFTWARE LISTING

```
240!*******************************************************************************
250!*******************************************************************************
260 Intro_screen:  ! DISPLAYS INTRO SCREEN
270   DISP
280   PRINT CHR$(12)
290   GRAPHICS ON
300   GCLEAR
310   MOVE 22,70
320   CSIZE 30,.5
330   LABEL "ANTS"
340   CSIZE 8,.4
350   MOVE 2,65
360   LABEL "CONTINUITY-INSULATION RESISTANCE TESTER"
370   MOVE 44,55
380   LABEL Am$
390   DISP "Press ANY key when ready"
400   ON KBD GOTO Start
410 Loop_a:GOTO Loop_a
420 !******************************************************************************
430 !******************************************************************************
440 Start:!
450   IF Re<>0 THEN OUTPUT KBD USING "#,K";" |"
460   GOSUB Init_const
470   GOSUB Init_test_var
480   GOSUB Addressing
490   GCLEAR
500   DISP ""
510   GOSUB Safe_tester
520   PRINT CHR$(12)
530   GOSUB Reset_ir
540   GOSUB Main_menu
550   GOSUB Ds_menu
560   GOSUB Data_disk
570   GOSUB Header_info
580   GOSUB Header_verify
590 !************************************************************************
600 !************************************************************************
610 Main_menu: ! GIVES TEST AND DATA CHOICES
620   GCLEAR
630   New_lot=0
640   Name$="****** MAIN MENU ******"
650   DATA " TEST NEW LOT "
660   DATA "TEST_NEW"
670   DATA " CONTINUE TESTING "
680   DATA "CONT_TEST"
690   DATA " RUN SIMULATORS "
700   DATA "SIM"
710   DATA " EDIT DATA DISK "
720   DATA "EDIT_DATA"
730   DATA " PRINT DATA DISK"
740   DATA "PRINT_DATA"
750   RESTORE 640
```

```
760   Menue_items=5
770   GOSUB Menue
780   OUTPUT KBD USING "#,K";" {"      ! USER SOFTKEYS
790   OUTPUT KBD USING "#,K";" |"      !SOFTKEYS ON
800   ON KEY 1 LABEL "UP" GOSUB Move_down
810   ON KEY 2 LABEL "DOWN" GOSUB Move_up
820   ON KEY 3 GOTO Loop_b
830   ON KEY 4 LABEL "SELECT" GOTO Selecta
840   ON KEY 5 GOTO Loop_b
850   ON KEY 6 GOTO Loop_b
860   ON KEY 7 GOTO Loop_b
870   ON KEY 8 GOTO Loop_b
880   ON KBD GOTO Loop_b
890 Loop_b:GOTO Loop_b
900 Selecta:OFF KEY
910   OUTPUT KBD USING "#,K";" |"
920   OFF KBD
930   PRINT CHR$(12)
940   IF Select$="EDIT_DATA" THEN GOTO Edit_data
950   IF Select$="PRINT_DATA" THEN GOTO Print_data
960   IF Select$="SIM" THEN GOTO Simulator_ck
970   IF Select$="TEST_NEW" THEN New_lot=1
980   RETURN
990  !***************************************************************************
1000 !***************************************************************************
1010 Ds_menu:!                              SELECTS DATA SOURCE CODE
1020  IF NOT (New_lot) AND Re=0 THEN RETURN
1030  GCLEAR
1040  Name$="***** SOURCE CODE MENUE *****"
1050  DATA " PRODUCTION TESTING (NO DATA DISK)"
1060  DATA "PRO_TEST"
1070  DATA " ENGINEERING EVALUATION (CODE L)"
1080  DATA "L"
1090  DATA " PRODUCT ACCEPTANCE (CODE S)"
1100  DATA "S"
1110  DATA " PRODUCT REACCEPTANCE (CODE R)"
1120  DATA "R"
1130  DATA " STOCKPILE SAMPLE (CODE Q)"
1140  DATA "Q"
1150  DATA " QAIA TESTING (CODE Z)"
1160  DATA "Z"
1170  DATA " DEVELOPMENT TESTING AT MOUND (CODE D)"
1180  DATA "D"
1190  RESTORE 1040
1200  Menue_items=7
1210  GOSUB Menue
1220  OUTPUT KBD USING "#,K";" |"
1230  ON KEY 1 LABEL "UP" GOSUB Move_down
1240  ON KEY 2 LABEL "DOWN" GOSUB Move_up
1250  ON KEY 3 GOTO Loop_c
1260  ON KEY 4 LABEL "SELECT" GOTO Selectb
1270  ON KEY 5 GOTO Loop_c
```

```
1280  ON KEY 6 GOTO Loop_c
1290  ON KEY 7 GOTO Loop_c
1300  ON KEY 8 GOTO Loop_c
1310  ON KBD GOTO Loop_c
1320  IF Re<>0 THEN
1330    CONTROL CRT;1,18
1340    PRINT "DATA SOURCE CODE ON DISK IS "&Data_source$&".  MAKE SELECTION"
1350  END IF
1360 Loop_c:GOTO Loop_c
1370 Selectb:OFF KEY
1380  OUTPUT KBD USING "#,K";" |"
1390  OFF KBD
1400  PRINT CHR$(12)
1410  IF Select$="PRO_TEST" THEN
1420    Select$="PRODUCTION TESTING"
1430    IF Re<>0 THEN Ds_menu
1440    Pro_test=1
1450  END IF
1460  Data_source$=Select$
1470  Output$="DATA SOURCE CODE "&Data_source$
1480  RETURN
1490 !***************************************************************************
1500 !***************************************************************************
1510 Menue:  ! CONTROLS ALL MENUE CHOICES
1520 !
1530  Longest=0
1540  FOR I=1 TO Menue_items
1550    READ Menue$(I)
1560    READ Progr$(I)
1570    IF LEN(Menue$(I))>Longest THEN Longest=LEN(Menue$(I))
1580  NEXT I
1590    !
1600  PRINT CHR$(12)
1610  Min_row=INT(10-(Menue_items/2))+2
1620  Row=Min_row
1630  Max_row=Min_row+Menue_items-1
1640  IF LEN(Name$)>Longest THEN Longest=LEN(Name$)
1650  Col=INT(40-(Longest/2))
1660  CONTROL CRT;Col-1,Row-2
1670  PRINT Name$;
1680  FOR Row=Min_row TO Max_row
1690    CONTROL CRT;Col,Row
1700    IF Row=Min_row THEN
1710      PRINT CHR$(129);CHR$(140);Menue$(Row-Min_row+1);CHR$(128);CHR$(136);
1720      DISP "Selection is '";Menue$(Row-Min_row+1);"'"
1730      Select$=Progr$(Row-Min_row+1)
1740    ELSE
1750      PRINT Menue$(Row-Min_row+1)
1760    END IF
1770  NEXT Row
1780  Row=Min_row
1790  RETURN
```

```
1800 !**************************************************************************
1810 Move_down:!
1820  CONTROL CRT;Col,Row
1830  PRINT Menue$(Row-Min_row+1);
1840  Row=Row-1
1850  IF Row<Min_row THEN Row=Max_row
1860  CONTROL CRT;Col,Row
1870  PRINT CHR$(129);CHR$(140);Menue$(Row-Min_row+1);CHR$(128);CHR$(136);
1880  DISP "Selection is '";Menue$(Row-Min_row+1);"'"
1890  Select$=Progr$(Row-Min_row+1)
1900  RETURN
1910  !*************************************************************************
1920 Move_up: !
1930  CONTROL CRT;Col,Row
1940  PRINT Menue$(Row-Min_row+1);
1950  Row=Row+1
1960  IF Row>Max_row THEN Row=Min_row
1970  CONTROL CRT;Col,Row
1980  PRINT CHR$(129);CHR$(140);Menue$(Row-Min_row+1);CHR$(128);CHR$(136);
1990  DISP "Selection is '";Menue$(Row-Min_row+1);"'"
2000  Select$=Progr$(Row-Min_row+1)
2010  RETURN
2020  !
2030  !*************************************************************************
2040  !*************************************************************************
2050 Get_old_header: !  THIS ROUTINE GETS PREVIOUSLY ENTERED DATA FROM DATA DISK
2060                  !
2070  IF Pro_test THEN RETURN
2080  IF New_lot THEN RETURN
2090                  !
2100  ENTER @Stat,1;Record$
2110  Record=VAL(Record$)
2120  Rd$=Record$
2130  IF Re<>0 THEN Rd$=Re$
2140  Rd=VAL(Rd$)
2150  ENTER @Data,Rd;Str_in$
2160  Pn_pos=POS(Str_in$,"PN")
2170  Pn_pos=Pn_pos+3
2180  Part_num$=Str_in$[Pn_pos;9]
2190  Ps_pos=POS(Str_in$,"PS")
2200  Ps_pos=Ps_pos+6
2210  Test_spec_num$=Str_in$[Ps_pos;8]
2220  Ln_pos=POS(Str_in$,"LN")
2230  Ln_pos=Ln_pos+3
2240  Lot_num$=Str_in$[Ln_pos;4]
2250  Ds_pos=POS(Str_in$,"DS")
2260  Data_source$=Str_in$[Ds_pos+3;1]
2270  Date_pos=POS(Str_in$,"TD")
2280  Date$=Str_in$[Date_pos+3;8]
2290  Sn_pos=POS(Str_in$,"SN")
2300  Serial_num$=Str_in$[Sn_pos+3;6]
2310  Dm_pos=POS(Str_in$,"DM")
```

```
2320  Date_manuf$=Str_in$[Dm_pos+3;3]
2330  RETURN
2340  !
2350  !***********************************************************************
2360  !***********************************************************************
2370 Header_info:! ROUTINE ENTERS HEADER INFORMATION
2380 !
2390  IF NOT (New_lot) THEN RETURN
2400 E_pn: ! ENTERS PART NUMBER
2410  Example$="XXXXXX-XX"
2420  Pos=9
2430  Output$=Part_num$
2440  Item$="PART NUMBER SUFFIX? "
2450  GOSUB Kbd_entry
2460  Part_num$=Output$
2470  IF Re<>0 THEN RETURN
2480 E_ps: ! ENTERS PRODUCT SPECIFICATION NUMBER
2490  Example$="XXXXXX-X"
2500  Pos=9
2510  Item$="TEST SPEC. NUMBER ISSUE"
2520  Output$=Test_spec_num$
2530  GOSUB Kbd_entry
2540  Test_spec_num$=Output$
2550  IF Re<>0 THEN RETURN
2560  !
2570 E_ln: ! ENTERS LOT NUMBER
2580  Example$="XXXX"
2590  Pos=1
2600  Output$=Lot_num$
2610  Item$="LOT NUMBER"                  !  ENTERS LOT NUMBER
2620  GOSUB Kbd_entry
2630  Lot_num$=Output$
2640  IF Re<>0 THEN RETURN
2650 E_date: ! ENTERS DATE TESTED
2660  Example$="XX-XX-XX"
2670  Pos=1
2680  Output$=Date$
2690  Item$="DATE: (MM-DD-YY)"      !  ENTERS DATE
2700  GOSUB Kbd_entry
2710  Date$=Output$
2720  RETURN
2730  !
2740 E_sn:! ENTERS SERIAL NUMBER
2750  Example$="XXXXXX"
2760  Item$="SERIAL NUMBER: "                 !  INPUT SERIAL NUMBER
2770  IF Re=0 THEN
2780    Output$=""
2790  ELSE
2800    Output$=Serial_num$
2810  END IF
2820  Pos=1
2830  Dum$=KBD$
```

```
2840  GOSUB Kbd_entry
2850  Point=POS(Output$,"X")
2860  Serial_num$=Output$
2870  IF Re<>0 THEN RETURN
2880 E_dm: !  ENTERS DATE MANUFACTURED CODE
2890  Example$="XXX"
2900  IF Re=0 THEN
2910    Output$=""
2920  ELSE
2930    Output$=Date_manuf$
2940  END IF
2950  Pos=1
2960  Item$="DATE MANUFACTURED  CODE "
2970  GOSUB Kbd_entry
2980  Date_manuf$=Output$
2990  IF Date_manuf$="KKK" THEN Date_manuf$="   "
3000  IF Re<>0 THEN RETURN
3010  RETURN
3020  !
3030  !****************************************************************************
3040  !****************************************************************************
3050 Kbd_entry: ! CONTROLS OPERATOR KEYED INPUTS
3060  Temp=Pos
3070  Exit=No
3080  IF Re<>0 THEN
3090    DISP "MAKE CHANGE TO ";Item$&" THEN DEPRESS ENTER BUTTON"
3100  ELSE
3110    DISP ""
3120    DISP "ENTER  > ";Item$&" Depress ENTER button"
3130  END IF
3140  IF LEN(Output$)<1 THEN Output$=Example$
3150  IF LEN(Output$)<>LEN(Example$) THEN Output$=""
3160  Key$=CHR$(255)&CHR$(60)
3170  GOSUB Process_key1
3180  ON KBD GOSUB Process_key
3190  IF Re<>0 THEN OFF KEY
3200  REPEAT
3210  UNTIL Exit
3220  IF POS(Output$,"X") THEN
3230    Output$=Example$
3240    Pos=Temp
3250    GOTO Kbd_entry
3260  END IF
3270  CONTROL CRT,1;15
3280  PRINT "                    "
3290  CONTROL CRT,1;16
3300  PRINT "                    "
3310  RETURN
3320 !
3330 Process_key:Key$=KBD$
3340 Process_key1:Key_code=NUM(Key$[1;1])
3350  SELECT Key_code
```

```
3360  CASE 45 TO 57
3370    GOSUB Fix_string
3380  CASE 65 TO 90
3390    GOSUB Fix_string
3400  CASE 97 TO 122
3410    GOSUB Fix_string
3420  CASE 255
3430    IF Key$[2;1]=CHR$(60) THEN
3440      Pos=Pos-1
3450      IF Pos<1 THEN Pos=1
3460      Temp$=Example$[Pos;1]
3470      IF Temp$="-" OR Temp$="." THEN Pos=Pos-1
3480      IF Pos<1 THEN Pos=1
3490    END IF
3500    IF Key$[2;1]=CHR$(62) THEN
3510      Pos=Pos+1
3520      Temp$=Example$[Pos;1]
3530      IF Temp$="-" OR Temp$="." THEN Pos=Pos+1
3540      IF Pos=14 THEN Pos=15
3550    END IF
3560    IF Pos>LEN(Example$) THEN Pos=LEN(Example$)
3570    IF Key$[2;1]=CHR$(69) THEN
3580      Exit=Yes
3590    END IF
3600    IF Key$[2;1]=CHR$(67) THEN
3610      Exit=Yes
3620    END IF
3630    IF Key$[2;1]=CHR$(88) THEN
3640      Exit=Yes
3650    END IF
3660  CASE ELSE
3670    Pos=Pos+1
3680    IF Pos>LEN(Example$) THEN Pos=LEN(Example$)
3690  END SELECT
3700  CONTROL CRT,1;15
3710  PRINT Output$
3720  CONTROL CRT,1;16
3730  CONTROL CRT,0;1
3740  PRINT "                                             "
3750  CONTROL CRT,1;16
3760  CONTROL CRT,0;Pos
3770  PRINT "^"
3780  RETURN
3790 Fix_string:Output$[Pos;1]=Key$
3800  Pos=Pos+1
3810  Temp$=Example$[Pos;1]
3820  IF Temp$="-" OR Temp$="." THEN Pos=Pos+1
3830  IF Pos>LEN(Example$) THEN Pos=LEN(Example$)
3840  RETURN
3850 !
3860 Question: !
3870  ON KBD GOTO Answer
```

```
3880  CONTROL CRT,1;17
3890  PRINT Item$
3900 Re_loop:DISP "PRESS Y OR N KEY"
3910  WAIT .25
3920  DISP "PRESS   OR   KEY"
3930  WAIT .25
3940  GOTO Re_loop
3950 Answer:Keyb$=KBD$
3960  IF Keyb$="Y" THEN
3970    OFF KBD
3980    Answer$="Y"
3990    CONTROL CRT,1;17
4000    PRINT USING "50X"
4010    DISP
4020    RETURN
4030  END IF
4040  IF Keyb$="N" THEN
4050    OFF KBD
4060    Answer$="N"
4070    CONTROL CRT,1;17
4080    PRINT USING "50X"
4090    DISP
4100    RETURN
4110  END IF
4120  GOTO Question
4130  !
4140  !*************************************************************************
4150  !*************************************************************************
4160 Header_verify:! GIVES CHANCE TO VERIFY OPERATOR INPUTS
4170  PRINT CHR$(12)
4180  DISP ""
4190  IF NOT (New_lot) THEN GOSUB E_date
4200  CONTROL CRT;10,5
4210  GOSUB Pr_to_printer
4220 Edit_q:ON KBD GOTO Edit_ans
4230  DISP "DO YOU NEED TO EDIT THIS DATA(Y/N)?"
4240 Wait11:GOTO Wait11
4250 Edit_ans:A$=KBD$
4260  IF A$="Y" THEN
4270    IF New_lot THEN
4280      GOSUB Ds_menu
4290      GOSUB Header_info
4300      GOTO Header_verify
4310    END IF
4320    GOSUB Close_file
4330    GOTO Start
4340  END IF
4350  IF A$<>"N" THEN Edit_q
4360  PRINT CHR$(12)
4370  DISP ""
4380  GOSUB E_sn
4390  !*************************************************************************
```

```
4400  !***************************************************************************
4410 Test_choice:! GIVES CHOICE TO TEST I.R./TEST CONT./STORE DATA
4420  DISP ""
4430  PRINT CHR$(12)
4440  CONTROL CRT;1,15
4450  PRINT "DO YOU WANT TO TEST CONTINUITY OR INSULATION RESISTANCE?"
4460  OUTPUT KBD USING "#,K";" |"
4470  ON KEY 1 LABEL " CONT." GOTO Cont_sel
4480  ON KEY 2 GOTO Wait
4490  ON KEY 3 GOTO Wait
4500  ON KEY 4 LABEL "  I.R." GOTO Ir_sel
4510  ON KEY 5 LABEL "  STORE   DATA  " GOTO Print_out
4520  ON KEY 6 GOTO Wait
4530  ON KEY 7 GOTO Wait
4540  ON KEY 8 LABEL "   END   TESTING" GOTO End_t
4550  ON KBD GOTO Wait
4560 Wait:GOTO Wait
4570 End_t:OUTPUT KBD USING "#,K";" |"
4580  GOSUB Close_file
4590  GOTO Start
4600  !
4610  !***************************************************************************
4620 Cont_sel:! GETTING READY TO TEST CONTINUITY
4630  OFF KBD
4640  OUTPUT KBD USING "#,K";" |"
4650  DISP ""
4660  PRINT CHR$(12)
4670  CONTROL CRT;1,15
4680  Disp$="PLACE PART IN CONTINUITY FIXTURE AND PLACE IN SAFETY CHAMBER AND CL
OSE DOOR."
4690  GOSUB Wait_for_key
4700 Cont_sel2:! GIVES CHOICE TO TEST 1 OR ALL CONTINUITY TESTS
4710  OFF KBD
4720  PRINT CHR$(12)
4730  DISP ""
4740  ON KBD GOTO Cont_sel3
4750  Cont_test_num=0
4760  DISP "DO YOU  WANT TO AUTOMATICALY RUN THROUGH ALL THREE(3) CONTINUITY TES
TS?(Y/N)"
4770 Wait3:GOTO Wait3
4780 Cont_sel3:A$=KBD$
4790  OFF KBD
4800  DISP ""
4810  IF A$="Y" THEN Test_cont
4820  IF A$<>"N" THEN Cont_sel2
4830 Man_cont:! PREPARE TO MANUALLY RUN THROUGH CONT. TESTS
4840  INPUT "ENTER CONTINUITY TEST NUMBER (1-3)",Cont_test_num$
4850  IF Cont_test_num$<>"1" THEN
4860    IF Cont_test_num$<>"2" THEN
4870      IF Cont_test_num$<>"3" THEN Man_cont
4880    END IF
4890  END IF
```

```
4900  Cont_test_num=VAL(Cont_test_num$)
4910  GOSUB Test_cont
4920  RETURN
4930  !
4940  !****************************************************************************
4950  !****************************************************************************
4960 Ir_sel: ! GETTING READY TO TEST INSULATION RESISTANCE TESTS
4970  OFF KBD
4980  OUTPUT KBD USING "#,K";" |"
4990  DISP ""
5000  PRINT CHR$(12)
5010  CONTROL CRT;1,15
5020  Disp$="PLACE PART IN INSULATION RESISTANCE FIXTURE AND PLACE IN SAFETY CHA
MBER AND     CLOSE DOOR "
5030  GOSUB Wait_for_key
5040 Ir_sel2: ! GIVES CHOICE TO RUN 1 OR ALL INSULATION RESISTNACE TESTS
5050  OFF KBD
5060  PRINT CHR$(12)
5070  DISP ""
5080  ON KBD GOTO Ir_sel3
5090  Ir_test_sel=0
5100  Ir_test_sel$=VAL$(Ir_test_sel)
5110  DISP "DO YOU WANT TO AUTOMATICALLY RUN THROUGH ALL NINE(9) I.R. TESTS?(Y/N
)"
5120 Wait5:GOTO Wait5
5130 Ir_sel3:A$=KBD$
5140  OFF KBD
5150  IF A$="Y" THEN Test_ir
5160  IF A$<>"N" THEN Ir_sel2
5170 Man_ir:! PREPARE TO MANUALLY RUN THROUGH INSULATION RESISTACE TESTS
5180  INPUT "ENTER I.R. TEST NUMBER (1-9)",Ir_test_sel$
5190  IF Ir_test_sel$<>"1" THEN
5200    IF Ir_test_sel$<>"2" THEN
5210      IF Ir_test_sel$<>"3" THEN
5220        IF Ir_test_sel$<>"4" THEN
5230          IF Ir_test_sel$<>"5" THEN
5240            IF Ir_test_sel$<>"6" THEN
5250              IF Ir_test_sel$<>"7" THEN
5260                IF Ir_test_sel$<>"8" THEN
5270                  IF Ir_test_sel$<>"9" THEN Man_ir
5280                END IF
5290              END IF
5300            END IF
5310          END IF
5320        END IF
5330      END IF
5340    END IF
5350  END IF
5360  Ir_test_sel=VAL(Ir_test_sel$)
5370  GOSUB Test_ir
5380  RETURN
5390 !****************************************************************************
```

```
5400 !********************** VARIABLES AND LIMITS *****************************
5410 !***************************************************************************
5420 Init_test_var: !INITIALIZE VARIABLES FOR DATA
5430                !
5440  FOR I=1 TO 9
5450    Ir_stat$(I)=""
5460  NEXT I
5470  FOR I=1 TO 3
5480    R$(I)=""
5490    R(I)=0
5500  NEXT I
5510  RETURN
5520  !
5530  !**************************************************************************
5540 Init_const: !
5550  Part_num$=""
5560  Test_spec_num$=""
5570  Lot_num$=""
5580  Date$=""
5590  Serial_num$=""
5600  Date_manuf$=""
5610                !
5620  !*************************** LIMITS **************************************
5630  Up_volt_lim(1)=(A+C)/1000
5640  Low_volt_lim(1)=(A-C)/1000
5650  FOR I=2 TO 9
5660    Up_volt_lim(I)=(B+D)/1000
5670    Low_volt_lim(I)=(B-D)/1000
5680  NEXT I
5690  !
5700  !*********************** LIMITS FOR CONTINUITY *************************
5710  R_up$(1)="x.xxxx"
5720  R_low$(1)="y.yyyy"
5730  R_up$(2)="x.xxxx"
5740  R_low$(2)="y.yyyy"
5750  R_up$(3)="x.xxxx"
5760  R_low$(3)="y.yyyy"
5770  !
5780  FOR I=1 TO 3
5790    R_up(I)=VAL(R_up$(I))
5800    R_low(I)=VAL(R_low$(I))
5810  NEXT I
5820  !
5830  !*************************** VARIALBES ***********************************
5840  Test_code$="AA"
5850  No=0
5860  Yes=1
5870  Door_mon=0  ! BIT=1 IF DOOR IS OPEN
5880  Key_mon=1   ! BIT=0 IF TESTER IS ARMED
5890  Comparator_mon=2   ! BIT=0 IF IR REJECT
5900  Sim_reject=0
5910  RETURN
```

```
5920  !
5930  !**************************************************************************
5940  !**************************************************************************
5950 Monitor: ! MONITORS SAFETY CHAMBER DOOR, ARM STATE, COMPARATOR STATE
5960  DISP ""
5970  PRINT CHR$(13)
5980  GOSUB Door_open
5990  GOSUB Arm_tester
6000  GOSUB Reset_ir
6010  RETURN
6020  !**************************************************************************
6030  !**************************************************************************
6040 Door_open:    ! MAKES SURE SAFETY CHAMBER DOOR IS CLOSED
6050  ENTER @Gpio_14 USING "#,W";Monitor
6060  IF BIT(Monitor,Door_mon)=1 THEN
6070    CONTROL CRT;25,15
6080    PRINT CHR$(130);"****** CLOSE CHAMBER DOOR *****";CHR$(128)
6090    WAIT 3
6100    GOTO Door_open
6110  END IF
6120  PRINT CHR$(12)
6130  RETURN
6140  !**************************************************************************
6150  !**************************************************************************
6160 Safe_tester:! MAKES SURE KEY IS IN SAFE POSITION
6170  ENTER @Gpio_14 USING "#,W";Monitor
6180  IF BIT(Monitor,Key_mon)=0 THEN
6190    DISP ""
6200    CONTROL CRT;25,15
6210    PRINT CHR$(130);"****** SAFE TESTER *****";CHR$(128)
6220    WAIT 3
6230    GOTO Safe_tester
6240  END IF
6250  PRINT CHR$(12)
6260  RETURN
6270  !**************************************************************************
6280  !**************************************************************************
6290 Arm_tester:! MAKES SURE KEY IS IN ARMED POSITION
6300  ENTER @Gpio_14 USING "#,W";Monitor
6310  IF BIT(Monitor,Key_mon)=1 THEN
6320    CONTROL CRT;25,15
6330    PRINT CHR$(130);"****** ARM TESTER *****";CHR$(128)
6340    WAIT 3
6350    GOTO Arm_tester
6360  END IF
6370  RETURN
6380  !**************************************************************************
6390  !**************************************************************************
6400 Reset_ir:! INSTRUCTS TO RESET INSULATION RESISTANCE BUTTON
6410  ENTER @Gpio_14 USING "#,W";Monitor
6420  IF BIT(Monitor,Comparator_mon)=0 THEN
6430    PRINT TABXY(18,8);CHR$(130);"****** INSULATION RESISTANCE FAILURE ON TEST
```

```
";Ir_test_num;" *****";CHR$(128)
6440    PRINT TABXY(25,13);CHR$(130);"****** RESET I.R. BUTTON *****";CHR$(128)
6450    FOR I=1 TO 3
6460      BEEP 500,.5
6470      WAIT .5
6480    NEXT I
6490    GOTO Reset_ir
6500  END IF
6510  RETURN
6520  !*************************************************************************
6530  !*************************************************************************
6540 Addressing:!
6550  ASSIGN @Volt TO 72300
6560  OUTPUT @Volt;"F1R1"
6570  ASSIGN @Gpio_12 TO 12
6580  ASSIGN @Gpio_14 TO 14
6590  ASSIGN @Keithley TO 72500
6600  RETURN
6610  !
6620  !*************************************************************************
6630  !************************** I.R. TESTING ROUTINE *************************
6640  !*************************************************************************
6650 Test_ir:! ROUTINE TO PERFORM I.R. TESTS
6660  LOCAL LOCKOUT 7
6670  !************************** RELAYS FOR IR TESTING ************************
6680  Volt_to_uut_12(1)=2^0+2^6+2^7+2^10
6690  Volt_to_uut_14(1)=2^14+(-2^15)
6700  Volt_to_uut_12(2)=2^0+2^6
6710  Volt_to_uut_14(2)=2^1+(-2^15)
6720  Volt_to_uut_12(3)=2^0+2^6
6730  Volt_to_uut_14(3)=2^5+(-2^15)
6740  Volt_to_uut_12(4)=2^0+2^6
6750  Volt_to_uut_14(4)=2^1+2^5+2^6
6760  Volt_to_uut_12(5)=2^0+2^6
6770  Volt_to_uut_14(5)=2^0+(-2^15)
6780  Volt_to_uut_12(6)=2^0+2^6
6790  Volt_to_uut_14(6)=2^4+(-2^15)
6800  Volt_to_uut_12(7)=2^0+2^6
6810  Volt_to_uut_14(7)=2^0+2^4+2^6
6820  Volt_to_uut_12(8)=2^0+2^6
6830  Volt_to_uut_14(8)=2^8+(-2^15)
6840  Volt_to_uut_12(9)=2^0+2^6
6850  Volt_to_uut_14(9)=2^7+(-2^15)
6860  !*************************************************************************
6870  !*************************************************************************
6880  FOR Ir_test_num=1 TO 9
6890    GOSUB Monitor
6900    DISP ""
6910    PRINT CHR$(12)
6920    CONTROL CRT;20,15
6930    IF Ir_test_sel<>0 THEN Ir_test_num=Ir_test_sel ! FOR MANUAL TEST OF ONE
TEST
```

```
6940     Ir_test_num$=VAL$(Ir_test_num)
6950     PRINT "TESTING SERIAL NUMBER "&Serial_num$
6960     CONTROL CRT;25,17
6970     PRINT " I.R. TEST NUMBER "&Ir_test_num$
6980     !************************************************************************
6990 Testing:      !
7000     OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ir_test_num)!CLOSE ALL RELAYS
7010     OUTPUT @Gpio_14 USING "#,W";Volt_to_uut_14(Ir_test_num)!EXCEPT TO TURN O
N HIGH VOLT P.S.
7020     IF Ir_test_num=1 THEN
7030       OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ir_test_num)+2^13+2^9!TURN
ON VOLT TO HIGH VOLT P.S.(K111,K110)
7040     ELSE
7050       OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ir_test_num)+2^13!TURN ON V
OLT TO HIGH VOLT P.S.(K111)
7060     END IF
7070     !
7080     !************************* CHECKING VOLTAGE ****************************
7090     IF Sim_reject=0 THEN
7100 Ck_volt:!VERIFY VOLTAGE WITHIN LIMITS
7110       WAIT 1
7120       OUTPUT @Volt;"F1R1"
7130       ENTER @Volt;Volt$
7140       Volt=VAL(Volt$)
7150       ENTER @Gpio_14 USING "#,W";Monitor
7160       IF Serial_num$<>"SIMULATOR" THEN
7170         IF NOT BIT(Monitor,Comparator_mon) THEN Ir_reject
7180       END IF
7190     !
7200       IF Volt>Up_volt_lim(Ir_test_num) THEN Volt_out_range
7210       IF Volt<Low_volt_lim(Ir_test_num) THEN Volt_out_range
7220     !
7230     END IF
7240     FOR I=1 TO 4
7250       WAIT 1
7260       ENTER @Gpio_14 USING "#,W";Monitor
7270       IF NOT BIT(Monitor,Comparator_mon) THEN Ck_comparator
7280     NEXT I
7290     WAIT 1
7300     !************************************************************************
7310 Ck_comparator:   ! CHECKS STATE OF COMPARATOR
7320     OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ir_test_num)!TURNS OFF VOLT T
O HIGH VOLT P.S.
7330     WAIT 3
7340     OUTPUT @Gpio_14 USING "#,W";0
7350     WAIT 4
7360     OUTPUT @Gpio_12 USING "#,W";0
7370        !
7380     IF Serial_num$="SIMULATOR" THEN RETURN
7390     ENTER @Gpio_14 USING "#,W";Monitor
7400     IF NOT BIT(Monitor,Comparator_mon) THEN Ir_reject
7410     WAIT 1
```

```
7420    ENTER @Gpio_14 USING "#,W";Monitor
7430    IF NOT BIT(Monitor,Comparator_mon) THEN Ir_reject
7440 Ir_accept:  !
7450    Ir_stat$(Ir_test_num)="ACCEPT"
7460    GOTO Ir_end
7470 Ir_reject:             !
7480    Ir_stat$(Ir_test_num)="REJECT"
7490    OUTPUT @Gpio_12 USING "#,W";0
7500    OUTPUT @Gpio_14 USING "#,W";0
7510    PRINT CHR$(12)
7520    CONTROL CRT;20,12
7530    GOSUB Reset_ir
7540    IF NOT BIT(Monitor,Comparator_mon) THEN Ir_reject
7550    GOTO Ir_end
7560    !************************************************************************
7570 Volt_out_range:! HIGH VOLT SUPPLY(S) OUT OF LIMITS
7580    OUTPUT @Gpio_12 USING "#,W";0 !CLEARS I/O
7590    OUTPUT @Gpio_14 USING "#,W";0
7600    IF Serial_num$="SIMULATOR" THEN RETURN
7610    PRINT CHR$(12)
7620    Ps$="B VOLT"
7630    IF Ir_test_num=1 THEN Ps$="A VOLT"
7640    CONTROL CRT;20,10
7650    PRINT Ps$&" POWER SUPPLY OUT OF LIMITS "
7660    CONTROL CRT;20,13
7670    PRINT "SAFE TESTER AND NOTIFY AREA FOREMAN"
7680    ON KBD GOTO Start
7690    GOSUB Close_file
7700    DISP "PRESS ANY KEY TO RETURN TO MAIN MENU"
7710 Wait2:GOTO Wait2
7720 Ir_end:!
7730    OUTPUT @Gpio_12 USING "#,W";0 ! CLEARS I/O
7740    OUTPUT @Gpio_14 USING "#,W";0
7750    OUTPUT @Volt;"F1R1"
7760    PRINT CHR$(12)
7770    IF Ir_test_sel<>0 THEN GOSUB Safe_tester
7780    PRINT CHR$(12)
7790 Test_another_ir:!ALLOWS RETEST OR TESTING OF NEXT MANUAL TEST
7800    IF Ir_test_sel<>0 THEN
7810      CONTROL CRT;30,5
7820      PRINT "SERIAL NUMBER "&Serial_num$
7830      CONTROL CRT;25,7
7840      PRINT "I.R. TEST NUMBER "&Ir_test_num$&"     "&Ir_stat$(Ir_test_num)
7850      ON KBD GOTO Test_ir_again
7860      DISP "DO YOU WANT TO TEST I.R. AGAIN?(Y/N)"
7870 Wait7:GOTO Wait7
7880 Test_ir_again:A$=KBD$
7890      IF A$="Y" THEN Ir_sel2
7900      IF A$<>"N" THEN Test_another_ir
7910      GOSUB Safe_tester
7920      GOTO Test_choice
7930    END IF
```

```
7940  NEXT Ir_test_num
7950  GOSUB Safe_tester
7960  CONTROL CRT;30,5
7970  PRINT "SERIAL NUMBER "&Serial_num$
7980  FOR I=1 TO 9
7990    I$=VAL$(I)
8000    CONTROL CRT;25,7+I
8010    PRINT "I.R. TEST NUMBER "&I$&"     "&Ir_stat$(I)
8020  NEXT I
8030 Retest_ir: ! ALLOWS RETEST IF IN AUTOMATIC TEST MODE
8040  ON KBD GOTO Retest_ir_ans
8050  DISP "DO YOU WANT TO RETEST?(Y/N)"
8060 Wait8:GOTO Wait8
8070 Retest_ir_ans:A$=KBD$
8080  IF A$="Y" THEN Ir_sel2
8090  IF A$<>"N" THEN Retest_ir
8100  GOSUB Safe_tester
8110  GOTO Test_choice
8120    !*************************************************************************
8130    !******************* CONTINUITY TESTING ROUTINE ************************
8140    !*************************************************************************
8150 Test_cont:   ! PERFORMS CONTINUITY TESTS
8160  !
8170  GOSUB Monitor
8180  PRINT CHR$(12)
8190  IF Cont_test_num=2 THEN GOTO Cont_2 ! TESTS SELECTED
8200  IF Cont_test_num=3 THEN GOTO Cont_3 ! CONT. TEST IF IN MANUAL MODE
8210  !
8220 Cont_1:OUTPUT @Gpio_12 USING "#,W";2^1+2^3
8230  OUTPUT @Keithley;"R0P0Z0T5G1X"
8240  PRINT CHR$(12)
8250  CONTROL CRT;20,15
8260  PRINT "TESTING SERIAL NUMBER "&Serial_num$
8270  CONTROL CRT;22,17
8280  PRINT "CONTINUITY TEST NUMBER 1"
8290  WAIT 1
8300  ENTER @Keithley;R$(1)
8310  R(1)=VAL(R$(1))
8320  R(1)=INT(R(1)*10^4)/10^4
8330  R$(1)=VAL$(R(1))
8340  FOR I=1 TO 3
8350    IF LEN(R$(1))<6 THEN
8360      R$(1)=R$(1)&"0"
8370    END IF
8380  NEXT I
8390  IF LEN(R$(1))>6 THEN R$(1)=R$(1)[1,6]
8400  IF Cont_test_num<>0 THEN Cont_end
8410 Cont_2:OUTPUT @Gpio_12 USING "#,W";2^1+2^2
8420  OUTPUT @Keithley;"R0P0Z0T5G1X"
8430  PRINT CHR$(12)
8440  CONTROL CRT;20,15
8450  PRINT "TESTING SERIAL NUMBER "&Serial_num$
```

```
8460  CONTROL CRT;22,17
8470  PRINT "CONTINUITY TEST NUMBER 2"
8480  WAIT 1
8490  ENTER @Keithley;R$(2)
8500  R(2)=VAL(R$(2))
8510  R(2)=INT(R(2)*10^4)/10^4
8520  R$(2)=VAL$(R(2))
8530  FOR I=1 TO 6
8540    IF LEN(R$(2))<6 THEN
8550      R$(2)=R$(2)&"0"
8560    END IF
8570  NEXT I
8580  IF LEN(R$(2))>6 THEN R$(2)=R$(2)[1,6]
8590  IF Cont_test_num<>0 THEN Cont_end
8600 Cont_3:OUTPUT @Gpio_12 USING "#,W";2^1+2^11
8610  OUTPUT @Keithley;"R0P0Z0T5G1X"
8620  PRINT CHR$(12)
8630  CONTROL CRT;20,15
8640  PRINT "TESTING SERIAL NUMBER "&Serial_num$
8650  CONTROL CRT;22,17
8660  PRINT "CONTINUITY TEST NUMBER 3"
8670  WAIT 1
8680  ENTER @Keithley;R$(3)
8690  R(3)=VAL(R$(3))
8700  R(3)=INT(R(3)*10^4)/10^4
8710  R$(3)=VAL$(R(3))
8720  FOR I=1 TO 5
8730    IF LEN(R$(3))<5 THEN
8740      R$(3)=R$(3)&"0"
8750    END IF
8760  NEXT I
8770  IF LEN(R$(3))>5 THEN R$(3)=R$(3)[1,5]
8780 Cont_end: !
8790  OUTPUT @Gpio_12 USING "#,W";0
8800  OUTPUT @Gpio_14 USING "#,W";0
8810  PRINT CHR$(12)
8820  !
8830  GOSUB Safe_tester
8840  PRINT CHR$(12)
8850  PRINT TAB(11);"TEST";TAB(29);"LIMITS(OHMS)";TAB(45);"VALUE(ohms)";TAB(60);
"STATUS"
8860  PRINT
8870  IF Cont_test_num<>0 THEN
8880    Cont_stat$(Cont_test_num)="C"
8890    IF R_up(Cont_test_num)<R(Cont_test_num) THEN Cont_stat$(Cont_test_num)="
H"
8900    IF R_low(Cont_test_num)>R(Cont_test_num) THEN Cont_stat$(Cont_test_num)=
"L"
8910    IF Cont_stat$(Cont_test_num)="C" THEN
8920      Cont_ar$(Cont_test_num)="ACCEPT"
8930    ELSE
8940      Cont_ar$(Cont_test_num)="REJECT"
```

```
8950    END IF
8960    C=Cont_test_num
8970    PRINT TAB(5);"CONT. TEST NUMBER ";Cont_test_num$;TAB(27);R_up$(C);TAB(35
);R_low$(C);TAB(47);R$(C);TAB(60);Cont_ar$(C)
8980  ELSE
8990    FOR I=1 TO 3
9000      I$=VAL$(I)
9010      Cont_stat$(I)="C"
9020      IF R_up(I)<R(I) THEN Cont_stat$(I)="H"
9030      IF R_low(I)>R(I) THEN Cont_stat$(I)="L"
9040      IF Cont_stat$(I)="C" THEN
9050        Cont_ar$(I)="ACCEPT"
9060      ELSE
9070        Cont_ar$(I)="REJECT"
9080      END IF
9090      PRINT TAB(5);"CONT TEST NUMBER ";I$;TAB(27);R_up$(I);TAB(35);R_low$(I)
;TAB(47);R$(I);TAB(60);Cont_ar$(I)
9100    NEXT I
9110  END IF
9120  IF Serial_num$="SIMULATOR" THEN RETURN
9130 Test_another_co:! ALLOWS RETEST OR NEXT MANUAL TEST
9140  ON KBD GOTO Test_co_again
9150  DISP "DO YOU WANT TO RETEST CONTINUITY ?(Y/N)"
9160 Wait9:GOTO Wait9
9170 Test_co_again:A$=KBD$
9180  IF A$="Y" THEN Cont_sel2
9190  IF A$<>"N" THEN Test_another_co
9200  GOSUB Safe_tester
9210  GOTO Test_choice
9220  !
9230  !*************************************************************************
9240  !*************************************************************************
9250 Data_disk: ! SETS UP DATA DISK IN DRIVE 1
9260  IF Pro_test THEN RETURN
9270  ON ERROR GOTO Dk_eras
9280  GOSUB Init_disk
9290  IF NOT (New_lot) THEN Old_data_disk
9300 New_data_disk: !
9310  CREATE BDAT "DATA:,700,1",400,350
9320  CREATE BDAT "STAT:,700,1",20,80
9330  !
9340  !*************************************************************************
9350 Old_data_disk: !
9360  ASSIGN @Data TO "DATA:,700,1";FORMAT ON
9370  ASSIGN @Stat TO "STAT:,700,1";FORMAT ON
9380  !
9390  GOSUB Get_old_header
9400  RETURN
9410  !
9420  !*************************************************************************
9430  !*************************************************************************
9440 Init_disk:!
```

```
9450 !  THIS PROGRAM INITIALIZES A BRAND NEW DISK OR ERASES AN EXISTING DISK
9460 !
9470  PRINT CHR$(12)
9480  GCLEAR
9490  DISP "CHECKING DISK IN DRIVE 1"
9500  ON ERROR GOTO Dk_eras
9510  PRINT CHR$(12)
9520  CAT ":HP,700,1";COUNT Entries
9530  DISP ""
9540  IF NOT (New_lot) THEN RETURN
9550  IF New_lot THEN Loop
9560  GOTO Key_press_1
9570 Loop:Pos=1
9580  Output$=""
9590  INPUT "DO YOU WANT TO ERASE THIS DISK? (type 'YES' then depress ENTER butt
on to erase)",A$
9600  IF A$<>"YES" THEN Key_press_1
9610  OFF KBD
9620  PRINT CHR$(12)
9630  DISP "ERASING DISK IN DRIVE 1"
9640  GOTO Eras
9650 Key_press_1:PRINT CHR$(12)
9660  DISP "INSERT DATA DISK INTO DRIVE 1"
9670  GOSUB Wait_for_key
9680  GOTO Init_disk
9690 Dk_eras:OFF ERROR
9700  DISP "CHECKING DISK"
9710  PRINT CHR$(12)
9720  IF ERRN=85 THEN
9730    GOTO Eras
9740  ELSE
9750    IF ERRN=80 THEN
9760      IF NOT (New_lot) THEN
9770        Disp$="Insert data disk into drive 1"
9780        GOSUB Wait_for_key
9790        RETURN
9800      END IF
9810      GOSUB Wait_for_key
9820      GOTO Init_disk
9830    END IF
9840    PRINT CHR$(12)
9850    Disp$="DISK ERROR--PLEASE CHECK DISK"
9860    GOSUB Wait_for_key
9870    PRINT CHR$(12)
9880    DISP ""
9890    GOTO Init_disk
9900  END IF
9910 Eras:   !*********
9920  ON ERROR GOTO Dk_eras
9930  DISP ""
9940  PRINT CHR$(12)
9950  CONTROL 1;1,15
```

```
9960  Dr$=VAL$(Dr)
9970  PRINT "DISK IN DRIVE 1 IS BEING INITIALIZED"
9980  INITIALIZE ":HP9122,700,1",0,4
9990  CONTROL 1;1,15
10000 PRINT USING "50X"
10010 OFF ERROR
10020 RETURN
10030 !
10040 Wait_for_key:!*******
10050 ON KBD GOTO Exit1
10060 CONTROL 1;1,18
10070 PRINT Disp$
10080 Wait_loop1:DISP "Press ANY KEY when ready."
10090 WAIT .25
10100 DISP
10110 WAIT .25
10120 GOTO Wait_loop1
10130 Exit1:OFF KBD
10140 DISP
10150 CONTROL 1;1,18
10160 PRINT USING "50X"
10170 Disp$=""
10180 OFF ERROR
10190 RETURN
10200 !
10210 !****************************************************************************
10220 !****************************************************************************
10230 Print_out:! PRINTS OUT TESTED DATA IN TABLE FORM TO CRT AND PRINTER
10240 IF Printer=0 THEN
10250   OUTPUT KBD USING "#,K";" |"
10260   PRINT CHR$(12)
10270 END IF
10280 PRINT TAB(1);"SERIAL NUMBER: "&Serial_num$;TAB(38);"DATE MANUFACTURE CODE:
 ";Date_manuf$
10290 PRINT
10300 OUTPUT 701 USING "#,K";""          !SET PRINTER FOR UNDERLINING
10310 PRINT CHR$(132);TAB(11);"TEST";TAB(29);"Limits(Ohms)";TAB(45);"VALUE(ohms)
";TAB(60);"STATUS";CHR$(128)
10320 OUTPUT 701 USING "#,K";""          !TURN OFF UNDERLINING
10330 !
10340 FOR I=1 TO 3
10350   I$=VAL$(I)
10360   PRINT TAB(5);"CONT TEST NUMBER ";I$;TAB(27);R_up$(I);TAB(35);R_low$(I);T
AB(45);R$(I);TAB(60);Cont_ar$(I)
10370 NEXT I
10380 FOR I=1 TO 9
10390   I$=VAL$(I)
10400   PRINT TAB(5);"I.R. TEST NUMBER "&I$;TAB(60);Ir_stat$(I)
10410 NEXT I
10420 IF Printer=1 THEN RETURN
10430          !
10440 Store_q:          ! VERIFYS ALL TESTS PERFORMED
```

```
10450 ON KBD GOTO Store_ans
10460 DISP "ARE YOU READY TO STORE THIS DATA?(Y/N)"
10470 Wait10:GOTO Wait10
10480 Store_ans:A$=KBD$
10490 IF Serial_num$<>"SIMULATOR" THEN
10500   IF A$="N" THEN Test_choice
10510 END IF
10520 IF A$<>"Y" THEN Store_q
10530 FOR I=1 TO 3
10540   IF R$(I)="" THEN Not_all_tested
10550 NEXT I
10560 FOR I=1 TO 9
10570   IF Ir_stat$(I)="" THEN Not_all_tested
10580 NEXT I
10590 PRINTER IS 701
10600 Printer=1
10610 OUTPUT 701 USING "#,K";CHR$(27)&"(s5hu6B"      ! PRINT EXPANDED CHARACTERS
10620 Pr_to_printer:!PRINTS DATA ON PRINTER
10630 PRINT
10640 PRINT
10650 PRINT TAB(16);"ANTS"
10660 PRINT TAB(1);"CONTINUITY/INSULATION RESISTANCE TESTER"
10670 PRINT
10680 OUTPUT 701 USING "#,K";CHR$(27)&"(s10huB"     !PRINT NORMAL SIZE
10690 PRINT "PROCUCT SPECIFICATION: ";Test_spec_num$;TAB(38);"SOFTWARE: ";Am$
10700 PRINT "PART NUMBER: ";Part_num$;TAB(38);"DATE: ";Date$
10710 PRINT "LOT NUMBER: ";Lot_num$;TAB(38);"TEST CODE: ";Test_code$
10720 PRINT TAB(38);"DATA SOURCE CODE: ";Data_source$
10730 PRINT
10740 IF Printer=0 THEN RETURN
10750 GOSUB Print_out
10760 PRINTER IS CRT
10770 Printer=0
10780 !
10790 !**************************************************************************
10800 !**************************************************************************
10810 Store_on_disk:!STORES DATA ON DATA DISK ON DRIVE 1
10820             !
10830 IF Serial_num$="SIMULATOR" THEN RETURN
10840 IF Pro_test THEN End_store
10850 Record=Record+1
10860 Record$=VAL$(Record)
10870 OUTPUT @Stat,1;Record$
10880             !
10890 Str1$="H, ID TRANSD, TD "&Date$&", DS "&Data_source$&", PN "&Part_num$&",
MF MAD, TC "&Test_code$&", PS PS-"&Test_spec_num$&", LN "&Lot_num$
10900 Str2$=", TE ANTS, $T, ID TRANSD, SN "&Serial_num$&", DM "&Date_manuf$&", C
R1 "&R$(1)&Cont_stat$(1)&", CR2 "&R$(2)&Cont_stat$(2)
10910 Str3$=", CR3 "&R$(3)&Cont_stat$(3)
10920 Str4$=", IR1 "&Ir_stat$(1)[1,1]&", IR2 "&Ir_stat$(2)[1,1]&", IR3 "&Ir_stat
$(3)[1,1]&", IR4 "&Ir_stat$(4)[1,1]&", IR5 "&Ir_stat$(5)[1,1]
10930 Str5$=", IR6 "&Ir_stat$(6)[1,1]&", IR7 "&Ir_stat$(7)[1,1]&", IR8 "&Ir_stat
```

```
$(8)[1,1]&", IR9 "&Ir_stat$(9)[1,1]&", $"
10940 Str$=Str1$&Str2$&Str3$&Str4$&Str5$
10950 !
10960 OUTPUT @Data,Record;Str$
10970 !
10980 End_store:!
10990 PRINT CHR$(12)
11000 DISP ""
11010 Next_sn_q:ON KBD GOTO Next_sn_a
11020 DISP "DO YOU WANT TO TEST NEXT SERIAL NUMBER?(Y/N)"
11030 Wait14:GOTO Wait14
11040 Next_sn_a:A$=KBD$
11050 IF A$="N" THEN
11060   GOSUB Close_file
11070   GOTO Start
11080 END IF
11090 IF A$<>"Y" THEN Next_sn_q
11100 GOSUB Init_test_var
11110 GOSUB E_sn
11120 GOTO Test_choice
11130 !
11140 !*******************************************************************************
11150 Not_all_tested:! NOT ALL TESTS PERFORMED
11160 PRINT
11170 Disp$="NOT ALL TESTS HAVE BEEN PERFORMED"
11180 GOSUB Wait_for_key
11190 GOTO Test_choice
11200 !
11210 !*******************************************************************************
11220 !*******************************************************************************
11230 Print_data:! THIS ROUTINE PRINTS OUT GEISHA DATA FROM THE DATA DISK
11240           !
11250 ON ERROR GOTO Dk_eras2
11260 GOSUB Old_data_disk
11270 DISP TAB(25);"***** PRINTING DATA *****"
11280 PRINTER IS 701
11290 ENTER @Stat,1;Record$
11300 Record=VAL(Record$)
11310 IF Record=1 THEN
11320   ENTER @Data,1;Data$
11330   PRINT Data$
11340 ELSE
11350   FOR I=1 TO Record
11360     ENTER @Data,I;Data$
11370     PRINT Data$
11380   NEXT I
11390 END IF
11400 PRINTER IS CRT
11410 GOSUB Close_file
11420 GOTO Main_menu
11430 !
11440 Dk_eras2:OFF ERROR
```

```
11450 IF ERRN=80 THEN DISP "INSERT DATA DISK INTO DRIVE 1"
11460 WAIT 2
11470 GOTO Print_data
11480         !
11490 !
11500 !***************************************************************************
11510 !***************************************************************************
11520 Edit_data:! THIS ROUTINE EDITS THE DATA DISK
11530         !
11540 In:INPUT "ENTER RECORD NUMBER TO EDITED ",Re
11550 Re$=VAL$(Re)
11560 ON ERROR GOTO No_record
11570 GOTO Edit_rec
11580 No_record:              !
11590 OFF ERROR
11600 DISP "RECORD "&Re$&" DOES NOT EXIST"
11610 WAIT 2
11620 GOSUB Close_file
11630 GOTO Start
11640 Edit_rec:                    !
11650 Edit_menu:!
11660 GOSUB Old_data_disk
11670 Name$="****** EDIT MENU ******"
11680 DATA " DATA SOURCE CODE "
11690 DATA "DS"
11700 DATA " PART NUMBER "
11710 DATA "PN"
11720 DATA " TEST SPECIFICATION NUMBER "
11730 DATA "PS"
11740 DATA " LOT NUMBER "
11750 DATA "LN"
11760 DATA " DATE "
11770 DATA "DATE"
11780 DATA " SERIAL NUMBER "
11790 DATA "SN"
11800 DATA " DATE MANUFACTURE CODE "
11810 DATA "DM"
11820 RESTORE 11670
11830 Menue_items=7
11840 GOSUB Menue
11850 OUTPUT KBD USING "#,K";" |"
11860 ON KEY 1 LABEL "UP" GOSUB Move_down
11870 ON KEY 2 LABEL "DOWN" GOSUB Move_up
11880 ON KEY 3 GOTO Loop_d
11890 ON KEY 4 LABEL "SELECT" GOTO Selectc
11900 ON KEY 5 LABEL "  END   EDITING" GOTO End_edit
11910 ON KEY 6 GOTO Loop_d
11920 ON KEY 7 GOTO Loop_d
11930 ON KEY 8 LABEL "DELETE" GOTO Delete
11940 ON KBD GOTO Loop_d
11950 Loop_d:GOTO Loop_d
11960 Selectc:OFF KEY
```

```
11970 OUTPUT KBD USING "#,K";" |"
11980 OFF KBD
11990 PRINT CHR$(12)
12000 Edit_select$=Select$
12010 IF Select$="DS" THEN GOSUB Ds_menu
12020 IF Select$="PN" THEN GOSUB E_pn
12030 IF Select$="PS" THEN GOSUB E_ps
12040 IF Select$="LN" THEN GOSUB E_ln
12050 IF Select$="DATE" THEN GOSUB E_date
12060 IF Select$="SN" THEN GOSUB E_sn
12070 IF Select$="DM" THEN GOSUB E_dm
12080 P_output:CONTROL CRT,1;15
12090 PRINT Output$
12100 Item$="IS THIS CHANGE CORRECT?"
12110 GOSUB Question
12120 IF Answer$="N" THEN 12010
12130 IF Answer$<>"Y" THEN P_output
12140 Store:DISP CHR$(12)
12150 Te_pos=POS(Str_in$,"TE")
12160 Te_pos=Te_pos-2
12170 !
12180 IF Edit_select$="SN" THEN Change_individ
12190 IF Edit_select$="DM" THEN Change_individ
12200 IF Edit_select$="DATE" THEN Change_individ
12210 Change_individ:Str_out$=Str_in$[1,Date_pos+2]&Date$&Str_in$[Date_pos+11,Sn
_pos+2]&Serial_num$&Str_in$[Sn_pos+9,Dm_pos+2]&Date_manuf$&Str_in$[Dm_pos+6]
12220 OUTPUT @Data,Re;Str_out$
12230         !
12240 IF Edit_select$="DS" THEN Change_common
12250 IF Edit_select$="PN" THEN Change_common
12260 IF Edit_select$="PS" THEN Change_common
12270 IF Edit_select$="LN" THEN Change_common
12280 Change_common:FOR I=1 TO Record
12290   ENTER @Data,I;Str_in$
12300   IF Str_in$="" THEN
12310     I=I+1
12320     GOTO 12290
12330   END IF
12340   Str_out$=Str_in$[1,30]&Data_source$&", PN "&Part_num$&", MF MAD, TC "&Te
st_code$&", PS PS-"&Test_spec_num$&", LN "&Lot_num$&Str_in$[Te_pos]
12350   OUTPUT @Data,I;Str_out$
12360 NEXT I
12370 GOSUB Close_file
12380 GOTO Edit_menu
12390 End_edit:!
12400 GOSUB Close_file
12410 Re=0
12420 OUTPUT KBD USING "#,K";" |"
12430 GOTO Start
12440 !
12450 Delete:! DELETES A RECORD FROM THE DATA DISK
12460 OUTPUT KBD USING "#,K";" |"
```

```
12470 PRINT CHR$(12)
12480 Re$=VAL$(Re)
12490 Item$="DELETE RECORD "&Re$&" ?"
12500 GOSUB Question
12510 IF Answer$="N" THEN Edit_menu
12520 IF Answer$<>"Y" THEN Delete
12530 Str_out$=""
12540 DISP "DELETING RECORD "&Re$
12550 OUTPUT @Data,Re;Str_out$
12560 IF Re$=Record$ THEN
12570   Record=Record-1
12580   Record$=VAL$(Record)
12590   OUTPUT @Stat,1;Record$
12600 END IF
12610 GOSUB Close_file
12620 WAIT 3
12630 GOTO Start
12640 Close_file:!
12650 ASSIGN @Stat TO *
12660 ASSIGN @Data TO *
12670 RETURN
12680 !
12690 !****************************************************************************
12700 !****************************************************************************
12710 Simulator_ck:   !VERIFYS SIMULATORS
12720 PRINT CHR$(12)
12730 Continuity_sim:     !
12740 !
12750 PRINT CHR$(12)
12760 DISP ""
12770 MOVE 22,70
12780 CSIZE 8,.4
12790 LABEL "VERIFYING CONTINUITY SIMULATOR"
12800 PRINT TABXY(8,10);"     PLACE CONTINUITY SIMULATOR IN CONTINUITY SIDE OF F
IXTURE"
12810 GOSUB Wait_for_key
12820 Serial_num$="SIMULATOR"
12830 GOSUB Test_cont
12840 !
12850 ON KBD GOTO Cont_sim_again
12860 DISP "DO YOU NEED TO RUN THE CONTINUITY SIMULATOR AGAIN(Y/N)?"
12870 Wait1:GOTO Wait1
12880 Cont_sim_again:      !
12890 Cs$=KBD$
12900 IF Cs$="Y" THEN Continuity_sim
12910 IF Cs$<>"N" THEN 12850
12920 Ir_sim_ck:              !
12930 PRINT CHR$(12)
12940 PRINT TABXY(8,10);"PLACE INSUL. R. SIMULATOR IN INSUL. R. SIDE OF FIXTURE"
12950 GOSUB Wait_for_key
12960 Ck_test=1
12970 Ck_test$="1"
```

```
12980 Sw_pos$="1"
12990 GOSUB Ck_accept
13000 Ck_test=8
13010 Ck_test$="8"
13020 Sw_pos$="2"
13030 GOSUB Ck_accept
13040 Ck_test=9
13050 Ck_test$="9"
13060 Sw_pos$="3"
13070 GOSUB Ck_accept
13080 Ck_test=2
13090 Ck_test$="2"
13100 Sw_pos$="4"
13110 GOSUB Ck_accept
13120 Ck_test=3
13130 Ck_test$="3"
13140 Sw_pos$="5"
13150 GOSUB Ck_accept
13160 Ck_test=5
13170 Ck_test$="5"
13180 Sw_pos$="6"
13190 GOSUB Ck_accept
13200 Ck_test=6
13210 Ck_test$="6"
13220 Sw_pos$="7"
13230 GOSUB Ck_accept
13240 Ck_test=4
13250 Ck_test$="4"
13260 Sw_pos$="8"
13270 GOSUB Ck_accept
13280 Ck_test=7
13290 Ck_test$="7"
13300 Sw_pos$="8"
13310 GOSUB Ck_accept
13320 GOTO Print_sim
13330 Ck_accept:                        !
13340 Ir_test_sel=Ck_test
13350 PRINT CHR$(12)
13360 PRINT TABXY(8,10);"VERIFYING INSUL. RES. SIMULATOR TEST "&Ck_test$
13370 Disp1$="ON INSUL .R. SIMULATOR BOX, MOVE SWITCHES TO POSITION "&Sw_pos$&"
AND ACCEPT"
13380 PRINT TABXY(8,11);Disp1$
13390 GOSUB Wait_for_key
13400 PRINT CHR$(12)
13410 GOSUB Test_ir
13420 !
13430 ENTER @Gpio_14 USING "#,W";Monitor
13440 IF NOT BIT(Monitor,Comparator_mon) THEN Sim_fail
13450 PRINT CHR$(12)
13460 GOSUB Safe_tester
13470 !
13480 Ck_reject:Sim_reject=1
```

```
13490 PRINT CHR$(12)
13500 PRINT TABXY(8,10);"VERIFYING INSUL. RES. SIMULATOR TEST "&Ck_test$
13510 Disp1$="ON INSUL. R. SIMULATOR BOX, MOVE SWITCHES TO POSITION "&Sw_pos$&"
AND REJECT"
13520 PRINT TABXY(8,11);Disp1$
13530 GOSUB Wait_for_key
13540 GOSUB Test_ir
13550 OUTPUT @Gpio_14 USING "#,W";0
13560 OUTPUT @Gpio_12 USING "#,W";0
13570 !
13580 Sim_reject=0
13590 ENTER @Gpio_14 USING "#,W";Monitor  !CHECKS STATE OF COMPARATOR
13600 IF BIT(Monitor,Comparator_mon) THEN Sim_fail
13610 PRINT CHR$(12)
13620 GOSUB Safe_tester
13630 ENTER @Gpio_14 USING "#,W";Monitor
13640 IF BIT(Monitor,Comparator_mon)=0 THEN
13650   CONTROL CRT;20,15
13660   PRINT " ***** RESET I.R. BUTTON ***** "
13670   WAIT 3
13680   GOTO 13630
13690 END IF
13700 RETURN
13710 Print_sim:!
13720 FOR I=1 TO 9
13730   Ir_stat$(I)="PASSED"
13740 NEXT I
13750 GOSUB Print_out
13760 Ir_test_sel=0
13770 GOTO Main_menu
13780  !
13790 Sim_fail:!
13800 PRINT CHR$(12)
13810 CONTROL CRT;25,15
13820 PRINT "***** SIMULATOR FAILURE *****"
13830 BEEP 500,.5
13840 WAIT .5
13850 BEEP 500,.5
13860 PRINT CHR$(12)
13870 GOSUB Safe_tester
13880 GOSUB Reset_ir
13890 PRINT CHR$(12)
13900 MOVE 22,70
13910 PRINT TABXY(8,10);"SIMULATOR FAILURE.  CHECK SWITCH POSITIONS ON SIMULATOR
 BOX"
13920 PRINT TABXY(8,11);Disp1$
13930 ON KBD GOTO Ck_again
13940 DISP "DO YOU NEED TO CHECK THIS POSITION AGAIN (Y/N)?"
13950 Wait6:GOTO Wait6
13960 Ck_again:Ck$=KBD$
13970 IF Ck$="Y" THEN
13980   IF Disp1$[61,66]="REJECT" THEN Ck_reject
```

```
13990   GOTO Ck_accept
14000 END IF
14010 IF Ck$<>"N" THEN Wait6
14020 PRINT CHR$(12)
14030 DISP ""
14040 CONTROL CRT;30,12
14050 PRINT "NOTIFY MAINTENANCE"
14060 GOTO End
14070 End:END
```

APPENDIX D

SELECTED CALIBRATION SOFTWARE LISTING

```
200 !**************************************************************************
210 !**************************************************************************
220 Intro_screen:  ! DISPLAYS INTRO SCREEN
230   DISP
240   PRINT CHR$(12)
250   GRAPHICS ON
260   GCLEAR
270   MOVE 22,70
280   CSIZE 30,.5
290   LABEL "ANTS"
300   CSIZE 8,.4
310   MOVE 2,65
320   LABEL "CONTINUITY-INSULATION RESISTANCE TESTER"
330   MOVE 36,55
340   LABEL "CALIBRATION PROGRAM"
350   MOVE 44,45
360   LABEL Am$
370   DISP "Press ANY key when ready"
380   ON KBD GOTO Set_up
390 Loop_a:GOTO Loop_a
400 !**************************************************************************
410 !**************************************************************************
420 Set_up:  !
430   GCLEAR
440   LOCAL LOCKOUT 7
450   ! RELAYS FOR IR TESTING
460   !
470   Volt_to_uut_12(1)=2^0+2^6+2^7+2^10
480   Volt_to_uut_14(1)=2^14+(-2^15)
490   Volt_to_uut_12(2)=2^0+2^6
500   Volt_to_uut_14(2)=2^1+(-2^15)
510   Volt_to_uut_12(3)=2^0+2^6
520   Volt_to_uut_14(3)=2^5+(-2^15)
530   Volt_to_uut_12(4)=2^0+2^6
540   Volt_to_uut_14(4)=2^1+2^5+2^6
550   Volt_to_uut_12(5)=2^0+2^6
560   Volt_to_uut_14(5)=2^0+(-2^15)
570   Volt_to_uut_12(6)=2^0+2^6
580   Volt_to_uut_14(6)=2^4+(-2^15)
590   Volt_to_uut_12(7)=2^0+2^6
600   Volt_to_uut_14(7)=2^0+2^4+2^6
610   Volt_to_uut_12(8)=2^0+2^6
620   Volt_to_uut_14(8)=2^8+(-2^15)
630   Volt_to_uut_12(9)=2^0+2^6
640   Volt_to_uut_14(9)=2^7+(-2^15)
650   Door_mon=0
660   Key_mon=1
670   Comparator_mon=2
680 !**************************************************************************
690 !**************************************************************************
700 Addressing: !
710   ASSIGN @Volt TO 72300
```

```
720   OUTPUT @Volt;"F1R1"
730   ASSIGN @Gpio_12 TO 12
740   ASSIGN @Gpio_14 TO 14
750   ASSIGN @Keithley TO 72500
760 !****************************************************************************
770 !****************************************************************************
780 As_found_val:      !VERIFIES "AS FOUND" VALUES
790   PRINT CHR$(12)
800   Label$="AS FOUND HIGH VOLTAGE P.S."
810   Disp1$="CONNECT STANDARD VOLT METER ACROSS TP101 AND TP102 (TP102 GROUND)"
820   Disp2$="BEHIND CAL. DOOR"
830   Dips3$=""
840   GOSUB Wait_for_key
850 As_found_b_volt:  !
860   GOSUB Door_open
870   GOSUB Arm_tester
880   OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(2)
890   OUTPUT @Gpio_14 USING "#,W";Volt_to_uut_14(2)
900   OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(2)+2^13
910   PRINT CHR$(12)
920   Label$="   AS FOUND B VOLT P.S."
930   Disp1$="(CP17) RECORD READING ON STD. METER 'AS FOUND'"
940   Disp2$="        SHOULD BE (B VOLT) +/- (D VOLT)"
950   Disp3$="(CP19) VERIFY HP3478A DVM IS (CP17)/1000 +/- .000D V"
960   GOSUB Wait_for_key
970   OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(2)
980   WAIT 3
990   OUTPUT @Gpio_14 USING "#,W";0
1000  WAIT 4
1010  OUTPUT @Gpio_12 USING "#,W";0
1020  GOSUB Reset_ir
1030 As_found_a_v:!
1040  GOSUB Door_open
1050  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(1)
1060  OUTPUT @Gpio_14 USING "#,W";Volt_to_uut_14(1)
1070  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(1)+2^13+2^9
1080  PRINT CHR$(12)
1090  Label$="  AS FOUND A VOLT P.S."
1100  Disp1$="(CP18) RECORD THIS READING 'AS FOUND'"
1110  Disp2$="        SHOULD BE (A VOLT)+/-(C VOLT)"
1120  Disp3$="(CP20) VERIFY HP3478A DVM IS (CP18)/1000 +/- .000C V"
1130  GOSUB Wait_for_key
1140  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(1)
1150  WAIT 3
1160  OUTPUT @Gpio_14 USING "#,W";0
1170  WAIT 4
1180  OUTPUT @Gpio_12 USING "#,W";0
1190  GOSUB Safe_tester
1200  GOSUB Reset_ir
1210  !
1220  Label$=""
1230  Disp1$="                REMOVE STANDARD VOLTMETER"
```

```
1240  Disp2$=""
1250  Disp3$=""
1260  GOSUB Wait_for_key
1270 !**************************************************************************
1280 !**************************************************************************
1290 Calibrate: !  SOFTWARE AIDED CALIBRATION
1300 Cal_com_1:           ! TRIP AT bb.b MICROAMPS
1310  Label$="CALIBRATING THE COMPARATORS"
1320  Disp1$="PLACE I.R. SIMULATOR IN I.R. FIXTURE AND CLOSE BARRICADE DOOR."
1330  Disp2$="ON SIMULATOR BOX, POSITION SWITCHES ON POS. 2 AND REJECT"
1340  Disp3$="REMOVE SHORTING PLUG(TP104) IN CAL DOOR AND PLUG IN STANDARD
          AMMETER(B GROUND)"
1350  GOSUB Wait_for_key
1360  GOSUB Arm_tester
1370  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(8)
1380  OUTPUT @Gpio_14 USING "#,W";Volt_to_uut_14(8)
1390  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(8)+2^13
1400  PRINT CHR$(12)
1410  Label$="CALIBRATING COMPARATOR 1"
1420  Disp1$="ADJUST COMPARATOR 1 UNTIL REJECT LIGHT STAYS OFF"
1430  Disp2$="ADJUST B VOLT P.S. UNTIL STD. AMMETER READS (bb.b-2) +/- 1 MICROAM
PS"
1440  Disp3$="ADJUST COMPARATOR 1 JUST UNTIL IT TRIPS(REJECT LAMP WILL LIGHT)"
1450  GOSUB Wait_for_key
1460  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(2)
1470  WAIT 3
1480  OUTPUT @Gpio_14 USING "#,W";0
1490  OUTPUT @Gpio_12 USING "#,W";0
1500  GOSUB Reset_ir
1510  GOSUB Safe_tester
1520  PRINT CHR$(12)
1530  Label$=""
1540  Disp1$="POSITION SWITCH ON SIMULATOR BOX TO POS. 1 AND REJECT"
1550  Disp2$=""
1560  Disp3$=""
1570  GOSUB Wait_for_key
1580 !
1590 Cal_com_2:! TRIP AT aa.a MICROAMPS
1600  GOSUB Door_open
1610  GOSUB Arm_tester
1620  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(1)
1630  OUTPUT @Gpio_14 USING "#,W";Volt_to_uut_14(1)
1640  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(1)+2^13+2^9
1650  PRINT CHR$(12)
1660  Label$="CALIBRATING COMPARATOR 2"
1670  Disp1$="ADJUST COMPARATOR 2 UNTIL REJECT LIGHT STAYS OFF"
1680  Disp2$="ADJUST A VOLT P.S. UNTIL STD. AMMETER READS (aa.a-3)+/-2 MICROAMPS
"
1690  Disp3$="ADJUST COMPARATOR 2 JUST UNTIL IT TRIPS(REJECT LAMP WILL LIGHT)"
1700  GOSUB Wait_for_key
1710  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(1)
1720  WAIT 3
```

```
1730  OUTPUT @Gpio_14 USING "#,W";0
1740  OUTPUT @Gpio_12 USING "#,W";0
1750  GOSUB Safe_tester
1760  GOSUB Reset_ir
1770 !****************************************************************************
1780 Cal_ps:          !
1790  PRINT CHR$(12)
1800  Label$="CALIBRATING B VOLT P.S."
1810  Disp1$="POSITION SWITCH ON SIMULATOR BOX TO ACCEPT"
1820  Disp2$="REPLACE SHORTING PLUG IN TP104"
1830  Disp3$="CONNECT STANDARD VOLT METER ACROSS TP101 AND TP102 (TP102 GROUND)"
1840  GOSUB Wait_for_key
1850 Cal_b_v_ps:GOSUB Door_open
1860  GOSUB Arm_tester
1870  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(2)
1880  OUTPUT @Gpio_14 USING "#,W";Volt_to_uut_14(2)
1890  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(2)+2^13
1900  PRINT CHR$(12)
1910  Disp1$="(CP17) ADJUST B VOLT P.S. UNTIL STD. VOLTMETER READS (B VOLT) +/-
(D/10)V"
1920  Disp2$="(CP19) VERIFY HP3478A DVM IS (CP17)/1000 +/-.0000D V"
1930  Disp3$="       IF NOT, NOTIFY MAINTENANCE"
1940  GOSUB Wait_for_key
1950  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(2)
1960  WAIT 3
1970  OUTPUT @Gpio_14 USING "#,W";0
1980  WAIT 4
1990  OUTPUT @Gpio_12 USING "#,W";0
2000  GOSUB Reset_ir
2010 Cal_a_v_ps:GOSUB Door_open
2020  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(1)
2030  OUTPUT @Gpio_14 USING "#,W";Volt_to_uut_14(1)
2040  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(1)+2^13+2^9
2050  PRINT CHR$(12)
2060  Label$="CALIBRATING A VOLT P.S."
2070  Disp1$="(CP18) ADJUST A VOLT P.S. UNTIL STD. VOLTMETER READS (A VOLT) +/-
(C/10)V"
2080  Disp2$="(CP20) VERIFY HP3478A DVM IS (CP18)/1000 +/- .0000C V"
2090  Disp3$="       IF NOT, NOTIFY MAINTENANCE"
2100  GOSUB Wait_for_key
2110  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(1)
2120  WAIT 3
2130  OUTPUT @Gpio_14 USING "#,W";0
2140  WAIT 4
2150  OUTPUT @Gpio_12 USING "#,W";0
2160  GOSUB Safe_tester
2170  GOSUB Reset_ir
2180 !****************************************************************************
2190 Verify_timing: !
2200  Label$="VERIFYING REQUIRED TIME INTERVAL"
2210  Disp1$=" CONNECT THE STORAGE SCOPE ACROSS TP102 AND TP103 (TP102 GROUND)"
2220  Disp2$=" SWITCH INSULATION RESISTANCE BOX TO ACCEPT AND POS. 2"
```

```
2230  Disp3$=" CHECK TIMING OF B/1000 VOLT SIGNAL"
2240  GOSUB Wait_for_key
2250  GOSUB Door_open
2260  GOSUB Arm_tester
2270  PRINT CHR$(12)
2280  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(2)
2290  OUTPUT @Gpio_14 USING "#,W";Volt_to_uut_14(2)
2300  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(2)+2^13
2310  !****** MOST OF THE FOLLOWING LINES ARE USED TO IMITATE THE TIMING DURING
              TESTING *****
2320  WAIT 1
2330  ENTER @Volt;Volt$
2340  Volt=VAL(Volt$)
2350  ENTER @Gpio_14 USING "#,W";Monitor
2360  IF Imitate_t=Imitate_t THEN
2370    IF NOT BIT(Monitor,Comparator_mon) THEN 2380
2380  END IF
2390  IF 0>1 THEN Volt_out_range
2400  IF 1<0 THEN Volt_out_range
2410  FOR I=1 TO 4
2420    WAIT 1
2430    ENTER @Gpio_14 USING "#,W";Monitor
2440    IF NOT BIT(Monitor,Comparator_mon) THEN 2450
2450  NEXT I
2460  WAIT 1
2470  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(2)
2480  WAIT 3
2490  OUTPUT @Gpio_14 USING "#,W";0
2500  WAIT 4
2510  OUTPUT @Gpio_12 USING "#,W";0
2520  GOSUB Safe_tester
2530  GOSUB Reset_ir
2540  ON KBD GOTO Time_again
2550  DISP "DO YOU NEED TO MAKE THIS MEASUREMENT AGAIN (Y/N)?"
2560 Wait2:GOTO Wait2
2570 Time_again:        !
2580  T$=KBD$
2590  IF T$="Y" THEN Verify_timing
2600  IF T$<>"N" THEN 2540
2610  Label$=""
2620  Disp1$="(CP21) RECORD THIS READING (FROM THE 50% POINTS) AS THE TIME INTER
VAL"
2630  Disp2$="IT SHALL BE BETWEEN 2 AND 9 SECONDS"
2640  Disp3$="REMOVE CALIBRATION EQUIPMENT AND CLOSE CAL. DOOR"
2650  GOSUB Wait_for_key
2660 !**************************************************************************
2670 Simulator_ck:    !VERIFYS SIMULATORS
2680  PRINT CHR$(12)
2690 Continuity_sim:     !
2700  !
2710  R_up(1)=1.0002  ! REPRESENTED BY x.xxxx
2720  R_low(1)=1.0001 ! REPRESENTED BY y.yyyy
```

```
2730  R_up(2)=1.0002  ! REPRESENTED BY x.xxxx
2740  R_low(2)=1.0001 ! REPRESENTED BY y.yyyy
2750  R_up(3)=1.0002  ! REPRESENTED BY x.xxxx
2760  R_low(3)=1.0001 ! REPRESENTED BY y.yyyy
2770  PRINT CHR$(12)
2780  DISP ""
2790  Label$="VERIFYING CONTINUITY SIMULATOR"
2800  Disp1$="       PLACE CONTINUITY SIMULATOR IN CONTINUITY SIDE OF FIXTURE"
2810  Disp2$=""
2820  Disp3$=""
2830  GOSUB Wait_for_key
2840  GOSUB Door_open
2850  GOSUB Arm_tester
2860  PRINT CHR$(12)
2870  !
2880 Cont_1:OUTPUT @Gpio_12 USING "#,W";2^1+2^3
2890  OUTPUT @Keithley;"R0P0Z0T5G1X"
2900  PRINT CHR$(12)
2910  CONTROL CRT;25,15
2920  PRINT "TESTING SIMULATOR"
2930  CONTROL CRT;22,17
2940  PRINT "CONTINUITY TEST NUMBER 1"
2950  WAIT 1
2960  ENTER @Keithley;R$(1)
2970  R(1)=VAL(R$(1))
2980  R(1)=INT(R(1)*10^4)/10^4   !IMPLIED ZERO
2990  R$(1)=VAL$(R(1))
3000  IF Cont_test_num<>0 THEN Cont_end
3010 Cont_2:OUTPUT @Gpio_12 USING "#,W";2^1+2^2
3020  OUTPUT @Keithley;"R0P0Z0T5G1X"
3030  PRINT CHR$(12)
3040  CONTROL CRT;25,15
3050  PRINT "TESTING SIMULATOR"
3060  CONTROL CRT;22,17
3070  PRINT "CONTINUITY TEST NUMBER 2"
3080  WAIT 1
3090  ENTER @Keithley;R$(2)
3100  R(2)=VAL(R$(2))
3110  R(2)=INT(R(2)*10^4)/10^4   !IMPLIED ZERO
3120  R$(2)=VAL$(R(2))
3130  IF Cont_test_num<>0 THEN Cont_end
3140 Cont_3:OUTPUT @Gpio_12 USING "#,W";2^1+2^11
3150  OUTPUT @Keithley;"R0P0Z0T5G1X"
3160  PRINT CHR$(12)
3170  CONTROL CRT;25,15
3180  PRINT "TESTING SIMULATOR"
3190  CONTROL CRT;22,17
3200  PRINT "CONTINUITY TEST NUMBER 3"
3210  WAIT 1
3220  ENTER @Keithley;R$(3)
3230  R(3)=VAL(R$(3))
3240  R(3)=INT(R(3)*10^4)/10^4   !IMPLIED ZERO
```

```
3250  R$(3)=VAL$(R(3))
3260 Cont_end: !
3270  OUTPUT @Gpio_12 USING "#,W";0
3280  OUTPUT @Gpio_14 USING "#,W";0
3290  PRINT CHR$(12)
3300  !
3310  GOSUB Safe_tester
3320  PRINT CHR$(12)
3330  !
3340  PRINT TAB(11);"TEST";TAB(29);"LIMITS(OHMS)";TAB(45);"VALUE(ohms)";TAB(60);
"STATUS"
3350  PRINT
3360  FOR I=1 TO 3
3370    I$=VAL$(I)
3380    Cont_ar$(I)="ACCEPT"
3390    IF R_up(I)<R(I) THEN Cont_ar$(I)="REJECT"
3400    IF R_low(I)>R(I) THEN Cont_ar$(I)="REJECT"
3410    PRINT TAB(5);"CONT TEST NUMBER ";I$;TAB(27);R_up(I);TAB(35);R_low(I);TAB
(47);R$(I);TAB(60);Cont_ar$(I)
3420  NEXT I
3430  !
3440  ON KBD GOTO Cont_sim_again
3450  DISP "DO YOU NEED TO RUN THE CONTINUITY SIMULATOR AGAIN(Y/N)?"
3460 Wait1:GOTO Wait1
3470 Cont_sim_again:        !
3480  Cs$=KBD$
3490  IF Cs$="Y" THEN Continuity_sim
3500  IF Cs$<>"N" THEN 3440
3510 Ir_sim_ck:                  !
3520  Ck_test=1
3530  Ck_test$="1"
3540  Sw_pos$="1"
3550  GOSUB Ck_accept
3560  Ck_test=8
3570  Ck_test$="8"
3580  Sw_pos$="2"
3590  GOSUB Ck_accept
3600  Ck_test=9
3610  Ck_test$="9"
3620  Sw_pos$="3"
3630  GOSUB Ck_accept
3640  Ck_test=2
3650  Ck_test$="2"
3660  Sw_pos$="4"
3670  GOSUB Ck_accept
3680  Ck_test=3
3690  Ck_test$="3"
3700  Sw_pos$="5"
3710  GOSUB Ck_accept
3720  Ck_test=5
3730  Ck_test$="5"
3740  Sw_pos$="6"
```

```
3750  GOSUB Ck_accept
3760  Ck_test=6
3770  Ck_test$="6"
3780  Sw_pos$="7"
3790  GOSUB Ck_accept
3800  Ck_test=4
3810  Ck_test$="4"
3820  Sw_pos$="8"
3830  GOSUB Ck_accept
3840  Ck_test=7
3850  Ck_test$="7"
3860  Sw_pos$="8"
3870  GOSUB Ck_accept
3880  GOTO Post_cal
3890 Ck_accept:                    !
3900  PRINT CHR$(12)
3910  Label$=""
3920  Disp1$="VERIFYING INSUL. RES. SIMULATOR TEST "&Ck_test$
3930  Disp2$="ON I.R. SIMULATOR BOX, MOVE SWITCHES TO POSITION "&Sw_pos$&" AND A
CCEPT"
3940  Disp3$=""
3950  GOSUB Wait_for_key
3960  GOSUB Arm_tester
3970  PRINT CHR$(12)
3980  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ck_test)
3990  OUTPUT @Gpio_14 USING "#,W";Volt_to_uut_14(Ck_test)
4000  IF Ck_test=1 THEN
4010    OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ck_test)+2^13+2^9
4020  ELSE
4030    OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ck_test)+2^13
4040  END IF
4050  ENTER @Gpio_14 USING "#,W";Monitor
4060  IF NOT BIT(Monitor,Comparator_mon) THEN Sim_fail
4070  WAIT 2
4080  ENTER @Gpio_14 USING "#,W";Monitor
4090  IF NOT BIT(Monitor,Comparator_mon) THEN Sim_fail
4100  OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ck_test)!TURNS OFF VOLT TO HIGH
 VOLT P.S.
4110  WAIT 3
4120  OUTPUT @Gpio_14 USING "#,W";0
4130  WAIT 4
4140  OUTPUT @Gpio_12 USING "#,W";0
4150  ENTER @Gpio_14 USING "#,W";Monitor
4160  IF NOT BIT(Monitor,Comparator_mon) THEN Sim_fail
4170  GOSUB Safe_tester
4180  !
4190 Ck_reject: !
4200  PRINT CHR$(12)
4210  Label$=""
4220  Disp1$="VERIFYING INSUL. RES. SIMULATOR TEST "&Ck_test$
4230  Disp2$="ON INSUL. R. SIMULATOR BOX, MOVE SWITCHES TO POSITION "&Sw_pos$&"
AND REJECT"
```

```
4240 Disp3$=""
4250 GOSUB Wait_for_key
4260 GOSUB Arm_tester
4270 PRINT CHR$(12)
4280 OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ck_test)
4290 OUTPUT @Gpio_14 USING "#,W";Volt_to_uut_14(Ck_test)
4300 IF Ck_test=1 THEN
4310   OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ck_test)+2^13+2^9
4320 ELSE
4330   OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ck_test)+2^13
4340 END IF
4350 WAIT 2
4360 OUTPUT @Gpio_12 USING "#,W";Volt_to_uut_12(Ck_test)!TURNS OFF VOLT TO HIGH
 VOLT P.S.
4370 ENTER @Gpio_14 USING "#,W";Monitor    !CHECKS STATE OF COMPARATOR
4380 IF BIT(Monitor,Comparator_mon) THEN Sim_fail
4390 WAIT 3
4400 OUTPUT @Gpio_14 USING "#,W";0
4410 WAIT 4
4420 OUTPUT @Gpio_12 USING "#,W";0
4430 ENTER @Gpio_14 USING "#,W";Monitor  !CHECKS STATE COMPARATOR
4440 IF BIT(Monitor,Comparator_mon) THEN Sim_fail
4450 PRINT CHR$(12)
4460 GOSUB Safe_tester
4470 GOSUB Reset_ir
4480 RETURN
4490 !*****************************************************************************
4500 Sim_fail: ! INFORMS OF SIMULATOR FAILURE ON I.R. TESTING
4510 PRINT CHR$(12)
4520 CONTROL CRT;25,15
4530 PRINT "****** SIMULATOR FAILURE ******"
4540 BEEP 500,.5
4550 WAIT .5
4560 BEEP 500,.5
4570 PRINT CHR$(12)
4580 GOSUB Safe_tester
4590 GOSUB Reset_ir
4600 PRINT CHR$(12)
4610 MOVE 22,70
4620 PRINT TABXY(8,10);"SIMULATOR FAILURE.  CHECK SWITCH POSITIONS ON SIMULATOR
 BOX"
4630 PRINT TABXY(8,11);Disp2$
4640 ON KBD GOTO Ck_again
4650 DISP "DO YOU NEED TO CHECK THIS POSITION AGAIN (Y/N)?"
4660 Wait5:GOTO Wait5
4670 Ck_again:Ck$=KBD$
4680 IF Ck$="Y" THEN
4690   IF Disp2$[61,66]="REJECT" THEN
4700     GOTO Ck_reject
4710   ELSE
4720     GOTO Ck_accept
4730   END IF
```

```
4740 END IF
4750 IF Ck$<>"N" THEN Wait5
4760 PRINT CHR$(12)
4770 DISP ""
4780 CONTROL CRT;20,18
4790 PRINT "NOTIFY MAINTENANCE"
4800 GOTO End
4810 !**************************************************************************
4820 Door_open:    ! MAKES SURE SAFETY CHAMBER DOOR IS CLOSED
4830 ENTER @Gpio_14 USING "#,W";Monitor
4840 IF BIT(Monitor,Door_mon)=1 THEN
4850   CONTROL CRT;25,15
4860   PRINT CHR$(130);"****** CLOSE CHAMBER DOOR *****";CHR$(128)
4870   WAIT 3
4880   GOTO Door_open
4890 END IF
4900 PRINT CHR$(12)
4910 RETURN
4920 !**************************************************************************
4930 Safe_tester:! MAKES SURE KEY IS IN SAFE POSITION
4940 ENTER @Gpio_14 USING "#,W";Monitor
4950 IF BIT(Monitor,Key_mon)=0 THEN
4960   DISP ""
4970   CONTROL CRT;25,15
4980   PRINT CHR$(130);"****** SAFE TESTER *****";CHR$(128)
4990   WAIT 3
5000   GOTO Safe_tester
5010 END IF
5020 PRINT CHR$(12)
5030 RETURN
5040 !**************************************************************************
5050 Arm_tester:! MAKES SURE KEY IS IN ARMED POSITION
5060 ENTER @Gpio_14 USING "#,W";Monitor
5070 IF BIT(Monitor,Key_mon)=1 THEN
5080   CONTROL CRT;25,15
5090   PRINT CHR$(130);"****** ARM TESTER *****";CHR$(128)
5100   WAIT 3
5110   GOTO Arm_tester
5120 END IF
5130 RETURN
5140 !**************************************************************************
5150 Reset_ir:! INSTRUCTS TO RESET INSULATION RESISTANCE BUTTON
5160 ENTER @Gpio_14 USING "#,W";Monitor
5170 IF BIT(Monitor,Comparator_mon)=0 THEN
5180   CONTROL CRT;20,15
5190   PRINT CHR$(130);"****** RESET I.R. BUTTON *****";CHR$(128)
5200   WAIT 3
5210   GOTO Reset_ir
5220 END IF
5230 RETURN
5240 !*******************************************************************
5250 Wait_for_key: !*******
```

```
5260  ON KBD GOTO Exit1
5270  MOVE 22,70
5280  CSIZE 8,.4
5290  LABEL Label$
5300  PRINT TABXY(8,10);Disp1$
5310  PRINT TABXY(8,11);Disp2$
5320  PRINT TABXY(8,12);Disp3$
5330 Wait_loop1:DISP "Press ANY KEY when ready."
5340  WAIT .25
5350  DISP
5360  WAIT .25
5370  GOTO Wait_loop1
5380 Exit1:OFF KBD
5390  DISP
5400  CONTROL 1;1,18
5410  PRINT USING "50X"
5420  OFF ERROR
5430  PRINT CHR$(12)
5440  DISP ""
5450  RETURN
5460  !
5470 Post_cal:!
5480  PRINT TABXY(15,9);"CONTINUE WITH SECTION 3.10 OF CALIBRATION PROCEDURE"
5490 End:END
```

BIBLIOGRAPHY

1. Fink, D. G., ed. Standard Handbook for Electrical Engineers. New York: McGraw-Hill Book Company, 1969.

2. Hayt, Jr., W.H. Engineering Electromagnetics. New York: McGraw-Hill, 1981.

3. Hewlett-Packard, Basic 4.0 Reference Manuals for the HP9000 Series 200/300 Computers. Fort Collins: Hewlett-Packard, 1986.

4. Hewlett_packard, Operator's Manual 3478A Multimeter. Loveland, Colorado: Hewlett-packard, 1981.

5. Keithley Instruments, Model 580 Micro-ohmmeter Manuals. Cleveland: Keithley Instruments Inc., 1987.

6. Sedra, Adel S., and Smith, Kenneth C. Micro-Electronic Circuits. New York: CBS College Publishing, 1982.

R008704819