# Mercury: A Satellite Ground Station Control System

James W. Cutler
Christopher A. Kitts
Space Systems Development Laboratory
Durand Building, Room 250
Stanford University, Stanford, CA 94305-4035
(650) 723-6021, (650) 725-6794
jwc@stanford.edu, kitts@leland.stanford.edu

*Abstract* - As part of its research program in spacecraft operations, Stanford University's Space Systems Development Laboratory (SSDL) is developing a ground station control system to support advanced command and telemetry operations. Known as Mercury, this control system provides direct human console control, remote human tele-operation, and script/program based autonomous control. Each of these modes supports the ability to configure and monitor station equipment as well as to conduct command and telemetry operations with spacecraft. The Mercury system is being implemented on SSDL's OSCAR-class amateur radio ground station. Mercury is currently being used to operationally test two SSDL microsatellites being prepared for launch in 1999. This paper describes the design and operation of the Mercury system, its implementation in the SSDL OSCAR station, its use to conduct mission operations with SSDL microsatellites, and its role in SSDL's multi-satellite, multi-ground station mission operations architecture.

TABLE OF CONTENTS



## 1. INTRODUCTION

Automation is an enabling technology for reducing the cost and enhancing the performance of operating space systems. A variety of developers are currently exploring new automated reasoning paradigms (expert systems, model-based reasoning, case-based reasoning, etc.), the optimal deployment of control authority (centralized vs. decentralized, ground-based vs. spacecraft-based, etc.), and the application of this technology to performing specific tasks (planning, scheduling, execution, fault management, etc.).

One specific area of interest lies in the development of ground segment automation capability and its integration with the space segment of the mission architecture. This domain is of interest to a vast community and is an active area of research within NASA, the military, and the commercial sector. Specific low-level automation objectives target issues such as ground station equipment control as well as command and telemetry processing. More advanced objectives include issues such as robust contact execution and high level control.

Stanford's Space Systems Development Laboratory (SSDL) is developing a simple but comprehensive spacecraft operations network to develop and validate operations technologies such as these [1]. The SSDL network will include a number of microsatellites developed by SSDL and its partners, a variety of globally distributed ground stations for communicating with the microsatellites, Internet and amateur radio based communication links, and a centralized mission control complex for system management. Although composed of simple elements, this system allows real-world experiments to be executed in a controlled environment capable of accommodating high risk and providing experimental validation.

For this space system, automating ground segment tasks serves as a means to explore ground automation issues; it also enables efficient operation of the space system in support of other research initiatives in planning, scheduling, high level product specification, anomaly management, and data summarization.

## 2. SSDL GROUND STATION

SSDL has a low-cost, small-scale ground station fully equipped to contact OSCAR[1]-class satellites in the 2m and 70cm amateur radio bands (140Mhz and 430Mhz). The station will be used to command and control SSDL's microsatellites upon their launch into orbit. Current ground station use includes students obtaining satellite contact experience by contacting various orbiting satellites and other space vehicles containing amateur radio such as Mir and the Space Shuttle. Students also use the station to perform operations experiments on Sapphire, SSDL's first microsatellite, while it is waiting for a launch opportunity.

Figure 1 shows the general layout of SSDL's ground station. The main equipment includes:

[1] OSCAR – Orbiting Satellite Carrying Amateur Radio.

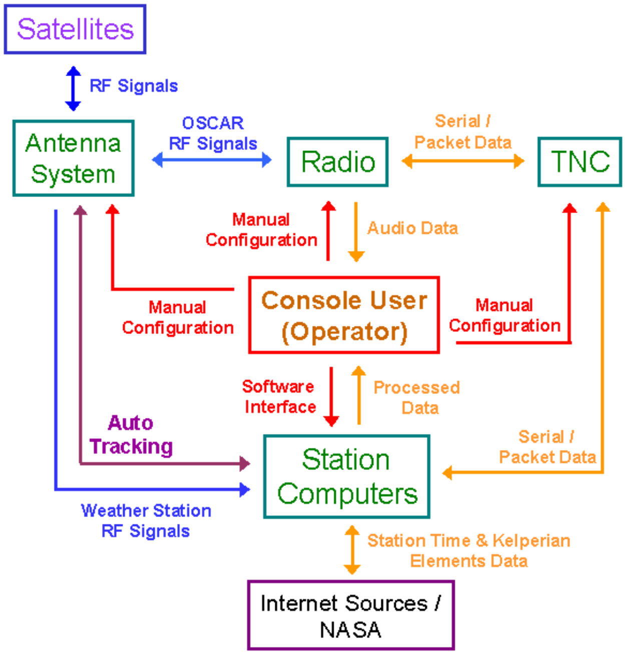



**Figure 1** – Block Diagram of SSDL OSCAR station

- A full duplex transceiver operating on the 2m and 70cm bands with a 160W output amplifier for the 2m band.
- Two Yagi antennas for the 2m and 70cm bands with preamplifiers and position control motors.
- A terminal node controller (TNC) that serves as a modem and uses the amateur radio packet protocol, AX 25, at rates up to 9600 baud.
- A 486 personal computer (PC) that interfaces to the terminal node controller and also controls the antenna position.
- A WEFAX weather station.

A full list of ground station equipment is found at http://aa.stanford.edu/~ssdl/facilities/gs/index.html.

A student present in the ground station control room manually operates station equipment. Satellite communication sessions are conducted at the PC. Transmit and receive frequencies are manually adjusted with the transceiver tuning dial. Figure 2 shows a photograph of the transceiver and TNC. The software package, InstantTrack, running on the PC controls antenna positioning. Figure 3 shows an InstantTrack output screen tracking the space station Mir.

The current operational scenario outlined above has significant deficiencies. First, the simultaneous, manual operation of station equipment required during a satellite contact is often difficult at best. For example, the constant frequency adjustments of the transceiver required to compensate for doppler shift[2] in satellite frequencies make it difficult for a single operator to conduct a satellite contact. Second, station operators are constrained to be physically present at the station. This limits ground station location possibilities and satellite contact opportunities. These and other deficiencies produce an inefficient operational scenario for the ground station.

**Figure 2** – Photograph of TNC (top) and transceiver (bottom)

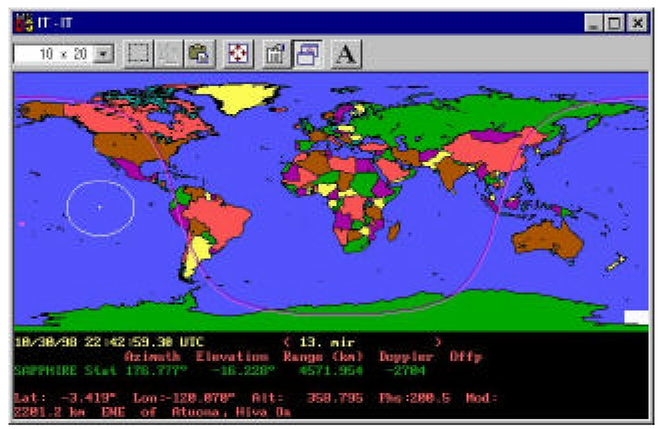


**Figure 3** – Screen capture of InstantTrack output

## 3. MERCURY OVERVIEW

SSDL is developing a ground station control system named Mercury to improve the operational efficiency of the station. Mercury accomplishes this by providing a centralized software interface to control all ground station equipment, software routines to automate station operation, and an

[2] The doppler effect is the change in transmitted frequency as seen by the receiver due to a relative velocity between the two. An example is the high pitched roar of a car as it approaches one and then the sudden decrease in pitch as it drives by.

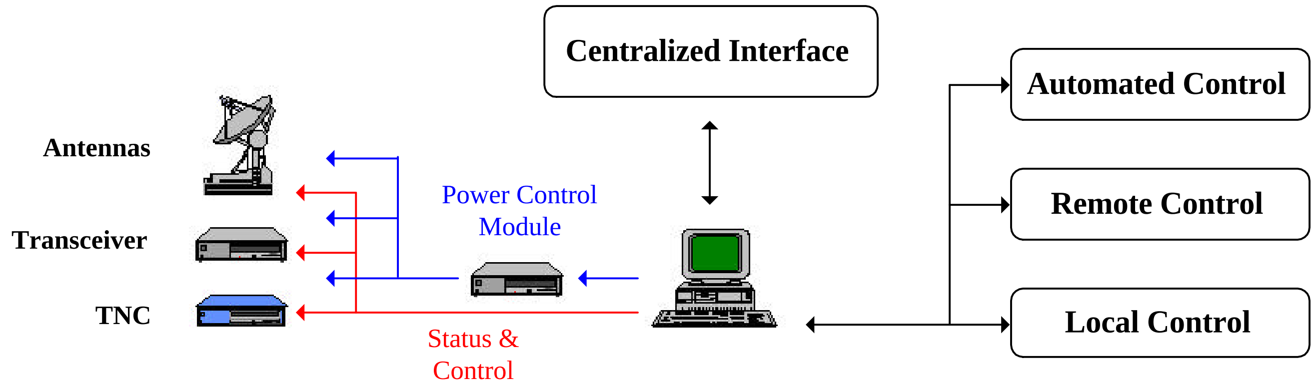



**Figure 4** - Block diagram of Mercury system

Internet gateway to access the centralized interface. Figure 4 shows block diagram of the Mercury control system.

*Centralized Interface*

Many of the ground station components are capable of computer assisted control. The antenna positioning motors can be directed through a PC expansion card. The transceiver has a serial interface to tune frequencies and determine receive signal levels. The terminal node controller is interfaced through a standard RS-232 serial port found on computers. Mercury centralizes software control drivers for these components and provides a single user interface to configure the ground station.

A common interface now exists for the operator to control all station equipment from a single location. The operator is freed from multiplexing between physically separated equipment control panels. Complex equipment operation procedures involving fine-tuning of control panels are now mapped to simpler software commands. This centralized interface simplifies manual operation of the station equipment.

Complementing the software control drivers is a power control module. Power management of station is controllable through the software interface.

*Automation Routines*

The centralized software interface enables automation of station functions. These automation routines include:

- Positioning of antennas to properly track a satellite during its pass.[3]
- Closed loop frequency adjustment of the transceiver to compensate for doppler shift.
- Archiving and formatting of data collected from the TNC.
- Station configuration (communication protocols for the TNC, satellite frequencies, tracking data) for a particular satellite contact session.
- Equipment diagnostics to monitor equipment performance.
- Routine station keeping such as clock synchronization and Keplerian element update.[4]

Automation of station procedures greatly improves operator efficiency. The operator is freed from the manual, time intensive support functions of the station to concentrate on the primary purpose of the station, satellite command and control.

*Internet gateway*

The centralization of station control into a single software package enables the development of custom interface gateways to the software. The primary interface to Mercury is an Internet gateway. An operator with an Internet connection can obtain full access to the ground station control system through a standard telnet application. This Internet interface provides numerous global access channels to the station resulting in a dramatic increase in the accessibility of the ground station.

Mercury reduces the dependence of geographic placement of stations on the geographic location of operators. Stations can now be placed in geographic locations that optimize satellite contact windows. For example, the SSDL ground station has about 30 minutes of prime contact time with a satellite in a low-earth, polar orbit per day. A ground station equipped with Mercury placed in Kiruna, Sweden, a city north of the Arctic Circle, would allow SSDL operators over 300 minutes of contact time per day.

Though designed for operator interface through a telnet application, the text based command protocol of the Internet gateway allows other interface possibilities. A Java applet could be developed to provide a graphical user interface to Mercury. A C program or Perl script could be written to entirely automate a satellite communication session.

[3] This automation was achieved previously through the use of InstantTrack, but incorporating this function into a single interface has several advantages, especially when considering the Internet gateway that is discussed in the following section.

[4] Keplerian elements are data sets published by NASA used to generate azimuth and elevation angles for satellite tracking and antenna positioning.

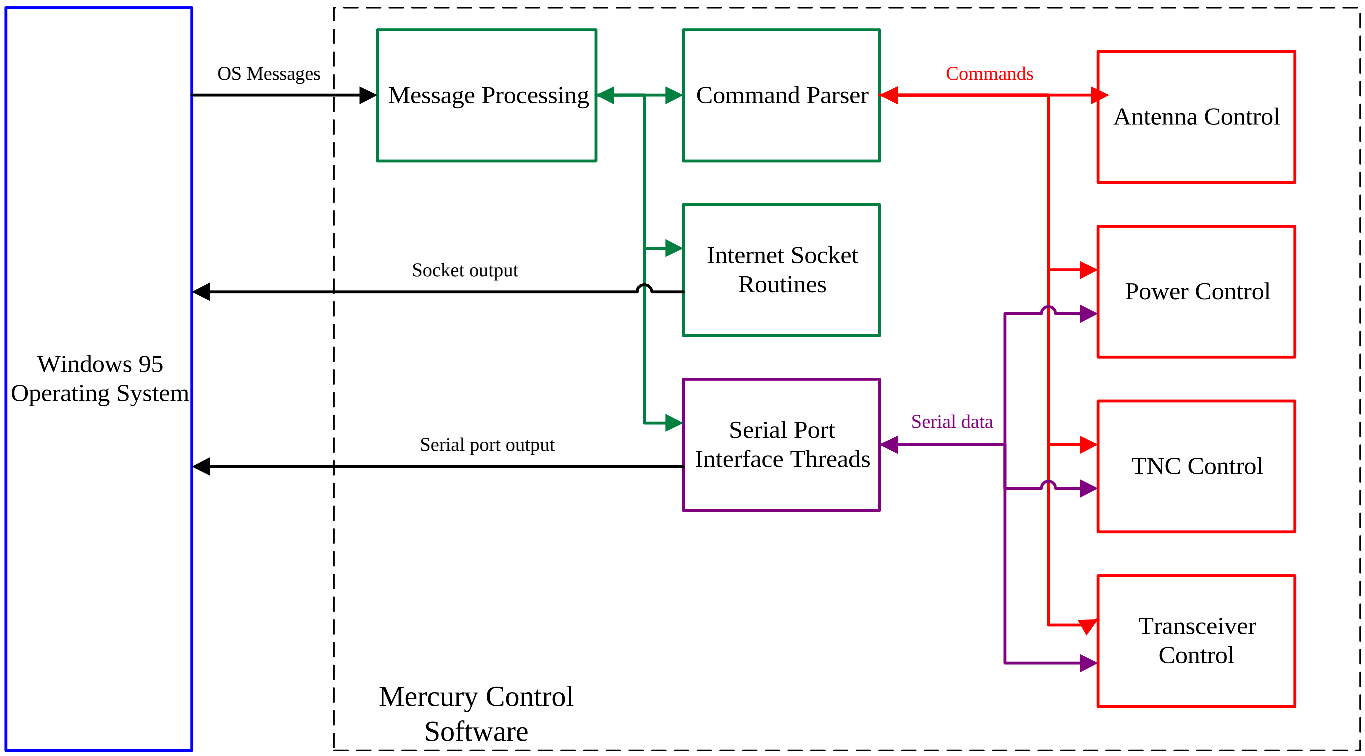



**Figure 5** - Software block diagram

## 4. MERCURY IMPLEMENTATION

A prototype Mercury system has been implemented in the SSDL ground station. Key features of the system are operational including the centralized software interface, the automated control routines, and the Internet gateway.

The centralized software interface resides on an Intel 486 computer running Windows 95. The code was written in the C programming language using the Win32 API, an application programming interface containing system defined functions to access operating system features. The Windows 95 platform was chosen because of its widespread use, the readily available online documentation and support for programming, and the existence of key software drivers to interface to station equipment.

Windows 95 is a thread-based multitasking operating system [2]. Mercury utilizes multitasking by spawning concurrent threads to handle each of the serial ports opened for communication. Mercury communicates to the operating system through function calls of the Win32 API. These functions allow Mercury to allocate memory, process keyboard strokes, manipulate windows, output to the screen, monitor serial port and Internet socket activity, etc. Windows 95 communicates to Mercury through a message passing system. These messages act like software interrupts allowing Mercury to process operating system data like pressed keys from the keyboard, data packets from Internet sockets, and serial port activity. Figure 5 is a software flow diagram for Mercury.

Antenna positioning is performed through function calls to a dynamically linked library[5], RR.DLL, written by Joe Holman [3]. These functions allow the antenna azimuth and elevation angles to be specified. The DLL file interfaces to a specialized computer expansion card, the Kansas City Tracker [4], which is electrically connected to the antenna position motor controller.

The TNC is interfaced through standard serial port communication at a default of 9600 baud, eight data bits, even parity, and one stop bit. These port parameters can be altered to optimize communication with the TNC. Mercury supports both binary and text output from the TNC.

Similarly, the transceiver is controlled through serial port communication. Transmit and receive frequencies are tunable and the transceiver can be polled for receive signal strength. Voltage level converter circuitry is required to convert the RS-232 voltage levels of the serial port (bipolar voltages of +-5V to +-15V) to the TTL voltage levels of the transceiver (0V to 5V).

The Internet gateway uses the Transmission Control Protocol/ Internet Protocol (TCP/IP) [5] to implement a text based command protocol. A typical telnet application can be used to command Mercury through the gateway. Mercury loosely emulates the telnet protocol[6] and allows single user access. Figure 6 is a screen shot showing a login

---

[5] A DLL file is a group of software functions collected together. DLL files allow for efficient programs shared use of common code.

[6] The telnet protocol is defined by RFC 854 "Telnet Protocol Specification". RFC stands for "Request For Comments", written definitions of the protocols and policies of the Internet.

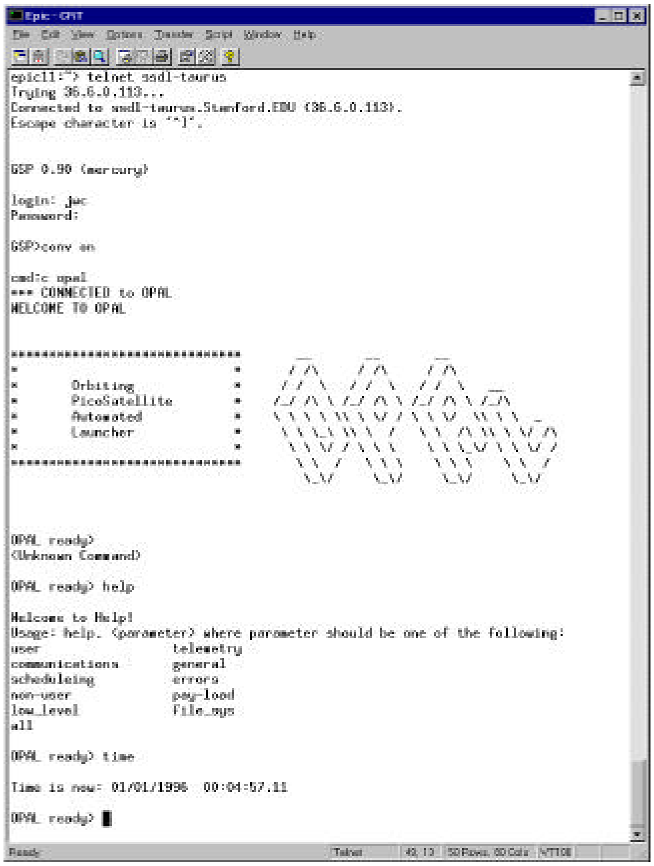



**Figure 6** - Screen capture of Mercury control session

and several command executions through the Internet gateway with the Opal microsatellite. The Internet gateway is used for all Mercury contacts, both remote and local.

Antenna position control has been automated in the Mercury prototype. Code from the Simple Tracking Program by Christie Harper [6] was incorporated into Mercury to calculate azimuth and elevation angles for satellite passes. These calculations are derived from Keplerian elements, orbital description parameters that NASA publishes approximately every two weeks.

## 5. Results

The Mercury prototype system is under development. Members of SSDL are performing operational testing of Mercury while conducting satellite communication sessions with SSDL satellites. Preliminary qualitative assessment of SSDL's Mercury-enabled ground station demonstrates that satellite communication sessions through Mercury are improved over operation using a conventional ground station.

### *SSDL Operational Testing*

The Mercury prototype is currently in use by SSDL students. Design teams are performing operational testing of SSDL's microsatellites, Sapphire [7] and Opal [8]. Student researchers are also conducting operational experiments on Sapphire.

The Sapphire and Opal design teams are developing operational procedures requiring the use of the vehicle under several simulated orbit conditions (orbital position, solar panel currents, etc). Mercury enables flexibility in a designer's physical location during work. The simulated satellite contact times are often at late, inconvenient hours. Mercury frees designers from coming to the station at these late hours and enables them to access the station from their home computers.

The Opal design team has an extreme example of freedom in the designer's physical location. The primary CPU programmer is studying abroad this quarter in Germany. Mercury has allowed this designer to access the Opal engineering model vehicle and perform operational testing. In addition, several modifications to Mercury could be done to also allow the designer to recompile Opal's operating system code and download to the Opal vehicle. Therefore he could continue software development despite the fact that he is in Germany and Opal is in California.[7]

Mercury has also simplified satellite data collection and analysis during development. The Opal design team has written several software programs that automatically collect payload data, process the raw data by generating plots, and publish it to easily accessible web pages. Opal designers can now analyze processed, formatted satellite data within moments of the vehicle downloading it.

Several students are performing health monitoring experiments on Sapphire [9]. Sapphire transmits a simple beacon encoded with the health status of the vehicle. A beacon receiving station has been developed to detect this beacon and notify vehicle operators through a standard personal pager system. Mercury allows the operators to respond to beacon status without having to be physically present at the ground station. The "on call" operators with pagers are free to carry on daily activities as long as they are within Internet access.

### *Preliminary Qualitative Assessment*

The Mercury system consists of three functional components: the centralized interface, the internet gateway, and the automation routines. Table 1 summarizes the benefits of these functional components. The qualitative assessment shows that a Mercury enabled ground station improves an operator's ability to conduct satellite communication sessions. Mercury enables tele-operation of unstaffed, completely automated OSCAR-class ground stations.

---

[7] An estimated 20 lines of code could be added to perform this task but the designer is enjoying his time in Germany, so we decided to give him a break.

Table 1 – Qualitative Assessment of Mercury Benefits

| Mercury Functional Component | Summary of Benefits |
|---|---|
| Centralized Interface | 1) Reduces operator multiplexing between station equipment by bringing control to a single location. 2) Simplifies operation of station equipment by creating a similar command sequence for all equipment. 3) Enables computer automation of routine station functions. |
| Internet Gateway | 1) Reduces dependence of geographic location of operators on the geographic location of stations. 2) Improves accessibility to stations by allowing remote control. 3) Enables access to unmanned stations placed world wide from a single location. |
| Automation Routines | 1) Increases ground station performance by enabling a computer to perform functions more precisely and more quickly than a human operator could. 2) Reduces cost of station operation by reducing human attention to the station. |

The implementation of Mercury may not be optimal. The student developed custom software of Mercury is a risk. A commercially available product might be available to implement the functions of Mercury while being backed by commercial support. Also, the cost of a commercial product could be less than the required cost of paying students to develop the software.

## 6. FUTURE DEVELOPMENT

The Mercury system is a prototype and under development. Future work will expand software capabilities, add hardware system components, and implement station automation routines.

*Software*

The current version of software provides basic functionality, but significant additions are needed to complete Mercury. Security protocols are rudimentary and do not provide adequate safety measures. Routines need to be written to provide robust security measures. The current orbital engine is less accurate than desired and does not calculate satellite velocity needed for doppler shift correction. Custom software is being developed to replace the current engine and provide this functionality. A software watchdog will be added to monitor Mercury software performance. If an error or failure is detected, the watchdog will restart Mercury or reboot the host PC. Also, a graphical user interface will be developed using Java to enhance connectivity to Mercury's Internet gateway.

*Hardware*

Hardware expansion will first focus on implementation of the power control module. Figure 7 shows a block diagram of the module. Mercury will control individual power relays for each station component through an $I^2C$ bus interface. I/O expander modules residing on the $I^2C$ bus will receive command information from the software and toggle the appropriate relay control lines. The $I^2C$ bus is two-wire serial bus developed by the Philips Corporation for integrated component communication [10].

Another improvement will be the addition of transceiver diagnostic hardware to verify proper transceiver operation.

*Automation*

Control of station equipment through a centralized interface has been achieved. Station automation involves computer control of routine station operations that free an operator to focus on satellite communication. Development of station automation will also be a primary focus of future development.

Closed loop control routines to correct for doppler shift will be written in order to precisely tune the station transceiver thereby maximizing received signal strength. In addition, Keplerian element updates will be automated by writing functions to download these elements from established NASA web sites. At a higher level, an intelligent command and telemetry executive capable of managing satellite contacts will be added. This system

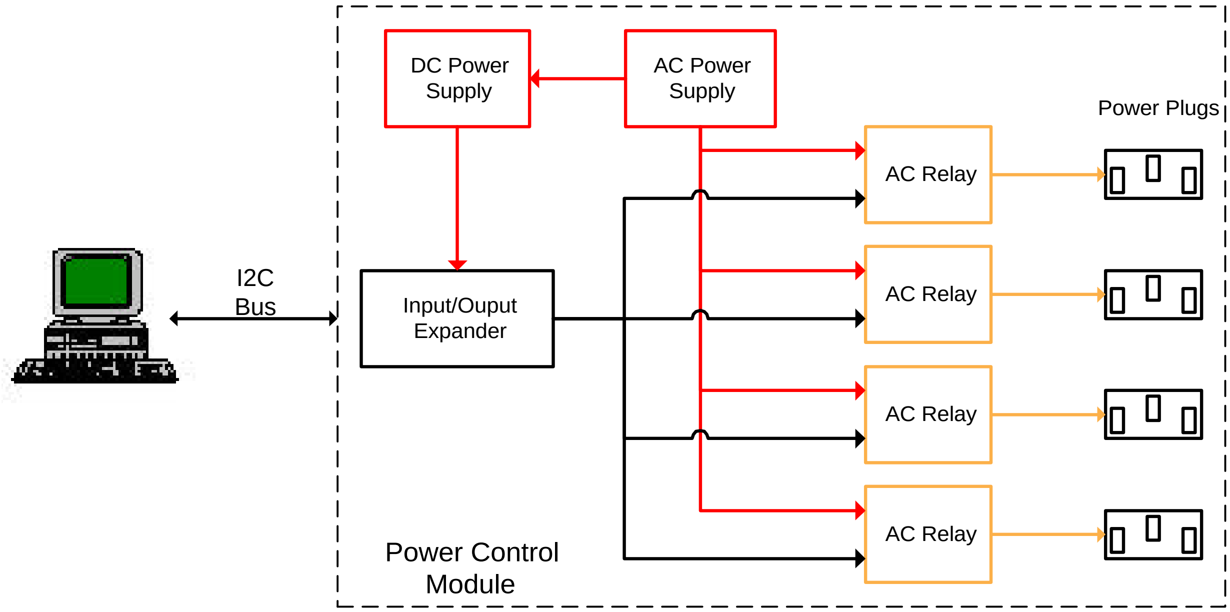



Figure 7 – Block diagram of Power Control Module

will take contact-specific tasks (such as ‘download file’ or ‘schedule a photo to be taken’), derive appropriate spacecraft commands and telemetry responses, and robustly execute these procedures while accounting for typical contact problems.

Finally, the Mercury-enabled ground stations will be integrated within the overall autonomy architecture of the SSDL spacecraft command and control network. Ultimately, this system will accept very high level direction (direct client specification of products or services) which will be planned and scheduled for subsequent execution over a series of ground station contacts. In addition, the stations will serve as part of a multi-satellite health beacon network in order to enable cost-effective anomaly detection.

## 6. Conclusions

SSDL had developed a ground station control system, Mercury, that centralizes station equipment control into a single interface, provides Internet access to the interface, and automates ground segment tasks. This system allows SSDL to explore ground station automation issues. It also enables direct human console control, remote human tele-operation, and script/program based autonomous control of station equipment.

A low-cost prototype of Mercury has been implemented in the SSDL OSCAR-class ground station. Preliminary Mercury results demonstrate an increase in operator efficiency. Centralization of control has decreased the multiplexing of operator time between control panels of equipment. Automation of ground tasks has freed operators to focus on satellite contacts rather than operating support equipment such as antenna tracking and transceiver tuning.

Mercury has also decreased the dependence of ground station location on operator location. The operators are now able to control the ground station via the Internet, thereby providing virtual global access to the station.

## Acknowledgements

The authors wish to thank NASA Ames and the NASA New Millennium Program for its support in developing this research program. Special thanks are given to Prof. Robert Twiggs for administering this project. Appreciation is also extended to SSDL’s operations research students Mike Swartwout, Brian Engberg, and Carlos Niederstrasser, to the Opal software engineer Greg Hutchins, and to Santa Clara University student Corina Hu for their useful feedback and assistance in the design and implementation of this system. Apprecition is also extended to the reviewers of this paper. Their valuable comments helped This work is being performed in partial satisfaction of graduate studies at Stanford University.

## References

[1] Kitts, Christopher A., "A Global Spacecraft Control Network for Spacecraft Autonomy Research," In *Proceedings of SpaceOps '96: The Fourth International Symposium on Space Mission Operations and Ground Data Systems, Munich, Germany*, September 16-20, 1996.

[2] Schildt, Herbert, *Windows 95 Programming Nuts & Bolts: For Experienced Programmers*, Berkeley: Osborne McGraw-Hill, 1995.

[3] Holman, Joe (AD7D), "32-Bit KCT Driver for Win95", 1996, ftp://ftp.amsat.org/amsat/software/win32/rotor-control/kctrr95.zip.

[4] *Kansas City Tracker Web Site*, L. L. Grace Communications, http://www.llgrace.com/.

[5] Stevens, W. Richard, *TCP/IP Illustrated, Volume 1*, Reading, Mass.: Addison Wesley, 1994.

[6] Harper, Christie, "STP: Simple Tracking Program", 1994, ftp://ftp.amsat.org/amsat/software/ source/stp.zip.

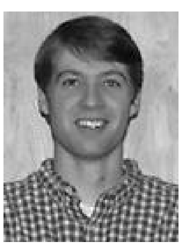
[7] Twiggs, Robert J., and Michael A., Swartwout, "SAPPHIRE - Stanford's First Amateur Satellite", In *Proceedings of the 1998 AMSAT-NA Symposium*, Vicksberg, MI, October 14-16, 1998.

[8] Engberg, Brian, Jeff Ota, and Jason Suchman, "The OPAL Satellite Project: Continuing the Next Generation Small Satellite Development," In *Proceedings of the 9th Annual AIAA/USU Conference on Small Satellites, Logan, Utah*, September 19-22, 1995.

[9] Swartwout, Michael, and Christopher A. Kitts, "Automated Health Operations for the SAPPHIRE Spacecraft," In *Proceedings of ITC/USA '96: The 33rd Annual International Telemetering Conference, Las Vegas, NV*, October 30- November 1, 1997.

[10] Paret, Dominique and Carl Fenger, *The $I^2C$ Bus, From Theory to Practice*, New York: John Wiley and Sons, 1997.

## BIOGRAPHY



**James Cutler** is a doctoral candidate in Electrical Engineering at Stanford University. His research focuses on developing reliable space systems from commercial off the shelf products. Through his research, Mr. Cutler is developing a low-cost, space flight testbed to study radiation effects on microelectronics. Mr. Cutler is project leader of SSDL's second microsatellite, Opal, and is a principal investigator on the third microsatellite, Emerald. He was also a research assistant at Purdue University's NASA Specialized Center of Research and Training in bio-regenerative life support. He developed an autonomous growth chamber and studied plant seedling response to stress. Mr. Cutler received a BS in Electrical Engineering from Purdue University and an MS in the same field from Stanford University.

**Christopher Kitts** is a doctoral candidate in Stanford University's Design Division where he specializes in spacecraft design and command and control systems. He holds a joint position as the Graduate Student Director of Stanford's Space Systems Development Laboratory and as the Co-Director of the Santa Clara Remote Extreme Environment Mechanisms Laboratory at Santa Clara University. Mr. Kitts is also a space systems engineer with Caelum Research Corporation at NASA's Ames Research Center where he develops spacecraft design and autonomy strategies for NASA's New Millennium Program. Mr. Kitts has served in the Air Force as a mission controller of and the Chief of Academics for the Defense Satellite Communications System III spacecraft constellation. He has held a research position at the Air Force Phillips Laboratory and has taught numerous graduate courses in space system design. Mr. Kitts received a BSE from Princeton University, an MPA from the University of Colorado, and an MS from Stanford University.