Mars Rover

By:

Daniel Dunn

Colin Shea

Eric Spiller

Project Advisors:

Dr. Brian Huggins

Dr. Alexander Malinowski

Mr. Chris Mattus

For Partial Completion of EE419 and EE 451 December 18, 2002

Project Summary

The Mars Rover Project continues the work of the Precision Robotic Platform designed by Robert Shockency and Randall Satterthwaite and initiated by the 2001 Telerobotic project, in which a third party robotic platform was used.

The Precision Robotic Platform project consisted of designing, constructing, and testing a robotic platform for use in various autonomous mobile platform applications including telerobotics. The system was comprised of two Pittman motors, an H-bridge motor driver, two CPLDs and an EMAC 8051 microprocessor. Variable PWM signals were generated for precise turning and speed control and synchronized RPM data received from rotary decoders kept the platform on a precise heading. Lack of a more powerful CPU precluded operation as a web server and transmission of images from a web camera at a reasonable rate.

The main objective of the Mars Rover Project is to redesign the Precision Robotic Platform for long battery life so it operates for a 7 day period without recharging. In addition, a powerful PC 104 based CPU is used, in addition to an EMAC 8051 microprocessor, to allow the platform to operate as a mobile web and image server. In particular, the PC104 computer runs high level software to control the interface between the user and the Mars Rover. The EMAC 8051 microprocessor translates interpreted user's commands into specific platform operations.

Detailed Description

Functional Description

The goal of this Capstone project is to build a semi-autonomous vehicle to operate on the ECE department's Mars landscape. This vehicle will have similar functionality as its predecessor, Pioneer 1, but the Mars Rover will have a 7 day battery life. In addition to running a web server, there is an Ethernet connection via a wireless card, a camera mounted on the front of it, and is lacking an umbilical cord. The Mars Rover has five modes of operation: wait mode, sleep mode, low battery, charge mode, and user mode. The details of operation are in the paragraphs that follow.

<u>*Modes of Operation*</u>

Wait mode - All systems are powered, except the motors, and the CPU monitors the wireless card for activity from the user. The last image captured from the camera is stored on the hard drive. If the user does not give a command within a specific amount of time, the rover goes into sleep mode, where most subsystems are powered down. The exceptions are the CPU which is active and monitoring battery level and the wireless network card

Sleep mode - The sub-systems are powered down except for the CPU and the wireless network card. If possible, the CPU will run in a reduced power mode. The rover monitors activity from a remote user through the wireless card port, and receives battery status from the CPU. The wireless card only receives data to conserve battery power. The rover remains in sleep mode until it receives a signal that the user wants it to perform a specific action.

Low battery mode - When the battery drops below a set level, the rover goes into low battery mode. At this point, the rover sends an email to Dr. Malinowski telling him to charge the rover and performs a software shutdown to prevent deep discharge of the lead-acid batteries and possible damage. Before the rover shuts down, it saves all information to the hard drive, so on power up the system knows why the rover shutdown.

Charge mode – The operator must press the 'ON' button for rover to power up. Once the rover powers up, it checks to see why it shutdown; if the rover shutdown because of low battery and the charger has been hooked up, then charge mode takes over, otherwise the rover goes into wait mode. The rover remains on the charger and converts to a trickle charge once a full charge is obtained. The CPU monitors the network for any activity; once activity has been detected, the rover acquires a new camera image, issues a wake up statement to the rover, and moves approximately one meter forward from the charging station.

User mode – The user dictates the distance the rover must travel and in which direction. The instructions for the rover in this mode are as follows: move forward or reverse continuously, forward or reverse for a certain distance, and how many degrees to rotate clockwise or counterclockwise. As a precaution, an immediate stop command is used in case the remote user needs to stop the rover from previous commands.

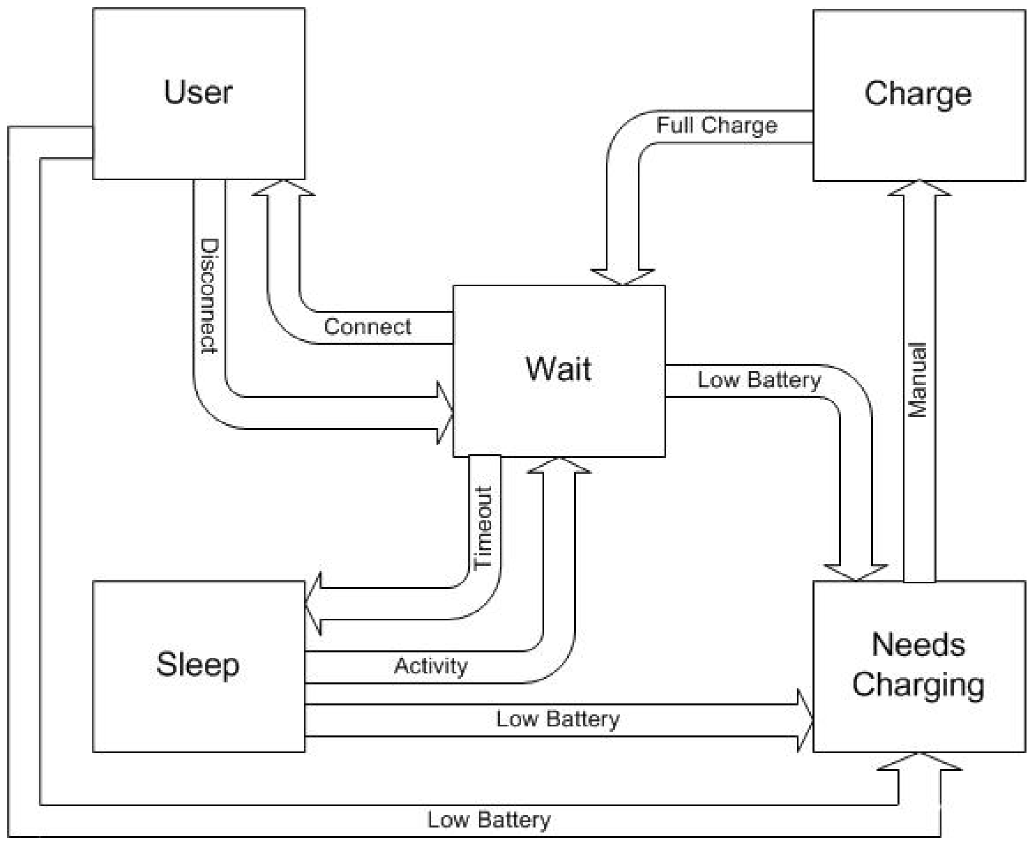

Figure 1 → Modes of Operation Flow Diagram

System I/O

Inputs-

Web/ Software-

Stop- The stop input halts the rover.

Reverse/ Forward motion continuous- The continuous motion command allows the rover to move forward or backward continuously until the stop command is called, an obstacle is detected to be too close, or a maximum distance is reached.

Reverse/ Forward motion distance- The distance motion command allows the rover to move a distance specified by the user either forward or backward.

Clockwise/Counterclockwise rotation- The turn command rotates the rover a specified angle in a clockwise or counterclockwise, from a top view.

Hardware-

ON button – Powers up the rover.

Battery Voltage- Lead Acid batteries are used to power the rover and the terminal voltage is an input to the CPU.

Camera- The camera captures photons that are directly in front of the rover. The image produced by these photons is stored on the hard drive in a buffer.

Front/ Back acoustic sensors- The sensors receive an echo signal, which is the signal that was emitted by the acoustic transmitters bounced off of an object. The distance from an obstacle is proportional to the time between the transmitted and received signals.

Wheel sensors- The wheel sensors track the movement of the wheels. They emit a digital bit stream that is captured by the CPU and decoded to find the direction and distance traveled. Once the desired distance is achieved, the CPU can stop the rover.

Wireless Card- The wireless card provides a link between the internet and the rover.

Outputs-

Web/ Software-

Low battery- An email is sent to Dr. Malinowski telling him to charge the rover. This output also issues the CPU to save all information to the hard drive and shutdown everything.

Camera image- The camera image, composed of photons captured by the camera, is stored in a buffer, and sent to the web interface where the user can view what the rover 'sees.'

Obstacle distance front/back- Through software, the distance away from objects is calculated by measuring the time between the signals emitted and acquired by the acoustic transmitters and receivers. This distance can then be displayed on the users screen to warn of potential hazards. If an object is too close, the rover will stop and will not allow the user to advance in that direction.

Hardware-

Front/ Back acoustic transmitters- The acoustic transmitters emit a signal that bounces off of objects and received by the acoustic sensors to determine how far away an object is.

Vehicle motion- The user enters the direction and distance they want the rover to move.

Wireless Card- The wireless card sends images and data from the rover to the web interface so that the user may control the rover.

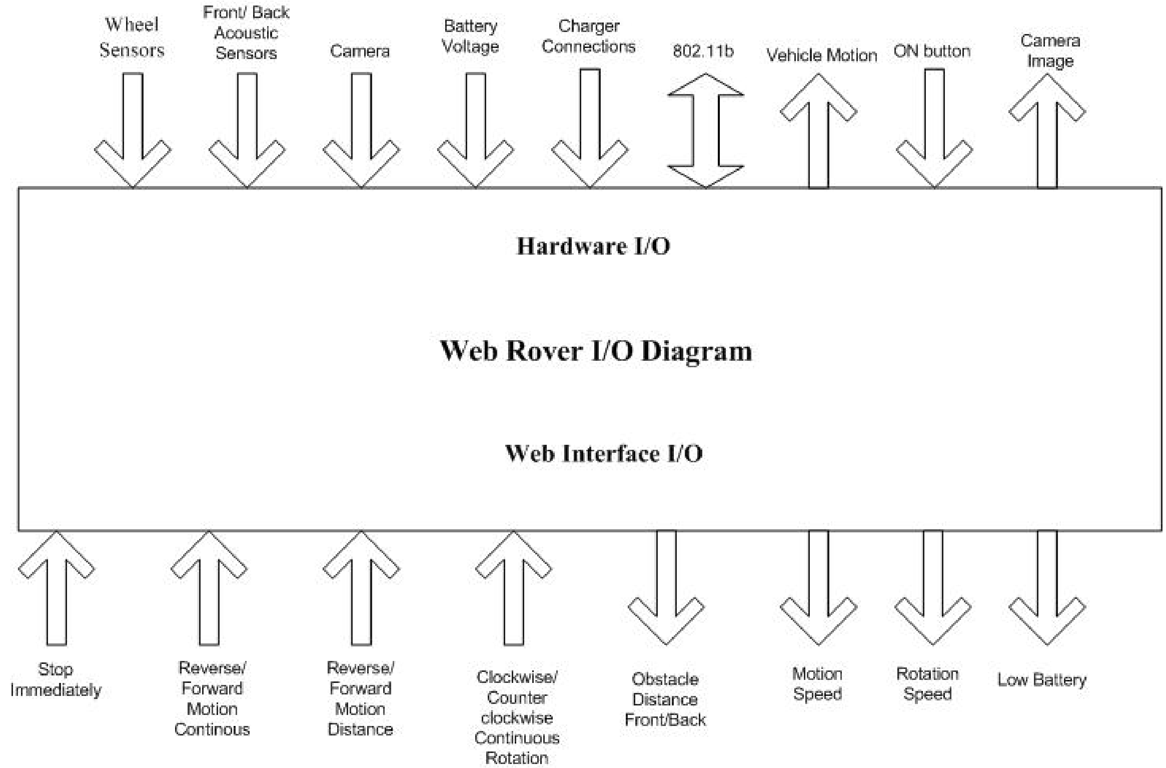

Figure 2 → System I/O

System Block Diagram

An overall system block diagram, shown in Figure 3, displays a detailed layout of the inputs and outputs of the system. The block diagram consists of four sub-blocks: User Computer, Network Card, Microprocessor, and the Embedded System. Each sub-block has multiple inputs and outputs that will be explained in detail.

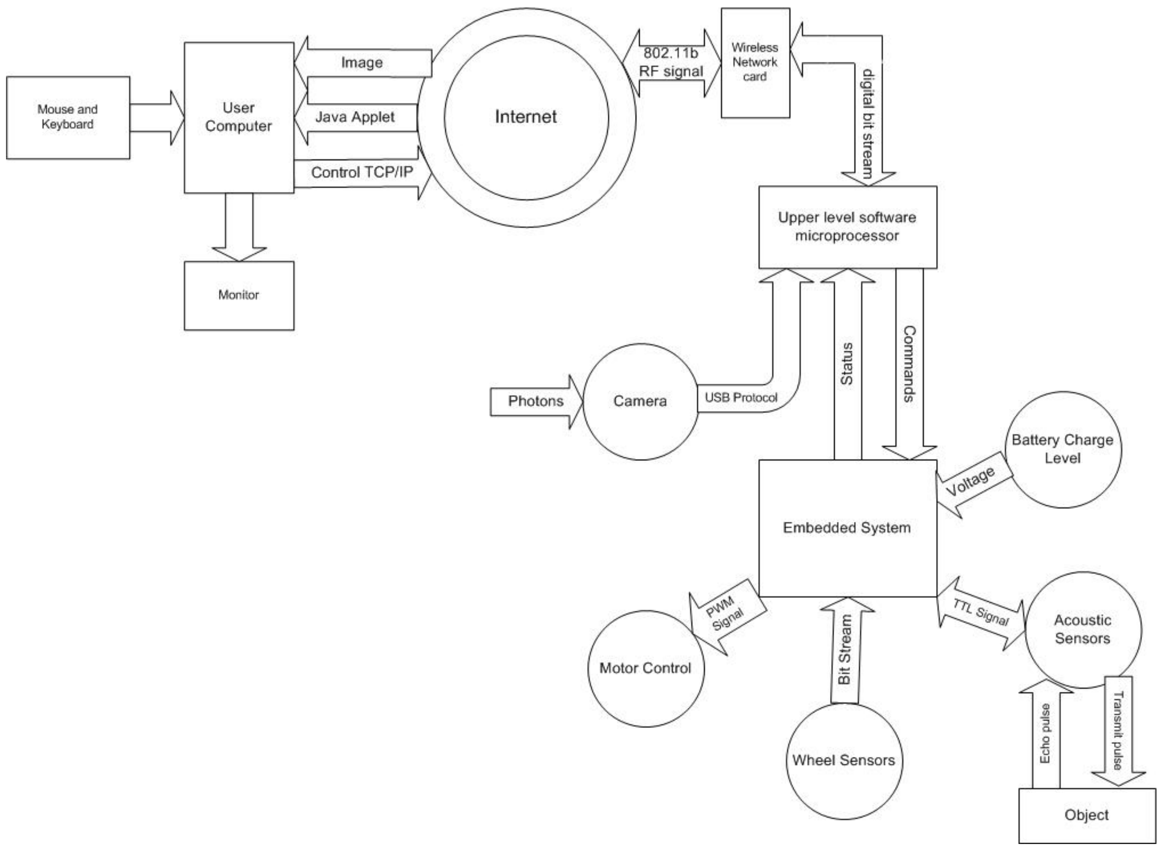

Figure 3
Overall Block Diagram for Mars Rover

<u>User Computer</u>

Inputs	Outputs
Mouse & Keyboard	Monitor
Image from Camera	Control TCP/IP
Java Applet	

<u>Inputs</u>

Java Applet Download – When the user connects to the rover, a Java Applet is downloaded to their computer so they can control the rover.

Image – The image downloads to the user computer so the user can view where the rover is heading. This way the user can direct the rover to avoid object collision.
Mouse click, key stroke – The user controls their computer by using the mouse and keyboard; the java applet will use these to determine how the user wants to control the robot.

Outputs
Control TCP/IP – The java applet will maintain a constant TCP/IP connection as long as the user keeps the window open and doesn't remain on idle for more than 2 minutes. From this connection, the user inputs commands to move the rover.
Monitor – The image is displayed on the user's monitor as well as the java applet used to control the rover.

Function
Wait Mode- The java applet displays the last image captured by the camera and waits for an input.

User Mode- The java applet displays the current image captured by the camera and updates every 5 seconds. The user is able to input a distance to move or a degree to rotate the rover.

Sleep Mode- The java applet displays the last image captured by the camera and waits for an input.

Needs Charging- The java applet displays the last image captured by the camera and does not allow the user to move the rover.

Charging- The java applet displays the last image captured by the camera and does not allow the user to move the rover until the batteries are fully charged.

Wireless Interface

Inputs	Outputs
802.11b/ RF Signal	802.11b/ RF Signal
Digital Bit Stream	Digital Bit Stream

Wireless Network Card-

Inputs/ Outputs
Wireless Access Point- The system will use a wireless access point to send and receive data to and from the internet and wireless card.
802.11b RF Signal – This is the protocol that the wireless card transmits data over. A TCP/IP connection is maintained over the wireless connection.

Function
Wait Mode- Wireless Interface is idle and waits until a signal is sent from the user.

User Mode- Wireless Interface sends and receives data from the user and from the rover. The interface sends the camera image to the user through the java applet. The interface sends information about the motion of the rover in terms of distance or rotation in degrees.

Sleep Mode- The Wireless Interface is idle until a signal is sent from the user.

Needs Charging- The Wireless Interface sends the last image captured by the camera and then shuts down to conserve power.

Charging- The Wireless Interface does not allow the user to access the rover by not allowing signals to be sent or received during this mode until the batteries are fully charged.

Upper Level Software Microprocessor

Inputs	Outputs
Digital Bit Stream	Digital Bit Stream
USB Protocol	Commands
Status	

Upper Level Software Microprocessor-

Inputs/ Outputs

Digital Bit Stream – The digital bit stream is the connection between the wireless network card and the Upper Level Software Microprocessor; the bit stream carries all the information that the user inputs to the rover, and all the information that the rover sends to the user.

Inputs

Camera – The camera captures photons and translates them into a 320x280 16-bit color picture, which is transmitted via a USB protocol to the upper level software microprocessor.

Status – The embedded system sends the upper level software microprocessor status concerning the distance to an object, battery voltage, and distance the wheels have traveled.

Outputs

Commands – The upper level software microprocessor sends commands from the user to the embedded system to control the rover.

Function

Wait Mode- Last image from camera is sent to user interface and the microprocessor waits until the wireless interface sends a signal.

User Mode- The microprocessor sends data to the embedded system, takes images from the camera, and sends data to the wireless interface.

Sleep Mode- The microprocessor is powered down and waits until the wireless interface sends a signal to the microprocessor to perform an action.

Needs Charging- The microprocessor is shut down to conserve the batteries until charging after the last camera image was sent to the wireless interface.

Charging- The microprocessor is shut down until the batteries are fully charged. Once the batteries are fully charged the microprocessor waits until a signal is sent from the wireless interface.

Embedded System

Inputs	Outputs
Commands	Status
Battery Voltage	Acoustic Sensor TTL Signal
Acoustic Sensor TTL Signal	Motor Control PWM Signal
Wheel Sensor Bit Stream	

Embedded System-

Inputs/ Outputs

Acoustic Sensors – The embedded system sends a TTL signal to the acoustic sensors which produce a transmit pulse. The transmit pulse bounces off of an object and returns to the acoustic sensors as an echo pulse. The sensor sends a TTL signal back to the embedded system.

Inputs	Outputs
TTL trigger	TTL signal delayed indicating distance to detected obstacle detected

Inputs

Wheel Sensors – The wheel sensors send a bit stream that the embedded system translates into distance traveled.

Inputs	Outputs
Wheel motion	Bit Stream

Battery System– The voltage across the battery terminals is measured by the embedded system and the charge of the battery is determined.

Outputs

Power from battery delivered to Rover Subsystems

Monitor of terminal voltage

Outputs

Motor Control – The embedded system sends a PWM signal to control the speed of the motors.

Inputs	Outputs
PWM signal from EMAC	Motor rpm

Function

Wait Mode- The Embedded system waits until a signal is sent from upper level software and all subsystems are powered up.

User Mode- The Embedded system receives inputs from the battery, wheel sensors, and acoustic sensors. The data is sent to the upper level software.

Sleep Mode- The Embedded system is powered down and no actions by the user are allowed to occur.

Needs Charging- The Embedded system is powered down. No signals are sent or received to conserve the batteries until they are recharged.

Charging- The Embedded system waits until the batteries are fully charged. Once the batteries are fully charged, the system waits until the user accesses the system.

Software Flowcharts

High Level Software

Control:

All upper level software is based on a Linux operating system, Red Hat 7.1 or above. This handles the requests for communication through the wireless network card, and also runs apache web-server for the requested webpage. The apache web-server software, receives the request by the user for the webpage, through the wireless network card. Then embedded in the webpage, are both the control java applet and the imaging java applet. The applets run on the rover, the communication between the user and the rover is done through HTML protocol.

The rover and video feed are controlled using high level software. Both are embedded into an html page as separate java applets. Control is based on a forward/backward motion and a rotation. Control for the motion forward and backward is based on either the user pressing and releasing the up or down arrow on the keypad, moving the rover forward by a predetermined distance, or by entering the distance in centimeters in a text field. Control for the rotation is done with a visual representation of the rover on a 0° to 360° axis. The user then inputs into a text field the rotation. Once the robot has completed the move, the software loops to the beginning again and waits for the next control event.

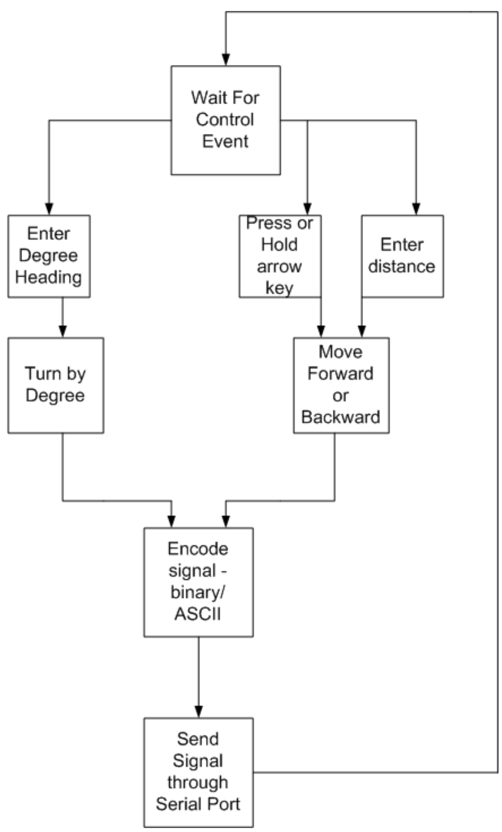

Figure 4
Control Software for the Mars Rover.

Image:

The camera image is compressed into a 320x280 16 bit color jpeg still frame by the camera's software and updated to the user through a java applet. In-between the image is stored to a predefined memory location, after a two second delay occurs; the image is updated once again. This process repeats itself until the user logs off.

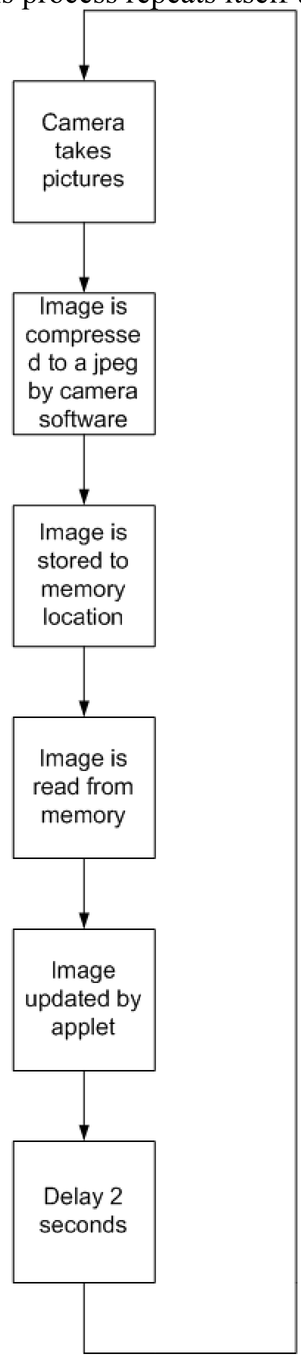

Figure 5
Imaging applet software for the Mars Rover

Low Level Software

Motor Control:

The Micropac 535 will control each of the motors individually with a PWM signal created with a timer interrupt routine. Each timer routine will have two registers, one for high time and one for low time, these registers will control the PWM duty cycle. A desired distance, entered by the user, is stored in a register and compared to the actual distance measured by the wheel sensors. The wheel sensors produce a set number of pulses per revolution of the shaft; this number is counted using a counter. The value stored in the counter is retrieved every .1 seconds. This value is subtracted from the total distance the rover should travel, and the counter is cleared once the value has been retrieved. Comparing these values will allow the Micropac to detect if one wheel has traveled a longer distance than the other, which implies that the rover is veering slightly to one direction; the Micropac can then straighten the rover by speeding up the lagging wheel, or by increasing the high time register and decreasing the low time register for the lagging wheel's timer routine. The motors are only powered in user mode.

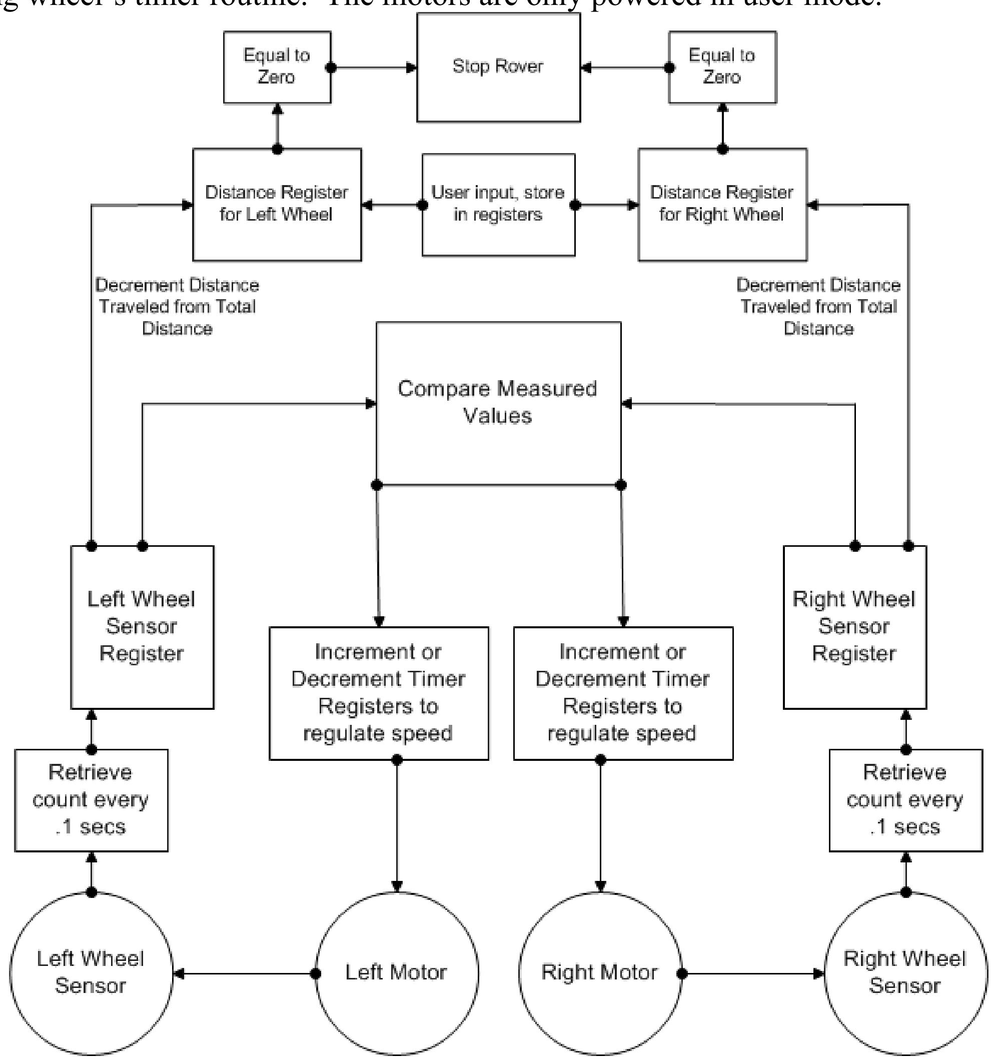

Figure 6
Motor control and sensor feedback

Object Detection:

With the use of acoustic sensors, the rover can detect objects directly in front and in back of it. The transmit signal is sent to the sensor from the Micropac; the sensor transmits a sound wave and detects an echo if an object is nearby. When an echo is detected, the acoustic sensor sends a signal back to the Micropac. The time between the transmit and receive pulses is directly proportional to the distance from an object. If the rover is determined to be too close to an object, the rover is stopped. The acoustic sensors are only powered in user mode.

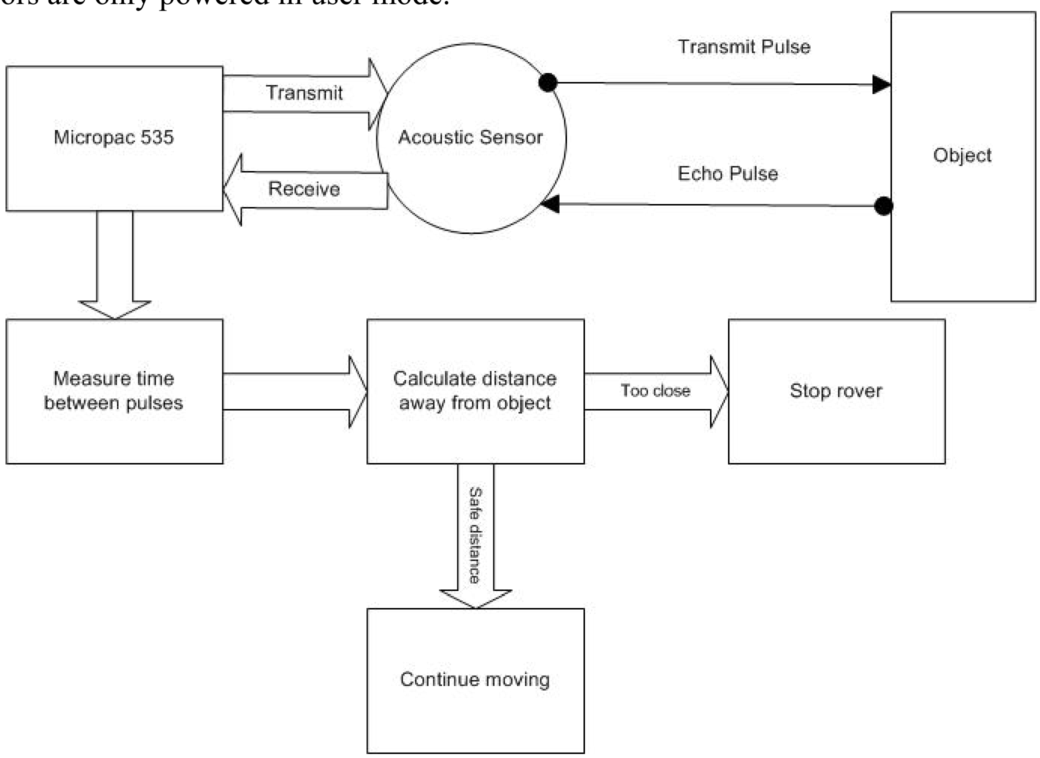

Figure 7
Acoustic sensor control

Battery Voltage Level:

The battery voltage is read in from the A/D converter and is compared to a value in the data table. If a user is present, an approximation of the charge level is displayed on their monitor. If the charge is sufficient for operation, the rover keeps running. Otherwise it switches to low charge mode. The battery voltage is monitored in every mode, but is not measured constantly; it is only measured every ten minutes in order to conserver battery life.

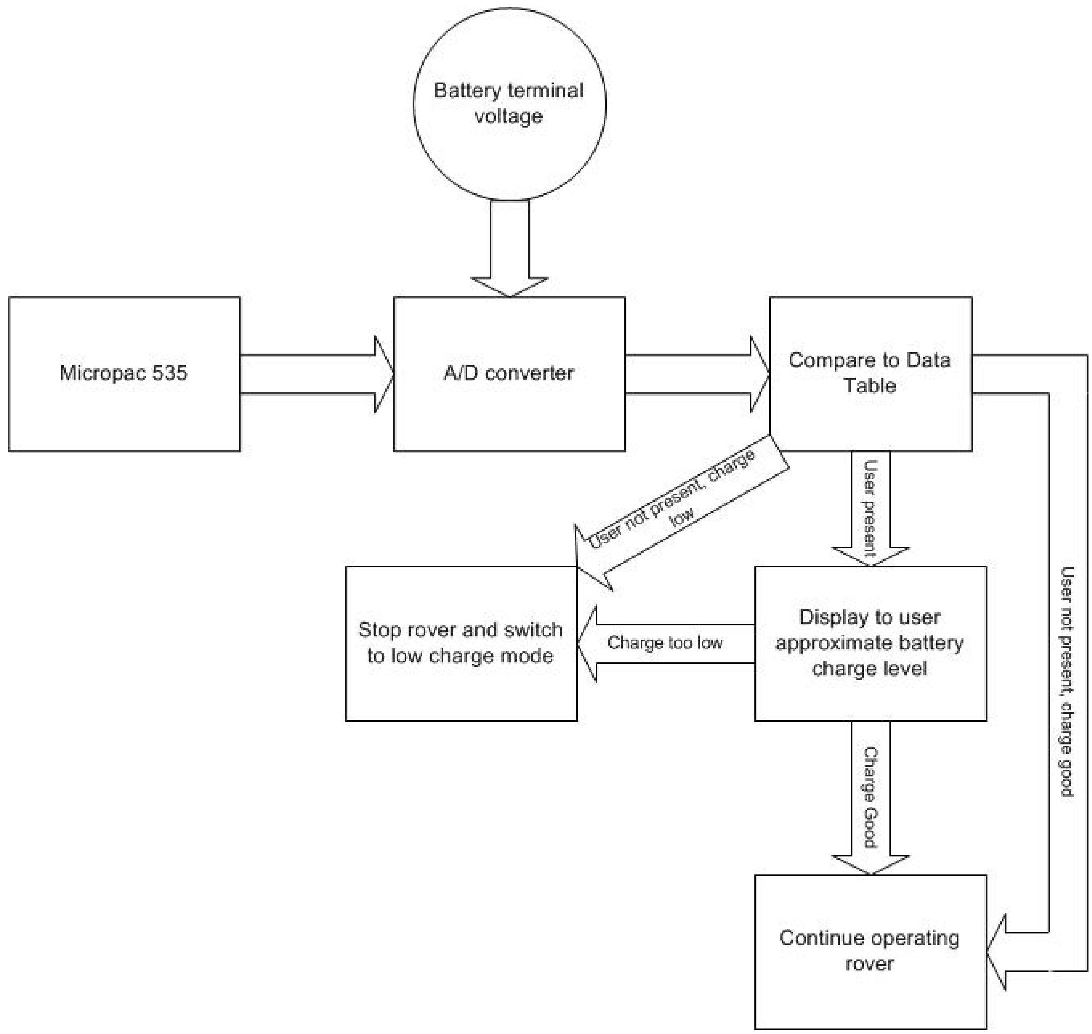

Figure 8
Battery charge level measurement

Preliminary Research

System Software:

The most critical part of the high level computing had been the decision to choose the Linux kernel for this application. This decision was based on the operating system's minimum requirement.

Linux Red Hat 8.0
X86 processor
16MB RAM
500MB hard disk.
Cost $0.00

Windows 98,
X86 processor
64MB RAM
1.0 Gig hard disk
Cost $90.00

Real time operating systems had been considered too, however, due to their complexity, cost, and time to learn, this project decided to use a dual processor system. With the lower hardware requirements of Linux, the project will be able to make use of a lower powered processor. Linux's native support of java, standard wireless communication cards, and apache server included, enhanced Linux's appeal.

Hardware – Computer Platforms:

In the upper level computing platform three different avenues were explored, PC/104, PC/104+, and SBC. Power consumptions of the three were very close. The PC/104 platform allowed for the best options in expansion. The SBC, single board computers, were in predetermined configurations and thus not customizable. The PC/104+ offers the best performance, with 32bit expansion bus, yet, finding expansion modules had been very hard and largely expensive. Thus two different parts lists were devised and included, one with a PC/104 and one with a PC/104+.

Hardware – Computer Processors:

All three types of platform considered incorporated an Intel processor, a National Semiconductor processor, or a Transmeta Crusoe processor. All were x86 architecture based. The only concerning characteristics thus were power.

All powers in sleep mode, 90% idle

Intel PIII ULV	.85 Watts
Nation Semiconductor Geode	.50 Watts
Transmeta Crusoe	.1 Watts

The power characteristics were taken from data sheets on each of the processor, and described a state in which the processor was the only device running on the board. The processor decided on was the National Semiconductor Geode GX1 based on price and power consumption.

Hardware – Wireless cards:

The main concern with wireless card research was not only the power they consumed during operation, but also Linux compatibility. The Dell Truemobile wireless card was the best in power consumption, but the Truemobile lacked compatibility with Linux. Cisco made a very good wireless card with Linux compatibility, but the power consumption was mediocre. The Linksys wireless card had lower power consumption then the Cisco card and was also compatible with Linux. The Linksys card was chosen due to its device drivers support for Linux from the manufacturer and its low power requirements.

Hardware – Hard drives:

A standard IDE hard drive drew too much power for this project, thus other storage methods were researched. There included flash cards, 2.5 inch IDE, and micro-drives. A flash card had excellent power consumption, but unfortunately offered too little storage. 2.5 inch laptop hard drives pulled less current then 3.5 inch IDE hard drive, yet still had high power consumption. A PCMCIA based micro-drive had the best power consumption, but the PCMCIA drive could not be booted from. The project will then use a PCMCIA hard drive with a boot device, such as a flash card through USB, disk-on-chip, or IDE to flash card adapter. With the PC/104+ a disk-on-chip option was chosen, while the PC/104 board will use an IDE to flash card adapter.

Hardware – Batteries:

At first, 6V batteries were researched since a 5V supply was needed for the PC104 computer. Enough current was needed to allow the rover to run for 7 days without a recharge, but as the current capacity of the batteries went up, so did the weight. Unfortunately, 5V motors were not strong enough to drive the rover, so 12V batteries, along with 12V motors, were needed. An idea of using 2 batteries arose, one 6V for the computer and one 12V for the motors, but the rover would need two different chargers, so the idea was quickly ruled out.

The advisers suggested using the 12V batteries already installed on the Shockency and Satterwaithe project, along with a DC to DC converter to get 5V. The batteries were 2 - 12V @ 7.2Ah, which would provide for a 7 day recharge period (see power calculations). These 2 – 12V batteries will be the power for the Mars Rover.

Hardware – Motors

Research was started on 5V motors, but was quickly declared useless since the rover probably wouldn't be able to move with such small motors. Maxon motors were considered for their low power consumption. Hansen motors were investigated also. The best motor seemed to be the Pittman motors because of previous work with the Pittman motors. Also, the advisers suggested using the Pittman motors from Shockency and Satterwaithe's project, which was agreed on.

Power Calculations:

Power Consumption for Sleep Mode:

PC104 computer	.026A
PC104 PCMCIA module	.07A
PCMCIA Hard drive	.015A
PCMCIA Wireless Card	.009A
EMAC	.045A
	+
Total	.165A

24hrs * 7days = 168hrs
168hrs * .165A = 27.72 Ah @ 5V
27.72Ah * 5V = 138.6Wh
Using 2 - 12 Volts, 7.2Ah batteries:
12V * 7.2Ah * 2 = 172.8Wh available

Power Consumption for User Mode:

PC104 computer	.8A
PC104 PCMCIA module	.07A
PCMCIA Hard drive	.4A
PCMCIA Wireless Card	.285A
EMAC	.045A
Camera	.1A
2 Polaroid Ultrasonic 6500	.2132A for a transmit pulse at 10 Hz *below*
	+
Total	1.9132A

The motors chosen by the Robotic Platform Design project were Pittman GM9236, which pull 16.9A max.
Total with motors 1.9132A + 16.9A * 2 = <u>35.7A</u>

If we assume that user is connected 3% (or 5 hrs out of a week) of the time, then power consumption is as follows:

(1.9132A * 3% + .165A * 97%) * 168 = 36.5Ah without motors
(35.7A * 3% + .165A * 97%) * 168 = 206.8Ah with motors @ maximum current

36.5Ah * 5V = 182.48Wh required if motor current is negligible

((33.8A * 12V + 1.9066A * 5V) * 3% + .165A * 97% * 5V)*168hrs=2226.7Wh required

New motors we want to use are the Pittman GM9X12 motors which pull 4.56A @ 12V, calculations are below:
((9.12A * 12V + 1.9066A * 5V) * 3% + .165A * 97% * 5V)*168hrs =726.8Wh required
To meet the requirements to run for 7 days without a recharge using 2 – 12V @ 7.2Ah batteries, the user would only be able to run the rover for 1.5hrs a week
((9.12A * 12V + 1.9066A * 5V) * .9%+.165A*99.1% * 5V)*168hrs =179.1Wh required
The motors are specified only at maximum current, if it is assumed that they will pull less then .5A at normal operation, then the power consumption will be almost negligible, which means the rover will possibly last for 7 days if the user connects <= 3% of the time. Since time has not been taken to measure motor current during normal operation, a specific value cannot be stated.

below
Polaroid Ultrasonic 6500
2A for 330us (transmit period)
.1A otherwise
10 second pulse (.1 Hz)

$$2A * \frac{.0033}{10} + .1A * (\frac{10 - .0033}{10} \sim 1) = .1066A$$
$$2A * .1066 = .2132A$$

Data Sheet Specifications

Turning accuracy - ± 5° for an individual turn command
Driving accuracy - ± 5cm and ± 2° for a 100cm command
Camera capture speed – 5 frames/sec @ 324x288 resolutions for a 10BaseT connection
Weight – ~28lbs
Battery life – 7 days without a recharge if user connects <= 5 hours a week
Top speed – 10cm/s
Speed range – 1 cm/s to 10 cm/s
Embedded System
    8051 MicroPac – 5 Volts at .045Amps
Server
        PC/104 or PC/104+ - 5Volts at .8Amps(max)
        (80% load)
Hard Drive
    Toshiba PCMCIA Hard disk - 3.3Volts at .4Amps
PCMCIA Adapter – 5Volts at .07Amps
Wireless Ethernet Card – 3.3Volt at .285Amps
Web Camera – 5Volt at .1Amps
    326x288 dpi
    16bit color
    Jpeg images
Acoustic sensors –
    Transmit frequency – 10 hertz
    Farthest object detection – 400cm
    Closest object detection – 50cm
    TTL 5Volts at .2132A for a transmit pulse at 10 Hz
Motors –
    Model number – GM9236
    Gearing – 1:65.5
    Max current – 16.9A
    Voltage – 12V
Wheel Sensors –
    Output – TTL
    Pulses per revolution of shaft – 512
    Voltage required – 5V
Battery charge level accuracy - ± 5%
Wireless protocol – 802.11b
Dimensions – 31.4cm x 46.4cm x 21cm (L x W x H)
Battery – 2 - 12V @ 7.2Ah
Wheels – 5cm x 16cm (Width x Diameter)

Patents

The following patents pertaining to our project were collected from the IBM Patent Search searchable on-line database using the keyword *Telerobotics*.

5,231,693 Telerobotics.

Abstract:

A telerobotic system including an operator interface for developing and controlling complex tasks for execution on a remote robot manipulator by developing sequences of parameterized tasks for execution and testing on a local simulator. The task sequences, and their parameterization, may be interactively modified by teleoperation and/or editing during sequence development and task execution. Adjustable time delays are provided for execution simulation including delays expected to occur between task command and execution at the remote robot manipulator.

The following patents are relevant to our project were collected from the United States Patent and Trademark Office searchable on-line database using the keywords *robotic, platform, mobile, and telerobtics*

.

Keyword: Telerobotics and Mobile Platform

5,737,500 Mobile dexterous seven degree of freedom robot arm with real-time control system

Abstract:

The present invention is a mobile redundant dexterous manipulator with a seven-degree-of-freedom robot arm mounted on a 1 degree-of-freedom mobile platform with a six-degree-of freedom end effector including a real-time control system with multiple modes of operation. The manipulator-plus-platform system has two degrees-of-redundancy for the task of hand placement and orientation. The redundancy resolution is achieved by accomplishing two additional tasks using a configuration control technique. This mobile manipulator with control system allows a choice of arm angle control or collision avoidance for the seventh task, and platform placement or elbow angle control for the eighth task. In addition, joint limit avoidance task is automatically invoked when any of the joints approach their limits. The robot is controlled by a processor employing a 6-by-7 Jacobian matrix for defining location and orientation of the end effector

Keyword: Telerobotics

5,581,666 Modular architecture for robotics and teleoperation

*Abstract***:**

Systems and methods for modularization and discretization of real-time robot, telerobot and teleoperation systems using passive, and network based control laws. Modules consist of network one-ports and two-ports. Wave variables and position information are passed between modules. The behavior of each module is decomposed into uncoupled linear-time-invariant, and coupled, nonlinear memory less elements and then are separately discretized.

Keyword: Mobile platform

6,122,572 Autonomous command and control unit for ***mobile platform***

Abstract:

In a vehicle designed for the execution of a mission, a programmable decision unit capable of managing and controlling the execution of the mission by utilizing a plurality of subsystems and database capable of holding and managing data including pre-stored data and data acquired by and received from the plurality of subsystems. The programmable decision unit includes a mission plan (MP) for accomplishing the execution of the mission includes a succession of iterations that include each assignment of a mission segment associated with a current mission stated to one or more of the subsystems. Each iteration further includes receipt from the subsystems report data which include data indicative of the execution status of the mission segment by the corresponding subsystem; and evaluation of the report data for determining either normal behavior or an exceptional event. The programmable decision unit is capable of managing and controlling the execution of the mission in an autonomous fashion whereby the vehicle becomes an autonomous vehicle.

Keyword: Mobile Robotic

5,936,240 Mobile autonomous robotic apparatus for radiological characterization

Abstract:

A mobile robotic system that conducts radiological surveys to map alpha, beta, and gamma radiation on surfaces in relatively level open areas or areas containing obstacles such as stored containers or hallways, equipment, walls and support columns. The invention incorporates improved radiation monitoring methods using multiple scintillation detectors, the use of laser scanners for maneuvering in open areas, ultrasound pulse generators and receptors for collision avoidance in limited space areas or hallways, methods to trigger visible alarms when radiation is detected, and methods to transmit location data for real-time reporting and mapping of radiation locations on computer monitors at a host station. A multitude of high performance scintillation detectors detect radiation while the on-board system controls the direction and speed of the robot due to pre-programmed paths. The operators may revise the preselected movements of the robotic system by Ethernet communications to monitor areas of radiation or to avoid walls, columns, equipment, or containers. The robotic system is capable of floor survey speeds of from 1/2-inch per second up to about 30 inches per second, while the on-board processor collects, stores, and transmits information for real-time mapping of radiation intensity and the locations of the radiation for real-time display on computer monitors at a central command console.

Standards

Major of the internet standards are documented in Internet Request For Comments, which are indexed at Ohio State University. Java is trademarked by Sun Computer Systems.

RFC 791 - Internet Protocol (IP)
RFC 793 - Transmission Control Protocol (TCP)
RFC 826 - An Ethernet Address Resolution Protocol (ARP)
RFC 893 - Internet Protocol on Ethernet Networks
RFC 1866 - Hypertext Markup Language (HTML/2.0)
RFC 1945 - Hypertext Transfer Protocol (HTTP/1.0)

USB Standard was found from USB.org, the document is part of a zip file that also includes information on the newest standard USB 2.0.

802.11b standard was found at IEEE.org.

Schedule

Laboratory Week	Project Milestones
January 19, 2003	Assemble PC104 and interface with previous Robotic Platform Design project.
January 26, 2003	Create boot software for Linux.
	Install drivers for all components in Linux.
February 2, 2003	Develop and test motor control software on Micro Pac 535.
	Develop software to interpret wheel sensor bit streams.
February 9, 2003	Continue working on software development for motor control and feedback loop.
February 16, 2003	Develop software to capture image from camera and send to user.
	Continue working on software development for motor control and feedback loop.
	Work on web server development.
February 23, 2003	Create Java applet for user interface.
March 2, 2003	Continue with Java applet
	Work on software to estimate battery charge level.
March 9, 2003	Finish working on software to estimate battery charge level
March 16, 2003	Spring Break
March 23, 2003	Develop software to operate acoustic sensors
March 30, 2003	Finish Java applet.
April 6, 2003	Testing of individual components and overall system.
April 13, 2003	Testing of individual components and overall system.
April 20, 2003	Preparation for presentation and final report
April 27, 2003	Presentation

Laboratory Week	Dan Dunn	Colin Shea	Eric Spiller
January 19, 2003	Assemble Mars Rover	Assemble Mars Rover	Assemble Mars Rover
January 26, 2003	Install drivers for Linux	Boot software	Motor H/W
February 2, 2003	Motor control S/W	Java Applet	Motor H/W
February 9, 2003	Motor control S/W	Java Applet	Motor H/W
February 16, 2003	Wheel Sensor S/W	Server side applets	Wheel Sensor H/W
February 23, 2003	Battery Charge S/W	Web Interface	Motor feedback control
March 2, 2003	Acoustic Sensor S/W	Camera interface	Camera interface
March 9, 2003	Serial Comm	Serial Comm	Serial Comm
March 16, 2003	Spring Break	Spring Break	Spring Break
March 23, 2003	Setbacks	Setbacks	Setbacks
March 30, 2003	Testing Complete System	Testing Complete System	Testing Complete System
April 6, 2003	Testing Complete System	Testing Complete System	Testing Complete System
April 13, 2003	Expo	Expo	Expo
April 20, 2003	Presentation prep	Presentation prep	Presentation prep
April 27, 2003	Presentation	Presentation	Presentation

Bibliography

Dell, **Dell Corporation**, Nov 20th 2002
<http://www.dell.com>

IBM Company. **IBM Company**, Dec 02nd 2002
<http://www.ibm.com>

IEEE, **IEEE Organization,** Dec 02nd 2002
<http://www.ieee.org>

Intel, **Intel Corporation,** Nov 15th 2002
<http://www.intel.com>

Linksys, **Linksys,** Nov 15th 2002
<http://www.linksys.com>

Microsoft, **Microsoft Windows,** Nov 15th 2002
<http://www.microsoft.com>

National Semiconductor, **National Semiconductor Corp,** Nov 15th 2002
<http://www.national.com>

Red Hat, **Linux Red Hat,** Nov 15th 2002
<http://www.redhat.com>

Rob Harding, Greg Harmon, John Kioldasa, Jon Kujanski **Telerobotics Senior Project Homepage,** Nov 28th 2002. <http://cegt201.bradley.edu/projects/proj2000/prjtelrb/>

Shockency, Robert and Randall Satterthwaite. Robotic Platform Design. Bradley University. Dec 3rd 2002 < http://cegt201.bradley.edu/projects/proj2002/traxx/>

Sean Grady, Tim Booth, Geoff Mikosz **Telerobotics 2001,** Nov 28th 2002.
<http://cegt201.bradley.edu/projects/proj2002/telerobo/>

SuSE, **SuSE the Linux Experts,** Nov 15th 2002
<http://www.suse.com>

Ted Konetski and Brain Davis. **Web Controlled Robot**. Dec 02nd 2002.
<http://cegt201.bradley.edu/projects/proj1999/web/webbot/WebRBot/WebRBot.htm>

Toshiba, **Toshiba Corporation,** Nov 15th 2002
<http://www.toshiba.com>

Transmeta, Transmeta Corporation Crusoe, Nov 15th 2002
<http://www.transmeta.com/>

United States Patent and Trademark Office Home Page. **United States Patent and Trademark**, Dec 01st 2002 <http://www.uspto.gov/patft/index.html/>

USB INFO. **USB Implementers Forum**, Dec 01st 2002 <www.usb.org>

Wireless Fidelity, **WiFi Alliance**. Dec 06th 2002 < http://www.weca.net/OpenSection/index.asp?TID=1>

Logitech, **Logitech Website Homepage.** Nov 7th, 2002 < http://www.logitech.com/lang/pdf/qc_express.pdf>

Ezonics, **Ezonics We make high tech.** Nov 7th, 2002 < http://ezonics.com >

Lexar, **Lexar Media.** Nov 7th, 2002 < http://www.digitalfilm.com/digfilm/index_cf.html >

Toshiba, **Toshiba America.** Nov 7th, 2002 < http://www.toshiba.com/ >

Batteries America, **Batteries America.** Nov 7th, 2002 < http://www.batteriesamerica.com/ >

Battery Mart, **Batteries of Battery Mart.** Nov 7th, 2002 < http://www.batterymart.com >

Maxon Motors, **Products & Specs.** Nov 7th, 2002 < http://www.maxonmotorusa.com/products/motors.cfm?prod=a >

Hansen Motors, **Hansen Corp DC Motors.** Nov 7th, 2002 < http://www.hansen-motor.com/dcindex.html >

Pittman, **Pittman – Electric Motors.** Nov 7th, 2002 < http://www.pittmannet.com/ >

Equipment List for Mars Rover

Part	Qty.	Website	Manufacturer	Location of Vendor	Part #	Price
5 Gb PC/MCIA Harddrive	1	www.pricewatch.com	Toshiba	www.legendmicro.com	HDD1232CZP41002	$191.00
128 Mb RAM	1	www.pricewatch.com	Infineon	www.18004memory.com	LG1064U/064/G3VAC	$14.20
PC/MCIA Wireless Card	1	www.pricewatch.com	Linksys	www.newegg.com	wpc11	$65.00
USB Webcam	1	www.pricewatch.com	Logictech	www.enpc.com	961137-0403	$16.00
PC104+ 200MHz w/ USB	1	www.square1industries.com	National Semi	www.square1industries.com	CM-589	$399.00
Dual PC/MCIA Adaptor PC/104+	1	http://www.prestico.com/	PAC	Lane Rowland HTS. CA	PCM-225/2	
						$685.20

The Energy Conservation Laboratory and Laboratory Room ___ will be used to construct the project.

All return policies are at 15 days 15% restock if non-defective.

Part	Qty.	Website	Manufacturer	Location of Vendor	Part #	Price
5 Gb PC/MCIA Harddrive	1	www.pricewatch.com	Toshiba	www.legendmicro.com	HDD1232CZP41002	$191.00
128 Mb RAM	1	www.pricewatch.com	Infineon	www.18004memory.com	LG1064U/064/G3VAC	$14.20
PC/MCIA Wireless Card	1	www.pricewatch.com	Linksys	www.newegg.com	wpc11	$65.00
USB Webcam	1	www.pricewatch.com	Logictech	www.enpc.com	961137-0403	$16.00
PC/104 300Mhz w/USB	1	www.square1industries.com	National Semi	www.square1industries.com	CM-588	$398.00
Dual PC/MCIA Adaptor PC/104	1	www.square1industries.com	square1	www.square1industries.com	NC-893	$85.00
IDE to Compact Flash	1	www.acscontrol.com	ACS	www.acscontrol.com	ACS-CF-IDEToCFA	$25.00
						$794.20

There also is a 2.5IDE adapter version, ACS-CF-IDESFFToCFA that could work and does not require the 5volts that the other one does. The other item required is a compact flash card, whose size is still unknown, yet can be picked up at any electronics store locally.