MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0355064 5

ORNL/TM-11746

ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

A Multifunction Ion-Irradiation/Surface Modification Ultrahigh Vacuum Chamber System

R. A. Buhl

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

MANAGED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-11746

Metals and Ceramics Division

A Multifunction Ion-Irradiation/Surface Modification Ultrahigh Vacuum Chamber System

R. A. Buhl

Date Published: April 1991

NOTICE: This document contains information of a preliminary nature. It is subject to revision or correction and therefore does not represent a final report.

Prepared for
DOE Office of Basic Energy Sciences
KC 02 01 04 0

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6285
managed by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0355064 5

CONTENTS

ABSTRACT 1

1. INTRODUCTION 1

2. MODULAR CHAMBER DESCRIPTION 2

2.1. CONFIGURATION 2

2.2. LOADING/ VAPOR DEPOSITION CHAMBER 2

2.3. MULTIPLE ION-IRRADIATION CHAMBER 4

2.4. SURFACE CHARACTERIZATION CHAMBER 6

3. CHAMBER LATERAL TRANSLATION 8

4. SPECIMEN MANAGEMENT AND TRANSFER SYSTEM 8

4.1. STANDARD SPECIMEN HOLDER 8

4.2. TROLLEY 11

4.3. TRACK AND DRIVE SYSTEM 11

5. HEAT AND TEMPERATURE CONTROL SYSTEM 14

5.1. THERMOCOUPLE 14

5.2. ELECTRON GUN HEATERS 14

5.3. TEMPERATURE CONTROLLER AND FEEDBACK CIRCUIT 16

6. COMPUTER AND CONTROLLER SYSTEM 16

6.1. HARDWARE 16

6.1.1. Touch Screen 16

6.1.2. Talker 19

6.1.3. IEEE-488 Interface 19

6.1.4. Controller 19

6.1.5. Controller Modules 20

6.2 SOFTWARE 21

6.2.1. Shell Program 21

6.2.2. DOS Level Interrupt 23

6.2.3. Run Algorithm 23

6.2.3.1. Setup 23

6.2.3.2. Disk Control and Data Storage 24

6.2.3.3. Data Presentation and Options 25

6.2.3.4. Beam Integration Algorithm 25

6.2.3.5. Restart 25

7. REFERENCES 25

APPENDIX A. COMPUTER PROGRAM 29

APPENDIX B. OPERATING PROCEDURES 99

LIST OF FIGURES

Fig. 1. (Beam Exit Side View) Modular Ion-irradiation UHV Chamber System 3

Fig. 2. (END View) Modular Ion-irradiation UHV Chamber System 5

Fig. 3. (Beam Entrance Side View) Modular Ion-irradiation Chamber System 7

Fig. 4. Specimen Loading Apparatus 9

Fig. 5. Specimen Mounted in Holder 10

Fig. 6. Specimen Holder Trolley 12

Fig. 7. View Inside UHV Chambers 13

Fig. 8. Water-cooled Heater Assembly 15

Fig. 9. Data Acquisition System 17

Fig. 10. Touch Screw Display of Shell Program 18

Fig. 11. Touch Screw Display of Initialize Menu 22

Fig. 12. Data Displayed During RUN 26

Fig. 13. Drawing of Mind-Fariday Cup Assembly 27

Fig. 14. Menu-selected Mini-cup, Beam Distribution Program. 28

A MULTIFUNCTION ION-IRRADIATION/SURFACE MODIFICATION ULTRAHIGH VACUUM CHAMBER SYSTEM*

R. H. Buhl

ABSTRACT

A modular ultrahigh vacuum chamber system has been built to facilitate experiments at the Triple Ion Beam Accelerator Facility at the Oak Ridge National Laboratory (ORNL). The system currently consists of three chambers, linked by ultrahigh vacuum valves through which groups of specimens are transferred via a track-type shuttle mechanism. A loading/vapor deposition dual purpose chamber serves not only as a vacuum interlock for loading and unloading groups of specimens, but also as a dual-electron gun-type metal vapor deposition station. The ion-irradiation chamber, located adjacent to the loading chamber, is positioned at the intersection of beam lines from three Van de Graaff accelerators. An Auger surface analyzer with an in situ differentially pumped sputter-gun assembly is located in the third chamber. The chambers are mounted on a frame with "airfloat" pads beneath the feet so that a lateral translation permits the beam lines to be connected to other chambers in the system. The modular assembly concept in conjunction with the lateral mobility and the specimen transfer mechanism enhances the ability to accommodate future expansion in the system. A Hewlett Packard (HP) Vectra PC connected via an IEEE-488 interface to an upgraded HP 3852A controller provides data acquisition and control.

1. INTRODUCTION

The technique of ion irradiation of materials has evolved as a powerful tool not only in the study of surface and near-surface related physical properties but also in the modification of these properties. Some examples of the applications of this technique are the alteration of the electronic properties of semiconductors and the improvement of mechanical properties and corrosion resistance of machine parts and human joint implants.

At ORNL, a facility is dedicated to ongoing work in radiation-effects research, ion beam treatment, and ion beam characterization. This unique facility consists of three accelerators and is referred to as the Triple Ion Beam Irradiation Facility.[1] The accelerators are capable of generating positive ion beams in the energy range of 100 keV to 5 MeV. Beam lines from the three accelerators converge with an angle of 15°, between adjacent lines and intersect at the location of the new chamber (Sect. 2), making it possible to use one or any combination of

*Research sponsored by the Division of Materials Sciences, U.S. Department of Energy, under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

three simultaneous or sequential beams in the ion irradiation of an individual specimen. The new chamber system was built to extend the capabilities of the facility. This report describes the principles, operation, and detailed configuration of the system.

2. MODULAR CHAMBER DESCRIPTION

2.1 CONFIGURATION

The triple-beam chamber is located in the facility target room [Room 111, Bldg. 5500 (Fig.1)]. The system consists of three in-line ultrahigh vacuum (UHV) chambers, each designed to perform a specific task. The chambers, constructed of 6-in. OD vacuum grade stainless steel tubing, are linked by UHV-type valves. Groups of specimens are transferred on a track-trolley mechanism (described in Sect. 4) through these valves to any chamber in the system. For preliminary evacuation, each chamber has valved access to a common vacuum line. Services to the chambers (i.e., electricity, water, air, both digital and analog data lines, and control signal lines) were designed to accommodate lateral movement.

2.2 LOADING/ VAPOR DEPOSITION CHAMBER

The first chamber in the system is the loading/vapor deposition module [(*a*) in Fig. 1]. In the loading mode, the chamber acts as an interlock for introducing specimens into and removing them from the UHV environment that is maintained in the adjacent chambers. This technique has several advantages such as reducing the turn-around time involved in reestablishing UHV conditions in the main chamber, which is essential for an ion-irradiation experiment. If the ion-irradiation chamber is exposed to the atmosphere, it requires an extended period of time and/or a bake-out cycle to regain its vacuum integrity. Maintaining UHV conditions in the ion-irradiation chamber also eliminates poisoning of the dispenser cathode heater assembly used to heat the specimens during elevated temperature irradiations. Exposing the dispenser cathode surface to air deteriorates its electron emission characteristics, which can only be reinstated by overheating the cathode for an extended period of time in a UHV environment. This conditioning process, shortens the operating life of the heater assembly.

The loading/vapor deposition chamber also houses a dual electron-gun (E-gun) type metal-vapor deposition system. The design concepts for this system are currently being updated to enhance the capabilities of the facility. The deposition system is scheduled for construction in the near future. Two independently controlled and operated evaporation sources will be mounted in the offset vertical section of the chamber. The E-guns are to be enclosed in individual compartments, with a port common to both, and aligned with the surface of the specimen. A shutter assembly, located in the port, controls which area of the specimen receives the vapor from each source. A computer, using feedback from evaporation rate controllers, will be used to vary the thickness of the layers. In this configuration it will be possible to deposit films of various thicknesses and compositions laterally across the specimen. In the post-deposition evaluation it will be possible to examine a variety of compositions from a single specimen by selecting the appropriate location along the specimen.

YP 11235

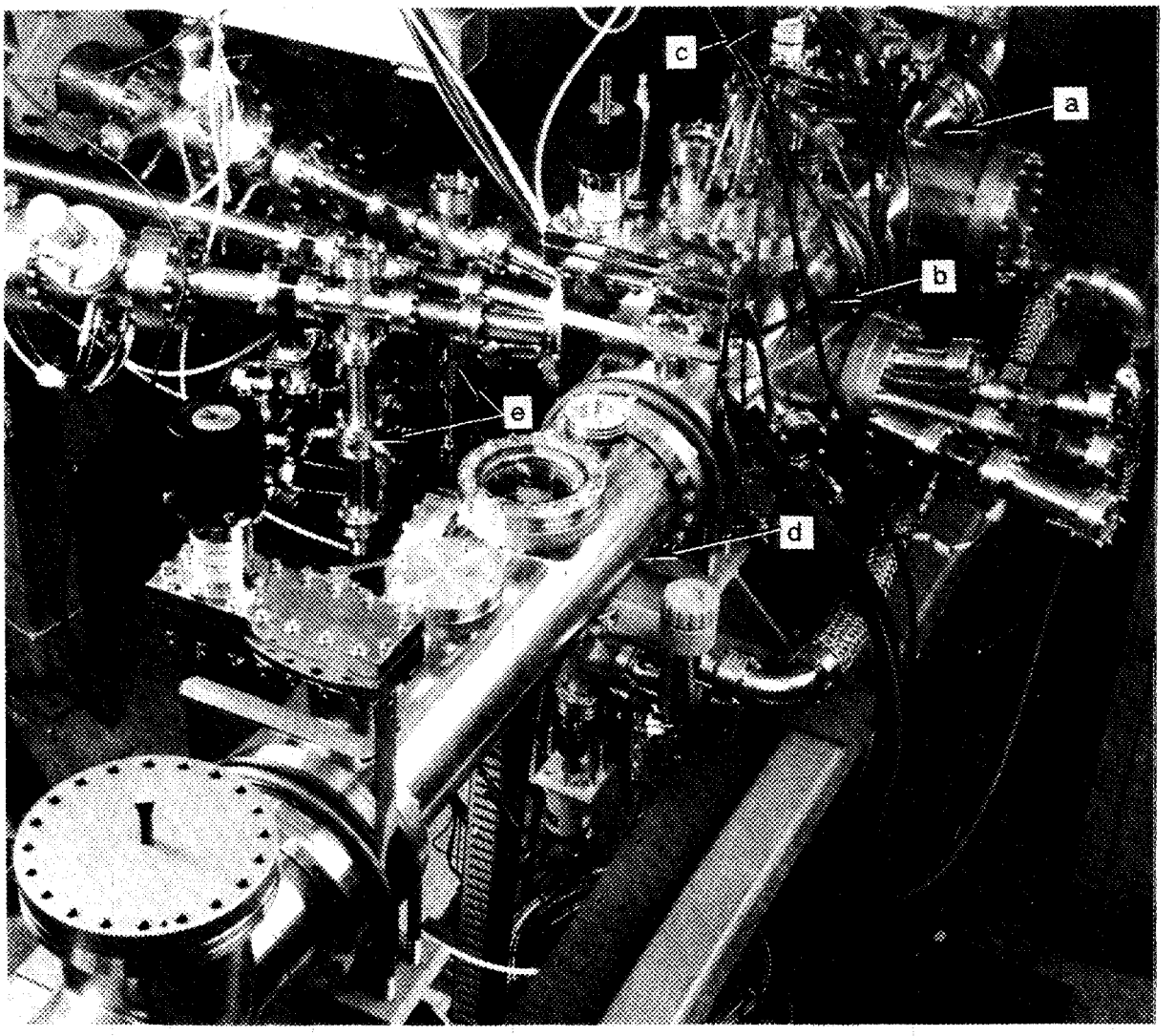

Fig. 1. Modular ion-irradiation UHV chamber system: (*a*) loading/vapor deposition module; (*b*) main ion-irradiation chamber; (*c*) pneumatic feedthrough for spring-loaded water-cooled electron boron heater assembly; (*d*) surface analysis chamber, and (*e*) external view of mini-Faraday cup assemblies.

The evaporation rate of each E-gun is to be measured by an oscillating quartz-crystal monitor. Changes in the oscillation frequency caused by metal being deposited on the quartz-crystal surface are measured by a rate controller and converted into thickness. An analog feedback signal from the rate controller regulates, according to a preset deposition rate, the current output from individual power supplies connected to the E-guns.

The vapor deposition fixture is attached to a UHV flange. Mechanically operated shutters will be used to cover the crystal sensors to extend their operating lifetime. Shutters will also be used to cover view ports when not in use.

The loading/vapor deposition chamber is routinely pumped from atmospheric pressure down to the 10^{-8} torr UHV range using a Balzer TPU 510 turbopump with 500 L/s pumping speed (calibrated for air). The turbopump is used in conjunction with a model DUO 16B backing pump [(*a*) in Fig. 2]. During specimen loading, the chamber is pumped into the 10^{-7} torr range prior to opening the valve connecting it to the adjacent chamber. The chamber roughing pressure is monitored using a convectron gauge, and the ultimate pressure is measured by a nude ion gauge. Both gauges are connected to a Granville Phillips model 307 vacuum gauge controller. The controller is interfaced to a computer via an RS-232-C line.

An Aerovac Mass Spectrometer computer-operated residual gas analyzer is attached to the system. It is used to identify gaseous components in the vacuum by their respective atomic mass numbers and to measure and record the partial pressures in the system.

2.3 MULTIPLE ION-IRRADIATION CHAMBER

The multiple ion-irradiation chamber, the principal environment for implementing ion irradiations, is the second chamber in the system and is normally located at the intersection of the three accelerator beam lines. However, as described in Sect. 3, lateral translation of the system allows the beam lines to be connected to other chambers in the system. In this configuration, ion beams can be customized in species composition and energy variations to cover a broad spectrum of projectile-target interactions.

The area of each beam at the point of intersection is 1.1 cm^2. The beams intersect horizontally 2.54 cm ahead of the center of the chamber.

The ion-irradiation chamber with its beam entrance and exit ports is shown as item *(b)* in Fig.1. On each beam line, the beam exit port terminates as a Faraday cup. With an ion beam present, a visual representation of its profile and relative strength is projected on a thick quartz-glass disk (coated with a conductive film) that is located inside a view port at the end of each Faraday cup. Two auxiliary viewing ports are symmetrically located above and below the Faraday cups, exposing the back sides of the trolley and specimen holder assemblies. This assists in confirming which specimen holder is at the intersection of the beams and/or if the specimen holder is blocking the beam from the Faraday cups (final beam-specimen alignment is performed on the beam entrance side of the chamber).

The beam entrance side of the chamber resembles the exit side with the exception of additional viewing ports. A quartz-glass disk with the specimen boundary lines etched in the

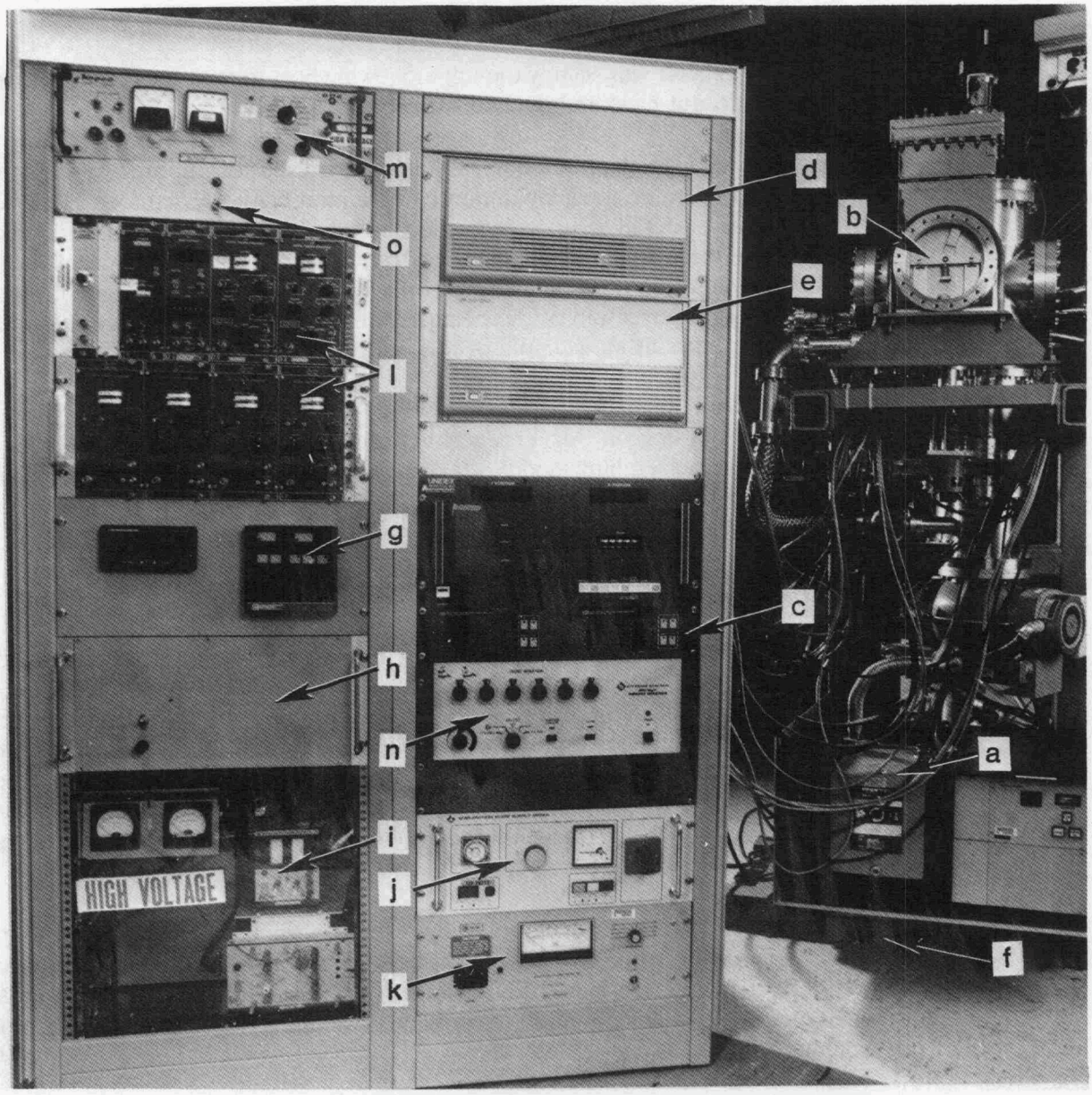

Fig. 2. (*a*) turbo pump parts; (*b*) loading port; (*c*) vacuum guage controller; (*d*) HP 3853 extender crate #1; (*e*) HP 3853 extender crate #2; (*f*) air-float pad frame; (*g*) temperature controller; (*h*) high-voltage translation box; (*i*) high-voltage crate; (*j*) titanium sublimation pump controller; (*k*) varian vac-ion pump controller; (*l*) current integrators; (*m*) high-voltage power supply; (*n*) stepper motor controller for trolley; and (*o*) power switch for high-voltage section.

surface is placed in a specimen holder and used to determine the exact beam-target alignment. Once this alignment has been established, the computer-interfaced positioning indexer will remember the exact placement of the other specimen holders. The positioning process will be automated in the future. (The specimen indexer is discussed in Sect. 4).

An infrared pyrometer will occupy the upper viewing port to monitor the surface temperature of specimens that do not have an attached surface thermocouple.

The top flange serves a dual function. Primarily it is used to support a spring-loaded water-cooled electron beam heater assembly mounted on a pneumatically actuated UHV feedthrough [(*c*)in Fig. 1]. This is detailed in Sect.5. Secondarily, the top flange affords access to the intersection of the three beam lines. Devices requiring configurations or environments not offered in the original design can be mounted on a UHV flange and suspended from the top of this chamber into the intersection of the beams. A liquid nitrogen cryostage has been designed and fabricated and serves as an example of this option.

The ion-irradiation chamber operates in the 10^{-8} to 10^{-9} torr UHV range. Vacuum for the chamber is produced using an Air Products AP-8 1000 L/s cryopump [(*a*) in Fig. 3] and is monitored by a Granville Phillips model 307 computer-interfaced vacuum gauge controller [(*c*) in Fig. 2].

2.4 SURFACE CHARACTERIZATION CHAMBER

The surface characterization chamber [(*b*) in Fig. 3] is presently being fabricated. In this chamber, Auger electron spectroscopy (AES) is used in conjunction with ion sputter-etching to profile surface/ near-surface regions of specimens both prior to and following treatments carried out in other chambers of the system. Ion sputter-etching is also used to remove oxides or other possible contaminants on the sample surface prior to ion irradiation or vapor deposition. A distinct advantage of this combination is that with the inter-chamber transfer system it is possible to eliminate the interim surface oxidation that would occur if the specimens had to be removed from the UHV environment to be analyzed.

The Auger electrons are spectrographically resolved using an electrostatic cylindrical mirror analyzer, that focuses electrons of a chosen energy (25N-2500 eV) through an aperture into an electron multiplier. A Perkin-Elmer PHI model 10-155 cylindrical Auger head mounted on a Thermionics model FB-150 manipulator (to provide positioning) is used in the design specification of this chamber.

A Perkin-Elmer PHI model 04-300 differentially pumped sputter-ion gun is focused on the specimen in the same area as the Auger head. This arrangement will simplify depth profile analysis. The sputter-ion gun can produce current densities > 8 $\mu A/cm^2$ at a working distance of 7 cm. The beam energy is continuously adjustable up to 4 keV. In the differentially pumped mode the ion gun will operate in the mid-10^{-8} torr UHV range. Auger analysis requires a UHV in the range of 10^{-10} torr or better at the specimen to reduce surface contamination. UHV operating techniques such as baking and inert gas backfilling must be employed to achieve and maintain this quality of vacuum. UHV pumping on this chamber is produced using a 400 L/s Varian model 912-7022 VacIon pump [(*c*) in Fig. 3] and a Vacuum Generator model VZST22

YP 11233

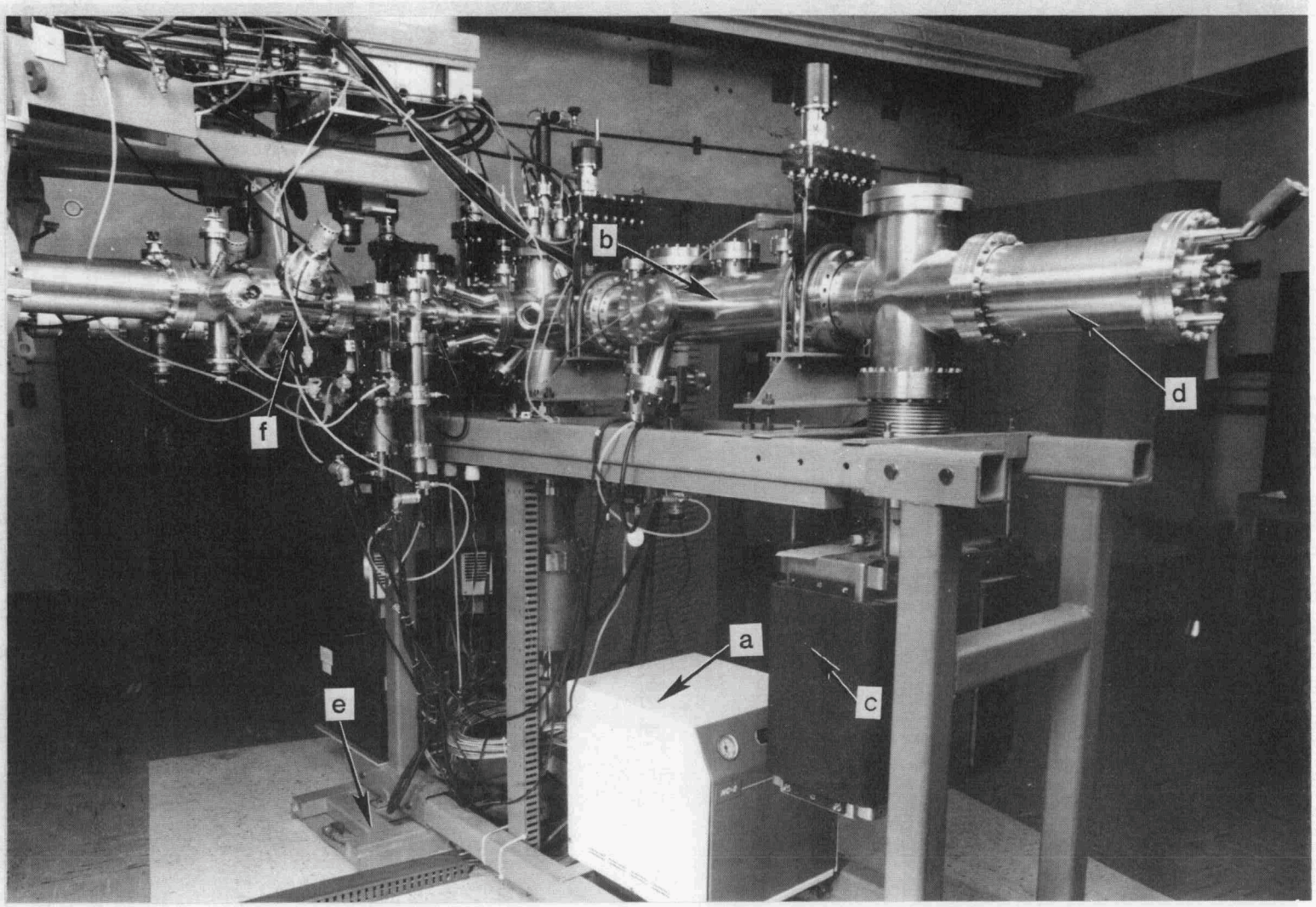

Fig. 3. Modular chamber system (*a*) cynpump for main chamber; (*b*) surface analysis chamber; (*c*) varian vac-ion pump; (*d*) lithinium sublimation pump; (*e*) air-float pad; and (*f*) beam profilometer.

titanium sublimation pump [(*d*) in Fig. 3] with a ~ SPS6A power supply. A computer-interfaced Granville Phillips model 307 vacuum indicator with a UHV electrometer uses a convectron gauge and a nude ion gauge sensor to monitor chamber pressure from atmosphere down into the mid-10^{-12} torr vacuum range.

3. CHAMBER LATERAL TRANSLATION

Lateral translation of the entire modular multiple-function chamber system may be carried out to provide beam line access for any chamber in the system. The chambers are mounted on a frame that is supported by two airfloat pads, one on each end of the structure [see (*e*) in Fig.3 and (*d*) in Fig. 2]. Each pad contains two 508 mm-diam air suspension cushions symmetrically placed for stability on a 533.4 mm by 1066.8 mm base. Built into each base is an air accumulation/ distribution system that individually regulates the air flow to each cushion. In this configuration, the stipulated lift capacity is 20 tons total, and the air consumption is 80-120 SCFM at 60 psi. Quick disconnects are provided at each base to accommodate a flexible hose manifold attached to the air-lift control system that operates the water-filled shielding door to the target room. A diverter valve, located behind the shielding door (when the door is open), determines which system (the door or the chambers) will receive air when the switch, mounted on the door, is closed.

4. SPECIMEN MANAGEMENT AND TRANSFER SYSTEM

Ion-irradiation specimens require a flat, highly polished metallographic quality surface that is free of scratches, defects, and contamination. These specimens must not be mishandled or abused as this could also produce unwanted deformation or surface scratches. Although a large variety of materials of different sizes, shapes, and compositions can be successfully irradiated in the chamber system, they all must have at least two things in common. They must be mounted in a holder capable of (1) fitting within the vacuum chamber and (2) being positioned in the beam. Specimen holders can either be attached to a trolley mechanism (discussed in Sect. 4.2) or suspended from the top flange of the chamber on a specially developed stage, the former being the more common technique.

4.1 STANDARD SPECIMEN HOLDER

The most common specimen is of the Transmission Electron Microscope (TEM) type, a disk 3 mm in diameter and 0.380-0.510 mm thick. The holder for these standard specimens consists of two parts, a thermalizer block and a faceplate. The faceplate has up to 9 individual TEM disk-size countersunk holes on a 3 by 3 array inscribed in a square area < ^ 1.1 cm^2. Using a loading device (Fig. 4), a small compressible ring of pure, fully annealed platinum wire is placed in the landing of the countersunk hole in the back side of the faceplate and a specimen is placed face-down on top of the ring. By mechanically compressing the sandwich in a clamping device, all specimens are forced into contact with the thermalizer block. The sandwich is held together by

ORNL-PHOTO 5953-77

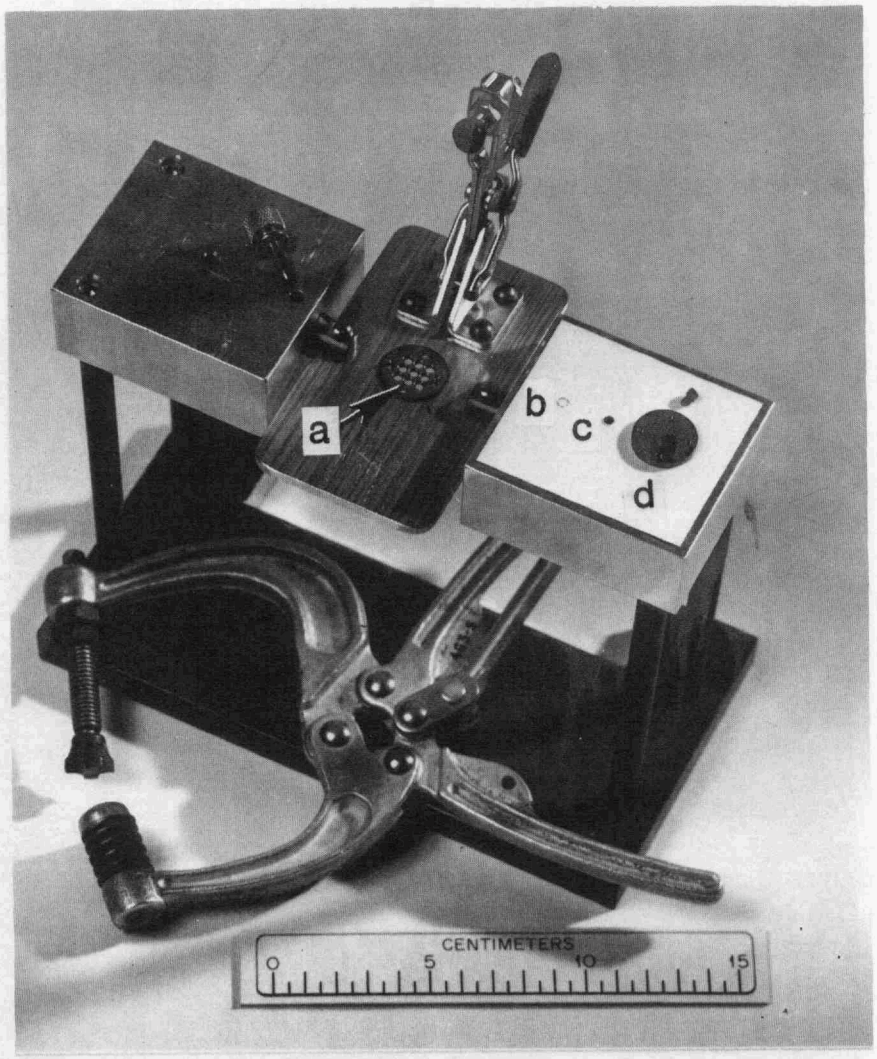

Fig. 4. Specimen holder and loading apparatus: (*a*) faceplate; (*b*) platinum wire gasket; (*c*) specimen; and (*d*) thermalizer block.

YP 6667

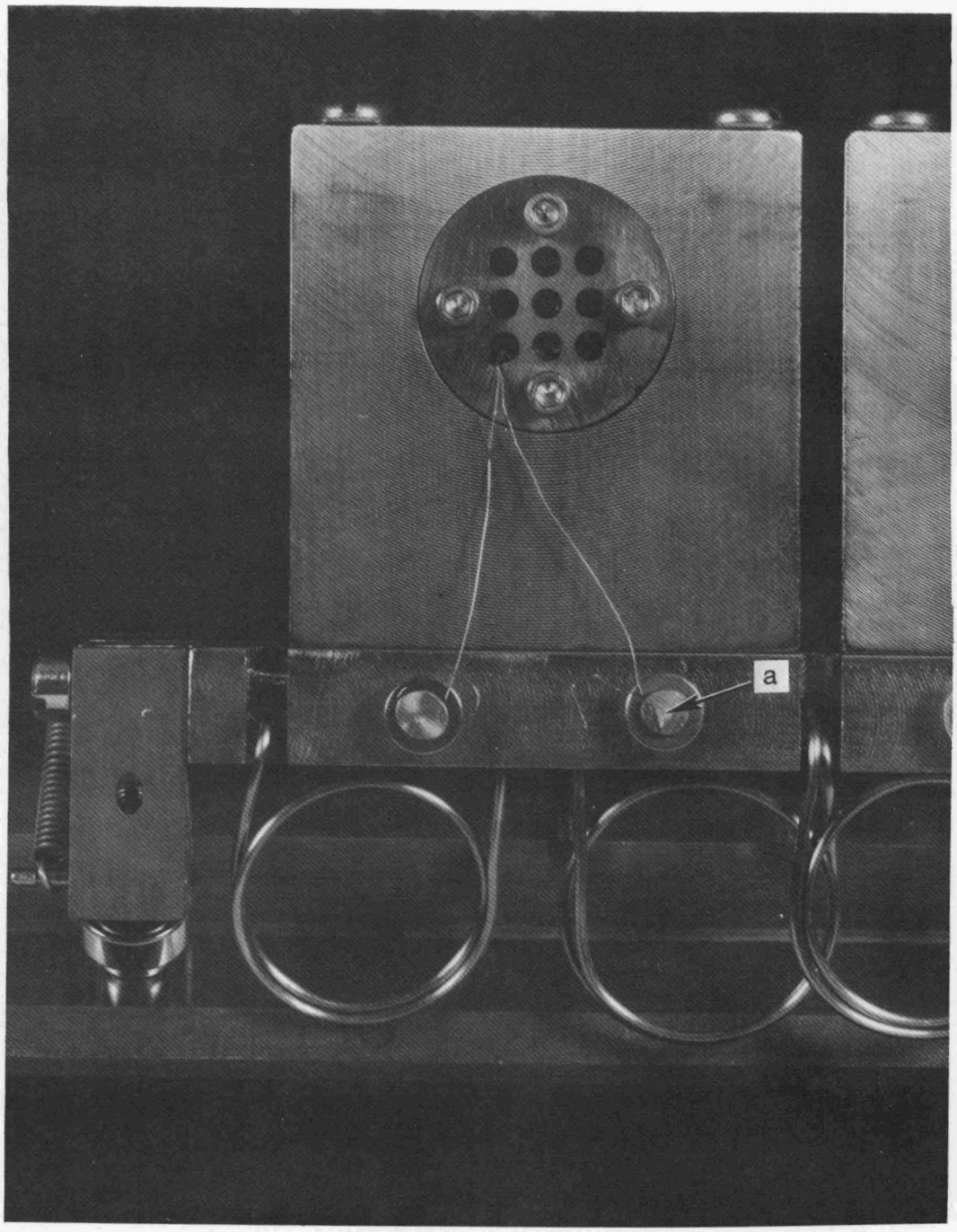

Fig. 5. View of a specimen holder mounted in a L-bracket located on the end station of the trolley. (*a*)Spring loaded thermocouple wire contact for surface mounted thermocouple.

four small screws applied through the faceplate into the thermalizer block. This arrangement produces a more uniform heat distribution during ion irradiation.

The sandwiched specimens are loaded into an L-shaped bracket (Fig. 5). They are held in place by two spring-loaded, sheathed thermocouples that pass from the trolley up through the L-shaped bracket into holes in the thermalizer block. The L-shaped bracket is attached to the trolley using a set of long screws. Figure 5 shows the brackets mounted on the trolley.

Other nonstandard specimen holders were designed to accommodate ion irradiations of specimens having different shapes and sizes. For example, a deep dish holder has a countersunk area 16 mm in diam by 3.5 mm deep. A faceplate with four small screws holds the specimen in position. The faceplate has a window 12.5 mm long by 6 mm wide.

4.2 TROLLEY

The trolley is the device that the specimen brackets are mounted on for transport through the chamber system. Figure 6 shows a cross section of the track and the end view of the trolley with the L-shaped specimen holder brackets in place. The trolley rides on three sets of ball bearings (one set in the center and a set at each end) with a set of spring-loaded bearings grasping the track at each end. This allows the end of the trolley to pass off the end of one track section of a chamber, pass through an open valve, and mate with the track in the adjacent chamber.

4.3 TRACK AND DRIVE SYSTEM

The track and drive system provides locomotion for the trolley. The trolley is driven along the track (Fig. 7) by a rack and pinion gear arrangement.

The track section in each vacuum chamber is powered independently by a stepper motor mounted on a UHV rotary feedthrough. One stepper motor controller is used to drive all three stepper motors in unison. Inside the chamber a vertical shaft translates the rotary motion (by a helical gear arrangement) to a horizontal drive shaft that runs along an inverted V-shaped cavity in the bottom of the track. In the middle and at each end of this drive shaft the rotary motion is translated (using worm gears) to pinion gears. A rack located along the lower rear side of the trolley engages a least one, and no more than two, of these pinion gears when the trolley is sitting on the track. While the independent pinion gears of a particular track section are synchronized with each other, they are not synchronized with the gears of the next track section. To compensate for the mismatch of the gears, the horizontal and vertical shafts are spring loaded to absorb motion created when the teeth of the rack, driven by one motor, engage the teeth of an unsynchronized pinion gear driven by the motor for the adjacent chamber. Once the rack of the trolley has cleared the pinion gear of the previous chamber, the spring tension is released as rotary motion.

YP 6668

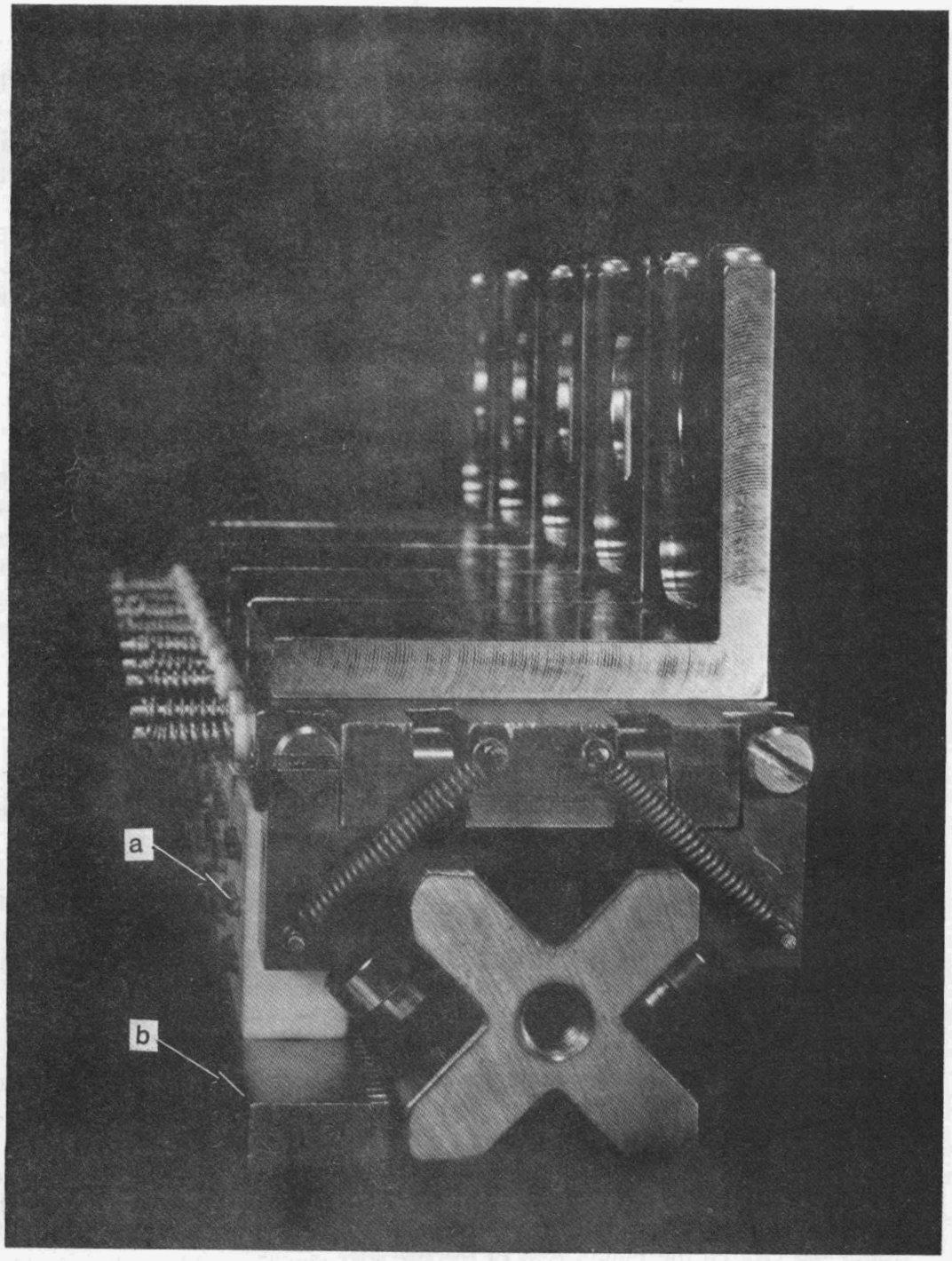

Fig. 6. End view of the trolley positioned on a section of track: (*a*) chromel-alumel thermocouple contacts on back side of trolley; and (*b*) rack gear for locomotion.

YP 6659

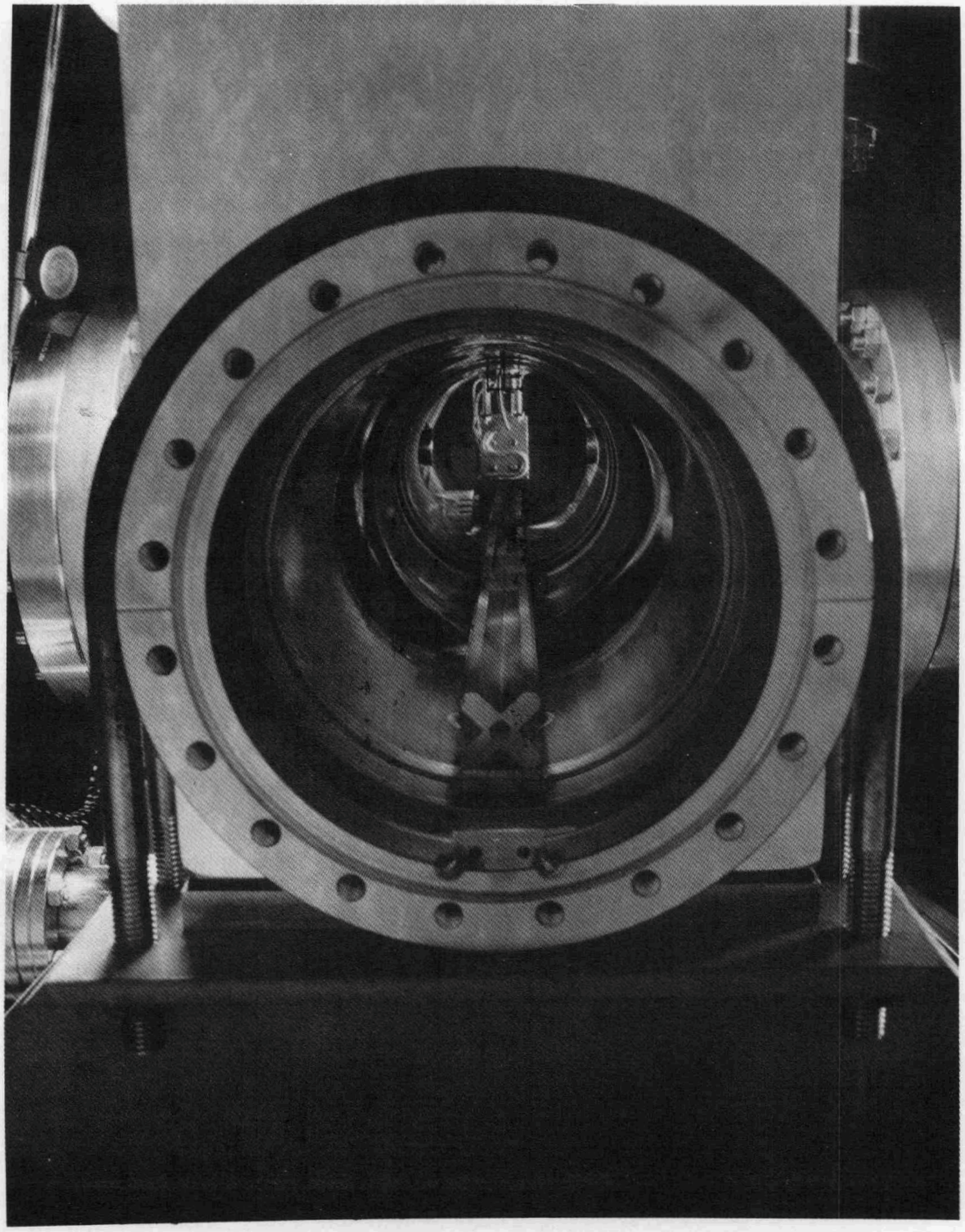

Fig. 7. A view through the loading chamber into the ion-irradiation chamber shows the side view of the water-cooled heater assembly and a side view of the thermocouple contact block.

5. HEAT AND TEMPERATURE CONTROL SYSTEM

In the standard configuration, the heating and control system will provide stable specimen operating temperatures from 350 to 850°C. These temperatures will be maintained to within 1°C of the preset temperature. Without the heating control system, the temperature of specimens being ion irradiated is governed by the energy of the ion beam and the specimen's thermal conducting and radiative properties. In some cases, uncompensated beam heating can raise the temperature hundreds of degrees celsius.

A liquid nitrogen cyrostage was constructed to permit ion irradiations with specimen temperatures approaching liquid nitrogen temperature. At present, the cyrostage has specimen temperature monitoring but has no provision for feedback temperature control, although plans for future modification include the possibility of incorporating a controllable reheating section.

5.1 THERMOCOUPLE

Each station on the trolley has three type K (chromel-alumel) thermocouples, two of which are inserted into parallel holes in the thermalizer block behind the specimens and the other is spot-welded to the surface of a specimen. The two thermalizer thermocouples are grounded junction sheathed types and are 1.02 mm in diam., the surface thermocouple consists of 0.127-mm diam bare chromel and alumel wires. The thermalizer thermocouples are spring-loaded to ensure positive thermal contact with surfaces in the thermalizer block. As shown in Fig. 5, the thermalizer thermocouples make a loop in front of the trolley to accommodate the downward displacement when the spring-loaded couple bottoms out in the thermalizer block and are routed to the back side through holes in the trolley. The thermocouple wires are connected to electrical contacts mounted on a machinable glass substrate located on the back side of the trolley [(*a*) in Fig. 6]. When a station is positioned in line with the ion beam, these contacts (three sets per station) mate with matched sets of spring-loaded contacts that are permanently positioned in the ion-irradiation chamber. The third set of contacts in this switch-type assembly is connected to a spring-loaded mechanism that captures the bare wire of the surface thermocouple on the front side of the trolley [(*a*) in Fig. 5].

5.2 ELECTRON GUN HEATERS

Specimen heating in the ion-irradiation chamber is produced using a dispenser cathode type triode assembly. Figure 8 shows a prototype of the triode assembly mounted in a spring-loaded water-cooled housing. This housing is attached to a UHV pneumatically driven linear motion feedthrough. When the trolley is positioned in front of the ion beam, the heater housing assembly is lowered onto the L-shaped specimen mounting bracket. In this position the heater is directly behind the specimen thermalizer block, which is used as an anode in the triode assembly. The cathode and grid structure is electrically insulated from the heater housing and operates at a negative voltage with respect to the anode, which is at ground potential.

YP 6662

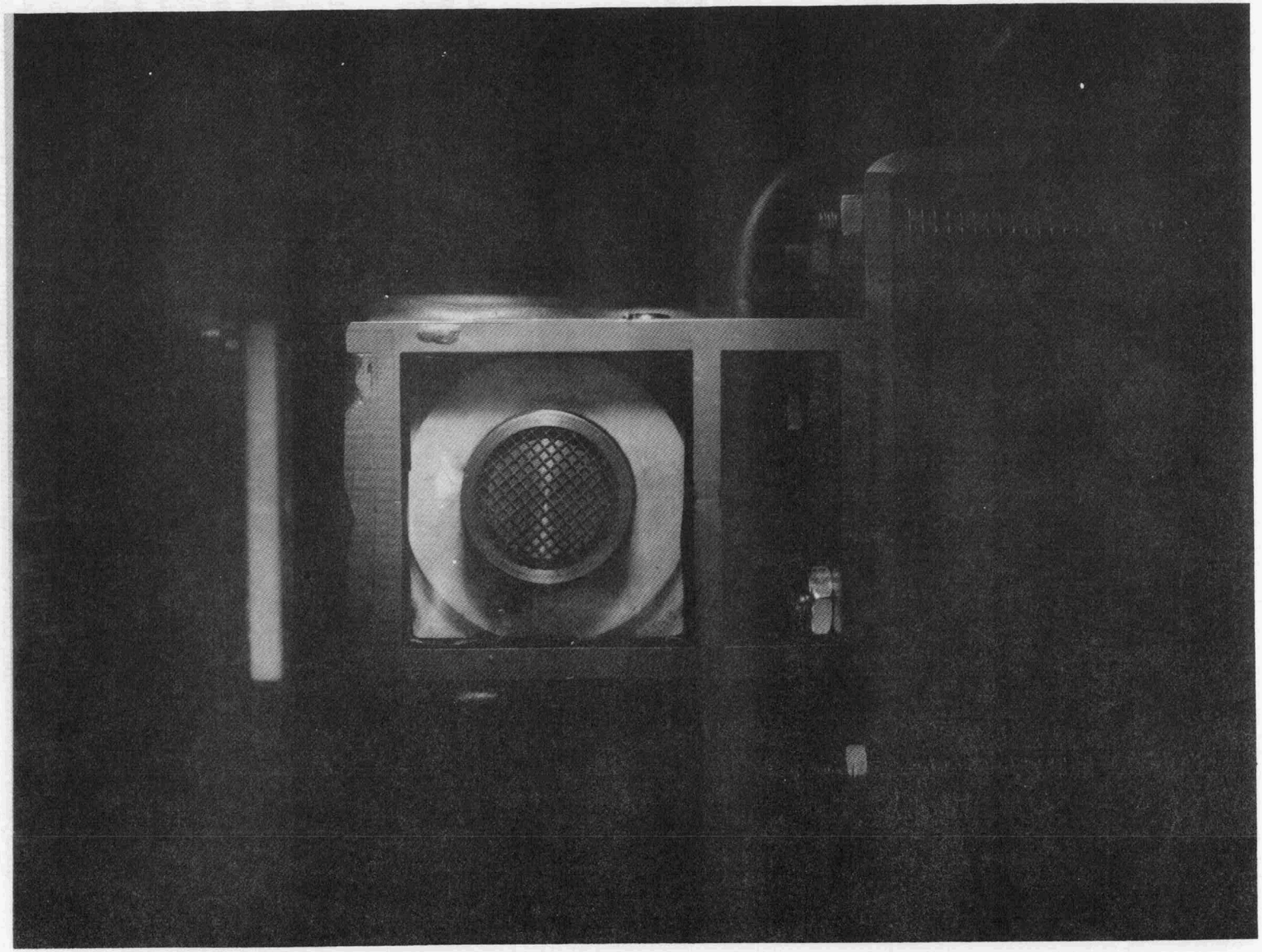

Fig. 8. A view of the front side of the spring-loaded water-cooled heater assembly.

5.3 TEMPERATURE CONTROLLER AND FEEDBACK

Because the dispenser cathode assembly is operated at a negative potential, the filament heater power supply and the grid power supply must float at a negative potential with respect to ground. To accomplish this, a high-voltage isolation transformer is mounted in a large Plexiglas cage with the power supplies. Isolating the control signals for the power supplies is accomplished by using a voltage-to-frequency converter to drive an optical coupler tied to a frequency-to-voltage converter. The control signal is produced by a Leeds and Northrup Electromax V Plus temperature controller [(*d*) in Fig. 2] connected to the thermocouple from the thermalizer block.

6. COMPUTER AND CONTROLLER SYSTEM

The computer-controlled data acquisition and control system integrated into the Triple Ion Beam Irradiation Facility makes it possible for one person to operate the entire facility.

An HP 80286 Vectra (MS-DOS) personal computer is the central controlling unit in the data acquisition and control system. The Vectra is connected via an IEEE-488 interface card to an HP upgraded 3852A controller, located in the Van de Graaff control room (Fig. 9). The HP 3852A controller is in turn connected to an HP 3853A extender module that is located in the ion-irradiation chamber room ~25 m away. The HP 3852A controller was updated by replacing the set of ROM chips with a later version and increasing the internal memory to 2 MB.

6.1 HARDWARE

6.1.1 Touchscreen

In order to incorporate the control of a large number of functions in a readily accessible format, a touchscreen-based approach was developed in conjunction with a shell-type resident program with pull-down menus [Fig. 10 and (*a*) in Fig. 9]. The shell program will be described in Sect. 6.2.1.

The touchscreen is a Hewlett Packard product developed for an entirely different computer system. A software device driver had to be written to make it compatible with the Vectra computer. The driver was written in Microsoft Quick Basic version 4.5, which allows direct access to the Vectra's Central Processing Unit (CPU) registers. By preloading a particular set of these registers with a special set of hexadecimal numbers and executing a software interrupt, it is possible to examine the results of a hardware-driven interrupt resulting from the activation of the touchscreen. The driver sequentially polls the CPU registers at 1-s intervals to determine if the screen has been touched. If the screen was not touched, the computer goes on with what it was doing, but if the screen was touched, the computer reads the registers containing the information specifying the touched area. A touched location represents 1 of 80 horizontal and 1 of 25 vertical positions. Particular areas of the screen are programmed to respond in a specific manner, depending on the last menu that is displayed on the screen. This algorithm affords the operator an option of activating one of several menus and directing the control system to perform tasks that are listed in these menus. The content and extent of this control is described in Sect. 6.2.1.

YP 11240

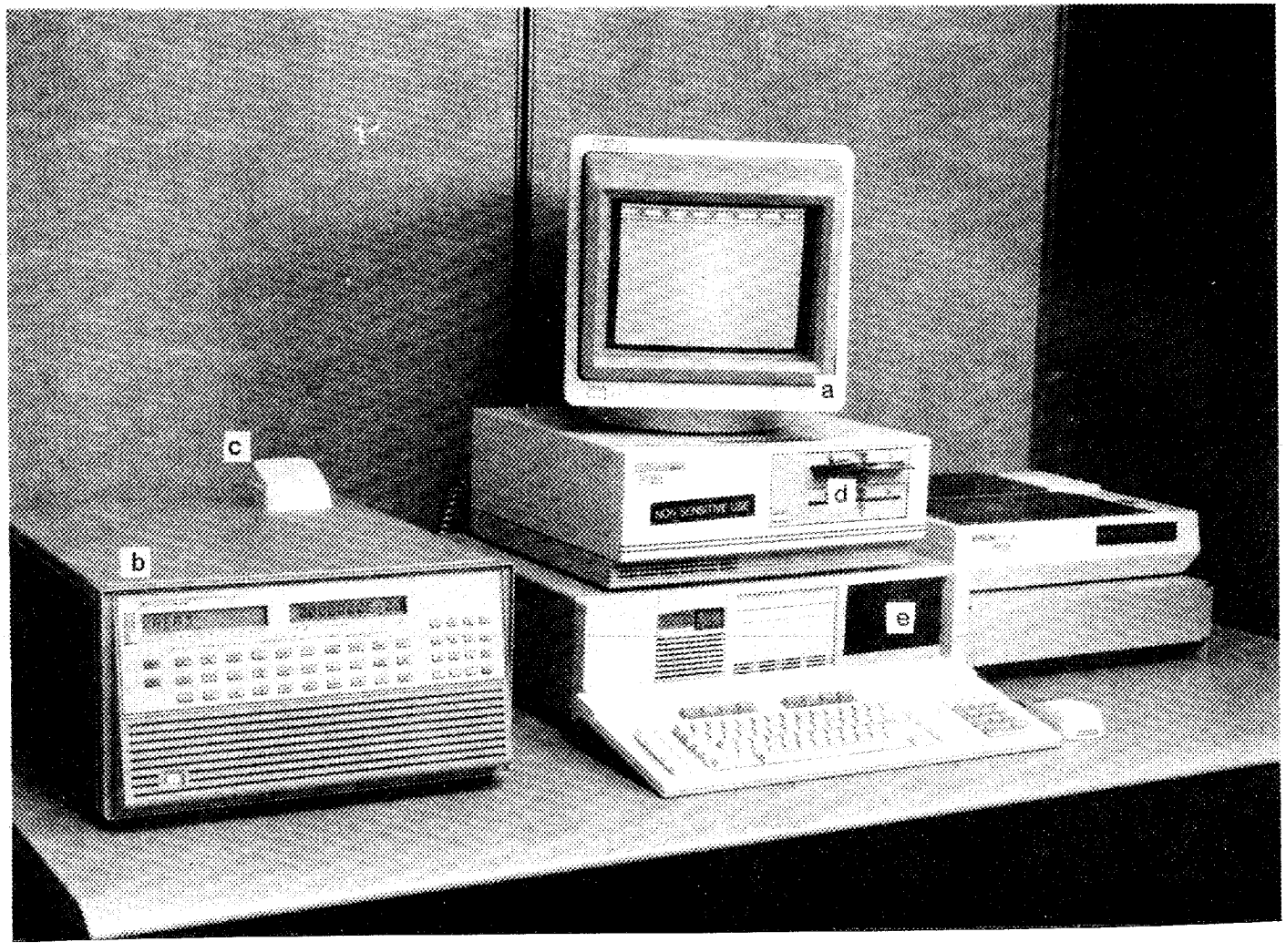

Fig. 9. Data aquisition system: (*a*) touchscreen monitor; (*b*) HP 3852A controller; (*c*) speaker for speach synthesizer; (*d*) HP Vectra PC; and (*e*) Vectra PC bus extension crate.

YP 11246

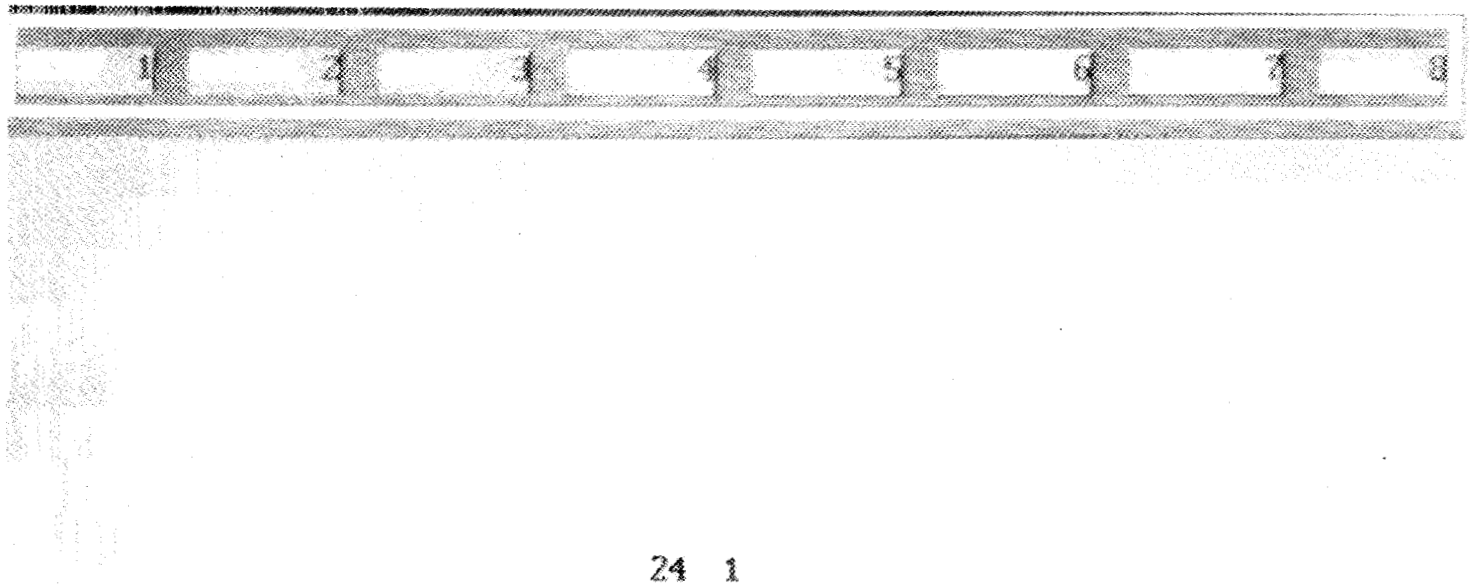

Fig. 10. The touchscreen shell main menu is used to access and initiate all functions in the data acquisition and control system.

6.1.2 Talker

The computer system is equipped with a speech synthesizer that verbalizes combinations of both characters and numbers. The unit is a Heathkit HV-1000 card with an external speaker. The speech synthesizer has the ability to speak alpha-numeric values read into the data acquisition system and to warn the operator when a certain parameter exceeds a preset limit. It has the ability to speak written text as well as results of mathematical manipulations.

6.1.3 IEEE-488 Interface

All communications between the Vectra computer and the HP 3852A controller are handled through an IEEE-488 interface card. The card, IOtech model GP488A, resides in a short slot in the AT bus structure of the Vectra. Communications to the HP 3852A are in the form of literals and are written to the software device driver called \DEV\IEEEOUT. Communications from the HP 3852A are also in the form of literals and are read through a device driver known as \DEV\IEEEIN. Subroutines are down-loaded from the Vectra prior to the beginning of a run. These subroutines must be written in strings representing the programming language and control structure recognized by the HP 3852A and must be formatted in the protocol dictated by the GP488A interface card. A small subroutine in the main computer program called OUTDO is used to perform the down-loading process.

The IOtech GP488A uses the light pen interrupt line to give the HP 3852A the ability to interrupt the Vectra computer to perform data transfers. After each interrupt, the GP488A card requires a software reset from the Vectra computer prior to further communication. This task is incorporated in the interrupt service routine in the main computer program.

6.1.4 Controller

An HP 3852A [(*b*) in Fig. 9] with an upgraded ROM kit and 2 MB of on-board memory is the central controller in the data acquisition system. The HP 3852A is connected via an IEEE-488 to the Vectra, and both units are located in the control room in Bldg. 5500. An HP# 3853A [(*d*) and (*e*) in Fig. 2] is an extension of the HP 3852A and is located in the target room ~25 m away. Both units are card-cage crate types with the HP 3852A accepting eight I/O modules and the HP 3853A extender having a larger capacity of ten I/O modules.

The HP 3852A uses a Motorola 68000 CPU and has a ROM-type basic operating system that accepts either directly executable commands or stored volatile subroutines that have been down-loaded from the main computer or that have been keyed in from the front panel keyboard. These subroutines can be actuated from the main computer, from the front keyboard, or from other subroutines in the controller.

The programming language used in the controller is similar in many respects to BASIC, with a large portion of the commands written specifically for a particular I/O module or certain groups of modules. One of the major drawbacks in the language, (with the exception of FOR....NEXT and WHILE....END WHILE statements) is that direct branching back is disallowed. There are no GO TO or DO statements. Branching back can be accomplished by constructing smaller subroutines that are called from a number of constructs such as IF....THEN....ELSE commands.

The language supports software interrupts, allowing the controller to activate interrupt service routines in the host computer.

6.1.5 Controller Modules

A large variety of I/O modules are available for the controller and the extender. These include the following: high-speed digital voltmeter, multiplexers, stepping motor controllers, digital input, digital output, digital-to-analog converters, counter/totalizer, and relay contact closures. Some of the I/O modules are capable of initiating hardware-type interrupts, whereas others are capable of scaling inputs from thermocouples to indicate temperatures.

The HP 3852A has a Digital Integrating Voltmeter (HP 4401A) capable of autoranging at full speed of 100,000 readings/s for scanned static signals. The voltmeter uses the successive approximation method of digitally approximating an analog voltage and has a 12-bit resolution. The unit has a standard 8k (actually 8,192) reading storage memory in the I/O module with the ability to hold up to 64k (actually 65,536). The unit can be used as a voltmeter, an ohmmeter, an ammeter, and a thermocouple or resistance temperature detector (RTD) temperature-measuring meter (with conversion for most types of sensors) and is reference-junction compensated.

The HP 3852A has three other I/O modules. They are the HP 44708A relay-type multiplexer used to multiplex thermocouple inputs into the digital voltmeter, an HP 44721A 16-channel digital input module for importing digital TTL level signals into the controller system, and an HP 44725A 16-channel general purpose switch for turning on or off up to 16 independent devices.

Most of the acquisition and control in the system takes place in the HP 3853A extender unit in the target room. The HP 3853A has eight I/O cards in its crate, starting with an HP 44725A 16-channel general purpose switch, which is used to sequentially switch the current signal from ground to the input of the current integrators in sets of three to measure the beam current in the three mini-Faraday cup assemblies. The respective cup locations in each mini-cup are connected to one of three current integrators for simultaneous measurements, reducing the total time needed to block the beam.

The next I/O unit is the HP 44721A digital input used to keep track of the position of the valves in the vacuum chamber specimen transfer system. The computer needs to know when the valves are open during the process of transferring the specimen trolley through the valve.

The HP 44724A occupying the next slot is a 16-channel digital output card. The TTL-level output from this card drives a bank of solid state relays located in a power strip mounted below the upper portion of the chamber system support structure. The computer (through the controller) can be programmed to activate or deactivate these relays, supplying 117 VAC (up to 40 A) to anything plugged into the outlet. At present only seven of the sixteen channels are dedicated to the power strip, saving nine for future expansion. The solenoid-activated pneumatic valves that drive the mini-cup assemblies into the path of the beam are collectively connected to one of the outlets.

The next series of four I/O modules are five-channel counter/totalizer units (model HP 44715A). Two of these I/O modules are used to count pulses that emanate at the same time

interval from six current integrators, three of which integrate current from three beam profilometers and three of which integrate current from the mini-cup assemblies. The other two I/O modules are connected to position encoders on the trolley transport stepper motors to give the location of the trolley.

The last I/O module in the crate is an HP 44724A digital output used to communicate with a stepper motor controller.

6.2 SOFTWARE

The software controlling the data acquisition and control system is written in Microsoft Quick Basic 4.5 (see Appendix A).

6.2.1 Shell Program

The main program resides in a shell-type program with pull-down menus. Figure 10 shows the main menu window. The topics listed across the top of the screen can be explored simply by touching the area of the screen where they appear. To return to the main menu you must touch the area where the word RETURN appears. If you touch the screen in the area in which a menu item is written, that preprogrammed function will be initiated. If you touch another topic in the main header at the top of the screen with another menu on the screen, the second menu will replace the first.

Touching the left most topic on the main menu labeled SETUP 1 brings down the menu shown in Fig. 11. This window is mainly for pre-run control and run parameter selection. INITIALIZE RUN is the section that lists the complete set of default selections and prompts the user for information about the pending run. READ DEEP CUP, if activated, will insert the mini-Faraday cup assemblies in each beam line, measure the beam currents, retract the cup assemblies, list currents temporarily on the screen, and print the data gathered on the printer. CHANGE RUN PARAMETER allows the user to change any of the run parameters previously entered. DEFAULT ALTERATION affords the opportunity to alter the default parameters for the pending run.

MANUAL 2, the second entry at the top of the main menu, is used to control manual functions in the chamber system. The first two entries that will be introduced in this function are VALVES and INSTRUMENTS. These categories, currently being programmed, will involve opening and closing valves and the manual control of instruments. INSERT CUPS and REMOVE CUPS headings follow in order and will activate the mini-Faraday cup assemblies to block the beams from the target.

Section 3 of the main menu is START. As the name implies, it is the function activated to start an ion-irradiation run. START RUN, RESTART, CHG PARAMETERS, and LST CONTROL fall under this heading. START RUN will initiate the down-loading of subroutines to the controller and proceed with the run until the required dose has been delivered to the target. During the run, the data is stored on a floppy diskette, displayed on screen, and printed on the printer. Details of these transactions will be covered in Sect 6.2.3.2. After the run is complete, the computer will revert back to the main menu. RESTART is double-faceted in that

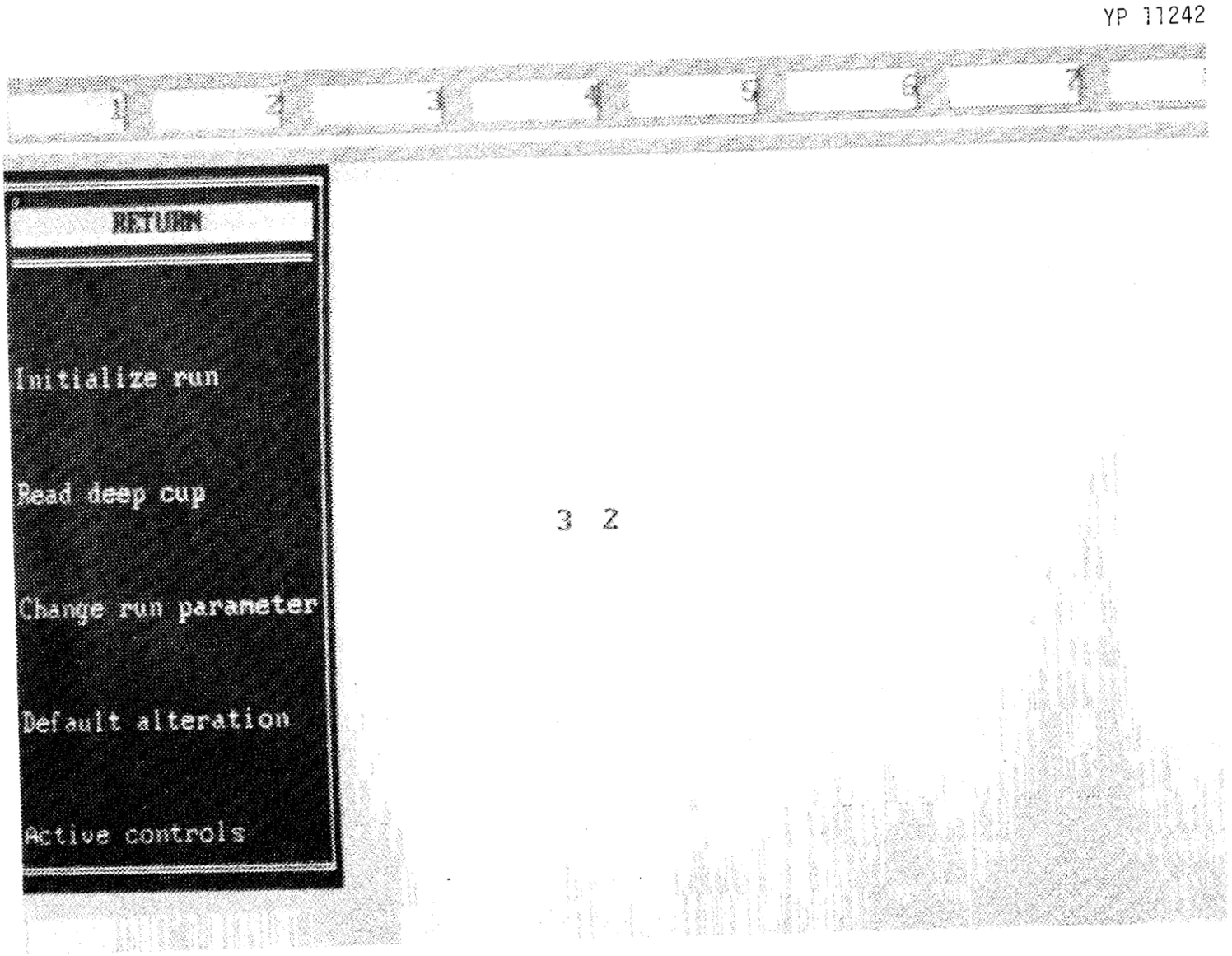

Fig. 11. A pull-down menu generated by touching the area "Setup 1" on the monitor screen.

it can be initiated to continue a run after a major system disruption that crashes the computer or it can be used to continue a run that was previously performed to augment the accumulated target dose. CHG PARAMETERS duplicates a feature available in the SETUP section of the program. LST CONTROL simply states the configuration of instruments (e.g., in which chamber the ion guage will be used to report vacuum conditions).

With the exception of section MONITOR 6, the remainder of the functions are currently being programmed. The second heading under MONITOR is VACUUM; when activated, it reports the vacuum condition in the chamber selected under SETUP/ACTIVE CONTROLS. The third heading under MONITOR is DEEP CUP. It is a beam diagnostic program that inserts the mini-Faraday cup assemblies in all beam lines and displays the beam current as the controller switches from one mini-Faraday cup to the next. The display shown in Fig. 14 shows the results of this exercise.

6.2.2 Dos Level Interrupt

Microsoft Corporation not only developed the operating system usually termed MS-DOS for IBM PC and compatibles, but they also developed a highly advanced form of BASIC, which they named Quick Basic. The advances of Quick Basic over BASIC are enormous, and a description of these advances is beyond the scope of this report. This is mentioned here to clarify possible confusion to someone trying to follow the program listing that is presented in Appendix A.

Quick Basic allows the programmer direct access to the tools the DOS kernel uses in managing the BIOS and extended BIOS in all IBM PC compatible machines. As a prerequisite to an interrupt call, the software program in Quick Basic must declare the CPU registers with a REGTYPE statement. Hexadecimal numbers are loaded into specific registers dictated by the function call to be made in the low-byte-high-byte order.

6.2.3 Run Algorithm

6.2.3.1 Setup

As previously mentioned, the START RUN menu selection is included in the main menu under the heading START. Before using this function it is mandatory to establish the parameters that will control the ensuing run (see SETUP/ INITIALIZE RUN). Entering INITIALIZE RUN through SETUP invokes sequential prompts to input data. The first prompt allows a view of the data used with the last run recorded on the diskette. If this default offer is declined (N or n), then input is requested about run parameters. Each question mark requires a response. If a particular machine is not to be used in the run, further inquiries about that machine will not appear.

After all data is entered, it will be displayed on the screen. An input prompt will be displayed giving the operator the option of listening to a verbal repetition of input data. After reviewing the input data, the option to make changes will be presented. This option for alterations includes the set of basic parameters used in the calculations that could be changed if the run involves a nonstandard situation. The program will then ask for a data disk in drive A:\ (it must be a formatted 2S/HD diskette). If the diskette was not previously used to store data, it will be initialized. IF A DISKETTE HAS DATA RECORDED FROM A PREVIOUS RUN AND IF

A RESPONSE INDICATES THAT IT IS A NEW DISKETTE, THE DATA ON THE DISKETTE WILL BE OVERWRITTEN AND DESTROYED! The run parameters are overwritten in a file on the diskette called PARAM, and the screen reverts back to the main menu, reactivating the touch functions.

6.2.3.2 Disk Control and Data Storage

When START RUN is selected, an initiation process begins. The floppy diskette in drive A:\ (Sect. 6.2.3.1) contains a file named INDEX that contains a computer-generated name of the file used for storing data during the last run. This file name consists of two parts, the first four digits indicate the name assigned to the floppy diskette and when it was first used, and the second four digits represent a sequential file number on that particular diskette. The file number is incremented by one and stored back into the INDEX file, replacing the previous entry. This newly created file number is the file name in which data for the impending run will be stored. The floppy diskette also contains a file called CATALOG. This file is used for cataloging the names of the data storage files with cross references to the run number given by the operator and the date of the run. The file initiation process is concluded by creating two files on the diskette for data storage. The first file uses the name generated by the computer (mentioned above), and the second file, used for storing augmented data, uses the same number except that the fifth digit from the left is replaced with an "A".

The second phase of the initiation process concerns preparation of the data acquisition and control system. After resetting the HP 3852A controller, subroutines and controlling parameters are down-loaded from the Vectra computer. The subject of controller communications was discussed in detail in Sect. 6.1.3 and 6.1.4.

At the beginning of the run, the first response for the system is to measure the beam currents that are applicable to the run and calibrate the appropriate beam profilometers. This process takes place at 10-min intervals during the run and is discussed in detail in Sect. 6.2.3.4. The data obtained is stored on the floppy diskette under a heading indicating mini-Faraday cup reading, time of day, and incremental run time. The actual data stored is the average of the cup readings, the average of the beam profile monitor (BPM), a factor relating the BPM to the cup readings, and the distribution from the average for each series of cups for all machines participating in the run.

After the cup readings, the target specimens are exposed to the single or multiple ion beams, and the accumulation of integrated data is gathered and recorded at 1-min intervals on the diskette for the next 10 minutes. This type of data is stored under a heading indicating that the data is converted BPM with distribution factors. The data is stored in the following sequence: average ions/cm^2 for each beam, an accumulation of these averages, and the distribution of the beam calculated from the average and the distribution measured during the last cup readings.

It is possible to generate a short computer program that can retrieve stored data from the diskette for plotting or for further analysis.

6.2.3.3 Data Presentation and Options

Data acquired during the run is displayed on the screen and is printed on a dot matrix printer. Displayed data is updated on a 1-min interval for the screen and a 10 min schedule for the printer. Figure 12 illustrates the layout of the data displayed on the screen. The printed information is basically a screen dump of the displayed data after a cup reading, which is augmented by a list of parameters used in the run.

6.2.3.4 Beam Integration Algorithm

Each beam line associated with the Triple Ion Beam Irradiation Facility has both a BPM and a mini-Faraday cup assembly mounted relatively close to the target chamber. The mini-Faraday cups are shown in Fig. 13. They are mounted on pneumatically actuated vacuum feedthroughs and are inserted into the beam via computer control. The BPM consists of an insulated Z-shaped wire that is rotated in the beam at an orthogonal 45° angle with respect to the direction of the beam. As the wire rotates, it scans the beam in the X and Y axes. Electrons generated by collisions of the accelerated ions with the rotating wire are collected in a Faraday cup. An electrometer connected to the Faraday cup produces a measurable signal with an output amplitude directly related to the strength of the intersected beam at any given time. This signal is used to visualize the beam profile on an oscilloscope. A resistor voltage-dividing network diverts a small part of this signal into a current integrator with an output of pulses equivalent to 1×10^{-6} C/pulse. Current is a measure of coulombs per unit time, and an ampere is defined as 1 C/s. By counting the pulses produced by the current integrator, it is possible to calculate the current. This representation of the beam current can be calibrated by measuring the actual beam with mini-Faraday cups while simultaneously measuring the signal from the BPM and is the justification for interrogating the beam with the mini-Faraday cups on a 10-min schedule. Beam distribution is derived by averaging the mini-Faraday cup measurements and calculating the decimal fraction of the average for each cup in the mini-cup assembly.

6.2.3.5 Restart

As a subfunction under the main menu heading START, the RESTART touchscreen option allows the operator to restart a run after a system shutdown that was intentional or otherwise. For example, the computer system and controller will crash if the electrical power to the building is interrupted. In this event, it is possible to restart the run from the point of termination. To do this, the computer program examines the file containing the data taken up to the point of interruption. It will select the last full set of data recorded. The data on the disk contains not only the accumulated dose at the time of termination but also the original parameters used in the run. The operator should answer all prompts presented on the screen to activate this option.

7. REFERENCES

1. M. B. Lewis, et al., *Triple Ion Beam Irradiation Facility*, ORNL/TM-10867, December, 1988.

YP 11244

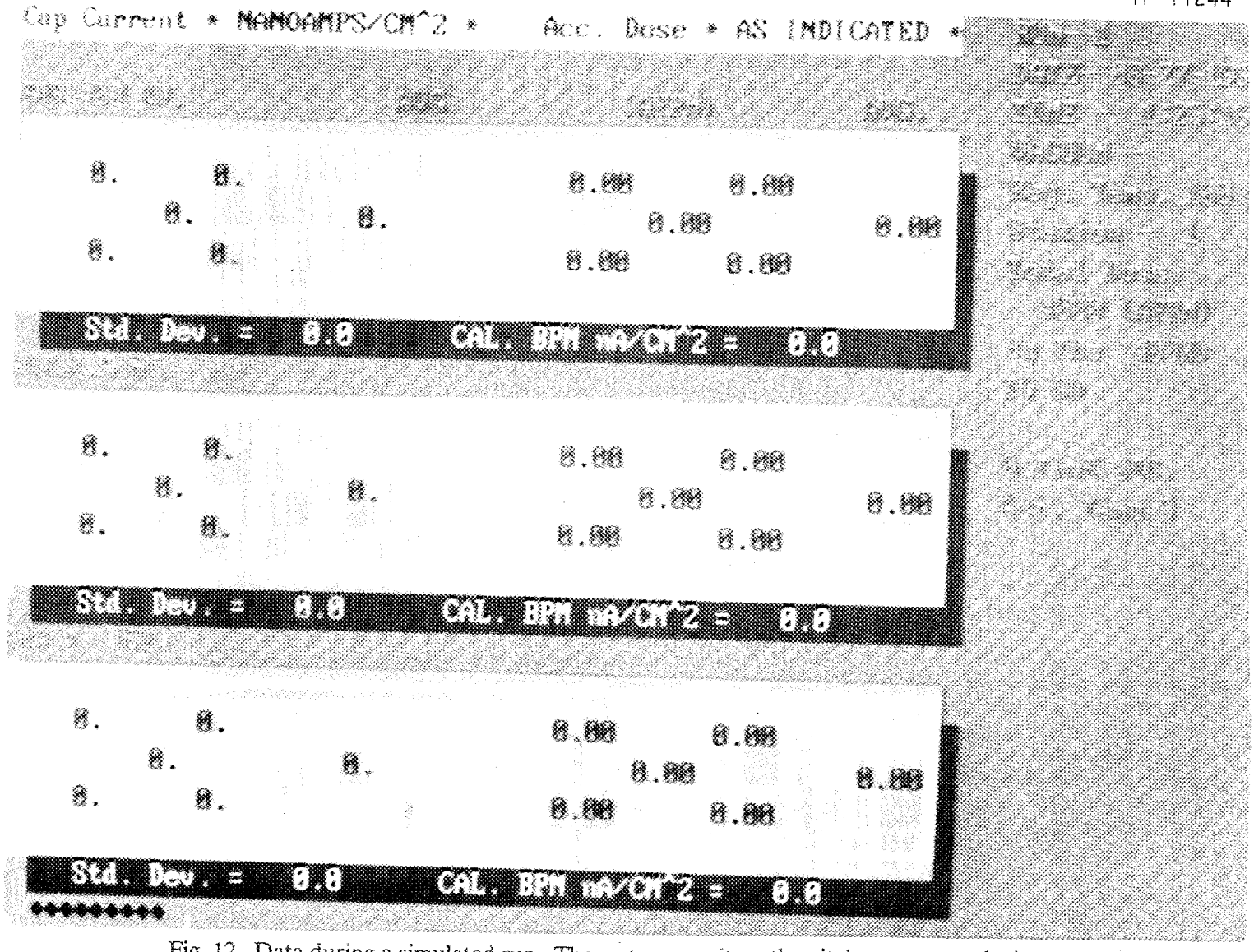

Fig. 12. Data during a simulated run. The system monitors the vital parameters during a run, records the information on disk, and prints a hard copy of the results.

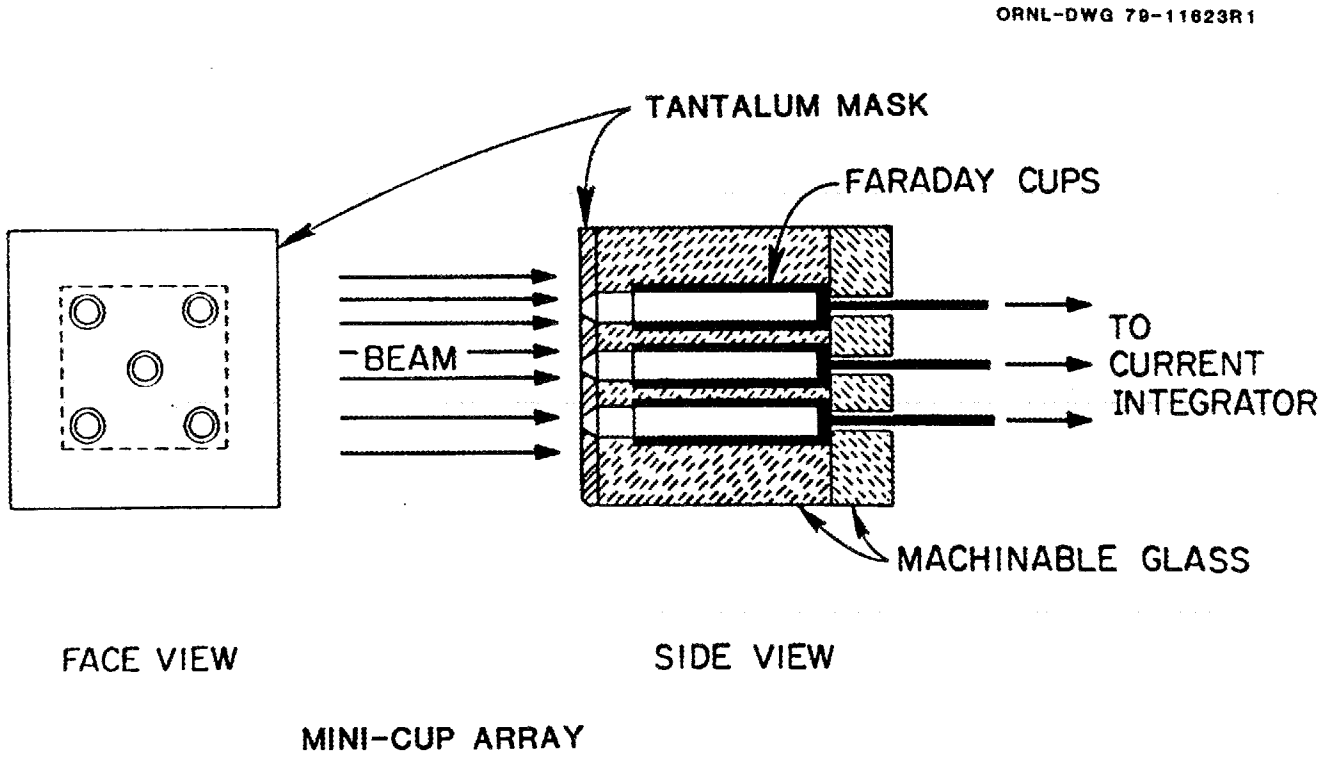

Fig. 13. Mini-cup schematic. The ion-beam is measured in the location of the specimens. The cross section of the beam measures 10 mm X 10 mm.

YP 11243

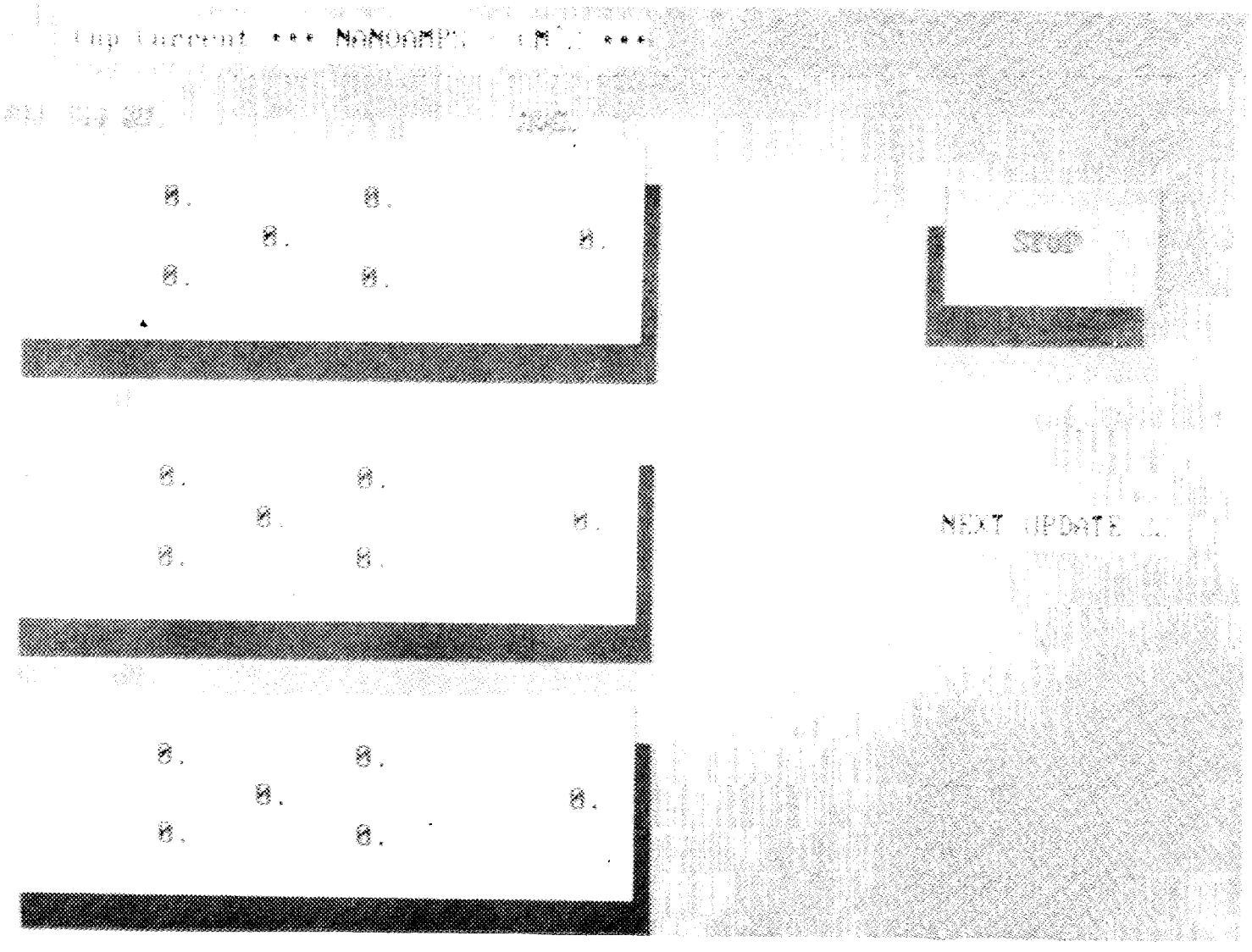

Fig. 14. Diagnostic program depicting the beam current distribution on the target surface.

YP 11234

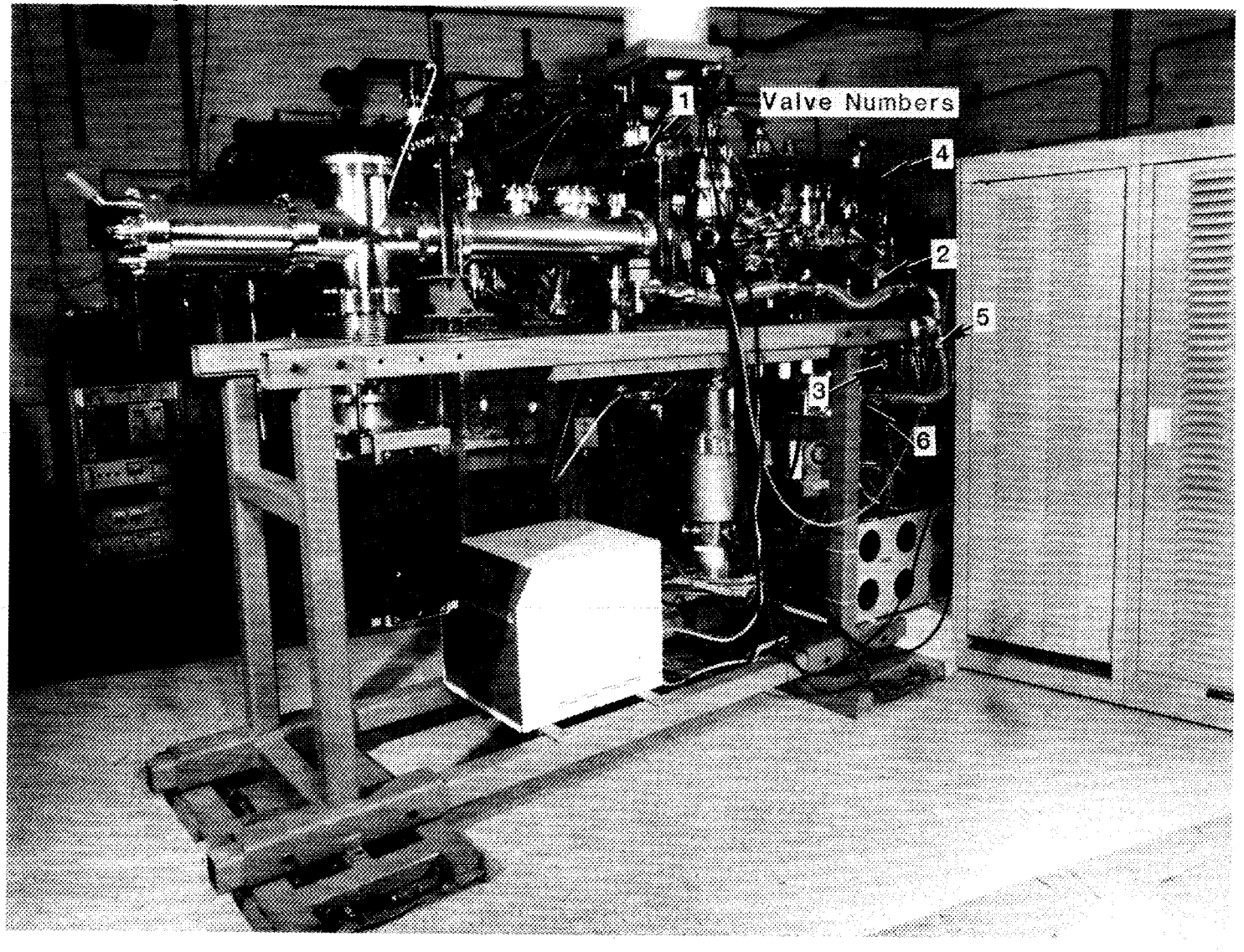

Fig. 15. Valve locations listed for operating procedures.

APPENDIX A

COMPUTER PROGRAM

```
DECLARE SUB TILCURA ()
DECLARE SUB IONMAG ()
DECLARE SUB TILCTP ()
DECLARE SUB MAGCAL ()
DECLARE SUB POPLIST (HEADER$, SHOWITEMS%, MAXITEMS%, ITEM$(), FORE%,
BACK%, HFORE%, HBACK%, QUADRANT$, SHADOW%, SLCT%)
DECLARE SUB MAKEWIND (ULR%, ULC%, LRR%, LRC%, FRAME%, FORE%,
BACK%, GROW%, SHADOW%, LABEL$)
DECLARE SUB OUTDO (A$)
DEFINT I-Q
DEFSNG O
OPTION BASE 0
DIM I$(15)
DIM C(40), Z(15)
 TYPE RegType
   AX AS INTEGER
   BX AS INTEGER
   CX AS INTEGER
   DX AS INTEGER
   BP AS INTEGER
   SI AS INTEGER
   DI AS INTEGER
   FLAGS AS INTEGER
   DS AS INTEGER
   ES AS INTEGER
 END TYPE
DIM InRegs AS RegType, OutRegs AS RegType
DIM SCR%(2000)
     C(1) = 0: C(2) = 0: C(3) = 0: C(4) = .1: C(5) = 0: C(6) = 9.1
     C(7) = .1: C(8) = 0: C(9) = 1!: C(10) = 1!: C(11) = 165: C(12) = 1000
     C(13) = 8000: C(14) = 1!: C(15) = 0: C(16) = .1: C(24) = 1: C(25) = 1: C(26) = 1

    OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
    OPEN "\DEV\IEEEIN" FOR INPUT AS #2
PRINT #1, "FILL ERROR"
PRINT #1, "CONFIG /I"
PRINT #1, "TIME OUT 70"
A$ = "CLROUT"
CALL OUTDO(A$)
A$ = "RST"
CALL OUTDO(A$)
```

```
    CLOSE #1
    CLOSE #2
 TIMER ON
 STATUS = 0
 STATE = 0
 ROW = 0
 COL = 0
  Q = 0
 CLS
  L = 0
ENT1:
PRI = -8
  FOR X = 2 TO 9
    PRI = PRI + 10
    CL(X) = PRI
    CH(X) = PRI + 6
    RL(X) = 1: RH(X) = 4
  NEXT X

 N = 9:   ' NUMBER OF SPOTS TO POLL
 nr = 0: RL(1) = 19: RH(1) = 22: CL(1) = 28: CH(1) = 50: DR = 930
 nc = 0
COLOR 15, 6
lab$ = ""
CLS
CALL MAKEWIND(2, 2, 2, 78, 2, 14, 1, 1, 0, lab$)
LOCATE 2, 2: PRINT "SETUP  "; : COLOR 16, 6: PRINT "1"; : COLOR 15, 6: LOCATE 2,
12: PRINT "MANUAL "; : COLOR 16, 6: PRINT "2"; : COLOR 15, 6
LOCATE 2, 22: PRINT "START  "; : COLOR 16, 6: PRINT "3"; : COLOR 15, 6: LOCATE 2,
32: PRINT "TRAIN  "; : COLOR 16, 6: PRINT "4"; : COLOR 15, 6
LOCATE 2, 42: PRINT "TEMP.  "; : COLOR 16, 6: PRINT "5"; : COLOR 15, 6: LOCATE 2,
52: PRINT "MONITOR"; : COLOR 16, 6: PRINT "6"; : COLOR 15, 6
LOCATE 2, 62: PRINT "PROGRAM"; : COLOR 16, 6: PRINT "7"; : COLOR 15, 6: LOCATE
2, 72: PRINT "event "; : COLOR 16, 6: PRINT "8"; : COLOR 15, 6
' THIS WILL BE REPLACED WITH REAL HEADER ie. GOSUB HEADER
ON TIMER(1) GOSUB ENTERUPT
710 GOTO 720
720 GOTO 710

END

ENTERUPT:
  InRegs.AX = &H600
  InRegs.BP = &HC6
780 CALL INTERRUPT(&H6F, InRegs, OutRegs)
  STATUS = OutRegs.AX: STATE = OutRegs.DX: ROW = OutRegs.CX + 1
```

```
  COL = OutRegs.BX + 1
 IF STATE = 32640 THEN GOTO EXT ELSE RETURN
 RETURN
EXT:

    ON Q GOTO MNU1, MNU2, MNU3, MNU4, MNU5, MNU6, MNU7, MNU8
   FOR r = 1 TO N
    IF ROW >= RL(r) AND ROW <= RH(r) AND COL >= CL(r) AND COL <= CH(r)
THEN DRT = r
   NEXT r

    LOCATE 13, 35: COLOR 0, 6: PRINT " "; ROW; COL; " "; : COLOR NF, NB

IF DRT = 1 THEN
    LOCATE 19, 28: COLOR 14, 4: PRINT SPACE$(23);
    LOCATE 20, 28: COLOR 14, 4: PRINT "    THANK YOU     "; : COLOR NF, NB
     FOR I = 1 TO 10000: NEXT I
    COLOR 7, 0
    CLS
    SYSTEM
ELSEIF DRT = 2 THEN
    GOSUB EVNT1
ELSEIF DRT = 3 THEN
    GOSUB EVNT2
ELSEIF DRT = 4 THEN
    GOSUB EVNT3
ELSEIF DRT = 5 THEN
    GOSUB EVNT4
ELSEIF DRT = 6 THEN
    GOSUB EVNT5
ELSEIF DRT = 7 THEN
    GOSUB EVNT6
ELSEIF DRT = 8 THEN
    GOSUB EVNT7
ELSEIF DRT = 9 THEN
    GOSUB EVNT8
ELSE
    ' SET PARAMETERS AND CALL POPLIST  ect. TO CASE = 9
END IF
  RETURN

MNU1:
       IF ROW < 8 THEN
    GOSUB RETCRT
    Q = 0: L = 0
    RETURN
```

```
    ELSE
     ON INT(((ROW * 3) + 7) / 10) - 2 GOTO INITIAL, READ.CUP, CHANGE,
DEFAULT.CHG, ACTIVE.CONT
      END IF

    GOSUB RETCRT
    Q = 0: L = 0: DRT = 0
    RETURN
INITIAL:
    COLOR 15, 1
    LOCATE 1, 1
    CLS
    C(1) = 0: C(2) = 0: C(3) = 0: C(4) = .1: C(5) = 0: C(6) = 9.1
    C(7) = .1: C(8) = 0: C(9) = 1!: C(10) = 1!: C(11) = 165:
C(12) = 1000
    C(13) = 8000: C(14) = 1!: C(15) = 0: C(16) = .1: C(24) = 1: C(25) = 1: C(26) = 2
   PRINT "*** IF THIS IS THE FIRST ENTRY ON A NEW DISKETTE YOU CAN **
NOT**", "***AUGMENT THE LAST ENTRY AND MUST REPLY  N  TO NEXT INPUT
***"
   INPUT "*** DO YOU WANT TO AUGMENT LAST RUN ENTRIES *** (Y=YES,
N=NO) ? ", LastEntry$
   IF UCASE$(LastEntry$) = "Y" THEN GOTO PAST2

INPUTCHG:
    INPUT "RUN # ="; RUNNUM$
    INPUT "STATION #"; STATION
    INPUT "TEMPERATURE (C) ="; TEMP
    GOTO INTIME
INPOINTA:
          INPUT "TIME(24hr.)    HOUR = ?"; HOUR$
          IF VAL(HOUR$) > 24 OR VAL(HOUR$) < 0 THEN GOTO INPOINTA
INPOINTB:
          INPUT " MIN. = ?"; MIN$
          IF VAL(MIN$) > 60 OR VAL(MIN$) < 0 THEN GOTO INPOINTB
INPOINTC:
          INPUT "SEC. = ?"; SEC$
          IF VAL(SEC$) > 60 OR VAL(SEC$) < 0 THEN GOTO INPOINTC
          TIMEXP$ = HOUR$ + ":" + MIN$ + ":" + SEC$: PRINT TIMEXP$
          TIME$ = TIMEXP$
INTIME:
    TIMEXP$ = TIME$: PRINT "THE TIME IS ", TIMEXP$
    INPUT " IS THIS CORRECT Y=YES  N=NO  ? ", CORECT$
    IF UCASE$(CORECT$) = "N" THEN GOTO INPOINTA
    GOTO INDATE
INDATS:
          INPUT "THE DATE IS ?  MONTH = ? ", MONTH$
```

```
        IF VAL(MONTH$) > 12 OR VAL(MONTH$) < 0 THEN GOTO INDATS
INDATS1:
        INPUT "THE DATE IS ?  DAY = ? ", DAY$
        IF VAL(DAY$) > 31 OR VAL(DAY$) < 0 THEN GOTO INDATS1
INDATS2:

        INPUT "THE DATE IS ?  YEAR = ? ", YEAR$
        IF VAL(YEAR$) > 1995 OR VAL(YEAR$) < 1989 THEN GOTO
INDATS2
        DT$ = MONTH$ + "/" + DAY$ + "/" + YEAR$
        DATE$ = DT$
INDATE:
    DT$ = DATE$: PRINT "THE DATE IS ", DT$
    INPUT " IS THIS CORRECT Y=YES  N=NO  ? ", CORECT$
    IF CORECT$ = "N" THEN GOTO INDATS
INCMD:
    PRINT "WHICH MACHINE(S) WILL BE USED IN THIS RUN"
    INPUT "     (1=YES, 0=NO)    400Kv"; Z(1): INPUT "
       THE AN2500"; Z(2): INPUT "                THE AN5000";
Z(3)
    INPUT "TARGET DOSE AND TERMINATION CONTROLLED BY 1=400Kv,
2=AN2500, OR 3=AN5000"; Z(4)
    IF Z(Z(4)) <> 1 THEN PRINT "THIS MACHINE NOT USED ! ":  GOTO
INCMD
    DOSE$(1) = " 400Kv": DOSE$(2) = " AN2500": DOSE$(3) = " AN5000"
    LINE INPUT "THE USER AND HIS SPECIMEN ID# FOR THIS RUN (1 LINE) = ";
USER$
    INPUT "THE ATOMIC DENSITY ( x 10^-22 ) OF THE TARGET IS = ";
C(6)
    IF Z(1) = 1 THEN
      PRINT " "
      INPUT "400 KV MACHINE VOLTAGE, USE AVERAGE IF RAMPING (KV)
= "; C(17)
      INPUT " "  "    "  RAMP INTERVAL RANGE (MAZ. - MIN.) IN
KV = "; C(18)
      INPUT " "  "    "  PROJECTILE ION ATOMIC MASS NUMBER =
"; C(19)
      INPUT " "  "    "  CHARGE STATE FOR THE 400 PROJECTILE
ION = "; C(24)
      PRINT " "  "    "  DOSE (1 = IONS/CM^2)"
      PRINT "                (2 = DPA) OR "
      INPUT "                (3 = APPM) = "; Z(5)
      IF Z(5) = 2 THEN INPUT "MICROCOULOMB/DPA FOR 400 KV (# INPUT IS FOR
SINGLY CHARGED IONS)", C(13)

       IF Z(4) = 1 THEN                        ' nested
```

```
if statement
        INPUT "FINAL TARGET DOSE IS = ?", FINALDOSE     ' nested if statement
        ING = Z(5)
        END IF                                  ' nested if statement
      C(7) = C(7) * (1 + C(18) / 165)                   ' M. Lewis added correlation
    ELSE
   END IF
    IF Z(2) = 1 THEN
      PRINT " "
      INPUT "2500 KV MACHINE VOLTAGE (KV) = "; C(20)
      INPUT " "   "   "  PROJECTILE ION ATOMIC MASS NUMBER = "; C(21)
      INPUT " "   "   "  CHARGE STATE FOR THE 2500 PROJECTILE ION = "; C(25)
      PRINT " "   "   "  DOSE (1 = IONS/CM^2)"
      PRINT "                (2 = DPA) OR "
      INPUT "                (3 = APPM) = "; Z(6)
      IF Z(6) = 2 THEN INPUT "MICROCOULOMB/DPA FOR 2500 KV (# INPUT IS FOR
SINGLY CHARGED IONS)", C(12)


        IF Z(4) = 2 THEN                        ' nested if statement
        INPUT "FINAL TARGET DOSE IS = ?", FINALDOSE     ' nested if statement
        ING = Z(6)
        END IF                                  ' nested if statement

    ELSE
   END IF
    IF Z(3) = 1 THEN
      PRINT " "
      INPUT "5 MV MACHINE VOLTAGE (MV) = "; C(22)
      INPUT " "     "  PROJECTILE ION ATOMIC MASS NUMBER =
"; C(23)
      INPUT " "     "  CHARGE STATE FOR THE 5MV PROJECTILE
ION = "; C(26)
      PRINT " "     "  DOSE (1 = IONS/CM^2)"
      PRINT "                (2 = DPA) OR "
      INPUT "                (3 = APPM) = "; Z(7)
      IF Z(7) = 2 THEN INPUT "MICROCOULOMB/DPA FOR 5 MV (# INPUT IS FOR
SINGLY CHARGED IONS)", C(11)


        IF Z(4) = 3 THEN                        ' nested
if statement
        INPUT "FINAL TARGET DOSE IS = ?", FINALDOSE     ' nested
if statement
        ING = Z(7)
        END IF                                  ' nested
```

if statement

```
    ELSE
   END IF
     ESOD$(1) = "(IONS/CM^2) x 10^15": SOD$(1) = " I ONS  PER SQUARE
CENTIMETER "
     ESOD$(2) = "(DPA)": SOD$(2) = " D P A "
     ESOD$(3) = "(APPM)": SOD$(3) = " A  P P M "
    ON Z(4) GOSUB 10, 20, 30
     GOTO PAST1
10 :
   ON Z(5) GOSUB 11, 21, 31
   RETURN
11 :
   TGDOS = FINALDOSE
   RETURN
21 :
   ' Convert To Ions/Cm^2 From DPA 400
   TGDOS = FINALDOSE * C(13) * 6.225E+12
   RETURN
31 :
   ' Convert To Ions/Cm^2 From APPM 400
   TGDOS = FINALDOSE * C(6) * C(7) * 2.5E+12
   RETURN
20 :
   ON Z(6) GOSUB 12, 22, 32
   RETURN
12 :
   TGDOS = FINALDOSE
   RETURN
22 :
   ' Convert To Ions/Cm^2 From DPA 2500
   TGDOS = FINALDOSE * C(12) * 6.225E+12
   RETURN
32 :
   ' Convert To Ions/Cm^2 From APPM 2500
   TGDOS = FINALDOSE * C(6) * C(4) * 2.5E+12
   RETURN
30 :
   ON Z(7) GOSUB 13, 23, 33
   RETURN
13 :
   TGDOS = FINALDOSE
   RETURN
23 :
   ' Convert To Ions/Cm^2 From DPA 5 MV
```

```
    TGDOS = FINALDOSE * C(11) * 6.225E+12
    RETURN
33 :
    ' Convert To Ions/Cm^2 From APPM 5 MV
    TGDOS = FINALDOSE * C(6) * C(16) * 2.5E+12
    RETURN

PAST1:
    GOTO PAST3
PAST2:
    OPEN "A:\PARAM" FOR INPUT AS #4
    FOR I = 1 TO 26
    INPUT #4, C(I)
    NEXT I
    FOR I = 1 TO 10
    INPUT #4, Z(I)
    NEXT I
    INPUT #4, RUNNUM$, STATION, TEMP, USER$, FINALDOSE, ESOD$(1), ESOD$(2),
ESOD$(3), TGDOS, SOD$(1), SOD$(2), SOD$(3), DOSE$(Z(4))
    CLOSE #4
    TIMEXP$ = TIME$: DT$ = DATE$

PAST3:

     CLS

     PRINT "THIS RUN WILL BE"
     PRINT "   RUN #     ", RUNNUM$
     PRINT "   STATION # ", STATION
     PRINT "   TEMP. (C) ", TEMP
     PRINT "   THE TIME IS ", TIMEXP$
     PRINT "   DATE      ", DT$
     PRINT "   USING     ";
     IF Z(1) = 1 THEN PRINT " 400Kv,";  ' IF Z(1) = 1 THE 400KV MACHINE WILL BE
USED
     IF Z(2) = 1 THEN PRINT " AN2500";  ' IF Z(2) = 1 THE AN2500 MACHINE WILL BE
USED
     IF Z(3) = 1 THEN PRINT ", AN5000";  ' IF Z(3) = 1 THE AN5000 MACHINE WILL
BE USED
     PRINT "   DOSE CONTROL"; DOSE$(Z(4))
     PRINT "   USER ID.  "; USER$
     PRINT "   ATOMIC DENSITY (x 10^-22) OF TARGET "; C(6)
    IF Z(1) = 1 THEN
      PRINT "   400 KV MACHINE VOLTAGE (KV) IS "; C(17)
      PRINT "   400 RAMP INTERVAL RANGE IS "; C(18)
      PRINT "   400 PROJECTILE ION ATOMIC MASS NUMBER IS "; C(19)
```

```
    PRINT "   400 DOSE IN "; ESOD$(Z(5))
  ELSE
 END IF
  IF Z(2) = 1 THEN
      PRINT "   2500 KV MACHINE VOLTAGE (KV) IS "; C(20)
      PRINT "   2500 PROJECTILE ION ATOMIC MASS NUMBER IS "; C(21)
      PRINT "   2500 DOSE IN "; ESOD$(Z(6))
     ELSE
    END IF
   IF Z(3) = 1 THEN
        PRINT "   5 MV MACHINE VOLTAGE (MV) IS "; C(22)
        PRINT "   5 MV PROJECTILE ION ATOMIC MASS NUMBER IS "; C(23)
        PRINT "   5 MV DOSE IN "; ESOD$(Z(7))
     ELSE
   END IF
        INPUT "DO YOU WANT TO HEAR READBACK (Y=YES, N=NO) ? ", Q$
        IF UCASE$(Q$) = "N" THEN GOTO NOHEAR
    OPEN "O", 3, "HV"
    PRINT #3, "THIS  RUN  WILL  BE  RUN  NUMBER  "; RUNNUM$
    PRINT #3, "   USING STATION "; STATION
    PRINT #3, "    OPERATING  AT  "; TEMP; "  DEGREASE   SEE      "
    PRINT #3, "   THE TIME IS "; TIMEXP$
    PRINT #3, " THE  DATE  IS   "; DT$
    PRINT #3, " USING";
    IF Z(1) = 1 THEN PRINT #3, "    THE  400 K E V";
    IF Z(2) = 1 THEN PRINT #3, "    THE A N 20 500 ";
    IF Z(3) = 1 THEN PRINT #3, "    THE A N 5000 ";
    PRINT #3, "       THE  TOE TAL  DOSE  WILL  BE  CONTROLLED  BY  THE ";
DOSE$(Z(4))
    PRINT #3, "       THE  TOE TAL  DOSE IS  "; FINALDOSE
    PRINT #3, " USER I D  IS   "; USER$
    PRINT #3, " THE ATOMIC DENSITY OF THE TAR GET IS "; C(6)
  IF Z(1) = 1 THEN
   PRINT #3, "  THE 400 K E V MASH SHENE VOLTAGE  "; C(17)
   PRINT #3, "  THE RAMP INTERVAL RANGE  "; C(18)
   PRINT #3, "  AND THE PROJECTILE   I ON   ATOMIC   MAS   NUMBER   IS ";
C(19)
   PRINT #3, "  THE PROJECTILE I ON CHARGE STATE IS  PLUS "; C(24)
   PRINT #3, "  THE DOSE UNITS IN "; SOD$(Z(5))
     ELSE
   END IF
  IF Z(2) = 1 THEN
      PRINT #3, "  THE  A N 20 500 MASH SHENE VOLTAGE IS "; C(20)
      PRINT #3, "  THE A N 20 500 PROJECTILE   I ON   ATOMIC   MAS   NUMBER
 IS "; C(21)
      PRINT #3, "  THE PROJECTILE I ON CHARGE STATE IS  PLUS "; C(25)
```

```
      PRINT #3, "  AND THE DOSE UNITS IN "; SOD$(Z(6))
       ELSE
      END IF
       IF Z(3) = 1 THEN
         PRINT #3, "  THE A N 5000 MASH SHENE VOLTAGE IS "; C(22)
         PRINT #3, "  THE PROJECTILE    I ON    ATOMIC    MAS    NUMBER    IS ";
C(23)
         PRINT #3, "  THE PROJECTILE I ON CHARGE STATE IS  PLUS "; C(26)
         PRINT #3, "  AND THE DOSE UNITS IN "; SOD$(Z(7))
      ELSE
   END IF
   CLOSE #3
NOHEAR:
   PRINT " "
   INPUT "   DO YOU WANT TO MAKE ANY CHANGES ( Y = YES, N = NO ) ?",
RUNCHG$
   IF UCASE$(RUNCHG$) = "Y" THEN GOTO INPUTCHG

DEFLT.CHG:
   CLS
   PRINT "THE PROGRAM DEFAULT OPTIONS ARE "
   PRINT " "
   PRINT "  (1)  USED IN PROGRAM IN 3852A CONTROLLER = "; C(1)
   PRINT "  (2)  AUXILIARY INPUT FOR FUTURE USE "; C(2)
   PRINT "  (3)  AUXILIARY INPUT FOR FUTURE USE ", C(3)
   PRINT "  (4)  DEVIATION OF DISTRIBUTION OF INPLANTED IONS FOR 2.5 MV
BEAM = "; C(4)
   PRINT "  (5)  AUXILIARY INPUT FOR FUTURE USE = "; C(5)
   PRINT "  (6)  AN INPUT VALUE FOR ATOMIC DENSITY ( x 10^-22 ) OF THE
TARGET WAS = "; C(6)
   PRINT "  (7)  DEVIATION OF DISTRIBUTION OF INPLANTED IONS FOR 400 KV
BEAM = "; C(7)
   PRINT "  (8)  AUXILIARY INPUT FOR FUTURE USE  "; C(8)
   PRINT "  (9)  RECALIBRATION FACTOR FOR THE 400 KV BEAM EQUATIONS =
"; C(9)
   PRINT "  (10) RECALIBRATION FACTOR FOR THE 2.5 MV BEAM EQUATIONS =
"; C(10)
   PRINT "   (11) MICROCOULOMB/DPA IN 5.0 MV BEAM CALCULATION (+1
CHARGED IONS) = "; C(11)
   PRINT "   (12) MICROCOULOMB/DPA IN 2.5 MV BEAM CALCULATION (+1
CHARGED IONS) = "; C(12)
   PRINT "   (13) MICROCOULOMB/DPA IN 400 KV BEAM CALCULATION (+1
CHARGED IONS) = "; C(13)
   PRINT "  (14) RECALIBRATION FACTOR FOR THE 5.0 MV BEAM EQUATIONS =
"; C(14)
   PRINT "  (15) AUXILIARY INPUT FOR FUTURE USE  "; C(15)
```

```
    PRINT "  (16) DEVIATION OF DISTRIBUTION OF INPLANTED IONS FOR 5 MV
BEAM = "; C(16)
    PRINT "  (24) CHARGE STATE FOR THE 400KV PROJECTILE ION = "; C(24)
    PRINT "  (25) CHARGE STATE FOR THE 2500KV PROJECTILE ION = "; C(25)
    PRINT "  (26) CHARGE STATE FOR THE 5MV PROJECTILE ION = "; C(26)
INPUT "INDICATE HOW MANY DEFAULT CHANGES YOU WANT TO MAKE =";
NUMBERCHG
INNUCHG:
    IF NUMBERCHG < 1 THEN GOTO NUCHGPT
INPUT "INDICATE BY # IN ( ) WHICH DEFAULT TO CHANGE = "; SUBSCPT
INPUT "WHAT VALUE WOULD YOU RATHER USE = "; NEWVALUE
    C(SUBSCPT) = NEWVALUE
    NUMBERCHG = NUMBERCHG - 1: GOTO INNUCHG
NUCHGPT:
    GOSUB DISKMGR

    GOSUB RETCRT
    Q = 0
RETURN

READ.CUP:
      COLOR 15, 1
      GOSUB RETCRT
      GOSUB DEEPCUP
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0
RETURN

CHANGE:
      COLOR 15, 1
      GOSUB RETCRT
      GOTO INPUTCHG
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0
RETURN
DEFAULT.CHG:
      COLOR 15, 1
      GOSUB RETCRT
      GOTO DEFLT.CHG
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0
RETURN
```

```
ACTIVE.CONT:
      COLOR 15, 1
      GOSUB RETCRT
      INPUT "IN WHICH CHAMBER WILL THE VACUUM BE OBSERVED (1) FOR
LOADING (2) FOR TARGET  ?=", KY
      OPEN "COM1:300,N,7,2,RS" FOR RANDOM AS #11
      IF KY = 1 THEN FART$ = "IG2 ON" ELSE FART$ = "IG1 ON"
      PRINT #11, FART$ + CHR$(13) + CHR$(10)
      CLOSE #11
      PRINT " THIS PORTION OF THE PROGRAM WILL SPECIFY EQUIPMINT TO BE
USED IN THIS RUN"
      IF KY = 2 THEN TLK$ = "  TAR GET   CHAMBER "
      OPEN "O", 1, "HV"
      IF KY = 1 THEN TLK$ = " LOADING CHAMBER "
      PRINT #1, " YOU  HAVE  CHOSEN THE   I ON   GAYGE   FOR  THE  "; TLK$
      CLOSE #1
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0
RETURN

MANUAL:
     COLOR 15, 1
     GOSUB RETCRT
     ' SHOW THE PARTS THAT CAN BE CONTROLLED MANUALLY, REPORT
CURRENT STATUS, INQUIRE WHAT IS TO BE DONE AND BY WHAT PART.
     PRINT "THIS IS THE MANUAL MODE"
     GOSUB DOLITTLE
     GOSUB RETCRT
     Q = 0
RETURN
TRAIN:
     COLOR 15, 1
     GOSUB RETCRT
     PRINT "THIS IS THE TRAIN CONTROL SYSTEM"
     GOSUB DOLITTLE
     GOSUB RETCRT
     Q = 0
RETURN
TEMP:
     COLOR 15, 1
     GOSUB RETCRT
     PRINT "THIS SECTION CONTROLAS THE TEMPERATURE"
     GOSUB DOLITTLE
     GOSUB RETCRT
     Q = 0
```

```
RETURN

MONITOR:
    COLOR 15, 1
    GOSUB RETCRT
    PRINT "THIS WINDOW SHOWS THE STATUS OF THE SYSTEM"
    GOSUB DOLITTLE
    GOSUB RETCRT
    Q = 0
RETURN

EVENT18:
    COLOR 15, 1
    GOSUB RETCRT
    PRINT "THIS WINDOW IS FOR FURTURE EXPANSION"
    GOSUB DOLITTLE
    GOSUB RETCRT
    Q = 0
RETURN


MNU2:
    IF ROW < 8 THEN
    GOSUB RETCRT
    Q = 0: L = 0: DRT = 0
    RETURN
      ELSEIF COL < 11 OR COL > 19 OR ROW > 20 THEN
      RETURN
        ELSE
        ON INT(((ROW * 3) + 7) / 10) - 2 GOTO VALVES, INSTRU, SUP1, SUP2
          END IF
          RETURN
VALVES:
      COLOR 15, 1
      GOSUB RETCRT
      PRINT "THIS SECTION WILL OPEN OR CLOSE VALVES IN THE SYSTEM"
      GOSUB VALVEACTUATOR
      GOSUB DOLITTLE
      GOSUB RETCRT
    Q = 0: L = 0: DRT = 0
    RETURN
VALVEACTUATOR:

    RETURN

INSTRU:
```

```
      COLOR 15, 1
      GOSUB RETCRT
      PRINT " THIS SECTION WILL OPERATE INSTRUMENTS IN THE SYSTEM"
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0: L = 0: DRT = 0
      RETURN
SUP1:
      COLOR 15, 1
      GOSUB RETCRT
      PRINT " THIS SECTION WILL INSERT MINI-CUPS INTO THE BEAM PATH"
    OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
    OPEN "\DEV\IEEEIN" FOR INPUT AS #2

A$ = "RST"
CALL OUTDO(A$)

      A$ = "CLOSE 1206"
      CALL OUTDO(A$)
      CLOSE #1
      CLOSE #2
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0: L = 0: DRT = 0
      RETURN

SUP2:
      COLOR 15, 1
      GOSUB RETCRT
     PRINT " THIS SECTION WILL REMOVE MINI-CUPS OUT OF THE BEAM PATH"
    OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
    OPEN "\DEV\IEEEIN" FOR INPUT AS #2

A$ = "RST"
CALL OUTDO(A$)

      A$ = "OPEN 1206"
      CALL OUTDO(A$)
      CLOSE #1
      CLOSE #2

      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0: L = 0: DRT = 0
      RETURN
```

```
MNU4:
    IF ROW < 8 THEN
    GOSUB RETCRT
    Q = 0: L = 0: DRT = 0
    RETURN
      ELSEIF COL < 31 OR COL > 39 OR ROW > 20 THEN
      RETURN
        ELSE
        ON INT(((ROW * 3) + 7) / 10) - 2 GOTO STATUS, MOVE, OPTION1, SUP3
          END IF
          RETURN

STATUS:

    COLOR 15, 1
    GOSUB RETCRT
    PRINT "THIS SECTION WILL BE USED TO SHOW POSITION AND STATUS OF
TRANSFER SYSTEM"
    GOSUB DOLITTLE
    GOSUB RETCRT
    Q = 0: L = 0: DRT = 0
    RETURN

MOVE:
    COLOR 15, 1
    GOSUB RETCRT
    PRINT " THIS SECTION WILL ESTABLISH PARAMETERS AND INITIATE"
    PRINT " TRANSFER OF SPECIMENS                              "
    GOSUB DOLITTLE
    GOSUB RETCRT
    Q = 0: L = 0: DRT = 0
    RETURN

OPTION1:
    COLOR 15, 1
    GOSUB RETCRT
    PRINT " THIS SECTION WILL ALLOW PULSE STEPPING OF TRANSFER
SYSTEM"
    GOSUB DOLITTLE
    GOSUB RETCRT
    Q = 0: L = 0: DRT = 0
    RETURN

SUP3:
    COLOR 15, 1
    GOSUB RETCRT
```

```
      PRINT " THIS SECTION WILL ACCOMIDATE FUTURE GROWTH"
      PRINT " CHANGES CAN BE MADE USING THE MENU SETUP                    "
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0: L = 0: DRT = 0
      RETURN

MNU3:
     IF ROW < 8 THEN
     GOSUB RETCRT
     Q = 0: L = 0: DRT = 0
     RETURN
       ELSEIF COL < 21 OR COL > 29 OR ROW > 20 THEN
       RETURN
         ELSE
         ON INT(((ROW * 3) + 7) / 10) - 2 GOTO STARTRUN, RESTART, CHGPARM,
LSTCONT
           END IF
           RETURN

STARTRUN:
       SIGMD = 0
       SIGME = 0
       SIGMF = 0
       TOTLDOS = 0
       FOR I = 0 TO 4
       FSD(I) = 0
       FSE(I) = 0
       FSF(I) = 0
       NEXT I

       SIGMDLT = 0
       ENTC = 1
       PRNT = 1

      COLOR 15, 1
      GOSUB RETCRT
    CLS

   GOSUB RUNDWNLOD
   GOSUB PARMPAS

RUNLOOP:
       OPEN "O", 3, "HV"

    A$ = "RQS ON"
```

```
    CALL OUTDO(A$)
    A$ = "RQS FPS"
    CALL OUTDO(A$)

    A$ = "STA?"
    CALL OUTDO(A$)
      PRINT #1, "ARM SRQ"
      ON PEN GOSUB SERVICEINTERUPT
       PRINT #1, "ABORT"
       PEN ON

 '*** DOWNLOAD PARAMETERS NEEDED IN SUBS  ( SEE PARAMPAS: )

    A$ = "CALL START"
    CALL OUTDO(A$)

       PONS = 1
        GOTO PRINTSCREEN

SERVICEINTERUPT:

       A$ = "VREAD ENTC;"
       CALL OUTDO(A$)
       PRINT #1, "ENTER09;"
       INPUT #2, ENTC

      ON ENTC GOTO DIR1, DIR2, DIR2, DIR2

      GOTO BYPASS
DIR1:
       BEEP

       BEEP
       A$ = "VREAD DSTAT;"            'STATICS MIN. MAX. MEAN. STD DEV.
       CALL OUTDO(A$)
       PRINT #1, "ENTER09;EOI"
       INPUT #2, DSTAT(0), DSTAT(1), DSTAT(2), DSTAT(3), Q1

       A$ = "VREAD ESTAT;"            'STATICS MIN. MAX. MEAN. STD DEV.
       CALL OUTDO(A$)
       PRINT #1, "ENTER09; EOI"
       INPUT #2, ESTAT(0), ESTAT(1), ESTAT(2), ESTAT(3), Q1

       A$ = "VREAD FSTAT;"            'STATICS MIN. MAX. MEAN. STD DEV.
       CALL OUTDO(A$)
       PRINT #1, "ENTER09; EOI"
```

```
INPUT #2, FSTAT(0), FSTAT(1), FSTAT(2), FSTAT(3), Q1

A$ = "VREAD AC;"                'AVERAGE CUP COUNTS
CALL OUTDO(A$)
PRINT #1, "ENTER09; EOI"
INPUT #2, AC(0), AC(1), AC(2), Q1

A$ = "VREAD AB;"                'AVERAGE BPM COUNTS
CALL OUTDO(A$)
PRINT #1, "ENTER09; EOI"
INPUT #2, AB(0), AB(1), AB(2), Q1

A$ = "VREAD CN4;"               'CALIBRATION FACTORS
CALL OUTDO(A$)
PRINT #1, "ENTER09; EOI"
INPUT #2, CN4(0), CN4(1), CN4(2), Q1

A$ = "VREAD VARD;"              'DISTRIBUTION OF BPM 400
CALL OUTDO(A$)
PRINT #1, "ENTER09; EOI"
INPUT #2, VARD(0), VARD(1), VARD(2), VARD(3), VARD(4), Q1

A$ = "VREAD VARE;"              'DISTRIBUTION OF BPM 2500
CALL OUTDO(A$)
PRINT #1, "ENTER09; EOI"
INPUT #2, VARE(0), VARE(1), VARE(2), VARE(3), VARE(4), Q1

A$ = "VREAD VARF;"              'DISTRIBUTION OF BPM  5 MV
CALL OUTDO(A$)
PRINT #1, "ENTER09; EOI"
INPUT #2, VARF(0), VARF(1), VARF(2), VARF(3), VARF(4), Q1
 A$ = "VREAD D"                 'CUP COUNTS 2 SEC. INTERVAL 400KV
CALL OUTDO(A$)
PRINT #1, "ENTER09; EOI"
INPUT #2, D(0), D(1), D(2), D(3), D(4), Q1
A$ = "VREAD E"                  'CUP COUNTS 2 SEC. INTERVAL 2500KV
CALL OUTDO(A$)
PRINT #1, "ENTER09; EOI"
INPUT #2, E(0), E(1), E(2), E(3), E(4), Q1

A$ = "VREAD F"                  'CUP COUNTS 2 SEC. INTERVAL 5MV
CALL OUTDO(A$)
PRINT #1, "ENTER09; EOI"
INPUT #2, F(0), F(1), F(2), F(3), F(4), Q1

A$ = "VREAD DELTIM;"
```

```
      CALL OUTDO(A$)
      PRINT #1, "ENTER09;"
      INPUT #2, DELTIM

       ' READ VACUUM AND TEMPERATURE HERE

       A$ = "VREAD LC2;"
      CALL OUTDO(A$)
      PRINT #1, "ENTER09;"
      INPUT #2, LC2
   GOSUB RESETINTERUPT

     GOSUB DISKCUPUPDATE

     A$ = "CALL THREE"
    CALL OUTDO(A$)

         FOR I = 0 TO 4

        IF VARD(I) = 0 AND Z(1) = 1 THEN
          OPEN "O", 7, "HV"
          PRINT #7, " THE 400 BEAM IS OFF OR VERY LOW"
          CLOSE #7
        END IF
        '24.9128 CONVERTS AREA OF CUP TO 1 CM^2 * .1 (10^-10 TO 10^-9) =
2.49128
       CMD(I) = (D(I) / 2) * 2.49128  ' 400 KV CUP CURRENTS

        IF VARE(I) = 0 AND Z(2) = 1 THEN
          OPEN "O", 7, "HV"
          PRINT #7, " THE 2500 BEAM IS OFF OR VERY LOW"
          CLOSE #7
         END IF
        '24.9128 CONVERTS AREA OF CUP TO 1 CM^2 * .1 (10^-10 TO 10^-9) =
2.49128
        CME(I) = (E(I) / 2) * 2.49128   ' 2500 KV CUP CURRENTS

        IF VARF(I) = 0 AND Z(3) = 1 THEN
          OPEN "O", 7, "HV"
          PRINT #7, " THE 5 M E V BEAM IS OFF OR VERY LOW"
          CLOSE #7

        END IF
        '24.9128 CONVERTS AREA OF CUP TO 1 CM^2 * .1 (10^-10 TO 10^-9) =
2.49128
        CMF(I) = (F(I) / 2) * 2.49128   ' 5MV CUP CURRENTS
```

```
        NEXT I
      FOR I = 0 TO 2
        '24.9128 CONVERTS AREA OF CUP TO 1 CM^2 * .1 (10^-10 TO 10^-9) =
2.49128
        CMAC(I) = (AC(I) / 2) * 2.49128   ' AVERAGES CUP CURRENTS
      NEXT I

        PONS = 2
        RETURN

DIR2:

   BEEP
   A$ = "VREAD SCUP;"            ' CONVERTED BPM TO SIMULATED CUP COUNTS
   CALL OUTDO(A$)
      PRINT #1, "ENTER09; EOI"
      INPUT #2, SCUP(0), SCUP(1), SCUP(2), SCUP(3), Q1
   A$ = "VREAD DLT;"              ' DELTA TIME
   CALL OUTDO(A$)
      PRINT #1, "ENTER09; "
      INPUT #2, DLT

   A$ = "VREAD TOTLDOS;"          ' TARGET TOTAL ACCUMULATED DOSE
   CALL OUTDO(A$)
      PRINT #1, "ENTER09; "
      INPUT #2, TOTLDOS
   A$ = "VREAD LC1;"              ' LOOP COUNTER
   CALL OUTDO(A$)
      PRINT #1, "ENTER09; "
      INPUT #2, LC1

    GOSUB RESETINTERUPT
      'PULSES/SEC(10^-10 COULOMB/PULSE)(.625E+19 IONS/COULOMB/CHARGE
STATE OF ION)(24.91285 CUP AREA/CM^2) = 1.557053E+10 IONS/CM^2/SEC
      'SCUP IS SIMULATED CUP COUNTS/SEC DERIVED FROM CONVERSION
FACTOR MEASURED EVERY 10 MIN DURING THE RUN.
      'DLT IS DELTA TIME IN SEC
      'C(24), C(25), C(26)  IS THE CHARGE STATES OF THE RESPECTIVE IONS
      CONVD = SCUP(1) * 1.55705E+10 * DLT / C(24)    ' COUNTS TO IONS/CM^2
400
      CONVE = SCUP(2) * 1.55705E+10 * DLT / C(25)    '  "    "    "    2500
      CONVF = SCUP(3) * 1.55705E+10 * DLT / C(26)    '  "    "    "    5 MV

   '24.9128 CONVERTS AREA OF CUP TO 1 CM^2 * .1 (10^-10 TO 10^-9) = 2.49128
   SIMCURD = SCUP(1) * 2.49128   ' 400 KV CUP CURRENTS (SIMULATED)
```

```
SIMCURE = SCUP(2) * 2.49128   ' 2500 KV CUP CURRENTS (SIMULATED)

SIMCURF = SCUP(3) * 2.49128   ' 5000 KV CUP CURRENTS (SIMULATED)

A$ = "CONFMEAS TEMPK,1800,USE 1900"
CALL OUTDO(A$)
PRINT #1, "ENTER09;"
INPUT #2, btemp
PRINT #1, "ENTER09;"
INPUT #2, btemp

    SIGMD = SIGMD + CONVD
    SIGME = SIGME + CONVE
    SIGMF = SIGMF + CONVF

    FOR I = 0 TO 4
    FSD(I) = FSD(I) + CONVD * VARD(I)
    FSE(I) = FSE(I) + CONVE * VARE(I)
    FSF(I) = FSF(I) + CONVF * VARF(I)
    NEXT I

    SIGMDLT = SIGMDLT + DLT

    IF ENTC = 4 THEN
      GOSUB DISKUPDATE
      CLOSE #1
      CLOSE #2
      CLOSE #3
      PONS = 2
      RETURN
    END IF
  IF ENTC = 3 THEN
   A$ = "CALL CUP_BPM"
   CALL OUTDO(A$)
  ELSE

   A$ = "CALL THREE"
   CALL OUTDO(A$)

  END IF


   OPEN "COM1:300,N,7,2,RS" FOR RANDOM AS #12
   TURD$ = "DS IG"
   PRINT #12, TURD$ + CHR$(13) + CHR$(10)
   SLEEP 1
```

```
      INPUT #12, VAC$
      CLOSE #12
        GOSUB DISKUPDATE
        PONS = 3
        RETURN

PARMPAS:

      FOR I = 0 TO 10
        A$ = "Z(" + STR$(I) + ")=" + STR$(Z(I))
       CALL OUTDO(A$)
      NEXT I
         C(1) = C(23 + Z(4))

      FOR I = 0 TO 26
        A$ = "C(" + STR$(I) + ")=" + STR$(C(I))
        CALL OUTDO(A$)
      NEXT I
     A$ = "TGDOS = " + STR$(TGDOS)
     CALL OUTDO(A$)
      A$ = "TOTLDOS = " + STR$(TOTLDOS)
      CALL OUTDO(A$)
    OPEN "A:\PARAM" FOR OUTPUT AS #4
    FOR I = 1 TO 26
    WRITE #4, C(I)
    NEXT I
    FOR I = 1 TO 10
    WRITE #4, Z(I)
    NEXT I
    WRITE #4, RUNNUM$, STATION, TEMP, USER$, FINALDOSE, ESOD$(1), ESOD$(2),
ESOD$(3), TGDOS, SOD$(1), SOD$(2), SOD$(3), DOSE$(Z(4))
    CLOSE #4
   RETURN

RESETINTERUPT:

     PEN OFF
     A$ = "STA?;STB?;"
     CALL OUTDO(A$)

     PRINT #1, "RESET"
     A$ = "RQS ON"
     CALL OUTDO(A$)
     A$ = "RQS FPS"
     CALL OUTDO(A$)
       PRINT #1, "ARM SRQ"
```

```
      ON PEN GOSUB SERVICEINTERUPT
       PRINT #1, "ABORT"
       PEN ON

RETURN

DISKMGR:

    CLS
      INPUT "Load the DATA Diskette In Drive A:\ and Press ENTER Key ", Z
      INPUT "IS THIS A NEW DATA DISK (Y=YES, N=NO) ? ", Q$
      IF UCASE$(Q$) = "Y" THEN
      DSKRUN$ = STR$(VAL(RIGHT$(DATE$, 2)) * 1000 + VAL(LEFT$(DATE$, 2)) *
1000000)
      GOTO PAST4
      ELSE
      END IF
      OPEN "A:\INDEX" FOR INPUT AS #5
         INPUT #5, DSKRUN$
      CLOSE #5
PAST4:

      DSKRUN$ = STR$(VAL(DSKRUN$) + 1)

         InRegs.AX = &H3600
         InRegs.DX = &H1
          CALL INTERRUPT(&H21, InRegs, OutRegs)
         SectPerClust = OutRegs.AX: ClustPerDriv = OutRegs.DX
         BytesPerSect = OutRegs.CX: AvailClust = OutRegs.BX
          IF ClustPerDrive = &HFFFF THEN GOTO ErDsk
          AvBytes = SectPerClust * BytesPerSect * AvailClust
           PRINT "DISK SPACE AVAILABLE IS "; AvBytes
           IF AvBytes < 10000 THEN

     OPEN "A:\PARAM" FOR INPUT AS #4
    FOR I = 1 TO 26
    INPUT #4, C(I)
    NEXT I
    FOR I = 1 TO 10
    INPUT #4, Z(I)
    NEXT I
    INPUT #4, RUNNUM$, STATION, TEMP, TIMEXP$, DT$, USER$, FINALDOSE,
ESOD$(1), ESOD$(2), ESOD$(3), TGDOS, SOD$(1), SOD$(2), SOD$(3)
    CLOSE #4
            INPUT "Disk is Full. Replace With a New Formated 2S\HD Disk and Press ENTER
key ", Z
```

```
OPEN "A:\PARAM" FOR OUTPUT AS #4
FOR I = 1 TO 26
WRITE #4, C(I)
NEXT I
FOR I = 1 TO 10
WRITE #4, Z(I)
NEXT I
WRITE #4, RUNNUM$, STATION, TEMP, TIMEXP$, DT$, USER$, FINALDOSE,
ESOD$(1), ESOD$(2), ESOD$(3), TGDOS, SOD$(1), SOD$(2), SOD$(3)
CLOSE #4

        ELSE
        END IF

      OPEN "A:\INDEX" FOR OUTPUT AS #6
        WRITE #6, DSKRUN$
      CLOSE #6

    OPEN "A:\CATALOG" FOR APPEND AS #7
      WRITE #7, DSKRUN$, RUNNUM$, DATE$
    CLOSE #7

    SKRUN$ = STR$(10000000000# + VAL(DSKRUN$))
    CURDSK$ = "A:\" + (RIGHT$(SKRUN$, 8))
    ACURDSK$ = LEFT$(CURDSK$, 7) + "A" + RIGHT$(CURDSK$, 3)
    OPEN CURDSK$ FOR OUTPUT AS #9

      IDENT1$ = "The File Name is " + CURDSK$
      IDENT2$ = "The Date is " + DATE$
      IDENT3$ = "The Run ID is " + RUNNUM$
      IDENT35$ = "The User and Specimen is " + USER$
      IDENT4$ = "The Station is " + STR$(STATION)
      IDENT5$ = "The Operating Temperature is " + STR$(TEMP)
      IDENT6$ = "The C(0) to C(25) and Z(0) to Z(10) Parameters are "
       WRITE #9, IDENT1$, IDENT2$, IDENT3$, IDENT35$, IDENT4$, IDENT5$,
IDENT6$
        FOR I = 0 TO 26
          WRITE #9, C(I)
        NEXT I

        FOR I = 0 TO 10
          WRITE #9, Z(I)
        NEXT I
        CLOSE #9

      OPEN ACURDSK$ FOR OUTPUT AS #9
```

```
            SAY1$ = "PARALLEL FILE TO" + CURDSK$ +
"CAL.FACTOR/,TIME,VACUUM,TEMP"
        WRITE #9, SAY1$
        CLOSE #9

      RETURN

ErDsk:
   PRINT "CHECK DISK DRIVE A: AND DISK! PRESS ENTER WHEN CORRECTED"
   CLOSE #5
   GOTO DISKMGR

DISKUPDATE:

   OPEN CURDSK$ FOR APPEND AS #9
    MARKER2$ = "Converted BPM with Distribution"
     WRITE #9, MARKER2$, DLT, TIME$
     WRITE #9, CONVD, CONVE, CONVF, SIGMD, SIGME, SIGMF
      FOR I = 0 TO 4
       WRITE #9, FSD(I), FSE(I), FSF(I)
      NEXT I
    CLOSE #9

    OPEN ACURDSK$ FOR APPEND AS #9
     WRITE #9, TIME$, VAC$, btemp
    CLOSE #9

    RETURN


DISKCUPUPDATE:
   OPEN CURDSK$ FOR APPEND AS #9
    MARKER1$ = "BPM_CUP Readsings With Calibration"
     WRITE #9, MARKER1$, TIME$, DELTIM
      FOR I = 0 TO 2
       WRITE #9, AC(I), AB(I), CN4(I)
      NEXT I
      FOR I = 0 TO 4
       WRITE #9, VARD(I), VARE(I), VARF(I)
      NEXT I
     CLOSE #9

    OPEN ACURDSK$ FOR APPEND AS #9
     WRITE #9, CN4(0), CN4(1), CN4(2)
    CLOSE #9
```

```
    RETURN
PRINTSCREEN:

   CLS
   LOCATE 2, 2: COLOR 0, 7
    PRINT "Cup Current * NANOAMPS/CM^2 *     Acc. Dose * AS INDICATED *":
COLOR 2, 1: LOCATE 2, 64: PRINT "RUN- "; RUNNUM$: COLOR 0, 6
    COLOR 2, 1

      PRINT " ": LOCATE 3, 64: PRINT "DATE- "; DATE$

    PRINT " FOR 400 KV.          AVG.          "; ESOD$(Z(5)); "        AVG."
T$ = TIME$
Hr = VAL(T$)
IF Hr > 12 THEN Hr = Hr - 12
LOCATE 4, 64: PRINT "TIME - "; STR$(Hr); RIGHT$(T$, 6)
  IF LC1 < 1 THEN GOTO UNDER

    DOSTOGO = TGDOS - TOTLDOS
    ON Z(4) GOSUB DOSCHK400, DOSCHK2500, DOSCHK5000
     GOTO YONDER

DOSCHK400:
   IF CONVD = 0 THEN CONVD = 1E-20

   TIMETOGO = DOSTOGO / (CONVD / DLT)
   RETURN

DOSCHK2500:
   IF CONVE = 0 THEN CONVE = 1E-20
   TIMETOGO = DOSTOGO / (CONVE / DLT)
   RETURN

DOSCHK5000:
   IF CONVF = 0 THEN CONVF = 1E-20
   TIMETOGO = DOSTOGO / (CONVF / DLT)
   RETURN

YONDER:
   ETAHOURS = FIX(TIMETOGO / 3600): ETAMIN = FIX((TIMETOGO - ETAHOURS
* 3600) / 60): ETASEC = FIX(TIMETOGO - ETAHOURS * 3600 - ETAMIN * 60)
   RUNTIME$ = STR$(ETAHOURS) + ":" + STR$(ETAMIN) + ":" + STR$(ETASEC)
UNDER:
    ON Z(5) GOSUB X400, Y400, Z400
     GOTO R400
X400:
```

```
    ' SAME
    FOR I = 0 TO 4
      REPORTD(I) = FSD(I) / 1E+15
    NEXT I
      REPORTD(5) = SIGMD / 1E+15
    RETURN
Y400:
    ' CONVERT IONS/CM^2 TO DPA
    FOR I = 0 TO 4
      REPORTD(I) = FSD(I) / (6.225E+12 * C(13))
    NEXT I
      REPORTD(5) = SIGMD / (6.225E+12 * C(13))
    RETURN
Z400:
    ' CONVERT IONS/CM^2 TO APPM
    FOR I = 0 TO 4
      REPORTD(I) = CINT(FSD(I) * 4E-13 / (C(6) * C(7)))
    NEXT I
      REPORTD(5) = CINT(SIGMD * 4E-13 / (C(6) * C(7)))
   RETURN
R400:

    CALL MAKEWIND(6, 3, 8, 30, 2, 7, 6, 1, 0, lab$)
    CALL MAKEWIND(6, 33, 8, 59, 2, 7, 6, 1, 0, lab$)
      LOCATE 5, 64: PRINT "VACUUM -"; VAC$
       COLOR 0, 6
   LOCATE 6, 3: PRINT USING "#####."; CMD(0); SPC(2); CMD(1): LOCATE 6, 33:
PRINT USING "#####.##"; REPORTD(0); SPC(2); REPORTD(1): LOCATE 6, 61:
COLOR 1, 0: PRINT " ":  COLOR 2, 1
   LOCATE 6, 64: PRINT "Req. Temp."; TEMP; : COLOR 0, 6
   LOCATE 7, 3: PRINT USING "#####."; SPC(5); CMD(2); SPC(6); CMAC(0): LOCATE
7, 38: PRINT USING "#####.##"; REPORTD(2); SPC(6); REPORTD(5): LOCATE 7, 61:
COLOR 1, 0: PRINT " ": COLOR 2, 1
   LOCATE 7, 64: PRINT "Station - "; STATION; : COLOR 0, 6
   LOCATE 8, 3: PRINT USING "#####."; CMD(3); SPC(2); CMD(4): LOCATE 8, 33:
PRINT USING "#####.##"; REPORTD(3); SPC(2); REPORTD(4): LOCATE 8, 61:
COLOR 1, 0: PRINT " ": LOCATE 9, 61: PRINT " ": COLOR 2, 1
     LOCATE 8, 64: PRINT "Total Dose";
     LOCATE 9, 65: PRINT FINALDOSE; ESOD$(ING);
     LOCATE 10, 4: COLOR 1, 0: PRINT SPACE$(58);
     COLOR 2, 1: LOCATE 10, 64: PRINT "By The "; DOSE$(Z(4));
     COLOR 7, 0: LOCATE 10, 7: PRINT "Std. Dev. = "; : PRINT USING "###.#";
(DSTAT(3) / (DSTAT(2) + 1E-20)) * 100; : LOCATE 10, 30: PRINT "CAL. BPM nA/CM^2
="; : PRINT USING "####.#"; SIMCURD
     LOCATE 11, 2: COLOR 2, 1
    PRINT "FOR 2.5 MV.                          "; ESOD$(Z(6)):  COLOR 2, 1: LOCATE 11,
```

```
63: PRINT RUNTIME$; " TO GO"
    PRINT " "
    COLOR 0, 6

    ON Z(6) GOSUB X2500, Y2500, Z2500
      GOTO R2500
X2500:
    ' SAME
    FOR I = 0 TO 4
      REPORTE(I) = FSE(I) / 1E+15
    NEXT I
      REPORTE(5) = SIGME / 1E+15
    RETURN
Y2500:
    ' CONVERT IONS/CM^2 TO DPA
    FOR I = 0 TO 4
      REPORTE(I) = FSE(I) / (6.225E+12 * C(12))
    NEXT I
      REPORTE(5) = SIGME / (6.225E+12 * C(12))
    RETURN
Z2500:
    ' CONVERT IONS/CM^2 TO APPM
    FOR I = 0 TO 4
      REPORTE(I) = CINT(FSE(I) * 4E-13 / (C(6) * C(4)))
    NEXT I
      REPORTE(5) = CINT(SIGME * 4E-13 / (C(6) * C(4)))
    RETURN
R2500:
    btemp$ = LEFT$(STR$(btemp), 6)
     CALL MAKEWIND(13, 3, 15, 30, 2, 7, 6, 1, 0, lab$)
     CALL MAKEWIND(13, 33, 15, 59, 2, 7, 6, 1, 0, lab$)
    LOCATE 13, 3: PRINT USING "#####."; CME(0); SPC(2); CME(1): LOCATE 13, 33:
PRINT USING "#####.##"; REPORTE(0); SPC(2); REPORTE(1): LOCATE 13, 61:
COLOR 1, 0: PRINT " ": LOCATE 13, 63: COLOR 2, 1: PRINT FIX(SIGMDLT); "TIME
SEC.": COLOR 0, 6
    LOCATE 14, 3: PRINT USING "#####."; SPC(5); CME(2); SPC(6); CMAC(1): LOCATE
14, 38: PRINT USING "#####.##"; REPORTE(2); SPC(6); REPORTE(5): LOCATE 14, 61:
COLOR 1, 0: PRINT " ": LOCATE 14, 64: COLOR 2, 1: PRINT "Act. Temp"; btemp$: COLOR
0, 6
    LOCATE 15, 3: PRINT USING "#####."; CME(3); SPC(2); CME(4): LOCATE 15, 33:
PRINT USING "#####.##"; REPORTE(3); SPC(2); REPORTE(4): LOCATE 15, 61:
COLOR 1, 0: PRINT " ": LOCATE 16, 61: PRINT " ": COLOR 0, 6
     LOCATE 17, 4: COLOR 1, 0: PRINT SPACE$(58);
     COLOR 7, 0: LOCATE 17, 7: PRINT "Std. Dev. = "; : PRINT USING "###.#";
(ESTAT(3) / (ESTAT(2) + 1E-20)) * 100; : LOCATE 17, 30: PRINT "CAL. BPM nA/CM^2
="; : PRINT USING "####.#"; SIMCURE
```

```
    LOCATE 18, 2
    COLOR 2, 1
   PRINT "FOR 5   MV.                          "; ESOD$(Z(7))
   PRINT " "
     COLOR 0, 6

   ON Z(7) GOSUB X5, Y5, Z5
     GOTO R5
X5:
   ' SAME
   FOR I = 0 TO 4
     REPORTF(I) = FSF(I) / 1E+15
   NEXT I
     REPORTF(5) = SIGMF / 1E+15
   RETURN
Y5:
   ' CONVERT IONS/CM^2 TO DPA
   FOR I = 0 TO 4
     REPORTF(I) = FSF(I) / (6.225E+12 * C(11))
   NEXT I
     REPORTF(5) = SIGMF / (6.225E+12 * C(11))
   RETURN
Z5:
   ' CONVERT IONS/CM^2 TO APPM
   FOR I = 0 TO 4
     REPORTF(I) = CINT(FSF(I) * 4E-13 / (C(6) * C(16)))
   NEXT I
     REPORTF(5) = CINT(SIGMF * 4E-13 / (C(6) * C(16)))
   RETURN

R5:
     CALL MAKEWIND(20, 3, 22, 30, 2, 7, 6, 1, 0, lab$)
     CALL MAKEWIND(20, 33, 22, 59, 2, 7, 6, 1, 0, lab$)
   LOCATE 20, 3: PRINT USING "#####."; CMF(0); SPC(2); CMF(1): LOCATE 20, 33:
PRINT USING "#####.##"; REPORTF(0); SPC(2); REPORTF(1): LOCATE 20, 61:
COLOR 1, 0: PRINT " ": COLOR 0, 6
   LOCATE 21, 3: PRINT USING "#####."; SPC(5); CMF(2); SPC(6); CMAC(2): LOCATE
21, 38: PRINT USING "#####.##"; REPORTF(2); SPC(6); REPORTF(5): LOCATE 21, 61:
COLOR 1, 0: PRINT " ": COLOR 0, 6
   LOCATE 22, 3: PRINT USING "#####."; CMF(3); SPC(2); CMF(4); : LOCATE 22, 33:
PRINT USING "#####.##"; REPORTF(3); SPC(2); REPORTF(4): LOCATE 22, 61:
COLOR 1, 0: PRINT " ": LOCATE 23, 61: PRINT " ": COLOR 0, 6
     LOCATE 24, 4: COLOR 1, 0: PRINT SPACE$(58);
    COLOR 7, 0: LOCATE 24, 7: PRINT "Std. Dev. = "; : PRINT USING "###.#";
(FSTAT(3) / (FSTAT(2) + 1E-20)) * 100; : LOCATE 24, 30: PRINT "CAL. BPM nA/CM^2
="; : PRINT USING "####.#"; SIMCURF;
```

```
    IF ENTC = 4 THEN
        CLOSE #1
        CLOSE #2
        CLOSE #3
      GOTO FINALPRINT
    END IF
      I = 4

CLOCKY:
      IF PONS > 1 THEN
      PONS = 1
      GOTO PRINTSCREEN
      ELSE
      LOCATE 25, I: COLOR 0, 1: PRINT CHR$(4); : SLEEP 2: I = I + 1
      END IF
      IF I = 5 AND ENTC = 1 AND PRNT = 2 THEN CALL INTERRUPT(&H5, InRegs,
OutRegs)
          IF PRNT = 1 THEN
          GOSUB PRNTPARAM
          PRNT = 2
          END IF
            IF LC1 = 9 THEN
              OPEN "O", 17, "HV"
              PRINT #17, "CUPS  WILL  GO  IN  NEXT   MINANT"
              CLOSE #17
              END IF
      GOTO CLOCKY

     COLOR 0, 7
RUNDWNLOD:
OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
OPEN "\DEV\IEEEIN" FOR INPUT AS #2
DIM A$(94), G(5)
PRINT #1, "FILL ERROR"
PRINT #1, "CONFIG /I"
PRINT #1, "TIME OUT 70"
CLS
   A$ = "SET TIME = 0"
   CALL OUTDO(A$)
   A$ = "CLROUT"
   CALL OUTDO(A$)

A$ = "RST"
CALL OUTDO(A$)

A$(0) = "REAL FSD(4), FSE(4), FSF(4)"
```

```
A$(1) = "REAL BD,BE,BF,BPMD,BPME,BPMF,SCUP(3),TOTLDOS,DLT,TGDOS,C(27),Z(11)"
A$(2) = "INTEGER I,J,L,N,P,M,R,S,T,ENTC,LC1,LC2,BY"
A$(3) = "REAL D(4),DI(4),E(4),EI(4),F(4),FI(4),CN4(2),DSTAT(3),ESTAT(3),FSTAT(3)"
A$(4) = "REAL A, AC(2), AB(2),AD(2), AE(2), VARD(4), VARE(4), VARF(4), TDTA,
TDTB, TDTC, TR(30), DELTIM"
A$(5) = "TDTB=0; LC2=0; SET TIME 0; AD(0)=0; AD(1)=0; AD(2)=0; AE(0)=0; AE(1)=0;
AE(2)=0;"
A$(6) = "SUB CUP_BPM"
A$(7) = "CLOSE 1206; TDTC=TDTB; BY=Z(4)"
A$(8) = "WAIT .7; TIME INTO TDTA; WAIT 1"
A$(9) = "USE 1400; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(10) = "USE 1401; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(11) = "USE 1500; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(12) = "USE 1501; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(13) = "USE 1600; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(14) = "USE 1601; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(15) = "FOR T = 0 TO 2; AC(T) = 0; AB(T) = 0; NEXT T"
A$(16) = "FOR I = 0 TO 4"
A$(17) = "M = 1000"
A$(18) = "M=M+I;N=M+5;P=M+10"
A$(19) = "CLOSE M,N,P"
A$(20) = "WAIT 1"
A$(21) = "FOR L = 1400 TO 1401"
A$(22) = "R = L + 100; S = L + 200"
A$(23) = "TRIG SGL USE R"
A$(24) = "TRIG SGL USE L"
A$(25) = "TRIG SGL USE S"
A$(26) = "NEXT L"
A$(27) = "CLOSE 1207"
A$(28) = "WAIT 1.995"
A$(29) = "OPEN 1207"
A$(30) = "CHREADZ 1400 INTO D(I); CHREADZ 1401 INTO DI(I); CHREADZ 1500
INTO E(I)"
A$(31) = "CHREADZ 1501 INTO EI(I); CHREADZ 1600 INTO F(I); CHREADZ 1601 INTO
FI(I)"
A$(32) = "AC(0)=AC(0)+D(I); AC(1)=AC(1)+E(I); AC(2)=AC(2)+F(I)"
A$(33) = "AB(0)=AB(0)+DI(I); AB(1)=AB(1)+EI(I); AB(2)=AB(2)+FI(I)"
A$(34) = "OPEN M,N,P"
A$(35) = "NEXT I"
A$(36) = "OPEN 1206; WAIT .6; TIME INTO TDTB"
A$(37) = "STAT DSTAT,DSTAT,DSTAT,DSTAT,D"  ' STATISTICS ON CUP
A$(38) = "STAT ESTAT,ESTAT,ESTAT,ESTAT,E"
A$(39) = "STAT FSTAT,FSTAT,FSTAT,FSTAT,F"
A$(40) = "FOR T = 0 TO 2;IF AC(T) = 0 THEN; AC(T) = 1.0E-20; END IF;
AC(T)=AC(T)/5; AE(T)=AE(T)+AC(T)"  ' AVERAGE COUNTS FOR CUPS  AND
```

```
ACCUMULATION
A$(41) = "AB(T)=AB(T)/5; AD(T)=AD(T)+AB(T)"                    ' AVERAGE COUNTS
FOR BPM  AND ACCUMULATION
A$(42) = "IF AD(T) = 0 THEN; AD(T) = 1.0E-20; END IF; CN4(T)=AE(T)/AD(T); NEXT
T"                    ' CONSTANT CONVERTS BPM TO CUP CAL.
A$(43) = "FOR I = 0 TO 4"
A$(44) = "VARD(I)=D(I)/AC(0)"            ' VARIANCE OF INDIVIDUAL CUPS 400
A$(45) = "VARE(I)=E(I)/AC(1)"            ' VARIANCE OF INDIVIDUAL CUPS 2500
A$(46) = "VARF(I)=F(I)/AC(2)"            ' VARIANCE OF INDIVIDUAL CUPS 5000
A$(47) = "NEXT I"
' READ VACUUM AND TEMP
A$(48) = "DELTIM = TDTA - TDTC; TR(0)=TDTB; LC1=1; ENTC=1; LC2=LC2+1; SRQ;
SUBEND"


A$(49) = "SUB BPM_READ"
A$(50) = "USE 1401; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(51) = "USE 1501; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(52) = "USE 1601; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(53) = "REAL FSD(4), FSE(4), FSF(4)"
A$(54) = "REAL BD,BE,BF,BPMD,BPME,BPMF,SCUPD,SCUPE,SCUPF"
A$(55) = "INTEGER I"
A$(56) = "BD=0; BE=0; BF=0"
A$(57) = "FOR I = 0 TO 29"                ' ***********************************
A$(58) = "TRIG SGL USE 1401"                ' * READS COUNTS FOR 2 SEC., ACCUM. *
A$(59) = "TRIG SGL USE 1501"                ' * COUNTS FOR 1  MIN., CONVERTS    *
A$(60) = "TRIG SGL USE 1601"                ' * BPM COUNTS TO CUP COUNTS AND
*
A$(61) = "CLOSE 1207"                     ' * CALCULATES DISTRIBUTION.        *
A$(62) = "WAIT 1.995"                     ' ***********************************

A$(63) = "OPEN 1207"
A$(64) = "CHREADZ 1401 INTO BPMD; CHREADZ 1501 INTO BPME; CHREADZ 1601
INTO BPMF"
A$(65) = "BD=BD+BPMD; BE=BE+BPME; BF=BF+BPMF"
A$(66) = "NEXT I"
A$(67) = "BPMD=BD/60; BPME=BE/60; BPMF=BF/60; VREAD VARD; VREAD VARE;
VREAD VARF"
' Dividing by 60 instead of 30 eliminates dividing by 2 later (interval of time over which counts
were taken)
A$(68) = "VREAD CN4; SCUP(1)=BPMD*CN4(0); SCUP(2)=BPME*CN4(1);
SCUP(3)=BPMF*CN4(2)"
A$(69) = "FOR I = 0 TO 4"
A$(70) = "FSD(I)=SCUP(1)*VARD(I); FSE(I)=SCUP(2)*VARE(I);
FSF(I)=SCUP(3)*VARF(I)"
A$(71) = "NEXT I"
```

```
A$(72) = "TIME INTO TR(LC1); ENTC=2; SUBEND"
     A$(73) = "SUB THREE"
     A$(74) = "CALL BPM_READ"
     A$(75) = "DLT= (TR(LC1)-TR(LC1 - 1))"
     A$(76) = "TOTLDOS= TOTLDOS + SCUP(BY) * 1.55705E+10 * DLT / C(1)"
     A$(77) = "IF TOTLDOS >= TGDOS THEN"
     A$(78) = "ENTC=4"
     A$(79) = "SRQ"
     A$(80) = "PAUSE"
     A$(81) = "ELSE"
     A$(82) = "LC1=LC1+1"
     A$(83) = "END IF"
     A$(84) = "ENTC=2"
     A$(85) = "IF LC1 >= 11 THEN"
     A$(86) = "ENTC=3"
     A$(87) = "END IF"

     A$(88) = "SRQ"
     A$(89) = "SUBEND"
     A$(90) = "SUB START"
     A$(91) = "CALL CUP_BPM"
     A$(92) = "ENTC=1"
     A$(93) = "SRQ"
     A$(94) = "SUBEND"
FOR I = 0 TO 94
A$ = A$(I)
CALL OUTDO(A$)
NEXT I
CHECKRUN = 1
RETURN

PRNTPARAM:

    LPRINT "THIS IS  "
    LPRINT "   RUN #     ", RUNNUM$; "   USER ID.  "; USER$
    LPRINT "   THE TIME IS ", TIMEXP$; "   DATE      ", DT$
    LPRINT "   STATION # ", STATION; "   TEMP. (C) ", TEMP
    LPRINT "   USING";
    IF Z(1) = 1 THEN LPRINT " PN400,";  ' IF Z(1) = 1 THE 400KV MACHINE WILL BE
USED
    IF Z(2) = 1 THEN LPRINT " AN2500,";  ' IF Z(2) = 1 THE AN2500 MACHINE WILL
BE USED
    IF Z(3) = 1 THEN LPRINT " CN5.0 ";  ' IF Z(3) = 1 THE AN5000 MACHINE WILL
BE USED
    LPRINT " TERMINATION CONTROL BY"; DOSE$(Z(4)); : LPRINT " IS"; : LPRINT
FINALDOSE; ESOD$(ING);
```

```
    LPRINT "   ATOMIC DENSITY (x 10^-22) OF TARGET "; C(6)
   IF Z(1) = 1 THEN
     LPRINT "   400 KV MACHINE VOLTAGE (KV) IS "; C(17)
     LPRINT "   400 RAMP INTERVAL RANGE IS "; C(18)
     LPRINT "   400 PROJECTILE ION ATOMIC MASS NUMBER IS "; C(19)
     LPRINT "   400 DOSE INDICATED IN "; ESOD$(Z(5))
     LPRINT "   400 DEVIATION OF DISTRIBUTION OF IMPLANTED ION IS "; C(7)
     LPRINT "   400 MICROCOULOMB/DPA IS "; C(13)
     LPRINT "   400 PROJECTILE ION CHARGE STATE IS "; C(24)
  ELSE
 END IF
  IF Z(2) = 1 THEN
      LPRINT "   2500 KV MACHINE VOLTAGE (KV) IS "; C(20)
      LPRINT "   2500 PROJECTILE ION ATOMIC MASS NUMBER IS "; C(21)
      LPRINT "   2500 DOSE INDICATED IN "; ESOD$(Z(6))
      LPRINT "   2500 DEVIATION OF DISTRIBUTION OF IMPLANTED ION IS "; C(4)
      LPRINT "   2500 MICROCOULOMB/DPA IS "; C(12)
      LPRINT "   2500 PROJECTILE ION CHARGE STATE IS "; C(25)

    ELSE
   END IF
   IF Z(3) = 1 THEN
        LPRINT "   5 MV MACHINE VOLTAGE (MV) IS "; C(22)
        LPRINT "   5 MV PROJECTILE ION ATOMIC MASS NUMBER IS "; C(23)
        LPRINT "   5 MV DOSE INDICATED IN "; ESOD$(Z(7))
        LPRINT "   5 MV DEVIATION OF DISTRIBUTION OF IMPLANTED ION IS ";
C(16)
        LPRINT "   5 MV MICROCOULOMB/DPA IS "; C(11)
        LPRINT "   5 MV PROJECTILE ION CHARGE STATE IS "; C(26)

    ELSE
  END IF

  RETURN



FINALPRINT:
    LPRINT CHR$(12)
  ' LINEPRINT DATA ASSOCIATED WITH THIS RUN ON PRINTER

         CALL INTERRUPT(&H5, InRegs, OutRegs)


    LPRINT "THIS IS  "
```

```
    LPRINT "   RUN #     ", RUNNUM$; "   USER ID.  "; USER$
    LPRINT "   THE TIME IS ", TIMEXP$; "   DATE      ", DT$
    LPRINT "   STATION # ", STATION; "   TEMP. (C) ", TEMP
    LPRINT "   USING";
    IF Z(1) = 1 THEN LPRINT " PN400,";  ' IF Z(1) = 1 THE 400KV MACHINE WILL BE
USED
    IF Z(2) = 1 THEN LPRINT " AN2500,";  ' IF Z(2) = 1 THE AN2500 MACHINE WILL
BE USED
    IF Z(3) = 1 THEN LPRINT " CN5.0 ";  ' IF Z(3) = 1 THE AN5000 MACHINE WILL
BE USED
    LPRINT " TERMINATION CONTROL BY"; DOSE$(Z(4)); : LPRINT " IS"; : LPRINT
FINALDOSE; ESOD$(ING);

    LPRINT "   ATOMIC DENSITY (x 10^-22) OF TARGET "; C(6)
   IF Z(1) = 1 THEN
     LPRINT "   400 KV MACHINE VOLTAGE (KV) IS "; C(17)
     LPRINT "   400 RAMP INTERVAL RANGE IS "; C(18)
     LPRINT "   400 PROJECTILE ION ATOMIC MASS NUMBER IS "; C(19)
     LPRINT "   400 DOSE INDICATED IN "; ESOD$(Z(5))
     LPRINT "   400 DEVIATION OF DISTRIBUTION OF IMPLANTED ION IS "; C(7)
     LPRINT "   400 MICROCOULOMB/DPA IS "; C(13)
     LPRINT "   400 PROJECTILE ION CHARGE STATE IS "; C(24)
     LPRINT " TOTAL AVERAGE IN TERMS OF"
     LPRINT "   IONS/CM^2 * 10^15 = ", SIGMD / 1E+15
     LPRINT "   DPA              = ", SIGMD / (6.225E+12 * C(13))
     LPRINT "   APPM              = ", CINT(SIGMD * 4E-13 / (C(6) * C(7)))


  ELSE
 END IF
  IF Z(2) = 1 THEN
      LPRINT "   2500 KV MACHINE VOLTAGE (KV) IS "; C(20)
      LPRINT "   2500 PROJECTILE ION ATOMIC MASS NUMBER IS "; C(21)
      LPRINT "   2500 DOSE INDICATED IN "; ESOD$(Z(6))
      LPRINT "   2500 DEVIATION OF DISTRIBUTION OF IMPLANTED ION IS "; C(4)
      LPRINT "   2500 MICROCOULOMB/DPA IS "; C(12)
      LPRINT "   2500 PROJECTILE ION CHARGE STATE IS "; C(25)
      LPRINT " TOTAL AVERAGE IN TERMS OF"
      LPRINT "   IONS/CM^2 * 10^15 = ", SIGME / 1E+15
      LPRINT "   DPA              = ", SIGME / (6.225E+12 * C(12))
      LPRINT "   APPM              = ", CINT(SIGME * 4E-13 / (C(6) * C(4)))

     ELSE
    END IF
   IF Z(3) = 1 THEN
        LPRINT "   5 MV MACHINE VOLTAGE (MV) IS "; C(22)
```

```
        LPRINT "   5 MV PROJECTILE ION ATOMIC MASS NUMBER IS "; C(23)
        LPRINT "   5 MV DOSE INDICATED IN "; ESOD$(Z(7))
        LPRINT "   5 MV DEVIATION OF DISTRIBUTION OF IMPLANTED ION IS ";
C(16)
        LPRINT "   5 MV MICROCOULOMB/DPA IS "; C(11)
        LPRINT "   5 MV PROJECTILE ION CHARGE STATE IS "; C(26)
        LPRINT " TOTAL AVERAGE IN TERMS OF"
        LPRINT "   IONS/CM^2 * 10^15 = ", SIGMF / 1E+15
        LPRINT "   DPA              = ", SIGMF / (6.225E+12 * C(11))
        LPRINT "   APPM              = ", CINT(SIGMF * 4E-13 / (C(6) * C(16)))

    ELSE
  END IF



      LPRINT CHR$(12)

       CALL INTERRUPT(&H5, InRegs, OutRegs)


    LPRINT "THIS IS  "
    LPRINT "   RUN #     ", RUNNUM$; "   USER ID.  "; USER$
    LPRINT "   THE TIME IS ", TIMEXP$; "   DATE      ", DT$
    LPRINT "   STATION # ", STATION; "   TEMP. (C) ", TEMP
    LPRINT "   USING";
    IF Z(1) = 1 THEN LPRINT " PN400,";  ' IF Z(1) = 1 THE 400KV MACHINE WILL BE
USED
    IF Z(2) = 1 THEN LPRINT " AN2500,";  ' IF Z(2) = 1 THE AN2500 MACHINE WILL
BE USED
    IF Z(3) = 1 THEN LPRINT " CN5.0 ";  ' IF Z(3) = 1 THE AN5000 MACHINE WILL
BE USED
    LPRINT " TERMINATION CONTROL BY"; DOSE$(Z(4)); : LPRINT " IS"; : LPRINT
FINALDOSE; ESOD$(ING);

    LPRINT "   ATOMIC DENSITY (x 10^-22) OF TARGET "; C(6)
   IF Z(1) = 1 THEN
     LPRINT "   400 KV MACHINE VOLTAGE (KV) IS "; C(17)
     LPRINT "   400 RAMP INTERVAL RANGE IS "; C(18)
     LPRINT "   400 PROJECTILE ION ATOMIC MASS NUMBER IS "; C(19)
     LPRINT "   400 DOSE INDICATED IN "; ESOD$(Z(5))
     LPRINT "   400 DEVIATION OF DISTRIBUTION OF IMPLANTED ION IS "; C(7)
     LPRINT "   400 MICROCOULOMB/DPA IS "; C(13)
     LPRINT "   400 PROJECTILE ION CHARGE STATE IS "; C(24)
     LPRINT " TOTAL AVERAGE IN TERMS OF"
```

```
    LPRINT "   IONS/CM^2 * 10^15 = ", SIGMD / 1E+15
    LPRINT "   DPA              = ", SIGMD / (6.225E+12 * C(13))
    LPRINT "   APPM              = ", CINT(SIGMD * 4E-13 / (C(6) * C(7)))


  ELSE

 END IF
  IF Z(2) = 1 THEN
      LPRINT "   2500 KV MACHINE VOLTAGE (KV) IS "; C(20)
      LPRINT "   2500 PROJECTILE ION ATOMIC MASS NUMBER IS "; C(21)
      LPRINT "   2500 DOSE INDICATED IN "; ESOD$(Z(6))
      LPRINT "   2500 DEVIATION OF DISTRIBUTION OF IMPLANTED ION IS "; C(4)
      LPRINT "   2500 MICROCOULOMB/DPA IS "; C(12)
      LPRINT "   2500 PROJECTILE ION CHARGE STATE IS "; C(25)
      LPRINT " TOTAL AVERAGE IN TERMS OF"
      LPRINT "   IONS/CM^2 * 10^15 = ", SIGME / 1E+15
      LPRINT "   DPA              = ", SIGME / (6.225E+12 * C(12))
      LPRINT "   APPM              = ", CINT(SIGME * 4E-13 / (C(6) * C(4)))

    ELSE
   END IF
   IF Z(3) = 1 THEN
        LPRINT "   5 MV MACHINE VOLTAGE (MV) IS "; C(22)
        LPRINT "   5 MV PROJECTILE ION ATOMIC MASS NUMBER IS "; C(23)
        LPRINT "   5 MV DOSE INDICATED IN "; ESOD$(Z(7))
        LPRINT "   5 MV DEVIATION OF DISTRIBUTION OF IMPLANTED ION IS ";
C(16)
        LPRINT "   5 MV MICROCOULOMB/DPA IS "; C(11)
        LPRINT "   5 MV PROJECTILE ION CHARGE STATE IS "; C(26)
        LPRINT " TOTAL AVERAGE IN TERMS OF"
        LPRINT "   IONS/CM^2 * 10^15 = ", SIGMF / 1E+15
        LPRINT "   DPA              = ", SIGMF / (6.225E+12 * C(11))
        LPRINT "   APPM              = ", CINT(SIGMF * 4E-13 / (C(6) * C(16)))


    ELSE
   END IF


       LPRINT CHR$(12)


BYPASS:
     GOSUB DOLITTLE
     GOSUB RETCRT
```

```
      Q = 0: L = 0: DRT = 0
      RETURN

RESTART:
      COLOR 15, 1
      GOSUB RETCRT
      CLS
      PRINT "If You Have  NOT  HARD RESET THE COMPUTER AND THE MAIN
CONTROLLER  Do so Now"
      INPUT "Press Enter If HAVE RESET", PQ$
      CLS
      INPUT "DATA DISK SHOULD BE IN DRIVE A: (PRESS ENTER TO CONTINUE)",
PQ$
      CLS
      PRINT "DO YOU WANT TO RESTART LAST RUN OR DO YOU WANT TO
AUGMENT DOSE OF AN OLD RUN"
      INPUT "L = LAST; O = OLD ", PQ$
      IF UCASE$(PQ$) = "O" THEN
      INPUT "INPUT OLD FILE NUMBER (example 12890030 ) ", FileNo$
      ELSE
      OPEN "A:\INDEX" FOR INPUT AS #14
      INPUT #14, FileNo$
      CLOSE #14
      END IF
      FileNo$ = RIGHT$(FileNo$, 8)
      FileNo$ = "A:\" + FileNo$
    OPEN "A:\PARAM" FOR INPUT AS #14
    FOR I = 1 TO 26
    INPUT #14, C(I)
    NEXT I
    FOR I = 1 TO 10
    INPUT #14, Z(I)
    NEXT I
    INPUT #14, RUNNUM$, STATION, TEMP, USER$, FINALDOSE, ESOD$(1),
ESOD$(2), ESOD$(3), TGDOS, SOD$(1), SOD$(2), SOD$(3), DOSE$(Z(4))
    CLOSE #14
      OPEN FileNo$ FOR INPUT AS #14
  FOR I = 1 TO 7
    INPUT #14, AIN$(I)
   NEXT I
     RUNNUM$ = "RERUN" + AIN$(3)
     USER$ = AIN$(4)
     STATION = VAL(AIN$(5))
     TEMP = VAL(AIN$(6))
  FOR J = 0 TO 26
    INPUT #14, C(J)
```

```
  NEXT J

  FOR I = 0 TO 10
   INPUT #14, Z(I)
   NEXT I


  CLS
  LC1 = 0

  J = 1
  K = 1

DO WHILE NOT EOF(14)


    INPUT #14, TMARKER1$, TSTIME$, DELTIM
      FOR I = 0 TO 2
       INPUT #14, AC(I), AB(I), CN4(I)
        NEXT I
      FOR L = 0 TO 4
        INPUT #14, VARD(L), VARE(L), VARF(L)
          NEXT L


     LC1 = 0
     DO WHILE NOT EOF(14) OR LC1 < 11
      LC1 = LC1 + 1
      INPUT #14, TMARKER2$, DLT, TTIME$
      SIGMDLT = SIGMDLT + DLT
        INPUT #14, CONVD, CONVE, CONVF, SIGMD, SIGME, SIGMF
        LC1 = LC1 + 1
        FOR I = 0 TO 4
          INPUT #14, FSD(I), FSE(I), FSF(I)
            NEXT I

        LOOP
        LOOP


    CLOSE #14

    ON Z(4) GOSUB 100, 200, 300
      GOTO 400
100 :
    TOTLDOS = SIGMD
```

```
     RETURN
200 :
   TOTLDOS = SIGME
     RETURN
300 :
   TOTLDOS = SIGMF
     RETURN
400 :


    ENTC = 1
    PRNT = 1
    GOSUB DISKMGR
    COLOR 15, 1
    INPUT " DO YOU WANT TO CHANGE TARGET DOSE ( Y = YES; N = NO ) ",
PDQ$
    IF UCASE$(PDQ$) = "N" THEN GOTO PST2
    PRINT " WHAT VALUE FOR TARGET DOSE WOULD YOU LIKE ( MUST BE
GREATER THAN PREVIOUS )"
    INPUT " (AND MUST BE IN THE SAME UNITS AS THAT OF THE ORIGINAL
RUN)", FINALDOSE


    ON Z(4) GOSUB A10, A20, A30
      GOTO PST2
A10:
   ON Z(5) GOSUB A11, A21, A31
   RETURN
A11:
   TGDOS = FINALDOSE
   RETURN
A21:
   ' Convert To Ions/Cm^2 From DPA 400
   TGDOS = FINALDOSE * C(13) * 6.225E+12
   RETURN
A31:
   ' Convert To Ions/Cm^2 From APPM 400
   TGDOS = FINALDOSE * C(6) * C(7) * 2.5E+12
  RETURN
A20:
   ON Z(6) GOSUB A12, A22, A32
   RETURN
A12:
   TGDOS = FINALDOSE
   RETURN
A22:
```

```
    ' Convert To Ions/Cm^2 From DPA 2500
    TGDOS = FINALDOSE * C(12) * 6.225E+12
    RETURN
A32:
    ' Convert To Ions/Cm^2 From APPM 2500
    TGDOS = FINALDOSE * C(6) * C(4) * 2.5E+12
    RETURN
A30:
    ON Z(7) GOSUB A13, A23, A33
    RETURN
A13:
    TGDOS = FINALDOSE
    RETURN
A23:
    ' Convert To Ions/Cm^2 From DPA 5 MV
    TGDOS = FINALDOSE * C(11) * 6.225E+12
    RETURN
A33:
    ' Convert To Ions/Cm^2 From APPM 5 MV
    TGDOS = FINALDOSE * C(6) * C(16) * 2.5E+12
    RETURN

PST2:
        PONS = 1
        ENTC = 1
        PRNT = 1
    GOSUB RUNDWNLOD
    GOSUB PARMPAS

    GOTO RUNLOOP

      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0: L = 0: DRT = 0
      RETURN

CHGPARM:
      COLOR 15, 1
      GOSUB RETCRT
      PRINT " TO CHANGE THE PARAMETERS OF THE IMPENDING RUN GO TO
SETUP"
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0: L = 0: DRT = 0
      RETURN
```

```
LSTCONT:
      COLOR 15, 1
      GOSUB RETCRT
      PRINT " THIS SECTION WILL LIST THE CONTROLS TO BE USED IN THIS RUN"
      PRINT " CHANGES CAN BE MADE USING THE MENU SETUP                  "
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0: L = 0: DRT = 0
      RETURN

MNU5:
     IF ROW < 8 THEN
     GOSUB RETCRT
     Q = 0: L = 0: DRT = 0
     RETURN
       ELSEIF COL < 41 OR COL > 49 OR ROW > 20 THEN
       RETURN
         ELSE
         ON INT(((ROW * 3) + 7) / 10) - 2 GOTO SETUP, HEAT, PYRO, SHUT
           END IF
           RETURN

SETUP:

      COLOR 15, 1
      GOSUB RETCRT
      PRINT "THIS SECTION WILL BE USED TO INITIALIZE AND CONTROL
TEMPERATURE"
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0: L = 0: DRT = 0
      RETURN

HEAT:
      COLOR 15, 1
      GOSUB RETCRT
      PRINT " THIS SECTION WILL SET THE PARAMETERS FOR SPECIMEN HEAT
UP"
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0: L = 0: DRT = 0
      RETURN

PYRO:
      COLOR 15, 1
      GOSUB RETCRT
```

```
        PRINT " THIS CONTROL PYROMETER"
        GOSUB DOLITTLE
        GOSUB RETCRT
        Q = 0: L = 0: DRT = 0
        RETURN

SHUT:
        COLOR 15, 1
        GOSUB RETCRT
        PRINT " THIS SECTION WILL ESTABLISH SHUT OFF CONTROL ITEMS"
        GOSUB DOLITTLE
        GOSUB RETCRT
        Q = 0: L = 0: DRT = 0
        RETURN

       RETURN

MNU6:
       IF ROW < 8 THEN
       GOSUB RETCRT
       Q = 0: L = 0: DRT = 0
       RETURN
         ELSEIF COL < 51 OR COL > 59 OR ROW > 20 THEN
         RETURN
           ELSE
           ON INT(((ROW * 3) + 7) / 10) - 2 GOTO PYROMETER, VACUUM, INSTRU2,
SUP4
             END IF
             RETURN

PYROMETER:

        COLOR 15, 1
        GOSUB RETCRT
        PRINT "THIS SECTION WILL BE USED TO READ PYROMETER"
        GOSUB DOLITTLE
        GOSUB RETCRT
        Q = 0: L = 0: DRT = 0
        RETURN

VACUUM:
        COLOR 15, 1
        GOSUB RETCRT
        OPEN "COM1:300,N,7,2,RS" FOR RANDOM AS #1
        TURD$ = "DS IG"
        PRINT #1, TURD$ + CHR$(13) + CHR$(10)
```

```
      SLEEP 1
      INPUT #1, VAC$
      PRINT " THE VACUUM IS = "; VAC$
      CLOSE #1
      OPEN "O", 1, "HV"
      IF KY = 1 THEN SQK$ = "FOR THE LOADING CHAMBER" ELSE SQK$ = "FOR
THE MAIN TARGIT CHAMBER"
      PRINT #1, "%THE VACCQUM IS", VAC$, SQK$
      CLOSE #1
      GOSUB DOLITTLE
      GOSUB RETCRT
      Q = 0: L = 0: DRT = 0
      RETURN

INSTRU2:
      COLOR 15, 1
      GOSUB RETCRT

MONDEEPCUP:

    IF DOWNLOADMON <> 1 THEN GOSUB MONDWNLOD
       OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
       OPEN "\DEV\IEEEIN" FOR INPUT AS #2
       OPEN "O", 3, "HV"
   CLS
      LOCATE 6, 60: COLOR 16, 4: PRINT SPACE$(13)
     FOR I = 7 TO 8
      LOCATE I, 59: COLOR 9, 0: PRINT SPACE$(1); : COLOR 16, 4: PRINT SPACE$(13)
     NEXT I
      LOCATE 7, 64: PRINT "STOP": COLOR 10, 1
      LOCATE 9, 59: COLOR 6, 0: PRINT SPACE$(13)
REDODEEP:

    A$ = "CALL MONCUP;"
    CALL OUTDO(A$)

       A$ = "VREAD DA;"
       CALL OUTDO(A$)
       PRINT #1, "ENTER09; EOI"
       INPUT #2, DA(0), DA(1), DA(2), DA(3), DA(4), DA(5)
       A$ = "VREAD EA;"
       CALL OUTDO(A$)
       PRINT #1, "ENTER09; EOI"
       INPUT #2, EA(0), EA(1), EA(2), EA(3), EA(4), EA(5)
       A$ = "VREAD FA;"
       CALL OUTDO(A$)
```

```
PRINT #1, "ENTER09; EOI"
INPUT #2, FA(0), FA(1), FA(2), FA(3), FA(4), FA(5)

A$ = "VREAD K;"
CALL OUTDO(A$)
PRINT #1, "ENTER09; "
INPUT #2, K
 'PULSES/SEC*(COULOMB/PULSE   10^-10)=10^-10   COULOMB/SEC;   1
COULOMB/SEC = 1 AMP AND 1 NANOAMP/10^-9 AMP IS UNITY
 ' THEREFORE PULSES/SEC * 10^-10/10^-9 * 24.912805 CUP AREAS/CM^2 =
NANOAMPS/CM^2
    FOR I = 0 TO 4
   ONCMD(I) = (DA(I) / 2) * .1 * 24.9128
   ONCME(I) = (EA(I) / 2) * .1 * 24.9128
   ONCMF(I) = (FA(I) / 2) * .1 * 24.9128
    NEXT I
   ONCMAC(0) = 0: ONCMAC(1) = 0: ONCMAC(2) = 0
 FOR I = 0 TO 4
   ONCMAC(0) = ONCMAC(0) + ONCMD(I)
   ONCMAC(1) = ONCMAC(1) + ONCME(I)
   ONCMAC(2) = ONCMAC(2) + ONCMF(I)
 NEXT I
    ONCMAC(0) = ONCMAC(0) / 5
    ONCMAC(1) = ONCMAC(1) / 5
    ONCMAC(2) = ONCMAC(2) / 5
LOCATE 2, 7: COLOR 0, 7
 PRINT "Cup Current *** NANOAMPS / CM^2 ***": COLOR 2, 1
  PRINT " "

 PRINT " FOR 400 KV.                    AVG."
 COLOR 0, 6
CALL MAKEWIND(6, 4, 8, 40, 2, 7, 6, 1, 0, lab$)
LOCATE 6, 7: PRINT USING "#######."; ONCMD(0); SPC(4); ONCMD(1): LOCATE
6, 42: COLOR 1, 0: PRINT " ": COLOR 0, 6
LOCATE 7, 7: PRINT USING "#######."; SPC(6); ONCMD(2); SPC(11); ONCMAC(0):
LOCATE 7, 42: COLOR 1, 0: PRINT " ": COLOR 0, 6
LOCATE 8, 7: PRINT USING "#######."; ONCMD(3); SPC(4); ONCMD(4): LOCATE
8, 42: COLOR 1, 0: PRINT " ": LOCATE 9, 42: PRINT " ": COLOR 0, 6
 LOCATE 10, 4: COLOR 1, 0: PRINT SPACE$(39);
 LOCATE 11, 2: COLOR 2, 1
PRINT "FOR 2.5 MV.       "
PRINT " "
 COLOR 0, 6
 CALL MAKEWIND(13, 4, 15, 40, 2, 7, 6, 1, 0, lab$)
LOCATE 13, 7: PRINT USING "#######."; ONCME(0); SPC(4); ONCME(1): LOCATE
13, 42: COLOR 1, 0: PRINT " ": COLOR 0, 6
```

```
   LOCATE 14, 7: PRINT USING "#######."; SPC(6); ONCME(2); SPC(11);
ONCMAC(1): LOCATE 14, 42: COLOR 1, 0: PRINT " ": COLOR 0, 6
   LOCATE 15, 7: PRINT USING "#######."; ONCME(3); SPC(4); ONCME(4): LOCATE
15, 42: COLOR 1, 0: PRINT " ": LOCATE 16, 42: PRINT " ": COLOR 0, 6
    LOCATE 17, 4: COLOR 1, 0: PRINT SPACE$(39);
    LOCATE 18, 2
    COLOR 2, 1
   PRINT "FOR 5   MV.       "
   PRINT " "
     COLOR 0, 6
     CALL MAKEWIND(20, 4, 22, 40, 2, 7, 6, 1, 0, lab$)
   LOCATE 20, 7: PRINT USING "#######."; ONCMF(0); SPC(4); ONCMF(1): LOCATE
20, 42: COLOR 1, 0: PRINT " ": COLOR 0, 6
   LOCATE 21, 7: PRINT USING "#######."; SPC(6); ONCMF(2); SPC(11);
ONCMAC(2): LOCATE 21, 42: COLOR 1, 0: PRINT " ": COLOR 0, 6
   LOCATE 22, 7: PRINT USING "#######."; ONCMF(3); SPC(4); ONCMF(4); :
LOCATE 22, 42: COLOR 1, 0: PRINT " ": LOCATE 23, 42: PRINT " ": COLOR 0, 6
      LOCATE 24, 4: COLOR 1, 0: PRINT SPACE$(39);
     COLOR 0, 7
     ON K + 1 GOTO 1, 2, 3, 4, 5
1 :
 K = 11: GOTO 6
2 :
 K = 13: GOTO 6
3 :
 K = 22: GOTO 6
4 :
 K = 31: GOTO 6
5 :
 K = 33: GOTO 6
6 :
 LOCATE 14, 60: PRINT "NEXT UPDATE"; K

              GSTOP = 0
             GOSUB STMONCUP
              IF GSTOP <> 1 THEN GOTO REDODEEP
               A$ = "OPEN 1206"
               CALL OUTDO(A$)
               CLOSE #1
               CLOSE #2
               CLOSE #3
                RETURN


STMONCUP:
```

```
  InRegs.AX = &H600
  InRegs.BP = &HC6
   CALL INTERRUPT(&H6F, InRegs, OutRegs)
  STATUS = OutRegs.AX: STATE = OutRegs.DX: ROW = OutRegs.CX + 1
  COL = OutRegs.BX + 1
IF STATE = 32640 THEN GOTO CHEKSTOP ELSE RETURN
   RETURN

CHEKSTOP:
    IF ROW >= 5 AND ROW <= 8 AND COL >= 60 AND COL <= 75 THEN GSTOP
= 1
  RETURN

MONDWNLOD:

OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
OPEN "\DEV\IEEEIN" FOR INPUT AS #2
DIM AA$(21)
PRINT #1, "FILL ERROR"
PRINT #1, "CONFIG /I"
PRINT #1, "TIME OUT 70"
A$ = "CLOSE 1206;"
CALL OUTDO(A$)
K = 0
CLS
A$ = "INTEGER IA,MA,NA,PA,TA,K;"
CALL OUTDO(A$)
A$ = "REAL DA(4),EA(4),FA(4);"
CALL OUTDO(A$)
A$ = "CLROUT;"
CALL OUTDO(A$)
AA$(0) = "SUB MONCUP"
AA$(1) = "USE 1400; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
AA$(2) = "USE 1500; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
AA$(3) = "USE 1600; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
AA$(4) = "MA = 1000"
AA$(5) = "MA = MA +K; NA = MA + 5; PA = MA +10"
AA$(6) = "CLOSE MA,NA,PA"
AA$(7) = "WAIT 1"
AA$(8) = "TRIG SGL USE 1400"
AA$(9) = "TRIG SGL USE 1500"
AA$(10) = "TRIG SGL USE 1600"
AA$(11) = "CLOSE 1207"
AA$(12) = "WAIT 1.995"
AA$(13) = "OPEN 1207"
AA$(14) = "CHREADZ 1400 INTO DA(K); CHREADZ 1500 INTO EA(K); CHREADZ 1600
```

```
INTO FA(K)"
AA$(15) = "OPEN MA,NA,PA"
AA$(16) = "K = K + 1"
AA$(17) = "IF K = 5 THEN "
AA$(18) = "K=0"
AA$(19) = "ELSE"
AA$(20) = "END IF"
AA$(21) = "SUBEND"
FOR IA = 0 TO 21
A$ = AA$(IA)
CALL OUTDO(A$)
NEXT IA
 DOWNLOADMON = 1
  CLOSE #1
  CLOSE #2
   RETURN


   GOSUB DOLITTLE
     GOSUB RETCRT
     Q = 0: L = 0: DRT = 0
     RETURN

SUP4:
     COLOR 15, 1
     GOSUB RETCRT
     GOSUB EV1
     OPEN "O", 1, "HV"
     PRINT #1, "THE  LOADING CHAMBER VALVE  IS " + V1$
     PRINT #1, "THE MAIN CHAMBER VALVE  IS " + V2$
     PRINT #1, "THE   O  J    CHAMBER VALVE  IS " + V3$
     CLOSE #1
     GOSUB DOLITTLE
     GOSUB RETCRT
     Q = 0: L = 0: DRT = 0
     RETURN

MNU7:
    IF ROW < 8 THEN
    GOSUB RETCRT
    Q = 0: L = 0: DRT = 0
    RETURN
      ELSEIF COL < 61 OR COL > 69 OR ROW > 20 THEN
      RETURN
        ELSE
        ON INT(((ROW * 3) + 7) / 10) - 2 GOTO SUP5, SUP6, SUP7, SUP8
```

```
            END IF
            RETURN

SUP5:

        COLOR 15, 1
        GOSUB RETCRT
        CALL IONMAG
        GOSUB RETCRT
        Q = 0: L = 0: DRT = 0
        RETURN

SUP6:
        COLOR 15, 1
        GOSUB RETCRT
        CALL MAGCAL
        GOSUB RETCRT
        Q = 0: L = 0: DRT = 0
        RETURN

SUP7:
        COLOR 15, 1
        GOSUB RETCRT
        CALL TILCTP
RETTCTP:
        GOSUB RETCRT
        Q = 0: L = 0: DRT = 0
        RETURN

SUP8:
        COLOR 15, 1
        GOSUB RETCRT
        CALL TILCURA
RETTURA:
        GOSUB RETCRT
        Q = 0: L = 0: DRT = 0
        RETURN

MNU8:
        IF ROW < 8 THEN
        GOSUB RETCRT
        Q = 0: L = 0: DRT = 0
        RETURN
          ELSEIF COL < 71 OR COL > 79 OR ROW > 20 THEN
          RETURN
            ELSE
```

```
        ON INT(((ROW * 3) + 7) / 10) - 2 GOTO SUP9, SUP10, SUP11, SUP12
          END IF
          RETURN

SUP9:

     COLOR 15, 1
     GOSUB RETCRT
     PRINT " THIS SECTION WILL ACCOMIDATE FUTURE GROWTH IN THE
SYSTEM"
     GOSUB DOLITTLE
     GOSUB RETCRT
     Q = 0: L = 0: DRT = 0
     RETURN

SUP10:
     COLOR 15, 1
     GOSUB RETCRT
     PRINT " THIS SECTION WILL ACCOMIDATE FUTURE GROWTH IN THE
SYSTEM"
     GOSUB DOLITTLE
     GOSUB RETCRT
     Q = 0: L = 0: DRT = 0
     RETURN

SUP11:
     COLOR 15, 1
     GOSUB RETCRT
     PRINT " THIS SECTION WILL ACCOMIDATE FUTURE GROWTH IN THE
SYSTEM"
     GOSUB DOLITTLE
     GOSUB RETCRT
     Q = 0: L = 0: DRT = 0
     RETURN

SUP12:
     COLOR 15, 1
     GOSUB RETCRT
     PRINT " THIS SECTION WILL ACCOMIDATE FUTURE GROWTH IN THE
SYSTEM"
     PRINT " CHANGES CAN BE MADE USING THE MENU SETUP                "
     GOSUB DOLITTLE
     GOSUB RETCRT
     Q = 0: L = 0: DRT = 0
    RETURN
EV1:
```

```
    OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
    OPEN "\DEV\IEEEIN" FOR INPUT AS #2
    PRINT #1, "FILL ERROR"
    INFO$ = "OUTPUT09; READ 1100"
    PRINT #1, INFO$
    PRINT #1, "ENTER09"
    INPUT #2, AIN$
    AIN = VAL(AIN$)
    CLOSE #1
    CLOSE #2

    V1$ = "CLOSED"
    V2$ = "CLOSED"
    V3$ = "CLOSED"
     IF AIN < 256 AND AIN > 1792 GOTO RETN1
      IF AIN >= 1024 THEN V3$ = "OPEN": AIN = AIN - 1024
      IF AIN >= 512 THEN V2$ = "OPEN": AIN = AIN - 512
      IF AIN >= 256 THEN V1$ = "OPEN"

RETN1:
   RETURN
DEEPCUP:
   IF DOWNLOADEEP <> 1 THEN GOSUB DWNLOD
       OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
       OPEN "\DEV\IEEEIN" FOR INPUT AS #2
       OPEN "O", 3, "HV"
    A$ = "CALL DEEP_CUP"
    CALL OUTDO(A$)

       A$ = "VREAD D"
       CALL OUTDO(A$)
       PRINT #1, "ENTER09; EOI"
       INPUT #2, D(0), D(1), D(2), D(3), D(4), D(5)
       A$ = "VREAD E"
       CALL OUTDO(A$)
       PRINT #1, "ENTER09; EOI"
       INPUT #2, E(0), E(1), E(2), E(3), E(4), E(5)
       A$ = "VREAD F"
       CALL OUTDO(A$)
       PRINT #1, "ENTER09; EOI"
       INPUT #2, F(0), F(1), F(2), F(3), F(4), F(5)
       A$ = "VREAD AC"
       CALL OUTDO(A$)
       PRINT #1, "ENTER09; EOI"
       INPUT #2, AC(0), AC(1), AC(2), AC(3)
         CLS
```

```
FOR I = 0 TO 4: PRINT D(I): NEXT I

PRINT " HEADINGS "
 FOR I = 0 TO 4
ONCMD(I) = (D(I) / 2) * .1 * 24.9128

ONCME(I) = (E(I) / 2) * .1 * 24.9128
ONCMF(I) = (F(I) / 2) * .1 * 24.9128
 NEXT I
  FOR J = 0 TO 2
   ONCMAC(J) = (AC(J) / 2) * .1 * 24.9128
  NEXT J
     PRINT " Cup Current *** NANOAMPS / CM^2 ***"
     PRINT " "
     PRINT USING "\      \"; SPC(23); "11"; "13"; "22"; "31"; "33"; "AVG."
     PRINT " "
   PRINT " FOR 400 KV.       ";
  PRINT  USING  "#######."; ONCMD(0); ONCMD(1); ONCMD(2);
ONCMD(3); ONCMD(4); ONCMAC(0)
   PRINT " FOR 2.5 MV.       ";
  PRINT  USING  "#######."; ONCME(0); ONCME(1); ONCME(2);
ONCME(3); ONCME(4); ONCMAC(1)
   PRINT " FOR 5   MV.       ";
  PRINT  USING  "#######."; ONCMF(0); ONCMF(1); ONCMF(2);
ONCMF(3); ONCMF(4); ONCMAC(2)
   PRINT " "
   PRINT " "
     LPRINT " Cup Current *** NANOAMPS / CM^2 ***"
      LPRINT " "
     LPRINT USING "\      \"; SPC(23); "11"; "13"; "22"; "31"; "33"; "AVG."
       LPRINT " "
   LPRINT " FOR 400 KV.       ";
  LPRINT  USING  "#######."; ONCMD(0); ONCMD(1); ONCMD(2);
ONCMD(3); ONCMD(4); ONCMAC(0)
   LPRINT " FOR 2.5 MV.       ";
  LPRINT  USING  "#######."; ONCME(0); ONCME(1); ONCME(2);
ONCME(3); ONCME(4); ONCMAC(1)
   LPRINT " FOR 5   MV.       ";
  LPRINT  USING  "#######."; ONCMF(0); ONCMF(1); ONCMF(2);
ONCMF(3); ONCMF(4); ONCMAC(2)

 CLOSE #1
 CLOSE #2
 CLOSE #3
  RETURN
```

```
DWNLOD:

OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
OPEN "\DEV\IEEEIN" FOR INPUT AS #2

PRINT #1, "FILL ERROR"
PRINT #1, "CONFIG /I"
PRINT #1, "TIME OUT 70"
CLS
A$ = "CLROUT"
CALL OUTDO(A$)
A$(0) = "SUB DEEP_CUP"
A$(1) = "CLOSE 1206"
A$(2) = "WAIT 2"
A$(3) = "USE 1400; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(4) = "USE 1500; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(5) = "USE 1600; TRIG HOLD; TERM ISO, ISO; FUNC TOTAL; EDGE LH, HI"
A$(6) = "INTEGER I,M,N,P,T"
A$(7) = "REAL D(4),E(4),F(4)"
A$(8) = "REAL AC(2), AB(2)"
A$(9) = "FOR T = 0 TO 2; AC(T) = 0; AB(T) = 0; NEXT T"
A$(10) = "FOR I = 0 TO 4"
A$(11) = "M = 1000"
A$(12) = "M=M+I;N=M+5;P=M+10"
A$(13) = "CLOSE M,N,P"
A$(14) = "WAIT 1"
A$(15) = "TRIG SGL USE 1400"
A$(16) = "TRIG SGL USE 1500"
A$(17) = "TRIG SGL USE 1600"
A$(18) = "CLOSE 1207"
A$(19) = "WAIT 1.995"
A$(20) = "OPEN 1207"
A$(21) = "CHREADZ 1400 INTO D(I); CHREADZ 1500 INTO E(I); CHREADZ 1600 INTO
F(I)"
A$(22) = "AC(0)=AC(0)+D(I); AC(1)=AC(1)+E(I); AC(2)=AC(2)+F(I)"
A$(23) = "OPEN M,N,P"
A$(24) = "NEXT I"
A$(25) = "OPEN 1206"
A$(26) = "FOR T = 0 TO 2; AC(T)=AC(T)/5"  ' AVERAGE COUNTS FOR CUPS
A$(27) = "NEXT T"
A$(28) = "SUBEND"
FOR I = 0 TO 28
A$ = A$(I)
CALL OUTDO(A$)
NEXT I
 DOWNLOADEEP = 1
```

```
  CLOSE #1
  CLOSE #2
   RETURN

 END

'This version is for QuickBASIC version 4.0 ONLY!
DEFINT A-Z

'This version of POPLIST is for QuickBASIC version 4.0 ONLY!

EVNT1:
    COLOR 16, 6:
      DRT$ = STR$(DRT)
      A$ = " RETURN "
      B% = 15: M% = 15: I$(1) = "": I$(2) = "": I$(3) = "Initialize run": I$(4) = "": I$(5) = ""
      I$(6) = "Read deep cup": I$(7) = "": I$(8) = "": I$(9) = "Change run parameter": I$(10)
= ""
      I$(11) = "": I$(12) = "Default alteration": I$(13) = "": I$(14) = "": I$(15) = "Active
controls"
      F% = 5: BA% = 16: HF% = 16: HB% = 1: QU$ = "4:02": SH% = 0

    Q = 1
    IF L > 0 GOTO POP
    GOSUB SAVCRT
    L = L + 1
POP:
    CALL POPLIST(A$, B%, M%, I$(), F%, BA%, HF%, HB%, QU$, SH%, SE%)
    RETURN

EVNT2:
    COLOR 16, 6:
      DRT$ = STR$(DRT)
      A$ = " RETURN "
      B% = 12: M% = 12: I$(1) = "": I$(2) = "": I$(3) = "VALVES": I$(4) = "": I$(5) = ""
      I$(6) = "INSTRUMENTS": I$(7) = "": I$(8) = "": I$(9) = "INSERT CUPS": I$(10) = ""
      I$(11) = "": I$(12) = "REMOVE CUPS"
      F% = 5: BA% = 16: HF% = 16: HB% = 1: QU$ = "4:12": SH% = 0


      Q = 2
      IF L > 0 GOTO POP
      GOSUB SAVCRT
      L = L + 1
      GOTO POP
```

```
EVNT3:
   COLOR 16, 6:
    DRT$ = STR$(DRT)
    A$ = " RETURN "
    B% = 12: M% = 12: I$(1) = "": I$(2) = "": I$(3) = "START RUN": I$(4) = "": I$(5) =
""
    I$(6) = "RESTART   ": I$(7) = "": I$(8) = "": I$(9) = "CHG PARAMETERS": I$(10) =
""
    I$(11) = "": I$(12) = "LST CONTROL"
    F% = 5: BA% = 16: HF% = 16: HB% = 1: QU$ = "4:22": SH% = 0


    Q = 3
    IF L > 0 GOTO POP
    GOSUB SAVCRT
    L = L + 1
    GOTO POP


EVNT4:
   COLOR 16, 6:
    DRT$ = STR$(DRT)
    A$ = " RETURN "
    B% = 12: M% = 12: I$(1) = "": I$(2) = "": I$(3) = "STATUS": I$(4) = "": I$(5) = ""
    I$(6) = "MOVE": I$(7) = "": I$(8) = "": I$(9) = "OPTION": I$(10) = ""
    I$(11) = "": I$(12) = "SUPLIMENT"
    F% = 5: BA% = 16: HF% = 16: HB% = 1: QU$ = "4:32": SH% = 0


    Q = 4
    IF L > 0 GOTO POP
    GOSUB SAVCRT
    L = L + 1
    GOTO POP

EVNT5:
   COLOR 16, 6:
    DRT$ = STR$(DRT)
    A$ = " RETURN "
    B% = 12: M% = 12: I$(1) = "": I$(2) = "": I$(3) = "SETUP": I$(4) = "": I$(5) = ""
    I$(6) = "HEAT UP": I$(7) = "": I$(8) = "": I$(9) = "PYROMETER": I$(10) = ""
    I$(11) = "": I$(12) = "SHUT DOWN"
    F% = 5: BA% = 16: HF% = 16: HB% = 1: QU$ = "4:42": SH% = 0


    Q = 5
```

```
    IF L > 0 GOTO POP
    GOSUB SAVCRT
    L = L + 1
    GOTO POP


EVNT6:
   COLOR 16, 6:
     DRT$ = STR$(DRT)
     A$ = " RETURN "
     B% = 12: M% = 12: I$(1) = "": I$(2) = "": I$(3) = "PYROMETER": I$(4) = "": I$(5) =
""
     I$(6) = "VACUUM": I$(7) = "": I$(8) = "": I$(9) = " DEEP CUP ": I$(10) = ""
     I$(11) = "": I$(12) = "  VALVES  "
     F% = 5: BA% = 16: HF% = 16: HB% = 1: QU$ = "4:52": SH% = 0


    Q = 6
    IF L > 0 GOTO POP
    GOSUB SAVCRT
    L = L + 1
    GOTO POP


EVNT7:
   COLOR 16, 6:
     DRT$ = STR$(DRT)
     A$ = " RETURN "
     B% = 12: M% = 12: I$(1) = "": I$(2) = "": I$(3) = "  ION-MAG  ": I$(4) = "": I$(5) =
""
     I$(6) = "  MAG-CAL  ": I$(7) = "": I$(8) = "": I$(9) = "  TIL-CTP  ": I$(10) = ""
     I$(11) = "": I$(12) = "  TIL-CURA  "
     F% = 5: BA% = 16: HF% = 16: HB% = 1: QU$ = "4:62": SH% = 0


    Q = 7
    IF L > 0 GOTO POP
    GOSUB SAVCRT
    L = L + 1
    GOTO POP
```

```
EVNT8:
   COLOR 16, 6:
     DRT$ = STR$(DRT)
     A$ = " RETURN "
     B% = 12: M% = 12: I$(1) = "": I$(2) = "": I$(3) = "SUPLIMENT 1": I$(4) = "": I$(5) =
""
     I$(6) = "SUPLIMENT 2": I$(7) = "": I$(8) = "": I$(9) = "SUPLIMENT 3": I$(10) = ""
     I$(11) = "": I$(12) = "SUPLIMENT 4"
     F% = 5: BA% = 16: HF% = 16: HB% = 1: QU$ = "4:68": SH% = 0


    Q = 8
    IF L > 0 GOTO POP
    GOSUB SAVCRT
    L = L + 1
    GOTO POP



SAVCRT:
  ame$ = "D:Bss"
  DEF SEG = &HB800
  BSAVE ame$, &H0, &H1FFF
  DEF SEG
  'saves row col displayed although they are altered and are not as displayed on return
RETURN

RETCRT:
  DEF SEG = &HB800
  BLOAD "D:Bss", &H0
  DRT = 0: L = 0
  'rem the value of row col displayed is not the actual value only the what they were when saved.
RETURN
DOLITTLE:
      FOR I = 1 TO 1000
      FOR J = 1 TO 200
      NEXT J
      NEXT I
RETURN
END



DEFSNG A-Z
```

```
SUB IONMAG
DEFSNG A-Z
CLS
PRINT "CALCULATE THE MAGNET CURRENT(amp) AND FREQUENCY(MHz) FOR
E1 M1 Q1-E2 M2 Q2 where"
PRINT "E1=ION ENERGY(MEV), M1=ION MASS(AMU),  of ion before stripper"
PRINT "E2,M2,Q2=ENERGY,MASS,AND CHARGE OF BREAK-UP ION after stripper"
LPRINT "THE STANDARD USED IN THIS CODE IS I=255-amp, F=6522-MHz FOR 4MEV
Ni++"
LPRINT
EST = 4: MST = 58: QST = 2: Q1 = 1
INPUT "enter: ENERGY,MASS of ion before stripper"; E1, M1
INPUT "enter: MASS,CHARGE of ion after stripper"; M2, Q2
PRINT
E2 = E1 * M2 / M1
RATIO1 = (M1 * E1 * QST ^ 2 / MST / EST / Q1 ^ 2) ^ .5
RATIO2 = (M2 * E2 * Q1 ^ 2 / M1 / E1 / Q2 ^ 2) ^ .5
AMP = 255 * RATIO1 * RATIO2
FREQ = 6522 * RATIO1 * RATIO2
PRINT
LPRINT USING "ENERGY OF THE BREAK-UP ION= ###."; E2
LPRINT USING "MAGNET CURRENT SHOULD BE APPROXIMATELY= #####.";
AMP
LPRINT "MAGNET FREQUENCY SHOULD BE = ", FREQ
LPRINT
LPRINT
LPRINT
LPRINT
LPRINT
END SUB

'

SUB MAGCAL
CLS
PRINT "Calibrate a Magnet by Reading One Calibration Point for Normalization"
LPRINT "": LPRINT ""
INPUT "enter: E1(MEV), M1(AMU), Q, IMAG(AMP)"; E1, M1, Q, AMP
INPUT "enter: mass range: MIN, MAX"; MIN, MAX

CAL = AMP / SQR(E1 * M1)
LPRINT "E1, M1, AMP, CAL= "; E1, M1, AMP, CAL: LPRINT ""
 PRINT "E1, M1, AMP, CAL= "; E1, M1, AMP, CAL:  PRINT ""
LPRINT "E1 = ION ENERGY(MEV)   M1 = ION MASS   AMP = MAGNET CURRENT
FOR Q="; Q
 PRINT "E1 = ION ENERGY(MEV)   M1 = ION MASS   AMP = MAGNET CURRENT
FOR Q="; Q
```

```
PRINT ""
FOR M = MIN TO MAX: AM = M: MAGAMP = CAL * SQR(E1 * AM):
 LPRINT USING "#.###"; E1; SPC(18); : LPRINT USING "##"; M; SPC(14);
 LPRINT USING "###.##"; MAGAMP
 PRINT USING "#.###"; E1; SPC(18); : PRINT USING "##"; M; SPC(14);
 PRINT USING "###.##"; MAGAMP
 NEXT M
END SUB

SUB MAKEWIND (ULR%, ULC%, LRR%, LRC%, FRAME%, FORE%, BACK%, GROW%,
SHADOW%, LABEL$) STATIC

    DEFINT A-Z

    IF GROW% = 0 THEN GOSUB MAKEWIND.STD:                    GOTO
MAKEWIND.DONE


    SHADOW% = 0

    X1 = ULC% + (INT((LRC% - ULC%) \ 2))
    X2 = LRC% - (INT((LRC% - ULC%) \ 2))
    Y1 = ULR% + (INT((LRR% - ULR%) \ 2))
    Y2 = LRR% - (INT((LRR% - ULR%) \ 2))

MAKEWIND.NXT:
    IF X1 > ULC% THEN X1 = X1 - 3:          IF X1 < ULC% THEN X1 = ULC%

    IF X2 < LRC% THEN X2 = X2 + 3:          IF X2 > LRC% THEN X2 = LRC%

    IF Y1 > ULR% THEN Y1 = Y1 - 1

    IF Y2 < LRR% THEN Y2 = Y2 + 1

    GOSUB MAKEWIND.SETUP

    IF (X1 = ULC%) AND (X2 = LRC%) AND (Y1 = ULR%) AND (Y2 = LRR%) THEN
GOTO MAKEWIND.DONE

    GOTO MAKEWIND.NXT

MAKEWIND.STD:
    X1 = ULC%
    X2 = LRC%
    Y1 = ULR%
    Y2 = LRR%
```

```
MAKEWIND.SETUP:
    ATTR = (BACK% AND 7) * 16 + FORE%

    IF FRAME% = 0 THEN GOSUB MAKEWIND.NOFRAME ELSE ON FRAME%
GOSUB MAKEWIND.H1V1, MAKEWIND.H2V2, MAKEWIND.H1V2, MAKEWIND.H2V1
    IF (LABEL$ = "") OR (LEN(LABEL$) > LEN(TOP$) - 5) THEN GOTO
MAKEWIND.SHADE

    MID$(TOP$, (LEN(TOP$) / 2) - ((LEN(LABEL$) + 1) / 2)) = "[" + LABEL$ + "]"

MAKEWIND.SHADE:
    IF SHADOW% = 0 THEN GOTO MAKEWIND.MAKE

    COL = X1 - 3
    DAT$ = STRING$((X2 - X1) + 3, 32)
    BLACK = 0

    FOR I = Y1 TO (Y2 + 2)
     ROW = I
     CALL FASTPRT(DAT$, ROW, COL, BLACK)
    NEXT

    SHADOW% = 0
MAKEWIND.MAKE:
    ROW = Y1 - 1
    COL = X1 - 1
    CALL FASTPRT(TOP$, ROW, COL, ATTR)

    FOR I = Y1 TO Y2
     ROW = I
     COL = X1 - 1
     CALL FASTPRT(MIDL$, ROW, COL, ATTR)
    NEXT

    ROW = Y2 + 1
    COL = X1 - 1
    CALL FASTPRT(BOTTM$, ROW, COL, ATTR)
    RETURN

MAKEWIND.H1V1:
    TOP$ = CHR$(218) + STRING$((X2 - X1) + 1, 196) + CHR$(191)
    MIDL$ = CHR$(179) + STRING$((X2 - X1) + 1, 32) + CHR$(179)
    BOTTM$ = CHR$(192) + STRING$((X2 - X1) + 1, 196) + CHR$(217)
    RETURN

MAKEWIND.H2V2:
```

```
    TOP$ = CHR$(201) + STRING$((X2 - X1) + 1, 205) + CHR$(187)
    MIDL$ = CHR$(186) + STRING$((X2 - X1) + 1, 32) + CHR$(186)
    BOTTM$ = CHR$(200) + STRING$((X2 - X1) + 1, 205) + CHR$(188)
    RETURN

MAKEWIND.H1V2:
    TOP$ = CHR$(214) + STRING$((X2 - X1) + 1, 196) + CHR$(183)
    MIDL$ = CHR$(186) + STRING$((X2 - X1) + 1, 32) + CHR$(186)
    BOTTM$ = CHR$(211) + STRING$((X2 - X1) + 1, 196) + CHR$(189)
    RETURN

MAKEWIND.H2V1:
    TOP$ = CHR$(213) + STRING$((X2 - X1) + 1, 205) + CHR$(184)
    MIDL$ = CHR$(179) + STRING$((X2 - X1) + 1, 32) + CHR$(179)
    BOTTM$ = CHR$(212) + STRING$((X2 - X1) + 1, 205) + CHR$(190)
    RETURN

MAKEWIND.NOFRAME:
    TOP$ = SPACE$((X2 - X1) + 3)
    MIDL$ = TOP$
    BOTTM$ = TOP$
    RETURN

MAKEWIND.DONE:
    GROW% = 0

END SUB

SUB OUTDO (A$) STATIC
INFO$ = "OUTPUT09;" + A$
PRINT #1, INFO$
END SUB

DEFSNG A-Z

    SUB POPLIST (HEADER$, SHOWITEMS%, MAXITEMS%, ITEM$(), FORE%,
BACK%, HFORE%, HBACK%, QUADRANT$, SHADOW%, SLCT%) STATIC
    DEFINT A-Z
    FRAME% = 4
    IF SHOWITEMS > MAXITEMS THEN SHOWITEMS = MAXITEMS

    WINDLEN = LEN(HEADER$)
    FOR J = 1 TO MAXITEMS
      IF LEN(ITEM$(J)) > WINDLEN THEN WINDLEN = LEN(ITEM$(J))
    NEXT J
```

```
     IF INSTR(QUADRANT$, ":") <> 0 THEN GOSUB GETORD: GOTO GO1

     QUADRANT = VAL(QUADRANT$)
     IF QUADRANT > 4 OR QUADRANT < 0 THEN QUADRANT = 0
     IF QUADRANT = 0 THEN CROW = 12: CCOL = 40 ELSE ON QUADRANT GOSUB
QUAD1, QUAD2, QUAD3, QUAD4
     ULR = CROW - ((SHOWITEMS + 2) / 2 - .5)
     ULC = CCOL - ((WINDLEN / 2) - .5)
     LRR = ULR + SHOWITEMS + 1
     LRC = ULC + WINDLEN - 1

GO1:    CALL MAKEWIND(ULR, ULC, LRR, LRC, FRAME, FORE, BACK, GROW,
SHADOW, LABEL$)


     TEMPHDR$ = SPACE$(WINDLEN)
     IF LEN(HEADER$) <> WINDLEN THEN GOSUB PUTHDR

     ATTR = (HBACK AND 7) * 16 + HFORE
     ROW = ULR: COL = ULC
     CALL FASTPRT(HEADER$, ROW, COL, ATTR)
     ATTR = (BACK AND 7) * 16 + FORE
     ROW = ULR + 1: COL = ULC
     DAT$ = STRING$(WINDLEN, 205)
     CALL FASTPRT(DAT$, ROW, COL, ATTR)


     SLCT = 1
     BEGVAL = 1
     ENDVAL = SHOWITEMS
     GOSUB FILL

LOPE:  GOTO DONE


PRESS: KP$ = INKEY$
     IF KP$ = "" THEN GOTO PRESS
     IF KP$ = CHR$(13) THEN RETURN
     IF KP$ = CHR$(27) THEN SLCT = 0: RETURN
     IF LEN(KP$) = 1 THEN SOUND 1000, 1: SOUND 1500, 2: SOUND 500, 1: GOTO
PRESS


DOWN:  IF ASC(RIGHT$(KP$, 1)) = 80 THEN OLD = SLCT: SLCT = SLCT + 1 ELSE
GOTO UP
     IF SLCT > MAXITEMS THEN SLCT = MAXITEMS: SOUND 1000, 1: SOUND 1500,
```

```
2: SOUND 500, 1
    IF (SLCT > ENDVAL) AND (SLCT = MAXITEMS) THEN BEGVAL = BEGVAL +
1: ENDVAL = ENDVAL + 1: GOSUB FILL: RETURN
    IF (SLCT > ENDVAL) AND (SLCT <> MAXITEMS) THEN BEGVAL = BEGVAL
+ 1: ENDVAL = ENDVAL + 1: GOSUB FILL: RETURN
    GOSUB FILL: RETURN


UP:   IF ASC(RIGHT$(KP$, 1)) = 72 THEN OLD = SLCT: SLCT = SLCT - 1 ELSE GOTO
PG.UP
    IF SLCT < 1 THEN SLCT = 1: SOUND 1000, 1: SOUND 1500, 2: SOUND 500, 1
    IF SLCT < BEGVAL AND SLCT = 1 THEN BEGVAL = BEGVAL - 1: ENDVAL =
ENDVAL - 1: GOSUB FILL: RETURN
    IF SLCT < BEGVAL AND SLCT <> 1 THEN BEGVAL = BEGVAL - 1: ENDVAL
= ENDVAL - 1: GOSUB FILL: RETURN
    GOSUB FILL: RETURN


PG.UP: IF ASC(RIGHT$(KP$, 1)) = 73 THEN OLD = SLCT: SLCT = SLCT - SHOWITEMS
ELSE GOTO PG.DN
    IF SLCT < 1 THEN SLCT = 1: SOUND 1000, 1: SOUND 1500, 2: SOUND 500, 1
    BEGVAL = BEGVAL - SHOWITEMS: ENDVAL = ENDVAL - SHOWITEMS
    IF BEGVAL < 1 THEN BEGVAL = 1: ENDVAL = SHOWITEMS
    GOSUB FILL: RETURN

PG.DN: IF ASC(RIGHT$(KP$, 1)) = 81 THEN OLD = SLCT: SLCT = SLCT +
SHOWITEMS ELSE GOTO HOME
    IF SLCT > MAXITEMS THEN SLCT = MAXITEMS: SOUND 1000, 1: SOUND 1500,
2: SOUND 500, 1
    BEGVAL = BEGVAL + SHOWITEMS: ENDVAL = ENDVAL + SHOWITEMS
    IF ENDVAL > MAXITEMS THEN ENDVAL = MAXITEMS: BEGVAL = ENDVAL
- SHOWITEMS + 1
    GOSUB FILL: RETURN

HOME:  IF ASC(RIGHT$(KP$, 1)) = 71 THEN OLD = SLCT: SLCT = 1 ELSE GOTO
ENDK
    BEGVAL = 1: ENDVAL = BEGVAL + SHOWITEMS - 1
    GOSUB FILL: RETURN

ENDK:  IF ASC(RIGHT$(KP$, 1)) = 79 THEN OLD = SLCT: SLCT = MAXITEMS ELSE
GOTO ERRCHK
    ENDVAL = MAXITEMS: BEGVAL = ENDVAL - SHOWITEMS + 1
    GOSUB FILL: RETURN

ERRCHK: SOUND 1000, 1: SOUND 1500, 2: SOUND 500, 1: GOTO PRESS
```

```
FILL:  K = 1
     IF BEGVAL < 1 THEN BEGVAL = 1
     IF ENDVAL > MAXITEMS THEN ENDVAL = MAXITEMS
     OFFSET = ENDVAL - SLCT: IF OFFSET < 0 THEN OFFSET = 0 ELSE IF OFFSET
> SHOWITEMS - 1 THEN OFFSET = SHOWITEMS - 1
     FOR J = BEGVAL TO ENDVAL
       ATTR = (BACK AND 7) * 16 + FORE
       ROW = (ULR + 1 + K): COL = ULC
       DAT$ = ITEM$(J)
       CALL FASTPRT(DAT$, ROW, COL, ATTR)
       K = K + 1
     NEXT J
     ATTR = (FORE AND 7) * 16 + BACK
     IF BEGVAL = 1 AND SLCT = 1 THEN ROW = ULR + 2 ELSE IF SLCT >=
BEGVAL AND SLCT <= ENDVAL THEN ROW = ULR + 2 + SLCT
     IF ENDVAL = MAXITEMS AND SLCT = MAXITEMS THEN ROW = LRR ELSE
ROW = LRR - OFFSET
     COL = ULC
     DAT$ = ITEM$(SLCT)
     CALL FASTPRT(DAT$, ROW, COL, ATTR)
     GOSUB MORE
     RETURN

MORE:  MCOL = ULC + ((LRC - ULC) / 2) - 1
     IF BEGVAL > 1 THEN MROW = ULR + 1: DAT$ = " " + CHR$(30) + " ": GOSUB
DISP
     IF ENDVAL < MAXITEMS THEN MROW = LRR + 1: DAT$ = " " + CHR$(31) +
" ": GOSUB DISP
     IF SLCT = 1 THEN MROW = ULR + 1: DAT$ = STRING$(3, 205): GOSUB DISP
     IF SLCT = MAXITEMS THEN MROW = LRR + 1: DAT$ = STRING$(3, 205):
GOSUB DISP
     RETURN

DISP:  ATTR = (BACK AND 7) * 16 + FORE
     CALL FASTPRT(DAT$, MROW, MCOL, ATTR)
     RETURN

QUAD1: CROW = 7
     CCOL = 20
     RETURN
QUAD2: CROW = 7
     CCOL = 60
     RETURN
QUAD3: CROW = 18
     CCOL = 60
     RETURN
```

```
QUAD4: CROW = 18
     CCOL = 20
     RETURN

GETORD:

     ULR = VAL(LEFT$(QUADRANT$, 2)) + 1
     ULC = VAL(RIGHT$(QUADRANT$, 2))
     LRR = ULR + SHOWITEMS + 1
     LRC = ULC + WINDLEN - 1
     RETURN

PUTHDR:

     PAD = (WINDLEN / 2) - (LEN(HEADER$) / 2) - .5
     MID$(TEMPHDR$, PAD + 1, LEN(HEADER$)) = HEADER$
     HEADER$ = TEMPHDR$
     RETURN

DONE:  END SUB

DEFSNG A-Z
SUB TILCTP

DEFSNG A-S, U-Z
DEFSTR T
DIM C(26), Z(10), TSTIME$(60), DELTIM(60), AC0(60), AC1(60), AC2(60), AB0(60),
AB1(60), AB2(60), CN40(60), CN41(60), CN42(60)
DIM DLT(600), TTIME$(600), AIN$(10)
DIM CONVD(600), CONVE(600), CONVF(600), SIGMD(600), SIGME(600), SIGMF(600)
CLS
INPUT "INPUT THE DATA FILE ID # "; DISK$
INPUT "INPUT THE TEMPERATURE-PRESSURE DATA FILE ID # "; DISKTP$
DISK$ = "A:" + DISK$
DISKTP$ = "A:" + DISKTP$
OPEN DISK$ FOR INPUT AS #10
OPEN DISKTP$ FOR INPUT AS #11
  FOR I = 1 TO 7
    INPUT #10, AIN$(I)
  NEXT I
  FOR I = 1 TO 6
    LPRINT AIN$(I)
  NEXT I
LPRINT " "
LPRINT "  TIME     CURRENT (NANOAMPS/CM^2)    BEAM     TEMPERATURE
PRESSURE"
```

```
LPRINT "  OF DAY    400     2500      5000     TIME(M)    DEGREES C     TORR"
LPRINT ""
FOR J = 0 TO 26
   INPUT #10, C(J)

  NEXT J

  FOR I = 0 TO 10
   INPUT #10, Z(I)

  NEXT I

  CLS
  LC1 = 0
BTIME = 0

INPUT #11, DUMMY$
DO WHILE NOT EOF(10)
    LC1 = LC1 + 1: LC2 = 0
   INPUT #10, TMARKER1$, TSTIME$(LC1), DELTIM(LC1)
       INPUT #10, AC0(LC1), AB0(LC1), CN40(LC1)
       INPUT #10, AC1(LC1), AB1(LC1), CN41(LC1)
       INPUT #10, AC2(LC1), AB2(LC1), CN42(LC1)
       INPUT #10, VARD0, VARE0, VARF0
       INPUT #10, VARD1, VARE1, VARF1
       INPUT #10, VARD2, VARE2, VARF2
        INPUT #10, VARD3, VARE3, VARF3
        INPUT #10, VARD4, VARE4, VARF4

           C400 = AC0(LC1) * 2.49 / 2
           C2500 = AC1(LC1) * 2.49 / 2
           C5 = AC2(LC1) * 2.49 / 2
LPRINT USING "####.#"; SPC(11); C400; SPC(2); C2500; SPC(5); C5
INPUT #11, dumy1, dumy2, dumy3

POINT2:
DO WHILE NOT EOF(10) AND LC2 < 10
      LC2 = LC2 + 1
      LC3 = 10 * LC1 + LC2
INPUT #10, TMARKER2$, DLT(LC3), TTIME$(LC3)
'CONVD,E,F=400,2500,5 ave. ions/cm2; SIGM=accumulated ions/cm2
       INPUT #10, CONVD(LC3), CONVE(LC3), CONVF(LC3), SIGMD(LC3),
SIGME(LC3), SIGMF(LC3)
         INPUT #10, FSD0, FSE0, FSF0
         INPUT #10, FSD1, FSE1, FSF1
         INPUT #10, FSD2, FSE2, FSF2
```

```
        INPUT #10, FSD3, FSE3, FSF3
        INPUT #10, FSD4, FSE4, FSF4
'SC=current deduced from bpm data; C(24,25,26)=ion charge state
 SCD = CONVD(LC3) * 2.49128 * C(24) / (1.55705E+10 * DLT(LC3))
 SCE = CONVE(LC3) * 2.49128 * C(25) / (1.55705E+10 * DLT(LC3))
 SCF = CONVF(LC3) * 2.49128 * C(26) / (1.55705E+10 * DLT(LC3))
BTIME = BTIME + DLT(LC3) / 60
INPUT #11, DUMMY$, PRES$, TEMP$
LPRINT ; TTIME$(LC3); USING "####.#"; SPC(3); SCD; SPC(2); SCE; SPC(5); SCF;
SPC(4); BTIME;
LPRINT SPC(4); TEMP$; SPC(7); PRES$
LOOP
LOOP
CLOSE #10
CLOSE #11
END SUB

DEFINT A-Z
SUB TILCURA
DEFSNG A-S, U-Z
DEFSTR T
DIM C(26), Z(10), TSTIME$(60), DELTIM(60), AC0(60), AC1(60), AC2(60), AB0(60),
AB1(60), AB2(60), CN40(60), CN41(60), CN42(60)
'DIM VARD0(60), VARD1(60), VARD2(60), VARD3(60), VARD4(60)
'DIM VARE0(60), VARE1(60), VARE2(60), VARE3(60), VARE4(60)
'DIM VARF0(60), VARF1(60), VARF2(60), VARF3(60), VARF4(60)
DIM DLT(600), TTIME$(600), AIN$(10)
DIM CONVD(600), CONVE(600), CONVF(600), SIGMD(600), SIGME(600), SIGMF(600)
'DIM FSD1(600), FSD2(600), FSD3(600), FSD4(600), FSD0(600)
'DIM FSE1(600), FSE2(600), FSE3(600), FSE4(600), FSE0(600)
'DIM FSF1(600), FSF2(600), FSF3(600), FSF4(600), FSF0(600)
CLS
INPUT "INPUT THE DATA FILE # "; DISK$
DISK$ = "A:\" + DISK$
'read and
'print run-identification
OPEN DISK$ FOR INPUT AS #10
  FOR I = 1 TO 7
    INPUT #10, AIN$(I)
  NEXT I
  FOR I = 1 TO 6
    LPRINT AIN$(I)
  NEXT I
LPRINT " "
LPRINT "    TIME            CURRENT  (NANOAMPS/CM^2)          BEAM
---IONS(E15)/CM2----"
```

```
LPRINT "  OF DAY    400     2500      5000     TIME(M)    400      2500      5000"
LPRINT ""
'read without printing run-parameters c(i) and z(i)
FOR J = 0 TO 26
    INPUT #10, C(J)

  NEXT J

  FOR I = 0 TO 10
    INPUT #10, Z(I)

  NEXT I

  CLS
  LC1 = 0
'point1 are the cup measurements: AC0,1.2=400,2500,5 ave. cup;AB0,1,2=bpm;CN40,1,2=conv.
fact.
'read digital info.
'btime = total time of beam on target
BTIME = 0
DO WHILE NOT EOF(10)
     LC1 = LC1 + 1: LC2 = 0
    INPUT #10, TMARKER1$, TSTIME$(LC1), DELTIM(LC1)

        INPUT #10, AC0(LC1), AB0(LC1), CN40(LC1)
        INPUT #10, AC1(LC1), AB1(LC1), CN41(LC1)
        INPUT #10, AC2(LC1), AB2(LC1), CN42(LC1)
'read only individual cup info.
          INPUT #10, VARD0, VARE0, VARF0
          INPUT #10, VARD1, VARE1, VARF1
          INPUT #10, VARD2, VARE2, VARF2
          INPUT #10, VARD3, VARE3, VARF3
          INPUT #10, VARD4, VARE4, VARF4
'convert cup data to ave. nanoamps/cm2
              C400 = AC0(LC1) * 2.49 / 2
              C2500 = AC1(LC1) * 2.49 / 2
              C5 = AC2(LC1) * 2.49 / 2
LPRINT USING "####.#"; SPC(11); C400; SPC(2); C2500; SPC(5); C5
'point2 are the bpm data points for the one minute intervals
APOINT2:
DO WHILE NOT EOF(10) AND LC2 < 10
       LC2 = LC2 + 1
       LC3 = 10 * LC1 + LC2
'DLT=60sec bpm measuring interval; TTIME=clock time
INPUT #10, TMARKER2$, DLT(LC3), TTIME$(LC3)
'CONVD,E,F=400,2500,5 ave. ions/cm2; SIGM=accumulated ions/cm2
```

```
        INPUT #10, CONVD(LC3), CONVE(LC3), CONVF(LC3), SIGMD(LC3),
SIGME(LC3), SIGMF(LC3)
          INPUT #10, FSD0, FSE0, FSF0
          INPUT #10, FSD1, FSE1, FSF1
          INPUT #10, FSD2, FSE2, FSF2
          INPUT #10, FSD3, FSE3, FSF3
          INPUT #10, FSD4, FSE4, FSF4
'SC=current deduced from bpm data; C(24,25,26)=ion charge state
 SCD = CONVD(LC3) * 2.49128 * C(24) / (1.55705E+10 * DLT(LC3))
 SCE = CONVE(LC3) * 2.49128 * C(25) / (1.55705E+10 * DLT(LC3))
 SCF = CONVF(LC3) * 2.49128 * C(26) / (1.55705E+10 * DLT(LC3))
 ACCUD = SIGMD(LC3) / 1E+15
 ACCUE = SIGME(LC3) / 1E+15
 ACCUF = SIGMF(LC3) / 1E+15
 BTIME = BTIME + DLT(LC3) / 60
LPRINT ; TTIME$(LC3); USING "####.#"; SPC(3); SCD; SPC(2); SCE; SPC(5); SCF;
SPC(4); BTIME; SPC(4); ACCUD; SPC(3); ACCUE; SPC(4); ACCUF
        LOOP
        LOOP
CLOSE #10

END SUB
```

APPENDIX B OPERATING PROCEDURES FOR ION-IRRADIATION CHAMBER

B.1. SPECIMEN EXCHANGE

B.1.1. Loading Specimen Holder with Transmission Electron Microscope (TEM) Disk Specimens

1. Inspect faceplate for residual platinum ring material and remove if necessary.

2. Clean faceplate and thermalizer blocks with alcohol.

3. Place faceplate [(*a*) in Fig. 4] with beam side down into the loading jig (Fig. 4), locating the plate's outer notch in the lower right 33 (row 3, column 3) position when inverted. Please note that position notation is inverted because you are looking at the back side of the specimen array.

4. If a mask is to be used it should be placed into the recess first.

5. Load a crushable platinum ring into the recess and form it to the bottom of the recess by pressing gently with the chamfered end of the shank portion of a #31 drill.

6. Load the thermocouple specimen facedown with wires aligned with slot, and while holding the specimen flat in the recess, straighten leads and position them to balance the specimen in its position when released.

7. Load the remaining specimens facedown, taking precaution not to disturb the thermocouple specimen.

8. Carefully place the thermalizer block (smooth side down) over the specimens, taking care not to dislodge them.

9. Apply the over-center clamp on the loading device to retain the thermalizer block in position.

10. Remove the knurled index pin and invert the pivoting table. Two of the 2-56 X 1/8-in socket head screws can be inserted through the faceplate and tightened.

11. Invert the pivoting table. Release the over-center clamp. Take the specimen holder out of the loading fixture. Insert the remaining screws and tighten.

12. Place the specimen holder in the compression clamp with the thermocouple wire located in the split prong of the anvil and compress. Tighten the screws uniformly snug and remove the specimen holder.

13. Place insulators on the thermocouple wires if necessary.

B.1.2. Breaking Vacuum and Unloading Specimen

1. The specimen trolley must be transported through valve #1 [(*b*) in Fig. 2] into the loading chamber. On the stepper motor controller select a number of steps greater than 44,000. Check that valve #1 is open. Select out direction and press index switch. Valve #1 must be open!

2. Looking into the loading chamber, press the clear switch on the stepper motor controller as the trolley approaches the loading port valve (valve #4, Fig. B.1). You should be looking down on the first or second station just coming into view.

3. Close valve #1 and close valve #3. Turn off the turbopump and turn off ion gauge #2 [(*c*) in Fig. 2].

4. Bring loading chamber up to atmospheric pressure by backfilling with argon through valves #5 and #2 while monitoring the pressure at thermocouple gauge B.

5. When the pressure inside the chamber is equal to the outside pressure, open valve #4 and reach through the valve opening (operator should be wearing clean surgical gloves), tilting the trolley slightly forward to disengage the rack-pinion gears, and gently pull trolley out of the chamber.

6. Close valve #4. Unscrew two long screws holding the L brackets to the trolley. Disconnect the surface thermocouples from the front connectors. Lift the bracket clear of the trolley, taking care not to drop the specimen holders.

7. Unscrew two opposite socket-head screws on the faceplate. Place the specimen holder in the loading jig and clamp in place with the over-center clamp.

8. Invert the pivoting table to remove the two remaining screws and re-invert so the thermalizer block is on top.

YP 11234

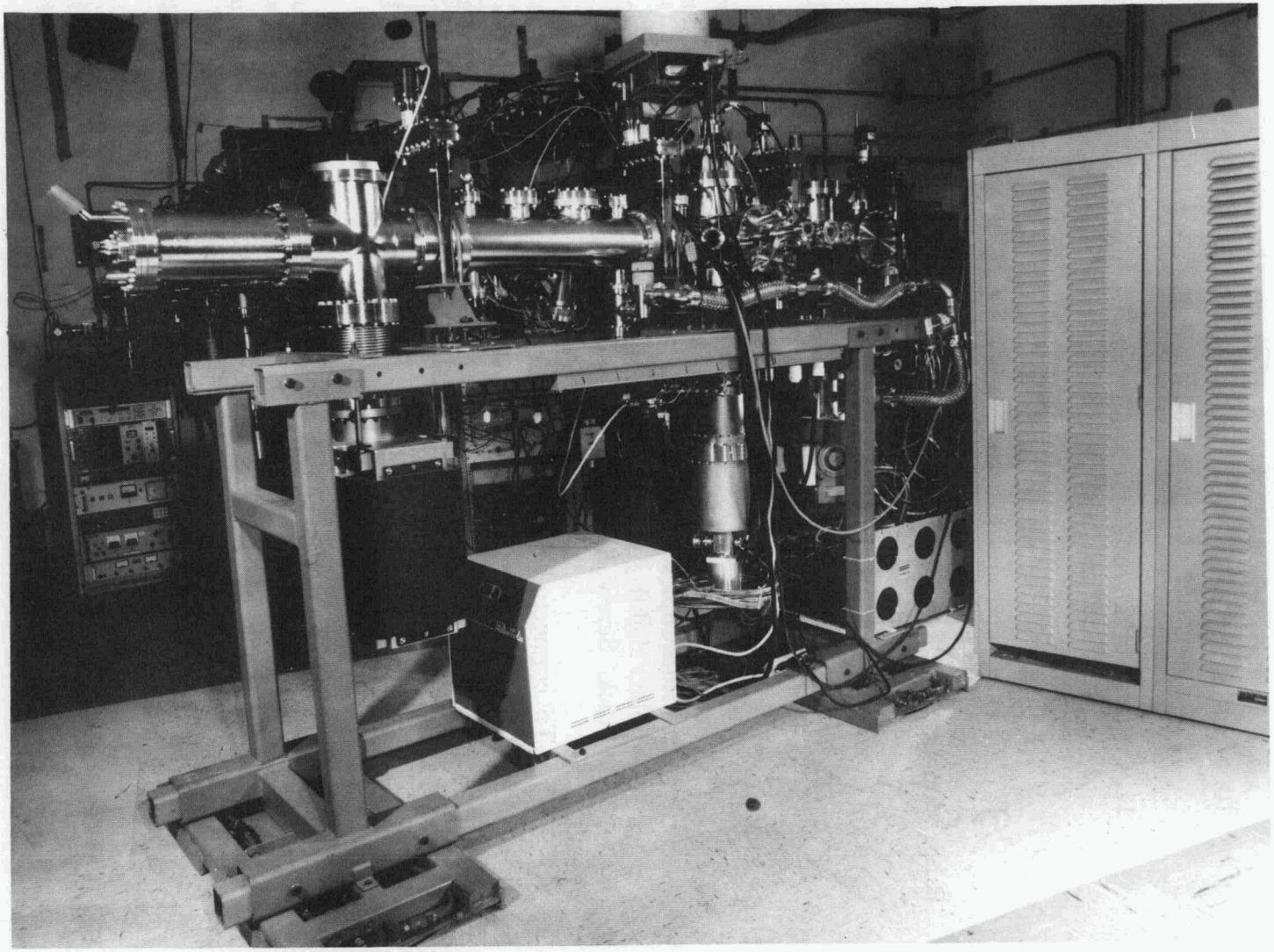

Fig. B.1. Valve locations listed for operating procedures.

9. Carefully unclamp the specimen holder and gently lift the thermalizer block off the top, taking care that no specimens are dislodged or stuck to the thermalizer block.

10. Using tweezers, very gently lift the individual specimens out of the faceplate one at a time, and place them individually into prelabeled vials.

11. Clean off all residual platinum ring remnants from the faceplate and ream spot faced area with the flat-end reamer especially designed for this purpose.

12. Collect and save all platinum scrap for recycle.

B.1.3. Mounting the Specimen Holder and Loading the Trolley

1. Wearing surgical gloves, place the specimen holder into the L bracket with the thermocouple holes in the thermalizer block aligned with the holes in the bracket.

2. Insert the two 6-32 X 2in long screws into the L bracket from the top side and lower the assembly down over the spring-loaded parallel thermocouples on the trolley. You may need to rotate the specimen holder slightly from side-to-side as the thermocouples enter the thermalizer block. Screw the long screws into the trolley and tighten.

3. To connect the surface thermocouple, press on the back side of the trolley's surface thermocouple connector and wrap the lead once around the area between the end of the post and the contact washer. With a small magnet, check to see which of the wires of the surface thermocouple is ferromagnetic and connect it to the connector with the same magnetic properties.

4. With the loading chamber at atmospheric pressure, open valve #4. Reaching through the valve, place the forward end of the trolley on the track while tilting it slightly forward to avoid contact of the rack with the pinion gears. Insert the trolley fully onto the track, indexing the end of the trolley precisely with the end of the track. Lay a straightedge along the end to check alignment.

5. Close valve #4. Open valve #2 and open valve #6 to vent the chamber to the turbopump. Open valve #3 and close valves #2 and #6.

6. Start turbopump and monitor the evacuation of the chamber. If the turbopump fails to reach operating speed within a certain time frame, the unit will shut down. If this happens and no vacuum leaks exist, it may be necessary to manually press the start button and hold it in until the RPM meter registers operating speed (manual override).

7. Continue monitoring the vacuum until the pressure is in the 10^{-7} torr range. Open valve #1.

8. On the stepper motor controller, set direction switch to + and set number of steps to 43,341 to index to center of station #5; 40,796 to index to center of station #4; 38,251 to index to center of station #3; 35,706 to index to center of station #2; and 33,161 to index to center of station #1 or use the following formula to calculate the number of steps: STEPS = 30,616 + 2545.45 * STATION NUMBER.

9. With valve #1 open, press index button on the stepping motor controller and the trolley should be on its way to the preselected station. Look through the sight port on top of the loading chamber to check direction of travel. (Valve #1 must be open.)

10. When the stepping motor has stopped indexing, check the position of the trolley by sighting through the viewer. Because of the possibility of a shift occuring when the trolley passes from one chamber to the other, a small adjustment may be necessary: a maximum shift of one tooth spacing of the rack drive may occur.

B.1.4. Specimen Heating and Temperature Control

1. Check water flow to heater assembly. Energize the air solenoid that is connected to the heater positioning linear feedthrough and observe that the heater assembly has moved into its operationing position directly behind the specimen holder.

2. Turn on the master switch to the heating system. Turn on the plate-grid high- voltage power supply. Select 400N-500 VDC for standard heating cycle or 100N-200 VDC for conditioning electron gun.

3. Slowly turn up the filament voltage on the power supply in the Plexiglas cage to 10 V (over a 1 min time frame).

4. With controller in the manual mode, increase the grid voltage by pressing the "+" pad on the temperature controller. The plate current should

respond slowly to the temperature request if it does not slowly increase the filament voltage to 12 V (MAX!) and wait for a response.

5. Press the "PROCESS VARIABLE/SET POINT" pad so the red light emitting diode (LED) is on next to "SET POINT". Use the "SET PT DECR" and "SET PT INCR" pads to select the desired operating temperature. Press the "PROCESS VARIABLE/SET POINT" pad so the red LED is not on. Once the desired temperature is obtained press the auto pad on controller.

6. Once the temperature has settled down, fine adjustments on the "SET POINT" can be made to bring the temperature to the desired value.

7. When the heater is functioning well it may be possible to decrease the filament voltage back to 10 V. This should prolong the life of the electron gun.

8. If the electron gun heater assembly has been exposed to air, the dispenser cathode surface will have been poisoned, and the above process may take several hours to recondition the cathode surface.

B.1.5. Computer and Electronics Initialization

1. Referring to the blue instrument rack associated with the ion-irradiation chamber in the target room (Room 111, Bldg. 5500), turn on the HP 3853A extender crate.

2. In reference to the current integrators in the same rack, the following settings should be observed for all integrators: 10^{-10} C/pulse, FS current 10^{3}, pos polarity, function switch STANDBY. During start-up, before the run begins, check the BPM integrators when beam is on. If meter is near full scale, decrease the BPM preamp gain on the beamline BPM assembly. This procedure will eliminate meter saturation.

3. With all function switches set to STANDBY, check the top analog meters of all integrators to see that they are centered on a reading of 0. If not, they should be calibrated according to the procedure in the current integrator manual.

4. Switch the function switch to operate on all integrators that will be used for the impending runs; that is, if the 400-keV machine is to be used, then the 400 CUP and 400 BPM integrators should be in the operate mode.

5. In the control room (Room 113, Bldg. 5500), with no floppy diskettes in drive A or B, turn on the computer, printer, and the HP 3852A controller. The computer goes through a self-check procedure and utters a verbal remark that confirms the operation of the talker and the bus extender module. The computer has a lockout program in the autoexec.bat file that is automatically executed when the system is turned on to prevent unauthorized access to the system. Key in the password and hit the enter key.

6. Once the computer is operating it will have automatically loaded Quick Basic into the system. Use the mouse to move the cursor over the area titled FILE and click the left button.

7. Move the mouse cursor over LOAD FILE and press the left button.

8. Move the mouse into the larger central box and press the left button (only one time). This should have darkened an area over one of the file names.

9. Move the shaded area by pressing the up-down left-right keys on the auxiliary numeric key pad until the shaded area is over a file called ZZZ-9224.BAS.

10. With the mouse on "OK," press the left button on the mouse twice to load the file.

11. Place the mouse cursor over the heading titled RUN and press the left button. Position the cursor over the area titled START in this new window and click the left button. This brings up the master touch-screen shell program.

12. A variety of options are presented at this juncture. A list of the functions available is included in Appendix A, along with a description of the actions preformed; however, to star a run, follow the instructions given in 13N-18.

13. Before starting the run, the run input parameters must be established. With the index finger, touch the area of the screen in the upper left-hand corner titled SETUP.

14. On the menu that appears, touch the area titled INITIALIZE run. Sequentially reply with the requested data, taking care to answer questions properly.

15. The input data will be displayed on the screen. Read it carefully to

confirm its accuracy before accepting it. An optional verbal presentation of the data is available if desired.

16. The system will ask for a floppy disk for data storage. This may be a new formatted AT 2S-HD-type diskette or one that was previously used in recording data of the previous runs. The program automatically checks

    the space available for data storage and asks for a new diskette if at least 8 h of run time cannot be stored.

17. The initialized run data is stored on disk, and the system reverts back to the main menu.

18. Touch the header titled START 3 to bring down a menu to start the run. Touching the first entry on the menu, START RUN, will start the run process. Zero values will first appear on the screen while run parameters are being printed.

ACRONYMS

BPM	beam profile monito
HP	Hewlett Packard
ORNL	Oak Ridge National Laboratory
TEM	Transmission electron Microscope
UHV	ultrahigh vacuum
RTD	resistance temperature detector
LED	light emitting diode.

INTERNAL DISTRIBUTION

1-2. Central Research Library
3. Document Reference Section
4-5. Laboratory Records Department
6. Laboratory Records, ORNL RC
7. ORNL Patent Section
8-10. M&C Records Office
11. W. R. Allen
12. J. Bentley
13-17. R. A. Buhl
18. S. W. Cook
19. D. R. Childress
20. R. E. Clausing
21. D. F. Craig
22. L. Heatherly, Jr.
23. R. L. Heestand
24. L. L. Horton
25. M. J. Kania
26. E. A. Kenik
27. E. H. Lee
28. M. B. Lewis
29. C. M. Malone
30-42. L. K. Mansur
43. C. J. McHargue
44. M. K. Miller
45. S. H. Overbury
46. N. H. Packan
47. J. E. Pawel
48. R. E. Pawel
49. D. F. Pedraza
50. P. S. Sklad
51. R. E. Stoller
52. L. J. Turner
53. F. W. Wiffen
54. J. M. Williams
55. D. F. Wilson
56. S. J. Zinkle
57. A. D. Brailsford (Consultant)
58. Y. A. Chang (Consultant)
59. H. W. Foglesong (Consultant)
60. J. J. Hren (Consultant)
61. M. L. Savitz (Consultant)
62. J. B. Wachtman, Jr. (Consultant)

EXTERNAL DISTRIBUTION

63. DOE, OAK RIDGE OPERATIONS, P. O. Box 2001, Oak Ridge, TN 37831-8600

Assistant Manager for Energy Research and Development

64-75. DOE, OFFICE OF SCIENTIFIC AND TECHNICAL INFORMATION, Office of Information Services, P. O. Box 62, Oak Ridge, TN 37831

For distribution by microfiche as shown in DOE/OSTI-4500 Distribution Category UC-25 (Materials).

☆U.S. GOVERNMENT PRINTING OFFICE: 1991-548-134/40008