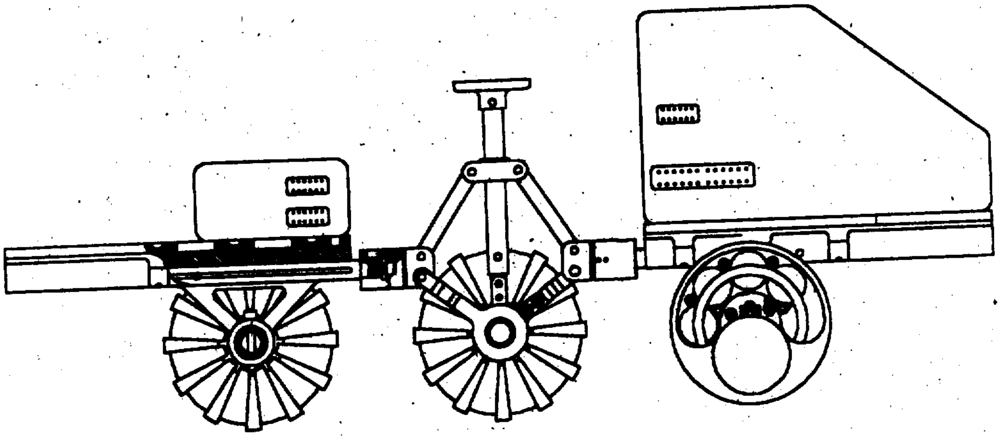
NASA-CR-195520

IN-37-CR
204277
103P

DESIGN OF A WHEELED ARTICULATING LAND ROVER

NASW-4435



UNIVERSITY OF IDAHO

SPONSORED BY USRA/NASA

(NASA-CR-195520) DESIGN OF A WHEELED ARTICULATING LAND ROVER (Idaho Univ.) 103 p

N94-24806

Unclas

G3/37 0204277

DESIGN OF A WHEELED ARTICULATING LAND ROVER

UNIVERSITY OF IDAHO

MECHANICAL ENGINEERING DEPT.

PROFESSOR: LARRY STAUFFER

GRADUATE ASSISTANTS: MATHEW DILORENZO AND BARBARA YANDLE

STUDENTS: Eric Baicy, John Barinaga, Brian Block, Wayne Bunce, John Clausen, Jay Constable, William Cresse, Travis Fulton, John Gadbois, Darryn Lunders, Jonathon Martinez, Martin Maxwell, Ron Stanley, Dave Walker, Mike Webster, Richard Williams

ABSTRACT

The WALRUS is a wheeled articulating land rover that will provide Ames Research Center with a reliable, autonomous vehicle for demonstrating and evaluating advanced technologies. The vehicle is one component of the Ames Research Center's on-going Human Exploration Demonstration Project. Ames Research Center requested a system capable of traversing a broad spectrum of surface types and obstacles. In addition, this vehicle must have an autonomous navigation and control system on board and its own source of power.

The resulting design is a rover that articulates in two planes of motion to allow for increased mobility and stability. The rover is driven by six conical shaped aluminum wheels, each with an independent, internally coupled motor. Mounted on the rover are two housings and a removable remote control system. In the housings, the motor controller board, tilt sensor, navigation circuitry, and QED board are mounted. Finally, the rover's motors and electronics are powered by thirty C-cell rechargeable batteries, which are located in the rover wheels and recharged by a specially designed battery charger.

ACKNOWLEDGMENTS

We would all like to thank Dr. Vicki Johnson of the USRA; Terry Fong, Butler Hine, Jay Steele and Diane Cook of Ames Research Center; Mike Drews of the University of California; Dr. Donald Elger, Dr. Dean Edwards, Dr. Donald Blackketter, Dr. Blaine Tew, Professor Jasper Avery, Dr. Richard Gill, Darrel Brown, Terry Long, Fred Lock, Cal Finn, Dr. David Atkinson, Dr. Jeff Young, Dr. George Hespelt, Dr. Mike Anderson and Jim Hill of the University of Idaho for their input and help on this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In the fall semester of 1992, the University of Idaho began a project for NASA's Human Exploration Demonstration Project (HEDP). The University of Idaho's goal was to provide Ames Research Center in California with a autonomous test platform for advanced technologies, such as virtual reality. Several groups of computer, electrical and mechanical engineering seniors set out to complete this vehicle by the end of the summer of 1993. This project entailed the design, construction, and testing of a land rover, nicknamed "WALRUS," which stands for Wheeled Articulating Land Rover *by Unemployed Seniors.*

2. OBJECTIVE

The objective of this project was to design, build and test a semi-autonomous land rover capable of traversing a broad spectrum of surface types and obstacles. Also, this rover must have on-board navigation and control systems which can interface with a virtual reality system. Finally, it must have a system to maintain power to the motors and electronics.

3. METHODOLOGY

This design project was divided into three sub-projects as follows: 1) The frame and wheel system. 2) The navigation and control system. 3) The power system. Quality Function Deployment (QFD), functional decomposition, decision matrices and Failure Mode and Effects Analysis (FMEA) were used to develop a rover that would best satisfy the objectives of the project. These design methods facilitated a systematic approach to the design of the rover.

4. FRAME & WHEEL DESIGN

4.1 Customer Requirements.

The following are customer requirements that the rover frame and wheel system must meet (see appendix A for a complete listing of requirements):

1. Traverse hard floor, sand and lunar like surface types.
2. Operate at a moderate speed, climb 30 degree hills and clear obstacles.
3. Provide a easily recharged power system to support all on-board systems.
4. Support all external sensors, camera equipment and other payloads.
5. Be of sufficient size to allow placement of internal sensors and control/navigation systems.
6. Be of sufficient size to allow placement of external sensors and a camera.
7. Easily assembled and maintained.
8. Use University of Idaho manufacturing facilities.
9. Cost less than $800.00.

4.2 Functional Decomposition.

With the customer requirements defined we were able to define the function of the rover. We used functional decomposition to focus our design on the individual functions (see appendix B for functional decomposition taxonomy).

4.3 Literature Research.

A literature search was conducted to find information on previous lunar rover vehicles and associated technologies. The designs we investigated included analysis of both wheeled and tracked systems. We were specifically interested in comparing the performance of each locomotion type with respect to our specific customer requirements.

The research showed that the use of wheels provides a simple and sufficient locomotion system for our particular application.

According to the research done by Eagle Engineering, Inc., for the Advanced Programs office at NASA's Johnson Space Center [1], tracks are used on earth when a large footprint is required to support a large load in soft soil. However, tracks are complicated mechanisms, that are susceptible to frequent breakdowns. Eagle Engineering concluded that wheels were superior to tracks because of the mechanical simplicity and can be designed into a lightweight system with matchless reliability. The drawback of wheeled systems is the smaller footprint they provide.

In another study conducted by the Old Dominion University of NASA/USRA [2] concluded that the contact surface area was of prime importance to the navigation of rough terrain. A wheeled vehicle integrated with a articulated system provided a larger contact area improving the rover's ability to clear obstacles and maintain stability. This idea was further supported by an article from Mechanics of Mobile Robotic Platforms. In it, the shortcomings of tracked systems maneuverability through rough terrain was illuminated. When a tracked vehicle climbs over an obstacle, the vehicle will balance on two points, significantly reducing ground contact area and compromising vehicle stability.

The results from the research were organized into a decision matrix. As shown in Table 1, the wheeled vehicle is superior to a track system in many different aspects. Based upon the customer's requirements we decided a wheeled vehicle combined with an articulating frame mechanism would provide the best possible rover design.

	Wheel	Track
Reliability	+	-
Efficiency	+	-
Simplicity	+	-
Weight	+	-
Footprint	-	+
Load Carrying	-	+
Speed	+	-

Table 1. Wheel Versus Track Decision Matrix

4.4 Concept Evaluation.

Through the use of research, patent searches, brain storming, and trade journals; we developed sixteen vehicle design concepts. These concepts were evaluated based on their ability to meet our customer requirements. Unfeasible concepts were eliminated using a GO-NO GO decision method, leaving eight feasible design alternatives. A decision matrix was used to evaluate the remaining concepts (see appendix C for the decision matrix) based upon our customer requirements.

Using these design techniques, we found that the concept that best fit our customer requirements was a six wheeled articulating vehicle, with conical shaped wheels.

4.5 Detailed Design.

We designed a six wheeled articulating demonstration vehicle, which we have named the WALRUS. The vehicles uses a modular design to facilitate expansion and interchangeability of on board systems. The frame is symmetrical about it's central axis, thereby minimizing the numbers of custom parts needed to be fabricated. Figure 1 shows the rover design as it existed on paper after the first sub-project was completed in December of 1992.

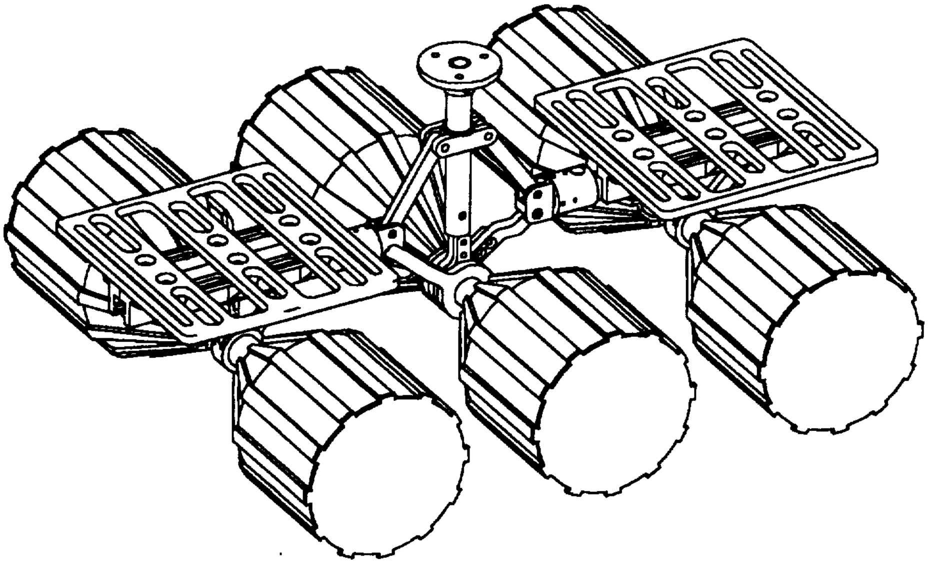


Figure 1. Schematic of Frame and Wheels.

With a full payload the total rover weight is equal to 45 pounds and the vehicle run time is approximately one hour. At this maximum weight, the rover is capable of climbing a 30 degree incline starting from zero velocity. The maximum RPM of the motor limits the speed of the rover to 1.5 feet per second.

4.6 Wheel Design.

The rover is equipped with 6 identical wheels, each containing a 35 oz-inch 12 volt DC motor, and five C-size nickel cadmium batteries. This group of five batteries is referred to as a carousel. The two center wheels also include an optical encoder to be used to monitor the speed of the vehicle.

The aluminum wheel housing is built in two parts; a treaded cone-cylinder section and an end cap. Traction on smooth surfaces is enhanced by installing 2 O-rings onto the machined grooves of the cylinder housing section. Access into the wheel is gained through the end cap by releasing three half turn fasteners. A section view of the wheel assembly is shown in Figure 2.

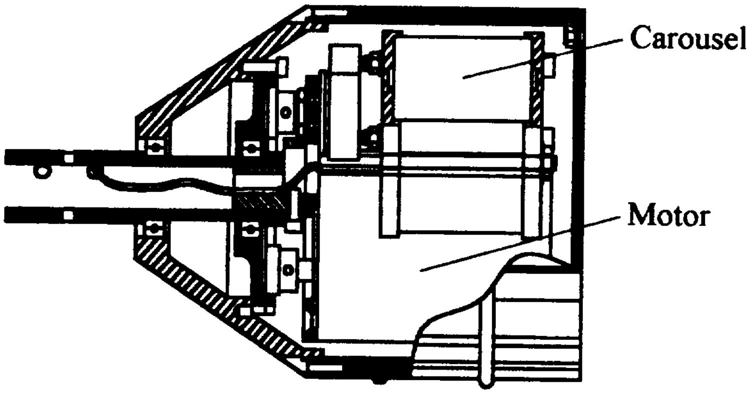


Figure 2. Section View of Wheel Assembly.

The wheel housing rotates about an aluminum shaft which is rigidly attached to the rover frame. The rotation occurs about two bearings; the first of which is connected between the shaft and the cone, the second which is attached from the shaft to a bearing support. The aluminum bearing support is rigidly secured to the cone housing. This bearing arrangement allowed us to terminate the shaft next to the bearing support, thus opening the entire wheel cavity for placement of the motor and battery carousel.

The battery carousel, optical encoder, motor are mounted to an aluminum support plate which is attached to the shaft. The interface between the plate and shaft is a hollow, stainless steel component through which the wires are routed out of the wheel and into the control system.

Each battery carousel contains five batteries clamped between two nylon support fixtures. This modular unit can be removed from the wheel though the end cap for easy inspection or maintenance. Also, through this access, the spur gear of the motor can be adjusted to properly mesh with the internal ring gear on the wheel so that gear backlash is minimized and maximum torque is transferred from the motor to the wheel.

4.7 Wheel Articulation System.

We found from our literature search that articulated vehicles exhibit superior mobility characteristics than vehicles of rigid body construction. The WALRUS is designed to pivot in both the longitudinal and lateral planes. This articulation mechanism allows all wheels to remain in contact with the ground while traveling across rough terrain and over obstacles.

Figure 3 shows a cross section of the rover exposing the articulation system. The mechanism is composed of three parts: two bearing assemblies and the center shaft assembly.

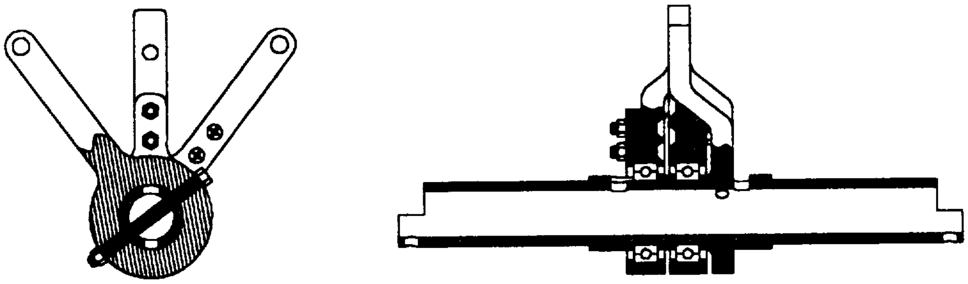


Figure 3. Center Shaft Assembly.

Longitudinal articulation occurs through the bearing assemblies. Two bearings are press fit onto a stainless steel hinge component. An external snap ring provides a redundant means of security to ensure that the bearings remain tight. A stainless steel housing is press fit over the bearings and redundantly secured by four set screws. The front frame is connected to the mid-frame through the bearing assembly by means of a threaded fastener.

Lateral articulation occurs through the center shaft assembly. With this mechanism, the forward and rear frame assemblies are capable of a 55 degree lift, and a 45 degree drop angle. This angular path is restricted by the bushing which slides along the camera shaft. The bushing is hinged to the lower frame by two mid frame struts. Maximum lift occurs when the bushing contacts the camera support plate. Maximum drop occurs when the bushing contacts the fastener connecting the camera shaft to the camera support arm of the center assembly.

In addition to increased vehicle mobility, this articulation system increases the forward looking capability of the camera mounted to the camera support plate. As the front frame assembly tilts up or down, the pitch angle of the camera is half that of the angular deflection of the front frame. For example, a 30 degree upward pitch of the front wheels will cause only a 15 degree lift of the camera field of vision. This effect minimizes the magnitude of the intervention of the pan and tilt mechanisms controlling the camera's position, increasing the response time of any necessary camera repositioning.

The center axle assembly consists of a pivoting arm and bearing cap, a camera support with bearing cap, a fixed arm and a center axle. The bearing caps are identical in form and function. Their purpose is to hold the bearings firmly to the pivoting arm and center shaft. The fixed arm of the axle assembly is bolted to the center shaft to provide a rigid support for mounting of the wheels. Axle shims are used to precisely locate the pivoting arm and the camera support. Additional shims are used to position the wheel on either side of the axle assembly.

4.8 Frame.

The frame is symmetric about the center camera shaft. This increases the manufactureability and maintainability of the rover. The use of symmetry also promotes interchangeability of platforms allowing the rover to be used in a myriad of demonstration applications with minimal setup requirements.

The modular frame assembly is made up of three components: two symmetrical platform supports and a top plate. The platform assembly slides onto the front frame flange and is held in place by a single clevis pin. The platform assembly is designed to support on board control and navigation systems, external sensing devices and other equipment.

4.9 Safety.

We conducted a FMEA and standards search to provide safety guidelines for our design. We identified three areas of concern: the motors and gears, the electrical circuitry components and performance of the rover.

We have followed the National Electrical Manufacturers Association (NEMA) safety guidelines for the degree of enclosure of our motors. With this design we have 100% enclosure of the motors and gears, eliminating any possible hazards exposed to the user. We have designed the wire management system such that wire bundles are routed through the shaft, away from the moving gears. Service loops are confined to the of the wheel cavity, away from the rotating wheel housing.

We have specified the use of 18 gauge Teflon coated wire which is rated at 10 amps. The motors draw a maximum current of 4 amps, therefore we are within safe operating limits of the wire. Overload protection has been specified in the form of a 10 amp fuse for the motors and 7.5 amp fuse for the electronics. The main function of these fuses is to protect the components from power spikes due to any possible short circuits. An emergency cut off switch that will enable the user to interrupt power system has also been specified. The switch must be located at a point that is easily and quickly accessible to the user. To reduce the possibility of injury we have required the manufacturer to remove all sharp edges and burrs from the machined parts as suggested by the American Society for Testing and Materials (ASTM). Additionally, ASTM also outlines vehicle stability requirements to minimize hazards that may occur if the vehicle becomes unstable during operation. We have met these requirements by keeping our center of gravity low to the ground, thus eliminating the possibility of roll over in the prescribed operating environment.

4.10 Cost Analysis.

We estimated the cost of the wheel and frame system to be $4320.00. The raw material costs were approximately $270.00, or about 20% of the overall cost. The manufacturing costs were estimated to be $3000.00. The remaining $1050.00 was for the purchase of off the shelf components such as gears and bearings.

5. NAVIGATION AND CONTROL SYSTEM DESIGN

5.1 Customer Requirements.

The following are customer requirements that the rover navigation and control system must meet:

1. Total weight of navigation and control system less than five pounds.
2. Housings must contain the on board electronics and sensors.
3. Housings must be less than eight inches in width.
4. Housings must be less than twelve inches in length.
5. Housings must be less than eight inches in height.
6. Housings must allow access to circuitry in three steps or less.

7. Remote control (RC) system most be detachable from the rover.
8. Remote control system must control all six motors.
9. Remote control system must weight less than five pounds.
10. Tilt sensor must detect a 30 degree slope.
11. Tilt sensor must fit into a 4" by 4" area.
12. Test environment must fit into a 10' by 4' area.
13. Test environment fill material must simulate Ames Research Center's soil.
14. Budget for navigation and control system must not exceed $1,000.00.
15. Rover must come within three centimeters of specified target.
16. Rover must avoid objects obstructing its path.
17. Rover must not traverse terrain with slopes of greater than 30 degrees.
18. Motors must have variable speed control.
19. Motor controllers must have analog control from QED board.
20. Motors must have supply voltage of 12 volts at 1 amp per motor.
21. Motors must have independent left and right motor control.
22. Motors must achieve 1800 rpm at .8 amps.

5.2 Literature Research.

We expected that there would be quite a bit of literature on autonomous robots, and this was indeed the case. However, a vast majority of the information that was found in our literature search dealt primarily with the electrical aspects of autonomous robots. Although much of the research available concerning the problem statement deals with much larger rovers, we found it helpful in determining a direction for our initial designs. Many faculty members responded positively to our requests for information and their help was greatly appreciated. Our textbooks provided another invaluable source of information on a variety of topics from amplifier design to the Friis transmission formula. As far as the mechanical components were concerned, the literature was relatively nonexistent. Existing designs of navigation and control systems of autonomous robots were investigated, however, no design information was uncovered. In addition to this, our design was constrained to be an original design since it was to be retrofitted to the existing WALRUS design.

Most of the texts that we reviewed spent a great deal of time defining autonomy and how autonomous robots function. Simply put, an autonomous robot is one that has the ability to complete a desired function without any human intervention. Although the information that we found in our literature search did not directly apply to our design, it did help give us a more general understanding of the magnitude of this project, as well as the experience in research.

Autonomous Mobile Robots, by A. Meystel covers a number of interesting topics involved with autonomous robots. One of these topics is the evolution of autonomous mobile robots; from ELSIE (Electro-light-sensitive Internal-External), one of the first AMRs, to the Drexel-buggy, which is a state of the art AMR. One interesting point that was uncovered was that all present autonomous rovers reviewed utilized off-board computer control, which was not an alternative for the WALRUS design. In addition to evolution of AMRs, Autonomous Mobile Robots describes the need for nested hierarchical control of AMRs. Nested hierarchical control is the use of computer programs to setup a problem solving system for the robot. This is important because it allows for efficient problem solving techniques for the AMR.

This book also talks about the "Intelligent Module," or the use of a Planner and a Navigator to set up a global positioning scheme for the nested hierarchical control to use in its problem solving. [3]

Another book that we uncovered, How to Build a Computer Controlled Robot, described how to build, control and use an autonomous robot from the ground up. Unfortunately for us, this book only talks about a single robot, named MIKE. MIKE is a large robot (approximately four feet tall at completion) and is designed specifically to move about a room. Due to the difference in physical structure and navigational system objectives, designs from MIKE were not applicable to the WALRUS design. [4]

Iyengar and Elfes, in their book Autonomous Mobile Robots: Control, Planning, and Architecture, describe the use of VITS, a vision system for autonomous vehicle navigation that directly ties in with the future objective of the WALRUS – virtual reality. This book also describes mobile robot modeling and control, describing the kinematics involved in robot design. This is more in line with the design of the chassis and propulsion system of WALRUS. In addition, this book reiterates a lot of the same information about decision making, hierarchical control and global location that was found in Meystel's Autonomous Mobile Robots. [5]

Far more information was available to us as far as radio controlled systems were concerned. One of our group members has quite a bit of experience with radio controlled aircraft, and his knowledge was found to be invaluable.

Radio control systems have been used for controlling model cars, trucks and airplanes for several years. We used several hobby magazines as a source for information about the use of radio control in various vehicles. In addition to the personal knowledge of one of our group members, we used magazine and catalog information, and advice from a local hobby shop when designing the radio control system for the WALRUS.

In addition to the literature review that was performed, we obtained background information on navigation and control of autonomous robots. This information was found to be even more valuable to us than the material uncovered in the literature review because it was put to us in laymen's terms. A brief synopsis of obstacle avoidance information can be found in appendix A, and information concerning global location is in appendix B.

Another goal of our project was to examine the use of Polaroid ultrsonics in obstacle avoidance. Polaroid ultrsonics are high voltage (approximately 300 volt bias), large band width ultrasonic transmitters and receivers. Due to technical problems associated with the piezoelectric ultrsonics (low voltage and high bandwidth), we believed that it would be worthwhile to examine Polaroid ultrasonics. We came to this conclusion through a combination of talking to faculty and realizing that Polaroid ultrasonics are commonly used in commercial cameras for automatic focusing.[6] We performed testing of the response of Polaroid transducers to various objects. We wanted to characterize the response of Polaroid ultrasonics to various objects in an effort to calibrate an ultrasonic based obstacle avoidance system.

A transducer frequency of 55 kHz and an amplitude of ten volts was used. A biasing voltage of 300 volts was used. A filter was used to separate the desired signals from background noise.

All objects were held approximately 0.5 meters from the face of the transducer. The transducer produced a continuous signal and the transducer signal together with the response signal were displayed. A QuickBASIC computer program was used to "capture" the oscilloscope screen in a data file on a personal computer.

The results received from the response signal were in terms of voltages. The voltage amplitude of the response signal is proportional to the strength of the signal. As suggested by faculty, we based our analysis on the standard deviation of the voltage signals. The greater the amplitude of the voltage, the greater the standard deviation of the voltage, from the mean voltage of zero, will be. Therefore, the standard deviation of the signal is a measure of the power of that signal. The results of our testing can be found in appendix F. A basic trend that was noticed was that the more vertical surface that an object had, the more powerful its return signal would be.

5.2 Concept Evaluation.

The design of the navigation and control system for the rover was divided into two parts. The mechanical engineering aspects were assigned to a group of mechanical engineering seniors. The electrical engineering aspects were assigned to a group of electrical engineering seniors.

5.2.1 Mechanical

Two basic issues arose in the design of the housings for the navigation and control system of the rover. First, a decision had to be made on whether to place all the electronics in one main housing (centralized housing) or divide the electronics up into two different housings (decentralized housing). It should be noted that the housing question deals only with the QED, tilt-sensor, ultrasonic control boards and motor control board. The RC system was to be removable from the rover when not in use, so it would require its own housing. Second, a decision had to be made whether to go with vendored housings or custom manufactured housings.

The decision of whether to have centralized or decentralized housing was based on four criteria. The most important criterion was camera visibility. At the beginning of the project, we were told that a pan and tilt video camera might later be mounted on the center post of the rover. This camera was supposed to be able to view the areas directly around the rover, including the area directly in front of the rover. Therefore, it was imperative that the housings not interfere with the line of sight of the camera. The next most important concern was the compatibility of needs of the QED board and the motor control board. According to its specifications, the QED board was sensitive to dust. Therefore, it should be isolated from the test environment as much as possible. The only circuit that produced any significant amount of heat was the motor control board. The motor control board could generate up to 15 watts of heat when it was in operation (that is, when the rover is changing speed or direction). Experimentation revealed that a combination venting system and cooling fins would be sufficient to cool the motor control board, however, venting of the motor control board and isolation of the QED did not easily go together. The next most important criterion was easy access to all circuitry. The least important criterion was centralized wiring. The less spread out the wiring was, the less likely it was to get entangled with some part of the rover.

Table 2 is a decision matrix that shows how the two alternatives (centralized or decentralized housing) rated in terms of each criterion. The criterion were weighted in order of importance (from 4 to 1). The alternative that best met the criterion was given a raw score or 1 and the alternative that least met the criterion was given a 0. This raw score was than multiplied by the weight factor to get a weighted score. Based on the size of a single housing that would house all the electronics, going with two housings would definitely cut down on the amount of interference with the video camera. Likewise, two would eliminate the compatibility in needs between the QED and the motor control board (they could be put into separate housings so the ventilation required by the motor control board would not interfere with the isolation required by the QED). In terms of easy access, both alternatives could be easily accessed from a lid on the top. Finally, more centralized wiring would definitely be favored by only having one housing. As table 1 shows, when the scores were totaled up, decentralized housing was the best choice.

		Centralized	**Decentralized**
	Weighting	Score	
Camera	4	0	1
Compatibility	3	0	1
Access	2	1	1
Central Wiring	1	1	0
Total		**3**	**9**

Table 2. Decision Matrix for Centralized Versus Decentralized Housing.

The decision concerning whether to go with custom of vendored housings was based on two basic criteria. The most important criterion was space utilization. Do to weight constraints, it was important that the housings be no larger than absolutely necessary. The least important criterion was machinablity. Various external connections would need to be added later, so it was important that we be able to easily machine the housings.

Table 3 is a decision matrix showing how the alternatives rated with each criterion. It has the same

		Custom	**Vendor**
	Weighting	Score	
Space Utilization	2	1	0
Machining	2	1	0
Total		**4**	**0**

Table 3. Decision Matrix for Custom Manufactured Versus Vendored Housings.

scoring format as Table 2, except the weightings range from two to one. As expected, the custom housing would allow for more freedom both in terms of space utilization (we could build to fit whatever board configuration we needed) and machining (since we built it, we could easily machine it). Therefore, a custom housing was the best choice, at least for the "front" housing, which would house the QED, tilt-sensor, ultrasonic boards and the RC housing (the exact design of these housings will be covered later in the discussion of our design). Since the back housing housed only the motor control board, and required only two 12-pin connector to be added later, we were able to find a vendored housing (from Radio Shack), which fit the board perfectly. Therefore, we used both custom housings (for the "front and RC housings) and a vendored housing (for the motor control board).

As mentioned previously, one of the project goals was to design and construct a method for on board determination of the slope of the ground on which the rover was located.

One of the alternatives suggested by Ames Research Center was to look into a two-axis tilt sensor made by Lucas-Schevitz. This sensor would send out voltages that correspond to degrees of slope up to 20 degrees (for further detail, refer to the discussion of our design). Based on our experiences with model airplanes and helicopters, another option that we found was to use a gyroscope.

The two alternatives were judged according to four criteria. The most important criterion was the ability of the alternative to interface with the QED. If it could not interface, it would not work. The next most important criterion was the cost of the system. As mentioned in the customer requirements, the project budget was $1,000.00. The next most important criterion was the accuracy of the system. Based on weight constraints, the size of the alternative was the least important criterion..

Table 4 is a decision matrix showing how the two alternatives rated with each criterion. It is scored the same as tables two and three. The clinometer sends out a voltage signal which could be read directly by the QED. The gyroscope was designed to control servo motors, so it would not be so easily interfaced. In terms of cost, the tilt-sensor was $85.00 while the gyroscope was $150.00. While the tilt sensor was designed to give voltages responding to slope, the gyroscope was designed to respond to drastic changes in slope, so it was not as accurate. The tilt-sensor and gyroscope were about the same size. When one adds on the fact that the tilt-sensor was customer recommended, it was the best choice.

		Clinometer	**Gyroscope**
	Weighting	Score	
Interface	4	1	0
Cost	3	1	0
Accuracy	2	1	0
Size	1	1	0
Total		**10**	**0**

Table 4. Decision Matrix for Tilt Sensor Versus Gyroscope.

In terms of our RC system, there really were not any alternatives that we could find. All commercially available RC systems that we found were for control of at most two motors. The rover requires control of six motors. Therefore, we knew we had to go with a custom made system.

5.2.2 Electrical

The navigation portion of this project was broken into two different sections. One concerning obstacle avoidance and the other concerned with the global location of the rover inside the test area.

To meet the requirements of the global location scheme three different alternatives were discussed:

1. Radio frequency alternative.
2. Ultrasonic alternative.
3. Combination external optical/audio and internal compass alternative.

The first alternative addressed in the selection of a global location scheme was the use of Radio Frequencies. The idea of determining the location of the rover was based on the Friis Transmission Formula [7].

$$P_L/P_t = (A_{e1}A_{e2})/(r^2\lambda^2) \qquad \text{EQ - 1}$$

In equation 1 above, P_L and P_t are the transmitted and received powers, respectively. A_{e1} and A_{e2} are the effective areas of the transmitting and receiving antennas, r^2 is the square of the distance separating the atennas and λ^2 is sthe square of the wavelength. Knowing the transmitting and receiving powers, the wavelength of the signal and the effective areas of the two attennas, the distance between the two can easily be calculated. Once in the laboratory, this alternative fell apart. Measuring the effective areas of the antennas requires a very error prone procedure using the network analyzer and once the measurement has been made, any jostling or other handling of the antennas is apt to change the areas. The other downfall of this method is the inability of a precise measurement of transmitted and received power due to reflections and other interference. The third and most destructive fault of this system is the small differences in power seen within the test area. The precision of the system as a whole is lost when the previous faults are coupled with the extremely small changes that must be measured in order to determine the rover's global location.

The second alternative for determining the rover positionsing was through ultrsonic beacons. The beacons would be placed around the test area. By knowing the speed of sound and by timing the pulses to and from the beacons, the QED board could determine the distance from each beacon and update its internal map as to its X and Y coordinates. The main problem with this application is that ultrasonics are not normally used for this application and, consequently, there is very little available research to draw upon when attempting to design such a system. Also the ultrasonic transmitter/receivers that were available for use began to lose accuracy at approximately six meters. Since the system requires three centimeters of accuracy at a maximum of ten meters of distance, this alternative clearly would not accomplish the task.

The third design alternative combines the use of external beacons with an internal compass. While the compass can continually monitor the speed and heading of the rover, it lacks the ability to correct itself once off track. To match the internal map with the actual position of the rover on the test area, the external beacons will be used. By accessing the external beacons at regular intervals the accuracy of the internal map is insured. The problem of accuracy falling off as the distance increases is solved by the use of light in the beacons configuratoin instead of sound. The problem of lost accuracy due to an object blocking the external signal is solved by using the internal compass. This scheme also makes use of the two axis tilt sensor for more than just a warning device, by taking the cosine of the angle present on the tilt sensor, the horizontal distance traveled can be determined.

In order for a global location scheme to be practical it must be accurate, within the specifications at the maximum possible distance in the test area, i.e. the diagonal distance of 14.14 meters. The scheme must be able to account for the rover changing direction due to an obstacle or an incline that is determined too steep to climb. Lastly, the scheme must incorporate some sort of "Dead Reckoning" tp update the internal map when the rover is behind a hill and incapable of being updated by the external beacons.

To solve the problem of obstacle avoidance, three different sensors were considered:

1. Infrared.
2. Ultrsonics (Piezoelectric versus Polaroid).
3. Tactile.

For the obstacle avoidance scheme to be practical it must be able to determine the difference between an obstacle it can conquer and an obstacle that is too large for the rover to cross safely. The system must account for drop-offs and not be fooled into thinking that a simple upward slope is some large obstacle.

The schemes for the obstacle avoidance and the schemes for the global location must interface with the QED board.

The design alternatives for the motor control circuit were limited. Given the power supply of the rover and the specifications that the motor control circuit must satisfy, the choice of different designs was constrained to:

1. Solid state TTL control.
2. Bridge control though the use of Darlington pairs.

Solid state TTL control uses integrated circuits to control the motors speed and direction. Speed is varied through current limiting and the direction of the motor rotation is governed by exchanging the positive and negative terminals, thus changing the direction of current flow.

The design using Darlington pairs produces a voltage across the terminals of the motor when the transistor is turned on by creating a path from the positive bus line to ground, through the motor. Changing the motors direction is accomplished simply by turning on the opposite pair of Darlingtons, creating a path to ground through the motor in the opposite direction.

For the motor control to be practical it must be able to supply high currents to the motors and survive corresponding voltage spikes that accompany the control of the motors.

The problem of system control for the rover could be solved through the use of any one of the following four devices:

1. Microprocessors.
2. Digital signal processing integrated circuits (IC's).
3. Input-Output (I/O) control devices.
4. QED microcomputer board.

Each system must have the ability to handle the task of global location, obstacle avoidance and motor control in a near simultaneous manner. There are a considerable number of ways that one could implement a control/navigation system for WALRUS given the variety of microprocessors, digital signal processing IC's and I/O control devices available on the market. The cost of many of these devices would be well within the budget constraints of this project. Control/navigation systems, however, tend to be very complex in nature and even the most rudimentary autonomous vehicles usually have fairly sophisticated processors and control circuitry on board. WALRUS fortunately has a fairly straightforward task in terms of its navigational activity. NASA Ames Research Center has specified that the rover must be able to navigate from some starting point (X_0,Y_0) to a destination point (X_1,Y_1) autonomously, or via remote control. Moving from one point to another is a fairly easy task to implement, if the rover were only required to move over a completely flat surface with no obstacles. The terrain to be used at the NASA center will be a simulation of a hilly and rocky surface such as that found on the moon or on Mars. The challenge is to provide a control system that allows the rover to properly control its wheels, determine its location at regular intervals, detect obstacles, avoid obstacles and find its predetermined destination within three centimeters. These tasks can be fairly complex by themselves, not to mention that they all must be performed somewhat simultaneously.

While a CRAY with 80 gigabytes of random access memory (RAM) and an ethernet link would probably be overkill for the WALRUS, providing the required level of control cannot be accomplished using a few logic gates and s simple microcontroller chip. The WALRUS will require enough processing power to perform its internal as well a I/O routines in a timely manner. There are many external devices that must be controlled and monitored, therefore the onboard computer must be I/O rich and have a well designed "operating system." The operating system is the computer software written to manage system resources, both internal and external. Because several tasks are being performed simultaneously as the rover moves about, a multitasking operating system wouls be appropriate. A multitasking operating system allows several computer programs to run at the same time. If the computer has only one central processing unit (CPU), then of course there can be only one instruction being executed at a given instant in time, but a multitasking operating system (OS) forces the programs to alternately share the CPU, thus making it appear that they are running simultaneously. OS2, UNIX and X-Windows are common examples of multitasking operating systems that are used on large computers and some PC systems. Writing a good multitasking OS is a senior design project by itself, not to mention the complex array of hardware necessary to properly interface all of the external devices on the rover to the main CPU (usually a microprocessor or microcontroller). NASA engineers, who were aware of this problem when the WALRUS project was given to the University of Idaho, decided on a suitable computer system for the rover which met all of the above criteria. The decision (made in the Fall of 1992) was to use the QED controller board by Mosaic Industries, Inc.

The QED board is a Motorola 68HC11 microcontroller based computer system. This board has a considerable number of I/O ports including twelve digital ports, eight 8-bit resolution and eight 12-bit resolution analog to digital (A/D) ports, eight 8-bit digital to analog (D/A) ports, two serial ports, eight megabytes of addressable external memory and a keyboard/liquid crystal display (LCD) interface. The board also has a 128K battery-backed RAM which can be used for software development and data storage. The software included with the QED is a Forth version 2.0 compiler and multitasking operating system, stored on a PROM IC. The QED-Forth compiler accepts Forth source code from its serial input port, compiles it into machine code and stores it at a user specified location in the battery-packed RAM. The Forth compiler vastly improves software development time,due to the fact that Forth is a third generation language (like FORTRAN or C) versus the traditional assembly language that is usually necessary to program most microcontroller IC's. In other words, software that would normally take three semesters to write in assembler, could be written in only one using the Forth compiler. The QED board is easily interfaced to an IBM (compatible) PC via RS232 cable. The software developer simply writes the source code on his or her editor of choice and downloads the code to the QED board from their computer.

Another major advantage of the QED board over a conventional microcontroller is its multitasking operating system. As mentioned before, the multitasking OS is necessary if many processes are to be preformed at the same time. The WALRUS will at the very least have obstacle avoidance, triangulation and motor control taking place at almost the same time. The QED-Forth environment has a built-in multitasker, whereby the programmer specifies wthat programs go where in memory, what hardware resources are needed by each specific program and how much available RAM each program can use. The QED operating system takes care of the rest once the system starts running. Multitasking is reality causes the multiple programs to take turns using the CPU at a high rate of speed. An extremely high rate can couse a lot of time to be wasted doing OS "administrative" tasks, thus QED-Forth also allows the user to specify the switching rate. Despite its computing power and abundance of I/O cpabilities, the QED board draws only 500 milliampers of current. The processor runs at eight or sixteen MHz depending on the crystal used, giving it the speed and throughput (approximately 0.0125 MFLOPS) similar to that of a 286 IBM PC. The entire board is only 3" by 4" and about 1" high. This compactness is another desirable trait, given the limited space inside the rover's housings. Given that the QED board meets the above criteria for speed, I/O control and multitasking capability, it has been decided that this device will be suitable for the control and navigation of WALRUS.

5.3 Detailed Design.

5.3.1 Mechanical

Figure 4 shows what the final assembled rover will look like. The rover has been designed so that it can travel with equal efficiency and agility regardless of it's direction of travel. For this report, however, we will refer to the end of the rover with the main housing as the "front" of the rover.

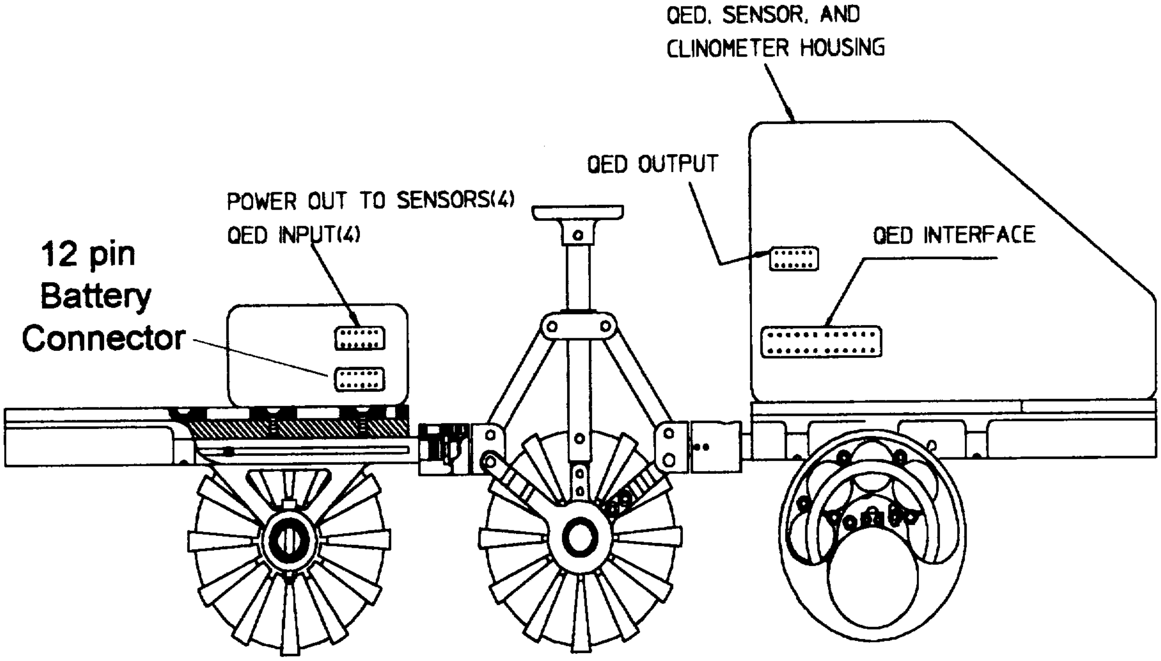


Figure 4. Assembled Rover.

The main housing is mounted to the front of the rover. This housing is connected to the batteries, motors and rear housing with cables that run between the 12-pin and 25-pin ports that are seen on the sides of the housings. The rear housings and it's platform are, of course, mounted to the rear of the rover frame. Again, cables from and to the ports shown on the side of the rear housing are used to connect this housing to the other components on the rover. Either the front or rear housing can be replaced by the radio control system housing. The RC housing will have a direct link to the motors and batteries and will not be connected to the other housings during operation. Cabling between the housings will be attached to the rover frame with a loose cable tie. This tie will prevent the cables from dangling but will, at the same time, allow free movement of the cables during articulation of the rover frame. The cables connecting the motors and batteries to the housings will be routed through the axles and frame members of the rover as specified in the original rover design.

The main housing for the WALRUS rover will contain the QED microcontroller, the 2-axis tilt sensor, the global navigation circuitry and the obstacle sensor circuitry. It weighs 4.8 pounds, slightly below the goal of five pounds. Because of the delicate nature of the QED circuitry, this housing needed to be dust free and be able to withstand the stresses that would be associated with a roll-over. Due to the complex geometry of the rover, it was hard to analytically determine the exact nature of forces that might occur during a roll-over. Therefore, we estimated that the maximum load the housing would carry would be the maximum weight of the rover, which was set at 45 pounds (including payload).

We decided to use 1/8 inch thick, 6061 aluminum for the housing material. The high strength an low weight of aluminum were the deciding factors in our choice of material. Moreover, we were able to take advantage of an experienced aluminum welder who offered to weld the housings and supply the aluminum at no cost. The welds would give the housing an excellent seal against dust and added joint strength. Another important factor was the fact that the rest of the rover was made from aluminum, so the housing would match. We used 1/8 inch thick aluminum so that the housing could withstand a roll-over. In some preliminary testing, the housing held well over 45 pounds. Furthermore, the 1/8 inch aluminum could also support most peripheral devices that Ames Research Center might later mount to the housing. We could have used thinner plates with ribs for added support, but this would have made the manufacturing more difficult.

The final design for the main housing is shown in figure 5. The slope of the lid is intended to allow better vision of the front of the rover by a camera mounted on the center post. The horizontal portion of the lid was left in case NASA desires to mount peripherals on the top of the box.

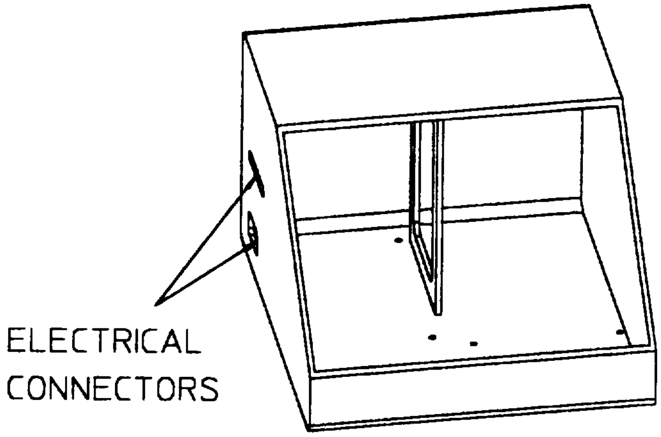


Figure 5. Front Housing Design.

The interior of the main housing has the layout shown in figure 6. This layout was designed with the idea of maximum space utilization. Room was also left for wiring and a global navigation system. The

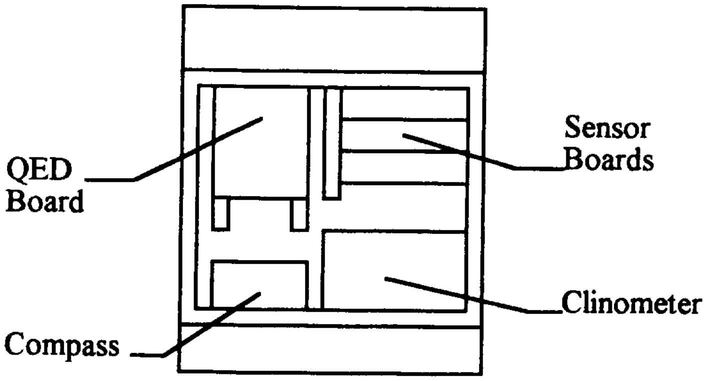


Figure 6. Layout of Front Housing.

ultrasonic sensors' circuit boards are stacked in the back part of the housing. They are held in place with circuit board slide mounts purchased from Digi-Key mail order. The mounts are epoxied in place within the housing. We tested the epoxy bond in the shop by having two of our group members pull as hard as they could on an epoxy bond. The bond did not yield. The amount of force used in the test of the epoxy bond would have been enough to break the circuit boards being held by the mounts. Because of this, we feel that the epoxy bonds will be sufficient for this application. The QED board is mounted to the floor of the housing, in the rear, next to the ultrasonic circuits. This placement was chosen to reduce the length of wiring necessary to connect the ultrasonic boards with the microcontroller. The 2-axis tilt sensor is mounted in the front of the housing with bolts that extend through the bottom of the housing which would result in erroneous readings. We left room next to the 2-axis tilt sensor for the circuitry needed for the global location system.

All of the components within the main housing are connected to the QED microcontroller with wiring harnesses. The QED board and the ultrasonic circuits are connected to the motor controller and sensors via a 12-pin plug that has been mounted in the side of the housing. We also included a 25-pin port in the side of the main housing to allow programming of the QED board. This port will allow the programming of the QED board while it is still mounted in the housing without the added space requirements for the necessary keypad and screen. The keypad and screen are contained in a seperate housing that were provided with the microcontroller. The keypad and screen will be connected to the 25-pin port with a ribbon wire.

The rear housing materials and method of construction were originally planned to be the same as the main housing. We realized, however, that this housing did not have the same strength requirements as the main housing. With this limitation removed we chose to use a housing available from Radio Shack. This housing has a sheet metal floor and plastic sides and top. While the black plastic does not match the aluminum main housing, the combination of the two materials still fits in with the appearance we felt was appropriate for a space research vehicle.

We are comfortable with specifying a plastic housing for the rear housing because we know from the rover's frame design that the rear housing will not come in direct contact with the ground in the event of a roll over. The rover's center post and the extension of the rear platform will support the majority of the rover's weight in a roll over situation. If the rear housing's plastic lid damaged in a collision, it's replacement cost is approximately two dollars. We feel that this is an acceptable risk.

The motor controller board will be housed in the rear housing. The motor controller board has the potential of producing 15 watts of heat during operation., This heat could possibly cause problems with the QED board and 2-axis tilt sensor. As mentioned before, this is the reason we chose to separate the motor controller board from the rest of the circuits. Heat will be dissipated from the transistors on the motor controller board with finned heat sinks. These fins will transfer the transistor's heat to the air inside the motor controller housing. The air within the housing will circulate with outside air through filtered vents in the housing.

The rear housing, like the main housing, will have 12-pin plugs mounted in it's side to allow interface with the QED microcontroller, the rover power supply and the rover's motors. These plugs will be epoxied into slots cut in the side of the housing.

The platforms for the front and rear of the WALRUS rover serve three purposes. The two purposes directly related to our project are to secure the housings and to hold the obstacls sensors. The third purpose for the platforms is to provide an interfafce between the rover frame and the housings.

The materials chosen for the platforms were a combination of aluminum plate and aluminum sheet. The platforms themselves are constructed of 0.25 inch aluminum plate. The ultrasonic obstacle sensors will be mounted in aluminum sheet that is attached to the front of the platform. Figure 7 shows the desingn of the platforms and sensor mountings.

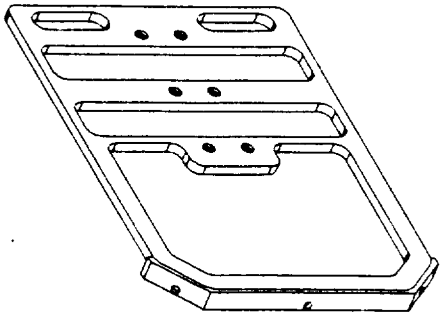


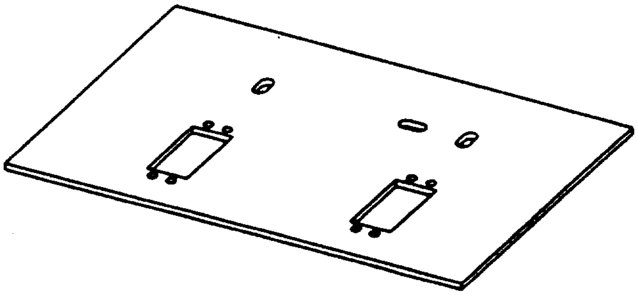
Figure 7. Mounting Platform Design.

To simplify manufacturing, the two platforms were designed to be almost identical. The platforms will mount to the rover frame via mounting rails. These rail were specified as part of the original rover design and are being manufactured by NASA. The rails will attach to the platforms through the six countersunk holes in the center of the platforms. The platforms will be attached to the housings with bolts.

We knew from our preliminary discussions with NASA that an optional radio control system for the WALRUS rover was desired. Because of this desire and also because of the difficulties encountered with the semi-autonomous navigation system, we decided to desingn and build a radio control system. The resulting design consists of the radio control system and it's housing. The goal of this RC system was to provide an alternative to the semi-autonomous control system being designed for the WALRUS rover. This RC system could be used to control the rover in situations beyond the capabilities fo the semi-autonomous system or to calibrate the autonomous system to different environments.

The housing for the RC system was constructed by the same individual who built our main housing. 1/8 inch thick, 6016 aluminum was chosen for the same reasons as outlined earlier. Within the housing, the on-board components of the RC system are held in a tray. This tray is constructed of 0.0625 inch thick aluminum sheet which is supported in the housing by four 0.375 inch square aluminum posts at the tray corners. The weight of the tray and contents are sufficient to hold the tray in place within the housing.

The design and layout of the RC housing and tray are shown in figure 8. All wiring and circuitry necessary for this system are contained below the tray.



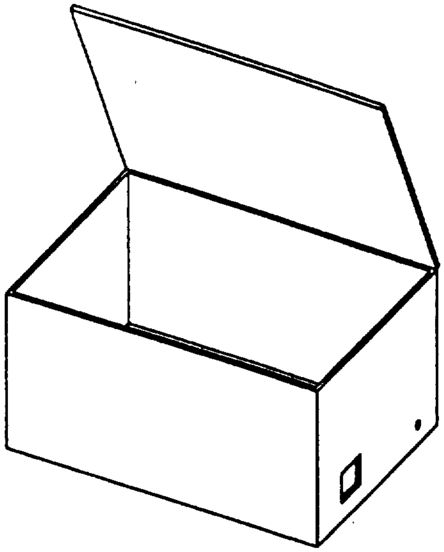


Figure 8. RC Housing and Tray

The RC system will interface with the rover batteries and motors via a 12-pin plug located in the side of the housing. As done with the other housings, the plug will be held in place with epoxy. The RC housing moounts to a smaller version of the platforms used for the main and rear housings. Like the other platforms, the RC platforms will be attached to a mounting rail which will attach to the rover frame.

The choice of a radio system was driven mostly by cost. While more expensive radio systems have extra features that would be fun to have, these features are not necessary given the increased cost associated with them. In the end, we decided to use a simple, two channel, AM radio system. This system, a FUTABA, uses two standard servos, a two stick transmitter, a standard receiver and AA batteries for transmitter/receiver/servo power.

The method of driving the potentiometers with the servos was dictated by the specifications of these two components. The servos have a total range of motion of 90 degrees. The potentiometers we found have a total range of motion of 312 degrees from full-on to full-off. This made the use of gears our only feasible alternative. We decided to use a three inch gear for the potentiometers and a one inch gear for the servos. The servo/single gear combination is the speed control for the rover and is controlled by the left stick on the transmitter. As the left stick is pushed forward, the potentiometer shaft is rotated, reducing the resistance of the potentiometer. The servo/two gear combination controls the direction the rover is traveling. This part of the system is controlled by the right stick on the RC transmitter. As the right stick is pushed to the right, the left potentiometer's resistance is reduced. At the same time, as a result of the series connection of the gears, the right side potentiometer's resistance is increased. This causes the rover to turn to the right. Pushing the right transmitter stick to the left reverses the process csusing the rover to turn to the left.

The servos and potentiometers are mounted in the servo tray. A switch controlling the RC system's receiver/servo batteries is also mounted in the tray surface. The RC system batteries and the receiver are mounted under the tray. Velcro patches prevent the batteries and receiver from rattling around in the housing and getting damaged. Also mounted under the tray is the small circuit that relays the rover battery power through the potentiometers and to the rover's motors. This circuit consists of a 1.1 kilo-ohm resistor and TPI transistors. A a flow diagram of this circuit and it's connections to the potentiometers, batteries and motors is shown in figure 9. Power from the rover's batteries and to the rover's motors is routed through the 12-pin plug mentioned previously.

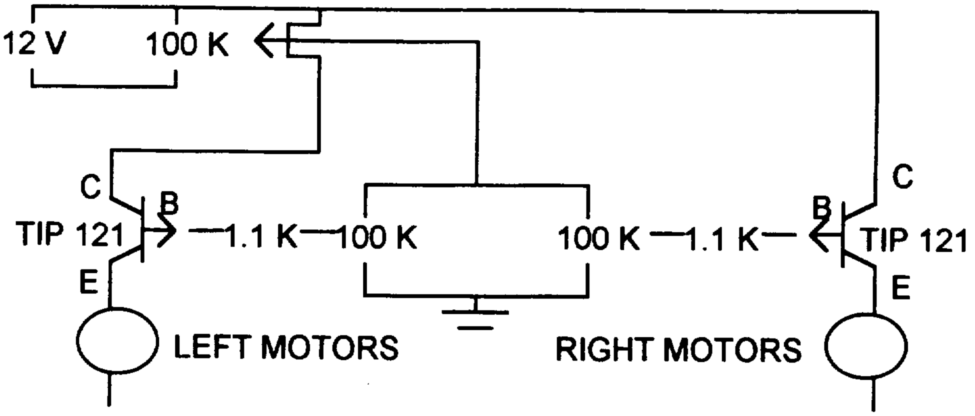


Figure 9. Remote Control Circuit Flow Diagram.

The sensor chosen for angle of tilt determine is a commercially available two axis tilt sensor purchased from Lucas-Schevitz. This sensor is able to communicate two axes of tilt simultaneously and has the ability to accurately and linearly measure angles up to 20 degrees. The original design specifications called for the ability to determine up to a 30 degree incline. We found, however, that there was an extremely large price difference between the 20 degree sensor and the 30 degree sensor. We presented this situation to NASA and they requested that we use the less expensive, 20 degree option. We have connected this sensor to the QED microcontroller and tested it's response to differing angles of inclination. As mentioned before, the sensor responed in a repeatable, linear fashion up to a 20 degree tilt. The sensor also had no problem communicationg with the QED board and the program that was written for it by our computer engineer.

5.3.2 Electrical

The computer control system used for WALRUS is based primarily on the QED computer board designed by Mosaic Industries Inc. The general background on this board was discussed previously, where it was mentioned that several important criteria had to be met by any control device used for navigation and control of the rover. The QED board was decided upon as an acceptable device which meets all of our basic requirements within a reasonable cost. The following is an explanation of how the QED-Forth software operates and how its processor and multiple I/O circuits are used to control the proper navigation of WALRUS.

As mentioned previously, the QED board relieves the programmer of the burden of assembly language programming. While the QED system does allow assembly programming if one desires, in most cases it is far more efficient and easy to use the on-chip Forth compiler. This compiler is an actual PROM chip on the QED board itself, which takes Forth source code from a serial port and directly compiles it to memory. The software development system is very easy to use, requiring only an ASCII text editor (such as the EDIT program in DOS), the Kermit communications software (provided by Mosaic Industries) and an IBM compatible PC. Forth is a third generation language; this language can probably be most likened to a combination of Hewlett-Packard calculator language and C language. The programming is done on a PC and when the program is finished, it is simply downloaded onto the QED via the RS232 port. The Forth compiler also performs dynamic error checking as the source code is downloaded. Once the program is loaded, QED-Forth also has a debugging feature which allows single stepping through each source command so that runtime errors can be found.

The flow chart of figure 10 shows the basic outline of the navigational software used to control WALRUS. While it appears that each program block is being executed in a linear fashion, in fact there

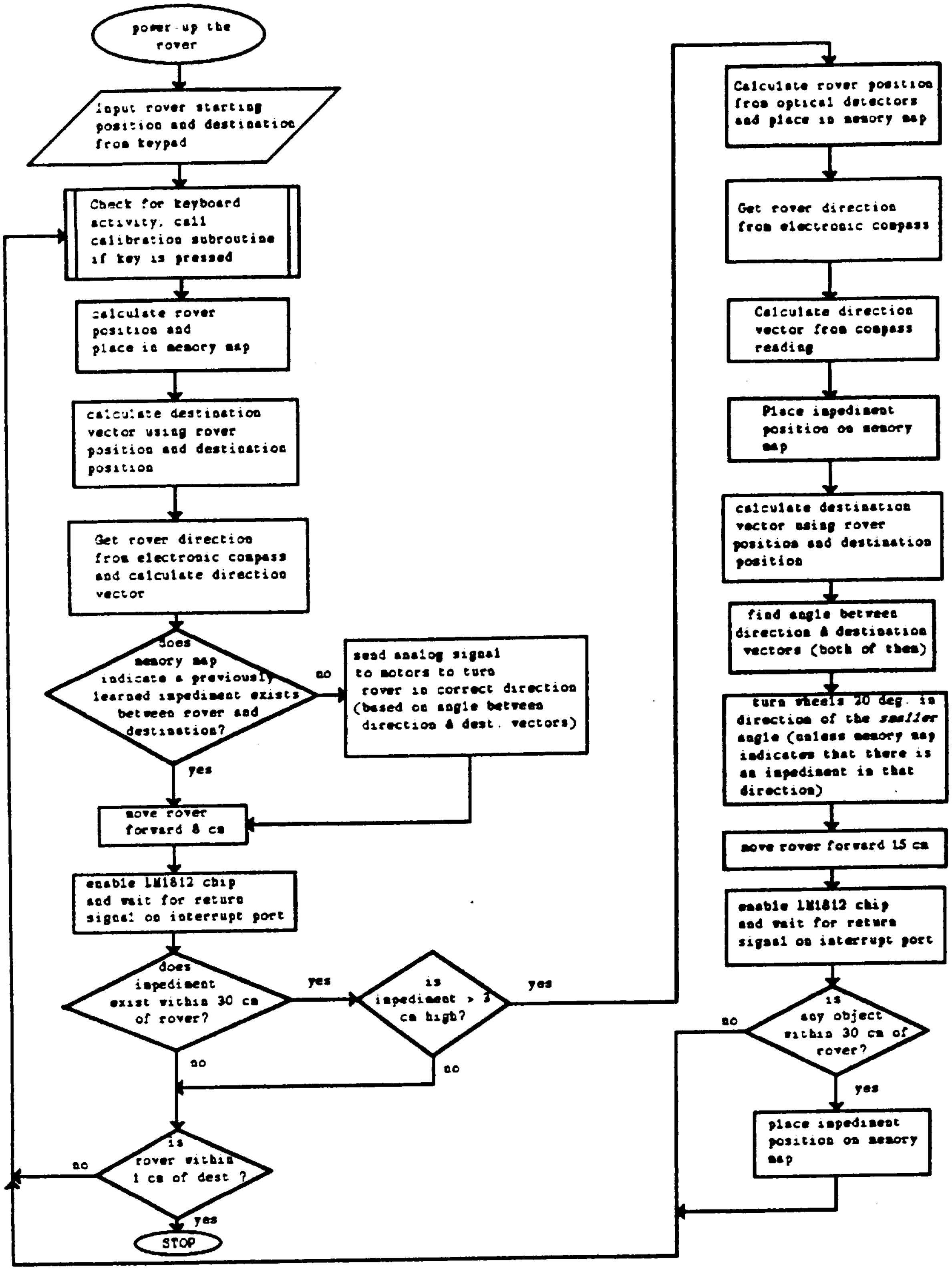

power-up the rover
input rover starting position and destination from keypad
Check for keyboard activity, call calibration subroutine if key is pressed
calculate rover position and place in memory map
calculate destination vector using rover position and destination position
Get rover direction from electronic compass and calculate direction vector
does memory map indicate a previously learned impediment exists between rover and destination?
no
send analog signal to motors to turn rover in correct direction (based on angle between direction & dest. vectors)
yes
move rover forward 8 cm
enable LM1812 chip and wait for return signal on interrupt port
does impediment exist within 30 cm of rover?
yes
is impediment > 3 cm high?
yes
no
no
is rover within 1 cm of dest ?
no
yes
STOP
Calculate rover position from optical detectors and place in memory map
Get rover direction from electronic compass
Calculate direction vector from compass reading
Place impediment position on memory map
calculate destination vector using rover position and destination position
find angle between direction & destination vectors (both of them)
turn wheels 30 deg. in direction of the smaller angle (unless memory map indicates that there is an impediment in that direction)
move rover forward 15 cm
enable LM1812 chip and wait for return signal on interrupt port
is any object within 30 cm of rover?
no
yes
place impediment position on memory map

Figure 10. QED Software Flow Diagram.

are usually several tasks being performed simultaneously. This brings up the issue of the multitasking operating system. Computers that utilize a simple operating system run programs whereby one instruction is followed by another. No two functions can be running concurrently. This leads to major problems for real-time systems such as WALRUS. For example, some mathematical calculations can take several seconds to perform. If the result of that calculation is needed immediately by some future program instruction, thane the calculation cannot be put off. The calculation will need to be performed concurrently as the program goes about its real-time operations.

The QED Forth multitasking operating system allows such concurrent operation. The QED board has only on CPU; this is contained in the Motorola 68HC11 microcontroller chip, With only one CPU, the system software (i.e. the operating system, not the Forth source code) must manage program execution in such a way that there appears to be several CPUs running. The QED does this using a method known as time slicing. Time slicing allows several programs to appear to run simultaneously by forcing each program to share the CPU. For example, assume that there are three programs needing concurrent execution. The operating system first allocates each program its own memory space and environment variables. Each program is then placed on a circular "round robin loop." The first program begins execution and runs for five milliseconds (or whatever time base is preferred). Immediately when the five milliseconds has expired, the second program begins execution and obviously the first one must stop since there is just one CPU. At the end of second five milliseconds, the second program stops and the third one begins. After another five milliseconds, program three stops and execution begins again with program one. This process continues indefinitely; each program taking its turn on the CPU. In this particular example, every program will get to run at least 67 times per second. This causes it to appear as though the QED has three slower CPUs, rather than only a single fast one. Of course there is a tradeoff between speed and program handling. As mentioned before, however, in many real-time systems it is not acceptable to wait for some calculation that could have been done while the CPU was performing some mundane task)such as turning on a motor or waiting for some type of navigation signal).

The WALRUS navigation system has three main programs which have their own memory space and can essentially run independent of each other. They are the obstacle avoidance program, the location sensing program and the motor control program (Note: from now on these programs may be referred to as "processes," which is the correct term for software that is actually running on a CPU.).

The obstacle avoidance program makes use of the LM1812 ultrasonic receiver/transmitter circuit. This circuit when enabled (by setting the PPA pins on the QED board to high), sends out an ultrasonic pulse at approximately 40 KHz using a transducer mounted on the front of the rover. If an obstacle lies within close proximity of the rover, the ultrasonic sound wave will reflect off the object and be detected by the receiver unit which is also mounted on the rover. The wave form is read from the QED A/D ports into a free place in memory. It is the responsibility of the obstacle avoidance program to determine if the signal coming from the transducer is actually a valid reflected ultrasound wave or not. This program makes use of a built-in Forth command called "FFT," which performs a fast fourier transform on a given time domain analog signal. Once the FFT calculation has been performed (requiring 600 milliseconds for a 64 point transform), the program examines the 40 kilo-Hertz discrete Fourier component and its amplitude. If the amplitude is significantly higher than the noise floor, the program will acknowledge the signal as a reflection wave. Having calculated the time between enabling the LM1812 and receiving the reflected wave, the obstacle's distance can then be calculated. When an obstacle is encountered and its distance is found, this program must then place this newly discovered obstacle onto the internal "terrain map." The terrain map is actually a 40 X 40 matrix that represents the 10 X 10 meters box that will be the rover's real-world environment. This breaks the 10 X 10 meter box up into 1600 smaller squares, where each matrix value represents a square; 0 for an open space; 1 for an impediment. Finding the correct spot on the matrix grid to record the impediment requires that WALRUS know its bearing. The location sensing program (discussed later) will be continuously updating the rover's position on the map and also dynamically updating the direction vector. The obstacle avoidance program will check the direction vector, and using trigonometric math functions, will determine where on the X-Y grid that the obstacle exists. The motor control program (discussed later) will then cause the rover to turn if it detects a nearby obstacle on the terrain map matrix.

The second program running on the QED board is used chiefly to determine the rover's bearing and present location at any given point in time. Other processes are always needing current information about the rover's position and direction; another reason why multitasking operating system is very beneficial. The location and bearing calculation is twofold: a magnetic compass is read for an almost instantaneous direction reading and a slower (but much more accurate) calculation of present position is determined from the optical beacon hardware. The location sensing program uses the compass because it is very quick to give a bearing. However minor rover distance calculation errors due to wheel slippage and/or stray magnetic fields can propagate directional errors very quickly. It is therefore very necessary to update the rover's location and direction using the optical beacon system on a frequent basis. WALRUS can even depend solely on the beacon system, however there will be cases where the rover's speed will be sacrificed.

A reading from the compass is simply the process of reading a voltage from 12-bit analog ports two and four and then scaling the readings to the appropriate value in radians (all trigonometric functions use radians). The beacon positioning program routine is somewhat more complicated. First the location sensing program waits for an audio beep signal on 12-bit analog port seven. This tells the program that rotating beacon number one has started its sweep and is now at zero degrees. The program starts a stopwatch timer at this point and waits for a second beep on port seven. Note also that beeps are not only detected, but their frequency is checked; frequencies other than the beacon frequencies are ignored. When the second beep is detected, the timer is stopped and an elapsed time between beeps is calculated. The angular velocity multiplied by this elapsed time gives the first radian angle used to triangulate the rover's position. This same process is completed for beacon number two, giving a second triangulation angle. Using the angle information and one known distance, a law of sines calculation can then determine the rover's X-Y coordinates and a polar representation of its bearing (refer to figure 11).

Each time a bearing and location are determined (which unfortunately takes about six seconds), the global variable "current position" is updated. This variable can then be accessed by any of the other two programs at any time. This location calculation, although slower than the compass reading, is usually accurate to within one percent, given that the rotating beacon speeds are well regulated. The direction and destination vectors are also updated by this program every time the current position is updated.

The motor control program determines what direction the rover should travel in based on information obtained by the obstacle avoidance and location sensing programs. This program has direct control of analog output one through four, which are connected to the motor control circuit of figure 12. The motor control program by default places three volts on ports one and three, while grounding ports two and four. This simply causes the rover to move in a forward motion. The direction vector is constantly checked against the destination vector, and whenever the two differ, a decision must be made as to how to change the rover's resent course. In the simplest case, this program activates a left turn if the difference in direction angles is negative and a right turn if the difference is positive. Most of the time however, obstacles prevent such a simple maneuver.

Several times per second, the motor control program checks the variable "current position" against the numeric values in the "terrain map." If matrix values which represent a close proximity to the rover's present position are equal to 1.0, this indicates that obstacles are nearby. If the obstacle on the terrain map is in direct line with the direction vector, the rover must turn. By default, the rover will turn left unless turning right brings the directing and destination vectors closer together and doing so would not place the rover in the path of yet another obstacle. This program also continually checks the two axis tilt sensor in order to prevent the rover from attempting to navigate extremely steep inclines. The motor reverse mode is available for backing out of situations such as collisions with obstacles or discovering that the rover is attempting to climb an unusually steep incline. When the "current position" and "destination position" variables are separated by a distance of less than three centimeters, the motors turn off.

There are a series of software utilities designed to help calibrate and diagnose problems with certain devices interfaced with the QED board. These utilities are activated by connecting the keyboard/display unit to the WALRUS 25-pin D-sub connector on the main housing . If any key is pressed within five seconds of turning on the rover, the utilities menu will appear on the LCD. Inside the utilities mode, all rover processes are made inactive, and the CPU is entirely dedicated to the utilities programs during this time. The menu system works much like the Hewlett-Packard (HP) calculator menus: a list of options will be listed at the bottom of the screen and the buttons directly below are pressed to select the corresponding option. When the utilities mode is first entered, the options "RUN," "CAL" and "TASK" appear at the bottom of the LCD screen. "RUN" will return the rover to its normal active mode. "CAL" is used to calibrate and test specific hardware devices. "TASK" is used to run one of the three main programs (discussed above) by itself.

Selecting the "CAL" option on the utilities menu brings up another menu of four hardware calibration options and one option for updating the destination point.

The "SND" option is used to test the functionality of the ultrasonic receiver. When this option is chosen, and a transducer is connected to analog port 0, the QED will read a wave form when the user presses the ENTER key. After the ENTER key is pressed, an A/D converter will sample the incoming signal for 100 milliseconds. The sampler also requires a 0.25 volt signal in order to trigger a read. After the signal is read, an FFT is performed on the input wave form. The "SST" menu option will allow the user to scan a frequency spectra from DC to 19.2 KHz in 621 Hz increments. When a peak amplitude is determined, the user can press "SET." This will store the value of the ultrasound frequency that the obstacle avoidance program will use from that point on. This frequency can then be changed at any time in the future should there be a need for calibration. Pressing the re SHIFT key will then return the user to the main menu.

The "MOT" option allows a test of the motor control circuit board. The motor test menu will display a blinking cursor pointing to the motor direction option. The UP-DOWN arrows are used to choose a forward, reverse, left of right motor test. When an option is chosen, the QED will enable the appropriate analog output ports and run the motors for five seconds.

The "TILT" option is used for a hardware calibration of the two axis tilt sensor. Selecting the "X-axis" or "Y-axis" option and then pressing ENTER will cause a display of the sensor's angle (in degrees) which is updated ten times per second. This utility requires that the user place the sensor on a known level surface, whereby the sensor null pots can be adjusted until a zero degree reading is shown on the LCD.

The "NAV" option tests the accuracy of the optical beacon system. Selecting this option will cause the QED to wait for a signal on the 12-bit analog port seven. If a one kilo-Hertz signal is emitted from the beacon and detected, the QED will wait for a second signal at the same frequency. When the second signal is detected, an angular reading will appear on the screen. This reading is intended to show the angle swept by the rotating beacon. Several tests should be done in order to determine the accuracy of the beacon rotation speed. It should also be noted that the beacon detector need not necessarily respond only to one kilo-Hertz; this constant can easily be modified in the source code (see appendix G).

The "LOC" option allows the user to specify the rover's starting and destination points inside the 10 X 10 meter box. The arrow keys and the ENTER key are used to select a desired coordinate. The coordinates entered must be between 0.0 and 9.999. These coordinates are stored inside the QED battery-backed RAM, and therefore do not need to be entered every time the rover is powered up.

The following discussion concerns the global location electronics. The beacons will consist of two light sensing arrays (figure 11). The test area is ten meters square and the beacons will be placed at the corners on the side opposite the target area (see figure 11). A light beacon and microphone will be onboard the rover. The light sensitive array will be shielded from the light beacon by a rotating hood. A thin slit will be cut into one of the sides of the hood. The initializing diode (see figure 11) is oriented such that as the slit in the hood passes in the perpendicular path of the lower boundary of the test area the light sensing array will be exposed and thus produce a reference signal. The reference signal is an audio pulse emitted from the speaker. When the first signal is detected by the rover the QED board starts a timer. The timer will run until the next audio signal of the same frequency is detected. From this time interval, and knowing the speed of the rotating hood, θ_a can be determined through the relation:

$$\theta = \omega t \qquad \text{EQ-2}$$

θ_b is determined in the same manner a θ_a but with the speaker transmitting at a different frequency. The two frequencies are necessary for the QED board to maintain the distinction between signals. From θ_a and θ_b, θ_1 and θ_2 are easily determined and from these two angles θ_3 is an elementary calculation. The location of the rover can now be determined using the angles θ_1 and θ_2 and the Law of Sine's:

$$\sin(\theta_3)/10\text{m} = \sin(\theta_1)/D_{2r} \qquad \text{EQ-3}$$

$$\sin(\theta_3)/10\text{m} = \sin(\theta_2)/D_{1r} \qquad \text{EQ-4}$$

By solving for D_{1r} and D_{2r} the rover's position can now be determined through a simple sine and cosine relation:

$$-X = D_{1r} \times \cos(\theta_a) \qquad \text{EQ-5}$$

$$-Y = D_{1r} \times \sin(\theta_a) \qquad \text{EQ-6}$$

This calculation is then repeated with θ_b and D_{2r} to check the first calculation for accuracy. It should be noted that the calculation using θ_b and D_{2r} yield X and Y values with respect to the lower left hand corner of the test area grid.

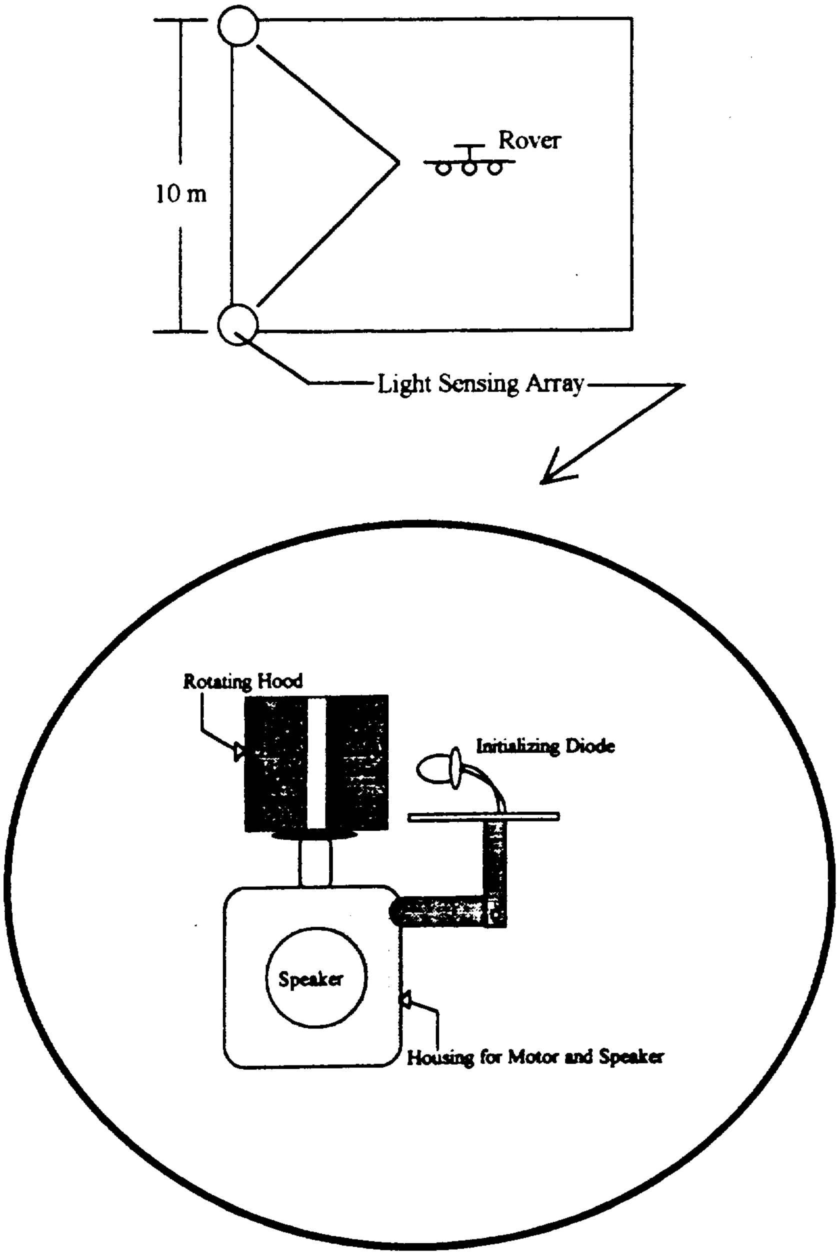


Figure 11. Test Site Layout and Light Sensing Array.

The principle applied in object avoidance through ultrasonics depends on the reflected characteristics of a transmitted signal. Signals when reflected from an object have three basic qualities. The first is the amount of power reflected from an object, the second is if the object is moving, there will be a shift in the frequency and lastly there is a time delay between transmission and reception which allows one to calculate the distance to the object.

Initially, we wanted to sweep the ultrasonic transmitter through a range of frequencies in order to examine the spectral content of the scattered energy. This information could be easily processed by the QED and interpreted into a pattern that could be recognized as different objects. Unfortunately, the piezo transducers for ultrasonics had too small of a bandwidth to make this technique useful. Therefore, we designed the transceiver to transmit and receive on one frequency and examine the time delay and received power. Using the time delay, we are able to find the distance to the object and the received power gives us a rough idea of the size of the object.

National Semiconductor makes an ultrasonic transceiver chip called LM1812, which among its features allows both pulse delays and received powers to be calculated. The circuit was assembled following the manufacturer's design equations and recommendations as found in the specifications in appendix H.

The following discussion concerns the motor control circuitry. Essentially, each motor is connected in a bridge configuration allowing current to flow in either the forward or reverse direction. Figure 12 illustrates the basic principal behind the connections for the motor. A important aspect behind this type of connection requires that before one voltage can be activated, the other must be at zero volts. If both $V_{forward}$ and $V_{reverse}$ are activated, current will be shunted directly to ground and the motor will not respond. Current limiting is provided by resistors that prevent any sudden surges in case such an event should occur. This limiting also keeps the transistors from becoming saturated and drawing too much current when the motor is under heavy torque.

Each motor will be connected identically to the motor pictured in figure 12 and additional circuitry completes the design. Three motors on each side will share the $V_{forward}$ and $V_{reverse}$ lines. Thus, exactly four control lines are needed; $V_{forward}$ and $V_{reverse}$ for the left side and likewise for the right half. The transistors are special high power darlington pair NPN transistors, called TIP121. Each transistor is capable of handling ten amps and 80 volts of power demand. The minimum forward gain for the TIP121 is 1000. With such a large forward gain, only one milli-ampere must be supplied to the base junction in order to achieve a one ampere output current at the emitter.

T vo reasons why an operational amplifier was used to drive the transistors on each side of the rover had to do with interfacing with the QED and drive current demands. The digital to analog port was used to control the motors because it provided 256 analog steps which was ideal for the type of circuitry we designed. However, the range of the analog device was only three volts and a twelve volt signal was required for full speed operation. So the operational amplifier was used to increase the range from three to twelve volts. The QED digital to analog drivers can only supply about 100 micro-amperes, so some current buffering was required to attempt to supply one milli-ampere to each transistor in operation. For just one side of the rover, two transistors are operating per motor at any given time; therefore, a total of six milli-amperes is required per side of the rover. The operational amplifier used (LM324) can supply about 50 milli-amperes maximum and produce the gain required to achieve a twelve volt range. The LM324 is also specially designed to operate with a single power supply, which simplifies the design. A simplified representation of this process is shown in figure 13.

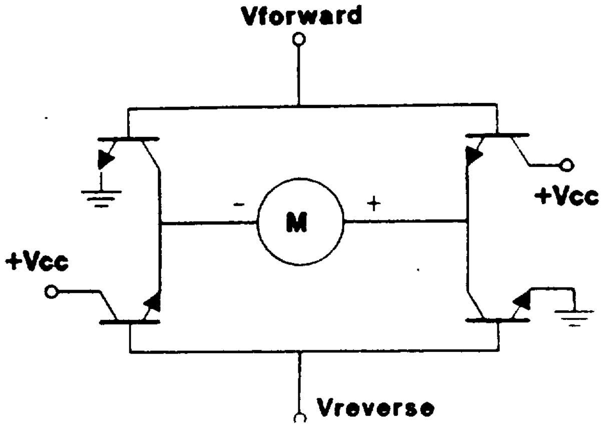


Figure 12. Motor Control.

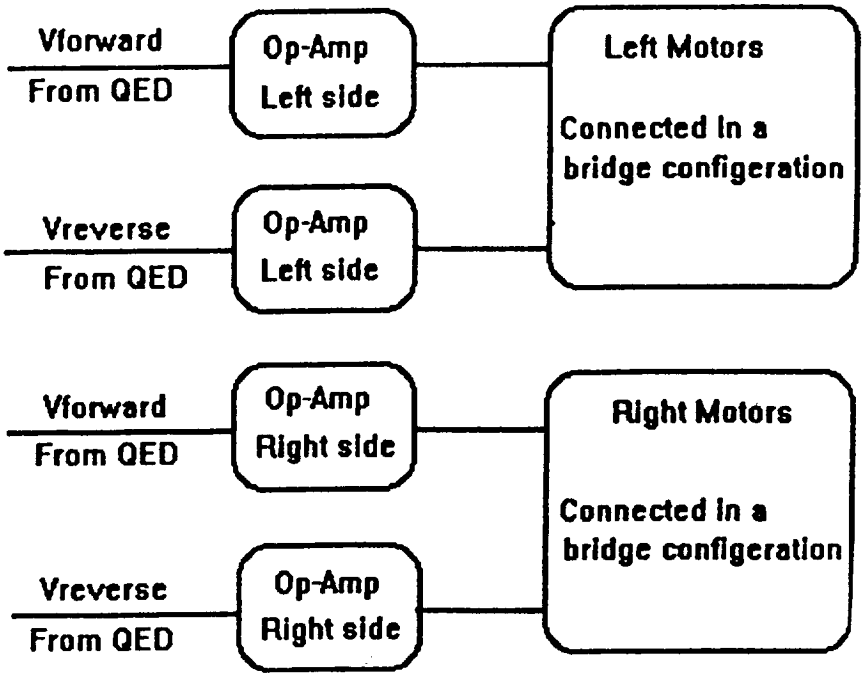


Figure 13. Motor Control Connections.

5.4 Safety.

Safety was addresses by the navigation and control group in our design. One of the major factors for having the housings, aside from protecting the electronics from the outside environment, was to safeguard users from the electronics themselves. The major safety issues that we looked at were the structural stability of the housings and platforms and the insulating of the electronic circuit boards. The housings and platforms all need to be able to withstand a roll-over by the rover, which was considered in the design. The electronic circuit boards were all mounted with insulating plastic snap connectors.

In addition, we decided to incorporate a centrally located on-off power switch to allow for quick shut down of the WALRUS if necessary. In manufacturing the housing and platform components, we made certain that no sharp edges were left that could potentially cut users.

For the radio control, the same safety requirements were necessary. Users need to be protected from moving parts within the gear box. The radio control housing incorporates this concern into its design.

5.5 Test Environment.

Before the assembled rover was sent to Ames Research Center, a test environment similar to Ames Research Center's had to be assembled. The test environment needed to test the rover's obstacle avoidance, traction and navigation system, in order to insure that the rover performed as expected. A frame for this environment has been built in a mechanical engineering laboratory at the University of Idaho. Obstacle avoidance will be tested using a variety of different sized rocks, ranging from two inches to twelve inches. A small hill in the center of the environment will test the capability of the tilt sensor to detect slope. One side will be over 20 degrees, the other side will be less than 20 degrees.

The bottom four inches of the test environment are filled with ordinary soil. The top inch of soil will be sifted and sorted in order to match the soil at Ames Research Center. For details of the top soil, refer to Appendix I.

5.6 Cost Analysis.

Table five shows the final cost breakdown for the navigation and control portion of the WALRUS project.

Clinometer	$85.00
RC Equipment	$180.00
Miscellaneous	$40.00
Sub Total	$305.00
Electronics	$200.00
Total	**$505.00**

Table 5. Cost Breakdown for Navigation and Control System.

6. POWER SYSTEM DESIGN

6.1 Customer Requirements.

The following are customer requirements that the power system must meet:

1. Provide one hour of run time at full payload per charge.
2. Maintain power to rover and on-board electronics.
3. Recharge six five-battery (30 Ni-Cd rechargable batteries) carousels simaltaneously.
4. Recharge all batteries in six hours or less.
5. Recharge batteries while still in rover.
6. Avoid overcharging batteries (automatic shut-off at 1.3 volts per C-cell).
7. Power system must be easily operated by one person (small in size and less than 15 lbs and four steps to accomplish recharging).
8. Power system must be safe to operator and rover's components (maximum temperature of any component is 60 degrees Celcius and all electronics must be grounded).
9. Cost less than $300.00.
10. Manufacture at Universtiy of Idaho facilities.

6.2 Literature Search.

Batteries were researched to ensure that the preliminary specification of the battery type was the best choice. The most important consideration for the selection of the batteries was the size consideration. The only battery sizes acceptable were C-cell size or smaller, unless an irregular shaped battery could be found that would fit in the space of the carousel. The preliminary batteries selected were Gates ULTRAMAX fast charge nickel cadmium (Ni-Cd), C-cell batteries. Terry Fong at NASA recommended that Gel cell batteries be considered also. A literature search revealed literally hundreds of types of Gel-Cell batteries. The most common type encountered was the lead-acid battery with a gelled electrolyte. These batteries were researched and it was discovered that they were not readily available in C-cell size. The smallest common size available was D-cell size. Even in the larger battery size the capacity rating of the lead-acid type was less than the Ni-Cds. Due to the cost of high tech, state of the art battery technology, our investigation of battery alternatives was limited. Another primary factor influencing the selection of batteries was the fact that the power control board and the electronics all rely on 12 volts. This gives the advantage to the Ni-Cd batteries because the combination of ten cells make up 12 volts. Therefore the 30 batteries carried in the rover provide 3 sets of 12 Volt power sources.

A battery charging book from Gates and a thorough search of vendor catalogs was performed to see what types of charger technology is available. A technical manual from Gates [8] and an interview with V. M. Stanley [9] indicated a wide variety of battery charging methods. Three of the most common methods are time control, temperature sensing and voltage sensing. The first method simply charges the batteries for a predetermined amount of time. The second method monitors the temperature to determine when batteries are fully charged. When the batteries are fully charged, the battery temperature will begin to increase rapidly. The third method monitors the voltage across the battery. When the voltage rises to the predetermined voltage the batteries are fully charged. To charge a battery, a current needs to be sent to the batteries. The amount of current determines how fast the batteries will be charged. However, once the batteries are charged, they can only accept a maximum of 250 milli-amperes without causing damage. If the batteries are being charged at more than 250 milli-amperes in order to charge them faster, the current must be lowered to 250 milli-amperes once they are charged.

The operating environment for the rover required the charger to be able to survive in a room temperature, indoor environment. Due to the amount of heat produced by the charger electronics, a method of cooling these components was required.

6.3 Concept Evaluation.

The first decision made was the selection of the batteries. We choose the C-cell CADNICA Ni-Cd batteries from Sanyo because of their high power capacity and the fact that 10 cells could provide 12 volts. The initial Gates batteries were not selected because they were on backorder, and the Sanyo batteries had similar specifications.

The second design decision was whether the batteries were to be charged in or out of the vehicle. Charging the batteries inside the rover posed potential problems with dissipating the heat generated in the batteries. The customer stated that the batteries should be charged in the rover, so a heat transfer analysis was undertaken. To supplement the heat transfer analysis a heat test chamber was made and calibrated. Figure 14 shows a drawing of the test apparatus.

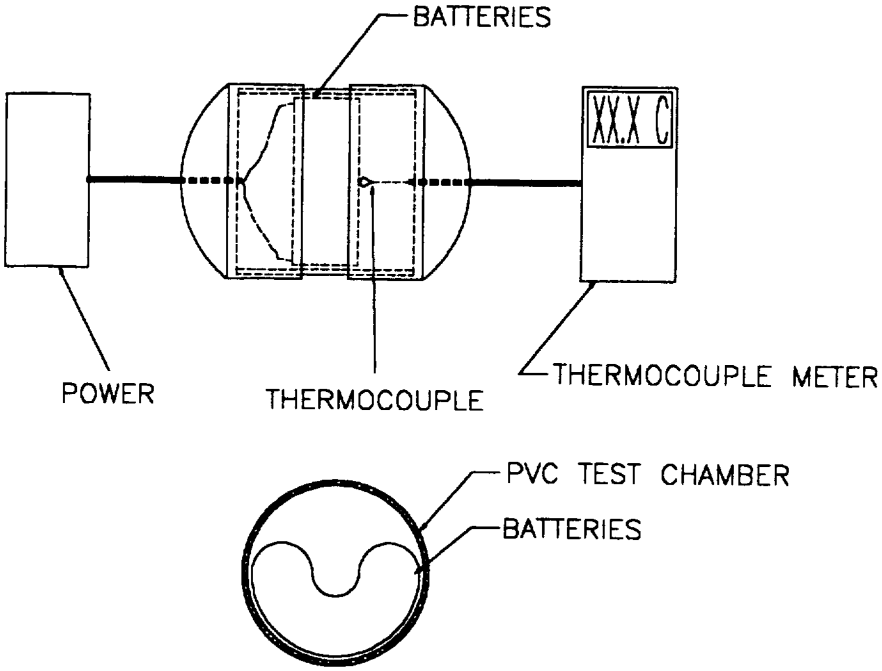


Figure 14. Test Chamber.

The chamber was calibrated using a light bulb at several different wattages. A carousel was then placed into the chamber and recharged. The temperature versus time curve was compared to the calibration curves to determine the heat produced by the charging battery.

The next design decision was to decide between ordering a custom made charger, buying an automotive charger, or designing one ourselves. The first option was to order a custom charger. To do this we contacted over 20 different battery charger manufacturers. The majority of the manufacturers were unable to help us, due to our low production quantities. Therefore, only 3 actual bids were received. The minimum bid received was for $2000. The maximum bid received was $18000. The advantages of this option were that it would be reliable, it would have automatic shutoff, and it would meet safety standards. The disadvantages were that it was very expensive and it could not be delivered by our deadline.

The second option was to purchase an automotive battery charger. The advantages of this option were that it was lightweight, low cost and had low power consumption. The disadvantages of this option were that it did not have an auto shutoff feature, it could not charge all the batteries at once and would be difficult to use.

The third option was to design and build the charger ourselves in the Moscow area. The advantages of this option were that it had the auto shutoff feature, it could charge all the batteries at once and it was very easy to use. The disadvantages of this option is the ease of maintenance and possible reliability problems with the system.

A decision matrix (see Table 6) was used to aid the selection of the optimal design. The decision matrix was constructed by listing the competing designs across the top and the specifications along the left side of the matrix. A weighting system was then established in order to rank the importance of each requirement. The competing designs were assigned numbers that represent their ability to meet the requirements (a scale of 1-5 was used with 5 being the best). Finally, the numbers representing their ability to meet the respective requirements were multiplied by the weighting factor and totaled at the bottom, with the design having the most points being the best design.

DESIGN ALTERNATIVES		UI Designed	Custom Made	Automotive
Requirements	Weighting			
Weight	7	4	3	5
Cost	12	3	1	4
Reliability	9	3	5	3
Manufacturablity	10	4	5	5
Auto Shutoff	12	5	5	2
Maintenance	8	3	4	4
Low Power Use	3	4	4	5
Size	7	4	3	4
Ease of Use	8	5	4	2
Delivery Time	12	4	1	4
Safety	12	4	5	3
TOTAL	100	391	357	359

Table 6. Decision Matrix.

The option that we chose, using the decision matrix, was to design, manufacture, and assemble the system at the University of Idaho. Also, after decision evaluation, we chose a pre-built container for the charger electronics. This decision was made because we found a housing that would satisfy all our needs, including delivery time. This allowed us to concentrate on the inside of the charger and the electronics rather than the housing. This option also appeared more professional and reduced manufacture and assembly time.

Using the information discussed previously in the literature research, we had to consider which method to use to charge the batteries. The easiest method to charge the batteries is to send a constant current of no more than 250 milli-amperes to the batteries. While no charge cut-off circuitry is needed for this design, a current of 250 milli-amperes will charge the batteries in only 16-20 hours, which does not meet the customer requirements. To charge the batteries in six hours, a current of approximately one ampere is needed. Once they are charged, the current needs to be lowered to less than 250 milli-amperes. Three of the most common methods of accomplishing this are time control, temperaturesensing and voltage sensing. The first method simply charges the batteries for a predetermined amount of time at a given current. This method is only useful if it is known how discharged the batteries are since it takes longer to charge a fully discharged battery than a partially discharged battery. Also, with the rover, since some batteries are used for the electronics and some for the motors (which draw much more current), the batteries will be at different levels of discharge. The second method monitors the temperature to determine when the batteries are fully charged. This uses the fact that the batteries heat up while thay are charging. When the batteries are fully charged, they are theoretically at a certain temperature. This method, however, is not suitable for the rover since the charging will be done in the wheels where the temperature will be different than if the batteries were outside of the rover. The third method monitors the voltage across the battery. When the voltage rises to the predetermined voltage the batteries are fully charged and the current must be lowered.

6.4 Detailed Design.

The final charging system for the rover is enclosed in a rectangular, 0.05 inch thick, aluminum housing purchased from Newark Electronics (see Figure 15). There are six LED (light emitting diode) display lights and a power switch mounted on the front. The rover recharging cord comes out the front panel, and the power cord comes out the rear of the box. A carrying handle is located on the top of the charger. It weighs 9 pounds, measures 8" X 8" X 10.125" and stands on four short, plastic feet. The charger has louvers on the side and a 150 CFM box fan mounted in the rear of the housing. The louvers and fan provide the cooling needed to keep the electronics from getting damaged.

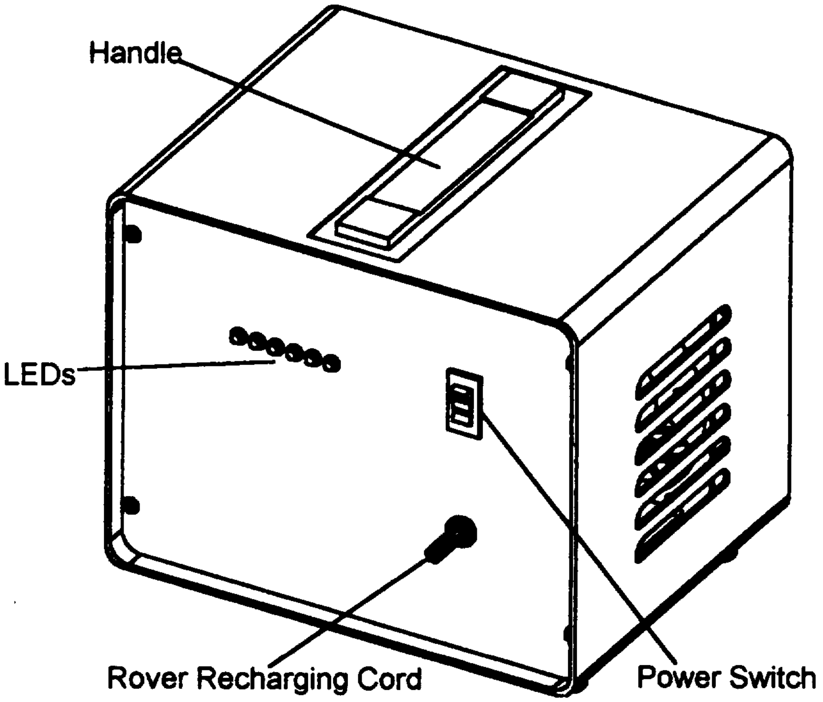


Figure 15. Overall Schematic of Charger.

A detailed heat transfer analysis of the batteries and charger electronics was performed (see appendix J). A heat transfer model for natural convection between two concentric cylinders was used from De Witt [10] to approximate the heat transferred inside the wheel enclosure. The model included volume heating inside the batteries to simulate the power lost due to the internal resistance of the batteries. With the heat rate and the ambient temperature as knowns, the battery surface temperature was calculated using the equation:

$$q = \frac{2\pi \cdot k_{eff}}{\ln \frac{D_o}{D_i}} \cdot (T_i - T_o)$$

q = Heat rate

k_{eff} = Effective thermal conductivity for air inside enclosure EQ-7

$D_{o,i}$ = Diameters of outer/inner cylinders

$T_{o,i}$ = Temperature of outer/inner cylinders

For the charger electronics a heat transfer correlation for a flat plate with a heat flux was used yielding the equation

$$\frac{q''}{T_e - T_\infty} = \frac{k}{L} \cdot 0.0308 \cdot \mathrm{Re}^{\frac{4}{5}} \cdot \mathrm{Pr}^{\frac{1}{3}}$$

q'' = Heat flux

$T_{e,\infty}$ = Electronics , Ambient Temperatures

k = Thermal conductivity of air

L = Length of plate

Re = Reynolds number

Pr = Prandtl number

EQ-8

We found that the batteries would reach a maximum temperature of 20.9 degrees Celsius using this model. This is well within the safe temperature range, so there should be no problem with the batteries overheating while they are being charged. For the charger electronics, a maximum surface temperature of 60 degrees Celsius was chosen, and a fan with a discharge rating of approximately 140 cubic feet per minute is required.

To ensure that our battery charger design was safe, we evaluated it using the Failure Mode and Effects Analysis (FMEA). This analysis showed that the most serious safety hazards were blocking the flow of cooling air for the electronics, electric shock, and water damage. To keep the airflow from being blocked we specified in the instructions not to put the charger in close proximity to other objects that can obstruct the airflow through the louvers and fan. A guard was also placed over the fan to protect it from foreign objects. To safeguard against electrical shock the circuit board was connected to the charger base using insulating plastic connectors, and the circuit was designed such that it is properly grounded. In order to safeguard against water damage we put cautions and warning messages into the charger procedure that state to avoid contact with liquids. It is also recommended to tape the cords to the floor (if the location is permanent) to prevent a tripping hazard. Appendix K contains the detailed safety analysis and results.

The rear electrical cord plugs into a standard three-prong, 120 Volt electrical wall outlet. The front electrical cord connects to the rover using a twelve pin connector. The front panel display consists of a power switch that illuminates when the switch is turned on. Also, there are six small LED that are lit initially and go out to signify when each carousel has reached full charge.

The charger housing contains a circuit board mounted to the bottom with plastic snap connectors, a transformer fastened with bolts and nuts, a fan attached with bolts into threaded holes, and a capacitor fastened to the fan. The main manufacturing process used was modifying the purchased housing to suit our needs. This was done by drilling holes into the housing to mount the circuit board, fan, LEDs, and transformer. We also machined openings for the power switch and the fan. Also, the circuit board was cut to size, using a band saw, and four mounting holes were drilled. Figure 16 shows the location of the internal components. An aluminum plate was machined to size and drilled for use as a cover for the opening in the back of the charger next to the fan.

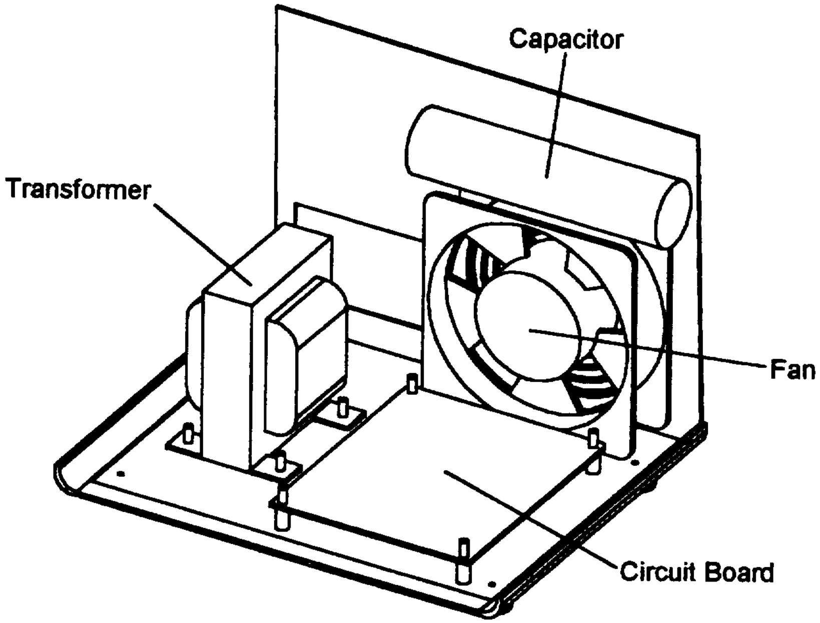


Figure 16. Internal Components.

The voltage transformer is located on one side of the charger; it's function is to convert the standard 120 volts from the wall outlet to the 12 volts required by the charger circuit. The circuit board is mounted in front of the fan, in order to aid the cooling of the electronics. The function of the circuit is to filter and convert the alternating signal from the wall outlet to the direct current form needed to charge the batteries. It also controls which carousels are charged and at what rate. Figure 17 shows the circuit flow chart.

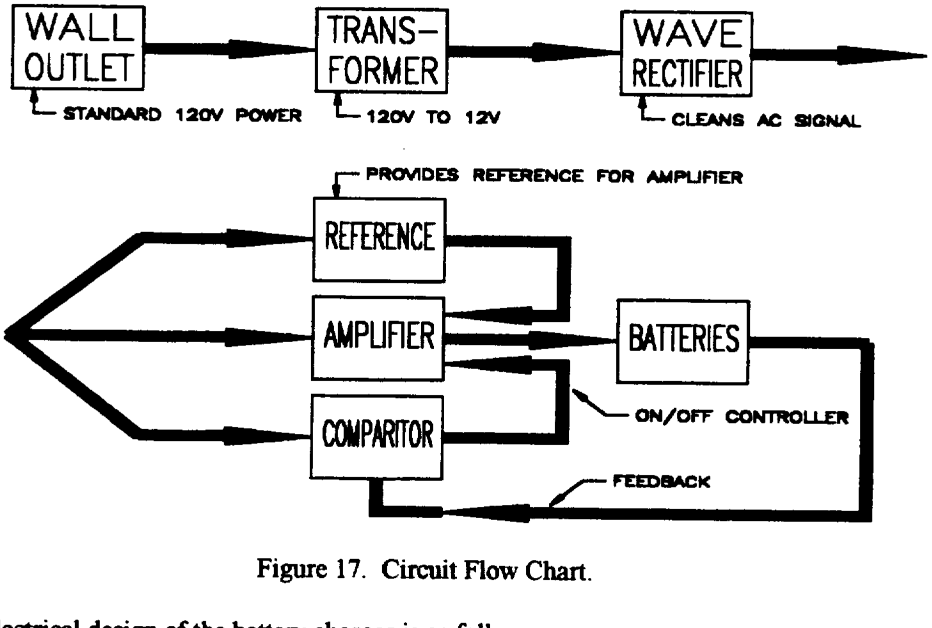


Figure 17. Circuit Flow Chart.

The basic electrical design of the battery charger is as follows:

Power to the charger comes from a cord plugged into a 120 volt outlet. This 120 volt signal is then sent through an On/Off switch and then through a transformer. The transformer lowers the voltage to 12 V_{rms}.

To provide a useful signal, the 12 V_{rms} output of the transformer is sent through a full wave rectifier. While a half wave rectifier also works, a full wave rectifier is much more efficient. Connected to the rectifier is a 11000 micro-Farad capacitor to further clean up the signal (this capacitor was used because it was donated).

The comparator takes a reference voltage set by a voltage divider and compares the reference to the current battery voltage. This is accomplished using an LM324 quad operational amplifier and external resistors. The output of the comparator is a direct relationship to the difference in the voltages multiplied by a gain set by the external resistors. These resistors are set to keep the operational amplifier railed until the difference in the voltages is very small. When the operational amplifier in not railed, the output is linear until the difference is essentially zero. The slope of the output voltage of the operational amplifier when not railed and until it reaches zero is set bby a single resistor. Changing this value has an effect on how big the difference in voltages has to be to keep the operational amplifier railed.

Six LEDs are connected to the outputs of the comparators. When the differences between the voltages are large and the outputs of the comparators are high, the LEDs are lit. When the differences aare small, the LEDs turn off. This provides a visual indication to the user of when all of the batteries are charged.

The amplifier boosts the output of the comparator with the use of a TIP121 darlington. The TIP121 darlington was chosen because fo its high gain (1000) and its ability to handle 50 volts at ten amperes. If the operational amplifier is railed, the output of the amplifier is at its maximum, which is set to approximately one ampere. As the output of the operational amplifier falls, so does the output of the amplifier.

The filtered and converted electrical signals are then sent through the front charger cord to the rover. A twelve pin connector is used to connect the charger cord to the rover. The rover wiring goes from the connector to the 1.2 volt batteries (which are connected in series in each carousel) in the wheels of the rover.

The process used to charge the rover batteries with this charger is quite simple. The charger box is plugged into a standard wall outlet. The charger can be moved, if required, using its handle. Next, the power cord on the rover that runs from the batteries to the QED box is unplugged. The charger cord is now connected to the power cord that was just unplugged. The power switch is then turned on and the operator can leave the charger. The charging process takes six hours and the operator can check to see if the batteries are charged by noting if all six LED lights are off. Each light is keyed to a specific carousel, allowing the operator to determine if there is a charging problem with a carousel. The operator does not have to worry about leaving the charger on too long and therefore overcharging the batteries, because the circuit board has automatic overcharge shut-off capabilities. Finally, the charger cords are unplugged, and the rover is ready to run. A complete charging procedure is included in appendix L.

7. RESULTS

The results of this project are best summarized through the use of pictures. Figure 18 shows some of the details of the wheel design, including the battery carousels and motors. Figure 19 shows the interface of the frame, wheels and mounting platforms. Figure 20 shows the completed rover, including the two electronics housings. Figure 21 shows the battery charger.

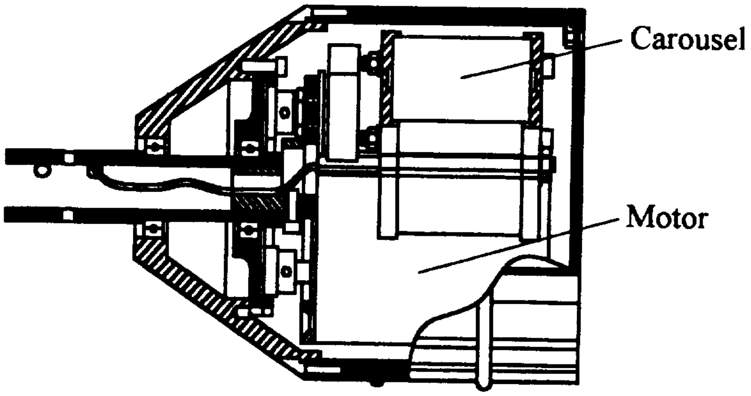


Figure 18. Section View of Wheel Revealing Interior.

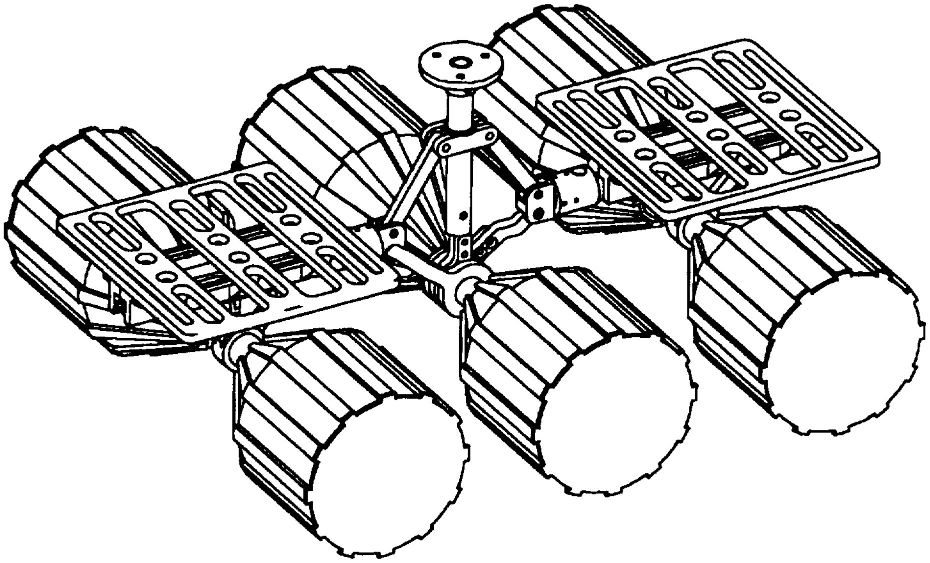


Figure 19. Frame and Wheel Assembly Showing Camera Shaft and Mounting Platform.

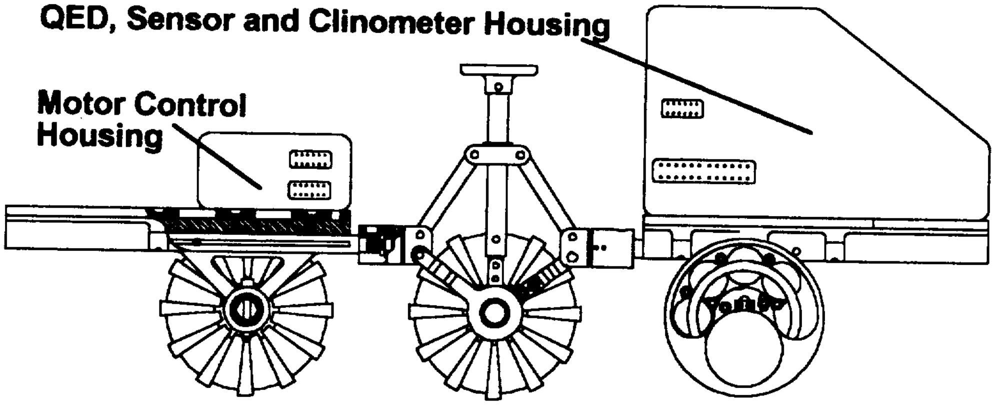


Figure 20. Completed Rover Including Housings.

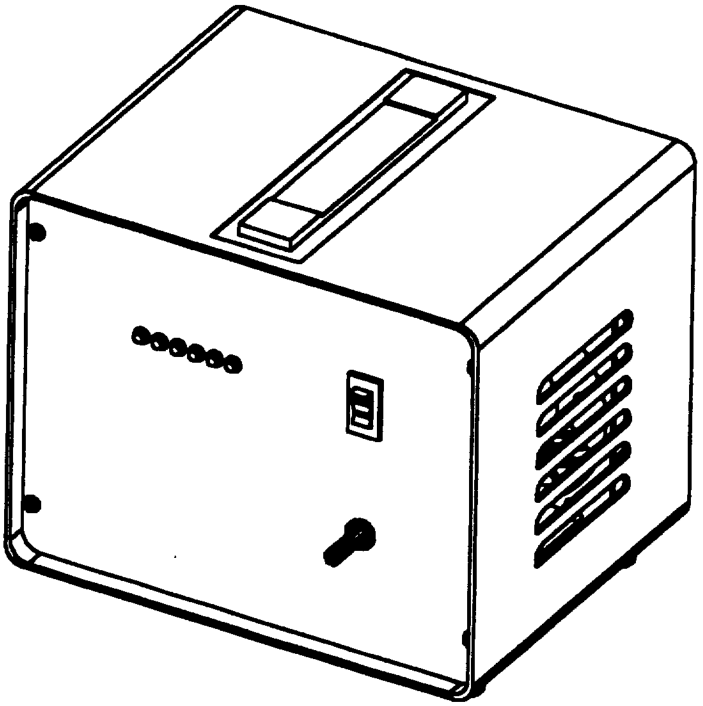


Figure 21. Charger for the Rover's Batteries.

The final specifications of the rover and its systems are best given using a table. Table seven gives these specifications.

Length	19.7 in.
Width	14.5 in.
Height	8.95 in.
Weight	25 lb.
Center of Gravity	2.5 in. from ground
Payload Weight	5 lb.
Payload Capacity	20 lb.
Run Time	1 hr. at full payload
Charge Time	6 hr.
Turning Radius	0 in.
Slope Detectability	20 degrees
Climbing Ability	30 degrees
Positional Accuracy	± 1.2 inches
Maximum Speed	1.5 ft./sec.
U of I Cost	$5000.00

Table 7. Final Specifications of Rover.

8. CONCLUSIONS

This design of an autonomous land rover fulfills our customer's requirements. These requirements were met through the use of systematic design methods and economic tradeoffs. Due to time and budget constraints the rover wheels are still in the process of being machined on NASA's computer numerical control lathes. All other parts are completed. The rover will be tested and delivered to Ames Research Center by the end of the summer of 1993.

9. REFERENCES

1. The Enabler: A Concept for a Lunar Work Vehicle, James A. Austin, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia, 1992.

2. Lunar Construction Utility Vehicle, NASA/USRA Summer Report, Old Dominum University, Norfolk, Virginina, 1989.

3. Autonomous Mobile Robots, A. Meystel, World Scientific, Singapore, 1991.

4. How to Build a Computer Controlled Robot, Tod Loofbourrow, Hayden Book Company, Inc., Rochelle Park, New Jersey, 1978.

5. Autonomous Mobile Robots: Control, Planning and Architecture, Volume II, S.S. Iyengar, A. Elfes, Computer Society Press, Los Alamitos, California, 1991.

6. Dr. Mike Anderson, private conversation on April 4, 1993.

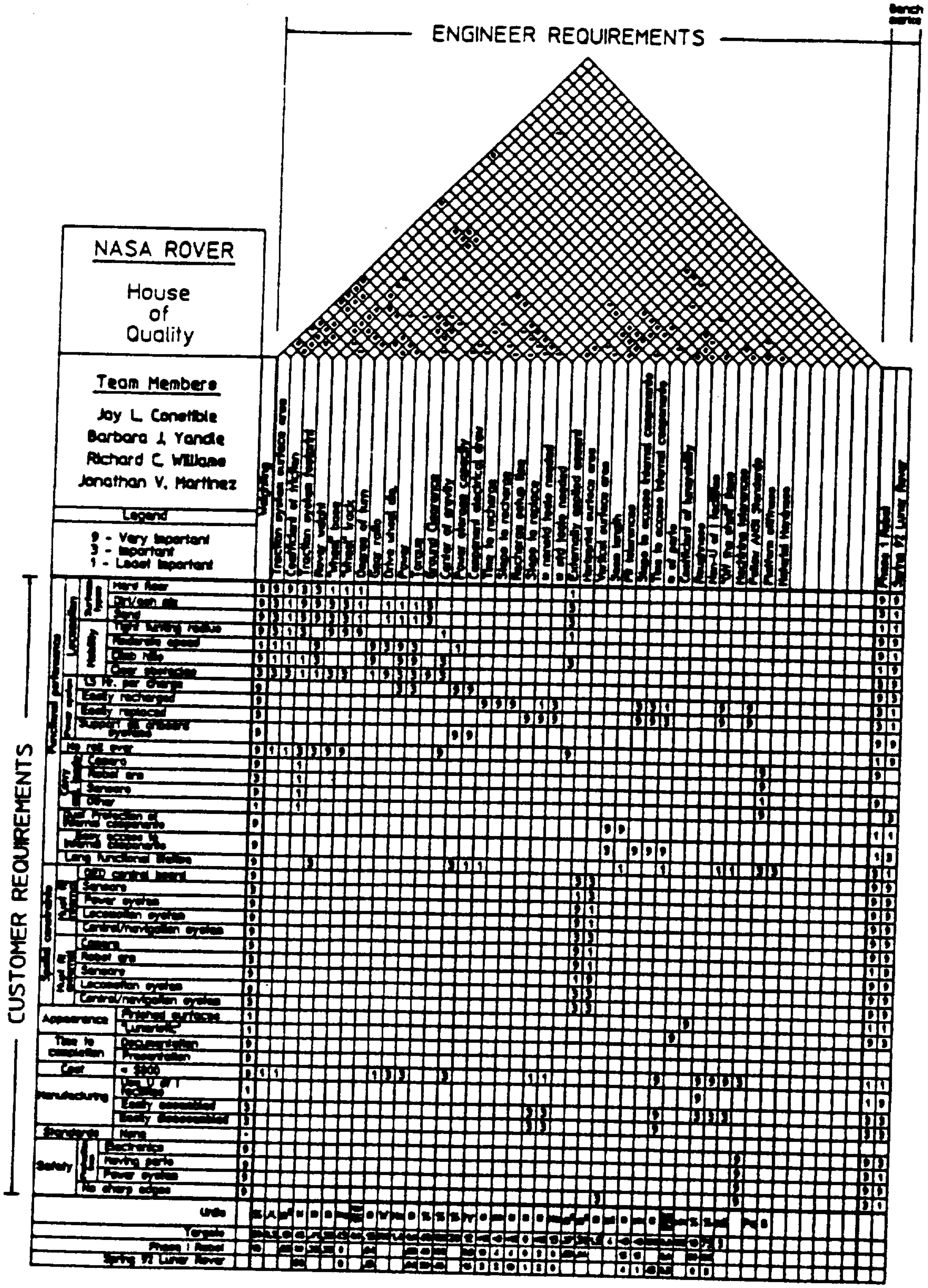
7. Field and Wave Electromagnetics, Second Edition, Adison-Wesley Publishing Company, Inc., 1990.

8. Sealed Rechargeable Battery Applicatoin Manual, Gates Energy Products.

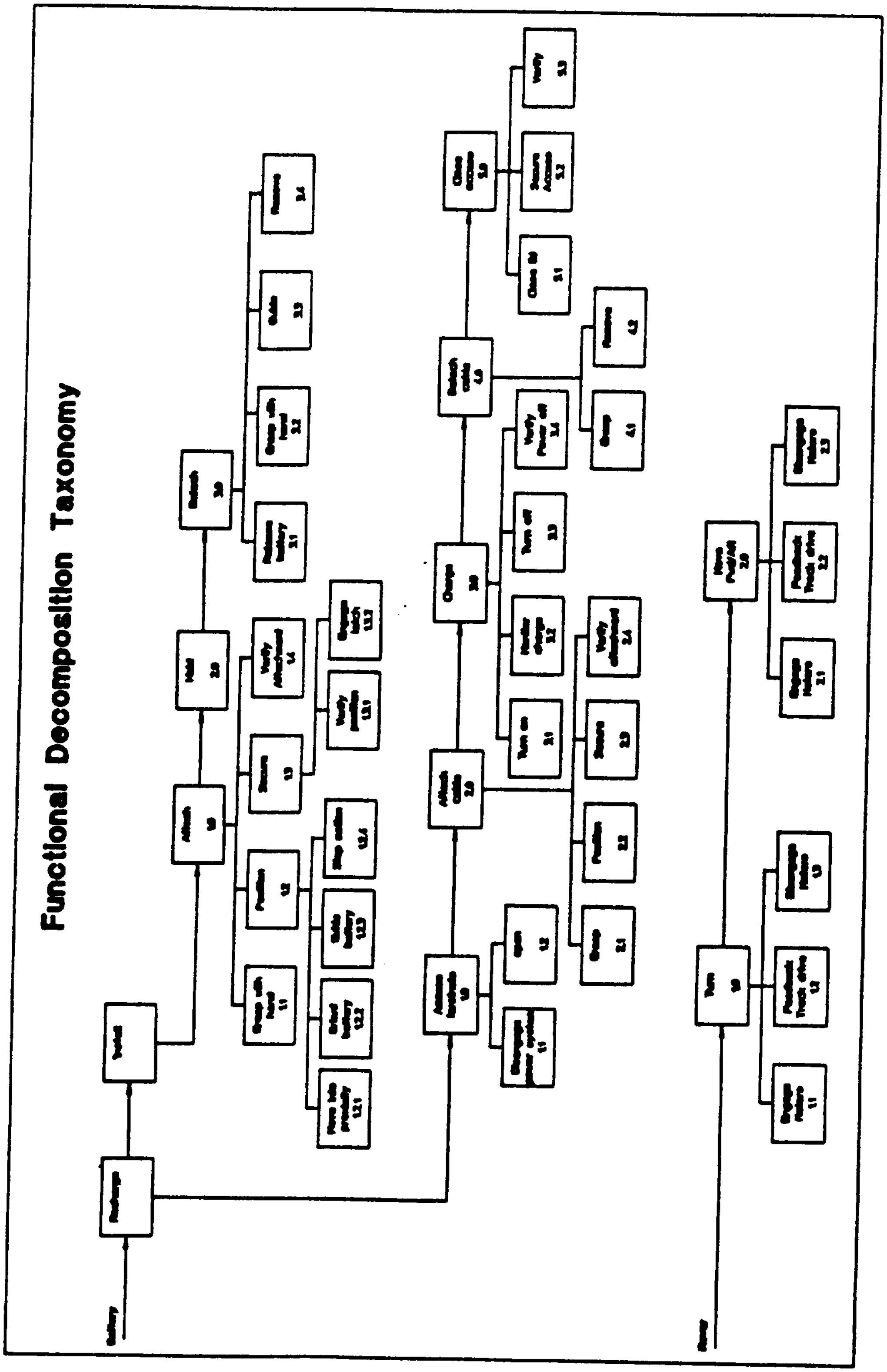
9. V.M. Stanley, private conversation on March 17, 1993.

10. Fundamentals of Heat and Mass Transfer, 3rd ed., John Wiley & Sons, New York, 1990.

APPENDIX A - HOUSE OF QUALITY



APPENDIX B - FUNCTIONAL DECOMPOSITION TAXONOMY



APPENDIX C - DECISION MATRIX

NASA ROVER

DECISION MATRIX

Legend

S = Same
+ = Better
- = Not as Well

CUSTOMER REQIREMENTS	WEIGHT	RICHES ROUGH RYDER	LAWN MOWER	S.M.R.	J's THREE TRACK	J's TWO TRACK	J's FOUR WHEEL	LARRY's THREE TRACK	RATLER
Hard Floor	2	S	S	-	S	S	S	S	DATUM
Dirt/Ash	3	+	-	+	+	+	S	+	
Sand	4	+	-	+	+	+	S	+	
Tight Turning Radius	4	S	S	S	S	S	-	-	
Moderate Speed	1	-	S	S	-	-	S	-	
Climb Hills	4	+	-	+	+	+	S	+	
Clear Obstacles	2	-	-	+	+	-	-	S	
No Roll Over	4	+	-	+	+	+	S	+	
External Load Carrying	3	+	-	+	+	+	S	+	
Internal Load Carrying	4	+	+	+	+	+	+	+	
Dust Protection	4	+	-	+	-	-	S	-	
Easy Access	4	-	S	S	-	S	S	-	
Long Life	4	-	-	+	-	-	S	-	
Lunaristic	1	+	+	+	-	-	-	-	
Use Uol Facilities	1	S	S	S	S	S	S	S	
Easy to Assemble/Disassemble	2	-	-	S	-	-	-	-	
Exposed Moving Parts	4	S	-	-	-	+	-	-	
Sharp Edges	3	S	+	-	S	S	S	-	
TOTAL +		8	3	10	7	7	1	6	
TOTAL +		6	10	3	7	5	5	8	
TOTAL +		2	-7	7	0	2	-4	-2	
WEIGHTED TOTAL		14	-26	24	4	12	-9	-2	

APPENDIX D - OBSTACLE AVOIDANCE

Obstacle Avoidance Techniques:

Ultrasonics - Both piezoelectric and Polaroid ultrasonic transducers use a combinations of transmitters and receivers. The signal is processed using an oscilloscope and a personal computer.

Pros-	Cons-
-resident expert in Dr. Anderson	-sensor used is usually high voltage
-widely used for this application	-return noise is a problem
-inexpensive	-piezoelectrics have narrow bandwidth - can not see some things
-low power consumption	
-driver chip availability	
-Polaroid's work good for this application	

Infrared - Infrared sensors transmit a light beam in the direction that is to be checked. If the receiver gets a signal back, there is something in front of the sensor.

Pros-	Cons-
-currently in use	-a lot of scattering
-good for flat surfaces	-high power consumption
-fast	-too fast for QED to process

Tactile - Tactile sensors receive sensor input by coming in contact with an obstacle. After contacting an object, the rover must stop, back up and turn away from the obstacle.

Pros-	Cons-
-easy to install	-reactive not proactive
-low power consumption	-can not measure slope
	-low lunarability
	-too sensitive

APPENDIX E - GLOBAL LOCATION

Radio Frequency - This navigation alternative would implement beacons sending a modulated radio frequency and a receiver located on the rover. The receiver would catch signals from the beacons and the QED would determine rover position in the test environment by measuring the power between the sender and the receiver. This power measurement is determined through the use of a simple equation.

Pros-	Cons-
-lower power consumption	-a lot of interference
-theoretical ease	-low repeatability
-limited hardware	-not commonly used
	-not practical

Ultrasonics - Polaroid sensors that are located in post beacons, send signals to a rotating receiver on the rover. Through the use of trigonometric relationships, the position of the rover can be determined.

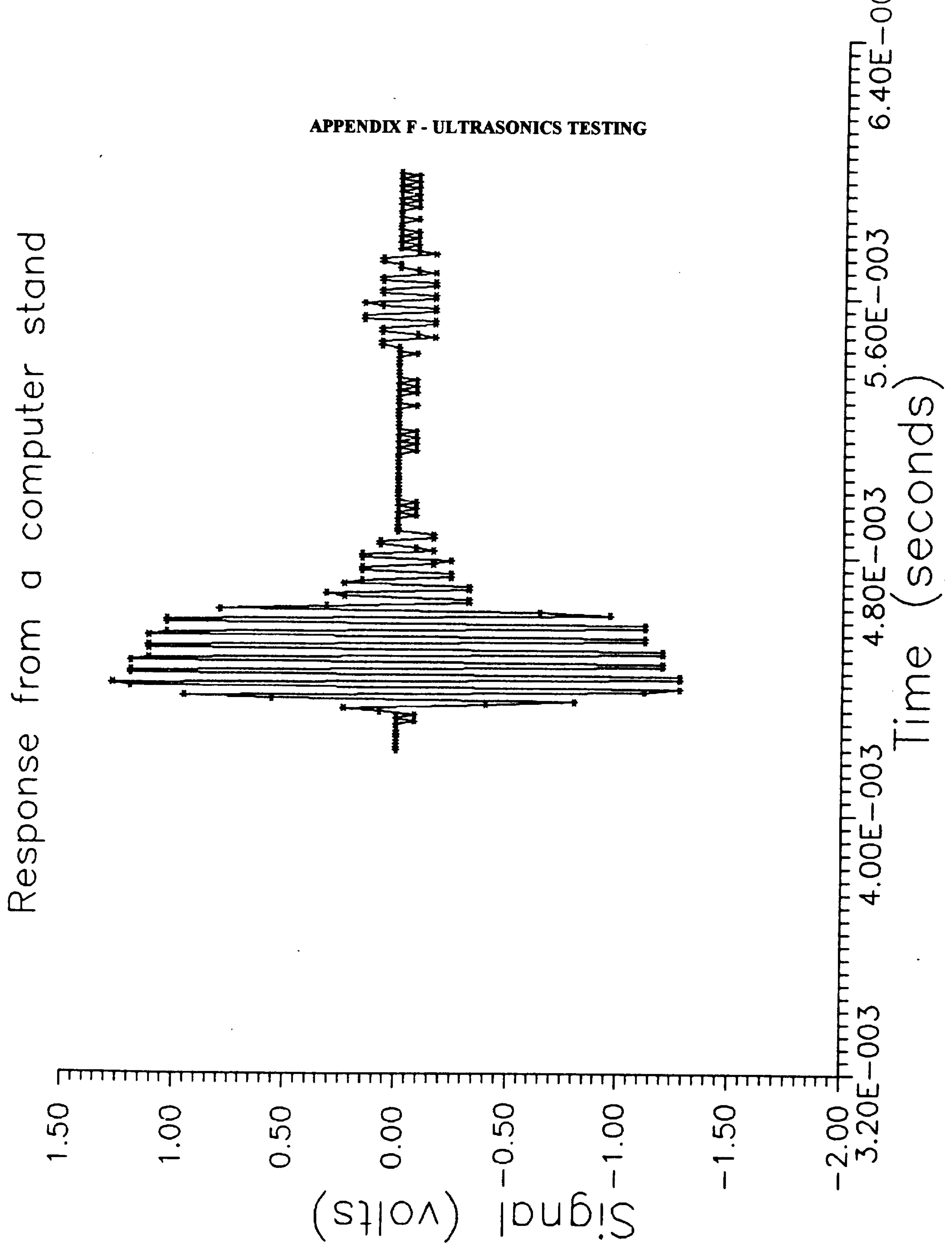
Pros-	Cons-
-in use at Ames Research Center	-too much hardware
-sound is slow enough for QED to process	-resolution is bad
-resident expert in Dr. Anderson	

Optical - Optical systems incorporate transmitting beacons in conjunction with a rotating receiver on the rover. Through the use of time calculations and trigonometry, the position can be calculated.

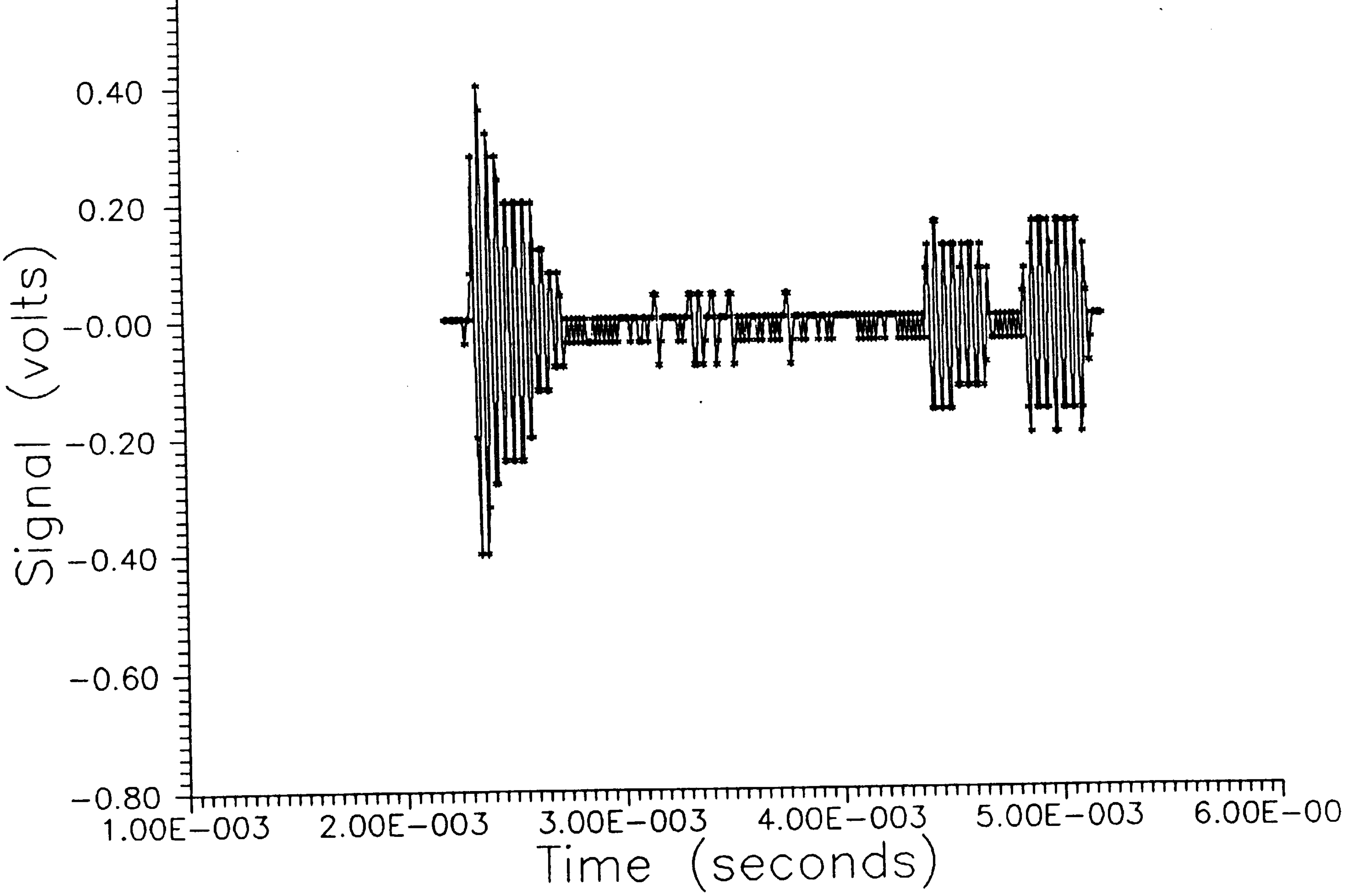
Pros-	Cons-
-simple to install	-rotating receiver
-easy circuits	-safety
-accurate	
-cost	

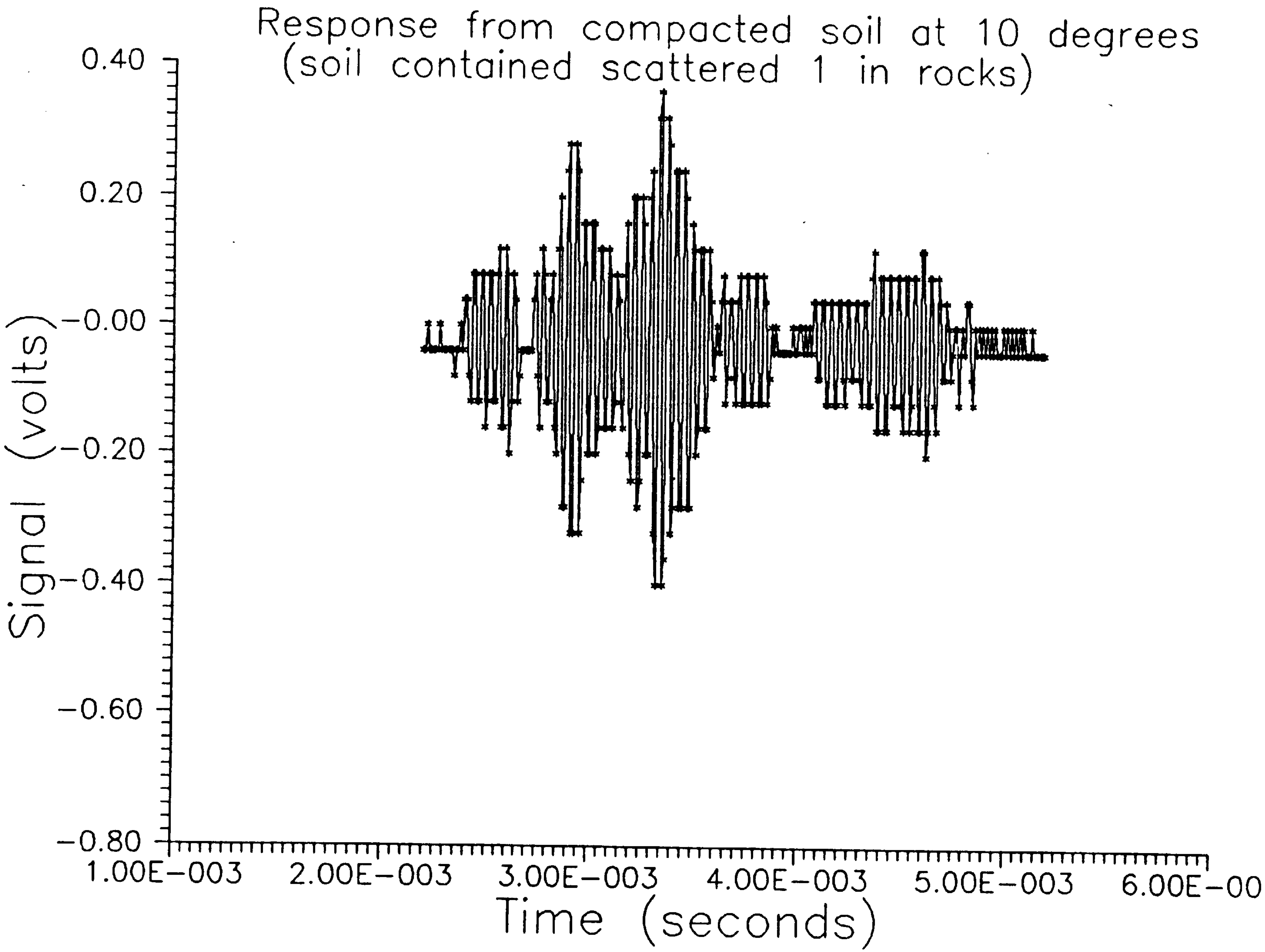
Lasers - A rotating transmitter on the center post of the rover would send out "bursts" to two separate reflectors located in corners of the test environment. The reflector posts reflect the burst back to the rover which measures the time delay between the two posts. From this information, and trigonometric relationships, the QED is able to calculate the rover locations.

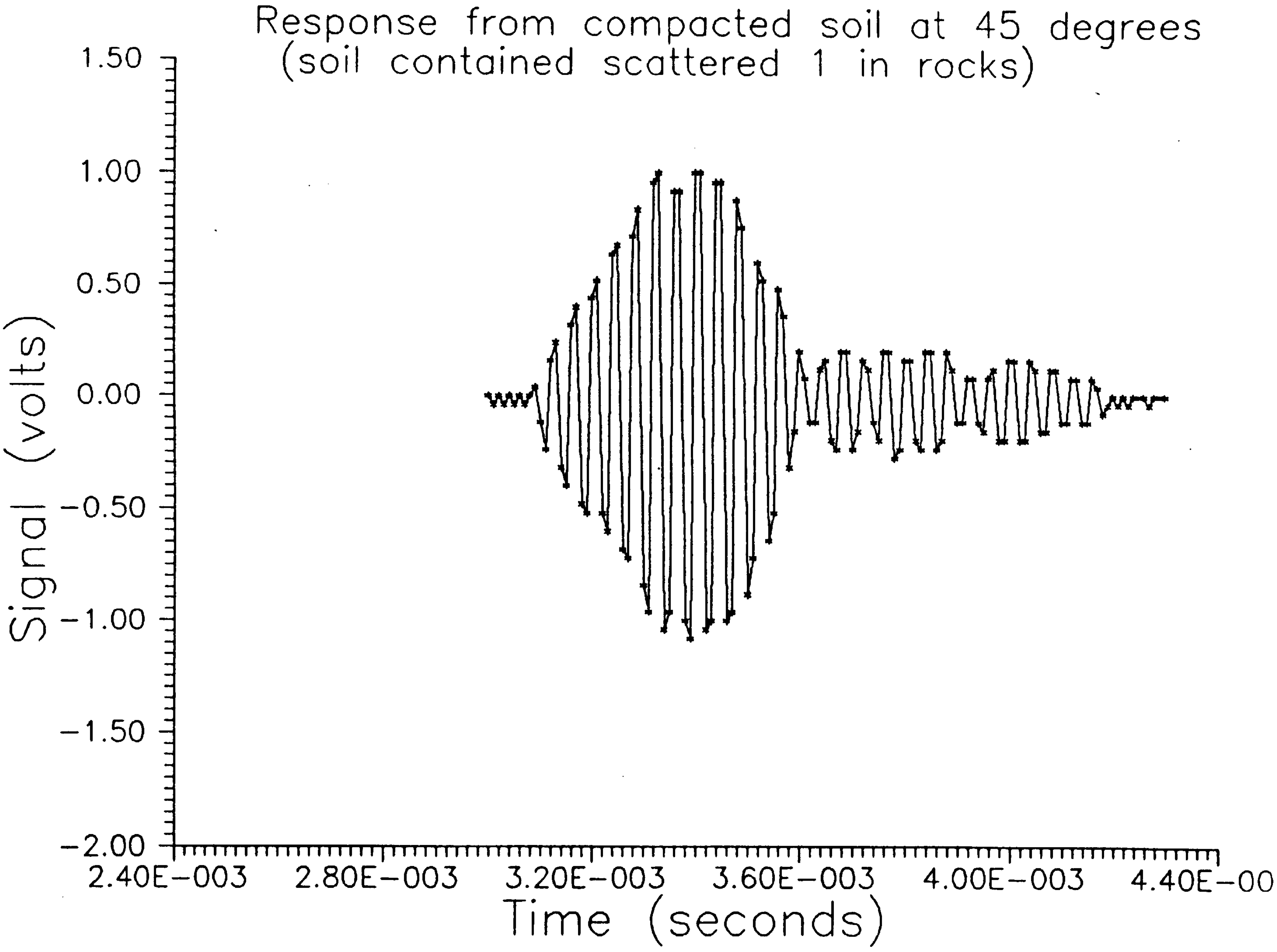
Pros-	Cons-
-very accurate	-high power consumption
-deterrent	-extra equipment on rover
-low interference	-cost
	-safety
	-mechanically difficult


Response from a computer stand
1.50
1.00
0.50
0.00
−0.50
−1.00
−1.50
−2.00
Signal (volts)
3.20E−003
4.00E−003
4.80E−003
5.60E−003
6.40E−00
Time (seconds)

Response from a motorcycle helmet

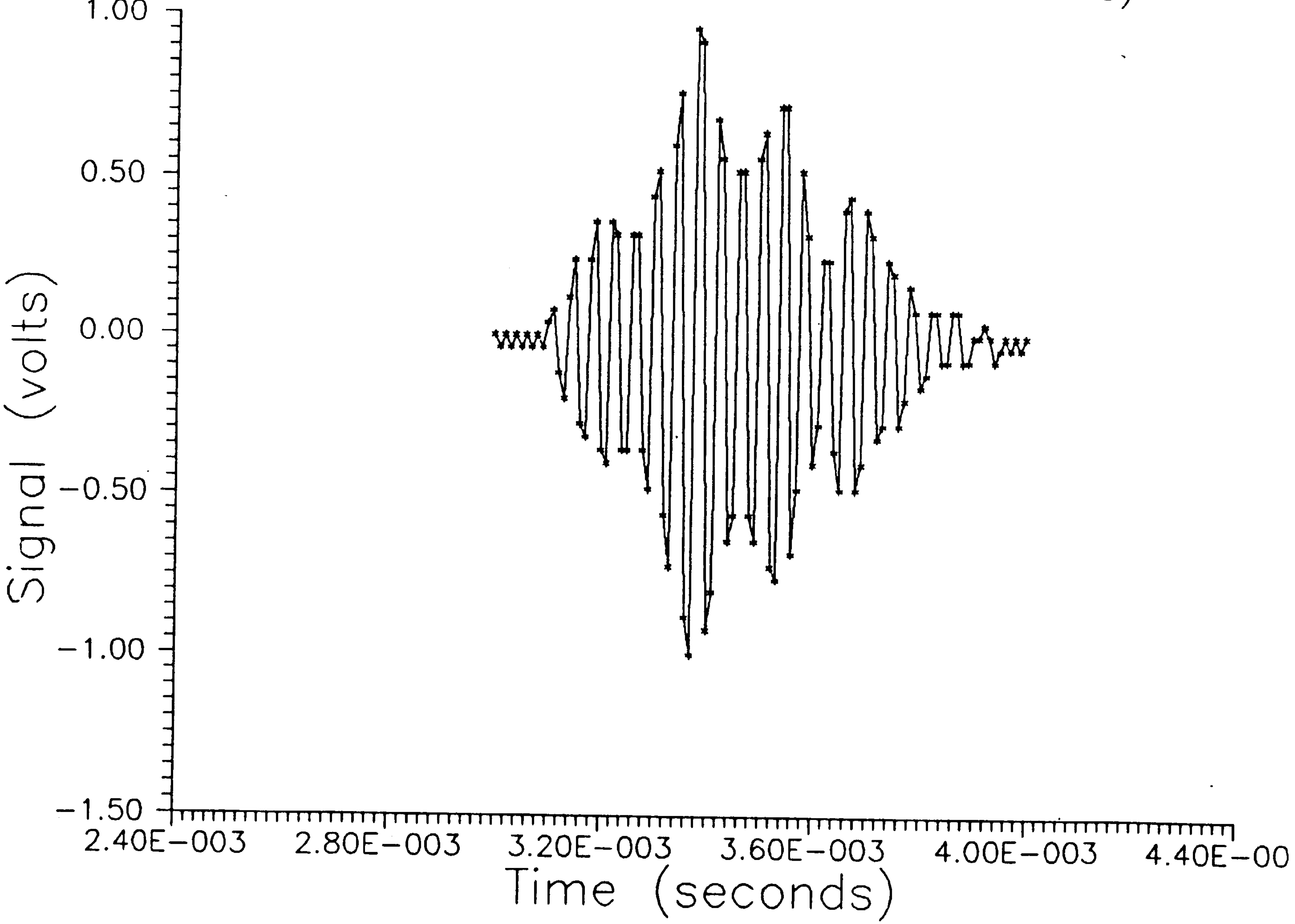


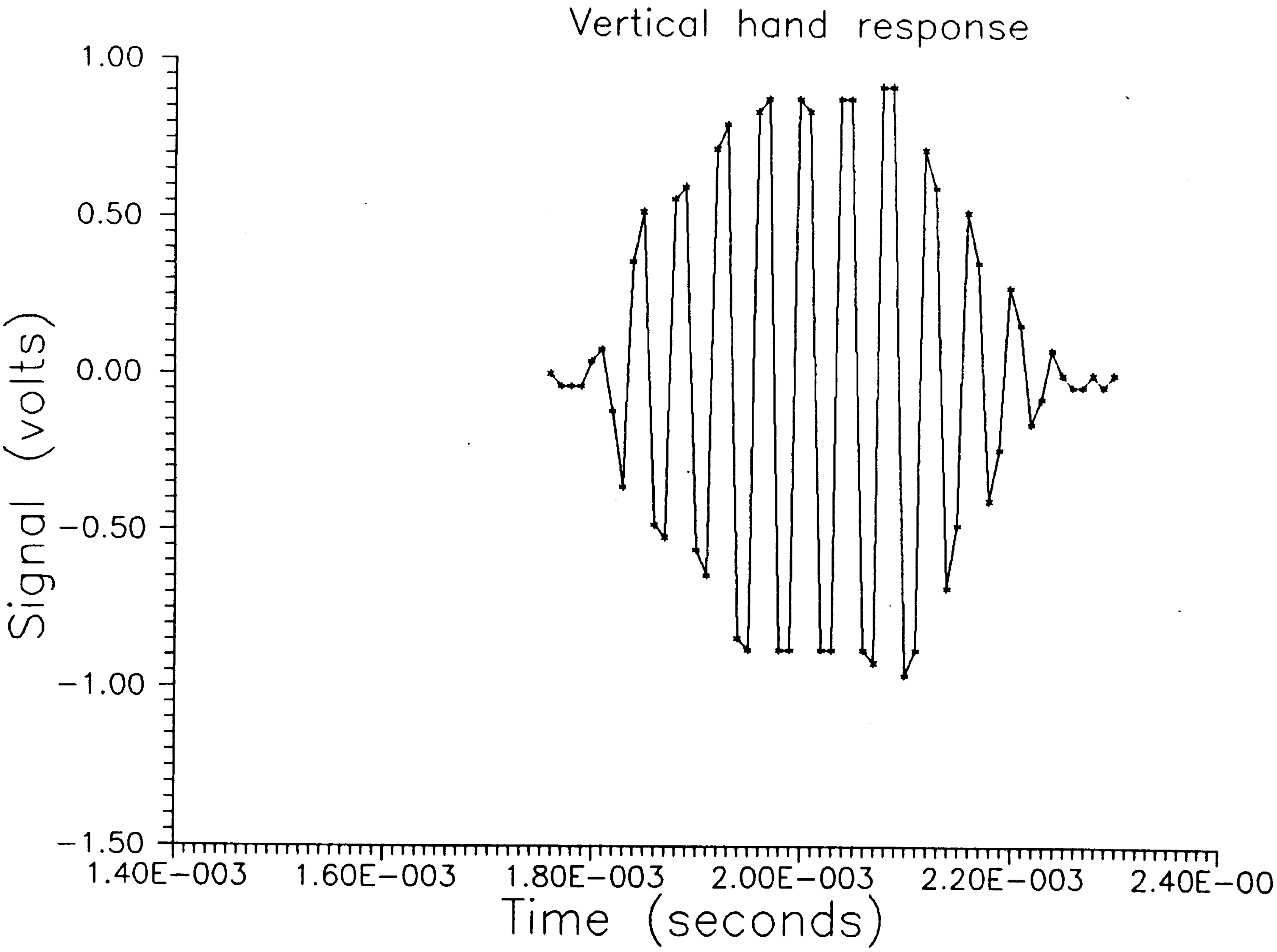

Response from compacted soil at 10 degrees
(soil contained scattered 1 in rocks)
Signal (volts)
0.40
0.20
−0.00
−0.20
−0.40
−0.60
−0.80
1.00E−003
2.00E−003
3.00E−003
4.00E−003
5.00E−003
6.00E−00
Time (seconds)



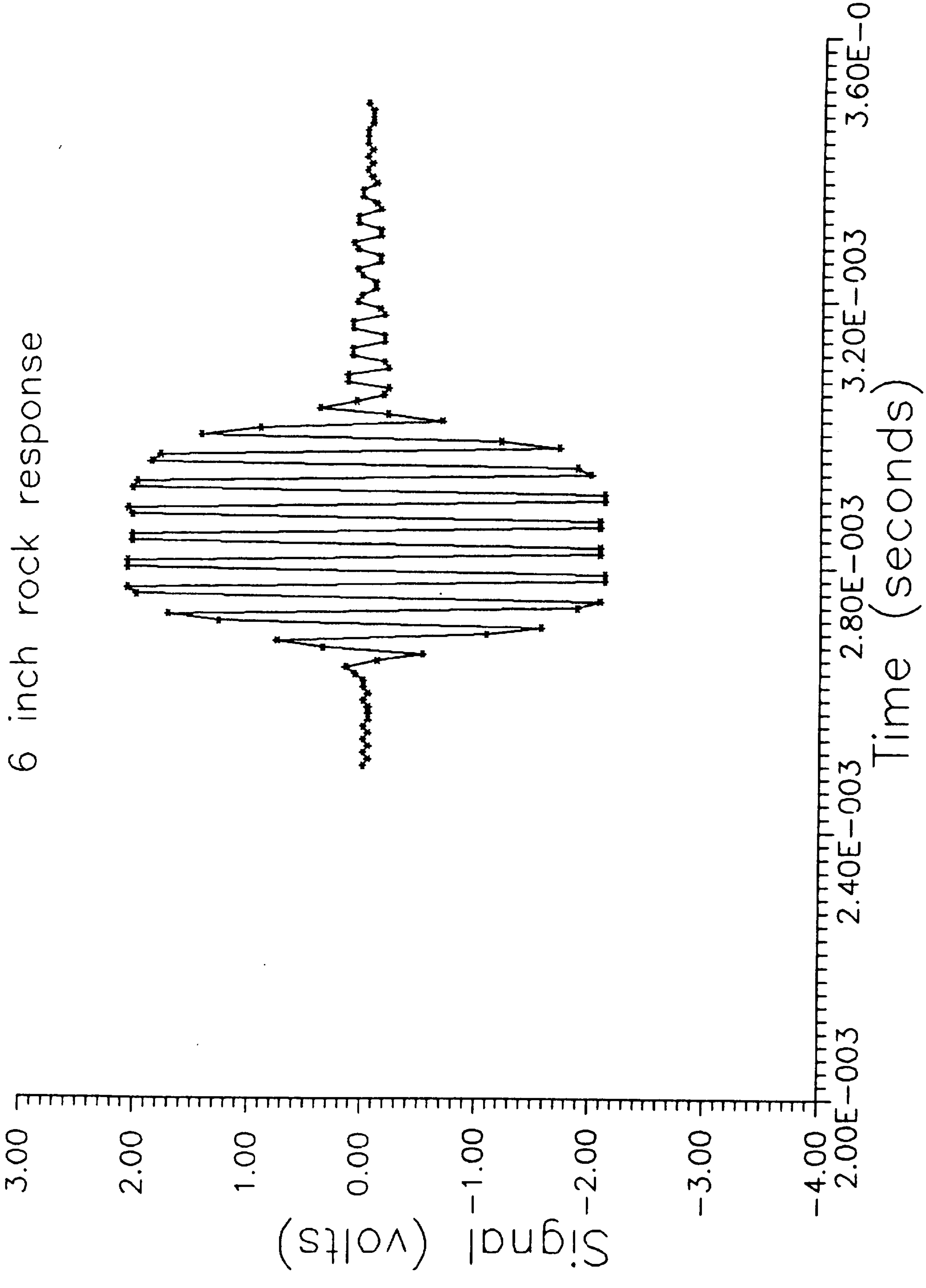
Response from compacted soil at 45 degrees
(soil contained scattered 1 in rocks)

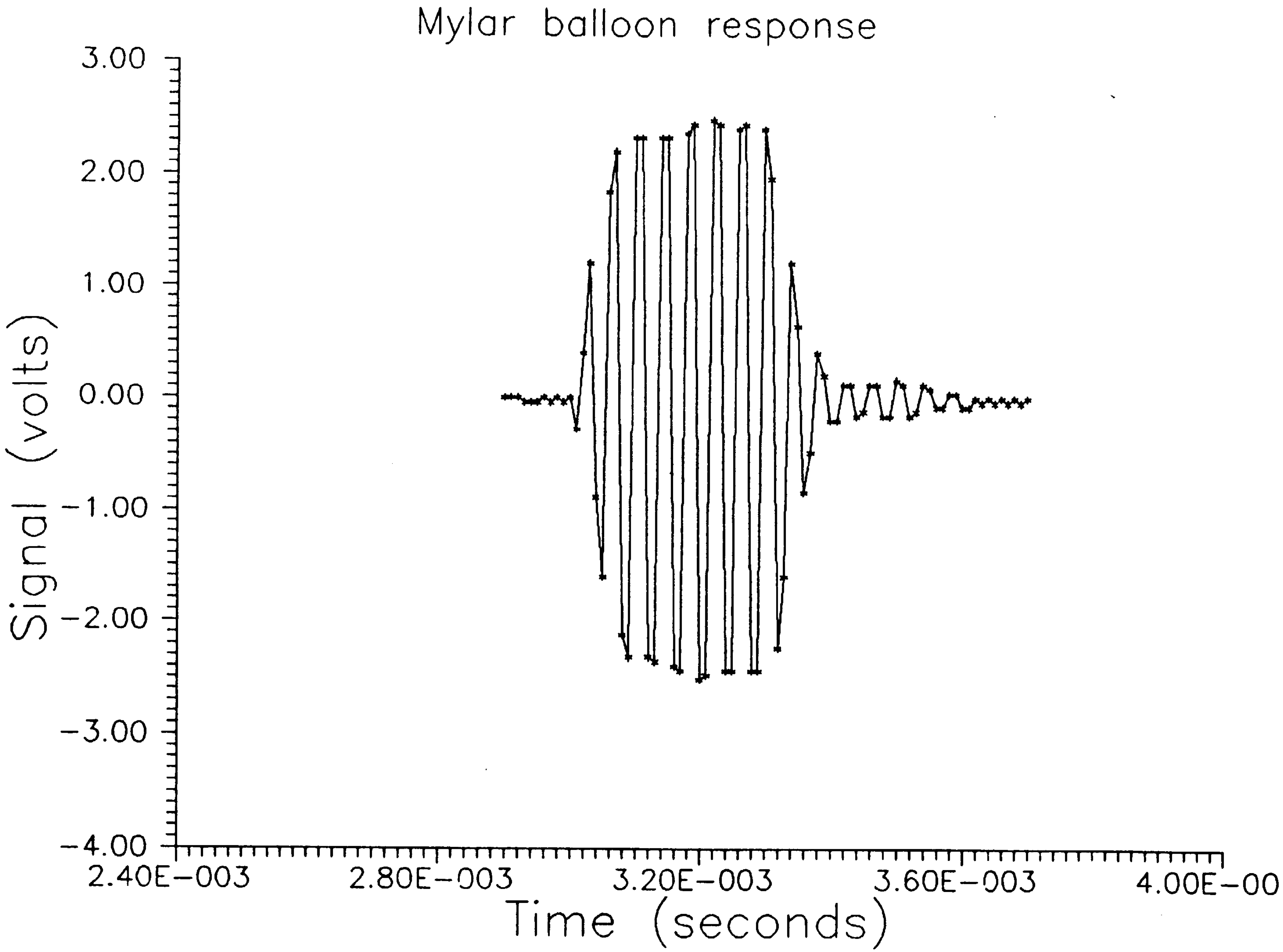
Response from vertical, compacted soil
(soil contained scattered 1 in rocks)



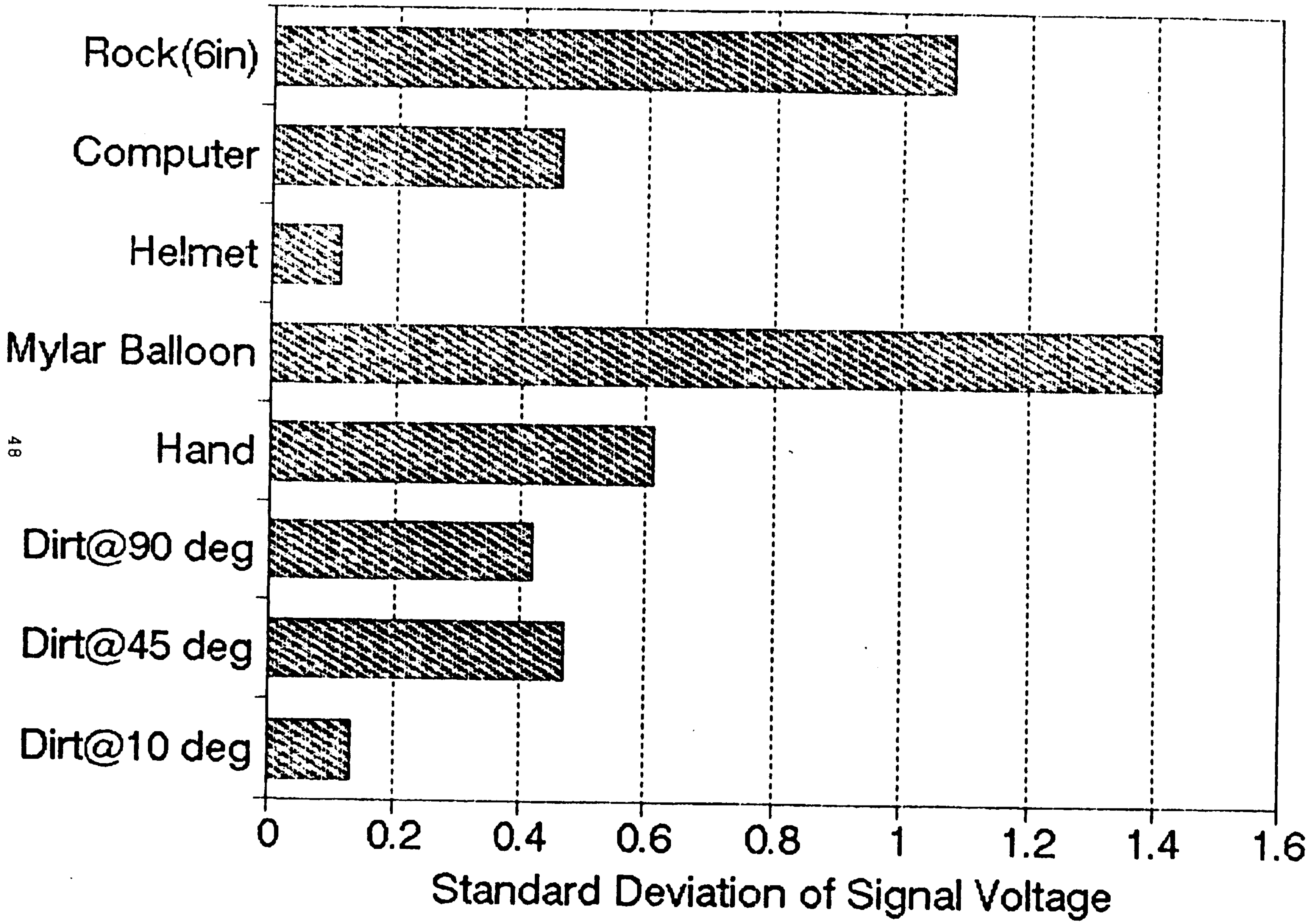


Vertical hand response


6 inch rock response


Mylar balloon response
Signal (volts)
3.00
2.00
1.00
0.00
−1.00
−2.00
−3.00
−4.00
2.40E−003
2.80E−003
3.20E−003
3.60E−003
4.00E−00
Time (seconds)

Comparison of Response Signals
(Polaroid transducer @ 55 kHz)



APPENDIX G - QED SOFTWARE

```
hex
10 width !  \ Names are saved by first 16 letters \
0000 06 DP X!  \ 32K for code \
4000 04 NP X!  \ 16K name area \
0000 05 XCONSTANT BOTTOM.OF.UPDATE.HEAP
7FFF 05 XCONSTANT TOP.OF.UPDATE.HEAP
BOTTOM.OF.UPDATE.HEAP TOP.OF.UPDATE.HEAP IS.HEAP
decimal
0.00156 FCONSTANT T   \ Time constant used for ultrasonic FFT's \
FVARIABLE ultra_freq1
0.0 ultra_freq1 f!
1000 CONSTANT beacon1_freq
2000 CONSTANT beacon2_freq


variable SAFE  \ Semaphor variable used for critical processes \
-1 SAFE !
variable in_menu
0 in_menu !
variable last_sonar_time
0 last_sonar_time !
variable last_position
0 last_position !
variable last_findpos_time
0 last_findpos_time !


matrix: waveform_sample           \ Matrix to store analog waveform \
2 64 ' waveform_sample dimmed     \ sample of ultrasonic receiver.  \
' waveform_sample zero.matrix


matrix: audio_sample              \ Matrix to store analog waveform sample \
2 64 ' audio_sample dimmed        \ from an audio (.5 - 5KHz) source.      \
' audio_sample zero.matrix


matrix: terrain_map            \ "memory-map" of 10 m. X 10 m. terrain that \
40 40 ' terrain_map dimmed     \ the rover navigates through. This matrix   \
' terrain_map zero.matrix      \ records the placement of obstacles as they \
                               \ are detected.                              \

: init.dacs                    \ Initializes the motors to 0 volts \
9 1 do 0 I >dac loop
;


array: char_buffer
20 1 1 ' char_buffer dimensioned
' char_buffer zero.array
```

```
: blank.buff
20 0 do 32 I char_buffer c! loop
;
blank.buff


array: start_position                  \ (X,Y) coordinate of rover's start \
2 1 4 ' start_position dimensioned     \ position on memory map.           \
' start_position zero.array


array: current_position                \ (X,Y) coordinate of rover's current \
2 1 4 ' current_position dimensioned   \ position on memory map.             \
' current_position zero.array


array: destination_position                \ (X,Y) coordinate of destination \
2 1 4 ' destination_position dimensioned   \ point.                          \
' destination_position zero.array


array: direction_vector                \ Polar (Mag,Dir) vector representation \
2 1 4 ' direction_vector dimensioned \ of rover's direction of travel.         \
' direction_vector zero.array


array: destination_vector              \ Polar representation of the vector \
2 1 4 ' destination_vector dimensioned \ from rover's current point on the  \
' destination_vector zero.array        \ memory map to the destination      \
                                       \ point on memory map.               \

array: motor_level                  \ Analog voltage output to the motor \
4 1 2 ' motor_level dimensioned     \ controller board.                  \
' motor_level zero.array
: init.motors
4 0
do
0 I motor_level !
loop
;
init.motors


array: tilt                     \ Stores the analog signal from the 2-axis \
2 1 2 ' tilt dimensioned        \ tilt sensor.                             \
' tilt zero.array
```

```
: init.tilt
2 0
do
2048 i tilt !
loop
;
init.tilt


array: keypad_index                   \ Provides a mapping of the actual \
20 1 2 ' keypad_index dimensioned     \ keyboard array values to those   \
' keypad_index zero.array             \ of the user's template.          \
: init.keypad
13 0
do
keypad_index !
loop
;
7 18 4 17 1 16 8 14 5 13 2 12 9 10 6 9 3 8 27 6 0 4 13 0 -2 5
init.keypad



38400 0                     \ Sets up the environment for the obstacle    \
task: update.map            \ avoidance process. The build.standard.task  \
16384 15 20479 15 28672 15  \ command initializes the heap and variable   \
update.map                  \ areas and places this process in the "round \
build.standard.task         \ robin" loop of the multitasker.             \


39424 0                     \ Sets up the beacon positioning process \
task: get.location
20480 15 24575 15 28928 15
get.location
build.standard.task


40448 0                     \ Sets up the motor control process \
task: motor.control
24576 15 28671 15 29184 15
motor.control
build.standard.task
```

```
: direction.status
\
\  Sets task environment variable in order to block a process
\  or awaken it:
\
\     <status> direction.status
\
\  (note: status is either 0 or -1)
\
status get.location task's.user.var !
;

: motor.status
status motor.control task's.user.var !
;

: sonar.status
status update.map task's.user.var !
;


: get_time.s
\
\  Gets seconds from system time
\
read.elapsed.time
3 ndrop
swap
drop
;



: get_time.m
\
\ Gets milliseconds from system time
\
read.elapsed.time
drop drop 0 0
fswap drop fswap drop
flot fswap swap drop
flot
1000.0
f* f+
;
```

```
: get_user_number
\
\  Gets a floating point number from user keypad:
\
\     <line> <col> get_user_number
\
-1 0 0 0 0 0.0 locals{ f&user_val char place line col error }
to col
to line
begin
   error
while
   blank.buff
   0 to place
   begin
      keypad
      to char
      place 0 =
      if
         6 0 4 c!
         32
         7 1
         do
            dup
            I 4 c!
         loop
         0 4 line col 1 + $>display
         update.display
         drop
      endif
      char keypad_index @ 48 +
      to char
      place 1+ to place
      char 61 = not
      if
         line col place + put.cursor
         char char>display
         char place char_buffer c!
      endif
      place 6 >=
      char 61 =
      or not
   while
   repeat
   char 61 =
   if
      place 1 - to place
   endif
   place 2 + 0 4 c!
   place 2 + 0
   do
      I char_buffer c@
      I 1 + 4 c!
   loop
   0 4 line col $>display
   update.display
   0 4 $>f
   if
      fdup to f&user_val
         10.0 f>
            if
               -1 to error
            else
               0 to error
            endif
   else
      -1 to error
   endif
   error
   if
      6 0 4 c!
      32 82 79 82 82 69
      7 1
      do
      I 4 c!
      loop
      0 4 line col 1 + $>display
      update.display
      line col 1 + put.cursor
   endif
repeat
f&user_val
;


: get_audio_tone
\
\  Gets audio sample from 12-bit analog port #8, performs an FFT, and
\  returns the amplitude of the < frequency > requested:
\
\     < frequency > get_audio_tone
\
0 locals{ freq }
200 /
to freq
' audio_sample zero.matrix
SAFE @
if
asleep motor.status
asleep sonar.status
endif
```

```
45056
0
64
-1
7
a/d12.multiple
1 0
do
   64 0
   do
     45056
     64
     J *
     +
     I 2 * +
     0
     @
     flot
     819.0
     f/
     J I audio_sample F!
   loop
loop
' audio_sample FFT
1 freq audio_sample F@
fabs
fdup f*
0 freq audio_sample F@
fabs
fdup f*
f+ fsqrt
31.4159
f/
SAFE @
if
awake motor.status
awake sonar.status
endif
;


: find_position
0 0.0 0.0 0.0 0.0 0.0 0.0 locals{ f&theta1 f&theta2 f&d1 f&d2
                                  f&delta_t f&r bad }
\
\   Calculates current rover position based on signals that
\   are received from the optical beacon system. Due to the
\   time neccessary for the FFT signal processing, it is
\   recommended that the beacons rotate at no more than 5 RPM.
\
decimal
```

```
begin
" L" 0 18 $>display
update.display
0 to bad
-1 7 a/d12.sample
256 >
if
   beacon1_freq
   get_audio_tone
   0.25 f>
   if
      get_time.m
      begin
         -1 7 a/d12.sample
         16 >
      while
      repeat
      begin
         -1 7 a/d12.sample
         16 <
      while
      repeat
      get_time.m
      beacon1_freq
      get_audio_tone
      0.25 f>
      if
         fswap f- to f&delta_t
         0.0005235 f&delta_t f*     \ 0.0005235 assumes 5 RPM; change
                                    \ this if neccessary.
         to f&theta1
         begin
            beacon2_freq
            get_audio_tone
            0.25 f<
         while
         repeat
         get_time.m
         begin
            -1 7 a/d12.sample
            16 >
         while
         repeat
         begin
            -1 7 a/d12.sample
            16 <
         while
         repeat
         get_time.m
         beacon2_freq
         get_audio_tone
```

```
            0.25 f>
            if
               fswap f- to f&delta_t
               0.0005235 f&delta_t f*        \ 0.0005235 assumes 5 RPM; change
                                             \ this if neccessary.
               to f&theta2
            else
               -1 to bad
               fdrop
            endif
         else
            -1 to bad
            fdrop
         endif
      else
      -1 to bad
      endif
      bad not
      if
         1.571 f&theta1 f- to f&theta1
         1.571 f&theta2 f- to f&theta2
         f&theta2 fsin 10.0 f*
         3.142 f&theta1 f&theta2 f+ f- fsin f/      \ law of sines \
         to f&r
         1.571 f&theta1 f- to f&theta1
         f&r f&theta1 fcos f* 0 current_position f!
         f&r f&theta1 fsin f* 1 current_position f!
         0 current_position f@ f.
         1 current_position f@ f.
      endif
   endif
   "  " 0 18 $>display
   update.display
   again
   halt
   ;


   : look
   \
   \   Enables the ultrasonic transmitter IC and counts the number of
   \   milliseconds between enabling the LM1812 IC and receiving a
   \   reflection signal from an object in the near proximity. Knowing
   \   the speed of sound and the elapsed time, this function can calculate
   \   the obstacle's distance from the rover.
   \
   0 0 0.0 0.0 0.0 0.0 0.0 locals{ f&delta_t f&start_time f&distance
                                   f&delta_x f&delta_y x_coord y_coord }
   decimal
   get_time.s
   last_sonar_time @ =
   not
```

```
          1.0 40 y_coord - x_coord terrain_map f!     \ Obstacle is recordea on \
      endif                                           \ the terrain memory map. \
      get_time.s
      last_sonar_time !
      "  " 0 17 $>display
      update.display
      awake direction.status
      awake motor.status
  endif
  ;



  : get.destination.vector
  \
  \ Calculates destination vector in polar form
  \
  0.0 0.0 locals{ f&x f&y }
     0 destination_position f@
     0 current_position f@ f- to f&x
     f&x fdup f*
     1 destination_position f@
     1 current_position f@ f- to f&y
     f&y fdup f* f+
     fsqrt
     0 destination_vector f!
     f&y f&x f/ fatan
     1 destination_vector f!
  ;



  : test.sonar
  \
  \   Tests the frequency response of the ultrasonic transducers. When an
  \   ultrasound tone is received on analog port 0, the spectrum of frequencies
  \   that are detected are displayed on the LCD screen. This function allows
  \   the user to also specify which frequency should be used for the
  \   ultrasound "aliasing" frequency.
  \
  decimal
  0 locals{ fft.option }
  a/d8.on
  begin
     init.display
     " Ultrasonic receiver " 0 0 $>display
     " spectrum analyzer:  " 1 0 $>display
     " SHIFT for main menu " 2 0 $>display
     " ENTER to sample     " 3 0 $>display
     update.display
     ' waveform_sample zero.matrix
     0
```

```
   to fft.option
   keypad
   0 =
while
init.display
decimal
" waiting for signal  " 0 0 $>display
" input...            " 1 0 $>display
update.display
begin
   0 a/d8.sample
   16
   <
while
repeat
init.display
update.display
asleep status 33792 0 task's.user.var !
45056
0
64
0
a/d8.multiple
awake status 33792 0 task's.user.var !
1 0
do
   64 0
   do
     45056
     64
     J *
     +
     I +
     0
     c@
     flot
     51.2
     f/
     J I waveform_sample F!
   loop
loop
' waveform_sample FFT
" DUMP SST SET        " 3 0 $>display
update.display
cr
cr
begin
keypad
to fft.option
fft.option
19 =
fft.option
```

```
15 =
or
not
while
repeat
32 0
do
   2 1
   do
     fft.option
     15 =
     if
     begin
     ?keypad
     not
     while
     repeat
     to fft.option
     endif
     fft.option
     11 =
     if
       J 1 -
       flot T f/
       ultra_freq1 f!
       2 0
       do
          "         " 0 10 $>display
          update.display
          5000 0
          do
          loop
          ultra_freq1 f@
          f>floating$
          drop
          0 10 $>display
          update.display
          5000 0
          do
          loop
       loop
       begin
       ?keypad
       not
       while
       repeat
       drop
       15
       to fft.option
     endif
     J flot T f/ f.
     I J waveform_sample F@
     fabs
```

```
      fdup f*
      0 J waveform_sample F@
      fabs
      fdup f*
      f+ fsqrt
      31.4159
      f/
      fdup f.
      cr
      f>floating$
      drop
      "                   " 0 0 $>display
      "                   " 1 0 $>display
      update.display
      1 10 $>display
      "      Freq:"
      0 0 $>display
      " Amplitude:"
      1 0 $>display
      J flot T f/
      0.0 f=
      if
      "  DC" 0 10 $>display
      else
      J flot T f/
      f>floating$
      drop
      0 10 $>display
      endif
      update.display
   loop
loop
init.display
" ultrasound detector" 0 0 $>display
" set to " 1 0 $>display
ultra_freq1 f@
f>floating$
drop
1 8 $>display
"  Hz" 1 14 $>display
update.display
begin
   ?keypad
   not
while
repeat
drop
repeat
-1 in_menu !
init.display
;
```

```
: activate.motors
\
\   Causes the rover motors to activate on two specified ports
\   for a given number of seconds + milliseconds:
\
\      <porta> <portb> <seconds> <milliseconds> activate.motors
\
0 0 0.0 0 0 0 0 0.0 locals{ f&current start_sec start_mil sec mil f&total_start_mils
p1 p2 }
to mil
to sec
to p2
to p1
p1 1+ to p1
p2 1+ to p2
32 0
do
  I 8 * p1 >dac
  I 8 * p2 >dac
loop
read.elapsed.seconds
drop
to start_sec
to start_mil
start_sec
flot
1000.0 f*
start_mil
flot f+
to f&total_start_mils
begin
   read.elapsed.seconds
   drop flot
   to f&current
   f&current
   1000.0 f*
   to f&current
   flot f&current f+
   to f&current
   f&current
   f&total_start_mils f-
   mil flot sec flot
   1000.0 f* f+
   f>=
   not
while
repeat
32 0
do
  255 I 8 * - p1 >dac
  255 I 8 * - p2 >dac
loop
;
```

```
: forward
\
\   Moves the rover forward for a specified number of seconds + milliseconds:
\
\       <seconds> <milliseconds> forward
\
0 0 locals{ secs mils }
to mils
to secs
0 2 secs mils activate.motors
;


: reverse
\
\   Moves the rover backward for a specified number of seconds + milliseconds:
\
\       <seconds> <milliseconds> forward
\
0 0 locals{ secs mils }
to mils
to secs
1 3 secs mils activate.motors
;


: left
\
\   Turns the rover left for a specified number of seconds + milliseconds:
\
\       <seconds> <milliseconds> forward
\
0 0 locals{ secs mils }
to mils
to secs
1 2 secs mils activate.motors
;


: right
\
\   Turns the rover right for a specified number of seconds + milliseconds:
\
\       <seconds> <milliseconds> forward
\
0 0 locals{ secs mils }
to mils
to secs
0 3 secs mils activate.motors
;
```

```
: set_motors
\
\   Starts motors 5 seconds after the rover is powered up.
\
decimal
begin
init.display
" M" 0 19 $>display
" Running...          " 1 0 $>display
"                     " 2 0 $>display
"                     " 3 0 $>display
update.display
1 5000 do loop
1 0 forward
"  " 0 19 $>display
update.display
again
halt
;



: test.motor
\
\   Calibration utility for testing the operation of the rover motors
\   in any user specified direction.
\
0 0 0 0 0 0 0 locals{ motor select cursor
                      DAC1 DAC2 escape system_motor_num }
init.display
" Select type  > FWD  " 0 0 $>display
" of test:       REV  " 1 0 $>display
"                 L   " 2 0 $>display
"                 R   " 3 0 $>display
update.display
-1 0 -1 display.options
0 13 put.cursor
begin
escape not
while
begin
   32 char>display
   cursor 13 put.cursor
   62 char>display
   cursor 13 put.cursor
   keypad
   to select
   select
   0 =
   select
   6 = or not
while
   select 1 =
   cursor 3 <
   and
   if
      cursor 1+
      to cursor
   endif
   select 2 =
   cursor 0 >
   and
   if
      cursor 1-
      to cursor
   endif
repeat
select 6 =
if
-1 to escape
endif
select 0 =
if
cursor
case
   0 of -1 0 0 display.options 5 0 forward endof
   1 of -1 0 0 display.options 5 0 reverse endof
   2 of -1 0 0 display.options 5 0 left endof
   3 of -1 0 0 display.options 5 0 right endof
endcase
endif
-1 0 -1 display.options
repeat
-1 in_menu !
;



: test.tilt
\
\   Utility for testing and manually calibrating the 2-axis tilt sensor.
\   The tilt angle is displayed on the LCD continuously, allowing the
\   user to adjust the calibration pots on the sensor until a zero
\   reading is made.
\
2 1 0 0 locals{ escape axis display_pointer pressed }
2 right.places !
1 to display_pointer
init.display
begin
" Axis to align:      " 0 0 $>display
"     X-axis          " 1 0 $>display
"     Y-axis          " 2 0 $>display
"                     " 3 0 $>display
update.display
-1 0 -1 display.options
display_pointer 12 put.cursor
```

```
60 char>display
display_pointer 12 put.cursor
escape
pressed 6 =
or not
while
0 to axis
begin
     keypad
     to pressed
     pressed 1 =
     if
     1 12 put.cursor
     32 char>display
     2 12 put.cursor
     60 char>display
     2 12 put.cursor
     2 to display_pointer
     endif
     pressed 2 =
     if
     2 12 put.cursor
     32 char>display
     1 12 put.cursor
     60 char>display
     1 12 put.cursor
     1 to display_pointer
     endif
     pressed 6 =
     if
     -1 to escape
     endif
     pressed 0 =
     escape or
     not
while
repeat
escape
if
6
endif
escape not
if
display_pointer 1 -
to axis
clear.display
-1 0 0 display.options
axis 0 =
if
"  X-axis:            " 1 0 $>display
update.display
endif
axis 1 =
if
"  Y-axis:            " 1 0 $>display
update.display
endif
begin
3000 0
do loop
"              " 1 10 $>display
update.display
-1 axis a/d12.sample
flot
819.2 f/ 2.5 f-
.075 f/
f>fixed$
drop 1 10 $>display
update.display
?keypad not
while
repeat
endif
to pressed
repeat
3 right.places !
-1 in_menu !
;



: test.beacon
\
\   Utility for testing the response of the optical beacon system.
\   This will test the accuracy of the angles measured by the QED
\   when the beacons are rotating.
\
0 0 0.0 0.0 locals{ f&theta1 f&delta_t bad escape }
decimal
2 right.places !
escape not
if
init.display
" Waiting for signal  " 0 0 $>display
" from beacon...      " 1 0 $>display
update.display
0 to bad
begin
-1 7 a/d12.sample
256 < escape not and
while
   ?keypad
   if
      6 =
      if
         -1 to escape
      endif
   endif
repeat
escape not
if
   init.display
   get_time.m
   beacon1_freq
   get_audio_tone
   0.25 f>
   if
      " First tone detected " 0 0 $>display
      "                     " 1 0 $>display
      update.display
      0 10 0 do
      -1 7 a/d12.sample
      +
      loop 10 /
      128 >
      if
      begin
         0 10 0 do
         -1 7 a/d12.sample
         +
         loop 10 /
         128 >
      while
      repeat
      endif
      begin
         0 10 0 do
         -1 7 a/d12.sample
         +
         loop 10 /
         128 <
      while
      repeat
      get_time.m
      beacon1_freq
      get_audio_tone
      0.25 f>
      if
         " Second tone detected" 0 0 $>display
         "                    " 1 0 $>display
         update.display
         fswap f- to f&delta_t
         0.0005235 f&delta_t f*     \ 0.0005235 assumes 5 RPM; change
                                    \ this if neccessary.
         to f&theta1
         0 10 0 do
        -1 7 a/d12.sample
         +
         loop 10 /
         128 >
         if
            begin
               0 10 0 do
               -1 7 a/d12.sample
               +
               loop 10 /
               128 >
            while
            repeat
         endif
      else
         fdrop fdrop
         -1 to bad
      endif
   else
   fdrop
   -1 to bad
   endif
   bad not
   if
      " theta1=" 0 0 $>display
      f&theta1 >degrees
      f>fixed$
      drop
      0 7 $>display
      " deg." 0 16 $>display
      update.display
   endif
endif
endif
3 right.places !
escape not
if
   keypad
   drop
endif
init.display
-1 in_menu !
;


: test.direction
\
\  Function used to read direction if an electronic compass signal
\  is connected to the rover.
\
;
```

```
: setup.coordinates
\
\   This function allows the user to enter the starting and destination
\   points from the keypad. The coordinates entered must be values between
\   0.0 and 9.999 meters.
\
1 0 0 locals{ select escape menu_option }
asleep direction.status
asleep motor.status
2 right.places !
2 left.places !
init.display
" Select coordinates  " 0 0 $>display
" to change:          " 1 0 $>display
" DEST. (     ,     ) " 3 0 $>display
0 start_position f@
f>fixed$
drop 2 6 $>display
1 start_position f@
f>fixed$
drop 2 12 $>display
update.display
" START (" 2 0 $>display
" ," 2 12 $>display
" ) " 2 18 $>display
update.display
0 destination_position f@
f>fixed$
drop 3 6 $>display
1 destination_position f@
f>fixed$
drop 3 12 $>display
update.display
" DEST. (" 3 0 $>display
" ," 3 12 $>display
" ) " 3 18 $>display
update.display
-1 0 -1 display.options
2 5 put.cursor
62 char>display
2 5 put.cursor
begin
     keypad
     to select
     select 1 =
     if
        2 5 put.cursor
        32 char>display
        3 5 put.cursor
        62 char>display
        3 5 put.cursor
        2 to menu_option
     endif
     select 2 =
     if
        3 5 put.cursor
        32 char>display
        2 5 put.cursor
        62 char>display
        2 5 put.cursor
        1 to menu_option
     endif
     select 6 =
     if
        -1 to escape
        0 to menu_option
     endif
     select 0 =
     escape or
     not
while
repeat
menu_option
case
    1 of 2 7 put.cursor
         2 6 get_user_number
         0 start_position f!
         2 13 put.cursor
         2 12 get_user_number
         1 start_position f!
      endof
    2 of 3 7 put.cursor
         3 6 get_user_number
         0 destination_position f!
         3 13 put.cursor
         3 12 get_user_number
         1 destination_position f!
      endof
endcase
2 right.places !
5 left.places !
init.display
awake direction.status
awake motor.status
-1 in_menu !
;



: cal.menu
\
\   Menu function for various device utilities.
\
init.display
" Device Calibration  " 0 0 $>display
" Menu:               " 1 0 $>display
"                     " 2 0 $>display
" SND MOT TILT NAV LOC" 3 0 $>display
```

```
update.display
keypad
case
  19 of test.sonar endof
  15 of test.motor endof
  11 of test.tilt  endof
   7 of test.beacon endof
   3 of setup.coordinates endof
endcase
;



: main_menu
\
\   This function is responsible for checking the presence of a keypad
\   connection; if a key is pressed, WALRUS goes into the calibration
\   mode. In this mode, a menu is displayed which allows testing of
\   various devices used by the rover.
\
0 locals{ menu_option }
0 start_position f@ 0 current_position f!
1 start_position f@ 1 current_position f!
begin
   get.destination.vector
   ?keypad
   in_menu @
   or
   if
      depth
      0 =
      not
      if
         drop
      endif
      asleep direction.status
      asleep motor.status
      0 SAFE !
      0 in_menu !
      init.display
      " WALRUS Test          " 0 0 $>display
      " Environment:         " 1 0 $>display
      "                      " 2 0 $>display
      " RUN CAL TASK         " 3 0 $>display
      update.display
      keypad
      to menu_option
      menu_option
      15 =
      if
         cal.menu
      endif
      init.display
      update.display
      -1 SAFE !
      awake direction.status
      awake motor.status
   endif
   look
again
halt
;


: system.start
\
\   "system.start" initializes all necessary hardware ports, starts the
\   system timers, allocates memory for each process that will be
\   executed in the multitasking environment, and finally causes all
\   processes to begin execution (all processes are asleep until this
\   function is executed).. This is the function which is pointed to by
\   the PRIORITY.AUTOSTART vector.
\
decimal
init.spi
-1 -1 init.pia
init.a/d12&dac
a/d8.on
init.elapsed.time
start.timeslicer
init.dacs
20480 15 24575 15 28928 15
get.location
build.standard.task
16384 15 20479 15 28672 15
update.map
build.standard.task
24576 15 28671 15 29184 15
motor.control
build.standard.task
50
*100us=timeslice.period
0 in_menu !
cfa.for main_menu update.map activate
cfa.for find_position get.location activate
cfa.for set_motors motor.control activate
;


\ execute: CFA.FOR SYSTEM.START PRIORITY.AUTOSTART when all debugging
\ is finished.
```

APPENDIX H - ELECTRONICS MANUFACTURER'S SPECIFICATIONS

EYQM

	MODEL NO.		33300-61	33200-500	73100-500	33700-500	63600-61	73500-500
	Voltage (V)	NOTE	12	12	12	24	24	24
PERFORMANCE AT MAXIMUM EFFICIENCY (typical)								
S load	Load Speed at Max Efficiency (RPM)		[illegible]	4500	4300	[illegible]	[illegible]	[illegible]
	Load Speed at Max Efficiency (rad/s)		492	471	450	450	[illegible]	419
	Power Output at Max Efficiency (Watts)		3.06	3.47	4.52	2.42	[illegible]	3.78
	Power Output at Max Efficiency (mHp)		[illegible]	4.6	6.1	3.2	[illegible]	5.1
	Efficiency (%)		[illegible]	57.4%	61.4%	54.9%	[illegible]	59.5%
I load	Load Current at Max Efficiency		0.45	0.50	0.61	0.18	[illegible]	0.26
RATED MOTOR PARAMETERS (typical)								
T load	Maximum Rated Torque (oz-in)	1.5	[illegible]	2.99	4.03	2.31	[illegible]	3.89
	Maximum Rated Torque (mN-m)	1.5	[illegible]	21.09	28.43	16.29	[illegible]	27.47
S load	Minimum Rated Speed (RPM)	1.5	[illegible]	2400	2500	2000	[illegible]	2000
	Minimum Rated Speed (rad/s)	1.5	[illegible]	251	262	209	[illegible]	209
	Power Output at Max. Rated Torque (Watts)	1.5	[illegible]	5.30	7.45	3.41	[illegible]	5.76
	Power Output at Max. Rated Torque (mHp)	1.5	[illegible]	7.1	10.0	4.6	[illegible]	7.7
	Efficiency (%)	1.5	[illegible]	36.58%	41.47%	31.10%	[illegible]	35.22%
I load	Maximum Rated Current (A)	1.5	[illegible]	1.21	1.50	0.46	[illegible]	0.68
I lock	Stall Current (A)		[illegible]	2.03	2.69	0.69	[illegible]	1.11
T lock	Stall Torque (oz-in)		[illegible]	5.25	7.54	3.63	[illegible]	6.60
	Stall Torque (mN-m)		[illegible]	37.07	53.28	25.65	[illegible]	46.64
I nl	No Load Current (A)		[illegible]	0.105	0.110	0.042	[illegible]	0.051
S nl	No Load Speed (RPM)		[illegible]	5777	5440	5206	[illegible]	4672
	No Load Speed (rad/s)		[illegible]	605	570	545	[illegible]	489
T cog	Cogging Torque (oz-in)		[illegible]	0.14	0.20	0.11	[illegible]	0.20
T d	Torque to Demagnetize (oz-in)	6	[illegible]	12.2	16.1	9.1	[illegible]	16.1
T fi	Internal Motor Friction (oz-in)		[illegible]	0.25	0.28	0.20	[illegible]	0.28
	Internal Motor Friction (mN-m)		[illegible]	1.77	1.98	1.41	[illegible]	1.98
J arm	Armature Inertia (oz-in-s^2)		[illegible]	2.40E-04	3.10E-04	1.60E-04	[illegible]	3.10E-04
	Armature Inertia (kg-m^2)		[illegible]	1.69E-06	2.19E-06	1.13E-06	[illegible]	2.19E-06
R t	Terminal Resistance (ohms)	3	[illegible]	5.91	4.46	34.69	[illegible]	21.55
L a	Terminal Inductance (mH)	4	[illegible]	3.09	2.42	16.91	[illegible]	12.10
K f	Frictional Damping Coefficient (oz-in-s)		[illegible]	1.53E-04	1.90E-04	1.28E-04	[illegible]	1.90E-04
K d	Damping Coefficient (oz-in-s)		[illegible]	8.20E-03	1.35E-02	5.20E-03	[illegible]	1.35E-02
	Max. Winding Temp (C)		[illegible]	130	130	130	[illegible]	130
R th	Thermal Dissipation Factor (C Rise/Watt)	1.3	[illegible]	11.0	9.5	13.1	[illegible]	9.5
	Thermal Dissipation Factor (C Rise/Watt)	2.3	[illegible]	20.2	17.8	23.6	[illegible]	17.8
T h	Thermal Time Constant (min)	1.3	[illegible]	4.6	6.3	2.5	[illegible]	6.3
	Thermal Time Constant (min)	2.3	[illegible]	11.9	12.9	8.7	[illegible]	12.9
K m	Motor Constant (oz-in/√Watt)		[illegible]	1.114	1.377	0.941	[illegible]	1.331
T m	Mechanical Time Constant (ms)		[illegible]	27.3	23.1	25.6	[illegible]	24.7
T e	Electrical Time Constant (ms)		[illegible]	0.52	0.54	0.49	[illegible]	0.56
	No Load Acceleration (rad/s^2)		25400	21900	24300	22700	[illegible]	21300
K b	Back EMF Constant (V/K RPM)		[illegible]	2.01	2.15	4.10	[illegible]	4.57
	Back EMF Constant (V/(rad/s))		[illegible]	0.019	0.021	0.039	[illegible]	0.044
K t	Torque Constant (oz-in/A)		2.43	2.71	2.91	5.54	5.78	6.18
	Torque Constant (mN-m/A)		[illegible]	19.14	20.55	39.12	40.88	43.64
MOTOR PERFORMANCE (SEE PERFORMANCE CHARTS BELOW)			①	②	③	①	②	③

Note 1 - Motor mounted horizontally on a 10 x 10 x 1/4 Aluminum plate

Note 2 - Motor in free air

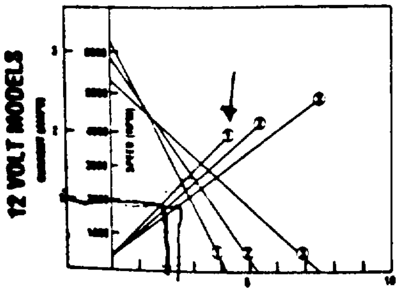
Note 3 - Rotor locked

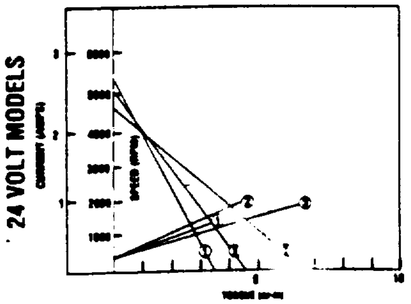
Note 4 - Inductance measured using a Sencor variable frequency inductance meter

Note 5 - Continuous duty

Note 6 - Causes a 5% loss of Full Load output torque

TYPICAL PERFORMANCE





▨ NORMALLY STOCKED

□ SPECIAL ORDER

TECHNICAL SUPPORT/ ORDERING INFORMATION:
Contact your local Barber-Colman representative, distributor or our sales office for assistance.

Issued: 8-92

Gates Energy Products [illegible] Nickel-Cadmium

ULTRAMAX C Fast Charge Cell

GLF-2800C

- Maximum runtime for demanding applications.
- Fast charge capability provides one hour recharge.
- Excellent rechargeable power source for low to moderate discharge rates.
- Typical applications: hand-held electronic devices, consumer electronics.

Model Number	GLF-2800C	
Cell Size	C	
Type	ULTRAMAX Fast Charge	
Voltage[1]	1.2V	
Capacity[1]	Minimum @ 5 hour rate	2800 mAhr
	Minimum @ 1 hour rate	2500 mAhr
Effective Internal Resistance (Re)	Maximum	21 mΩ
	Typical	19 mΩ
Maximum Discharge Current	Continuous	21 A
	Momentary (1 second)	42 A

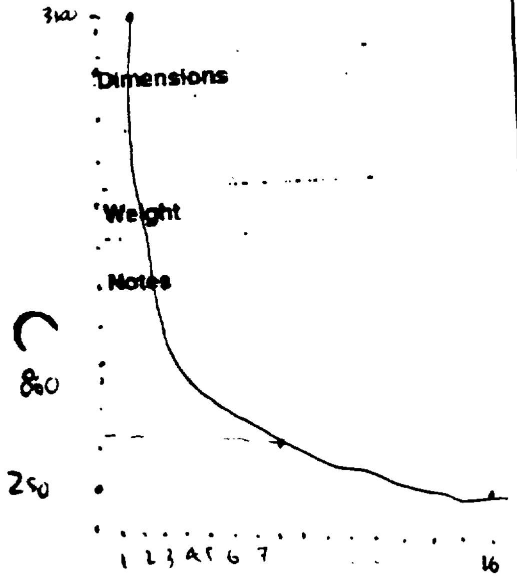
Charging[1]		Standard	Quick[3]	Fast[4]
	Min. Charge Rate (mA)	125	125	125
	Time Required (h)	36–48	36–48	36–48
	Max. Charge Rate (mA)	250	860	3100
	Time Required (h)	16–20	4–5	1–1.2
	Min. Overcharge (mA)	50	50	50
	Max. Overcharge (mA)	250	250	250

Temperature[2]		Standard Min.	Standard Max.	Quick Min.	Quick Max.	Fast Min.	Fast Max.
	Storage (°C)	–40	+50	–40	+50	–40	+45
	Discharge (°C)	–20	+50	–20	+50	–20	+45
	Charge (°C)	0	+50	+10	+50	+15	+45

Dimensions		Min.	Max.
	Diameter mm (in)	25.1 (0.99)	25.5 (1.00)
	Height mm (in)	48.5 (1.91)	49.0 (1.93)

Weight		
	Typical g (oz)	72.4 (2.55)

Notes

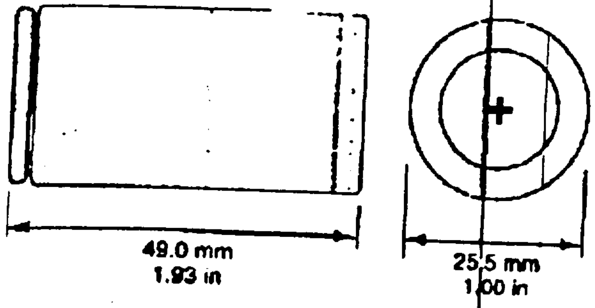


[1] Per American National Standards (ANSI) C18.2 - 1984.

[2] Cells can be charged at lower than specified temperatures but current must be appropriately reduced. Brief excursions to temperatures higher than specified may b tolerated. Cell life is decreased by heat from over-charging or ambient sources.

[3] Quick charge must be terminated or reduced to acceptable overcharge rate when ce reaches full charge.

[4] Fast charge requires charge control to terminate fast charge or switch to acceptable overcharge rate when cell reaches full charge.



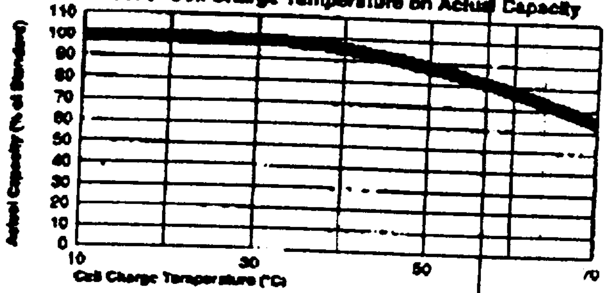
Cells are shown actual size

Single Cell Order Numbers

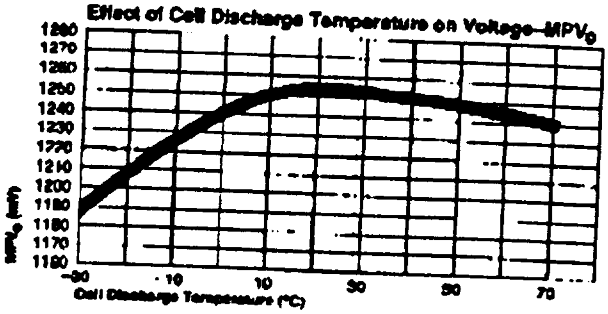
Bare	41B
Sleeve	41B037AB01001
Sleeve and tab	41B037AB01201

Refer to Nickel Cadmium Battery Selection Guide for additional information.

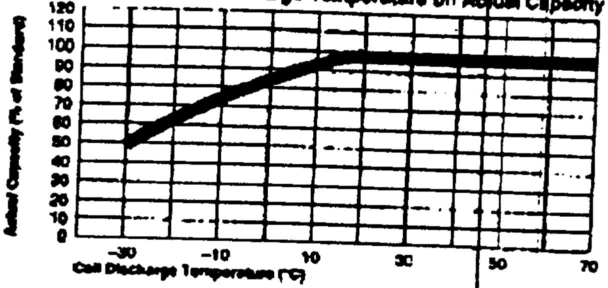
Effect of Cell Charge Temperature on Actual Capacity



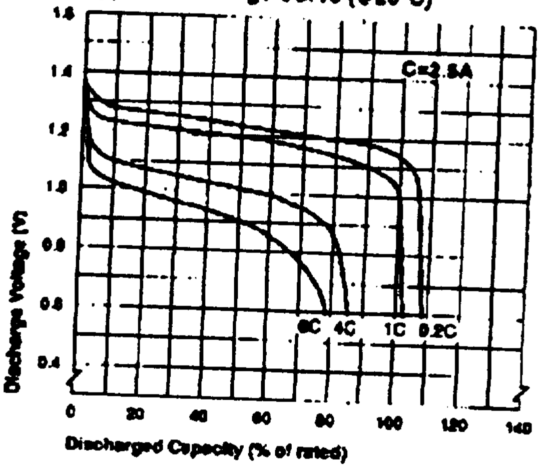
Effect of Cell Discharge Temperature on Voltage–MPV$_0$



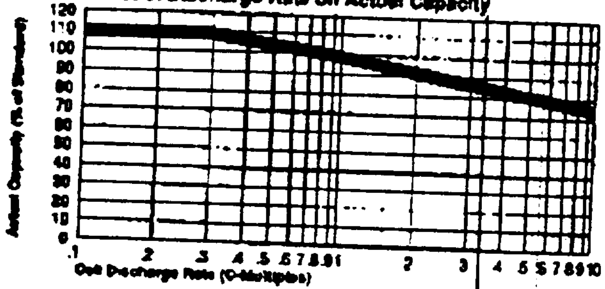
Effect of Cell Discharge Temperature on Actual Capacity



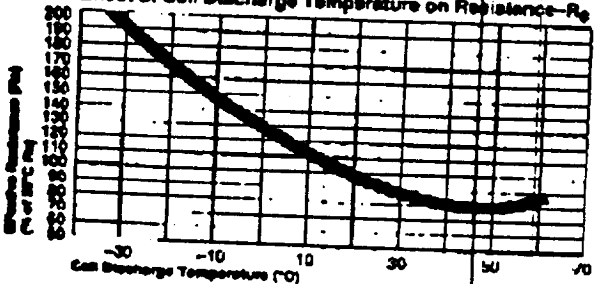
Typical Discharge Curve (@23°C)



Effect of Discharge Rate on Actual Capacity



Effect of Cell Discharge Temperature on Resistance–R$_e$



Gates Energy Products
P.O. Box 147114
Gainesville, FL 32614-7114

ORIGINAL PAGE IS
OF POOR QUALITY

Weight	Typical g (oz) 77
Notes	Per American National Standards (ANSI) C18.2 - 1984.

temperatures but current must be

The QED Board Integrates I/O-Ri
Hardware and Easy To Use Software

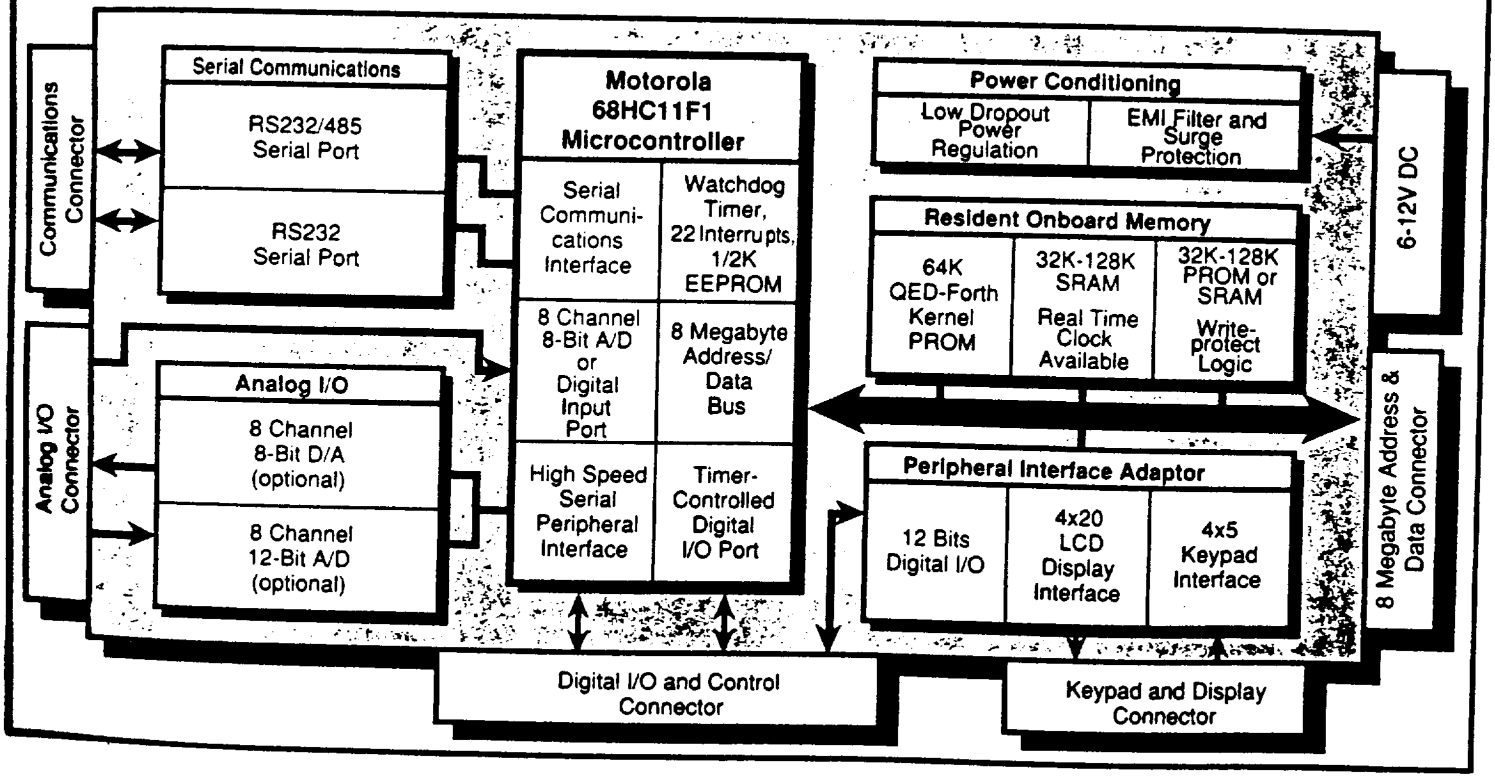
Tremendous Capability on a Palm-Sized Board

- Up to 32 digital I/O lines including 8 timer-controlled signals
- Up to 24 analog I/O lines with optional 12-bit A/D and 8-bit D/A
- Two serial ports, RS232 and RS232/485
- Built-in hardware/software interfaces for keypad, LCD display, and real-time clock
- 8 Megabyte addressable memory with up to 384K battery-backed RAM/ROM onboard
- Write-protection switches allow PROM-less programming
- 8 or 16 MHz 68HC11 microcontroller with built-in watchdog timer and 1/2K EEPROM
- Battery operable, runs on 6-12 volts, draws only 100mA

Complete Programming, Mathematics, anc Debugging Environment – Right On The Board

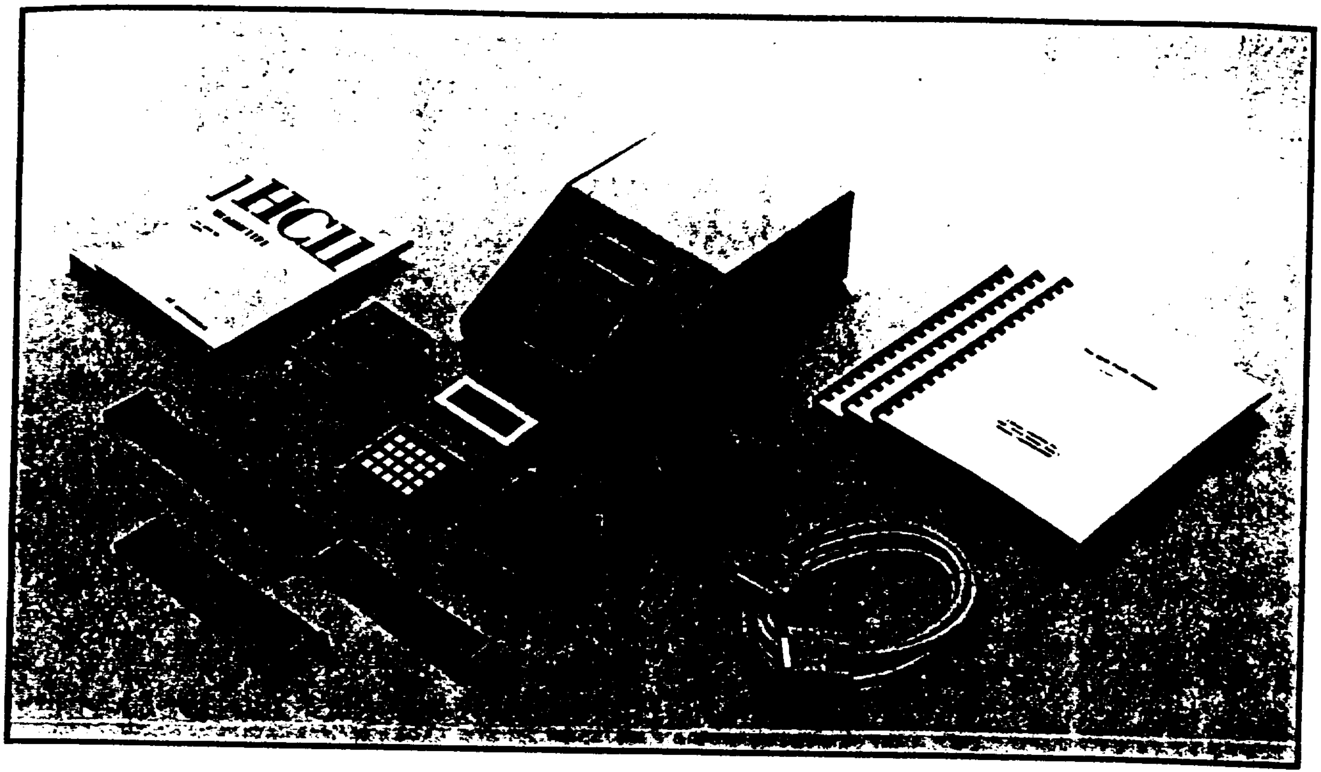
- Enhanced QED-Forth high level language
- Resident multitasking executive, interpreter, compiler, anc assembler
- Complete floating point math capabilities including trigonometric, log/exponential functions, and formatted real number input/output
- Workstation-caliber matrix math package with calibration functions, simultaneous equation solution, curve fitting, and FFTs
- Pre-coded drivers for keypad, display, A/D, D/A, and other I/O
- Built-in autostarter automatically runs your program at power up
- Powerful interactive debugger supports program trace, single step, break point insertion, and register viewing

Block Diagram of the QED Board


Communications Connector
Serial Communications
RS232/485 Serial Port
RS232 Serial Port
Analog I/O Connector
Analog I/O
8 Channel 8-Bit D/A (optional)
8 Channel 12-Bit A/D (optional)
Motorola 68HC11F1 Microcontroller
Serial Communi-cations Interface
Watchdog Timer, 22 Interrupts, 1/2K EEPROM
8 Channel 8-Bit A/D or Digital Input Port
8 Megabyte Address/ Data Bus
High Speed Serial Peripheral Interface
Timer-Controlled Digital I/O Port
Power Conditioning
Low Dropout Power Regulation
EMI Filter and Surge Protection
6-12V DC
Resident Onboard Memory
64K QED-Forth Kernel PROM
32K-128K SRAM
Real Time Clock Available
32K-128K PROM or SRAM
Write-protect Logic
8 Megabyte Address & Data Connector
Peripheral Interface Adaptor
12 Bits Digital I/O
4x20 LCD Display Interface
4x5 Keypad Interface
Digital I/O and Control Connector
Keypad and Display Connector

Rapidly Prototype Your Instrument Using

The QED Product Design Kit



The QED Product Design Kit™ integrates the QED Board, keypad, display, prototyping board, and power supply in a versatile instrument enclosure.

Fully Integrated Instrument Hardware

The QED Product Design Kit saves you time and effort by integrating all the ingredients of an impressive user-friendly instrument, including the powerful QED computer board, analog and digital I/O, serial communications, keypad and display. Built-in device drivers and pre-wired interfaces make the hardware easy to use so you can concentrate on the creative and unique aspects of your product.

Rapid Software Development

With the QED Board's programming environment you'll be up and running in no time. Everything you need to develop your software is included: superbly documented high level Forth language, multitasker, pre-coded I/O drivers, complete floating point math and matrix algebra package, and user friendly interactive debugger. And with the flip of a switch you can convert the onboard battery-backed RAM into write-protected "emulated ROM", so you won't waste time burning PROMs during product development.

Turn Your Smart Product Ideas Into Reality

The QED Board supports your product from idea to production. With the QED Product Design Kit you can quickly prototype and debug your instrument. And when your prototype is done, the application software is easily installed on the QED Board in your production instrument. OEM users can take advantage of the QED Board's economical volume discounts.

Mosaic Industries, Inc. • 5437 Central Ave. Suite 1 • Newark, CA 94560 • Phone 510-790-1255 • **Fax 510-790-0925**

The QED Board
Software Development Environment
Specifications

High Level Programmability: The QED Board™ includes an interpreter, compiler, assembler, math library, debugger, and multitasking real-time operating system in its 64K on-board ROM. The board is easily programmed with high level QED-Forth commands from any PC or terminal via an RS-232 serial link. QED-Forth is a superset of the Forth language that includes hundreds of pre-coded library functions, a floating point and matrix math package, access to an integrated assembler and symbolic debugger, and features such as local variables that simplify function definitions and enhance code readability.

Decision Making/Flow Control: QED-Forth supports conditional IF ... ELSE ... ENDIF statements, multiple-branch CASE statements, definite looping with DO ... LOOP and FOR ... NEXT, indefinite looping with BEGIN ... UNTIL, and conditional looping with BEGIN ... WHILE ... REPEAT statements.

Interrupt Support: Twenty 68HC11F1 interrupts have associated names in the QED-Forth dictionary. Interrupt service routines can be coded in either high level Forth or in assembly code, and are debugged just like any other routine. A simple ATTACH command installs the service routine so that it will be called whenever the specified interrupt is activated.

Device Drivers: Built-in high level routines provide easy access to onboard hardware including two RS-232 communications channels, keypad, display, 8 bit A/D, 12 bit A/D, 8 bit D/A, and real-time clock.

Integer Math: Dozens of integer and double number math routines are provided.

Floating Point Math: The full featured floating point math package includes simple math, log, trigonometric, and exponential functions, and formatted real number input/output. Floating point numbers maintain 5 decimal digits of precision over a range of 10^{-38} to 10^{-38}. Among the many floating point functions are: F+, F-, F*, F/, F**, FSQRT, FCOS, FSIN, FTAN, FATAN, FLN, FALN, FLOG10, FALOG10, FIXED, FLOATING, SCIENTIFIC, RANDOM.GAUSSIAN, PI, F>, F<, and F=.

Matrix Math: QED-Forth includes a complete matrix algebra library ranging from matrix editing functions to simultaneous equation solution, curve fitting, and fast fourier transforms. The following is a sample of the many dozens of QED-Forth matrix functions: MATRIX:, MATRIX*, MATRIX+, COPY.MATRIX, SWAP.MATRIX, ROW/COL.INSERTED, DOT.PRODUCT, MATRIX.VARIANCE, INVERTED, LEAST.SQUARES, SOLVE.EQUATIONS, and FFT.

Heap Memory Management: A heap-style memory manager facilitates dynamic run-time dimensioning of arrays, matrices, and data structures for optimal use of available RAM. Heaps and data structures as large as 8 Megabytes may be allocated and accessed as a single contiguous block of memory.

The QED Board Hardware Specifications

CPU: 8 MHz or 16 MHz Motorola 68HC11F1 microcontroller.

Address Space: 8 Megabytes, memory map optimized for high speed code execution.

On Board Memory: 96K minimum up to 384K maximum.
There are 3 memory sockets on the board:
Socket 1 accomodates a 64K or 128K ROM; each board is shipped with a 64K development ROM in this socket.
Socket 2 accommodates a 32K or 128K RAM with optional battery backup and real-time clock.
Socket 3 accomodates 32K or 128K standard or battery-backed RAM, or 32K to 128K PROM. RAM in this socket can be write-protected for PROM-less program development.
320 bytes of EEPROM are available for the user.

BACKUP TIMER !

Software: Complete QED-Forth software development environment resides in 64K onboard ROM.
Includes interactive Forth interpreter and compiler, assembler, symbolic debugger, comprehensive floating point and matrix math package, interrupt support, and device drivers.
The QED Board is easily programmed from any PC or terminal via RS232 serial link.

Real-Time Clock: An optional real-time clock resides in a sealed battery-backed RAM in memory socket 2. It is fully supported by the QED-Forth software.

Timer Controlled I/O: 3 or 4 input capture functions facilitate accurate detection of pulse edges and measurement of pulse widths. ← SIZE DETERMINATION / DISTANCE
4 or 5 output compare functions make it easy to create complex waveforms and pulse-width modulated signals.
1 pulse accumulator function facilitates frequency measurement and pulse counting.

SENSOR !

A/D Conversion: 8 channels of 8-bit analog-to-digital conversion with up to 45kHz sampling frequency. (90 kHz with 16MHz CPU.)
8 channels single-ended or 4 channels differential 12-bit analog-to-digital conversion with up to 13 kHz sampling frequency. (26kHz with 16MHz CPU.)

LF Control Frequencies

D/A Conversion: 8 channels of 8-bit multiplying digital-to-analog conversion.
Pairs of D/A converters may be combined to yield high resolution (11+ bits) D/A conversion.

Keypad Interface: 4 x 5 keypad connects via direct cable interface and is read by a pre-coded keypad scanning routine.

Display Interface: LCD display of up to 4 lines by 20 characters connects via direct cable interface and is controlled by pre-coded driver routines.

CONDITION STATEMENTS
RUN TIME - CLOCK
TEMP - MONITOR
BATTERY LEVEL - CHARGNG

Digital I/O: An onboard peripheral interface adaptor (PIA) provides three 8-bit I/O ports named PPA, PPB and PPC. The 68HC11F1 processor provides an additional 20 digital I/O lines for a total of 44 digital I/O signals.

The following table groups I/O signals by function and notes their alternate uses.

I/O signal group	Alternate use
8 I/O bits, PPA, bytewise configurable	
4 I/O bits, 1/2 PPC, nibble configurable	
8 Input bits, 68HC11F1 port E	(8 Bit A/D)
8 I/O bits, 68HC11 Port A, bitwise config	(Timer subsystem)
4 I/O bits, 68HC11 Port D, bitwise config	(DAC, 12-bit A/D, SPI)
8 I/O bits, PPB, bytewise configurable	(Display/keypad)
4 I/O bits, 1/2 PPC, nibble configurable	(Keypad)
44 Total digital signals	

Runtime Security: Watchdog timer and clock monitor ensure orderly reset if a processor error occurs.

Interrupts: 21 interrupts are associated with the 68HC11's on-chip subsystems.

Connectors: All connectors use standard shrouded single or dual row .025" square posts on a .1" centerline pitch.

40 pin digital I/O and control bus, dual row.

40 pin analog I/O bus, dual row.

40 pin address/data bus, dual row.

34 pin interface connector (keypad and display), dual row.

10 pin serial communications connector, dual row.

6 pin power connector, single row.

Communications: A hardware UART supports either RS232 or RS485 and a second software UART implements an RS232 interface.

A fast synchronous serial peripheral interface (SPI) is also provided.

Power: 100 mA average current consumption at +5 volts.

Onboard regulation converts 6-12VDC into +5V regulated 750 mA primary supply and 10 mA secondary supply used for analog circuitry. The excess capacity of this supply is available to peripheral devices.

Digital shutdown signal can turn off the primary supply, reducing power consumption to 20 mW.

Board can be directly powered by a regulated +5Volt supply.

Power monitor ensures orderly power up/power down.

Power fail early warning signal is generated onboard.

EMI filter and power surge protection is provided on the board.

Custom Configurations: Contact Mosaic Industries to discuss custom hardware configurations for your production volume orders.

QED-Forth

A Problem Solving Language

The Forth programming language is ideal for real-time control, instrumentation, and data analysis. Its simple yet powerful structure allows an interpreter and compiler to reside on the target hardware, giving the programmer the ability to interactively define new routines and immediately execute and debug them. This leads to an iterative programming style involving rapid definition, testing, and redefinition of small, modular routines tailored to the application.

Like the processors it runs on, Forth is stack based. Parameters are passed between routines on a push-down data stack. Stack-based parameter passing speeds program execution and unifies Forth syntax. All arithmetic, logical, I/O, and decision operators remove required data from the stack and leave their results on the stack. This leads to "postfix" syntax, with the operation being called after the data is placed on the stack.

QED-Forth™ is a powerful superset of the Forth language. It provides a comprehensive library of logical, computational, control, and I/O functions right on the QED Board.

A Robotics Program

To demonstrate the simplicity of Forth, let's look at the control of a robotic arm. The arm can move up and down and travel horizontally, and uses an electromagnet to move metallic objects from one pile to another. An object is picked up by positioning the arm just above the object and activating the electromagnet. It is then transported to the desired location and the electromagnet is de-activated to let go of the object.

We first define the following routines (called "words" in Forth) to control the basic actions of the robot:

```
GRAB            \ activate electromagnet
LET.GO          \ de-activate electromagnet
LOWER.ARM       \ activate vertical motor in - direction
RAISE.ARM       \ activate vertical motor in + direction
TRAVEL.TO       \ activate horizontal motor until destination is reached
```

The definitions of these words are simple. Each involves activating or de-activating an I/O signal that controls a magnet or motor. By combining these actions, we can automate the movement of objects.

Next let's define some locations expressed in the coordinate system of the robot. For example,

```
  0       CONSTANT  HOME.BASE
100       CONSTANT  FIRST.BASE
```

When HOME.BASE is later executed, it leaves its value, 0, on the stack. Likewise, FIRST.BASE leaves 100 on the stack.

We now define a robot control word that moves an object from one location to another:

```
: MOVE.OBJECT    ( origin\destination -- )
   LOCALS{ &destination  &origin }
   &origin TRAVEL.TO
   LOWER.ARM   GRAB    RAISE.ARM
   &destination TRAVEL.TO
   LOWER.ARM  LET.GO  RAISE.ARM  ;
```

The : (colon) tells the compiler that the word MOVE.OBJECT is to be added to the dictionary as a new definition. The comment at the end of the first line is a "stack picture" that reminds the programmer that this word removes two items from the data stack. The remainder of the commands up to the terminating ; (semicolon) form the body of the new definition. The LOCALS statement names the two input stack items as local variables to enhance the readability of the code. The action of the robot when MOVE.OBJECT is executed mirrors the definition: it travels to the starting point, lowers the arm, grabs the metallic object, raises the arm, travels to the destination, lowers the arm, lets go of the object, and raises the arm. The names of the fundamental motions were well chosen, so the code is virtually self-documenting.

Now that the definition has been compiled, we can simply state a starting point, a destination, and the acti

```
HOME.BASE   FIRST.BASE   MOVE.OBJECT
```

and QED-Forth will execute the command. We can watch the robot arm move the object and decide if it is functioning properly. If not, the code can be easily modified and re-compiled until it works correctly, and then higher level words can be constructed until the robotics application is complete.

Solving Simultaneous Equations the Easy Way

To demonstrate the computational power of QED-Forth, let's see how easy it is to solve a set of simultaneous equations such as:

$$
\begin{aligned}
1.07\,X_1 + 0.19\,X_2 + 0.23\,X_3 &= 2.98\\
0.38\,X_1 + 1.00\,X_2 + 0.74\,X_3 &= 3.48\\
0.12\,X_1 + 0.53\,X_2 + 1.20\,X_3 &= 3.70
\end{aligned}
$$

The equations can be represented by the matrix equation

$$C\,X = R$$

where C is the 3 by 3 matrix of coefficients, X is the 3 by 1 matrix of unknows (X_1,X_2,X_3) and R is the 3 by 1 matrix of Right-hand-side values. The following commented code solves the system of equations:

```
MATRIX: C       3 3 'C DIMMED           \ create and dimension coefficient matrix
MATRIX C =      1.07 0.19 0.23          \ initialize coefficients matrix
                0.38 1.00 0.74
                0.12 0.53 1.20

MATRIX: R       3 1 'R DIMMED           \ create and dimension right hand side
MATRIX R =      2.98 3.48 3.70          \ initialize right hand side matrix

MATRIX: X                               \ create the matrix of unknowns

'C 'R 'X  SOLVE.EQUATIONS               \ calculate the answer
'X  M..                                 \ print the solution
```

QED-Forth responds:

```
Matrix X =
2.099
0.8259
2.509
```

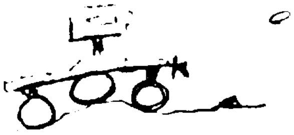
With just these few lines of code, we've solved the set of linear equations with floating point coefficients. Larger sets of equations with many more unknowns are handled with very similar code.

The Power of QED-Forth

These examples demonstrate the power of QED-Forth for control and data processing applications. But there's much more. The multitasking real-time operating system makes it easy to create modular applications comprising a number of cooperative tasks. The interactive on-board debugger supports program tracing, single stepping, break point insertion, and register content reporting. In addition, this integrated programming environment provides easy access to the QED Board's I/O-rich hardware. Pre-coded library routines simplify the use of the board's A/D and D/A converters, keypad, LCD display, and battery-backed real-time clock.

With QED-Forth, the fundamental capabilities that you need are already programmed for you, so you can concentrate on creatively solving the unique aspects of your application. The result is a powerful tool for solving instrumentation, automation, and embedded control problems.

Mosaic Industries, Inc.
5437 Central Avenue Suite 1, Newark, CA. 94560
(510) 790-1255



PROCESS WHILE WORKING

Multitasking: An integrated real-time operating system implements cooperative and/or preemptive (timesliced) multitasking with only 60 μsec task switch time. Resource variables and mailboxes facilitate inter-task communication and resource sharing.

Stand-alone Operation: The QED Board is easily configured to execute a user-defined program at power-up or reset using the QED-Forth command AUTOSTART.

Assembler: An integrated assembler with psuedo-high-level iteration and decision making macros simplifies coding of time-critical routines. Assembly code and high level QED-Forth code can be combined without restriction.

Symbolic Debugging and Trace: Any user-defined high level or assembly coded routine can be examined with the debugging tools. Single stepping, program tracing, register content reporting, and setting of software breakpoints make it easy to isolate bugs. The full power of QED-Forth is available in the debugging mode. For example, even if the user has suspended operation in the middle of a routine that is being debugged, memory contents can be verified and variables can be checked; simply typing two carriage returns from the terminal then resumes execution of the suspended routine where it left off.

Error Handling: QED-Forth messages are easily understood sentences. Custom error condition functions can be defined by the user.

Time Critical Coding Support: The BENCHMARK: routine calculates the execution time of any specified function.

Sample QED-Forth Benchmarks:
(8MHz crystal)

Operation	Typical execution time
F+, F-	0.170 msec
F*	0.160 msec
F/	0.150 msec
FSQRT	1.550 msec
FSIN	2.770 msec
FLN	2.480 msec
3x3 matrix multiply	0.017 sec
10x10 matrix inversion	0.800 sec
64 point complex FFT	0.600 sec
Task switch	60.00 μsec
Interupt latency	17.00 μsec
Interupt exit time	10.00 μsec

LONG TIME TO COMPLETE = 1½ FT/SEC

These execution times apply to QED Boards with 8MHz crystal frequencies. 16MHz QED Boards execute at twice this speed.

Mosaic Industries, Inc.
5437 Central Avenue Suite 1, Newark, CA. 94560
(510) 790-1255

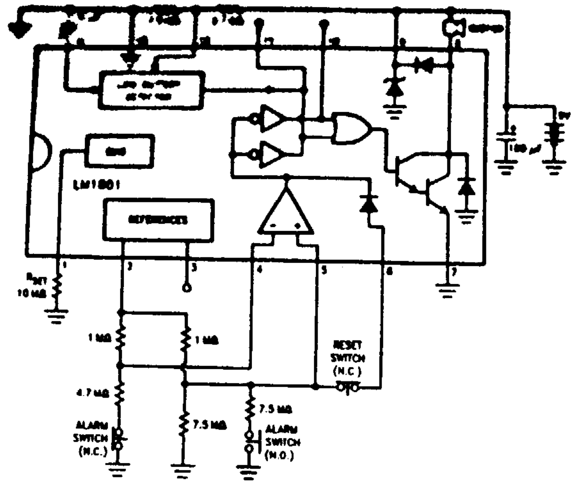


FIGURE 6. Simple Alarm Circuit

TL/H/8139-6

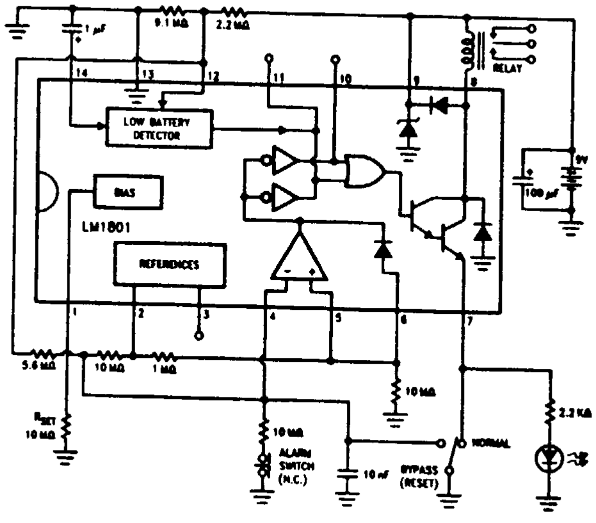


FIGURE 7. Full-Featured Intrusion Alarm

TL/H/8139-7

78

National Semiconductor

(5) 7, (10), 12
13, (15), 16

LM1812 Ultrasonic Transceiver

General Description

The LM1812 is a general purpose ultrasonic transceiver designed for use in a variety of ranging, sensing, and communications applications. The chip contains a pulse-modulated class C transmitter, a high gain receiver, a pulse modulation detector, and noise rejection circuitry.

A single LC network defines the operating frequency for both the transmitter and receiver. The class C transmitter output drives up to 1A (12W) peak at frequencies up to 325 kHz. The externally programmed receiver gain provides a detection sensitivity of 200 µVp-p. Detection circuitry included on-chip is capable of rejecting impulse noise with external programming. The detector output sinks up to 1A.

Applications include sonar systems, non-contact ranging, and acoustical data links, in both liquid and gas ambients.

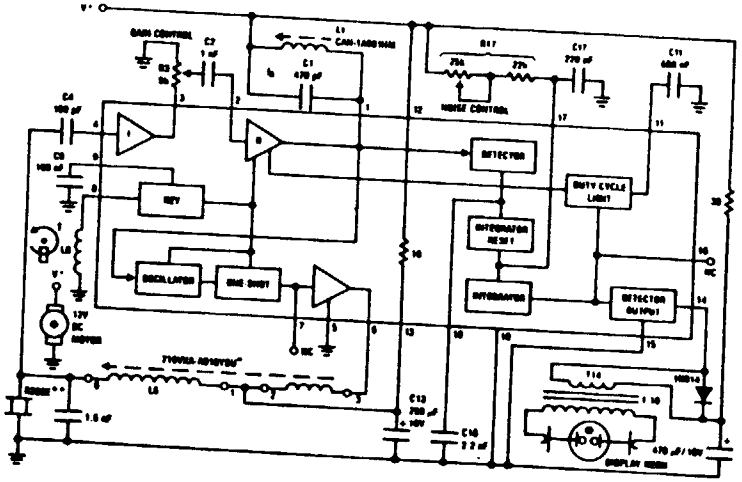
Features

- One or two-transducer operation
- Transducers interchangeable without realignment
- No external transistors
- Impulse noise rejection
- No heat sinking
- Protection circuitry included
- Detector output drives 1A peak load
- Ranges in excess of 100 feet in water, 20 feet in air
- 12W peak transmit power

Applications

- Liquid level measurement
- Sonar
- Surface profiling
- Data links
- Hydroacoustic communications
- Non-contact sensing
- Industrial process control

Typical Application



Order Number LM1812N
See NS Package Number N18A

TL/H/7882-1

†Note: Echo returns are displayed by a neon lamp on a motor driven disc. Connections to the neon are made through brushes and slip rings. Rotating with and counterbalancing the neon lamp is a permanent magnet whose field induces a pulse in a stationary coil (L8) as it passes by. This pulse keys the LM1812's transmitter.

*Available from Toko America, 1250 Feehanville Drive, Mount Prospect, Illinois 60056 Tel. (312) 297-0070

**Available from Massa Products Corporation, 280 Lincoln Street, Hingham, Massachusetts 02043 Tel. (617) 749-4800

FIGURE 1. 200 kHz Depth Sounder, 5 Feet to 100 Feet

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage, V+ (Pin 12)	18V
Power Dissipation (Note 1)	1700 mW
Peak Current (Pins 6, 14)	1A
Input Current (Pins 4, 8)	50 mA
Operating Temperature	0°C to +70°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	260°C

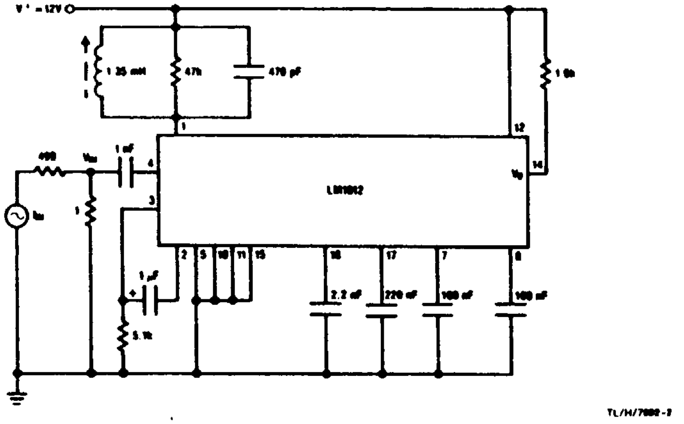
Electrical Characteristics $V^+ = 12V$, $T_A = 25°C$, unless otherwise noted

Parameter	Conditions	Min	Typ	Max	Units
Input Sensitivity (Note 2)	*Figure 2*		200	600	μVp-p
Input Noise			10		μVp-p
Transmitter Output, V_{SAT}	I_6 = 1A		1.3	3	V
Transmitter Output Leakage	V6 = 36V V8 = 0V		0.01	1	mA
Detector Output, V_{SAT}	I_{14} = 1A		1.5	3	V
Detector Output Leakage	V14 = 36V		0.01	1	mA
Transmitter Key Threshold	I_8 = 1 mA	0.55	0.7	0.9	V
Supply Current	$I_1 + I_{12}$ Receive Mode	5	8.5	20	mA
V8 for Receive Mode				0.3	V
Maximum Operating Frequency	Transmit Mode	200	325		kHz

Note 1: For operating at high temperatures, the LM1812 must be derated based upon a 125°C maximum junction temperature and a thermal resistance of 58°C C/W which applies for the device soldered in a printed circuit board and operating in a still air ambient. Due to the low duty cycle operation, only a small average power is dissipated in the package.

Note 2: A 47k resistor is added in parallel with the receiver tank at pin 1 to swamp variations in the coil's unloaded Q. The resistor reduces sensitivity (see equation 4) and is unnecessary in an actual applications circuit.

Test Circuit



TL/H/7992-2

f_{IN} 200 kHz

Input sensitivity = minimum V_{IN} for V_O to go low

FIGURE 2. Sensitivity Test Circuit

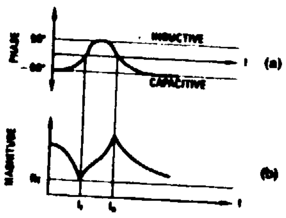
Application Hints

External Component Descriptions

Pin	Component	Typical Values	Pin Description	Component Function
1	L1, C1	500 μH–50 mH 250 pF–2.2 nF	Second gain stage output/ transmitter oscillator	Set the operating frequency (f_O) for the transmit oscillator and receiver
2	C2	500 pF–10 nF	Second gain stage input	Couples first and second gain stage
3	R3	5.1 kΩ	First gain stage output	Terminates emitter-follower output
4	C4	100 pF–10 nF	First gain stage input	Input coupling for the first gain stage
6	L6	50 μH–10 mH	Transmitter output	Matches LM1812 to the transducer
7	—	—	Transmitter driver	—
8	R8	1 kΩ–10 kΩ	Transmitter key	Current limiter for keying pulses up to 12V
9	C9	100 nF–10 μF	Receiver second stage delay	Sets the receiver turn-on delay after transmit (*Figure 10*)
11	C11	220 nF–2.2 μF	Detector output duty cycle limit	Limits the duty cycle of the detector output (short to ground to defeat)
13	C13	100 μF–1000 μF	Transmitter supply decoupling	Decouples the transmitter power supply
14	T14	$L_p \geq 50$ mH $N_S/N_P = 10$	Detector output	Drives neon display lamp
16	—	—	Output driver	—
17	R17, C17	22k–Open 10 nF–10 μF	Pulse integrator	Controls integration time constant (*Figure 13*)
18	C18	1 nF–100 μF	Pulse integrator reset	Controls integrator reset time constant (*Figure 14*)

TRANSDUCERS

The most common transducer used with the LM1812 is the piezo-ceramic type which is electrically similar to a quartz crystal. Piezo-ceramic transducers are resistive at only two frequencies, termed the resonant and antiresonant (f_r, f_a) frequencies. Elsewhere these transducers exhibit some reactance as shown in *Figure 3*.



TL/H/7992-3

FIGURE 3. Phase and Magnitude of Transducer Impedance

For transmitting (to maximize electrical to mechanical efficiency), the transducer should be operated at its resonant frequency. For receiving (to maximize mechanical to electrical efficiency), optimum operation is at antiresonance. In two-transducer systems the resonant frequency of the transmit transducer is matched to the antiresonant frequency of the receiver.

The LM1812 is primarily used with a single transducer performing both transmit and receive functions. In this mode, maximum echo sensitivity will occur at a frequency close to resonance.

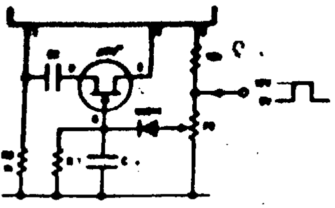
Transducer ringing is a troublesome phenomenon of single transducer systems. After a transducer has been electrically driven in the transmit mode, some time is required for the mechanical vibrations to stop. Depending on the amount of damping, this ringing may last from 10 to 1000 cycles. This mechanical ring produces an electrical signal strong enough (>200 μVp-p) to hold the detector ON, thus masking any echo signals occurring during this time.

A solution to this ring problem is to vary the receiver gain from a minimum, just after transmit, to a maximum, when the ring signal has dropped below the full-gain detection threshold. Since near-range echo signals are much stronger than ring signals, close echos will still be detected in spite of the reduced gain.

The gain is varied by attenuating the signal between pins 2 and 3 of the LM1812. *Figure 4* shows such an arrangement. An externally generated 12V pulse (*Figure 17*) keys the transmitter and activates the attenuator. This pulse charges C to a voltage set by P8, turning the FET OFF. C slowly discharges through R, decreasing the gate voltage, which in turn decreases the attenuation of the signal passing from pin 3 to pin 2. R and C are selected so that the FET is not

88

Application Hints (Continued)



TL/H/7882-4

*Available from National Semiconductor Corporation

FIGURE 4. Time Variable FET Attenuator

completely turned ON until all detectable ringing has stopped. The duration of the ring is rarely specified by the transducer manufacturer and must be experimentally determined.

When designing an ultrasonic ranging system, three *transducer* parameters are very important:

1) resonant impedance (R_T in *Figure 3b*)

2) maximum peak-to-peak voltage

3) resonant frequency, f_r

This data, used in conjunction with the curves given in *Figure 6*, results in a functional output stage design.

TRANSMITTER

The transmitter (*Figure 5*) consists of an oscillator, a 1 µs one-shot, and a power amplifier.

When the transmitter is keyed ON at pin 8 the L1-C1 tank is switched to the oscillator mode. An on-chip 1 µs one-shot is triggered with each cycle of the oscillator and, in turn, drives a power amplifier. This one-shot has a reset time of 2 µs, limiting the maximum operating frequency to about 325 kHz. A transformer couples the transducer to the output stage.

The oscillator frequency is set by L1-C1 and can be calculated from

$$f_O = \frac{1}{2\pi\sqrt{L1C1}} \tag{1}$$

The L1-C1 tank must have a minimum R_p of 10 kΩ where

$$R_p = 2\pi f_O Q L1 \tag{2}$$

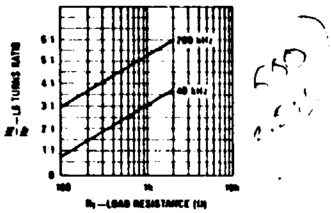
and Q = unloaded Q of L1-C1 tank.

The output transformer (L6) is designed with the aid of *Figure 6*. Curves are shown for two common frequencies: 40 kHz and 200 kHz. For a given load impedance (R_T, *Figure 3b*), a turns ratio for L6 is determined. In order not to exceed the transducer's specifications, the peak-to-peak output voltage may need to be adjusted using the equation:

$$V_{p\text{-}p} = 2V^+ \left(\frac{N_S}{N_P}\right) \tag{3}$$

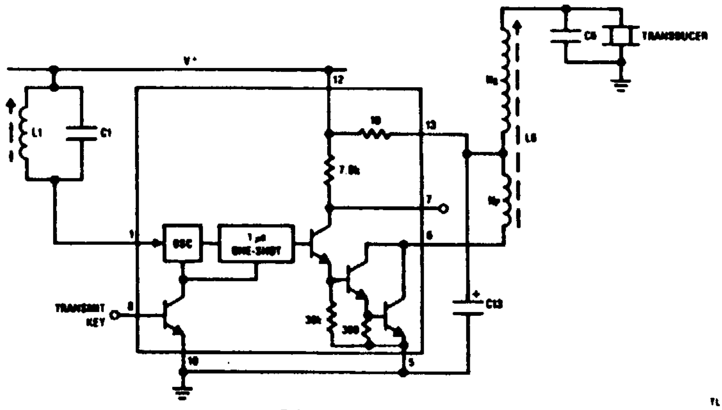
To ensure that the output stage is not overloaded, a current measurement must be made at pin 6. While the first few pulses of each transmit period may reach 2A or 3A, the steady-state current spikes must not exceed 1A. Current spikes are reduced by decreasing the turns ratio of L6.

The secondary of L6 tunes with C6 at the operating frequency, f_O.



TL/H/7882-5

FIGURE 6. L6 Turns Ratio vs Load Resistance



TL/H/7882-6

FIGURE 5. Transmitter

Where additional power is desired, a pulse amplifier or a pulse stretcher can be used as shown in *Figure 7*. The pulse amplifier (*Figure 7a*) increases output current up to 5A. The pulse stretcher (*Figure 7b*) increases output current *and* pulse width. The wider pulse of *Figure 7b* is especially useful at lower frequencies where the relatively narrow 1 µs pulse creates a large peak current demand for a given power level. Pulse width as a function of R is plotted in *Figure 8*.

Pin 8 performs the function of switching the LM1812 into either the transmit or receive mode. When pin 8 is held high, the chip is in the transmit mode. When held low, it is in the receive mode. The input current at pin 8 should be designed to operate within a 1 mA–10 mA range.

RECEIVER

The receiver section (*Figure 9*) contains two seperate gain stages.

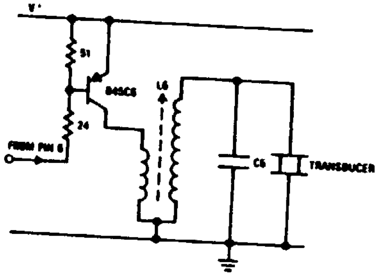
In some applications large voltages are applied across the transducer during transmit. Since the receiver input is coupled to the transducer, some protection is necessary to limit the input current spikes to less than 50 mA. Where the voltage across the transducer is less than 200 Vp-p, a C4 reactance of 5 kΩ at the operating frequency is adequate protection. Above 200 Vp-p, a 5 kΩ resistor should be inserted in series with C4.

Since the L1-C1 tank circuit is shared with the oscillator, both the transmitter and receiver are always tuned to the same frequency. The second stage voltage gain is given by:

$$A_V = \frac{Q}{70}\sqrt{\frac{L1}{C1}} \tag{4}$$

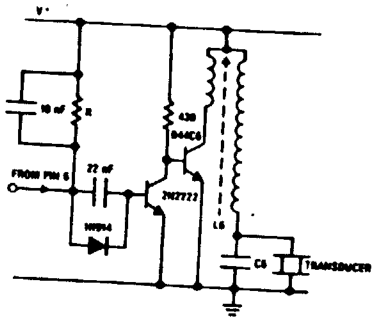
where Q = unloaded Q of L1-C1 tank

When the LM1812 is in the transmit mode, the second gain stage is turned OFF. When switching back to the receive mode, the gain stage does not turn ON immediately, but instead turns ON after a slight delay as programmed by C9. This delay blanks the receiver (and therefore the detector) momentarily, giving the transducer time to stop ringing.



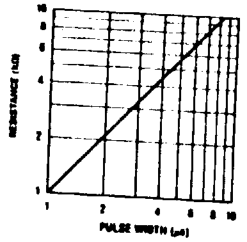
TL/H/7882-7

FIGURE 7a. Pulse Amplifier



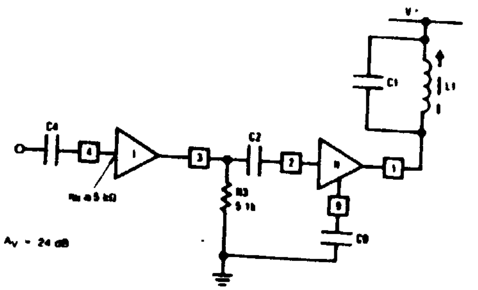
TL/H/7882-8

FIGURE 7b. Pulse Stretcher



TL/H/7882-9

FIGURE 8. Pulse Stretcher Resistance vs Pulse Width



TL/H/7882

FIGURE 9. Receiver Section

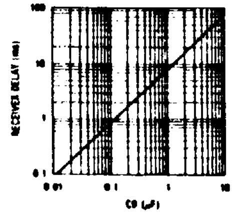
Application Hints (Continued)

Delay as a function of C9 is plotted in Figure 10. The second gain stage may be shut OFF independently of pin 9 by pulling pin 9 low.

Due to the high gain of the receiver, care must be taken to avoid oscillations. Oscillation problems are reduced by keeping the components associated with pins 1 and 4 well seperated (*Figure 11*). The transducer must be connected to the circuit with shielded cable. This not only helps avoid oscillation, but also reduces electrical noise pick-up. As a last resort, receiver gain can be reduced with R3 as in *Figure 1*.

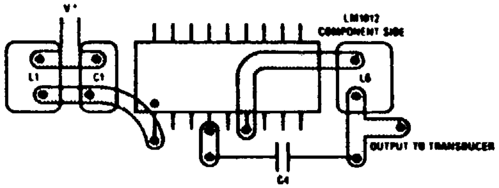
PULSE DETECTOR

The pulse detector circuitry (*Figure 12*) consists of five distinct stages: 1) threshold detector, 2) pulse integrator reset, 3) pulse integrator, 4) output driver, 5) power output stage. The detector (Q1, Q2) switches on all pin 1 signals that exceed 1.4 Vp-p. Since noise pulses are also detected, filtering is done by an integrator stage, C17 and R17, whose time constant is typically 10% to 50% of the transmit time. Integration starts when Q3 turns OFF, which occurs at the same moment Q1 and Q2 detect a signal. Pins 16 and 14 go low after the integration delay.



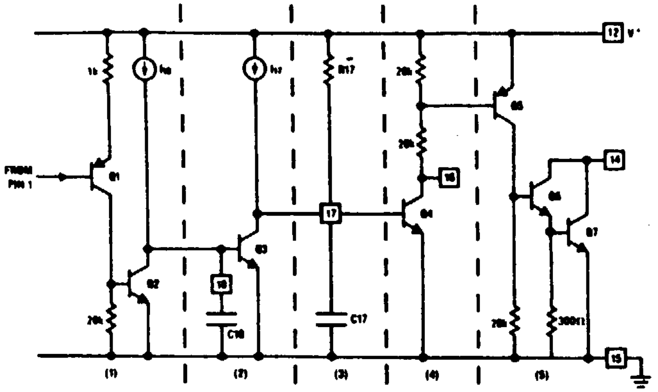
TL/H/7882-11

FIGURE 10. Receiver Delay vs C9



TL/H/7882-12

FIGURE 11. Component Side of Layout Showing Isolation of Receiver Input and Output



TL/H/7882-13

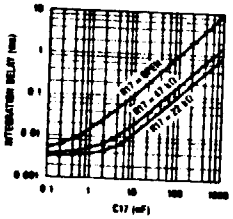
FIGURE 12. Simplified Circuit Diagram of Detector

When the voltage at pin 1 becomes too small to activate the detector (< 1.4 Vp-p), the integrator is reset by Q3 after a delay introduced by C18. A delay of 1 to 10 cycles of the transmitted frequency is typical. These integration and reset delays, as a function of the external component values, are shown in *Figures 13* and *14*.

Pin 16 provides a CMOS compatible logic output. For driving high-intensity displays, pin 14 will sink up to 1A. When driving a transformer such as T14 in *Figure 1*, it is possible for the primary current to integrate up to destructive levels under conditions of multiple echo reception. Pin 11 is employed to protect the power output (pin 14). C11 integrates an internal current source while pin 14 is low. When V11 reaches a 0.7V threshold, the second gain stage is turned OFF. With the receiver OFF, no signal will be applied to the detector, and pin 14 will turn OFF. After another delay C11 is discharged and the receiver is then again activated. With C11 = 680 nF and a continuous echo return, the receiver will cycle ON and OFF every 6 ms. This function can be defeated by grounding pin 11.

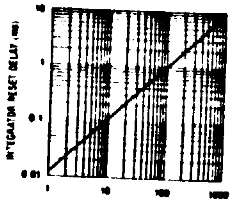
TYPICAL OPERATION

Figure 15 shows typical waveforms at pins 1 and 16 for 200 kHz operation, with pin 9 left open. The pin 1 oscillator signal (5 Vp-p) lasts for 200 µs. The next 900 µs show a ring signal so strong that it is clipped by the receiver. The exponential nature of the decaying ring is seen for the next 500 µs. An echo return appears at 3.9 ms. Note that the detector is held low during the transmit period and for the duration of the ring.



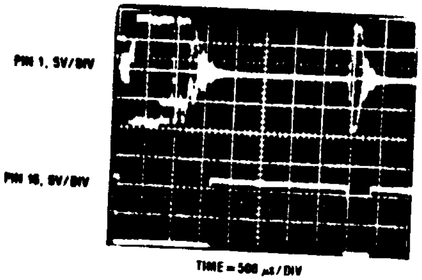
TL/H/7882-14

FIGURE 13. Integration Delay vs C17



TL/H/7882-15

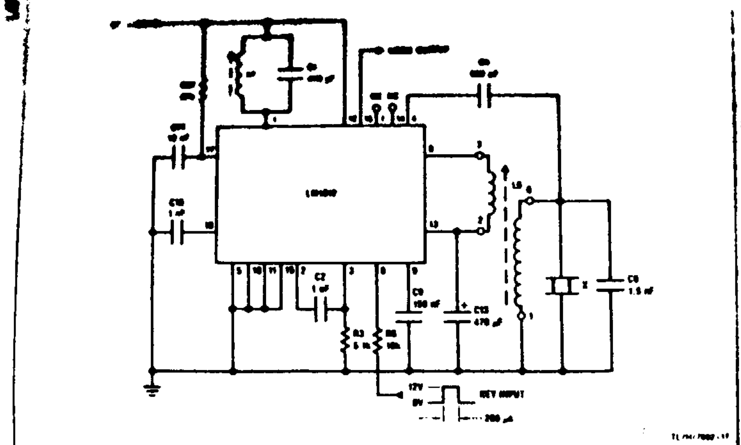
FIGURE 14. Integrator Reset Delay vs C18



TL/H/7882-16

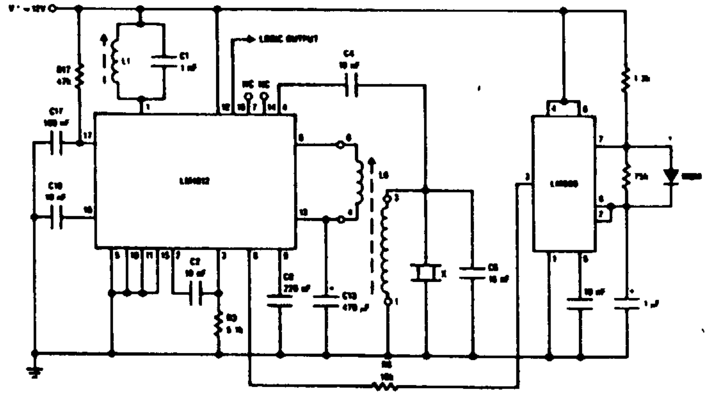
FIGURE 15. Typical Transmit/Receive Waveforms

89



L1 - CAN 1A801HM (Toko)
L6 - 719VXA A018YSU (Toko)
X - R283E (Massa Products)

FIGURE 16. 200 kHz Ultrasonic Ranging System for 4 Inches to 6 Feet in Air



L1 - CLN-2A880HM (Toko)
L6 - 719VXA A017AD (Toko)
X - EFR-OTB40K2 (Available from Panasonic Company, 1 Panasonic Way, Secaucus, NJ 07094. Tel (201) 392-4651)

FIGURE 17. 40 kHz Ultrasonic Ranging System Covering a Range of 3 Feet to 20 Feet

National Semiconductor

LM1815 Adaptive Sense Amplifier

General Description

The LM1815 is an adaptive sense amplifier and default gating circuit for motor control applications. The sense amplifier provides a one-shot pulse output whose leading edge coincides with the negative-going zero crossing of a ground referenced input signal such as from a variable reluctance magnetic pick-up coil.

In normal operation, this timing reference signal is processed (delayed) externally and returned to the LM1815. A logic input is then able to select either the timing reference or the processed signal for transmission to the output driver stage.

The adaptive sense amplifier operates with a positive-going threshold which is derived by peak detecting the incoming signal and dividing this down. Thus the input hysteresis varies with input signal amplitude. This enables the circuit to sense in situations where the high speed noise is greater than the low speed signal amplitude. Minimum input signal is 100 mVp-p.

Features

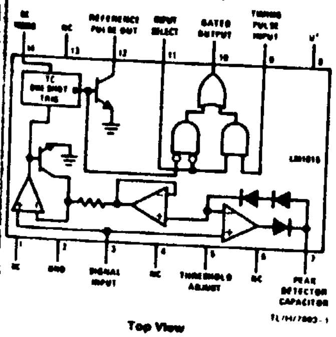
- Adaptive hysteresis
- Single supply operation
- Ground referenced input
- True zero crossing timing reference
- Operates from 2V to 12V supply voltage
- Handles inputs from 100 mV to over 120V with external resistor
- CMOS compatible logic

Applications

- Position sensing with notched wheels
- Zero crossing switch
- Motor speed control
- Tachometer
- Engine testing

Connection Diagram

Dual-In-Line Package



Top View

Order Number LM1815N
See NS Package Number N14A

Truth Table

Signal Input	Input Select	Timing Input	Gated Output
Pulses	L	X	Pulses
X	H	Pulses	Pulses

APPENDIX I - SOIL SPECIFICATIONS

The test environment located a NASA's Ames Research Center tries to copy what is on the moon. The environment that we built building in the Gauss Mechanical Engineering Laboratory at the University of Idaho is similar to that of Ames. In the fall of 1992 a test container was built and mass calculations of environment material were completed. It was determined that 1500 pounds of basalt sand were necessary to fill the container. The test environment is made of different grit sizes of basalt sand, The composition of which follows:

Screen Number	Moon Composition (Percent of Total)	Ames Composition (Percent of Total)	Mass Needed (lbm.)
8-16	5	10	150
16-30	3	6	90
30-50	10	20	300
50-100	12	24	360
100-200	20	40	600
200+	50	0	000
Totals	100	100	**1500**

The screen numbers shown are a range of screen numbers. For example, of the moon's surface composition, twelve percent of it is made up of material that is in the range of a 50 to a 100 screen. The numbers indicate how many holes there are in the screen per square inch. The reason that Ames Research Center and University of Idaho are not using the 200+ screens is because the particles that are obtained from this screen are so small that they are more trouble in the environment (too many dust problems) than they are worth.

APPENDIX J - POWER SYSTEM HEAT TRANSFER ANALYSIS

For a horizontal cylinder: L = 0.049 meters; Incropera & Dewitt equation 9.34;

$$\bar{h} = \frac{k\overline{Nu}}{D} = \frac{k}{D}\left[0.6 + \frac{0.387 Ra_D^{1/6}}{(1+(\frac{0.559}{\text{Pr}})^{9/16})^{8/27}}\right]^2$$

$$Ra_D = \frac{g\text{B}(T_s - T_\infty)D^3}{\nu D}$$

@ 300 Kelvin:

k = 26.3×10^{-3} W/mK; $\alpha = 22.5\times10^{-6}$ m^2/sec; $\nu = 15.89\times10^{-6}$ m^2/sec; Pr = 0.707; D = 0.114 inch

$Ra_D = 135505.19\times(T_s - T_\infty)$

$$\bar{h} = 0.231\left(0.6 + 0.3212 \bullet Ra^{1/6}\right)^2$$

ΔT	Ra_D	$\bar{h}(\frac{W}{mk})$
30	4065156	5.01
20	2710103.8	4.46
10	1355051.9	3.657
5	677525.95	3.01

For batteries in rover wheels: Incropera & Dewitt equation 9.58;

$$q' = \frac{2\pi k_{eff}}{\ln(\frac{D_o}{D_i})}(T_i - T_o)$$

$$Ra_c = \frac{\left(\ln(\frac{D_o}{D_i})\right)^4}{\left(D_i^{-3/5} + D_o^{-3/5}\right)^5} \bullet \frac{g\text{B}(T_i - T_o)}{\nu\alpha}$$

D_o = 0.108 meters; D_i = 0.0252 meters

$$Ra_c = \frac{\left(\ln(\frac{.108}{.0252})\right)^4}{\left(.108^{-3/5} + .0252^{-3/5}\right)^5} \bullet \frac{(9.81)(1/300)(T_i - T_o)}{(15.89\times10^{-6})(22.5\times10^{-6})}$$

$\mathrm{Ra}_c = 114665 \times (T_i - T_o)$

$$k_{eff} = 0.386k\left(\frac{\Pr}{0.861+\Pr}\right)^{1/4}\left(1146.65 \times (T_i - T_o)\right)^{1/4}$$

$$k_{eff} = (0.386)(26.3\times10^{-3})\left(\frac{.707}{0.861+.707}\right)^{1/4}(5.819)(T_i - T_o)^{1/4}$$

$$k_{eff} = 0.04841(T_i - T_o)^{1/4}$$

$$q' = \frac{2\pi(.04841)(T_i - T_o)^{5/4}}{\ln\left(\frac{.108}{.0252}\right)}$$

$$q' = .209(T_i - T_o)^{5/4}$$

$$\frac{T_o - T_\infty}{\frac{1}{h_o A}} = q$$

$$T_o = \frac{q}{h_o A} + T_\infty$$

$$q' = 0.209\left(T_i - \frac{q}{hA} - T_\infty\right)^{5/4}$$

$A = \pi(.108\text{ m})(.049\text{ m}) = 0.0166\text{ m}^2$

$R = 5.5\text{ m}\Omega$

$q = I^2R = (5)(0.5\text{ A})^2(5.5\text{ m}\Omega) = 0.007$ Watts

$q' = (0.007\text{ W})/(0.049\text{ m}) = 0.140$ Watts per meter

$$0.140 = .209\left(T_i - \frac{0.007}{(3)(.1066)} - 293\right)^{5/4}$$

T_i = 20.9 °C ⇒ Temperature of Battery in Wheel

For Charger Circuit: Incropera & Dewitt equation 7.50;

$q'' = h_{min} \times (T_E - T_\infty)$

$$h_x = \frac{k}{L} \times 0.0308 \times \mathrm{Re}^{4/5}\,\Pr^{1/3}$$

$T_f = 300$ Kelvin

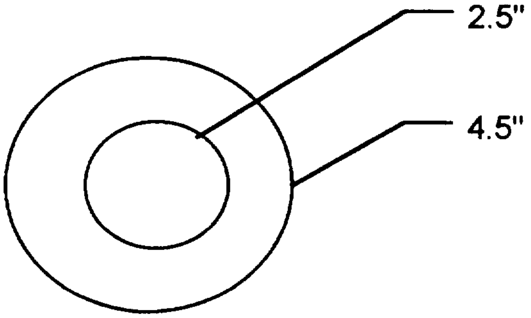
$$\frac{q''}{(T_E - T_\infty)} = \frac{k}{L} \times 0.0308 \times \mathrm{Re}^{4/5}\,\mathrm{Pr}^{1/3}$$

$$\mathrm{Re} = \left(\frac{q'' \times L}{(T_E - T_\infty) \times k \times 0.0308 \times \mathrm{Pr}^{1/3}}\right)^{5/4}$$

$$V = \frac{\nu}{L} \times \left(\frac{q'' \times L}{(T_w - T_\infty) \times k \times 0.0308 \times \mathrm{Pr}^{1/3}}\right)^{5/4}$$; where $(T_W - T_\infty) = \Delta T$

q = 80 Watts; $\Delta T = 40$, where T_W = circuit temperature and T_∞ = ambient temperature

FAN



Circuit Board Surface Area (square inches)	Volume Flow Rate (CFM)
48	207
72	126.7
96	87
144	52.6

(Fan Rating, where CFM = cubic feet per minute) ⇑

q = 40 Watts; $\Delta T = 40$; $Q = V \times A$, where Q = Volume Flow Rate, V = Volume and A = Area

Circuit Board Surface Area (square inches)	Volume Flow Rate (CFM)
48	87.3
72	53.3
93	36.7
144	22.1

(Fan Rating, where CFM = cubic feet per minute) ⇑

APPENDIX K - SAFETY ANALYSIS OF CHARGER AND CHARGING PROCESS

The following safety analysis was undertaken to help quantify and rank the possible failure modes of the charging system. First, the frequency of occurrence of each hazard is listed, and a frequency value of A, B, C, D, or E is assigned to show how often each hazard occurs. Next, the consequence of each hazard is listed, and a value of I, II, III, or IV is assigned to that particular hazard to show the severity of the outcome of each hazard. Finally, The frequency and consequence of each hazard are related by a number. These numbers (shown in Table C-1), called hazard risk indexes, are given by Military Standard 882B.

Table C-1. Hazard assessment matrix.

FAILURE MODE	FAILURE EFFECTS	INDEX	ACTION
Electric shock	Death	D, I: 8	Insulate and ground
Handle failure	Crushed toes, damaged equipment	D, II: 10	Carry with two hands
Foreign object in fan	Hurt fingers, broken fan	C, III: 11	Fan gaurd
Blockage of air flow	Electronics overheating/failure	B, II: 5	Leave space for airflow
Burn yourself	Burned fingers	D, III: 14	Avoid electronics contact
Plastic PCB holder failure	Damage to circuit board, shock	D, II: 10	Place on level surface
Trip over cord	Damage charger, minor abrasions	B, III: 9	Tape cords down
Water damage	Electric shock, shorting	C, II: 6	Don't place liquid near
Display damage	Charger malfunction	C, III: 11	Protect display panel
Improper removal of cover	Electrical shock, charger damage	D, I: 8	Unplug when service

<u>Frequency</u>

A. Frequent
B. Probable
C. Occasional
D. Remote
E. Improbable

<u>Hazard Category</u>

I. Catastrophic
II. Critical
III. Marginal
IV Negligible

<u>Hazard Risk Index</u>

1-5 Unacceptable Situation
6-9 Undesirable Situation
10-17 Acceptable with Review
18-20 Acceptable without Review

APPENDIX L - CHARGING PROCEDURE

Figure C-1 shows the schematics of the charger and the rover as well as the terminology used in this procedure.

Step 1:
Plug the charger power cord (in back) into a standard 120 Volt wall outlet.

Step 2:
Disconnect the 12 pin battery connector cord from the rover box.

Step 3:
Plug in the rover recharging cord to the 12 pin battery connector cord. Make sure connection is secure.

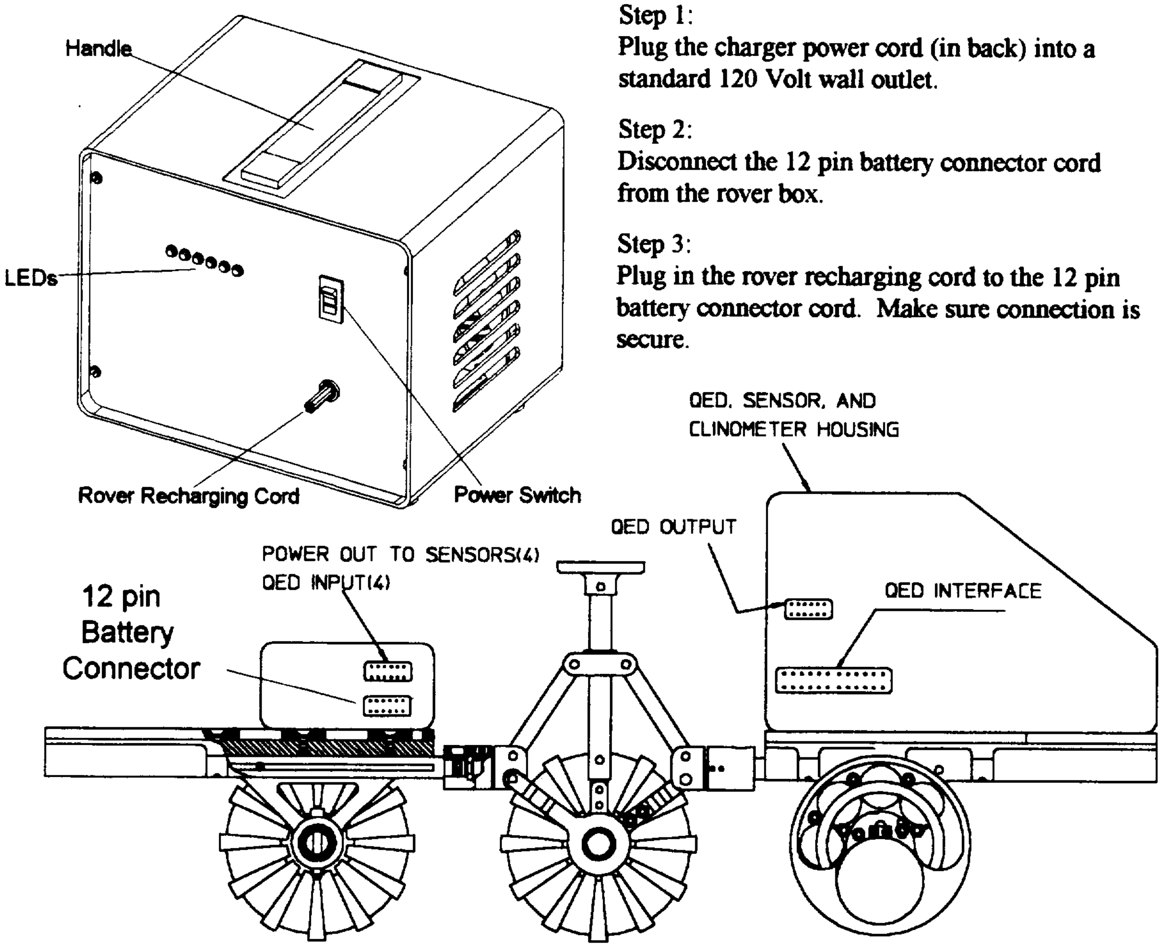


Figure C-1. Charger and Rover connection schematics.

Step 4:
Turn the power switch to the on position. (Switch will illuminate when it is in the on position.) All six LED's should also be lit. The operator can now leave the charger unattended.

WARNING: DO NOT OBSTRUCT AIRFLOW THROUGH LOUVERS OR FAN.

The charger will stop charging automatically when the batteries are at full capacity. Wait approximately six hours until all of the LED's turn off. This signals that the batteries are completely charged. Turn the power switch to the off position, disconnect the rover charging cord and reconnect the 12 pin battery connector cord.

APPENDIX M - FINAL DRAWINGS

Provided Upon Request