

Mars Rover:
System Block Diagram

November 19, 2002

By:

Dan Dunn
Colin Shea
Eric Spiller

Advisors:

Dr. Huggins
Dr. Malinowski
Mr. Gutschlag

System Block Diagram

An overall system block diagram, shown in Figure 2-1, 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.

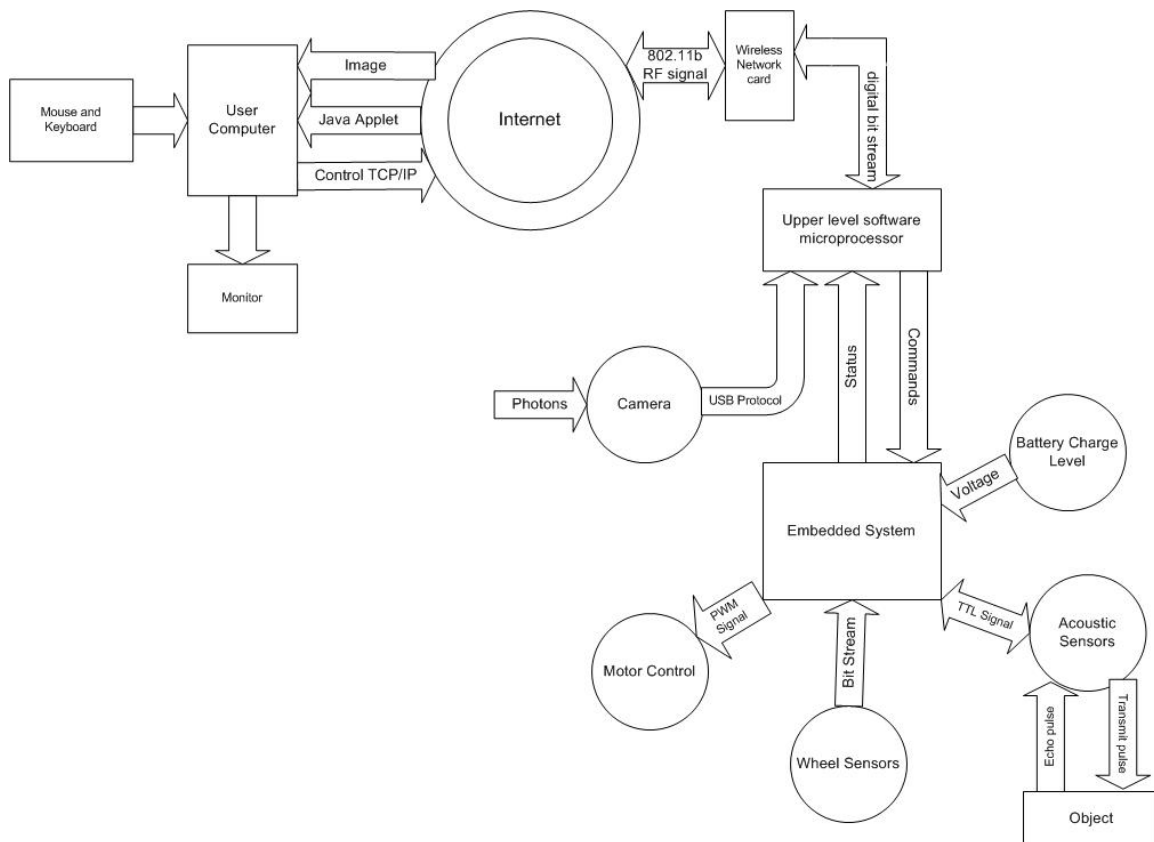


Figure 2-1
Overall Block Diagram for Mars Rover

User Computer

Inputs

Mouse & Keyboard
Image from Camera
Java Applet

Outputs

Monitor
Control TCP/IP

Inputs

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

802.11b/ RF Signal

Digital Bit Stream

Outputs

802.11b/ RF Signal

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

Digital Bit Stream

USB Protocol

Status

Outputs

Digital Bit Stream

Commands

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

Commands
Battery Voltage
Acoustic Sensor TTL Signal
Wheel Sensor Bit Stream

Outputs

Status
Acoustic Sensor TTL Signal
Motor Control PWM Signal

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

TTL trigger

Outputs

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

Wheel motion

Outputs

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

PWM signal from EMAC

Outputs

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.

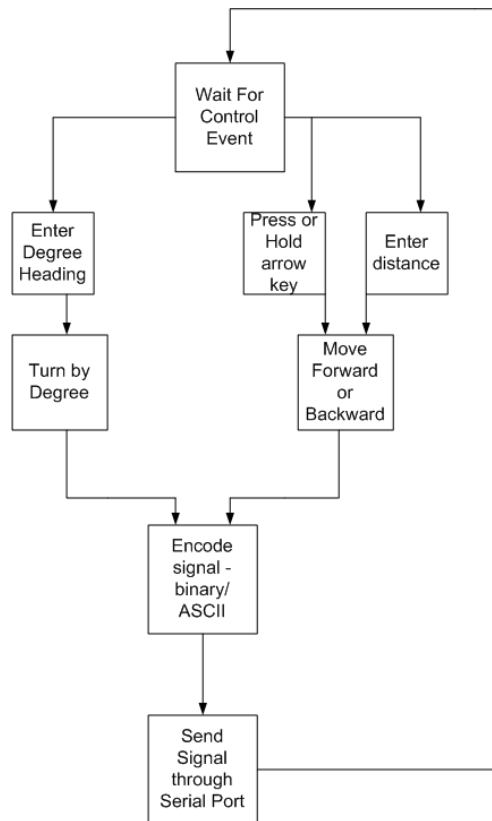


Figure 7-1
Control Software for the Mars Rover.

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.

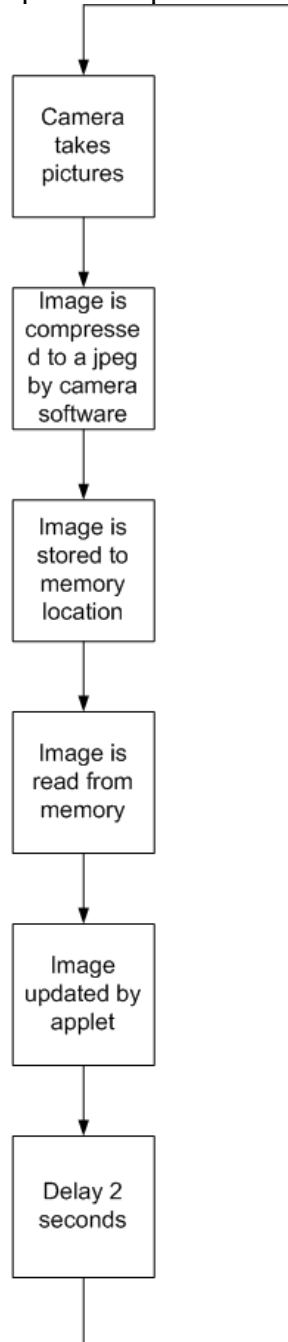


Figure 8-1
Imaging applet software for the Mars Rover

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.

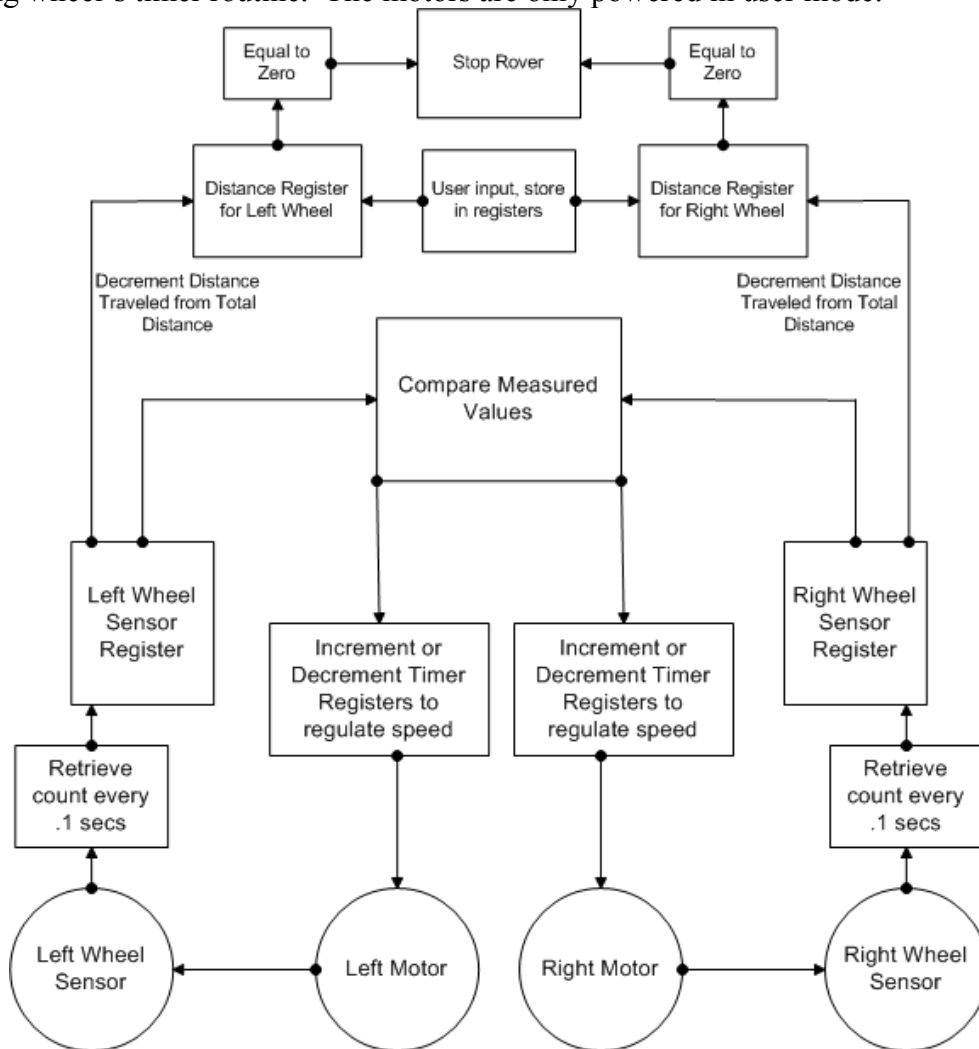


Figure 9-1
Motor control and sensor feedback

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.

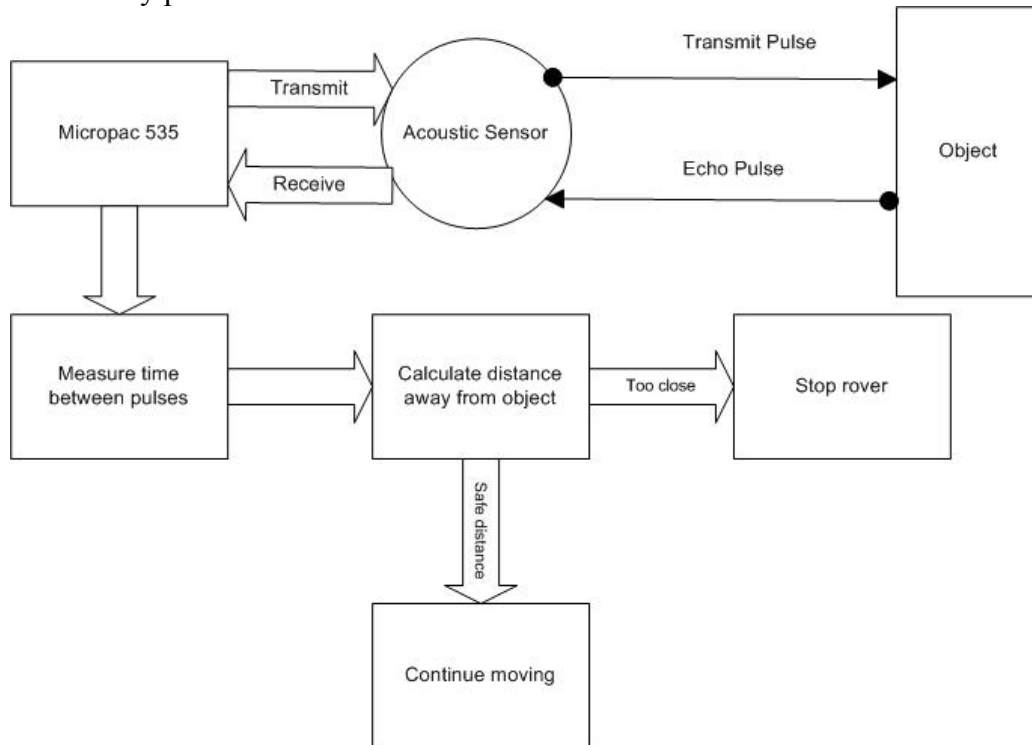


Figure 10-1
Acoustic sensor control

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 conserve battery life.

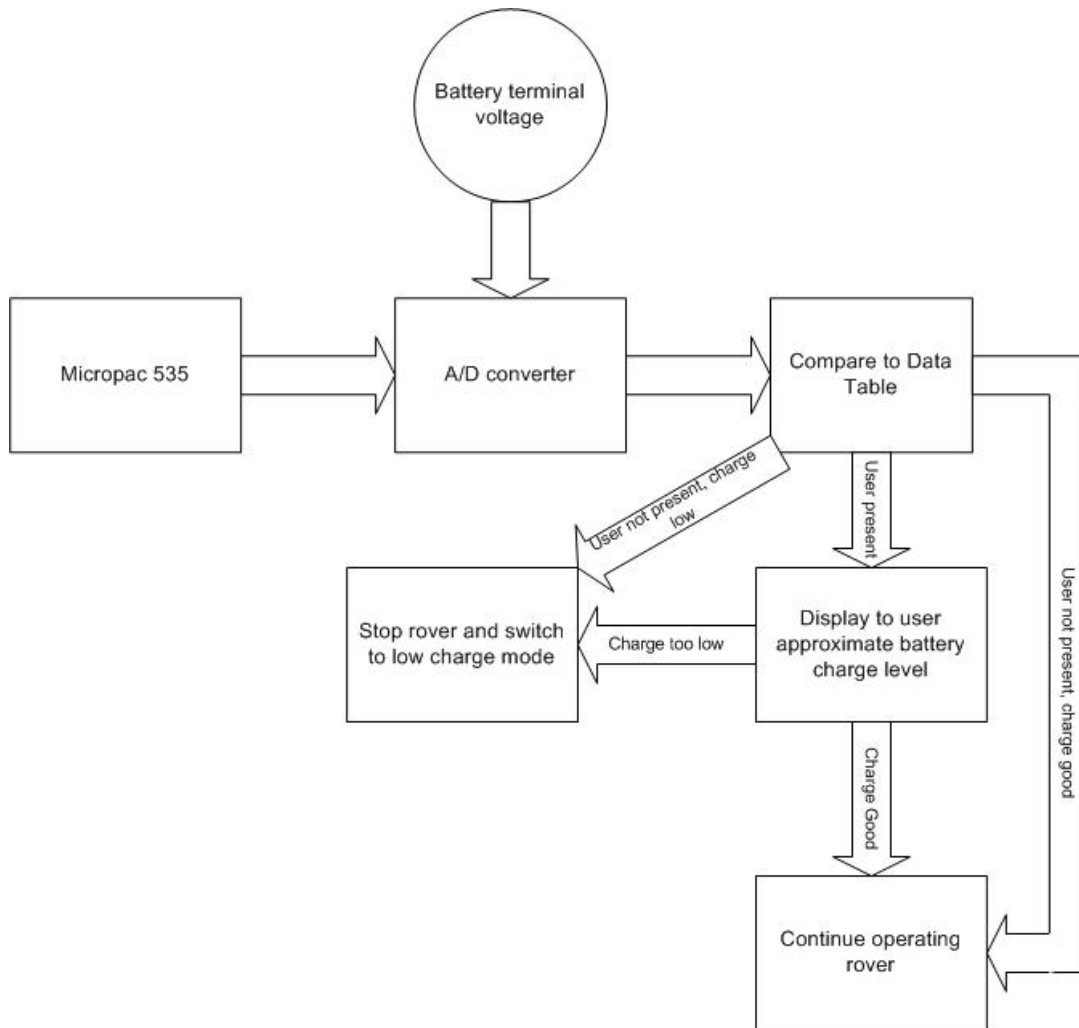


Figure 11-1
Battery charge level measurement