Electrical Analysis of Low Energy Argon Ion Bombarded GaAs

by

Eric D. Cole

Dissertation submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering

APPROVED:

Dr. L. C. Burton, Chairman

Dr. F. W. Stephenson Dr. S. Onishi

Dr. R. Zallen Dr. J. Dillard

June 1988
Blacksburg, Virginia
An electrical analysis was done on Al and Au Schottky diodes fabricated on n-type (100) GaAs which had been bombarded with low energy Ar ions. The purpose of this study was to quantify electrically damage caused by the Ion Beam Etching (IBE) as functions of energy and fluence.

Electrical studies included Deep Level Transient Spectroscopy (DLTS), Current-Voltage (I-V), Capacitance-Voltage (C-V), Conductance-Voltage (G-V), Capacitance-Temperature (C-T), and Activation Energy Analysis. These electrical measurements were carried out on GaAs which had been exposed to a variety of treatments after IBE (such as chemical etch removal) to determine damage depth.

At the lowest energy studied, 0.5keV, Schottky reverse saturation currents ($I_{sat}$) increased by over 4 orders of magnitude from the virgin case. The ideality factor, $n$, increased slightly while the breakdown voltage decreased. The most prominent changes occurred in the DLTS spectrum where it was observed that the native arsenic defect EL2 peak disappeared completely after ion etching. Concurrently a sharp increase in the diode conductivity with temperature was seen. It was found that chemical removal of 100Å of GaAs by chemical means could restore most of the diode parameters and the EL2 peak. It is proposed that the loss of EL2 is not related to a true physical reduction (i.e. an arsenic deple-
tion) since calculations showed that the As loss would have extended beyond 3000Å for detectable DLTS changes. Also, the EL2 peak could be made to artificially disappear on a virgin sample with an external diode shunting resistor. The loss of the EL2 peak is, rather, attributed to a thin low resistivity surface layer having a partly amorphous non-stoichiometric crystal structure which can desensitize or mask the DLTS measurement. Surface chemical etch studies over the top of the Schottky diodes recovered 25% of the EL2 peak supporting this conclusion. Lower fluences had no effect at 0.5keV.

Increasing ion bombardment energy showed a steady degradation in diode ideality factors. The reverse breakdown voltage increased past the unetched value and the DLTS spectrum began to show a very slight return of EL2. At 3keV the ideality factor was large, indicating the presence of a somewhat thicker high resistance layer. In fact recovery of diode parameters and EL2 did not occur until after 1000Å removal. This was much deeper than expected at this energy, according to theory.

Physical and lumped R-C electrical models are reported with an accompanying computer simulation of experimental DLTS results. The simulation used both thin low resistance and thick high resistance top layers to show that EL2 could be removed artificially. The models were also somewhat successful in explaining previously reported capacitance dispersion found in IBE GaAs.
ACKNOWLEDGEMENTS

The author would like to thank Dr. L. C. Burton, who served as his committee chairman, for providing him with research funding and facilities as well as invaluable information concerning research and teaching that will serve well in the author's future professional career. Thanks also to Dr. R. Zallen and Dr. J. Dillard for serving on the author's committee, and their helpful suggestions and assistance during the course of this research, including the use of the Surface Analysis Lab and communication of their own results which helped guide this study. The author would also like to thank Dr. F. W. Stephenson not only for serving on the committee but also for use of the Hybrid Microelectronics Lab facilities; and Dr. S. Onishi for being on the committee and for many interesting, insightful conversations.

Much gratitude to for the many hours he spent helping to develop the MEDUSA software and his patience for my continuously changing designs. Thanks to for her many hours spent preparing the GaAs samples. Thanks to for his helpful assistance in making "good" diodes, and his and efforts in bringing the big evaporator up to par. The author appreciates the help and skill of for his contributions to the entire project.

The author would also like to thank his lab and research colleagues and many others for their friendship, advice and good humor.
Lastly I would like to thank my wife for her hard work, good management and constant support throughout my life with her. Also to my family and my wife's family who provided us an oasis in the desert. And finally to my daughter whose smiling face melts through the tension of the day like the sun in winter.

This research was funded by Texas Instruments and the Virginia Center for Innovative Technology.
This work is dedicated to my father-in-law I am truly privileged and fortunate to have encountered such an individual in my lifetime. And though the sun still rises and the pages of books still turn, these things are forever different for me. I weep at his passing but my heart rejoices that I knew and loved him and the course of my life is and ever shall be enriched.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>iv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xii</td>
</tr>
<tr>
<td>Chapter 1: Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Basic Properties of GaAs</td>
<td>1</td>
</tr>
<tr>
<td>1.2 GaAs Device Fabrication</td>
<td>7</td>
</tr>
<tr>
<td>1.2.1 Ion Beam Etching (IBE)</td>
<td>14</td>
</tr>
<tr>
<td>1.3 Literature Review and Research Objectives</td>
<td>18</td>
</tr>
<tr>
<td>Chapter 2: Schottky Diodes on GaAs</td>
<td>24</td>
</tr>
<tr>
<td>2.1 Schottky Contacts</td>
<td>24</td>
</tr>
<tr>
<td>2.1.1 Ideal</td>
<td>24</td>
</tr>
<tr>
<td>2.1.2 Surface States</td>
<td>27</td>
</tr>
<tr>
<td>2.1.3 Non-ideal</td>
<td>29</td>
</tr>
<tr>
<td>2.1.4 Current-Voltage</td>
<td>31</td>
</tr>
<tr>
<td>2.1.5 Capacitance-Voltage</td>
<td>38</td>
</tr>
<tr>
<td>2.2 Ohmic Contacts</td>
<td>41</td>
</tr>
<tr>
<td>Chapter 3: Experimental Methods</td>
<td>46</td>
</tr>
<tr>
<td>3.1 Schottky Diode Fabrication</td>
<td>46</td>
</tr>
<tr>
<td>3.2 Automated Electrical Analysis (MEDUSA)</td>
<td>52</td>
</tr>
<tr>
<td>3.2.1 Hardware</td>
<td>52</td>
</tr>
<tr>
<td>3.2.2 Software</td>
<td>55</td>
</tr>
<tr>
<td>3.3 Electrical Measurements</td>
<td>56</td>
</tr>
<tr>
<td>3.4 Sample Grouping</td>
<td>61</td>
</tr>
<tr>
<td>Chapter 4: Results and Discussion</td>
<td>63</td>
</tr>
<tr>
<td>4.1 Vendor and Metal</td>
<td>63</td>
</tr>
<tr>
<td>4.2 Energy</td>
<td>66</td>
</tr>
<tr>
<td>4.3 Fluence</td>
<td>78</td>
</tr>
<tr>
<td>4.4 Chemical Etch</td>
<td>86</td>
</tr>
<tr>
<td>4.4.1 Surface Removal</td>
<td>97</td>
</tr>
<tr>
<td>4.4.2 Shunt Resistor</td>
<td>101</td>
</tr>
<tr>
<td>4.5 Anneal</td>
<td>101</td>
</tr>
<tr>
<td>Chapter 5: Modelling and Computer Analysis</td>
<td>105</td>
</tr>
<tr>
<td>5.1 Physical Model</td>
<td>105</td>
</tr>
<tr>
<td>5.2 Electrical Model</td>
<td>107</td>
</tr>
<tr>
<td>5.2.1 DLTS Simulation</td>
<td>107</td>
</tr>
<tr>
<td>5.2.2 Capacitance-Frequency Simulation</td>
<td>110</td>
</tr>
<tr>
<td>Chapter 6: Conclusions</td>
<td>116</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>GaAs crystal structure</td>
<td>2</td>
</tr>
<tr>
<td>2.</td>
<td>Electronic band structure of GaAs</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>Dependency of EL2 density on GaAs melt stoichiometry</td>
<td>8</td>
</tr>
<tr>
<td>4.</td>
<td>Radial distribution of EL2 across wafer with EPD correlation</td>
<td>9</td>
</tr>
<tr>
<td>5.</td>
<td>Flowchart of a typical GaAs integrated circuit fabrication process</td>
<td>10</td>
</tr>
<tr>
<td>6.</td>
<td>Cutaway of GaAs wafers showing the effects of different etch rates on structure definition</td>
<td>13</td>
</tr>
<tr>
<td>7.</td>
<td>Typical system used for ion-beam etching</td>
<td>16</td>
</tr>
<tr>
<td>8.</td>
<td>Typical uses of low energy IBE including (a) wafer cleaning and (b) MESFET gate recessing</td>
<td>17</td>
</tr>
<tr>
<td>9.</td>
<td>Band diagram for an ideal Schottky contact to an n-type semiconductor</td>
<td>25</td>
</tr>
<tr>
<td>10.</td>
<td>Band diagram of a semiconductor containing surface states</td>
<td>28</td>
</tr>
<tr>
<td>11.</td>
<td>Bardeen model of a non-ideal Schottky contact</td>
<td>30</td>
</tr>
<tr>
<td>12.</td>
<td>Schematic indicating current components in forward bias</td>
<td>32</td>
</tr>
<tr>
<td>13.</td>
<td>Effect of thin (30Å) interfacial oxide layer on I-V characteristics of Schottky diodes</td>
<td>37</td>
</tr>
<tr>
<td>14.</td>
<td>Schottky contact indicating space charge region</td>
<td>39</td>
</tr>
<tr>
<td>15.</td>
<td>Typical ohmic contact for $\phi_m &lt; \phi_s$ for an n-type semiconductor</td>
<td>42</td>
</tr>
<tr>
<td>16.</td>
<td>Tunnel type ohmic contact</td>
<td>43</td>
</tr>
<tr>
<td>17.</td>
<td>Diagram of as received wafer indicating (a) crystal orientations, (b) cut pieces and (c) IBE angles</td>
<td>47</td>
</tr>
<tr>
<td>18.</td>
<td>Schottky diode processing flowchart</td>
<td>48</td>
</tr>
<tr>
<td>19.</td>
<td>Evaporation jig and mask arrangements for depositing ohmic and Schottky contacts</td>
<td>50</td>
</tr>
<tr>
<td>Page</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>20.</td>
<td>TO-8 package and attached diode</td>
<td></td>
</tr>
<tr>
<td>21.</td>
<td>MEDUSA hardware layout</td>
<td></td>
</tr>
<tr>
<td>22.</td>
<td>DLTS pulse biasing and electron trapping process</td>
<td></td>
</tr>
<tr>
<td>23.</td>
<td>Comparison of DLTS spectra done on diodes made on wafers from Airtron and Morgan</td>
<td></td>
</tr>
<tr>
<td>24.</td>
<td>Current-Voltage curves for ion energy comparison (Inset shows In(I) vs V of the forward biased diodes.)</td>
<td></td>
</tr>
<tr>
<td>25.</td>
<td>Typical C-V dopant profile for IBE samples</td>
<td></td>
</tr>
<tr>
<td>26.</td>
<td>Conductance-temperature comparison of different IBE energies with the Airtron sample included for further comparison</td>
<td></td>
</tr>
<tr>
<td>27.</td>
<td>DLTS spectra comparing different IBE energies</td>
<td></td>
</tr>
<tr>
<td>28.</td>
<td>DLTS spectra showing loss of EL2 in 3keV IBE energy Airtron samples as compared to virgin material</td>
<td></td>
</tr>
<tr>
<td>29.</td>
<td>Current-Voltage characteristics for 0.5keV energy with variation of fluence</td>
<td></td>
</tr>
<tr>
<td>30.</td>
<td>Conductance-temperature characteristics for 0.5keV energy with variation of fluence</td>
<td></td>
</tr>
<tr>
<td>31.</td>
<td>DLTS spectra for 0.5keV energy, at different ion fluences</td>
<td></td>
</tr>
<tr>
<td>32.</td>
<td>Current-voltage characteristics for 3keV energy with variation of fluence</td>
<td></td>
</tr>
<tr>
<td>33.</td>
<td>DLTS spectra for 3keV energy at different ion fluences</td>
<td></td>
</tr>
<tr>
<td>34.</td>
<td>Current-voltage characteristics for 0.5keV energy depth profile</td>
<td></td>
</tr>
<tr>
<td>35.</td>
<td>Conductance-temperature characteristics for 0.5keV energy depth profile</td>
<td></td>
</tr>
<tr>
<td>36.</td>
<td>DLTS spectra for 0.5keV energy depth profile</td>
<td></td>
</tr>
<tr>
<td>37.</td>
<td>Current-Voltage characteristics of 3keV energy depth profile</td>
<td></td>
</tr>
<tr>
<td>38.</td>
<td>DLTS spectra for 3keV energy depth profile</td>
<td></td>
</tr>
<tr>
<td>39.</td>
<td>Conductance-temperature virgin, 0.5keV and 2000 Å surface removal</td>
<td></td>
</tr>
</tbody>
</table>
40. DLTS spectra comparing virgin, 0.5keV and 2000 Å surface removal.........................................................100
41. Experimental setup for shunt resistor test.................................102
42. DLTS spectra for shunt resistor test......................................103
43. Physical model of IBE damaged GaAs........................................106
44. Electrical model corresponding to the physical model.....................108
45. Curves showing dependency of EL2 peak magnitude on RA and Rs variation.........................................................111
46. Simulated DLTS for 0.5keV layer model....................................112
47. Simulated DLTS for 3keV layer model.......................................113
48. Capacitance–frequency dispersion simulated using electrical model as compared to Sen's results............................114
49. Flowchart for MEDUSA software............................................127
50. Flowchart for general simulation software....................................289
<table>
<thead>
<tr>
<th>#</th>
<th>Table Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>GaAs physical and electrical properties</td>
<td>5</td>
</tr>
<tr>
<td>2.</td>
<td>Common defects found in GaAs</td>
<td>6</td>
</tr>
<tr>
<td>3.</td>
<td>Common wet etchants used on GaAs along with their etch rates</td>
<td>12</td>
</tr>
<tr>
<td>4.</td>
<td>Effects of Plasma Etching and Reactive Ion Etching on GaAs and GaAs devices</td>
<td>15</td>
</tr>
<tr>
<td>5.</td>
<td>Theoretical ion implantation parameters for argon in GaAs (RP = range, DRP = straggle)</td>
<td>19</td>
</tr>
<tr>
<td>6.</td>
<td>Summary of useful literature of IBE on GaAs</td>
<td>21</td>
</tr>
<tr>
<td>7.</td>
<td>Common Schottky metals to various semiconductors</td>
<td>26</td>
</tr>
<tr>
<td>8.</td>
<td>Common ohmic metal systems used on GaAs</td>
<td>44</td>
</tr>
<tr>
<td>9.</td>
<td>Schottky diode parameters and their graphical derivations</td>
<td>57</td>
</tr>
<tr>
<td>10.</td>
<td>DLTS trap parameters for Airtron and Morgan wafers</td>
<td>65</td>
</tr>
<tr>
<td>11.</td>
<td>Energy and fluence ion etching schedule</td>
<td>67</td>
</tr>
<tr>
<td>12.</td>
<td>Diode parameters for different ion energies</td>
<td>69</td>
</tr>
<tr>
<td>13.</td>
<td>Diode parameters for different fluences</td>
<td>79</td>
</tr>
<tr>
<td>14.</td>
<td>Chemical etch schedule used in depth profile</td>
<td>88</td>
</tr>
<tr>
<td>15.</td>
<td>Diode parameters for 0.5keV depth profile</td>
<td>89</td>
</tr>
<tr>
<td>16.</td>
<td>Diode parameters for 3keV depth profile</td>
<td>93</td>
</tr>
</tbody>
</table>
Chapter 1: INTRODUCTION

GaAs is not a new semiconductor (it was first made in the 1920's by Goldschmidt) but it is beginning to emerge from the laboratory into the market place [1]. Once touted as a replacement for Si, GaAs is now seen as a material that fills the gaps left by Si in microwave and high speed digital component, electro-optic (lasers, photodetectors) and power component applications. Presently the major consumer of GaAs, due to its high cost and specialized uses, is the military. This will change as material quality and device processing yield improves. The aim of this thesis is to report in detail on one processing step, Ion-Beam Etching (IBE), and its effects on Schottky device quality. First, however, it will be instructive to review some of the basic properties of GaAs and processing steps involved in device fabrication.

1.1 Basic Properties of GaAs

GaAs is a III-V compound semiconductor having a zincblende crystal structure (see Fig. 1), with a lattice constant \( a = 5.6533\text{Å} \) and the two sublattices separated by \( 2.44793\text{Å} \) along the body diagonal of the unit cube.

Figure 2 shows the band structure of GaAs from which several important properties can be interpreted. GaAs has a direct bandgap with \( E_G = 1.423\text{eV} \), making it ideally suited for electro-optic applications, since electron transitions from the valence band to the conduction band and vice versa at \( k = 0 \) occur through pure photon interaction (i.e. no phonons necessary) [4]. These properties are exploited in GaAs lasers and photodetectors. As a result of the wider bandgap and the presence
Figure 1. GaAs crystal structure (black = Ga, white = As) [taken from Ref. 2].
Figure 2. Electronic band structure of GaAs [taken from Ref. 3].
of a native arsenic defect, named EL2, GaAs is semi-insulating in its undoped state, giving excellent device isolation compared to Si, thus eliminating the need for guard ring structures. (EL2 is further explained later in this section.)

Another salient feature of the band structure can be derived from the sharpness of the conduction band at \( k = 0 \) where the electron effective mass is smallest [5]. This results in an electron effective mass to rest mass ratio \( \left( \frac{m^*}{m_0} \right) \) of .067 for GaAs compared to .98 for Si. Thus for a given applied electric field the electron will attain a higher saturated velocity in GaAs than in Si, thus making GaAs an electrically faster material. This is usually seen in a comparison of the electron mobilities: \( \mu(\text{GaAs}) = 8500 \text{cm}^2/\text{V—s}, \mu(\text{Si}) = 1500 \text{cm}^2/\text{V—s} \). The hole mobilities are nearly equal \( (\approx 400 \text{cm}^2/\text{V—s}) \) indicating that n-type unipolar devices are the most advantageous structures in GaAs [6]. Being a faster material allows devices to be fabricated that can operate at higher frequencies \( (>100 \ \text{GHz}) \) and switch in shorter times \( (<100 \text{ps}) \) [7].

The band structure of GaAs also indicates the possibility of inter-valley scattering via phonon interaction (i.e. transferred electron effect). This property is exploited in microwave devices such as the Transferred Electron Device (TED). A summary of these and other GaAs properties is shown in Table 1.

It is appropriate at this point to discuss the traps and defects native to or typically found in GaAs in view of the study to be reported later in this paper. An extensive literature search was performed, with the results given in Table 2 [9—33]. As can be seen the most commonly reported electron trap in all types of GaAs is EL2.
Table 1. GaAs physical and electrical properties [taken from Ref. 8].

<table>
<thead>
<tr>
<th>Properties</th>
<th>GaAs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atoms/cm³</td>
<td>$4.42 \times 10^{22}$</td>
</tr>
<tr>
<td>Atomic weight</td>
<td>144.63</td>
</tr>
<tr>
<td>Breakdown field (V/cm)</td>
<td>$-4 \times 10^6$</td>
</tr>
<tr>
<td>Crystal structure</td>
<td>Zincblende</td>
</tr>
<tr>
<td>Density (g/cm³)</td>
<td>5.32</td>
</tr>
<tr>
<td>Dielectric constant</td>
<td>13.1</td>
</tr>
<tr>
<td>Effective density of states in conduction band, $N_C$ (cm⁻³)</td>
<td>$4.7 \times 10^{16}$</td>
</tr>
<tr>
<td>Effective density of states in valence band, $N_V$ (cm⁻³)</td>
<td>$7.0 \times 10^{16}$</td>
</tr>
<tr>
<td>Effective Mass, m²/mₐ, Electrons</td>
<td>0.067</td>
</tr>
<tr>
<td>Holes</td>
<td>$m_f = 0.082$</td>
</tr>
<tr>
<td></td>
<td>$m_i = 0.43$</td>
</tr>
<tr>
<td>Electron affinity, $\chi(V)$</td>
<td>4.07</td>
</tr>
<tr>
<td>Energy gap (eV) at 300 K</td>
<td>1.424</td>
</tr>
<tr>
<td>Intrinsic carrier concentration (cm⁻³)</td>
<td>$1.79 \times 10^{16}$</td>
</tr>
<tr>
<td>Intrinsic Debye length (Å)</td>
<td>2250</td>
</tr>
<tr>
<td>Intrinsic resistivity (Ω-cm)</td>
<td>$10^3$</td>
</tr>
<tr>
<td>Lattice constant (Å)</td>
<td>5.6533</td>
</tr>
<tr>
<td>Linear coefficient of thermal expansion, $\Delta L/L = \alpha T$ (°C⁻¹)</td>
<td>$6.86 \times 10^{-4}$</td>
</tr>
<tr>
<td>Melting point (°C)</td>
<td>1238</td>
</tr>
<tr>
<td>Minority carrier lifetime (s)</td>
<td>$-10^{-5}$</td>
</tr>
<tr>
<td>Mobility (drift) (cm²/V-s)</td>
<td>8500</td>
</tr>
<tr>
<td></td>
<td>400</td>
</tr>
<tr>
<td>Optical-phonon energy (eV)</td>
<td>0.035</td>
</tr>
<tr>
<td>Phonon mean free path $\lambda_p$ (Å)</td>
<td>58</td>
</tr>
<tr>
<td>Specific heat (J/g-°C)</td>
<td>0.35</td>
</tr>
<tr>
<td>Thermal conductivity at 300 K (W/cm°C)</td>
<td>0.46</td>
</tr>
<tr>
<td>Thermal diffusivity (cm²/s)</td>
<td>0.24</td>
</tr>
<tr>
<td>Vapor pressure (Pa) at 1050°C</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>1 at 900°C</td>
</tr>
</tbody>
</table>
Table 2. Common defects found in GaAs [9-33].

<table>
<thead>
<tr>
<th>Common Name</th>
<th>$E_r$ (eV)</th>
<th>$N_r$ (cm$^{-3}$)</th>
<th>x-sect (cm$^2$)</th>
<th>GaAs</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>EL2</td>
<td>0.72-0.82</td>
<td>$10^5$</td>
<td>$10^{-12}$</td>
<td>LEC, VPE, MBE, S.I., MOCVD</td>
<td>As$_x$, cluster</td>
</tr>
<tr>
<td>EL0</td>
<td>0.82</td>
<td>$10^5$</td>
<td>$10^{-12}$</td>
<td>LEC, S.I.</td>
<td>oxygen</td>
</tr>
<tr>
<td>EL12</td>
<td>0.72</td>
<td>$10^6$</td>
<td>$10^{-12}$</td>
<td>LEC</td>
<td>EL2?</td>
</tr>
<tr>
<td>Cr?</td>
<td>0.62</td>
<td>--</td>
<td>$10^{-12}$</td>
<td>LEC, VPE</td>
<td>chromium</td>
</tr>
<tr>
<td>EL3</td>
<td>0.58</td>
<td>$10^6$</td>
<td>$10^{-12}$</td>
<td>LEC, HB, HGF</td>
<td></td>
</tr>
<tr>
<td>EL4</td>
<td>0.52</td>
<td>$10^6$</td>
<td>$10^{-12}$</td>
<td>LEC</td>
<td></td>
</tr>
<tr>
<td>EL5</td>
<td>0.42</td>
<td>--</td>
<td>$10^{-12}$</td>
<td>LEC, HB</td>
<td></td>
</tr>
<tr>
<td>EL6</td>
<td>0.34</td>
<td>$10^6$</td>
<td>$10^{-12}$</td>
<td>LEC, HB</td>
<td></td>
</tr>
<tr>
<td>EL8</td>
<td>0.27</td>
<td>--</td>
<td>$10^{-12}$</td>
<td>LEC, HB, MOCVD</td>
<td></td>
</tr>
<tr>
<td>EL14</td>
<td>0.22</td>
<td>$10^5$</td>
<td>$10^{-12}$</td>
<td>LEC, HGF</td>
<td>oxygen</td>
</tr>
</tbody>
</table>
EL2 is a trap which is generally attributed to an $\text{As}_{\text{Ga}}$ antisite or As cluster defect, though much controversy and research have surrounded EL2 and its source. Early reports showed oxygen impurities to be a possibility since its energy was very close, but much data supports arsenic defects, including crystal stoichiometry-EL2 density relationships [34-40].

Many factors affect the density of EL2 in wafers and epi-layers including melt stoichiometry (As inbalances), anneal temperature and even saw speed (refer to Fig. 3) [41]. The radial distribution of EL2 across LEC wafers is of prime importance in device fabrication and is usually measured using optical scanning techniques, though a chemical etch with KOH gives an etch pit density (EPD) which often correlates with the EL2 distribution (refer to Fig. 4) [42]. EL2 is generally considered responsible for giving GaAs its semi-insulating characteristic in the undoped state since it is located approximately 0.75eV below the conduction band, near mid-gap, effectively pinning the bulk Fermi level at midgap for S.I. behavior. EL2 as measured by DLTS will be seen later to be a very important monitor of device quality.

Many other physical and electrical properties can be seen in the next section on device processing.

1.2 GaAs Device Fabrication

Fabrication of GaAs devices is a long, involved process from ingot growth to finished device. A typical processing flowchart is given in Fig. 5. Each fabrication step could and has been the subject of much research activity, and a few are briefly described below.
Figure 3. Dependency of EL2 density on GaAs melt stoichiometry [taken from Ref. 41].
Figure 4. Radial distribution of EL2 across wafer with EPD correlation [taken from Ref. 42].
Figure 5. Flowchart of a typical GaAs integrated circuit fabrication process [taken from Ref. 1, pp. 14-15].
Industrial wafers are typically 2" in diameter (3" wafers are presently gaining popularity) grown using Liquid Encapsulated Czochralski (LEC) techniques, though Horizontal Bridgman growth is also used. Epitaxial layers for device fabrication can be made using ion-implantation, MBE, LEC or MOCVD techniques, depending on the device to be produced. The wafer and epi or active layer quality are critical, since defects can lower electron mobility drastically, effectively destroying devices. These defects may be introduced by anything from ingot pull rate to implant energy, as was seen in the previous section.

Wafer processing also includes some wet cleaning and etching steps for surface preparation and structural definition. Typical organic solvents used for degreasing and general purpose cleaning include: ethanol, acetone, benzene and propanol. Etchants for GaAs along with their etch rates can be seen in Table 3. The H₂SO₄:H₂O₂:H₂O and Br₂:CH₃OH systems are the most popular with the second being useful for polishing. Due to its crystal structure, GaAs wet etches at different rates along different planes, with variables such as solution temperature and stir speed also affecting etch rates. This creates a problem in resolution, design and controllability during processing which can affect yield and cost (refer to Fig. 6).

To circumvent some of the drawbacks of wet etching, dry processing techniques can be used and are gaining popularity. These techniques include plasma assisted etching (PE), Reactive Ion Etching (RIE) and Ion Beam Etching (IBE). Plasma etching occurs in a similar fashion to metal deposition using Rf sputtering, except that the target becomes the GaAs wafer itself. The etch rate is the highest of the three and plasma
Table 3. Common wet etchants used on GaAs along with their etch rates [taken from Ref. 43].

(proportions are equal volumes unless otherwise noted)

<table>
<thead>
<tr>
<th>Etchant</th>
<th>Etch Rate (μm/min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCl/CH$_3$COOH/H$_2$O</td>
<td>2.0</td>
</tr>
<tr>
<td>HCl/CH$_3$COOH/(1N-K, Cr, O$_2$)</td>
<td>0.40</td>
</tr>
<tr>
<td>HCl/H$_3$PO$_4$/H$_2$O</td>
<td>0.75</td>
</tr>
<tr>
<td>HCl/H$_3$PO$_4$/(1N-K, Cr, O$_2$)</td>
<td>0.04</td>
</tr>
<tr>
<td>HCl/HNO$_3$ (1:1)</td>
<td>0.50</td>
</tr>
<tr>
<td>(1:2)</td>
<td>0.75</td>
</tr>
<tr>
<td>(2:1)</td>
<td>0.25</td>
</tr>
<tr>
<td>HCl/HNO$_3$/H$_2$O</td>
<td>0.83</td>
</tr>
<tr>
<td>HCl/HNO$_3$/H$_2$O$_2$</td>
<td>1.0</td>
</tr>
<tr>
<td>HCl/HNO$_3$/CH$_3$COOH</td>
<td>1.3</td>
</tr>
<tr>
<td>HNO$_3$/H$_2$O</td>
<td>7.0</td>
</tr>
<tr>
<td>HNO$_3$/CH$_3$COOH</td>
<td>-</td>
</tr>
<tr>
<td>HNO$_3$/CH$_3$COOH/H$_2$O</td>
<td>4.5</td>
</tr>
<tr>
<td>HNO$_3$/H$_2$PO$_4$</td>
<td>10.0</td>
</tr>
<tr>
<td>HNO$_3$/H$_2$PO$_4$/H$_2$O$_2$</td>
<td>3.5</td>
</tr>
<tr>
<td>H$_2$SO$_4$/H$_2$O$_2$/H$_2$O</td>
<td>5.0</td>
</tr>
<tr>
<td>H$_2$SO$_4$/CH$_3$COOH/H$_2$O</td>
<td>2.5</td>
</tr>
<tr>
<td>H$_2$SO$_4$/H$_2$PO$_4$/H$_2$O</td>
<td>3.0</td>
</tr>
<tr>
<td>H$_2$SO$_4$/HC/H$_2$O (1N-K, Cr, O$_2$)</td>
<td>0.75</td>
</tr>
<tr>
<td>H$_2$PO$_4$/H$_2$O$_2$/H$_2$O</td>
<td>4.0</td>
</tr>
<tr>
<td>H$_2$PO$_4$/CH$_3$COOH/H$_2$O</td>
<td>2.0</td>
</tr>
<tr>
<td>H$_2$PO$_4$/CH$_3$COOH/H$_2$O$_2$</td>
<td>2.5</td>
</tr>
<tr>
<td>HF/HNO$_3$/H$_2$O</td>
<td>10.0</td>
</tr>
<tr>
<td>HF/HNO$_3$/H$_2$O$_2$</td>
<td>8.0</td>
</tr>
<tr>
<td>HF/HNO$_3$/CH$_3$COOH</td>
<td>80.0</td>
</tr>
<tr>
<td>HF/HNO$_3$/H$_2$PO$_4$</td>
<td>50.0</td>
</tr>
<tr>
<td>HF/H$_2$SO$_4$/H$_2$O</td>
<td>21.0</td>
</tr>
<tr>
<td>HBr/HNO$_3$</td>
<td>0.75</td>
</tr>
<tr>
<td>HBr/HNO$_3$/H$_2$O</td>
<td>0.05</td>
</tr>
<tr>
<td>HBr/CH$_3$COOH/(1N-K, Cr, O$_2$)</td>
<td>1.5</td>
</tr>
<tr>
<td>HBr/H$_3$PO$_4$/(1N-K, Cr, O$_2$)</td>
<td>1.0</td>
</tr>
<tr>
<td>4% Br$_2$/CH$_3$OH</td>
<td>6.0</td>
</tr>
<tr>
<td>1% Br$_2$/CH$_3$OH</td>
<td>0.67</td>
</tr>
<tr>
<td>(1% Br$_2$/CH$_3$OH)/CH$_3$COOH</td>
<td>0.20</td>
</tr>
<tr>
<td>(1% Br$_2$/CH$_3$OH)/H$_2$PO$_4$</td>
<td>0.02</td>
</tr>
<tr>
<td>NaOCl</td>
<td>1.8</td>
</tr>
<tr>
<td>NaOCl/HCl (5:1)</td>
<td>0.75</td>
</tr>
<tr>
<td>(1N-NaOH)/H$_2$O/O/H$_2$O (1:1:10)</td>
<td>0.38</td>
</tr>
<tr>
<td>(1N-NaOH)/(1N-NaOH)/NH$_4$OH (5:1:1)</td>
<td>1.1</td>
</tr>
<tr>
<td>NH$_3$/H$_2$O/H$_2$O/O (1:1:5)</td>
<td>1.8</td>
</tr>
<tr>
<td>(1N-KOH)/H$_2$O/O/H$_2$O (1:1:10)</td>
<td>0.5</td>
</tr>
<tr>
<td>(1N-KOH)/(1N-KOH)/O/H$_2$O (5:1:1)</td>
<td>1.4</td>
</tr>
</tbody>
</table>
Figure 6. Cutaway of GaAs wafers showing the effects of different etch rates on structure definition [taken from Ref. 44].
etching is often used for via hole fabrication. RIE is similar to PE except in addition to the inert plasma ion a reactive species such as CF$_4$ is included to improve the anisotropic nature of the etch. A survey of the effects of these two techniques is given in Table 4. Processing variables in PE and RIE included ion energy, vacuum quality and wafer angle which can affect etch rates and crystal stoichiometry, degrading device quality. IBE is somewhat different from the above, and since it is the main subject of this research, its description is taken up in the next section.

This section briefly touched on a few major points involved in GaAs device processing. It was not extensive but was included to give the reader a flavor of the complicated nature of this subject and to show that device quality depends on a near infinity of variables, including those involved in IBE.

1.2.1 Ion Beam Etching (IBE)

Ion Beam Etching (IBE), or ion milling as it is sometimes called, is an important and useful processing technique for GaAs. A typical system is shown in Fig. 7 where it can be seen that IBE is simply the bombardment of a given surface with gas ions of various species, energies and fluences [46]. At low energies (100eV), inert gas ions, (usually Ar$^+$ for this study) are used to clean wafer surfaces of organic, oxide and other contaminations which could impede device fabrication. At higher energies (>500eV) the ion beam can be used to define structures for devices such as isolation mesas and, more importantly, recessed channels for gate metalization of MESFETs (Fig.
Table 4. Effects of Plasma Etching and Reactive Ion Etching on GaAs and GaAs devices [taken from Ref. 45, table entry references are given there].

<table>
<thead>
<tr>
<th>Gas used</th>
<th>Type of system used</th>
<th>Etch conditions</th>
<th>Observations, etch rates, surface and electrical effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCl₂F₂, CCl₄,</td>
<td>parallel plate, PE</td>
<td>power supplied to both electrodes; 200-500 mTorr</td>
<td>etched both GaAs and oxides; GaAs etch rate much higher than oxide etch rate</td>
</tr>
<tr>
<td>PC₁₀, HCl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cl₂COCl₂</td>
<td>same</td>
<td>same</td>
<td>GaAs was etched, oxide was not; etch rate ~ 5 µm min⁻¹ at 5 W cm⁻² for Cl₂, very rough surface</td>
</tr>
<tr>
<td>CCl₂F₂/Ar/O₂</td>
<td>RF-RIE (1:1)</td>
<td>0.5 W cm⁻²; 5 mTorr, 500 V self-bias; 10 mTorr, 650 V self-bias</td>
<td>etch rate ~ 5 µm min⁻¹ at 5 mTorr, 3800 Å min⁻¹ etch rate; at 10 mTorr, 0 Å min⁻¹ etch rate; very sharp angle etch walls</td>
</tr>
<tr>
<td>Ar</td>
<td>DC-RIE, -3 kV on bottom; 40 mTorr</td>
<td>400 Å min⁻¹ etch rate</td>
<td></td>
</tr>
<tr>
<td>Ar/CCl₂F₂</td>
<td>same</td>
<td>25-60 mTorr</td>
<td>500-2000 Å min⁻¹ etch rate</td>
</tr>
<tr>
<td>(9:1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCl₂F₂</td>
<td>same</td>
<td>10-40 mTorr</td>
<td>2000-800 Å min⁻¹ etch rate; surface carbon build-up observed and ion bombardment helps remove it; rate-limiting factor is GaF₃ sputter removal</td>
</tr>
<tr>
<td>CCl₂F₂/He</td>
<td>RF-RIE (1:1)</td>
<td>40 mTorr, 0.18 W cm⁻²</td>
<td>GaAs etch rate = 200:1, with ~ 20 Å min⁻¹ for AlGaAs; more surface degradation observed for lower He</td>
</tr>
<tr>
<td>CCl₂F₂/O₂</td>
<td>RF-RIE (1:1)</td>
<td>0.1 mTorr, 50 W</td>
<td>7 µm min⁻¹ GaAs etch rate</td>
</tr>
<tr>
<td>(55 kHz)</td>
<td>(area unknown)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCl₃F/O₂</td>
<td>300 °C, 160 mTorr</td>
<td></td>
<td>addition of O₂ binds carbon and releases Cl and F ions for GaAs etching; high O₂ concom lowers etch rate due to GaAs surface oxidation</td>
</tr>
<tr>
<td>Br₂</td>
<td>0.1-14 MHz</td>
<td>300 mTorr, 30 SCCM Br₂, 100 °C, &lt; 0.5 W cm⁻² for &lt; 0.3 W cm⁻², etch rates (111) Ga &lt; (110) &lt; (100) &lt; (111) As; first demonstration of preferential crystallographic plasma etching of GaAs</td>
<td></td>
</tr>
<tr>
<td>CF₃, CHF₃</td>
<td>RF-RIE, 500 V self-bias</td>
<td>φₐ and ½ₐ increase, but new deep levels appear</td>
<td></td>
</tr>
</tbody>
</table>

mTorr = mTorr.
Figure 7. Typical system used for ion-beam etching [taken from Ref. 46].
Figure 8. Typical uses of low energy IBE including (a) wafer cleaning and (b) MESFET gate recessing.
IBE processing is very useful in the latter application due to the extreme anisotropic nature of the beam. This minimizes mask undercutting, thus improving resolution of device features.

IBE, however, is not without its detriments. Since energetic ions are incident on the surface, radiation damage in the semiconductor is inevitable. Further, in GaAs it is found that arsenic is preferentially etched, making crystal stoichiometry with associated defects a problem [47]. This damage in general reduces the quality and reproducibility of devices made on their surfaces.

Ar\(^+\) IBE can also be viewed as low energy ion-implantation with an unreactive ion species. Therefore implant theory and tables may be used for prediction of depth of penetration and straggle. Table 5 lists values for Ar\(^+\) in GaAs assuming a Gaussian distribution. Further use of this table will be made in subsequent chapters.

1.3 Literature Review and Research Objectives

Although there is a vast amount of literature on GaAs in general, the amount which includes studies of electrical effects of IBE is somewhat small. However a review and summary of these results will be helpful in outlining further research.

Most of these studies have been done on either Al or Au Schottky diodes fabricated on (100) n-GaAs [49,50]. Of course these parameters are dependent upon processing techniques. The barrier height has been shown to be more or less independent of the metal due to surface state pinning of the Fermi level [51]. Al is more thermally stable than Au up to temperatures of 450°C, while Au tends to react with the substrate and getter Ga at 200°C [52].
Table 5. Theoretical ion implantation parameters for argon in GaAs
(RP = range, DRP = straggle [taken from ref. 48].)

<table>
<thead>
<tr>
<th>E</th>
<th>RP</th>
<th>DRP</th>
<th>DY</th>
<th>SK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>20</td>
<td>17</td>
<td>0.43</td>
</tr>
<tr>
<td>3</td>
<td>45</td>
<td>36</td>
<td>31</td>
<td>0.42</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>47</td>
<td>41</td>
<td>0.42</td>
</tr>
<tr>
<td>10</td>
<td>93</td>
<td>70</td>
<td>61</td>
<td>0.40</td>
</tr>
<tr>
<td>20</td>
<td>153</td>
<td>108</td>
<td>95</td>
<td>0.37</td>
</tr>
<tr>
<td>30</td>
<td>213</td>
<td>142</td>
<td>125</td>
<td>0.34</td>
</tr>
<tr>
<td>40</td>
<td>273</td>
<td>175</td>
<td>155</td>
<td>0.30</td>
</tr>
<tr>
<td>50</td>
<td>335</td>
<td>207</td>
<td>184</td>
<td>0.27</td>
</tr>
<tr>
<td>60</td>
<td>397</td>
<td>238</td>
<td>212</td>
<td>0.24</td>
</tr>
<tr>
<td>70</td>
<td>461</td>
<td>268</td>
<td>241</td>
<td>0.22</td>
</tr>
<tr>
<td>80</td>
<td>525</td>
<td>298</td>
<td>269</td>
<td>0.19</td>
</tr>
<tr>
<td>100</td>
<td>657</td>
<td>358</td>
<td>325</td>
<td>0.14</td>
</tr>
<tr>
<td>120</td>
<td>792</td>
<td>415</td>
<td>381</td>
<td>0.11</td>
</tr>
<tr>
<td>140</td>
<td>930</td>
<td>471</td>
<td>436</td>
<td>0.07</td>
</tr>
<tr>
<td>160</td>
<td>1070</td>
<td>526</td>
<td>491</td>
<td>0.03</td>
</tr>
<tr>
<td>180</td>
<td>1212</td>
<td>579</td>
<td>545</td>
<td>-0.01</td>
</tr>
<tr>
<td>200</td>
<td>1356</td>
<td>632</td>
<td>599</td>
<td>-0.04</td>
</tr>
<tr>
<td>220</td>
<td>1501</td>
<td>683</td>
<td>652</td>
<td>-0.07</td>
</tr>
<tr>
<td>240</td>
<td>1647</td>
<td>733</td>
<td>705</td>
<td>-0.10</td>
</tr>
<tr>
<td>260</td>
<td>1794</td>
<td>782</td>
<td>757</td>
<td>-0.13</td>
</tr>
<tr>
<td>280</td>
<td>1943</td>
<td>830</td>
<td>808</td>
<td>-0.16</td>
</tr>
<tr>
<td>300</td>
<td>2091</td>
<td>877</td>
<td>858</td>
<td>-0.19</td>
</tr>
<tr>
<td>350</td>
<td>2465</td>
<td>989</td>
<td>982</td>
<td>-0.25</td>
</tr>
<tr>
<td>400</td>
<td>2841</td>
<td>1096</td>
<td>1101</td>
<td>-0.31</td>
</tr>
<tr>
<td>500</td>
<td>3593</td>
<td>1292</td>
<td>1328</td>
<td>-0.41</td>
</tr>
<tr>
<td>600</td>
<td>4341</td>
<td>1470</td>
<td>1539</td>
<td>-0.49</td>
</tr>
<tr>
<td>700</td>
<td>5081</td>
<td>1632</td>
<td>1737</td>
<td>-0.57</td>
</tr>
<tr>
<td>800</td>
<td>5811</td>
<td>1779</td>
<td>1923</td>
<td>-0.64</td>
</tr>
<tr>
<td>900</td>
<td>6530</td>
<td>1914</td>
<td>1098</td>
<td>-0.70</td>
</tr>
<tr>
<td>1000</td>
<td>7237</td>
<td>2039</td>
<td>2262</td>
<td>-0.76</td>
</tr>
</tbody>
</table>
Some of the major papers on studies of IBE are tabulated in Table 6 [53-60]. It must be noted that only two studies have exceeded 1keV in energy, and the Kawabe article [53] did not discuss any electrical studies. The Holloway study [60] also is somewhat specialized in that the Schottky metals were deposited in situ after IBE. Thus there was no oxide present between the substrate and the metal; but the oxide is a very real and important part of Schottky devices. (Nicollian [61] reports that the oxide serves to change I-V's, and can even be conductive.)

In general, the studies given in Table 6 [55-60] showed that $\phi_b$ dropped while $J_{sat}$ and $n$ increased with IBE. Also, arsenic appeared to be preferentially etched at the surface since XPS showed an increase in the Ga/As ratio [60]. Holloway, however, showed this trend to change somewhat in that the 5keV sample appeared arsenic rich. All papers agree that a partly amorphous layer of less than 100Å in thickness is developed, while defects can diffuse into the bulk at room temperature. Pang [57] showed that for a 1000eV ion etch I-V characteristics recovered after a 250Å wet etch, but C-V curves did not recover until 750Å of material was removed. The depth of the diffused damage can be quite deep. Ghandhi reports that an Ar$^+$ IBE at ~100eV produced damage observable at 900Å from the surface. Detectable damage was found as deep as 2000Å by Kawabe using He backscatter and enhanced chemical activity.

Holloway has performed DLTS on his samples and found peaks at .3eV and .6eV below the conduction band edge. He failed to note, however, what traps were present before IBE. This will be shown to be an important distinction in the results of this study.
Table 6. Summary of useful literature of Ar⁺ IBE on GaAs.

<table>
<thead>
<tr>
<th>Energy (eV)</th>
<th>Experiments performed</th>
<th>ref #</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 W plasma</td>
<td>Rutherford backscatter, Raman</td>
<td>53</td>
</tr>
<tr>
<td>100-2000</td>
<td>He backscatter, photoluminescence, chemical activity</td>
<td>54</td>
</tr>
<tr>
<td>500</td>
<td>I-V</td>
<td>55</td>
</tr>
<tr>
<td>100-1000</td>
<td>I-V, C-V</td>
<td>56</td>
</tr>
<tr>
<td>250, 500, 1000</td>
<td>I-V, C-V, XPS, DLTS on RIE only</td>
<td>57</td>
</tr>
<tr>
<td>100</td>
<td>C-V</td>
<td>58</td>
</tr>
<tr>
<td>50W Sputter</td>
<td>I-V</td>
<td>59</td>
</tr>
<tr>
<td>1000-5000</td>
<td>DLTS, I-V, C-V, XPS (UHV Schottky deposition after IBE in situ)</td>
<td>60</td>
</tr>
</tbody>
</table>
J. Feng et al. [62] reports that 4keV bombarded surfaces showed a change in the UV reflectivity toward an amorphous crystal structure. It was also shown by J. Epp using ESCA that low energy ions (0.5keV) did not completely remove the native oxide, while higher energies were As deficient and contained a somewhat distorted oxide (different composition compared to virgin) when exposed to atmosphere [62]. Sen reports capacitance-frequency dispersion with 3keV IBE which indicated deep traps and a possible amorphous layer near the surface. He showed that weak reverse bias and reduced temperatures could eliminate the effect [63].

It can be seen from this brief review that there is a general lack of data concerning electrical properties of IBE samples over 1keV. Trap analysis using DLTS is practically nonexistent. Electrical analysis and modelling of low energy ion-etching, and its effects on GaAs surfaces, are necessary if this technology is to be used effectively for device fabrication [64-67]. It is the intent of this thesis to provide this data along with physical and electrical models quantifying IBE induced damage.

In order to be contributory, this study focused on higher energies (>5keV) and low currents (<100μA) which brought the IBE studied here into the realm of ion cleaning. This does not reduce the importance of this process since dry cleaning techniques are gaining increased popularity and little has been reported about their damaging effects.

The following questions were addressed in this study:

1. How does ion cleaning affect Schottky diode device parameters as
functions of ion energy and fluence?

2. To what depth does electrically detectable damage occur?

3. What is the physical nature of that damage and how does it relate to the electrical measurements?

4. How can the induced damage be modelled?

5. What can be done to repair the residual damage to improve device quality?

Other objectives related to, but not directly associated with, these questions included:

1. Design and construction of an automated electrical test station to increase through-put and reduce the analysis burden.

2. Production of "good" working Schottky diodes starting from the polished wafer.

The methods used to answer these and other questions can be found in Chapter 3.
Chapter 2: Schottky Diodes on GaAs

The Schottky diode was the device chosen for probing crystal and electrical damage due to its simplicity, sensitivity and versatility in detailing electrically related materials parameters. The diode itself consists of two basic contacts, the Schottky or rectifying contact and the ohmic or linear contact. Each is detailed below.

2.1 Schottky Contacts

2.1.1 Ideal

A Schottky or rectifying contact is formed by contacting a metal of a given work function $\phi_M$ to a semiconductor of a given work function $\phi_S$. The resulting band-diagram is given in Fig. 9 for an n-type semiconductor with $\phi_M > \phi_S$. The resulting barrier height looking from the metal to the semiconductor is given as

$$\phi_B = \phi_M - \phi_S$$

Looking from the semiconductor to the metal gives a barrier

$$V_{do} = \phi_M - x_S$$

where $x_S$ is the electron affinity of the semiconductor [68,69]. Table 7 gives a listing of measured barrier heights for various semiconductors and metals.

Unfortunately the picture for real Schottky diodes is not quite as simple as the ideal Schottky contact shown in Fig. 9. Problems such as surface states and oxide layers can completely change Schottky
Figure 9. Band diagram for an ideal Schottky contact to an n-type semiconductor.
Table 7. Common Schottky metals to various semiconductors [taken from Ref. 70].

| Semiconductor | Type | $E_o$ (eV) | Ag | Al | Au | Cr | Cu | Hf | In | Mg | Mo | Ni | Pb | Pd | Pt | Ta | Ti | W |
|---------------|------|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Diamond       | p    | 5.47      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Ge            | n    | 0.66      | 0.54 | 0.48 | 0.59 | 0.52 | 0.64 | 0.49 | 0.38 |    |    |    |    |    |    |    |    |    |    |
| Ge            | p    | 0.50      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Si            | n    | 1.12      | 0.78 | 0.72 | 0.80 | 0.61 | 0.58 | 0.50 | 0.46 | 0.42 | 0.51 | 0.55 | 0.61 | 0.45 |    |    |    |    |    |    |
| Si            | p    | 0.54      | 0.58 | 0.34 | 0.30 | 0.46 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| SiC           | n    | 3.00      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| AlAs          | n    | 2.16      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| AlSb          | p    | 1.63      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| BN            | p    | 7.50      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| BP            | p    | 6.00      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| GaSb          | n    | 0.67      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| GaAs          | n    | 1.42      | 0.88 | 0.80 | 0.90 | 0.82 | 0.72 |    |    |    |    |    |    |    |    |    |    |    |    |    |
| GaAs          | p    | 0.63      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| GaP           | n    | 2.24      | 1.20 | 1.07 | 1.30 | 1.06 | 1.20 | 1.84 | 1.04 | 1.13 | 1.27 | 1.45 |    |    |    |    |    |    |    |
| GaP           | p    | 0.72      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| InSb          | n    | 0.16      | 0.18 | 0.17 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| InAs          | p    | 0.33      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| InP           | n    | 1.29      | 0.54 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| InP           | p    | 0.76      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CdS           | n    | 2.43      | 0.56 | 0.78 | 0.50 |    |    | 0.45 | 0.59 | 0.62 | 1.10 | 0.84 |    |    |    |    |    |    |    |    |
| CdSe          | n    | 1.70      | 0.43 | 0.49 | 0.33 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CdTe          | n    | 0.81      | 0.76 | 0.71 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ZnO           | n    | 3.20      | 0.68 | 0.65 | 0.45 | 0.30 |    |    |    |    | 0.68 | 0.75 | 0.30 |    |    |    |    |    |    |    |
| ZnS           | n    | 3.60      | 1.65 | 0.80 | 2.00 | 1.75 | 1.50 | 0.82 |    |    | 1.87 | 1.84 | 1.10 |    |    |    |    |    |    |    |
| ZnSe          | n    | 1.21      | 0.76 | 1.36 | 1.10 | 0.91 |    |    |    |    | 1.16 | 1.40 |    |    |    |    |    |    |    |    |
| PbO           | n    | 0.95      |    |    |    |    |    |    |    |    |    |    | 0.93 | 0.96 | 0.95 |    |    |    |    |    |

*77 K.*
characteristics. This is especially true with GaAs. It is therefore necessary that these effects be better understood so that a more complete model can be developed for the Schottky contact, and experimental results better understood.

2.1.2 Surface States

All materials have a boundary between themselves and their surroundings which defines the surface of the material. At the surface there are atoms which will have no neighbor to one side and the resulting dangling bond, along with other crystalline defects and surface contaminants, may contribute to the overall surface state distribution. To illustrate the electrical effect of surface states at semiconductor interfaces two cases will be considered, one where the semiconductor is in contact with a metal and the other where the surface is exposed to free space.

Consider the first case as shown in Fig. 10. \( E_o \) marks the surface state neutral level, i.e., if \( E_f \) is above \( E_o \), the surface is acceptor-like (-) while if \( E_f \) is below \( E_o \), the surface is donor-like (+) [71]. Consider an n-type semiconductor with \( E_f \) slightly below \( E_o \). The surface will be positively charged, partially compensating the metal, reducing the space charge in the semiconductor. This reduces the band bending and thus the barrier height. With a high density of states any aberration of \( E_f \) from \( E_o \) would mean massive surface charge build up. The bands will therefore only bend enough to keep \( E_o \) close to \( E_f \), otherwise this large uncompensatable surface charge would result. With the band bending so restricted, the barrier height is effectively
Figure 10. Band diagram of a semiconductor containing surface states (a) donor-like, (b) acceptor-like [taken from Ref. 71].
"pinned" by $E_o$ so that

$$
\phi_B = E_g - E_o
$$

Similarly for the free surface, $E_f$ must be maintained near $E_o$ or else uncompensatable surface charge again results. This forces $E_f$ to be pinned at $E_o$ for the free surface [72].

Clearly, with large surface state densities the barrier height is almost independent of $\phi_M$, completely deleting the ideal model results. Unfortunately GaAs has large enough surface states to have a pinned barrier height regardless of metal type. This is one of the major problems with GaAs and much research is presently dedicated to passivating the GaAs surface in an attempt to unpin the Fermi level [73,74].

2.1.3 Non-ideal

With regard to the surface states and the presence of interfacial oxides on GaAs (~20 Å) (whose electrical effects will be seen shortly) a non-ideal model has been given in Fig. 11. This is the Bardeen model of the real metal-semiconductor interface. From this diagram $\phi_B$ may have some bias dependence due to voltage dropped across the interfacial oxide layer given by

$$
\phi_b = \phi_B - \alpha E_{\text{max}}
$$

with

$$
\alpha = \frac{\delta \varepsilon_S}{\varepsilon_i + q\delta D_S}
$$

where $\varepsilon_S$ is the semiconductor permittivity, $\varepsilon_i$ is the insulator permittivity,
Figure 11. Bardeen model of a non-ideal Schottky contact [taken from Ref. 68].
δ is the insulator thickness and $D_S$ is the density of surface states [75]. This bias dependent barrier height effect is usually small however.

Image force lowering can also decrease the barrier height. The effect is to lower the barrier by

$$\Delta \phi = \left( \frac{qE}{4\pi \varepsilon_S} \right)^{1/2}$$

(6)

where $E$ is the field maximum. The barrier peak occurs at

$$x_m = \left( \frac{9}{16\pi \varepsilon_S E} \right)^{1/2}$$

(7)

where $x_m$ is the location from the surface in cm of the barrier height maximum. The lowering is on the order of a few tenths of a volt for moderate fields [76].

2.1.4 Current-Voltage

Figure 12 shows a schematic of the Schottky contact with the forward biased current mechanisms numbered. The numbers correspond to:

1. Thermionic emission
2. Tunneling
3. Recombination in the space charge layer
4. Hole injection or recombinaton in the neutral region
Figure 12. Schematic indicating current components in forward bias [taken from Ref. 68].
For forward and "near" reverse bias, thermionic emission dominates the other components and is considered the most correct theory of current transport. Sze [3] has developed a combination of diffusion and thermionic emission which will be discussed later.

Thermionic emission currents originate from electrons which gain or have enough thermal energy to surmount the contact barrier. From kinetic theory the number of electrons incident on the barrier is

\[ J_{SM} \text{ or } J_{MS} = \frac{qn\bar{v}}{4} \] (8)

where \( n \) is the electron density and \( \bar{v} \) is their average velocity. For electrons in the semiconductor

\[ n = N_c \exp \left[ -\frac{q(\phi_B - V)}{kT} \right] \] (9)

giving

\[ J_{SM} = q \frac{N_c \bar{v}}{4} \exp(-q\phi_B/kT) \] (10)

At equilibrium and \( V = 0 \) the number of electrons incident from the metal side must be equal to those incident from the semiconductor side. Thus

\[ J_{MS} = \frac{qN_c \bar{v}}{4} \exp(-q\phi_B/kT) \] (11)

The total current, \( J = J_{SM} - J_{MS} \), is given by

\[ J = \frac{qN_c \bar{v}}{4} \exp(-q\phi_B/kT)[\exp(qV/kT) - 1] \] (12)
Considering all of the electron's energy to be kinetic in one direction for a Maxwellian distribution gives

$$ v = \left( \frac{8kT}{\pi m^*} \right)^{1/2} \tag{13} $$

and substituting for $N_c$, the effective density of states in the semiconductor gives, after simplification,

$$ J = A^* T^2 \exp(-\phi_B/kT) \left[ \exp(qV/kT) - 1 \right] \tag{14} $$

where $A^*$ is the Richardson constant given as

$$ A^* = 4\pi m^* q k^2 / h^3 \tag{15} $$

where $m^*$ is the electron effective mass, $q$ the electron charge, $k$ is Boltzmann's constant and $h$ is Planck's constant. With $V > 3kT/q$ the current reduces to

$$ J = J_{sat} \exp(qV/kT) \tag{16} $$

This is very similar to the p-n diode equation with the only difference being that the Schottky diodes have a different and often larger reverse saturation current $J_{sat}$.

If the effect of image force and bias dependency of the barrier height is considered as
\[ \phi_{BE} = \phi_B - \Delta \phi + \beta V \] (17)

where \( \beta \) is the coefficient determining the strength of bias dependency, then after substituting in Eqn. 14 with \( 1/n = 1 - \beta \) and reducing gives

\[ J = J_{sat} \exp(qV/nkT) \] (18)

where \( n \) is called the diode ideality factor and is often used as a measure of diode quality [77].

Sze [3] has developed a hybrid model which considers both thermionic emission and diffusion theory. The resulting Current-Voltage relation is

\[ J = \frac{qN_c v_R}{1 + v_R/v_D} \exp\left(\frac{-q\phi_R}{kT}\right)[\exp\left(\frac{-qV}{kT}\right) - 1] \] (19)

where \( v_R \) is the surface recombination velocity at the interface and \( v_D \) is the effective diffusion velocity for electrons transported from the edge of the space charge layer. If \( v_D > v_R \) than thermionic emission applies, if \( v_R > v_D \) the process is diffusion limited. This paper will consider thermionic emission to dominate since it has been shown to be a more correct description of Schottky diode current flow [68].

Further modification exists for \( A^* \). Considering the possibility of electrons emitted from the semiconductor being scattered back, the Richardson constant may be modified to

\[ A^{**} = \frac{f_p f_n A^*}{1 + f_p f_n v_R/v_D} \] (20)
But since it is assumed that thermionic emission applies \( (v_D > v_R) \), we obtain

\[
A^{**} = f_p f_q A^* \tag{21}
\]

where \( f_p \) represents the fraction of electrons scattered back by phonon interaction in the metal and \( f_q \) the fraction quantum mechanically reflected from the barrier [78].

Reverse breakdown generally occurs via an avalanche process and is usually expressed for a one-sided abrupt junction, (a Schottky can be modeled as a \( p^+n \)), as

\[
V_B = \frac{\varepsilon E_{\text{MAX}}^2}{2qN_D} \tag{22}
\]

where \( V_B \) is the reverse breakdown voltage, \( E_{\text{MAX}} \) the peak field at breakdown and \( N_D \) the dopant density. The breakdown voltage is also dependent upon the carrier mobility in that the velocity the electron attains under the applied field must be large enough for impact ionization to occur. For lightly doped \( n \)-type semiconductors with \( N_D = 5 \times 10^{16}\text{cm}^{-3} \), \( V_B \), as calculated from the above equation, is approximately 11 V [79].

A thin interfacial layer can have many adverse effects on the Current-Voltage characteristics in both forward and reverse bias (see Fig. 13). Since electrons must tunnel through this layer, the current varies with layer thickness for a given applied bias. This leads to a
Figure 13. Effect of thin interfacial layer on I-V characteristics of Schottky diodes [taken from Ref. 80].
softening of the forward and reverse characteristics as shown. Also, part of the voltage is dropped across this layer, which causes a bias dependent barrier height as discussed earlier. This results in non-saturation of the reverse bias current and affects the ideality factor.

Surface states can also affect Current-Voltage characteristics by introducing tunneling states at the interface and by thinning the barrier which also aids in tunneling in both directions. Field crowding can also occur because of uneven charge distribution near the surface causing an early breakdown.

2.1.5 Capacitance-Voltage

For all intensive purposes the Schottky diode may be treated as a one-sided abrupt junction with a space charge layer on the n-side as shown in Fig. 14. Solving the Poisson equation across the space-charge layer results in

\[ C = \left( \frac{q\varepsilon S N_D}{2} \right)^{1/2} \left( V_{bi} + V - kT/q \right)^{-1/2} \]  \hspace{1cm} (23)

This may be related to the parallel plate approximation as

\[ C = \frac{\varepsilon S}{W} \]  \hspace{1cm} (24)

where \( W \) is the space-charge width. Solving for \( W \) using the above two equations gives [81]

\[ W = \left( \frac{2\varepsilon (V_{bi} + V)}{qN_D} \right)^{1/2} \]  \hspace{1cm} (25)
Figure 14. Schottky contact indicating space charge region [taken from Ref. 81].
An interfacial layer modifies dependence of the space charge in the depletion region on the bias voltage through barrier variations and voltage drops across the layer. These are usually nonlinear effects. Surface states, because of their charged nature, vary the zero bias capacitance and C-V characteristic of a surface-state-free or ideal diode, since space-charge compensation is either reduced or increased depending on the surface charge state and applied voltage affecting $W$.

Deep traps in the bulk of the semiconductor can also have effects on the capacitance of the Schottky diode depending on the trap depth, characteristics and location with respect to the Fermi level. Assume a deep donor in two separate cases. If the trap emission frequency $\omega_T = 1/\tau_T$ is less than the measuring signal frequency $\omega_s$ the traps will not respond fast enough and the measured capacitance will be due to $N_D$ only. If however $\omega_T < \omega_s$ the traps can respond and

$$C = \frac{\varepsilon_s (N_D + N_T)}{N_D W + N_T Y}$$  \hspace{1cm} (26)$$

where $Y$ is the trap width in the space charge layer [82].

Deep traps also vary their occupation and emission rate with temperature, affecting the transient and thermal capacitance behavior of the diode. This is exploited in the Deep Level Transient Spectroscopy (DLTS) measurement which will be explained more fully in the next chapter.
2.2 Ohmic Contacts

Ohmic contacts are linear in that they generally obey Ohm's Law and may, in general, be formed in two ways to semiconductors. One way, for n-type semiconductors, is to choose a metal with a work function less than the semiconductor work function ($\phi_M < \phi_S$). The resulting band diagram when the two are joined is shown in Fig. 15 where it can be seen that no effective barrier exists looking from either the semiconductor to the metal or from the metal to the semiconductor. This results in ohmic behavior [83]. However, as was seen earlier, GaAs tends to have a somewhat uncontrollable oxide layer and surface state distribution, effectively pinning the semiconductor Fermi level. Thus regardless of the metal work function a rectifying contact will most likely result, making this method of ohmic contact formation to GaAs ineffective.

Another method is to form a so-called tunnel contact as seen in the band diagram given in Fig. 16, which has been shown to give ohmic behavior. This is indeed the common contact method used in the GaAs industry, and in this research. Table 8 illustrates a variety of metal systems used to form ohmic contacts in this manner. Au-Ge-Ni is by far the most popular system used to date and is the ohmic contact system that was used in this study. (This technique was also used to replicate the Texas Instruments diodes.)

Fabrication takes place in several steps [85], as follows

1. An 88:12 Au:Ge alloy is deposited.
2. A Ni cap is deposited on top of the Au:Ge.
3. The contact is annealed in forming gas (95% N₂:5% H₂) for several minutes at ~440°C.
Figure 15. Typical ohmic contact for $\phi_m < \phi_S$ for an n-type semiconductor.
Figure 16. Tunnel type ohmic contact [taken from Ref. 83].
Table 8. Common ohmic metal systems used on GaAs [taken from Ref. 84].

<table>
<thead>
<tr>
<th>n-Type GaAs</th>
<th>Metal System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ag-In</td>
<td>75% Ag, 25% In; sinter 500°C</td>
</tr>
<tr>
<td>Ag-In-Ge</td>
<td>90% Ag, 5% In, 5% Ge; alloy 600°C</td>
</tr>
<tr>
<td>Ag-Sn</td>
<td>98% Ag, 2% Sn; alloy 550–650°C</td>
</tr>
<tr>
<td></td>
<td>33% Ag, 67% Sn; alloy</td>
</tr>
<tr>
<td>Au-Ge</td>
<td>12% Ge (alloy); alloy 450°C</td>
</tr>
<tr>
<td></td>
<td>Au overlayer</td>
</tr>
<tr>
<td>Au-Ge-Ni</td>
<td>12% Ge (alloy), Ni; alloy 480°C</td>
</tr>
<tr>
<td>Au-Ge-In</td>
<td>Ni plated to In</td>
</tr>
<tr>
<td>Au-Si</td>
<td>alloy 425°C</td>
</tr>
<tr>
<td>Au-Sn</td>
<td>20% Sn; alloy 450°C</td>
</tr>
<tr>
<td>Au-Sn-Ni-Au</td>
<td>alloy 300°C</td>
</tr>
<tr>
<td>Au-Te</td>
<td>10% Te; laser alloy</td>
</tr>
<tr>
<td></td>
<td>2% Te; alloy 500°C</td>
</tr>
<tr>
<td>In</td>
<td>300°C melt</td>
</tr>
<tr>
<td>In-Al</td>
<td>alloy 320°C</td>
</tr>
<tr>
<td>In-Au</td>
<td>90% In; alloy 550°C</td>
</tr>
<tr>
<td>In-Ni</td>
<td>Ni plated to In</td>
</tr>
<tr>
<td>Pd-Ge</td>
<td>sinter 500°C for two hours</td>
</tr>
<tr>
<td>Sn-Ni</td>
<td>Ni plated to Sn; alloyed</td>
</tr>
<tr>
<td>Sn-Sb</td>
<td>4% Sb; alloy 300–350°C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p-Type GaAs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ag-In-Zn</td>
</tr>
<tr>
<td>Ag-Zn</td>
</tr>
<tr>
<td>Au-Be</td>
</tr>
<tr>
<td>Au-Zn</td>
</tr>
<tr>
<td>In-Zn</td>
</tr>
</tbody>
</table>
The anneal step drives the Ge into the near surface of the GaAs, heavily doping it. This forms the narrow, easily tunneled-through barrier, as illustrated earlier. The role of the Ni is less clear however, but it is known that it acts as a wetting agent for the Au during the anneal, so that the Au contact does not ball-up and loose adhesion. It also appears that the Ni can act as a diffusion barrier to Ga, which is gettered by the Au during the anneal. The contact resistance is very sensitive to alloy temperature and time, and some success has been made improving the contact resistance and stability by depositing a thin (~10Å) Ni layer before Au:Ge [86].

Contact resistance, tunneling probability, reproducibility and reliability of ohmic contacts are subjects of much research and analysis which will not be taken up or presented here. It will be assumed that a small change in contact resistance will not affect the measured results of IBE damage and in fact can be neglected since ohmic contacts were made before IBE was done. The actual processing of ohmic and Schottky contacts for this research is discussed in the next chapter.
Chapter 3: Experimental Methods

This chapter details the methods used to study the effects of IBE. The fabrication process is discussed in detail along with the measurement system and the parameters that are derivable from electrical measurement. Also described are the experimental groups into which the samples were divided for study. DLTS is described in some detail as well.

3.1 Schottky Diode Fabrication

As mentioned earlier, the Schottky diode was chosen in this study due to its sensitivity to surface effects, and its relatively simple fabrication. What follows is a detailed description of the fabrication process used to make diodes for experiment.

Wafers were initially purchased from Airton Inc. but were later obtained from Morgan Semiconductor, since Texas Instruments uses Morgan as a primary wafer vendor. They were standard 2" diameter LEC GaAs wafers 15mils thick, doped n-type in the melt to ~5 x 10^{16} cm^{-3} with Si, and cut 2° off (100). Electron mobility was on the order of 4200 cm^2/V·s as given in the manufacturer's data sheet provided by Morgan. Wafers were received chemically and mechanically polished on one side for device fabrication. The samples were initially cut from the wafer to .5" x .5" squares and marked for crystal orientation as shown in Fig. 17, using a diamond scribe. After cutting, the samples were subjected to the cleaning, metalization and packaging process given in Fig. 18 [87].
Figure 17. Diagram of as received wafer indicating (a) crystal orientations, (b) cut pieces and (c) IBE angles.
SAMPLE #:  
WAFER #:  

PROCESSING (etch, cleaning, anneal, metals, specia, . . . . . etc. / date):

1. ( / ) 10 min. in hot tetrachloroethylene
2. ( / ) 10 min. in hot methanol
3. ( / ) hot HCl dip : D.I. rinse
4. ( / ) 4 min. in 8:1:1 (H₂SO₄:H₂O₂:H₂O) : D.I. rinse : dry
5. ( / ) Au/Ge(88%/12%) evaporation about 2000Å at slow rate
6. ( / ) Ni evaporation about 700Å
7. ( / ) ohmic anneal 2 min. at 440°C in forming gas
8. ( / ) 4 min. 8:1:1 or until white haze subsides : D.I. rinse and dry
9. ( / ) 10 min. HCl : D.I. rinse and dry
10. ( / ) Ar⁺ ion etch at x keV, x Amps/cm², x min. : store in argon atmosphere when complete
11. ( / ) Al Schottky evaporation about 2000Å at slow rate
12. ( / ) scribe cut diodes : mount in TO-8 package with 1 mil Au wire and silver epoxy : cure in oven at 65°C for 3hrs

DEVICE/pinout: pin 1=Al Schottky, pin 2=Au/Ge ohmic

COMMENTS: wafer marked for orientation

Figure 18. Schottky diode processing flowchart.
The first step involved a cleaning and degreasing process to remove any fingerprints and dirt present from shipping and handling. This involved 10 minutes in a hot trichloroethylene bath followed by a 10 minute hot methyl alcohol wash.

Samples were then rinsed in deionized water and dipped briefly in hot (~40°C) 10% HCl solution. This was done to clean and remove any oxides (Ga₂O₃, As₂O₃, etc.) present; a DI rinse followed.

As suggested by Texas Instruments, the cleaned as-received surface was not used for fabrication. Instead this surface was exposed for 4 minutes to an 8:1:1 H₂SO₄:H₂O₂:H₂O chemical etch which removes GaAs at approximately 1μm/min. This yielded clean surfaces with ~4μm removed which were then DI rinsed and surface-tension-pulled dry.

Four samples, processed in parallel, were then placed in an Al sample jig which was fitted with two stainless steel masks consisting of 4 evaporation holes and 2 alignment holes (Fig. 19). The masks were arranged so that ohmic contacts were evaporated using holes 1/16" in diameter. Schottky contacts were evaporated using the bottom alignment holes on the 1/32" dot diameter mask. This produced 4 diodes per piece or 16 diodes per run.

The ohmic contacts were produced by placing the above jig into a Denton Vacuum evaporator, approximately 9" above two tungsten filaments containing an 88:12 Au:Ge slug of approximately 5 grams in one boat, and 99.999% pure Ni in the other. (Both metals were purchased from Materials Research Corporation.) After achieving a vacuum of 6μtorr, approximately 2000Å of Au:Ge was deposited, as measured by a Kronos Model QM-311 thickness monitor. Under the same vacuum, 700Å of Ni was next
Figure 19. Evaporation jig and mask arrangements for depositing ohmic and Schottky contacts.
was next deposited on top of the Au:Ge. The chamber was then vented with dry N₂ and the samples annealed at 440°C for 2 minutes in forming gas (95% N₂:5% H₂) (This is the Ge drive-in step explained in Chapter 2).

It was found that Schottky contacts deposited on these annealed surfaces produced inferior, unreproducible diodes. Therefore a second 4 min. 8:1:1 H₂SO₄:H₂O₂:H₂O etch was incorporated to remove any possible surface damage introduced by the anneal. Upon exposure to the etch a white, "milky" coating appeared on the sample surfaces. This is called the "white haze" in industrial circles and is possibly the result of micro-surface cracks exposed by the etch [88]. After sufficient time (~5 min.) all surfaces recovered to their near optically clean appearance.

The samples were next cleaned in 1:1 HCl for 10 min. and placed in a Perkin-Elmer 5300 XPS system for Ar⁺ etching. Energies between .5 keV and 3 keV were used with current densities correspondingly between 0.5 μA/cm² and 60 μA/cm². Ion etching took place along the (001) direction at 50° off normal, as oriented by an "X" on the sample back (refer to Fig. 17). Fluences of 10¹⁴ cm⁻² and 10¹⁶ cm⁻² were used, with fluence defined as fluence = current density • etch time. ESCA measurements sometimes followed the IBE, and such results will be presented where appropriate.

Al or Au Schottky contacts were then deposited using the 1/32" dot mask and jig described earlier. After approximately 2000Å of Schottky metal deposition, the samples were electrically tested for quality using a Tektronix 575 Transistor-Curve Tracer after which good individual
diodes were scribed and cut from the larger pieces and mounted into a TO-8 package with Si based thermal grease. Electrical contact was made using 1mil Au wire and room temperature cured thermal epoxy (ACME E-Solder). A resulting device is illustrated in Fig. 20. The Schottky diodes were then ready for electrical test.

3.2 Automated Electrical Analysis (MEDUSA)

Due to the large number of samples and measurements necessary to study this subject and the requirement that data be reproducible, an automated electrical test station was designed and constructed. This station was given the name MEDUSA for Materials and Electronic Device Unified System Analyzer.

3.2.1 Hardware

The hardware layout is given in Fig. 21. All instruments including the magnet (through an optical power relay) are connected to the IEEE 488 bus controlled by the IBM-AT using a Scientific Solutions board. Each piece of equipment performed several tasks:

HP 4280A: (Capacitance Bridge) C,G; C,G-V; C,G-t; DLTS (coupled with HP 8112A pulse generator)

HP 4140B: (Current Meter and Voltage Source) I-V, C-V

Keithley 195A DVM: Van de Pauw (coupled with 4140B, 4280A, Varian Magnet), 4-pt. Resistivity (coupled with 4140B)
Figure 20. TO-8 package and attached diode.
Figure 21. MEDUSA hardware layout.
3.2.2 Software

The software consists of three separate levels: (1) Parameter entry and error check, (2) Experimental run and data collection, (3) Data processing and graphical display. (Flowcharts and the actual software are given in Appendix A.) Graphs derivable from the system include:

1. Hand Graph Entry
2. Capacitance vs. Time
3. Conductance vs. Time
4. Capacitance vs. Temperature
5. Conductance vs. Voltage
6. Capacitance vs. Temperature (TSCAP)
7. Conductance vs. Temperature
8. $1/C^2$ VS. Voltage
9. Dopant Profile
10. Barrier vs. Temperature
11. DLTS Spectrum
12. Current vs. Voltage
13. Log (I) vs. Voltage
14. Ln(I) vs. $qV/kT$
15. Ln($I_f/T^2$) vs. $q/kT$
16. $I_{sat}$ vs. Temperature
17. Ideality Factor vs. Temperature
18. Resistivity vs. Temperature
19. Mobility vs. Temperature
20. Carrier Concentration vs. Temperature
21. Activation Z vs. Temperature
22. 1/C³ vs. Voltage
23. 4-point Resistivity vs. Temperature
24. Ln(I_{sat}) vs. Temperature

3.3 Electrical Measurements

It is appropriate at this point to discuss how some of the above graphs produce vital data characterizing the diode. The major parameters to be considered for the diodes were \( \phi_b, n, I_{sat}, N_D(x), E_{Traps(E_T)}, N_{Traps(N_T)}, V_{Turn-on(V_T)}, V_{breakdown(V_B)}, G \) and \( C \). Table 9 gives the parameter and graphs from which they are derived \[89\]. Mobility measurements were not done on these samples since any surface damage would be shunted by the low resistivity bulk. The trap parameters were derived from the DLTS measurement, which deserves further explanation (along with TSCAP).

DLTS (Deep Level Transient Spectroscopy) was a major part of this research, and the principles by which it works need further clarification. Essentially DLTS is performed by reverse biasing a diode and pulsing it to zero bias, after which the capacitance is monitored as a function of time at a given temperature. During the pulse, the traps in the reduced space charge fill, changing their charge state, thus effecting the capacitance. These trapped electrons are thermally released exponentially at a rate dependent upon the sample temperature after the bias pulse has subsided. The thermal release rate for electrons is given by
Table 9. Schottky diode parameters and their graphical derivations.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Graphical Derivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_b$</td>
<td>x-intercept of $1/C^2$ vs. $V$, slope of $\ln(I, /T^2)$ vs. $q/kT$ (activation energy, $E_a = V, /n$-slope)</td>
</tr>
<tr>
<td>$n$</td>
<td>slope of $\ln(I)$ vs. $qV/kT$ ($n=1/slope$)</td>
</tr>
<tr>
<td>$I_{Sat}$</td>
<td>y-intercept of $\ln(I)$ vs. $qV/kT$</td>
</tr>
<tr>
<td>$V_B$</td>
<td>45° slope point of reverse $I-V$</td>
</tr>
<tr>
<td>$V_T$</td>
<td>extrapolated x-intercept of $I-V$</td>
</tr>
<tr>
<td>$N_D(x)$</td>
<td>$2/qE[-1/slope of 1/C^2]$ with $W$ calculated from depletion width</td>
</tr>
<tr>
<td>$E_T$</td>
<td>$23.7kT,...$ of DLTS spectrum</td>
</tr>
<tr>
<td>$N_T$</td>
<td>$2[\delta C/CJN]$, where $\delta C$ comes from DLTS peaks in spectrum</td>
</tr>
</tbody>
</table>
where \( v \) is the release frequency. By sweeping the sample temperature, measuring the capacitance vs. time after pulse biasing, and looking for a specific rate of decay, (rate window), a DLTS spectrum is obtained. This process is illustrated in Fig. 22.

The spectrum is a plot of the magnitude of the capacitive transient versus temperature, where majority carrier traps are indicated by negative peaks in the spectrum. The magnitude of the peaks may be used to determine the trap density through the relation

\[
\frac{\Delta C}{C} = - \frac{N_T}{2N_D} \left[ 1 - \frac{2\lambda}{W(V)} \left( \frac{C(V)}{C(0)} - \frac{C(V)}{C(0)} \right)^2 \right]
\]

(28)

where \( \lambda \) is the so-called edge region thickness, \( W \) the space charge thickness, \( N_T \) the trap density, \( C \) the reverse biased capacitance and \( \Delta C \) the transient capacitance given by

\[
\Delta C = C(t_1) - C(t_2)
\]

(29)

where \( t_1 \) and \( t_2 \) are the boxcar integrator times. If the edge region is neglected then the trap density \( N_T \) can be found from the relation

\[
N_T = 2N_D \left| \frac{\Delta C}{C} \right|
\]

(30)

assuming \( \Delta C \ll C \). DLTS is much more sensitive to traps located near the
Figure 22. DLTS pulse biasing and electron trapping process [taken from Ref. 90].
edge of the space charge layer, as can be seen in the relation

\[ \frac{\Delta C}{C} \cdot x = -\frac{n(x)}{N_D \omega^2} x \Delta x \quad (31) \]

The trap energy may be determined using an Arrhenius plot of log of the transient decay rate versus 1/T. This is implemented in DLTS by scanning the rate window which in turn changes the peak temperature accordingly. A simpler approximate method is to fix the rate window at 50 s⁻¹, giving the relation

\[ E_T = 23.7 \, kT_M \quad (32) \]

where \( v = 10^{12} \, s^{-1} \) and \( T_M \) is the peak temperature. The above calculated energies are accurate to ±10% of their actual values.

The rate window is implemented in the computer by simulating a double boxcar integrator set-up where a capacitance sample is taken at time \( t_1 \) and again later at \( t_2 \). The DLTS spectrum will then peak when

\[ \tau_M = (t_1 - t_2) / \ln(t_1/t_2) \quad (33) \]

This relation was used by fixing \( t_1 \) and \( t_2 \) so that \( 1/\tau_M = 50 \, s^{-1} \), as required for Eqn. 32.

Trap scattering crosssections, \( \sigma_e \) and \( \sigma_h \), may be found by observing peak magnitude changes due to pulse bias width variations, but were not done in this study.

The thermally scanned capacitance, TSCAP, was done in parallel with
the DLTS as a cross-check. During a thermal scan, a step can be observed in the capacitance when a trap goes from being mostly full to mostly empty. The trap energy may be calculated using the relation

\[ E_T = 30.7 \, kT_{\text{step}} \]  

(34)

where 30.7 has been linearly interpolated from Lang [90] and \( T_{\text{step}} \) is the mid-point of the capacitance transition. However TSCAP is not as sensitive to deep traps as DLTS since the trap step may be superimposed on a large slope, and since it is not a differential measurement, the capacitance change may be difficult to detect.

3.4 Sample Grouping

Having discussed sample preparation and the electrical test procedures it is necessary to explain how these were applied in this study to deduce electrical effects of IBE. The samples were broken up into the following groups:

1. Test to indicate differences in wafer vendors, and metals used for Schottky contact.
2. IBE with energy between 0.5keV and 3keV at constant fluence.
3. IBE at 0.5keV and 3keV at two different fluences.
4. Chemical etch after IBE and before and after Schottky deposition at 0.5keV and 3keV constant fluence.
5. Anneal 0.5keV and 3keV samples before Schottky deposition.

Group 1 was a catch all for other sundry variables which might
influence the main diode studies. Things such as wafer and Schottky metal variations were looked at briefly.

Group 2 was meant to determine the effects of energy on the diodes. All diodes were fabricated directly on top of the IBE surface with fluence held constant during bombardment so that only the effects of beam energy would be seen.

Group 3 was done to discriminate the effects of ion fluence at both the lowest and highest ion energies used in Group 1. It must be noted here that the fluence was studied, meaning beam currents and exposure times varied between the 0.5keV and 3keV samples.

Group 4 was used for two major studies. The first was to determine the depth of the IBE damage by successively building diodes on top of chemically reduced surfaces. The second study was meant to decouple the role of surface damage between contact damage by chemically removing the region between the ohmic and leaving damage under the Schottky contact only, neglecting undercutting.

Group 5 was done to see if damage could be repaired by a brief, low temperature (350°C) thermal anneal. Recovery could indicate the degree of amorphicity and the role of any surface oxides.

As experiments proceeded, a few side tests were performed as an aid in understanding the data. These tests and their results will be mentioned in the next Chapter where appropriate.
Chapter 4: Results and Discussion

The results of the Group 1 experiments are reported first, since chronologically they were performed first. Also the results of this group helped to standardize the rest of the measurements. Each table and curve represents the most typical results obtained for each sample treatment described with samples starting from the whole wafer. At least two pieces were tested for each experiment (reproducibility of results confirmed) ranging up to 8 pieces for some studies.

4.1 Vendor and Metal

As mentioned earlier, wafers from Airtron were used initially to fabricate Schottkys. Later, Morgan wafers were used in an attempt to standardize with Texas Instruments. Differences were seen immediately as the diodes were processed, in that the Airtron wafers did not require the post anneal etch, as did the Morgan. After adjusting the process, both wafers showed the same basic diode parameters $n$ and $I_{sat}$ while the breakdown voltage was less for the Morgan, due to approximately an order of magnitude higher dopant density (Airtron $\sim 3 \times 10^{15} \text{cm}^{-3}$, Morgan $\sim 5 \times 10^{16} \text{cm}^{-3}$).

The most striking difference appeared in the DLTS spectra comparison shown in Fig. 23. The Airtron wafers contained 5 distinct peaks, indicating 5 deep traps. The Morgan wafers, however, contained only one trap, EL2. (TSCAP confirmed the results of the DLTS.) The energies and densities are given in Table 10. Clearly the Morgan wafers were the better choice for study since any change in the spectrum due to IBE would be immediately detectable as well as exclusively attributable to IBE damage, and the sensitivity would be highest, since any other peaks
Figure 23. Comparison of DLTS spectra done on diodes made on wafers from Airtron and Morgan.
Table 10. DLTS trap parameters for Airtron and Morgan wafers.

<table>
<thead>
<tr>
<th>Wafer</th>
<th>Trap#</th>
<th>Name</th>
<th>$E_T$(eV)</th>
<th>$N_T$(cm$^{-3}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Morgan</td>
<td>1</td>
<td>EL2</td>
<td>.77</td>
<td>2*10$^{15}$</td>
</tr>
<tr>
<td>Airtron</td>
<td>1</td>
<td>EL2</td>
<td>.72</td>
<td>5*10$^{15}$</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>EL3</td>
<td>.54</td>
<td>3*10$^{14}$</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>EL6</td>
<td>.34</td>
<td>1*10$^{15}$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>EL8</td>
<td>.30</td>
<td>3*10$^{14}$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>EL14</td>
<td>.23</td>
<td>6*10$^{13}$</td>
</tr>
</tbody>
</table>
would not be buried in those found in the Airtron. The Airtron wafers
were, however, used briefly for some IBE damage studies, and their
results are given in the next section.

Both Al and Au Schottky contacts were used initially, and both
yielded identical Current-Voltage characteristics and DLTS spectra.
Stability was good through 400K and no apparent metal GaAs reactions
were evident since I-V characteristics were linear in $\ln(I) \ vs. \ V$
plots, even though some workers report AlAs formation and Au
diffusion. Both the Au and Al gave similar results on the IBE treated
samples. For these reasons and project economics, Al was the Schottky
metal chosen for further studies.

Other sundry trials were performed on silver epoxies and Schottky
deposition rates. Clean DLTS spectra and I-V characteristics showed
that the ACME E·solder room temperature cure epoxy worked best, along
with a slow initial Schottky deposition rate (~1Å/sec) for the
first 100Å.

4.2 Energy

Al Schottky contacts were deposited on GaAs surfaces ion beam
etched according to the schedule given in Table 11. The Current-Voltage
characteristics are given in Fig. 24 with the accompanying diode param-
eters in Table 12. Several effects are immediately evident.

The diode barrier height, $\phi_B$, dropped at 0.5keV ion energy, and
increased slightly with IBE energy. This is also seen in the literature
for plasma etching and RIE [91]. Since it is assumed that the barriers
are pinned by the surface states in GaAs, this may indicate a shift in
Table 11. Energy and fluence ion etching schedule.

<table>
<thead>
<tr>
<th>Fluence (cm(^{-2}))</th>
<th>Energy (keV)</th>
<th>Current density ((\mu A/cm^2))</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>(10^{16})</td>
<td>0.5</td>
<td>.28</td>
<td>57s</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>1.00</td>
<td>16s</td>
</tr>
<tr>
<td>(10^{18})</td>
<td>0.5</td>
<td>.30</td>
<td>89s</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>1.20</td>
<td>22s</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>14.00</td>
<td>115s</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>50.00</td>
<td>32s</td>
</tr>
</tbody>
</table>
Figure 24. Current-Voltage curves for ion energy comparison.
(Inset shows $\ln(I)$ vs. $V$ of the forward biased diodes.)
Table 12. Diode parameters for different ion energies.

<table>
<thead>
<tr>
<th>Ion Energy (keV)</th>
<th>$E_n$ (eV)</th>
<th>n</th>
<th>$I_{son}$ (A)</th>
<th>$V_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.92</td>
<td>1.3</td>
<td>$1.8 \times 10^{-10}$</td>
<td>11.7</td>
</tr>
<tr>
<td>0.5</td>
<td>0.62</td>
<td>1.8</td>
<td>$3.5 \times 10^{-8}$</td>
<td>8.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.70</td>
<td>5.1</td>
<td>$2.8 \times 10^{-7}$</td>
<td>11.1</td>
</tr>
<tr>
<td>2.0</td>
<td>0.71</td>
<td>6.8</td>
<td>$1.6 \times 10^{-7}$</td>
<td>15.6</td>
</tr>
<tr>
<td>3.0</td>
<td>0.71</td>
<td>6.8</td>
<td>$7.5 \times 10^{-7}$</td>
<td>14.3</td>
</tr>
</tbody>
</table>
the surface state distribution. ESCA showed the 0.5keV etched surface to be slightly As rich, while at higher energies it became Ga rich or As depleted [92]. Clearly a lack of stoichiometry near the surface is an indicator that surface and near surface defects may exist.

The presence of these near surface defects was also evidenced by capacitance dispersion with frequency for IBE treated samples, as reported by S. Sen. He reports that as IBE energy increased the value of C increased to the limit of the bridge, while frequency decreased, indicating deep, slow defects in the near surface region [93].

The reverse saturation current, \( I_{\text{sat}} \), increased dramatically with IBE. At 0.5keV there were approximately 4 orders of magnitude increase from the virgin diodes, but as ion energy increased \( I_{\text{sat}} \) decreased slightly. This indicates that the diodes had become extremely leaky, also reported in the literature, and this may be related to surface defects which could be providing tunneling/hopping states through the barrier [94]. Surface shunting is another possibility through a conductive layer between the contacts since ESCA shows that at 0.5keV the surface oxide is not removed completely, and Nicollian reports that the oxide can be conductive [95].

Ideality factors increased with IBE energy indicating the presence of a possible insulating layer between the metal and bulk GaAs. This layer could take the form of an oxide or perhaps partly amorphous layer, which has been reported at higher ion energies using UV reflectivity [96]. For the 0.5keV diodes, there was a slight decrease in the turn-on voltage, while at ion energies greater than 1keV, forward turn-on increased until at 3keV no real turn-on could be observed. This is also
indicative of an insulating layer which could be of higher resistance, as indicated in Chapter 2.

Similarly the breakdown voltage, \( V_B \), first decreased at 0.5keV, then increased again at higher ion energies. This is contradictory to published results which show almost a total loss of \( V_B \). In fact it was proposed that the introduction of tunneling states mentioned earlier could be used to form ohmic contacts by reducing \( V_B \) to zero [97]. The increased \( V_B \) at higher energies could be explained by the presence of a lower mobility region which increases \( V_B \) by increasing the field maximum necessary for impact ionization velocity to be attained, while at lower IBE energies (<0.5keV) near virgin mobility, coupled with increased carrier density, could decrease \( V_B \).

Capacitance-voltage measurements showed good linearity for \( 1/C^2 \) vs. \( V \) plots, but the \( \phi_B \) values were unusually high. The dopant profiles remained flat through the measurable thickness. An example is given in Fig. 25. The average dopant density was approximately \( 4 \times 10^{16} \text{cm}^{-3} \), which is within 10% of the manufacturer's data sheet.

At higher energies the room temperature capacitance decreased slightly at both 0V and -4V bias, but was not always consistent. As temperature was swept the capacitance showed a typical increase, with only the virgin sample showing the definite presence of a trap step, i.e. TSCAP showed no trap structure for all IBE diodes.

The conductance vs. temperature is plotted in Fig. 26. The virgin piece is relatively flat through 400K while the 0.5keV sample showed a sharp increase after 300K. As energy increased past 1keV the conductance again dropped. The 0.5keV increased so much that it appeared as a
Figure 25. Typical C-V dopant profile for IBE samples.
Figure 26. Conductance-Temperature comparison of different IBE energies with the Airtron sample included for further comparison.
short to the HP 4280A, thus killing the G and C measurement. This is possibly attributable to the surface states acting as generation-recombination centers, increasing carrier concentration with temperature and creating a shunt path. At higher energies the mobilities may be sufficiently decreased to negate the effects of the increased carrier concentration to some extent.

The most interesting results were seen in the DLTS measurements shown in Fig. 27. (All spectra had a tendency of being somewhat noisy but not unreadable.) The virgin sample showed an EL2 peak at $E_T = 0.77\text{eV}$ and $N_T = 4 \times 10^{15}\text{cm}^{-3}$. At 0.5 keV the EL2 peak was completely removed and a trap was seen at $0.53\text{eV}$ and $1.5 \times 10^{14}\text{cm}^{-3}$ density. For IBE at 1 keV, EL2 was still missing and traps were seen at $0.55\text{eV}$ and $0.53\text{eV}$ with densities of $2.7 \times 10^{14}\text{cm}^{-3}$ and $1.7 \times 10^{14}\text{cm}^{-3}$. At 2 keV, EL2 was seen slightly ($\sim 2.9 \times 10^{14}$) with other traps given by $0.52\text{eV}$, $0.46\text{eV}$, $0.59\text{eV}$ and $0.54\text{eV}$, with densities averaging $5 \times 10^{14}$. Finally at 3 keV, there was again some EL2 present but still small compared with other traps at $0.58\text{eV}$ and $0.49\text{eV}$ and densities of $8 \times 10^{14}\text{cm}^{-3}$. This loss of EL2 was seen in the Airtron samples as well, and is shown in Fig. 28. Thus the loss of EL2 was not unique to the Morgan wafers, and can be considered a general result of IBE. This does not indicate that EL2 is physically depleted, loss of As$_{Ga}$, since the the 0.5 keV samples showed As richness with no EL2. However, the DLTS and TSCAP loss of EL2 is a real effect. If it is assumed that the diode capacitance obeys a parallel plate capacitor model, the depths of DLTS study were approximately: 1790Å–4140Å for the virgin, 1530Å–4010Å for 0.5 keV, 1540Å–4040Å for 2 keV, and 3500Å–5410Å for 3 keV.
Figure 27. DLTS spectra comparing different IBE energies.
Figure 28. DLTS spectra showing loss of EL2 in 3keV IBE energy Airtron sample as compared to virgin material.
From these depths it might be interpreted that the damage and EL2 loss were reaching as deep as 1500Å into the sample, even at 0.5keV. This, however, is highly unlikely when reference is made to the ion implant tables (Table 5) and was explored further in the chemical etch study (discussed later).

Another factor which could be affecting the DLTS is the increased diode conductivity with temperature. It has already been seen that the HP 4280A is effectively shorted for the 0.5keV bombarded sample, and the capacitance measurement drops accordingly. This question will be further pursued in a later section.

The following is a summary of the important results of the energy study:

1. Diode parameters degraded rapidly with IBE. The 0.5keV GaAs diode appeared leakiest, with decreased turn-on and breakdown voltages. The 3keV samples had slightly lower (less than 0.5keV) leakage ($I_{sat}$) with increased turn-on and breakdown voltages. Ideality factors increased and saturated above 2keV while barrier heights, $\Phi_B$, decreased, in agreement with reports.

2. Dopant profiles were unaffected.

3. Zero and reversed bias capacitance decreased slightly with IBE.

4. Diode conductance showed a strong temperature increase for 0.5keV diodes, while the effect was less strong for >1keV IBE GaAs diodes.

5. TSCAP and DLTS indicated a total loss of EL2 for 0.5keV and 1keV samples, while some EL2 signal (almost unmeasurable) was seen in 2keV and 3keV diodes. Other electron traps were introduced into the spectra, but the densities were small compared to virgin EL2
6. Depth analysis, assuming a parallel plate capacitor model, indicated that IBE was affecting the GaAs down to at least 1500Å, even for the 0.5keV IBE GaAs samples.

Possible reasons for the above results have been discussed and further analysis in the thesis (below) will help clarify them.

4.3 Fluence

To study the effects of ion fluence on IBE damage, samples at the low (0.5keV) and high (3keV) energies were subjected to two different fluences, $10^{14}$ions·cm$^{-2}$ and $10^{16}$ions·cm$^{-2}$ before Al Schottky deposition. Diode parameters for this IBE schedule are given in Table 13.

At 0.5keV there was a clear threshold where damage had occurred. A fluence of $10^{14}$ions·cm$^{-2}$ appeared to do no real damage to current-voltage characteristics in that $\phi_B$, $I_{sat}$, $n$, $V_B$, $V_T$ showed no degradation. In fact, the I-V's given in Fig. 29 appear to be sharpened or improved over the virgin case. However, after $10^{16}$ions·cm$^{-2}$, as reported earlier, I-V damage was evident in the extreme increase in $I_{sat}$ and the decrease in both $V_B$ and $V_T$. Clearly, the fluence is an important factor in IBE at this energy for I-V characteristics.

Conductance versus temperature characteristics are given in Fig. 30. Again at $10^{14}$ion·cm$^{-2}$ fluence there was no perceptible change, indicating, together with the I-V parameters, that the supposed surface state changes had not as yet occurred. Thus there was no sharp temperature dependence in the conductivity.
<table>
<thead>
<tr>
<th>Ion Energy</th>
<th>Fluence (ions cm$^{-2}$)</th>
<th>$\phi_B$ (eV)</th>
<th>$n$</th>
<th>$I_{Sat}$ (A)</th>
<th>$V_B$ (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>virgin</td>
<td></td>
<td>92</td>
<td>1.3</td>
<td>$1.8 \times 10^{-10}$</td>
<td>11.7</td>
</tr>
<tr>
<td>.5 keV</td>
<td>$10^{14}$</td>
<td>1.05</td>
<td>1.2</td>
<td>$2.5 \times 10^{-12}$</td>
<td>11.3</td>
</tr>
<tr>
<td></td>
<td>$10^{16}$</td>
<td>.62</td>
<td>1.8</td>
<td>$3.5 \times 10^{-6}$</td>
<td>8.0</td>
</tr>
<tr>
<td>3 keV</td>
<td>$10^{14}$</td>
<td>.67</td>
<td>1.6</td>
<td>$1.5 \times 10^{-8}$</td>
<td>12.0</td>
</tr>
<tr>
<td></td>
<td>$10^{16}$</td>
<td>.70</td>
<td>5.4</td>
<td>$7.5 \times 10^{-7}$</td>
<td>14.3</td>
</tr>
</tbody>
</table>

Table 13. Diode parameters for different fluences.
Figure 29. Current-Voltage characteristics for 0.5keV energy with variation of fluence.
Figure 30. Conductance-Temperature characteristics for 0.5keV energy with variation of fluence.
The above is borne out in the DLTS spectra shown in Fig. 31. Note that at $10^{14}$ ions cm$^{-2}$ there was no change in the EL2 peak. Therefore it appears that at low fluence and low energy there is no perceptable damage.

This, however, was not the case for the 3keV samples; refer to Table 12 and Fig. 32. The diode parameters indicate clear damage at $10^{14}$ ions cm$^{-2}$, and a considerably decreased $\phi_B$. In fact, the barrier was slightly lower than for the $10^{16}$ ions cm$^{-2}$ samples. $I_{sat}$ showed a 3-order of magnitude increase at $10^{14}$ over virgin samples, and continued to increase another order of magnitude at $10^{16}$, while $n$ increased only slightly at first until it saturated at $10^{16}$. $V_B$ and $V_T$ remained relatively unaffected at $10^{14}$ ions cm$^{-2}$. Conductivities increased with fluence and dopant profiles remained flat.

The DLTS spectra are given in Fig. 33. The virgin spectrum is used as a gauge since EL2 is plainly visible. At $10^{14}$ fluence there did appear to be an EL2 peak with $E_T = 0.79$ eV and $N_T = 6.2 \times 10^{14}$ cm$^{-3}$, with the $10^{16}$ ion cm$^{-2}$ fluence having a very similar spectrum. Thus it appears that all detectable DLTS damage had occurred by $10^{14}$ cm$^{-2}$ fluence at 3keV ion energy.

The results of the fluence study thus indicate that:

1. Low fluence ($10^{14}$) and low energy (.5keV) had no perceptable effect on diodes. However, the effective surface cleaning performed by such a beam was questionable since ESCA indicated the presence of an oxide after IBE. Thus, at low energies, fluence is an important parameter for damage effects.
Figure 31. DLTS spectra for 0.5keV energy, at different ion fluences.
Figure 32. Current-Voltage characteristics for 3keV energy with variation of fluence.
Figure 33. DLTS for spectra for 3keV energy, at different ion fluences.
2. Higher energy (3keV) bombarded GaAs materials were much less sensitive to IBE fluence, since damage was similar at $10^{14}$ and $10^{16}$ fluences.

4.4 Chemical Etch

Previous results have shown the effective depth of the damage induced by IBE to be uncertain since it appeared from DLTS that EL2, a bulk trap, could be completely removed from the crystal with a low energy (0.5keV) surface etch. This appears quite unlikely since damage would have to extend past 1500Å at 0.5keV for this to occur, the ion-implant tables indicating penetration to no more than 45Å at 1keV [98]. Since, however, the IBE surface treatment is affecting the spectra, an electrical depth profiling DLTS was not advised. Instead, a chemical etch removal of the surface was chosen to perform depth profiling of the IBE damage.

To etch the surfaces a weak $\text{H}_{2}\text{SO}_4:\text{H}_2\text{O}_2:\text{H}_2\text{O}$ 1:1:100 (1:1 $\text{H}_2\text{SO}_4$, 30% $\text{H}_2\text{O}_2$) solution was used which has been found to etch crystalline GaAs at 530Å/min, and amorphous GaAs at 660Å/min [99]. Since the nature of the damaged layer was somewhat questionable, it was assumed that the damage in general consisted of a 50% crystalline and 50% amorphous composition, which was dependent on energy. Therefore, the 1:1:100 solution was assumed to etch the IBE surfaces at approximately 600Å/min.

The etch solution was checked by suspending a piece of S.I. GaAs over the etchant so that half of the sample extended into the solution. Etching was allowed to proceed for three hours, after which time the sample profile was taken using a Sigmascan stylus profilometer.
Etching was found to occur at approximately 500Å/min, in good agreement with reported results. Thus, the assumed 600Å/min rate represented a maximum depth or worst case scenario.

Samples were prepared using 0.5keV and 3keV IBE energies, with $10^{16}$ ions•cm$^{-2}$ fluence to maximize damage. The surfaces were chemically removed according to the schedule in Table 14, after which Al Schottky contacts were deposited, giving a damage depth analysis for the IBE surfaces.

Table 15 indicates the diode parameters for the 0.5keV I-V characteristics given in Fig. 34. As indicated in the parameters and the curves, the characteristics appeared to recover somewhat after 100Å removal, and appeared fully recovered after 500Å. This places the damage depth relating to for the I-V's at about 100Å. Recovery was also seen in the conductance-temperature curve given in Fig. 35. The shunting effect also disappeared after 100Å of removal.

Surprisingly, the DLTS spectra also recovered after only 100Å removal as can be seen by the 60% recovered EL2 peak in Fig. 36. DLTS recovery was complete by 500Å. Therefore it appears that at this low energy the DLTS measurement, which is typically seen as a bulk measurement technique, was being affected by a near surface damaged region. This effect has not been reported in the literature, and appears to be an important one.

The diode parameters and I-V's for the 3keV diodes are given in Table 16 and Fig. 37. The diode and the conductance-temperature characteristics did not appear to recover until removal of 1000Å of damaged GaAs. This is far deeper than implant theory predicts (the
Table 14. Chemical etch schedule used in depth profile.

<table>
<thead>
<tr>
<th>Depth in Å</th>
<th>Etch time (600Å/min)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10s</td>
</tr>
<tr>
<td>500</td>
<td>50s</td>
</tr>
<tr>
<td>1000</td>
<td>1m 40s</td>
</tr>
<tr>
<td>5000</td>
<td>8m 20s</td>
</tr>
</tbody>
</table>

*All solutions were stirred during etch.
Table 15. Diode parameters for 0.5 keV depth profile.

<table>
<thead>
<tr>
<th>Depth (Å)</th>
<th>$\Phi_B$ (eV)</th>
<th>n</th>
<th>$I_{Sat}$ (A)</th>
<th>$V_B$ (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.62</td>
<td>1.80</td>
<td>$3.5 \times 10^{-6}$</td>
<td>8.0</td>
</tr>
<tr>
<td>100</td>
<td>.80</td>
<td>2.60</td>
<td>$1.2 \times 10^{-6}$</td>
<td>11.5</td>
</tr>
<tr>
<td>500</td>
<td>.94</td>
<td>1.40</td>
<td>$9.0 \times 10^{-12}$</td>
<td>11.2</td>
</tr>
<tr>
<td>1000</td>
<td>1.05</td>
<td>1.10</td>
<td>$6.0 \times 10^{-14}$</td>
<td>11.3</td>
</tr>
<tr>
<td>5000</td>
<td>.99</td>
<td>1.05</td>
<td>$1.0 \times 10^{-12}$</td>
<td>11.3</td>
</tr>
</tbody>
</table>
Figure 34. Current-Voltage characteristics for 0.5keV energy depth profile.
CONDUCTANCE VS. TEMPERATURE

1-0Ω
2-100Ω
3-500Ω
4-1000Ω
5-5000Ω

Figure 35. Conductance-Temperature characteristics for 0.5keV energy depth profile.
Figure 36. DLTS spectra for 0.5keV energy depth profile.
### Table 16. Diode parameters for 3keV depth profile.

<table>
<thead>
<tr>
<th>Depth (Å)</th>
<th>$\phi_B$ (eV)</th>
<th>n</th>
<th>$I_{\text{sat}}$ (A)</th>
<th>$V_B$ (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.70</td>
<td>5.4</td>
<td>$7.5 \times 10^{-7}$</td>
<td>14.3</td>
</tr>
<tr>
<td>100</td>
<td>.71</td>
<td>5.0</td>
<td>$1.7 \times 10^{-7}$</td>
<td>15.5</td>
</tr>
<tr>
<td>500</td>
<td>.78</td>
<td>5.4</td>
<td>$1.5 \times 10^{-8}$</td>
<td>13.5</td>
</tr>
<tr>
<td>1000</td>
<td>.84</td>
<td>2.7</td>
<td>$2.1 \times 10^{-9}$</td>
<td>12.5</td>
</tr>
<tr>
<td>5000</td>
<td>.89</td>
<td>2.6</td>
<td>$2.9 \times 10^{-10}$</td>
<td>10.5</td>
</tr>
</tbody>
</table>
Figure 37. Current-Voltage characteristics of 3keV energy depth profile.
tables give ≈ 80Å). This depth is not completely unexpected, however, since several workers have seen detectable damage down to 2000Å. [100]. Pang, in a similar study of RIE on VPE GaAs showed wet etch recovery of I-V's done at 1keV after removal of 250Å [101]. All papers tend to agree that a partly amorphous layer ≈ 100Å thick is generated during ion treatment while defects can diffuse into the bulk at room temperature.

Similarly, after removal of 1000Å by sequential wet etch showed recovery in the DLTS spectra, as shown in Fig. 38. It must be noted that no studies showed diffused bulk damage at this energy any deeper than 2000Å, which would be necessary to generate a true EL2 loss. Thus it appears that the EL2 recovery is tightly coupled to the DLTS measurement and its interaction with the damaged surface. Another interesting result of the DLTS measurement is the appearance, after 500Å removal, of two prominent electron trap peaks at .58eV, $3.8 \times 10^{15}\text{cm}^{-3}$ and .51eV, $2.7 \times 10^{15}\text{cm}^{-3}$. These may be attributable possibly to buried or shallow depth traps introduced by IBE, and previously hidden from the DLTS due to the damaged surface.

Further evidence that EL2 was not truly being removed at any energy is given by the electrical depth of study assuming a parallel plate model, where it was seen that the actual depth studied did not actually change while the EL2 peak recovered. Clearly this is an indicator of a desensitized DLTS measurement which has been distorted due to surface damage.

The major results of this section are:

1. At IBE of 0.5keV, $1 \times 10^{16}$ions-cm$^{-2}$, diode parameters, conductivity and EL2 recovered after 100Å of surface removal. Damage layer thickness is approximately 100Å.
Figure 38. DLTS spectra for 3keV energy depth profile.
2. At 3keV IBE, diode parameters, conductivity and EL2 recovered after 1000Å, of surface removal. Damage layer thickness is thus estimated at 1000Å for 3keV IBE material.

3. Depth of analysis shows that the loss of EL2 is most probably not real, but rather a result of a desensitized DLTS measurement due to the top damaged layer.

4. The same damage responsible for diode degradation is desensitizing DLTS measurements.

At this point, however, the nature of the damaged layers appears somewhat different at 0.5keV than 3keV in that the 0.5keV IBE seems to result in a thin somewhat conductive layer, while the 3keV causes a thicker and much less conductive layer. The electrical role of the bulk and surface damage may be somewhat different also. In order to further delineate their effects and clarify the damaging results of IBE two other tests were performed. In one test the surface layer between the contacts was chemically removed. In a second test, a variable resistor was placed in shunt with a virgin diode. The procedures and results are given below.

4.4.1 Surface Removal

In order to further delineate the effects of bulk or under contact damage from the surface or between contact damage both 0.5keV and 3keV (10^16 ions·cm⁻²) diodes were chemically etched with the 1:1:100 etchant so that 2000Å were removed from the surface between the contacts. The damage beneath the contacts remained the same, so that only the effect of damage under the Schottky survived.
No real improvement occurred in the I-V characteristics for the 0.5keV diodes, while the conductivity-temperature characteristic improved toward the virgin material, as shown in Fig. 39. This indicates that the surface shunt path may have been removed and the remaining conductivity was a result of damage beneath the contact. More surprisingly, however, was the DLTS spectra given in Fig. 40, in which it can be seen that EL2 was 25% recovered. This clearly indicates that the loss of EL2 is a result of some desensitized DLTS measurement in that the true spectrum is being mased by a surface effect.

At 3keV, 2000Å surface removal caused no improvement in the diode parameters and only slight improvement in the conductance temperature (G-T) characteristic. The DLTS spectrum was identical to that of unetched 3keV sample. It appears that the 3keV ion etch damage shows itself not through the surface between the contacts but, exclusively by damage under the contacts.

The results of these tests are:

1. For 0.5keV samples with 2000Å removal showed improvement in G-T and DLTS measurements. Therefore, damage at low energies occurs in both the surface and bulk. Further, the reduction in shunting conductance lends itself to the notion that there exists a low resistance layer.

2. For 3keV samples, removal of 2000Å around the contacts showed no perceptable recovery. This indicates that a higher resistance region may exist under the Schottky metal.

3. The nature of the damaged layer (low resistance, high resistance) is somewhat different for low and high energy IBE.
CONDUCTANCE VS. TEMPERATURE

1-0.5keV
2-2000Å SURFACE REMOVAL
3-VIRGIN

Figure 39. Conductance-Temperature virgin, 0.5keV and 2000Å surface removal.
Figure 40. DLTS spectra comparing virgin, 0.5keV and 2000Å surface removal.
4.4.2 Shunt Resistor

As further proof of the effects of a shunt conductance, a virgin diode was placed in the cryostat with an external shunting potentiometer connected as in Fig. 41. As the resistance was reduced from approximately 9kΩ to 240kΩ, the EL2 peak reduced as shown in Fig. 42, and the measured conductance increased from .1mS to 4.2mS, which is similar to values for the actual 0.5keV sample.

Therefore, the 0.5keV model of a low resistance layer shunting the diode appears correct, since the reduction of EL2 was nearly the same as the recovery of EL2 in the 0.5keV sample in which 2000Å of GaAs between the contacts had been removed. This will be seen further in section 4.6.

4.5 Anneal

Having partially understood the nature of the damage as being confined to a layer containing traps, with a partly amorphous crystal structure and variable depth according to IBE energy, an attempt was made to repair the damage. To do this, 0.5keV and 3keV (10^{16} ions·cm^{-2}) samples were thermally annealed in forming gas at approximately 350°C for 15min. before Schottky metal deposition. This temperature was chosen since it was believed that it may be high enough to repair damage without being hot enough to change the ohmic contact characteristics [102]. This was not an extensive study, but merely a brief attempt based upon previous knowledge of IBE damage found in this study to improve diode characteristics through a low temperature anneal.
Figure 41. Experimental setup for shunt resistor test.
Figure 42. DLTS spectra for shunt resistor test.
For the 0.5keV samples, no improvement was seen in the DLTS spectra while the G-T curve did flatten somewhat toward a virgin characteristic. This was accompanied by an order of magnitude decrease in $I_{\text{sat}}$, with an increase of $V_B$ to 11.0 V. Thus it appears that the brief anneal did slightly improve the I-V characteristics and conductivity, which could indicate a slight change in the defect density of the damaged layer toward the virgin state.

At 3keV there was again no improvement in the DLTS spectrum, while the conductivity vs. temperature characteristic took the shape of the conductivity vs. temperature characteristic of the 0.5keV IBE sample. This could indicate that the mobility of the damaged layer was somewhat improved, while defect densities were maintained, which would tend to resemble the 0.5keV case. This was supported in the I-V's in that $V_B$ and $n$ dropped while $I_{\text{sat}}$ increased.
Chapter 5: Modelling and Computer Analysis

From experimental results based on the electrical study taken up here, along with ESCA, UV reflectivity and capacitance-frequency dispersion, physical and electrical models were developed, and are given below.

5.1 Physical Model

Shown in Fig. 43 is a simple physical model based on the literature and previous results. The thicknesses given are variable, depending on the IBE parameters and total surface treatment. The first layer (D1) consists of an oxide of approximately 20Å in thickness and of variable composition, since ESCA (done by June Epp) showed that 0.5keV samples never completely lost their pre-etch oxide, while 3keV samples showed several different As and Ga oxides present after atmospheric exposure. The second layer (D2) is considered to be a partly amorphous layer of thickness less than 100Å to 1000Å (depending on beam energy), as seen by UV reflectivity (done by Jeff Feng) and the chemical etch profile. The degree of amorphicity is dependent upon energy since it seems that the lower energy IBE still retains most of its crystallinity, while higher energies indicate a marked loss in crystal structure. The third layer (D3) is somewhat ambiguous in thickness since it represents room temperature diffused damage which has been indicated in the literature. In this study, the damage appears to vary in depth from 100Å to 1000Å, depending on Ar⁺ energy. This is a somewhat generalized and simplified model, though electrical and computer analyses will be shown to help validate it.
Physical damage model:

Figure 43. Physical model of IBE damaged GaAs.
5.2 Electrical Model

A reverse bias electrical model was developed from the above physical structure by assignment of lumped R-C components to corresponding damaged layers. The result is given in Fig. 44, with the components listed. This model was used in computer analysis to simulate DLTS and capacitance-frequency measurements. As will be seen, the simulations are in good agreement with experimental results.

5.2.1 DLTS Simulation

In order to simulate the DLTS measurement so that results corresponding to experiment could be directly compared the simulation took measurements in the same fashion as in the actual experiment. Any capacitance bridge analyzes a network's capacitance from a measurement of its impedance at a given frequency (fixed at 1 MHz for HP 4280A). The network capacitance is defined as

\[
C = -\frac{1}{\omega X}
\]  

(35)

where \(X\) is the imaginary part of the impedance \(Z = R + jX\), [103]. The impedance of the electrical model is

\[
Z_M = \frac{(R_{SE} + R_{SF}^2)}{G^2 + \omega^2} + j \frac{(R_{SF} - R_{SG})}{G^2 + \omega^2}
\]  

(36)

where

\[
E = \frac{R_A}{1 + \omega^2 C_A R_A^2} + \frac{R_D}{1 + \omega^2 C_D R_D^2}
\]  

(37)
Electrical damage model:

\[ R_A, C_A \] - amorphized series layer
\[ R_S \] - surface shunt layer
\[ R_D, C_D \] - ideal diode
\[ C_T \] - trap capacitor(EL2)

Figure 44. Electrical model corresponding to the physical model.
and

\[ F = \frac{C_A R_A^2}{1 + \omega^2 C_A^2 R_A^2} + \frac{C_D R_D^2}{1 + \omega^2 C_D^2 R_D^2} \]  \quad (38)

with \( C = C_A + R_S \) and \( C = C_D - C_T \), where the negative sign has been added to indicate the electron trap nature of EL2. The measured capacitance from the above impedance is given as

\[ C_M = \frac{1}{\omega F} + \frac{2E}{\omega F R_S} + \frac{E^2}{\omega F R_S^2} + \frac{F}{\omega R_S^2} \]  \quad (39)

Clearly as \( R_S \to \infty \) and \( R_A \to 0 \), \( C_M \to C_D \), (ideal diode) with \( C_T \) as a transient capacitance. Since \( C_T \) represents EL2, it is to be treated as a transient capacitance whose value will be given as

\[ C_T = C_0 \exp(-r_T t) \]  \quad (40)

where \( C_0 \) is the capacitance change amplitude and \( r_T \) is the trap decay rate given by

\[ r_T = \nu \exp(-E_T/kT) \]  \quad (41)

with \( \nu = 10^{12} \text{s}^{-1} \) (the thermal release frequency) and \( E_T = 0.77 \text{eV} \) (the EL2 trap energy).

These equations were programmed as detailed in Appendix B, and DLTS spectra simulated.
Using the above equations, the EL2 peak magnitude was plotted as a set of curves (Fig. 45) developed from variation of $R_A$ and $R_S$ with all other components held constant ($C_A = 2\, \text{pF}$, $C_D = 150\, \text{pF}$, $R_D = 10\, \text{M\Omega}$). $C_A$ was held constant since it was believed that the resistivity of these layers played a more important role, as seen in the resistor shunt experiment. Two regions of EL2 loss can be seen, even though no true loss has occurred. These regions may be directly related to the IBE results (loss of EL2) and confirm that the DLTS was being desensitized by the surface damage layer.

The first loss region relates the 0.5keV samples to a low resistance shunt layer. Figure 46 indicates the progression of decreasing shunt and the corresponding EL2 magnitude. It is also interesting to note that these results correspond directly to both the shunt resistor and surface removal experimental results. The second region of loss (higher resistances) can be related to the 3keV samples, in which it is believed that a higher resistance, lower mobility layer was generated by the high energy IBE. This progression can be seen in Fig. 47.

The strong resemblance of the simulated DLTS spectra to experiment suggests the validity of both the physical and corresponding electrical models.

5.2.2 Capacitance-frequency Simulation

As mentioned earlier S. Sen reported capacitance-frequency dispersion for 3keV IBE diodes. The same electrical model was used to simulate his results and are given in Fig. 48 assuming high resistance layers corresponding to the virgin and 3 keV cases (network values corresponded to those used in the 3keV DLTS simulations that reduced
Figure 45. Curves showing dependency of EL2 peak magnitude on $R_L$ and $R_S$ variation.
Figure 46. Simulated DLTS for 0.5keV layer model.
Figure 47. Simulated DLTS for 3keV layer model.
Figure 48. Capacitance-Frequency dispersion simulated using electrical model as compared to Sen's results.
EL2). As can be seen, the computer results accurately predict the
dispersion, and compare well with his results. Therefore, this
electrical model helps to explain both.

(It must be noted that the models used were first developed for
Sen's work. It is therefore even more gratifying that with only the
addition of $C_T$ for DLTS simulation, the results could be simulated with
the same model.)
Chapter 6: Summary and Conclusions

Several important and useful results were obtained from this research, and they are summarized below with recommendations for further analysis.

6.1 Summary

First, the fabrication process for "good" reproducible diodes was developed, starting from vendor wafers. This took approximately three months to perfect, using trial-and-error, changing such things as evaporation rates, chemical etch times and silver bonding epoxies. The result was discussed in section 3.1.

Second was the development of a fully automated test station, which allows the user to connect a sample and run experiments via a computer. The computer runs the electrical tests, collects the data and graphically displays it on either the screen, printer or plotter. The graphs in this dissertation are examples of that output. This system has not only capabilities for Schottky diodes but is adaptable to many other devices including the new high $T_c$ superconductors.

Third was the analysis of the electrical effects of low energy argon ion etching of GaAs through the use of the electrical testing of Schottky diodes fabricated on these surfaces. A more complete understanding of this process was achieved, with the major results showing that:

1. At low energies (0.5keV) and moderate fluences IBE seems to create a thin ($\approx 100\AA$), low resistance layer (higher resistance with respect to virgin samples) with possible near surface traps. This damaged layer causes leaky diodes with reduced reverse breakdown
voltages. The EL2 peak is hidden by the desensitizing effects of this layer and the corresponding electrical network (resulting from IBE) on DLTS. Low energies at low fluences produce no discernable damage to the crystal.

2. At higher energies (=3keV) the IBE seems to generate a somewhat thicker (=1000Å), higher resistance (due possibly to lowered mobility), partly amorphous layer with associated traps and defects which tend to increase $V_B$ and $V_T$ as well as making diodes leaky. These effects are essentially independent of fluence. Being of higher resistance, the conductivity, and thus the effect of the surface path, is all but eliminated from influencing the measurements as in the 0.5keV samples. The DLTS is also desensitized by this higher resistance series layer, which is sufficient to create an apparent loss of EL2.

3. Annealing at low temperatures (350°C) for 15min. resulted in no improvement in DLTS characteristics for both 0.5keV and 3keV samples. However, I-V and G-T characteristics showed movement for the 0.5keV sample towards virgin results, while the 3keV sample moved towards 0.5keV energy sample results. Though diode recovery was not achieved, sample changes could be seen.

4. The DLTS (usually considered a bulk measurement) appears to be quite sensitive to ion surface treatment through the correspondingly modified layers generated by IBE.

5. The EL2 DLTS peak appears to be useful as a sensitive monitor of surface layer changes due to IBE.

6. Computer analysis of the proposed physical and electrical models
successfully reproduced several experimental results, including DLTS spectra and previously reported capacitance-frequency dispersion. This indicates that some validity exists in these models.

6.2 Future Recommendations

A somewhat clearer picture of IBE damage and its electrical effects on Schottky diodes has been presented. However, there is room for improvement in several areas which would clarify the nature of the damage.

- Study IBE effects in MBE grown layers, since carrier mobility can be measured without the shunting effects of the bulk.
- Study IBE effects on MESFET operation.
- Extend the annealing study to optimize the process for temperature and time as functions of IBE energy and fluence to detail damage sensitivity to heat treatments.
- Perform photo-electric measurements on IBE surfaces such as spectral response and quantum efficiency to study surface related defects as could be seen in photocapacitance measurements.
- In situ deposition of Schottky metals directly after IBE to eliminate oxide effects caused by exposure to atmosphere.
- Extension of experiments to RIE and plasma etch for correlation with these results.
- More detailed modelling and computer analysis to include possible temperature and voltage dependence of the damage layer and network components to aid in understanding of damage and possible repair mechanisms.
This study, coupled with the above recommendations, could extend understanding, optimization and use of this most effective device processing technique.
REFERENCES


4. Same as [1], p. 22.


6. Same as [3], p. 850.

7. Same as [2], p. 528.

8. Same as [3], p. 850.


120


34. Same as [2], p. 62.


37. Same as [27].

38. Same as [28].

39. Same as [29].

40. Same as [31].

41. Conversation with Keith Ritala of Cominco.

42. Same as [2], p. 75.

43. Same as [1], p. 120.

44. Same as [1], p. 104.

45. Same as [2], p. 151, 156.

46. Same as [1], p. 199.


52. Same as [2], p. 207.


62. Same as [47].


69. Same as [3].

70. Same as [3], p. 291.


72. Same as [68], p. 27.


74. Same as [51].

75. Same as [68], p. 33.

76. Same as [3], p. 250.

77. Same as [68], p. 77-91.

78. Same as [37], p. 259-263.

79. Same as [3], p. 100.
80. Same as [68], p. 125.
81. Same as [3], p. 248.
82. Same as [68], p. 147-152.
83. Same as [2], p. 200.
84. Same as [1], p. 233.
85. Conversations with Rick Hudgens of Texas Instruments.
86. Seminar by Dr. Wu of IBM Thomas J. Watson Research Center.
87. Same as [85].
88. Same as [41].
89. Same as [3], p. 245-295.
92. Same as [47].
93. Same as [61].
94. Same as [2], p. 153.
95. Same as [61].
96. Same as [47].
97. Same as [2], p. 201.
98. Same as [48].
100. Same as [54].
101. Same as [57].
102. Same as [85].
Appendix A: MEDUSA Software

The software is divided into three separate sections as mentioned earlier in 3.2 and the flowcharts are given in Fig. 49.

The system allows for 3 device environments to be used. These include:
1. Temperature sweep from 30K to 600K in increments of 1K.
2. Time sweep at a fixed holding temperature with experiments run every 5min. for an indefinite period.
3. Room temperature, no cryostat use.

Within these environments the sample may be exposed to either of 4 experimental groups as required by instrumentation. These include:
1. C-t, G-t, C-V, G-V, C, G
2. I-V, C-V
3. Mobility
4. 4-pt. Resistivity

After experimental completion the software automatically resets all instruments and cycles into a graphical display routine. These graphics allow for device analysis from the computer screen as mentioned in Section 3.2.

The complete documented software written in Better Basic follows given in the same four discrete sections as it was written and operates:
1. MEDUSA Batch: ties and runs all software together as a unit.
3. Runnit: runs experiments and collects data.

*Note assistance of Phillip Johnson
Figure 49. Flowchart for MEDUSA software.
ECHO OFF
:REPEAT
CD \BB
BB.COM \MEDUSA\NEW.CNF/C run "\MEDUSA\PARAMETER.CMP"
IF EXIST C:\DATA\END GOTO GRAPH
CD \BB
BB.COM \MEDUSA\NEW.CNF/C run "\MEDUSA\RUNIT.CMP"
IF EXIST C:\DATA\REDO GOTO REPEAT
:GRAPH
IF EXIST C:\DATA\END ERASE C:\DATA\END
CD \BB
BB.COM \MEDUSA\NEW.CNF/C run "\MEDUSA\GRAPHICS.CMP"
CLS
CD\
PROCEDURE: Border
INTEGER: MX
9 REM
10 REM
11 REM *
12 REM * The Procedure BORDER prints '*' around the menu.
13 REM *
14 REM * to call type BORDER (ROWNUMBER%) (RETURN)
15 REM *
16 REM * INTEGER: MX - used in a FOR-NEXT loop
17 REM *
18 REM * INTEGER ARGUMENT: Rownumber%/VAR - used to delineate
19 REM * the maximum row for the '*'
20 REM *
21 REM
22 REM
23 REM 100 COLOR 2,0
24 REM 110 LOCATE 3,1
25 REM 120 PRINT "*******************************"
26 REM 130 PRINT "*******************************"
27 REM 140 Rownumber% = Rownumber% + 1
28 REM 150 FOR MX = 4 TO Rownumber%
29 REM 160 LOCATE MX, 1
30 REM 170 PRINT "*
31 REM 180 LOCATE MX, 79
32 REM 190 PRINT "*
33 REM 200 NEXT MX
34 REM 210 Rownumber% = Rownumber% + 1
35 REM 220 LOCATE Rownumber%, 1
36 REM 230 PRINT "*******************************"
37 REM 240 PRINT "*******************************"
38 REM 250 Rownumber% = 0
39 REM 260 COLOR 7,0
40 REM 270 EXIT
END PROCEDURE

PROCEDURE: Title
    INTEGER: Column%
    9 REM
    10 REM ******************************************************************************
    11 REM *
    12 REM * The Procedure TITLE prints the title of the menu transferred from the main program by TITLE
    13 REM * in the location 1, col%
    14 REM * to call: TITLE (COL%, TITLE$)
    15 REM *
    16 REM * INTEGER ARGUMENT: Col% - gives the 1-80 column number
    17 REM *
    18 REM * STRING ARGUMENT: Title$ - gives the title string
    19 REM *
    20 REM ******************************************************************************
    21 REM *
    22 REM ******************************************************************************
    23 REM
    100 Column% = CINT((79 - LEN(TITLE$))/2)
    110 COLOR 5,0:CLS
    120 LOCATE 1, Column%
    130 PRINT Title$
    140 COLOR 7,0
    150 EXIT
END PROCEDURE

PROCEDURE: Menu
    INTEGER: Col%
    STRING: Number$[4]
    INTEGER: Row1%
    9 REM
    10 REM ******************************************************************************
    11 REM *
    12 REM * The Procedure MENU takes the names transferred by the
    13 REM *
13 REM * calling program and prints them as a screen. *
14 REM *
15 REM * to call: MENU (NUMBER%, ROWNUMBER%, NAME$, ROW%) *
16 REM *
17 REM * INTEGER: Co1% *
18 REM *
30 REM * INTEGER ARGUMENT: Number%/VAR, Rownumber%/VAR *
32 REM *
40 REM * STRING: Number$[4] *
43 REM *
50 REM * STRING ARGUMENT: Name$ *
59 REM *
60 REM ************************************************************

**************
61 REM
100 COLOR 3,0
110 Number% = Number% + 1
120 Number% = STR%(Number%)
130 DEL$(Number%, 1, 1)
140 IF Number% < 13 THEN Co1% = 5 ELSE Co1% = 42
150 IF Number% = 13 THEN Row% = 1 ELSE Row% = Row% + 1
160 LOCATE Row%+4, Co1%  
170 PRINT "(";Number$;")"; Name$  
180 IF Rownumber% < CSRLIN THEN Rownumber% = CSRLIN
190 COLOR 7,0
200 EXIT
END PROCEDURE

PROCEDURE: Finish
STRING: Number$[4]
INTEGER: Column%
9 REM
10 REM ************************************************************

**************
11 REM *
12 REM * The Procedure FINISH prints the message which allows *

13 REM * the user to exit the menu.
   *
14 REM *
   *
15 REM * to call:  FINISH (ROWNUMBER%, NUMBER%, COL%,
   ,NAME%) *
16 REM *
   *
20 REM * INTEGER ARGUMENT:  Rownumber%/VAR, Number%/VAR,col% *
22 REM *
   *
30 REM * STRING:  Number$[4] *
32 REM *
   *
40 REM * STRING ARGUMENT: Name$ *
43 REM *
   *
50 REM **************************************************************

51 REM
100 Column% = CINT((77 - LEN(Name$))/2)
110 COLOR 3,0
120 Rownumber% = Rownumber% + 1
130 Number% = Number% + 1
140 Number$ = STR$(Number%)
150 DEL$(Numbers$,1,1)
160 LOCATE Rownumber%,Column%
170 PRINT "(";Number$;") ";Name$
180 COLOR 7,0
190 EXIT
END PROCEDURE

PROCEDURE: Filecheck
 STRING: Chk*[16]
 INTEGER: T%,I%
 STRING: A[*?]
 INTEGER: Maxloop%
 9 REM
10 REM **************************************************************

11 REM *

12 REM * The Procedure FILECHECK sees if the direc
tory specified *
13 REM * has been previously used. If it has then
it asks the *
14 REM * user if they want to erase all previous f
iles. If the *
15 REM * directory has not been used, it then creates the asked *
16 REM * for directory.
17 REM * to call: FILECHECK (FILE$, REDO$)
18 REM *
20 REM * INTEGER ARGUMENT: Rownumber%/VAR, Number%
23 REM *
30 REM * STRING: Number$[4]
32 REM *
40 REM * STRING ARGUMENT: Name$
43 REM *
50 REM******************************************************************************
51 REM
99 CLOSE
100 ON ERROR GOTO 10000
110 OPEN "\DATA\" + File$ + "\INFO" FOR INPUT AS #1
120 IF ERR = 1001 OR ERR=1007 THEN GOTO 180 'If no file goto erase routine
130 CLOSE #1
140 LOCATE 20,5;COLOR 7,0
150 PRINT "THE FILE NUMBER YOU CHOSE HAS ALREADY BEEN USED."
160 INPUT " DO YOU WANT TO OVERWRITE THE EXISTING FILES (Defaults to NO)";Chk$
170 IF INSTR(Chk$,"Y") <> 0 OR INSTR(Chk$,"y") <> 0 THEN GOTO 180 ELSE GOTO 1000
180 RESTORE,50000
190 READ Maxloop%
210 IX = IX + 1
220 READ A$
230 KILL "\DATA\" + File$ + "\" + A$
265 IF IX < Maxloop% THEN GOTO 210
270 RMDIR "\data\" + File$
280 MKDIR "\data\" + File$
290 Redo$ = 0
300 GOTO 1010
1000 Redo$ = 1
1010 ON ERROR 0
1020 COLOR 7,0
1030 LOCATE 19,5;PRINT SPC(79);PRINT SPC(79);PRINT SPC(79);PRINT SPC(79)
1040 CLEAR
1050 EXIT
10000 REM
10001 REM ###################################################################
10002 REM * These are the ERROR handlers which allows
10003 REM * the Procedure *
10004 REM * to check to see if the files or directory
10005 REM * have been *
10006 REM *
10007 REM ###################################################################
10010 IF ERR = 1001 THEN RESUME NEXT
10020 IF ERR=18 THEN RESUME,270
10030 IF ERR = 1007 THEN RESUME NEXT
10040 IF ERR=1022 THEN RESUME,280
10050 PRINT "I'm sorry but the FILECHECK procedure stil
l doesn't work."
10060 PRINT "This is error number ";ERR;" from line "ER
10070 END
50000 REM
50001 REM ###################################################################
50002 REM *
50003 REM * The DATA statements are for erasing the p
50004 REM * possible files *
50005 REM * from the hard disk.
50006 REM *
50007 REM ###################################################################
50008 DATA 9
50010 DATA info,C-t,CGV,CG,G-t,IV,CV,Mob,Res
50020 END PROCEDURE

PROCEDURE: Placeampersand
INTEGER: Col%,Pos%
9 REM
10 REM ###################################################################
11 REM *
12 REM * The Procedure PLACEAMPERSAND takes the nu
numbers passed by CHOICE% and NUMBER% and arranges them so that an "@" is placed in the appropriate place to show that the item chosen has actually been chosen.

17 REM to call: PLACEAMPERSAND (NUMBER%,CHOICE%)

18 REM

20 REM INTEGER: Pos%, Col%

30 REM INTEGER ARGUMENT: NUMBER%, CHOICE%

40 REM

***************

41 REM

100 IF Number% < 12 THEN Col% = 3
110 IF Number% > 12 THEN Col% = 40
120 IF Choice% > 12 THEN Pos% = Choice% - 12 ELSE Pos% = Choice%

130 LOCATE Pos%+4,Col%
140 PRINT "@
150 EXIT

END PROCEDURE

PROCEDURE: Removeampersand

INTEGER: Col%,Pos%

100 IF Number% < 12 THEN Col% = 3 ELSE Col% = 40
110 IF Choice% > 12 THEN Pos% = 12 - Choice% ELSE Pos% = Choice%

120 LOCATE Pos%+4,Col%
130 PRINT " 
140 EXIT

END PROCEDURE

PROCEDURE: Settempparam

REAL: Initialtemp!,Finaltemp!,Deltatemp!,Chk!,Chk1!,Min!,Max!,Test!
INTEGER: Error%,Line%
STRING: Chk$[4]
REAL: Test
INTEGER: Delay%
REAL: Tyme
STRING: Sign*[16]
REAL: Increment!, Incrementtemp!, Maximumtemp!

9 REM
10 REM ******************************
************
12 REM * The Procedure SETTEMPPARAM allows the user to 1) set the initial, final and increment temperature for each experiment, 2) set the temperature for a TIME run, or 3) choose a room temperature run. This procedure also has a variety of error checks to make sure the chosen temperatures are possible.

18 REM *
19 REM *
20 REM * to call: SETTEMPPARAM (TIME%, COL%, INITIALTEMP, FINALTEMP, DELTATEMP, EXPT$, NAME$) *
22 REM *
30 REM * INTEGER: error%, line%
32 REM *
40 REM * INTEGER ARG: time%, col%
41 REM * min!, max!, test!
42 REM *
50 REM * REAL: initialtemp!, finaltemp!, deltatemp!, chk!, chk1!
51 REM * min!, max!, test!
52 REM *
60 REM * REAL ARG: minimumtemp!/VAR, maximumtemp/VAR, incrementtemp!/VAR
61 REM *
62 REM *
70 REM * STRING: chk$
REM *
* STRING ARG: expt$, name$
*
** *********************************************
***************
ON ERROR GOTO 30000
IF Timex = 3 THEN GOTO 800
IF Timex = 1 THEN GOTO 600
IF Expt$ <> "Cryostat" THEN GOTO 130
COLOR 2,0:CLS:LOCATE 1,Colx
PRINT "This sets the temperature for the ";Expt$ ":"
COLOR 4,0:LOCATE 3,5
PRINT "These are the Cryostat temperature ";Name$	"
COLOR 3,0:LOCATE 5,16
PRINT "Initial";SPC(15);"Final";SPC(15);"Increment"
COLOR 9,0:LOCATE 6,16
PRINT Minimumtemp!;SPC(17);Maximumtemp!;SPC(18); Incrementemp!
COLOR 5,0:LOCATE 10,5
LOCATE 10,1:PRINT SPC(79)
LOCATE 10,5:INPUT "Enter the initial temperature ";Initialtemp!
IF Initialtemp! = Minimumtemp! THEN GOTO 290
Errorx = 1:Linex = 1:GOTO 2000
IF Initialtemp! ( Maximumtemp! THEN GOTO 310
Errorx = 2:Linex = 1:GOTO 2000
LOCATE 12,5:INPUT "Enter the final temperature ";Finaltemp!
IF Finaltemp! ( Maximumtemp! THEN GOTO 340
Errorx = 2:Linex = 1:GOTO 2000
LOCATE 14,5:INPUT "Enter the temperature increment ";Deltatemp!
IF Deltatemp! ( Incrementemp! THEN GOTO 410
390 IF Deltatemp! = Incrementemp! THEN GOTO 410
400 Errorx = 3:Line% = 3:GOTO 2000
410 GOSUB 1000
420 COLOR 7,0:LOCATE 20,1
430 INPUT "Do you want to change any of the above (Defaults to No) ";Chk$
440 Chks = MID$(Chks,1,1)
450 IF Chks = "Y" OR Chks = "y" THEN GOTO 110
460 IF Expt$ = "Cryostat" THEN GOTO 560 'If for cryostat goto var. set
470 IF SGN(Incrementtemp!) = SGN(Deltatemp!) THEN GOTO 560
480 IF SGN(Incrementtemp!) = SGN(Deltatemp!) THEN GOTO 560
490 Minimumtemp! = Finaltemp!
500 Maximumtemp! = Initialtemp!
510 Incrementtemp! = SGN(Incrementtemp!) * ABS(Deltatemp!
520 EXIT
560 Minimumtemp! = Initialtemp!
570 Maximumtemp! = Finaltemp!
580 Incrementtemp! = Deltatemp!
590 EXIT
599 REM
600 REM *******************************************************
601 REM *
602 REM * This section is used for a time run. It allows the *
603 REM * user to enter the temperature which will be used. *
604 REM *
605 REM *******************************************************
610 COLOR 2,0=CLS=LOCATE 12,5
620 PRINT "Qt what temperature do you want to do the time run (must be ";
630 PRINT ":";Minimumtemp!;", and ":Maximumtemp!
640 INPUT ", K)";Min!
650 IF Min! >= Minimumtemp! THEN GOTO 670
660 Min! = 10=Error% = 1:Line% = 4:GOTO 2000
670 IF Min! <= Maximumtemp! THEN GOTO 690
680 Max! = 600:Error% = 2:Line% = 4:GOTO 2000
690 Minimumtemp! = Min!
700 Maximumtemp! = Min!
710 Incrementtemp! = 0
720 EXIT
799 REM
800 REM
810 CLS:COLOR 2,0=LOCATE 12,5
820 INPUT "Enter The Room's Temperature(defaults to 290K)";Minimumtemp!
825 IF Minimumtemp!=0 THEN Minimumtemp!=290
830 Maximumtemp! = Minimumtemp!
840 Incrementtemp! = 400.0
139

850 EXIT
899 STOP
999 REM
1000 REM ************************************************************
1001 REM *
1002 REM * This section checks to see if the temperatures set *
1003 REM * intersect those of the cryostat.
1004 REM *
1005 REM ************************************************************
1006 REM
1009 GOTO 1030
1010 IF Initialtemp! + ABS(Deltatemp!) >= Finaltemp! THEN GOTO 1030
1020 Error% = 5:Line% = 1:GOTO 2000
1030 Chk! = (Initialtemp! - Minimumtemp!)/Incrementtemp

1040 Chk1! = FIX(Chk!)
1050 Test! = Chk1! - Chk!
1060 IF Test < 0.0001 OR Test > 0.0001 THEN GOTO 1080
1070 Error% = 6:Line% = 1:GOTO 2000
1080 Chk! = Deltatemp!/Incrementtemp!
1090 Chk1! = FIX(Chk!)
1100 Test! = Chk1! - Chk!
1110 IF Test < 0.0001 OR Test > 0.0001 THEN GOTO 1130
1120 Error% = 7:Line% = 3:GOTO 2000
1130 RETURN
1999 REM
2000 REM ************************************************************
2001 REM *
2002 REM * This is the section which prints up the error messages *
2003 REM * for temperature settings which are not allowed. *
2004 REM *
2005 REM ************************************************************
2006 REM
2008 COLOR 7,0:LOCATE 20,5
2010 GOTO 2000:GOSUB 2100,2150,2200,2250,2300,2350,240,2450
2030 Delay% = 3:GOSUB 20000
2040 COLOR 5,0:LOCATE 20,1:PRINT SPC(79):PRINT SPC(79)
2050 ON Line% GOTO 2500,2550,2600,2650
2099 REM
2100 REM --------This is for a temperature less than
the minimum--------
2101 REM ----------------cryostat temperature-------
-----
2102 REM
2110 PRINT "The temperature must be greater than ";Minimumtemp!;"!"
2120 RETURN
2149 REM
2150 REM --------This is for a temperature greater than
the maximum--------
2151 REM ----------------cryostat temperature-----
-----
2152 REM
2160 PRINT "The temperature must be less than ";Maximumtemp!;"!
2170 RETURN
2199 REM
2200 REM --------This is for an increment less than
the cryostat--------
2201 REM ----------------temperature-------------
---
2202 REM
2210 PRINT "The temperature increment must be greater
than";Incrementtemp!;"!"
2220 RETURN
2249 REM
2250 REM --------This is for an increment which is
going in-------------
2251 REM ----------------the wrong direction-----
-----
2252 REM
2255 IF SGN(Finaltemp! - Initialtemp!) = 1 THEN Sign$ = "positive"
2257 IF SGN(Finaltemp! - Initialtemp!) = -1 THEN Sign$ = "negative"
2258 IF SGN(Finaltemp! - Initialtemp!) = 0 THEN Sign$ = "no slope"
2260 PRINT "The temperature increment must be ";Sign$;"!
2270 RETURN
2299 REM
2300 REM --------This is for an intersection of less than 2-------------
2301 REM
2310 PRINT "The parameters you have set has only 1 experimental reading in"
2320 PRINT "it. It must have at least 2. ";
2340 RETURN
2349 REM
2350 REM -----------This is for a wrong initial temperature-----------
2351 REM
2360 PRINT "The cryostat temperatures will never intersect with the initial";
2370 PRINT "temperature set"
2380 RETURN
2399 REM
2400 REM -----------For an impossible temperature increment-----------
2401 REM
2410 PRINT "The increment you want is not possible!"
2420 RETURN
2430 REM
2440 PRINT "The Final Temperature must be greater than " ; Initialtemp!
2450 RETURN
2469 REM
2470 REM -----------This sends the program back to the initial temp-----------
2480 REM
2490 GOTO 250
2509 REM
2510 REM -----------This takes the experiment to the final temperature-----------
2520 REM
2530 LOCATE 12, 1; PRINT SPC(79)
2540 GOTO 310
2559 REM
2560 REM -----------This takes the experiment back to the increment-----------
2570 REM
2580 LOCATE 14, 1; PRINT SPC(79)
2590 GOTO 360
2609 REM
2610 REM -----------This takes the experiment back to the time segment-----------
2620 REM
2630 GOTO 610
2640 REM
2650 REM *******»*************************************
2660 REM * °
2670 REM * This allows a timed delay for error messages *
2680 REM *    °
2690 REM *******¤
20005 REM
20010 Tyme = TIMER + Delay%
20020 IF Tyme > TIMER THEN GOTO 20020
20030 RETURN
30000 PRINT "The error is "; ERR;" and the line number is "; ERL
END PROCEDURE

PROCEDURE: Timeparam
STRING: Chk$[4]
9 REM
10 REM ************
20 REM * The Procedure TIMEPARAM sets the final and incremental times for a constant temperature time run *
30 REM * to call: TIMEPARAM (FINALTIME!, DELTATIME !) *
40 REM * REAL ARG: Finaltime!/VAR, Deltatime!/VAR *
50 REM *
60 REM *
70 REM *
80 REM *
90 REM *
100 ON ERROR GOTO 10000
110 COLOR 2,0:CLS:LOCATE 1,25
120 PRINT "This sets the TIME parameters"
130 COLOR 5,0
140 LOCATE 4,3:INPUT "Enter the amount of time this program is to run (> 5 minutes)"; Finaltime!
150 LOCATE 6,3:PRINT "Enter the length of time between experiments"
160 COLOR 3,0:PRINT " set ";
170 COLOR 5,0:INPUT "runs ( > 5 minutes)"; Deltatime!
180 COLOR 7,0:LOCATE 20,5
190 INPUT "Do you want to change any of the above (Defaults to No)"; Chk$
200 Chk$ = MID$(Chk$, 1, 1)
210 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
220 Finaltime! = 60 * Finaltime!
230 Deltatime! = 60 * Deltatime!
240 EXIT
10000 CLS:LOCATE 15,5:PRINT "The error is ";ERR;" in line";ERL
10010 STOP
END PROCEDURE

PROCEDURE: Svolt
STRING: Chk$[3]

9 REM
10 REM ************************************************************
11 REM *
12 REM * The Procedure SVOLT allows the user to enter the *
13 REM * voltages (including a break voltage and two different *
14 REM * increments. It does not include any error checking *
15 REM * for entering incorrect voltages.
16 REM *
17 REM * to call SVOLT (START!,STOP1!,STEP1!,STOP2!
,STEP2!) *
18 REM *
30 REM * REAL ARGUMENTS: Start!/VAR, Stop1!/VAR, Stop2!/VAR *
32 REM * Step1!/VAR, Step2!/VAR *
40 REM *
41 REM ************************************************************
42 REM
100 ON ERROR GOTO 10000
110 LOCATE 3,1
120 DO 20 TIMES
130 PRINT SPC(79)
140 REPEAT
145 LOCATE 4,24:COLOR 4,0
147 PRINT "This sets the Voltage settings"
150 COLOR 5,0
150 LOCATE 6,5:INPUT "Enter the Start Voltage (-100 to 100 V)";Start!
170 LOCATE 8,5:INPUT "Enter the first Stop Voltage (-100 to 100 V)";Stop1!
180 LOCATE 10,5:INPUT "Enter the first Step Voltage (}
abs[0.01 V]";Step1!

190 LOCATE 12,5:INPUT "Enter the second Stop Voltage (-100 to 100 V)";Stop2!

200 LOCATE 14,5:INPUT "Enter the second Step Voltage () abs[0.01 V]";Step2!

210 COLOR 7,0:LOCATE 20,1

220 INPUT "Do you want to change any of the above (De faults to No)";Chk$

230 Chk$ = MID$(Chk$,1,1)

250 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110

260 ON ERROR 0

270 Step1! = VAL(STR$(Step1!))

280 Step2! = VAL(STR$(Step2!))

290 EXIT

1000 REM

10010 IF ERR = 1008 THEN RESUME

10020 COLOR 7,0:LOCATE 18,3:PRINT "Sorry the procedure II 3

10030 COLOR 20,0:PRINT "SVOLT ";

10040 COLOR 7,0:PRINT "is bombing. This is error ";ERR ;

END PROCEDURE

PROCEDURE: DVdt

STRING: Chk$[3]

9 REM

10 REM **________________________________________________________________________**

11 REM *

12 REM * The Procedure dvdt allows the user the set the slope *

13 REM * for the C-V experiment in group 2.

14 REM *

15 REM * to call: DVDT (dVdt!,start!,stop!,step!)

16 REM *

30 REM * real arg: dVdt!/VAR

32 REM *

40 REM **________________________________________________________________________**

41 REM

47 REM * real arg:start!/var,stop!/var,step!/var

100 ON ERROR GOTO 10000

110 LOCATE 3,1

120 DO 20 TIMES
130 PRINT SPC(79)
140 REPEAT
150 COLOR 4,0:LOCATE 4,23
160 PRINT "This sets the dV/dt slope"
170 COLOR 5,0
180 LOCATE 6,5:INPUT "Enter the dV/dt setting (0.001
to 1 V/s)";DVdt!
185 LOCATE 8,5:INPUT "Enter the Start Voltage (-100 t
to 100 V)";Start!
187 LOCATE 10,5:INPUT "Enter the Stop Voltage (-100 t
to 100 V)";Stop!
188 LOCATE 12,5:INPUT "Enter the Step Voltage (-10 to
10 V)";Step!
190 LOCATE 20,5:COLOR 7,0
200 INPUT "Do you want to change any of the above (De
faults to No)";Chk$
210 Chk$ = MID$(Chk$,1,1)
220 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
230 EXIT

10000 REM
10001 REM *******************************************************
10002 REM *
10003 REM * This is the error procedures. It correct
10004 REM * the expected * errors and kills the program and writes w
10005 REM * which error * and line number for an unexpected error
10006 REM *
10007 REM *******************************************************
10008 REM
10010 IF ERR = 1008 THEN RESUME
10020 COLOR 7,0:LOCATE 20,5:PRINT "Sorry the procedure
"
10030 COLOR 20,0:PRINT "dVdt ";
10040 COLOR 7,0:PRINT "bombing. The error is ";ERR;" in
line number";ERL
10050 STOP
END PROCEDURE

PROCEDURE: Settime
STRING: Chk$[3]
REAL: Tyme!
INTEGER: MX
10 REM
11 REM *******************************************************
The Procedure SETTIME allows the user to enter the Step Delay Time, Initial Hold Time, Final Hold Time and Delta Hold Time in seconds.

to call: SETTIME (STEPDELAY!, INITIALHOLD!, FINALHOLD!, DELTAHOLD!, MINHOLDTIME!, MINDELA
R, TYPE%)

REAL ARG: Stepdelay!/VAR, Initialhold!/VAR, Deltahold!/VAR, Finalhold!/VAR

**********

100 ON ERROR GOTO 10000
110 LOCATE 3,1
120 DO 21 TIMES
130 PRINT SPC(79)
140 REPEAT
150 LOCATE 4,25:COLOR 4,0
160 PRINT "This sets the Time Parameters"
170 COLOR 5,0
180 IF Type% = 3 THEN GOTO 210
190 LOCATE 6,5:PRINT "Enter the Step Delay Time, in seconds (t > Mindelaytime!);"
200 INPUT " seconds"; Stepdelay!
210 LOCATE 8,5:PRINT "Enter the Initial Hold Time, in seconds (t > Minholdtime!);"
220 INPUT " seconds"; Initialhold!
230 IF Type% = 1 THEN GOTO 290
240 IF Type% = 3 THEN GOTO 290
250 LOCATE 10,5:PRINT "Enter the Final Hold Time, in seconds (t > Minholdtime!);"
260 INPUT " seconds"; Finalhold!
270 LOCATE 12,5:PRINT "Enter the Delta Hold Time, in seconds (t > Minholdtime!);"
280 INPUT " seconds"; Deltahold!
290 IF Stepdelay! < Mindelaytime! OR Initialhold! < Minholdtime! THEN GOSUB 10100
300 LOCATE 20,5:COLOR 7,0
310 INPUT "Do you need to change any of the above (Default to No)"; Chk$
320 Chk$ = MID$(Chk$,1,1)
330 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
340 EXIT
9999 STOP
10000 REM
10001 REM *****************************************************
10002 REM * This error routine takes care of any expected errors. *
10003 REM * If an unexpected error occurs, the program prints the *
10004 REM * procedure name, error number and line number to screen *
10005 REM * then ends the program. *
10006 REM *
10007 REM *
10008 REM *****************************************************
10009 REM
10010 IF ERR = 1008 THEN RESUME
10020 LOCATE 23,5:COLOR 7,0
10030 PRINT "Sorry the Procedure ";
10040 COLOR 20,0:PRINT "settime";
10050 COLOR 7,0:PRINT " has bombed. The error number is ";
10059 PRINT Chk$;
10060 PRINT " from line ";
10069 STOP
10070 REM
10100 REM
10101 REM *****************************************************
10102 REM *
10103 REM * This error is given if either the initial hold time or *
10104 REM * the step delay time is less than the asked for values. *
10105 REM * It then returns them to the beginning of this routine. *
10106 REM *
10107 REM *****************************************************
10108 REM
10110 LOCATE 20,5:COLOR 7,0
10120 PRINT "The times you have chosen are not within the meter's resolution."
10130 LOCATE 21,5:PRINT "Please choose different parameters..."
TERS.
10140 Tyme! = TIMER + 5
10150 IF TIMER < Tyme! THEN GOTO 10150
10160 RETURN,100
10170 STOP
END PROCEDURE

PROCEDURE: Setbias
STRING: Chk$[3]

9 REM
10 REM ***********************************************
11 REM *
12 REM * The Procedure SETBIAS sets the different
bias voltages *
13 REM *
14 REM * to call: SETBIAS (INITIALBIAS!,FINALBIAS *
15 REM *
30 REM * real arg:initialbias!/>VAR,deltabias!/>VAR,
finalbias!/>VAR *
32 REM *
40 REM ***********************************************

41 REM
100 ON ERROR GOTO 10000
110 LOCATE 3,1
120 DO 21 TIMES
130 PRINT SPC(79)
140 REPEATE
150 LOCATE 4,25:COLOR 4,0
160 PRINT "This sets the Bias Voltages"
170 COLOR 5,0
180 LOCATE 6,5:INPUT "Enter the Initial Bias Voltage
(-99 to 99 V)";Initialbias!
190 LOCATE 8,5:INPUT "Enter the Final Bias Voltage (-
99 to 99 V)";Finalbias!
200 LOCATE 10,5:INPUT"Enter the Delta Bias Voltage (>
abs(0.01) V";Deltabias!
210 LOCATE 20,5:COLOR 7,0
220 INPUT "Do you need to change any of the above (De
faults to No)";Chk$
230 Chk$ = MID$(Chk$,1,1)
240 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
250 EXIT
10000 REM
10001 REM ***********************************************
These are the error messages which take care of the expected errors. If an unexpected error takes place an error messages is given along with the line number and the program stops.

**------------------------------**

**procedure: Setsamples**

**STRING: Chk$**

**15 REM * to call: SETSAMPLES (samples!)**
**30 REM * REAL ARG: SAMPLES!/var**

**100 ON ERROR GOTO 10000**
**110 LOCATE 3,1**
**120 DO 21 TIMES**
**130 PRINT SPC(79)**
**140 REPEAT**
**150 LOCATE 4,25:COLOR 4,0**
**160 PRINT "This sets the Number of Samples"**
**170 COLOR 5,0**
**180 LOCATE 6,5:INPUT "Enter the number of Samples needed ( ) 1)";SAMPLES!**

**190 LOCATE 20,5:COLOR 7,0**
**200 INPUT "Do you want to change any of the above (Default to No)";Chk$$**
**210 Chk$ = MID$(Chk$,1,1)**
**220 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110**
**230 SAMPLES! = INT(SAMPLES!)**
**240 EXIT**
**10000 STOP**

**------------------------------**
10003 REM * This corrects any expected errors. If an
error is
10004 REM * unexpected the program bombs and an error
message with
10005 REM * its line number.

10006 REM *

10007 REM ************************************************************************

10008 REM
10010 IF ERR = 1008 THEN RESUME
10020 LOCATE 1,5:COLOR 7,0
10030 PRINT "The procedure ";
10040 COLOR 20,0:PRINT "SETSAMPLES ";
10050 COLOR 7,0:PRINT "has bombed. The error is ";ERR;
10060 STOP
10060 STOP
END PROCEDURE

PROCEDURE: Setpulse
STRING: Chk$[3]

9 REM
10 REM ************************************************************************

11 REM *

12 REM * The Procedure SETPULSE allows the user to
enter the
13 REM * Initial High Pulse, Final High Pulse, Delta
14 REM * High Pulse and Low Pulses for the voltages in the C-t
15 REM * program

16 REM * to call: SETPULSE (INITIALPULSE!,FINALPULSE!,

17 REM * DELTAPULSE!,LOWPULSE!)

18 REM *

20 REM * REAL ARG: Initialpulse! VAR, Finalpulse!
/VAR,
21 REM * Deltapulse!/VAR
22 REM *

30 REM * STRING: Chk$[3]
32 REM *
*  
40 REM 40 REM 42 REM
110 LOCATE 3,1
120 DO 21 TIMES
130 PRINT SPC(79)
140 REPEAT
150 LOCATE 4,25:COLOR 4,0
160 PRINT "This sets the Pulse Voltages"
165 COLOR 5,0
170 LOCATE 6,5:INPUT "Enter the Initial High Pulse (-7 to +7 V)";Initialpulse!
180 LOCATE 8,5:INPUT "Enter the Final High Pulse (-7 to +7 V)";Finalpulse!
190 LOCATE 10,5:INPUT "Enter the Delta High Pulse ( 0.01 V)";Deltapulse!
200 LOCATE 12,5:INPUT "Enter the Low Pulse (-7 to +7 V)";Lowpulse!
210 LOCATE 20,5:COLOR 7,0
220 INPUT "Do you want to change any of the above (Defaults to No)";Chk$
230 Chk$ = MID$(Chk$,1,1)
240 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
250 IF Deltapulse! = 0 THEN Finalpulse! = 0 THEN Finalpulse! = 0
260 EXIT
END PROCEDURE

PROCEDURE: Copytodisk
INTEGER: Drive%
STRING: Chk$[3],R$[?]
20 REM * integer:drive%
30 REM * string:chk$[4]
40 REM * STRING ARG:FILE$
100 Drive% = 1
110 CLS:COLOR 3,0:LOCATE 12,5
120 PRINT "Do you want to copy the files to a ";
130 COLOR 26,0
140 ON Drive% GOSUB 1000,2000
150 LOCATE 12,44:COLOR 3,0:INPUT " floppy diskette (Defaults to No)";Chk$
160 Chk$ = MID$(Chk$,1,1)
170 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 180 ELSE GO TO 280
180 LOCATE 15,5:COLOR 6,0
190 PRINT "Please insert a BLANK FORMATTED ";
200 COLOR 24,0
210 ON Drive% GOSUB 1000,2000
220 LOCATE 15,7:COLOR 6,0:PRINT " floppy diskette into drive";
230 ON Drive% GOSUB 1050,2050
240 PRINT " and press any key to begin copying."
250 A$ = INKEY$: IF A$ = "" THEN GOTO 250
260 CLS
270 ON Drive% GOSUB 1100,2100
280 Drive% = Drive% + 1
290 IF Drive% < 3 THEN GOTO 110 ELSE EXIT
1000 REM
1010 PRINT " 1.2 Mbyte";
1020 RETURN
1050 PRINT " A";
1060 RETURN
1100 SHELL "\SEMI\COPYCTOA.BAT " + FILES$
1110 RETURN
2000 REM
2010 PRINT " 360 Kbyte";
2020 RETURN
2050 PRINT " B";
2060 RETURN
2100 SHELL "\SEMI\COPYCTOB.BAT " + FILES$
2110 RETURN
END PROCEDURE

PROCEDURE: Timedelay
REAL: Tyme!
22 REM * integer arg:delay%/opt = 5
32 REM * real: tyme!
100 Tyme! = TIMER
110 IF TIMER < Tyme! + Delay% THEN GOTO 110
120 EXIT
END PROCEDURE

PROCEDURE: Clearscreen
INTEGER: M%
1 REM
2 REM *****************************************************
**************
3 REM *
* 5 REM * to call: CLEARSCREEN (ROW%)
* 19 REM *
20 REM * INTEGER ARG:row%
* 21 REM *
31 REM *****************************************************
**************
32 REM
100 FOR M% = Row% TO 23
110 LOCATE M%,1:PRINT SPC(79)
120 NEXT M%
130 EXIT
END PROCEDURE

' MAIN Program:

200 REM
201 REM******************************************************************************
202 REM *
203 REM * This section prints a welcome message to the screen, *
204 REM * clears all variables, changes the drive and directory *
205 REM * to "C:", sets the appropriate devices to remote and *
206 REM * initializes the error handler. *
207 REM *
208 REM******************************************************************************
209 REM
210 CLS:STATUSLINE OFF
220 COLOR 2,0,8:LOCATE 12,30
230 PRINT "MEDUSA welcomes you"
240 LOCATE 14,12:PRINT "Materials and Electronic Device, Ultimate System Analyzer"
250 Timedelay (5)
260 CLEAR
270 DNT ERRDR GOSUB 9500
280 DRIVE$ = "C:";DIR$ = "\"
290 PARAM$ = "INIT/1/H310/P/";GOSUB 50000
300 PARAM$ = "SDR/5,12,16,17,8/";GOSUB 50000
310 REM
311 REM******************************************************************************
312 REM *
313 REM * This checks to see if the batch file has come from *
314 REM * the program "RUNIT" and if it has it branches *
315 REM * see if the user wants to use the same information *
316 REM * block. If the batch file hasn't come from " *
317 REM * a message about turning on the printer is sent to *

318 REM * screen and the printer is sent a code to configure it. *
319 REM *
320 REM ************************************************************************

321 REM
330 OPEN "\DATA\REDO" FOR INPUT AS #1
340 GOSUB 9000
350 COLOR 2,0,0:CLS:LOCATE 12,25
360 PRINT "Please turn on the printer"
390 Timedelay(3)
400 GOTO 1210
401

REM*************402 REM *
403 REM * information to correctly identify the sample *
405 REM *
406 REM ************************************************************************

407 REM
410 Redo% = 0
420 COLOR 2,0:CLS:LOCATE 3,5
430 INPUT "Enter the Directory number (between 1 and 999)";File$
440 Information$(1) = File$
450 IF VAL(File$) ( 1 OR VAL(File$) ) 999 THEN GOTO 420
460 Filecheck (File$,Redo%)
470 IF Redo% = 1 THEN GOTO 410
480 Information$(2) = DATE$
490 COLOR 2,0
500 LOCATE 5,5
505 LINE INPUT "Enter Experimenter 's Name. ";Information$(3)
510 LOCATE 7,5
515 LINE INPUT "Enter the sample number. ";Information$(4)
520 LOCATE 9,5
525 LINE INPUT "Enter the Cryostat chronometer reading. ";Information$(5)
530 LOCATE 11,5
535 LINE INPUT "Enter any comments. ";Information$(6)
760 LOCATE 20,5:COLOR 7,0
770 INPUT "Do you want to change any of the above (Defaults to No)";Chk$
780 Chk$ = MID$(Chk$,-1,1)
790 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 800 ELSE GOTO 1510
800 REM
801 REM ********************************************
802 REM *
803 REM * This allows the user to change any part of t
804 REM * the information block, using the various Procedu
805 REM *
806 REM ********************************************
807 REM
810 Number% = 0:Row% = 0
820 Title$ = "INFORMATION CHANGES"
830 Title (Title$)
840 RESTORE,60010
850 READ Maxloop%
860 DO Maxloop% TIMES
870 READ Name$
880 Menu (Number%,Rownumber%,Name%,Row%)
890 REPEAT
900 Name$ = "Finished changing information"
910 Finish (Rownumber%,Number%,Name$)
920 Border (Rownumber%)
930 DO
940 Clearscreen (20)
950 COLOR 6,0:LOCATE 20,5
960 INPUT "Enter which number you want to change";Choice%
970 IF Choice% < 1 OR Choice% > Number% THEN GOTO 940
980 IF Choice% = Number% THEN EXIT TO,1510
990 Placeampersand (Number%,Choice%)
995 Clearscreen (20)
1000 LOCATE 20,3;COLOR 4,0
1010 PRINT "The current information is: ";Information$(Choice%)
1020 LOCATE 22,1;LINE INPUT ". What is the new informatio
n? ";Information$(Choice%)
1030 IF Choice% <> 1 THEN GOTO 1100
1040 Redo% = 0
1050 Clearscreen (20)
1060 Filecheck (Information$(1),Redo%)
1070 IF Redo% = 1 THEN GOTO 1090
1080 File$ = Information$(1)
1090 Information$(1) = File$
1100 Removeampersand (Number%,Choice%)
1110 REPEAT
1200 REM
1201 REM ****************************************************
---------
1202 REM *

1203 REM * This section allows the user to choose which
* type of *
1204 REM * temperature run he wishes. The choices are a
* 1) Time run *
1205 REM * 2) Temp run; 3) Room Temp; 4) Exit to graphs *
1206 REM *

1207 REM *********************************************************

************
1208 REM
1210 Number% = 0: Row% = 0
1220 Title$ = "Menu for choosing TIME/TEMPERATURE Run"
1230 Title (Title$)
1240 RESTORE, 60410
1250 READ Maxloop%
1260 DO Maxloop% TIMES
1270 READ Name$
1280 Menu (Number%, Rownumber%, Name$, Row%)
1290 REPEAT
1300 Name$ = "EXIT to Graphing Routines"
1310 Finish (Rownumber%, Number%, Name$)
1320 Border (Rownumber%)
1330 Clearscrew (20)
1340 COLOR 6, 0: LOCATE 20, 5
1350 INPUT "Choose which type of run"; Choice%
1360 IF Choice% < 1 OR Choice% > Number% THEN GOTO 1330
1370 IF Choice% = Number% THEN GOTO 8910
1380 Expt% (26) = Choice%
1390 RESTORE, 60010
1400 READ M%
1410 M% = M% + 1
1420 RESTORE, 60600
1430 DO Choice% TIMES
1440 READ Information$(M%)
1450 REPEAT
1455 IF INSTR(UPPER$(Chk$), "Y") = 0 THEN GOTO 410
1500 REM
1501 REM *********************************************************

************
1502 REM *

1503 REM * This section allows the user to choose between groups. *
1504 REM *

1505 REM *********************************************************
1506 REM
1510 Number% = 0: Row% = 0
1520 RESTORE, 60100
1530 Title$ = "GROUP TYPES"
1540 Title (Title$)
1550 READ Maxloop%
1560 DO Maxloop% TIMES
1570 READ Name$
1580 Menu (Number%, Rownumber%, Name$, Row%) REPEAT
1590 RESTORE, 60200
1600 Border (Rownumber%)
1610 Clearscreen (20)
1620 COLOR 6, 0; LOCATE 20, 5
1630 INPUT "Which group of experiments do you want to perform"; Choice%
1640 IF Choice% < 1 OR Choice% > Number% THEN GOTO 1610
1650 Expt%(25) = Choice%
1700 REM
1701 REM ???????????????????????????????????????????????????????
1702 REM *
1703 REM * This section allows the user to pick which experiments *
1704 REM * he wants to run from the group chosen previously *
1705 REM *
1706 REM ???????????????????????????????????????????????????????
1707 REM
1710 Number% = 0: Row% = 0
1720 RESTORE, 60200
1730 Title$ = "EXPERIMENTAL MENU"
1740 Title (Title$)
1750 READ B%
1760 FOR M% = 1 TO B%
1770 READ Group%(M%)
1780 NEXT M%
1790 GOSUB 41000 'Used to set Read Pointer
1800 DO Group%(Expt%(25)) TIMES
1810 READ Name$
1820 Menu (Number%, Rownumber%, Name$, Row%) REPEAT
1830 REM
1840 Name$ = "Finished choosing experiments"
1850 Finish (Rownumber%, Number%, Name$)
1860 Border (Rownumber%)
1870 DO
1880 Clearscreen (20)
1890 LOCATE 20, 5; COLOR 6, 0
1900 INPUT "Which experiment do you want to run"; Choice%
1910 IF Choice% < 1 OR Choice% THEN GOTO 1880
1920 IF Choice% = Number% THEN EXIT
1930 Placeampersand (Number%,Choice%)
1940 Expt%(Choice%) = Choice%
1950 REPEAT
1960 REM
1961 REM -----------This asks if any experiments are to be deleted---------
1962 REM
1970 DO
1980 Clearscreen (20)
1990 LOCATE 20,1:COLOR 7,0
2000 INPUT "Do you need to delete any of the above (Defaults to No)"; Chk$
2010 Chk$ = MID$(Chk$,1,1)
2020 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 2030 ELSE EXIT
2030 Clearscreen (20)
2040 LOCATE 20,1:COLOR 6,0
2050 INPUT "Which of the above do you need to delete"; Choice%
2060 IF Choice% < 1 OR Choice% THEN GOTO 20 90
2070 Removeampersand (Number%,Choice%)
2080 Expt%(Choice%) = 0
2090 REPEAT
2100 REM
2102 REM -----------This asks if any experiments are to be added----------
2103 REM
2110 DO
2120 Clearscreen (20)
2130 LOCATE 20,1:COLOR 7,0
2140 INPUT "Do you want to add any to the above (Defaults to NO)"; Chk$
2150 Chk$ = MID$(Chk$,1,1)
2160 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 2170 ELSE EXIT
2170 Clearscreen (20)
2180 COLOR 6,0;LOCATE 20,5:INPUT "Which experiment do you want to add";Choice%
2190 IF Choice% < 1 OR Choice% THEN GOTO 20 20
2200 Expt%(Choice%) = Choice%
2210 Placeampersand (Number%,Choice%)
2220 REPEAT
2230 REM
2231 REM ************************************************************
2232 REM *
2233 REM * This section returns the program to the Temperature *
2234 REM * run type choosing section if no experiments were chosen *
2235 REM *
2236 REM ******************************************************************************
2237 REM
2238 FOR M? = 1 TO 24
2239 IF Expt%(M?) < 0 THEN EXIT TO, 2500
2240 NEXT M?
2241 CLS: COLOR 7, 0: LOCATE 12, 30
2242 PRINT "No experiments were chosen"
2243 Timedelay (4)
2244 GOTO 1210
2245 REM
2246 REM ******************************************************************************
2247 REM * This allows the user to set the Cryostat temperature(s) *
2248 REM * and the times if it is a TIME run. *
2249 REM *
2250 REM ----------
2251 REM ON Expt%(26) GOSUB 2600, 2700, 2900
2252 REM STOP
2253 REM
2254 REM ----------
2255 REM Parameter!(25, 1) = 30.0
2256 REM Parameter!(25, 2) = 600.0
2257 REM Parameter!(25, 3) = 1.0
2258 REM GOSUB 3500
2259 REM Timeparam(Parameter!(25, 5), Parameter!(25, 6))
2260 RETURN, 4000
2261 STOP
2700 REM
2701 REM *********************************************************
2702 REM *
2703 REM * This subroutine allows the user to set the temperatures *
2704 REM * for a temp run.
2705 REM *
2706 REM *********************************************************
2707 REM
2710 Parameter!(25,1) = 30.0
2720 Parameter!(25,2) = 600.0
2730 Parameter!(25,3) = 1.0
2740 GOSUB 3500
2750 RETURN, 4000
2799 STOP
2900 REM
2901 REM *********************************************************
2902 REM *
2903 REM * This subroutine allows the user to set the room *
2904 REM * temperature.
2905 REM *
2906 REM *********************************************************
2907 REM
2910 GOSUB 3500
2930 RETURN, 4000
3499 STOP
3500 REM
3501 REM *********************************************************
3502 REM *
3503 REM * This section goes to the procedure SETTEMPPARAM to *
3504 REM * allow the user to set what temperatures for the *
3505 REM * cryostat.
3506 REM *
3507 REM ******
3508 REM
3510 Col% = 20
3520 Expt$ = "Cryostat"
3530 Name$ = "limits"
3540 Settempparam (Expt%(26), Col%, Parameter!(25,1), Parameter!(25,2), Parameter!(25,3), Expt$, Name$)
3550 Parameter!(25,4) = Parameter!(25,1)
3560 RETURN
3999 STOP
4000 REM
4001 REM ****************************************************************************************************

4002 REM *
4003 REM * This section branches to the parameter setting *
4004 REM * subroutines selected by the user. *
4005 REM *
4006 REM ****************************************************************************************************

4007 REM
4010 ON Expt%(25) GOTO 4100, 4200, 4300, 4400
4020 STOP
4100 REM
4101 REM ****************************************************************************************************

4102 REM *
4103 REM * This is to branch to the set-up routine for *
4104 REM * the experiments in group I. *
4105 REM *
4106 REM ****************************************************************************************************

4107 REM
4110 L% = 1
4120 ON Expt%(L%) GOSUB 10000, 10000, 12000, 13000
4130 L% = L% + 1
4140 IF L% (= Group%(Expt%(25))) THEN GOTO 4120
4150 GOTO 8010
4199 STOP
4200 REM
4201 REM ****************************************************************************************************

4202 REM *
4203 REM * This is to branch to the set-up routine for the experiments in group II.
4204 REM *
4205 REM *
4206 REM ************************************************************************************
4207 REM
4210 LX = 1
4220 ON Expt%(L%) GOSUB 14000,15000
4230 L% = L% + 1
4240 IF L% <= Group%(Expt%(25)) THEN GOTO 4220
4250 GOTO 8010
4299 STOP
4300 REM
4301 REM ******************************************************************************
4302 REM *
4303 REM * This is the branch to Group III. Since no parameters need to be set it goes immediately to the print statements.
4305 REM *
4306 REM *
4307 REM ******************************************************************************
4308 REM
4310 CLS:LOCATE 1,20:COLOR 2,0
4320 PRINT "This sets the";
4330 COLOR 18,0:PRINT " MOBILITY";
4340 COLOR 2,0:PRINT " Parameter Settings"
4350 COLOR 14,0 : SET CURSOR 13,6 : INPUT "Enter the Current Bias in mA (9 mA ). ",Parameter!(1,17)
4360 IF Parameter!(1,17) <= 0 OR Parameter!(1,17) > 9 THEN GOTO 4310
4370 Parameter!(1,17) = Parameter!(1,17) * 1E-03
4380 GOTO 8010
4400 REM
4401 REM ******************************************************************************
4402 REM *
4403 REM * This is the branch to Group IV. Since no parameters need to be set it goes immediately to the print statements.
**4406 REM **

**4407 REM **********************************************************

**4408 REM
4410 CLS:LOCATE 1,15:COLOR 2,0
4420 PRINT "This sets the";
4430 COLOR 18,0:PRINT " 4-point Resistivity";
4440 COLOR 2,0:PRINT " Parameter Settings"
4450 COLOR 14,0:SET CURSOR 13,6:INPUT "Enter the Current Bias in mA (100 mA ). ",Parameter!(1,17)
4460 IF Parameter!(1,17) (= 0 OR Parameter!(1,17) > 100 THEN GOTO 4410
4480 GOTO 8010

**8000 REM
8001 REM **********************************************************

**8002 REM *
8003 REM * This section saves all the information necessary to
8004 REM * running the experiments. It also prints out a list
8005 REM * of the information block, the cryostat settings,
8006 REM * time settings and all the parameters for each
8007 REM * experiment chosen.

**8008 REM *

**8009 REM **********************************************************

**8010 REM
8011 REM --------This prints and saves the information block-----
8012 REM
8014 LPRINT CHR$(24);CHR$(27);CHR$(58)
8015 LPRINT CHR$(27);CHR$(68);CHR$(35);CHR$(45);CHR$(55);CHR$(65);CHR$(75);CHR$(85);CHR$(0)
8020 OPEN "\DATA\" + File$ + "\INFO" FOR OUTPUT AS #1
8030 RESTORE,60010
8040 READ M%
8050 Maxloop% = M% + 1
8060 LPRINT CHR$(9);CHR$(9);CHR$(9);CHR$(9);CHR$(9);CHR$(9);CHR$(9);CHR$(9);CHR$(9);CHR$(9);DATE$
8070 FOR M% = 1 TO Maxloop%
8080 READ Name$
8090 PRINT #1,Name$
8100 PRINT #1,Information$(M%)
8110 IF MX = 2 THEN GOTO 8140
8120 LPRINT SPC(10);Name$
8130 LPRINT SPC(10);Information$(MX)
8140 NEXT MX
8150 CLOSE #1
8160 REM
8161 REM —————-—This prints and saves the Cryostat and Time Settings————-
8162 REM
8170 OPEN "\DATA\PARAM" FOR OUTPUT AS #1
8180 RESTORE,60320
8190 LPRINT CHR$(10);CHR$(10)
8200 LPRINT SPC(10);"These are the Cryostat and Time settings"
8210 LPRINT FOR M% = 1 TO 6
8220 READ Name$
8230 PRINT #1,Parameter!(25,M%)
8240 IF M% = 4 THEN GOTO 8270
8250 LPRINT SPC(10);Name$,Parameter!(25,M%)
8260 NEXT MX
8270 NEXT M%
8280 CLOSE #1
8290 LPRINT CHR$(10)
8300 REM
8301 REM ————This prints the parameter settings———-
8310 LPRINT SPC(10);"These are the parameter settings for the chosen experiments"
8320 GOSUB 40000
8330 LPRINT
8340 LPRINT SPC(10);"PARAMETER";CHR$(09);
8350 FOR M% = 1 TO Group%(Expt%(25))
8360 READ Name$
8370 IF Expt%(M%) = 0 THEN GOTO 8390
8380 LPRINT Name$;CHR$(09);
8390 NEXT M%
8400 LPRINT CHR$(10)
8410 RESTORE,60310
8420 READ Maxloop%
8430 FOR M% = 1 TO Maxloop%
8440 READ Name$
8450 IF M% > 3 AND M% < 7 THEN GOTO 8520
8460 LPRINT SPC(10);Name$;CHR$(09);
8470 FOR N% = 1 TO Group%(Expt%(25))
8480 IF Expt%(N%) = 0 THEN GOTO 8500
8490 LPRINT Parameter!(N%,M%);CHR$(09);
8500 NEXT N%
8510 LPRINT CHR$(13);
8520 NEXT M%
8530 LPRINT CHR$(12)
8540 REM
8541 REM --------This saves which experiments are to be run--------
8542 REM
8550 OPEN "\DATA\EXPT" FOR OUTPUT AS #1
8560 PRINT #1, File$
8570 FOR M% = 1 TO 30
8580 PRINT #1, Expt%(M%)
8590 NEXT M%
8600 CLOSE #1
8610 REM
8611 REM --------This saves the parameters for the experiments--------
8612 REM
8620 OPEN "\DATA\PARAM" FOR APPEND AS #1
8630 RESTORE, 60310
8640 READ Maxloop%
8650 FOR N% = 1 TO Group%(Expt%(25))
8660 IF Expt%(N%) = 0 THEN GOTO 8700
8670 FOR M% = 1 TO Maxloop%
8680 PRINT #1, Parameter!(N%, M%)
8690 NEXT M%
8700 NEXT NX
8710 CLOSE #1
8800 REM
8801 REM ************************************************************
8802 REM *
8803 REM * This exits the "PARAMETER" program and goes to the *
8804 REM * program "RUNIT". *
8805 REM *
8806 REM ************************************************************
8807 REM
8810 REM
8820 SYSTEM
8830 STOP
8840 REM
8850 REM ************************************************************
8860 REM *
8870 REM * This allows the user to go directly to the graphics *
8880 REM * routines if he has chosen to do so in the TEMP RUN *
8890 REM * section.
166

```basic
8906 REM *
8907 REM ************************************************************
8908 REM
8910 OPEN '"\DATA\END" FOR OUTPUT AS #1
8920 PRINT #1,"Howdy there pardner"
8930 CLOSE
8940 CLS
8950 SYSTEM
8999 STOP
9000 REM
9001 REM ************************************************************

9002 REM *
9003 REM This subroutine allows the user to reuse the
9004 REM information block used in the previous experiment. *
9005 REM *
9006 REM *************************************************************

9007 REM
9010 CLOSE
9013 LPRINT CHR$(24)
9020 CLS:COLOR 10,0,0:LOCATE 12,5
9030 INPUT "Do you want to use the same information block
(Defaults to No)";Chk$
9040 Chk$ = MID$(Chk$,1,1)
9050 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 9090 ELSE GOTO 9260
9090 OPEN '"DRTR\REDO" FOR INPUT QS #1
9100 INPUT #1,FILE1$
9110 CLOSE #1
9160 RESTORE,60010
9170 READ Maxloop%
9180 OPEN '"\DATA\" +FILE1$ + '"\INFO" FOR INPUT AS #1
9190 INPUT #1,BS$,BS$
9200 FOR M% = 2 TO Maxloop%
9210 INPUT #1,BS$,Information$(M%)
9220 NEXT M%
9230 CLOSE #1
9233 Clearscreen (17):COLOR 10,0,0
9234 LOCATE 17,5:INPUT "Enter the directory number (betwe
en 1 and 999)";File$
9235 IF VAL(File$) < 1 OR VAL(File$) > 999 THEN GOTO 9233
9236 Information$(1) = File$
9237 Redo% = 0
```
REM Filecheck (File$, Redo%)
9239 IF Redo% = 1 THEN GOTO 9233
9240 KILL "\DATA\REDO"
9250 GOTO 1210
9260 KILL "\DATA\REDO"
9270 GOTO 1210
9499 STOP
9500 REM ************************************************
9501 REM * 
9502 REM * This is the ERROR handling routines. It checks to see * 
9503 REM * what error is and attempts to fix it. 
9504 REM * 
9505 REM ************************************************
9506 REM 
9510 IF ERR = 1008 THEN RESUME NEXT 
9520 IF ERR = 1001 THEN RESUME, 350 
9530 IF ERR = 1007 THEN RESUME, 350 
9540 CLS:COLOR 7, 0:LOCATE 11, 5 
9550 PRINT "Sorry the main program is bombing. This is the error "; ERR 
9560 LOCATE 12, 5:PRINT "The line number is "; ERL 
9570 STOP 
9999 STOP 
10000 REM 
10001 REM ************************************************
10002 REM * 
10003 REM * This is the Capacitance vs. Time and Conductance vs. 
10004 REM * Time Subroutine. It sets the parameters and then sends * 
10005 REM * them to the C-V meter and Pulse Generator for checking. * 
10006 REM * 
10007 REM ************************************************
10008 REM 
10010 CLS:LOCATE 1, 19:COLOR 2, 0 
10020 IF L$ = 1 GOTO 10060 'If C-t skip next 2 instructions 
10030 Function$ = "FN6" 
10040 Name$ = "G-t" 
10050 GOTO 10080 'Since G-t skip C-t
set-up
10060 Function$ = "FN5"
10070 Name$ = "C-t"
10080 COLOR 2,0:PRINT "This sets the ";
10090 COLOR 18,0:PRINT Name$;
10100 COLOR 2,0:PRINT " Experiment Parameters"
10110 Parameter!(L%,1) = Parameter!(25,1)
10120 Parameter!(L%,2) = Parameter!(25,2)
10130 Parameter!(L%,3) = Parameter!(25,3)
10140 Parameter!(L%,4) = Parameter!(25,4)
10150 IF Expt%(25) = 3 OR Expt%(26) = 1 THEN GOTO 10220
10160 REM
10161 REM ----Sets Temperature Parameters-----
10162 REM
10170 Expt$ = Name$ + " Experiment"
10180 Name$ = "Settings"
10190 Col% = 20
10200 Settempparam (Time%,Col%,Parameter!(L%,1),Parameter!(L%,2),Parameter!(L%,3),Expt$,Name$)
10210 REM
10211 REM ------Sets Bias Voltages------
10212 REM
10220 Setbias (Parameter!(L%,17),Parameter!(L%,18),Parameter!(L%,19))
10230 Parameter!(L%,4) = Parameter!(L%,1)
10240 REM
10241 REM -----Sets Number of Samples-----
10242 REM
10250 Setsamples (Parameter!(L%,20))
10260 REM
10261 REM ------Sets Pulse Times------
10262 REM
10270 Settime (Parameter!(L%,13),Parameter!(L%,14),Parameter!(L%,15),Parameter!(L%,16),1.0E-05,1.0E-05,0)
10280 REM
10281 REM ------Sets Pulse Voltages------
10282 REM
10290 Setpulse (Parameter!(L%,21),Parameter!(L%,22),Parameter!(L%,23),Parameter!(L%,24))
10300 REM
10301 REM ******************************************
10302 REM *  
10304 REM * This section checks the parameters with the  
10305 REM * C-V meter *  
10306 REM * and Pulse generator.  
10307 REM ******************************************
169

*************
10308 REM
10310 REM----Checks the Initial High Pulse Voltage with the Pulse Generator----
10311 REM
10320 DO 2 TIMES
10330 PARAM$ = "SER.POLL/13/";GOSUB 50000
10340 PARAM$ = "SDR/13/";GOSUB 50000
10350 DATA_STRING$ = "M4,CTO,T1,W1,HIL" + STR$(Parameter!(LX,21)) + "V,LOL" + STR$(Parameter!(LX,24)) + "V"
10360 PARAM$ = "WR.STR/13//EOS/";GOSUB 50000
10370 PARAM$ = "RD.STR/13//EOS/";GOSUB 50000
10380 PARAM$ = "SER.POLL/13/";GOSUB 50000
10390 REPEGT
10400 IF POLL_RESPX GND &H40 = 64 THEN GOSUB 10740
10410 IF Parameter!(LX,23) = 0 THEN GOTO 10520
10420 REM
10421 REM—-—Checks the Final High Pulse Voltage with the Pulse Generator——
10422 REM
10430 DO 2 TIMES
10440 PARAM$ = "SER.POLL/13/";GOSUB 50000
10450 DATA_STRING$ = "M4,CTO,T1,W1,HIL" + STR$(Parameter!(LX,22)) + "V,LOL" + STR$(Parameter!(LX,24)) + "V"
10460 PARAM$ = "WR.STR/13//EOS/";GOSUB 50000
10470 PARAM$ = "RD.STR/13//EOS/";GOSUB 50000
10480 PARAM$ = "SER.POLL/13/";GOSUB 50000
10490 REPEGT
10500 IF POLL_RESPX AND &H40 = 64 THEN GOSUB 10740
10510 PARAM$ = "SDL/13/";GOSUB 50000
10520 REM
10521 REM------Checks the Hold Times vs. Bias Voltage with the C-V meter------
10522 REM
10530 MX = 14
10540 NX = 17
10550 DATA_STRING$ = Function$ + "CN13TR3LE25A1PC" + STR$( Parameter!(LX,NX)) + ";PN" + STR$(Parameter!(LX,20)) + ";PH" + STR$(Parameter!(LX,MX)) + ";PT" + STR$(Parameter!(LX, 13)) + "SW1"
10560 PARAM$ = "WR.STR/17//EOS/";GOSUB 50000
10565 DATA_STRING$ = "SWO"
10567 PARAM$ = "WR.STR/17//EOS/";GOSUB 50000
10570 PARAM$ = "SER.POLL/17/";GOSUB 50000
10580 IF POLL_RESPX AND &H40 = &H40 THEN GOSUB 10870
10590 NX = NX + 1
10600 IF Parameter!(LX,19) = 0 OR Parameter!(LX,18) = 0 THEN GOTO 10660
10610 IF NX = 18 THEN GOTO 10550
10620 IF Parameter!(LX,15) = 0 THEN GOTO 10660
10630 MX = MX + 1
10640 IF M% = 15 THEN GOTO 10540
10650 REM
10651 REM ————This checks to see if delaytime/holdtime is > 200———
10652 REM
10660 IF Parameter!(L%, 14) / Parameter!(L%, 13) > 200 THEN GOTO 10680
10670 IF Parameter!(L%, 16) / Parameter!(L%, 13) <= 200 THEN GOTO 10710
10680 COLOR 7, 0:CLS:LOCATE 13, 5
10690 PRINT "The delay time divided by the pulse times must be less than 200!"
10700 GOTO 10270
10710 PPRPMS = "SDC/13, 17/":GOSUB 50000
10720 RETURN
10730 STOP
10738 STOP
10740 REM ******************************************************
10741 REM This section prints the error messages if any of the parameters evoke an "illegal" call from the generator or C-V meter.
10742 REM
10743 REM Prints out error message for the Pulse Generator——
10744 REM
10745 REM
10746 REM ******************************************************
10747 REM
10748 REM ———Prints out error message for the Pulse Generator——
10749 REM
10750 DATA_STRING$ = "IERR"
10760 PARAM$ = "WR, STR/13//EOS/":GOSUB 50000
10770 PARAM$ = "RD, STR/13//EOS/":GOSUB 50000
10780 Clearscreen (3)
10790 LOCATE 10, 5:COLOR 7, 0
10800 PRINT "The settings chosen are not viable. You must select new settings."
10810 LOCATE 11, 5:PRINT "To aid you the error from the HP8 112 is "
10820 LOCATE 11, 47:COLOR 5, 0:PRINT MID$(DATA_STRING$, 2, 18) :COLOR 7, 0
10830 LOCATE 12, 5:PRINT "Please look in the manual on page s 3-21 to 3-23."
10840 PARAM$ = "SDL/13/":GOSUB 50000
10850 Timedelay(6)
10860 RETURN, 10280
10869 STOP
10870 REM
10871 REM -----Prints out error message for the C-V meter-----
10872 REM
10880 DATA_STRING$ = "ERR?"
10890 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
10900 PARAM$ = "RD.STR/17//EOS/":GOSUB 50000
10910 Startnoerror% = INSTR(DQTQ_STRING$,"ER00.0")
10920 IF Startnoerror% = 0 THEN RETURN
10930 Clearscreen (3)
10940 LOCATE 10,5;COLOR 7,0
10950 PRINT "The settings chosen are not viable. You must select new settings."
10960 PRINT "To aid you the error number from the HP4280 A is ";
10970 COLOR 5,0;PRINT DATA_STRING$;COLOR 7,0
10980 PRINT "Please look in the manual on pages 3-23 to 3-30.)"
10990 Timedelay(6)
11000 PARAM$ = "SDC/17/":GOSUB 50000
11010 RETURN,10220
11999 STOP
12000 REM
12001 REM **********************************************
12002 REM * This subroutine control the C-G-V experiment and allows *
12003 REM * the user to enter the parameter settings and checks for *
12004 REM * any possible errors. *
12005 REM *
12006 REM *
12007 REM **********************************************
12008 REM
12010 CLS;LOCATE 1,20;COLOR 2,0
12020 PRINT "This sets the"
12030 COLOR 18,0;PRINT " C-G-V";
12040 COLOR 2,0;PRINT " Parameter Settings"
12050 Parameter!(3,1) = Parameter!(25,1)
12060 Parameter!(3,2) = Parameter!(25,2)
12070 Parameter!(3,3) = Parameter!(25,3)
12080 Parameter!(3,4) = Parameter!(25,4)
12090 IF Expt%(26) = 3 OR Expt%(26) = 1 THEN GOTO 12150
12100 REM
12101 REM --------Sets the Temperature Parameters--
Expt$ = "C-B-V Experiment"

Col1% = 19
Name$ = "Settings"

Settempparam (Time%, Col1%, Parameter!(3, 1), Parameter!(3, 2), Parameter!(3, 3), Expt$, Name$)

---------Sets the Start, Stop and Delta Voltages---------

Svolt (Parameter!(3, 7), Parameter!(3, 8), Parameter!(3, 9), Parameter!(3, 10), Parameter!(3, 11))

---------Sets the Hold and Step Delay Time---------

Settime (Parameter!(3, 13), Parameter!(3, 14), Parameter!(3, 15), Parameter!(3, 16), 0.1, 0.1, 1)

---------------This checks the parameters entered with the C-V meter.---------------

DQTQ_STRING$ = "FN1IB2TR3PS"+STR$(Parameter!(3, 7)) +";PP"+STR$(Parameter!(3, 8)) +";PE"+STR$(Parameter!(3, 9)) +";PD"+STR$(Parameter!(3, 13)) +";PL"+STR$(Parameter!(3, 14))

PQRQMS = "wR.STR/17//EOS/";GOSUB 50000
PQRQMS = "SER.POLL/17/";GOSUB 50000
IF POLL_RESP% AND &H40 = 64 THEN GOSUB 12500

---------------This checks the first start, stop and delta voltages---------------

DATA_STRING$ = "FN1IB2TR3PS"+STR$(Parameter!(3, 8)) +";PP"+STR$(Parameter!(3, 10)) +";PE"+STR$(Parameter!(3, 11)) +";PD"+STR$(Parameter!(3, 13)) +";PL"+STR$(Parameter!(3, 14))

PARAM$ = "WR.STR/17//EOS/";GOSUB 50000
PARAM$ = "SER.POLL/17/";GOSUB 50000
IF POLL_RESP% AND &H40 = 64 THEN GOSUB 12500

---------------This checks the second start, stop and delta voltages---------------

DATA_STRING$ = "FN1IB2TR3PS"+STR$(Parameter!(3, 9)) +";PP"+STR$(Parameter!(3, 13)) +";PE"+STR$(Parameter!(3, 11)) +";PD"+STR$(Parameter!(3, 13)) +";PL"+STR$(Parameter!(3, 14))

PARAM$ = "WR.STR/17//EOS/";GOSUB 50000
PARAM$ = "SER.POLL/17/";GOSUB 50000
IF POLL_RESP% AND &H40 = 64 THEN GOSUB 12500
RETURN
12499 STOP
12500 REM
12501 REM ************************************************************
12502 REM *
12503 REM * This subroutine prints the error found by the C-V *
12504 REM * meter, then returns the program to reenter the *
12505 REM * parameters.
12506 REM *
12507 REM ***************************************************************
12508 REM
12509 DATA_STRING$ = "ERR?"
12510 PARAM$ = "WR.STR/17//EOS/";GOSUB 50000
12511 PARAM$ = "RD.STR/17//EOS/";GOSUB 50000
12512 Startnoerror% = INSTR(DATA_STRING$,"ER00.0")
12513 IF Startnoerror% <> 0 THEN RETURN
12514 Clearscreen (3)
12515 LOCATE 10,5:COLOR 7,0
12516 PRINT "The settings chosen are not viable. You must select new settings."
12517 PRINT "To aid you the error number from the HP4280 A is ";
12518 COLOR 5,0:PRINT DATA_STRING$;COLOR 7,0
12519 PRINT "Please look in the manual on pages 3-23 to 3-30."
12520 Timedelay(6)
12521 LOCATE 10,5:COLOR 7,0
12522 PRINT "This sets the";}
COLOR 18,0:PRINT " C-G";
COLOR 2,0:PRINT " Experiment Parameters"
Parameter!(4,1) = Parameter!(25,1)
Parameter!(4,2) = Parameter!(25,2)
Parameter!(4,3) = Parameter!(25,3)
Parameter!(4,4) = Parameter!(25,4)
IF Expt%(26) = 3 OR Expt%(26) = 1 THEN GOTO 13150
REM
-------Sets the Temperature Parameters----

Expt$ = "C-G Experiment" 'Sets title and names for the
Col% = 19 'Procedure
Name$ = "Settings"
Settempparam (Time%,Col%,Parameter!(4,1),Parameter!(4,2),Parameter!(4,3),Expt$,Name$)
REM
-------Sets the Bias Voltages---------

Setbias (Parameter!(4,17),Parameter!(4,18),Parameter!(4,19))
REM
*****************+****************************
************* *
This section checks to see if the parameters set are allowed by the C-V meter.
************* *

DQTQ_STRING$ = "FN1CN1IO1RA1MS2SL2TRPV" + STR$(Parameter!(4,17))
PQRQMS = "wR.STR/17/":GOSUB 50000
PQRQMS = "SER.POLL/17/":GOSUB 50000
IF POLL_RESP% = &H40 = 64 THEN GOSUB 13500
REM
-------This checks the start voltage------

DATA_STRING$ = "FN1CN1IO1RA1MS2SL2TRPV" + STR$(Parameter!(4,17))
PARAM$ = "WR.STR/17/:GOSUB 50000
PARAM$ = "SER.POLL/17/:GOSUB 50000
IF POLL_RESP% AND &H40 = 64 THEN GOSUB 13500
REM
-------This checks the final voltage------

DATA_STRING$ = "FN1CN1IO1RA1MS2SL2TRPV" + STR$(Parameter!(4,18))
PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
PARAM$ = "SER.POLL/17/":GOSUB 50000
IF POLL_RESP% AND &H40 = 64 THEN GOSUB 13500
PARAM$ = "SDC/17/":GOSUB 50000
RETURN
STOP
REM
PROGRAM NAME: "WR.STR/17//EOS/"
PARAM$ = "SER.POLL/17/":GOSUB 50000
IF POLL_RESP% AND &H40 = 64 THEN GOSUB 13500
PARAM$ = "SDC/17/":GOSUB 50000
RETURN
STOP
REM
PROGRAM NAME: "SER.POLL/17/"
GOSUB 50000
RETURN
STOP
REM
PROGRAM NAME: "SDC/17/"
GOSUB 50000
RETURN
STOP
REM
**********

This subroutine prints the error found by the C-V meter, then returns the program to reenter the parameters.

REM

DATA_STRING$ = "ERR?"
PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
PARAM$ = "RD.STR/17//EOS/":GOSUB 50000
Startnoerror% = INSTR(DATA_STRING$,"ERROR")
IF Startnoerror% = 0 THEN RETURN
Clearscreen (3)
LOCATE 10,5:COLOR 7,0
PRINT "The settings chosen are not viable. You must select new settings."
PRINT "To aid you the error number from the HP4280 is ";
COLOR 5,0;PRINT DATA_STRING$;COLOR 7,0
PRINT "Please look in the manual on pages 3-23 to 3-30."
Timedelay(6)
PARAM$ = "SDC/17/":GOSUB 50000
RETURN,13150
STOP
REM
PROGRAM NAME: "SDC/17/"
GOSUB 50000
RETURN
STOP
REM
**********

This subroutine prints the error found by the C-V meter, then returns the program to reenter the parameters.

REM

PROGRAM NAME: "SDC/17/"
GOSUB 50000
RETURN
STOP
REM
**********

This is the Current vs. Voltage subroutine. It sets up the parameters necessary to run the program. It also checks to see if the parameters are feasible.
14006 REM *
  *
14007 REM **********************************************************
14008 REM
14010 CLS:LOCATE 1,20:COLOR 2,0
14020 PRINT "This sets the ";
14030 COLOR 18,0:PRINT "I-V";
14040 COLOR 2,0:PRINT " Experiment Parameters"
14050 Parameter!(1,1) = Parameter!(25,1)
14060 Parameter!(1,2) = Parameter!(25,2)
14070 Parameter!(1,3) = Parameter!(25,3)
14080 Parameter!(1,4) = Parameter!(25,4)
14090 IF Expt%(26) = 3 OR Expt%(26) = 1 THEN GOTO 14150
14100 REM
14101 REM -----------Sets the Temperature Parameters-----
14110 Expt$ = "I-V Experiment"
14120 Col% = 20
14130 Name$ = "Settings"
14140 Settempparam (Time%,Col%,Parameter!(1,1),Parameter!(
1,2),Parameter!(1,3),Expt$,Name$)
14150 REM
14151 REM -----------Sets the Start, Stop and Delta Voltages--------
14160 Svolt (Parameter!(1,7),Parameter!(1,8),Parameter!(1,
9),Parameter!(1,10),Parameter!(1,11))
14170 REM
14171 REM -----------Sets the Hold and Step Delay Times--------
14180 Settime (Parameter!(1,13),Parameter!(1,14),Parameter!
(1,15),Parameter!(1,16),0.7,0.1,1)
14190 REM
14191 REM **********************************************************
14192 REM *
14193 REM * This checks the parameters set with the I-V meter. *
14194 REM *
14195 REM **********************************************************
14196 REM
14200 MX = 7
14210 NX = 8
14215 X$ = STR$(VAL(STR$(Parameter!(1,NX+1))))
14220 DATA_STRING$ = "F2I2L3PS" + STR$(Parameter!(1,MX)) +
14230 PARAM$ = "WR. STR/5//EOS/";GOSUB 50000
14240 DATA_STRING$ = "W7"
14250 PARAM$ = "WR. STR/5//EOS/";GOSUB 50000
14260 PARAM$ = "SER.POLL/5/";GOSUB 50000
14270 IF POLL_RESPX AND &H40 = 64 THEN GOSUB 14500
14280 IF Parameter!(1,11) = 0 THEN GOTO 14320
14290 M$ = M$ + 1
14300 N$ = N$ + 2
14310 IF M$ = 8 THEN GOTO 14215
14320 PARAM$ = "SDC/5/";GOSUB 50000
14330 RETURN
14499 STOP
14500 REM
14501 REM ************************************************
14502 REM *************
14503 REM *
14504 REM * This prints the error message sent in binary
14505 REM * by the I-V meter. It then sends the program back to
14506 REM * reenter the parameters.
14507 REM *
14508 REM ************************************************
14509 REM
14510 Clearscreen (3)
14520 COLOR 7,0;LOCATE 10,5
14530 PRINT "The settings chosen are not viable. You must
14540 select new settings."
14540 LOCATE 11,5
14550 PRINT "If you are not sure why this occurred please
14560 check the HP 4140B."
14560 LOCATE 12,5;PRINT "The BINARY error code is ";
14570 COLOR 5,0;PRINT MID$(BIN$(POLL_RESPX),8)
14580 Timedelay (6)
14590 PARAM$ = "SDC/5/";GOSUB 50000
14600 RETURN,14160
14999 STOP
15000 REM
15001 REM ************************************************
15002 REM *
15003 REM * This is the Capacitance vs. Voltage subroutine. It *
15004 REM * sets up the parameters necessary to run the program. *
15005 REM * It also checks to see if the parameters are feasible. *
15006 REM *
15007 REM ************************************************************
15008 REM
15009 CLS:LOCATE 1,20:COLOR 2,0
15010 PRINT "This sets the";
15011 COLOR 18,0:PRINT " C-V ";
15012 COLOR 2,0:PRINT "Experiment Parameters"
15013 Parameter!(2,1) = Parameter!(25,1)
15014 Parameter!(2,2) = Parameter!(25,2)
15015 Parameter!(2,3) = Parameter!(25,3)
15016 Parameter!(2,4) = Parameter!(25,4)
15017 IF Expt%(26) = 3 OR Expt%(26) = 1 THEN GOTO 15160
15100 REM
15101 REM ------Sets the Temperature Parameters------
15110 Expt$ = "C-V Experiment"
15120 Co1% = 20
15130 Name$ = "Settings"
15140 Settempparam (Time%,Col%,Parameter!(2,1>,Parameter!(2,2),Parameter!(2,3),Expt$,Name$)
15150 REM
15151 REM ------Sets the Start, Stop and Delta Voltages------
15160 DVdt (Parameter!(2,12),Parameter!(2,7),Parameter!(2,8),Parameter!(2,9))
15170 REM
15171 REM ------Sets the Hold Time ------
15180 Settime (Parameter!(2,13),Parameter!(2,14),Parameter!(2,15),Parameter!(2,16),0.1,0,3)
15190 REM
15191 REM *************************************************************
15192 REM *
15193 REM * This checks the parameters set with the I-V meter. *
15194 REM *
15195 REM *************************************************************
15196 REM
15004 REM * sets up the parameters necessary to run the program. *
15005 REM * It also checks to see if the parameters are feasible. *
15006 REM *
15007 REM **********************************************************************************************
15008 REM
15009 REM
15010 CLS:LOCQTE 1,20:COLOR 2,0
15020 PRINT "This sets the";
15030 COLOR 18,0:PRINT " C—V ";
15040 COLOR 2,0:PRINT " Experiment Parameters";
15050 Parameter!(2,1) = Parameter!(25,1)
15060 Parameter!(2,2) = Parameter!(25,2)
15070 Parameter!(2,3) = Parameter!(25,3)
15080 Parameter!(2,4) = Parameter!(25,4)
15090 IF Expt%(26) = 3 OR Expt%(26) = 1 THEN GOTO 15160
15100 REM
15101 REM ———Sets the Temperature Parameters———
15102 REM
15110 Expt$ = "C-V Experiment"
15120 Col% = 20
15130 Name$ = "Settings"
15140 Settempparam (Time%,Col%,Parameter!(2,1),Parameter!(2,2),Parameter!(2,3),Expt$,Name$)
15150 REM
15151 REM ————Sets the Start, Stop and Delta Voltages———
15152 REM
15160 DVdt (Parameter!(2,12),Parameter!(2,7),Parameter!(2,8),Parameter!(2,9))
15170 REM
15171 REM ————Sets the Hold Time ————
15172 REM
15180 Settime (Parameter!(2,13),Parameter!(2,14),Parameter!(2,15),Parameter!(2,16),0,1,0,3)
15190 REM
15191 REM **********************************************************************************************
15192 REM *
15193 REM * This checks the parameters set with the I-V meter. *
15194 REM *
15195 REM **********************************************************************************************
DATA STRING$ = "F3I2L3PS" + STR$(Parameter!(2,7)-Parameter!(2,9)) + PT + STR$(Parameter!(2,8)+Parameter!(2,9)) + PE + STR$(Parameter!(2,9)) + PV + STR$(Parameter!(2,12)) + PH + STR$(Parameter!(2,14)) + W1

PARAM$ = "WR.STR/5//EOS/";GOSUB 50000

DATA STRING$ = "W7"

PARAM$ = "WR.STR/5//EOS/";GOSUB 50000

PQRQMS = "SER.POLL/5/";GOSUB 50000

IF POLL_RESP% AND &H40 = 64 THEN GOSUB 15500

PARAM$ = "SDC/5/";GOSUB 50000

RETURN

REM

REMARK ************************************************

REMARK *************

REMARK * This prints the error message sent in binary by the I-V meter. It then sends the program back to reenter the parameters.

REMARK ************************************************

Clearscreen (3)

COLOR 7,0:LOCATE 10,5

PRINT "The settings chosen are not viable. You must select new settings."

LOCATE 11,5

PRINT "If you are not sure why this occurred please check the HP 4140B."

LOCATE 12,5:PRINT "The BINARY error code is ";

COLOR 5,0:PRINT MID$(BIN$(POLL_RESP%),8)

Timedelay(6)

PARAM$ = "SDC/5/";GOSUB 50000

RETURN,15150

STOP

REMARK ************************************************

REMARK *************

REMARK * This subroutine places the data pointer to the right experimental group, for the print out.

REMARK *************
1 81

40006 REM ***********************************************************
40007 REM
40008 ON Expt%(25) GOTO 40020,40030,40040,40045
40010 RESTORE,60510:GOTO 40050
40020 RESTORE,60520:GOTO 40050
40030 RESTORE,60530:GOTO 40050
40040 RESTORE,60540:GOTO 40050
40045 RESTORE,60550:GOTO 40050
40050 RETURN
40999 STOP
41000 REM
41001 REM ***********************************************************
41002 REM *
41003 REM * This subroutine places the data pointer to the right *
41004 REM * experimental group, for the experiment menu. *
41005 REM *
41006 REM ***********************************************************
41007 REM
41010 ON Expt%(25) GOTO 41020,41030,41040,41050
41020 RESTORE,60220:GOTO 41060
41030 RESTORE,60230:GOTO 41060
41040 RESTORE,60240:GOTO 41060
41050 RESTORE,60250:GOTO 41060
41060 RETURN
49988 REM IEEE—488 INTERFACE FOR THE IBM PC V4_2
49989 REM WRITTEN IN ADVANCED BASIC
49990 REM AND INCORPORATING ASSEMBLY LANGUAGE ROUTINES TO IMPLEMENT
49991 REM DMA—DRIVEN GPIB TRANSACTIONS
49992 REM THE ASSEMBLY LANGUAGE ROUTINES MUST BE LOADED PRIOR TO ENTERING
49993 REM BASICA BY TYPING "SUBLIB", THEN TYPE "BASICA", LOAD IEEE488_BAS,
49994 REM AND CALL SUBROUTINES AS DESCRIBED IN THE MANUAL
49995 REM
49997 REM
49998 REM ******************************************************* START OF SUBROUTINE *******
49999 STOP
60000 REM
60001 REM *******************************************************
* 60003 REM * These DATA statements are used for the information * 60004 REM * block print statements and correction routine. * 60005 REM * 60006 REM ****************************************************************************** 60007 REM 60010 DATA 6 60020 DATA "Directory Number","Date ","Experimenters Name", "Sample Number","Cryostat Chronometer Reading","Comments", "Type of Run" 60099 STOP 60100 REM 60101 REM ****************************************************************************** 60102 REM * 60103 REM * These DATA statements allow the user to choose which * 60104 REM * group of experiments to run. * 60105 REM * 60106 REM ****************************************************************************** 60110 DATA 4 60120 DATA "GROUP I C-t, G-t, C-G-V, C-G" 60130 DATA "GROUP II I-V, C-V" 60140 DATA "GROUP III Van der Pauw/Mobility" 60150 DATA "GROUP IV 4-point Resistivity" 60199 STOP 60200 REM 60201 REM ****************************************************************************** 60202 REM * 60203 REM * These DATA statements allow the user to choose which * 60204 REM * experiments he wants to run from the group chosen * 60205 REM * earlier. * 60206 REM * 60207 REM ****************************************************************************** 60208 REM 60210 DATA 4,4,2,1,1
DATA "Capacitance (C) vs. Time","Conductance (G) vs. Time","C and G vs. Voltage","Capacitance and Conductance"
DATA "Current vs. Voltage","Capacitance vs. Voltage"
DATA "Van der Pauw/Mobility"
DATA "4-point Resistivity"
STOP
REM
DATA 24
DATA "Initial Temp","Final Temp","Delta Temp",Current Temp"
DATA "Final Time","Delta Time"
DATA "Start Volt","Stop1 Volt","Step1 Volt","Stop2 Volt","Step2 Volt"
DATA "DV/dt"
DATA "Delay Time","Initial Hold","Final Hold","Delta Hold"
DATA "Initial Bias","Final Bias","Delta Bias"
DATA "Number of Samples"
DATA "Start High Pulse","Final High Pulse","Delta High Pulse","Low Pulse"
STOP
REM
DATA 3
DATA "Time Run ( ) 5 Minutes; 30-600 K)","Temperature Run (30 — 600 K)","Room Temperature Run (290 K)"

DATA statements are for the PARAMETER (x,y) array.
DATA statements are used for the type of temperature run section.
STOP
REM
*/
REM * These data statements are for the print out of the parameter set.
REM *
REM **********************************************
DATA "C-t", "G-t", "C-G-V", "C-G"
DATA "I-V", "C-V"
DATA "Van der Pauw"
DATA "4-point Resistivity"
STOP
REM
REM **********************************************
DATA "TTEMP", "LTEMP", "RTEMP"
ENDFILE
STRING: X$[?]  
REAL: TEMP_SET!, TEMP_PEAK!, Temp1!, Temp2!, TIFF!  
REAL: Temp3!  
INTEGER: Print_temp%, ROW%, P%  
REAL: ACTUALTEMP, N

PROCEDURE: TIMEDELAY()  
    REAL ARG: DELAY!/OPT=5!  
END PROCEDURE

PROCEDURE: CLEARSCREEN()  
    INTEGER ARG: ROW%  
END PROCEDURE

PROCEDURE: TIMEDELAY  
    REAL: Tyme!  
    STRING: OVERHEAT$[?], Data_string$[?]  
    REAL: TEMP_PEAK!, PEAK_TEMP!  
    22 REM * integer arg: delay%/opt = 5  
    32 REM * real: tyme!  
    100 Tyme! = TIMER  
    110 IF TIMER < Tyme! + DELAY! THEN GOTO 110  
    120 EXIT  
END PROCEDURE

PROCEDURE: CLEARSCREEN  
    INTEGER: N%  
    100 FOR N% = ROW% TO 24  
    110 SET CURSOR NX, 0  
    120 PRINT SPC(80)  
    130 NEXT N%  
END PROCEDURE

' MAIN Program:

200 REM  
201 REM ****************************#################################  
202 REM *  
203 REM * This section prints the entrance message to  
204 REM * RUNIT. It  
205 REM * also initializes the devices on the IEEE bus  
206 REM  
207 REM
210 CLEAR
220 ON ERROR GOSUB 42000
230 CLS : STATUSLINE OFF : SCREEN 0,0,0 : COLOR 2,0 : LOCATE 12,23
240 PRINT "Hit any key to begin Experiments"
250 COLOR 3,0 : LOCATE 14,10
260 PRINT "All instruments must be on and the proper connections be made"
270 TIME DelAY (2)
280 A$ = INKEY$: IF A$ = "" THEN GOTO 250
290 PARAM$ = "INIT/1/&H310/P/" : GOSUB 50000
300 PARAM$ = "SDR/5,7,8,12,13,16,17/" : GOSUB 50000
310 REM
311 REM **************************************************
312 REM *
313 REM * This section loads the data from the files stored by *
314 REM * the program PARAMETER. *
315 REM *
316 REM **************************************************
317 REM
320 CLS : COLOR 2,0,0 : LOCATE 12,30
330 PRINT "Now LOADING data"
340 DRIVE$ = "C:"
350 DIR$ = "\"
360 OPEN " \DATA\EXPT" FOR INPUT AS #1
370 INPUT #1,File$
380 FOR MX = 1 TO 30
390 INPUT #1,ExptX(MX)
400 NEXT MX
410 CLOSE #1
420 OPEN " \DATA\PARAM" FOR INPUT AS #1
430 FOR MX = 1 TO 6
440 INPUT #1,Parameter!(25,MX)
450 NEXT MX
460 RESTORE,60010
470 READ MaxloopX
480 FOR MX = 1 TO MaxloopX
490 READ GroupX(MX) < (25, M)
500 NEXT MX
505 RESTORE,60410
510 READ MaxloopX
510 FOR N% = 1 TO GroupX(ExptX(25))
520 FOR MX = 1 TO MaxloopX
530 IF ExptX(N%) = 0 THEN EXIT 1 LEVELS
540 INPUT #1,Parameter!(N%,MX)
550 NEXT M\% 560 NEXT N\% 570 CLOSE #1 575 ACTUALTEMP=Parameter!(25,4) 600 REM 601 REM "******************************************************************************** 602 REM * 603 REM * This section finds which type of temperature run was * 604 REM * requested, then stores in in the string XTEM P\$ * 605 REM * 606 REM "******************************************************************************** 607 REM 610 RESTORE,60110 620 FOR MX = 1 TO 30 630 REQD Xtemp$ 640 IF MX = ExptX(26) THEN EXIT 650 NEXT MX 700 REM 704 PX = 0 2 Print_tempX = 1 705 RESTORE,62000 710 DO 720 IF ExptX(25) () 3 THEN EXIT TO,1000 730 CLS 2 COLOR 4,0,0 2 SET CURSOR 0,35 740 PRINT "MOBILITY EXPERIMENT" 750 COLOR 3,0,0 2 SET CURSOR 7,6 2 PRINT "Connect the cables as follows" 760 COLOR 2,0,0 770 FOR N\% = 1 TO 5 780 READ Name$ 800 SET CURSOR 7 + N\%*2,10 810 PRINT Name$ 820 NEXT N\% 830 COLOR 7,0,0 : SET CURSOR 22,6 840 PRINT "Hit any key to continue" 850 A$ = INKEY$: IF A$ = "" THEN GOTO 850 860 P\% = P\% + 1 870 GOSUB 15000 871 IF P\% < 3 THEN GOTO 710 872 CLEARSCREEN(2) 873 COLOR 2,0,0 : SET CURSOR 13,6 874 PRINT "Hit any key to begin the experiment run" 875 A$ = INKEY$: IF A$ = "" THEN GOTO 875 880 Print_temp% = 2:GOTO 1000 1000 REM
1001 REM #################################################################################
1002 REM *
1003 REM * This section print up the proper screen for each type of temperature run. Prints the time if a time run, and goes to the proper subroutine for the experiments. It also exits to finish routines if either the final temperature or final time has been reached. *
1008 REM *
1009 REM #################################################################################
1010 REM
1020 Mintemp! = Parameter!(25,1)
1030 Maxtemp! = Parameter!(25,2)
1040 COLOR 5,0:CLS
1050 IF Expt%(26) = 3 THEN PRINT SPC(60);"Room Temperature"
1060 COLOR 4,0:IF Expt%(26) = 3 THEN PRINT SPC(63);Parameter!(25,4);"K"
1070 COLOR 3,0:LOCATE 12,28:PRINT "The Current Program is:
1080 IF Expt%(26) = 1 THEN GOTO 1100
1090 GOSUB 9000
1100 IF Expt%(26) <> 1 THEN GOTO 1150
1110 Timerun! = 0.0
1120 TIME$ = "00:00:00"
1130 LOCATE 1,8:COLOR 5,0:PRINT "Time"
1140 LOCATE 2,6:COLOR 4,0:PRINT TIME$
1150 COLOR 2,0
1160 ON Expt%(25) GOSUB 2000,2500,3010,4010
1170 IF Expt%(26) = 1 THEN GOTO 1220
1180 Parameter!(25,4) = Parameter!(25,4) + Parameter!(25,3)
1190 IF Parameter!(25,4) > Maxtemp! THEN GOTO 40000
1210 GOTO 1040
1220 IF Timer >= Parameter!(25,5) THEN GOTO 40000
1230 Timerun! = Timerun! + Parameter!(25,6)
1240 TIMEDELY (.5)
1250 LOCATE 14,35:PRINT "TIME SET"
1260 LOCATE 2,6:COLOR 4,0:PRINT TIME$
1270 IF Timer <= Timerun! THEN GOTO 1250
1280 GOTO 1150
1999 STOP
2000 REM
This subroutine branches to the proper experiments chosen in group I. It also checks after each experiment to see if the COLD HEAD has overheated (except in the case of a Room Temperature run).

```
2001 REM ************************************************
2002 REM *
2003 REM * This subroutine branches to the proper experiments chosen in group I. It also checks after each experiment to see if the COLD HEAD has overheated (except in the case of a Room Temperature run).
2004 REM *
2005 REM *
2006 REM *
2007 REM *
2008 REM ************************************************
2009 REM
2010 LX = 1
2020 ON ExptX(LX) GOSUB 10000,10000,11000,12000
2030 LOCATE 14,1:PRINT SPC(79)
2040 LX = LX + 1
2050 IF ExptX(26) <> 3 THEN GOSUB 9500
2060 IF LX <= GroupX(ExptX(25)) THEN GOTO 2020
2070 RETURN
2499 STOP
2500 REM
2501 REM ************************************************
2502 REM *
2503 REM * This subroutine branches to the proper experiments chosen in group II. It also checks after each experiment to see if the COLD HEAD has overheated (except in the case of a Room Temperature run).
2504 REM *
2505 REM *
2506 REM *
2507 REM *
2508 REM ************************************************
2509 REM
2510 LX = 1
2520 ON ExptX(LX) GOSUB 13000,14000
2530 LOCATE 14,1:PRINT SPC(79)
2540 IF ExptX(26) <> 3 THEN GOSUB 9500
2550 LX = LX + 1
2560 IF LX <= GroupX(ExptX(25)) THEN GOTO 2520
2570 RETURN
3000 REM
```
3001 REM *****************************
3002 REM *
3003 REM * This subroutine branches to the proper experiments *
3004 REM * chosen in group III. It also checks after each *
3005 REM * experiment to see if the COLD HEAD has overheated *
3006 REM * (except in the case of a Room Temperature run). *
3007 REM *
3008 REM *****************************
3009 REM
3010 GOSUB 15000
3020 IF Expt%(26) <> 3 THEN GOSUB 9500
3030 RETURN
4000 REM
4001 REM *****************************
4002 REM *
4003 REM * This subroutine branches to the proper experiments *
4004 REM * chosen in group IV. It also checks after each *
4005 REM * experiment to see if the COLD HEAD has overheated *
4006 REM * (except in the case of a Room Temperature run). *
4007 REM *
4008 REM *****************************
4009 REM
4010 GOSUB 16000
4020 IF Expt%(26) <> 3 THEN GOSUB 9500
4030 RETURN
8999 STOP
9000 REM
9001 REM *****************************
9002 REM *
9003 REM * This sets the Cryostat to the correct temperature and *
9004 REM * then lets the temperature to stabilize to +/-
9005 -1 degree *
9005 REM *
9007 REM *****************************************************************************
9008 REM
9010 COLOR 2, 0: LOCATE 14, 24
9020 PRINT "Setting the Cryostat Temperature"
9030 LOCATE 1, 30: PRINT SPC(49)
9040 COLOR 5, 0: LOCATE 1, 60: PRINT "Sample Setpoint"
9050 COLOR 4, 0: LOCATE 2, 63: PRINT Parameter(25, 4); "K"
9060 Test_Passed% = 0: Cryostat% = 0
9065 IF Parameter!(25, 2) > 290 THEN Offset! = 0 ELSE Offset! = 0
9070 Temp_wanted! = Parameter!(25, 4) + Offset!
9075 IF Temp_wanted! => 15 THEN Heater$ = "7" ELSE Heater$ = "8" ELSE Heater$ = "9"
9080 IF Temp_wanted! <= 15 AND Temp_wanted! < 20 THEN Heater$ = "7"
9090 IF Temp_wanted! > 290 THEN Int_Gain$ = "00100000400" ELSE Int_Gain$ = "00100000500"
9100 IF Temp_wanted! > 100 THEN Int_Gain$ = "00150000600" ELSE Int_Gain$ = "00200000700"
9110 Delta_temp! = 1
9120 Sensor$ = "33"
9130 DATA_STRING$ = "P"
9140 PARAM$ = "WR.STR/7//EOS/"; GOSUB 50000
9150 Temp_wanted$ = MID$(STR$(Temp_wanted! + 1000), 3)
9160 DATA_STRING$ = Temp_wanted$ + Int_Gain$ + Sensor$ + Heater$ + "505"
9170 PARAM$ = "WR.STR/7//EOS/"; GOSUB 50000
9180 PARAM$ = "RD.STR/7//EOS/"; GOSUB 50000
9190 PARAM$ = "RD.STR/7//EOS/"; GOSUB 50000
9200 TIMEDELAY(2)
9210 DO 2 TIMES
9220 PARAM$ = "RD.STR/7//EOS/"; GOSUB 50000
9230 Temp! = VAL(MID$(DATA_STRING$, 2))
9240 IF Temp! = Temp_wanted! + Delta_temp! THEN EXIT TO, 9300
9250 TIMEDELAY(1)
9260 REPEAT
9270 Cryostat% = 0
9275 Test_Passed% = 0
9280 IF Temp!( Temp_wanted! - 2 OR Temp!) Temp_wanted! + 2 THEN GOSUB 5000
9290 GOTO 9130
9299 STOP
9300 Test_Passed% = Test_Passed% + 1
9310 IF Test_Passed% > 1 THEN GOTO 9400
9320 TIMEDELAY(35)
1930 GOTO 9130
9400 LOCATE 1,30:PRINT SPC(79)
9410 COLOR 5,0:LOCATE 1,59:PRINT "Sample Temperature"
9415 COLOR 4,0:LOCATE 2,65:PRINT Parameter!(25,4)
9417 ACTUALTEMP=Temp!
9420 COLOR 2,0
9430 Cryostat% = 1
9440 RETURN
9499 STOP
9500 REM
9501 REM **************************************************+
9502 REM *
9503 REM * This subroutine checks the Cold Head to make
9504 REM * sure its *
9505 REM *
9506 REM **************************************************+
9507 REM
9510 IF Cryostat% = 0 THEN GOTO 9540
9520 LOCATE 14,1:PRINT SPC(79)
9530 COLOR 2,0:LOCATE 14,30:PRINT "Checking Cold Finger"
9540 DATA_STRING$ = "P"
9550 PARAM$ = "WR STR/7//EOS/":GOSUB 50000
9555 Sensor1$ = "31"
9560 DATA_STRING$ = Temp_wanted$ + Int_Gain$ + Sensor1$ +
9570 PARAM$ = "WR STR/7//EOS/":GOSUB 50000
9580 DATA_STRING$ = "R"
9590 PARAM$ = "WR STR/7//EOS/":GOSUB 50000
9600 DO 5 TIMES
9610 PARAM$ = "RD STR/7//EOS/":GOSUB 50000
9620 Temp! = VAL(MID$(DATA_STRING$,2))
9630 IF Temp! > 325 AND Temp! < 900 THEN GOTO 9700
9640 REPEAT
9650 IF Cryostat% = 0 THEN GOTO 9670
9660 LOCATE 14,1:PRINT SPC(79):LOCATE 14,1
9670 RETURN
9699 STOP
9700 REM
9701 REM **************************************************+
9702 REM *
9703 REM * This subroutine is used if the Cold Finger o
9704 REM * verheats. *
It writes death parameters to the Cryostat and gives a warning message.

*********************************************************************************

**DGTG_STRING$ = "P"

**PGRGMS = "WR.STR/7//EOS/";GOSUB 50000
**DGTG_STRING$ = "O021000000001117501"
**PGRGMS = "NR.STR/7//EOS/";GOSUB 50000
**DGTG_STRING$ = "R"

**PGRGMG = "HR.STR/7//EOS/";GOSUB 50000

**PGRGMS = "RD.STR/7//EOS/";GOSUB 50000

**CLS:COLOR 2,0;LOCATE 11,21

**PRINT "The ";

**COLOR 19,0:PRINT "COLD FINGER";

**COLOR 2,0

**PRINT " has ";

**COLOR 20,0:PRINT "Overheated"

**LOCATE 14,16:COLOR 7,0

**PRINT "PLEASE CHECK FOR DAMAGE AND CONTACT ERIC COLE"

**LOCATE 16,12:COLOR 1,0:PRINT "ALSO TURN THE HEATER SWITCH ABOVE THE CONTROLLER OFF"

**CLEAR

**GOTO 9880

**STOP

**REM

**This subroutine run the experiment [either C-t (1) or G-t (2)] specified by 1%. It stores the data in the file C:\DGTG\##\?-t where ? is C or G respectively.

**file C:\DATA\##\?-t where ? is C or G respectively.

**Function$ = "FN6"

**Name$ = "\G-t"
10060 Moniker$ = "CONDUCTANCE"
10070 GOTO 10110
10080 Function$ = "FN5"
10090 Name$ = "\C-t"
10100 Moniker$ = "CAPACITANCE"
10110 OPEN "C:\DATA\" + File$ + Name$ FOR APPEND AS #1
10120 LOCATE 14,20;PRINT "Taking ";Moniker$;" vs. TIME Measurements"
10130 Pulsevolt! = Parameter!(LX,21)
10140 DATA_STRING$ = "M4CTOT1T1W1HIL" + STR$(Pulsevolt!) + "VLOL" + STR$(Parameter!(LX,24)) + "VDO"
10150 PARAM$ = "WR.STR/13//EOS/";GOSUB 50000
10160 PARAM$ = "RD.STR/13//EOS/";GOSUB 50000
10170 Bias! = Parameter!(L%,17)
10180 Time! = Parameter!(L%,14)
10190 IF Time! <= 0.01 THEN GOSUB 10580:6OTO 10220
10200 IF Time! < 10.1 * Parameter!(LX,13) GND Time! < .1 THEN GOSUB 10580:6OTO 10220
10210 GOSUB 10650
10220 NX = 1
10230 PRINT #1,Xtemp$;Parameter!(25,4);"ACTUALTEMP"
10240 PRINT #1,"TIME = ";Timerun!;" ;"" ;"TIMER;" ;"
10250 PRINT #1,"BIAS = ";Bias!
10260 PRINT #1,"HIGH PULSE = ";Pulsevolt!
10270 PRINT #1,"HOLD TYME = ";Time!
10280 PARAM$ = "RD.STR/17//EOS/";GOSUB 50000
10290 NCMX = INSTR(DATA_STRING$,"M") + 1
10300 Cdata! = VAL(MID$(DATA_STRING$,NCMX))
10310 TX = INSTR(DATA_STRING$,"T") + 1
10320 Xdata! = VAL(MID$(DATA_STRING$,TX))
10330 PRINT #1,Cdata!;",";Xdata!
10340 NX = NX + 1
10350 IF NX <= Parameter!(LX,20) THEN GOTO 10280
10360 PRINT #1,"END"
10370 IF Parameter!(LX,15) = 0 OR Parameter!(LX,16) = 0 THEN GOTO 10400
10380 Time! = Time! + Parameter!(LX,16)
10390 IF Time! <= Parameter!(LX,15) THEN GOTO 10190
10400 IF Parameter!(LX,18) = 0 OR Parameter!(LX,19) = 0 THEN GOTO 10430
10410 Bias! = Bias! + Parameter!(LX,19)
10420 IF Bias! <= Parameter!(LX,18) THEN GOTO 10180
10430 IF Parameter!(LX,23) = 0 THEN GOTO 10460
10440 Pulsevolt! = Pulsevolt! + Parameter!(LX,23)
10450 IF Pulsevolt! <= Parameter!(LX,22) THEN GOTO 10140
10460 Parameter!(LX,4) = Parameter!(LX,4) + Parameter!(LX,3)
10470 CLOSE #1
10480 DATA_STRING$ = "BL0"
10490 PARAM$ = "WR.STR/17//EOS/";GOSUB 50000
10500 REM
10540 IF Parameter!(L%,4) < Parameter!(25,4) THEN Parameter!(L%,4) = Parameter!(L%,3) + Parameter!(L%,4)
10550 IF Parameter!(L%,4) > Parameter!(L%,2) THEN Parameter!(L%,4) = 600
10560 PARAM$ = "SDC/13,17/";GOSUB 50000
10570 RETURN
10579 STOP
10580 REM
10581 REM ****************************
10582 REM *
10583 REM * This subroutine is used if the step delay time and *
10584 REM * hold times does require the meter to be put into *
10585 REM * block mode.
10586 REM *
10587 REM ****************************
10588 REM
10589 DATA_STRING$ = Function$ + "CN13TR3LE2SA1PC" + STR$(Bias!) + ";PN" + STR$(Parameter!(L%,20)) + ";PH" + STR$(Time!) + ";PT" + STR$(Parameter!(L%,13)) + ";SW1"
10600 PARAM$ = "WR.STR/17//EOS/";GOSUB 50000
10610 TIMEDELAY(10)
10620 DATA_STRING$ = "BD"
10630 PARAM$ = "WR.STR/17//EOS/";GOSUB 50000
10640 RETURN
10649 STOP
10650 REM
10651 REM ****************************
10652 REM *
10653 REM * This subroutine is used if the step delay time and *
10654 REM * hold times do not require the meter to be put into *
10655 REM * block mode.
10656 REM *
10657 REM ****************************
10658 REM
10659 DATA_STRING$ = Function$ + "CN13TR3LE2SA1PC" + STR$(Bias!) + ";PN" + STR$(Parameter!(L%,20)) + ";PH" + STR$(Time!) + ";PT" + STR$(Parameter!(L%,13)) + ";SW1"
PARAM$ = "WR.STR/17//EOS/"; GOSUB 50000
RETURN
STOP
REM
1001 REM ****************************
*************
1002 REM *
1003 REM  This subroutine runs the C-G-V experiment and stores the data in the file "\DATA\##\CGV".
*
1005 REM *
1006 REM ****************************
1007 REM
1010 IF Expt$(26) = 1 OR Expt$(26) = 3 THEN GOTO 11030
1020 IF Parameter!(3,4) <> Parameter!(25,4) THEN GOTO 11310
1030 OPEN "C:\DATA\" + File$ + "CGV" FOR APPEND AS #1
1040 PRINT #1, Xtemp$; Parameter!(25,4)";QCTUQLTEMP
1050 PRINT #1, "TIME = "; Timerun!; " ("; TIMER; ")"
1060 LOCATE 14,10; PRINT "taking CPPQCITPNCE QND CONDUCTANCE measurements"
1070 N% = 6
1080 M% = 6
1090 N% = N% + 1
1100 M% = M% + 2
1110 IF Parameter!(3, M%+1) = 0 THEN GOTO 11270
1120 DATA_STRING$ = "FN1CN10IB2LE2TR3PS" + STR$(Parameter!(3,N%)) + "; PP" + STR$(Parameter!(3, M%)) + "; PE" + STR$(Parameter!(3, M%+1)) + "; PL" + STR$(Parameter!(3, 14)) + "; PD" + STR$(Parameter!(3, 13)) + "; SW1"
1130 PARAM$ = "WR.STR/17//EOS/"; GOSUB 50000
1140 L! = Parameter!(3, N%)
1150 PARAM$ = "RD.STR/17//EOS/"; GOSUB 50000
1160 NCM% = INSTR(DATA_STRING$, "M") + 1
1170 Cdata! = VAL(MID$(DATA_STRING$, NCM%))
1180 NGM% = INSTR(DATA_STRING$, "GM") + 2
1190 Gdata! = VAL(MID$(DATA_STRING$, NGM%))
1200 V% = INSTR(DATA_STRING$, "V") + 1
1210 Xdata! = VAL(MID$(DATA_STRING$, V%))
1220 PRINT #1, Cdata!;", "; Gdata!; "; Xdata!
1240 IF L! < Parameter!(3, M%) THEN L! = L! + Parameter!(3, M%+1) GOTO 11150
1250 DATA_STRING$ = "SWO"
1260 PARAM$ = "WR.STR/17//EOS/"; GOSUB 50000
1270 IF N% = 7 THEN GOTO 11090
1280 Parameter!(3,4) = Parameter!(3,4) + Parameter!(3,3)
1290 PRINT #1, "END"
1300 CLOSE #1
11310 REM
11350 IF Parameter!(3,4) < Parameter!(25,4) THEN Parameter!(3,4) = Parameter!(3,4) + Parameter!(3,3)
11360 IF Parameter!(3,4) > Parameter!(3,2) THEN Parameter!(3,4) = 600
11370 PARAM$ = "SDC/17":GOSUB 50000
11380 RETURN
11999 STOP
12000 REM
12001 REM ************************************************************
12002 REM *
12003 REM * This section takes the C-G readings and puts them in  *
12004 REM * file \DATA\##\CG"
12005 REM *
12006 REM *************************************************************
12007 REM
12010 IF Expt%(26) = 1 OR Expt%(26) = 3 THEN GOTO 12030
12020 IF Parameter!(4,4) < Parameter!(25,4) THEN GOTO 12250
12030 OPEN "C:\DATA\" + File$ + "\CG" FOR APPEND AS #1
12040 Bias! = Parameter!(4,17)
12050 LOCATE 14,15:PRINT "taking CAPACITANCE AND CONDUCTANCE measurements"
12060 DATA_STRING$ = "FN1CN10IB1RA1MS3SL2LE2TR3PV" + STR$(Bias!)
12070 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
12072 DATA_STRING$ = "V01"
12074 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
12080 PRINT #1,Xtemp$;Parameter!(25,4);"ACTUALTEMP
12090 PRINT #1,"TIME = ";Timerun!;";"TIME;");""
12100 PRINT #1,"BIAS = ";Bias!
12105 TIMEDELAY(5)
12110 DATA_STRING$ = "EX"
12120 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
12130 PARAM$ = "RD.STR/17//EOS/":GOSUB 50000
12140 NCM% = INSTR(DATA_STRING$,"M") + 1
12150 Cdata! = VAL(MID$(DATA_STRING$,NCM%))
12160 NGMV% = INSTR(DATA_STRING$,"GM") + 2
12170 Gdata! = VAL(MID$(DATA_STRING$,NGMV%))
12180 PRINT #1,Cdata!,"Gdata!","ACTUALTEMP
12185 DATA_STRING$ = "V00"
12187 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
12190 IF Parameter!(4,19) = 0 THEN GOTO 12220
12200 Bias! = Bias! + Parameter!(4,19)
12210 IF Bias! (= Parameter!(4,19) THEN GOTO 12060
12220 Parameter!(4,4) = Parameter!(4,4) + Parameter!(4,3)
12225 TIMEDELAY(10)
12230 CLOSE #1
12240 PARAM$ = "SDC/17/":GOSUB 50000
12250 REM
12290 IF Parameter!(4,4) ( Parameter!(25,4) THEN Parameter!(4,4) = Parameter!(4,4) + Parameter!(4,3)
12300 IF Parameter!(4,4) > Parameter!(4,2) THEN Parameter!(4,4) = 600
12310 PARAM$ = "SDC/17/":GOSUB 50000
12320 RETURN
12999 STOP
13000 REM
13001 REM
13002 REM *
13003 REM * This is the subroutine for actually running the I-V *
13004 REM * experiment. It stores the data in the file *
13005 REM * \DATA\#\IV *
13006 REM *
13007 REM ************************************************
13008 REM
13010 IF Expt%(26) = 1 OR Expt%(26) = 3 THEN GOTO 13030
13020 IF Parameter!(1,4) (> Parameter!(25,4) THEN GOTO 13290
13030 OPEN "C:\DQTR" + File$ + "IV" FOR PPPEND QS #1
13040 LOCQTE 14,20:PRINT "taking CURRENT vs. VOLTAGE measurements"
13050 PRINT #1,Xtemp$;Parameter!(25,4);"ACTUALTEMP
13060 PRINT #1,"TIME = "; timerun!;"("; "")"
13070 M% = 7
13080 N% = 8
13085 X$ = STR$(VAL(STR$(Parameter!(1,N% + 1))))
13090 DATA_STRING$ = "F2I2L3PS" + STR$(Parameter!(1,M%)) + ";PT" + STR$(Parameter!(1,N%)) + ";PE" + X$ + ";PH" + STR$(Parameter!(1,14)) + ";PD" + STR$(Parameter!(1,13)) + ";W"
13100 TIMEDELAY(1)
13110 PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
13120 BN = 0
13130 PARAM$ = "RD.STR/5//EOS/":GOSUB 50000
13140 NCM% = INSTR(DATA_STRING$,"I") + 1
13150 Idata! = VAL(MID$(DATA_STRING$,NCM%))
13160 V% = INSTR(DATA_STRING$,"A") + 1
13170 Xdata! = VAL(MID$(DATA_STRING$,V%))
13180 PRINT #1, Idata!","Xdata!
13190 B% = INSTR(DATA_STRING$, "L")
13200 IF B% = 0 THEN GOTO 13130
13210 TIMEDELAY(2)
13215 PARAM$ = "RD.STR/5//EOS/":GOSUB 50000
13220 IF Parameter!(1,11) = 0 THEN GOTO 13260
13230 M% = M% + 1
13240 N% = N% + 2
13250 IF M% = 8 THEN GOTO 13085
13260 Parameter!(1,4) = Parameter!(1,4) + Parameter!(1,3)
13270 PRINT #1,"END"
13280 CLOSE #1
13290 REM
13300 IF Parameter!(1,4) < Parameter!(25,4) THEN Parameter!(1,4) = Parameter!(1,4) + Parameter!(1,3)
13340 IF Parameter!(1,4) > Parameter!(1,2) THEN Parameter!(1,4) = 600
13350 PQRRMS = "SDC/5/":GOSUB 50000
13360 RETURN
13399 STOP
14000 REM
14001 REM ************************************************
14002 REM *************
14003 REM ** This actually runs the Capacitance vs. Voltage experiment.**
14004 REM * experiment. *
14005 REM *
14006 REM ************************************************
14007 REM
14010 IF Expt%(26) = 1 OR Expt%(26) = 3 THEN GOTO 14030
14020 IF Parameter!(2,4) <> Parameter!(25,4) THEN GOTO 14040
14030 OPEN "C:\DATA\" + File$ + "\CV" FOR APPEND AS #1
14040 LOCATE 14,18:PRINT "taking CAPACITANCE vs. VOLTAGE measurements"
14050 B% = 0
14060 PRINT #1,Xtemp$;Parameter!(25,4);"\"ACTUALTEMP
14070 PRINT #1,"TIME = ";Timerun!;" (\";TIMER;\")
14080 DATA_STRING$ = "F3I2L3PS" + STR$(Parameter!(2,7) + Parameter!(2,9)) + ";PT" + STR$(Parameter!(2,8) + Parameter!(2,9)) + ";PE" + STR$(Parameter!(2,9) + Parameter!(2,12)) + ";PV" + STR$(Parameter!(2,14) + ";W1"
14090 PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
14100 PARAM$ = "RD.STR/5//EOS/":GOSUB 50000
14110 NCM% = INSTR(DATA_STRING$,"C") + 1
14120 Cdata! = VAL(MID$(DATA_STRING$,NCM%))
14130 VX = INSTR(DATA_STRING$, "A") + 1
14140 Xdata! = VAL(MID$(DATA_STRING$, VX))
14150 PRINT #1, Cdata!", Xdata!
14160 BX = INSTR(DATA_STRING$, "L")
14170 IF BX = 0 THEN GOTO 14100
14180 Parameter!(2, 4) = Parameter!(2, 4) + Parameter!(2, 3)
14190 PRINT #1, "END"
14200 CLOSE #1
14210 REM
14250 IF Parameter!(2, 4) ( Parameter!(25, 4) THEN Parameter!
14260 IF Parameter!(2, 4) > Parameter!(2, 2) THEN Parameter!
14270 PARAM$ = "SDC/5/": GOSUB 50000
14280 RETURN
14999 STOP
15000 REM
15001 REM ************************************************
15002 REM
15003 REM * This subroutine runs the Van der Pauw/Mobility
15004 REM * experiment and stores it in the file MOB.
15005 REM *
15006 REM ************************************************
15007 REM
15009 IF Print_tempX = 1 THEN CLEQRSCREEN(1)
15010 COLOR 2, 0; LOCATE 14, 20
15020 PRINT "taking Van der Pauw/Mobility Measurements"
15030 OPEN "\DATA\" + File$ + "\MOB" FOR APPEND AS #1
15032 DO
15035 IF Print_tempX < 2 THEN EXIT 1 LEVELS
15040 PRINT #1, Xtemp$; Parameter!(25, 4)"; "ACTUALTEMP
15050 PRINT #1, "TIME = "; Timerun!; ("; TIME; ")"
15055 END DO
15060 DATA_STRING$ = "FN1CN10IB2TR3PS—10;PP10;PE.01;PL2;PD
15070 PARAM$ = "WR.STR/17//EOS/": GOSUB 50000
15078 DATA_STRING$ = "L3;"
15079 PARAM$ = "WR.STR/5//EOS/": GOSUB 50000
15089 DATA_STRING$ = "RO" + CHR$(13) + "X"
15090 PARAM$ = "WR.STR/16//EOS/": GOSUB 50000
15092 TIMEDELAY(15)
15095 DATA_STRING$ = "Z1" + CHR$(13) + "X"
15096 PARAM$ = "WR.STR/6//EOS/": GOSUB 50000
15100 FOR M% = 1 TO Print_tempX
15110 DATA_STRING$ = "F2RA1I2J0A3B2L3PS0;PT10;PE0.01;PH1
201

;PD.1;W2
15120   PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
15130   DO 1000 TIMES
15140   DATA_STRING$ = "W6"
15150   PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
15160   TIMEDELAY (2)
15170   PARAM$ = "RD.STR/5//EOS/":GOSUB 50000
15180   CURRENT! = VAL(MID$(DATA_STRING$,4))
15190   VOLT! = VAL(MID$(DATA_STRING$,16))
15200   IF CURRENT! => Parameter!(1,17) THEN EXIT
15210   REPEAT
15211   DATA_STRING$ = "S1" + CHR$(13) + "X"
15212   PARAM$ = "WR.STR/16//EOS/":GOSUB 50000
15235   TIMEDELAY (10)
15237   PARAM$ = "RD.STR/16//EOS/":GOSUB 50000
15240   VOLT2! = VAL(MID$(DATA_STRING$,5))
15250   PRINT #1,M%;",";CURRENT!;",";VOLT!;",";VOLT2!
15260   IF M% = 1 AND Print_temp% = 2 THEN DATA_STRING$ = "S1" ELSE DATA_STRING$ = "SWO"
15270   PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
15280   DATA_STRING$ = "W7"
15290   PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
15300   IF M% = 1 AND Print_temp% = 2 THEN TIMEDELAY (20):
15310   DATA_STRING$ = "Z1" + CHR$(13) + "X":PARAM$ = "WR.STR/16//EOS/":GOSUB 50000:TIMEDELAY (15)
15310   NEXT M%
15320   PRINT #1,"END"
15330   CLOSE #1
15340   PARAM$ = "SDC/5,16,17/":GOSUB 50000
15350   RETURN
16000   REM
16001   REM ************************************************
16002   REM ************************************************
16003   REM ************************************************
16004   REM ************************************************
16005   REM ************************************************
16006   REM ************************************************
16007   REM
16030   OPEN "C:\DATA\" + File$ + "\RES" FOR APPEND AS #1
16040   Bias! = Parameter!(1,17)
16050   LOCATE 14,17:PRINT "taking 4-POINT RESISTIVITY measurements"
16080   PRINT #1,Xtemp$;Parameter!(25,4);";";ACTUALTEMP
16090   PRINT #1,"TIME = ";Timerun!;" (";";TIMER;" )"
16100   PRINT #1,"BIAS = ";Bias!
DATA_STRING$ = "RO" + CHR$(13) + "X"
PARAM$ = "END/16/";GOSUB 50000
TIMEDELAY(5)
PARAM$ = "END/16/";GOSUB 50000
VOLT2! = VAL(MID$(DATA_STRING$, 5))
PRINT #1, VOLT2!", "VOLT2!", "ACTUALTEMP
TIMEDELAY(1)
CLOSE #1
PARAM$ = "END/16/";GOSUB 50000
RETURN
STOP
REM ********************&****************************
This stops the program after a successful run.
REM ********************&****************************
IF ExptX(25) <> 3 THEN GOSUB 41500
IF ExptX(25) = 1 THEN RESTORE, 60210
IF ExptX(25) = 2 THEN RESTORE, 60220
IF ExptX(25) = 3 THEN RESTORE, 60230
IF ExptX(25) = 4 THEN RESTORE, 60240
IF ExptX(25) = 5 THEN RESTORE, 60250
IF ExptX(25) = 6 THEN RESTORE, 60260
IF ExptX(25) = 7 THEN RESTORE, 60270
IF ExptX(25) = 8 THEN RESTORE, 60280
IF ExptX(25) = 9 THEN RESTORE, 60290
IF ExptX(25) = 10 THEN RESTORE, 60300
FOR M$ = 1 TO GroupX(ExptX(25))
READ Name$
OPEN "\DATA\" + File$ + ":\" + Name$ FOR APPEND AS #1
PRINT #1, "END"
CLOSE #1
NEXT M$
REM ********************&****************************
This section
REM ********************&****************************
finds out whether the user wants to
41005 REM * another experimental run.

41006 REM *

41007 REM ************************************************************
41008 REM *
41009 REM *************
41010 REM *

41011 REM PARAM$ = "ABORT/":GOSUB 50000
41012 REM CLS:COLOR 10,0,0:LOCATE 12,5
41013 INPUT "Do you want to run another set of experiments (Defaults to No)";Chk$
41014 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 41060 ELSE GOTO 41100
41015 OPEN "\DATA\REDO" FOR OUTPUT AS #1
41016 PRINT #1,Chk$
41017 CLOSE #1
41018 SYSTEM
41019 STOP

41020 REM *
41021 REM ************************************************

41022 DQTR_STRING$ = "P"
41023 PQRRMS = "NR.STR/7//EOS/":GOSUB 50000
41024 DDTR_STRING$ = "00210000000001117501"
41025 PQRQMS = "NR.STR/7//EOS/":GOSUB 50000
41026 DRTQ_STRING$ = "R"
41027 PQRQMS = "NR.STR/7//EOS/":GOSUB 50000
41028 PQRRMS = "RD.STR/7//EOS/":GOSUB 50000
41029 RETURN

41030 REM *
41031 REM ************************************************

41032 REM * This subroutine gives the Cryostat default parameters when the program RUNIT is finished.

41033 REM *

41034 REM ************************************************

41035 DATA_STRING$ = "P"
41036 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
41037 DATA_STRING$ = "00210000000001117501"
41038 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
41039 DATA_STRING$ = "R"
41040 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
41041 PARAM$ = "RD.STR/7//EOS/":GOSUB 50000
41042 RETURN

41043 REM *
41044 REM ************************************************

41045 REM * This is the ERROR handling routines. It checks to see what error is and attempts to fix it.
* 42006 REM *

* 42007 REM **************************************************

42008 REM IF ERR = 1008 THEN RESUME

42020 PRINT "Sorry the main program is bombing. This is the error "; ERR
42030 PRINT "The line number is "; ERL
42040 STOP

42050 REM 4

42060 REM IEEE-488 INTERFACE FOR THE IBM PC V4_2

42070 REM WRITTEN IN ADVANCED BASIC

42080 REM AND INCORPORATING ASSEMBLY LANGUAGE ROUTINES TO IMPLeMENT

42090 REM DMA - DRIVEN GPIB TRANSACTIONS

42100 REM THE ASSEMBLY LANGUAGE ROUTINES MUST BE LOADED PRIOR TO ENTERING

42110 REM BASICA BY TYPING "SUBLIB"_ THEN TYPE "BASICA", LOAD IEEE488_BAS,

42120 REM AND CALL SUBROUTINES AS DESCRIBED IN THE MANUAL_

42130 REM


42150 REM

42160 REM ************************************************** START OF SUBROUTINE ******

42170 END

60000 REM

60001 REM **************************************************

60002 REM *

60003 REM * This data statement tells the program how many

60004 REM * experiments exist in each group.  

60005 REM *

60006 REM **************************************************

60007 REM

60010 DATA 4,4,2,1,2

60099 STOP

60100 REM

60101 REM **************************************************

60102 REM *
60103 REM * This data statement stores what type of temperature it is in the string XTEMP$.
60104 REM *
60105 REM *
60106 REM ******************************************************************************************************
60107 REM
60108 DRT9 "TTEMP = ", "LTEMP = ", "RTEMP = 
60109 STOP
60120 REM
60121 REM *
60122 REM * These data statements are used to print the word "END" in each of the data files to show where the file quits. *
60123 REM *
60124 REM ************************************************
60125 *************
60126 REM
60127 DQT9 "C—T", "G—T", "CBV", "CG"
60128 DQTQ "IV", "CV"
60129 DQT9 "MOB"
60130 DQTR "RES"
60131 DQT9 
60132 DRTR 
60133 DRTQ 
60134 DRTQ 
60135 DQTQ 
60136 DRTQ 24
60140 REM
60141 DGTR "CQBLE 1 to Va", "CQBLE 2 to I high", "CQBLE 3 to V- of the Keithley", "CQBLE 4 to V+ of the Keithley", "I low to Ground"
60142 DQTQ "CQBLE 1 to V- of the Keithley", "CQBLE 2 to I high", "CQBLE 3 to Va", "CQBLE 4 to V+ of the Keithley", "I low to Ground"
60143 DQTQ "CQBLE 1 to Va", "CQBLE 2 to V+ of the Keithley", "CQBLE 3 to I high", "CQBLE 4 to V- of the Keithley", "I low to Ground"
60144 ENDFILE
SOURCE
PRECISION= 7
AUTODEF=ON
OPTION BASE=1
ERR=ON
ERRORMODE=LOCAL
RESUME=LINE
FORMODE=BB
SCOPE=ON
PROCS=49
STRING ARRAY(5,24)[32]: Graph_name$
STRING ARRAY(2,2)[5]: Directory$
STRING ARRAY(2)[20]: Tyme$, Temp$, Bias$, Hold$, Rate$, Slope$, Y_int$, X_int$
STRING ARRAY(2)[20]: Range$
STRING ARRAY(2)[20]: Correlation$, Pulse$
STRING ARRAY(2)[3]: Temp_type$
INTEGER ARRAY(5): Maxloop%
INTEGER ARRAY(2): Max_graph_points%
BYTE ARRAY(4003): Menu1, Menu2, Menu3, Menu4
BYTE ARRAY(10): Plus
REAL ARRAY(2,2,1200): Graph!
REAL ARRAY(2): X_int!, Y_int!, Slope!, Correlation!
REAL ARRAY(2): Min!, Max!
INTEGER: Rownumber%, Col%, Menu_chosen%, Graph_chosen%, Error%, Minloop%, Print_it%
INTEGER: Which_graph%, Graph_type%, M%, N%, Next_screen%, Choice%, Comp_graph%, Graph%
INTEGER: File%, Graph_pick%, ID%, Print_out%, Info_pointer%, Max_x_graph_number%, Pointer%, Return%
INTEGER: Position%
REAL: MinX!, MaxX!, MinY!, Q!, K!, E_sub_S!, Area!, Slope!
REAL: AeA!, Vf!, If!, Min!, Max!, Y_int!, X_int!, Graph!
REAL: MaxY!, Temp_sought!, Time_sought!, Dummy!, DeltaX!, DeltaY!, Y_plot!, X_plot!
REAL: Min_tyme!, Max_tyme!, Bias_sought!, Pulse_sought!, Hold_sought!, Temp!, Y!, Nc!
REAL: Nd!, X!, E_sub_O!, Elec_thick!, I!, V1!, V2!, R1!
REAL: R2!, G!, Eric!
STRING: File$[5], Xtitle$[77], Ytitle$[77], A$[3], Check$[3], Temp_found$[25]
STRING: Time_found$[16], Bias_found$[12], Pulse_found$[17], Hold_found$[33], Name$[30], Save_file$[40], Area$[16], Nd$[16]
STRING: V$[?]
INTEGER: Counter%, Skip%
STRING ARRAY(12)[32]: Hand_info$
INTEGER: Run_type%
STRING ARRAY(2)[18]: GRange$
INTEGER: Minloop1%, Maxloop%
STRING: Change$[?], Fit$[?], Symbol$[?]
INTEGER: Error_LS%
STRING: Title$, Dummy$, I$]
INTEGER: Max_graph_points%
STRING ARRAY(?): Ra$
STRING: V1$, V2$]
REAL: Power!
INTEGER: Pen_selected%
STRING: Bias$, Line_type[5], Line_type$[16], SYMBOLS$]
STRING ARRAY(5)[16]: SYMBLS$]
STRING ARRAY(5)[16]: LINETYPE$
INTEGER ARRAY(8): PEN
REAL: X_plot1!, Y_plot1!
INTEGER: Graph1or2%, G1, G2, G
INTEGER: Onoff
STRING: SOURCE$[16$
REAL: N, T, Voltage, In, Volts, O
INTEGER ARRAY(?): Skippoint%
REAL ARRAY(?): Plot!
INTEGER: 0%, P%
INTEGER ARRAY(?): Min%
INTEGER: Min%
REAL: Thick!, BW
STRING: HG$[?$
REAL: CF
INTEGER: MOD, GFLAG, Saveflag
STRING: SPEED$[5$

PROCEDURE: TITLE()
  STRING ARG: Title$
END PROCEDURE

PROCEDURE: MENU()
  STRING ARG: Name$
  INTEGER ARG: Number%
  INTEGER ARG: Rownumber%/VAR
END PROCEDURE

PROCEDURE: FINISH()
  STRING ARG: Title$
  INTEGER ARG: Rownumber%/VAR, Col%/VAR
  INTEGER ARG: Number%
END PROCEDURE

PROCEDURE: BORDER()
  INTEGER ARG: Rownumber%/VAR
END PROCEDURE

PROCEDURE: CLEARSCREEN()
  INTEGER ARG: Row%
END PROCEDURE

PROCEDURE: DIRECTORY()
PROCEDURE: TIMEDELAY()
    REAL ARG: Delay!/OPT=5!
END PROCEDURE

PROCEDURE: XTITLE()
    STRING ARG: Title$
    REAL ARG: Min!, Max!
END PROCEDURE

PROCEDURE: YTITLE()
    STRING ARG: Title$
    REAL ARG: Min!, Max!
END PROCEDURE

PROCEDURE: LEASTSQUARES()
    INTEGER ARG: Which_graph%
    REAL ARG: Min!/VAR, Max!/VAR, Slope!/VAR, Correlation!/VAR
    , X_int!/VAR, Y_int!/VAR
    INTEGER ARG: Skip%
    INTEGER ARG: Error_LS%/VAR
END PROCEDURE

PROCEDURE: RUNTYPE()
    INTEGER ARG: Graph1orE%, Skip%
END PROCEDURE

PROCEDURE: DELTAMULTIPLY()
    INTEGER ARG: Graph1orE%, XorY%
    REAL ARG: Amount!
END PROCEDURE

PROCEDURE: DELTA1n()
    INTEGER ARG: Graph1orE%, XorY%
    REAL ARG: Amount!
END PROCEDURE

PROCEDURE: DATARETRIEVAL()
    INTEGER ARG: Graph_ch0san%, Graph1orE%, File%
    INTEGER ARG: Error%/VAR
    INTEGER ARG: Run_type%
END PROCEDURE

PROCEDURE: DATAPRINT
END PROCEDURE

PROCEDURE: PARAMETERSET()
PROCEDURE: PLOTTERAXES()
    STRING ARG: Title$
END PROCEDURE

PROCEDURE: SAVEDATA()
    INTEGER ARG: Graph%
END PROCEDURE

PROCEDURE: LOADDATA()
    INTEGER ARG: Graph%
END PROCEDURE

PROCEDURE: INVERSEPOWER()
    INTEGER ARG: Graph1or2%,Xory%
    REAL ARG: Amount!,Power!
END PROCEDURE

PROCEDURE: MAINKEY
END PROCEDURE

PROCEDURE: PLOTTER
END PROCEDURE

PROCEDURE: MODIFY
END PROCEDURE

PROCEDURE: COLORS
END PROCEDURE

PROCEDURE: SYMBOLS
END PROCEDURE

PROCEDURE: LINES
END PROCEDURE

PROCEDURE: Info
END PROCEDURE

PROCEDURE: CAPTION
PROCEDURE: Fixpoints
   INTEGER ARG: Typa%
   REAL ARG: Plotit!
   REAL ARG: P1t!/VAR, P1t1!/VAR
END PROCEDURE

PROCEDURE: Cleargraph
   INTEGER ARG: G1orG2
END PROCEDURE

PROCEDURE: CLRGRPH
END PROCEDURE

PROCEDURE: ADDNUM
END PROCEDURE

PROCEDURE: Mltnum
END PROCEDURE

PROCEDURE: POWER
END PROCEDURE

PROCEDURE: NATLOG
END PROCEDURE

PROCEDURE: EXPNT
END PROCEDURE

PROCEDURE: ABSLT
END PROCEDURE

PROCEDURE: DELPT
END PROCEDURE

PROCEDURE: ADDPT
END PROCEDURE

PROCEDURE: SPEED
END PROCEDURE

PROCEDURE: TITLE
   INTEGER: Co1%
   EXTERNAL: G1, G2, GRAPH%, GFLAG
   100 CLS: COLOR 5, 0
   104 SET CURSOR 0, 70: COLOR 2, 7: PRINT GFLAG: COLOR 5, 0
   105 SET CURSOR 0, 74: COLOR 8, 7: PRINT G1; GE: COLOR 5, 0
   110 Co1% = CINT((80 - LEN(Title$))/2)
   120 SET CURSOR 0, Co1%
130 PRINT Title$
END PROCEDURE

PROCEDURE: MENU
INTEGER: M%, Placement%, Row%, Col%
STRING: Number$[5]
90 COLOR 3,0
95 IF Number% = 1 THEN Row% = -1
100 FOR M% = 1 TO 4
110 IF Number% <= M% * 24 THEN EXIT
120 NEXT M%
130 Placement% = Number% - (M%-1)*25
140 IF Number% > 24 THEN Placement% = Placement% + M% - 1
150 Number$ = STR$(Number%)
160 DEL$(Number$,1,1)
170 IF Placement% < 13 THEN Col% = 5 ELSE Col% = 42 + 1
180 IF Placement% = 13 THEN Row% = 0 ELSE Row% = Row% + 1
190 SET CURSOR Row% + 4,Col%
200 PRINT "(";Number$;") ";Name$
210 IF Rownumber% < CSRLIN THEN Rownumber% = CSRLIN
220 COLOR 7,0
230 EXIT
END PROCEDURE

PROCEDURE: FINISH
STRING: Number$[5]
100 COLOR 3,0
110 Rownumber% = Rownumber%
120 Number$ = STR$(Number%)
130 DEL$(Number$,1,1)
140 IF Col% = 0 THEN Col% = CINT((73 - LEN(Tit1e$))/2)
150 SET CURSOR Rownumber%,Col%
160 PRINT "(";Number$;") ";Tit1e$
170 Col% = 0
180 COLOR 7,0
190 EXIT
END PROCEDURE

PROCEDURE: BORDER
INTEGER: M%
100 COLOR 2,0
110 SET CURSOR 2,1
120 PRINT "******************************************************************************
******************************************************************************
";
130 PRINT "****************************************************************************
140 Rownumber% = Rownumber% + 1
150 FOR M% = 3 TO Rownumber%
160 SET CURSOR M%,1
170 PRINT "*"
180 SET CURSOR M% , 79
190 PRINT "*"
200 NEXT M%
210 Rownumber% = Rownumber% + 1
220 SET CURSOR Rownumber%, 1
230 PRINT "*******************************************************************************"
240 PRINT "*******************************************************************************"
250 Rownumber% = 0
260 COLOR 7, 0
270 EXIT
END PROCEDURE

PROCEDURE: CLEARSCREEN
INTEGER: M%
100 FOR M% = Row% TO 23
110 SET CURSOR M%, 0
120 PRINT SPC(80)
130 NEXT M%
140 EXIT
END PROCEDURE

PROCEDURE: DIRECTORY
INTEGER: M%, Start%, Comp_graph%
EXTERNAL: Directory$(>, Number$[8])
INTEGER: N%, Stop%
REAL: M
STRING: A$[16]
EXTERNAL: SOURCE$

PROCEDURE: TIMEDELAY
END PROCEDURE

PROCEDURE: TIMEDELAY
REAL: Time!
100 Time! = TIMER + 4
110 IF TIMER < Time! THEN GOTO 100
120 EXIT
130 END
END PROCEDURE

90 CLOSE
100 ON ERROR GOTO 10000
110 Start% = 1
115 Stop% = 1
120 Error% = 0
131 SET CURSOR 4, 6: COLOR 7, 0
133 INPUT "IS THE DATA SOURCE C OR B DRIVE"; SOURCE$
134 SOURCE$ = UPPER$(SOURCE$)
140 FOR M% = Start% TO Stop%
213
141 FOR N% = 5+2*M% TO 23
142   LOCATE N%,1:PRINT SPC(79)
143   NEXT N%
170 CLEAR (ERR)
180 COLOR 2,0:SET CURSOR 5+2*M%,6
190 PRINT "Enter the data source directory number (OR SAMPLE/RUN#);"
200 INPUT Directory$(Graph1or2%,M%)
210 IF Directory$(Graph1or2%,1) = "" AND Comp_graph% = 0 THEN GOTO 170
230 IF INSTR(SOURCE$,"H")=0 THEN OPEN "\DGTG\" + Directory$(Graph1or2%,M%) + File$ FOR INPUT AS #M% ELSE OPEN "B:" + File$ + "." + Directory$(Graph1or2%,M%) FOR INPUT AS #M%
240 NEXT M%
245 Directory$(Graph1or2%,2) = "NO"
250 Comp_graph% = 1
260 EXIT
270 STOP
10000 REM
10010 COLOR 7,0:SET CURSOR 20,5
10020 IF ERR = 1001 AND M% = 1 AND Comp_graph% = 0 THEN GOSUB 10500:RESUME,260
10030 IF ERR = 1001 AND M% = 1 AND Comp_graph% = 1 THEN GOSUB 10600:RESUME,110
10040 IF ERR = 1001 AND M% = 2 THEN GOSUB 10600:RESUME,120
10050 IF ERR = 1007 THEN GOSUB 10700:RESUME,120
10300 PRINT "THIS IS AN UNEXPECTED ERROR IN PROCEDURE DIRECTORY IN LINE ";ERL;"AND ERROR NUMBER";ERR
10310 STOP
10500 REM
10510 PRINT "The graph you asked for does not exist in the directory chosen."
10520 SET CURSOR 21,5:PRINT "Please choose another graph.
10530 TIMEDELY
10540 Error% = 1
10550 RETURN
10600 REM
10610 PRINT "The graph you asked for does not exist in the directory chosen."
10620 SET CURSOR 21,5:PRINT "Please choose another directory."
10630 TIMEDELY
10640 Start% = M%
10650 RETURN
10700 REM
10710 PRINT "The directory you asked for does not exist."
10720 SET CURSOR 21,5:PRINT "Please choose another directory."

```plaintext
PROCEDURE: TIMEDELAY
  REAL: Waiting!
  100 Waiting! = TIMER + Delay!
  110 IF TIMER < Waiting! THEN GOTO 110
  120 EXIT
END PROCEDURE

PROCEDURE: XTITLE
  INTEGER: ColZ,MZ
  100 ON ERROR GOTO 10000
  110 LOCATE 22,4:PRINT USING "#.##^^^^";Min!
  120 LOCATE 22,72:PRINT USING "#.##^^^^";Max!
  150 DRAW "BM105,163 D3 BR57 U3 BR57 D3 BR57 U3 BR57 D3 BR57 U3 BR57 D3 BR57 U3 BR57 D3 BR57 U" 
  155 DO 1 TIMES
  160 DO 1 TIMES
  170 IF RBS(Min!) (> Max!) THEN EXIT
  180 LOCATE 22,42:PRINT "0"
  185 EXIT 2 LEVELS
  190 REPEAT
  200 DO 1 TIMES
  210 IF (Max! - Min!) MOD 10 = 0 THEN EXIT
  220 FOR MZ = 1 TO 9
  230 LOCATE 22,5 + MZ*7.5
  240 IF MZ + Min! = 0 THEN EXIT
  250 NEXT MZ
  260 IF MZ < 10 THEN PRINT "0"
  270 REPEAT
  275 REPEAT
  280 Co1Z = CINT((79 - LEN(Tit1e$))/2)
  290 LOCATE 23,ColZ:PRINT Tit1e$
  300 EXIT
  10000 REM
  10010 VIEW:SCREEN 0,0,0:COLOR 7,0:SET CURSOR 12,5
  10020 PRINT "Sorry the procedure XTITLE is bombing. This is error number ";ERR;" from line number ";ERL
  10030 STOP
END PROCEDURE

PROCEDURE: YTITLE
  INTEGER: Max%,Row%,M%
  STRING: Y$[3]
  100 ON ERROR GOTO 10000
  110 DRAW "BM48,15 R570 BL570 BD15 R3 BD15 L3 BD15 R3 BD15 L3 BD15 R3 BD15 L3 BD15 R3 BD15 L3 BD15 R3 BD15 BL3 U
```
215"

120 MaxX = LEN(Title$)
130 Row% = CINT(((23 - MaxX)/2) - 2
135 IF Row% < 0 THEN Row% = 0
137 IF MaxX > 23 THEN MaxX = 23
140 FOR MZ = 1 TO MaxX
150 Y$ = MIDS(Title$, MZ, 1)
160 SET CURSOR Row% + MZ,0:PRINT Y$
170 NEXT MZ
180 SET CURSOR 1,0:PRINT USING "#.##^^^^";Max!
190 SET CURSOR 20,0:PRINT USING "#.##^^^^";Min!
200 EXIT
10000 CLS: VIEW: SCREEN 0,0,0: COLOR 7,0: SET CURSOR 11,5
10010 PRINT "The Xprocedure YTITLE is mbing. This is e
rror ";ERR;" from line ";ERL
10020 STOP
END PROCEDURE

PROCEDURE: LEASTSQUARES
EXTERNAL: Graph!(),Max_graph_points%()
INTEGER: Minloop%,Maxloop%,M%,Total_samples%
REAL: SumX2!,SumY2!,SumXY!,SumX!
REAL: SumY!,TotalX2!,TotalY2!,Xmean!,Ymean!,Sxx!,Syy!,S
xy!
INTEGER: Chich_graph%
REAL: Min1!,Max1!
INTEGER: N%
REAL: XTOTAL!,YTOTAL!
STRING: Min1$[?],Max1$[?]
INTEGER: Minloop1%,Step%,Maxloop1%
STRING ARRAY(?): SLOPES,X_INTS,Y_INTS
STRING ARRAY(?): RANGES,CORRELATIONS,GRANGES
REAL#: Correlation!
80 Error_LS% = 0
90 ON ERROR GOTO 60000
95 Minloop1% = 1: Maxloop1% = Max_graph_points%(Which
h_graph%):Step% = 1
100 CLEAR (SumX2!,SumY2!,SumX!,SumY!,SumXY!,Sxx!,Syy!
,Sxy!,Xmean!,Ymean!,XTOTAL!,YTOTAL!)
105 IF Skip% = 1 THEN GOTO 165
107 GOSUB 20000
110 COLOR 2,0; SET CURSOR 13,1
120 INPUT "Over what X-axis range do you want your le
ast square fit (min,max)";Min1$,Max1$
125 Min1! = VAL(Min1$); Max1! = VAL(Max1$)
130 GOSUB 20000
135 DO
140 SET CURSOR 13,32
150 PRINT "Now LOADING data"
160 END DO
165 IF Graph!(Which_graph%,1,Max_graph_points%(Which_
(Graph!(Which_graph%,1,1) THEN Minloop1% = Max_graph_points%(Which_graph%):Maxloop1% = 1 :Step% = -1
168 Min! = Min1! : Max! = Max1!
170 FOR MX = Minloop1% TO Maxloop1% STEP Step%
180 IF Graph!(Which_graph%,1,MX) =) Min1! THEN EXIT
190 NEXT MX
200 Minloop% = MX
210 FOR MX = Minloop1% TO Maxloop1% STEP Step%
220 IF Graph!(Which_graph%,1,MX) =) Max1! THEN EXIT
230 NEXT MX
240 Maxloop% = MX
300 FOR MX = Minloop% TO Maxloop% STEP Step%
310 SumX2! = SumX2! + Graph!(Which_graph%,1,MX) ^ 2
320 SumY2! = SumY2! + Graph!(Which_graph%,2,MX) ^ 2
330 SumXY! = SumXY! + Graph!(Which_graph%,1,MX) * Graph!(Which_graph%,2,MX)
340 SumX! = SumX! + Graph!(Which_graph%,1,MX)
350 SumY! = SumY! + Graph!(Which_graph%,2,MX)
360 NEXT MX
400 DO
410 TotalX2! = SumX! * SumX!
420 TotalY2! = SumY! * SumY!
430 Total_samples% = Maxloop% - Minloop% + 1
440 Xmean! = SumX!/Total_samples%
450 Ymean! = SumY!/Total_samples%
460 Sxx! = SumX2! - TotalX2!/Total_samples%
470 Syy! = SumY2! - TotalY2!/Total_samples%
480 Sxy! = SumXY! - (SumX! * SumY!)/Total_samples%
490 END DO
600 DO
610 Slope! = Sxy!/Sxx!
620 Correlation! = Sxy!/SQR(Sxx! * Syy!)
630 Y_int! = Ymean! - (Slope! * X_mean!)!
640 X_int! = -1 * (Y_int! / Slope!)
650 END DO
660 Min! = Min1! : Max! = Max1!
700 EXIT
10010 STOP
20000 DO
20010 FOR N% = 3 TO 23
20020 LOCATE N%,1 : PRINT SPC(79)
20030 NEXT N%
20035 END DO
20040 RETURN
60000 IF ERR < 2 AND ERR > 3 THEN GOSUB 20000 ELSE GOTO 62000
60010 COLOR 7,0 : SET CURSOR 13,1
60999 COLOR 7,0 : SET CURSOR 13,1
61000 PRINT "The procure LEASTSQUARES is boxbing. ThX
217

s is error; ERR; from line ; ERL
61010 FOR Mx = -30000 TO 30000 : NEXT Mx
61020 Error_LS = 1
61030 EXIT
62000 Slope! = 0: Correlation! = 0: Y_int! = 0: X_int! = 0
62010 EXIT
END PROCEDURE

PROCEDURE: RUNTYPE
STRING: Check$[10]
INTEGER: Mx
EXTERNAL: Temp_sought!, Time_sought!, Temp_type$()
STRING: T$[10]
100 DO
110 FOR Mx = 2 TO 23
120   LOCATE Mx, 1: PRINT SPC(79)
130   NEXT Mx
140 COLOR 2, 0, 0: SET CURSOR 5, 6
150 INPUT "Enter the Type of Temperature run (T, L, R )"; Temp_type$(Graph1or2%)
160   Temp_type$(Graph1or2%) = UPPER$(Temp_type$(Graph1or2%))
170 IF INSTR(Temp_type$(Graph1or2%), "L") <> 0 THEN EXIT
180 IF INSTR(Temp_type$(Graph1or2%), "R") <> 0 THEN EXIT TO, 400
190 IF INSTR(Temp_type$(Graph1or2%), "T") <> 0 THEN EXIT TO, 600
200 REPEAT
210   IF Skipx = 1 THEN EXIT
220   SET CURSOR 7, 6
230   INPUT "Enter the Temperature at which to read the data"; Temp_sought!
240 END DO
250 DO
260   COLOR 7, 0, 0: SET CURSOR 22, 6
270   INPUT "Do you want to change any of the above ( Defaults to NO)"; Check$
280   Check$ = UPPER$(Check$)
290 IF INSTR(Check$, "Y") <> 0 THEN EXIT TO, 100
300 END DO
310 EXIT
320 DO
330   IF Skipx = 1 THEN EXIT 1 LEVELS
340   SET CURSOR 7, 6
350   INPUT "Enter the time at which to read the data (min)"; T$
360   Temp_sought! = CINT(60*VAL(T$))
370 END TO, 400
380 END DO
PROCEDURE: DELTAMULTIPLY

EXTERNAL: Graph!(), Max_graph_points%(), Miny!, Maxy!, Minx!, Maxx!
INTEGER: M%, Xor%
REAL: Min!, Max!
EXTERNAL: G1, G2

10 DO
15 IF G1=1 AND Graph1or2% = 2 THEN EXIT 1 LEVELS
20 DO
25 IF XorY% = 2 THEN EXIT 1 LEVELS
30 Minx! = 1E+29 : Maxx! = -1E+29
35 EXIT 2 LEVELS
37 END DO
40 DO
45 Miny! = 1E+29 : Maxy! = -1E+29
50 END DO
55 END DO
100 DO
120 Min! = 1E+26 : Max! = -1E+27
130 END DO
500 FOR M% = 1 TO Max_graph_points%(Graph1or2%)
510 Graph!(Graph1or2%, XorY%, M%) = Graph!(Graph1or2%, XorY%, M%) * Amount!
520 IF Min! > Graph!(Graph1or2%, XorY%, M%) THEN Min! = Graph!(Graph1or2%, XorY%, M%)
530 IF Max! < Graph!(Graph1or2%, XorY%, M%) THEN Max! = Graph!(Graph1or2%, XorY%, M%)
540 NEXT M%
550 DO
560 IF XorY% = 2 THEN EXIT 1 LEVELS
570 IF Min! > Min! THEN Minx! = Min!
580 IF Max! < Max! THEN Maxx! = Max!
590 EXIT 2 LEVELS
600 END DO
610 DO
620 IF Miny! > Min! THEN Miny! = Min!
630 IF Maxy! < Max! THEN Maxy! = Max!
640 END DO
END PROCEDURE

PROCEDURE: DELTALn

EXTERNAL: Graph!(), Max_graph_points%(), Miny!, Maxy!, Minx!, Maxx!
REAL: Min!, Max!
INTEGER: M%
EXTERNAL: G1, G2
100 DO
110 DO
120 IF G1=1 AND Graph1or2% = 2 THEN EXIT
130 Min! = 1E+19
140 Max! = -1E+29
150 EXIT 2 LEVELS
160 END DO
170 DO
180 IF XorY% = 2 THEN EXIT
190 Min! = Minx!
200 Max! = Maxx!
210 EXIT 2 LEVELS
220 END DO
230 DO
240 Min! = Miny!
250 Max! = Maxy!
260 EXIT 2 LEVELS
270 END DO
280 END DO
300 FOR M% = 1 TO Max_graph_points%(Graph1or2%)
310 DO
320 IF Graph!(Graph1or2%,XorY%,M%) <> 0 THEN EXIT 1 LEVELS
330 Graph!(Graph1or2%,XorY%,M%) = -1000
340 EXIT 2 LEVELS
350 END DO
360 DO
370 Graph!(Graph1or2%,XorY%,M%) = LOG(Abs(Graph!(Graph1or2%,XorY%,M%))*Amount!)
380 END DO
390 IF Min! > Graph!(Graph1or2%,XorY%,M%) THEN Min! = Graph!(Graph1or2%,XorY%,M%)
400 IF Max! < Graph!(Graph1or2%,XorY%,M%) THEN Max! = Graph!(Graph1or2%,XorY%,M%)
410 NEXT M%
500 DO
510 DO
520 IF XorY% = 2 THEN EXIT 1 LEVELS
530 Minx! = Min!
540 Maxx! = Max!
550 EXIT 2 LEVELS
560 END DO
570 DO
580 Miny! = Min!
590 Maxy! = Max!
600 END DO
610 END DO
1000 EXIT
END PROCEDURE

PROCEDURE: DATA_RETRIEVAL
EXTERNAL: Graph!(), Max_graph_points%(), Min![], Max![], Miny![], Maxy![], Temp_sought!, Bias_found$
EXTERNAL: Hold_found$, Pulse_found$, Time_sought!, Bias_sought!, Pulse_sought!, Hold_sought!, Temp_type$(), Temp_found$
EXTERNAL: Time_found$
INTEGER: Position%, Magnet_on%, Graph%, M%, N%
STRING: Data1$[32], Data2$[32], Data3$[32]
REAL: Min!, Max!
REAL: X!, Y1!, Y2!, X1!, Tyme!, R31!, R32!, R3!
REAL: Dummy!, Y1
EXTERNAL: Mintyme!, Maxtyme!
INTEGER: Pointer%
REAL ARRAY(?): Graph
EXTERNAL: G!, Elec_thick!
INTEGER: Mistake%
EXTERNAL: Menu_chosen%

20 REM * GRAPH_CHOSEN%
21 RUN
30 REM * 1 = C-t, G-t
40 REM * 1 = C-V, 1/C**2, 1/C**3
50 REM * 2 = C-T
60 REM * 3 = G-V
70 REM * 4 = C-T
80 REM * 5 = G-V
90 REM * 6 = I-V
100 REM * 7 = C-V, 1/C**2
110 REM * 8 = DLTS
120 REM * 9 = MDO, Rho vs. Temp
130 REM * 10 = MDO, Mu vs. Temp
140 Error% = 0
150 ON ERROR GOTO 50000
160 DO
170 CLS: COLOR 7, 0, 0: SET CURSOR 13, 6
180 PRINT "ERROR, the graph_chosen% variable is not defined in the procedure dataretrieval"
190 STOP
200 DO 1 TIMES
221

530 INPUT #File%, Temp_found$
540 IF INSTR(Temp_found$, "END") <> 0 THEN Error
% = 1: EXIT TO, 60000
550 IF INSTR(Temp_type$(Graph1or2%), "T") <> 0 OR INSTR(Temp_type$(Graph1or2%), "R") <> 0 OR Graph_chosen% = 4 OR Graph_chosen% = 5 OR Graph_chosen% = 8 THEN EXIT 1 LEVELS
560 Position% = INSTR(Temp_found$, ":") + 1
570 IF VAL(MID$(Temp_found$, Position%)) <> Temp_sought! THEN EXIT 2 LEVELS
580 Do 1 TIMES
590 INPUT #File%, Time_found$
600 IF INSTR(Temp_type$(Graph1or2%), "T") = 0 OR Graph_chosen% = 4 OR Graph_chosen% = 5 OR Graph_chosen% = 8 THEN EXIT 1 LEVELS
610 Position% = INSTR(Time_found$, ":") + 1
620 IF VAL(MID$(Time_found$, Position%)) <> Time_sought! THEN EXIT 2 LEVELS
630 Do 1 TIMES
640 IF Graph_chosen% <> 1 AND Graph_chosen% <> 4 AND Graph_chosen% <> 5 AND Graph_chosen% <> 8 THEN EXIT 3 LEVELS
650 INPUT #File%, Bias_found$
660 Position% = INSTR(Bias_found$, ":") + 1
670 IF VAL(MID$(Bias_found$, Position%)) <> Bias_sought! THEN EXIT 2 LEVELS
680 Do 1 TIMES
690 IF Graph_chosen% <> 1 AND Graph_chosen% <> 8 THEN EXIT 3 LEVELS
700 INPUT #File%, Pulse_found$
710 Position% = INSTR(Pulse_found$, ":") + 1
720 IF VAL(MID$(Pulse_found$, Position%)) <> Pulse_sought! THEN EXIT 2 LEVELS
730 Do 1 TIMES
740 IF Graph_chosen% <> 4 AND Graph_chosen% <> 5 THEN EXIT 3 LEVELS
750 INPUT #File%, Hold_found$
760 Position% = INSTR(Hold_found$, ":") + 1
770 IF VAL(MID$(Hold_found$, Position%)) <> Hold_sought! THEN EXIT 2 LEVELS
780 Do 1 TIMES
790 INPUT #File%, Data1$
800 IF Graph_chosen% <> 4 AND Graph_chosen% <> 5 THEN EXIT 1 LEVELS
810 Do 1 TIMES
820 INPUT #File%, Data1$, Data1$
897 EXIT TO, 100
898 REPEAT
900 IF INSTR(Data1$, "END") <> 0 THEN EXIT 1 LEVELS
S
980 REPEAT
990 IF INSTR(Data1$, "END") <> 0 THEN EXIT 1 LEVELS
S
2000 FOR M% = Max_graph_points%(Graph1or2%) TO 1200
2005 IF Graph_chosen% = 9 OR Graph_chosen% = 10 THEN INPUT #File%, Magna_on$
2010 IF Graph_chosen% <> 1 AND Graph_chosen% < 6 THEN INPUT #File%, Data2$
2020 IF Graph_chosen% = 9 OR Graph_chosen% = 10 THEN INPUT #File%, Data3$
2030 IF INSTR(Data1$, "END") <> 0 OR INSTR(Data2$, "END") <> 0 OR INSTR(Data3$, "END") <> 0 THEN EXIT 1 LEVELS
2050 X! = VAL(Data1$)
2060 Y1! = VAL(Data2$)
2070 Y2! = VAL(Data3$)
2071 DO
2072 IF Graph_chosen% <> 2 AND Graph_chosen% <> 6 OR Graph_chosen% <> 8 THEN EXIT 1 LEVELS
2074 IF Y1! < Min! OR Y1! > Max! THEN EXIT TO, 2005
2075 END DO
2076 DO
2077 IF Graph_chosen% <> 5 THEN EXIT 1 LEVELS
2078 IF Y2! < Min! OR Y2! > Max! THEN EXIT TO, 100
2079 END DO
2080 DO
2081 IF Graph_chosen% <> 4 THEN EXIT 1 LEVELS: PRINT Y1!, 2081
2082 IF Y1! < Min! OR Y1! > Max! THEN EXIT TO, 100
2083 END DO
2084 DO
2085 IF Graph_chosen% <> 3 THEN EXIT 1 LEVELS
2086 IF Y2! < Min! OR Y2! > Max! THEN EXIT TO, 2005
2087 END DO
2088 DO 1 TIMES
2090 ON Graph_chosen% GOTO 2100, 2100, 2200, 2300, 230
0, 2100, 2100, 2600, 3000, 3300
2100 DO
2110 Graph!(Graph1or2%, 1, M%) = X!
2120 Graph!(Graph1or2%, 2, M%) = Y1!
2130 EXIT 2 LEVELS
2140 REPEAT
2200 DO
2210 Graph!(Graph1or2%, 1, M%) = X!
2220 Graph!(Graph1or2%, 2, M%) = Y2!
2230 EXIT 2 LEVELS
2240 REPEAT
2230    REM
2310    DO
2400    Graph!(Graph1or2%, 1, MK) = X!
2410    DO
2420    IF Graph_chosen% <> 4 THEN EXIT 1 LEVELS
2430    Graph!(Graph1or2%, 2, MK) = Y1!
2440    END DO
2450    DO
2460    IF Graph_chosen% <> 5 THEN EXIT 1 LEVELS
2470    Graph!(Graph1or2%, 2, MK) = Y2!
2480    END DO
2490    Max_graph_points%(Graph1or2%) = Max_graph_points%(Graph1or2%) + 1
2500    EXIT 3 LEVELS
2530    REPEAT
2600    REM
2610    DO 1 TIMES
2620    IF X! <> Mintyme! THEN EXIT 1 LEVELS
2630    DO
2635    IF INSTR(Data2$, "END") <> 0 OR INSTR(Data1$, "END") <> 0 THEN EXIT 4 LEVELS
2636    IF INSTR(Data2$, "END") <> 0 OR INSTR(Data1$, "END") <> 0 THEN EXIT TO, 100
2640    IF Y1! > Min! AND Y1! < Max! THEN EXIT 1 LEVELS
2650    INPUT #FileK, Data2$, Data1$: Y1! = VAL(Data2$)
2670    REPEAT
2680    Graph!(Graph1or2%, 2, MK) = Y1!
2690    N% = 2
2700    EXIT TO, 2010
2710    REPEAT
2750    DO 1 TIMES
2760    IF X! <> Maxtyme! THEN EXIT 1 LEVELS
2770    DO
2780    IF Y1! > Min! AND Y1! < Max! THEN EXIT 1 LEVELS
2790    INPUT #FileK, Data2$: INPUT #FileK, Data1$: Y1! = VAL(Data2$) GOTO 2770
2810    REPEAT
2820    Graph!(Graph1or2%, 2, MK) = Graph!(Graph1or2%, 2, MK) - Y1!
2830    N% = 0
2840    Max_graph_points%(Graph1or2%) = N% + 1
2850    DO 1 TIMES
2860    Position% = INSTR(Temp_found$, ";") + 1
2870    Graph!(Graph1or2%, 1, MK) = VAL(MID$(Temp_f
2890    DO
2900    INPUT #FileK, Data1$
2910   IF INSTR(Data1$, "END") <> 0 THEN EXIT TO, 4955
2920   REPEAT
2930   REPEAT
2935   EXIT TO, 2010
3000   DO
3010   DO
3020   IF INSTR(Magnet_on$, "1") <> 0 THEN EXIT 1
3030   INPUT #File%, Dummy$, Magnet_on$, Data2$, Data3$, Data1$
3040   REPEAT
3050   Y2! = VAL(Data3$)
3060   Y1! = VAL(Data2$)
3070   Graph!(Graph10r2%, 2, M%) = Y2! / Y1! * (Elec_thick!/G!)
3080   DO
3090   IF INSTR(Temp_type$(Graph10r2%), "T") <> 0 THEN EXIT 1
3100   Position% = INSTR(Temp_found$, ";") + 1
3110   Graph!(Graph10w2%, 1, M%) = VAL(MID$(Temp_found$, Position%, Position%))
3120   OPEN ".BUCKET" FOR APPEND AS #3
3130   PRINT #3, Graph!(Graph10r2%, 2, M%)
3140   CLOSE #3
3150   EXIT TO, 3700
3160   REPEAT
3170   DO
3180   Position% = INSTR(Time_found$, ";") + 1
3190   Graph!(Graph10r2%, 1, M%) = VAL(MID$(Time_found$, Position%, Position%))
3191   OPEN ".BUCKET" FOR APPEND AS #3
3192   PRINT #3, Graph!(Graph10r2%, 2, M%)
3193   CLOSE #3
3200   EXIT TO, 3700
3210   REPEAT
3220   REPEAT
3230   DO
3240   DO
3250   IF INSTR(Magnet_on$, "2") <> 0 THEN Mistake = 1 : EXIT TO, 60000
3260   R31! = X!/Y1!
3270   END DO
3280   DO
3290   INPUT #File%, Magnet_on$, Data2$, Data3$, Data1$
3300   IF INSTR(Magnet_on$, "1") <> 0 THEN Mistake = 1 : EXIT TO, 60000
3310   X! = VAL(Data2$)
3320   Y1! = VAL(Data1$)
3330   R32! = Y1!/X!
3410 END DO
3420 DO
3430 R3! = ABS(R32! - R31!)
3440 Graph!(Graph1or2%,2,M%) = R3!/(Graph!(Graph1or2%,2,M%) * 3.4E-05)
3450 END DO
3460 DO
3470 IF INSTR(Tamp_typa$(Graph1or2%),"T") <> 0 THEN EXIT 1 LEVELS
3480 Positi0n% = INSTR(Tamp_f0und$,:""") + 1
3490 Dummy! = VAL(MID$(Tamp_f0und$,Position%)
3500 EXIT TO,3700
3510 EXIT TO,3700
3520 REPEAT
3530 DO
3540 Position% = INSTR(Time_f0und$,:""") + 1
3550 Dummy! = VAL(MID$(Time_f0und$,Position%)
3560 END DO
3570 IF Dummy! <> Graph!(Graph1or2%,1,M%) THEN
3580 Mistake% = 2 : EXIT TO,60000
3590 EXIT TO,3700
3600 REPEAT
3610 REPEAT
3620 DO
3630 INPUT #File%,Data1$
3640 IF INSTR(Data1$,"END") <> 0 THEN EXIT 1 LEVELS
3650 REPEAT
3660 DO
3670 INPUT #File%,Temp_found$
3680 IF INSTR(Temp_found$,:"TEMP") <> 0 THEN EXIT 1 LEVELS
3690 REPEAT
3700 INPUT #File%,Time_found$
3710 EXIT 1 LEVELS
3720 IF Graph_ch0sen% = 1 THEN EXIT 1 LEVELS
3730 IF Graph_ch0san% = 2 THEN EXIT 1 LEVELS
3740 IF Minx! > Graph!(Graph1or2%,1,M%) THEN Minx! = Graph!(Graph1or2%,1,M%)
3750 IF Maxx! < Graph!(Graph1or2%,1,M%) THEN Maxx! = Graph!(Graph1or2%,1,M%)
3760 IF Miny! > Graph!(Graph1or2%,2,M%) THEN Miny! = Graph!(Graph1or2%,2,M%)
3770 IF Maxy! < Graph!(Graph1or2%,2,M%) THEN Maxy! = Graph!(Graph1or2%,2,M%)
IF Maxy! < Graph!(Graph1or2%,2,M%) THEN Maxy! 
= Graph!(Graph1or2%,2,M%)

IF Graph_chosen% = 8 THEN EXIT TO,500
END DO
NEXT M%
DO

IF Graph_chosen% = 4 OR Graph_chosen% = 5 THEN EXIT TO,100

IF Graph_chosen% < 9 OR Graph_chosen% > 10 THEN Max_graph_points%(Graph1or2%) = M% - 1 ELSE Max_graph_points%(Graph1or2%) = M%

IF Run_type% = 0 THEN EXIT 1 LEVELS DO

IF Graph_chosen% = 6 OR Graph_chosen% > 8 THEN EXIT 1 LEVELS

INPUT #File%,Data1$
END DO

IF INSTR(Data1$,"TEMP") = 0 THEN EXIT 1 LEVELS
ELS

Position% = INSTR(Data1$,"=") + 1
Temp_sought! = VAL(MID$(Data1$,Position%))
EXIT 4 LEVELS

REPEAT
DO

IF Graph_chosen% <> 2 THEN EXIT 1 LEVELS
ELS

Position% = INSTR(Data3$,"=") + 1
Temp_sought! = VAL(MID$(Data3$,Position%))
EXIT 4 LEVELS
REPEAT

IF Graph!(Graph1or2%,2,M%—1) = 0 THEN Error% = 1 : EXIT TO,50000

Error% = 1 : EXIT 3 LEVELS

REPEAT
DO

IF Graph_chosen% < 5 THEN EXIT 1 LEVELS
INPUT #File%,Data1$
END DO

IF INSTR(Data1$,"TIME") = 0 THEN EXIT 1 LEVEL
S

Position% = INSTR(Data1$,"=") + 1
Time_sought! = VAL(MID$(Data1$,Position%))
EXIT 3 LEVELS

REPEAT

IF Graph!(Graph1or2%,2,M%—1) = 0 THEN Error% = 1 : EXIT TO,50000
227

5410   Error% = 1 : EXIT 2 LEVELS
5420   REPEAT
5430   EXIT
10000  EXIT
50000  REM
50010  COLOR 7,0,0:SET CURSOR 24,6
50500  PRINT "The procedure DATARETRIEVAL is bombing. This is error ";ERR;" from line ";ERL
50510  STOP
50000  REM
60001  FOR M% = 2 TO 24
60002   LOCATE M%,1 : PRINT SPC(79)
60003   NEXT M%
60010   IF Graph_chosen% = 4 AND Graph!(Graph1or2%,2,1) < 0 THEN Error% = 0:Max_graph_points%(Graph1or2%) = Max_graph_points%(Graph1or2%) - 1 : EXIT
60015   IF Graph_chosen% = 5 AND Graph!(Graph1or2%,2,1) < 0 THEN Error% = 0:Max_graph_points%(Graph1or2%) = Max_graph_points%(Graph1or2%) - 1 : EXIT
60020   IF Graph_chosen% = 8 AND Graph!(Graph1or2%,2,2) < 0 THEN Error% = 0:Max_graph_points%(Graph1or2%) = Max_graph_points%(Graph1or2%) - 1
60030   IF Graph_chosen% \r 8 AND Graph_chosen% \r 11 AND Graph!(Graph1or2%,2,1) < 0 THEN Error% = 0
60040   IF Mistake% = 1 THEN GOTO 60500
60050   IF Mistake% = 2 THEN GOTO 60600
60060   IF Graph_chosen% = 8 AND Graph!(Graph1or2%,2,1) < 0 THEN Error% = 0 : EXIT
60070   IF Graph_chosen% = 6 AND Graph!(Graph1or2%,2,2) < 0 THEN Error% = 0 : EXIT
60100   FOR N% = 2 TO 23
60110   SET CURSOR 2,0 : PRINT SPC(80)
60120 NEXT N%
60130  COLOR 7,0 : SET CURSOR 13,6
60140  PRINT "The parameters you have set are not in the directory chosen. Please choose new parameters"
60150  Tyme! = TIMER + 5
60160  IF Tyme! = TIMER THEN GOTO 60150
60200 EXIT
60300 DO
60510  CLS : COLOR 2,0,0 : SET CURSOR 13,6
60520  PRINT "The magnet data does not line up correctly. This is a major bomb. Sorry"
60530 END
60540 DO
60510  CLS : COLOR 7,0,0 : SET CURSOR 13,6
60520  PRINT "The temperature does not line up from the Rho to Mu data. This is a major bomb. Sorry"
60630 END
60640 DO
PROCEDURE: DATAPRINT
INTEGER: N%
50 FOR N% = 1 TO 23
60 SET CURSOR N%,0 : PRINT SPC(80)
70 NEXT N%
100 COLOR 2,0:SET CURSOR 13,23
110 PRINT "The ";
120 COLOR 3,0:PRINT "DATA";
130 COLOR 2,0:PRINT " is now being loaded"
140 EXIT
END PROCEDURE

PROCEDURE: PARAMETERSET
EXTERNAL: Bias_sought!,Hold_sought!,Pulse_sought!,Mintyme!,Maxtyme!
INTEGER: M%,Row%
REAL: L
STRING: Check$[10],Dummy$[?],Dummy2$[?]
REAL: Min!,Max!:
REAL#: Max!:
38 REM * The numbers correspond to graph_type%
39 REM *
40 REM * 1 = C-t, G-t
41 REM * 2 = C-G-V
42 REM * 3 = C-G
43 REM * 4 = DLTS
44 REM *
100 Row% = 2:GOSUB 10000
110 COLOR 2,0,0
200 DO 1 TIMES
210 IF Graph_type% <> 1 AND Graph_type% <> 3 AND Graph_type% <> 4 THEN EXIT L LEVELS
220 SET CURSOR Row%,6
230 INPUT "Enter the BIAS Voltage";Dummy$
235 Bias_sought! = VAL(Dummy$)
240 Row% = Row% + 2
250 REPEAT
300 DO 1 TIMES
310 IF Graph_type% <> 1 AND Graph_type% <> 4 THEN EXIT 1 LEVELS
320 SET CURSOR Row%,6
330 INPUT "Enter the PULSE Voltage";Dummy$
335 Pulse_sought! = VAL(Dummy$)
340 Row% = Row% + 2
350 REPEAT
400 DO 1 TIMES
410 IF Graph_type% <> 1 AND Graph_type% <> 4 THEN EXIT 1 LEVELS
XIT 1 LEVELS

420  SET CURSOR Row%, 6
430  INPUT "Enter the HOLD Time in seconds"; Dummy$
435  Hold_sought! = VAL(Dummy$
440  Row% = Row% + 2
450  REPEAT
500  DO 1 TIMES
510  IF Graph_type% <> 4 THEN EXIT 1 LEVELS'
520  SET CURSOR Row%, 6
530  INPUT "Enter the box car times in seconds (min, max)"; Dummy$, Dummy2$
535  Mintyme! = VAL(Dummy$); Maxtyme! = VAL(Dummy2$
540  Row% = Row% + 2
550  REPEAT
8990  DO
9000  COLOR 7, 0, 0: SET CURSOR 22, 6
9010  INPUT "Do you need to change any of the above (Defaults to NO)"; Check$
9020  Check$ = UPPER$(Check$
9030  IF INSTR(Check$, "Y") <> 0 THEN EXIT TO, 100
9040  END DO
9050  EXIT
10000 FOR M% = Row% TO 23
10010  LOCATE M%, 1: PRINT SPC(79)
10020  NEXT M%
10030  RETURN
END PROCEDURE

PROCEDURE: PLOTTING_POINTS
EXTERNAL: Graph!(), Minx!, Maxx!, Miny!, Maxy!, Max_graph_points%(), Info_pointer%, Directory$
EXTERNAL: Temp$(), Bias$(), Pulse$(), Hold$(), Tyme$(), Rate$
(), Slope$(), X_int$
STRING: Line_type$[3], A$[3]
INTEGER: NZ, Y_pointer%
EXTERNAL: Y_int$(), Range$(), Correlation$(), GRANGE$
REAL: X_factor!, Y_factor!, X!
REAL: Y!
STRING: Dummy$[
EXTERNAL: Pen_selected%
INTEGER: M%
STRING: FUCK$[
REAL: Ymin!, Xmin!
EXTERNAL: G, SYMBOL$(), LINETYPE$(), PEN()
REAL ARRAY(?): LINETYPE
EXTERNAL: ONOFF, SPEED$

PROCEDURE: TIMEDELAY()
REAL ARG: Delay!/OPT=1!
END PROCEDURE
PROCEDURE: CLEARSCREEN()
   INTEGER ARG: Line%
END PROCEDURE

PROCEDURE: EDITDATA()
   STRING ARG: DUMMY$VAR
END PROCEDURE

PROCEDURE: TIMEDELAY
   REAL: Tyme!
      100 Tyme! = TIMER + Delay!
      110 IF Tyme! > TIMER THEN GOTO 110
      120 EXIT
END PROCEDURE

PROCEDURE: CLEARSCREEN
   INTEGER: NZ
      100 FOR NZ = Line% TO 23
      110 LOCATE NZ,1 : PRINT SPC(79)
      120 NEXT NZ
      130 EXIT
END PROCEDURE

PROCEDURE: EDITDATA
   REAL: DUMMY!
   INTEGER: Position%, Sign%, N%
   STRING: Saver$(14)
   REAL: Saver!
   INTEGER: Position1%
   STRING: A$(3)
   REAL ARRAY(?): Sng
      50 OPEN "CONVERT" FOR OUTPUT AS #1
      60 CLOSE #1
      80 IF INSTR(DUMMY$,"to") <> 0 THEN EXIT 1 LEVELS
      90 CLOSE #1
      100 OPEN "\CONVERT" FOR OUTPUT AS #1
      110 DO
         120 IF INSTR(DUMMY$"," ") = 0 THEN EXIT 1 LEVELS
         130 DUMMY! = VAL(DUMMY$)
         135 Sign% = SGN(DUMMY!)
         137 Position1% = INSTR(DUMMY$"," ")-1
         145 FOR N% = 1 TO 2
         160 DO
            185 IF Sign% = -1 THEN Position% = INSTR(DUM
               185 MY$,"-") ELSE Position% = 1
               200 Saver$ = MID$(DUMMY$,Position%,Position1
               %)
      210 PRINT #1 Saver$;
      220 IF N% = 1 THEN PRINT #1,", to ";
      230 Position% = Position1% + 2
Position1% = 20
END DO
DUMMY$ = MID$(DUMMY$, Position1%)
Sign% = SGN(VAL(DUMMY$))
NEXT N%
EXIT TO, 5250
REPEAT
DO
IF INSTR(DUMMY$, "X") = 0 THEN EXIT 1 LEVELS
PRINT #1, "X"
EXIT TO, 5250
END DO
DUMMY! = VAL(DUMMY$)
IF Sign% = -1 THEN DUMMY$ = " " + MID$(DUMMY$, INSTR(DUMMY$, ",") + 1)
IF INSTR(DUMMY$, "E") <> 0 THEN GOTO 5180
DO
IF DUMMY! = 0 THEN EXIT 1 LEVELS
PRINT #1, MID$(DUMMY$, INSTR(DUMMY$, "O"))
EXIT TO, 5250
REPEAT
IF DUMMY! > 10 OR DUMMY! < -10 THEN EXIT 1 LEVELS
IF Sign% = -1 THEN Saver$ = "-" + MID$(DUMMY$, 2, 4)
IF Sign% = 1 THEN Saver$ = MID$(DUMMY$, 2, 4)
PRINT #1, Saver$
EXIT TO, 5250
REPEAT
IF DUMMY! > 1000 OR DUMMY! < -1000 THEN EXIT 1 LEVELS
IF Sign% = -1 THEN Saver$ = "-" + MID$(DUMMY$, 2, 5)
IF Sign% = 1 THEN Saver$ = MID$(DUMMY$, 2, 5)
PRINT #1, Saver$
EXIT TO, 5250
REPEAT
DO
DUMMY! = VAL(DUMMY$)
Sign% = SGN(DUMMY!)
IF INSTR(DUMMY$, "E") <> 0 THEN EXIT TO, 5180
END DO
EXIT 1 LEVELS
DO
IF Sign% = -1 THEN Saver$ = "-" + MID$(STR$(ABS(DUMMY! - .00001)), 2)
IF Sign% = 1 THEN Saver$ = MID$(STR$(DUMMY! + .00001)), 2)
5080 PRINT #1 USING "\Saver$
5090 EXIT 2 LEVELS
5100 DO
5110 IF Sign% = -1 THEN Saver! = DUMMY! - 1
5110 IF Sign% = 1 THEN Saver! = DUMMY! + 1
5140 PRINT #1 USING ".###^^^^"Saver!
5150 EXIT 2 LEVELS
5160 REPEAT
5170 REPEAT
5180 DO
5190 Position% = INSTR(DUMMY$, "E")
5192 Position1% = 4
5193 IF Position% < 5 THEN Position1% = Position%
5200 IF Sign% = 1 THEN Saver$ = MID$(DUMMY$, 2, 4) + MID$(DUMMY$, Position1%)
5205 IF Sign% = -1 THEN Saver$ = "-" + MID$(DUMMY$, 2, Position1%) + MID$(DUMMY$, Position%)
5210 PRINT #1, Saver$
5220 CLOSE #1
5230 EXIT TO, 5250
5240 REPEAT
5250 CLOSE #1
5260 OPEN "\CONVERT" FOR INPUT AS #1
5265 CLEAR (DUMMY$)
5270 INPUT #1, DUMMY$
5280 CLOSE #1
5290 KILL "\CONVERT"
5300 EXIT
END PROCEDURE
70 CLOSE #3
80 OPEN "COM1 : 9600,N,8,1,RS,CS65535,DS,CD" FOR OUT
PUT AS #3
200 DO
210 X_factor! = (6634) / (Maxx! - Minx!)
220 Y_factor! = (5969) / (Miny! - Maxy!)
225 Ymin! = Miny!
226 Xmin! = Minx!
230 END DO
950 DO
955 CLEARSCREEN (2): SET CURSOR 0,70: COLOR 4,7:PRINT G
960 COLOR 2,0,0 : SET CURSOR 13,32
970 PRINT "PLOTTING POINTS"
980 END DO
1000 DO
1010 PRINT #3, "IN ; IP381,1016,6350,7650 ; "
1020 PRINT #3, "IN381,1016,6350,7650 ; "
1030 PRINT #3, "SP"PEN(G)" ; SI.2,.2 ; SM"SYMBL$(G)" ; LT"LINETYPE$(G)" ; DI 0,1 ;"
1040 END DO
233

1100 FOR NZ = 1 TO Max_graph_points% (Graph1 or 2%) STEP 1
1110    DO
1120        X! = Graph!(Graph1 or 2%, 1, NZ) * X_factor! + 10 γ: m
1130        Y! = +Graph!(Graph1 or 2%, 2, NZ) * Y_factor! + 63 (6 targets)
1132        IF X! > 32767 OR X! < -32768 THEN EXIT TO, 121
1133        IF Y! > 32767 OR Y! < -32768 THEN EXIT TO, 121
1140    END DO
1150    DO
1160        IF NZ < 1 THEN EXIT 1 LEVELS
1170        PRINT #3, "PA PU "Y!", "X!" ; PD ;"
1180    END DO
1190    PRINT #3, "PA "Y!", "X!" ; PD ;"
1200    IF VQL(SPEED$) = 0 THEN TIMEDELQY(1) ELSE TIMEDEL
AY(VAL(SPEED$))
1210 NEXT NZ
1220 IF ONOFF=0 THEN GOTO 4000
2000 DO
2010    Info_pointer% = Info_pointer% + 1
2020    IF Info_pointer% > 5 THEN EXIT 1 LEVELS
2030    Y_pointer% = 1762 + 1179 * (Info_pointer% - 1)
2040    PRINT #3, "IW ; SM ; SP"PEN(G)" ; SI .16,.2 ; DR 0,1 ;"
2050    DO
2060        PRINT #3, "PA PU 7754,"Y_pointer%" ; LB"SYMBOL$ \(G) ; CHR$(3)
2065        TIMEDELAY(5)
2070    END DO
2080    DO
2090        PRINT #3, "PA PU 7954,"Y_pointer%" ; LB"Direct ory$(Graph1 or 2%, 1);CHR$(3)
2095        TIMEDELAY(5)
2100    IF INSTR(Directory$(Graph1 or 2%, 2), "NO") = 0 THEN EXIT 1 LEVELS
2110    PRINT #3, "PA PU 7954,"Y_pointer%+480"; LB"Direct ory$(Graph1 or 2%, 2); CHR$(3)
2115    TIMEDELAY(5)
2120    END DO
2130    DO
2133        Dummy$ = " " + Temp$(Graph1 or 2%)
2135        EDITDATA(Dummy$)
2140    PRINT #3, "PA PU 8154,"Y_pointer%" ; LB"Dummy$ \(3 ; CHR$(3)
2145    TIMEDELAY(5)
2150    END DO
2151    DO
2152        EDITDATA(Tyme$(Graph1 or 2%))
Graph1or2% ; CHR$(3)
2154  PRINT #3, "pa pu 8354, "Y_pointer" ; LB"Tyme$(
2155  END DO
2160  DO
2165  EDITDATA(Bias$(Graph1or2%))
2170  PRINT #3, "PA PU 8554, "Y_pointer" ; LB"Bias$(
2175  END DO
2180  PRINT #3, "PA PU 8754, "Y_pointer" ; LB"Pulse$(
2200  END DO
2205  END DO
2225  EDITDATA(Hold$(Graph1or2%))
2230  PRINT #3, "PA PU 8954, "Y_pointer" ; LB"Hold$(
2235  END DO
2255  EDITDATA(Rate$(Graph1or2%))
2260  PRINT #3, "PA PU 9154, "Y_pointer" ; LB"Rate$(
2265  END DO
2272  EDITDATA(Grange$(Graph1or2%))
2273  PRINT #3, "PA PU 9354, "Y_pointer" ; LB"Grange$(
2274  END DO
2285  EDITDATA(Slope$(Graph1or2%))
2290  PRINT #3, "PA PU 9554, "Y_pointer" ; LB"Slope$(
2300  END DO
2315  EDITDATA(X_int$(Graph1or2%))
2320  PRINT #3, "PA PU 9754, "Y_pointer" ; LB"X_int$(
2325  END DO
2345  EDITDATA(Y_int$(Graph1or2%))
2350  PRINT #3, "PA PU 9954, "Y_pointer" ; LB"Y_int$(
2355  END DO
2360 END DO
2370 DO
2375 EDITDATA(Range$(Graph1or2%))
2380 PRINT #3,"PA PU 10154,"Y_pointer%" ; LB"Range $(Graph1or2%) ; CHR$(3)
2385 TIMEDELAY(5)
2390 END DO
2400 DO
2405 EDITDATA(Correlation$(Graph1or2%))
2410 PRINT #3,"PA PU 10354,"Y_pointer%" ; LB"Correlation$(Graph1or2%) ; CHR$(3)
2415 TIMEDELAY(5)
2420 END DO
3000 END DO
3001 LOCATE NZ,1 : PRINT SPC(79)
3002 NEXT NZ
3003 RETURN
END PROCEDURE

PROCEDURE: GETDATA
EXTERNAL: Graph!(),Minx!,Maxx!,Miny!,Maxy!,Max_graph_points%
INTEGER: M%,N%
REAL: Dummy,Dummy!
EXTERNAL: G1,G2
100 DO
110 IF G1=1 AND Graph%() <> 1 THEN EXIT 1 LEVELS
120 Minx! = 1E+29
130 Maxx! = -1E+29
140 Miny! = 1E+29
150 Maxy! = -1E+29
160 END DO
500 OPEN "\BUCKET" FOR INPUT AS #3
510 FOR M% = 1 TO Max_graph_points%(Graph%)
520 INPUT #3,Graph!(Graph%,1,M%), Dummy!, Graph!(Graph%,2,M%)
530 IF Minx! > Graph!(Graph%,1,M%) THEN Minx! = Graph!(Graph%,1,M%)
540 IF Maxx! < Graph!(Graph%,1,M%) THEN Maxx! = Graph!(Graph%,1,M%)
550 IF Miny! > Graph!(Graph%,2,M%) THEN Miny! = Graph!(Graph%,2,M%)
560 IF Maxy! < Graph!(Graph%,2,M%) THEN Maxy! = Graph!(Graph%,2,M%)
570 NEXT N%
1000 KILL "\BUCKET"
END PROCEDURE
PROCEDURE: PLOTTERAXES
EXTERNAL: Minx!, Miny!, Maxx!, Maxy!, Xtitle$, Ytitle$
STRING: Grid$[7], Name$[80], Dummy$[16]
REAL: Dummy!, Number!
INTEGER: M%, Maxloop%, Y_PTRINTER%, X_PTRINTER%, N%, Mini_pads%
EXTERNAL: ONOFF, Info_pointer%

PROCEDURE: TIMEDELAY()
REAL ARG: Delay!/OPT=10!
END PROCEDURE

PROCEDURE: CLEARSCREEN()
INTEGER ARG: Line%
END PROCEDURE

PROCEDURE: EDITDATA()
REAL ARG: DUMMY!
STRING ARG: DUMMY$/VAR
END PROCEDURE

PROCEDURE: TIMEDELAY
REAL: Tyme!
100 Tyme! = TIMER + Delay!
110 IF Tyme! > TIMER THEN GOTO 110
120 EXIT
END PROCEDURE

PROCEDURE: CLEARSCREEN
INTEGER: N%
100 FOR N% = Line% TO 24
110 LOCATE N%,1 : PRINT SPC(79)
120 NEXT N%
130 EXIT
END PROCEDURE

PROCEDURE: EDITDATA
REAL: SAVER!
INTEGER: Sign%
STRING: Saver$[?]
INTEGER: Position%, Position1%
EXTERNAL: Mini_pads%
REAL: DUMY!
1 ON ERROR GOTO 1320
100 CLOSE #1
110 OPEN "\CONVERT" FOR OUTPUT AS #1
111 DO
112 IF DUMMY! (.999 AND DUMMY! > -.999 THEN EXIT
113 IF DUMMY! ) 10000 OR DUMMY! < -10000 THEN EX
IT 1 LEVELS
   117 IF ABS(CINT(DUMMY!)-DUMMY!)<.0009 THEN DUMMY! =CINT(DUMMY!):DUMMY$=STR$(DUMMY!):EXIT TO,130
   118 END DO
   120 DUMMY$ = STR$(DUMMY!)
   130 Sign% = SGN(DUMMY!)
   135 DO
      136 IF Sign% <> -1 THEN EXIT 1 LEVELS
      137 DUMMY$ = MID$(DUMMY$,INSTR(DUMMY$,"-"))
   138 END DO
   200 DO
      205 IF Mini_pads% = 1 THEN EXIT 1 LEVELS
      210 IF DUMMY! > .000999 OR DUMMY! < -.000999 THEN EXIT 1 LEVELS
      220 PRINT #1 USING ";.##^"""" VAL(DUMMY$)
   230 CLOSE #1
   240 EXIT TO,1250
   250 END DO
   500 DO
      510 IF DUMMY! > 10 OR DUMMY! < -10 THEN EXIT 1 LEVELS
      515 IF INSTR(DUMMY$,".0000") <> 0 THEN DUMMY$ = "0"
      520 Saver$ = MID$(DUMMY$,1,6)
      530 PRINT #1,Saver$
      540 EXIT TO,1250
      550 REPEAT
      560 DO
         570 IF DUMMY! > 1000 OR DUMMY! < -1000 THEN EXIT 1 LEVELS
         580 Saver$ = MID$(DUMMY$,1,6)
         590 PRINT #1,Saver$
         600 EXIT TO,1250
         610 REPEAT
         620 DO
            630 IF DUMMY! > 9999.99 OR DUMMY! < -9999.99 THEN EXIT 1 LEVELS
            640 Saver$ = MID$(DUMMY$,1,6)
            650 PRINT #1,Saver$
            660 EXIT TO,1250
            670 REPEAT
   1000 DO
      1010 Position% = INSTR(DUMMY$,"E")
      1020 IF Position% = 0 THEN EXIT 1 LEVELS
      1025 DO
         1030 IF Position% <= 5 THEN EXIT 1 LEVELS
         1040 IF Sign% = -1 THEN Saver$ = MID$(DUMMY$,IN
            STR(DUMMY$,"-"),5) + MID$(DUMMY$,Position%)
            ELSE Saver$ = MID$(DUMMY$,1,5) + MID$(DUMMY$,Position%)
         1050 PRINT #1,Saver$
         1060 EXIT TO,1250
1070 REPEAT
1080 PRINT #1 USING "#.##•••••••••••••••••" DUMMY!
1090 EXIT TO,1250
1100 REPEAT
1200 DO
1210 SAVER! = VAL(DUMMY$)
1220 PRINT #1 USING "#.##•••••••••••••••••" SAVER!
1230 EXIT TO,1250
1240 REPEAT
1250 CLOSE #1
1260 OPEN "\CONVERT" FOR INPUT AS #1
1270 INPUT #1,DUMMY$
1280 CLOSE #1
1290 KILL "\CONVERT"
1300 IF INSTR(DUMMY$,"E+00") (> 0 THEN DUMMY$ = "0"
1310 EXIT
1320 PRINT "Error number "ERR" from line "ERL
1330 STOP

END PROCEDURE

90 Info_pointer%=O
100 CLEARSCREEN(2)
110 COLOR 2,0,0 : SET CURSOR 13,6
120 INPUT "Do you want the grid plotted (Defaults to NO)";Grid$
130 Grid$ = UPPER$(Grid$)
140 CLEARSCREEN(12)
150 COLOR 3,0,0 : SET CURSOR 13,36
160 PRINT "PLOTTING"
190 CLOSE #3
200 DO
210 OPEN "COM1 : 9600,N,8,1,RS,CS65535,DS,CD" FOR OUTPUT AS #3
220 PRINT #3,"IN ; IP 381,1016,6350,7650 ; "
230 END DO
300 DO
310 PRINT #3, "SP1 ; PA PU 381,1016 PD 381,7650,6350,7650,6350,1016,381,1016"
320 PRINT #3, " ; PU ;"
330 TIMEDELAY(5)
340 END DO
400 DO
410 Y_POINTER% = CINT(1016 + (6634 - (LEN(Xtitle$)*172))/2)
420 PRINT #3, "SP2 ; SI.3,.3 ; DR 0,1 ;"
430 PRINT #3, "PA PU 6858,"Y_POINTER%" ; LB"Xtitle$ ;CHR$(3)
440 TIMEDELAY(7)
450 END DO
500 DO
510 X_POINTER% = CINT(6340 - (5969 - (LEN(Ytitle$)*172))/2)
520  PRINT #3,"SP2 ; SI.3,.3 ; DR-1,0 ; "
530  PRINT #3,"PA PU "X_POINTER%",170 ; LB"Ytitle$;C
540  TIMEDELY()
550  END DO
560  DO
570  Y_POINTER% = CINT(1016 + (6634 - (LEN(Title$) * 179)) / 2)
580  PRINT #3, " SP1 ; SI.3,.3 ; DRO,1 ;"
590  PRINT #3, "PA PU 254,"Y_POINTER%;C
600  TIMEDELY()
610  END DO
620  DD
630  Y_PDINTERZ = CINT(1016 + (663 - (LEN(Title$) * 179)) / 2)
640  PRINT #3, "SP1 ; TL.9 ;"
650  FOR NZ = 1 TO 9
660  X_POINTER% = CINT(381 + NZ * 597)
670  PRINT #3, "PA PU "X_POINTER%;970 ; XT ;"
680  TIMEDELY(1)
690  NEXT NZ
700  FOR NZ = 1 TO 9
710  Y_POINTER% = CINT(1016 + NZ * 663)
720  PRINT #3, "PA PU 6350, "Y_POINTER%; YT;"
730  TIMEDELY(1)
740  NEXT NZ
750  END DO
760  IF INSTR(Grid$,"Y") = 0 THEN EXIT 1 LEVELS
770  IF NZ = 1 TO 9
780  PRINT #3, "LT1,0.5 ; SP2 ;"
790  FOR NZ = 1 TO 9
800  X_POINTER% = CINT(381 + NZ * 597)
810  PRINT #3, "PA PU "X_POINTER%;970 ; XT ;"
820  TIMEDELY(1)
830  NEXT NZ
840  PRINT #3, "PA PU 381,"Y_POINTER%; PD 6350,"Y_ POINTER%" ;"
850  TIMEDELY(25)
860  NEXT NZ
870  EXIT 1 LEVELS
880  END DO
890  IF NZ = 10 THEN Y_POINTER% = CINT(7585 - LEN(Dummy$) * 72)
1270 PRINT #3, "PA PU 6530," "Y_POINTER"\$" ; LB"Dummy$ ; CHR$(3)
1280 TIMEDELY (2)
1290 NEXT NZ
1300 END DO
1400 DO
1410 PRINT #3, "SP2 ; SI.2,1 ; DIO,1 ;"  
1420 FOR NZ = 0 TO 10 STEP 2
1425 IF Miny! > .00099 OR Miny! < -.00099 THEN Min
i_pads% = 1 ELSE Mini_pads% = 0
1430 Number! = Miny! + (Maxy! - Miny!) * (NZ/10)
1440 EDITDATA (Number!, Dummy$)
1450 X_POINTER% = CINT (6355 - NZ*597)
1460 Y_POINTER% = CINT (584 - LEN (Dummy$) * 36)
1470 PRINT #3, "PA PU "X_POINTER%", "Y_POINTER%" LB 
"Dummy$; CHR$(3)
1480 TIMEDELY (2)
1490 NEXT NZ
1500 END DO
1510 IF ONOFF=0 THEN GOTO 1710
1600 DO
1610 PRINT #3, "PU ; SI.16,.2 ; DRO,1 ; SP2 ;"
1620 RESTORE, 60100
1630 READ Maxloop%  
1640 FOR NZ = 1 TO Maxloop%
1650 READ Name$
1660 X_POINTER% = CINT (7554 + NZ*200)
1670 PRINT #3, "PA PU "X_POINTER%", 508 ; LB"Name$ ;
CHR$(3)
1680 TIMEDELY (4)
1690 NEXT NZ
1700 END DO
1710 PRINT #3, "SPO;"
1720 CLOSE #3
1730 EXIT
60000 STOP
60100 DATA 14
60110 DATA "SYMBOL", "DIRECTORIES", "TEMPERATURE", "TIME", "BIAS", "PULSE HIGH"
60120 DATA "HOLD TIME", "RATE WINDOW", "GRAPH RANGE", "SLO PE", "X-INTERCEPT"
60130 DATA "Y-INTERCEPT", "RANGE", "CORRELATION"
END PROCEDURE

PROCEDURE: SAVEDATA
EXTERNAL: Graph!(), Max_graph_points%(), Graph_name$(>, Xt itle$, Ytitle$, Pulse$(), Directory$(), Temp$()  
EXTERNAL: Tyme$(), Bias$(), Hold$(), Rate$(), Grange$(), Slo pe$(), X_int$(), Y_int$()
EXTERNAL: Range$()  
STRING: Save_file$[15]
INTEGER: Line%, Choice%, Position%, N%
EXTERNAL: Graph_chosen%, Menu_chosen%, Slope(), Y_int!, X_int!()
EXTERNAL: Correlation(), Min!, Max!, Correlation$(()), Y_int!(), Min()!, Max()!
INTEGER: M%
REAL ARRAY(?): RRange!
STRING: CHOICE$[2]
EXTERNAL: GFLAG, Saveflag

1 ON ERROR GOTO 60000
100 DO
110 COLOR 6, 0 : SET CURSOR 20, 6
120 PRINT "Enter the driver and name you want to save the file under"
130 SET CURSOR 21, 6 : PRINT "(Defaults to B:"; Graph_name$(Menu_chosen%, Graph_chosen%)") ";
140 INPUT Save_file$
150 IF Save_file$ = "" THEN Save_file$ = "B:" + Graph_name$(Menu_chosen%, Graph_chosen%)
160 END DO
170 Line% = 19 : GOSUB 20000
175 Choice% = Graph%
178 IF Saveflag=1 THEN Saveflag=0: GOTO 300
180 DO
200 SET CURSOR 20, 6 : COLOR 2, 0 : INPUT "Which do you want to save, (1) G1 or (2) G2 graph"; CHOICE$
202 Choice% = VAL(CHOICE$) : GFLAG = Choice%
203 SET CURSOR 0, 70: COLOR 2, 7: PRINT Choice%; COLOR 7, 0
210 IF Choice% (1 OR Choice% > 2 THEN EXIT TO, 170
220 EXIT 1 LEVELS
230 REPEAT
300 Line% = 19 : GOSUB 20000
304 KILL Save_file$
305 DO
310 Position% = LEN(Save_file$)
320 SET CURSOR 17, 10 : COLOR 5, 0 : PRINT "SAVING FILE UNDER 
330 SET CURSOR 17, 28 : COLOR 5, 0 : PRINT MID$(Save_file$, 1, Position%)
335 CLOSE
340 OPEN Save_file$ FOR OUTPUT AS #1
350 PRINT #1, Graph_name$(Menu_chosen%, Graph_chosen%)
360 PRINT #1, Xtitle$
370 PRINT #1, Ytitle$
380 PRINT #1, Directory$(Choice%, 1) "," Directory$(Choice%, 2)
390 PRINT #1, Temp$(Choice%)
400 PRINT #1, Tyme$(Choice%)
410 PRINT #1, Bias$(Choice%)
242 PRINT #1, Pulse$(Choice%)
430 PRINT #1, Hold$(Choice%)
440 PRINT #1, Rate$(Choice%)
450 PRINT #1, Grange$(Choice%)
460 PRINT #1, Slope!(Choice%)
470 PRINT #1, X_int!(Choice%)
480 PRINT #1, Y_int!(Choice%)
490 PRINT #1, Correlation!(Choice%)
495 PRINT #1, Min!(Choice%)
496 PRINT #1, Max!(Choice%)
500 PRINT #1, Max_graph_points!(Choice%)
510 FOR N% = 1 TO Max_graph_points!(Choice%)
520 PRINT #1, Graph!(Choice%, 1, N%)", " Graph!(Choice%
530 NEXT N%
540 END DO
600 Line% = 17 : GOSUB 20000
610 CLOSE #1
1000 EXIT
20000 FOR N% = Line% TO 24
20010 LOCATE N%, 1 : PRINT SPC(79)
20020 NEXT N%
20030 RETURN
60000 IF ERR = 1001 AND ERL = 304 THEN RESUME NEXT
60010 IF ERR > 999 THEN GOSUB 63000
62000 CLS : COLOR 7, 0 : SET CURSOR 13, 1
62010 PRINT "Sorry the procedure SAVEDATA is bombing.
This is error "ERR" from line "ERL"
62020 STOP
62030 Line% = 17 : GOSUB 20000
62040 NEXT N%
62050 EXIT
END PROCEDURE

PROCEDURE: LOADDATA
EXTERNAL: Graph!(.), Max_graph_points!(.), Grange_name$(.), Xti
itle$, Ytitle$, Pulse$(.), Directory$(.), Temp$(.)
EXTERNAL: Tyme$(.), Bias$(.), Hold$(.), Rate$(.), Grange$(.), Slo
e$(.), X_int$(.), Y_int$(.)
EXTERNAL: Range$(.)
INTEGER: Line%, Choice%, Position%, N%
STRING: Load_file$[15], Xdata$[18], Ydata$[18]
EXTERNAL: Correlation$(.), Min!(.), Max!(.)
EXTERNAL: Slope!(.), X_int!(.), Y_int!(.), Correlation!(.), Cle
ARGRAPH, G1, G2, COMP_GRAPH%
INTEGER: CHOICE%
1 ON ERROR GOSUB 60000
100 DO
COLOR 6,0 : SET CURSOR 20,6
PRINT "Enter the file name and source drive to be loaded"
SET CURSOR 21,6 : PRINT "(Defaults to B:";Graph_name$(1,1)"");
INPUT Load_file$
IF Load_file$ = "" THEN Load_file$ = "B:" + Graph_name$(1,1)
END DO
Line% = 16 : GOSUB 20000
Choice% = Graph%
Line% = 16 : GOSUB 20000
DO
Position% = LEN(Load_file$)
SET CURSOR 17,10 : COLOR 5,0 : PRINT "LOADING FILE"
SET CURSOR 17,23 : COLOR 5,0 : PRINT MID$(Load_file$,1,Position%) 
OPEN Load_file$ FOR INPUT AS #1
INPUT #1,Graph_name$(1,1)
INPUT #1,Xtitle$
INPUT #1,Ytitle$
INPUT #1,Directory$(Choice%,1), Directory$(Choice%,2)
INPUT #1,Temp$(Choice%) 
INPUT #1,Tyme$(Choice%) 
INPUT #1,Bias$(Choice%) 
INPUT #1,Pulse$(Choice%) 
INPUT #1,Hold$(Choice%) 
INPUT #1,Rate$(Choice%) 
DO
   INPUT #1,Xdata$
   IF INSTR(Xdata$,"X") <> 0 THEN EXIT 1 LEVELS
   INPUT #1,Ydata$
   Grange$(Choice%) = Xdata$ + "," + Ydata$
END DO
INPUT #1,Slope!(Choice%) 
INPUT #1,X_int!(Choice%) 
INPUT #1,Y_int!(Choice%) 
INPUT #1,Correlation!(Choice%) 
INPUT #1,Min!(Choice%) 
INPUT #1,Max!(Choice%) 
INPUT #1,Max_graph_points%(Choice%) 
FOR N% = 1 TO Max_graph_points%(Choice%) 
   INPUT #1,Xdata$,Ydata$
   Graph!(Choice%,1,N%) = VAL(Xdata$)
   Graph!(Choice%,2,N%) = VAL(Ydata$)
   NEXT N%
END DO
Line% = 18 : GOSUB 20000
CLOSE #1
1000 EXIT
20000 FOR N% = Line% TO 24
20010 LOCATE N%, 1 : PRINT SPC(79)
20020 NEXT N%
20030 RETURN
59999 EXIT
60000 IF ERR > 999 THEN GOSUB 61000: RESUME, 59999
60005 Line% = 2 : GOSUB 20000
60010 SET CURSOR 13, 6
60020 PRINT "The procedure LOADDATA is bombing. This is error "ERR" from line "ERL"
60030 STOP
60040 Line% = 19: GOSUB 20000
60050 SET CURSOR 19, 6 : PRINT "There is a file loading error. Check your files"
60060 FOR NZ = -30000 TO 30000 : NEXT NZ
60070 Line% = 13: GOSUB 20000
60080 RETURN
END PROCEDURE

PROCEDURE: INVERSEPOWER
EXTERNQL: MINX!, MQXX!, MINY!, MQXY!, GRQPH!(), MQX_GRQPH_POINTS()
INTEGER: M%, Minloop%
10 ON ERROR GOTO 50000
20 Minloop% = 1
84 DO
85 IF Xory% = 2 THEN EXIT 1 LEVELS
90 MINX! = 1.0E+37: MAXX! = -1.0E+37
91 END DO
92 DO
93 IF Xory% = 1 THEN EXIT 1 LEVELS
98 MINY! = 1.0E+37: MAXY! = -1.0E+37
99 END DO
100 FOR M% = Minloop% TO MAX_GRAPH_POINTS%(Graph1or2%)
110 GRAPH!(Graph1or2%, Xory%, M%) = 1 / (GRAPH!(Graph1or2%, Xory%, M%) * Amount!)^Power!
120 DO
130 DO
140 IF Xory% <> 1 THEN EXIT 1 LEVELS
150 IF MINX! > GRAPH!(Graph1or2%, 1, M%) THEN MIN X! = GRAPH!(Graph1or2%, 1, M%)
160 IF MAXX! < GRAPH!(Graph1or2%, 1, M%) THEN MAX X! = GRAPH!(Graph1or2%, 1, M%)
170 EXIT 2 LEVELS
180 END DO
190 DO
200 IF MINY! > GRAPH!(Graph1or2%, 2, M%) THEN MIN Y! = GRAPH!(Graph1or2%, 2, M%)
210 IF MAXY! < GRAPH!(Graph1or2%, 2, M%) THEN MAX Y!
Y! = GRAPH!(Graph1or2%,2,M%)
220     END DO
230     END DO
240    NEXT M%
250    EXIT
50000 DO
50010 IF ERR = 2 AND ERL = 110 THEN Minloop% = M% + 1
50020 EXIT TO 100
51000 PRINT "The procedure INVERSEPOWER is bombing.
This is error "ERR" from line "ERL"."
51010 END
60000 END DO
END PROCEDURE

PROCEDURE: MAINKEY
REAL: I,L,N
STRING ARRAY(10)[16]: KEYFUNC$,FUNC$
REAL: 0
EXTERNAL: G1,G2,Clearscreen,TITLE,GRAPH_NAME$(())
EXTERNAL: MENU_CHOSEN%,GRAPH_CHOSEN%,GRAPH%,GFLAG
10 Clearscreen(2):STATUSLINE OFF
14 SET CURSOR 0,70:COLOR 2,7:PRINT GRAPH%:COLOR 7,0
15 IF MENU_CHOSEN%=0 OR GRAPH_CHOSEN%=0 THEN GOTO 20
ELSE TITLE(GRAPH_NAME$(MENU_CHOSEN%,GRAPH_CHOSEN%)):SET CURSOR 0,70:COLOR 2,7:PRINT GFLQG:COLOR 7,0
20 FOR I=1 TO 10
30 IF COS(I*3.14159)<0 THEN L=2*I:N=1 ELSE N=14
40 READ KEYFUNC$(I)
50 READ FUNC$(I)
60 KEY I,FUNC$(I)+CHR$(13)
70 SET CURSOR L,N:COLOR 2,0:PRINT "F";I;" ":COLOR 2,7:SET CURSOR L,N+4:PRINT KEYFUNC$(I)
80 NEXT I
90 COLOR 2,0
95 STOP
100 DATA "SYSTEM ",SYSTEM,"SAVEDAT",EXIT TO 8555,"MODIFY ",MODIFY,"PLOTTER",EXIT TO 9000,"MENU ",EXIT TO 200,
"LEAST ",EXIT TO 6300,"LOAD G1",EXIT TO 1429,"LOAD G2",EXIT TO 8700,"SCALE ",EXIT TO 3000,"SCRNPLT",EXIT TO 4000
END PROCEDURE

PROCEDURE: PLOTTER
REAL: I,L,N
STRING ARRAY(10)[16]: KEYFUNC$,FUNC$
EXTERNAL: Onoff,Clearscreen,MENU_CHOSEN%,GRAPH_CHOSEN%,
G1
EXTERNAL: G2,TITLE,GRAPH_NAME$(()),G
10 Clearscreen(2)
15 IF MENU_CHOSEN%=0 OR GRAPH_CHOSEN%=0 THEN GOTO 20
ELSE TITLE(GRAPH_NAME$(MENU_CHOSEN%,GRAPH_CHOSEN%)):SET CURSOR 0,70:COLOR 4,7:PRINT G:COLOR 7,0
PROCEDURE: MODIFY
REAL: I, L, N
STRING ARRAY (10) [16]: KEYFUNC$ EXTERNAL: Clearscreen, MENU_CHOSEN%, GRAPH_CHOSEN%, G1, G2, TITLE EXTERNAL: GRAPH_NAME$( ) REAL: MD EXTERNAL: MOD
10 Clearscreen (2)
15 IF MENU_CHOSEN% = 0 OR GRAPH_CHOSEN% = 0 THEN GOTO 20 ELSE TITLE (GRAPH_NAME$(MENU_CHOSEN%, GRAPH_CHOSEN%)): SET CURSOR 0, 70: COLOR 1, 7: PRINT MOD: COLOR 7, 0
20 FOR I = 1 TO 10
30 IF COS (I * 3.14159) < 0 THEN L = 2 * I; N = 1 ELSE N = 14
40 READ KEYFUNC$ (I)
50 KEY I, KEYFUNC$ (I) + CHR$(13)
60 SET CURSOR L, N: COLOR 1, 0: PRINT "F"; I; "": COLOR 1, 7: SET CURSOR L, N + 4: PRINT KEYFUNC$ (I)
70 NEXT I
80 COLOR 1, 0
90 STOP
100 DATA "MAINKEY", "MAINKEY", "SPEED ", SPEED, "LINES ", LINES, "SYMBOLS", SYMBOLS, "COLORS ", COLORS, "PLOTXTS", EXIT TO 9570, "PLOT G1", EXIT TO 9640, "PLOT G2", EXIT TO 9220, "ON/OFF ", INFO, "CAPTION ", CAPTION
END PROCEDURE

PROCEDURE: COLORS
EXTERNAL: PLOTTER
REAL: I, J
STRING: A$[16]
REAL: N
INTEGER ARRAY (5): EXIST EXTERNAL: Pen_selected%
REAL ARRAY(?): PN
EXTERNAL: PEN(), G1, G2, Clearscreen
STRING: N$[?], J$[?]
10 EXIST(1)=G1:EXIST(2)=G2
11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
20 Clearscreen(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EXISTING GRAPHS:"
30 FOR I=I TO 4
40 IF EXIST(I)=0 THEN GOTO 60
50 IF PEN(I)<>0 THEN SET CURSOR 5,2*I+16:COLOR PEN(I)-2,7:PRINT EXIST(I) ELSE SET CURSOR 5,2*I+16:COLOR 0,7:PRINT EXIST(I)
60 NEXT I
70 SET CURSOR 10,1:COLOR 7,0:PRINT "COLORS:"
80 FOR I=1 TO 5
90 SET CURSOR 10,2*I+8:COLOR I-1,7:PRINT I+1
100 NEXT I
110 SET CURSOR 15,1:COLOR 7,0:INPUT "WHICH GRAPH" N$
111 N=VAL(N$)
120 SET CURSOR 17,1:INPUT"WHICH COLOR" J$
121 J=VAL(J$)
125 IF N<1 OR N>2 THEN GOTO 135
130 PEN(N)=J
131 FOR I=1 TO 4
132 IF EXIST(I)=0 THEN GOTO 134
133 IF PEN(I)<>0 THEN SET CURSOR 5,2*I+18:COLOR PEN(I)-2,7:PRINT EXIST(I) ELSE SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
134 NEXT I
135 COLOR 7,0
140 SET CURSOR 20,1:INPUT"FINISHED CHOOSING COLORS"; A$
150 IF INSTR(UPPER$(A$), "Y") THEN GOTO 160 ELSE GOTO 20
160 PLOTTER
170 STOP
END PROCEDURE

PROCEDURE: SYMBOLS
INTEGER ARRAY(10): EXIST, LINES
EXTERNAL: PLOTTER
REAL: I, J, N
STRING: A$[16]
REAL ARRAY(10): SYMBOLS
EXTERNAL: Symbol$, Symbol$( )
STRING ARRAY(5)[16]: Symbol$
EXTERNAL: G1, G2, Clearscreen
STRING: N$[?], J$[?]
10 EXIST(1)=G1:EXIST(2)=G2
11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
20 Clearscreen(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EXIST
ISTING GRAPHS:
30 FOR I=1 TO 5
40 IF EXIST(I)=0 THEN GOTO 60 ELSE SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
50 IF SYMBOLS(I)<>0 THEN SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
60 NEXT I
70 SET CURSOR 10,1:COLOR 7,0:PRINT "SYMBOLS: ":SET CURSOR 10,10:COLOR 13,0:PRINT " 1)NONE":SET CURSOR 10,23:COLOR 13,0:PRINT " 2) + ":SET CURSOR 10,28:COLOR 13,0:PRINT " 3) * ":SET CURSOR 10,33:COLOR 13,0:PRINT " 4) @ "
80 SET CURSOR 15,1:COLOR 7,0:INPUT "WHICH GRAPH" N$
81 N=VAL(N$)
85 SET CURSOR 17,1:INPUT "WHICH SYMBOL" J$
86 J=VAL(J$)
90 IF N<1 OR N>2 THEN GOTO 105
91 IF J=1 THEN SYMBOLS(N)=" "
92 IF J=2 THEN SYMBOLS(N)=" + "
93 IF J=3 THEN SYMBOLS(N)=" * "
94 IF J=4 THEN SYMBOLS(N)=" @ "
100 SYMBOLS(N)=J
101 FOR I=1 TO 5
102 IF EXIST(I)=0 THEN GOTO 104 ELSE SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
103 IF SYMBOLS(I)<>0 THEN SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
104 NEXT I
105 COLOR 7,0
110 SET CURSOR 20,1:INPUT "FINISHED CHOOSING SYMBOLS";
A$
120 IF INSTR(UPPER$(A$),"Y") THEN GOTO 130 ELSE GOTO 20
130 PLOTTER
140 STOP
END PROCEDURE

PROCEDURE: LINES
INTEGER ARRAY(10): EXIST, LINES
EXTERNAL: PLOTTER
REAL: I, J, N
STRING: A$[16]
EXTERNAL: Line_type$, LINETYPE$( ), G1, G2, CLEARSCREEN
STRING: N$[?], J$[?]
10 EXIST(1)=G1:EXIST(2)=G2
11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EXISTING GRAPHS:"
30 FOR I=1 TO 5
40 IF EXIST(I)=0 THEN GOTO 60 ELSE SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
50 IF LINES(I)<>0 THEN SET CURSOR 5,2*I+18:COLOR L
INES(I)+1,7:PRINT EXIST(I)
60 NEXT I
70 SET CURSOR 10,1:COLOR 7,0:PRINT "LINES: ":SET CURSOR 10,7:COLOR 2,0:PRINT "1)NONE":SET CURSOR 10,14:COLOR 3,0:PRINT "2)----":SET CURSOR 10,21:COLOR 4,0:PRINT "3)... ":SET CURSOR 10,28:COLOR 5,0:PRINT "4)---"
80 SET CURSOR 15,1:COLOR 7,0:INPUT"WHICH GRAPH" N$
81 N=VAL(N$
85 SET CURSOR 17,1:INPUT"WHICH LINE" J$
86 J=VAL(J$
90 IF N<1 OR N>2 THEN GOTO 105
91 IF J=1 THEN LINETYPES(N)="0"
92 IF J=2 THEN LINETYPES(N)=" 
93 IF J=3 THEN LINETYPES(N)="1"
94 IF J=4 THEN LINETYPES(N)="2"
100 LINES(N)=J
101 FOR I=1 TO 5
102 IF EXIST(I)=0 THEN GOTO 104 ELSE SET CURSOR 5,2 I+18:COLOR INES(I)+1,7:PRINT EXIST(I)
103 IF LINES(I)<0 THEN SET CURSOR 5,2*I+18:COLOR LINES(I)+1,7:PRINT EXIST(I)
104 NEXT I
105 COLOR 7,0
110 SET CURSOR 20,1:INPUT"FINISHED CHOOSING LINES";A$
120 IF INSTR(UPPERS(A$),"Y") THEN GOTO 130 ELSE GOTO 20
130 PLOTTER
140 STOP
END PROCEDURE

PROCEDURE: Info
EXTERNAL: Onoff,Plotter
10 IF Onoff=0 THEN Onoff=1 ELSE Onoff=0
20 Plotter
30 STOP
END PROCEDURE

PROCEDURE: CAPTION
STRING: CPTION$(80)
INTEGER: YPOINT
EXTERNAL: PLOTTER,TIMEDELAY
REAL: HELLO
5 WIDTH 80:CLS
10 SET CURSOR 15,5:COLOR 7,0:PRINT "ENTER THE PLOT CAPTION BELOW(80 CHARACTERS OR LESS)"
20 SET CURSOR 18,1:COLOR 0,7:INPUT"";CPTION$
30 CLOSE #3
40 OPEN "COM1 : 9600,N,8,1,RS,CS65535,DS,CD" FOR OUT
PUT AS #3
50 PRINT #3,"IN; "
70 PRINT #3, "SP2 ; SI.15,.25 ; DR 0,1 ;"
80 PRINT #3, "PA PU 7200,900 ; LB"CPTIO\$;CHR$ (3)
85 TIME DELAY (15)
86 PRINT #3, "SP0 ;"
90 CLOSE #3
95 COLOR 7,0
100 PLOTTER
110 STOP
END PROCEDURE

PROCEDURE: Fixpoints
INTEGER: 0%
10 ON Type% GOTO 100,200,300,400,500
100 Plt! = 48
110 IF Plt1! < 15 THEN Plt1! = 15 : EXIT
120 IF Plt1! > 166 THEN Plt1! = 166 : EXIT
198 EXIT
199 STOP
200 Plt! = 618
210 IF Plt1! < 15 THEN Plt1! = 15 : EXIT
220 IF Plt1! > 166 THEN Plt1! = 166 : EXIT
298 EXIT
299 STOP
300 Plt! = 166
310 IF Plt! < 48 THEN Plt! = 48 : EXIT
320 IF Plt! > 618 THEN Plt! = 618 : EXIT
398 EXIT
399 STOP
400 Plt1! = 15
410 IF Plt! < 48 THEN Plt! = 48 : EXIT
420 IF Plt! > 618 THEN Plt! = 618 : EXIT
500 EXIT
END PROCEDURE

PROCEDURE: Cleargraph
EXTERNAL: Pulse$(), Hold$(), Rate$(), Slope$(), X_int$(), Y_int$(), Range$(), Correlation$() (_CAPTURE)
EXTERNAL: Tyme$(), Temp$(), Grange$(), G1, G2, Comp_graph%
REAL: I, N
EXTERNAL: Bias$(), Max_graph_points%()
EXTERNAL: Graph$()
INTEGER: J
10 Bias$(G1orG2) = "X"
20 Pulse$(G1orG2) = "X"
30 Hold$(G1orG2) = "X"
40 Rate$(G1orG2) = "X"
50 Slope$(G1orG2) = "X"
60 X_int$(G1orG2) = "X"
70 Y_int$(G1orG2) = "X"
80 Range$(G1orG2) = "X"
90 Correlation$(G1orG2) = "X"
100 Tyme$(G1orG2) = "X"
110 Temp$(G1orG2) = "X"
120 Grange$(G1orG2) = "X"
130 IF Max_graph_points%(G1orG2)>300 THEN J=Max_graph_points%(G1orG2) ELSE J=300
140 FOR I=1 TO 2
150 FOR N=1 TO J
160 Graph!(G1orG2,I,N)=0
170 NEXT N
180 NEXT I
185 Max_graph_points%(G1orG2)=1
190 IF G1orG2=1 THEN G1=0
200 IF G1orG2=2 THEN G2=0:Comp_graph%=0
210 EXIT
220 STOP

END PROCEDURE

PROCEDURE: CLRGRPH

EXTERNAL: G2, COMP_GRAPH%, MODIFY, Cleargraph, G1
REAL ARRAY(10): Exist
EXTERNAL: Clearscrean
REAL: I
STRING: N$[?]
REAL: N
STRING: A$[?]
EXTERNAL: MOD
10 Exist(1)=G1:Exist(2)=G2
11 Exist(3)=0:Exist(4)=0:Exist(5)=0
20 Clearscrean(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EXISTING GRAPHS:"
30 FOR I=1 TO 4
40 IF Exist(I)=0 THEN GOTO 60
50 SET CURSOR 5,2*I+18:COLOR 0,7:PRINT Exist(I)
60 NEXT I
70 SET CURSOR 15,1:COLOR 7,0:INPUT"CLEAR WHICH GRAPH N$"
80 N=VAL(N$):MOD=N
81 IF N(1 OR N)>3 THEN GOTO 100
90 Cleargraph N
100 Clearscrean(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EXISTING GRAPHS:"
105 Exist(1)=G1:Exist(2)=G2
110 FOR I=1 TO 4
120 IF Exist(I)=0 THEN GOTO 140
130 SET CURSOR 5,2*I+18:COLOR 0,7:PRINT Exist(I)
140 NEXT I
150 SET CURSOR 18,1:COLOR 7,0:INPUT"FINISHED CLEARING GRAPHS" A$
160 A$=UPPER$(A$)
170 IF INSTR(A$,"Y")=0 THEN GOTO 10
180 MODIFY
190 STOP
END PROCEDURE

PROCEDURE: ADDNUM
  EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%(()  
  INTEGER ARRAY(8): EXIST  
  EXTERNAL: CLEARSCREEN  
  INTEGER: I,W  
  EXTERNAL: MODIFY  
  INTEGER: M  
  REAL: NUM  
  INTEGER: J  
  EXTERNAL: MOD
    10 EXIST(1)=G1:EXIST(2)=G2  
    11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0  
    20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EXISTING GRAPHS:"  
    30 FOR I=1 TO 4  
    40 IF EXIST(I)=0 THEN GOTO 60  
    50 SET CURSOR 5,E*I+1B:COLOR 0,7:PRINT EXIST(I)  
    60 NEXT I  
    70 SET CURSOR 15,1:COLOR 7,0:INPUT"ADD TO WHICH GRAPH" W$  
    80 W=VAL(W$):MOD=W  
    81 IF W<1 OR W>E THEN GOTO 100  
    82 SET CURSOR 17,1:COLOR 7,0:INPUT"ADD TO X OR Y" X$  
    83 X$=UPPER$(X$)  
    84 M=1  
    85 IF INSTR(X$,"X")=0 THEN M=2  
    87 SET CURSOR 19,1:COLOR 7,0:INPUT"ADD WHAT NUMBER" NUM$  
    88 NUM=VAL(NUM$)  
    90 FOR J=1 TO MAX_GRAPH_POINTS%(W)  
    92 GRAPH!(W,M,J)=GRAPH!(W,M,J)+NUM  
    94 NEXT J  
    100 REM  
    150 SET CURSOR 21,1:COLOR 7,0:INPUT"FINISHED ADDING" A$  
    160 A$=UPPER$(A$)  
    170 IF INSTR(A$,"Y")=0 THEN GOTO 10  
    180 MODIFY  
    190 STOP

END PROCEDURE

PROCEDURE: Mltnum
  EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%(()  
  INTEGER ARRAY(8): EXIST  
  EXTERNAL: CLEARSCREEN  
  INTEGER: I,W  
  EXTERNAL: MODIFY
INTEGER: M
REAL: NUM
INTEGER: J
EXTERNAL: MOD
10 EXIST(1)=G1:EXIST(2)=G2
11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EXISTING GRAPHS:"
30 FOR I=1 TO 4
40 IF EXIST(I)=0 THEN GOTO 60
50 SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
60 NEXT I
70 SET CURSOR 15,1:COLOR 7,0:INPUT"MLTNUM WHICH GRAPH" W$
80 W=VAL(W$):MOD=W
81 IF W<1 OR W>E THEN GOTO 100
82 SET CURSOR 17,1:COLOR 7,0:INPUT"MLT TO X OR Y" X$
83 X$=UPPER$(X$)
84 M=1
85 IF INSTR(X$,"X")=0 THEN M=2
87 SET CURSOR 19,1:COLOR 7,0:INPUT"MLT WHAT NUMBER"
90 FOR J=1 TO MAX_GRAPH_POINTS%(W)
92 GRAPH!(W,M,J)=GRAPH!(W,M,J)*NUM
94 NEXT J
100 REM
150 SET CURSOR 21,1:COLOR 7,0:INPUT"FINISHED MULTIPLYING" A$
160 A$=UPPER$(A$)
170 IF INSTR(A$,"Y")=0 THEN GOTO 10
180 MODIFY
190 STOP
END PROCEDURE
PROCEDURE: POWER
EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%()
INTEGER ARRAY(8): EXIST
EXTERNAL: CLEARSCREEN
INTEGER: I,W
EXTERNAL: MODIFY
INTEGER: M
REAL: NUM
INTEGER: J
EXTERNAL: MOD
10 EXIST(1)=G1:EXIST(2)=G2
11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EXISTING GRAPHS:"
30 FOR I=1 TO 4
40 IF EXIST(I)=0 THEN GOTO 60
50 SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
60 NEXT I
70 SET CURSOR 15,1:COLOR 7,0:INPUT"POWER WHICH GRAPH
" W$
80 W=VAL(W$):MOD=W
81 IF W(1 OR W)=2 THEN GOTO 100
82 SET CURSOR 17,1:COLOR 7,0:INPUT"POWER X OR Y" X$
83 X$=UPPER$(X$)
84 M=1
85 IF INSTR(X$,"X")=0 THEN M=2
87 SET CURSOR 19,1:COLOR 7,0:INPUT"WHAT POWER" NUM$
88 NUM=VAL(NUM$)
90 FOR J=1 TO MAX_GRAPH_POINTS%(W)
92 GRAPH!(N,M,J)=GRAPH!(W,M,J)^NUM
94 NEXT J
100 REM
150 SET CURSOR 51,1:COLOR 7,0:INPUT"FINISHED POWERING
" A$
160 A$=UPPER$(A$)
170 IF INSTR(A$,"Y")=0 THEN GOTO 10
180 MODIFY
190 STOP
END PROCEDURE

PROCEDURE: NATLOG
EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%(G)
INTEGER ARRAY(8): EXIST
STRING: W$(2),X$(2),NUM$(10),A$(2)
EXTERNAL: CLEARSCREEN
INTEGER: I,W
EXTERNAL: MODIFY
INTEGER: M
REAL: NUM
INTEGER: J
EXTERNAL: MOD
10 EXIST(1)=G1:EXIST(2)=G2
11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EXISTING GRAPHS:
" 30 FOR I=1 TO 4
40 IF EXIST(I)=0 THEN GOTO 60
50 SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
60 NEXT I
70 SET CURSOR 15,1:COLOR 7,0:INPUT"NATLOG WHICH GRAPH
" W$
80 W=VAL(W$):MOD=W
81 IF W<1 OR W>2 THEN GOTO 100
82 SET CURSOR 17,1:COLOR 7,0:INPUT"NATLOG X OR Y" X$
83 X$=UPPER$(X$)
84 M=1
85 IF INSTR(X$, "X")=0 THEN M=2
90 FOR J=1 TO MAX_GRAPH_POINTS%(W)
92 GRAPH!((W, M, J))=LOG(GRAPH!((W, M, J))
94 NEXT J
100 REM
150 SET CURSOR 21, 1: COLOR 7, 0: INPUT "FINISHED NATLOG"

A$
160 A$=UPPER$(A$)
170 IF INSTR(A$, "Y")=0 THEN GOTO 10
180 MODIFY
190 STOP
END PROCEDURE

PROCEDURE: EXPNT
EXTERNAL: GRAPH!(), G1, G2, MAX_GRAPH_POINTS%(())
INTEGER ARRAY (8): EXIST
EXTERNAL: CLEQRSCREEN
INTEGER: I, W
EXTERNAL: MODIFY
INTEGER: M
REAL: NUM
INTEGER: J
EXTERNAL: MOD
10 EXIST(1)=G1: EXIST(2)=G2
11 EXIST(3)=0: EXIST(4)=0: EXIST(5)=0
20 CLEQRSCREEN(2): SET CURSOR 5, 1: COLOR 7, 0: PRINT "EXISTING GRQPHS:"
30 FOR I=1 TO 4
40 IF EXIST(I)=0 THEN GOTO 60
50 SET CURSOR 5, 2*I+1: COLOR 0, 7: PRINT EXIST(I)
60 NEXT I
70 SET CURSOR 15, 1: COLOR 7, 0: INPUT "EXPNT WHICH GRQPH II80 w=VQL(w$: MOD=w$
81 IF W(1 OR W)2 THEN GOTO 100
82 SET CURSOR 17, 1: COLOR 7, 0: INPUT "EXPNT X OR Y" X$
83 X$=UPPER$(X$)
84 M=1
85 IF INSTR(X$, "X")=0 THEN M=2
90 FOR J=1 TO MAX_GRAPH_POINTS%(W)
92 GRAPH!(W, M, J)=EXP(GRAPH!(W, M, J))
94 NEXT J
100 REM
150 SET CURSOR 21, 1: COLOR 7, 0: INPUT "FINISHED EXPNT" A$
160 A$=UPPER$(A$)
170 IF INSTR(A$, "Y")=0 THEN GOTO 10
180 MODIFY
190 STOP
END PROCEDURE
PROCEDURE: ABSLT
EXTERNAL: GRAPH(), G1, G2, MAX_GRAPH_POINTS%
INTEGER ARRAY(8): EXIST
EXTERNAL: CLEARSCREEN
INTEGER: I, W
EXTERNAL: MODIFY
INTEGER: M
REAL: NUM
INTEGER: J
EXTERNAL: MOD
10 EXIST(1)=G1:EXIST(2)=G2
11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EXISTING GRAPHS:"
30 FOR I=1 TO 4
40 IF EXIST(I)=0 THEN GOTO 60
50 SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
60 NEXT I
70 SET CURSOR 15,1:COLOR 7,0:INPUT "ABSLT WHICH GRAPH" W$
80 W=VAL(W$):MOD=W
81 IF W<1 OR W>2 THEN GOTO 100
82 SET CURSOR 17,1:COLOR 7,0:INPUT "ABSLT X OR Y" X$
83 X$=UPPER$(X$)
84 M=1
85 IF INSTR(X$,"X")=0 THEN M=2
90 FOR J=1 TO MAX_GRAPH_POINTS%(W)
92 GRAPH!(W,M,J)=ABS(GRAPH!(N,M,J))
94 NEXT J
100 REM
150 SET CURSOR 21,1:COLOR 7,0:INPUT "FINISHED ABSLT" A$
160 A$=UPPER$(A$)
170 IF INSTR(A$,"Y")=0 THEN GOTO 10
180 MODIFY
190 STOP
END PROCEDURE

PROCEDURE: DELPT
EXTERNAL: GRAPH(), G1, G2, MAX_GRAPH_POINTS%
INTEGER ARRAY(8): EXIST
EXTERNAL: CLEARSCREEN
INTEGER: I, W
EXTERNAL: MODIFY
INTEGER: M
REAL: NUM
INTEGER: J, X
STRING: D$[4]
INTEGER: D, N, FLAG
STRING: DP$[?]
INTEGER: DP
REAL: MAIN
STRING: Null$[?]
EXTERNAL: MOD

10 EXIST(1)=G1: EXIST(2)=G2
11 EXIST(3)=0: EXIST(4)=0: EXIST(5)=0
20 CLEARSERIE(2): SET CURSOR 5, 1: COLOR 7, 0: PRINT "EXISTING GRAPHS:"
30 FOR I=1 TO 4
40 IF EXIST(I)=0 THEN GOTO 60
50 SET CURSOR 5, 2*I+18: COLOR 0, 7: PRINT EXIST(I)
60 NEXT I
70 SET CURSOR 15, 1: COLOR 7, 0: INPUT "DELETE FROM WHICH GRAPH" W$
80 W=VAL(W$): MOD=W
81 IF W<1 OR W>2 THEN GOTO 240
90 N=-2
100 M=1
101 FOR J=3 TO 20
102 SET CURSOR J—N, 40: COLOR 2, 0: PRINT " "
104 NEXT J
110 FOR J=M TO MAX_GRAPH_POINTS%(N)
120 SET CURSOR J—N, 40: COLOR 2, 0: PRINT J; ";GRAPH! (W, 1, J); ";GRAPH! (W, 2, J)
130 IF J—N=15 THEN EXIT TO, 150
135 FLAG=J
140 NEXT J
150 N=J—3: M=J
160 SET CURSOR 17, 1: COLOR 7, 0: INPUT "DELETE WHICH POINT" ; D$
170 D=VAL(D$)
180 IF D=0 AND FLAG<>MAX_GRAPH_POINTS%(W) THEN GOTO 1
190 IF D=0 THEN GOTO 240
210 FOR J=D TO MAX_GRAPH_POINTS%(W)
220 GRAPH! (W, 1, J)=GRAPH! (W, 1, J+1)
221 GRAPH! (W, 2, J)=GRAPH! (W, 2, J+1)
230 NEXT J
231 MAX_GRAPH_POINTS%(W)=MAX_GRAPH_POINTS%(W)-1
240 REM
250 SET CURSOR 21, 1: COLOR 7, 0: INPUT "FINISHED DELPT" A$
260 A$=UPPER$(A$)
270 IF INSTR(A$, "Y")=0 THEN GOTO 10
280 MODIFY
290 STOP
END PROCEDURE
PROCEDURE: ADDPT

EXTERNAL: GRAPH!(), G1, G2, MAX_GRAPH_POINTS%()
INTEGER ARRAY(8): EXIST
STRING: W$(2), X$(2), NUM$(10), A$(2)
EXTERNAL: CLEARSCREEN
INTEGER: I, W
EXTERNAL: MODIFY
INTEGER: M
REAL: NUM
INTEGER: J, X
STRING: AP$(4)
INTEGER: AP, N, FLAG
STRING: XAD$(10), YAD$(10)
REAL: XAD, YAD
EXTERNAL: MOD

10 EXIST(1)=G1: EXIST(2)=G2
11 EXIST(3)=0: EXIST(4)=0: EXIST(5)=0
20 CLEARSCREEN(2): SET CURSOR 5, 1: COLOR 7, 0: PRINT "EXISTING GRAPHS:"
30 FOR I=1 TO 4
40 IF EXIST(I)=0 THEN GOTO 60
50 SET CURSOR 5, 2*I+16: COLOR 0, 7: PRINT EXIST(I)
60 NEXT I
70 SET CURSOR 15, 1: COLOR 7, 0: INPUT "ADDPT TO WHICH GRAPH" W$
80 W=VAL(W$): MOD=W
81 IF W<1 OR W>2 THEN GOTO 240
90 N=-2
100 M=1
101 FOR J=3 TO 20
102 SET CURSOR J, 40: COLOR 2, 0: PRINT " 
104 NEXT J
110 FOR J=M TO MAX_GRAPH_POINTS%(W)
120 SET CURSOR J-N, 40: COLOR 2, 0: PRINT J; "; GRAPH!(W, 1, J); "; GRAPH!(W, 2, J)
130 IF J-N=15 THEN EXIT TO, 150
135 FLAG=J
140 NEXT J
150 N=J-3: M=J
160 SET CURSOR 17, 1: COLOR 7, 0: INPUT "ADD BEFORE WHICH POINT"; AP$
170 AP=VAL(AP$)
180 IF AP=0 AND FLAG() MAX_GRAPH_POINTS%(W) THEN GOTO 101
190 IF AP=0 THEN GOTO 240
191 SET CURSOR 19, 1: COLOR 7, 0: INPUT "ENTER NEW X, Y"; XAD$, YAD$
210 FOR J=MAX_GRAPH_POINTS%(W) TO AP STEP -1
220 GRAPH!(W, 1, J+1)=GRAPH!(W, 1, J)
221 GRAPH!(W, 2, J+1)=GRAPH!(W, 2, J)
'MQIN Pwogwam:

100 DD
110 SCREEN 0,0,0 : VIEW : CLS : STATUSLINE OFF : TROFF
120 COLOR 2,1:CLS:SET CURSOR 13,30:PRINT "Now Entering Graphics"
130 TIMEDELAY (3)
140 COLOR 2,0,0
145 CLS
150 END DO
155 SET CURSOR 0,74:COLOR 8,7:PRINT G1;G2:COLOR 7,0
160 MAINKEY
200 CLEAR:WIDTH 80
210 ERROR GOTO 55000
230 CHDIR "\"
240 MinX! = 1E+19
250 MaxX! = -1E+19
260 MinY! = 1E+19
270 MaxY! = -1E+19
275 FOR N% = 1 TO 2
280 Cleargraph N%
385 NEXT N%
390 Q! = 1.60218E-19
400 K! = 1.38066E-23
410 E_sub_0! = 8.85418E-14
490 Minloop% = 1
500 REM
510 RESTORE,60200
520 READ Max_graph_number%
530 FOR N% = 1 TO 5
540 FOR M% = 1 TO 24
550 READ Graph_name$(N%, M%)
560 IF (N%-1)*24 + M% => Max_graph_number% THEN EXIT
2 LEVELS
570 NEXT M%
580 Maxloop%(N%) = 24
590 NEXT N%
600 Maxloop%(N%) = M% - 1
605 IF M% = 24 THEN Maxloop%(N%) = 24
610 IF N% = 6 THEN Next_screen% = 5 ELSE Next_screen% = N%
700 REM
710 FOR N% = 1 TO Next_screen%
720 TITLE ("Graphics Menu")
730 FOR M% = 1 TO Maxloop%(N%)
740 MENU (Graph_name$(N%, M%), M%, Rownumber%)
750 NEXT M%
760 DO
770 IF N% = Next_screen% THEN EXIT
780 IF N% ) 1 THEN Col% = 5
790 FINISH ("Go to Next Graphics Menu", Rownumber%, Col%, Co1%, 25):EXIT
800 REPEAT
810 DO
820 IF N% < 2 THEN EXIT
830 IF N% < Next_screen% THEN Col% = 42
835 IF N% = Next_screen% THEN Col% = 5
840 FINISH ("Go to Previous Graphics Menu", Rownumber%, Col%, 26):EXIT
850 REPEAT
860 DO
870 IF N% < Next_screen% THEN EXIT
875 Col% = 42
880 FINISH ("Exit to DOS", Rownumber%, Col%, 27):EXIT
890 REPEAT
895 BORDER (Rownumber%)
900 REM
910 DO
920 ON N% GOTO 940, 950, 960, 970
930 CLS:PRINT "ERROR SAVE SCREEN ROUTINE":STOP
940 SAVE SCREEN 0, 0, 24, 79, MENU1:EXIT
950 SAVE SCREEN 0, 0, 24, 79, MENU2:EXIT
960 SAVE SCREEN 0, 0, 24, 79, MENU3:EXIT
970 SAVE SCREEN 0, 0, 24, 79, MENU4:EXIT
990 REPEAT
1000 REM
1010 DO
1020 CLEARSCREEN (21)
1030 COLOR 6,0:SET CURSOR 21,5
261

1040
INPUT "Enter the number to be executed";ChbiceZ
1050
IF ChoiceZ > 0 GND Ch¤iceZ <=Max1b0pZ(NZ) THEN E
XIT 2 LEVELS
1060
IF Ch0iceZ = 26 GND NZ ) 1 THEN EXIT TO,1100
1070
IF Ch0iceZ = 25 GND NZ < Next_screenZ THEN EXIT
TO,1300
1080
IF Ch0iceZ = 27 GND NZ = Next_screenZ THEN CLS:S
YSTEM
1090
REPEGT
1 100
REM
1110
DO
NZ = NZ — 1
1120
ON NZ GOTO 1150,1160,1170,1180
1130
1140
CLS:PRINT "ERROR IN RESTORE ROUTINE":STOP
1150
RESTORE SCREEN 0,0,24,79,MENU1:EXIT

1160

RESTORE SCREEN 0,0,24,79,MENU2:EXIT

1170
RESTORE SCREEN 0,0,24,79,MENU3:EXIT
1180
RESTORE SCREEN 0,0,24,79,MENU4:EXIT
1200
REPEGT
GOTO 1000
1210
1300
REM
1310 NEXT NZ
1320 C01Z = CINT((80 — LEN(Graph_name$(NZ,ChoiceZ)))/2)

1330 CLS:COLOR 3,0:SET CURSOR 0,C0lZ
1340 PRINT Graph_name$(NZ,ChoiceZ)
1350 Max_graph_p0intsZ(1) = 1
1360 Max_graph_pointsZ(2) = 1
1400 REM
1410 Menu_ch0senZ = NZ
1420 Braph_ch0senZ = Ch0iceZ
1425 GOTO 160
1429 Cleargraph 1:G1=1:GraphZ=1
1430 WIDTH 80:GFLGG=GraphZ:TITLE(Graph_name$(Menu_chbsenZ
,Graph_ch0senZ)):ON Menu_ch0senZ GOTO 1500,1600,1700,1800,
1900
1440 CLS:COLOR 7,0:PRINT "ERROR IN MENU_CHOSEN ROUTINE (1
420)":STOP
1500 REM
1510 ON Graph_ch0senZ GOSUB 20000,10000,10000,23000,23000
,17000,17000,23000,23000,23000,10000,11000,11000,11000,110
00,11000,11000,11000,11000,11000,15000,15000,15000,15000
1520 CLS:PRINT “ERROR IN FIRST MENU ROUTINE (1500)":STOP
1600 REM
1610 ON Graph_ch0senZ GOSUB 23000,23000,11000,25000,2000,
1620 CLS:PRINT "ERROR IN SECOND MENU ROUTINE (1600)":STOP
1700 REM
1720 CLS:PRINT "ERROR IN THIRD MENU ROUTINE (1700)":STOP
1800

REM


1820 CLS:PRINT "ERROR IN FOURTH MENU ROUTINE (1800)";STOP
1900 REM
1920 CLS:PRINT "ERROR IN FIFTH MENU ROUTINE (1900)";STOP
2000 CLS:PRINT "THESE GRAPHS PRE NOT INSTALLED."
2010 TIMEDELTA ();
2020 GOTO 200
2425 GRange$(Graph%) = STR$(Min!) + "," + STR$(Max!)
2430 GOTO 160
3000 WIDTH 80;Minloop%=1
3002 IF G1=1 THEN DELTAMULTIPLY(1,1,1.0);DELTAMULTIPLY(1,
2,1.0)
3003 IF G2=2 THEN DELTAMULTIPLY(2,1,1.0);DELTAMULTIPLY(2,
2,1.0)
3007 CLOSE
3010 DO
3020 TITLE ("Extrema for X and Y axes")
3030 COLOR 4,0;SET CURSOR 2,5
3040 PRINT "Minimum X";SPC(10);"Maximum X";SPC(10);"Min
imum Y";SPC(10);"Maximum Y"
3050 COLOR 3,0
3060 SET CURSOR 3,7;PRINT MinX!
3070 SET CURSOR 3,26;PRINT MaxX!
3080 SET CURSOR 3,45;PRINT MinY!
3090 SET CURSOR 3,64;PRINT MaxY!
3100 COLOR 7,0;SET CURSOR 19,5
3110 INPUT "Do you want to change any of the extrema (D
defaults to NO)";Check$
3120 Check$ = UPPER$(Check$)
3130 IF INSTR(Check$,"Y") = 0 THEN EXIT 1 LEVELS
3140 CLEARSCREEN (19)
3150 DO 1 TIMES
3160 COLOR 6,0;SET CURSOR 9,5
3170 PRINT "Enter the Minimum X value (Defaults to ");
MinX!;"");
3180 INPUT Change$
3190 IF Change$ () "" THEN MinX! = VAL(Change$)
3200 COLOR 6,0;SET CURSOR 11,5
3210 PRINT "Enter the Maximum X value (Defaults to ");
MaxX!;"");
3220 INPUT Change$
3230 IF Change$ () "" THEN MaxX! = VAL(Change$)
3240 COLOR 6,0;SET CURSOR 13,5
3250 PRINT "Enter the Minimum Y value (Defaults to ");
MinY!;"");
3260 INPUT Change$
3270 IF Change$ () "" THEN MinY! = VAL(Change$)
3280 COLOR 6,0;SET CURSOR 15,5
3290 PRINT "Enter the Maximum Y value (Defaults to "; MaxY!; ")";
3300 INPUT Change$
3310 IF Change$ <> "" THEN MaxY! = VAL(Change$)
3320 REPEQT
3330 REPEQT
3340 GOTO 160
4000 REM
4010 DO
4020 CLS:COLOR 2,0:SET CURSOR 11,5
4030 INPUT "Do you want a dot (1) or a line (2) graph";
Choice% = Choice% + Choice% * 3 THEN EXIT
4050 REM
4060 Graph_type% = Choice%
4500 REM
4510 DO TIMES
4520 COLOR 2,0
4530 SCREEN 2
4540 DRAW "BM200,100 U2 D4 U2 L2 R4"
4550 GET (198,102) — (202,98),PLUS
4560 CLS
4570 REPEQT
5000 REM
5010 Col% = CINT((79 - LEN(Graph_name$(Menu_chosen%,Graph_chosen%)))/2)
5020 LOCATE 1,Col%:PRINT Graph_name$(Menu_chosen%,Graph_chosen%)
5050 XTITLE (Xtitle$,MinX!,MaxX!)
5060 YTITLE (Ytitle$,MinY!,MaxY!)
5500 REM
5510 DeltaX! = (570)/(MaxX! - MinX!)
5520 DeltaY! = (150)/(MaxY! - MinY!)
6000 REM
6004 ERASE Plot!(),Skippoint%(1)
6005 DIM Skippoint%(2,600)
6010 FOR M% = 1 TO Max_graph_points%(1)
6015 DO
6017 IF MinX! > Graph!(1,1,M%) THEN Skippoint%(1,M%) = 1:EXIT 1 LEVELS
6019 IF MaxX! < Graph!(1,1,M%) THEN Skippoint%(1,M%) = 2:EXIT 1 LEVELS
6021 IF MinY! > Graph!(1,2,M%) THEN Skippoint%(1,M%) = 3:EXIT 1 LEVELS
6023 IF MaxY! < Graph!(1,2,M%) THEN Skippoint%(1,M%) = 4:EXIT 1 LEVELS
6025 X_plot! = DeltaX!*(Graph!(1,1,M%) - MinX!) +46
6026 Y_plot! = DeltaY!*(Graph!(1,2,M%) - MinY!) +165
6030 PSET (X_plot!,Y_plot!)
6031 Skippoint%(1,M%) = 5
6035 END DO
6040 NEXT M%
6045 DO 1 TIMES
6050 IF Comp_graph% = 0 THEN EXIT
6120 FOR M% = Minloop% TO Max_graph_points%(2)
6130 DO
6140 IF MinX! > Graph!(2,1,M%) THEN Skippoint%(2,M%) = 1 : EXIT 1 LEVELS
6145 IF MaxX! < Graph!(2,1,M%) THEN Skippoint%(2,M%) = 2 : EXIT 1 LEVELS
6150 IF MinY! > Graph!(2,2,M%) THEN Skippoint%(2,M%) = 3 : EXIT 1 LEVELS
6155 IF MaxY! < Graph!(2,2,M%) THEN Skippoint%(2,M%) = 4 : EXIT 1 LEVELS
6160 X_plot! = DataX!*(Graph!(2,1,M%) - MinX!) +46
6170 Y_plot! = -DataY!*(Graph!(2,2,M%) - MinY!) +16
6180 PUT (X_plot!,Y_plot!),PLUS,OR
6190 Skippoint%(2,M%) = 5
6200 END DO
6208 NEXT M%
6209 REPEAT
6210 DIM Plot!(2,2)
6212 DO
6213 IF Graph_type% <> 2 THEN EXIT 1 LEVELS
6214 FOR N% = 1 TO 2
6215 IF N% = 2 AND Comp_graph% <> 1 THEN EXIT 2 LEVEL
6216 FOR M% = 1 TO Max_graph_points%(N%) - 1
6217 DO
6218 FOR O% = 1 TO 2
6219 Plot!(O%,1) = DataX!*(Graph!(N%,1,M%+O%-1) - MinX!) +46
6220 Plot!(O%,2) = -DataY!*(Graph!(N%,2,M%+O%-1) - MinY!) +165
6224 Fixpoints Skippoint%(N%,M%+O%-1),0,Plot!(O%
6233 NEXT 0%
6235 IF N% = 2 AND (Skippoint%(1,M%) = 5 OR Skippoint%(2,M%) = 5) THEN Plot!(1,1) = Plot!(1,1)+ 2:Plot!(1,2 ) = Plot!(1,2)+2:Plot!(2,1) = Plot!(2,1)+2:Plot!(2,2) = Plot!(2
6238 LINE (Plot!(1,1),Plot!(1,2)) - (Plot!(2,1),P
6240 END DO
6242 NEXT M%
6244 NEXT N%
6245 END DO
6248 DO
6249 TIMEDELAY(2)
6250 A$ = INKEY$: IF INKEY$ = "" THEN GOTO 6250
6260 SCREEN 0,0,0 : VIEW : CLS
6265 ERASE Skippoint%(),Plot!()
6270 END DO
6275 GOTO 160
6300 WIDTH 80:TITLE ( Graph_name$(Manu_ch0sanZ,Gvaph_ch0s onZ))
6310 Fit$=UPPER$("Y")
7550 Error_LS%=0
7560 FOR M%= 1 TO 2
7562 DO 1 TIMES
7565 DO
7568 IF INSTR(Fit$,"Y")=0 THEN EXIT 1 LEVELS
7569 CLEAR (Slope!(M%),Correlation!(M%),X_int!(M%),Y_int!(M%)
7570 LEASTSQUARES (M%,Min!(M%),Max!(M%),Slope!(M%),Correlation!(M%),X_int!(M%),Y_int!(M%),M%-1,Error_LS%)
7572 IF Error_LS% <> 0 THEN Fit$ = "END"
7575 END DO
7640 REPEPT
7650 DO
7655 IF Error_LS% = 1 THEN EXIT TO,160
7660 IF INSTR(Fit$,"Y") = 0 THEN EXIT
7670 DO 1 TIMES
7680 IF M% = 2 THEN EXIT
7690 CLS : TITLE ("Least Squares Fit")
7694 COLOR 2,0,0
7700 SET CURSOR 8,0:PRINT "X — Intercept"
7710 SET CURSOR 10,0:PRINT "Y — Intercept"
7720 SET CURSOR 12,0:PRINT "Slope"
7730 SET CURSOR 14,0:PRINT "Correlation"
7740 SET CURSOR 16,0:PRINT "Range"
7750 REPEPT
7760 DO
7790 IF M% =1 THEN Co1% = 20
7800 IF M% = 2 THEN Co1% = 50
7810 SET CURSOR 6,Co1%
7820 IF M% = 1 THEN PRINT ". CURVE 1" ELSE PRINT "+ CURVE 2"
7830 FOR N% = 1 TO 6
7840 SET CURSOR N%*2 + 6,Co1%
7850 DO
7860 ON N% GOTO 7870,7880,7890,7900,7910,7920
7870 Dummy! = X_int!(M%):EXIT 1 LEVELS
7880 Dummy! = Y_int!(M%):EXIT 1 LEVELS
7890 Dummy! = Slope!(M%):EXIT 1 LEVELS
7900 Dummy! = Correlation!(M%):EXIT 1 LEVELS
7910 Dummy! = Min!(M%):EXIT 1 LEVELS
7920 SET CURSOR 16,Co1% + 8:PRINT ","
7930 SET CURSOR 16,Co1% + 10:Dummy! = Max!(M%):
EXIT 1 LEVELS
7940 REPEAT
7945 DO 1 TIMES
7950       IF Dummy! = .01 AND Dummy! (<= 1000 THEN PRINT USING ".";Dummy!:EXIT
7960       IF Dummy! (<= -.01 AND Dummy! >= -1000 THEN PRINT USING ".";Dummy!:EXIT
7970       IF Dummy! = 0 THEN PRINT "0" ELSE PRINT USING ".###^^^^";Dummy!:EXIT
7975       REPEAT
7980       NEXT N%
7990       END DO
7995       END DO
8000       NEXT M%
8015     DO 1 TIMES
8020     IF INSTR(Fit$,"Y") = 0 THEN EXIT 1 LEVELS
8030     TIMEDELY (4)
8040     SET CURSOR 21,5:COLOR 7,0:PRINT "HIT ANY KEY TO CONTINUE"
8050     A$ = INKEY$:IF A$ = "" THEN GOTO 8040
8060     REPEAT
8070     IF X_int!(1) (<> 0 OR X_int!(2) <> 0 THEN Fit$ = "Y"
8080     GOTO 160
8090     CLEARSCREEN(2) : SET CURSOR 13,6
8100     SAVEDATA(Comp_graph%+1)
8105     END DO
8120     GOTO 160
8130     Cleargraph 2;G2=2;Graph%=2;Comp_graph%=1
8140     CLS:COLOR 3,0:SET CURSOR 0,CINT((80 - LEN(Graph_name $(Menu_chosen%,Graph_chosen%)))/2)
8150     PRINT Graph_name$(Menu_chosen%,Graph_chosen%)
8160     EXIT TO,1430
8180     REM
8190     TITLE (Graph_name$(Menu_chosen%,Graph_chosen%))
8200     DO 1 TIMES
8210     IF Info_pointer% (<> 5 THEN EXIT 1 LEVELS
8220     COLOR 7,0 : SET CURSOR 13,6
8230     PRINT "WARNING!: If you plan on plotting the another graph, the information block"
8235     SET CURSOR 14,6 : PRINT "will not be printed!"
8240     TIMEDELY (5)
8245     PLOTTER
8250     REPEAT
8255     PLOTTER
8260     DO
8265     IF INSTR(Fit$,"Y") = 0 THEN EXIT 1 LEVELS
8270     Slope$(2) = STR$(Slope!(2))
8280     X_int$(2) = STR$(X_int!(2))
8290     Y_int$(2) = STR$(Y_int!(2))
8295     Range$(2) = STR$(Min!(2)) + "," +STR$(Max!(2))
8300     Correlation$(2) = STR$(Correlation!(2))
8305     END DO
8310     G=G2;Print_out%=1
8320     PLOTTING_POINTS(Symbol$,2,1)
9370 EXIT TO, 9999
9560 GOTO 9000
9570 DO
9580 Title$ = UPPER$(Graph_name$(Menu_chosen%, Graph_chosen%))
9590 PLOTTERAXES (UPPER$(Graph_name$(Menu_chosen%, Graph_chosen%)))
9600 END DO
9601 GOTO 9000
9640 DO
9650 IF INSTR(Fit$, "Y") = 0 THEN EXIT 1 LEVELS
9660 Slope$(1) = STR$(Slope!(1))
9670 X_int$(1) = STR$(X_int!(1))
9680 Y_int$(1) = STR$(Y_int!(1))
9690 Range$(1) = STR$(Min!(1)) + "," + STR$(Max!(1))
9695 Correlation$(1) = STR$(Correlation!(1))
9696 END DO
9700 DO
9701 Temp$(1) = " " + Temp$(1)
9702 Time$(1) = " " + Time$(1)
9703 Bias$(1) = " " + Bias$(1)
9704 Pulse$(1) = " " + Pulse$(1)
9705 Hold$(1) = " " + Hold$(1)
9706 Rate$(1) = " " + Rate$(1)
9707 GRange$(1) = " " + GRange!(1)
9708 G=G1
9710 PLOTTING_POINTS(Symbol$, 1, 0)
9715 Print_out$ = 1
9720 END DO
9999 GOTO 9000
10000 REM
10007 Error% = 0
10010 DO
10020 IF Graph_chosen% = 11 THEN EXIT 1 LEVELS
10030 Directory$(Graph%, 2) = "NO"
10040 END DO
10040 DO
10041 IF Graph_chosen% = 3 THEN File$ = "\G-T" ELSE File$ = "\C-T"
10045 IF Graph_chosen% = 11 THEN ID% = 0 ELSE ID% = 1
10040 DIRECTORY (File$, ID%, Graph%, Error%)
10040 IF Error% <> 0 THEN EXIT TO, 160
10434 DO 1 TIMES
10435 IF Graph_chosen% = 11 THEN ID% = 1 ELSE ID% = 0
10440 RUNTYPE(Graph%, ID%)
10445 DO
10446 IF Graph_chosen% <> 11 THEN EXIT 1 LEVELS
10447 IF INSTR(Temp_type$(Graph%), "T") = 0 AND INSTR (Temp_type$(Graph%), "R") = 0 THEN EXIT 1 LEVELS
10449 CLEARSCREEN(2)
10451 SET CURSOR 13, 6 : COLOR 7, 0
PRINT "The DLTS Spectrum cannot be done with either a TIME or ROOM TEMPERATURE run."
TIMEDELAY()
EXIT TO,10400
REPEAT
IF Graph_chosen% <> 11 THEN ID% = 1 ELSE ID% = 4
PARAME QUEST (ID%)}
IF Graph_chosen% = 11 THEN ID% = 8 ELSE ID% = 1
DATAPRINT
FOR N% = 1 TO 2
DO
IF INSTR(Directory$(Graph%,N%),"NO") <> 0 THEN EXIT 1 LEVELS
DATARETRIEVAL (ID%,Graph%,N%,Error%,0)
ENDIF <> 0 THEN EXIT TO,10400
END DO
NEXT N%
END DO
DO
IF Graph_chosen% <> 2 THEN EXIT 1 LEVELS
DELTAMULTIPLY (Graph%,2,1.0E+09)
Ytitle$ = "Capacitance in nF"
DELTAMULTIPLY (Graph%,1,1000.0)
Xtitle$ = "Time in milliSeconds"
IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp$(Graph%) = STR$(Temp_sought!)
ENDIF <> 0 THEN Tyms$(Graph%) = STR$(Tams_sought!)
Bias$(Graph%) = MID$(Bias_Found$,INSTR(Bias_foun d$,"=") + 2)
Pulse$(Graph%) = MID$(Pulse_foun d$,"=") + 2)
Hold$(Graph%) = MID$(Hold_foun d$,"=") + 2)
EXIT 2 LEVELS
END DO
DO
IF Graph_chosen% <> 3 THEN EXIT 1 LEVELS
DELTAMULTIPLY (Graph%,2,1000.0)
Ytitle$ = "Conductance in mMhos"
DELTAMULTIPLY (Graph%,1,1000.0)
Xtitle$ = "Time in milliSeconds"
IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp$(Graph%) = STR$(Temp_sought!)
ENDIF <> 0 THEN Tyms$(Graph%) = STR$(Time_sought!)
Bias$(Graph%) = MID$(Bias_foun d$,"=") + 1)
Pulse$(Graph%) = MID$(Pulse_foun d$,"=") + 1)

10745     Hold$(Graph%) = MID$(Hold_found$, INSTR(Hold_found$,
        "=") + 1)
10750     EXIT 2 LEVELS
10760     END DO
10770     DO
10780     IF Graph_chosen% <> 11 THEN EXIT 1 LEVELS
10790     DELETAMULTIPLY (Graph%, 2, 1E+13)
10800     Ytitle$ = "DLTS in pF"
10810     Xtitle$ = "Temperature in Kelvin"
10820     Bias$(Graph%) = MID$(Bias_found$, INSTR(Bias_found$,
        ",=") + 1)
10822     Pulse$(Graph%) = MID$(Pulse_found$, INSTR(Pulse_found$,
        ",=") + 1)
10823     Hold$(Graph%) = MID$(Hold_found$, INSTR(Hold_found$,
        ",=") + 1)
10824     Rate$(Graph%) = STR$(LOG(Mintyme! / Maxtyme!) / 
        (Mintyme! - Maxtyme!))
10825     EXIT 3 LEVELS
10830     END DO
10840     END DO
10980     END DO
10985     CLOSE
10990     RETURN, 160
11000     REM
11010     Counter% = 0
11030     File% = 1
11500     DO
11510     IF Graph_chosen% = 13 OR Graph_chosen% = 14 THEN F
11515     ile$ = "\CV" ELSE File$ = "\IV"
11517     IF Graph_chosen% > 16 AND Graph_chosen% < 25 THEN ID%
11520         = 0 ELSE ID% = 1
11517     IF Graph_chosen% = 3 THEN ID% = 0
11520     DIRECTORY(File$, ID%, Graph%, Error%)
11530     IF Error% <> 0 THEN EXIT TO, 160
11540     END DO
11550     DO
11560     IF Graph_chosen% > 16 OR Graph_chosen% = 3 THEN ID%
11570         = 1 ELSE ID% = 0
11574     RUNTYPE(Graph%, ID%)
11575     DO
11575     IF INSTR(Temp_type$(Graph%), "R") = 0 OR Graph_chosen% <> 17 THEN EXIT 1 LEVELS
11576     CLEARSCREEN(2)
11577     COLOR 7, 0, 0 : SET CURSOR 13, 6
11578     PRINT "You cannot do the chosen graph with a roo
11579     m temperature run."
11580     TIMEDELAY(5)
11581     REPEAT
11589     END DO
11590     DO
IF Graph_chosen% <> 20 THEN EXIT 1 LEVELS
CLEARSCREEN(2)
SET_CURSOR 13,6 : COLOR 2,0
INPUT "Enter the AE*G ",Dummy$
AeA! = VAL(Dummy$)
IF AeA! <= 0 THEN EXIT TO,11590
END DO

DO
IF Graph_chosen% <> 17 THEN EXIT 1 LEVELS
CLEARSCREEN(2)
SET_CURSOR 13,6 : COLOR 2,0,0
INPUT "Enter the forward bias. ",Dummy$
VF! = VAL(Dummy$)
END DO
DATAPRINT
DO
IF Graph_chosen% < 17 AND Graph_chosen% < 3 THEN EXIT 1 LEVELS
IF INSTR(Temp_type$(Swaph%),"T") <> 0 THEN Run_type% = 2 ELSE Run_type% = 1
IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp_type% = "R"
EXIT TO,11900
REPEAT
Run_type% = 0
DO
IF Graph_chosen% = 13 OR Graph_chosen% = 14 THEN ID% = 7 ELSE ID% = 6
DATARETRIEVAL (ID%,Graph%,File%,Error%,Run_type%)
IF Error% = 1 AND Graph!(Graph%,2,2) <> 0 AND Graph_chosen% = 16 THEN EXIT 1 LEVELS
IF Error% = 1 AND Graph!(Graph%,2,2) <> 0 AND Graph_chosen% = 12 THEN EXIT 1 LEVELS
IF Error% = 1 AND Graph!(Graph%,2,2) <> 0 AND Graph_chosen% = 15 THEN EXIT 1 LEVELS
IF Error% <> 0 AND Graph_chosen% < 17 AND Graph_chosen% <> 3 THEN EXIT TO,11500
END DO
END DO
DO
IF Graph_chosen% <> 12 THEN EXIT 1 LEVELS
DELTAMULTIPLY(Graph%,2,1000.0)
Xtitle$ = "Voltage in Volts"
Ytitle$ = "Current in mA"
EXIT 2 LEVELS
REPEAT
DO
DO
IF Graph_chosen% <> 13 THEN EXIT 1 LEVELS
DELTAMULTIPLY(Graph%,2,1E+09)
Ytitle$ = "Cap in nF"
EXIT 2 LEVELS
DO
  IF Graph_chosenX <> 14 THEN EXIT 3 LEVELS
  INVERSEPOWER(Graph%, 2, 1E+09, 2.000)
  Ytitle$ = "1/C**2 C in nF"
END DO
EXIT 2 LEVELS
REPEAT
DO
  IF Graph_chosenX > 16 OR Graph_chosenX = 3 THEN EXIT 1 LEVELS
  DELTAIn(Graph%, 2, 1.0)
  Ytitle$ = "log(I) I in A"
  DO
    IF Graph_chosenX <> 15 THEN EXIT 1 LEVELS
    Xtitle$ = "Voltage in Volts"
    DELTAMULTIPLY(Graph%, 2, 1/LOG(10))
  END DO
  EXIT 3 LEVELS
REPEAT
  PositionX = INSTR(Temp_found$, ";") + 1
  Temp! = VAL(MID$(Temp_found$, PositionX))
  DELTAMULTIPLY(Graph%, 1, (Q!/(K! * Temp!)))
  Xtitle$ = "QV / RT"
  IF Graph_chosenX = 17 THEN EXIT 1 LEVELS
END DO
EXIT 3 LEVELS
REPEAT
END DO
EXIT TO, 12530
DO
  IF Graph_chosenX <> 17 THEN EXIT 1 LEVELS
DO
  PositionX = INSTR(Temp_found$, ";") + 1
  Temp! = VAL(MID$(Temp_found$, PositionX))
END DO
CounterX = CounterX + 1
DO
  FOR M% = 1 TO Max_graph_points%(Graph%)
    IF Graph!(Graph%, 1, M%) = VF! THEN EXIT 2 LEVELS
  NEXT M%
  IF CounterX <> 1 THEN EXIT TO, 12530
CLEARSCREEN(2)
COLOR 7, 0 : SET CURSOR 13, 6
PRINT "The Forward bias you specified was not found"
TIMEDELAY (5)
EXIT TO, 11670
12570    REPEAT
12580    IF! = Graph!(Graph%,2,M%)
12590    X! = (Q!) / (K! * Temp!)
12600    Y! = LOG (ABS(If! / Temp!*2.0))
12610    OPEN "\BUCKET" FOR APPEND AS #3
12620    PRINT #3,X! "," , Y!
12630    CLOSE #File% : CLOSE #3
12632    IF INSTR(SOURCE$,"B")=0 THEN OPEN "\DATA\" + Directory$(Graph%,File%) FOR INPUT AS #File% ELSE OPEN "B:" + File$ + "."+Directory$(Graph%,File%) FOR INPUT AS #File%
12635    Max_graph_points%(Graph%) = 1
12637    IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp_type$(Graph%) = "L"
12640    IF Error% = 0 THEN EXIT TO,11900
12642    DO
12643    CLOSE #File%
12644    File% = File% + 1
12645    IF File% = 3 OR INSTR(Directory$(Graph%,2),"NO") <> 0 THEN EXIT 1 LEVELS
12646    IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp_type$(Graph%) = "R"
12647    EXIT TO,11900
12648    REPEAT
12650    DO
12660    Max_graph_points%(Graph%) = Counter%
12670    GETDATA(Graph%)
12671    Ytitle$ = "Ln (If / T**2) If in A"
12672    Xtitle$ = "q / KT"
12675    Graph_name$(1,17) = "Activation Energy"
12679    Bias$(Graph%) = STR$(Vf!)
12680    EXIT 3 LEVELS
12685    EXIT 2 LEVELS
12695    END DO
12700    Counter% = Counter% + 1
12750    REPEAT
12770    DO
12775    IF Graph_chosen% = 3 THEN EXIT 1 LEVELS
12780    IF Graph_chosen% < 18 OR Graph_chosen% > 20 THEN EXIT 2 LEVELS
12785    END DO
12790    Position$ = INSTR(Temp_found$,";") + 1
12795    Temp! = VAL(MID$(Temp_found$,Position%))
12800    DELTAln (Graph%,2,1.0)
12805    DELTAMULTIPLY (Graph%,1,(Q!/(K! * Temp!)))
12815    IF Counter% = 1 THEN ID% = 0 ELSE ID% = 1
12820    LEASTSQUARES (Graph%,Min!,Max!,Slope!,Dummy!,Dummy!,Y_int!,ID%,Error_LS%)
12832    IF Error_LS% = 1 THEN EXIT TO,160
12835    GRange$(Graph%) = STR$(Min!) + "," + STR$(Max!)
273

) 12870 END DO
12880 DO
12890 DO
12900 IF Graph_chosen% (>) 18 AND Graph_chosen% (>) 3 THEN EXIT 1 LEVELS
12905 DO
12906 IF Graph_chosen% = 3 THEN EXIT 1 LEVELS
12907 IF Y_int! < -39 OR Y_int! > 1 THEN Counter% = Counter% - 1 : EXIT TO,13313
12910 Y! = EXP(Y_int!)
12920 Ytitle$ = "Isat in Amps"
12924 Graph_name$(1,18) = "I Saturation"
12925 END DO
12926 DO
12927 IF Graph_chosen% (>) 3 THEN EXIT 1 LEVELS
12928 Y! = Y_int!
12929 Ytitle$ = "ln(Isat) in Amps"
12930 Graph_name$(2,3) = "ln(Isat)"
12935 END DO
12936 EXIT 2 LEVELS
12940 REPEAT
12950 DO
12960 IF Graph_chosen% (>) 19 THEN EXIT 1 LEVELS
12970 Y! = 1/Slope!
12980 Ytitle$ = "Ideality Factor"
12985 Graph_name$(1,19) = "Ideality Factor n"
12990 EXIT 2 LEVELS
13000 REPEAT
13010 DO
13020 IF Graph_chosen% (>) 20 THEN EXIT 1 LEVELS
13025 IF Y_int! < -39 OR Y_int! > 1 THEN Counter% = Counter% - 1 : EXIT TO,13313
13030 Y! = (K! * Temp!/ Q!) * LOG (AeA! * Temp!^2. 0/EXP(Y_int!))
13040 Ytitle$ = "Barrier in Volts"
13045 Graph_name$(1,20) = "Barrier Height (Current )"
13050 EXIT 2 LEVELS
13060 REPEAT
13070 END DO
13150 DO
13160 DO
13170 IF INSTR(Temp_type$(Graph%), "T") <> 0 THEN EXIT 1 LEVELS
13180 X! = Temp!
13190 Xtitle$ = "Temperature in Kelvin"
13200 EXIT 2 LEVELS
13210 REPEAT
13220 DO
13230 Position% = INSTR(Time_found$,"=") + 1
X! = VAL(MID$(Time_found$, Position%))
Xtitle$ = "Time in Seconds"
END DO
END DO
DO
OPEN "\BUCKET" FOR APPEND AS #3
PRINT #3, X!, "," Y!
CLOSE #File% : CLOSE #3
END DO
DO
Max_graph_points%(Graph%) = 1
CLOSE #File%
IF INSTR(SOURCE$, "B") = 0 THEN OPEN "\DATA\" + Directory$(Graph%, File%) + File$ FOR INPUT AS #File% ELSE OPEN "B:" + File$ + "." + Directory$(Graph%, File%) FOR INPUT AS #File%
IF INSTR(Temp_type$(Graph%), "T") = 0 THEN Temp_type$(Graph%) = "L"
END DO
IF Error% = 0 THEN EXIT TO, 11900
END DO
DO
CLOSE #File%
File% = File% + 1
IF File% = 3 OR INSTR(Directory$(Graph%, 2), "NO ") = 0 THEN EXIT 1 LEVELS
IF INSTR(Temp_type$(Graph%), "T") = 0 THEN Temp_type$(Graph%) = "R"
EXIT TO, 11900
REPEAT
DO
Max_graph_points%(Graph%) = Counter%
GETDATA(Graph%)
END DO
EXIT 2 LEVELS
END DO
END DO
DO
IF Graph_chosen% = 3 THEN EXIT 1 LEVELS
END DO
DO
IF INSTR(Temp_type$(Graph%), "T") = 0 THEN EXIT 1 LEVELS
Position% = INSTR(Temp_found$, ";" ) + 1
Temp$(Graph%) = MID$(Temp_found$, Position%, 14)
EXIT 2 LEVELS
REPEAT
DO
Position% = INSTR(Time_found$, "=" ) + 1
Tyme$(Graph%) = MID$(Time_found$, Position%, 14)
END DO
END DO
13560 RETURN, 160
15000 REM
15010 DO
15020 IF Graph_chosen% > 20 AND Graph_chosen% < 25 THEN EXIT 1 LEVELS
15030 CLEARSCREEN(2)
15040 COLOR 7, 0, 0 : SET CURSOR 13, 6
15050 PRINT "The graph for the mobility experiment chosen is not installed (oops)"
15060 END
15070 REPEAT
15090 File% = 1
15095 File$ = "\MOB"
15100 DO
15110 DIRECTORY(File$, 0, Graph%, Error%) 
15130 END DO
15140 RUNTYPE(Graph%, 1)
15200 DO
15210 CLEARSCREEN(2)
15220 COLOR 2, 0, 0 : SET CURSOR 10, 6
15230 INPUT "Enter the samples electrical thickness. ", Dummy$
15240 Elec_thick! = VQL(Dummy$)
15250 IF Elec_thick! <= 0 THEN EXIT TO, 15200
15260 END DO
15300 DO
15310 IF Graph_chosen% < 24 THEN EXIT 1 LEVELS
15320 CLEARSCREEN(16)
15330 SET CURSOR 16, 6
15340 INPUT "Enter the average dopant density. ", Dummy$
15350 Nd! = VQL(Dummy$)
15360 IF Nd! <= 0 THEN EXIT TO, 15300
15370 END DO
15380 DQPRINT
15500 DO
15510 DO
15516 IF INSTR(Temp_type$(Graph%), "T") = 0 THEN Temp_type$(Graph%) = "R"
15520 END DO
15520 IF INSTR(Directory$(Graph%, File%), "NO") <> 0 THEN EXIT 2 LEVELS
15530 INPUT #File%, Dummy$, I$, V1$, V2$, Dummy$
15540 R1! = VAL(V2$) / VAL(I$)
15550 INPUT #File%, Dummy$, I$, V1$, V2$, Dummy$
15560 R2! = VAL(V2$) / VAL(I$)
15570 INPUT #File%, Dummy$, I$, V1$, V2$, Dummy$
15575 Dummy! = MIN(R2!, R1!)
15576 Eric! = MAX(R2!, R1!)
15580 G! = 1.0 / ((3.141592654 * VAL(I$) / (2 * VAL(V1$) * .693147181)) * (R1! + R2!) * (1 - .138974234 * LOG(ABS(Eric! / Dummy!))))
15590  DATA RETRIEVAL (9, Graph%, File%, Error%, 0)
15600  IF Error% () 0 THEN EXIT TO, 15200
15610  END DO
15700  DO
15720  DO
15720  IF Graph_chosen% (21 THEN EXIT 1 LEVELS
15725  Graph_name$(1, 21) = "Resistivity"
15730  Ytitle$ = "Resistivity in ohms*cm"
15731  DO
15732   CLOSE #File%
15733   File% = File% + 1
15734   IF File% = 3 OR INSTR(Directory$(Graph%, 2), "NO ") () 0 THEN EXIT 1 LEVELS
15735   Max_graph_points%(Graph%) = Max_graph_points%(Graph%) + 1
15736   EXIT TO, 15500
15737   REPEAT
15740   EXIT 2 LEVELS
15750  END DO
15760  DO
15770  IF Graph_chosen% (22 OR Graph_chosen% ) 24 THEN EXIT 1 LEVELS
15780  CLOSE #File%
15790  DO
15810  IF INSTR(SOURCE$, "B")=0 THEN OPEN "/DATA/" + Directory$(Graph%, File%) + File$ FOR INPUT AS #File% ELSE OPEN "B:" + File$ + "." + Directory$(Graph%, File%) FOR INPUT AS #File%
15820  DO 15 TIMES
15830  INPUT #File%, Dummy$
15840  REPEAT
15850  END DO
15855  IF File% = 1 THEN Counter% = Max_graph_points%(Graph%) + 1
15856  IF File% = 1 THEN Max_graph_points%(Graph%) = 1
15857  IF File% = 2 THEN Max_graph_points%(Graph%) = Counter%
15860  DO
15870  IF INSTR(Directory$(Graph%, File%), "NO") () 0 THEN EXIT 1 LEVELS
15880  DATA RETRIEVAL (10, Graph%, File%, Error%, 0)
15890  IF Error% () 0 THEN EXIT TO, 15200
15900  END DO
15912  DO
15913  CLOSE #File%
15914  File% = File% + 1
15915  IF File% = 3 OR INSTR(Directory$(Graph%, 2), "NO ") () 0 THEN EXIT 1 LEVELS
15916  Max_graph_points%(Graph%) = Max_graph_points%(Graph%) + 1
15917  EXIT TO, 15500
REPEAT
DELTAMULTIPLY(Graph%, 2, Elec_thick!)
DO
IF Graph_chosen% <> 22 THEN EXIT 1 LEVELS
Ytitle$ = "Mobility in cm**2/V-s"
Graph_name$(1, 22) = "Mobility"
EXIT 3 LEVELS
END DO
DO
IF Graph_chosen% <> 23 AND Graph_chosen% <> 24 THEN EXIT 1 LEVELS
OPEN "\BUCKET" FOR INPUT AS #3
FOR N% = 1 TO Max_graph_points%(Graph%)
    INPUT #3, Y!
    Graph!(Graph%, 2, N%) = 1 / (Q! * Y! * Graph!(Graph%, 2, N%))
NEXT N%
EXIT 3 LEVELS
EXIT 4 LEVELS
END DO
DO
IF Graph_chosen% <> 23 THEN EXIT 1 LEVELS
Ytitle$ = "Carrier Conc in 1/cm**3"
Graph_name$(1, 23) = "Carrier Concentration"
EXIT 3 LEVELS
END DO
END DO
DO
DELTAMULTIPLY(Graph%, 2, 100.0/Nd!)
Ytitle$ = "% Activation"
Graph_name$(1, 24) = "Dopant Activation %"
EXIT 3 LEVELS
END DO
END DO
REPEAT
END DO
END DO
DO
IF INSTR(Temp_type$(Graph%), "T") <> 0 THEN EXIT 1 LEVELS
Xtitle$ = "Temperature in Kelvin"
EXIT 2 LEVELS
END DO
END DO
DO
Xtitle$ = "Time in Seconds"
END DO
END DO
END DO
KILL "\BUCKET"
RETURN, 160
REM
DO
IF Graph_chosen% = 6 OR Graph_chosen% = 7 THEN EXIT 1 LEVELS
278

17030 CLEARSCREEN(2)
17040 SET CURSOR 13,1 : COLOR 7,0
17050 PRINT "The graph chosen has not been implemented (17000)."
17060 STOP
17070 REPEAT
17210 DO
17220 DIRECTORY("\CG",0,Graph%,Error%)
17230 IF Error% <> 0 THEN EXIT TO,160
17235 RUNTYPE(Graph%,1)
17240 END DO
17250 DO
17260 CLEARSCREEN(2)
17270 SET CURSOR 13,6 : COLOR 2,0,0
17280 INPUT "Enter the Bias Voltage. ",Dummy$
17285 Bias_sought! = VQL(Dummy$)
17290 END DO
17500 FOR NX = 1 TO 2
17510 IF INSTR(Directory$(Graph%,NX),"NO") <> 0 THEN EXIT 1 LEVELS
17512 DATAPRINT
17515 IF NX = 2 THEN Max_graph_points%(Graph%) = Max_graph_points%(Graph%) + 1
17520 IF Graph_chosen% = 6 THEN ID% = 4 ELSE ID% = 5
17530 DATARETRIEVAL(ID%,Graph%,N%,Error%,0)
17540 IF Error% = 1 THEN EXIT TO,17210
17550 NEXT N%
17700 DO
17710 IF Graph_chosen% <> 6 THEN EXIT 1 LEVELS
17720 DELTAMULTIPLY(Graph%,2,1E+09)
17730 Ytitle$ = "Capacitance in nF"
17735 Graph_name$(1,6) = "Capacitance vs."
17740 END DO
17750 DO
17760 IF Graph_chosen% <> 7 THEN EXIT 1 LEVELS
17770 DELTAMULTIPLY(Graph%,2,1000.0)
17775 Graph_name$(1,7) = "Conductance vs."
17780 Ytitle$ = "Conductance in mMhos"
17790 END DO
17800 DO
17810 Position% = INSTR(Bias_found$,"=") + 1
17820 Bias%(Graph%) = MID$(Bias_found$,Position%)
17830 DO
17840 IF INSTR(Temp_type$(Graph%),"T") <> 0 THEN EXIT 1 LEVELS
17850 Xtitle$ = "Temperature in Kelvin"
17860 Graph_name$(1,Graph_chosen%) = Graph_name$(1,Graph_chosen%) + "Temperature"
17870 EXIT 2 LEVELS
17880 REPEAT
17890 DO
Xtitle$ = "Time in Seconds"
Graph_name$(1, Graph_chosen%) = Graph_name$(1, Graph_chosen%) + "Time"
END DO
END DO
RETURN, 160
REM
DO
CLEARSCREEN(2)
RESTORE, 62000
FOR N% = 4 TO 11
READ Hand_info$(N%)
COLOR 11, 0 : LOCATE N%, 1
PRINT Hand_info$(N%)
NEXT N%
END DO
LOCATE 4, 1, : COLOR 0, 7 : PRINT "MAKE SELECTION"
LOCATE 4, 1
A$ = INKEY$ : IF A$ = "" THEN GOTO 20220
IF ASC(RIGHT$(A$, 1)) = 13 THEN GOTO 20300
IF ASC(RIGHT$(A$, 1)) = 72 THEN GOSUB 20400
IF ASC(RIGHT$(A$, 1)) = 80 THEN GOSUB 20500
GOTO 20220
ON CSRLIN -3 GOSUB 30500
PRINT CSRLIN : STOP
RETURN, 20220
COLOR 11, 0 : LOCATE Position%, 1 : PRINT Hand_info$(Position%)
IF Position% < 5 THEN Position% = 13
COLOR 0, 7 : LOCATE Position% - 1, 1 : PRINT Hand_info$(Position% - 1)
RETURN, 20220
COLOR 11, 0 : LOCATE Position%, 1 : PRINT Hand_info$(Position%)
IF Position% > 10 THEN Position% = 3
COLOR 0, 7 : LOCATE Position% + 1, 1 : PRINT Hand_info$(Position% + 1)
RETURN, 20220
GOSUB 22500
DO
LINE INPUT "Enter the TITLE of the graph (up to 60 letters) ": Graph_name$(1, 1)
Position% = INSTR(Graph_name$(1, 1), ",") - 1
Graph_name$(1, 1) = MID$(Graph_name$(1, 1), 1, Position%)
CLEARSCREEN(20)
END DO
RETURN, 20200
20700 GOSUB 22500
20710 DO
20720  LINE INPUT "Enter the X AXIS title (up to 60 letters) ";Xtitle$
20725  Position% = INSTR(Xtitle$," ") - 1
20727  Xtitle$ = MID$(Xtitle$,1,Position%)
20730  CLEARSCREEN(20)
20740  END DO
20750  RETURN,20200
20800 GOSUB 22500
20810 DO
20820  LINE INPUT "Enter the Y AXIS title (up to 20 letters) ";Ytitle$
20825  Position% = INSTR(Ytitle$," ") - 1
20827  Ytitle$ = MID$(Ytitle$,1,Position%)
20830  CLEARSCREEN(20)
20840  END DO
20850  RETURN,20200
20900 GOSUB 22500
20920 PRINT "Enter the X,Y data point (RETURN) 
20930  FOR NZ = Minloop% TO 1200
20940  SET CURSOR 21,43 : PRINT SPC(34)
20950  SET CURSOR 21,43
20955  DO
20960  LINE INPUT V$
20961  IF V$ = "" THEN EXIT TO,21005
20962  Pointer% = INSTR(V$",")
20963  IF Pointer% = 0 THEN EXIT TO,21005
20964  END DO
20965  Graph!(Graph%,1,N%) = VAL(MID$(V$,1,Pointer%))
20966  Graph!(Graph%,2,N%) = VAL(MID$(V$,Pointer%+1))
20985  SET CURSOR 21,43 : PRINT SPC(34)
20990  CLEARSCREEN(22)
21000  NEXT N%
21005  Minloop% = N%
21007  Max_graph_points%(Graph%) = N%-1
21010  CLEARSCREEN (21)
21020  RETURN,20200
21050  RETURN
21100 GOSUB 22500
21110 SET CURSOR 3,40 : COLOR 5,0 : PRINT "Changing X-Y data"
21120  Minloop1% = 1
21130 FOR N% = Minloop1% TO Max_graph_points%(Graph%)
21140  COLOR 2,0 : SET CURSOR N% + 5 - Minloop1% ,45
21150  PRINT Graph!(Graph%,1,N%) "," Graph!(Graph%,2,N%)
21155  IF N%+5-Minloop1% ) 17 THEN EXIT 1 LEVELS
21160  NEXT N%
21170  Maxloop% = N% - 1
21200 FOR N% = Minloop1% TO Maxloop%
COLOR 0,2 : SET CURSOR N%+5-Minloop1%, 45
PRINT Graph!(Graph%, 1, N%) "," Graph!(Graph%, 2, N%)
COLOR 0,0 : CLEARSSCREEN(20)
SET CURSOR N%+5-Minloop1%, 45
COLOR 0,2 : LINE INPUT V$
COLOR 0,0 : CLEARSSCREEN(20)
COLOR 0,0 : Position% = INSTR(V$, ",") + 1
IF Position% = 1 THEN EXIT TO, 21400
Graph!(Graph%, 1, N%) = VAL(MID$(V$, 1, Position%))
Graph!(Graph%, 2, N%) = VAL(MID$(V$, Position%))
COLOR 2,0 : SET CURSOR N%+5-Minloop1%, 45
PRINT SPC(35)
COLOR 2,0 : SET CURSOR N%+5-Minloop1%, 45
PRINT Graph!(Graph%, 1, N%) "," Graph!(Graph%, 2, N%)
CLEARSSCREEN(20)
NEXT N%
DO
IF Maxloop% => Max_graph_points%(Graph%) THEN EXIT TO, 21500
FOR N% = 4 TO 23
COLOR 7,0 : SET CURSOR N%, 30
PRINT SPC(50)
NEXT N%
NEXT N%
RETURN, 20200
DO
Choice% = 1: Saveflag=1
SAVEDATA(Graph%)
END DO
SET CURSOR 13, 40 : PRINT SPC(40)
CLEARSCREEN(16)
CLOSE #1
RETURN, 20200
COLOR 7,0
CLEARSCREEN(2)
RETURN, 160
REM
COLOR 11,0 : LOCATE CSRLIN, 1 : PRINT Hand_info$(CSRLIN)
DO
COLOR 0,0 : CLEARSCREEN(12)
SET CURSOR 13, 40 : PRINT SPC(39)
SET CURSOR 21, 6 : COLOR 2,0
22590 RETURN
22599 STOP
22600 GOSUB 22500
22610 LOADDATA(Graph%)
22870 RETURN,20200
22999 STOP
23000 REM
23020 DO
23030 IF Graph_chosen% <> 10 OR Graph_chosen% <> 2 THEN Directory$(Graph%,2) = "NO"
23040 END DO
23050 DO
23055 IF Graph_chosen% = 10 OR Graph_chosen% = 2 THEN ID% = 0 ELSE ID% = 1
23060 DIRECTORY("\CGV",ID%,Graph%,Error%)
23070 END DO
23080 DO
23095 IF Graph_chosen% = 10 OR Graph_chosen% = 2 THEN EXIT 1 LEVELS
23100 CLEARSCREEN(2)
23110 SET CURSOR 13,6 : COLOR 2,0,0
23120 INPUT "Enter the area of the device in cm squared.
",Area$
23130 Area! =VAL(Area$)
23140 IF Area! = 0 THEN EXIT TO,23650
23150 END DO
23160 DO
23170 IF Graph_chosen% <> 10 AND Graph_chosen% <> 6 THEN EXIT 1 LEVELS
23175 CLEARSCREEN(15)
23180 SET CURSOR 15,6 : INPUT "Enter the type of material you have tested (GaAs, Si)";Name$
23185 Name$ = UPPER$(Name$)
23190 IF INSTR(Name$,"GAAS") = 0 AND INSTR(Name$,"SI") = 0 THEN EXIT TO,23671
23195 IF INSTR(Name$,"G") <> 0 THEN RESTORE,61000 ELSE RESTORE,61500
23200 READ Nc!,Dummy!
23205 E_sub_S! = E_sub_O! * Dummy!
23210 IF Graph_chosen% = 9 THEN EXIT 1 LEVELS
23215 CLEARSCREEN(17)
23220 SET CURSOR 17,6 : INPUT "Enter the donor density (Nd) ";Nd$
23225 Nd! = VAL(Nd$)
23230 IF Nd! = 0 THEN GOTO 23682
23687 END DO
23689 DATAPRINT
23695 DO
23696 DO
23697 IF Graph_chosen% < 10 AND Graph_chosen% <> 2 THEN EXIT 1 LEVELS
23698 IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp_type$(Graph%) = "R"
23699 IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Run_type% = 1 ELSE Run_type% = 2
23700 EXIT 2 LEVELS
23701 REPEAT
23702 Run_type% = 0
23703 END DO
23704 File% = 1
23705 DO
23710 IF Graph_chosen% = 5 THEN ID% = 3 ELSE ID% = 2
23720 DATA RETRIEVAL(ID%, Graph%, File%, Error%, Run_type%)
23730 IF Error% <> 0 AND Graph_chosen% = 10 THEN EXIT 1 LEVELS
23740 IF Error% <> 0 AND Graph_chosen% = 2 THEN EXIT 1 LEVELS
23745 IF Error% <> 0 THEN EXIT TD
23750 END DO
23800 DO
23810 DO
23820 IF Graph_chosen% <> 4 THEN EXIT 1 LEVELS
23830 DELTAMULTIPLY(Graph%, 2, 1E+09)
23840 Ytitle$ = "Cap in nF"
23850 EXIT 2 LEVELS
23860 REPEAT
23870 DO
23880 IF Graph_chosen% <> 5 THEN EXIT 1 LEVELS
23890 DELTAMULTIPLY(Graph%, 2, 1E+03)
23900 Ytitle$ = "Cond in mMhos"
23910 EXIT 2 LEVELS
23920 REPEAT
23930 DO
23940 IF Graph_chosen% <> 8 AND Graph_chosen% <> 1 THEN EXIT 1 LEVELS
23945 IF Graph_chosen% <> 8 THEN Power! = 3.000 ELSE Power! = 2.0000
23950 INVERSEPOWER(Graph%, 2, 1.0E+09/Area!, Power!)
23960 IF Graph_chosen% = 8 THEN Ytitle$ = "1/C**2 C in nF" ELSE Ytitle$ = "1/C**3 in nF"
23970 EXIT 2 LEVELS
23980 REPEAT
23990 DO
24000 IF Graph_chosen% <> 9 THEN EXIT 1 LEVELS
24010 INVERSEPOWER(Graph%, 2, 1.0/Area!, 2.000)
24015 MinX! = 1E+37 ; MaxX! = -1E+37
MinY! = 1E+37 : MaxY! = -1E+37

FOR N% = 1 TO Max_graph_points%(Graph%) - 1
    Dummy! = E_sub_S! * SQR(Graph!(Graph%,2,N%))
    Slope! = (Graph!(Graph%,2,N%+1) - Graph!(Graph%,2,N%)) / (Graph!(Graph%,1,N%+1) - Graph!(Graph%,1,N%))
    Graph!(Graph%,1,N%) = Dummy!
    IF Slope! = 0 THEN GOTO 24120
    Graph!(Graph%,2,N%) = (-2/(O! * E_sub_S!)) * 1/Slope!
NEXT N%
Max_graph_points%(Graph%) = Max_graph_points%(Graph%) - 1

Ytitle$ = "Dopant Density in 1/cm**3 X 1e+10"
Xtitle$ = "Depth in microns"

DELTAMULTIPLY(Graph%,2,1E-10)
DELTAMULTIPLY(Graph%,1,1E+04)
EXIT 2 LEVELS

REPEAT
DO
    IF Graph_chosen% <> 10 AND Graph_chosen% <> 2 THEN EXIT 1 LEVELS
    Counter% = Counter% + 1
    IF Graph_chosen% <> 10 THEN Power! = 3.0000 ELSE Power! = 2.0000
    INVERSEPOWDER(Graph%,2,1/Area!,Power!)
    DELTAMULTIPLY(Graph%,2,1E-15)
    IF Counter% = 1 THEN ID% = 0 ELSE ID% = 1
    LEASTSQUARES(Graph%,Min!,Max!,Dummy!,Slope!,X_int!,Y_int!,ID%,Error_LS%)
    IF Counter% = 1 THEN GRange$(Graph%) = STR$(Min!) + "," + STR$(Max!)
    IF Error_LS% = 1 THEN RUN,200
    Position% = INSTR(Temp_found$,";") + 1
    Temp! = VAL(MID$(Temp_found$,Position%))
    Y! = X_int! + ((K! * Temp!) / Q!) * LOG(Nc!/Nd!)
    DO
        IF INSTR(Temp_type$(Graph%),"T") <> 0 THEN EXIT 1 LEVELS
        X! = Temp!
        EXIT 2 LEVELS
        REPEAT
        DO
            Position% = INSTR(Time_found$,"=") + 1
            Temp! = VAL(MID$(Time_found$,Position%))
            X! = VAL(MID$(Time_found$,Position%))
        END DO
    END DO
    END DO
    OPEN "\BUCKET" FOR APPEND AS #3
    PRINT #3,X!","Y!
    CLOSE #3 : CLOSE #File%
    IF INSTR(Directory$(Graph%,File%),"NO") <> 0 THEN
N EXIT 1 LEVELS

24334 OPEN "A:\CGV." + Directory$(Graph%, File%) FOR INPUT AS #File%
24335 IF INSTR(Temp_type$(Graph%), "T") = 0 THEN Temp_type$(Graph%) = "LH"
24336 Max_graph_points% (Graph%) = 1
24340 IF Error% = 0 THEN EXIT TO, 23705
24401 DO
24402 File% = File% + 1
24403 IF File% = 3 OR INSTR(Directory$(Graph%, 2), "NO") <> 0 THEN EXIT 1 LEVELS
24404 CLOSE #1 : Error% = 0
24406 IF INSTR(Temp_type$(Graph%), "T") = 0 THEN Temp_type$(Graph%) = "R"
24407 EXIT TO, 23705
24408 REPEAT
24410 DO
24430 Max_graph_points% (Graph%) = Counter%
24440 GETDATA(Graph%)
24450 Graph_name$(1, 10) = "Barrier Height (Capacitive)"
24470 Ytitle$ = "Barrier in Volts"
24480 IF INSTR(Temp_type$(Graph%), "T") THEN Xtitle$ = "Time in Seconds" ELSE Xtitle$ = "Temperature in Kelvin"
24485 EXIT TO, 3000
24490 END DO
24500 END DO
24510 END DO
24520 DO
24530 IF Graph_chosen% = 9 THEN EXIT TO, 24560
24555 Xtitle$ = "Voltage in Volts"
24560 DO
24570 IF Temp_type$(Graph%) = "T" THEN EXIT 1 LEVELS
24580 Position% = INSTR(Temp_found$, ";") + 1
24590 Temp$(Graph%) = MID$(Temp_found$, Position%)
24600 EXIT 2 LEVELS
24610 REPEAT
24615 DO
24620 Position% = INSTR(Time_found$, "=") + 1
24630 Tyme$(Graph%) = MID$(Time_found$, Position%)
24640 END DO
24650 EXIT TO, 160
24660 END DO
24900 RETURN, 160
25000 REM
25210 DO
25220 DIRECTORY("\RES", 0, Graph%, Error%)
25230 IF Error% <> 0 THEN EXIT TO, 160
25235 RUN TYPE(Graph%, 1)
25240 END DO
25250 DO
CLEARSCREEN(2)
SET CURSOR 13,6: COLOR 2,0,0
INPUT "Enter the Current Bias (mA).", Dummy$
Bias_sought! = VAL(Dummy$)
SET CURSOR 15,6: COLOR 2,0,0
INPUT "Enter the 4-pt. Correction Factor(CF; pg. 31 of Sze).", Dummy$
CF = VAL(Dummy$)
SET CURSOR 17,6: COLOR 2,0,0
INPUT "Enter the Sample thickness.", Dummy$
Thick$ = VAL(Dummy$)
END DO
FOR N% = 1 TO 2
IF INSTR(Directory$(Graph%,N%),"NO") <> 0 THEN EXIT 1 LEVELS
DATAPRINT
IF N% = 2 THEN Max_graph_points%(Graph%) = Max_graph_points%(Graph%) + 1
IDZ = 4
DATARETRIEVGL(IDZ,GvaphZ,NZ,E»v0rZ,0)
IF Error% = 1 THEN EXIT TO,25210
NEXT N%
DELTMULTIPLY(Graph%,2,CF*Thick!/(.001*Bias_sought!))
Ytitle$ = "Resistivity in ohm*cms"
Graph_name$(1,6) = "Resistivity vs."
END DO
DO
Position% = INSTR(Bias_found$,"=") + 1
Bias$(Graph%) = MID$(Bias_found$,Position%)
DO
IF INSTR(Temp_type$(Graph%),"T") <> 0 THEN EXIT 1 LEVELS
Xtitle$ = "Temperature in Kelvin"
Graph_name$(1,Graph_chosen%) = Graph_name$(1,Graph_chosen%) + "Temperature"
EXIT 2 LEVELS
REPEAT
DO
Xtitle$ = "Time in Seconds"
Graph_name$(1,Graph_chosen%) = Graph_name$(1,Graph_chosen%) + "Time"
END DO
END DO
RETURN,160
STOP
IF ERR = 723 AND ERL=6205 THEN Minloop% = M% + 1: RESUME,6045
IF ERR = 1 AND ERL = 5520 THEN GOSUB 55500: RESUME,3
000
55030 IF ERR = 2 AND ERL = 5520 THEN GOSUB 55500: RESUME, 3
000
55100 REM
55200 PRINT "The main program is bombing. This is error "
55210 ";ERR;" from line ";ERL
55210 STOP
55499 STOP
55500 REM
55510 VIEW : SCREEN 0,0,0 : COLOR 7,0 : SET CURSOR 13,6
55520 PRINT "The Maximum minus Minimum points are beyond
55530 BetterBasic's Accuracy."
55530 TIMEDELAY(5)
55540 RETURN
60200 REM
60210 DATA 29
60220 DATA "Hand Graph Entry","Capacitance vs. Time","Conduc-
tance vs. Time","Capacitance vs. Voltage {4280}"","Conduc-
tance vs. Voltage","Capacitance vs. Temp. (Time)","Conduct-
ance vs. Temp. (Time)"
60230 DATA "1/C**2 vs. Voltage {4280}"","Dopant Profile {42-
80}"","Barrier vs. Temp (Time) 1/C**2","DLTS Spectrum"
60240 DATA "Current vs. Voltage","Capacitance vs. Voltage
{4140}","1/C**2 vs. Voltage {4140}"","log(I) vs. Voltage","l-

n(I) vs. qV/kT","ln(If/T**2) vs. q/kT"
60250 DATA "Isat vs. Temp (Time)","Ideality Factor vs Temp
(Time)","Barrier vs Temp. (Time) {4140}""Resistivity vs. Temp. (Time)","Mobility vs. Te-
mp. (Time)","Carrier Conc. vs. Temp. (Time)","Activation %
vs. Temp. (Time)"
60270 DATA "1/C**3 vs. Voltage {4280}"","Barrier vs. Temp (Time) 1/C**3","ln(Isat) vs. Temp (Time)","4-pt. Resistivit-
y vs. Temp(Time)"
60500 REM
61000 DATA 4.7E+17,13.1
61500 DATA 2.8E+19,11.9
62000 DATA "TITLE","X-AXIS TITLE","Y-AXIS TITLE","X-Y DATA
INPUT","CHANGE X-Y DATA","SAVE","LOAD","GO TO MAINKEY"

ENDFILE
Appendix B: Model Simulation Software

The flowchart for the general simulation software is given in Fig. 50 and is self explanatory. The actual documented software listing follows for both the DLTS and Capacitance-frequency simulations.
Physical constants & network parameters

Set time and temperature for DLTS
or frequency for C-f

Calculate C

Find δC for DLTS

Load into files for δC vs. T or C vs. f

Exit to graphics

Figure 50. Flowchart for general simulation software.
SOURCE
PRECISION= 7
AUTODEF=ON
OPTION BASE=0
ERL=OFF
ERRORMODE=LOCAL
RESUME=LINE
FORMODE=BB
SCOPE=ON
PROC8=0
REAL: A,B,C,D,E,F,G,R1
REAL: R2,RD,C1,CD,W,TP,TM,ET
REAL: V,Q,K,X1,X2,C0,CT1,CT2
REAL: E1,E2
REAL ARRAY(2,5): CM
REAL: M,T
REAL ARRAY(20,20): CZ
REAL: Con,Cap,Peak,N

' MAIN Program:

10 REM *THIS PROGRAM PRODUCES A DLTS PLOT USING COMPONENTS DERIVED FROM ANALYSIS
20 REM
30 REM *THESE ARE THE CONSTANS USED FOR CALCULATION
40 REM
50 Q=1.6E-19;V=1E12;K=1.38E-23;W=6.2836;Peak=0
60 REM
70 REM *THESE ARE THE NETWORK COMPONENTS: C0, RD-DIODE; C1, R1-LAYER; R2-SHUNT
80 REM
90 RD=1;C0=2;R1=1;C1=2;R2=1
100 REM
110 REM THESE ARE THE EL2 TRAP PARAMETERS
120 REM
130 E2=.77;CT2=20E-12
140 REM
150 REM *THIS SECTION PUTS THE DATA IN A FORM WHICH CAN BE USED BY MEDUSA
160 REM
170 OPEN "\ERIC\DATA#" FOR OUTPUT AS #1
180 PRINT #1 "DLTS SPECTRUM"
190 PRINT #1 "TEMP in K"
200 PRINT #1 "DLTS in pFs"
210 PRINT #1 ","
220 PRINT #1 "X"
230 PRINT #1 "X"
240 PRINT #1 "X"
250 PRINT #1 "X"
260 PRINT #1 "X"
270 PRINT #1 "X"
280 PRINT #1 "X"
290 PRINT #1 "O"
300 PRINT #1 "O"
310 PRINT #1 "O"
320 PRINT #1 "O"
330 PRINT #1 "O"
340 PRINT #1 "O"
350 PRINT #1 "61"
360 REM
370 REM *THIS SECTION SIMULATES THE DLTS
380 REM
390 FOR TP=100 TO 400 STEP 5
400 FOR M=1 TO 5
410 IF Q*E2/K/TP>45 THEN X2=1E-8:GOTO 430
420 X2=V*EXP(-Q*E2/K/TP)
430 IF Q*E1/K/TP>45 THEN X1=1E-8:GOTO 450
440 X1=V*EXP(-Q*E1/K/TP)
450 TM=.008*M
460 IF X1*TM<6 THEN X1=1000
470 IF X2*TM<6 THEN X2=1000
480 CD=CO-CT1*EXP(-X1*TM)-CT2*EXP(-X2*TM)
490 A=R1/(1+W^2*C1^2*R1^2):B=W*C1*R1^2/(1+W^2*C1^2*R1^2)
500 C=RD/(1+W^2*CD^2*RD^2):D=W*CD*RD^2/(1+W^2*CD^2*R1^2)
510 E=A+C:F=B+D
520 CZ(M,1)=1/W/F+2*E/W/F/R2+E^2/W/F/R2^2+F/W/R2^2
530 NEXT M
540 PRINT #1 " TP", "1E12*(CZ(1,1)-CZ(5,1))"
550 NEXT TP
560 CLOSE #1
570 STOP

ENDFILE
SOURCE
PRECISION= 7
AUTODEF=ON
OPTION BASE=0
ERL=OFF
ERRORMODE=LOCAL
RESUME=LINE
FORMODE=BB
SCOPE=ON
PROCS=0
REAL: A, B, C, D, E, F, G, R1
REAL: R2, RD, C1, CD, W, TP, TM, ET
REAL: V, Q, K, X1, X2, C0, CT1, CT2
REAL: E1, E2
REAL ARRAY(2, 5): CM
REAL: M, T
REAL ARRAY(20, 20): C1
REAL: Con, Cap, Peak, N

' MAIN Program:

10 REM *THIS PROGRAM CALCULATES CAPACITANCE FREQUENCY DISPERSION
20 REM
30 REM
40 REM *THESE ARE THE PHYSICAL CONSTANTS
50 REM
60 Q=1.6E-19; V=1E12; K=1.38E-23; Peak=0
70 REM
80 REM *THESE ARE THE NETWORK VALUES: DIODE—C0, RD; LAYER—C1, R1; SHUNT—R2
90 REM
100 C0=150E-12; C1=2E-12; R1=0; R2=1E7; RD=1E7
110 REM
120 REM *THESE ARE THE EL2 TRAP PARAMETERS
130 REM
140 E2=.77; CT2=20E-12
150 REM
160 REM *THIS SETS THE DATA FILE IN A FORM FOR MEDUSA
170 REM
180 OPEN "\ERIC\DATD1" FOR OUTPUT AS #1
190 PRINT #1 "CAP vs. FREQ"
200 PRINT #1 "FREQ in Hz"
210 PRINT #1 "CAP in pFs"
220 PRINT #1 ";:
230 PRINT #1 "X"
240 PRINT #1 "X"
250 PRINT #1 "X"
260 PRINT #1 "X"
270 PRINT #1 "X"
280 PRINT #1 "X"
290 PRINT #1 "X"
300 PRINT #1 " O"
310 PRINT #1 " O"
320 PRINT #1 " O"
330 PRINT #1 " O"
340 PRINT #1 " O"
350 PRINT #1 " O"
360 PRINT #1 " 101"
370 REM
380 REM *THIS FINDS CAPACITANCE DISPERSION
390 REM
400 FOR N=1 TO 501 STEP 5
410 W=2*3.14159*N*100
420 CD=C0
430 A=R1/(1+W^2*C1^2*R1^2):B=W*C1*R1^2/(1+W^2*C1^2*R1^2)
450 E=A+C:F=B+D
460 Con=((E+R2)^2+F^2)/(R2*E^2+E*R2^2+R2*F^2)
470 Cap=1/W/F+2*E/W/F/R2+E^2/W/F/R2^2+F/W/R2^2
480 PRINT #1 " ";W/2/3.14159;" , ";1E12*Cap
490 NEXT N
500 CLOSE #1
510 STOP
ENDFILE
The vita has been removed from the scanned document