TU Sat 1: A Novel Communications and Scientific Satellite

W.C. Holmes, Josh Bryson, Brent Gerig, Jacob Oehrig, Jonathan Rodriguez, Jeremy Schea, Noel Schutt, David Voss, Jonathan Voss, David Whittington, Adam Bennett, Chris Fennig, Stefan Brandle, Jeff Dailey, H. D. Voss

Department of Physics
Taylor University
236 W. Reade Ave
Upland, IN 46989
(765) 998-4672
wlholmes@tayloru.edu

Abstract. TU Sat 1 is a multipurpose nanosatellite scheduled for launch on the first CubeSat mission, a 650 km polar orbit. TU Sat 1 provides a low-cost store and forward email communication system for individuals in 3rd world nations. The satellite includes a 115 Kbps primary transceiver and a 1.2 Kbps secondary Ham transceiver. The primary data link uses a 900 MHz COTS data transceiver with a lightweight patch antenna. The satellite 386 CPU on a board processes scientific data, emails and attachments into a 32 Mbyte flash drive. The scientific data is also stored on the 386 before downlink to the ground support equipment. Certain satellite systems are duty cycled to keep the total power usage under 100 watt-hours per day. The main radio link requires a semi-stable attitude, which is achieved through the 30 m gravity gradient tether. In addition, TU Sat 1 measures space plasma density, tether field electrodynamics, and 3-axis magnetic field variations. The technology of TU Sat 1 makes it a good test bed for future solo and constellation missions. Integrated technologies and efficient power control allow TU Sat 1 to be a highly functional satellite despite its small size, 2000 cubic cm, and mass, 1.5 kg. This paper focuses on the design of TU Sat 1 and how these technologies could be used in future low-cost missions.

Introduction

Recent efforts to provide low cost access to space for education and space-based component testing has led to the CubeSat program.[1,2] The CubeSat program provides a standard satellite specification for a 1 kg, 10 X 10 X 10 cm^3 cube.[3] TU Sat 1 will launch on the premier flight of the CubeSat program, under coordination of Stanford University, California Polytechnic State University and One Stop Satellite Solutions. Since its origin in 2000, the TU Sat 1 Program has involved the work of over 40 undergraduate students and five faculty. The TU Sat 1 Program has several goals: demonstrate new technology, educate undergraduates, perform scientific measurements, and meet critical third world needs. This multifaceted satellite is intended to serve as a working prototype that will advance nanosatellite technology and provide a valuable communications service to people in developing countries.

Many of the five billion people in developing countries are being left behind during the Internet and communication revolutions because low-income nations do not have the resources to develop an extensive communications infrastructure. Any current access is concentrated in urban settings and reserved for the wealthy.[4,5] In contrast, the people living in developed countries have nearly unlimited opportunity for communication. This technological division between developed and developing nations has become known as the "digital divide."

Every year, the International Telecommunications Union (ITU) releases an exhaustive collection of statistics and analysis of current telecommunications development worldwide. The 2001 data displays a clear digital divide. For example, the 59 countries with lowest-income economies have less than 1% of their populations using the Internet, while the 48 countries with the highest-income economies have almost 40% of their populations using the Internet. Phone usage is not much better. For example, only 5.5% of Africans subscribe to a telephone service.[6]

The World Bank did a study of the impact and challenges of the network revolution on the third-world.

The study points out that advanced connectivity capabilities can bring a host of benefits, such as improved economic efficiency, strengthened social cohesion, and enhanced education, health care, and public administration.[7]

Satellite technology is the ideal method to provide communication to remote locations over a large area. The satellite industry offers a variety of communication options for individuals, but the cost is often prohibitively expensive. TU Sat 1 is taking advantage of technological advances to prove a low cost satellite e-mail system is possible for the isolated parts of the world.

To demonstrate this capability, TU Sat 1 will begin a communications Beta Test soon after deployment. The year-long test will service up to 50 needy users scattered throughout the third-world. The initial trial audience will include missions and humanitarian workers who currently lack an effective form of communication. Several international organizations will help facilitate this process, including a Sudanese based humanitarian organization.

TU Sat 1 relies on a highly functional design to offer both sophisticated communications as well as scientific data collection. Innovative techniques are required on several levels to produce this level of complexity built into its 1.5 kg mass. In our previous published paper,[8] we focused on the conceptual and mechanical design. Most of the ideas from the previous paper were carried over into satellite manufacture.

The software and hardware of TU Sat 1 is highly modular. This allows for easy integration of new experiments for future missions. For example, the entire power management and communications systems could be reused on future missions. This paper reviews the design of the satellite by first examining the overall system, including instruments, mechanical design, power system, and spacecraft control. The satellite has two main computing systems. One system is microcontroller based and interfaces through a custom made amateur radio system. The other system is a commercial off the shelf (COTS) Intel based system which interfaces with the ground through a COTS spread spectrum data transceiver. The software for these systems and their ground interfaces comprise the balance of this paper.

TU Sat 1 System Development

Figure 1 shows a model drawing of TU Sat 1. Conspicuous in the design are the two deployable solar flaps, the 30 m tether, the plasma probe boom, and the communication antennas. The extended tether mass is used for gravity gradient stabilization while the conducting tether wire enables electrodynamics experiments. The tether measurement devices include a VLF receiver, a sensitive electric field probe, and a variable current source that operates through the tether and the earth's magnetic field to adjust satellite altitude. The VLF receiver is for studying magnetospheric waves since they cannot penetrate through the ionosphere. The electric field probe collects data on both DC and amplified AC variations particularly in the aurora regions and over thunderstorms. The plasma probe measures plasma density and temperature. The email communication system uses the 900 MHz patch antenna on the nadir side of the spacecraft while both the 2 m transmitter and the 70 cm receiver use the same Nitinol wire antenna.

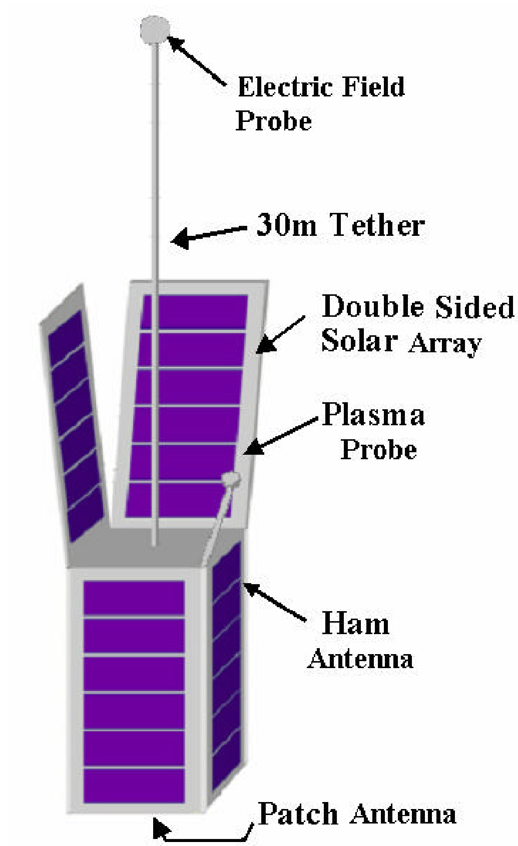

Figure 1 Conceptual Model of the satellite

The system block diagram is shown in Figure 2. The system is broken into three main sections: the instrument package, the computer and power system, and the control and communication system. The computer system uses four microcontrollers working in parallel with a 386 CPU on a board with a 32 Mbyte flash drive. An efficient power distribution system connects satellite components to two independent battery packs and the solar arrays. The magnetometer is used to give satellite orientation (low gain) and science data (high gain). The magnetorquers assist controlling the attitude of the spacecraft. The TU Sat 1 operates with three communication frequencies: a 900 MHz spread spectrum for email and scientific data transmission, a 146 MHz transmitter for diagnostic information, and a 437 MHz receiver for uploading new control parameters to the satellite. The breakthrough in this system is that the many components are included in a 1.5 kg mass budget, have been developed at low cost, and the system design is modular.

Mechanical System

TU Sat1 utilizes an integrated mechanical design allowing for structural integrity while meeting specific design criteria dictated by the CubeSat program.[3] Constructed primarily from 6061 T6 aluminum, TU Sat 1 is a double length CubeSat, outside dimensions of 10 cm x 10cm x 22cm and mass of 1.5 kg. Once ejected from the modular satellite container, the P-POD[9] TU Sat 1 will deploy two solar flaps 10cm x 20cm x 0.05cm providing additional solar cell area. A 30 m tether, can be let out or retracted in the zenith direction to provide stabilization and scientific measurements. The 2m and 70 cm amateur radio antenna, tucked under the solar flap during launch, will unfurl as the flaps are deployed extending 54.3 cm from the body of the satellite. The integrated design of the four walls provides a rigid backing for the solar panels, houses the magnetorquer, provides a Faraday EMI shield, and conducts thermal energy away from the solar cells. Thicker sections of the wall allow components to mount directly to the walls, eliminating the need for mounting brackets. With this design, the printed circuit boards act as shear planes with minimal stress to components. Low weight requirements drove the mechanical design, and the integrated design allows single components to perform many functions, which reduces satellite mass.

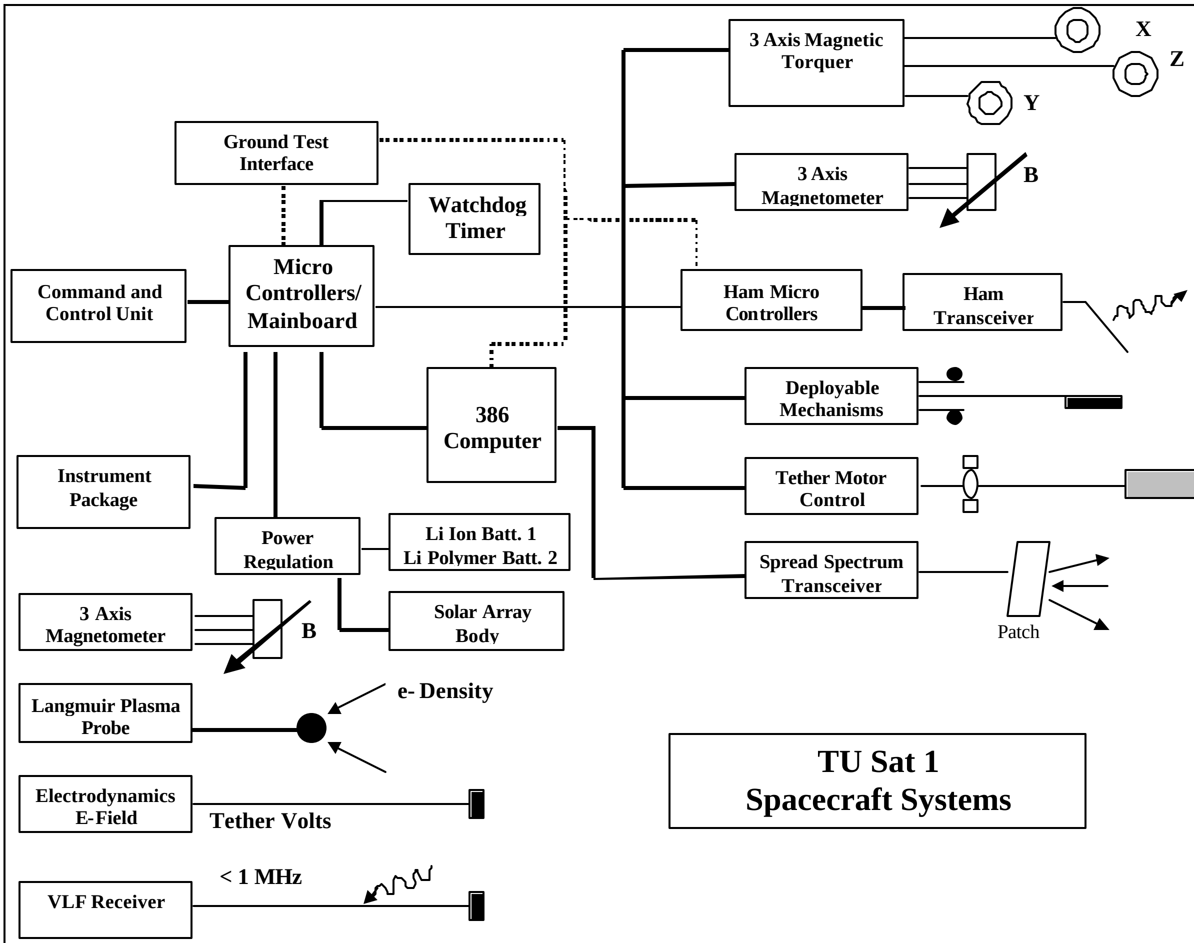

Figure 2 System diagram for TU Sat 1

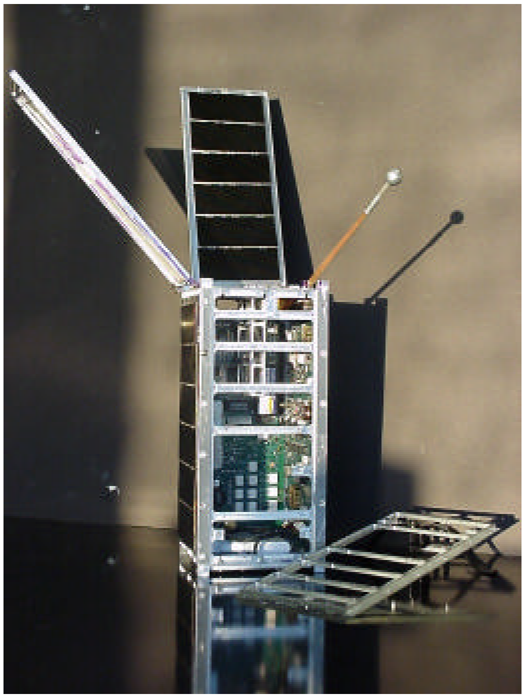

Figure 3. Partially Assembled Engineering Unit

Ball Aerospace performed a preliminary thermal analysis of the satellite. Results from this analysis show that spacecraft component temperatures can vary from -30^{o}C to 40^{o}C. The satellites' components have a temperature tolerance that exceeds this range.

Space Craft Control

As with most global communication satellites, spacecraft control and orientation is mandatory in order to achieve adequate connection with terrestrial stations. In order to improve transmitter efficiency, we use directional antennas, which must be pointed towards earth. We achieve attitude control by using a permanent magnet, magnetorquers, and a 30m Nitinol gravity gradient tether.

Once released from the P-POD, TU Sat 1 will be in a random tumble. Attached to the tip of the tether is a small, permanent magnet. Because of the earth's magnetic field, this magnet's pole will point earthward; therefore the antenna will point earthward when the satellite is in the northern hemisphere. This preferred orientation; along with eddy current dampening gives the satellite the correct orientation over the northern hemisphere. We will analyze magnetometer data returned by the beacon to determine the proper time to deploy the gravity gradient tether. We will then command the satellite from the ground to deploy the tether at a specific time. The tether will provide a one axis stabilized satellite with the satellite antennas pointing towards earth.

The 30m tether gives the satellite the needed axis of stabilization by using the gravity gradient between the satellite and the extended tether. The tether wire is composed of three strands of Nitinol coated with Teflon. The high tensile strength of Nitinol makes the tether come off the spool straight. The Teflon coating gives the tether wire electrical insulation and mechanical dampening.

The tether deployment mechanism, Figure 4, makes use of two rollers, a tension arm, two stepper motors, and a 2 inch spool, capable of holding 150 m of the Nitinol wire. The stepper motors (Faulhaber Series AM 1524[10]) are 2 phase and have 24 steps per revolution. Two motors are needed: one for deploying the tether and one for retracting the tether.

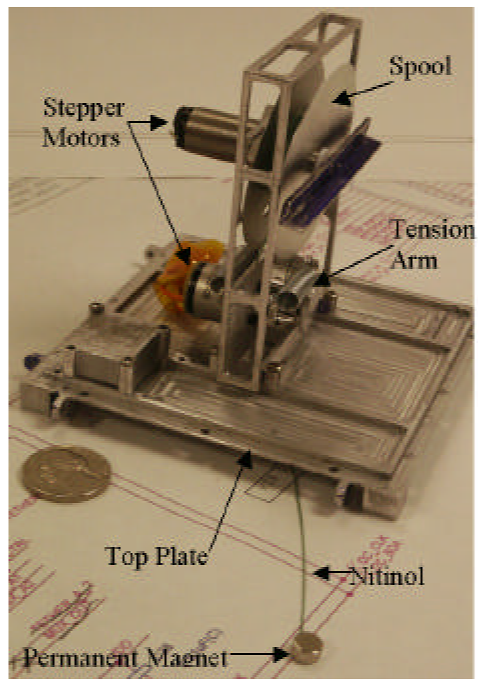

Figure 4 Tether deployment mechanism

As an alternative to the permanent magnet and eddy current stabilization, TU Sat 1 has 3 axis magnetorquers built into the satellite walls, Figure 5. We can activate these magnetorquers from the ground to provide initial stabilization or to perform spacecraft dynamics experiments in orbit.

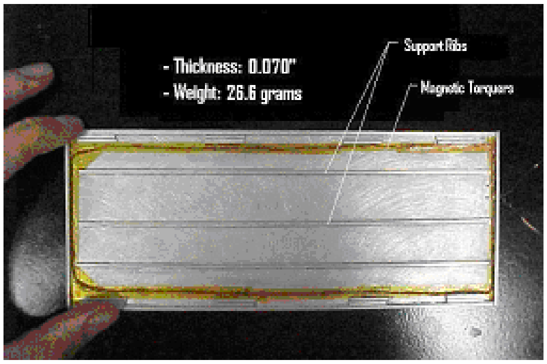

Figure 5 Satellite Wall with Magnetorquers

Power System Design

The power system is designed to efficiently distribute and regulate power. This task is especially important for a CubeSat because there is a limited surface area for solar cells. Another design goal of the power system is protecting the satellite from single point failures and providing short protection.

Figure 6 shows a simplified schematic of the satellite power system. Eight parallel solar panels, each with six Spectrolab 19% efficient Gallium Arsenide cells[11] connected in series, provide primary power for TU Sat 1. As shown in Figure 3, four panels attach to the main body walls and four attach to both sides of the two wings. This arrangement gives approximately two solar panels in full sunlight during the light phase of our orbit.

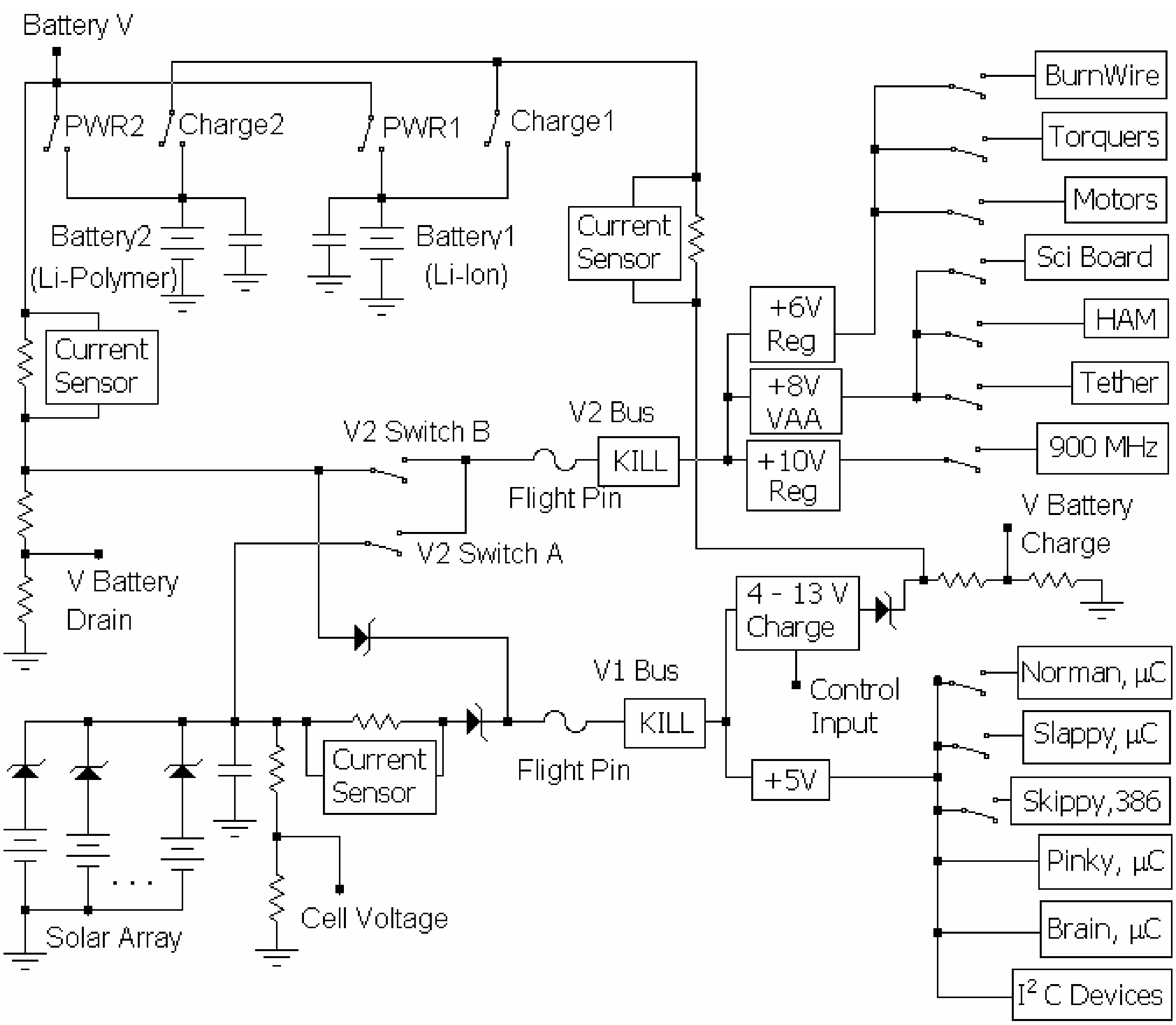

Figure 6. Power System Flow Diagram

Secondary satellite power comes from two parallel Lithium technology battery packs. One is a Lithium – Ion battery (Panasonic CGA103450[12]) chosen for its high energy density and reliability. The other is a Lithium polymer battery (Ultralife UBC525085[13]) chosen for its exceptional energy density and high current capability. The polymer battery also attaches effectively to the satellite frame because of its thin size and form factor. Table 1 lists voltage, maximum current, and capacity of the satellite's power systems. Note that this table lists the maximum current for a single solar panel. There are approximately two panels illuminated at any given time which doubles the current flow into the satellite.

The power from the solar cells and batteries flows into two separate unregulated buses, V1 and V2. The key difference between these two busses is that V1 is always connected to either the solar cells or (during night) the batteries whereas V2 can be completely turned off for both power conservation and short protection.

These unregulated lines feed four regulators (Micrel mic4681[14]) and one boost converter (National Semiconductor LM2621[15]). The always-on bus, V1, feeds the 5V regulator, provides power to all digital systems, and the variable-voltage, battery-charging regulator. We actively control the charging voltage to maximize power from the solar cells into the satellite and correctly charge the batteries. A microcontroller controlled digital potentiometer on the feedback pin of the regulator varies this voltage. The firmware for this feature is described in the microcontroller section. The 5V regulator is on the V1 bus since the satellite must have VDD to function. The variable voltage charging regulator is on the V1 bus since it must run from the solar cells.

In addition to the two voltages mention above, there are three other voltage regulators on the system. TU Sat 1 uses the 6V regulator primarily for deployment. This regulator gives short protection to the rest of the satellite. In addition, the deployment system is on a separate PCB from the rest of the system.

The voltage from the charging regulator is adjustable to provide optimal battery charging and solar cell power output. Because of the special nature of this regulator, TU Sat 1 needs a separate 8V supply for the amateur radio and the power amplifiers.

The last regulator is a boost regulator for the spread-spectrum radio. This radio needs 9 volts to transmit at full power. In addition, the radio requires more current than the solar cells can provide, and must be supplied by the batteries, which give only 8.4 volts.

The three above mentioned regulators are duty cycled to conserve power, so were placed them on a separate unregulated buss, V2. Having two unregulated buses allows the solar cells to operate at full voltage, 12.2V, while the batteries are supplementing solar power.

As a final step before power distribution to the various subsystems of TU Sat 1, an array of solid-state relays can turn off power to each subsystem independently. Pinky, the power management microcontroller, which we describe in microcontroller section, controls each of these relays to achieve the design goals of power management system.

Short protection is achieved by having Pinky monitor current and detect when an over-current condition exists. If a short occurs, the entire satellite is powered down. Since the VDD voltage falls below the microcontroller brown out voltage, all regulators and switches open. This removes the short and allows power up. At power-up, only those components connected to VDD receive power. Pinky detects this reset condition and subsequently performs a system check to isolate the short. By turning components on

Power Source	Description	Voltage	Current	Capacity
Battery Pack 1	Lithium Ion	8.4 V	1.4 A	1400 mAh / 11.76 J
Battery Pack 2	Lithium Polymer	8.4 V	3.0 A	1500 mAh / 12.9 J
Solar Panels (1 String)	Dual Junction GaAs / Ge	12.2 V @ 20 C	Min 273 mA	

Table 1. Power Sources for TU Sat 1

one at a time, Pinky determines the short cause, isolates the questionable system and reports this information to the ground. The ground must command Pinky before the processor will turn on that subsystem again.

Science Data

Magnetometer

TU Sat 1 utilizes a small (2.52 cm X 2 cm) 3-axis magnetometer (Honeywell HMC2003[16]) capable of detecting magnetic fields from 40 micro-gauss to ±2 gauss at 1V/gauss. The magnetometer has internal amplifiers that allow low-noise measurements from DC to 1kHz. Data is taken on the DC magnetic field as well as the AC variation superimposed upon the DC signal. The AC signal is amplified allowing for detection of faint variations within the DC field which allows analysis of the ionosphere crosscurrents within the earth's magnetic field. Data from the magnetometer receives a time stamp and is compared to data received from the plasma probe and electric field probe. Using data from the magnetometer we determine attitude by comparing the DC value of the x, y, and z axis to corresponding know areas of the earth's magnetic field.

Data is acquired at 50 Hz with burst mode sampling near 1 kHz. The sampling rate decreases as system bandwidth is transferred to other functions later in the life of the satellite.

Langmuir Plasma Probe

The Langmuir plasma probe measures plasma density and temperature. The data collected assists in the study of ionosphere, including auroral regions, equatorial fountains, plasma waves, and other plasma phenomena. A half-inch solid aluminum sphere that is suspended approximately 4 inches above the satellite takes the measurements. The spherical design of the probe allows omni-directional measurements. In addition, the probe points in the ram direction to improve the data gathered. There is a voltage applied to the probe, which can be ramped between 0 and +8 V to measure the electron temperature.

The data collection rate for the plasma probe is variable, depending on storage space and downlink time. The average data rate is around 20Hz with possible burst rates of up to 1kHz. This allows higher resolution study of certain areas of interest. Data collected is stored with 8-bit precision. Future plans include development of a constellation of satellites containing these probes in order to make multi-point measurements in similar areas simultaneously.

Electrodynamic and E-Field Tether

In order to reduce manufacturing costs, most parts of the satellite have multiple uses. For instance, the tether acts both as an attitude control device as well as an electric field data collector. An advantage to using the tether as an E-field detector is the large distance from the satellite, resulting in very accurate E-field measurements.[17]

A voltage-follower op-amp is used for high input impedance to the E-field analog to digital converter. A power MOSFET is used to control current in the tether. Instead of only measuring the natural current in the tether, the satellite can actually send a current through the tether. This current will interact with the magnetic filed lines of the Earth and cause the probe's orbital velocity to either speed up or slow down depending upon current direction. This causes the satellite's altitude to either increase or decrease respectively (See Figure 7).

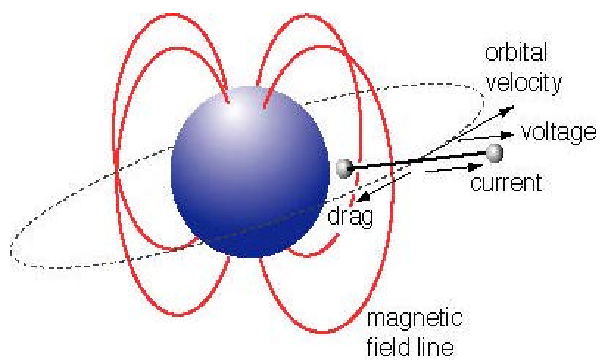

Figure 7. Electric field lines interacting with Electrodynamic tether [18]

VLF Receiver

Magnetospheric very low frequency (VLF) radio waves in the kHz range are impossible to measure at the earth's surface since they cannot get through our top ionosphere and are refracted and reflected by electron density. Moreover, these radio waves have long wavelengths and thus require large or expensive antennas to receive them. Conveniently, both of these problems are solved using the tether system.

The 30 m Nitinol tether is electrically conductive causing it to act as a very long monopole antenna attached directly to the main body. The tether antenna covers a much broader range (10 Hz to 1 MHz) than just VLF frequencies, however, its length makes it well suited for a VLF receiver.

Processor Name	Processor Type	Clock Speed	Memory	Purpose
Skippy	Intel 386	25 MHz	512 K and 32 Mbyte Flash Drive	Stores scientific and email data for download to the ground station, and interfaces to the ground station.
Brain	Microchip 16F877	4 MHz	8 K Program 368 Bytes Data 256 Bytes EEPROM	Records scientific data from the magnetometer, plasma probe, electrodynamic tether, and temperature sensors.
Pinky	Microchip 16F877	4 MHz	8 K Program 368 Bytes Data 256 Bytes EEPROM	Handles all power management functions including software short protection and recovery, monitoring power level and battery charging.
Slappy	Microchip 16F877	4 MHz	8 K Program 368 Bytes Data 256 Bytes EEPROM	Interfaces with the ground through the amateur radio. Gathers health and diagnostic data from other processors and sends this information out on the beacon. Also handles the satellite command link.
Norman	Microchip 16C773	4 MHz	4 K Program 256 Bytes Data	Deploys the tether and gives extra digital output pins.

Table 2. Processors on TU Sat 1

By AC coupling the same signal used in the electric field probe (described above) and using a high input impedance J-FET along with a double stage amplifier, the AC signal can easily be digitized and stored with relatively small amounts of noise error.

Microcontrollers

Processor Hardware Description

Table 2 lists the processors on TU Sat 1. There are five processors on the satellite: four COTS microcontrollers from Microchip[19] and one COTS single board computer[20] with an Intel 386 processor. Skippy, the single board computer, serves the email and high-speed data link system, which we describe in the next section. In this section we describe the hardware and firmware of the microcontrollers which are responsible for power management, data handling, and communication with the ground station through the amateur radio.

All our microcontrollers are from Microchip's midrange series. We decided on Microchip microcontrollers early in the conceptual design phase because of their general purpose nature, widespread use in both ground and space applications, and the familiarity of these processors to one of our authors. The microcontrollers have hardware support for serial communications, event timers (used to trigger data collection), watchdog timeout (used to protect against single event memory failure), eight inputs for Analog to Digital conversion, on chip EEPROM, pulse width modulation (used for scientific data) and in-circuit serial reprogramming. TU Sat 1 employs all these features.

Initial satellite designs called for one microcontroller; however, to enable the functionality desired in TU Sat 1, a multi-processor system soon became necessary. The decision to switch to a multi-processor environment was driven by the need for more digital outputs and our desire to have each processor dedicated to a specific task (Table 2).

As shown, Brain gathers scientific data, Pinky provides power management, Slappy sends beacon data and interfaces with the ground, and Norman deploys the tether and controls additional digital switches needed for deployment. While each processor has a specific task, the processors are also inter-related. For example, Slappy needs scientific data from Brain to send on the beacon, and Brain needs power status information from

Pinky to know if there is sufficient power to take scientific data.

The microcontrollers share information since they are all on the same Inter-Integrated Chip Bus[21,22] (I^2C), as shown in Figure 8. There is also an RS232 serial link between one microcontroller, Brain, and Skippy so that data collected by the microcontrollers can be stored locally for high-speed download. Also on the I^2C bus is 256 Kbytes of shared EEPROM (Atmel AT24C512[23]), two temperature probes (National Semiconductor LM83[24]), a real time clock (Xicor X1226[25]), and a digital potentiometer (Dallas Semiconductor DS1803[26]). Since all these devices are on the I^2C bus, any processor can access any attached device. For instance, Brain, Pinky and Slappy all access the real time clock. With the I^2C, there only needs to be one real time clock for the entire system.

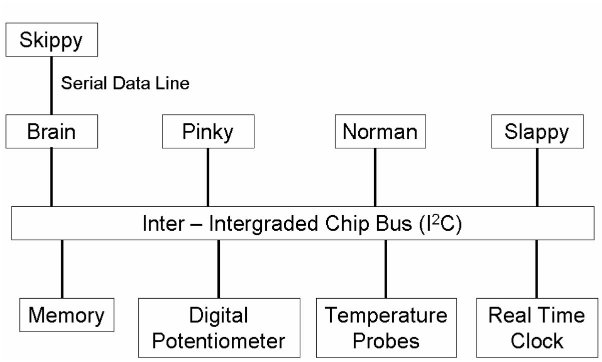

Figure 8 Hardware Diagram of Satellite Processors

The microcontrollers communicate with each other using an interface similar to memory access protocols. Each processor can directly read or write to another processor's memory. Each processor can also execute subroutines on the other processor. For instance, we could command Pinky to immediately execute the power analysis code via Slappy from the ground.

The microcontrollers must also work together to get data to Skippy's high-speed data link. For example, Pinky records power data, and needs to transfer this information to Skippy. This is one use of the common memory. Pinky can write data to the memory, and then command Brain to get that data from common memory and transmit it to Skippy for later download to the ground.

If we were to develop each processor independently, it would be a monumental task. To simplify development, we used a highly modular software design, especially for a microcontroller. Much of the code is used directly in two or more microcontrollers, and relatively few changes are needed to add new functionality or move functionality from one microcontroller to another. This greatly simplifies future mission development in addition to current development. If a future mission measures a different type of scientific data, the only software change needed is to add a module to Brain. In the following section, we describe this modular design.

Microcontroller Software Description

Many software modules are shared among Pinky (Figure 9), Brain (Figure 10), Norman (Figure 11), and Slappy (Figure 13). The primary function of Pinky is monitoring and regulation of the power system. Pinky maximizes the power flowing into the satellite from the solar cells by varying the battery charging voltage. It also monitors battery charge status and changes component duty cycles to protect critical systems. To accomplish these tasks, Pinky uses the hardware shown in Figure 6.

Pinky also queues the power measurements in the off-chip memory. When enough data is gathered, Pinky sends a signal to Brain. When Brain receives this signal, it transmits this data to Skippy for later transmission to the ground. Brain then returns a signal indicating the success or failure of the transmission.

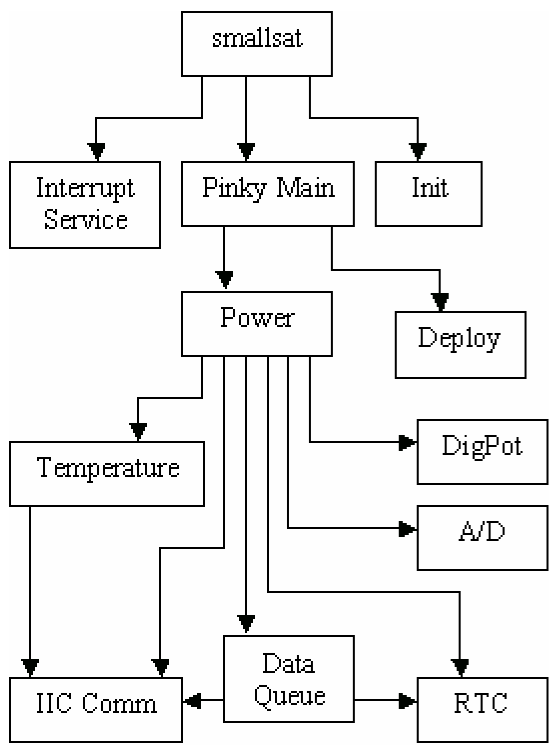

Figure 9 Software Module Diagram of Pinky

The primary function of Brain (Figure 10) is collection and transmission of scientific data. The two types of data collected are plasma probe and magnetometer data, although the modular design of the software allows the easy addition of other data collection devices. Each plasma probe sample consists of a voltage and temperature, and each magnetometer sample records the strength of the magnetic field on the x-axis, y-axis, and z-axis. All inputs except for the temperature are represented as analog signals. The analog signals are digitized with the embedded A/D converter before storage.

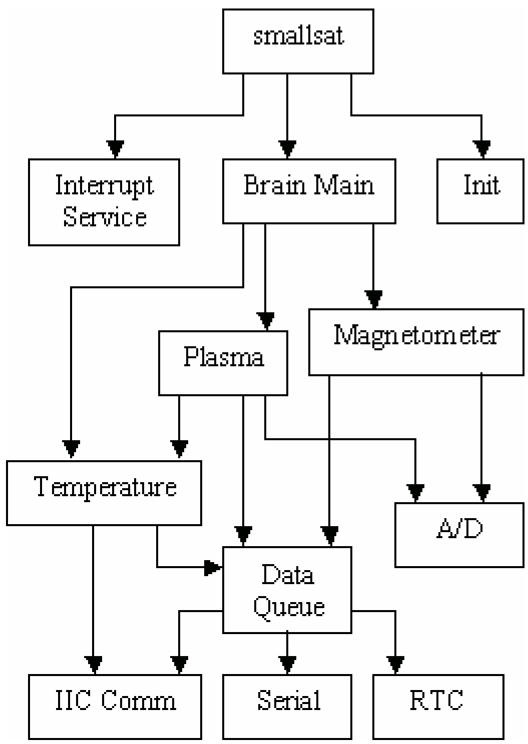

Figure 10. Software Module Diagram of Brain

The plasma data and magnetometer data are stored in separate queues in the EEPROM. These two data queues function identically to the two queues on Pinky. Whenever one of the queues reaches a certain size, transmission to Skippy is attempted. When duty cycling of the spread radio is active, data transmission only occurs when the radio power is off to avoid interfering with email transmission time.

Early in the design process, TU Sat 1 had a complicated deployment sequence that required multiple burn wires. There were not enough digital IO pins from the processors for the deployment. We then decided to add Norman, a microcontroller dedicated to deployment. In the final design, Norman controls two burn-wires and two stepper motors. Pinky completely controls this processor through the I^2C bus, which gives Norman a very simple software design (Figure 11). The motor module steps the two phase motors to the specified location set by an I^2C message from Pinky.

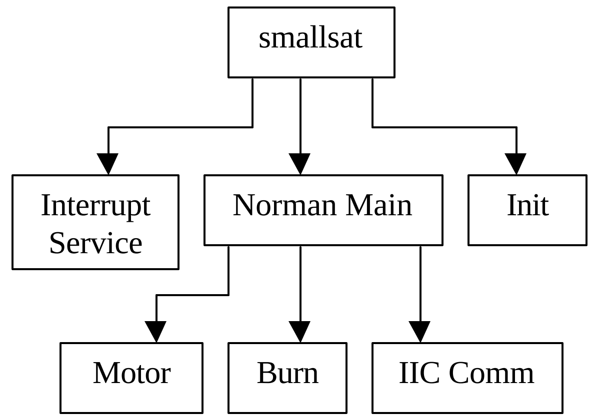

Figure 11. Software Module Diagram of Norman

Amateur Radio System

For a reliable beacon, health, and control data link, TU Sat 1 includes a low speed (1200 Baud) amateur radio system, Figure 12. This system is responsible for transmitting beacon data as well as satellite health data, and is the control link for the satellite. This system includes:

- A terminal node controller (TNC) microcontroller, Slappy (see Table 2)
- A direct Frequency Shift Keyed (FSK) synthesized 2 m transmitter
- A single-chip FSK 70 cm receiver with built-in data demodulator

Terminal Node Controller

Because of the inadequacy and power overhead of commercial systems, we decided to make our own custom TNC, which we named Slappy. This microcontroller is dedicated to interfacing with the ground through the amateur radios. Slappy automatically sends four beacons every two minutes, handles AX.25 packet protocol for the beacon, parses the command messages sent from the ground and sets the phase locked loop (PLL) frequency.

Making a custom TNC saves space, weight, and power, and allows us to place the TNC microcontroller directly on the I^2C bus. This means that Slappy can execute commands from the ground that directly access other devices on the I^2C bus. Since it is an in-house programmed microcontroller, it can also handle the beacon timing. Finally, a custom TNC allows Pinky to be completely dedicated to power management.

Figure 13 shows the software module diagram for Slappy. The beacon module pulls data directly from the real time clock, the temperature sensors, Pinky, and Brain. The beacon module passes this information to the AX.25 module which places the data into AX.25 unnumbered information frames (UI). This data is then passed to the radio through the serial module and port. By using standard AX.25 for the beacon, we allow anyone with a 2 m receiver and TNC to listen to our beacon signal.

Because the command link is only used by the control station at Taylor University and we desire the packet size to be as small as possible for reliability reasons, we decided the command link would use a custom packet protocol. Our custom packet protocol has only seven bytes of data and two bytes of overhead. Using the full eighteen overhead bytes of AX.25 UI frames would have significantly increased packet size and therefore would have decreased link reliability.

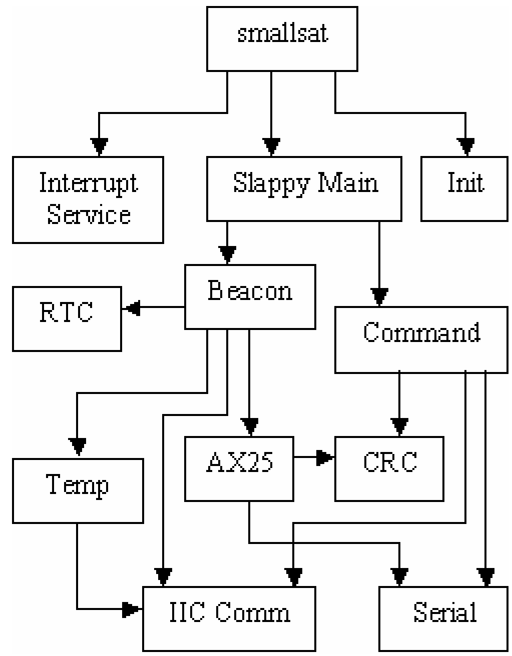

Figure 13 Software Module Diagram of Slappy

Linking Slappy, the satellite TNC, and the ground support equipment is a custom designed amateur radio receiver and transmitter. Because the command link is mission critical, the design emphasizes high reliability. This emphasis dictates the use of a simple design with a low speed data link.

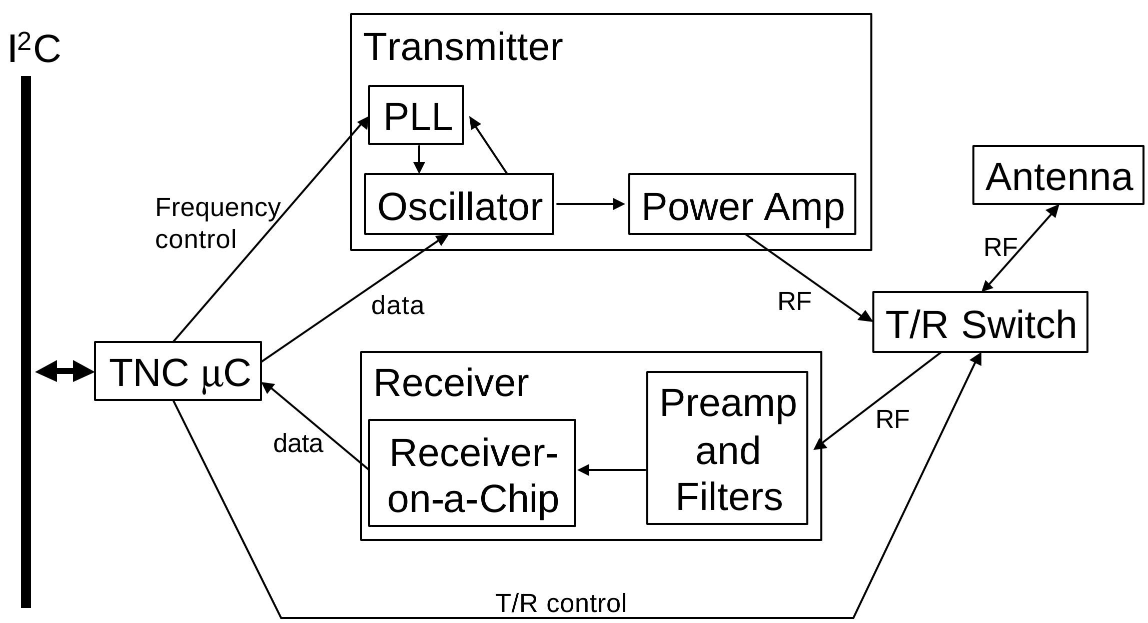

Figure 12. Amateur Radio System Diagram

Radio Hardware

As detailed by Oehrig[8], TU Sat 1's initial design was a modified COTS TNC with a handheld transceiver. Unfortunately, the COTS equipment we considered did not have the required reliability, and would have been difficult to modify for space flight. After evaluating several options, the only viable solution was to design our own radio and TNC. By designing our own radio, we could optimize the system for our specific frequency and simplify the radio design. This simplification increases reliability.

The transmitter is built around a series tuned Colpitts oscillator. A programmable single-chip PLL and a temperature compensated crystal oscillator stabilize the radio oscillator. The microcontroller applies modulation directly to the radio oscillator.

The satellite's entire amateur radio system (TNC, receiver, and transmitter) fits on a single PC-board. This allows efficient use of space and power. Because only one board is used, components can perform multiple tasks and interconnect components are eliminated. The simple design reduces possible points of failure.

By using direct FSK instead of AFSK, which is more common in amateur radio systems, we are able to simplify the transmitter design. Instead of sending the TTL data to a modem and transmitter audio section before the modulator, we send the data though a filter directly to the modulator. The TNC output passes through a filter so that the signal is not a square-wave. This keeps the transmitter harmonics down.

The amateur radio receiver uses a receiver-on-a-chip. By using a single-chip receiver, we are able to decrease the parts count and complexity of the radio. The single chip receiver gives good sensitivity and selectivity, a wide temperature range (-40 to +86 °C), and low power requirements (<6mA). The low external parts count is an additional benefit. The IC provides a built-in data filter and data demodulator. Along with using direct FSK modulation, this eliminates the need for an external modem chip.

The amateur radio antenna is a quarter-wave 2 m monopole antenna. This simplicity of antenna design allows it be easily integrated with the double sided solar array shown in Figure 1.

Ground Support Equipment and Satellite to Ground Communications

Our primary ground station is located at Taylor University. It is the satellite control and monitoring

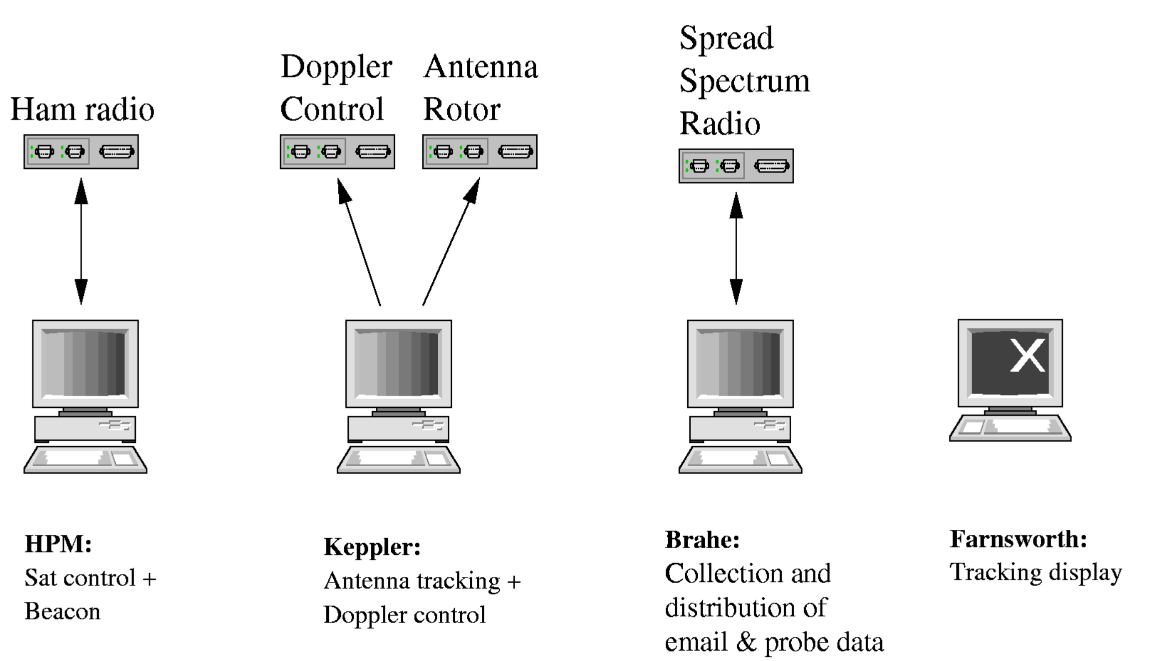

Figure 14. Primary Ground Station Modules

station as well as the processing point for all communication with the satellite. This includes email transfer to and from the satellite and scientific data retrieval from the satellite. The ground support equipment consists of several standard Intel systems running Linux.

The ground station itself consists of four modules (Figure 14):

- Control and monitoring
- Tracking and radio control
- Visual orbital display
- Email and scientific data collection and distribution

Control and Monitoring

We need to monitor and control satellite settings after launch. General monitoring health and safety information is obtained via the beacon. Our primary control mechanism is a flexible interface that allows us to read or write any memory location on any of the microcontrollers or any device on the I^2C bus. Both of these are accomplished via the 1200bps amateur radio link.

The satellite has a two-minute beacon packet transmission cycle during which it alternates between transmitting and listening for one minute, then powers down the radio for one minute. The beacon consists of a series of packets which contain instantaneous readings of power, temperature, and clock values, as well as a recent history of the magnetometer, power and temperature data. Figure 15 is a screenshot of the ground support software that displays the beacon data.

Figure 15 Display of beacon data on ground support equipment

The command mechanism allows direct access to the I^2C bus, thereby granting access to any memory location on any processor. This allows explicit control over virtually all of the satellite subsystems. The user interface to the control system allows multiple commands to be queued for when the ground station enters the satellite footprint.

All of this is done by a client-server pair of programs installed on HPM, our control machine (Figure 16). The control server, *Controld* (control daemon; daemons are continuously running background processes), for two purposes: 1) it transmits commands to the satellite, and 2) it receives, interprets, and stores the beacon data. When the command station receives a beacon, the station checks the command queue and transmits any commands.

The client program, *Control*, provides the user interface to issue commands to the satellite and to view beacon data. The benefit of the client-server design is that there can be multiple clients.

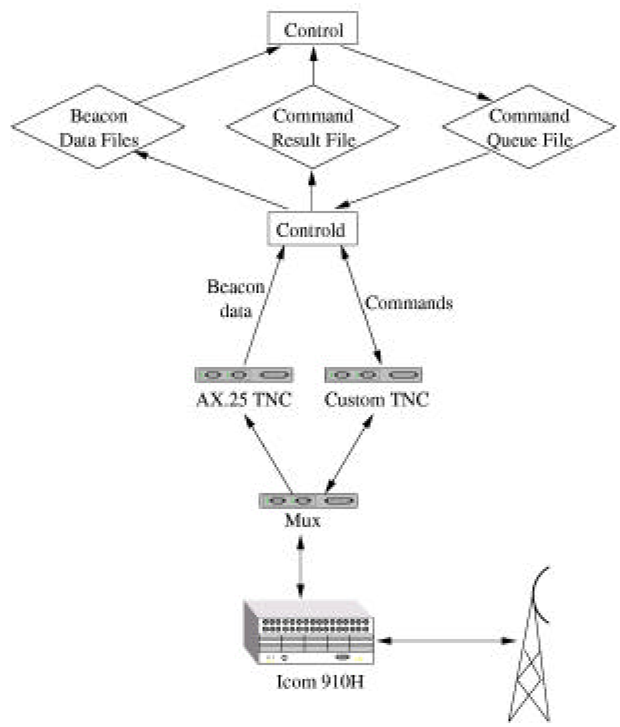

Figure 16. Command and Beacon Link Hardware

Tracking and radio control

In order to compensate for the relatively low power transmission from the satellite, we need high gain antennas at the ground station. This requires careful positioning of the antennas, including incremental antenna adjustment as the satellite moves overhead. We also need to compensate for the Doppler shift in the frequency on the amateur radio channel. We do this by making incremental frequency changes.

An open source software package, *Predict*[27], handles most of the calculations for us, including azimuth, elevation, and Doppler shift. We are using *Predict* in its server mode. In this mode, it makes the results of its calculations available to client software over the network. A custom-written *Predict* client reads tracking and Doppler data from the server and issues the necessary rotor movement commands as well as the frequency updates to the amateur radio. It also is responsible for power up and power down of the antenna rotor and radio. Another program periodically updates *Predict's* TLE data with current orbital data from NORAD.

Figure 17. Earthtrack Display Close-up

Visual orbital display

The visual orbital display is performed using another *Predict* client program, *Earthtrack*[28]. This program runs on Farnsworth to show satellite footprints and orbits, including TU Sat 1 (Figure 17).

Email and scientific data collection and distribution

Initially email sent to or from the satellite passes through Taylor's ground station. All email sent to a user of the satellite is initially sent over the Internet to Taylor, where the email gateway packages and queues it for delivery to the satellite. Similarly, the satellite delivers email from the remote users to the ground station. These email messages are then resent over the Internet using a table that maps the Internet email addresses to the remote station identifiers. As email comes in from the Internet, we filter, process, batch, and queue it for the next satellite pass. Email from the satellite is un-batched, processed, and delivered to the intended recipient via the email gateway.

Besides email, we also offload the collected scientific data as files from the satellite during each pass over the ground station. These include scientific magnetometer and plasma sampling data as well as satellite health information such as temperature and power logs.

High Speed Spread Spectrum Link

In this section, we describe the COTS high speed spread spectrum link. We first describe the hardware, then the software that allows this link to function.

Hardware

The spread spectrum radio provides a serial link between the satellite and the ground. Figure 18 shows the hardware block diagram for this data link. The microcontrollers can dump data to the ground through the spread spectrum system. This allows a much higher data rate than could be achieved with the amateur radio system alone. In the satellite, the serial data is received by Skippy, a COTS CPU on a board (JK microsystems μFlashTCP[29]) with a 32 Mbyte flash drive. This component, Figure 19, has a mass of 55 grams, measures 2.5" by 3.75", and runs a DOS 3.0 environment.

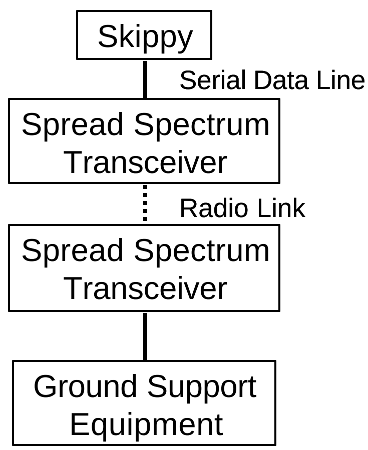

Figure 18 Hardware diagram for the spread spectrum data link

Figure 19 Picture of Skippy, the satellite computer on a board.

Skippy communicates with the ground through a 900 MHz spread spectrum data transceiver (Figure 20). This radio makes satellite to ground communications transparent. The radio provides a 115 Kbps transmission rate and 32-bit cyclic redundancy check with forward error correction. This allows the radio to detect and fix transmission errors without requiring retransmission of the data. Because there is no need for retransmission of data, the radio link provides 115 Kbps throughput.

Another feature of this radio is that it implements a frequency hopping technique (FH – CDMA). One advantage of our radio is that the high frequency of our signal allows a higher data rate than other transmission methods. Another advantage of our system is that it avoids portions of the bandwidth that are noisy. A final advantage of the frequency hopping is that it enhances link security by making it difficult for a third party to intercept the signal.

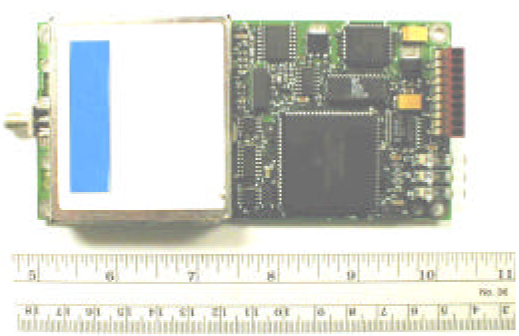

Figure 20 Picture of spread spectrum radio used for high speed communications.

The satellite antenna for the 900 MHz system is a patch antenna designed and built by Ball Aerospace. The directional nature of the antenna, requires the tether stabilization discussed on page 4.

Software

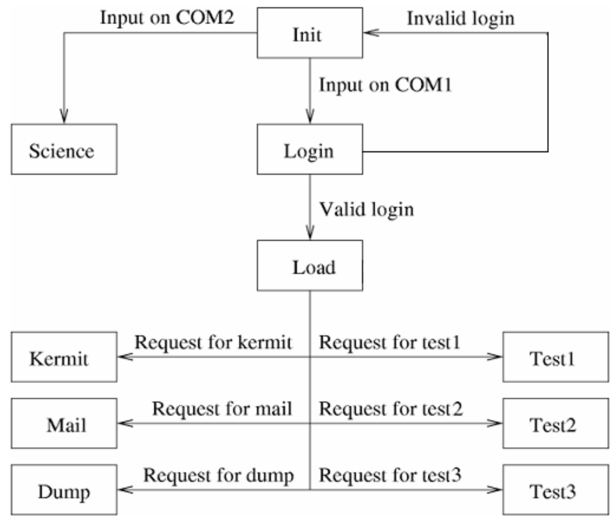

Figure 21 Control flow diagram of Skippy.

Security is important on TU Sat 1. It would be problematic if anyone with a spread spectrum radio could connect and login to our system. The hardware has some built-in protection, but there are also two security measures built into the software system. First, the satellite has a password-protected login system with multiple access levels. Second, with the exception of a Kermit administration session (only accessible with a master password), all data exchanged with the satellite is encrypted using PGP.

Skippy performs three main tasks. It connects with remote ground stations to allow email communication, with Brain to allow the transfer and storage of scientific data, and finally with the primary ground station to allow administration.

To achieve a high degree of simplicity and fault tolerance, Skippy's software is designed to be highly modular. The modular design makes the system less susceptible to complete system failures. The failure of a single module will not bring the system to a halt.

It is also easier to test a modular system because testing can be done on a per module basis, allowing a greater range of tests. Modules can be subjected to extreme inputs, which often uncovers problems in the programs that would otherwise have gone unnoticed. In addition, testing efforts can be focused on critical modules first.

With a modular design it is simple to integrate third-party software with our software. The use of existing software reduces the amount of code that must be written for the satellite and the amount of testing required because software packages that are already in widespread use have been well tested.

Skippy's software system (Figure 21) consists of ten programs, three of which (*Test1*, *Test2*, and *Test3*) are placeholders for future expansion and testing. The remaining seven programs consist of essential system programs (*Init*, *Login*, *Load*, and *Kermit*) and application programs (*Science*, *Mail*, and *Dump*). These programs are supplemented with two third-party utility programs (*Bzip*[30] and *PGP*[31]) that are called by the application programs.

In the future the three placeholders could be replaced with completely new software or upgraded version of any of the already existing software on the satellite. This will allow thorough testing of software on the satellite before any changes are made to the core systems.

The system programs handle critical tasks such as the initial response to COM port input and the loading of various sub-systems. Without these programs Skippy would not be able to function at all.

When Skippy powers on, *Init* loads. Its only function is to wait for input on either COM1 or COM2 and load another program based on which COM port the input was sensed. Once that program finishes, *Init* resumes waiting for input.

Login is the program that is spawned when Init finds input on COM1. Its purpose is to provide a secure way for ground station users to login to the satellite. It uses PGP to perform public/private key cryptography. It sends a request to ground station users for encrypted login information along with a public key to encrypt it. When a response is received *Login* decrypts the information it receives using its private key and performs validation of decrypted login data. At that point, if the user's login information is valid it executes the *Load* program. If the user's login information is incorrect the process is repeated.

Load performs the loading of the various different applications available on the satellite. It sends a prompt to the user asking what subsystem they would like to start. Certain subsystems may not be available based on a user's access level. *Load* then executes the program associated with the subsystem they request.

Kermit is an administration terminal application. It can be started by *Load* if a user has master access. Files can be uploaded to, downloaded from, and manipulated on the satellite using *Kermit*. *Kermit* is not a product of our development efforts. Rather, we opted to use it for our administrative system because it has already been heavily tested, is available for free, and has a robust feature set. It performs error checking and correction on all data transmitted. It also performs windowing and buffering of data during transmission that allows for high throughput when sending data to and from the satellite.

Science is the application that is executed when Init finds input on COM2. It communicates with Brain to retrieve scientific data, performs simple error checking and recovers from any errors that occur by requesting retransmission of the corrupted data. As the data is received it is saved in a file to be transmitted to the ground later.

Mail is used to transfer email to and from the satellite. *Mail* sends any email in the user's inbox and then waits for any mail that the user might want to send. All mail transmissions are compressed and encrypted, but the ground station performs both these functions.

Dump is a data retrieval program. It encrypts, compresses, and sends the data collected on the satellite to the primary ground station. This data consists of email, log files, and scientific data. *Kermit* is used by *Dump* to perform the actual transmission of the files and automatically performs buffering and error correction so that *Dump* doesn't need to perform these operations.

All compression of data being sent from the satellite is done using *Bzip2*. Encryption of information is performed using version 2.6 of MIT *PGP*.

Conclusion

TU Sat 1 demonstrates a sophisticated store and forward email communication system. The low cost of this system makes it ideally suited for use by people in third-world countries. Along with email communication, TU Sat 1 also records several types of data related to space plasma.

This project not only allows students to apply what they are learning, but also challenges them to be creative, perform interdisciplinary work, function in teams for innovation and cutting edge development, and solve real world technical and social related problems at the undergraduate level.

Some breakthrough innovations for TU Sat 1 include: distributed low power micro controllers and software, modular and integrated mechanical, thermal, and electrical design, advanced microelectronics, and integrated science instrumentation with spacecraft systems thereby eliminating multiple boxes and interfaces.

Acknowledgements

This work was supported by the Taylor University Science Research Training Program (SRTP) and the NASA Indiana Space Grant Consortium (INSGC). The authors of this paper would like to thank the following for their time and technical involvement in the design and construction of TU Sat 1: Ball Aerospace Corporation, Fort Wayne Metals, Vroom Engineering, HCJB engineering, and Freewave. We would also like to thank the many faculty, staff, students and their families who put their time and effort into this project.

References

[1] Heidt, H.; Puig-Suari, J.; Moore, A.; Nakasuka, R.; Twiggs, R., "CubeSat: A new Gereration of Picosatellite for Education and Industry Low-Cost Space Experimentation." 13th Annual AIAA/USU Conference on Small Satellites , Logan, UT, August 2000.

[2] Puig-Sauri, Jordi; Turner, Clark and Twigs, Robert J., "CubeSat: The Development and Launch Support Infrastructure for Eighteen Different Satellite Customers on One Launch." 15th Annual AIAA/USU Conference on Small Satellites, Logan, UT, August 2001.

[3] The CubeSat specification may be download from Stanford's CubeSat home page, http://ssdl.stanford.edu/cubesat/.

[4] Sakiko, Fukada-Parr, *Human Development Report 2001: Making New Technologies Work for Human Development*, Oxford University Press, New York, 2001.

[5] Volunteers in technical Assistance (VITA), http://www.vita.org

[6] Kelly, Tim; Minges, Michael; and Gray, Vanessa, *World Telecommunication Development Report 2002: Reinventing Telecoms*, International Telecommunications Union (ITU), 2002

[7] *The Network Revolution and the Developing World, Final Report for World Bank and InfoDev*, Analysis Consulting Report Number 00-216, 17 August 2000.

[8] Oehrig, Jacob et al. "TU Sat 1—An Innovative Low-Cost Communications Satellite." 15th Annual AIAA/USU Conference on Small Satellites, Logan, UT, August 2001.

[9] The P-POD is the tube that houses and launches CubeSats into space. For more details, please see Stanford's CubeSat home page, http://ssdl.stanford.edu/cubesat/.

[10] http://www.arsape.ch/pdf/motors/eAM_1524.pdf

[11] These cells do not Spectrolab's space specifications and can be purchased at reduced rates on their web site, http://www.spectrolab.com/stores/.

[12] http://www.panasonic.com/industrial/battery/oem/images/pdf/LiIon_CGA103450.pdf

[13] http://www.ulbi.com/techsheets/UBI-5091b_UBC525085.pdf

[14] http://www.micrel.com/product-info/products/mic4681.shtml

[15] http://www.national.com/pf/LM/LM2621.html

[16] http://www.ssec.honeywell.com/magnetic/datasheets/hmc2003.pdf

[17] Pfaff, Robert; Joseph Borovsky; David Young Ed, *Measurements Techniques in Space Plasmas Fields*. American Geophysical Union: Washington, DC, 2000.

[18] This figure was taken from Tethers Unlimited web site, http://www.tethers.com

[19] http://www.microchip.com

[20] We used the μFlash TCP from JKmicrosystems, http://www.jkmicro.com

[21] The I^2C bus was developed by Philips Semiconductors. Their I^2C home page is http://www.semiconductors.philips.com/buses/i2c/

[22] Peatman, John B., *Design with PIC Microcontrollers*, Prentice Hall, Upper Saddle River, NJ, 1998.

[23] http://www.atmel.com/atmel/acrobat/doc1116.pdf

[24] http://www.national.com/ds/LM/LM83.pdf

[25] http://www.xicor.com/folders/X1226.php

[26] http://www.maxim-ic.com/quick_view2.cfm?qv_pk=2779

[27] *Predict* is an open source software package developed by John A. Magliacane and is freely available for download at http://www.qsl.net/kd2bd/predict.html

[28] *Earthtrack* was also written by John A. Magliacane and is available packaged with *Predict*.

[29] http://www.jkmicro.com/newsite/products/uflashtcp.html

[30] bzip2 is a portable, lossless data compressor based on the Burrows-Wheeler transform. It was written by Julian Seward and is available freely from http://sourceware.cygnus.com/bzip2/

[31] MIT PGP is a high quality set of encryption utilities. MIT distributes PGP Freeware without cost for non-commercial use. It can be downloaded from http://web.mit.edu/network/pgp.html