Project Number: CS-CEW-0802

Lunar Micro Rover

A Major Qualifying Project Report
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
in Computer Science

by

Katy Levinson

Date: 1/9/2008

Approved:

Prof. C. Wills, Major Advisor

keywords
1. rover
2. embedded systems
3. NASA

Project Abstract

A light and small form factor lunar rover platform holds a significant promise of benefits for NASA (National Aeronautics and Space Administration) both financially and scientifically. The LMR (Lunar Micro Rover) project is an attempt to design such a platform on a low budget. This report outlines the progress of the LMR team made between August 2007 and December 2007 in software and decisions relating to software. This time frame encompasses Prototype 3, 4, and some planning of 5, the demonstration for Director Worden, image compression work, and VxWorks work. This report also provides a basic background of events that led up to this work beginning with the MAX (Mobile Autonamous eXploration) LMR demonstration in the Atacama Desert (2006).

Executive Summary

NASA has an interest in developing a reusable micro rover platform. There are many benefits to designing a rover platform that can be manufactured in large quantities, modified slightly for each mission, and then launched. Micro rovers are easier and cheaper to build once the original design is completed, and less expensive to fly. When used in a group, micro rovers are more robust, able to gather a larger variety of data in a single mission and able to exhibit more complex behavior. Micro rovers are easier to design launch and landing vehicles around.

This report outlines the progress of the LMR team made between August 2007 and December 2007 in software and decisions relating to software. During this time prototype 3 was designed, fabricated and tested. A demonstration for Director Pete Worden (Director of NASA Ames Research Center) was held so that Director Worden could evaluate the team's progress to that point. Data that was gathered during the demonstration was analyzed. Prototype 4 was designed, fabricated and tested. Design of Prototype 5 began. The issue of video compression was studied and the team concluded that single frame image compression was preferable over a video compression algorithm. The team also switched a different version of VxWorks (from 6.3 to 5.5) and began work in the new operating system researching how much overhead specific features used.

This report also provides a basic background of events that led up the work that was done during the MQP (Major Qualifying Project) time period. This began with the development of the MAX platform by Carnegie Mellon University and the 2006 MAX demonstration in the Atacama Desert of Chile to Director Pete Worden. It included the transition to the LMR series and the physical characteristics of the moon which caused a dramatic redesign of the series. Prototype 1 was designed, fabricated, and tested during that time as was Prototype 2. VxWorks work included getting VxWorks 6.3 operational.

Overall this report details the benefits of a micro rover platform, the work between the dates of August 2007 and December 2007 by the LMR team relating to software, and the history of the project up to August 2007.

Acknowledgments

Thank you to the LMR team. Mark Leon: our fearless leader, Khalid Al-Ali: the leader who remembers we might be wiser to be a little more fearful, Jacques Dolan who is easily the most dedicated coworker I have ever had, Ritchie Lee whose patience is appreciated, Chris Ishisoko and Jack Biesiadecki who can probably do anything between them, Josh Wainer, Daniela Buchman and Kerry Scharfglass: the summer crew and John Bowland and Andrew Khoflas: nobody ever thanks either of them and they both put up with a lot.

I am extraordinarily grateful to Professor Wills at Worcester Polytechnic Institute. He has permitted me to take on this unorthodox endeavor as my Major Qualifying Project and has also demonstrated incredible patience during the entire process.

Thank you to Corey Ippolito of NASA Ames for constantly answering technical questions.

I appreciate everybody who fixed grammatical errors in this report: Daniela Buchman, Patrick Crane, Nick Dufour, Elan Hourticolon, Ken Stein, Amers Tolle, and Ethan Truong.

Table of Contents

Report Roadmap

This project is unconventional due to the fact it does not follow the standardized MQP format. This section is meant to provide a basic outline of the report and what is covered in each chapter.

Chapter 1 is the introduction which discusses the main motivations of the project. It also discusses why a micro rover is a good solution to the current constraints on NASA.

The history of the project is outlined in Chapter 2. This discusses the 2006 Atacama demonstration, pre-prototype 1 concepts, prototype 1 and prototype 2. It also discusses a few of the ongoing challenges that were inherited into the MQP from VxWorks, the hardware, and the physical characteristics of the moon.

Chapter 3 detail the project work. This included video compression, single frame compression, various code architectures, switching from VxWorks 6.3 to 5.5, Prototype 3, a demonstration for Director Pete Worden, analyzing the data from the demonstration, Prototype 4, beginning to design Prototype 5, and some work in physics simulators.

Chapter 4 is the conclusions chapter. It reiterates the most important events detailed in the report.

Following the report is included an appendix of major acronyms used in this report and some photographs of various versions of the rover.

Chapter 1: Introduction

Spaceflight is expensive.

The entirety of NASA is run on approximately 17 billion dollars.[1] NASA has always wished to do as much work with the taxpayer money as possible, and since the presidential challenge and Project Constellation, NASA has further motivation to do more with less money.

The cost of a spaceflight can be effectively divided into three parts: the cost of development, the cost of launch, and the cost of running the actual mission. The cost of development can be mostly expressed as the sum of: the cost to develop the original concept, the cost of the raw materials and COTS (Commercial Off The Shelf) components needed to make the rover, the cost to manufacture the custom parts, the cost of the personnel who integrate the parts and design/build the non-COTS parts, the costs of hardening any non-radiation tolerant pieces and the costs of testing. The cost of launch consists of: the cost of using the actual launch area, the cost of the launch vehicle (which grows in cost proportionally to how much it must lift and how large the payload is), and the cost of the personnel who man the actual launch. The cost of running the actual mission is mainly derived from the cost of the tools required to complete the mission and the cost of the personnel required to man the operation.

This project took a thorough look at this breakdown of costs and designed LMR to be inexpensive. In response to the constraints on launch costs, the dimensions of the physical rover were shrunk and the weight was decreased. In an era when rovers such as Spirit and Opportunity are the size of golf carts and the soon-to-be-launched Mars Science Lab (or MSL) is approximately the size and mass of a Volkswagen Beetle, the LMR rover is tiny; it fills less than a volumetric foot of space and weighs only 10 kilograms. It also features reduced development costs by attempting to use COTS parts nearly exclusively for all electronics.

In addition to being a lower cost option, microrovers pose many advantages over their larger counterparts. First, is the added benefit of reliability. Putting all of the sensors on one rover is somewhat analogous to the phrase “putting all of your eggs in one basket.” If a single radio cable becomes unplugged, a wheel jams, a solar panel becomes disconnected or bent, or any one of many small other mishaps the entire mission and all of the experiments on the rover will fail. Compare this

---

[1] United States Executive Office of the President: Office of Management and Budget. 2007. Budget of the united states government: National aeronautics and space administration.

to the event when one rover out of many breaks; the mission can continue with the still functioning rovers. In some special and fortunate circumstances the other rovers can help the broken one compensate for its shortcomings and allow that mission be successful as well. Suppose a wheel becomes disconnected from the drive shaft, making one rover unable to move. Another rover can push the broken rover, thus still permitting it to deliver its payload.

Second, the manufacturing of several small rovers instead of one large one is far cheaper and much quicker to integrate. This is due to the fact that for smaller rovers the process is pioneered once and then replicated for every rover, but on a larger rover a team is always breaking new ground.

Thirdly there are scientific advantages to having different secondary science payloads on different rovers; this way, each of the rovers can go to a different location to carry out its experiments. Some examples of this include: a neutron detector, which may want to examine a small patch of ground quite thoroughly; a mapping payload, which may want to cover a great deal of flat ground rapidly; a water-seeking robot, which may wish to dive deep into one crater to explore it and then become unable to return due to thermal constraints or the slope of the walls.

Fourth, dividing one rover into many permits more complex behavior. At NASA there is a saying that “every articulation costs a million dollars.” This saying expresses the pressure on rovers and satellites to have as few moving pieces as possible and how unlikely a large complex rover is to have articulating arms for complex tasks. Many smaller rovers working together with good coordination can perform tasks that would be done with a complex articulating arm on Earth. Many small rovers also permit intelligent tasks not possible with a single rover, such as swarm-related logic and behavior.

Fifth, having several small rovers lessens the design pressure on the lander and launch vehicles as the rovers can serve as their own ballast. When a vehicle that will fly is designed it must be balanced or it will not fly correctly. Normally when the vehicle has a single large payload it must be designed to shift weight around that payload so it is still balanced. With the small rover design several rovers can be hung around the outside of the vehicle and balance each other. This encourages lander designs to be reusable as well as the rovers.

Overall there is a strong motivation to explore the possibility of using micro rovers. They are easier and cheaper to build once the first one is finished, cheaper to fly, more robust, a group is able to gather a larger variety of scientific data in a single mission, they can in a group exhibit more complex behavior, and they are easier for other vehicles to design around.

Chapter 2: Project History

2.1 MAX

2.1.1 The MAX (Mobile Autonomous Exploration) Series

The MAX is a rover and research platform rover developed by CMU West (Carnegie Mellon University West) currently being produced by a spin-off company called Senseta. It uses almost entirely COTS parts. It is designed to have roughly the same processing power, durability, and cost of a laptop computer but various sensing devices whose integration has been requested by various customers have spawned more expensive variants. An example of this is the MAX 5J for JPL (Jet Propulsion Laboratory).

Carnegie Mellon University West is a branch of Carnegie Mellon University. Founded in 2002 it grants master degrees in software and software-related fields.[2]

The MAX rover was developed in CMIL (Carnegie Mellon Innovations Lab), which is part of CMU West.[3] CMIL is also where a great deal of the development of the rest of LMR series was conducted.

2.1.2 MAX LMR Atacama Demonstration

In 2006 NASA Ames Research Center and Carnegie Mellon University West went to the Atacama Desert in Chile to test their new robotic creation: the MAX LMR rover.[4]

The Atacama Desert stretches from the northern region of Chile to the southern border of Peru. It is the driest place on earth[5] and in places no water or rainfall has been recorded there in the entirety of human history.[6] It is frequently used by NASA as an extra-terrestrial analogue site for scientific

---

[2] Carnegie Mellon University West. 2008. Carnegie Mellon University West. Silicon Valley California. Online. Available from the Internet. Available at http://west.cmu.edu/

[3] Carnegie Mellon Innovations Lab. 2007. Carnegie Mellon Innovations Lab. Silicon Valley California. Online. Available from the Internet. Available at http://cmil.west.cmu.edu/

[4] National Aeronautics and Space Administration Intelligent Systems Division. 2006. Micro rover Technology Demonstration. Silicon Valley California. Online. Available from the Internet. Available at http://ic.arc.nasa.gov/news/story.php?id=361

[5] National Geographic World. The Atacama Desert. Online. Available from the Internet. Available from http://magma.nationalgeographic.com/ngm/0308/feature3/

studies (primarily Mars related) because it is so dry that, if Viking had landed in one of the drier areas, it would read that no life exists at all.[7] This sort of extreme climate is also one of the best field test estimations available to see if a specific piece of hardware will perform reasonably well in an extra-terrestrial environment.

The rover was demonstrated to the Director of NASA Ames Research Center.[8] It was considered such a success that CMIL and NASA were requested to build a theoretical prototype that could work on the moon.

2.2 Transition from MAX to LMR

2.2.1 Physical Characteristics of the Moon

When it was decided that a version of the MAX should be made that would be suitable for use on the moon, the team began considering the physical properties of the moon that would be relevant to this mission. Most current candidate landing sites on the moon have not had previous landings at the site, therefore the project attempted to prepare for the worst combination of effects possible.

2.2.1.1 Lunar Regolith

Regolith is a Greek word and means “blanket rock”. It has different meanings on different planets, but on the moon it refers to all the soil between the bedrock of the moon and the surface.

Lunar regolith is an incredibly harsh and unforgiving material and was a serious issue during the Apollo missions. During extra-vehicular activities in Apollo 12 the dust wore through the astronaut's space suits in several places revealing insulation and causing the suits to leak oxygen. On Apollo 16 the specially made vacuum to clean the regolith from the lander became clogged with regolith and ceased to function. On Apollo 17 the astronauts became directly exposed to the dust when it entered their space suits and it made one of them ill.[9] Regolith varies in particle size from large

---

[7] National Aeronautics and Space Administration Ames Research Center. 2003. Mars-like Atacama Desert Could Explain Voyager “No Life” Report. Online. Available from the Internet. Available at http://www.nasa.gov/centers/ames/news/releases/2003/03_87AR.html

[8] National Aeronautics and Space Administration Intelligent Systems Division. 2006. Micro rover Technology Demonstration. Silicon Valley California. Online. Available from the Internet. Available at http://ic.arc.nasa.gov/news/story.php?id=361

[9] John Mangels. 2007. Coping with a Lunar Dust Up. Seattle Times. Seattle Washington. Online. Available from the Internet. Available at http://seattletimes.nwsource.com/html/nationworld/2003572876_moondust15.html

boulders to incredibly finely ground (about 20 microns in size). This small size of some particles means it can fit in nearly any gap. The regolith is frequently razor sharp since there is no atmosphere or water to grind it smooth. These sharp edges help it stick to most objects that it touches. The regolith is statically charged by the sun during daylight. This static electricity also makes it cling to objects.

Lunar regolith is formed two ways. It can be formed by meteors crashing into the surface of the moon that breaks apart existing rocks and turns them into smaller parts. It can also be formed by the existing regolith re-melting together in the heat of space into larger pieces.[10]

In some areas the regolith is magnetic due to the composition of the meteors that formed it. This poses a serious problem, as the dust can be charged by induction from the motors and electronic circuits, thus magnetically drawing the regolith into some of the most critical areas of the rover. It also is charged by the sun to gain large quantities of static electricity. These ESD (electro-static discharge) shocks are terrible for computer boards.

Another interesting property of lunar regolith is the fact that it absorbs large amounts of radiation. This phenomenon has not yet been studied in great depth but many payloads are interested in further study of this.

2.2.1.2 Randomized Radiation Events

Aside from the issue of the radiation that is constant in the regolith, the moon is known to randomly have a "radiation event." This is a large spike in the radiation level caused by a solar storm.[11] The statistical chance of it happening during the project is not significant unless the mission time frame is longer than one month.

Some hardware is more affected by radiation events than other hardware. Processor logic is frequently interrupted because the physical gates that make up basic mathematical operations receive random false positives. Volatile memory loses its integrity because random gates become flipped. Memory components that only flash-write are safe. In rare circumstances severe radiation events can damage circuit boards or radio receivers permanently.

Radiation events are likely to be more dangerous on the moon compared to LEO (Low Earth

---

[10] Department of Earth and Planetary Sciences Washington University in Saint Louis. 2007. Lunar Regolith Breccias. Saint Louis. Online. Available from the Internet. Available at http://meteorites.wustl.edu/lunar/regolith_breccia.htm

[11] Education and Public Outreach National Aeronautics and Space Administration. 2008. Radiation Belt Storm Probes Educational Website. Online. Available from the Internet. Available at http://rbsp.jhuapl.edu/

Orbit) and GEO (Geostationary Earth Orbit) satellites. LEO and GEO satellites are protected by the earth's Van Allen radiation belts, which are believed to intercept the more violent aspects of solar storms. This Van Allen radiation belt does not extend to the moon. The rover may have to pass through it on its way to the moon thus the rover might be exposed to at least some doses of radiation.[12]

2.2.1.3 Day Length

The length of a day on the moon causes severe design challenges in the areas of power and thermo. The span of noon-to-noon on the moon is approximately 29.5 days long.[13]

This is a serious concern for thermo. Because of these long days, as well as the lack of an atmosphere on the moon, the temperature varies greatly. A rover that becomes too cold during the night may freeze and break. Heating components could be added to a rover to survive the night but these require power to operate, take up space, and add extra weight. Rovers cannot be built with the goal of maintaining heat exclusively in mind because, during the equally long lunar day, the temperature becomes extremely hot, thus putting the rover in danger of melting. Larger objects can use their thermal inertia to remain cool and warm. Thermal inertia is the concept that if an object is large enough and the specific heat for that object is high enough the object will not finish heating or cooling before the trend reverses. The team has not found any viable material which would give this tiny rover sufficient thermal mass to accomplish this.

These long days also pose a serious problem for power. Solar-powered rovers do not gain any additional energy at night. Because of this issue the rover must be able to store enough power to keep itself warm for two weeks when the sun goes down or have an alternative power source or it will not survive the night. If the rover is to do anything during the night it must also have enough power to execute those actions as well.

2.2.2 Travel to the Moon

The greatest problem the LMR is likely to have traveling to the moon is launch vibration. Vibration on takeoff is severe and can pose many problems. Flimsy pieces are likely to snap. Most of

---

[12] Stern, David P. and Peredo, Mauricio The Exploration of Earth's Magnetosphere. National Aeronautics and Space Administration Goddard Space Flight Center. Florida USA. Online. Available from the Internet. Avaialable at http://www-istp.gsfc.nasa.gov/Education/Intro.html

[13] National Aeronautics and Space Administration Worldbook. 2007. Moon. National Aeronautics and Space Administration. Online. Available from the Internet. Available at http://www.nasa.gov/worldbook/moon_worldbook.html

the danger, however, comes from couplings; bolts are likely to vibrate and become loose, weak solders may break, and connections may become unplugged.

2.2.3 Hardening Feasibility

Almost as soon as the decision was made to space-harden the MAX LMR it was determined that the idea was completely infeasible without a major redesign. The MAX LMR was determined to have too many failure points for them all to be solved on the necessary budget. Most of these stemmed from parts such as suspension and locomotion parts that were exposed to the lunar regolith.

2.3 The Prototype Series

Eventually a basic design was arrived at for the LMR Prototype Series. This basic design has been modified for each version in the prototype but many parts of it have remained constant through the entire process.

2.3.1 Design Overview

2.3.1.1 Communications Design

Communications from the rover to the earth involve a number of steps in between. Here are the basic steps of the transmission from the rover to earth. A transmission from earth to the rover will take the same path in the opposite direction.

The rover will transmit to the lander using one UHF (Ultra High Frequency) radio in the rover and a matching UHF radio on the lander. These radios have RS232 inputs and outputs from the rover, but they internally correspond using TCP (Transmission Control Protocol). This means that if multiple rovers are sent, the lander will not need multiple UHF radios. Instead the UHF radios will be arranged like a simple wireless network.

The lander will complete its mission objectives before deploying the rover. At the point when the LMR is running, its only mission will be to receive transmissions from the LMR and forward them to the DSN (or Deep Space Network). This step is considered to be the greatest bottleneck in the communications. The traditional transmission system used also poses some issues for the team at this step. On the lander, packets are normally stored until they complete a frame and then the entire frame is sent back to earth at once. This means that unless the frame size is fixed to be small, the team may have a high average bandwidth but it will not be a constant amount of bandwidth.

The DSN will then take this information and bring it down to earth by satellite. The data can then be brought to mission control.

2.3.1.2 Mechanical Design

The mechanical design of the rover is relatively simple. A large rectangular prism e-box (or electronics box) holds the electronics. On either end of the e-box there is a hammerhead that consists of a small portion of box aluminum welded at a ninety-degree angle to the center of an aluminum tube so that the edge of the box aluminum meets the side of the aluminum tube. At the end of each side of the aluminum tube is mounted an aluminum circular plate (or wheel hub) which is the diameter of the wheel. At the edge of this circular plate the circle is connected to a sheet of metal that has been bent into a cylinder without the top and bottom. This cylinder is mounted to the edge of the wheel hub so that it extends inwards to cover all but the central few inches of the hammerhead. The insides of both hammerheads have been hollowed out: one end contains a motor in each end of the aluminum tube (the wires run back through the hammerhead into the e-box through a hole) and on the other end a space is hollowed out to hold the camera in the aluminum box. A pair of Kevlar belts run from the front wheels to the back, forming the drive system.

2.3.1.3 Software Concepts

While no piece of the LMR Prototype series coding has been fixed to any one piece of hardware or operating system there are some software concepts which the team is attempting to uphold during all versions whenever possible.

2.3.1.3.1 Flight Code Standards

NASA HQ (headquarters) does not set any standards that are far reaching enough that the team worried about following them this early in the prototyping stage. The launch site imposes most deep-reaching changes due to standards. Since we are not entirely sure on what launch vehicle the LMR would be launched with, let alone from what center it would be launched, we attempted to follow all standards we could find. We focused on GSFC (Goddard Space Flight Center) as they are a likely launch site and their standards were the easiest to access.

Most of the standards imposed are best expressed as standardized common sense and good coding practices, such as conventions for variable names, being clear about what units a variable is expressed in, conventions for file names, and placement of opening and closing brackets.[14] We ran into some interesting situations, however, as flight software generally strongly discourages nearly all use of the heap and run-time memory allocation. Flight software rules generally advocate simply making an array larger than necessary than to dynamically resize it at runtime.

2.3.1.3.2 RTOS

An RTOS (Real Time Operating System) is a type of OS (Operating System) in which the greatest priority is to minimize the latency time between a situation and a response. On the LMR, examples of these situations can be program commands, system interrupts, and sensor inputs. Internal latencies such as process switching and page swapping are absolutely minimized, often by design at the expense of robustness, fault recovery, and user-friendliness. An RTOS alone is not enough to guarantee a fully real-time deterministic system but it is an essential tool in the construction of such a system.

2.3.2 Pre- Prototype 1

Several pieces of Prototype 1 had multiple revisions. These small pieces were never fully assembled into a rover prototype but had individual prototype series of their own.

2.3.2.1 Metal Belts

A tested and rejected idea was to have the drive belts made of chain with metal tabs that grip the ground. This idea was rejected for mainly three reasons. The first reason was that there were too many moving parts. Each link of chain had two moving parts on it and each moving part was considered a failure point. The regolith, once inside, would prevent the part from moving and if the piece was not sealed the lubricant used to make the chain move was likely to evaporate. The second reason was the weight of the entire contraption. The chains and metal tabs took up an unacceptably high proportion of the mass budget. The third reason was that this weight was not only taking up an unacceptably large

---

[14] Code 582 Flight Software Division National Aeronautics and Space Administration Goddard Space Flight Center. 2003. C++ Coding Standards. Online. Available from the Internet. Available at http://software.gsfc.nasa.gov/AssetsApproved/PA2.4.1.3.1.pdf

amount of the mass budget, but also that this large quantity of weight must be turned to have the rover move. This meant that a great deal of motor energy was used in simply overcoming the inertia of the tread and so to achieve the same speeds a given motor would now have to work harder. If the motor needed to work harder it would use up a greater proportion of the power budget per second, thus shortening the life of the mission.

2.3.2.2 Relative Wheel Size to Electronics Box

Another idea that was discussed was the relative size of the wheels to the electronics box. Different wheel sizes had different properties.

If the wheels were smaller than the electronics box by a significant margin, the rover would be “pointed” at the ends like earth-tanks and thus able to navigate rough terrain better. It would also permit a far longer electronics box to exist because of the extra space. Smaller wheels would also mean that, in general, less of the rover tread is in contact with the surface of the moon. This shorter tread base causes less friction when the rover turns. One major disadvantage was that this configuration caused the belt to constantly push against the belly and ceiling of the electronics box, which introduced a lot of friction while simply moving forward. The other major concern is that the degree of contact between the rover wheel and the belt would not be enough to prevent the wheels from slipping.

Another possibility was to have the wheels be larger than the electronics box. This would provide over 180 degrees of contact between the belt and the wheels, and the belt will clear the electronics box without rubbing against it. The drawbacks are: lunar regolith may become caught in the space between the belts and the rover, the larger wheels would significantly eat into the space the electronics box can take up, the rover might have issues going over steep terrain, and the basic inertia to turn the treads would increase.

The last possibility was to have the wheels be the same size as the electronics box. This did not have any advantages inherent to it but it was eventually chosen because it was believed to not have any of the disadvantages of the former two options.

2.3.3 Prototype 1: Hephaestus

Prototype 1 was designed to be a proof of concept. It has absolutely no software onboard and is controlled directly through the UHF radios from the ground station. It holds a record as the first known fabric drive system. It was made to prove that, mechanically speaking, a belt-based drive system was viable. A listing below shows the more significant features that were added to this particular prototype and needed to be programmed.

2.3.3.1 Reflections Architecture

The Reflections Architecture is a program used by many projects at NASA ARC (Ames Research Center) including prototypes 2, 3 and 4 of the LMR, the MAX, and various UAV (Unmanned Aerial Vehicle) projects. It was designed and is primarily maintained by Corey Ippolito.

The reflections architecture works in essence like a large software breadboard. There are many “modules”, each of which represents a single hardware object or software concept (examples: a joystick, TCP or UDP (User Datagram Protocol) output, a particular brand and model of motor controller, or a USB (Universal Serial Bus) video camera). Each of these modules has a series of undefined inputs and outputs, but internally has all the methods required to complete all commands required for operation of the physical object or software concept. A special type of script file links the appropriate outputs from one module (example a joystick) to the correct inputs of another (example a motor controller). The great advantage of this is the flexibility of the code. Once the motor controller module works the joystick input could easily be switched to another module which represents an autopilot or a joystick of an entirely different make and standard convention with only a few lines of script.

Reflections runs on Windows and *nix. It does not have flight hardening or heritage. It is primarily used for proof of concept and prototyping.

On Prototype 1 Reflections was run on a ground computer. The signals were then sent through a radio and directly to the motor controllers on the other side with no computer to interpret in between.

2.3.3.2 50 Watt Motors Controllers

These motor controllers are far larger than the 20 Watt counterparts that were intended to be used for later prototypes but were used for early experimentation on controlling the motors on Prototype 1. They were controlled directly by the UHF Radio.

2.3.3.3 UHF Radio

This is a small UHF radio set that is designed to be used between the lander and the micro rover on the moon. The radios themselves were designed to be controlled over a serial RS232 connection, but internally they communicate using TCP/IP protocol. The radios must be configured as a private network to speak with one another. The three ports are: COM1, a serial port; COM2, which is also RS232 but the physical port only takes cat5 (Category 5, Cable and Telephone) cables; and a programming port which also takes input from a cat5.

2.3.4 Prototype 2: Prometheus

Prototype 2 was designed to be the group workhorse. Once the form factor was determined this particular rover was designed to be assembled and run for long periods of time to accumulate large quantities of test data. Varieties of parts were to be swapped in and out and data was to be collected on how these parts impacted the performance of the rover. The machining of the original concept was delayed by non-technical issues, and the eventual realization of the concept was to put the pieces of prototype 2 that had been finished on prototype 1 and continue work. This meant that the transition from prototypes 1 to 2 did not have any significant impact on software as there was no space for an onboard computer in the Prototype 1 electronics box.

2.3.4.1 20 Watt Motor Controllers

Once the system of using the 50 Watt motor controllers had been perfected the team switched to the 20 Watt motor controllers. These were the ones to be used on the final robot. The 20 Watt motor controllers in addition to only being roughly a quarter of the volume of the 50 Watt motor controllers draw far less power than their 50 Watt counterparts while still being strong enough. This was favorable because it extended the mission life of the rover.

2.4 Hardened Stack

The radiation-hardened computing stack comes in many pieces. A motherboard connects into a VMEbus backplane. This motherboard is programmed via a cat5 cable that plugs into a secondary board that piggybacks on the motherboard so that they fit within the same standardized slot. During actual operation this adapter board is replaced with a frame grabber board. This frame grabber board takes raw feed from the camera, compresses it, and returns it to the motherboard for processing and transmission. The VMEbus plane connects into an IOBoard. When the board stack is in the flight cage the VMEbus also connects into a power board.

2.4.1 Card Cage Issues

Early incarnations of LMR did not have the radiation-hardened set of cards on the actual system. This was because, in addition to the software issues of VxWorks, there was an issue on the integration of the power board and the conduction cooled flight card cage.

A flight card cage is an object that holds the boards of the computer together. It can be compared to a tower in a desktop computer. In LMR, however, it is a much more sophisticated object as it is also responsible for regulating the heat of the system. Most earth computers use fans for cooling, but this is not a possible solution due to the vacuum of space. Instead, card cages are designed to conduct heat to and away from the board through the frame of the board and the holder.

A power board can be compared to a power supply in an earth computer tower. It is only needed in the flight card cage. This is because in the development card cage there is already a complete power supply (although it is not radiation hardened.) The development card cage is not interchangeable with the flight card cage because of differences in the form factor and because the development card cage is only for use on earth, thus it has fans.

There was an integration issue between the power supply board and the flight card cage. A standardized form factor was listed for the power board and the flight card cage was designed to those specifications. In reality however the actual power board had an oversize capacitor that stuck out too far for the board to fit into the flight card cage. Working out these issues with the contractors absorbed a significant amount of the team's effort and also meant that it was increasingly difficult to have the development on the radiation hardened boards and the development of the rover hardware closely linked.

2.4.2 ITAR Restrictions

Another previous issue that had come up was that use of the radiation-hardened boards was difficult as the technology is ITAR restricted.

ITAR (International Traffic in Arms Regulations) is an American classification for technology that has potential military applications. It does not mean that the technology in question is a weapon as much as an acknowledgment that this specific technology could be used as a component of a weapon.

Restrictions make the use of such hardware quite difficult. Essentially the only documentation that can be easily obtained on the hardware is whatever specifications are given to the team by the manufacturer. Foreign nationals are not permitted in the same room as the hardware without a large number of waivers and paperwork. This became a significant issue because the only full-time electrical/computer engineer on the team is Canadian.

2.4.3 VxWorks Work

There was a significant amount of work done in VxWorks prior to the beginning of the MQP. Unfortunately most of this became less useful when the team switched from 6.3 to 5.5.

VxWorks is an RTOS that is developed and maintained by the company WindRiver. It has extensive flight heritage. VxWorks is documented as a POSIX (Portable Operating System Interface) based operating system that runs in several pieces.

The host machine is the development machine. This development comes mainly in two parts: the development of the kernel image that will be used on the rover and the development of the software that will run on that kernel. Development of the kernel is significant because the original image the system starts with using is absolutely as stripped down as possible. The host machine is also the machine where the manual pages can be installed (they are not installed by default. If any person ever wishes to program VxWorks, installation of these files is essential as the operating system is only loosely inspired by POSIX.)

The second main part of the system is the target boards. The target boards are essentially the boards that will run on the rover. The code is compiled and then fed into these boards, normally in machine code format, so that the rover hardware does not have to go through the effort of compiling it itself.

Several other small pieces are involved in the system of transferring code from the host computer to the target boards. An FTP (File Transfer Protocol) server does the actual transferring.

2.4.3.1 VxWorks 5.5

VxWorks 5.5 has 10 years of flight heritage. It is developed with the Tornado IDE (integrated development environment). It is the operating system that the team's radiation hardened boards were designed to use, but is not the latest version. This means that WindRiver (the company that makes VxWorks) does not provide as much support for VxWorks 5.5 compared to the VxWorks 6.x family, but the manufacturer of the boards only provides support making the hardware work for VxWorks 5.5. This mismatch causes a significant issue.

The operating system is stripped down. Since no third party applications are expected to run on a rover, most of the operating systems are designed for complete efficiency and expect the programmer to keep track of most of the placement of data and system priorities. It has no MMU (Memory Management Unit), no separation of kernel space and user space, and permits void * types for pointers (while 6.3 does not permit void * pointers). This can cause many issues when attempting to port code from VxWorks 5.5 to VxWorks 6.3.

2.4.3.2 VxWorks 6.3

VxWorks 6.3 is no longer technically the newest version of VxWorks, but it was at the beginning of the MQP. It does not have flight heritage, nor does any member of the VxWorks 6.x family. WindRiver redid large portions of the operating system for the transition between VxWorks 5.x and 6.x. It includes an MMU that can be disabled and unless this MMU is disabled it will strictly enforce user space and kernel space. This can make writing to RS232 ports difficult.

2.4.4 VxWorks 6.3 Setup / Bootrom

The VxWorks 6.3 operating system had been set up on the development host. This process was not particularly well documented, and so a significant portion of time was spent getting the host to interface with the target boards. An eventual realization was that while one operating system had been given to the team to install and use on the host, (6.3, which is currently the most thoroughly supported by WindRiver) an entirely different operating system had been loaded on the target, VxWorks 5.5. After this was discovered and the two pieces of hardware were put on the same operating system, the problems were resolved.

2.4.5 BSP (Board Support Package)

BSPs permit the computer to control some item of hardware. Normally this is the host computer controlling the target boards. They are best thought of as a loose hybrid between device drivers and libraries that correspond to a specific piece of hardware. They are normally written by the manufacturer of the hardware.

2.4.6 IOBoard

The radiation hardened IO board was a significant challenge. It is essentially a hardened FPGA (field programmable gate array) with a large number of 9 pin connectors on it. It plugs into the VMEbus backplane on both the flight and the development versions of the card cage to obtain its power and information from the other boards.

There was a misunderstanding about the firmware provided for the FPGA. The team was of the understanding when the item was purchased that there was already a gate configuration available so that all of these ports could be used as RS232 and RS422 ports and that that configuration was part of the purchase. In reality this gate configuration was not provided with the board. The configuration provided only had two of the ports working. As the board is a relatively complex object with so many ports this was a serious issue.

Another issue is that the board did not behave as expected. When the team ran simple queries about various statuses the outputs did not match the hardware manufacturer's expected output. They also did not match the outputs expected by the software company or the outputs expected by the resident expert on VxWorks at NASA who runs the same board in his lab. This to date has not been fully explained.

2.5 Summary

Previous to this MQP the LMR team had already made significant progress. The original MAX rovers had been designed, fabricated, tested and demonstrated. Significant revisions were required to the MAX, mainly because of issues with lunar regolith, before it would be a feasible design for the moon. The new concept version, Prototype 1, had been designed, fabricated and tested. Prototype 2 had gone through the same process. In parallel other work had begun on the VxWorks 6.3 operating system to get the hardened stack of boards operational. The team found many challenges in pursuing this track.

This progress set the stage for the work to be done during the project.

Chapter 3: Project Work

Many of the software choices are driven by non-software reasons. This is because NASA's interest for this project is not inherently in abstract areas of computer science, but in getting the LMR to the moon.

The work done for this project is not focused on any one particular aspect of computer science, but is done with the goal of accomplishing whatever was necessary to help the project succeed, given a background in computer science.

Many mechanical properties of the rover influence the software indirectly, such as the form factor of different components. Thermal properties strongly influence the choices of electronic components, and in some cases set a minimum and maximum speed at which objects can be run. Power requirements cap maximum processing power. Selection of components was originally limited mainly to radiation-hardened parts (most of which came with inherent quirks in them that did not assist the project.) The position of the camera is also determined by mechanical design constraints.

This project was a prototyping cycle during most of the MQP. Because of this there are different standards of success than in most other projects. The project is more interested in the ability to determine that a task was more effort than it was worth and to find an alternative solution that could be implemented easily, than by a demonstration in pure determination in solving any given problem.

3.1 Video Compression

Video compression is an important part of the rover. The project hopes to have the technical ability to drive the rover real-time from the earth while the rover is on the moon. The network bottleneck is at the lander to earth transmission that is capped at 128kbps. 100kbps of that has been dedicated to the video. This means that sufficient video data for a driver to drive the robot real time must be transmitted at no more than 100kbps. It must also be done live with minimal compression delay as there is already a 4.2 second round-trip delay in the time it takes the radio waves to travel the physical distance from earth to the moon. This was already considered a sufficient challenge for the driver and danger to the rover.

The low position on the rover at which the camera was mounted increased the challenge presented by video compression. This was unavoidable, however, as if the camera was mounted off-center vertically the rover would be better when driving on one side but far worse when driving upside down. The rover must be able to drive upside down because there is no guarantee of which way it would land when deployed from the lander. If a self-righting mechanism was installed it would increase mass budget, power budget, size budget, and add failure points. This was considered unacceptable by the project team. The height of the rover could not be increased because of the form factor constraints from the lander and launch vehicle, and the fact that it would raise the center of gravity.

3.1.1 Commercial Software Video Compression

Commercial software video compression products are not a good candidate solution, for several reasons. Most commercial video compression programs are not designed for the same operating systems as used in space. Porting the video compression software was not feasible within the time constraints. If porting the software somehow became feasible, the algorithms may not be particularly suited for the designated task.

3.1.2 Standard Video Compression Algorithms

Many advanced algorithms in video compression make assumptions about the scenes that they will be compressing. These assumptions were not expected to hold true on this particular project.

Some video compression schemes use what is effectively the equivalent of electronic multiplexing in software by combining the images of several frames in a row together. This solves the wrong problem on this mission. The greatest constraint on this mission is the lag. Because of this, the driver is only interested in the most recent frame. This increased frame rate might be useful if the compression method did not add lag to the transmissions (first by adding another step it must wait for the computer to finish and secondly a “multiplexed” image cannot be sent out until all the images it sends have been collected.

Most algorithms are made for a set of moving objects on a still backdrop. Imagine actors in a movie walking around a scene. Most of each frame in a given scene is simply repeating the existing backdrop and overwriting a few square pixels where the characters move.

Contrast this to the scene the rover was expected to see. Since the camera is mounted at a fixed point to the absolute front of the rover whenever the rover turns the camera will swing dramatically.

Since the entire rover is fairly low to the ground, whenever the rover goes over a small bump or incline the camera will also swing. This means that from the perspective of the video compression software it looks like the entire background is changing nearly every frame, making this a nearly pathological test case for most video compression algorithms.

Some video compression algorithms compensate for this phenomenon. A few better quality algorithms can realize that a camera has shifted, move the matching areas of the picture accordingly, and then only update the “new portion” of the picture. However, the camera is placed low on the rover (due to the mechanical constraints) and the vehicle was expected to be driving at high speeds. This was expected to cause so much change in each frame that the compression task would still flirt with the worst-case scenario, which is roughly equivalent to compressing each frame as an individual image.

3.1.3 Move to Single Picture Compression

Eventually single frame compression (picture compression) was chosen for use over compression that takes information from the relative frames around it (video compression). It was discovered that the M750, which had been ordered long before any of the current core project team had been hired, only did single-frame compression in its current state. The M750's FPGA could be possibly re-flashed for use as a video board if the team wrote a complete new gate configuration or an alternative board could be purchased. However the current project gained so little from current video compression technologies that this was hardly considered a reasonable expenditure of the team's funds and effort.

3.1.4 Sending Back Outlines

Sending back large portions of the picture as strictly blank or default values and only sending out the outlines of objects would save a significant amount of space.

The method to compress images in this manner is relatively simple. The brightness of all of the pixels in the image is stored. Then the derivative is taken of this data. The resulting highest points represent the points of greatest change in the image. These points generally represent hard edges because on hard edges the darkness and lightness change abruptly. The area around these highest points is remembered, and the rest go to a simple blank default color. This resulting image is far easier to compress and the important parts of the image can be sent back in much greater relative quality.

On earth additional work can be done to recover some of the image. On earth, the image can be reassembled with effectively any computer and nearly any power requirements. An earth computer can infer what color to “color in” the blank spots on the photograph to permit a driver to drive using a semi-realistic image.

3.1.5 A Problem: Variable Bit Rate Compression with a Fixed Frame Size

Currently the team is locked into using a fixed frame size. This is because of the physical limitations of the lander's radio. The lander has a fixed frame size on the radio (which we are permitted to decide but cannot change at runtime). On the moon the lander will gather packets to fill a “frame” and then transmit the entire “frame” at once to the earth. For the purposes of driving we are only interested in the most recent image because our greatest concern is the lag. This means that if there is a more recent image we are not interested in transferring the old one. Thus our best option is to send one image per frame. If the packet the rover sends with the data does not fill the whole frame, the lander will either wait for the frame to be filled, or fill the rest with empty data and send that. This is either a waste of bandwidth or an increase of lag. Because the team currently gains nothing from a variable packet size with a fixed frame size and there are serious disadvantages the team concluded that it would be best to have a fixed packet size (currently, one packet matches one frame).

Variable bit rate compression means that simpler images will have a smaller resulting file size while more complex images will have large ones. This is because most sophisticated algorithms will compress only the simpler parts of the image and leave the more complex ones. The more complex the overall image is the less “compressed areas” will result from the image compression process and so the larger the resulting file size. All of the compression schemes recommended so far are variable bit rate compression schemes.

Remember, however, that even though we can sometimes compress images to be smaller than normal under certain circumstances, all images must be eventually transferred through the lander's radio with its fixed frame size. This means that if an image is compressed beyond a certain threshold, no actual excess space would be saved as the rest of the space the image was transferred with would simply be blank spaces. This threshold must be set fairly high to accommodate worst-case scenarios in compression. This means that variable bit rate compression schemes are theoretically not optimized when used with fixed length data packets if each packet corresponds to one image.

This situation does not mean that the variable bit rate compression ideas are useless. If LMR ever flies it may fly on a lander using abnormal hardware that would not cause this problem. The team is considering methods of attempting to change the frame size at runtime. Some compression schemes (such as the outline-only method), while producing variable file sizes, still constantly produce incredibly smaller image sizes. Thus it may be worth it to take that algorithm's worst-case scenario case, make that the frame size, and deal with transmitting empty data. It is a less optimized approach but quite likely to produce better results. Despite this possibility, the team at this point started seriously looking in to non-variable bit rate compression ideas.

3.1.6 Non-Variable Bit Rate Compression Tricks

The first method discussed was sending black and white pictures by default. The camera for the hardened stack is a color camera. It was purchased because of the potential interest of being a science payload to obtain a full-color photograph of the lander and whatever experiments the lander is running. Most images of the moon, however, will be white or gray regolith against a black sky. This means that for default when driving if the rover cannot see the lander the full color image can be easily stripped to a four-bit gray scale without the human driver noting any loss of data at all.

Another method was cutting the sky out of the picture. Everything above the horizon of the moon is of absolutely no interest to the mission for navigational purposes. The main issue with this idea is that if the rover is on a relatively hilly area of the moon the positioning of the horizon line would move. This would result in a variable amount of each frame being cut and thus the entire bit rate would vary. This idea is still worth mentioning for its usefulness over relatively level terrain.

A third proposed method was to simply use a variable bit compression scheme and vary the compression level so that the resulting file size was always the same. This is, when applied, generally fairly impractical as the resulting file size of a variable bit rate compression algorithm can not be known until the compression process on the file in question is finished. If a specific target resulting file size is desired, it will probably take multiple compression attempts to find roughly the correct file size. Unless the rover has a phenomenal amount of spare computing resources available, this system will probably add too much lag to the transmission to make it a viable plan. However, if the rover's computer is changed to one with better computing power, or if the rover gains use of computing resources on the lander this plan might be worth revisiting.

3.2 Various Code Architectures

The original code design went on a polling system. This was because it would be easier to debug the specific parts of the program if the code continued to run deterministically and identically every time it was run. The code was divided into motor controls, radio, and video and would poll through each one in series.

The second iteration of the code design was in reflections. This also used a polling system by the nature of the reflections architecture. Functionality was divided instead this time mostly into hardware pieces and ideas that interfaced various items of hardware together.

The third iteration is not planned to be on a polling system. This is because on Prototype version 4 the transmission time of a single frame of video is long enough that the motor controllers have sampled and stored a large amount of data before the end of the transmission. This backup is not fixed because with the polling system the computer only sends one item of data at a time before going back to poll video. This is what spawned the original interest in benchmarking threads against processes in VxWorks.

3.3 VxWorks

During this particular development cycle VxWorks took a lower priority than the demonstration and the associated tasks. It did, however, still take up a decent amount of the team's effort.

3.3.1 Version 6.3 to 5.5

Not long before the demonstration the choice was made to revert from VxWorks version 6.3 to VxWorks version 5.5. This was done for many reasons. First, the hardware that the code was eventually supposed to run with was designed to be used with VxWorks 5.5. AiTech did not offer either a BSP or another form of a compatible version to 6.3, nor any experience in the operating system of 6.3.

The motherboard appeared to have full functionality in 6.3, but the IOBoard never checked out on diagnostics. It did not behave in the same manner as the manufacturer expected, the setup walk through expected, or the resident expert on the particular hardware at NASA expected. VxWorks 6.3 and 5.5 are tremendously different. While none of the challenges were unsolvable, they would demand more time and manpower than the team expected starting over in VxWorks 5.5 to take.

There were other advantages to using VxWorks 5.5 over VxWorks 6.3 in addition to the hardware compatibility. Pioneering a new Operating System into flight did not seem a good allocation of resources when the LMR does not gain any additional functionality from this new operating system. Porting the provided resources from VxWorks 5.5 to VxWorks 6.3 proved to be far more difficult than the team had originally been led to believe. VxWorks 6.3 has strictly enforced separated kernel and user address space and methods while VxWorks 5.5 does not.

3.3.2 Board on Fire

After switching to VxWorks 5.5 there was a significant delay. The operating system installation, target and host setup had been completed and then that area was turned over to another team member. This team member reported that the IOBoard appeared to be functioning aside from the fact that smoke came out of it when it was used. This halted software development in VxWorks for some time as the electrical team had to look at the connections on the ports that had been soldered on for debugging to determine why smoke was coming out. Eventually it was discovered that an DB9 connector which had been soldered onto an RS232 connection on the VMEbus on the development card cage for diagnostic purposes had a short.

3.3.3 5.5 BSP

Another challenge that arose was the BSP for VxWorks 5.5. The operating system did not appear to have a BSP for 5.5 in it for the boards that we had. Instead we took the copy of the BSP that had been provided to us by the maker of our boards for porting to 6.3 and forced the Operating System to acknowledge this as a valid BSP.

3.3.4 Threads vs. Processes

Once the demonstration was completed research was done in finding new ways to do the system architecture. It was obvious that the process would benefit from either a multi-threaded system or a multi-process system over the current prototyping flat polling system.

Normally the line between these two is quite clear. Threads are easier to switch between, have a shared memory, and are harder to backup if they fail. Processes by comparison take more overhead to switch between and do not have a shared memory. They are maintained by the operating system instead of threads that can be maintained either at the user level or with the assistance of the operating system.

The line between these two approaches has been severely blurred by VxWorks. Processes in VxWorks are described as “lightweight processes”, citing that this way they took up less space and had fewer overheads when they were switched between. This is confusing, because that analogy is frequently used to describe threads. To figure out the real differences between these modified processes and threads in VxWorks a series of benchmark tests were designed. Unfortunately, before the tests were executed the simulator broke. Repairing this was a much higher priority than computing benchmarks.

3.3.5 Other Issues

There were many issues with VxWorks that, while not technically difficult, consumed a lot of time for team members.

The VxWorks 5.5 installation is on roughly 25-30 CDs. Not all of these CDs are needed but no instruction was provided as to what order the CDs were supposed to be installed in. If there were three CDs (for example A B and C where A was suppose to come before B which is suppose to come before C) there was no way to tell which came first. The system only would say if a disk was skipped when the user attempted to add in the disk that was missed. For example, if disk A was installed and then disk C, the system would not complain until the user tried to install disk B. At this point all the disks needed to be uninstalled and the process would start over. This slowed the installation process significantly.

Another issue was that one day the VxWorks 5.5 simulated target system appeared to stop working. It had been installed and configured by a former member of the team who was not on the team at the time of the issue. This former team member could not remember how to fix it. Later when there is time the team hopes to devote some time to debugging it but currently there are higher priorities.

3.4 Prototype 3: Narcissus

Prototype 3 was designed mostly as a demonstration, but due to problems with machining prototype 2 it was the first rover that had a full onboard computer of the series (the PC104). It was also much longer than the previous two rovers and much heavier. Since it was the first rover to have a full onboard computer there were several features that were added specially to it.

3.4.1 Computer Access

Prototype 3, when being worked on in the lab is treated somewhat like a computer in a rather oddly shaped tower. A monitor, mouse and keyboard can be plugged in when the lid is open to ports on the PC104.

When prototype 3 is being tested or is fully assembled none of these ports are accessible. The case is designed to discourage lunar regolith from entering so there is no room to slide them into the case. The ports do not come out the side of the case because this opening in the case would be considered an additional failure point for lunar regolith to enter. Additionally, mechanical staff does not permit holes in the electronics box for ports, as it would also undermine the structural integrity of the walls.

3.4.2 50 Watt Motor Controllers

Prototype 3 is by far the largest of the rovers in the family, thus it was much heavier and required that a different set of motors and motor controllers be used as well (the 50 Watt set). An interesting challenge came up in the use of the original 20 Watt motors and motor controllers. They are in fact amp-limited, and so if not run at maximum nominal voltage from the system the output of the motors will not correspond to the published specifications. For this reason the team found it electrically easier to switch to a different set of motors and motor controllers, (which were already in the lab) than to change the nominal voltage of the system.

3.4.3 Battery Monitoring System (BMS)

The Battery Monitoring System was created because of the complexities of charging Lithium Batteries. The cells must be individually charged and monitored for overcharging. Discharging batteries are also generally monitored, because if something should go wrong and no alerts are signaled before the actual unit begins combustion the resulting fire is extremely dangerous.

Originally, the concept was to have the BMS chip as part of the form factor of the battery itself.

Unfortunately, during an unexpected staff change the individual who was negotiating with the battery company left suddenly and a great deal of communication was lost. The result was a misunderstanding about the form factor of the BMS and the team was mailed a large box of additional chips and wiring that is roughly the size of the battery itself instead of an integrated chip. This monitoring system was part of the reason for the expanded form factor of Prototype 3. Unfortunately, the BMS box caught fire and could not be used for Prototype 4. Eventually the team hopes to integrate the BMS chip onto the battery itself as early as Prototype 5.

3.5 The Demonstration

A deadline on the basic mobility and functionality of the rover had been set for August, 2007. Director Pete Worden (director of NASA Ames Research Center) was going to see the rover at this time and evaluate the project team's efforts and progress.

The rover was to be tested in an off-site lunar environment analogue site. The area was selected to be dry, relatively void of plant life, and rocky. Unfortunately due to a dispute between the NASA legal department and the lunar analogue site over liability in the case of injury to a team member the demonstration was delayed until September 2007 so that in the extended time an agreement could be reached.

Communications between the site and the RMOC (rover mission operations center) were a decent approximation of the actual communications system on the moon. A pair of satellite dishes linked the communications network through a geostationary satellite. One end of this network was in the RMOC and the other was at the extra-terrestrial analogue site.

Preparing for this demonstration absorbed a great deal of the effort of the team during the time of this MQP.

3.5.1 RS232 to TCP Issues

A great debate exists over the use of serial communications versus TCP and UDP communications in spaceflight. Most spacecraft that have flown, most spacecraft that are flying, and most spacecraft that will be flying soon still use serial communications. It is considered inevitable that serial will be replaced in the near future, but for this mission it was decided that the team would go the traditional route and use RS232 radios.

On earth, serial communications are considered outdated. The NASA ARC communications team's satellites run strictly on TCP protocol and the RS232 technologies are considered so outdated that they no longer maintain converters between RS232 and TCP. This is understandable because most of the research currently done at NASA ARC is either high-level theoretical research that will not be applied to actual missions for approximately a decade (at which point TCP will probably be the system used), wind tunnel testing which does not normally require a communications system, or UAV (Unmanned Aerial Vehicle) work which tends to use the more updated earth technologies.

The LMR was not aware that the communications team no longer carried RS232 to serial converters until less than a week before the demonstration. Vendors who sold RS232 to TCP converters had a 6-8 week shipping time that meant that they would not arrive in time, and so we had to build one ourselves.

The first attempt was to emulate the entire RS232 to TCP conversion using software. An individual affiliated with the communications team wrote a large Perl script that was supposed to do the conversion. This script was installed on two computers. One computer was placed in the simulated lander between the UHF radio and the simulated lander to earth communication. The other was placed at the other end of the communication directly before the command computer in the RMOC. Unfortunately, the Perl script did not work when larger amounts of data were transferred across the system; the team had a limited amount of time on the test equipment to get it to work, and neither the individual who wrote the script nor any other programming members of the communications team were available during the scheduled time on the test equipment. The team could not figure out how the Perl script was supposed to work during the time on the test equipment.

That night the team at CMIL examined some other alternatives. The UHF radios had a configuration port in the back which takes its commands in TCP. With some experimentation it was discovered that it was not hard to force the radios to take in TCP data through the configuration port and then retrieve the data on the other radio's configuration port. This is obviously not a documented feature, nor an intended use of the programming port, but it permitted the rover to interface to the existing hardware provided by the communications group. The software that outputs to the radios was rewritten so that it transferred in TCP instead of in RS232, and the team prepared to gain some additional testing time on the communications hardware the next morning to make sure this new system worked. The new radios and code were proven to work, however due to other issues with the rover, (see Other Challenges) the entire end-to-end communications system was not tested until after deployment.

3.5.2 Hard Drive Safety During Rover Operation

A concern arose about using an ordinary laptop hard drive in a rover that was going to go over bumps. Laptop hard drives are not designed to be able to sustain serious vibration even in storage. They are considered to not be prepared to be read from and written to during vibration. The team was concerned that the heads might crash through the platters if a serious impact occurred, and was requested to find some way to prevent this problem from happening.

The first series of plans involved attempting to use drives which did not have moving mechanical parts. These parts have many benefits aside from being more vibration friendly. They also consume less power, are faster, and are less likely to be permanently damaged by an unexpected power loss.

The first plan was to use a memory disk (similar to those used in digital cameras). However the documentation for how to make one of these drives a bootable drive was unclear. It was not until after normal business hours had closed that the team discovered this method required several tools that were not in stock (most difficult to find was a 3 ½ inch floppy drive). The demonstration deadline was approaching fast, and the mechanical team could not continue their tests until this problem was solved. The software team needed to get something functional in less than 24 hours.

The next attempt was to make a USB flash drive a bootable main hard drive. Making a sector of the flash drive bootable was trivial as was installing the Operating System of the PC104 onto the board. The issue arose that the computer at startup did not know how to read a USB port as a bootable device without instructions from another disk (for example a hard drive). It was hypothesized that the boot drivers for the Operating System could be modified to permit the computer to boot off a USB drive, but with the demonstration approaching the team did not have the time and effort to devote to such a project at that moment.

The final solution for the demonstration was to use “bumper beads.” These objects are essentially a small chunk of a rubbery plastic material with a small section of each end threaded. The two threaded bits do not ever touch each other. This object is made to absorb most of the shock that is created by the rover going over bumps. While this was a sub-optimal and temporary solution it was considered to be the right thing to do for the project. The number of hours before the demonstration was ticking down and the mechanical division had many tests of its own that it had to pass before the demonstration started.

3.5.3 Other Challenges

Many smaller challenges arose while preparing for the demonstration. While these were not particularly technically challenging tasks, they were important to the success of the mission and absorbed a significant amount of the team's time and effort.

Unfortunately, at some point during the transfer of the main ground station computer from the RMOC to the communications station for the first equipment testing, the hard drive on the computer died when the car drove over a significant bump in the road. The code that it was running had been backed up to SVN but the Operating System needed to be reinstalled and the system needed to be reconfigured for use as a main ground station.

Many other computers also needed to be configured to be used with the reflections architecture. They were required for use as RMOC pieces and computers in the lab for parallel development. This was not a particularly difficult task but it was time consuming.

Another issue was the “unregistered driver” problem. After it was discovered that the radios could be tricked into taking in TCP data and outputting TCP data the rover was taken back to the communications station for another round of equipment testing. When the entire system was connected, the rover appeared to have an electrical failure. It would boot, the operating system splash screen would show, and then the screen would go black and the entire system would shut off suddenly. Upon further inspection it was found that nothing in fact was wrong with the electrical system. When the team dove into the operating system it was discovered that a driver labeled “unregistered driver” had been automatically installed at some time between the last time the rover had been used and the boot up with an error in it. Removing the driver solved the problem. This problem continued to happen almost every time the team used a new monitor to plug into the system for diagnostics, but each time removing the “unregistered driver” solved the problem.

While the rover was being tested for electrical issues during the first time the team encountered the “unregistered driver” problem, a solution was needed to test the communications equipment. Waiting for the rover to come back from testing and potential repairs was not an option because the communications system was scheduled for deployment to the extra-terrestrial analogue site later that day. Instead, an ordinary laptop had reflections installed on it and was forced to emulate the rover inputs and outputs for the testing.

During the live demonstration for Director Worden there was a communications failure. This happened because pieces of the communications system had been run over by a truck. Some of the communications system was placed on a small street around the side of an office building at NASA. Although this street has been blocked off from traffic a truck decided to ignore these signs and drive on the street anyway. Thankfully the only part permanently damaged was a cat5 cable. The team strung a new cable down the outside of a two-story building under the vacuum tanks and into the communications system to solve this problem.

3.6 Study of Demonstration Results

Overall the demonstration was considered to be a success. Director Worden was pleased with the progress made. There were, however, some lessons that could be learned from the demonstration that the team spent some serious time studying in order to better the project.

3.6.1 Prototype 2 Unresponsiveness

During one of the days of the demonstration Prototype 2 became relatively unresponsive in the lunar simulator and the video transmission dropped in quality. The team recharged the batteries on Prototype 2 and attempted to replicate this problem but was unable to. It was hypothesized that this could be for one of two reasons.

The first possibility is that there was interference in the transmissions. Prototype 2 had only been previously tested in the lunar simulator at night between the hours of approximately 10PM and 3AM. The demonstration where the unresponsiveness took place was at approximately 11AM. Since the simulator was positioned near a relatively large office building at NASA it is possible that during demonstration hours there was a heightened level of the use of other electronic devices which could have interfered with the telecommunications between the RMOC and Prototype 2.

The other possibility is that the batteries on Prototype 2 were not completely charged. After the demonstration the rover was placed directly back on the charger before the level of charge on the battery was read. This means that the team cannot rule out the batteries being not fully charged during the demonstration.

Which scenario occurred is relatively simple to determine. If team members acquire some spare time at a later date, tests can be run during business hours with a confirmed full battery. If no significant interference is noticed the team is fairly comfortable with the assumption that a low battery was the cause of the issue.

3.6.2 Prototype 3 Unresponsiveness

During the second day of the demonstration, Prototype 3 became unresponsive. Team members who held the rover close to an ear could hear the CPU and fan turn on, the radio would become operational, and then the whole system appeared to be functioning normally. The power leads were drawing appropriate amounts of power. When pinged by other computers on the network, it did not respond. When the lid was opened nothing appeared to be wrong but the team also could not see much. This was because to get to all the hardware components requires some serious disassembly that the mechanical team did not want to have during the demonstration.

Upon opening the case after returning to the lab a loose wire was found. However, the wire was not connected to anything. The team is relatively confident that this wire did not cause a short during the demonstration because the rover continued to draw the correct amount of power.

The robot was then connected to the computer after the lid was removed. The operating system failed to boot and at the menu options screen the menu did not respond. The system eventually attempted to boot again and this time simply shut off after the splash screen. This was fairly reminiscent of the “unregistered driver” problem so the machine was rebooted, the drivers were examined, and a new “unregistered driver” was in fact present. The driver was removed and system continued to function normally after that.

This situation does not, however, explain Prototype 3's unresponsiveness during a short period of the second demonstration by itself. Previously the unregistered driver error had only shown up at reboot time. This error of becoming suddenly unresponsive happened after the rover had been left on for an extended period of time. It is possible that the rover heated up during this time. Air is an insulator that was not accounted for during the design of the rover but was present at the demonstration. If the rover heated too much and the radio shut off it would stop responding to pings. Then when the system was rebooted the “unregistered driver error” could take over and prevent the rover from becoming functional again.

3.7 Hard Drive Safety Precautions

After the demonstration the team began researching a more permanent solution to the issue of protecting the hard drive. During the attempts to protect the hard drive for the demonstration various people pursued multiple tracks at once in an attempt to circumvent the strict time limit. This left the team with three options effectively: getting the camera memory card to work, modifying the operating system to accept USB devices as bootable objects, and the use of a solid state hard drive.

Solid-state hard drives pose many advantages over normal hard drives and the camera card. With no mechanical moving parts they consume far less power. They also retrieve data faster. In this particular application they were considered a desirable concept because of those two reasons and the fact that they do not have heads to crash through the platters during vibration. The camera cards do not come in large sizes (generally no larger than two to four gigabytes). During this stage of testing the operating system had not been trimmed down to minimum size. Two gigabytes would be more than sufficient on a target board but the PC104 was being used both as a target board and a development board. This meant that the much larger space afforded by a full solid-state hard drive (about 16 gigabytes) was quite appealing to the team.

The only serious drawback is the write life of the hard drive. Solid state hard drives are reported to wear out after fewer individual bit flips. To solve this problem it was determined that the use of the hard drive would be split into two parts. All of the reading would be done from the solid-state hard drive. Any file that would be written to would be copied to a 512MB thumb drive and then all further modifications would be done to that file instead. This solution meant that the life of the solid-state hard drive would become long as the team would only write to it when changing the program that runs the rover. The 512MB thumb drives are cheap enough to be considered fairly disposable (about 5 dollars). This solution was also strongly favored because having one read-only disk and one read and write disk is part of NASA's normal practices. In space the read-only memory is generally flashed memory that the system could not modify even if it wished to. This makes it more tolerant to radiation. It is also favored because even if the memory is not flashed in for some reason it means that if the system has a power loss during writing, none of the startup data is corrupted. If the system could write to files on the drive that are normally made read-only, a write function interrupted half way through would make the written file unusable. This means the system is likely to be unable to start up from scratch and become fully functional again. When the drive is read-only, however, the data is not being modified, so if the power is lost during a read the file is not corrupted. The system can instead load everything off the read-only memory as if from the beginning. Having the two separate drives (the solid state and the 512MB USB) set up this way together will be a good simulation of the final flight hardware. This means that if code is written for this architecture now it will be easier to port to the final flight state.

3.8 Prototype 4

Prototype 4 began progress after the demonstration of prototype 3. It does not currently differ from prototype 3 in software; it was added to the family so that the mechanical design could be reworked. The main mechanical difference is that the size of the electronics box was reduced to less than a fourth of the original. It became the group workhorse robot and ironed out many mechanical issues.

3.9 Prototype 5

Prototype 5 is the design being worked on as of January 2008. It has a smaller electronics box than Prototype 4.

3.9.1 New Hardware

During the design for Prototype 5 it became painfully obvious that new hardware was needed. By this point the electronics box had shrunk so much that it was smaller than the VxWorks cards themselves (not counting the motor controllers, batteries, and other hardware that need to fit in the box.)

The overall expectations of the project were set higher for Prototype 5. NASA wished for the resulting reusable design to cost still less to manufacture, be smaller, and be lighter than previous incarnations.

Alternatives to using commercial off the shelf radiation-hardened parts were researched. The lower budget made buying radiation-hardened COTS parts far less feasible.

One option is to use a series of non-radiation hardened processors that all compute the same choices and then vote on what the results are. This system has been used historically with radiation-hardened chips for large important systems. Examples of such systems are human life support systems on the space shuttle. Other satellite projects for LEO and GEO have proposed using unhardened chips for voting system but to date the team has not found any examples of this system that have flown.

Another option that was considered was to run the entire system off an FPGA. Radiation-hardened self-healing FPGAs are a topic of modern research and while none are currently found to be proven on the market (or with flight heritage) they are a promising technology. They too use the process of executing the same instructions several times and then voting on the results to detect errors caused by radiations.

3.10 Physics

Another important task was to assist mechanical with calculations and simulations. Because the team is so small and many of the computations are complex, critical or both assisting with physics calculations is important.

Some of the more complex physics problems are solved by writing computerized Monte Carlo simulations. A Monte Carlo simulation is essentially setting up a series of rules (in most cases a basic physics engine), putting in randomized inputs, and recording the outputs. The most famous experiment of this type was to draw a circle that touched the edges of a square with a known area. Then darts were randomly thrown and the percentage that land inside the circle was calculated. This let the person estimate what percentage of the square the circle took up and thus what the area of a circle was relative to its diameter. This same idea can be applied to more complex ideas as a relatively simple but useful test.

One of the more complex challenges is the issue of spacing various moving components of the rover at an optimal distance that lunar regolith is unlikely to stick inside long enough for a significant mass to build up. To solve this problem the solution was to write a basic physics simulation system that would simulate the movement of the particles in the moving components. Additionally complex physics are added in as the simulation requires (such as inelastic collisions, and variable size particles).

3.11 Summary

During the project significant progress was made. Video compression was researched and, the team eventually discovered that single frame compression was better suited to the project's needs than video compression. The overall architecture of the code was reconsidered and the version of VxWorks was switched from 6.3 to 5.5. Prototype 3 was designed, fabricated and tested. Director Pete Worden observed a demonstration of Prototype 3 and evaluated the team's progress. After the demonstration the team spent some time analyzing the results. Following the demonstration of Prototype 3, Prototype 4 was designed, fabricated, and tested but no significant software changes were made during that iteration. Currently work is being done on designing Prototype 5 and building a physics simulator to assist the mechanical division.

Chapter 4: Conclusions

NASA has many reasons to be interested in using micro rovers. Micro rovers are cheaper to fly if only one is being flown. If NASA wishes to fill a large rover's payload space with several micro rovers the micro rovers are cheaper to manufacture. Micro rovers in groups also provide additional robustness, scientific advantages, more complex behavior, and ease the design constraints on the launch and lander vehicles.

Previous to the MQP a great deal had been accomplished. The MAX rover series had been designed, fabricated, tested and demonstrated to Director Pete Worden while in the Atacama Desert. The MAX had to undergo significant revisions, mainly because of issues with lunar regolith, before it would be a feasible design for the moon. A new concept version which embodied these changes, Prototype 1, had been designed, fabricated and tested. Prototype 2 had undergone the same process. In parallel with the development of prototypes 1 and 2 other work had begun on the VxWorks 6.3 operating system to get the hardened stack of boards operational.

During the project the team made significant progress. Research was conducted on video compression and a decision was made that single frame compression suited the project's needs better. The overall architecture of the code was reconsidered. VxWorks was switched from version 6.3 to 5.5 because of issues with the hardware. Prototype 3 was designed, fabricated and tested. A demonstration was conducted of Prototype 3 at which Director Worden evaluated the team's progress. After the demonstration some time was spent analyzing the results. Prototype 4 was designed, fabricated, and tested but no significant software changes were made during that iteration. Work is currently being done on designing Prototype 5. Other work involves building a physics engine to assist mechanical simulations.

The MQP portion of the project was a success. Requested deadlines were met and the demonstration for Director Worden received a favorable review.

Chapter 4: Appendices

Appendix A: List of Relevant Acronyms and Abbreviations

*nix: Unix and Linux

ARC: Ames Research Center

BSP: Board Support Package

cat5: Category 5, Cable and Telephone

CMIL: Carnegie Mellon Innovations Lab

CMU West: Carnegie Mellon University West

CMU: Carnegie Mellon University

Code EN: (Ames) Education Division

Code ES: (Ames) Special Projects Division

Code QS: (Ames) Occupational Safety, Health and Medical Services, System Safety and Mission Assurance Division

Code TI: (Ames) Intelligent Systems Division

COTS: Commercial Off The Shelf

DSN: Deep Space Network

Ebox: Electronics Box

ESD: Electro-Static Discharge

FPGA: Field Programmable Gate Array

FTP: File Transfer Protocol

GEO: Geostationary Earth Orbit

GSFC: Goddard Space Flight Center (a NASA center in Florida)

IDE: Integration Development Environment

ITAR: International Traffic in Arms Regulations

IV&V: (software) Verification and Validation

JPL: Jet Propulsion Laboratory (a NASA center in southern California)

LEO: Low Earth Orbit

LMR: Lunar Micro Rover

MAX: Mobile Autonomous eXploration

MCC: Mission Control Center

MMU: Memory Management Unit

MOC: Mission Operations Center

MQP: Major Qualifying Project

MSL: Mars Science Lab

NASA HQ: NASA Headquarters (in Washington D. C.)

NASA: National Aerospace and Space Administration

OS: Operating System

POSIX: Portable Operating System Interface

RAP: Robotics Alliance Project

RMOC: Rover Mission Operations Center

RTOS: Real Time Operating System

SAM: Student Assembled Microprobe, another name for LMR

TCP: Transmission Control Protocol

UAV: Unmanned Aerial Vehicle

UDP: User Datagram Protocol

UHF: Ultra High Frequency

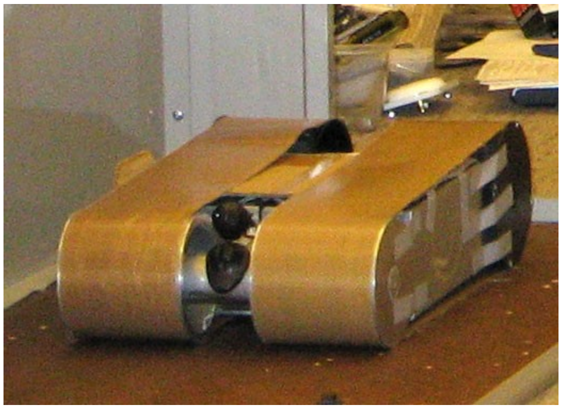
USB: Universal Serial Bus

VMEbus: A type of computer bus. In this project it was used in the card cages for the hardened boards.

WPI: Worcester Polytechnic Institute

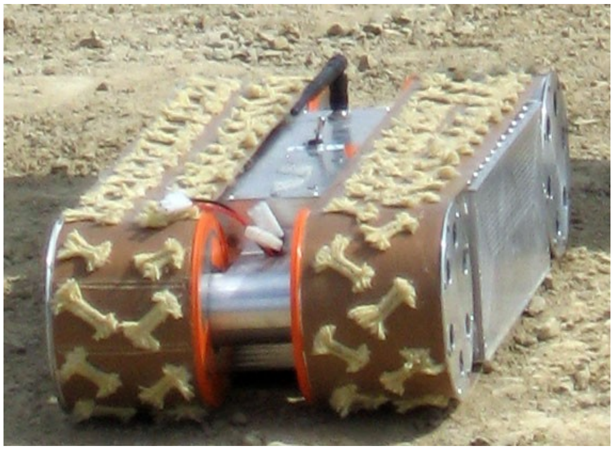
ΛMP: Lunar Micro Rover (Greek)

Appendix B: Images of Rover



Prototype 2 with the wheels taped so they can not move. This was done for a friction test of the belts.

Prototype 3 at the extra-terrestrial demonstration site.



References

Carnegie Mellon Innovations Lab. 2007. Carnegie Mellon Innovations Lab. Silicon Valley California. Online. Available from the Internet. Available at http://cmil.west.cmu.edu/

Carnegie Mellon University West. 2008. Carnegie Mellon University West. Silicon Valley California. Online. Available from the Internet. Available at http://west.cmu.edu/

Code 582 Flight Software Division National Aeronautics and Space Administration Goddard Space Flight Center. 2003. C++ Coding Standards. Online. Available from the Internet. Available at http://software.gsfc.nasa.gov/AssetsApproved/PA2.4.1.3.1.pdf

Department of Earth and Planetary Sciences Washington University in Saint Louis. 2007. Lunar Regolith Breccias. Saint Louis. Online. Available from the Internet. Available at http://meteorites.wustl.edu/lunar/regolith_breccia.htm

Education and Public Outreach National Aeronautics and Space Administration. 2008. Radiation Belt Storm Probes Educational Website. Online. Available from the Internet. Available at http://rbsp.jhuapl.edu/

John Mangels. 2007. Coping with a Lunar Dust Up. Seattle Times. Seattle Washington. Online. Available from the Internet. Available at http://seattletimes.nwsource.com/html/nationworld/2003572876_moondust15.html

National Aeronautics and Space Administration Ames Research Center. 2003. Mars-like Atacama Desert Could Explain Voyager "No Life" Report. Online. Available from the Internet. Available at http://www.nasa.gov/centers/ames/news/releases/2003/03_87AR.html

National Aeronautics and Space Administration Intelligent Systems Division. 2006. Micro rover Technology Demonstration. Silicon Valley California. Online. Available from the Internet. Available at http://ic.arc.nasa.gov/news/story.php?id=361

National Aeronautics and Space Administration Worldbook. 2007. Moon. National Aeronautics and Space Administration. Online. Available from the Internet. Available at http://www.nasa.gov/worldbook/moon_worldbook.html

National Geographic World. The Atacama Desert. Online. Available from the Internet. Available from http://magma.nationalgeographic.com/ngm/0308/feature3/

Stern, David P. and Peredo, Mauricio The Exploration of Earth's Magnetosphere. National Aeronautics and Space Administration Goddard Space Flight Center. Florida USA. Online. Available from the Internet. Avaialable at http://www-istp.gsfc.nasa.gov/Education/Intro.html

United States Executive Office of the President: Office of Management and Budget. 2007. Budget of the united states government: National aeronautics and space administration.