8/26.90 JS (1)
6

CONTRACTOR REPORT

SAND90-7003
Unlimited Release
UC-706

STL Global Control System, Technical Reference

Roger A. Mitchell
Ktech Corporation
901 Pennsylvania Ave., NE
Albuquerque, NM 87110

Prepared by Sandia National Laboratories Albuquerque, New Mexico 87185 and Livermore, California 94550 for the United States Department of Energy under Contract DE-AC04-76DP00789

Printed January 1990

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

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

NTIS price codes
Printed copy: A03
Microfiche copy: A01

STL GLOBAL CONTROL SYSTEM, TECHNICAL REFERENCE

January 30, 1990

SAND--90-7003

DE90 011258

Roger A. Mitchell
STL Operations Division 9343
Ktech Corporation

ABSTRACT

This report describes the major hardware and software components of the Global Control System (GCS) at the Simulation Technology Laboratory (STL) at Sandia National Laboratories. The GCS presently controls the transfer of oil and water between seven storage tanks and eight accelerator tanks. This report is intended to be a general introduction to the system for engineers and technicians involved in the maintenance and modifications of the system and may also be useful to people interested in designing and constructing a similar system. The scope of this report is to mention the hardware and software systems that are used that are common to the HERMES III and GCS control systems and to describe in detail the hardware and software that is unique to the GCS.

Table of Contents

List of Illustrations

Acknowledgement

I would like to thank all of the people who helped in collecting and reviewing the information contained in this report. Special thanks is expressed to G. A. Zawadzkas who provided encouragement for generating this documentation. I would also like to thank J. M. Farmer, R. A. Miesch, K. A. Mikkelson, and B. A. Woodard for providing and reviewing information in this report and B. A. Macias for editorial assistance and the preparation of this report for publication.

1.0 INTRODUCTION

The Global Control System (GCS) at the Simulation Technology Laboratory (STL) is a computer controlled system designed to control the transfer of Insulating Oil and Deionized Water between storage tanks and accelerator tanks in the STL facility. At the time of the completion of the system, the STL facility had seven storage tanks and eleven accelerator tanks. Because of the number of accelerators in the facility, the system was designed to be able to transfer fluids to and from all of the accelerators simultaneously, when permitted by the building plumbing configuration, and with little operator intervention. The system is also designed to monitor for fluid losses for inactive tanks during operational and nonoperational hours. If a fluid loss is detected, the system closes all valves in the facility, activates an audible alarm and calls facility personnel with a recorded message to indicate a fluid loss.

2.0 BACKGROUND

2.1 History

The Global Control System at STL was started at the beginning of the HERMES III Project, as part of the Control and Monitor Systems portion of the project, with David B. Davis as the Functional Representative. In the project phase, David B. Davis was the Work Package Manager for the Global Control System and designed and supervised the installation of most of the hardware used for the system. At the end of the project phase, in October 1988, the Sandia representation was taken over by Gerald A. Zawadzkas and Kenneth A. Mikkelson, 9343 and the design and coordination of the completion of the project, including the software design and implementation, was taken over by Roger A. Mitchell, 9343/Ktech Corporation. Extensive software support was provided by James D. Creager, EG&G with his work on the HERMES III C/M System and Bonnie A. Woodard, EG&G with her work on the HERMES III C/M System and much of the new software required for the GCS. System installation, maintenance, and documentation was provided by Ellis Dawson, EG&G, Jena M. Farmer and Richard A. Miesch, 9343/Ktech. Facility support for this project was provided by Harold L. Brown, 9343/Ktech, Dusty Ervin, 9343/Technical Support Corporation, and Matthew L. Garcia, 9343/Ktech.

At the end of the HERMES III project, when the system was turned over to Division 9343, a substantial amount of the current hardware had been installed and tested and an incomplete software package to control fluid transfer had been written. This hardware and software were evaluated and found to be inconsistent with other control systems in this area. We desired this system to be consistent with the HERMES III Control Monitor System at STL and the PBFA-II Control Monitor System and determined that it would be faster to implement, and more reliable, if the software was built upon the code foundations and design philosophy of these systems. In addition we would have the many development and system checkout and debugging tools that have been developed with these systems. We decided to build the fluid transfer system using the HERMES III Control Monitor System as a base. The HERMES III Control Monitor System, built upon the PBFA-II Control Monitor System, is a somewhat smaller version of the PBFA-II System and has been proven to be a very reliable control system. We decided to use the current hardware configuration, as it was usable and tested and it would take too much time to change, and modify the software as necessary for hardware differences. Because we used the original hardware, there are some inconsistencies with this design and the design philosophies of the HERMES III and PBFA-II Systems. Since much of the design of this system was based on the HERMES III and PBFA-II Control Systems the reader may want to reference the following documents for in-depth descriptions of the design of these systems: "The Structure and Function of the PBFA-II Control System" (SAND87-1563), "Design Criteria Document For HERMES III Control and Monitor System", "Initial Design Document For HERMES III Control and Monitor System", "Final Design Document For HERMES III Control and Monitor System", and the "Hermes III Remote Operation Control System Operators Manual". The document, "System Manager's Technical Guide for PBFA-II Control/Monitor System" (SAND87-1564) contains documentation on the system software, much of which is common to all three control systems, and procedures for configuration management. The design and implementation of the software for this system was started in October, 1988 and was finished six months later in April, 1989, requiring ten person-months of software effort. With the extensive use of software simulators, less than two weeks of live test time was required for the completion of this system.

2.2 Initial Criteria

For this system, it was desired to have independent control of the oil and water transfer systems with a common operator interface.

For the oil **and** water transfer, the system was to be able to:

- recirculate fluid through filters and the pipes of Building 970 to keep the fluid fresh and clean
- emergency drain an accelerator tank and cancel all other fluid transfers and not allow new fluid transfers until the emergency drain is finished
- emergency stop all current fluid transfers and put all the valves into a safe state
- simultaneous fills **or** drains of accelerator tanks
- ability to create a pending fluid transfer job if the transfer can not be currently started
- fluid loss surveillance with a phone call out for fluid losses.

For water transfer, the system was to:

- allow fills and drains for the PROTO, SPEED, and HYDRAMITE accelerator tanks
- fill and drain the HERMES III Pulse Forming Lines (PFLs).

For oil transfer the system was to allow:

- fills and drains for the PROTO, SPEED, HYDRAMITE, STF, HERMES III, RLA, and EPOCH accelerator tanks
- simultaneous fills **and** drains of accelerator tanks,
- the ability to recirculate oil through filters and the pipes in Building 970A.

The oil transfer system was allowed to have more features because the pipe and pump configurations allowed them. After this system was installed, it was modified to remove the fluid transfers for the HYDRAMITE and SPEED tanks and add fills and drains for the SABRE oil tanks and PFLs.

The operator interface to this system was to be located in a RFI shielded enclosure in Building 970 and provide a color graphical representation of accelerator and storage tank levels, pump and valve statuses, and fluid transfer job status. This display would also provide alarm message reporting and update the required sections of the graphics display when the system status changes. The operators would enter fluid transfer jobs by using buttons on the front of the enclosure. In addition this system would have the ability for point recording and plotting, alarm and action reporting and logging, and orderly shutdown in fault conditions. The finished system met all of these initial criteria.

2.3 Design Philosophy

To quickly implement a reliable and consistent control system, we chose to use the HERMES III Control system as a software base and make necessary software changes for the differences in the system hardware and control tasks. This system relies heavily on vendor-supplied hardware and software products. This system is based on the Hewlett Packard A-series technical computers that use a real-time operating system and have proven their reliability in the PBFA-II and HERMES III systems. The following software products are also heavily used: the RTE-A Real-Time Operating System, the DS/1000 Distributed System product, and the IMAGE/1000 Data Base product.

The flexibility of this system required a software-intensive design. Because there were no demanding timing requirements for this fluid transfer system, all of the control actions were software-based. For software maintainability, all of the software was written in structured FORTRAN 77.

3.0 HARDWARE SYSTEMS

The major hardware components of this system may be broken into the following groups: the Global Control System (GCS) computer and related hardware, the Panel Control System (PCS) computer, the Operator Interface terminal, the Point of Control (POC) circuitry, the latch relay circuitry, the pump control circuitry and pumps, the valves, and the level and position sensors. Figure 1 shows a diagram of how these systems are interconnected. When the system is operated in automatic mode all of these systems are used. The following sections give a description of these systems.

3.1 Computer Systems

For the GCS, two computers and a programmable data acquisition unit are used. The GCS computer is a HP-1000 A-400 computer located in the basement of Building 970A and is responsible for running all of the control and operator interface software. This computer is connected to the PCS computer, a HP-1000 A-600 computer located in the user screenroom of Building 970, via fiber-optic links. These links are terminated with Fibermux FX110 fiber-optic modems and transmit and receive at a speed of 256 kilobaud using the HDLC protocol. The PCS computer is used as a remote data acquisition node for the status of valves in Building 970 and is also responsible for updating the message boards in Building 970. The FCS computer is connected to the HP-3852A Data Acquisition Unit via an IEEE-488 interface bus. The HP-3852A is a programmable acquisition/control unit that , for this application, is used as a dumb I/O device. The PCS computer and the HP-3852 are used as only dumb I/O devices so that the structure of the software is more consistent with the HERMES III System structure and all of the processing is done on one computer.

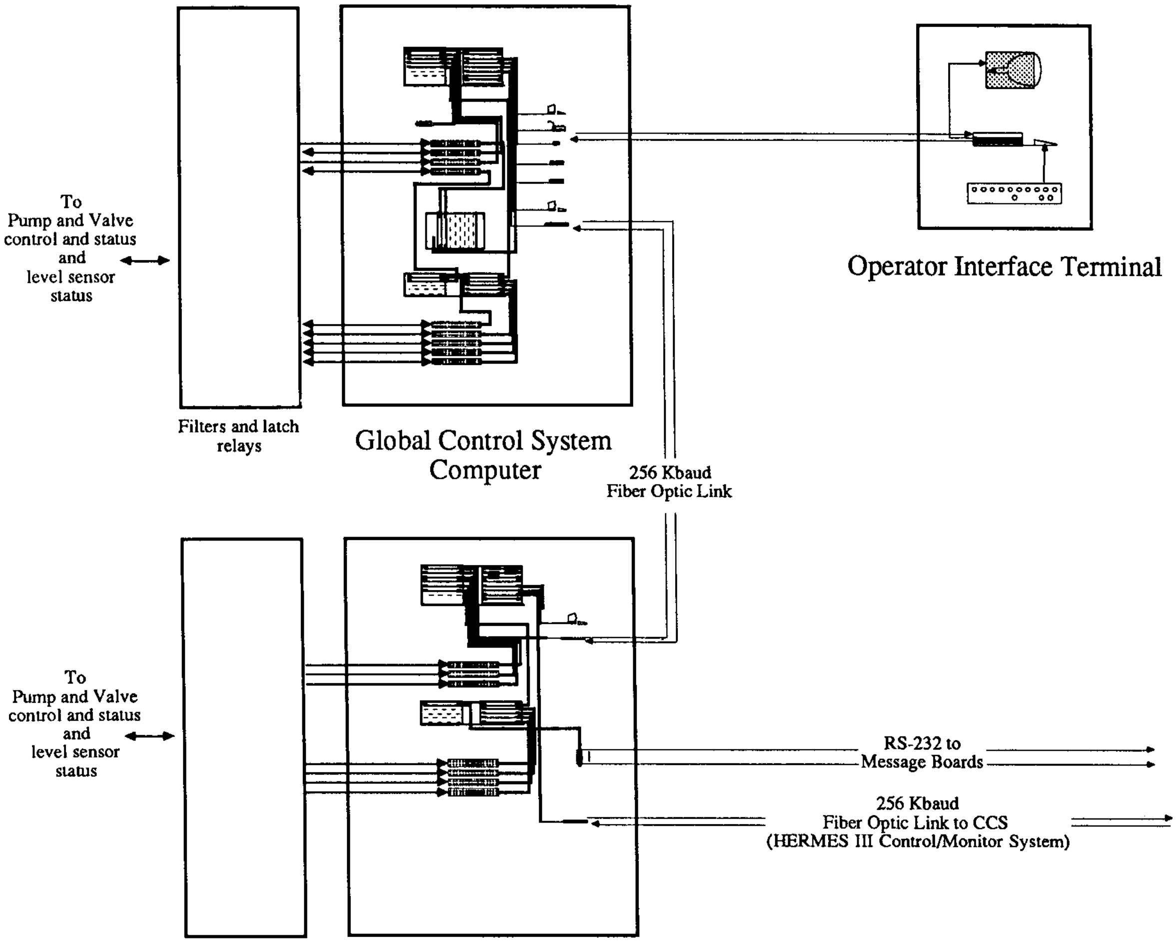


Figure 1. The Global Control System Hardware Components.

3.1.1 The GCS Computer System

The GCS computer performs both of the functions that the host computer system and a subsystem control unit computer in the HERMES III System perform. These are to run the generic control software including the database, message passing, and recording and plotting and to run the specific control action tasks required for the system being controlled. Both of these tasks were put on one computer because the application of fluid transfer was less time critical than accelerator control and the Operator Machine Interface (OMI) for this system was to be much less demanding on the computer than that of the HERMES III System. In addition only one computer had been purchased for this system.

The GCS computer consists of an A-400 computer and cabinet, a HP-1000 L-Series cabinet used for an I/O extender, a HP-3852A Data Acquisition Unit, a HP-7957A 81 megabyte disk, a HP-9133H 20 megabyte disk, a HP-150, an Epson FX-86e printer, a Fibermux FX110 fiber-optic modem, and the required computer and I/O cards. The hardware configuration of this system is illustrated in Figure 2.

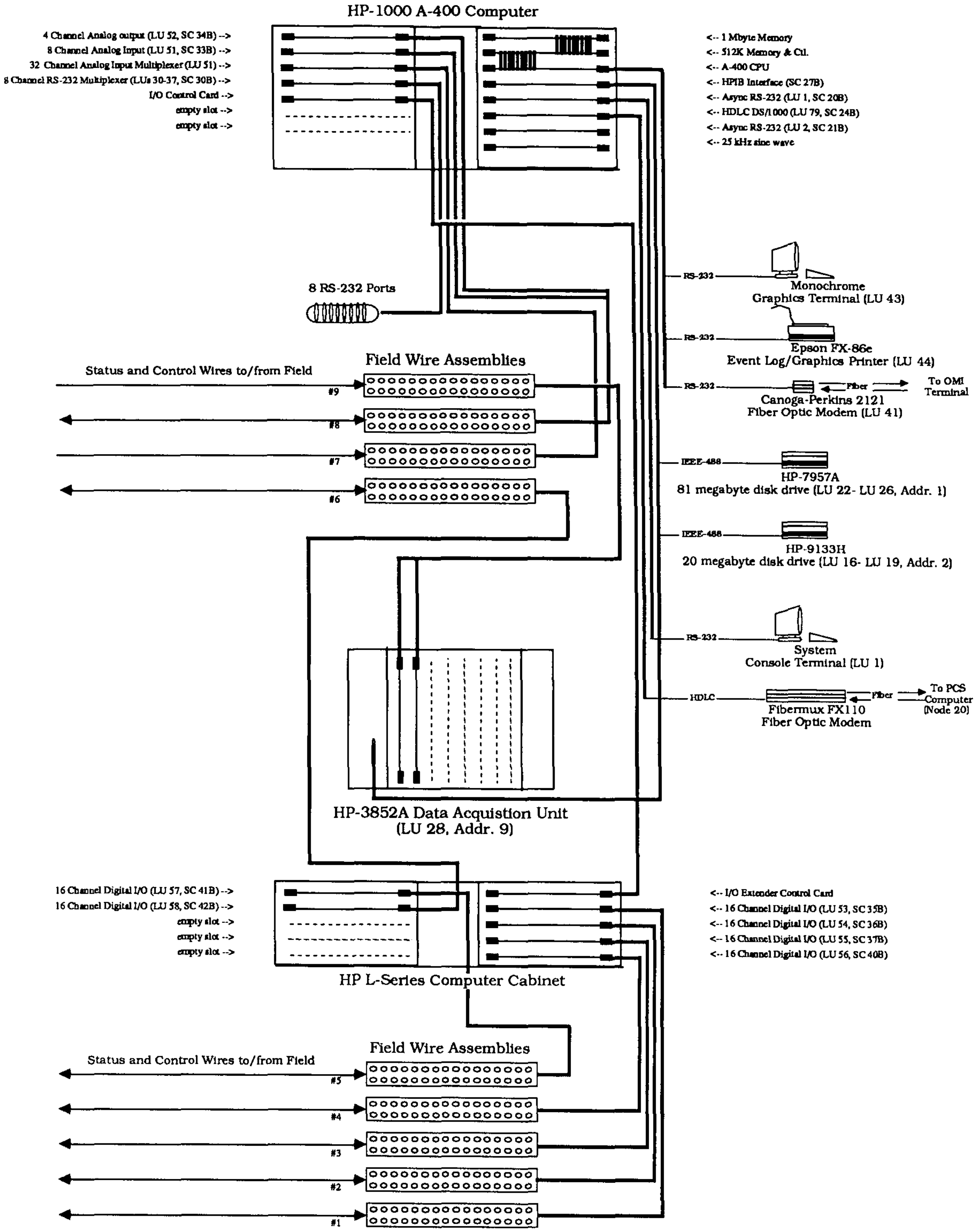


Figure 2. Global Control System Computer Hardware Configuration.

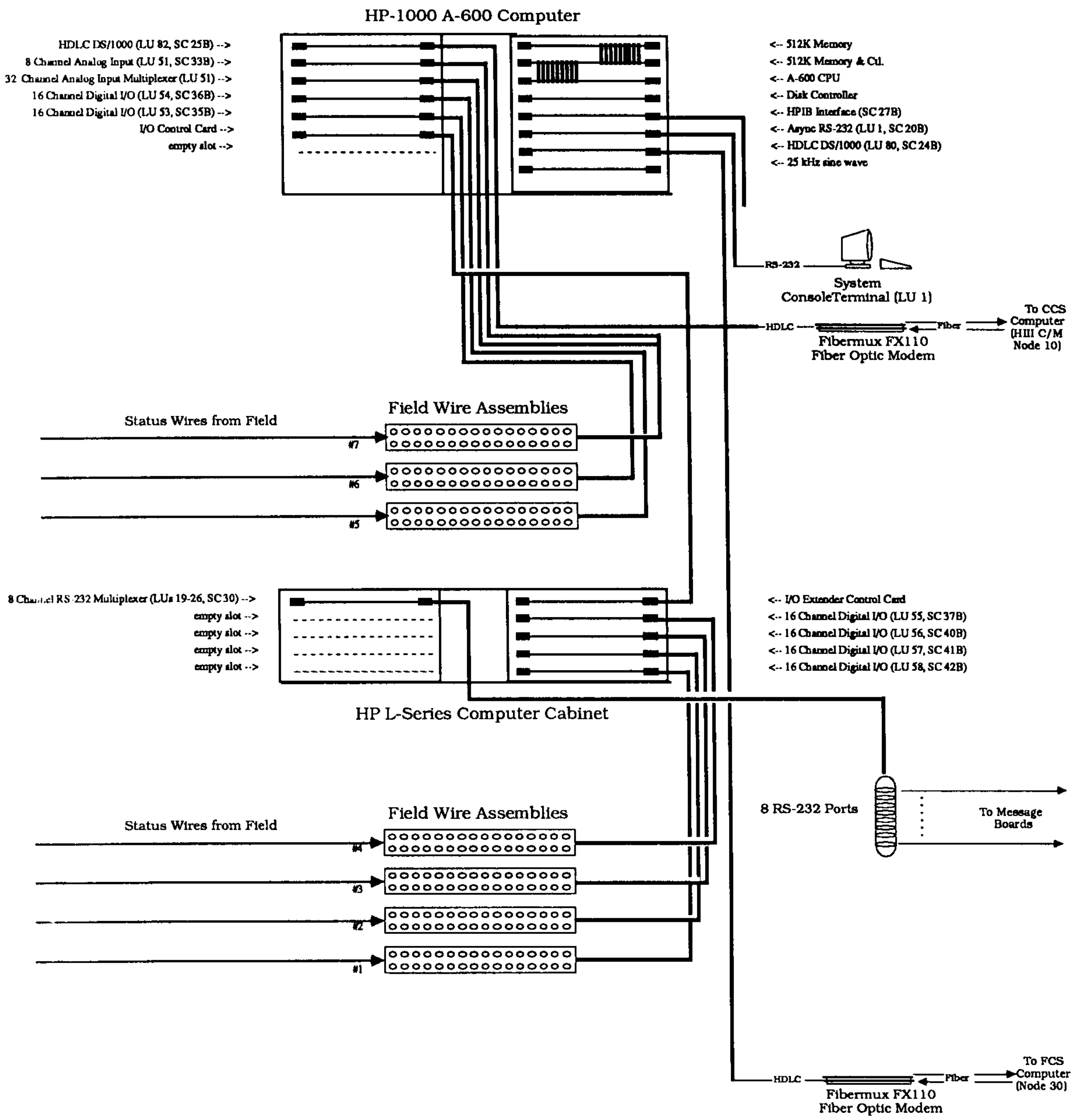


Figure 3. Panel Control System Computer Hardware Configuration.

3.1.2 The PCS Computer System

The PCS computer is used as a remote I/O device to read the status of the valves in Building 970. Theses valve statuses are read through the PCS computer because at the time the system was turned over to Division 9343 the necessary connections from the limit switches to the PCS computer had been made and tested and it would have taken too much time to move them over to the GCS computer. The PCS computer does not run any of the control software used for the system and only runs the DS/1000 network

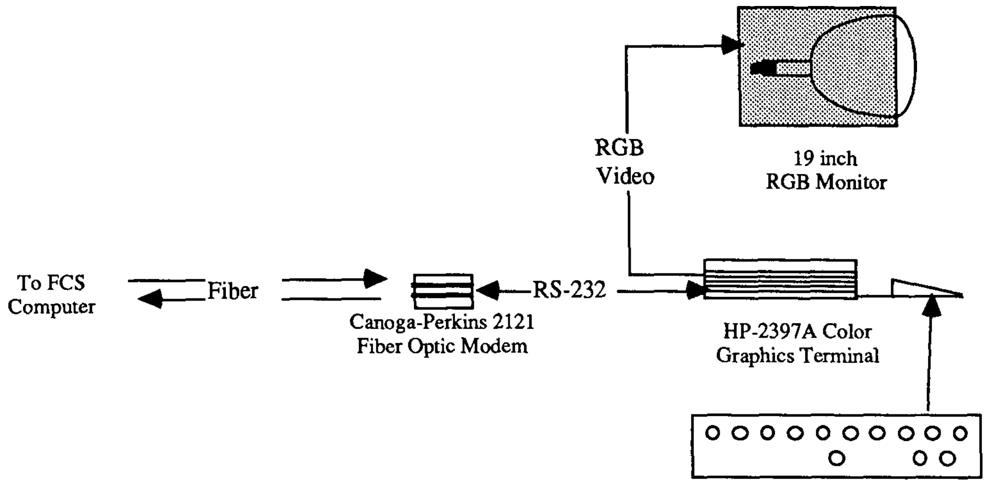


Figure 4. The GCS Operator Interface Terminal.

software required to perform the remote input calls.

The PCS computer is also responsible for updating the message boards in Building 970 with accelerator and general building status. More information about this function of the PCS computer may be found in the "PCS Technical Reference" in the STL

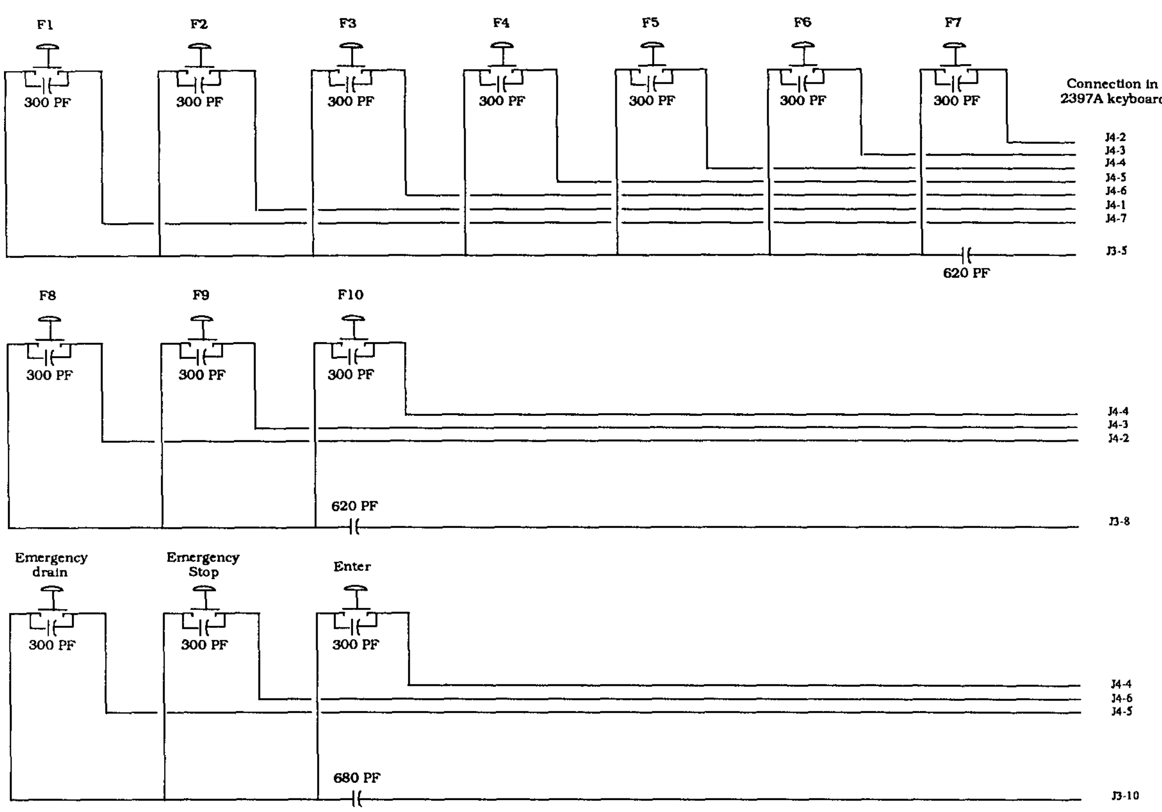


Figure 5. The Circuit Between the Operator Input Buttons and the Keyboard.

Document Library in Building 970A. This document describes that the program ALLMN reads the status of the valves in Building 970 and sends reports to the program RMTMN which sends reports to the GCS computer. This function of these programs is not used and has been replaced by the remote I/O calls in the oil and water control programs running on the GCS computer to produce a system that is more consistent with the HERMES III and PBFA-II control systems and to isolate all of the control software to one computer.

The PCS computer consists of an A-600 computer and cabinet, a HP-1000 L-Series cabinet used for an I/O extender, a 15 megabyte internal disk, a HP-2392A terminal, two Fibermux FX110 fiber-optic modems, and the required computer and I/O cards. The hardware configuration of this system is illustrated in Figure 3.

3.2 The Operator Interface Terminal

The operator interface to this system is a HP-2397A Color Graphics Terminal located in a RFI shielded enclosure on the Hi-Bay floor in Building 970. This terminal is connected to the GCS computer via a fiber optic link and two Canoga-Perkins 2121 RS-232 fiber optic modems and connected to a 19 inch color monitor via three RGB cables. Thirteen operator input buttons are located on the outside of the enclosure and are connected in a capacitive circuit to the keyboard of the 2397A. Figure 4 shows the configuration of the Operator Interface Terminal and Figure 5 shows the capacitive circuit used to connect the operator input buttons to the keyboard. The buttons F1 through F10 correspond to the keys 1 through 0 on the keyboard, the Emergency Drain button corresponds to the minus key, the Emergency Stop button corresponds to the equal key, and the Enter button corresponds to the Return key. The capacitive circuit is used to simulate the touching of a key on the terminal keyboard (a change in capacitance for the circuit). The different values of the capacitors on the return lines (620 pF and 680 pF) were required because of the different capacitances of the switches and wires in the circuit.

3.3 The Fluid and Pump Control Panels

The GCS computer interfaces to control the valves and pumps through the Fluid Control Panels (FCPs) and the Pump Control Panel. The schematics and wiring diagrams for these panels are extensively documented in the STL Facility Interfaces Documentation (STL FID) in the STL Document Library in Building 970A. The STL FID has been updated with changes that have been made in the circuits and wiring lists so the reader should consult with STL facility operations personnel when using this documentation. In this paper, drawings referenced are in the STL FID and changes in the original documentation are noted.

3.3.1 The Fluid Control Panels

There are two Fluid Control Panels for the Global Control System, one is located in the basement of Building 970A and the second is located in the Hi-Bay in Building 970. The Fluid Control Panel in Building 970A contains: point of control selection circuitry, point of control selection display, latch relay circuitry, valve control circuitry and switches for all of the valves used in the system, and valve position indicators for all of the valves. The Fluid Control Panel in Building 970 contains: point of control selection display,latch relay circuitry, valve control circuitry and switches for the valves in

Building 970, and valve position indicators for the valves in Building 970. The schematic for the 970A FCP is in the STL FID drawing CK-S83830 and the schematic for the 970 FCP is in the STL FID drawing CK-S83810. The only changes made to these drawings is in the point of control circuitry in the drawing of the 970A FCP.

The point of control circuitry consists of switches and relays in the 970A FCP. There are four possible positions for this switch: 970 control, 970A control, safe, and remote control. In the 970 position only the switches on the 970 FCP may be used, in the 970A position only the switches on the 970A FCP may be used, in the safe position all of the switches are inactivated as well as computer control, and in the remote position only the computer may control pumps and valves. When the switch is put into the remote position, the computer reads a status point and sets a remote accepted control output if it is in a state that allows computer control. When this output point is set the point of control circuitry allows the computer to have control. The point of control selection is displayed with lights on the 970A FCP and the 970 FCP. If the point of control switch is moved when the computer or the local control switches are in the process of changing (i.e. an output point is set and not cleared or a switch is pressed and not released) the valve position will change as if the valve control switch was released or the output point cleared.

The valves in the system are opened and closed by toggling the valve control switches when the system is in local mode or by setting and then releasing an output control when the system is in remote mode. If the valve was closed it will open and if it was open it will close. The latch relays are used as a T-Flip Flop to maintain the current control voltage of 115 VAC to the valve and open or close the valve when the relay is toggled. The schematic of the latch relays is shown in the STL FID drawing CK-570847. When the 115 VAC voltage is sent out to the valves, a pneumatic solenoid is activated and the valve opens, when the voltage is removed, the air pressure is removed and the valves close. The system is designed so that if there is any power loss or air pressure loss the valves will close.

The switches to open and close valves in local mode on the FCPs are arranged on the panels to show their positions in the buildings and the flow paths between them. These switches have red and green lights in them to show the status of the valves, open valves are indicated with a red light and closed valves with a green light. The valve statuses are also sent to the GCS computer from the 970A FCP for the valves in Building 970A and are read by the computer through channels on digital input cards. The 970A FCP has switches to control and show status for all of the valves in Buildings 970A and 970 and is illustrated in the STL FID drawing S97411. The 970 FCP has switches to control and show status for only the valves in Building 970 and is illustrated in the STL FID drawing S83625. The statuses of the valves in Building 970 are sent to the PCS computer from the 970 FCP and are read through digital input channels in the PCS computer by way of remote input calls from the GCS computer.

3.3.2 <u>The Pump Control Panel</u>

The Pump Control Panel is located in the basement of Building 970A. This panel contains controls for local mode control of the oil and water transfer pumps as well as display of pump on status, pump RPM status, pressure status, and panel active status. This panel also contains the circuitry for computer control of the pumps in remote mode. The schematic for the Pump Control Panel is in STL FID drawing S85926 and has been updated for changes including allowing for separate control of both water pumps.

The local mode control of the pumps is by way of pump power on switches and RPM set point knobs. In remote mode the computer controls the pumps by first strobing the pumps power on output and then setting an analog voltage to control the pumps RPM. The pump on status is displayed on the Pump Control Panel by lighting up the pump power switches when the pumps power is on, this information is also sent to the GCS computer and the computer reads it through channels on a digital input card. The pumps RPM and pressure statuses are displayed on meters on the Pump Control Panel and are also sent to the GCS computer to be read through analog input channels. Pump Control Panel active status is also displayed on the panel and sent to the GCS computer. This status is on when the system is not in the safe mode. The front of the Pump Control Panel showing the local control switches and status display is illustrated in STL FID drawing S85927. This drawing has been modified to show the controls for separate control of the water pumps.

3.4 The Valves

The valves used for GCS are pneumatical actuated valves and are documented in the "STL Process Oil Manual" and the "STL Process Water Manual". The locations of these valves are illustrated in the set of drawings "STL Building 970 Phase 2". All of these documents may be found in the STL Document Library in Building 970A.

3.5 The Level Sensors

The level sensors used for GCS are of two basic types, analog level sensors and digital level sensors.

3.5.1 The Analog Level Sensors

The analog level sensors are differential pressure transmitters that send a current of between 4 mA and 20 mA to the computer, based on the input pressure to the transmitter. This current is converted to 1 V to 5 V by precision resistors and read on channels on an analog input card in the GCS computer. The analog level sensors used are Rosemount Model 1151DP differential pressure transmitters. For the oil and water storage tanks, these pressure transmitters are plumbed into the system with the fluid on one side and air on the other side and measure the difference between the fluid pressure and the air pressure. For the accelerator tanks, the pressure transmitters are used in a bubbler type level measuring system. In this system pressurized air is piped to the bottom of the accelerator tanks and allowed to flow through the fluid to the top of the tank. The differential pressure transmitted measures what pressure of air is required for the air to flow up through the fluid. These sensors are located in the thirteen foot deep trench in Building 970 under each accelerator. The differential pressure transmitters and the bubbler measuring system are documented in the manual from Southwest Controls, "Sandia Bubblers Level Measuring System Job #A9017" located in the STL Document Library. Appendix A gives a description of the procedure used to calibrate these sensors.

3.5.2 The Digital Level Sensors

There are two types of digital level sensors used to tell if the fluid is above a certain point in a tank. The type used for the oil storage tanks are "Flowtect Flowswitch Model L6EPB5535" made by W. E. Anderson Inc. The type used for the water storage tanks are "SOR Model 201-24-1X". These sensors provide a switch closure when the fluid is above the center of the sensor. A voltage of 24 VDC is passed through the switches in the level sensors and read through channels on a digital input card in the GCS computer. The locations of these sensors are illustrated in the STL FID drawing S83856. For the storage tanks, there are sensors to measure if the tank is empty or if it is overfilled. The software checks to see that the empty sensor is on (not empty) and that the overfill sensor is off at all times. The type of sensors used for the oil and water accelerator tanks are "Linc 282 Series Pneumatic Level Control" sensors. These sensors provide an air pressure change when the fluid is above the center of the sensor. This air pressure change is converted to a switch closure by air switches and relays located in the cabinet in the thirteen foot deep trench that contains the differential pressure sensor for the accelerator. This switch closure is read in the same manner as the storage tank digital sensors by the GCS computer. The locations of these sensors are illustrated in the STL FID drawing S83840.

For the accelerator tanks, there are sensors to measure if the tank is empty, full or if it is overfilled. The software uses the empty sensors to tell that the accelerator tanks are empty when draining and the full sensors to tell that the accelerator tanks are full when filling. In addition to the digital sensors, the software checks the fluid level using the analog sensors to tell if a tank is full or empty when filling or draining to provide additional information and a backup should a digital sensor fail. If an overfill sensor should come on, the software stops all jobs and goes into a fault state. In addition, if an overfill sensor comes on, a relay is activated that will close the fill and drain valves for that accelerator tank. The schematic for this circuit is shown in the STL FID drawings S84397, S83843, and S84399.

4.0 SOFTWARE SYSTEMS

As mentioned earlier, the base for the GCS software came from the HERMES-III Control Monitor System software. The system uses the following Hewlett Packard software packages: DS/1000 Distributed System used to handle the network communications between computers, IMAGE/1000 Data Base used for the disc based database, and the RTE-A Real-Time Operating System used for system services such as interprocess communication, task synchronization, file and I/O handling, memory management etc. In addition to these software packages the system uses an in house written real-time control package consisting of over 90,000 lines of FORTRAN code. Of this code approximately 30,000 lines were written for GCS and 60,000 lines were transported over from the HERMES-III system. The major changes made in the software for use in GCS were in the input and output routines, the routines to initialize the disc based data base, the operator machine interface, and the actual control software to control the fluid transfer.

4.1 Software Subsystems

This section will describe the major software subsystems of the system, pointing out which subsystems were changed from the HERMES-III system and which subsystems are the same for both

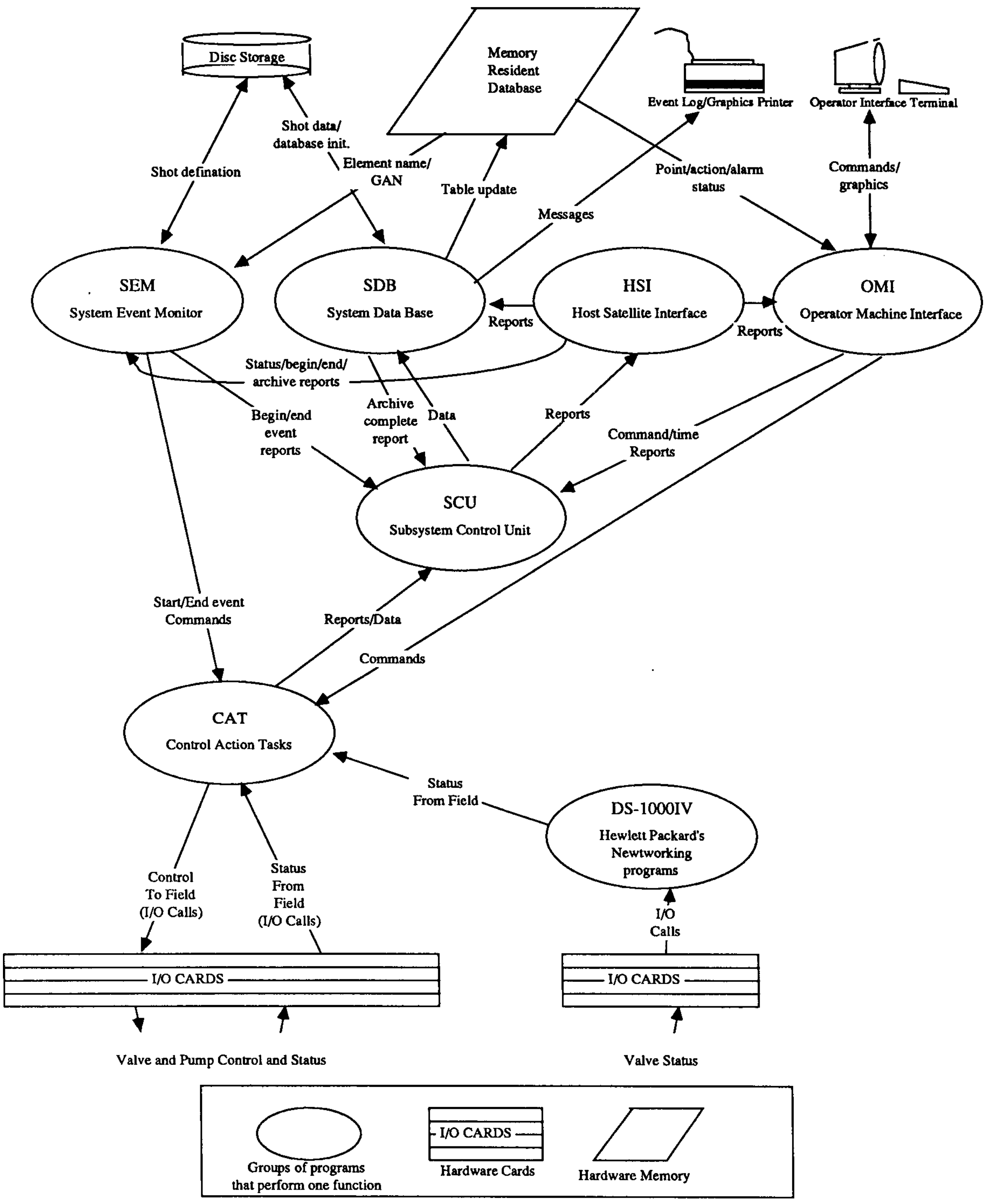


Figure 6. Fluid Control Software Subsystems Data Paths.

systems. The software subsystems that were changed will be covered in more detail than unchanged subsystems. The software system consists of six software subsystems and a set of software tools that are used for development and debugging. These subsystems are: The Control Action Tasks (CAT), The Subsystem Control Unit (SCU),The Host-Satellite Interface (HSI), The System Data Base (SDB), The System Event Monitor (SEM), and the Operator Machine Interface (OMI). The document " System Manager's Technical Guide For PBFAII Control/ Monitor System" describes the software system directory structure, software management, and building the executable control and system programs for the PBFA II system. The structure of the GCS software is very similar to the PBFA II structure so this document serves as an introduction to the software management for GCS. The initial and final design documents for the HERMES III Control/Monitor System give some description of the software design and software subsystems functions used for the control system. The data flow paths between the GCS software subsystems are shown in Figure 6. A complete diagram of all the data flow paths for all programs in the system is shown in Appendix B.

In the following sections it would be helpfull for the reader to have an understanding of some of the topics of the RTE-A Operating System, the DS-1000 Distributed System product and the Image/1000 Data Base such as: Standard I/O, Class I/O, Program Scheduling, Resource Numbers, Shareable Extended Memory, System Services, Network Program to Program Communications, and Database Access Routines. In depth information on these topics may be found in the "RTE-A Programmer's Reference Manual", the "DS/1000-IV Network Manager's Manuals", and the "IMAGE/1000 Data Base Management System Reference Manual".

4.1.1 The Control Action Tasks (CAT)

The CAT subsystem is the group of programs that actually control the pumps and valves for the fluid transfers. The subsystem consists of the following programs: CNFIO, the I/O card configuration program; RCIF/RCIFU the remote CAT interface program; OILCA, the oil transfer control program; OILCA_SIM, the simulator version of the oil transfer control program; OILCS, the oil transfer simulator program; WATCA, the water transfer control program; WATCA_SIM the simulator version of the water transfer control program; and WATCS, the water transfer simulator program.

4.1.1.1 CNFIO

The program CNFIO is scheduled at boot time from the WELCOME1.CMD file and is responsible for setting up the I/O cards on the GCS computer. The cards on the PCS computer are set up at boot time with the program CNFIO that runs on PCS. The CNFIO program resets all of the I/O cards and reports any configuration errors. The program CNFIO does not exist on the HERMES III System, it takes the place of the subroutine CHKCONF called by the program UPSAT that is run when the I/O processors are brought up.

4.1.1.2 RCIF/RCIFU

The program combination of RCIF/RCIFU is used to send commands to the control software and simulators in a debugging and test mode and to modify the model parameters used by the control software. The commands are sent to the control software by a schedule of that program with a buffer that contains the command as well as object and model parameters. The object parameters are used for parameters that may change with each command. The only object parameter that is sent for the fluid transfer system is the command. For the HERMES III Control System, the object parameters are used to send target values to the control software (such as target charge voltage). The model parameters are used for parameters that may be changed, but are not normally changed for each command. Some examples of model parameters used for the Fluid Transfer System are: allowed level tolerance for inactive tanks, pump RPM for recirculation state, normal low and high fill points for accelerator tanks, and allowed delays for valves to open. The model parameters are edited with the touchscreen driven program RCIF and are stored in the files /MACH_ELEMENTS/ SUBSYSTEM_NAME/MMDS.MMDS where SUBSYSTEM_NAME is OIL_TRANSFER or WTR_TRANSFER. Each time a command is sent to a control program, these files are read and the model parameters are sent to the program. RCIF is the only way to modify the model parameters.

The program RCIF can send most commands to the control programs in a test mode and is the only way to send commands to the simulators. Some of the simulator commands are to put the subsystem into local or remote mode and to simulate fault conditions for testing of the control programs. The program RCIF uses the program RCIFU because the size of the data segment for RCIF was too large to fit in one program. The only change from the HERMES III System RCIF to the GCS RCIF was to remove the commands for the subsystems used for HERMES III control and to add the commands for the subsystems used for Fluid Transfer control.

4.1.1.3 OILCA and WATCA

The oil and water control programs are built upon a control program shell that is used for the control programs for the HERMES III Control System with one or more subroutines that are specific for the system being controlled (oil transfer or water transfer for example). The shells of the control programs, for the most part, are unchanged from the HERMES III control shells, but the system specific subroutines used are one of the largest changes that was required for the fluid transfer system. These programs are initially scheduled by the program UPSAT, scheduled at boot time by the file WELCOME1.CMD, and are put on the time list to run periodicity based on a scan time that is set in the file /MACH_ELEMENTS /ELEMENT_MASTER. These programs are also scheduled, with a buffer, when a command is sent to them. In the normal operating mode, commands are sent from the program GCSDM (or GCSCM the color version) based on operator input. In addition, the program SYEMN sends start and end event commands. The programs RCIF and SUPER may also send commands to the control programs and are used only in a debugging and testing mode.

The control programs read and write to the I/O cards with standard I/O calls, send I/O point state data to the program RECOR with a class buffer, and send reports to the program REPOR with a class buffer. The CAT will send out I/O point data for all points that have a 1 in the record field for the point name in the file /GCS_CODE/CAT/SUBSYSTEM_NAME/FPN_LIST_XXXXX.FPN (where XXXXX is OILCA or WATCA for the oil or water system respectively). Reports may be Action Reports, Alarm Reports, Group Reports, or Status Reports. Action reports are used to tell that the control software has reached a certain state or that a fluid transfer job has started or finished. Alarm reports are used to report an alarm message during fault conditions. Group reports are used to tell that the status of an I/O point monitored by the system has changed. Status reports are used for start event and end event messages as well as for stall reports that are sent periodicity, based on a time that is set in the file /MACH_ELEMENTS/ELEMENT_MASTER, to tell that the control programs are still running. The stall reports are monitored by the program STALL. By the use of reports, programs that are interested in a particular type of report may suspend on a class get and be rescheduled when that type of report is sent out. This allows these programs to not have to poll and waste CPU time.

The routines that were changed the most from the HERMES III System were CNTRL_OILCA.FTN for the oil control and CNTRL_WATCA.FTN for the water control and the specific routines that these routines call. These routines are very specific for the subsystem being controlled but are similar to each other. CNTRL_WATCA may be thought of as a subset of the routine CNTRL_OILCA because the water control program does not perform as many functions as the oil control program. A summary of the command options available for the oil and water control and some considerations and error messages for these control programs is shown in Appendix C. The pump profiles that show what pump RPMs are used to fill and drain each accelerator tank are shown in Appendix D.

Another major change between the Fluid Transfer control programs and the HERMES III control programs are the routines BUILD_IN_XXXXX.FTN, GET_STATE_XXXXX.FTN, and SET_CTL_XXXXX.FTN. These routines required changing because the I/O cards used for the Fluid Transfer System are different than the cards used for the HERMES III System. The routine BUILD_IN_XXXXX fills in buffers with information telling the program what I/O cards must be read for that subsystem. This routine is called when the control programs first start up so that the control programs will not have to determine this in real time. The routine GET_STATE_XXXXX is the routine that reads the I/O cards and the routine SET_CTL_XXXXX sets the I/O points in an exception mode.

4.1.1.4 OILCA_SIM and WATCA_SIM

The programs OILCA_SIM and WATCA_SIM are the simulator versions of OILCA and WATCA. The difference between the simulator versions and the real versions are that the simulator versions do not call the BUILD_IN_XXXXX routine and call GET_SIM_XXXXX instead of GET_STATE_XXXXX and SET_SIM_XXXXX instead of SET_CTL_XXXXX. BUILD_IN_XXXXX is not called because the simulator versions do not perform actual point I/O. The routine GET_SIM_XXXXX schedules the simulator (OILCS for oil and WATCS for water) and passes it a buffer telling it to return point status. The routine SET_SIM_XXXXX schedules the simulator and passes it a buffer telling it to set points. These two routines have the same entry points as the real control routines and all other parts of the simulator control programs are the same as the real control programs. The program UPSAT determines if the real control programs or the simulators will be scheduled by a passed in parameter and if the program can be found in the system (RP'ed).

4.1.1.5 OILCS and WATCS

The programs OILCS and WATCS are the simulators for oil and water transfer. The simulators are used primarily for testing of the control software and operator interface screen. The simulators are scheduled by either the GET_SIM_XXXXX or the SET_SIM_XXXXX call. If the call was the GET_SIM_XXXXX the simulator processes through its loop and sets points based on control points set, system time, fault commands, etc. and passes the points back to the control program. If the call was the SET_SIM_XXXXX the simulator sets the correct output points in its internal buffers and takes action to set any points sent back to the control program based on these output points. The simulators may receive commands from RCIF telling them to change modes, simulate faults, etc. The simulators use the same shell as the simulators used for the HERMES III System with changes to the specific routines CNTRL_OILCS.FTN and CNTRL_WATCS.FTN.

4.1.2 The Subsystem Control Unit (SCU)

The SCU programs are responsible for starting the CAT programs, starting the DS/1000 network, passing reports from the CAT programs to the HSI subsystem, and passing data from the CAT programs to the SDB subsystem. In the HERMES III System, the CAT and SCU programs run on a satellite I/O processor nodes and the SCU subsystem is used to pass data and reports to the host processor. For GCS both the I/O processor and host functions were placed on the same computer so some of the paths of passing data and reports could have been combined. The programs ARCHV and ARC30 could have been combined and the programs REPOR and ASR30 could have been combined. To keep this system consistent with the HERMES III System all of the data paths were kept the same and the DS/1000 calls are still used but the destination node is the same as the source node. The SCU subsystem consists of the following programs: ARCHV, the data archiver master; NOD30 the DS/1000 initialization program; RECOR, the data recorder; REPOR, the message reporter; TMSYN, a system clock setting program; and UPSAT, the CAT subsystem starting program.

4.1.2.1 ARCHV

The program ARCHV reads data from the Shareable Extended Memory Area (SHEMA) partition "RECORD" , packages it up, and sends it to the data archiver slave program, ARC30, by using a DS/1000 Program To Program (PTOP) write. ARCHV is scheduled by the program UPSAT at boot time and runs in an endless loop with a process priority of 30000. Because of the priority of this program it only uses free CPU cycles. The size of the SHEMA partition may be increased so that, on a busy system, all of the data from an event (in this case the time from when the recorders are turned on to when they are turned off) may be stored in memory and transferred by ARCHV when the event is over. When ARCHV runs, it picks up all data from the SHEMA partition and sends a data buffer to ARC30 when it has 4096 words of data. At the end of the event , ARCHV sends the final buffer of data. In the normal operating mode of GCS, no event is active and the data recorders are not turned on. For this system the recorders are normally used only for testing, the HERMES III System has the recorders on for all automatically controlled shots. Because ARCHV is the master program and ARC30 is the slave program, ARCHV must be scheduled before ARC30. The GCS version of ARCHV is the same as the HERMES III version.

4.1.2.2 NOD30

The program NOD30 is scheduled at boot time from the WELCOME1.CMD file and starts up the DS/1000 link. This program is basically a data statement and calls a DS/1000 routine to set up the link. This version is slightly different from the HERMES III versions (NOD01, NOD02, HERMES_NOD10) because the links are set up differently.

4.1.2.3 RECOR

The program RECOR is scheduled by UPSAT and waits on a class buffer of data from the CAT control programs. The CAT programs receive RECOR's class number when they are scheduled. When a class buffer is received, RECOR puts the data into the SHEMA partition "RECORD" for ARCHV. The GCS version of RECOR is the same as the HERMES III version.

4.1.2.4 REPOR

The program REPOR is scheduled at boot time from the WELCOME1.CMD file and then waits for a class buffer from one of the following programs: OILCA, WATCA, ARC30, SET_KNOBS, GCSDM, or WATCH. The CAT programs receive REPOR's class number when they are scheduled, the other programs get REPOR's class number by calling the routine NAME_TO_CLASSES.FTN. When REPOR receives a class buffer it packages up the reports and sends them to ASR30, the report slave program, with a PTOP write. Because REPOR is the master program and ASR30 is the slave program, REPOR must be scheduled before ASR30. The GCS version of REPOR is the same as the HERMES III version.

4.1.2.5 TMSYN

The program TMSYN is scheduled by the program SYEMN, the system event monitor, when a new event is started, to set the system clock. This program was designed for the HERMES III System so that all of the computers would have the same time. For the GCS it serves no real purpose as it sets the system time to the current system time (only one computer used). It is put in for compatibility and is the same as the HERMES III Version.

4.1.2.6 UPSAT

The program UPSAT is scheduled at boot time from the file WELCOME1.CMD, to start the following programs: ARCHV, RECOR, REPOR, OILCA, and WATCA. UPSAT allocates class numbers for RECOR and REPOR and allocates resource numbers for ARCHV, RECOR, and REPOR. When UPSAT is finished running, it suspends saving resources so that these numbers stay allocated. UPSAT schedules the simulators if they RP'ed and if the mode passed into UPSAT was up_sim. UPSAT schedules the control programs based on the scan times in the file /MACH_ELEMENTS/ELEMENT_MASTER and reads the files /MACH_ELEMENTS/SUBSYSTEM_NAME/CCDS.CCDS for information on what physical addresses are assigned to the I/O points that the CAT is to use and what groups of points the CAT is to report and passes this information on to the CAT. The GCS version of UPSAT is the same as the HERMES III version except that the HERMES III version also starts up the DS/1000 link, checks the I/O cards and schedules DIGEM. The Fluid Transfer version zeros out the DO_BUFFER stored in the SHEMA partition "HGTAB" when it starts. The DO_BUFFER is a buffer of all of the output points that are currently set by the CATs. It must be shared between the two CATs so that if a point is set by one CAT the other CAT will not set that point to zero if it sets another point on the same I/O card (as the outputs are set one word at a time).

4.1.3 The Host Satellite Interface (HSI)

The HSI programs are used to route reports that are sent to REPOR to other programs, called monitors, that are interested in that type of report, or mapped into that report type. By using this type of program to program communication the same report may be sent to many programs that are mapped into that report type and a program may receive many different reports from many different sources. The programs that are mapped into reports are waiting on a class buffer and are rescheduled when the buffer is sent, to not use any CPU time until the buffer is sent. The HSI subsystem consists of the programs: ASR30, the reporter slave program; HSIMA, the HSI start up program; INAML, the HSI initialization program; MAMLT, the program to maintain the Active Monitor List; and RAMRP, the report routing program, as well as two utility programs: DUMP_HSI, to show what programs are mapped into which reports, and CLEAN_UP_HSI, to remove unused class numbers from the HSI tables. Programs map in for reports by putting a class number into the SHEMA table "HSIEMA".

4.1.3.1 ASR30

ASR30 is the report slave program that receives PTOP buffers from REPOR and sends them in a class buffer to RAMRP. Because ASR30 is a slave program it is scheduled by REPOR. The GCS version of ASR30 is the same as the HERMES III version (ASR10).

4.1.3.2 HSIMA

The program HSIMA is scheduled at boot time by the file WELCOME1.CMD and schedules the programs INAML and RAMRP. HSIMA allocates the class number used for the communication between ASR30 and RAMRP and puts it in the SHEMA partition "HSIEMA". When HSIMA is finished, it suspends saving resources so that this class number will stay allocated. The GCS version of HSIMA is the same as the HERMES III version.

4.1.3.3 INAML

INAML is scheduled by HSIMA and puts zeros in all of the locations of the Active Monitor List in "HSIEMA". The Active Monitor List is the list of all of the class numbers for the programs that are mapped into reports. The GCS version of INAML is the same as the HERMES III version .

4.1.3.4 MAMLT

MAMLT is scheduled, with a buffer, by each program that wants to map into reports. MAMLT puts the scheduling program's class number into the locations in the Active Monitor List for the type of reports it asks to be mapped into. MAMLT is also used to remove class numbers from the Active Monitor List. The GCS version of MAMLT is the same as the HERMES III version.

4.1.3.5 RAMRP

The program RAMRP is used to route reports from ASR30 to the monitor programs that are mapped into that report type. When RAMRP receives a buffer from ASR30, it decodes the type of report and then searches the Active Monitor List for class numbers of programs that want to receive this report. The Active Monitor List is an array, indexed by report types, of class numbers. RAMRP then sends, with a class write, the report to all of the mapped in monitors. The GCS version of RAMRP is the same as the HERMES III version .

4.1.3.6 CLEAN_UP_HSI

The program CLEAN_UP_HSI is used to clean out unused class numbers from the Active Monitor List. This program is most useful in a testing mode when programs are started and stopped often and do not correctly remove their class numbers from the list. The GCS version of CLEAN_UP_HSI is the same as the HERMES III version .

4.1.3.7 DUMP_HSI

The program DUMP_HSI can be used to produce a printout of what programs are mapped into which type of reports. It is used in a testing mode. The GCS version of DUMP_HSI is the same as the HERMES III version.

4.1.4 The System Data Base (SDB)

The SDB subsystem of programs perform a wide range of jobs related to creating and maintaining the disc and memory based databases and storing and recalling point state data. The routines that initialize the disc based database are: CFINT, the command file interpreter for PMM; GRBLD, the group building program; and PMM, the point modify and maintenance utility. The disc based IMAGE/1000 database is originally created by running the IMAGE/1000 program DBDS on the file /IMAGE2/GCS_SCHEMA, more information on this may be found in the "IMAGE/1000 Data Base Management System Reference Manual". All of the files to setup and manage the IMAGE database are located in the directory /IMAGE2. The IMAGE database itself is located on the FMGR Logical Unit (LU) 26. The program FPNCO is used to read the disc based database and produce files that are used by the CAT programs.

The programs that initialize the memory resident database are: RMRGA, to restore memory resident group attributes; RMRLA, to restore memory resident alarm attributes; and RMRPA, to restore memory resident point attributes. The programs that update the memory resident database are: ACMON, the action report monitor; ALMON, the alarm report monitor; PVDMB, the point value database monitor; STALL, the stall report monitor; and STMON, the status report monitor.

The programs that store the CAT point state data in disc are ARC30, the slave archiver; and LNKER, the linker of detail data sets. The programs that build files that are used to index the stored data are BUAMF, to build the archiver master file; and SUARF to set up the archiver files. The program RECAL is used to recall and plot data. The SDB also contains the program SUPER that can be used to look at point state data in real time and send commands to the CATs in a testing mode.

4.1.4.1 CFINT

When PMM is run in command file mode, the program CFINT is scheduled to add, modify, or delete I/O points from the database. CFINT uses the .DEF files in the directory /GCS_CONFIG to tell what points to add, modify or delete. For the current system the files GCS_POINTS_DEF.DEF and PCS_POINTS_DEF.DEF are used to add all of the points for the GCS and PCS computers and the files GCS_POINTS_DEL.DEF and PCS_POINTS_DEL.DEF are used to delete all of the points used for the system. CFINT and PMM modify the PTATTR section ot the IMAGE database. More information about CFINT and PMM may be found in the document "PM&M User's Manual" located in the STL Document Library. The GCS version of CFINT is the same as the HERMES III version.

4.1.4.2 GRBLD

The program GRBLD is used to create and delete a group of points. A group of points is 50 or less points grouped together so that a program may map into reports for that group of points and receive reports only when points that are in that group change. An I/O point may be in as many as five or as few as zero groups. A point is put into groups by putting the group numbers of the desired groups into the point's groups field in the IMAGE database. GRBLD reads the .GRP files in the directory /GCS_CONFIG to create and delete groups. The information about what points are in which groups is read from the IMAGE database by FPNCO and put in the CCDS.CCDS files for the CATs to use when they start up. The CAT will then send out a group report for a group if any point that the cat is monitoring (has a 1 in the report field for the point name in the file /GCS_CODE/CAT/SUBSYSTEM_NAME/FPN_LIST_XXXXX.FPN) in that group changes. The GCS version of GRBLD is the same as the HERMES III version.

4.1.4.3 PMM

The program PMM is used to add, modify, or delete I/O points from the database. Normally PMM is used in the command file mode and the database modification is done by CFINT. PMM may also be used in a forms mode to add, modify, or delete I/O points from the database. The GCS version of PMM is the same as the HERMES III version.

4.1.4.4 FPNCO

The program FPNCO is used to extract information from the IMAGE database and put it into files that the CATs use, it must be run whenever points or groups are modified. In the directories /GCS_CODE/CAT/SUBSYSTEM_NAME/ FPNCO reads the file FPN_LIST_XXXXX.FPN and creates the files I_SIZE_TBL_XXXXX.FTN and I_PNT_TAGS_XXXXX.FTN. The file I_SIZE_TBL_XXXXX contains information telling the CAT how many of each type of I/O point it is reading and setting and what are the size of the buffers the CAT reads data from the card into and writes data from. The major buffers that the CAT uses are STATE_VECTOR for the data read and CONTROL_VECTOR for the points that are to be set. The file I_PNT_TAGS_XXXXX contains labels that point into the STATE_VECTOR and CONTROL_VECTOR buffers. The labels or tags are used so that in the CAT the points may be addressed by STATE_VECTOR(tag) or CONTROL_VECTOR(tag) where the tag is the point name.

FPNCO creates, in the directories /GCS_CODE/CAT/SUBSYSTEM_NAME/XXXXX/, the files I_REC_PTS_XXXXX.FTN and I_REP_PTS_XXXXX.FTN to tell the CAT what points in FPN_LIST_XXXXX.FPN it should record and report respectively. These are created based on the ones or zeros after the point name in the file FPN_LIST _XXXXX.FPN. A CAT may monitor and set points but not record or send reports for them in this way.

FPNCO also creates the files /MACH_ELEMENTS/ SUBSYSTEM_NAME/CCDS.CCDS. These files are read by UPSAT and sent to the CATs when they start up. These files contain information about the physical addresses of I/O points, engineering units for points, coefficients to determine the value in engineering units of a analog point from the millivolt value read, and what groups a point belongs to. The differences between the GCS version of FPNCO and the HERMES III version is that the GCS version writes the node number that a point is on into the CCDS.CCDS file. GCS allows a CAT to use points that are on different computers (nodes) the HERMES III System does not.

4.1.4.5 Other Database Modifications for GCS

The HERMES III System has a program, IOCNF, that is run on the satellite I/O processors to initialize the node (PTNOD#) and address (PTADDR) sections of the IMAGE database. This program must be run before points or groups are added to the database. IOCNF determines what type of cards are in the HP-2250 I/O controller and adds this information into the database. For GCS this program was replaced by a set of QUERY command files that initialize these sections of the database. This method was faster to implement then to write a new IOCNF program but requires a person to edit these command files if the card configurations are changed. The files /IMAGE2/SETUP_NOD20.QUE and /IMAGE2/SETUP_NOD30.QUE initialize the PTNOD# section of the database with information on the types of cards in each slot and the number of channels on the cards. The files /IMAGE2/SETUP_ADD20.QUE and /IMAGE2/SETUP_ADD30.QUE initialize the PTADDR section of the database with information on the addresses for each point on each card.

4.1.4.6 RMRGA

The program RMRGA is scheduled at boot time from the WELCOME1.CMD file and reads information from the IMAGE database to fill the GROUP_TABLES in the memory resident database SHEMA partition "HGTAB". This information includes the group descriptor, the group number, the group size, the group point numbers, and the resource number to access these tables. The GCS version of RMRGA is the same as the HERMES III version.

4.1.4.7 RMRLA

The program RMRLA is scheduled at boot time from the WELCOME1.CMD file and reads information from the files /GCS_CODE/SDB/RMRLA/I_ALARMS_CONFIG.FTN and /GCS_CODE/SDB/RMRLA/I_ALARMS_LO_DBAS.FTN (actually read at compile time) to fill the ALARM_TABLES in the memory resident database SHEMA partition "HGTAB". This information includes the alarm active and acknowledged flags, the alarm activated and released times, the alarm category and severity, the alarm message, and the resource number to access these tables. The GCS version of RMRLA is the same as the HERMES III version.

4.1.4.8 RMRPA

The program RMRPA is scheduled at boot time from the WELCOME1.CMD file and reads information from the IMAGE database to fill the POINT_TABLES in the memory resident database SHEMA partition "HGTAB". This information includes the point descriptor, the point Facility Point Name (FPN), the coefficients to determine the value in engineering units of an analog point from the millivolt value read, the point gain value, the points current value, the allowed limits for a point, the feedback analog input point for analog outputs, the time of the last point update, and the total number of points in the system. Most of these values are indexed by the points Global Point Number (GPN) that is hashed out by IMAGE when the point is put into the database. The GCS version of RMRPA is that same as the HERMES III version.

4.1.4.9 ACMON

The program ACMON receives action reports through class buffers from RAMRP and updates the ACTION_TABLES in the SHEMA partition "HGTAB" and prints out the message and time portion of the action report on the log printer. ACMON is scheduled at boot time from the WELCOME1.CMD file. The GCS version of ACMON is the same as the HERMES III version.

4.1.4.10 ALMON

The program ALMON receives alarm reports through class buffers from RAMRP and updates the ALARM_TABLES in the SHEMA partition "HGTAB" and prints out the message and time portion of the action report on the log printer. ALMON is scheduled at boot time from the WELCOME1.CMD file. The GCS version of ALMON is the same as the HERMES III version.

4.1.4.11 PVDBM

The program PVDBM receives global reports through class buffers from RAMRP and updates the POINT_TABLES in the SHEMA partition "HGTAB". Global reports are the reports sent out from the CATs that actually contain the I/O point state data. Global reports are sent out before the group reports are sent out so that if a monitor program receives a group report, the data in the POINT_TABLES will be current. PVDBM is scheduled at boot time from the WELCOME1.CMD file. The GCS version of PVDBM is the same as the HERMES III version.

4.1.4.12 STALL

The program STALL is scheduled at boot time from the WELCOME1.CMD file and put on the time list to run at one-fourth of the shortest stall time of any CAT in the file /MACH_ELEMENTS/ELEMENT_MASTER. STALL is mapped into stall reports (a type of status report) from RAMRP and compares the time of the last report sent out with the current system time. If the difference between the report time and the system time is greater than the stall time of that CAT, STALL sends out a stall report, by using a class buffer, to REPOR and logs a message on the log printer. In this way STALL makes sure that all of the CATs are still running. The GCS version of STALL also checks to see if LU 44 is down and stops all jobs if it detects a fault.

4.1.4.13 STMON

The program STMON receives status reports through class buffers from RAMRP and updates the STATUS_TABLES in the SHEMA partition "HGTAB". The STATUS_TABLES contain information from the CATs telling if the CAT has sent out an event started, event aborted, or event ended report; if the CAT has sent out an archive completed report; if an event started or ended report has been sent from the System Event Monitor; the current event number; and the resource number to access these tables. STMON is scheduled at boot time from the WELCOME1.CMD file. The GCS version of STMON is the same as the HERMES III version.

4.1.4.14 ARC30

The program ARC30 is the slave data archiver, it is scheduled by the master archiver ARCHV. ARC30 receives a PTOP buffer of data from ARCHV and stores it on disc in the files /SHOT_XXXX/ERN_YYY/TEMPRY_ZZ_XXXXYYY where XXXX is the shot number, YYY is the event number, and ZZ is the node number that the CAT that this data came from is running on. These buffers are stored on disc in the format that ARC30 receives them. After the event is over, LNKER is scheduled to further process these files. When ARC30 has finished storing all of the data for an event on disc, it sends an archive complete report to REPOR. This report is then sent to the System Event Monitor, SYEMN. The only difference between the GCS version and the HERMES III version of ARC30 is that the GCS version was modified to allow for two digit node numbers (as all of the CATs for GCS run on node 30).

4.1.4.15 LNKER

The program LNKER is scheduled,with a buffer, by SYEMN when the event is over to further process the data files ARC30 stores on disc. LNKER creates the files /SHOT_XXXX/ERN_YYY/DETAIL_ZZ_XXXXYYY where XXXX is the shot number, YYY is the event number, and ZZ is the node number. LNKER is used to create linked list data files of point state data from the linear stored data in the temporary files. These files are used by the data recalling and plotting program, RECAL. When LNKER is finished, it purges the temporary data files and prints a message on the log printer. The differences between the GCS version and the HERMES III version of LNKER is that the GCS version was modified to allow for two digit node numbers and the total number of allowed points for the system was changed from 2701 to 1009. This change was initially made in the file /IMAGE2/GCS_SCHEMA when the IMAGE database was created. It was required to code this number into several programs, the most important place this number was changed was in the file /GCS_CODE/I_POINT_TABLE_SI.FTN (this file is included into many programs).

4.1.4.16 BUAMF

The program BUAMF is used to build the archiver master file template and must be run whenever the point database is changed. BUAMF creates the files /SHOT_SPEC/ARC00_FPN_MASTER and /SHOT_SPEC/ARC00_GPN_MASTER. The FPN_MASTER file is a alphabetically sorted list of Facility Point Names (FPNs) and their Global Point Numbers (GPNs). The program RECAL uses this list to preform a binary sort to find the GPN for a FPN. The GPN_MASTER file is a file, indexed by GPN, of all of the point's attributes. These include the point descriptor, the engineering units, the coefficients, the point gain value, and the allowed limits for a point. These files are copied to the files /SHOT_XXXX/ERN_YYY/FPN_SORT_XXXXYYY and /SHOT_XXXX/ERN_YYY/MASTER_XXXXYYY by SUARF at the beginning of an event. All of the information in these files is stored in the SHEMA tables for the current database configuration. This information is stored with the shot data so that RECAL can use the information that was current when the data was stored should the database be changed. The only difference between the GCS version and the HERMES III version of BUAMF is that the GCS version was modified so that the total number of allowed points for the system was changed from 2701 to 1009.

4.1.4.17 SUARF

The program SUARF is scheduled, with a buffer, by SYEMN at the beginning of the event. SUARF copies the file /SHOT_SPEC/ARC00_FPN_MASTER to /SHOT_XXXX/ERN_YYY/FPN_SORT_XXXXYYY, copies /SHOT_SPEC/ARC00_GPN_MASTER to /SHOT_XXXX/ERN_YYY/MASTER_XXXXYYY, and creates the file /SHOT_XXXX/ERN_YYY/TEMPRY_ZZ_XXXXYYY.

4.1.4.18 RECAL

RECAL is the point data recalling and plotting program. RECAL uses the files stored in the /SHOT_XXXX/ERN_YYY directories. The use of the RECAL program is documented in the "Hermes III Remote Operation Control System Operators Manual". The GCS version of RECAL is the same as the HERMES III System version except it allows for two digit node numbers in the input file names.

4.1.5 The System Event Monitor (SEM)

The SEM subsystem of programs were designed primarily for use in the HERMES III System and will be only briefly outlined in this section. For the HERMES III System, a shot is defined that contains information about shot type and target states of the CAT systems and the coordinator. This information is stored in the files /HOLD/SUBSYSTEM_NAME/EOF.EOF and /HOLD/SUBSYSTEM_NAME/SOF.SOF and are copied to the directory /SHOT_XXXX/ERN_YYY when the shot or event is started. There may be more than one event for each shot, but all of the information defined for the shot will be the same for all events taken for that shot. The information in the EOF.EOF files is sent to the CATs when a command is sent. For GCS the shot and event system was kept and is used primarily for recording data in a test mode. For GCS, the shots are defined as Local Shots (no subsystems in automatic mode) and new events are started to record data. The use of shots and events is documented in the "Hermes III Remote Operation Control System Operators Manual"

The SEM subsystem consists of: MBOG, the model based objective generator; SET_KNOBS, the event starting program; SMCI, the Shot Master Command Interface; SOSI, the Shot Objective Specification Interface; SQMGR, the Shot Queue Manager; SREPT, to provide shot specification reports; and SYEMN, the System Event Monitor. All of the SEM programs for GCS are the same as the HERMES III versions.

4.1.5.1 MBOG

The program MBOG is used to create the EOF.EOF files with target information for the CAT subsystems in them. These files are not used for GCS in normal operation. MBOG is scheduled, with a buffer passed, by SOSI.

4.1.5.2 SET_KNOBS

The program SET_KNOBS starts and ends new shots and events. This program builds directories for shot data, copies the SOF.SOF and EOF.EOF files to the correct directories, sets the correct shot and event number in the file /EVENT/OLC, and schedules SUARF to copy the FPN_MASTER and GPN_MASTER files. SET_KNOBS receives begin and end status reports from RAMRP. SET_KNOBS is scheduled by the program SMCI when the operator selects to start or end events of shots.

4.1.5.3 SMCI

SMCI is the program that schedules the programs: SET_KNOBS, SOSI, SQMGR, and SREPT based on operator input.

4.1.5.4 SOSI

The program SOSI is scheduled by the program SMCI when the operator selects to define a shot. SOSI creates the SOF.SOF files and schedules MBOG to create the EOF.EOF files.

4.1.5.5 SQMGR

The program SQMGR is scheduled by the program SMCI when the operator selects to examine, delete, or rotate shot numbers for pending shots in the shot queue.

4.1.5.6 SREPT

The program SREPT is scheduled by the program SMCI when the operator selects to print out a shot report for a previous or pending shot.

4.1.5.7 SYEMN

The program SYEMN is responsible for actually starting and ending events based on reports received from SET_KNOBS, ARC30, and the CATs (through REPOR/ASR30/RAMRP). SYEMN sends start and end eventcommands to the CATs, schedules TMSYN at the start of the event, and scheduled LNKER at the end of the event.

4.1.6 The Operator Machine Interface (OMI)

The OMI for GCS is completely different from the OMI for the HERMES III System. The reasons for this are that the hardware for display of the graphics and operator input were quite different and the OMI for GCS did not have many of the requirements of the HERMES III System OMI. For the HERMES III System, the graphics display is performed by HP-12065 Color Video Interface cards located in the A-900 computer and the operator input is performed by touchscreens connected to graphics terminals. For GCS, a color graphics terminal was used for the graphics display because the display is located quite a distance from the computer and connected by RS-232 fiber optic modems. The buttons connected to the terminal were used for the operator input because a touchscreen could not be used in the high electromagnetic interference environment.

The GCS OMI did not have the requirements of the HERMES III OMI of needing to show many different graphical displays on different monitors and because of this required fewer modules. The GCS OMI only had to display two pages of information on one color monitor.

The programs in the OMI subsystem are: CMDIN, the command input program; GCSDM (and GCSCM, the color version), the display monitor; and WATCH, the time report generator.

4.1.6.1 CMDIN

When GCSDM runs, it sets up CMDIN to be an interrupt scheduled program on the terminal GCSDM is running on. GCSDM also sets up the multiplexer for the LU that it is running on to have First In First Out (FIFO) buffering. As a result of this, whenever any key is pressed on the keyboard for the terminal that GCSDM is running on, CMDIN is scheduled and the value of the key pressed is stored in the multiplexer's buffers.

When CMDIN is scheduled, it reads the value of the key pressed from the multiplexer's buffer and sends this information, in a command report, to REPOR. GCSDM then receives this command report along with the other reports that it is mapped in for. This was set up to allow for fast single-character input and to allow GCSDM to be suspended on a class get and resume when it got any type of report. CMDIN does not exist on the HERMES III System.

4.1.6.2 GCSDM

GCSDM is the program that processes the operator's input and displays the graphical representation of the system's status. GCSCM is the color monitor version of GCSDM. GCSDM is originally scheduled at boot time from the file WELCOME1.CMD. When GCSDM starts up, it maps into action reports, alarm reports, command reports, group reports, and time reports. GCSDM then reads the initial point statuses and CAT states from "HGTAB" and draws the status display. After the initial display is drawn, GCSDM suspends on a class get and waits for reports from RAMRP.

When GCSDM receives an action report, it writes the action message on the screen (in yellow for oil transfer messages, blue for water transfer messages) and stores the CATs' states information in its internal buffers. GCSDM uses this state information, and information in the oil and water active and pending job tables in "HGTAB", to determine what buttons to present to the operator. By using this information, GCSDM will only present buttons for the operator for logical commands (will not present a fill button for a full or filling tank, will not present a drain button for an empty or draining tank, will not present a surveillance button for a system that is already in surveillance, etc.). If a system is in recirculation or surveillance, the recirculation or surveillance button is replaced with a cancel recirculation or cancel surveillance button.

When GCSDM receives an alarm report, it writes the alarm message on the screen, in red, and all of the buttons for that subsystems are removed except for a clear alarms button.

When GCSDM receives a group report, it reads the point values from "HGTAB" and updates it's display accordingly. If GCSDM receives a command report, it enters a sequence to process the command. The operator must first enter a code number to enter commands. Then a series of buttons for commands are presented. Most commands require the operator to select two or more buttons. For example if the operator wanted to fill the PROTO oil tank, the operator would enter the code and a new set of buttons of allowed commands would be displayed. The operator would then select the FILL OIL button and a new set of buttons, for oil tanks that were allowed to be filled, would be displayed. The operator would press the PROTO OIL button and GCSDM would then ask the operator to press the ENTER button to start the fill of the PROTO oil tank. When the enter button is pressed, GCSDM sends the command to OILCA. The EMERGENCY DRAIN and EMERGENCY STOP commands may be entered after the normal code is entered or after an alternate code, that is known to all personnel in the building, is entered.

Time reports are used when GCSDM is in this command sequence to allow for timeouts for operator input. If the timeout occurs in the command sequence without any new operator commands, GCSDM returns to the beginning of the command sequence and waits for the authorization code to be entered. Time reports are sent out by the program WATCH when it is scheduled by GCSDM. By using reports GCSDM can receive commands and updates asynchronously, and present graphics and allow for operator input at the same time without using excess CPU time.

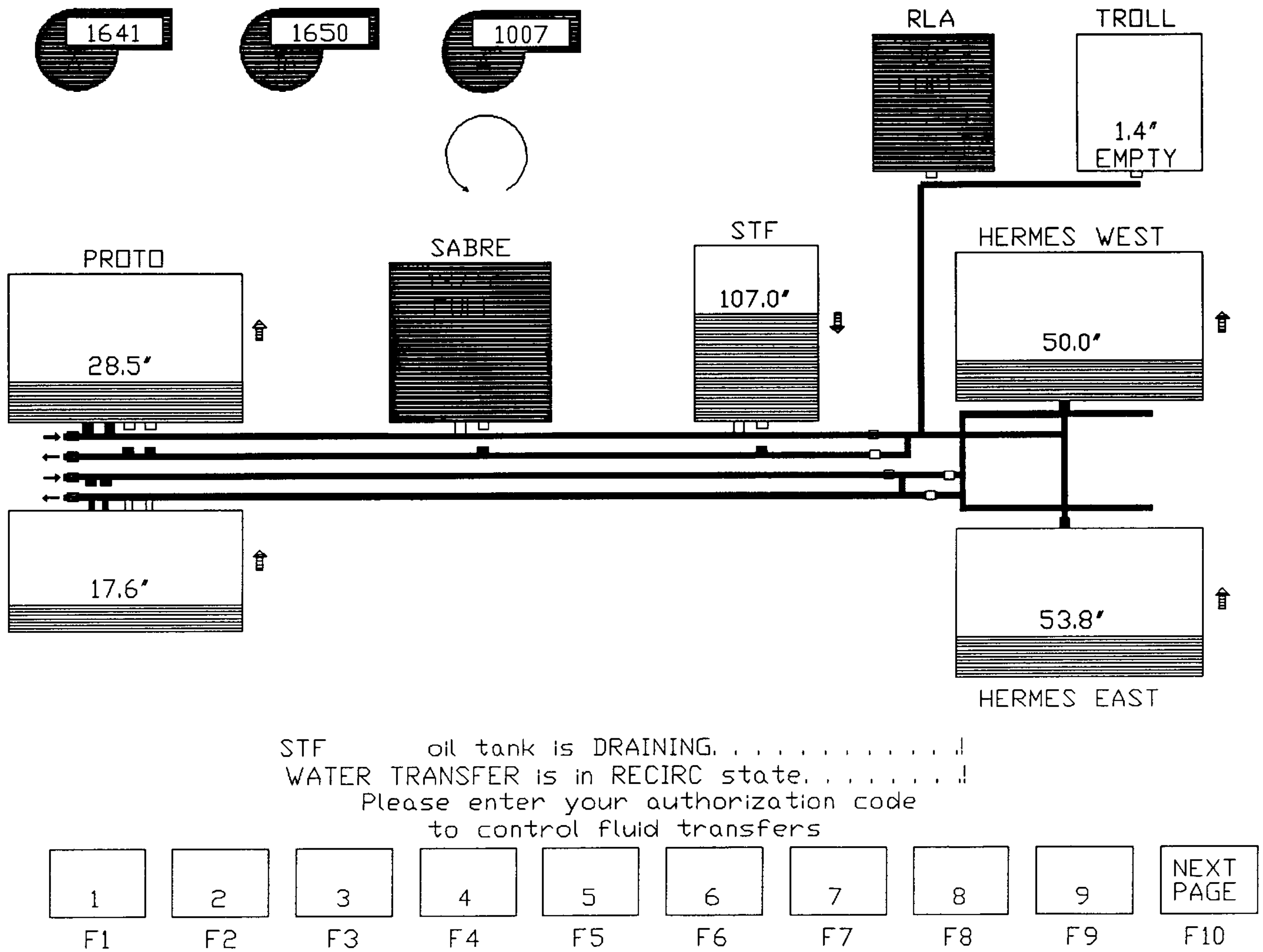


Figure 7. Page One of the Operator Interface Screen.

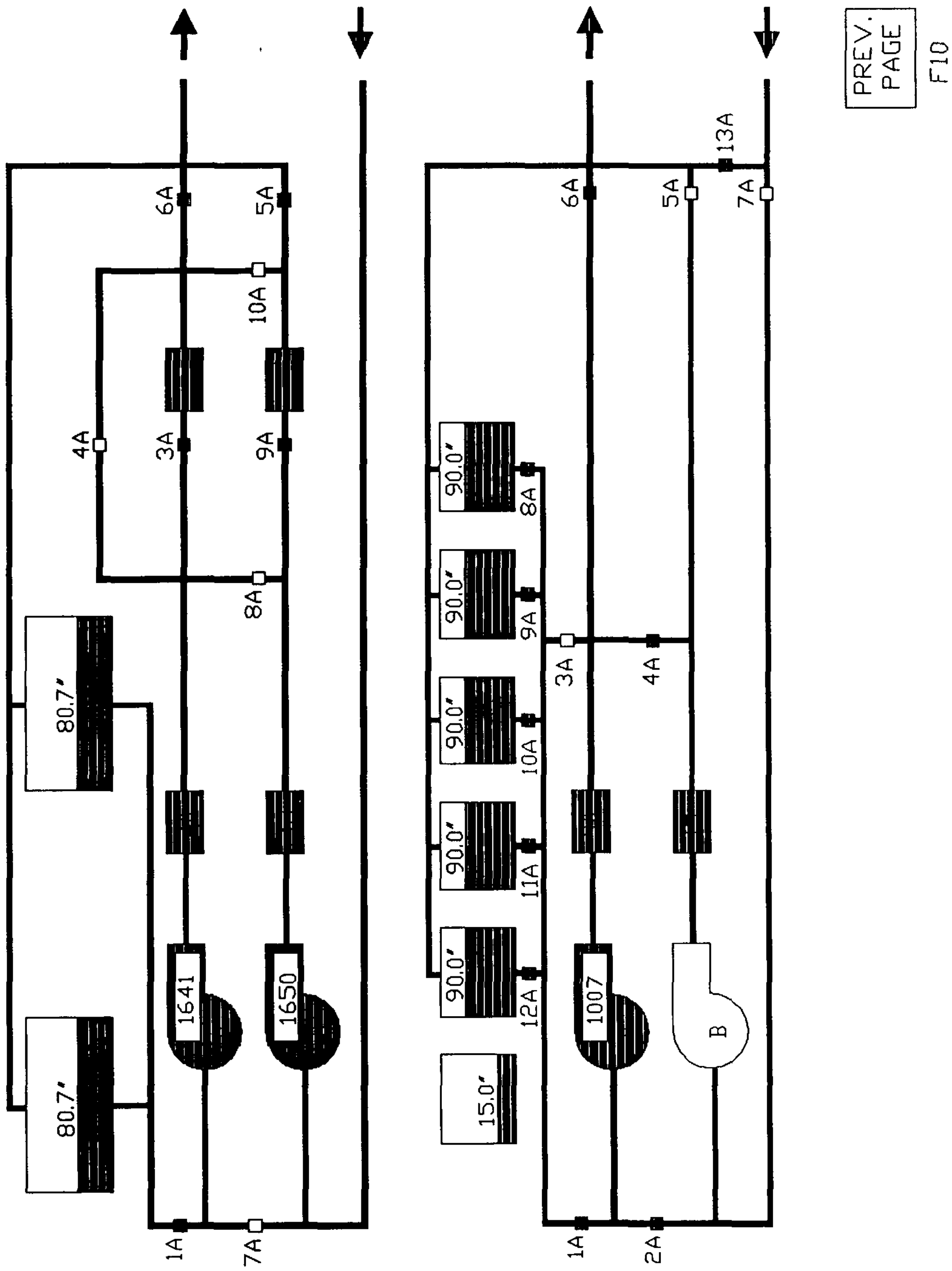


Figure 8. Page Two of the Operator Interface Screen.

Figure 7 shows page one of the OMI graphic display and Figure 8 shows page two. Page one shows the accelerator tanks,pipes, and valves in Building 970 and pump statuses if they are on. Page two shows the storage tanks, pipes, valves, and pumps in Building 970A. The oil transfer system is shown in yellow and the water system in blue. The fluid levels in the accelerator tanks are shown by the filled in colors in the tank with the inch values written inside the tank. If the tank is empty (the empty sensor off), the word empty is written in the tank. If the tank is full (the full sensor on) the word full is written on the tank and the fluid color in the tank is changed to dark yellow for oil and dark blue for water. The valves are shown with a hollow box for a closed valve and a filled box for a open valve.

On page one, if the fluid transfer pumps are on, they are shown above the tanks with a pump symbol and the pump RPM written inside the symbol. On page two the pumps are shown in the same way if they are on and shown with an unfilled pump symbol if they are off. A circular arrow is drawn in yellow if the oil system is in recirculation and in blue if the water system is in recirculation. If the oil system is in recirculation, the words 970, 970A, or DRAIN are written inside the circle to indicate the type of recirculation. A telescope symbol is drawn in yellow if the oil system is in surveillance and in blue if the water system is in surveillance. At the bottom of page one the action and alarm messages are written and the operator input buttons and instructions are shown. GCSDM does not exist on the HERMES III System.

4.1.6.3 WATCH

The program WATCH is scheduled by GCSDM to send out time reports, when GCSDM is in the command sequence, to wake up GCSDM to determine if there is an operator input timeout. WATCH sends time reports to REPOR with a class write. WATCH on the GCS is the same as the version on the HERMES III System used to wake up the coordinator program.

4.1.7 TOOLS

The following is a short description of some of the tools we wrote to use in software development and management.

4.1.7.1 CBULD

The program CBULD is used to create the command files that are used to compile all of the FORTRAN modules in a directory.

4.1.7.2 CMD_CONTINUE

CMD_CONTINUE is used in our compile command files to allow the programmer to correct an error in a program that is compiling and then resume the command file where it was stopped.

4.1.7.3 CMD_VALUE

CMD_VALUE is used to pass a value, from a user, into a running command file. Hewlett Packard provided a routine to do this in 5.1 and newer versions of RTE-A.

4.1.7.4 CNUM

CNUM is used to show the ASCII, octal and binary representation of a decimal number.

4.1.7.5 DSRCH

DSRCH is used to search for duplicate file names in directories.

4.1.7.6 ESTAT

ESTAT is used to show the status of events.

4.1.7.7 INCLD

INCLD is used to provide a printout of what include files are used by which modules and what modules use which include files.

4.1.7.8 ND

ND is a UNIX type of set working directory command. It was copied from a copy of HP Professional Magazine.

4.1.7.9 PSTRG

PSTRG is used to search for strings in files in directories.

4.1.7.10 RPSYS

RPSYS is used to RP programs into the system session.

4.1.7.11 RUSYS

RUSYS is used to run programs in the system session.

4.1.7.12 TIMER_A600

TIMER_A600 is used to check CPU utilization on an A-600 computer.

4.1.7.13 VTERM

VTERM is used to log onto other computers on a DS/1000 network. It was obtained from another division at Sandia.

4.1.7.14 WDSYS

WDSYS is used to change the working directory of the system session.

4.1.7.15 XQSYS

XQSYS is used to run programs in the system session and not wait for them to complete.

4.1.7.16 XREF

XREF is used to provide a printout of what modules are called by which modules and what modules call which modules.

4.2 Modifying the Software

The software for the GCS was designed so that the parameters used to control fluid transfers such as fill and drain pump rpm and the normal fill and drain heights could be fairly easily changed and accelerator tanks could be removed from the system or new accelerator tanks put in place of old tanks. It would be a large task to add more accelerator tanks than the system was designed for and would require extensive software and hardware changes.

The normal fill and drain heights for accelerator tanks may be changed by raising or lowering the digital level sensors and by using RCIF to change the model parameters. The pump rpm used for recirculation may also be changed with RCIF.

To change the pump profiles for fills and drains, the file INITIALIZE_XXXXX.FTN must be changed. To change the valves that are open for a particular state, the file I_VALVE_STATES.FTN must be changed. The program GCSDM must be changed to change the graphic display of the allowed commands.

5.0 CONCLUSIONS

The Global Control System has shown itself to be a reliable system with regular maintenance of the hardware components. The modification of the HERMES III Control System to be used for fluid transfer has produced a standard, reliable, and easy to maintain system. Hopefully this document will serve to be useful for people maintaining this system and designing future systems. At the time of the writing of this paper, the GCS software was being used as the base for a fluid control system at the Saturn accelerator and the HERMES III Control System is planned to be used as a base for the Saturn Control and Monitor System. When these systems are done, there will be a consistent control system for fluid transfer and accelerator charging and firing for the large accelerators in Area IV at Sandia.

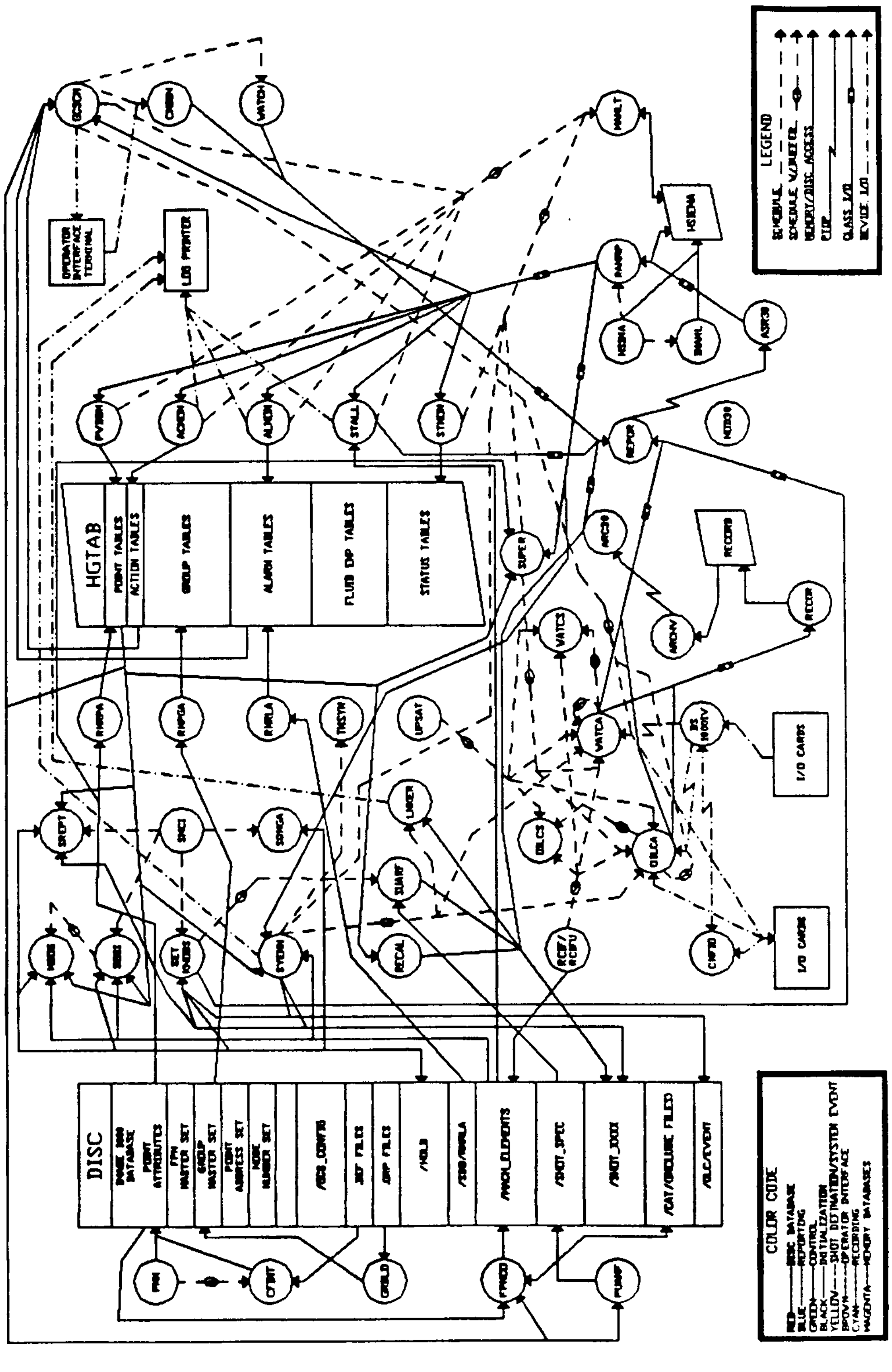
Appendix A . Procedure for Calibrating Analog Level Sensors.

1. Drain the tank to the desired empty level.
2. Measure the level of fluid in the tank.
3. Calculate the current in ma that the sensor should be putting out at that level with the equation:

   $$ma = (inches - c_0) / (c_1 \times 250)$$

   (should be somewhere between 3 and 5 ma).
4. Put the Fluid Control System into "Safe Mode".
5. Connect the amp meter in series with the wires that go to the computer.
6. Adjust the low level adjustment on the level sensor so that the amp meter reads the value calculated above.
7. Disconnect the air supply to the sensor.
8. Measure the value on the amp meter, this is the "no air ma value", record this value.
9. Reconnect the air to the sensor and the wires to the sensor.
10. Fill the tank to the desired full level in manual mode.
11. Reconnect the amp meter.
12. Adjust the high level adjustment to 20 ma.
13. Disconnect the air supply to the sensor.
14. Adjust the low level adjustment to the "no air ma value".
15. Reconnect the air supply.
16. Repeat steps 12-15 until the desired readings are obtained.

Appendix B . Complete Data Flow Path Diagram for all GCS Programs.



Appendix C . Fluid Control System Command Considerations.

FILL

OIL CANNOT fill in surveillance state
CANNOT fill in recirc state
CANNOT fill in Emergency Drain state

Multiple Fills
If filling a tank and a request for a larger volume tank is made
If level of larger tank is 24" > level of smaller tank:
*wait until smaller tank fills to level of larger tank
*then fill both tanks
ELSEIF level of larger is <= level of smaller tank:
*fill both tanks together

IF filling a tank and a fill request for a smaller volume tank is made
IF level of larger tank is 24"> level of smaller tank:
*suspend fill on larger tank
*fill smaller tank to level of larger tank
*then continue filling both tanks
ELSEIF level of larger tank is <= level of smaller tank
*fill both tanks together

WATER Cannot fill HERMES tank
Cannot fill in surveillance state
Can fill in recirc state
Cannot fill in Emergency Drain state

Multiple fills
Stipulations are same as Oil.

DRAIN

OIL CANNOT drain in surveillance state
CANNOT drain in recirc state
CANNOT drain in Emergency Drain state

Multiple Drains
If draining a tank and a request for a smaller volume tank is made
If level of larger tank is 24" > level of smaller tank:
*wait until larger tank drains to level of smaller tank
*then drain both tanks
ELSEIF level of larger is <= level of smaller tank:
*drain both tanks together

IF filling a tank and a fill request for a larger volume tank is made
IF level of larger tank is 24"> level of smaller tank:
*suspend drain on smaller tank
*drain larger tank to level of smaller tank

*then continue draining both tanks
ELSEIF level of larger tank is <= level of smaller tank
*drain both tanks together

WATER CANNOT drain HERMES
CANNOT drain in surveillance state
Can drain in recirc state
CANNOT drain in Emergency Drain state
Multilple drains
Stipulations are same as OIL

FILL/DRAIN

OIL

CANNOT fill and drain simultaneously two tanks beyond main loop. Tanks beyond main loop are HERMES EAST, HERMES WEST, RLA and TROLL.
Example: cannot fill HERMES and drain TROLL at same time.

WATER

CANNOT fill and drain at same time

RECIRC/FILL

OIL

CANNOT fill in recirculation state.

WATER

Command allows any tank to be filled while in the recirc state.

EMERGENCY DRAIN

Cancel all active commands and drain only tank specified in emergency drain. This command cannot be canceled but it can be over-riden by another emergency drain or an emergency stop command.
No other fill or drain will be started until the emergency drain tank is completed or terminated.

RECIRCULATION

OIL

There are two seperate recirc states. Recirc 970 recirculates the main loop. Recirc 970A recirculates the main loop and the storage tanks. Filling/draining and recircing cannot be done at the same time but recircing can be entered as a pending job which will start automatically when all active drains or fills are completed. Recirc must be cancelled before starting another job.

WATER

Can fill any tank in recirculation state

SURVEILLANCE

Surveillance monitors the pressure and levels of all accelerator tanks and storage tanks which are online. If a level exceeds or drops below a starting level +/- a tolerance then an alarm condition exists, valves are places in a static state and a callout is made until a response is made to this alarm condition. Filling/draining and surveillance cannot be done at the same time but surveillance can be entered as a pending job which will start automatically when all active drains or fills are completed. Surveillance must be cancelled before starting another job.

EMERGENCY STOP

Cancel all active and all pending jobs.

CANCEL

Request to cancel specific job.
CANNOT cancel emergency drains
CANNOT cancel suspended jobs

OFFLINE

Request to put specific tank offline.
This command discontinues monitoring pressure or level on specified tank. This is usefull if tank is undergoing maintenance or has a bad level sensor or pressure is bad for some reason.

FAULT

Faults are generally caused by some hardware failure. A message is printed to indicate the problem and a fault state is entered so that the operator will acknowledge that an error has occurred. The pumps are turned off and a static valve configuration is maintained. When the problem has been identified and corrected the operator can clear the fault state and continue operation.

Error messages and reasons for fault state are:

ERROR - Timeout before valves reached state
This indicates that a valve didn't close or open in the specified amount of time. Try again. This valve may be just a little slow.

Pressure is not right for tank _______

This error indicates that air pressure for a given tank is bad. Pressure is not checked for any tank which has been put offline.

ERROR - _______ tank overfull indicator is ON
This error indicates that the high level indicator is on. Level indicators are not checked for any tank which has been put offline. Check for an overfull condition or a faulty indicator.

ERROR - valve does not have status — cannot move
This indicates that computer does not know status of valve. It is neither closed nor open. Check for bad data point.

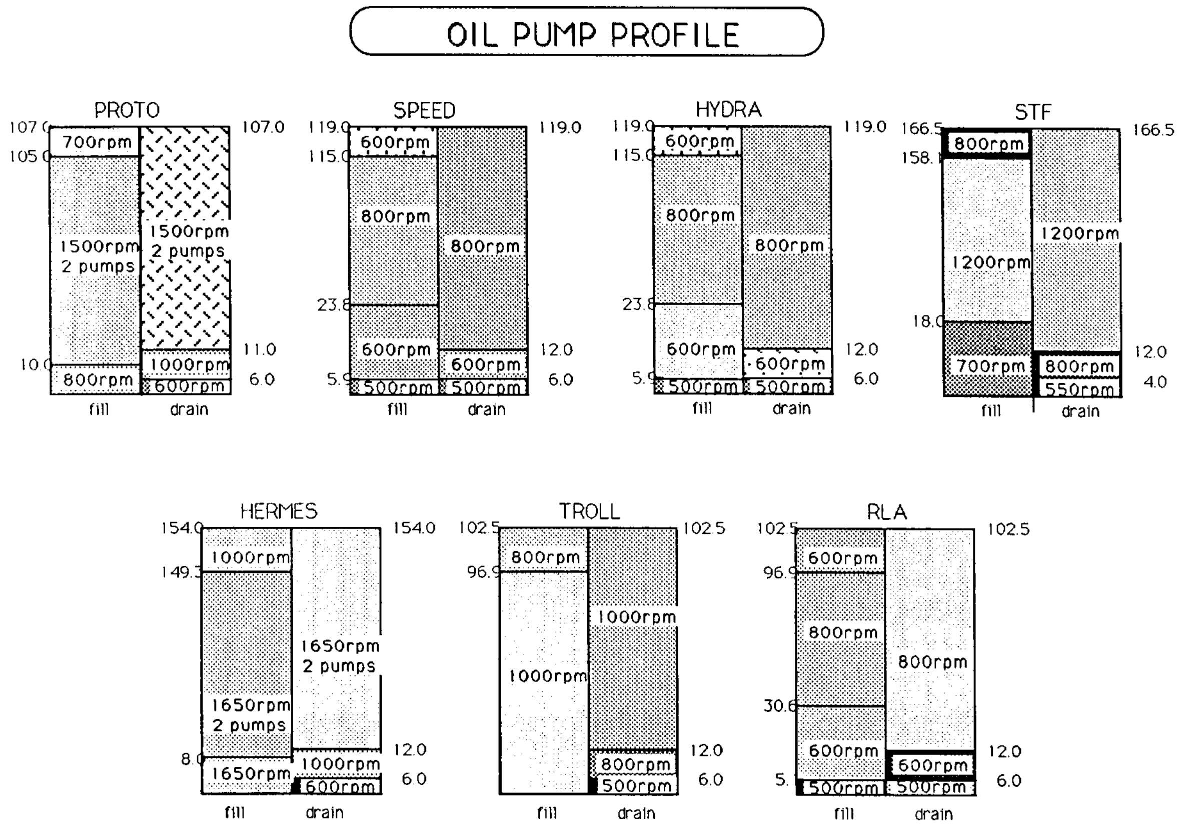
Level is not right for tank _______
Inactive tank levels are monitored at all times. If the levels go above or below a certian tolerance (which is a model parameter and can be adjusted) then the computer indicates an error.

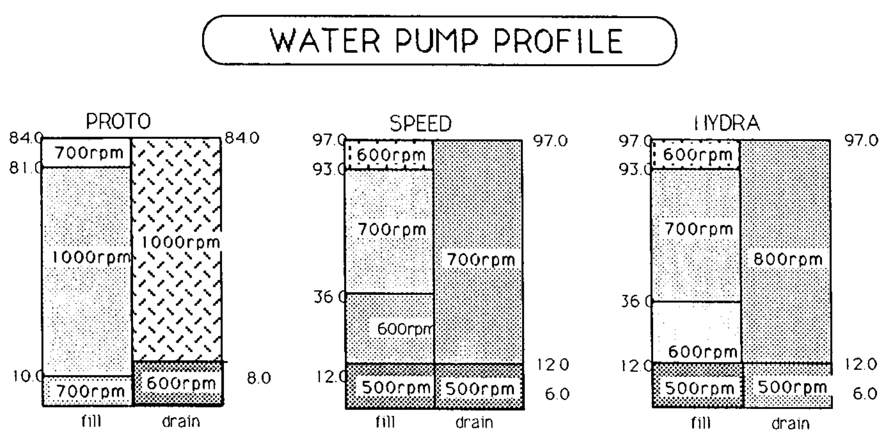
ERROR- pump or valve configuration fault
This error indicates that a valve was in the right position but for some reason has now changed states. Check for hardware failure.
OR
Pump rpm goes above or below the RPM which has been set + or - some tolerance. This tolerance can be adjusted through model parameters.
EXAMPLE: set to 600 RPM
tolerance is 80 RPM
Fault if RPM < 520rpm or > 680rpm

ERROR - pumps not off or bad valve configuration
This error would occur in the SHUTDOWN state and hardware status should be checked

Time-out occurred before pump reached RPM
Pumps have not reached RPM in a given amount of time. Time can be changed in model parameters.

Appendix D . Oil and Water Fill and Drain Pump Profile Tables.





References

[1] Goldsmith, S. Y., "The Structure and Function of the PBFA-II Control System", SAND87-1563, December 1988

[2] Creager, J. D., "Design Criteria Document for the HERMES III Control and Monitor System", unpublished project document, May 1986

[3] Creager, J. D., Mikkelson, K. A., " Initial Design Document for the HERMES III Control and Monitor System", unpublished project document, September 1986

[4] Creager, J. D., Mikkelson, K. A., " Final Design Document for the HERMES III Control and Monitor System", unpublished project document, December 1986

[5] Mitchell, R. A., " HERMES III Remote Operation Control System Operator's Manual", unpublished document, December 1988

[6] Simpson, C. E., "System Manager's Technical Guide for PBFA-II Control/Monitor System", SAND87-1564, April 1988

[7] Micono, J. G., "PCS Technical Reference", unpublished document, March 1988

[8] Southwest Controls, Inc., 1717N. 77th St., Scottsdale, AZ, "Sandia Bubblers Level Measuring System Job # A9017"

[9] Hewlett Packard Company, 1266 Kifer Rd., Sunnyvale, CA, "RTE-A Programmer's Reference Manual"

[10] Hewlett Packard Company, 19420 Homestead Rd. Cupertino, CA, "DS/1000-IV Network Manager's Manual"

[11] Hewlett Packard Company, 19483 Pruneridge Av. Cupertino, CA, "IMAGE/1000-II Data Base Management System Reference Manual"

[12] PBFA-II Control System Programming Team, "PM&M User's Manual", unpublished document, October 1984

DISTRIBUTION:

1200	J. P. VanDevender
1240	K. R. Prestwich
1253	M. T. Buttram
1266	D. D. Bloomquist
1266	R. J. Clevenger
1266	E. Dawson (EG&G)
1266	E. J. Mader (EG&G)
3141	S. A. Landenberger (5)
3141-1	C. L. Ward (8) for DOE/OSTI
3151	W. I. Klein (3)
3202	P. C. Pei
3202	G. J. Smith
3216	A. M. Fine
3216	W. Mairson
7532	D. B. Davis
7545	C. E. Simpson
7823	J. P. Furaus
7823	L. O. Seamons
8524	J. A. Wackerly
9300	J. E. Powell
9340	W. Beezhold
9342	L. M. Choate
9342	A. W. Sharpe
9342	B. A. Woodard (EG&G) (2)
9343	D. G. Baur
9343	H. L. Brown (Ktech) (2)
9343	J. M. Farmer (Ktech)
9343	R. A. Miesch (Ktech)
9343	K. A. Mikkleson (2)
9343	R. A. Mitchell (Ktech) (20)
9343	R. W. O'Rourke (Applied Physics)
9343	G. A. Zawadzkas (5)