Calhoun: The NPS Institutional Archive
DSpace Repository

Faculty and Researchers

Faculty and Researchers' Publications

1993-09

Navy Education Through Amateur Satellite Development

Horning, J.A.

Monterey, California: Naval Postgraduate School.

<https://hdl.handle.net/10945/37430>

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

NAVY EDUCATION THROUGH AMATEUR SATELLITE DEVELOPMENT

J. A. Horning[†]
Naval Postgraduate School
Monterey, California

Abstract

The Naval Postgraduate School (NPS) is developing a small satellite for digital store-and-forward communication using spread spectrum techniques. NPS is looking toward the amateur radio community in an effort to utilize cost-effective engineering and proven means of radio message relay. This cooperative initiative between NPS and the amateur radio community provides numerous benefits for the education of NPS officer students. The spacecraft will provide for amateur radio enthusiasts a new space communication mode utilizing spread spectrum modulation for packet radio. It also offers a means of evaluating spread spectrum in the increasingly congested frequency bands.

Introduction

The Space Systems Academic Group at the Naval Postgraduate School supports Space Systems Engineering and Space Systems Operations Curricula. The Space Systems Academic Group is developing a low-cost, small satellite for digital, spread spectrum communication using the amateur frequency band. This small satellite, the Petite Amateur Navy Satellite (PANSAT), enhances the education of officer students in the Space Systems curricula by providing hands-on system design, hardware and software development, system integration, testing, and operation studies.

PANSAT is providing practical research topics for officer student theses. Such topics have already included structural design and analysis of the spacecraft, software design, ground station design, operational analyses, and hardware designs and implementations of communication, electrical power, and digital subsystems. Once in space, PANSAT will facilitate officer students in understanding satellite communications and operations.

The amateur radio community has been involved with packet radio for over a decade and is responsible for standards in communication protocols, as well as packet radio, satellite development, and operations. PANSAT designers have worked closely with this group in order to profit from existing design and operations standards, and to learn from past experience.

PANSAT Mission Description

PANSAT will be a low-Earth orbit, free-tumbling satellite providing digital, store-and-forward communication using spread spectrum modulation. The satellite structure is composed of aluminum in a spherical configuration with a diameter of about 19 inches. PANSAT can be launched from a Shuttle Get Away Special (GAS) canister, which provides a typical orbit of 480 km at 28.5° inclination. An average of 6 minutes of communication per pass is possible with this orbit, providing brief communication possibilities throughout the day. The satellite is expected to remain operational for two years.

A general user on Earth will be able to connect with the orbiting PANSAT. While connected, the user may send and receive mail that is stored on-board PANSAT, upload and download files, and read spacecraft telemetry. Multiple users will have the capability of communicating with PANSAT during the same window of time over the one physical communication channel. The use of sophisticated protocols allows users to interleave their activities with PANSAT.

The NPS ground station will be used to command PANSAT. It will also be used to gather telemetry not destined for the ordinary user, and to command any one of the spacecraft's subsystems. Furthermore, the NPS ground station will be able to communicate with an on-board experiment module, or future software programs, all of which are still unspecified or not selected. These features give PANSAT the ability to evolve and also correct deficiencies in software.

Amateur Packet Radio

Amateur packet radio is a communication technique that allows high speed and error-free digital data exchange. The amateur radio community has developed a data link layer protocol that fits within the seven layer Open Systems Interconnection (OSI) Reference Model. The data link layer is considered the second level protocol that communicates with the physical level. This protocol layer is responsible for taking a transmission facility (such as a spread spectrum modem attached to a radio transmitter) and producing an error-free link. This is accomplished by sending and

[†] Computer Engineer

This paper is declared a work of the U.S. Government and is not subject to copyright protection in the United States

receiving streams of bits across the physical layer and applying structure to those streams of bits. The data link layer structures the streams of bits into units called frames. The frames are delimited by particular bit patterns called flags^{1*}.

AX.25 Protocol

The AX.25 protocol is a data link layer protocol adopted by the Amateur Radio community from the International Telegraph and Telephone Consultative Committee (CCITT) X.25 data link layer protocol². This protocol is designed to function independently of the physical layer that delivers the signal, and works equally well in either half-duplex or full-duplex environments.

Each frame is composed of several smaller groups of data called fields. The standard Information Frame is shown in Figure 1. The flag fields indicate the beginning and end of a frame. By using a technique called bit stuffing, this bit pattern never repeats within the frame. Amateur call signs are used in the address field to indicate the source and destination of a frame. The control and protocol identifier (PID) fields assist in identifying the type of frame being sent and controlling the connection. The information field contains up to 256 bytes of data. The frame-check sequence is a cyclic redundancy code (CRC) used to detect corruptions by the physical layer.

Information and Unnumbered information (UI) Frames:					
Start Flag	Address Bits	Control Bits	Information no more than 2048 Bits	CRC Bits	Stop Flag
01111110	112-560 Bits*	8 Bits	8 PID Bits	16 Bits	01111110

Figure 1. AX.25 Information Frame

There can be up to eight outstanding frames in a relay sequence. Thus, AX.25 frames are sent in bursts and can be acknowledged by the destination in small link-management frames, or with information frames that are sent back to the source. Currently, AX.25 is the most widely-used data link layer protocol for packet radio within the amateur community. There are also several PACSATs (satellites capable of amateur packet radio) that handle the AX.25 protocol using other modulation techniques (namely narrow-band frequency-shift keying).

User Ground Station

A typical amateur station includes a personal computer, a Terminal Node Controller (TNC), and radio transmission and receiving equipment. There are numerous software applications available providing user interfaces based on

personal preferences. A packet radio station can be used for both terrestrial and space-based communications. Generally, the user employs packet radio for message relays, file exchanges, and telemetry collecting in the case of PACSATs. The decoding of telemetry is often handled after contact with a passing satellite in order to reduce overhead with the short connection time.

In order to communicate with PANSAT, a typical user's ground station will include a personal computer attached to a TNC, that controls a PANSAT-specific spread spectrum modulator-demodulator (modem), which controls the radio transmitter and receiver. The spread spectrum modem is unique to a PANSAT ground station. NPS will provide software to run on the personal computer for a simple-to-use user interface that includes telemetry decoding. PANSAT will attempt to operate higher level communication protocols that have been developed within the Amateur Radio community.

PANSAT Implementation

The spacecraft exterior has 17 solar panels and four dipole antennae. Within are two battery boxes, the electrical power subsystem, a fully redundant digital control subsystem, a redundant communication subsystem, and a possible experiment payload. The following discussion of the PANSAT implementation focuses on the digital control hardware and software of the satellite.

System Architecture

The PANSAT computer system architecture is best viewed using a computer architecture model that incorporates both the hardware and software layers. Such a model places software tasks, which provide user services, on the top of the architectural hierarchy with protocol handlers, an operating system, and hardware as lower layers. Figure 2 shows the hierarchy of the hardware communications equipment, operating system, AX.25 protocol software, and other software tasks for both the spacecraft and ground systems.

Hardware Design

The digital control subsystem (DCS) supports an operating environment for the satellite software³. The DCS also provides control of the communications and electrical subsystems, an interface with the experiment payload, access to sensors for telemetry gathering, and a mass storage system to organize user mail, files, and telemetry. The DCS is implemented with redundant control boards that communicate via a shared status register. Upon failing to update this register on a regular basis, the secondary DCS control board will assume responsibility of spacecraft operations, disabling the primary control board. A ground command may also initiate the second control board to assume responsibility.

On each control board, system ROM and RAM are implemented in pairs, providing another layer of redun-

* Ref. 1, Section 4.1

Figure 2. PANSAT Computer Architecture

dancy. Upon hardware reset, the ROM and system RAM pairs will switch, allowing the alternate pair to operate. This design can prevent a system ROM or RAM failure from permanently disabling the operation of the spacecraft. However, this pair switching upon hardware reset implies that successive resets will alternate the functional hardware with the failed hardware. Nevertheless, the failed hardware will be unable to update an internal timer, causing a switch back to the operational hardware.

The M80C186XL microprocessor has been selected for several reasons. This microprocessor consumes low power, has tolerance to radiation, and integrates many common digital peripherals which reduces the design complexity. Also, the M80C186XL will provide a software environment that is object code compatible with PCs which are used for PANSAT software development. Most importantly, the M80C186XL supports a special preemptive multi-tasking operating system, the Space Craft Operating System (SCOS)⁴, and a companion product which implements the AX.25 protocol, BekTek AX.25 (BAX)⁵.

The mass storage unit is contained within the DCS and serves as a permanent storage mechanism for user mail, files, telemetry, and storage for updates to the system software. This mass storage will be implemented with either an array of static RAMs which will be backed-up by battery, or Flash memories. Flash memories have the advantage of requiring no power to maintain a written state as well as an erasure life of at least 100,000 cycles. Currently, the Intel Flash File memories are being considered since they will be available in a military package. However, the behavior of these devices in a radiation environment is unknown and must be better understood.

Software Design

The goal of software design for PANSAT is threefold. The first is to allow for maximum use of existing software implementations with reliable functionality. The second requires that software modifications, as well as new software modules, can be uploaded to and exercised on PANSAT after launch. The third goal is that software developed at NPS must be compatible with Amateur Radio software and standards already in use.

Computer Aided Software Engineering (CASE) tools are assisting in simplifying the software development process by providing software diagramming techniques and strict methods of software module interface design. These tools aid in visualizing the software design, gaining confidence that the functional requirements have been satisfied. CASE tools also assist in presenting software designs to others not involved with the software development process. The PANSAT software design group uses Evergreen's EasyCASE Plus which is available on the PC.

Certain low-level functions will be coded in 80186 assembly language in order to maintain strict timing requirements. To assist in generating initialization code for each peripheral embedded within the microprocessor Intel's ApBuilder is used. ApBuilder is a software development tool for embedded processor application programmers. However, the majority of the software will be implemented in C. C is a good choice because of the familiarity the team members possess with the language and also because of its extensive use in embedded systems. This second point is demonstrated because C is used as the language interface for SCOS and BAX. The family of Microsoft C compilers are used for code generation.

Use Of Existing Products And Standards

The ability to use existing software products and standards will be the most important element of success for PANSAT's software development. The challenge for students involved with the design of PANSAT is to first identify and then understand existing products and specifications. Thereafter, the focus is primarily concerned with features that are unique to PANSAT.

Two software components on-board PANSAT will be commercial products: SCOS and BAX. Also, the implementation of a higher level protocol, using AX.25 services for data delivery, is being considered, providing a whole suite of user services. This higher level protocol is called the File Transfer Level 0 (FTL0)⁶, and is implemented on several PACSATs to provide file transfer service. However, FTL0 is not limited to providing only file service; this protocol could easily be used as a mechanism to effectively transfer any type of information in a structured manner.

SCOS has a history of success providing multi-tasking capabilities to several existing PACSATs, including AO-16, DO-17, WO-18, LO-19, UoSAT-3, UoSAT-5, and KITSAT. SCOS provides a standard application program interface designed to assist in the development of multi-tasking applications. These services include a real-time multi-tasking kernel, message passing facilities for inter-task communication, and Direct Memory Access (DMA) and Interrupt driven Input/Output (I/O) drivers. Software development for PANSAT is vastly simplified by using these services. The SCOS services are accessed by using C function calls to the operating system. Thus, all of the inputs and outputs of the service function calls are well defined, and have been extensively used by other operating PACSATs. Some porting is necessary to make SCOS

operate on the PANSAT hardware, but this will be facilitated by the portability of SCOS.

BAX also has a history of success. A companion product of SCOS, BAX has been providing AX.25 services to many PACSATs. BAX frees the PANSAT software developers from implementing the AX.25 protocol. Instead, efforts and attention can be directed toward PANSAT specific issues. BAX runs as a self-contained task on SCOS. Multiple tasks will be able to concurrently use the services of BAX with the assurance that streams of data are sent and received between PANSAT and Earth without error. Like the SCOS software interfaces, BAX services are accessed by using C function calls.

For a task developer, Figure 3 provides a very simplified model of the functions performed by BAX for processing AX.25 frames. This figure is a modified flow chart with actions handled at the task layer denoted by the six-sided symbols. A task can be thought of as the BBS, telemetry request, or other service. All other actions and questions depicted in the figure occur at the BAX layer. As far as a task developer is concerned, the frames that BAX processes are a disconnect request, a negative acknowledgment, a request to connect, and a frame containing information for the task. When a ground station wishes to disconnect, BAX acknowledges this automatically, and then informs the task that the disconnection occurred. BAX also handles the automatic retransmission of outgoing frames (frames that the task has sent to the ground station). This occurs when a ground station has a CRC error with a previously sent frame, and thus indicates this problem by sending a NACK frame. If an incoming frame indicates that a ground station desires a connection, BAX notifies the task which is responsible for accepting or rejecting the request, and then sends the appropriate frame indicating the task's decision.

Figure 3. BAX AX.25 Interface

When an incoming frame contains data destined for the task, BAX first buffers the data. If there are any frames that BAX is required to send to the ground station, an acknowledgment of the incoming frame is placed within the outgoing frame using a method called piggy-backing. Otherwise, BAX just sends an acknowledgment frame. Finally, BAX notifies and passes the data to the task.

The complete FTL0 packet exchange protocol is well defined and will be implemented as a task on PANSAT. This packet exchange protocol will be used to implement file exchange services which are found in PACSAT bulletin board services (BBS)^{7,8,9}.

Software Modifications

The ability to perform software modifications and upload new software modules after launch is a powerful feature of PANSAT's software environment. This is a realistic approach to software development; although errors are not anticipated in the original launch software, not all of the spacecraft's operational scenarios can be forecast during the design process. Previous PACSATs have demonstrated that with the implementation of a simple and well tested boot process, small satellites do not need to rely on the fact that all of the implemented software cannot be modified after the time of launch. This does not mean that attention to details are being ignored at this time. On the contrary, attention to all details imaginable (but in a more limited scope) are being explored right now. At launch, PANSAT may not have all of the user services implemented, but the PANSAT ground station will have the ability to upload a version of the software on-board the spacecraft.

The boot process will consist of the minimum actions required to initialize the necessary hardware on PANSAT so that the spacecraft is capable of communicating with the ground station. The initial method of communication will provide the capability to upload any software component desired, including the operating system.

PANSAT Operational Scenario

PANSAT will remain in a receive mode at all times, ready to acknowledge a request to connect with a user on Earth. Upon a request to connect, PANSAT will grant an ordinary user access to some services. After engaging in the services, the user will then disconnect from PANSAT.

User Services

While remaining in a receive mode at all times possible, PANSAT will be ready to acknowledge a request to connect with a user on Earth. Upon a request to connect, PANSAT will grant the user access to a bulletin board service which offers several services. More services can be implemented in the future.

Orbiting Mailbox

As an orbiting mailbox, PANSAT will store mail for any amateur radio user who desires that a message be

delivered to another. A mail recipient must connect with PANSAT at a later point in time and request to read any messages currently stored. This implements a satellite version of the popular terrestrial network electronic mail (e-mail) systems without the need for wires or for a connection between the sender and receiver of the mail message. PANSAT's mail storage facility is limited in capacity. It will be necessary to remove old, unread messages when free storage is scarce.

File Transfer

File transfers work similarly to the orbiting mailbox concept. However, files sent to and retrieved from PANSAT are not limited to simple text-based messages between users. As a file transfer facility, PANSAT will allow the transfer of any format of file (e.g. binary, or text). These files can contain executable programs, graphics images, encoded voice, or perhaps a word-processed file. There will be a limit to the size, number, and how long the file remains on PANSAT. These limitations will apply to all users.

File transfers are also capable of being transmitted using a broadcast method. This allows multiple users who are listening to PANSAT, but not necessarily connected, to capture file data. This service is currently being explored and may appear as an option for amateur users. However, the types of files to be distributed in a broadcast fashion must be better understood.

Spacecraft Telemetry

Spacecraft telemetry is a popular item that amateur users receive from orbiting satellites. Current spacecraft telemetry will be available to any user who connects with PANSAT. Telemetry will be transmitted in an encoded form so that the number of bits of information exchanged between PANSAT and a ground station are minimized. In order to minimize the connection time, amateurs will perform the telemetry decoding after disconnecting from PANSAT. NPS will provide the information necessary to decode PANSAT's telemetry, as well as a program for the PC that will perform this chore.

Telemetry may also be broadcast when PANSAT is idle. This will make current telemetry available to any user who is currently listening but who has not necessarily connected with PANSAT. Thus, within a region, several users will be able to receive current telemetry all at the same time. This service will also aid in the initial set-up of a PANSAT ground station to verify a communication link.

Management and Maintenance

Management and maintenance of PANSAT will be handled by the NPS ground station, located within the facilities of the Space Systems Academic Group. The hardware will be similar to other amateur radio stations; most of the differences will be in the software.

This ground station will be required to gather and archive comprehensive telemetry from PANSAT. This telemetry will contain values for all on-board sensors, packet radio statistics, mail and file exchange activities, information about on-board tasks, and other characteristics that describe the state of the satellite. The NPS ground station will connect within a three day interval. If this interval is exceeded, PANSAT will stop all services with general users until contact is made with NPS. During this time, PANSAT will be broadcasting its general telemetry to any listening users.

Commanding of the spacecraft is necessary to incorporate new software modules, or to make changes to existing software. The subsystems of PANSAT may also need direct commanding from the ground to force certain functions such as battery discharging, modifying transmitter amplification, or forcing a redundant subsystem component to assume control. PANSAT and the ground station will periodically free mass storage space by purging out-of-date mail and files. A command interface is also needed to communicate with a possible on-board experiment module.

Summary and Conclusions

PANSAT has already proved a useful tool for educating students in the development of a small satellite capable of digital store-and-forward communication using spread spectrum techniques. While cooperating and learning from the abundant work produced by the amateur radio community, the design of PANSAT software has been simplified. And, at the same time, software reliability has increased since PANSAT will incorporate proven components and specifications. In the future, PANSAT will be a facility for students to understand and exercise satellite operations.

References

- 1 *Computer Networks*, Second Edition, Andrew S. Tanenbaum, Prentice-Hall, Englewood Cliffs, NJ, 1988.
- 2 *AX.25 Amateur Packet-Radio Link-Layer Protocol*, Version 2.0, T. L. Fox, Amateur Radio Relay League, Newington, CT, October 1984.
- 3 *Petite Amateur Navy Satellite Spacecraft Digital Control System A Hardware Design*, LT. John Ashe, USN, NPS M.S.E.E Thesis, March 1993.
- 4 *SCOS Reference Manual*, Harold Price, BekTek Corporation, Bethel Park, PA, March 1992.
- 5 *BAX Reference Manual*, Harold Price, BekTek Corporation, Bethel Park, PA, March 1992.
- 6 "PACSAT Protocol: File Transfer Level 0", Jeff Ward and Harold E. Price, Proceedings Of The ARRL 9th Computer Network Conference, ARRL, Newington, CT, 1990.
- 7 "PACSAT Protocol Suite - An Overview", Harold E. Price and Jeff Ward, Proceedings Of The ARRL 9th Computer Network Conference, ARRL, Newington, CT, 1990.
- 8 "PACSAT File Header Definition", Jeff Ward and Harold E. Price, Proceedings Of The ARRL 9th Computer Network Conference, ARRL, Newington, CT, 1990.
- 9 "PACSAT Broadcast Protocol", Harold E. Price and Jeff Ward, Proceedings Of The ARRL 9th Computer Network Conference, ARRL, Newington, CT, 1990.

AIAA 93-4211

**NAVY EDUCATION THROUGH
AMATEUR RADIO SATELLITE DEVELOPMENT**

J. A. Horning

Naval Postgraduate School

Monterey, CA

**AIAA Space Programs
and Technologies Conference
and Exhibit**

September 21-23, 1993 / Huntsville, AL