Microcontroller Design for the Montana EaRth Orbiting Pico-Explorer (MEROPE) Cubesat-class Satellite

Brian A. Larsen, David M. Klumpar, Micheal Wood, George Hunyadi, Steven Jepsen, Michael Obland
Space Science and Engineering Laboratory (SSEL)[1][2]
Montana State University
204 EPS Bldg
Bozeman, MT 59717
406-994-7383
ssel@ssel.montana.edu
http://www.ssel.montana.edu/merope

Abstract—Montana State University's Space Science and Engineering Laboratory (SSEL) under support from the Montana NASA Space Grant Consortium is engaged in an earth orbiting satellite student project that will carry a reproduction, using current-day technology, of the scientific payload flown on Explorer-1 in 1958 into a 650 km sun-synchronous polar orbit. On-board operations will be commanded by a Motorola MC68HC812A4 (HC12) microcontroller, chosen for its ease of use, processing power, and intrinsic features. Accompanying this will be an Integrated Device Technology CMOS Supersync First-in, First-Out (FIFO) IDT72291 150 Kbyte RAM chip, used for storing scientific data and system telemetry before downlink to ground station. The RAM was selected for the simplicity of the FIFO data flow. The HC12 is responsible for controlling antenna deployment, communication handling, and other system parameters. Using in-house Assembly code, the system has been designed to run an abbreviated main loop, using hardware and software interrupts to call subroutines, thereby allowing the mission critical tasks to be on the main loop for nearly constant monitoring. System interrupts allow the HC12 to seamlessly collect payload data, attitude data, battery conditions (voltage, current, charge state, and temperature), bus voltage, bus current, processor temperature, and stability of the payload high voltage power supply. This interrupt method allows new payload routines to be added with little code modification, maximizing satellite modularity for future experiments aboard Montana State University Cubesats. The design has the advantage of utilizing a powerful microcontroller therein eliminating the need for many of the external components needed by other systems (analog-to-digital converters, serial communications interfaces, slave micro-controllers, etc.), while remaining simple to program and implement. This paper describes MEROPE's computer subsystems in detail.

TABLE OF CONTENTS

1. INTRODUCTION

The Montana EaRth-Orbiting Pico Explorer (MEROPE) is the Montana Space Grant Consortium (MSGC) Cubesat program, being built by the Space Science and Engineering Laboratory (SSEL) at Montana State University in Bozeman. MEROPE will also be the first satellite ever built in Montana. The project is entirely student run, with faculty members acting as advisers. First and foremost, MEROPE is an educational project. Students are involved with every part of the satellite, including but not limited to: designing the satellite, constructing the on-board experiment and all sub-systems, fabricating a ground station to control and communicate with the satellite, testing the engineering and flight models, and contributing to public outreach and web site development. MEROPE is being constructed on a low-cost budget of less than $50,000, including launch, by using mostly off-the-shelf hardware. MEROPE is current day technologies re-flight of the Explorer-1 science payload in 1958 [1]. Many of the mission parameters are based on Explorer-1, for example a four-month mission lifetime would exceed that of Explorer-1. (For more information see [2]) The choice was made to not use specifically radiation hard components mainly for price and simplicity reasons, however with the planned minimum four-month mission life, this will minimize the problem of radiation latchup. More discussion of radiation issues in section 8. In order to correctly design the on-board computer system the mission goals and requirements had to be cleanly laid out.

[1] 0-7803-7231-X/01/$10.00/© 2002 IEEE
[2] IEEEAC paper #196, Updated Dec 4, 2001

- 4-month minimum lifetime
- 40 kHz data count rate capability (from scientific payload)
- Spatial resolution better than 5km
- 125 Kbyte memory capability
- Power consumption control
- _
- Telemetry gathering
- Single event upset (SEU)/power loss recovery

With these requirements in mind the processor choice and bit-stream had to be laid out to satisfy these requirements.

These requirements come from various sources and goals for the project. The four-month mission lifetime is a lower threshold of what success would mean. The scientific payload, a Geiger tube has a maximum count rate before saturation of 40 kHz.

2. Processor/External Memory Overview

The Motorola MC68HC812A4 (HC12) microcontroller is a CPU12, 16-bit central processing unit, with two asynchronous serial communications interfaces (SCIs), timer and pulse accumulator module, 8-bit on-board analog-to- digital (A/D) converters, 1 kbyte RAM, 4 kbytes of programmable read-only memory (EEPROM), and will be running at a 8 MHz clock rate. This processor was chosen for its easy assembly programming, the high-level language capabilities for future missions, its on-chip extras, and extreme reliability. A higher-level language (such as C) would allow more program complexity; but as code complexity increases so does its size, therein requiring larger program ROM and increased system costs. "Good" assembly can be one-third to one-half the size of a higher-level language routine of the same function. Hardware resource is shown in figure 2.

The external memory is an Integrated Device Technology CMOS Supersync First-in, First-Out (FIFO) IDT72291 150 kbyte RAM chip. FIFO memory was chosen for its ease of use and reliability. Not having to address the external RAM unlike standard RAM, frees up man hours for device testing and code optimization, as well as decreasing the number of wires that interface to the RAM, therein increasing the reliability of the data storage system.

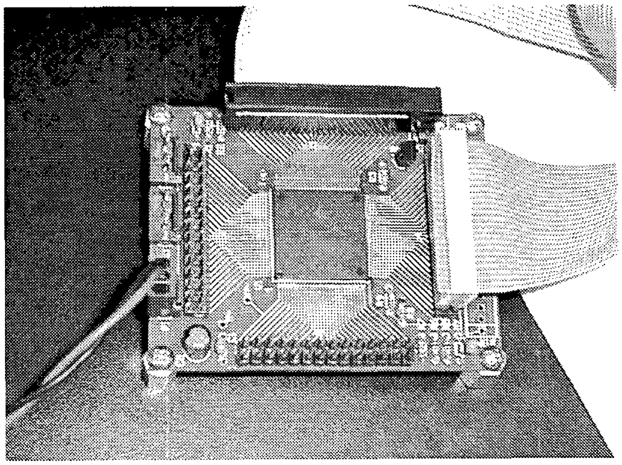


Figure 1. The MEROPE flight Motorola MC68HC812A4 mounted on a BDM12 Debugging Pod purchased from the Seattle Robotics Club.

We have employed an interrupt driven architecture. The main loop of the code monitors the telemetry and attitude information and writes out their data, as well as resetting the watchdog timer, and checking the receive buffer for input. Then the rest of the operations are carried out via interrupts. The system clock runs on an internal to microcontroller interrupt, which also increments attitude and telemetry counters. The scientific payload and associated data write are called from an external interrupt. When information is received from the communications system an interrupt gathers that data into the receive buffer.

The processor as in all spacecraft has the role of mission command and data handling as well as operation of all on-orbit duties, figure 3 shows the system level MEROPE.

3. Processor Duties

In one sentence, the processor must control all system processes. These include servicing payload interrupts, servicing incoming terminal node controller (TNC) data, verifying ground communications link, timer interrupts to maintain an accurate system clock, and collecting critical system telemetry data. As parts of the 40kHz data count rate capability requirement, most of the systems are run from either hardware of software interrupts to ensure more seamless data gathering. Every system timer overflow (~15ms) an interrupt occurs, then every five interrupts the system clock is incremented giving a time resolution of about 73ms. This puts the data collected at much better than the required 5km requirement as

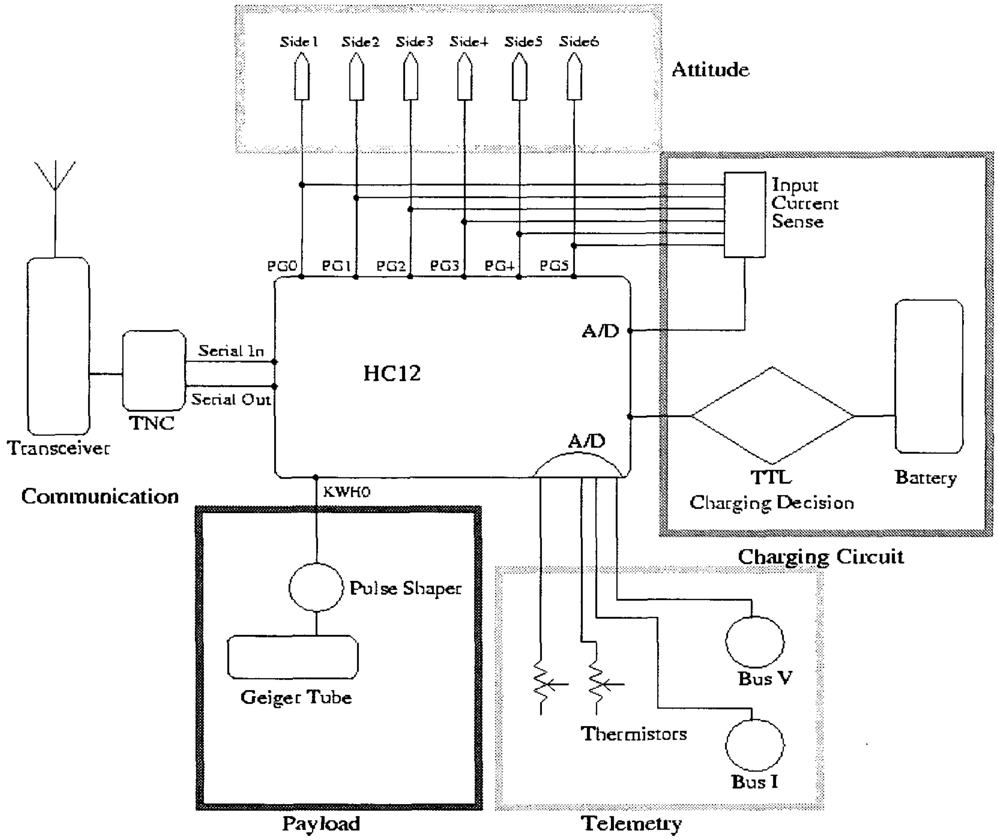


Figure 2. Hardware resources are blocked out and divided among the other sub-systems.

calculated from $\Delta x = v\Delta t$. With this 73ms time resolution the 3-byte system clock will roll over every 14 days and the addition will have to performed on the ground.

MEROPE has two modes in which it can be run, probe mode and orbit mode. In probe mode a serial cable is attached and the batteries can be charged, system diagnostics run, and processor reprogramming may be preformed. Probe mode will be used primarily in final integration and test phases while the Cubesats are being integrated into the P-POD deployer[3]. Orbit mode will be the on-orbit operation of the satellite. All on-orbit tasks are controlled autonomously by the microcontroller. This operation is broken up into three distinct segments, one: on deployment the Cubesats must wait to release antennae to ensure that all craft are far enough apart that no collisions will occur. The processor on power-up will delay and deploy the antennae after the time interval has passed (~20 min). After deployment the microcontroller and satellite will go into normal operating mode. This will consist of running the short main loop and interrupting to handle mission tasks. The third segment is transmitting data to the Montana State University ground station in Bozeman, Montana. When MEROPE receives a communication signal the processor will check it validity and if the communication is a valid command than the data dump to ground station will occur. This is the only time in which the payload requirement of a 40 kHz data rate will not be possible.

4. PAYLOAD INTERFACE

MEROPE carries a science payload as well as the engineering payload that is the entire satellite. The payload is a Geiger tube that will register counts of electrons with energies greater than 50 KeV. This data will be used to measure the lower portion of the inner Van Allen radiation belt. The Geiger tube has a 40kHz saturation rate so a collimator will be fitted to the tube to restrict incident angle and aperture area. The Geiger tube outputs a negative current pulse that will be shaped to a 5V TTL voltage pulse that is then counted by a key-wakeup interrupt on the microcontroller. This system works quite

[3] See http://ssdl.stanford.edu/cubesat/ for more details of the P-POD deployer and Cubesat in general.

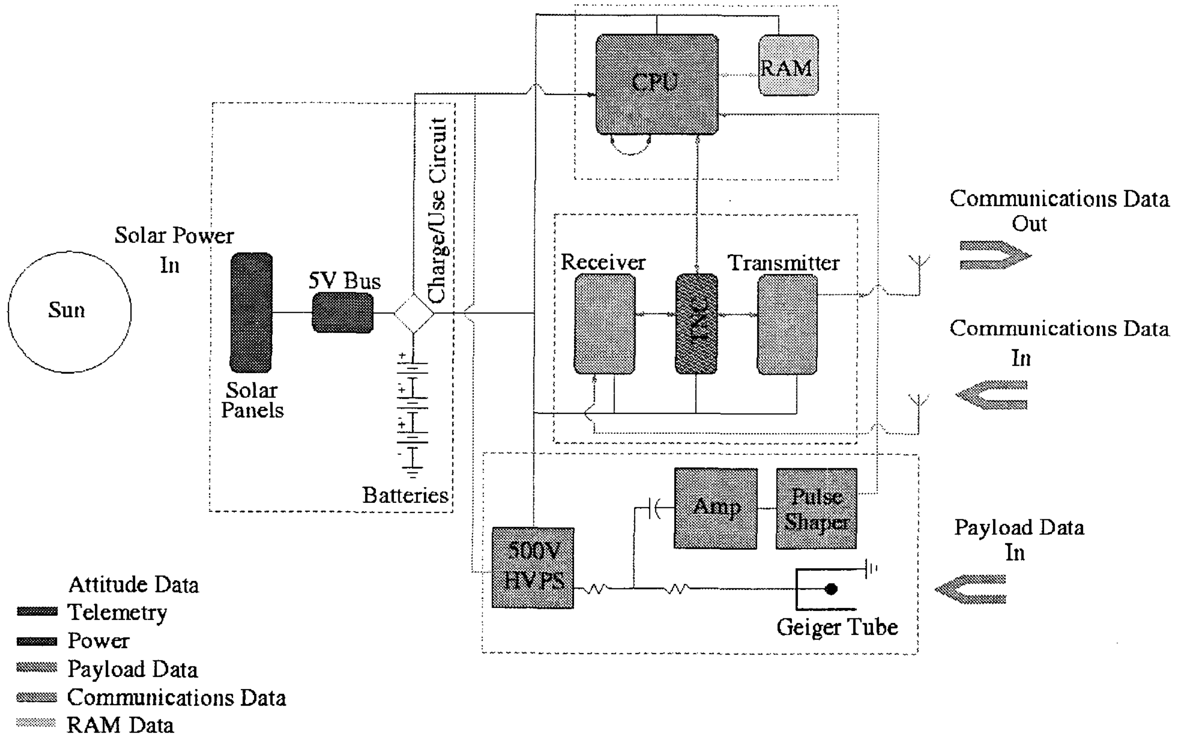


Figure 3. Systems level interface diagram for MEROPE.

effectively and a max count rate of 40kHz is easily accomplished by the microcontroller running at 8MHz. The only restriction is that the code must be designed that no routine runs atomically (this is, un-interruptible) for more than 200 clock cycles [3]. This limits each atomic subroutine to an average of 50~100 instructions. Normally Geiger tube data is taken in bins of time to give counts per second data. Instead MEROPE is incrementing a counter on each pulse and then when the pulse count reaches N the time is written to external memory. This is done in two ways, first is a Hard-Time-Stamp (HTS), which is the full three bytes of the system clock with its appropriate header information. The second is a Soft-Time-Stamp (STS), which is the least significant two-bytes of the three-byte system clock and the appropriate header. One HTS occurs for every 50 STS. These STS are numbered so that in the event of data loss the remaining STS can be interpreted properly. This simple system gives a "compression" of 25% over writing all HTS. This system also gives the payload about a 50% duty cycle since about 50% of the orbit has a very low count rate. In areas of high-count rate we achieve good spatial and temporal resolution and in areas of low count rates we have very low resolution, this fits the mission of "seeing" the Van Allen radiation belt perfectly. Since N never changes it is not necessary to send the collected counts in the downlink data stream.

5. Attitude Interface

MEROPE employs a passive magnetic stabilization system. This system consists of permanent magnets normally aligned to the Geiger tube and magnetically soft iron rods for hysteresis damping. This system has been used on other satellites with good success, however very few 1kg autonomous satellites have ever flown and we want to have the ability to analyze the success of the attitude data for future mission. Calculations yield an expected libration period about the field line of about 10s. So one must sample fast enough to be able to uniquely solve for the period, and do this without an on-board GPS or magnetometer. To achieve attitude information MEROPE will sample the solar cells to see which are receiving sunlight every 3s. This information will allow the reconstruction of the motion. Each of the six sides is run through a voltage cutoff/regulation circuit and then to an input pin on the microcontroller where a TTL logic high or low will be recorded every 3s. Once sampled, attitude data will then be written out to the external RAM. With the appropriate header and HTS or STS. This routine has one HTS after every 20 STS. The beauty of this system is that since this occurs on a fixed time interval the least two significant bytes can be replaced by the one byte STS packet counter that counts from 0-20. Saving one byte every three seconds.

6 Telemetry Interface

The remaining data taken consists of processor temperature, battery temperature and voltage, Geiger tube high voltage and high voltage stability, and bus voltage and current. The temperatures are taken using standard thermistors and read using the on-chip A/D converters. The bus voltage is taken in the same manner; it is fed through a voltage divider and then measured on the A/D converters. The high voltage supply sends out its own 0-3 volt reference to measure the 600 V supply. The accuracy of the A/D depends on the stability of the measured voltage as well as the ± 1 least significant bit accuracy inherent in the conversion, which is constant over the entire operating range. Four conversions are done and averaged to give the reading. The reading is $5V/255 = 19.6mV$ per division. The bus current is run through a current sense resistor and then measured. The telemetry is gathered on 60s intervals and also has a HTS/STS split. There are 10 STS for every HTS. As with the fixed time interval of the attitude the least two significant bytes can be replaced by the one byte STS counter from 0-10.

7 Communications Interface

Communication for MEROPE will be on the HAM radio bands. The communication system consists of a Yaesu VX-1R hand-held radio and a PacComm PicoPacket terminal node controller (TNC) [5]. To save power the TNC and radio will be cycled on an X minutes on Y minutes off cycle. The available system power does not allow continuous operation [5]. On each power-up cycle the TNC will have to be reinitialized. The TNC takes ASCII commands through its serial data line. Once configured the TNC will run in a "transparent" mode where everything sent in is packetized and sent out. The TNC has its own RAM buffer allowing the microcontroller-TNC interface to run at 9600 baud and the TNC-radio interface to run at 1200 baud [4]. During communication dump the microcontroller cannot maintain the 40kHz data input rate and this time needs to be minimized to avoid data loss due to I/O bottlenecks. Fortunately Bozeman, Montana at 45.75 degrees North is too far south to be below a high-count region thereby hopefully eliminating the need for the 40kHz data input rate. The communications will take place as the ground station calls out to MEROPE with its unique identifier. Once MEROPE has a successful verification of the identifier it will start dumping data and continue to do so until it is below the horizon or empty. Currently there is not a command up-link interface (aside from transmit now) on this satellite as it is a generation one Cubesat. The TNC handles all handshaking and data error detection and correction (EDAC), and has the ability to retransmit packets that don't successfully reach the ground station. This is the only EDAC built into MEROPE, on the first generation Cubesat it was decided that there wasn't time to implement an all-encompassing EDAC. To do the EDAC poorly and rushed would be worse than no EDAC at all.

8 Fault Precautions

Our 650km sun-synchronous orbit will place the satellite through the tails of the first Van Allen radiation belt. This orbit will not have a high overall radiation count, thereby lessening the worry of total radiation latch-ups. However single event upsets (SEU) will be quite common when passing through the radiation belt and the South Atlantic anomaly. Normal bit-flip occurrences for our orbit are 1 per million bits each day. The rate in the program ROM is significantly lower. Of these bit-flips about half will never be noticed since they will occur to a bit that does not yet contain valid data. The other half will occur either on stored data or stored program variables. To become a SEU the flipped bit will have to be a program variable that will crash the processor, which defines SEU, otherwise it is data corruption. To accommodate SEUs, the processor has several reset conditions; these are illegal operations code, watchdog timer, and reset. All of these reset conditions will cause the processor to reset information in all registers and return the program counter to the top of the code thus reinitializing all RAM variables and beginning the code tree again. This will not result in a total data loss for the stored information since the power on code will reinitialize the external RAM and reset conditions will not, so long at the external RAM does not lose power all data will be stored through an SEU. The internal watchdog timer is the main source for processor reset. It resets the processor on a period of about one second if the timer is not reset in that time. There will be an external backup to the watchdog timer on a slower cadence to ensure that in the event of a bit-flip disabled watchdog and an errant processor, reset will occur. The illegal operations code reset occurs if the processor starts to execute a command that is not a member of the instruction set, this can occur in a few ways, first and most likely the program counter can get flipped to a memory location that is data and not instructions, this will cause a reset; second the code can become corrupted and then the processor is lost, this is an unlikely possibility since the code is stored in EEPROM which is much less sensitive to bit-flips than standard RAM. Again in future generations there will be a code EDAC system. A bit-flip can cause any bit in RAM to flip changing the function of any part of the microcontroller. For example an unused pin that is grounded through a pull-down resistor could be made to sit normally high though a bit-flip causing the processor to have to source a large amount of current through that pin, which can ruin the processor. To avoid this problem on every reset of the watchdog timer all RAM variables will be reinitialized. So that more than every second the I/O pins, interrupt vectors, and other RAM variables will be reinitialized so that the above scenario can only occur for less than one second, which the processor will survive.

9 Conclusions

The success of MEROPE is a journey requiring a lot of testing and revisions. Since the number one killer of satellites is processor troubles we have tried to ensure a robust, KISS system. This system was put together to be a modular as possible using the interrupt system. For the second generation Cubesat one only has to add new interrupt routines and the routines such as attitude and telemetry gathering can be reused with little or no change. The design philosophy that went into this system was to use a powerful microcontroller to avoid having to utilize slave microcontrollers and numerous external components. Therein taking the complication of the satellite from hardware to software, which is more of our expertise.

This work is supported by the Montana Space Grant Consortium (NGT5-40092) and the Montana NASA EPSCoR Program (NCC5-240).

REFERENCES

[1] Van Allen, J.A., Origins of Magnetospheric Physics, Washington D.C.: Smithsonian Institution Press, 1983

[2] Obland, M., G. Hunyadi, S. Jepsen, B. Larsen, D. M. Klumpar, C. Kankelborg, and W. Hiscock, "The Montana State University NASA Space Grant Explorer-1 7.Science Reflight Commemorative Mission" 2001 AIAA/USU Small Satellites Conference Proceedings, August 13-16, 2001.

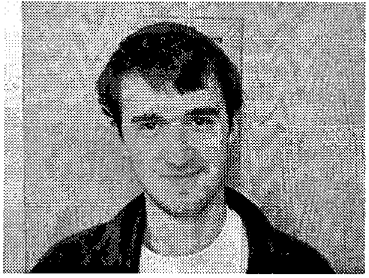
[3] Fredrick M. Cady and James M. Sibigtroth, Software and Hardware Engineering –Motorola M68HC12, New York: Oxford University Press, 2000.

[4] Fredrick M. Cady, Microcontrollers and Microcomputers, New York: Oxford University Press, 1997.

[5] Hunyadi, G., D. M. Klumpar, S. Jepsen, , B. Larsen, M. Obland. "A Commercial Off the Shelf (COTS) Packet Communications Subsystem for the Montana EaRth-Orbiting Pico-Explorer (MEROPE) Cubesat" 2002 IEEE Aerospace Conference Proceedings, March 9-16, 2002.

[5] Obland, M., D. M. Klumpar, S. Kim, G. Hunyadi, S. Jepsen, B. Larsen. "Power Subsystem Design for the Montana EaRth Orbiting Pico-Explorer (MEROPE) Cubesat-class Satellite" 2002 IEEE Aerospace Conference Proceedings, March 9-16, 2002.

***Brian Larsen** - Earned his B.S. in Physics and Mathematics in 2000 from Linfield College and is now a graduate student at Montana State University. He is trying (successfully?) to balance his duties as software and firmware lead and his need to ski, climb and bike.*



***Dr. David Klumpar** - David Klumpar is a research professor and director of the SSEL lab here at Montana State. He received his B.A. in physics and mathematics at University of Iowa. He received his M.S. in physics from the University of Iowa in 1968. In 1972 he received his Ph.D in physics from the University of New Hampshire.*

***Micheal Wood** - Micheal is completing his B.S. in Computer Engineering at Montana State University. When he is not riding or repairing his '78 Honda Express, he is busy at work as the principle firmware developer for the MEROPE project.*

***George Hunyadi** - Earned his B.S. in Physics at Montana State University (1997). Worked a total of one year at NASA's Goddard (Greenbelt, MD) and Marshall (Huntsville, AL) Space Flight Centers as staff and research associate for the NASA Academy. Currently pursuing M.S. in Electrical Engineering at MSU, with an emphasis in RF and Communications systems. George is Project Manager as well as the Communications, Telemetry & Data Handling (C, T & DH) subsystem lead. He enjoys amateur astronomy, skiing, climbing, and fishing.*

***Steven Jepsen** - Earned his B.S. in Geology and Physics from California State Univ, Fresno to then move to graduate work at Montana State University. Steve is the lead on mechanical & structure and is in charge of operations. He spends his winters telemark skiing and summers hiking/fishing. Steve also spends a lot of time reading.*

***Mike Obland** - Earned his B.A. in Physics and Mathematics in 2000 from the University of Montana. Currently he is a Ph.D student in physics at Montana State University. He is the Deputy Project Manager as well as the Payload and Business/Outreach sub-systems lead. In his few moments of spare time, he enjoys outdoor activities including exercising, camping, and fishing*