RB5X: MISSION TO MARS

AN INTERPLANETARY ADVENTURE FOR ALL GRADE LEVELS

Student Manual



GENERAL ROBOTICS CORPORATION

1978 South Garrison Street, Unit 6 ■ Lakewood, Colorado 80227-2243

BLANK PAGE

RB5X: MISSION TO MARS

An Interplanetary Adventure for all Grade Levels

Student Manual

Version 1.0, May 2000
General Robotics Corporation

General Robotics Corporation
1978 South Garrison Street, Suite 6
Lakewood, Colorado 80227-2243

Toll-Free: 800.422.4265
Local: 303.988.5636
Fax: 303.988.5303

Information: rb5x@generalrobotics.com
Technical Support: support@generalrobotics.com

Products & Corporate Information: www.generalrobotics.com
The RB5X and Education: www.edurobot.com

"RB5X," "RB5X The Intelligent Robot" and "RCL" (Robot Control Language) are trademarks of General Robotics Corporation.
"NSC Tiny Basic" is a trademark of National Semiconductor Corporation.

COVER PHOTO: The RB5X robot is superimposed over the desert landscape of Chryse Planitia, latitude 25N, longitude 45W, on Mars. In an effort to make the image realistic, RB5X's shadow, and the glare on its dome, have been adjusted for the position of the Sun at the time the photo was taken. This is how RB5X would really appear on the surface of Mars!

The ground in this area is strewn with rocks ejected from volcanoes, and has numerous small sand dunes that shift daily with the winds. The large rock, nicknamed "Big Joe," is about 2 meters in diameter. This image (minus RB5X, of course) was radioed back to Earth by the Viking 1 Lander on June 21st, 1976. Photo courtesy of NASA/JPL.

About the Author

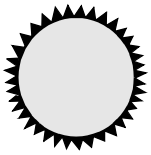
Dan Heim is a freelance consultant in educational multimedia and energy efficiency, and writes instructional materials in the fields of astronomy, physics and ecology. As a teacher of science for 25 years he enjoys a practiced insight into how students learn. That experience and insight was essential to the creation of RB5X: Mission to Mars.

Dan and his wife Sandi live at Heimhenge, a "home of the future" located in the Sonoran desert near New River, Arizona. Earth-sheltered in the foothills of rim country, their home incorporates passive solar design, photovoltaic electricity, high-tech insulation and energy-efficient appliances. The HVAC energy demand at Heimhenge is half that of a conventional structure.

RB5X: Mission to Mars was produced on a computer powered by free solar energy. All electrons were recycled, and no living creatures were harmed in the process.

They don't yet have a robot butler, but when they do it will be a solar-powered RB5X. Until then, Dan will continue answering his email personally at: heimhenge@compuserve.com.

May 2000



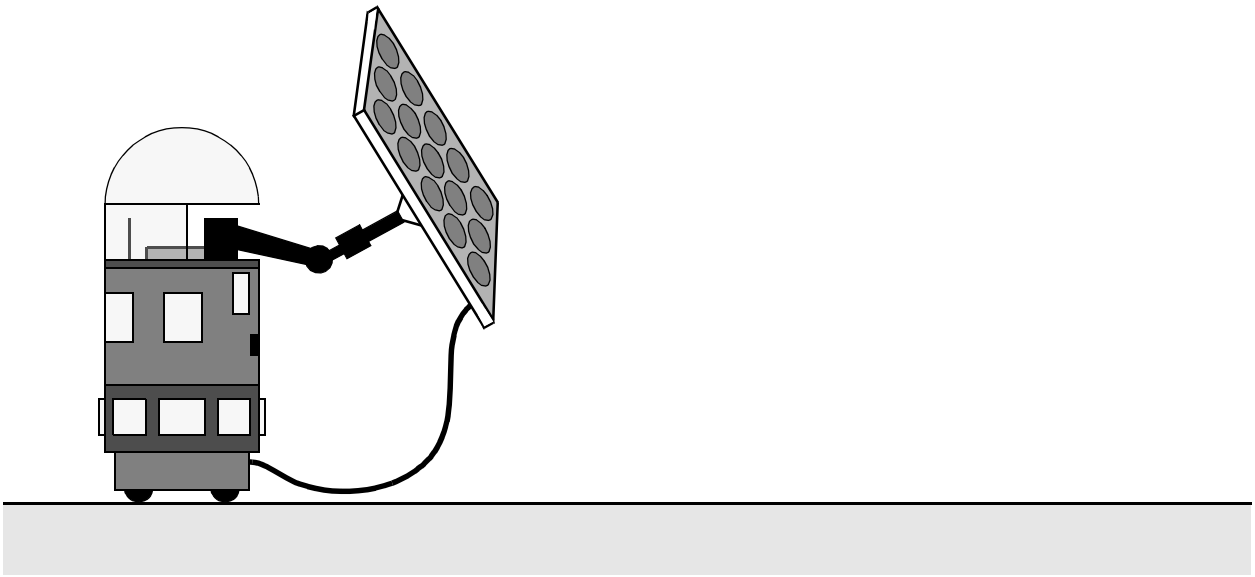


Table of Contents

		page
Mission Briefing		**7**
Mission 1:	Creating the Mars Diorama	**9**
Mission 2:	Robot Calibration	**12**
Mission 3:	Mapping	**14**
Mission 4:	Hazard Avoidance	**16**
Mission 5:	Navigation and Piloting	**18**
Mission 6:	Energy Management	**21**
Mission 7:	Sample Acquisition	**23**
Appendix 1:	Sample Diorama	**27**
Appendix 2:	The Heuristic Method	**28**
Appendix 3:	Mission to Mars Crossword Puzzle	**29**
Appendix 4:	Data for the Planets	**32**
Appendix 5:	Global View of Mars	**34**
Appendix 6:	The Surface of Mars at Ares Vallis	**35**
Appendix 7:	A Short History of Robotic Exploration	**36**
	RB5X on Terra Firma	**37**
	Sojourner Rover	**38**
	Venera	**39**
	Viking	**40**
	Lunokhod	**41**
	Luna (Late Series)	**42**
	Surveyor	**43**
	Ranger	**44**
	Luna (Late Series)	**45**
Appendix 8:	URL's You Can Use	**47**
Appendix 9:	Glossary	**48**
Appendix 10:	Programming Your RB5X	**52**
Appendix 11:	Calibrating Your RB5X	**66**

NOTE: Page numbers run sequentially from cover to end, with the cover as page 1, and facing blank pages in the printed manual also reproduced in the online manual. Should you decide to use both versions, you'll be pleased to find identical numbers on corresponding pages.

BLANK PAGE

Mission Briefing

Introduction:

NASA's recent success with Mars Pathfinder and Sojourner proved that robotic exploration of distant worlds, once the realm of science fiction, is now a valuable tool of science. Earlier missions to the Moon and planets used machines in place of astronauts quite effectively. The Viking landers on Mars (1976) had working arms that allowed them to do experiments in *exogeology* and *exobiology*. But these earlier robots lacked the mobility and *autonomy* of the Sojourner Rover. Sojourner was controlled from a computer workstation at JPL. The mission "pilots" drove Sojourner using a joystick and keyboard while "seeing" on the monitor through the robot's camera eyes. This type of remote control is known as *teleoperation* with *telepresence*, since the pilots see and reacts as though they were actually there. This is not to be confused with *virtual reality* systems; the world on the other end of the line at JPL was reality.

Your RB5X can duplicate many of the functions of Sojourner and perform a few additional tasks as well. In this series of Missions your class will set up a simulated Mars landscape for RB5X to explore. Along the way you will learn about the nuts and bolts of remote robotic exploration, do some real creative problem-solving and become intimately familiar with one of our nearest neighbors in space — planet Mars.

Assigned Missions:

Creating the Mars Diorama (an exercise in modeling and exogeology)
Robot Calibration (engineering and applied mathematics)
Mapping (exploring and charting unknown territory)
Navigation and Piloting (using maps and teleoperation)
Hazard Avoidance (safely negotiating a hostile environment)
Energy Management (planning exploration within energy constraints)
Sample Acquisition (identifying and retrieving rock samples)

Required Materials:

Your RB5X is ready and willing to assist you in this simulated exploration of Mars. Grade levels K-12 will need no special materials other than a few inexpensive common household items. For post-secondary classes, access to basic tools and construction materials is required. Missions are designed for the base RB5X package and do not require the purchase of any RB5X accessories. If you have the voice/speech synthesis option, *RF* com-link or video camera, or robotic arm, you will find relate exercises and suggested applications included in most Missions.

Caution!

Don't ignore these sidebars. They will tell you things you want to know, didn't know, or absolutely need to know ... just like the three you see on this page.

Did You Know?

*The letters JPL stand for the **J**et **P**ropulsion **L**aboratories, located in Pasadena, CA. JPL is an operating division of Caltech (California Institute of Technology), and a major contractor with NASA since the early days of space science.*

FAQ:

***Q:** Why are some of the words I'm reading printed in italics?*

***A:** Words that might be unfamiliar have been italicized, and are defined in the Glossary (p. 48). In the online version of your Student Manual the first occurrence of each such word is a Glossary hyperlink. Margin notes like this are always in italics.*

Did You Know?

The calculations for interplanetary space travel were first done by Walter Hohmann, in his book titled: The Possibility of Reaching Celestial Bodies (1925).

Did You Know?

To reach Mars from Earth a rocket must achieve a speed of 11,200 m/s. That's about 25,000 mph!

FAQ:

Q: *Ok, I'm getting used to the margin notes, but it seems as if there's still a lot of unused space on some pages. Isn't this a real waste of paper?*

A: *Marginal notes are typically added to subsequent editions in response to feedback and questions. Building this space into the document now makes later revisions easy.*

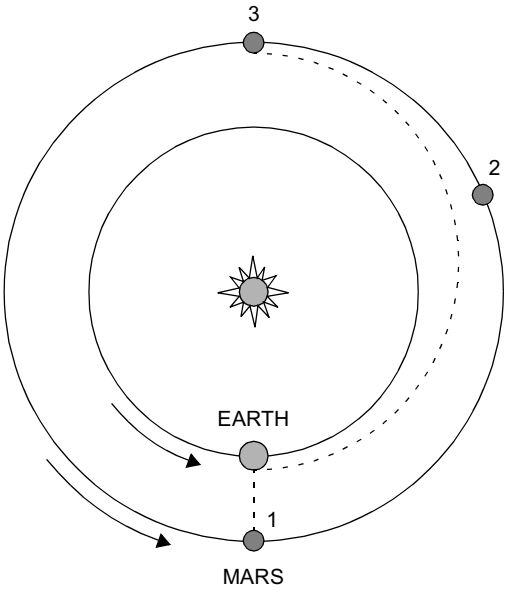
And until the margin is completely full, feel free to use this space for your own notes.

The Flight Plan:

Space travel takes a lot of fuel.

Earth is 150 million kilometers from the Sun; Mars 228 million kilometers. The 78 "Mkm" gap cannot be navigated easily, as the planets move at different speeds. The fastest possible path (to 1) would require more fuel than can be carried by current spacecraft. The long path (to 3) requires the least fuel, but would take about 8½ months one-way and would launch when Mars is at (2).

This interplanetary maneuver is known as the Hohmann Transfer.



The Delta II:

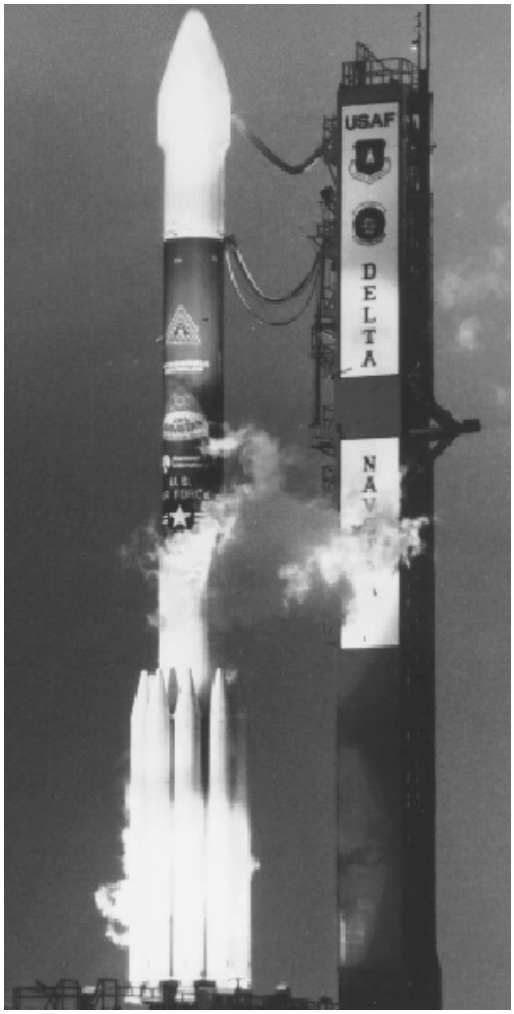
This photo shows the launch of a Delta II rocket, currently our only vehicle capable of the 78 million kilometer journey to Mars.

The rocket is 38 meters tall and, at its waist, 2 meters in diameter. Its engines produce a total thrust of 2.63 MN (mega-*newtons*) or approximately 600,000 pounds.

The rocket reaches a top speed of about 11.2 km/s; still, the journey to Mars lasts 8-9 months.

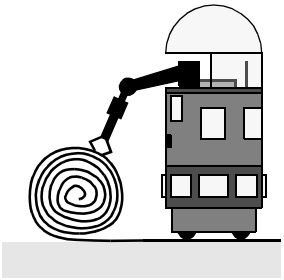
The payload (satellite or robotic explorer) is housed in the white conical shell at the top. The eight devices arranged around the base are solid-fueled rocket *boosters*.

The Delta II is made by McDonnell Douglas Space Systems.



DELTA II IMAGE COURTESY OF JPL/CALTECH/NASA

Mission 1: Creating the Mars Diorama



Introduction

All Missions will take place on a simulated landscape (diorama) that you plan and construct. This diorama represents RB5X's landing site on Mars. Floor space in your classroom must be set aside for the diorama. An ideal area would measure about 5 m x 5 m, with a 10 x 10 grid of 50 cm blocks. Smaller, rectangular areas can also be used. Study the sample diorama in Appendix 1 before sketching your own; it contains ideas and restrictions applicable to all valid dioramas.

Measure the length and width of your proposed diorama location, then use graph paper to plan the site. It's easier to make changes on paper than to physically alter a project already underway.

Creating the Mars Diorama Surface

When the location and shape of the diorama is decided, its floor area must be covered with paper. For a more realistic Mars-scape this paper can be colored randomly with light shades of pink and tan. The black crevice and finder tape must contrast strongly with the paper for RB5X's IR (infrared) sensor to work properly. Rolls of shelving paper or pieces of paper grocery bags work fine, provided creases and bumps are flattened. The paper must be taped to the floor or otherwise secured from sliding.

Next, measure and lay out grid lines 50 cm apart. Draw them first with a pencil and ruler, then retrace them with a narrow black marker. Next, label the grid blocks along two edges to establish a *coordinate system* according to the following scheme:

1. locate the origin at the corner west of RB5X's landing site
2. number the bottom row as 0,1,2,3 ... (*longitude* east)
3. number the left column as 0,1,2,3 ... (*latitude* north)

When the grid lines are drawn you can locate and create the crevice. It can be painted onto the surface or formed with separate pieces of black paper or overlapping strips of black tape. Finally, place the black tape which will guide RB5X to its charger nest.

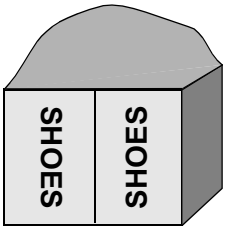
FAQ:

Q: *What's a good shape for the rocks in our diorama?*

A: *See below (paper maché cap optional)*

Creating the Rocks

You can make the rocks from paper maché, shoe boxes, or scrap cardboard. They should be flat on the bottom with a 20-25 cm squarish base and have flat, vertical sides 15-20 cm tall (for good contact with RB5X's bumpers). Two shoeboxes attached top-to-top and cut down to size work quite well.



FAQ:

Q: *How does one activate the self-diagnostic power-up?*

A: *With the Utility EPROM installed, simply turn on the robot. The self-diagnostic power-up will run automatically. It tests a variety of robot functions, including RB5X's drive motors.*

They can then be painted to resemble the three classes of rocks found on *terrestrial* planets: *Igneous*, *metamorphic* and *sedimentary.* Leave some means to add weight to the rock, as the next step requires experimentation.

RB5X must be able to push these rocks, but they need to be heavy enough to trigger its bumpers when contacted head-on. The required weight must be found experimentally, since it will depend on the rock-to-floor *friction*. The rocks must also be stable enough to slide without tipping over. To test a rock for "pushability" and stability, place it directly in front of the robot (≈ 5 cm from bumper 1) and activate the self-diagnostic power-up. RB5X will move forward about 10 cm, backward by the same amount, and rotate 90° twice. The rock should slide but not tip over. To test a rock for bumper triggering, place it directly in front of the robot (≈ 1 m from bumper 1) and activate RB5X's "wander mode." The rock should be ignored by the sonar but cause the robot to stop and turn when contacted by the bumper.

FAQ:

Q: *How does one activate the wander mode?*

A: *With the Utility EPROM installed, the robot on, and the self-diagnostic power-up completed, simply press bumpers 2 and 5 simultaneously.*

Creating the Mountain

The mountain can be constructed from any large cardboard box, 1x2 grid blocks to 1x4 grid blocks in horizontal size. It must fit closely inside the grid lines that bound it. Box-to-grid line gaps should be < 5 cm, so the box will probably need to be cut down to size. It must also be heavy enough to trigger the robot's bumpers and tall enough for *sonar* detection, with flat, vertical sides at least 40 cm in height. Beyond these specifications, you can decorate it to look like anything from Denali to Devil's Tower.

Label the Landing Site

Did You Know?

The mountain must fit closely inside the grid blocks that bound it. This ensures a good sonar profile at the scale of your diorama.

[see below]

Construct a sign showing the chosen name for RB5X's landing site. Otherwise, simply label it "RB5X: Mission to Mars" (with a brief explanation) so that all who see your diorama will know what's going on. Expect to get a lot of questions from curious students and teachers from other classes.

Optional Activities for all Grade Levels

These activities can be used as assignments or for extra credit. They are not required for completion of Mission 1, but will increase your knowledge of the Red Planet and provide useful insights for other Missions.

A. Find some pictures of Mars in the library or online. Paint or draw your own realistic "mars-scape".
B. Refer to the data charts in Appendix 4, or other resources, and write an essay contrasting Mars and Earth.
C. Complete the crossword puzzle(s) in Appendix 3.
D. Research a scientist, writer or artist connected with Mars, and write a biographical essay describing their Mars-related activities.

E. Read Appendix 4, 5, and 6, and write a technical report discussing the design requirements for a machine that must operate in the harsh surface environment on Mars.
F. Appendices 8-14 include images of other robotic explorers. Construct poster-sized drawings of these magnificent machines and create a "Wall of Fame" to decorate the classroom and set the mood for your adventure.
G. Appendix 6 (The Surface of Mars at Ares Vallis) shows a wide-angle view from the Mars Pathfinder Lander. A larger, color version of this view can be found in the IMAGES supplement. This was the terrain explored by Sojourner Rover. NASA investigators gave names to many of the prominent rocks, some descriptive, some puzzling. Giving the rocks names was no frivolous exercise; when planning Sojourner's route on the surface it was easier to call the rocks by name than to say something like "the rock at relative coordinates 1 meter north and 3.5 meters east."

Did You Know?

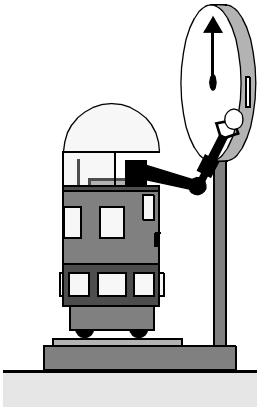
Sojourner was named in honor of the 19th century American civil rights activist Sojourner Truth.

There are coordinates around the image for specifying the positions of individual rocks. See if you can match the rocks with their names. Some will be obvious. Others, as stated earlier, are puzzling. If you can get 5 right out of 10 you are way above average (or very lucky).

Here's a hint to get you started: the image below is a close-up of the rock named "Stimpy." As you try to match this image with one of the rocks in the full picture you will be engaged in *pattern recognition*. This is very easy for humans to do, but very difficult for robots (or computers). As you scan the picture for a match, your brain automatically compensates for size, orientation, lighting and other expected differences. Getting a computer connected to a camera to do the same thing is not easy. Some of the best results obtained to date have involved only simple objects like spheres and cubes — the robots have a long way to go, but they're still learning.



IMAGE COURTESY OF NASA/JPL/CALTECH



Mission 2: Robot Calibration

Introduction

There are two modes of operation for RB5X (or for any robotic explorer):

autonomous: exploring on its own according to stored instructions
teleoperation: being controlled by a human from a remote location

The latter will be examined in Mission 5 (Navigation and Piloting). In this Mission you will evaluate RB5X's performance as an autonomous mobile vehicle, measuring several aspects of its drive system *kinematics*.

In autonomous operation the robot must carry out instructions and make decisions on its own without any human intervention. In reality, no robotic explorer to date has been totally autonomous — all had the ability to contact Mission Control for instructions if stumped by a problem they couldn't solve on their own. This is good.

The more distant the exploration, the greater the need for autonomy, as the time for radio communications to travel between robot and Earth become significant in the vastness of space. A radio signal traveling at the speed of light (300,000 km/s) will take 4.3 minutes to go from Earth to Mars when the two planets are closest! Imagine RB5X in a critical situation (boulder rolling towards it or drive motor overheating) having to wait 8.6 minutes for instructions about what to do.

Autonomous operation requires that the robot carry with it all instructions necessary to handle all possible critical situations, the topic of Mission 6 (Hazard Avoidance). Also required is a knowledge of its own capabilities and limitations, the measurement of which is the topic of this Mission.

Scenario

RB5X is still on Earth, in the lab, ready to be *calibrated*. The process of calibration involves measuring certain aspects of its performance that are needed for autonomous operation. Specifically, RB5X needs to know its:

1. speed on level ground
2. distance accuracy
3. directional accuracy
4. rotational accuracy

In this Mission you will measure RB5X's performance in the above areas. These measurements are useful data in any RB5X project, and are required data for programming solutions in Missions 3 through 7.

Mission Questions

A. Name some familiar machines on Earth that, if not actually robots, are at least partially autonomous. Hint: they must react to things — not just run without human care or intervention. A clock is not a robot.
B. In paragraph three of the Introduction, why is the crisis response time given as 8.6 minutes when it only takes 4.3 minutes for the radio signal to travel from Mars to Earth?
C. How long would it take for a radio signal to travel between Earth and Mars when they are farthest apart? Hint: study the orbital diagram at the top of page 8 and then do some arithmetic.
D. The Scenario lists four performance categories for RB5X that are very important to know. Why is it important to know the speed of the robot? Hint: remember that the robot will be exploring autonomously.

FAQ:

Q: What is the speed of a radio wave when measured in mph?
A: 186,000 miles/sec = 669,600,000 mph!

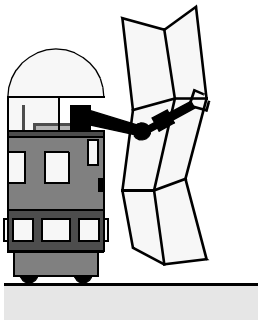
Mission Problem

Devise a method for measuring the following performance characteristics, perform the measurements, and express your results to the correct number of significant figures.

A. speed on level ground: How fast does RB5X move? Express your answer in units of "meters per second" (m/s).
B. directional accuracy: When RB5X is commanded to move forward in a straight line, how straight is that line? Hint: this parameter is most easily expressed as "X meters of veer in Y meters of distance."
C. distance accuracy: When RB5X is commanded to move X meters, how far does it actually move? Express your answer as a percentage error.
D. rotational accuracy: When you tell RB5X to turn 90° or 180°, how far does it actually rotate? Express your answer as a percentage error.

Exploring Further

A. When RB5X uses its sonar to measure a distance, how much error is involved? The absolute error (in "cm") will increase with distance, so the relative error (in "%") is a better measure. Graph your results, with relative error plotted as a function of distance. Obtain data from 1-5 m distance in increments of 0.10 m.
B. One advantage of sonar over *lasers* used in *ranging* measurements is that sonar can "see" through the same fog and dust that scatters laser light. Are there any other advantages? Are there any disadvantages? Write a two page essay discussing the topic of "sonar vs. laser" ranging.
C. If you have the RB5X robotic arm accessory, measure its maximum lifting force in units of "newtons" (N). Define "maximum" as the amount of weight that blows the 6 amp fuse in the power line to the arm.



Mission 3: Mapping

Introduction

An essential phase in the exploration of an unknown territory is mapping. You need to know where you, or your robotic emissary, are located relative to known territory. Most of the mapping done by NASA is accomplished from high orbit, where the cameras can image up to half a planet's surface. Smaller details like craters or other landmarks are added later by human or robotic explorers on the ground.

But a map is far more than just a copy of an image. A coordinate system of latitude and longitude lines must be superimposed over the image to allow specific features to be referred to unambiguously. So rather than saying "that small mountain next to the big crater just to the left of the three small valleys" you can say "the mountain at coordinates 20°N, 30°E."

Did You Know?

The prime meridian (0° longitude) on Earth was chosen to run through the Royal Observatory in Greenwich, England, one of the first centers of global mapping.

NASA uses the same cartographic coordinate system for all planets, Earth included. Here are the defining concepts and *conventions*:

1. north is the direction toward the pole that rotates counterclockwise when viewed from above
2. the equator is the great circle equidistant from both poles and dividing a planet in half
3. east is to the right when you are facing north
4. latitude numbers run from 90°S at the south pole (sometimes -90°) to 0° at the equator to 90°N at the north pole (sometimes +90°)
5. longitude numbers run from 0° at the prime meridian (an arbitrary point usually corresponding to a notable geographic feature) to 180°E and 180°W at the point halfway around the planet.

Did You Know?

The prime meridian (0° longitude) on Mars was chosen to run through the center of Terra Meridiani, a large dark lava plain that straddles the martian equator.

We will use a more basic coordinate system. Any true *cartographer* would appreciate its simplicity. The diorama grid is numbered much like a simple graph, with each block of the grid assigned an *ordered-pair* of numbers that describes its position relative to the origin (0,0). For example, the Sample Diorama (Appendix 1) assigns coordinates (6,0) to RB5X's landing site; the crevice is located at coordinates (0,8) through (3,8). The first number of each pair gives longitude east; the second number gives latitude north.

Scenario

RB5X has landed on Mars! Its first task is to map the local area, recording the positions of any mountains, crevices or large rocks. It must then report those positions back to Mission Control (i.e., you). RB5X already knows its starting coordinates (6,0) and must explore the entire surrounding area from coordinates (0,0) through (9,9).

Mission Questions

A. How can RB5X sense and tell the difference between a crevice, a rock and a mountain?
B. How can RB5X know how far it has moved, in total distance, since it left its landing site (6,0)? Note: RB5X does not have an *odometer*.
C. How can RB5X calculate the straight-line distance from its current position to its landing site (6,0)? Assume the site is out of sight.
D. How could RB5X use its video camera to find the direction of north, at night, on the surface of Mars?
E. RB5X knows it started at the landing site facing north. The scheduled exploration will consist of a series of linear motions connected by turns at specified rotation angles. How can RB5X tell which direction it is facing at the end of an exploration?

Did You Know?

Those increasingly popular GPS (Global Positioning Satellite) receivers are useless on other planets. They work on Earth because of the system of GPS transmitter orbiting our equator. After colonization, Mars is sure to get its own GPS system.

Mission Problem

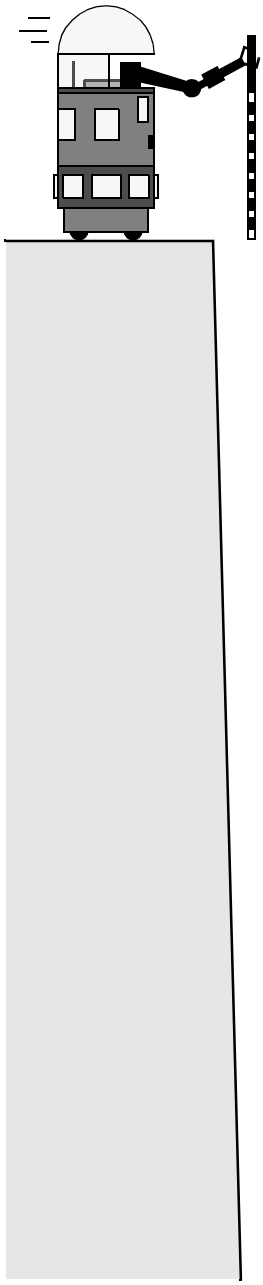
Write a program that enables RB5X to find the mountain without leaving the grid, record the mountain's coordinates for later output, return to the landing site, and signal its results via LED blinker code. You may assume the mountain is rectangular in shape, with grid dimensions 1 by X blocks, and has its long axis oriented east-west. You may also assume RB5X starts at coordinates (6,0) facing north.

Exploring Further

A. Write a program that will allow RB5X's sonar to discriminate between objects that are within the diorama area and those that are outside. RB5X should ignore sonar echoes from objects that are not a part of the simulated Mars-scape, and signal a detection when they are. Your first task in solving this problem is to decide what information RB5X needs to know, and what assumptions must be made to clearly define the geometry.
B. Mars has a weak magnetic field compared to Earth; some planets have none at all! Fortunately, compasses are not the only means for determining direction. Devise a method by which RB5X could use its video camera, during the daytime, to determine the direction of north.
C. Design and/or construct a device that would allow RB5X to use sonar to detect the position of a crevice. Note: this is as essential to mapping as it is to Hazard Avoidance (Mission 4).

Caution!

Never point a video camera directly at the Sun without using a properly rated filter over the lens.



Caution!

RB5X has a tough aluminum shell for skin, but the robotic arm, when extended, is vulnerable — even at normal operating speeds. Always stow the arm in its most compact position when engaged in exploratory roving.

Mission 4: Hazard Avoidance

Introduction

In our previous discussion of teleoperation we noted an inherent problem with interplanetary communications. Due to the vast distances separating planets there will be a significant delay between transmission of a message and reception of a reply. This delay is comprised of:

1. time for radio message from RB5X to travel from Mars to Earth
2. time for human teleoperator to decide what to do
3. time for radio commands from teleoperator to travel from Earth to Mars

The great distance between Mars and Earth, and the finite speed of radio waves, cause a problem with teleoperation. Radio waves are fast but they are not instantaneous. They move through space at 300,000 km/s, the same speed as light. Unfortunately, some of the hazards RB5X might encounter could require immediate decision and action.

This is why artificial intelligence is so important. AI can compensate for some of the limitations that signal delays place on teleoperation. In fact, the two are complimentary — good design strategy incorporates both AI and teleoperation to whatever extent is possible. Sojourner used AI for hazard avoidance and power management, but was teleoperated during sample acquisition.

Scenario

RB5X is on Mars, 21 *light-minutes* from tech support, and must now fend for itself. Using a "move and evaluate" exploration strategy it advances in short, straight-line segments only as far as it knows is safe. Even with this conservative approach, RB5X must continually monitor for the presence of hazards. In situations like this, AI is an indispensable survival tool.

The hazards presented by Nature on a foreign planet are too numerous to list, and include conditions not easily simulated in the average classroom (e.g., vacuum, radiation, temperature extremes). For our Mission we are concerned with only three: mountains, steep slopes and crevices.

Mountains are a hazard, though not in the sense of collisions. The robot moves too slowly to damage itself in an impact, but its drive motors can be burned out, or blow a fuse, if they push for too long against an immovable object. Steep slopes can cause RB5X to lose balance and tip over. If that happens, the Mission is effectively terminated — RB5X may never be able to right itself. Crevices present the greatest hazard, since a fall from even a few meters height would pretty much total the robot.

Mission Questions

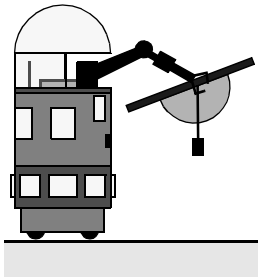
A. The distance from Earth to Mars changes continuously as they move around their orbits. Calculate the time it would take an RF command to go from Earth to Mars at closest approach. Calculate again for when they are farthest apart. Hint: you can find astronomical distances in Appendix 4.
B. As explained earlier, if RB5X ran into an immovable object and didn't stop or turn, its drive motor fuse might blow. Why should a stalled motor cause a fuse to blow? A stalled motor isn't moving. One expects a motor running too fast might blow a fuse, but it doesn't seem sensible for a stalled motor. Can you resolve this apparent paradox? Hint: do some research on the physics of electric motors.
C. Nine robotic explorers have landed on other worlds, as of this writing (ten including RB5X — who's still waiting for a ride). All were fortunate to land on reasonably level ground. Potential landing sites are surveyed in advance, when possible, using orbital reconnaissance. Unfortunately, level ground is usually scientifically uninteresting, so risk to the robot has to be weighed against the opportunity for better data. Consequently, balance is still an important design feature in landers and rovers. Specifications and photos of these ten explorers can be found in Appendix 7. Examine their mechanical structure and rank them in order from "most stable" to "least stable." Hint: do some research on the physics of *center-of-gravity.*
D. The Mission Scenario exaggerated. There may be a way for a toppled RB5X to right itself using its robotic arm. The arm is rated with a lifting force of 5.0 N. The robot weighs 120 N (with arm). Seems impossible, but there could be a way to do it using only RB5X's onboard resources. How? Hint: do some research on the physics of *simple machines*.

Mission Problem

Write a program that allows RB5X to avoid the crevice using its IR sensor. Ignore, for now, the fact that the sensor is underneath the robot and could not "see" the crevice until RB5X is directly over it and already falling.

Exploring Further

A. RB5X is moving forward. At t = 0, onboard sensors detect the need to stop as quickly as possible. What is the robot's minimum "response time" and "stopping distance" from sensor activation to cessation of motion?
B. Design and/or construct a device by which RB5X could use its IR to sense and recognize a crevice before falling into it.
C. Design and/or construct a device that would allow RB5X to sense the slope of the ground. Integrate this device with a hazard avoidance program that stops the robot and reverses motion before it tips over. Of course, you will first need to measure the maximum safe tilt angle for your RB5X.



Mission 5: Navigation and Piloting

Introduction

In Mission 3 we investigated mainly autonomous exploration and learned that RB5X can do a lot of science on its own. But there are many scenarios where human intervention and decision-making are essential and *artificial intelligence* (AI) is simply not up to the task. Consider a situation where RB5X needed to select a single rock sample from a rubble field. Textures, colors and shapes that are significant to a human geologist would not be meaningful to the robot. Could RB5X be programmed to recognize a fossil if he saw one? Not with today's computer technology. The experience and intuition that frequently guides human investigators is just not available to robots — AI has yet to evolve to that level.

Teleoperation is the solution to limited robot AI. A human teleoperator, or "remote pilot," can control RB5X's actions in response to what is relayed by its onboard sensors. What the robot's camera sees is displayed on the teleoperator's computer monitor, providing visual input for decisions that could change the course of the mission. In an emergency, the teleoperator can assume total command of the robot's functions.

Teleoperation, in practice, rarely involves continuous, protracted motion. Because of the signal delay times imposed by interplanetary distances, the teleoperator chooses each new location, commands the robot to go there, waits for visual confirmation of the move, surveys the area from the new vantage point, and then decides the next step. It's slow — but it works.

Both Sojourner and Lunokhod, the only autonomous mobile vehicles used to date in planetary exploration, combined teleoperation with crude AI to ensure mission success. Yet in both cases there were unforeseen problems that neither teleoperation nor AI could solve. Sojourner had a transmitter problem that amazingly seemed to fix itself, and Lunokhod had problems regulating its temperature in the harsh extremes on the Moon. Even with these glitches, what they accomplished was nothing short of spectacular.

FAQ:

Q: *Would a compass be able to find north on other planets?*

A: *In most cases, no. On the Moon, which has no magnetic field, a compass would point towards Earth, or your spaceship, or any nearby object that contains a lot of iron. On planets that do have magnetic fields, the magnetic north pole does not, in general, coincide with the rotational pole. On Earth, the magnetic north pole is in the middle of Hudson Bay, over 2000 miles from the rotational pole. So even here on Earth, a compass only points approximately to the north.*

Scenario

RB5X has mapped the area around its landing site and is ready for some routine reconnaissance. Unlike the Moon, which has no weather, things change daily on Mars. Rocks that were hidden are uncovered by blowing winds. The polar caps grow and shrink with seasonal temperature changes. Volcanoes, though long dormant, still vent steam and other gases. And not all scientists are convinced Mars is lifeless — the Viking missions of 1976 left that question open. One never knows what might turn up during a walk on the Red Planet. Once around the mountain should do it.

Mission Questions

A. Many of the exploratory spacecraft sent to other planets qualify as true robots, but only two were mobile — Sojourner and Lunokhod. Read more about these explorers in Appendix 7, and cite similarities and differences between their designs and that of RB5X.
B. RB5X's video camera is mounted only 36 cm above ground level. This presents a whole new view of any world, including Earth. What problems might a teleoperator encounter with such a low eye level?
C. Luna 9 was not much taller than RB5X. Based on the information in Appendix 7, how far away was Luna 9's horizon? Hint. The *Pythagorean Theorem* is all the mathematics you need to solve this problem.
D. *Triangulation* is a surveying procedure whereby the position of the observer can be deduced from the compass directions to any two known landmarks (usually distant mountains). RB5X has obtained the following information by using its video camera to sight two mountains on Mars:

Mount Kirk:	(coordinates 20° W, 40° N)	compass heading 60°
Mount Picard:	(coordinates 10° E, 30° S)	compass heading 140°

What are the coordinates of RB5X's location? Hint: get some graph paper, plot the mountains, use a protractor, and find the robot.

FAQ:

Q: *If we can send a rocket to Mars, why haven't we sent any astronauts yet?*

A: *Satellites and robots don't need to return to Earth, but astronauts do. This would require almost twice as much fuel — more than the Delta II can carry. Human exploration of Mars awaits the development of a new rocket transportation system.*

Mission Problem

Mountains are, by their nature, sites of erosion. Around their perimeters are *talus* fields. Within the talus are countless rock samples, excavated by erosion and easy to collect. RB5X mapped the coordinates of the mountain back in Mission 3, but suppose you know only the mountain's location and not its extent. Write a program that causes the robot to:

1. accept, as input, the coordinates of an arbitrarily specified grid block adjacent to the long side of the mountain
2. move from its landing site to the specified grid block
3. circumnavigate the mountain's perimeter
4. return to the landing site
5. report the length of the mountain (using horn, LED or voice)

Note: *Effluvial* regions, like flood plains and glacial deltas, are also great sites for acquiring samples. Sojourner landed in an ancient flood plain of the Ares Vallis region on Mars. Many interesting geological specimens were imaged and tested. No evidence of life was found. Data did support the hypothesis that a torrent of water flowed through the region ages ago. The source of that water was not rain, but more likely an outflowing from melting *permafrost* during an unusually warm era in the planet's history.

Exploring Further

Depending on which hardware options your RB5X has installed, one or all of the following activities are possible.

A. Basic RB5X (AI navigation)

Position the robot on the grid block farthest from (6,0), facing to the north. Write a program that will allow RB5X to find its way back to (6,0) subject to the following conditions:

1. it knows its starting coordinates and direction (as data in the program)
2. it does not know the location of the crevice, mountain or rocks
3. it must avoid all obstacles enroute to its goal

B. RB5X + RF Com-Link (limited teleoperation)

Here's a real test of teleoperator skill. Remove all but one rock from the diorama. Next, move the mountain, crevice and rock to new positions unknown to the pilot. Finally, set up a screen or remote terminal such that the diorama is not visible to the pilot, but the robot's horn or voice signals are still audible. The robot will start from its landing site at (6,0).

RB5X will provide *feedback* to the pilot about what it senses, but no visual images. Based on this feedback the pilot must choose the next course of action — a move to one of the four adjacent grid blocks. The goal is to find the rock, the location of which is unknown.

Write a program that does the following:

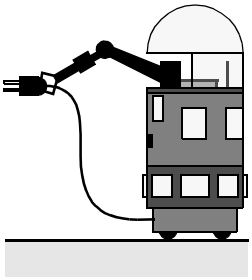
1. move the robot one grid block in a specified *cardinal direction*
2. probe the adjacent grid block sensing for crevice, mountain, or rock
3. report the results of the probe with horn or voice signals
4. stop to await instructions for the next move

The pilot can now upload a new program for the next move or, if a rock has been found, command RB5X to return to (6,0).

C. RB5X + RF Com-Link + RF Video Camera (full teleoperation)

Use the same setup as Exploring Further B. The video camera eliminates the need for probing and reporting. Thus, this exercise involves visually avoiding hazards, estimating safe movement distances, and programming movement commands. The goal is to simply touch the rock with bumper 1. You could add more crevices and/or mountains to increase the challenge.

Mission 6: Energy Management



Introduction

One thing all robotic explorers have in common is a finite supply of energy with which to perform their missions. Power supplies used to date include chemical batteries, *photovoltaic* electric systems, and *RTG's* (radioisotope thermoelectric generator). All had a limited storage capacity and thus were doomed to eventual failure. There is no such thing as an "infinite battery" or "perpetual motion machine."

The First Law of Thermodynamics states that energy can never be created or destroyed. The Second Law states that energy can be converted between forms (e.g., electrical energy into kinetic energy), but with less than 100% efficiency — the waste being lost as heat.

The use of *fuel cells*, *regenerative braking*, wind power (where there is air) and *flywheel* energy systems can increase the lifetime of a power supply, but the need to conserve energy and use what is available efficiently still creates constraints. Energy management remains a high priority in system design and mission planning. Research into new power sources continues, but it is unlikely the Laws of Thermodynamics will ever be repealed.

Working within the constraints imposed by a finite power supply is often a real challenge. One can only do so much with X *joules* of energy, and the trick is to get the most science done within those constraints.

Scenario

RB5X arrives on Mars with fully charged batteries. Its motor/sonar/LED battery has a capacity of 10 *amp-hours*, its electronics battery 5 amp-hours. If RB5X stayed in one place and simply acted as a remote sensor, then the electronics battery would be the first to expire. If RB5X were commanded to move forward indefinitely (assuming it could do so without running into an obstacle), then the motor/sonar/LED battery would expire first.

Typical missions require both electronic processing and motion, so either battery could be the limiting factor, depending on programmed activities. In most cases it will be the motor/sonar/LED battery that first expires, since motion takes more energy than thought (this is true for humans too).

The constraint a finite energy supply places on RB5X's operation needs to be translated into specific limitations on performance. What can be done within that constraint? What cannot be done? A solar-powered charger nest is, of course, available at the landing site. But for serious exploration, RB5X must venture far from the comfort and safety of its nest.

Did You Know?

The best chemical batteries available still have a limited number of recharge cycles before failure. Solar energy can be used to recharge them during the day, but the night will eventually claim them. Cold temperatures further accelerate the decline of a battery's charge.

Mission Questions

A. RB5X's motor/sonar/LED battery has a rated capacity of 10 amp-hours. What does that mean, and how is it related to the length of time RB5X can operate and/or the distance it can travel?
B. How does RB5X's total power output (measured in *watts*) compare to that of Sojourner?
C. RB5X can climb hills, but there is a maximum slope possible. On what external factors does this maximum slope depend?
D. RB5X weighs 24 pounds on Earth. What would RB5X weigh on Mars? What does this question have to do with energy management?

Mission Problem

Devise a means to predict RB5X's maximum range. You may assume it starts with a fully charged battery, moves straight forward with no turns, and travels over level ground. Of course, you could just find a large, flat, open area, turn on the robot, and follow it until the batteries shut down — but there's a much easier way to do this. Find that easier way.

Did You Know?

Scientists tend to be meticulous about the use of symbols, even to the distinction of capital and lowercase letters. In Exploring Further B there will be a need to use the symbols for speed (v) and voltage (V) in the same equation. Learn the language to avoid ambiguities.

Exploring Further

A. The internal factor determining how steep a slope RB5X can climb is the 6 A fuse protecting its drive motors. Use the equations for potential energy, work, power and speed to predict the maximum slope RB5X could climb (on Earth). The equations are presented here, but you may want to refer to a Physics text for help on how to use them.

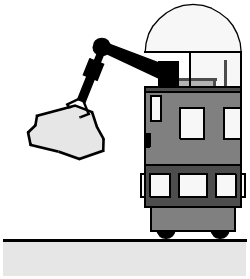
work: $W = F \cdot d$
potential energy: $E_p = mgh$
power: $P = \Delta E/\Delta t = F \cdot v = I \cdot \Delta V$
speed: $v = \Delta d/\Delta t$

Caution!

Whenever RB5X is made to move across an uneven or tilted surface, use a human spotter to provide for its safety. Don't ever let RB5X fall — its shell is pretty tough, but a sharp impact could easily damage internal components.

B. If you did Exploring Further A you discovered that, at least on Earth, and certainly on Mars, RB5X's drive motor is sufficiently powerful to climb any slope encountered. Friction (both *internal* and *rolling*) creates additional loads on the drive motor. Measure the friction experienced by RB5X on a smooth level surface at *1g* and 2g. Hint: three equations for power are given above — use one of them.
C. Which communications mode available to RB5X (LED, horn, voice, RF, or "arm waving") requires the least energy to transmit a message? You may assume, for comparison, the message to be sent is an "S O S".
D. Design and/or construct a solar charging system that RB5X could carry along on explorations, reducing or eliminating its dependency on the static charger nest. Hint: electrical specifications for the standard charger nest can be found in the Reference Manual.

Mission 7: Sample Acquisition



Introduction

Geology is the science of rocks, minerals, and topography, and the natural processes that shape them. Exogeology is the extension of geology beyond terrestrial limits. Other planets, moons, asteroids, meteors and comets are all possible subjects for exogeological study. Exogeologists learn a great deal by simply looking at other worlds through telescopes — they can learn even more if we, or our robot emissaries, go there.

The first object so studied was our own Moon. It was explored by several non-mobile robot landers that sent closeup pictures (Surveyor 1-7, 1966-8) and later by the Apollo 11-17 astronauts (1969-72). What we learned by seeing the Moon closeup, and by gathering and studying surface samples, changed our theory of the Moon's *primordial* origin.

Mars was the next target of robotic exploration. In 1976 the United States set down two Viking landers on the plains of Mars. These were also non-mobile robots, but they did have television cameras that could rotate and survey the surrounding landscape. One used a robotic arm to scoop up soil and deposit it into an onboard exobiology experiment. There the soil was watered, warmed and exposed to lights. No definite signs of life, plant or animal, were detected.

The Viking's cameras imaged the panorama around them. Many interesting features were visible. Black rocks that appeared volcanic in nature were strewn as far as the camera could see. The soil appeared distinctly reddish in color, suggesting the presence of *iron-oxide* (which colors the awesome "red rocks" throughout the American Southwest). Sand was seen blowing in the thin air, creating dunes that shifted from day to day. And in the early morning light, frost that had formed overnight on the ground was observed evaporating in the Sun's warmth.

Did You Know?

The oxygen in iron-oxide can be released by heating it. What you get is breathable O_2 plus a bonus of metallic iron (useful for construction). Most of what we'd need to colonize Mars is already there.

When the robotic explorer is mobile and can change its physical location, more exciting scientific possibilities arise. The explorer can be moved to visually interesting but remote objects and then examine or retrieve them. This is just what Sojourner did in July/August of 1997. When it rolled out of its Pathfinder Lander on the surface of Mars it became the first robotic explorer to be teleoperated on another planet. What Sojourner discovered profoundly changed our view of Mars, and added support to the hypothesis that large amounts of water once flowed on its surface. Most of that water is still there, frozen under the surface in a layer of permafrost. Future missions will include sample acquisition and sample return, allowing scientists to study the martian soil in a fully-equipped, Earth-based laboratory.

Scenario

The International Space Station (ISP) will be serviced by beachball-sized Aercam (Autonomous EVA Robotic Camera). These robots will use compressed N_2 *propulsion*, hazard avoidance, and self-recharging systems for continuous duty. Aercams will use AI to scout for changes to the outside structure of the ISP, reporting or taking action as needed. They were first tested successfully on *STS-87* in December of 1997.

Did You Know?

Your brain is wired to automatically detect changes in your field of view — even if you're not looking directly at whatever is changing. How often has a small crawling insect, seen only in your peripheral vision, caught your attention?

One of the few successes in visual AI, or pattern recognition, is the ability to recognize when something has changed in the view. Automated manufacturing quality control systems use this type of AI to spot defects; each inspected unit is compared to a virtual ideal model — visible differences are interpreted as flaws. A similar form of AI activates security cameras to record only when something in the field of view is moving, thereby making greatest use of whatever length tape is available.

Robotic explorers on the surface of another world could use the same AI strategy to look for signs of geological or biological activity. Any process that changes the environment is, by definition, interesting to scientists. When a robot is teleoperated, the teleoperator looks for such changes, often with the assistance of geologists and biologists. But when a robot is on its own, using only AI, that same strategy can still be applied.

Sample acquisition, by whatever means, adds significantly to the potential for scientific discovery. Once acquired, a sample can be subjected to testing otherwise nearly impossible. Onboard instruments can determine age (by carbon dating), soil density and porosity, chemical composition, and even transmit detailed images taken through powerful microscopes.

RB5X can't examine the samples it acquires, but it can return them to the landing site for safekeeping. Rocks left in the field, unless marked with stakes, can be covered by blowing sand, landslides or subsidence.

Mission Questions

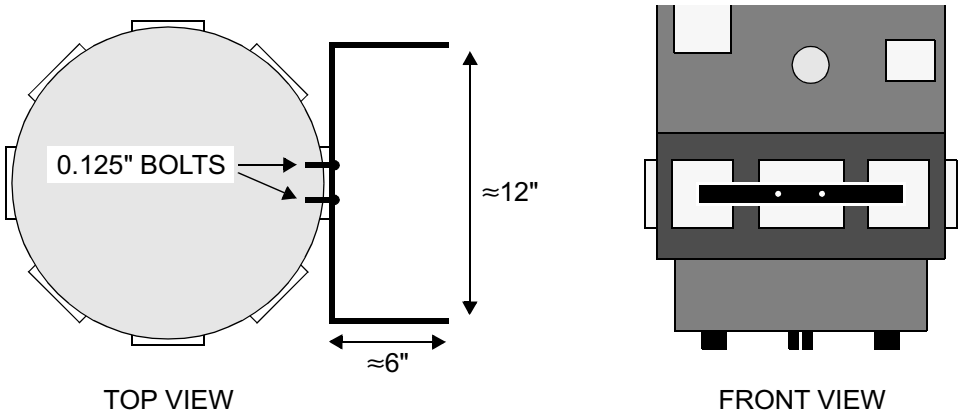
A. If you have access to a camera and tripod, position the camera at the back of your classroom and take a picture at the start of the day. Without moving the camera, take another picture at the end of the day. Examine the two photos closely and list all of the changes visible.
B. RB5X's robotic arm is rated at 5.0 N of lifting force. Trying to lift more will blow a fuse, rendering the arm inoperative. How could the weight of a rock seen with its camera be estimated before attempting to lift it?
C. If a sample of soil were returned from Mars, and if bacterial life forms were discovered in it, how could we be sure the bacteria weren't actually from Earth and just somehow got into the sample?

Mission Problem

In preparation for this exercise you need to remove all but one of the rocks from the diorama. You also need to construct a "rock acquisition device" (RAD) to attach to bumper #1. Get a piece of 1" x 24" x 0.125" aluminum flat and bend it at 90° as shown. Drill the flat and bumper to accept a pair of 0.125" x 1" bolts for attaching the RAD. There's ≈ 4-5 cm of hollow space behind that bumper to drill into — but be careful of the ribbon cable on the inside (it likes to hang in the way). Attach the RAD using bolts, nuts and lock-washers. If you're squeamish about operating on RB5X, you can use duct tape to fasten it in place. Whatever method you use, the bumper switch must remain functional as a sensor.

Caution!

Any modifications to the RB5X that involve drilling through the outer shell must be done with care. Remove the dome to get a clear view of the interior before you operate on your robot. See the Reference Manual for dome removal instructions.



Write a program that will cause RB5X to find the rock and return it to the landing site. Hazard avoidance is required.

Exploring Further

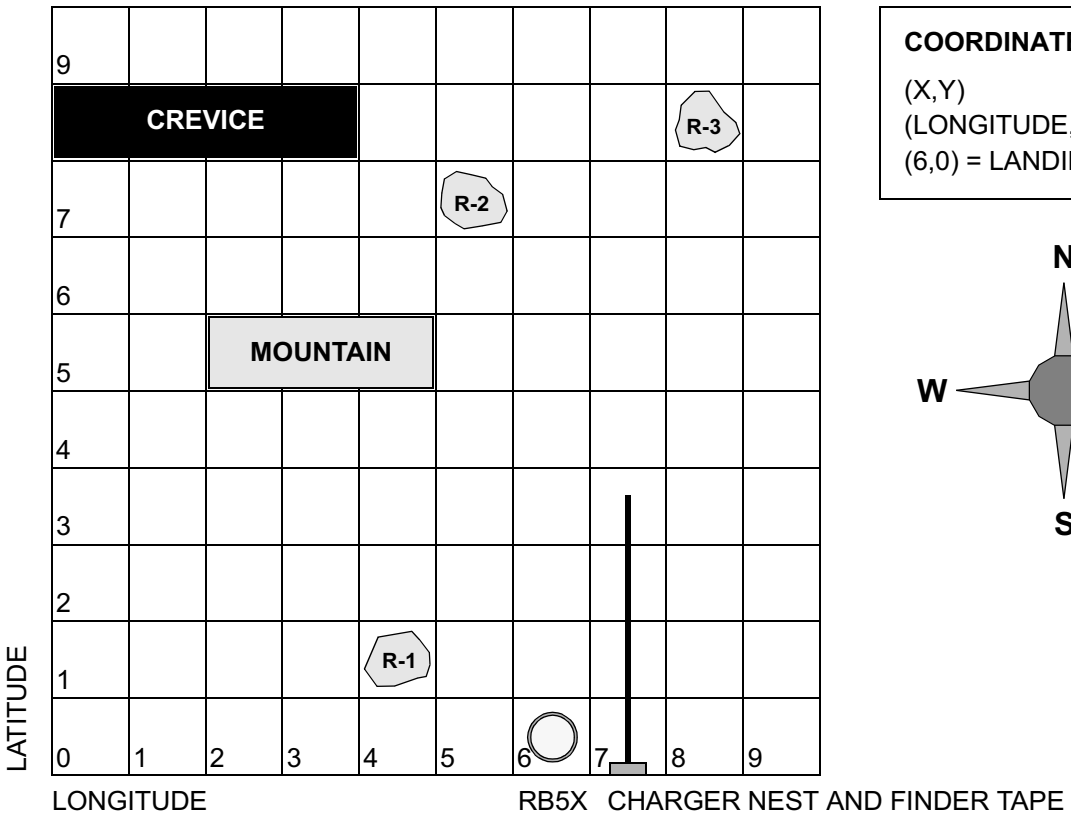
A. How can RB5X's teleoperator determine the size of a distant rock seen through the camera? The method of Mission Question B is not an option in this case. Hint: How do you suppose the image of RB5X on the cover was adjusted to the correct size, given the information in the photo caption?
B. Design and/or build a device that allows RB5X to acquire a sample too large to fit within the grip of its robotic arm (≈ 7 cm).
C. For RB5X with robotic arm option: cut a 10 cm length of wooden 2x4. Set the block on the ground 15-20 cm directly in front of the robot. Attach a 2-lb size coffee can as a sample collection "backpack" to the rear of the robot and in the least obtrusive location. Write a program that instructs RB5X's robotic arm to pick up the block and deposit it in the "backpack."
D. For RB5X with teleoperation options (RF com-link and video camera): guide the robot to a rock, engage it with the RAD (or the arm and "backpack" described in Exploring Further C), and return it to the landing site. This is a great exercise for team competition based on time.

BLANK PAGE

Appendix 1: Sample Diorama

Your diorama must meet the following criteria. Shown below is one example of a configuration that would work well in all Missions. If available floor space does not allow a 10 x10 square grid, other configurations that meet the criteria are acceptable. If you elect to complete only some of the Missions the diorama can be considerably smaller and simpler.

1. Grid blocks must be 50 cm square to fully contain RB5X's diameter.
2. The mountain may be from 1x2 to 1x4 grid blocks in size.
3. The mountain must near the diorama's center, oriented with its length in the E-W direction.
4. The rocks cannot be adjacent to the mountain, the crevice, or each other.
5. The crevice may be any size from 1x1 to 1x4, but may not be adjacent to a rock or mountain.
6. The crevice may be oriented in either the E-W or N-S direction.
7. Distance to crevice > distance to mountain (as measured from the landing site).
8. The row and column that RB5X starts on must be empty (no rocks, crevice or mountain).
9. Origin (0,0) must also be left empty.
10. RB5X starts on (6,0) facing north adjacent to the charger nest at (7,0).
11. The charger nest must be backed up by a wall (or very heavy object) so that it cannot move.
12. Leave at least 2 grid blocks open to either side of the charger nest.
13. Do not place objects such that any part of the grid is totally cut off from access.
14. The grid need not be square; a rectangle of minimum width 5 blocks wide may also be used.
15. Number the grid along two edges using a 2-axis coordinate system with origin as shown.



COORDINATES:

(X,Y)
(LONGITUDE, LATITUDE)
(6,0) = LANDING SITE

Appendix 2: The Heuristic Method

The field of knowledge concerned with general problem-solving methods is know as *heuristics*. It emerged as a subfield of mathematics early in this century and has evolved into the modern fields of *cybernetics* and artificial intelligence (AI).

It is perhaps amazing that such general problem-solving methods even exist, but when you really stop and think about it, you've probably already discovered some on your own. For example, you may recall noticing that, in the process of explaining a difficult idea to somebody else, your own understanding of it also improves. This is known as "restating the problem" in heuristics.

If you want to learn more about heuristics, the books Lateral Thinking, by Edward de Bono, and How To Solve It, by G. Polya, are highly recommended. Both include many great examples of how specific heuristic methods can be applied.

Presented below is a partial list of heuristic methods. They won't all work for a specific problem, but most of them will help. One of them will surely be the key that unlocks the door.

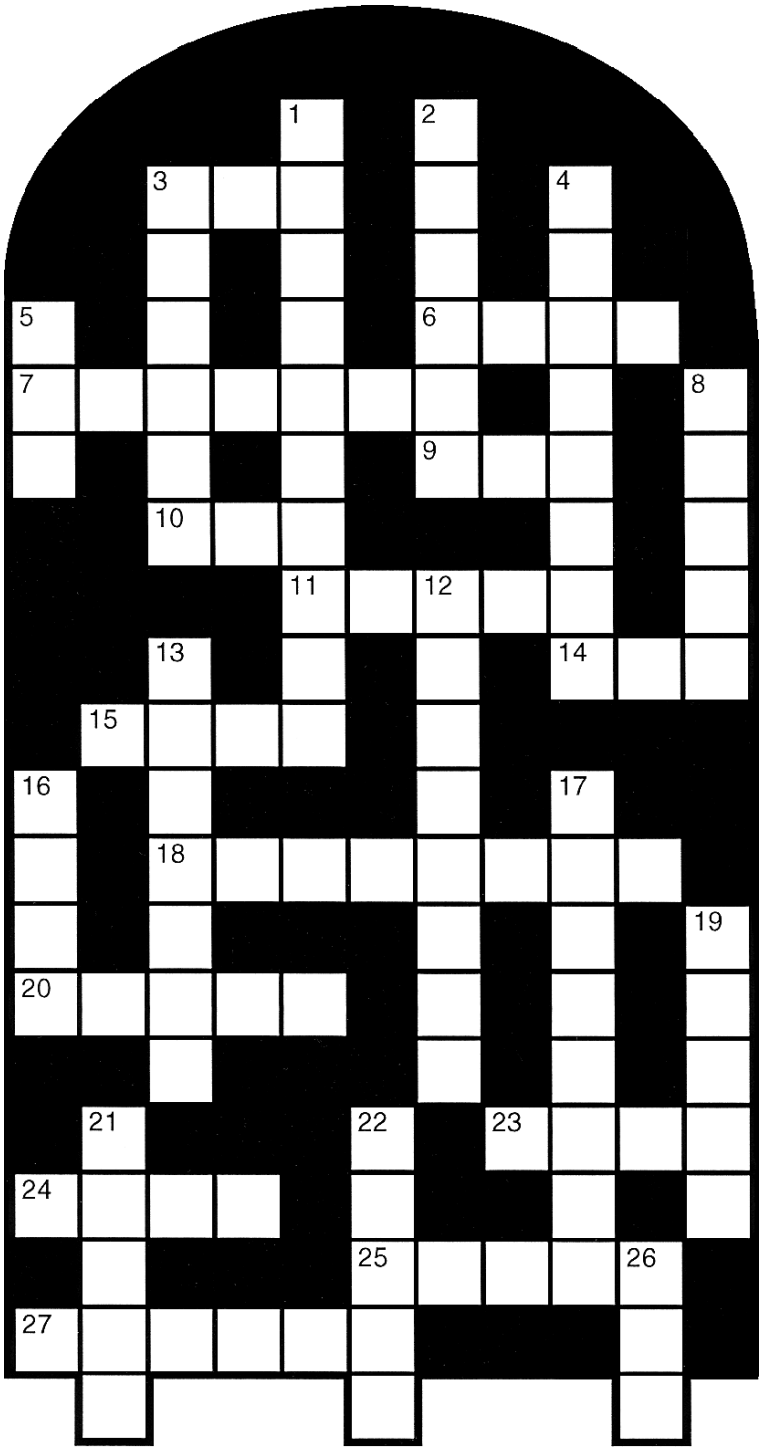
1. **State the problem clearly in a way that can be answered.** Write it down or state it verbally.
2. **Restate the problem in an equivalent way.** Ask the same question from a new point of view.
3. **Make an analogy.** Is there a similar problem whose answer you already know?
4. **Simplify.** You may already know the answer to a simpler problem and have a partial solution.
5. **Generalize.** Do you know of a larger problem, already solved, that contains this one?
6. **Brainstorm.** Discuss the problem with others, exchanging ideas and insights.
7. **Define your terms.** Be sure you know, or decide on, the meanings of critical terms.
8. **Break it down.** Divide the problem into parts and solve them separately.
9. **Draw a picture.** Visualizing often helps. Even abstract block diagrams may provide insights.
10. **Make a guess.** Try your first impression and see where your intuition leads you.
11. **Focus on the unknown.** Is the problem's answer a number? Word? Code? Device? Shape?
12. **Use notation.** Represent the solution, and ideas that may lead to it, with symbols.
13. **Work backwards.** Imagine you have solved the problem; then deduce the consequences.
14. **Look for symmetry.** Look at the problem upside down, inside out, or mirror-imaged.
15. **Give up. Temporarily.** Allow your subconscious to process the problem. Insights happen.

Add your own methods to the list. You must have discovered a few by now.

16. **?**
17. **?**
18. **?**
19. **?**
20. **?**

Appendix 3: Mission to Mars Crossword Puzzle

All the words needed to complete these puzzles are somewhere inside this manual. Hyphenated and multiple-word terms should be entered as a continuous string of letters, with no spaces or special symbols. This first puzzle is just a warm-up. Advanced crossworders may proceed to the second puzzle — but be ready for some challenging vocabulary exploration!



ACROSS

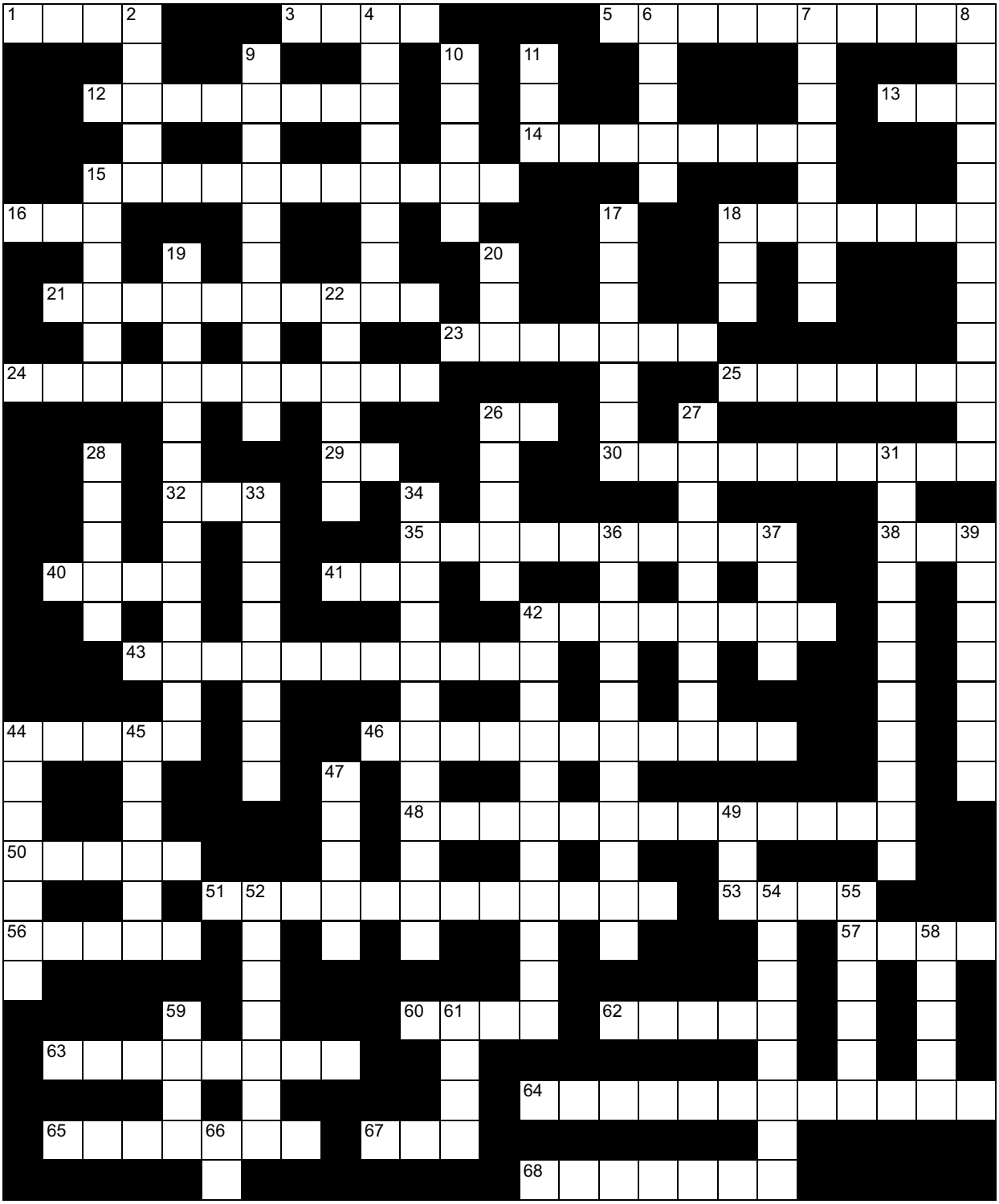
3. volts in RB5X's battery
6. its rust colors sand on Mars
7. can be recognized or moved in
9. this letter (spelled) = gravity
10. official name of the Sun
11. element 12 di-?
14. color of the fourth planet
15. radiation used by dentists
18. region where lava is found
20. ?-rockets act like brakes
23. molten rock
24. explored by Sojourner
25. rocket used for Mars travel
27. robotic explorer of Venus

DOWN

1. study of non-Earthly rocks
2. first robot on Mars
3. ?-Major is dark martian plain
4. keeps track of distance
5. built Sojourner with Caltech
8. Earth is this rock from the Sun
12. how RB5X senses light
13. holds things to the ground
16. 780 Earth days on Mars
17. 1000 watts (abbreviation)
19. cells that catch light energy
21. intense light source
22. used to measure range
26. can pick up a rock

Exolexicologist Certification Puzzle

Any resemblance between this puzzle and a diorama grid laced with multiple crevice hazards is purely coincidental. It's only hazardous to your vocabulary self-esteem. If you can survive this puzzle, completing it without error, you are a true *exolexicologist*.



Exolexicologist Certification Puzzle Clues

ACROSS

1. ten hundred instructions every second
3. basaltic liquid
5. never-melting ice in soil
12. friction not due to ground conditions
13. time from one sunrise to next
14. RB5X's moon-roving cousin
15. limb that is able to bend and rotate
16. a.k.a. the Space Shuttle
18. where RB5X goes when hungry
21. science of problem-solving
23. study of rocks, minerals, topography
24. right triangle theorem
25. what AI vision seeks to recognize
26. common communications band
29. smart, but not really
30. location on a numbered grid
32. video communications band
35. study of unearthly life
38. Mars is the _?_ planet
40. US government space organization
41. isotopic power source
42. referenced to the stars
43. Earth-to-Sun distance is 8.3 of these
44. flat area ringed by hills or mountains
46. related to measurement of weight
48. remote control
50. renewable energy from a star
51. process wherein light becomes electricity
53. fourth rock from the Sun
56. terra firma
57. add oxygen, get rust
60. harmful electromagnetic radiation
62. rockets used for deceleration
63. one thousand joules per second
64. senses force that affects compasses
65. first robot on Mars
67. voltage rating of RB5X's batteries
68. stores electrical energy

DOWN

2. RB5X's ears
4. non-defensive shield
6. chip with instructions for CPU
7. output that becomes input
8. being there without being there
9. wobble of a spinning mass
10. special light source
11. mission control in Pasadena
15. on the way up
17. referenced to the Earth
18. RB5X's brain
19. geometric alternative to GPS
20. one of these is 9.8 m/s^2
26. autonomous machine
27. RB5X's Mars-roving cousin
28. radio detection and ranging
31. using air for deceleration
33. stores energy in a spinning mass
34. type of braking that saves energy
36. two numbers in a set of coordinates
37. once around the Sun
39. molecule with two oxygen atoms
42. rocks formed by erosive layering
44. temporary extra thrust unit
45. can make a basin or a crater
47. rocket currently used for travel to Mars
49. unit of atmospheric pressure
52. his transfer gets us to other planets
54. able to function without assistance
55. the lever is one of these machines
58. molecule with one oxygen atom
59. target object for robotic acquisition
61. your favorite robot
66. channel linking CPU to peripheral

Appendix 4: Data for the Planets

Table 1: Physical Characteristics of the Planets

PLANET	SYMBOL	EQUATORIAL RADIUS		MASS		ROTATION PERIOD	AXIAL TILT (degrees)
		$\times 10^6$ m	$\oplus = 1$	$\times 10^{24}$ kg	$\oplus = 1$		
Mercury	☿	2.439	0.3824	3.303	0.05528	58.65 days	7.0
Venus	♀	6.051	0.9488	4.869	0.8150	242.9 days (R)	177.4
Earth	⊕	6.378	1.000...	5.9742	1.000...	$23^H\ 56^M\ 04^S$	23.45
Mars	♂	3.397	0.5326	0.6416	0.1074	1.026 days	25.2
Ceres	①	0.5	0.008	0.001	0.0002	**?**	**?**
Jupiter	♃	71.49	11.21	1899	317.8	0.4135 days	3.08
Saturn	♄	60.27	9.450	568.6	95.18	0.4440 days	26.73
Uranus	♅	25.56	4.008	86.86	14.54	0.7183 days (R)	97.92
Neptune	♆	24.76	3.882	102.5	17.15	0.6713 days	28.8
Pluto	♇	1.17	0.183	0.0132	0.0021	6.39 days	118

Table 1: Physical Characteristics of the Planets (continued)

PLANET	OBLATENESS	DENSITY (g/cm^3)	SURFACE GRAVITY (N/kg)	ESCAPE VELOCITY (km/s)	ALBEDO	MAX VISUAL MAGNITUDE	NUMBER OF SATELLITES
Mercury	≈ 0	5.43	3.62	4.2	0.6	-1.9	0
Venus	≈ 0	5.25	8.62	10.3	0.76	-4.4	0
Earth	0.00335	5.518	9.81	11.2	0.39	-------	1
Mars	0.00521	3.94	3.69	5.01	0.15	-2.8	2
Ceres	**?**	2.3	0.27	0.52	0.1	+4.9	0 ?
Jupiter	0.0649	1.33	24.9	59.7	0.51	-2.5	16+4 rings
Saturn	0.0980	0.688	10.5	35.6	0.50	-0.4	20+7 rings
Uranus	0.0229	1.27	8.62	21.0	0.66	+5.6	15+10 rings
Neptune	0.0171	1.64	11.2	23.6	0.62	+7.9	8+4 rings
Pluto	**?**	1.97	0.643	1.23	0.5	+14.9	1

Table 1 Notes:

Radius values cited are equatorial averages, useful for most *gravimetric* calculations. Values are measured at the average solid surface of terrestrial planets, and the 1 bar pressure level for gaseous planets.
Rotation Periods are *sidereal* values, measured with respect to the stars. *Synodic* periods are measured with respect to the Sun. Differences are due to the planets' revolution around their orbital paths as they rotate. R $\Rightarrow$ *retrograde*.
Oblateness is the fractional measure of a planet's equatorial bulge. Earth's polar radius is 6335 km and its equatorial radius is 6378 km, a difference of 43 km (about 28 miles); its average spherical radius = 6371 km.
Albedo is a measure of reflectivity. If a planet reflected 100% of the sun's illumination it would have an albedo of exactly 1.00. Oceans, clouds and ice all contribute to high albedos.
Visual Magnitude is the apparent brightness of a planet (or star). In the case of planets, it depends on both distance and albedo (stars shine by their own light). The dimmest objects visible to the unaided eye have a magnitude of about +6. A difference of 2.5 magnitudes equates to a factor of 10 in brightness (lower magnitudes are brighter).

Table 2: Orbital Characteristics of the Planets

PLANET	SIDEREAL PERIOD (years)	SYNODIC PERIOD (years)	SEMI-MAJOR AXIS (AU)	SEMI-MAJOR AXIS (m)	ORBITAL INCLINATION (degrees)	ORBITAL ECCENTRICITY (no units)	AVERAGE SPEED (km/s)
Mercury	0.24085	0.31727	0.3871	5.791×10^{10}	7.004	0.20563	47.8
Venus	0.61521	1.5987	0.7226	1.081×10^{11}	3.394	0.00679	35.0
Earth	1.000039	--------	1.000...	1.496×10^{11}	0.000...	0.01673	29.8
Mars	1.88089	2.1354	1.523	2.278×10^{11}	1.850	0.09337	24.2
Ceres	4.604	1.277	2.767	4.140×10^{11}	10.62	0.0765	17.9
Jupiter	11.86223	1.0921	5.203	7.783×10^{11}	1.305	0.04844	13.1
Saturn	29.4577	1.0352	9.552	1.429×10^{12}	2.490	0.05568	9.64
Uranus	84.013	1.0121	19.22	2.875×10^{12}	0.773	0.04721	6.81
Neptune	164.793	1.0061	30.11	4.504×10^{12}	1.774	0.00858	5.43
Pluto	247.686	1.0040	39.44	5.896×10^{12}	17.17	0.25024	4.64

Table 2 Notes:

Sidereal Period is measured in *tropical years*, where one tropical year is the time for one Earth orbit with respect to the *Vernal Equinox*. The difference is due to Earth's *precession*, which lengthens its sidereal year by 1,230.7 s.
Synodic Period is measured (also in tropical years) with respect to the Earth frame-of reference. This would be the time between successive *oppositions* (for a *superior planet*) or solar *conjunctions* (for an *inferior planet*).
Semi-major Axis is the average orbital radius as measured from the center of the Sun to the center of the planet.

Table 3: Parameters of the Sun-Earth-Moon System

PARAMETER		UNITS	SUN	EARTH	MOON
MASS		kg	1.9891×10^{30}	5.9742×10^{24}	7.35×10^{22}
AVERAGE RADIUS		m	6.955×10^{8}	6.371×10^{6}	1.738×10^{6}
SURFACE GRAVITY		N/kg	274.4	9.807	1.67
ESCAPE VELOCITY		km/s	618	11.2	2.37
ORBITAL RADIUS:	PERIAPSIS	m	2.6×10^{20}	1.471×10^{11}	3.6326×10^{8}
	APOAPSIS	m	2.8×10^{20}	1.521×10^{11}	4.0555×10^{8}
	AVERAGE	m	2.7×10^{20}	1.496×10^{11}	3.84403×10^{8}
ORBITAL ECCENTRICITY		---	0. 05	0.01673	0.055
ORBITAL PERIOD (IN UNITS OF CONVENIENCE)			230 Myr	1.000... solar yr	1 moonth
AVERAGE ORBITAL SPEED		km/s	[a] 232	29.8	1.01

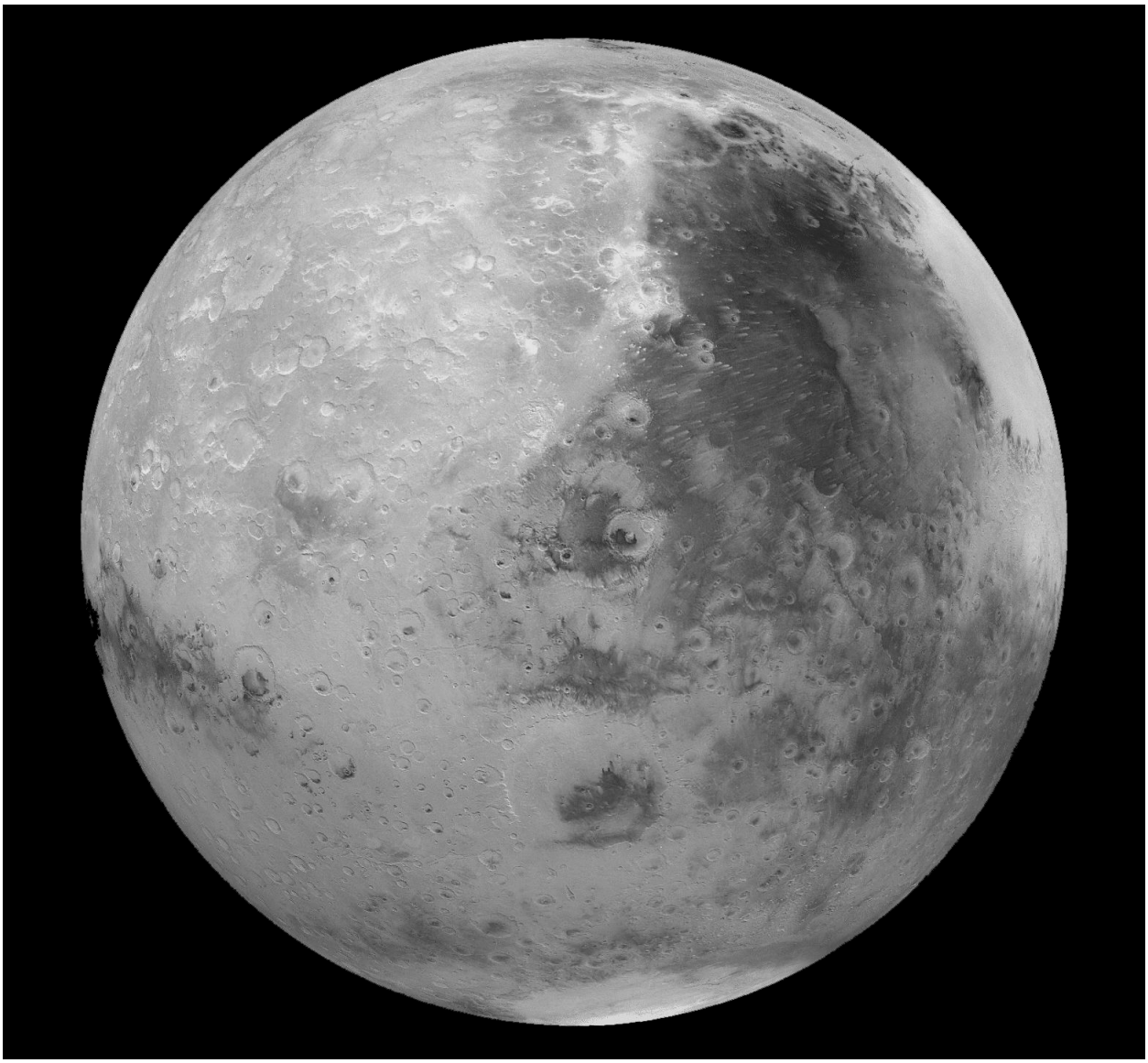
Table 3 Notes:

Orbital Data for the Sun describe its motion around the center of mass of the Milky Way Galaxy.
Orbital Period for the Earth, 1 *solar year* = 365.24219878 *mean solar days* (e.g., Vernal Equinox to Vernal Equinox)
Orbital Period for the Moon, 1 "moonth" = 27.32166 days (sidereal) = 29.53059 days (synodic)

Appendix 5: Global View of Mars

Behold the Syrtis Major hemisphere of Mars. The view is similar to that enjoyed from a spacecraft 2,000 kilometers above the surface. This mosaic is composed of about 100 Viking Orbiter images acquired in 1980 during the early northern summer on Mars. The center of this image is near latitude -1.0, longitude 306. Color variations have been enhanced by a factor of two. The large bright colored area, located at the upper left is known as Arabia. The dark area to the right of Arabia, called Syrtis Major Planus, is a low-relief *volcanic shield* probably composed of basalt. Bright white areas in the south, including the Hellas *impact basin* at extreme lower right, are covered by *carbon dioxide* frost. Regions to the west and south of Syrtis Major are heavily cratered and relatively old. The dark feature at lower left is Sinus Sabaeus. Mars is smaller than Earth, but its surface area is approximately equal to our total land area — 140 million km^2.

Image Processing by Jody Swann, Tammy Becker and Alfred McEwen, USGS Flagstaff, AZ



MARS IMAGE COURTESY OF NASA SPACE SCIENCE DATA CENTER AND USGS

Appendix 6: The Surface of Mars at Ares Vallis

Pathfinder landed in Ares Vallis, an ancient flood basin, at 19.5° N, 34° W. Features indicative of once-flowing waters include tumbled rocks and effluvial patterns, like the white stripe down the South Twin Peak (dubbed "Ski Slope" by NASA investigators). This view is to the west-south-west; the Twin Peaks are about 1 km distant and 50 meters in height.

Gravity on the surface is only 0.38 g, so a person who weighs 100 pounds on Earth would weigh only 38 pounds on Mars. The low gravity also allows formation of taller mountains, since their constituent rock weighs less. Consider: how many bricks could you stack in a pile before the weight of the pile crushes the bottom brick? Mars' tallest mountain is 44,000 meters!

Anyone who followed the Pathfinder mission heard of the rocks named "Yogi," "BooBoo" and "Barnacle Bill." NASA investigators named dozens of rocks around the site, many included in this image (Yogi and his friends are just to the right of this frame and not shown). Some of the names are fairly descriptive, but others are downright enigmatic. See if you can match the names on the list to their proper coordinates. It will help to get a print of the larger, color version of this view available in the IMAGES supplement. Your teacher can provide the answers.

ROCK NAMES		COORDINATES
Flat Top		**(CD,5.5)**
Flute Top	?	**(FG,3)**
Grommit		**(I,3)**
Halfdome		**(J,5.5)**
Hippo		**(J,6.5)**
Moe	?	**(K,3)**
Pumpkin		**(KL,4)**
Shark		**(L,3)**
Stimpy	?	**(V,8)**
Wedge		**(X,1.5)**

ON THE SURFACE ...

The reddish ground is reflected in the salmon-colored sky, pastel thin and only 0.01 *atm* pressure at (the now dry) sea level. The mix is 95% CO_2, 3% N_2, 2% Ar, with traces of O_2 and H_2O vapor.

It is dry and cold — polar winters hit -128 °C. An equatorial summer afternoon might get up to 17 °C, if the sky is clear. Dust storms are common in the summer.

Mars enjoys seasons just like Earth, but they last twice as long. Coincidentally, one Mars day is 24 hr, 37 min, 23 s; that's 1.026 Earth days and, by definition, "1 *sol*."

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z



0 1 2 3 4 5 6 7 8 9

SURFACE IMAGE COURTESY OF NASA/JPL/CALTECH

Appendix 7: A Short History of Robotic Exploration

It was recognized early on in the exploration of space that human emissaries required far more complex and expensive infrastructure than did machines. Without the additional constraints of life support and a return trip, the cost of exploration was reduced by about a factor of ten!

Thus began the research to apply the then-young fields of robotics and teleoperation to the exploration of other worlds and deep space. Not to diminish the accomplishments of Mariner, Pioneer, and Voyager, all of whom vastly increased our knowledge of other worlds, we will focus on that special breed of explorer — those who touched the surface of their target.

We present here a brief history of the robotic exploration of other worlds. Spanning over four decades, and in reverse chronological order, significant "firsts" in this field are noted along with mission data and commentary. The robotic explorers included are tabulated below:

ROBOT	YEAR	WORLD
RB5X	2000+	Earth (so far)
Sojourner	1997	Mars
Venera 13	1982	Venus
Viking 1&2	1976	Mars
Lunokhod 2	1973	Moon
Venera 7	1970	Venus
Lunokhod 2	1970	Moon
Luna 16	1970	Moon
Surveyor 1	1966	Moon
Luna 9	1966	Moon
Ranger 7	1964	Moon
Luna 3	1959	Moon
Luna 2	1959	Moon
Luna 1	1959	Moon (and beyond)

RB5X on Terra Firma

At present, RB5X has only explored its home planet Earth. It would like to go to Mars, but is still awaiting an invitation from NASA. Its drive system, best suited for flat and solid terrain, could become mired in dust or sand. However, the martian polar caps and lava plains would be ideal sites for RB5Xploration. This robot is capable of both autonomous and teleoperation.

RB5X's onboard processor is a National Semiconductor INS 8073, with an 8-bit CPU running at 4.0 MHz, and up to 26k of RAM. It utilizes 2 *I/O channels* for video and data and has a full array of sensors including IR and sonar. A fully articulated 3-axis arm allows the robot to manipulate its environment and acquire samples. Maximum speed is 0.12 m/s, with a range of about 2 km.

The power supply consists of three standard lead-acid batteries with a total capacity of 25 A·hr. Solar recharging options are currently under investigation by several student groups, as RB5X seems to prefer "clean" electricity drawn from *renewable* resources.

The image shows RB5X in a friendly encounter with an unidentified reptilian life form.

Dimensions: H = 58 cm, D = 33 cm
Mass: 10.9 kilograms



RB5X IMAGE COURTESY OF GENERAL ROBOTICS CORPORATION

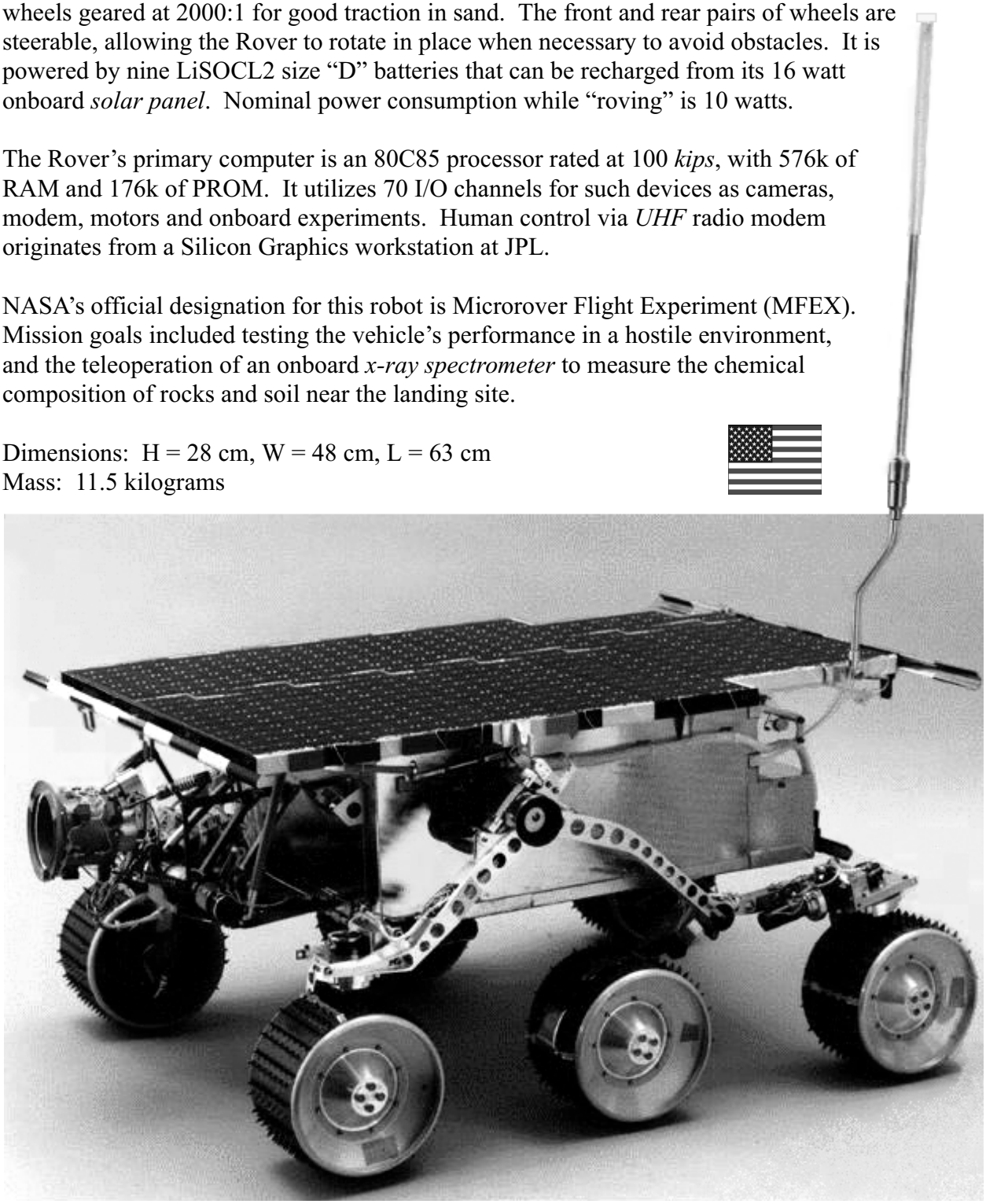
Sojourner Rover

Sojourner Rover was the payload of Mars Pathfinder. The craft landed at Ares Vallis on Mars, July 4th, 1997. The Rover can move at a top speed of 0.7 cm/s using six independently driven wheels geared at 2000:1 for good traction in sand. The front and rear pairs of wheels are steerable, allowing the Rover to rotate in place when necessary to avoid obstacles. It is powered by nine LiSOCL2 size "D" batteries that can be recharged from its 16 watt onboard *solar panel*. Nominal power consumption while "roving" is 10 watts.

The Rover's primary computer is an 80C85 processor rated at 100 *kips*, with 576k of RAM and 176k of PROM. It utilizes 70 I/O channels for such devices as cameras, modem, motors and onboard experiments. Human control via *UHF* radio modem originates from a Silicon Graphics workstation at JPL.

NASA's official designation for this robot is Microrover Flight Experiment (MFEX). Mission goals included testing the vehicle's performance in a hostile environment, and the teleoperation of an onboard *x-ray spectrometer* to measure the chemical composition of rocks and soil near the landing site.

Dimensions: H = 28 cm, W = 48 cm, L = 63 cm
Mass: 11.5 kilograms



SOJOURNER IMAGE COURTESY OF NASA/JPL/CALTECH

Venera

Venera 13 returned the first color images of the surface of another planet. It landed east of Phoebe Regio on Venus, March 1st, 1982, and survived about two hours (compare Venera 7).

Instruments included chemical and *isotopic* analysers, surface drills, seismometers and an x-ray fluorescence spectrometer. Its camera returned 14 color images of the surface.

During its descent through the atmosphere it measured gas composition, sunlight spectrum levels and electric discharges.

Dimensions: H ≈ 2 m, D ≈ 2 m
Mass = 750 kg



Venera 7 was the first spacecraft to return data from another planet. Launched by the Soviet Union, it touched down on the torrid surface of Venus on December 15th, 1970, landing south of the area known as Phoebe Regio.

Due to Venus' very thick atmosphere, only *aerobraking* and a parachute were needed to achieve a soft landing. It survived 30 minutes in 800 °F temperatures and 90 atm pressure. This is a pressure equal to that experienced at a depth of 1.5 miles in the oceans of Earth!

Instruments included a BW camera, chemical sensors for atmospheric composition, and an array of meteorological sensors. Its camera managed to send a single image of the surface before the extremely inhospitable conditions caused a total system failure.

It was the first of seven Venera series landers launched over a 12 year period.

Dimensions: H ≈ 3 m, D ≈ 1 m
Mass: 495 kg



VENERA IMAGES COURTESY OF NASA

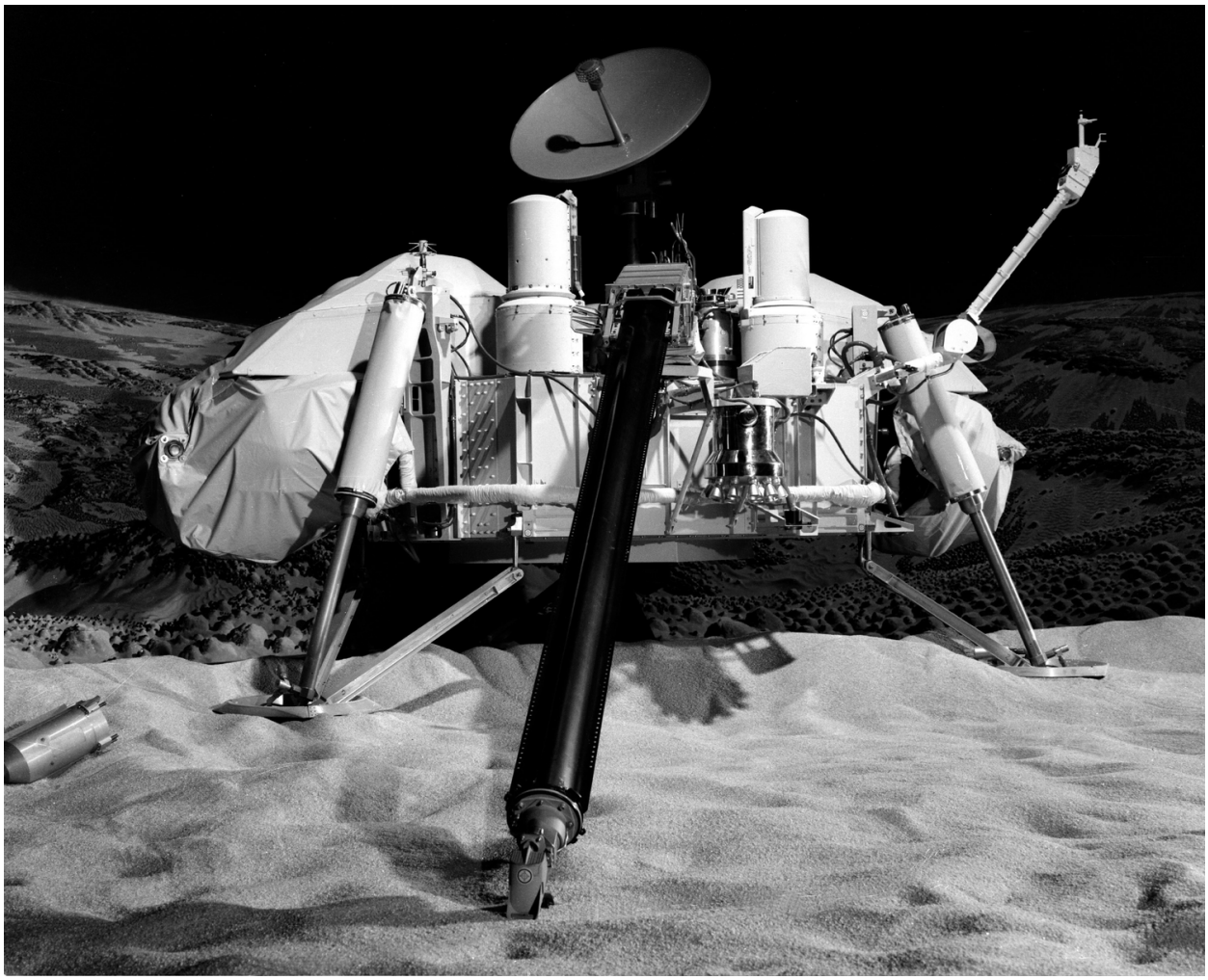
Viking

The first robotic mission to the Mars was NASA's Viking 1 Lander. Using aerobraking for initial deceleration, a parachute, and finally *retro-rockets*, it achieved a soft landing in Chryse Planitia on June 21st, 1976. It was followed by Viking 2 later that year.

Power was provided by two 70 watt RTG's with four rechargeable 8 amp-hour NiCad batteries for peak electric demands. Viking used a robotic arm for obtaining surface samples which were then tested for biological activity. Onboard instruments included a panoramic camera, weather station, *magnetometer* and *seismometer*. The Vikings were non-roving robots.

Viking's primary mission was to search for evidence of life, past or present, on Mars. It found no convincing evidence of either. The photo below shows a model on display at the Smithsonian Air and Space museum.

Dimensions: H ≈ 1 m, W ≈ 2 m, L ≈ 2 m
Mass: 600 kg



VIKING IMAGE COURTESY OF NASA/JPL

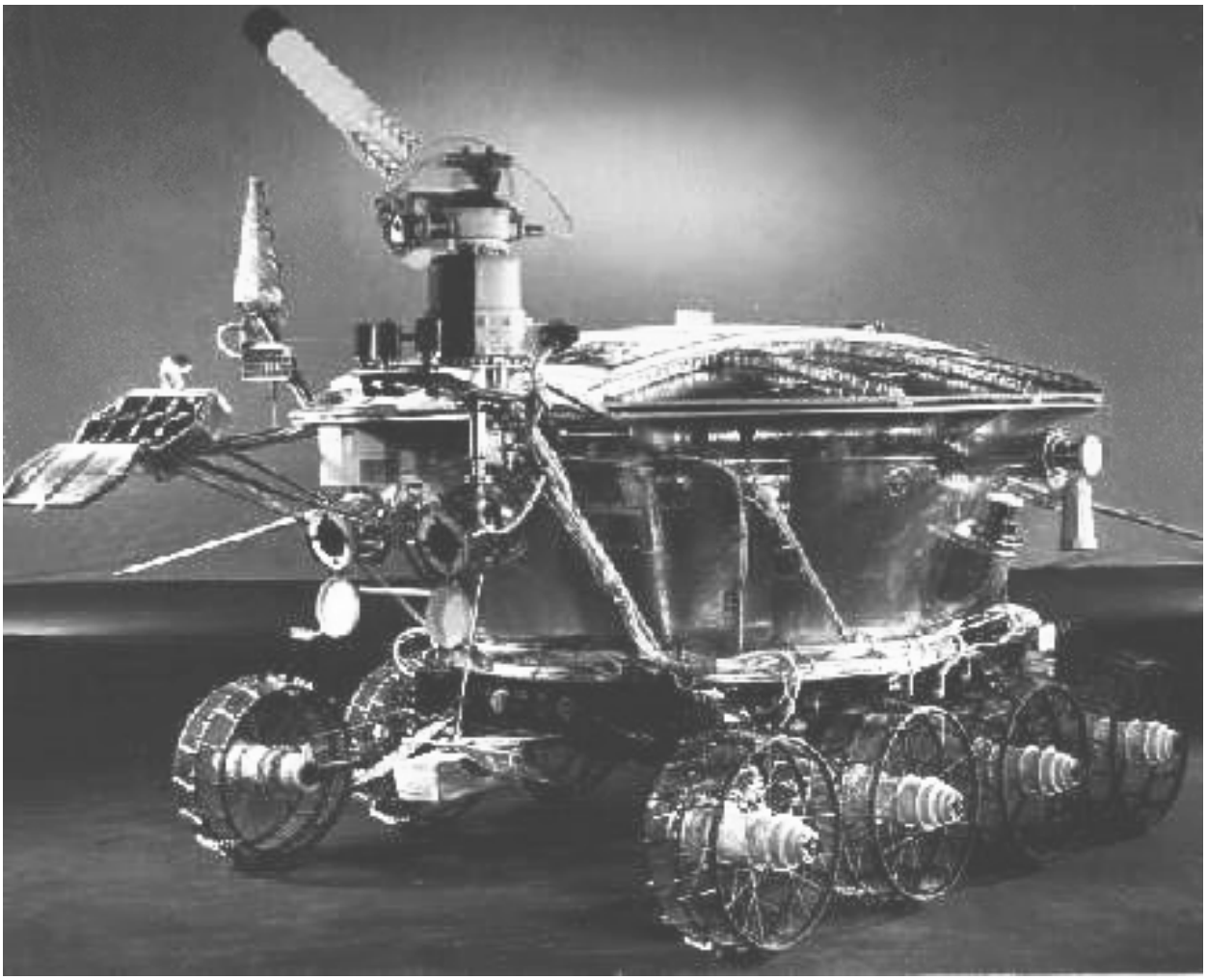
Lunokhod

Lunokhod 1 was the first mobile robot to explore the Moon. It was carried to the lunar surface as the payload of Luna 17, a Soviet spacecraft, on November 15th, 1970. Soft landing was achieved in Mare Imbrium (Sea of Rains) using retro-rockets for deceleration (parachutes only work in air). It was followed by a second rover, Lunokhod 2, in January of 1973.

Instrumentation included television cameras, extendable probes to test soil density and mechanical properties, an x-ray spectrometer and telescope, *cosmic-ray* detectors, and a *retro-reflector* for bouncing back a laser beam fired from Earth. Top speed was about 0.5 m/s, with power provided by a solar array for daytime operations only. During its 11 month exploration it traveled 10.5 km, returned over 20,000 TV images, and performed some 500 soil tests.

Lunokhod's mission was to collect images of the surface and measure light and radiation levels.

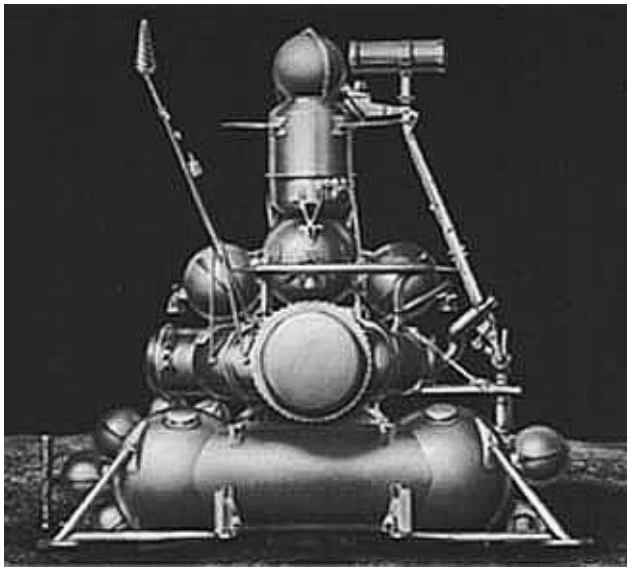
Dimensions: H = 1.4 m, W = 1.6 m, L = 1.7 m
Mass: 840 kg



LUNOKHOD IMAGE COURTESY OF NASA

Luna (Late Series)

Luna 16 was the first spacecraft to land on the Moon and return a sample to Earth. This non-roving robot landed near Webb crater in Mare Fecunditatis on September 20th, 1970. It used its robotic arm to acquire a 101 gram sample and load it into the *ascent stage* (the top 2/3 of the craft). The rocket reached Earth three days later, falling by parachute to a location in the outback of Kazakhstan.



LUNA 16 IMAGE COURTESY OF NASA

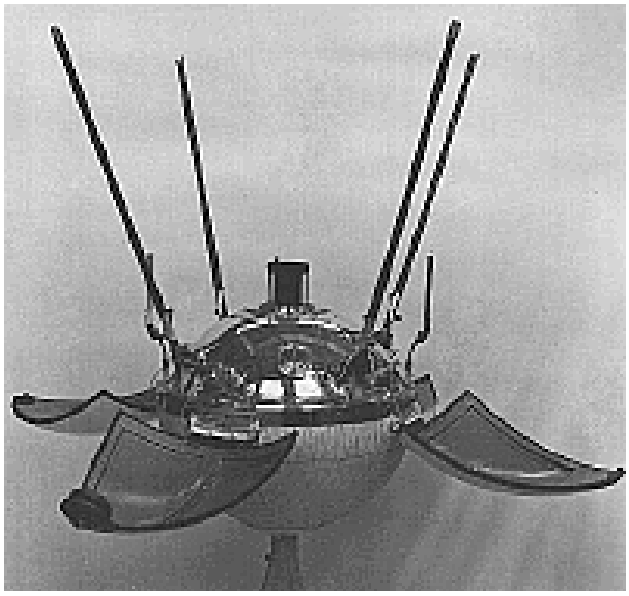
The first lunar rock samples were returned by American astronauts in July of 1969, but the first ever samples were returned by Luna 16.

Onboard instruments included temperature and radiation monitoring devices, and B/W television cameras.

Luna 16's primary mission was to return a sample of lunar soil to Earth.

Dimensions: H ≈ 3 m, D ≈ 3.3 m
Mass: 5600 kg

Luna 9 landed in Oceanus Procellarum on February 3rd, 1966. It was the first spacecraft to achieve a soft landing on the Moon, using an air bag to cushion its impact with the lunar surface. Four "petals" that formed the shell of the craft then opened to provide stability and allow deployment of spring-loaded antennas.



LUNA 9 IMAGE COURTESY OF NASA

Its modest scientific instrumentation was supplemented by an internal camera that imaged the surrounding area using a rotating and tilting mirror (visible top center). A panoramic view from the landing site was thus returned.

Luna 9's mission was to obtain images of the Moon's surface and transmit them to Earth.

Dimensions: D ≈ 1 m
Mass: 99 kg

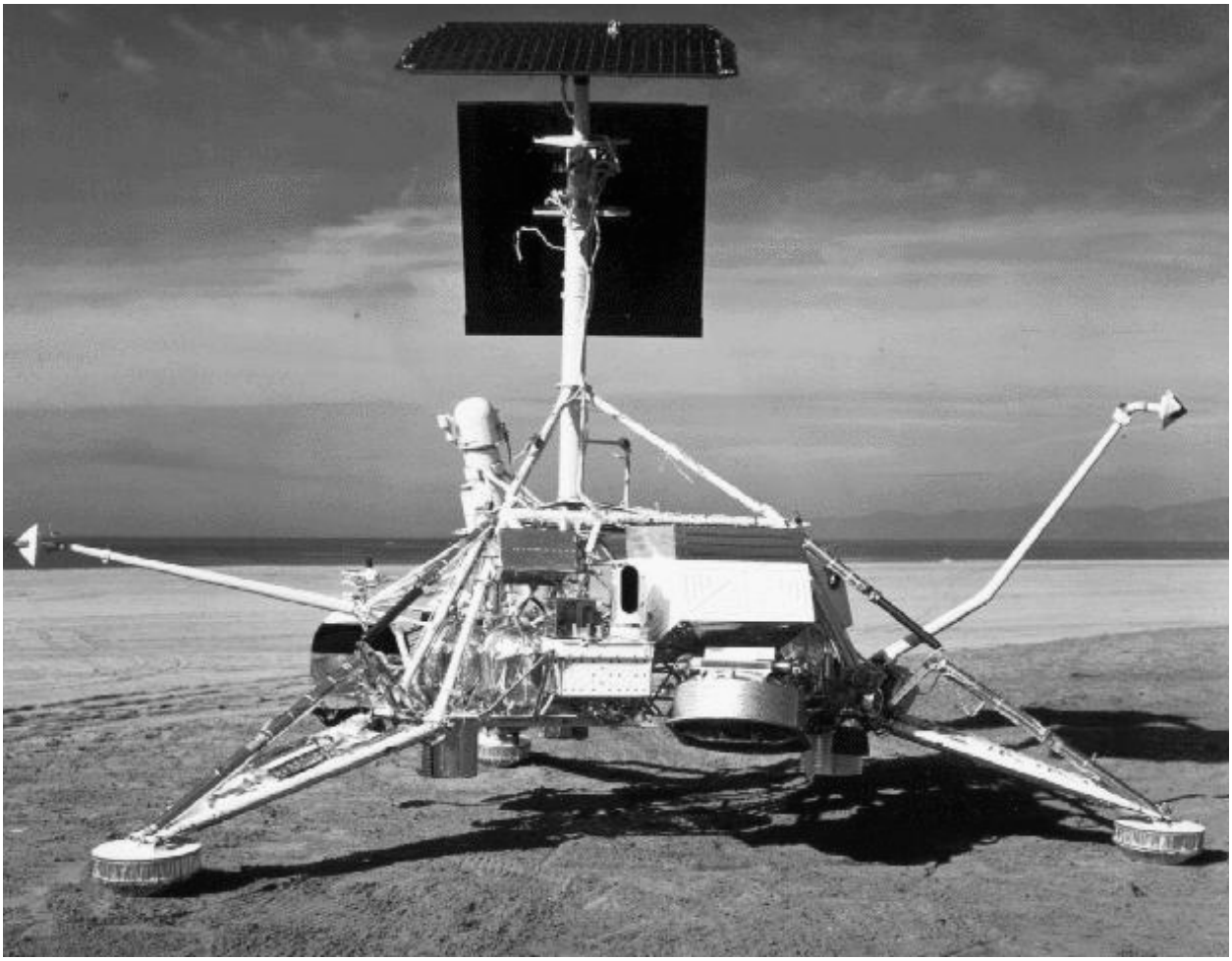
Surveyor

The Surveyor probes were the first American spacecraft to achieve soft landings on the Moon. Surveyor 1 arrived June 2nd, 1966, landing near the lunar equator. It was followed by five others over the next couple of years. All were of similar design and mission.

These non-roving robots carried high-resolution cameras, radiation monitors, soil sampling drills and chemical analyzers. Surveyor 3 included an *articulated* "scratcher arm" for increased sampling depth, and demonstrated limited mobility by "hopping" 2.5 m laterally using a controlled burst from its thrusters. Power was provided by an 85 watt solar panel, supplemented by batteries that could provide a peak power of about 4 *kilowatts*. The full-scale model pictured below was photographed on a beach near Los Angeles.

Surveyor's primary mission was to gather information about the lunar surface in preparation for the imminent manned missions to the Moon.

Dimensions: H = 3.3 m, D = 4.5 m
Mass: 1000 kg



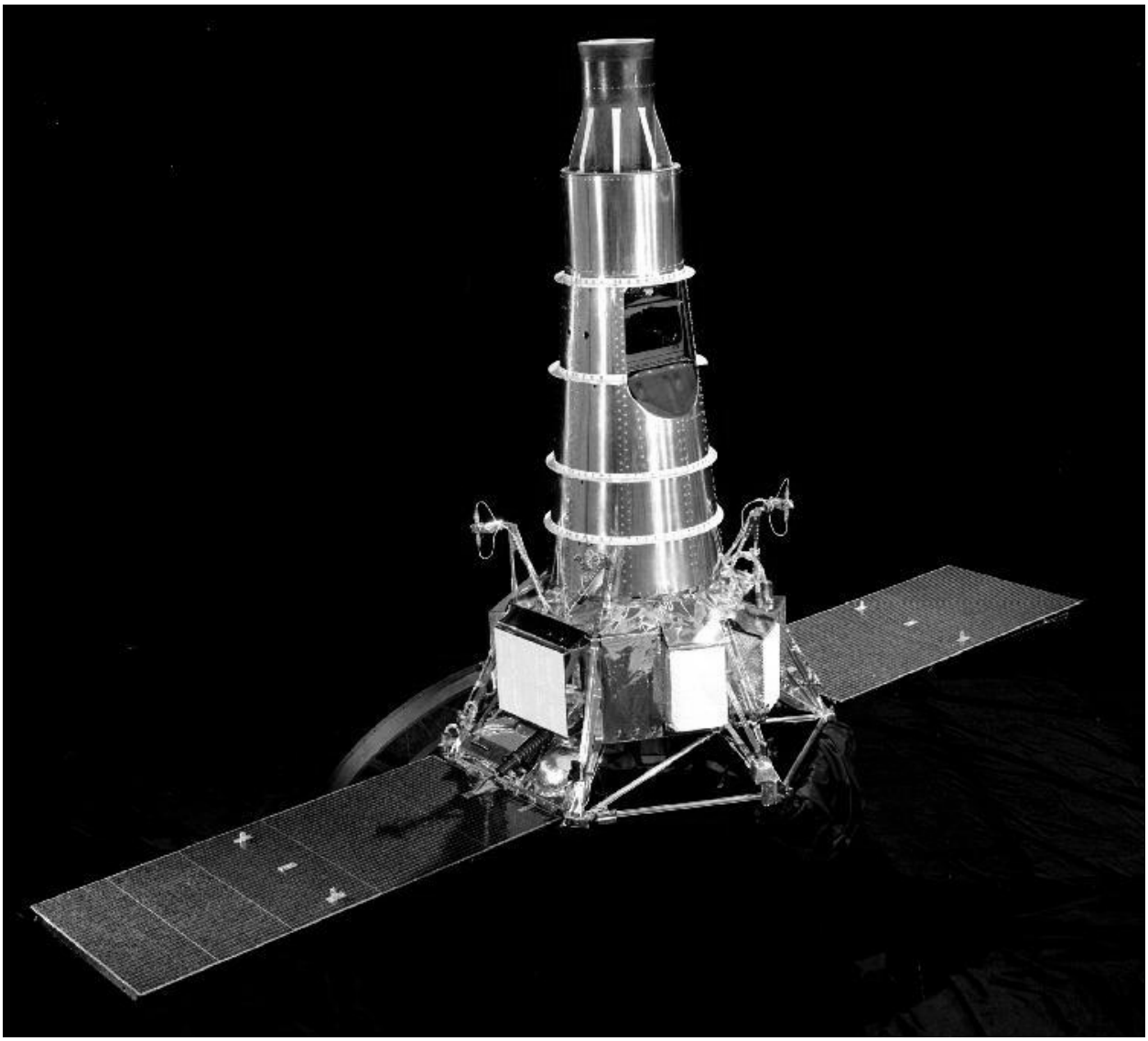
SURVEYOR IMAGE COURTESY OF NASA

Ranger

From 1961-65 nine Rangers were fired straight at the Moon, programmed to take pictures through six onboard cameras right up to the moment of impact. The final picture was typically shot about 0.3 seconds before impact at an altitude of 2000 ft. Some Rangers failed at launch, others missed the Moon entirely. Ranger 7 was the first success; on July 28th 1964, it returned surface images 1000 times more detailed than the best telescopic efforts. Power was provided by a 200 watt solar panel and 3600 *watt-hour* AgZnO battery pack.

Ranger's mission was to take the first detailed, close-up images of the lunar surface. These new images helped NASA planners choose landing sites for the manned missions of 1969.

Dimensions: H = 3.6 m, W = 4.6 m (including solar panels)
Mass: 362 kg



RANGER IMAGE COURTESY OF NASA

Luna (Early Series)

Luna 3 entered into a low lunar orbit, altitude 6200 km, on October 7th 1959. From this new vantage point it captured the first images of the Moon's never-before-seen "dark side."

The images were obtained using conventional photographic processes. Film was developed and scanned onboard, and then transmitted in FAX form back to Earth. Computer enhancement of these crude images allowed the first far-side maps to be constructed. This is why most far-side features have Russian names.

Other instrumentation included micrometeor and cosmic ray detectors. It was stabilized by a gyroscopic attitude control system. Power was provided by a battery and solar cells.

Luna 3's primary mission was to capture and return images of areas of the Moon's surface not visible from Earth.

Dimensions: H = 1.3 m, D ≈ 1 m
Mass: 279 kg



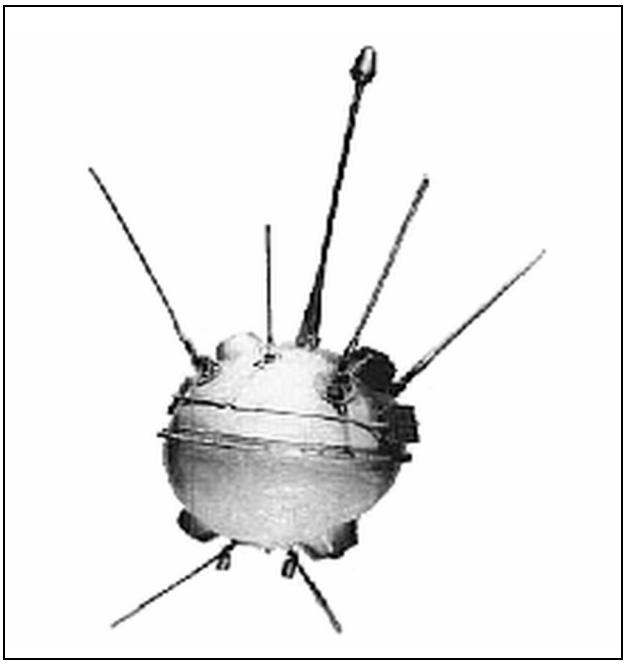
LUNA 3 IMAGE COURTESY OF NASA

Luna 2 became the first man-made object to touch the Moon on September 14th, 1959. It impacted the lunar surface in Mare Serenitatis near Archimedes Crater and deployed Soviet pennants to mark the achievement.

This craft contained no propulsion systems; it was, essentially, a ballistic projectile with the Moon as its target. Enroute it released a cloud of sodium gas visible from Earth, analysis of which proved the Moon lacked any significant magnetic field or radiation belts. Further data was provided by geiger counters, magnetometers, and micrometeor detectors.

Luna 2's primary mission was to hit the Moon.

Dimensions: R ≈ 1 m
Mass: 390 kg



LUNA 2 IMAGE COURTESY OF NASA

Luna 1 became the first man-made object to achieve a lunar "fly by" on January 4th, 1959. Like Luna 2, it was a ballistic projectile aimed at the Moon. At its closest approach it passed within 5995 km of the lunar surface.

Luna 1 was intended to hit the Moon. This is inferred from the fact that it was loaded with dozens of metallic tokens, bearing the Soviet coat of arms, engineered to scatter around the site at impact — redundant evidence of their accomplishment and, no doubt, prized trophies for future lunar explorers.

But Luna 1 missed the Moon. Its great speed propelled it into an elliptical "graveyard orbit" around the Sun, somewhere between the orbits of Earth and Mars. Its current orbital elements are unknown; Luna 1 is considered "lost."

Instrumentation and experiments were similar to Luna 2, providing data on Earth's radiation belts and the solar wind.

Dimensions: R ≈ 1 m
Mass: 361 kg



LUNA 1 IMAGE COURTESY OF NASA

A Short History of Robotic Exploration: Conclusions

From Luna to Sojourner we can see the evolution of robotics mirrored in the capabilities of these intrepid explorers. Luna 1 was little more than a data relay device — a remote sensor— a weather balloon in space. Sojourner Rover was mobile, teleoperable, had an "arm," and utilized artificial intelligence for semi-autonomous operation. Yet all qualified as "robots" in the sense that they performed tasks otherwise requiring human presence or intervention.

What we have learned over these 40-plus years of robotic exploration goes far beyond the realm of astronomy and planetary science. We have also learned how to make better robots. Missions previously thought to require human explorers are being increasingly handled by these mobile, autonomous machines. One might ask "Is there really any need to send humans to other worlds, given how well the robots are doing?" The answer to that question is likely to remain "Yes." for the foreseeable future. Regardless of developments in AI, the "need" comes not from the mission goals or budget constraints, but from within those explorers who build the robots. The robots may lead, but we shall surely follow.

Appendix 8: URL's You Can Use

Research on the internet can be time-consuming. If you are looking for information related to the activities in Mission to Mars, these websites will be a good starting point. There is an incredible amount of robotics research currently underway. Much of this work is in areas on the cutting edge of technology. Examples include:

- artificial intelligence
- visual pattern recognition
- self propulsion
- synthetic muscle
- sample acquisition

The following URL's were valid at the time of this printing (5/00). If you installed this Manual on a computer, and are currently online, these URL's will be active links. Get to any one of them and you'll probably find links to all the others.

Sites Related to Robotics:

Carnegie Mellon University Robotics Institute: http://www.ri.cmu.edu
JPL Robotics Lab: http://robotics.jpl.nasa.gov
Harvard Robotics Lab: http://yellowstone.harvard.edu
Stanford Robotics Lab: http://robotics.stanford.edu
Caltech Robotics Lab: http://robby.caltech.edu
MIT Artificial Intelligence Lab: http://www.ai.mit.edu
Sandia Labs Intelligent Micromachine Initiative: http://www.mdl.sandia.gov/Micromachine

Sites Related to Planetary Exploration:

NASA Home Page: http://www.nasa.gov
NASA Photo Archive: http://photojournal.jpl.nasa.gov
NASA Historical Archive: http://www.ksc.nasa.gov/history/history.html
National Space Sciences Data Center: http://nssdc.gsfc.nasa.gov/planetary/planetary_home.html
JPL Mars Exploration Center: http://mpfwww.jpl.nasa.gov

Sites for General Robotics Corporation:

Products and corporate information: http://www.generalrobotics.com
RB5X educational forum: http://www.edurobot.com

Appendix 9: Glossary

aerobraking	the use of air drag to decelerate a spacecraft in preparation for orbital insertion or landing, and an alternative to fuel-guzzling retro-rockets
amp-hour	quantity of charge delivered or received when a current of 1 ampere flows for one hour; exactly equal to 3600 coulombs; usually used for rating the storage capacity of batteries
articulated	able to flex, bend or rotate; opposite of rigid; used to describe mechanical components that mimic the action of human limbs
artificial intelligence	computer processing that mimics the action of the human mind, including such functions as learning, intuition, modeling, visual processing
ascent stage	separable component of a spacecraft used to regain orbit after landing
atm	abbreviation for "atmosphere," a quantity of pressure; 1 atm equals the average sea-level pressure of Earth's atmosphere; $\approx 1.0 \times 10^5$ N/m^2
atmosphere	the gaseous part of a planet (see also *lithosphere* and *hydrosphere*)
autonomy	ability to function without outside assistance, including, but not requiring, artificial intelligence, hazard avoidance, and self-maintenance
booster	temporary thrust-producing module used for brief speed changes, often ejected after use for disposal or refilling
calibrate	the process by which numerical performance specifications are obtained for mechanical and electrical devices
carbon-dioxide	CO_2, a component of most atmospheres; can be liquid or solid under suitable conditions of pressure and temperature
cardinal direction	one of the 4 primary compass directions (North, East, South, West)
cartographer	a person who designs or studies maps, globes or geological models
center-of-gravity	point in space representing the average position of matter in a solid body; equivalent to "center-of-mass" in most situations
conjunction	point in the motion of two astronomical bodies where they have equal celestial longitudes, but not necessarily equal celestial latitudes
convention	universally adopted agreement about how to measure or define an otherwise arbitrary quantity, e.g., the decision to label the equator latitude 0° is an international cartographic convention
coordinate system	geometric reference structure in any number of dimensions that allows sets of numbers to represent the locations of points
cosmic ray	very high energy atomic nuclei or electrons traveling nearly at the speed of light; believed to originate from catastrophic cosmic events
cybernetics	area of science concerned with similarities and differences between the human brain and electronic processors designed to model it
effluvial	related to the flow of water and its effects on topography
exogeology	area of science concerned with the geology of other planets, especially in the sense of comparison with Earth's own geology
exobiology	area of science concerned with the biology of other planets, especially in the sense of comparison with Earth's own biology
exolexicologist	not an official word, but an extrapolation of a legitimate word meaning "someone who studies words related to things beyond the Earth"
feedback	in general, the use of an output to modify or control an input

flywheel	device that stores energy in the form of rotational kinetic energy, most often through the use of high-rpm, high-density discs
friction	resistive force produced when two bodies in contact move relative to one another; can be rolling, sliding or viscous in nature
fuel cell	type of battery that stores energy in the form of dissociated chemical compounds, releasing electrical energy by chemical reaction; easily recharged and more efficient than standard electrolytic batteries
g (or gee)	1g is the amount of acceleration equal to 9.8 m/s^2, or amount of inertial reaction (also called g-force) equal to 9.8 N/kg; normal Earth gravity
geology	the area of science that investigates the relationships and evolution of lithospheric features on Earth (*geo* ≡ earth and *ology* ≡ study)
gravimetric	related to the measurement of gravitational forces or anomalies
heuristics	area of study concerned with the process by which the human mind learns how to solve problems; also applied to machine AI
hydrosphere	the liquid part of a planet (see also *lithosphere* and *atmosphere*)
igneous	rocks or minerals formed by high-temperature geologic processes
impact basin	flat, depressed, usually circular area created by the impact of a meteor
inferior planet	planet in an orbit closer to the Sun than the observer's planet
internal friction	friction generated within the drive train and motor of a machine
I/O channel	data link between a computer and some outside device
iron-oxide	most often Fe_2O_3, commonly known as "rust"
isotope	atomic nucleus having more or less than the average atomic weight due the an excess or deficiency of neutrons; typically less stable and hence more radioactive than the average nucleus
joule (J)	amount of energy required to exert a force of 1 N over a distance of 1 m [1 N·m, mechanical], or supply 1 W of power for 1 s [1 W·s, electrical]
kilowatt (kW)	amount of power equal to 1000 watts
kinematics	area of study concerned with the mathematical description of motion
kips	abbreviation for "thousands (kilo) of instructions per second;" usually used to rate the command processing speed of a computer
lithosphere	the solid part of a planet (see also *atmosphere* and *hydrosphere*)
laser	acronym for "light amplification by stimulated emission of radiation"; optical device that produces a beam of monochromatic coherent light
latitude	measure of distance, in degrees north or south of the equator
light-minute	distance traveled by light (or by a radio wave) in a time of 1 minute; equal to 1.8×10^7 km or about 11 million miles
longitude	measure of distance, in degrees east or west of the prime meridian
magnetometer	device used to measure the strength of a magnetic field; is most often calibrated to provide readings in units of "gauss" or "tesla"
metamorphic	rocks or mineral formed by high-pressure geologic processes
N_2 propulsion	system that produces thrust by venting high-pressure nitrogen gas; used for low thrust applications; safe, non-reactive exhaust; highly reliable
newton (N)	amount of force required to accelerate 1 kg of mass at a rate of 1 m/s^2
odometer	mechanical or electrical device used to measure total distance traveled
opposition	point in the motion of an astronomical body where its celestial longitude is 180 degrees away from the Sun

ordered-pair	set of two numbers representing the location of a point in a plane, and measured with respect to a 2-dimensional coordinate system
pattern recognition	sub-area of AI concerned with the process by which visual information is acquired, processed and interpreted
permafrost	water frozen permanently into the soil at a depth not affected by seasonal temperature variations
photovoltaic	physical process by which the radiant energy of a photon is converted into electrical energy; principle of operation underlying solar cells
precession	the circular drift or wobble of the axis of a rotating body caused by the application of a force off-axis to the body's rotation
primordial	relating to the distant past, usually in the era during which astronomical bodies were formed, e.g., primordial Earth 4.5 billion years ago
Pythagorean Theorem	mathematical relation between the sides of a right triangle relating the sides A and B to the hypothenuse C with: $C^2 = A^2 + B^2$
radar	acronym for "radio navigation and ranging"; process by which radio waves reflected from a remote body can be used to determine its distance and location relative to the radio transmitter
ranging	determination of the distance to a remote object by optical, acoustic or electromagnetic reflection
receiver	device designed to receive and process a signal from a transmitter
regenerative braking	method of deceleration accomplished via transfer of kinetic energy to a mechanical or electrical storage device, as opposed to frictional braking where pads or discs are pressed together converting energy into heat
renewable	as applied to an energy source, one that is replenished by natural processes with no action on our part, e.g., solar, wind or hydro energy
retrograde	rotation or revolution in the opposite direction defined by convention
retro-reflector	tetrahedronical mirror-like device capable of reflecting a signal exactly back to the source regardless of the angle of incidence
retro-rocket	thruster designed to fire in the direction opposite to normal motion and used for deceleration of spacecraft
rolling friction	friction created by the action of a wheel rolling, but not sliding, on a surface; function of both surface smoothness and wheel diameter
RF	abbreviation for "radio frequency"; that part of the electromagnetic spectrum below about 10^{13} Hz (or with wavelengths > 1 cm)
RTG	abbreviation for "radioisotope thermoelectric generator"; device that produces power by converting the heat from a radioactive substance, usually thorium, directly into electrical energy with no moving parts
sedimentary	rocks or mineral formed by erosive and layering geologic processes
seismometer	device designed to measure ground movement relative to an inertial system, usually a large mass mechanically attached to a recorder
SETI	acronym for "Search for Extra-Terrestrial Intelligence"; the ongoing process, begun in the 1950s, to find hard scientific evidence for the existence of other technological civilizations in our galaxy
sidereal	measured with respect to the motion of the stars
simple machine	mechanical device used to multiply force or speed; classic examples include the lever, inclined plane, wheel and axel, pulley and screw

sol	one apparent day measured with respect to the Sun; on Earth, 1 sol is 86,400 seconds; on Mars, 1 sol is 88,643 seconds
solar cell	thin, flat wafer of material, usually silicon doped with germanium, or gallium doped with arsenic, in which the photovoltaic effect converts the energy of an absorbed photon into electrical energy
solar panel	flat, usually rectangular, array of solar cells; used to convert sunlight into electrical energy at 12, 24 or 48 VDC
sonar	acronym for "sound navigation and ranging"; process by which sound waves reflected from a remote body can be used to determine its distance and location relative to the sound transmitter
STS-X	abbreviation for "Space Transportation System - mission #X", and the official NASA designation for the Space Shuttle
superior planet	planet in an orbit farther from the Sun than the observer's planet
synodic	measured with respect to the motion of Earth
talus	rocky debris eroded from a mountain by wind, rain or temperature
teleoperation	the process of controlling a device in a remote location, usually by means of 2-way radio communication
telepresence	the process of observing a remote location with multiple sensory input, including visual, aural and sometimes tactile information, usually by means of 2-way radio communication
terrestrial	relating to the Earth (Terra)
transmitter	device designed to process and send a signal to a remote receiver
triangulation	geometric method for locating observer's position in a plane; requires directional and coordinate information for two remote points
tropical year	amount of time for Earth to complete one orbit with respect to the Vernal Equinox; basis for the definition of the astronomical "second"
UHF	abbreviation for "ultra-high frequency"; that part of the RF spectrum with frequencies between 3×10^8 and 3×10^9 Hz; useful for video
URL	Universal Resource Locator, an internet address, e.g., http://www.etc.
Vernal Equinox	point on the celestial sphere of 0° longitude and 0° latitude, currently located in the constellation Pisces; changes slowly due to precession
virtual reality	artificial environment produced by a system of computers and sensory feedback transducers (for sight, hearing and touch) that interacts with the user in a real-time mode
volcanic shield	topographic feature consisting of a broad, flat, gently uplifted layer of igneous material, usually basalt, through which volcanoes vent
volt	amount of electromotive force that imparts an energy of 1 J to 1 C of electric charge
watt (W)	amount of power equal to an energy consumption rate of 1 J/s or 1 V·A
watt-hour (W·hr)	amount of energy equivalent to 1 W of power acting for 1 hr of time; exactly equal to 3600 J
x-ray	that part of the electromagnetic spectrum between frequencies of about 10^{17} and 10^{20} Hz; harmful to most life forms
x-ray spectrometer	device used to detect and measure x-rays; provides data in the form of x-ray intensity vs. frequency; used to identify species of nuclei present in a sample of matter

Appendix 10: Programming Your RB5X

Platform Conventions

The procedure for interfacing your RB5X with a computer is detailed in your Reference Manual and is platform-specific. The procedure given here is for the PC DOS environment, running any version of DOS ≥ 2.0. In other environments, the user will have to interpret these procedures with respect to platform-specific instructions in the Reference Manual.

Connecting RB5X to Your Computer

Before you do anything else, make a backup copy of your RCL disc. Programs can only be saved to this disc. When it's full (1.44 Mbytes) you will need another to continue. Do this now.

The first step is to connect RB5X to your computer and establish a two-way communication link. Follow these steps in the exact sequence indicated:

1. Connect RB5X to your computer's COM1 port using the RS-232 cable. COM1 is RCL's default setting. If COM1 is already used by another device, you can reset RCL to use COM2 by running the routine described in Appendix 11 (Calibrating Your RB5X).
2. Ensure the RS-232 switch next to RB5X's battery level indicator is set to "STD" (standard).
3. Carefully remove any EPROM module installed in the socket on RB5X's interface panel — this is a memory chip, so take the usual antistatic precautions. You can leave it installed if, after powering-up the robot, you press bumpers 2 and 4 to enter the command mode.
4. Turn on RB5X.
5. Turn on your computer and boot to DOS.
6. Insert the "RCL for DOS" disc into the floppy disc drive.
7. Type: A: <RET>
8. Type: RCL <RET>

This will launch RCL and display the title screen along with system information. RCL will then prompt you for a command by printing:

9. *WHAT WOULD YOU LIKE ME TO DO NOW?*

This is the main command mode prompt from RCL. You could, at this point, create a new program, or edit an existing program. However, before proceeding you should verify that two-way communication between your computer and RB5X has been established.

10. Type: DUMBO <RET>

If you get the error message *ROBOT AND COMPUTER ARE NOT COMMUNICATING!!!*, first be sure the robot is turned on and in "command mode" (press bumpers 2 and 4). Then check the cable connections, communications port calibration, and your Reference Manual (in that order). If you get the message *COMMUNICATION WITH ROBOT IS WORKING*, all systems are go. Enter F7 (the function key) to return to RCL.

About RCL

RCL (Robot Control Language) is an easy and intuitive programming environment. It is powerful enough for most robot control functions without having to resort to NSC Tiny Basic subroutines. We strongly recommend you acquire and refer to the GRC publication "Robot Control Language with Savvy® — Programmers Manual" for a complete listing of RCL commands and syntax.

If you've done programming in other languages you'll need to relearn some vocabulary. In RCL the words "program" and "command" are replaced by the single word "task," since its language commands are themselves programs (in NSC Tiny Basic). Other differences will be come clear as you work with the language. It is strongly recommended that even experienced programmers go through the steps of the sample program provided later in this section. It will acquaint you with the most important procedures and vocabulary for RCL programming.

An important feature of RCL is the way it accepts and processes individual program lines as they are entered. RCL is a dynamic programming environment, thus when you enter a command you often need enter only the first word — RCL will fill in the rest and provide the input fields for the parameters needed. In some cases, a few more words may be required to specify the command uniquely. For example, to command the robot to move forward 5 feet you would enter:

1. MOVE DISTANCE FORWARD <RET>

and RCL will respond with:

2. *MOVE DISTANCE FORWARD FOR THIS MANY FEET ___*

You then enter a "5" in the distance field, hit <RET>, and the line is complete. In fact, if you try to enter the entire line with distance included, RCL will not accept it.

To program a loop, you need to enter more than the just first word since there are three different types of loops, all of which start with the word BEGIN, i.e.,

BEGIN A LOOP
BEGIN A COUNTED LOOP
BEGIN A LIMITED LOOP

Programming Conventions Used in this Document

- Text typed in Arial caps is text you actually type, e.g., PREPARE THE ROBOT.
- Text typed in Arial italics is a prompt or output from RCL, e.g., *INVENT A TASK CALLED ___*
- The symbol <RET> means you should press the "Return" or "Enter" key.
- Anything in square brackets is an argument (number, variable, or function) you can specify, e.g., MOVE DISTANCE FORWARD THIS MANY FEET [5]. Note: The brackets are not entered.

- Though not shown in the examples provided, all variables entered in an RCL command must be placed in quotes, e.g. SET "B" EQUAL TO 3. The same rule applies to the logical operators (<, >, =, <>). Numbers can be entered without quotes.

Using Variables in RCL

RCL by itself is fine, provided all you need to do is program motions or voice events. However, the use of variables (the key to higher-level programming) requires inclusion of NSC Tiny Basic subroutines. Many of the RCL commands that are designed to handle variables are not functional and generate errors when used. Fortunately, NSC Tiny Basic provides the means for arithmetic and logical operations not otherwise possible.

There are 26 single-letter variables (A-Z) available for use by RCL. Some may already be taken by primary RCL tasks, so to avoid conflicts use only A,B,C,E,F,G,H,J and U. Nine variables are usually enough for all but the most complex problem solving. If you really need more, you can easily check which variables are in use by entering the command:

```
SHOW VARIABLES
```

which will display a screen showing the variables in use, and those still available, in the program most recently listed or edited. Other variables will occasionally be available, depending on what internal tasks RCL is using to run your program.

Two important functional commands for manipulating variables in RCL are:

```
RESERVE THE TINY BASIC VARIABLE [VARIABLE]
CLEAR ALL ITEMS
```

The first command is required at the start of a program to enable the use of a variable. The second sets all variables equal to zero (including any variables currently in use by NSC Tiny Basic).

One of the most powerful functional commands in RCL is:

```
SET THE TINY-BASIC VARIABLE [VARIABLE] EQUAL TO [EXPRESSION]
```

which can be used to set a variable equal to any valid integer (-32767 to +32767) or to a variable or variable expression. It will generate a non-fatal *REPORT MATH ERROR!* message in the latter two cases, but this will not affect the operation of the program (i.e., it's non-fatal). Examples of legal SET statements include:

```
SET THE TINY-BASIC VARIABLE [A] EQUAL TO [3]
SET THE TINY-BASIC VARIABLE [A] EQUAL TO [B]
SET THE TINY-BASIC VARIABLE [A] EQUAL TO [(B+C)/2]
```

The following command sequence is also functional, and can be used to create loops that repeat or terminate on variable values. Again, the use of a variable in more than one field of the same RCL command (i.e., line 3) will generate a non-fatal *REPORT MATH ERROR!* message.

1. BEGIN A LIMITED LOOP
2. (do something)
3. REPEAT THE LIMITED LOOP UNLESS [VARIABLE or VALUE] IS [<] TO [VARIABLE or VALUE]

Another functional loop sequence is the counted loop:

1. BEGIN A COUNTED LOOP CALLED [VARIABLE] BEGINNING AT [VARIABLE or VALUE] ENDING AT [VARIABLE or VALUE]
2. (do something)
3. REPEAT THE COUNTED LOOP FOR THE COUNTER [VARIABLE]

Note that when you use a counted loop, the underlined VARIABLE must be the same in both lines.

Unfortunately, there are some non-functional RCL commands that would seem to severely restrict what can be done with this language. In case you haven't already discovered them, they are:

```
CALCULATE [VARIABLE] = [VARIABLE or VALUE] [ + , - , * , / ] [VARIABLE or VALUE]
TEST IF [VARIABLE or VALUE] IS [ = , > , < , <> ]  COMPARED TO [VARIABLE or VALUE]
```

Fortunately, we can recover these important operations. By using the COMPILE command we can insert equivalent NSC Tiny Basic statements into the program. The form of this command is:

```
COMPILE THE BASIC STATEMENT [STATEMENT] WHICH MEANS [COMMENT]
```

The argument STATEMENT can be a calculation using variables and integers, such as

```
A=2
A=A+1
A=(A+B)*C/100
```

or any other legal operation. All calculated values will be truncated (rounded down) to integers. It is possible to include more than one STATEMENT in a single COMPILE command by using a colon (:) to separate individual STATEMENTS. For example:

```
COMPILE THE BASIC STATEMENT [A=3:IF A.>B GOTO 42] WHICH MEANS [COMMENT]
```

There is no limit to the number of individual STATEMENTS that can be included within a single COMPILE command, save for that imposed by program size restrictions.

The argument STATEMENT can also be a logical conditional such as

```
IF A<=10 GOTO 50
IF A=B GOTO 50
IF (A>3) AND (C=3) GOTO 42
```

Note that in compound conditionals such as the last example, each conditional must be enclosed in parentheses. Conditional statements can be used to simply jump to another line number in the program, or to terminate a loop by jumping out of it.

When using a GOTO statement in a COMPILE command you cannot GOTO an RCL line number. The line number specified must be the line number in the compiled NSC Tiny Basic version of the program. To see what that line number will be, temporarily enter a dummy line number like 333.

Then go into the RCL command mode and enter:

1. BUILD <RET>
2. *BUILD A PROGRAM FROM THE ROBOT TASK* ___
3. [NAME OF YOUR PROGRAM] <RET>

with the name of your program placed in quotes. Your RCL program will be compiled into NSC Tiny Basic. To list the program showing its NSC Tiny Basic commands and line numbers, enter LTP <RET> and scroll around to discover the actual line numbers. You can then return to RCL and edit the dummy line numbers accordingly.

Finally, the argument COMMENT in the command

```
COMPILE THE BASIC STATEMENT [STATEMENT] WHICH MEANS [COMMENT]
```

is simply a comment by you regarding the function of the command, such as COMPARE A&B or CALCULATE C. These comments can be helpful when reviewing and debugging programs.

Some of these methods may seem awkward, but they are the only way to regain full programming functionality. As you acquire experience using variables you will see the real problem-solving potential of this language, for it is the logical processing of variables that allows an autonomous machine to respond intelligently to changes in its internal state or external environment.

Using Random Numbers in RCL

It is sometimes desirable to incorporate random elements into a program. Artificial intelligence problem-solving algorithms often require this feature. The COMPILE statement provides a means to enter a random factor into an RCL program by using the RND function. The command:

```
COMPILE THE BASIC STATEMENT [A=RND(X,Y)] WHICH MEANS [RANDOM INTEGER]
```

will assign a integral value from X to Y inclusive to the variable A. The value of integer X must be less than the value of integer Y, and Y can be any integer from 1 to 32767.

The RND function can be used without a “seed” value, in which case it will return the identical sequence of pseudo-random numbers each time the robot is activated. To generate a somewhat less “pseudo” sequence, use the following command before calling the RND function:

```
RANDOMIZE USING A SEED VALUE OF:  [U]
```

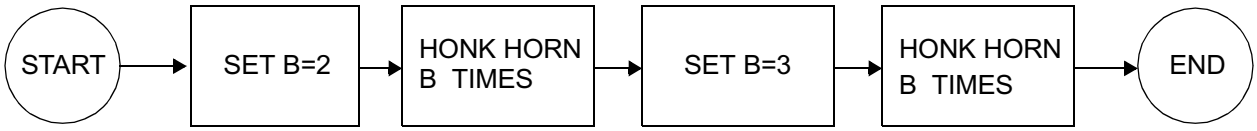
where U is an integer between 0 and 1000 inclusive. By using a variable that takes on different values based on some input (e.g., a counted loop for U that is exited on some external event) the random number generated becomes truly unpredictable.

Using Subroutines in RCL

The COMPILE command also allows the inclusion of subroutines, something that cannot be done in RCL alone. The use of subroutines for procedures that are executed often in the same program can tremendously reduce program size and complexity.

The advantage of a subroutine over an [IF something GOTO somewhere] statement is that the program execution sequence can jump into the subroutine from anywhere in the program and then return to where it jumped from without reference to a specific line number. You can also use a GOTO or JUMP statement to exit a subroutine before reaching the RETURN command.

This concept is best illustrated with a specific example. Consider a program that requires the robot to honk its horn a number of times equal to the variable B. Consider also that this variable could be set to different values at different points in the program. The flowchart below shows a simple program that follows this scheme.



To execute these steps in an RCL program we can use the following code.

HONK: An RCL Program to Demonstrate Subroutine Use

```
1.  PREPARE THE ROBOT
2.  RESERVE THE TINY-BASIC VARIABLE [A]
3.  RESERVE THE TINY-BASIC VARIABLE [B]
4.  SET THE TINY-BASIC VARIABLE [B] EQUAL TO [2]
5.  COMPILE THE BASIC STATEMENT [GOSUB 36] WHICH MEANS [HONK]
6.  SET THE TINY BASIC VARIABLE [B] EQUAL TO [3]
7.  COMPILE THE BASIC STATEMENT [GOSUB 30] WHICH MEANS [HONK]
8.  JUMP TO THE LINE CALLED [DONE]
9.  CALL THIS ROBOT TASK LINE [HONK]
10. BEGIN A COUNTED LOOP CALLED [A] BEGINNING AT [1] ENDING AT [B]
11.    HONK THE HORN FOR [1] SECONDS
12.    WAIT FOR [1] SECONDS
13. REPEAT THE COUNTED LOOP FOR THE COUNTER [A]
14. COMPILE THE BASIC STATEMENT [RETURN] WHICH MEANS [RETURN]
15. CALL THIS ROBOT TASK LINE [DONE]
```

Note how lines 9-13 are executed twice {by the GOSUB), but written only once. In this trivial example only four lines of code are eliminated, but the savings in a more complex program can be substantial. Note also that the subroutine itself is called as line 36 — this is the NSC Tiny Basic line number in the compiled version of the program. By starting the subroutine with a named line (HONK) we make its line number easier to find when scrolling through the compiled program listing. Remember that you must BUILD a program before you can LTP (list) it.

Data Input

The ability to accept input and provide output is crucial for many robotics applications. Input can come from the robot's local environment or from a remote programmer. Sensory information in the form of bumper contacts, sonar echoes, or IR (infrared) readings allow the robot to respond to what it encounters. Output in the form of horn, LED or voice signals can provide feedback to the programmer when acting in a teleoperational mode.

Unfortunately, there is no way to enable I/O (Input/Output) via onscreen prompts or printouts while a program is executing, even with the robot connected to the computer by cable (or the RF communications link option). There are, however, several alternate means for accomplishing I/O.

RB5X can sense its environment in three ways: bumper contacts, sonar and IR. In each of these cases, internal memory registers or NSC Tiny Basic variables take on certain values that can be used for decision-making or calculations within a program.

Bumper input is the easiest data to handle, as it requires nothing more than a few standard RCL commands. The bumpers are numbered clockwise, starting with #1 at the front of the robot, and will display an LED signal (on the correspondingly numbered red LEDs) when pressed. Here is an example of RCL code that uses bumper contact for decision-making in a program:

```
1. BEGIN A LOOP
2. EXIT IF BUMPER NUMBER [1] IS TOUCHED
3. REPEAT THIS LOOP
4. (do something)
```

where the bumper number entered can range from 1-8. Alternatively, line 2 could be replaced by

```
2. EXIT IF ANY BUMPER IS TOUCHED
```

If you need the robot to react to a specific bumper contact for manual data input while a program is executing, individual bumper values can be incorporated into a COMPILE statement. When any bumper is activated, its ID value is stored in a register accessible through NSC Tiny Basic. Here is an example of RCL code that uses bumper contact for numerical data input:

```
1. COMPILE THE BASIC STATEMENT [Y=@#7800:IF Y=251 GOTO 3] WHICH MEANS [BUMPER #1]
2. JUMP TO THE LINE CALLED [NEXT BUMPER]
3. SET THE TINY-BASIC VARIABLE [B] EQUAL TO [1]
```

where the value of the variable Y depends on which bumper is activated (in this case bumper 1), and the variable B is assigned whatever value is needed. Numerical data can be entered into a program by a series of such commands. Individual bumper values are shown in the adjacent chart.

BUMPER	Y=?
1	251
2	254
3	247
4	127
5	239
6	191
7	223
8	253

The bumper value is initially stored at hexadecimal address @#7800 and then transferred to the variable Y (already reserved by NSC Tiny Basic).

Sonar input can be accessed by RCL code modeled after the following example:

```
1. BEGIN A LOOP
2. EXIT IF SONAR DISTANCE IS [MORE] THAN [0.1] FEET
3. REPEAT THIS LOOP
4. CALL THIS ROBOT TASK LINE [MEASURE]
5. COMPILE THE BASIC STATEMENT [LINK#1100:DELAY20] WHICH MEANS [GET D]
6. COMPILE THE BASIC STATEMENT [A=(30*D - 30*73)/22] WHICH MEANS [CALC CM]
```

The loop in lines 1-3 is required to activate the sonar, and only needs to be used once near the start of the program. After that, any time the code in line 5 is used the variable D is assigned a value equal to the number of "counts" between sonar transmission and echo reception. To convert this value into an actual distance we need line 6, the specific details of which depend the units desired ("cm" in this case) and the calibration parameters of the robot (see Calibrating the Sonar).

IR input can be accessed by RCL code modeled after the following example:

```
1. COMPILE THE BASIC STATEMENT [X=#02] WHICH MEANS [TURN ON IR]
2. COMPILE THE BASIC STATEMENT [@#7801=@#7801 OR X] WHICH MEANS [SET BIT]
3. COMPILE THE BASIC STATEMENT [IF (@#7802 AND #40)=0 GOTO 6] WHICH MEANS [LIGHT]
4. SET THE TINY-BASIC VARIABLE [A] EQUAL TO [0]
5. JUMP TO THE LINE CALLED [CONTINUE]
6. SET THE TINY-BASIC VARIABLE [A] EQUAL TO [1]
7. CALL THIS ROBOT TASK LINE [CONTINUE]
```

The conditional "=0" in line 3 will sense light colors against a dark background. If you need to sense dark colors against a light background, change this to "=64." In this example, if the robot senses a light color, the variable A will be assigned a value of "1." If not, the variable A remains at a value of "0."

Direct, real-time, numerical input via keyboard is, as stated earlier, not possible once a program is executing, but by using the hierarchical nature of RCL it can be accomplished, albeit circuitously. The robot must, of course, be connected to the computer by cable or RF link. Consider a program named MOVE that calls, as a task, the sub-program named INPUT consisting of the following lines:

```
1. CALL THIS ROBOT TASK LINE [INPUT]
2. SET THE TINY-BASIC VARIABLE [A] EQUAL TO [42]
3. SET THE TINY-BASIC VARIABLE [B] EQUAL TO [33]
```

where the variables represent, for example, distances to move. When executed, MOVE terminates at a point intended for new data input. On termination, the operator/programmer can easily edit INPUT to include new variable values, and then re-BL (build and load) MOVE. Since INPUT is a task in MOVE, as MOVE is re-compiled these new values will be incorporated as input.

MOVE must be structured to accommodate this type of cyclic data input by including INPUT as a command line preceding any operations on the variables A and B.

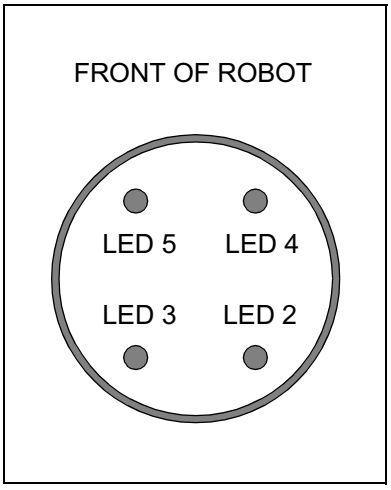
Data Output

As stated in the preceding section, there is no way for RB5X to report data directly to the screen of your computer, even when it is connected via cable or RF com link. Again, there are several alternative means by which the robot can communicated with its programmer. These include the horn, the LEDs, and the voice (if available).

The easiest method here is to use the horn. Conditional loops with included horn commands can be used to count off integral values, or signal any predefined code — even Morse if "text" output is required. Here is an example of a loop that signals the value of a calculated value B:

1. CALL THIS ROBOT TASK LINE [HONK]
2. SET THE TINY-BASIC VARIABLE [B] EQUAL TO [C+E/2]
3. BEGIN A LOOP CALLED [A] BEGINNING AT [1] ENDING AT [B]
4. HONK THE HORN FOR [1] SECONDS
5. WAIT FOR [1] SECONDS
6. REPEAT THE COUNTED LOOP FOR THE COUNTER [A]

Though not as omnidirectional as the horn, the robot's LEDs, if visible in the application, can also be used to signal output codes. There are four individually controllable LEDs numbered as shown in the adjacent diagram. This provides, essentially, four bits of information and allows for much more complex signal codes.



As with the horn, conditional loops are used for converting the values of internal variables into observable output. In practice, line 4 of the preceding example would be replaced by some combination of

```
TURN ON LED NUMBER 2
TURN ON LED NUMBER 3
TURN ON LED NUMBER 4
TURN ON LED NUMBER 5
```

Note that LED 1 is the IR emitting diode located on the underside of the robot. You can turn it on with this command, but it provides no visible output.

The final output option is the voice, available only through the optional voice/sound synthesis package. Using the voice eats up more memory, but provides a method of output most efficiently interpreted by humans. The RCL code for creating words from phonemes, and for speaking those words, is explained clearly in the manual included with that option. Suffice it to say that using a series of SPEAK commands in the form

```
SPEAK THE PHRASE CALLED:  [DIGIT or WORD]
```

with conditional JUMPs provides a powerful means for reporting <u>any</u> type of data.

Creating Your Own RCL Program

You are now ready to begin writing your own RCL programs. This is done by using an onscreen text-based editor. Programs can then be saved to the RCL disk, or downloaded to the robot and executed. If you select "Mode: Free" (as opposed to "Mode: Literal") RCL will assist you by correcting faulty language syntax. This may be helpful to novice programmers. The following example will step you through the programming process. Try this one for practice, and you will soon be sufficiently fluent in RCL to create your own RB5X programs.

You will be creating a simple test program called BEEPLITE. This program will cause RB5X to sound its horn and turn its LED lights on and off. It performs no useful function beyond verifying correct programming procedures. Start out by typing

1. INVENT <RET>

RCL will respond with

2. *INVENT A TASK CALLED* ___

prompting you for the name of the program you want to create. Note that you cannot have two tasks with the same name. If you want to reassign a name, the RENAME function (described later in this example) allows you to do just that. Enter

3. BEEPLITE <RET>

The RCL screen will change, list your program name as "Task BEEPLITE," and prompt:

4. -> ___

You are actually in the edit mode at this point, and can create the program by entering each line in sequence. The flashing cursor is where you begin entering your program. As you enter the lines, RCL will number them sequentially for you. Enter the following line of code:

5. PREPARE THE ROBOT <RET>

This command resets the memory registers to receive a new program. The most current version of RCL (v2.5, 1994) is said to no longer require this line, but programs seem to run more consistently when it is used. The command only costs you 5 bytes, so go ahead and use it.

It is frequently advisable to incorporate a delay between the loading and execution of a program. This allows you time to disconnect the cable (essential if any robot motion is involved), prepare for data acquisition, get out the way, or simply sit back and anticipate, with dramatic pause, your creative masterpiece in action. To include a controlled delay between the loading and execution of this program, enter

6. WAIT FOR ANY BUMPER PRESS <RET>

RCL will respond with

7. *WAIT FOR ANY BUMPER PRESS AND THEN WAIT FOR ___ SECONDS MORE*

allowing you to input the desired delay time. For now, enter

8. 3 <RET>

For programs without any physical motion, the above commands are the only ways to insert such a delay. For programs that do involve motion, RCL provides a calibration parameter that will do this for you automatically without requiring a command within the program. This parameter will be explained in Appendix 11 (Calibrating Your RB5X).

Now we get to the fun part. Time to program some real actions. Enter

9. HONK <RET>

RCL will respond with

10. *HONK THE HORN FOR ___ SECOND(S)*

prompting you here for the number of seconds to honk the horn. Enter

11. 3 <RET>
12. TURN ON ALL THE LIGHTS <RET>
13. WAIT <RET>

RCL will respond with

14. *WAIT FOR ___ SECOND(S)*

prompting you here for the number of seconds to wait. Enter

15. 3 <RET>
16. TURN OFF ALL THE LIGHTS <RET>

This completes the entry of BEEPLITE. You must now save the program to disc, download it to the robot, and run it. Enter

17. Q <RET>

This quits the edit mode, saves your program to the floppy disc, and returns you to the command mode. RCL will respond with

18. *WHAT WOULD YOU LIKE ME TO DO NOW?*

Note that you will not see this program listed on the disc if you list its directory in DOS — it has been assimilated into the file named RCL.SVY which acts as the RCL task library. You could see it in RCL, however, by using the SHOW TASKS command. Try this now. Then enter

19. BUILD AND LOAD (or more simply) BL <RET>

The RCL screen will change again and prompt you with

20. *PLEASE ENTER THE NAME OF THE ROBOT TASK* ___
21. BEEPLITE <RET>

The RCL screen will change several times as BEEPLITE is compiled into NSC Tiny Basic and download to the robot. When the download is complete the program would normally be executed immediately — in this case you'll need to press a bumper to begin the show. RB5X should wait three seconds, beep its horn for three seconds, and then turn on its LED lights for three seconds. If the cable is still connected, you can easily execute the program again. Just type RUN and the program will repeat.

Editing an RCL Program

If you want to experiment with RCL commands on your own, BEEPLITE can be modified by using the edit mode. Enter

1. LIST <RET>

RCL will respond with

2. *LIST THE* ___
3. BEEPLITE <RET>

RCL will list the program and also display its size (38 bytes). This is a small program. RCL will accommodate programs up to 1000 bytes, so fairly complex problem-solving is indeed possible.

To edit this program enter

4. EDIT <RET>

RCL will prompt with

5. *EDIT THE* ___
6. BEEPLITE <RET>

RCL will list your program and place the cursor after the program's final line. Look it over and review the lines you entered earlier. If you need to add, delete or modify any program lines, this is the place to do it. You are now in the edit mode.

While in the edit mode you can use the following tools:

CURSOR KEYS: to move cursor up and down through the program lines
TAB: to move cursor to the next input field in the current line
SHIFT + TAB: to move cursor to the previous input field in the current line
NUMBER: moves cursor to program line with that number

When the cursor is positioned in the right place, simply enter the desired program line and RCL will renumber all the other lines accordingly. You also have these options:

E <RET>: to edit the line above cursor
D <RET>: to delete the line above cursor
L <RET>: to list the program showing changes
Q <RET>: to quit the edit mode and save the program

There is no way to directly save an edited program under a new name. If you want to modify an existing program and rename it, you must first create a copy of the original with the desired new name before editing the original. Follow these steps from the main command prompt:

1. *WHAT WOULD YOU LIKE ME TO DO NOW?*
2. CLONE A TASK <RET>
3. *MAKE A CLONE FROM THE TASK NAMED .. ___*
4. [NAME OF TASK YOU WANT TO EDIT] <RET>
5. *GIVE THE NEW TASK A NAME .. ___*
6. [NAME YOU WANT FOR NEW TASK] <RET>

RCL will process your request, create a copy the original program, rename it, and save it to disc. You can then edit the cloned program as desired using the EDIT command explained earlier.

To simply rename as existing program, follow these steps from the main command prompt:

1. *WHAT WOULD YOU LIKE ME TO DO NOW?*
2. RENAME <RET>
3. *RENAME OLD NAME ___*
4. [OLD PROGRAM NAME] <RET>
5. *RENAME OLD NAME [OLD PROGRAM NAME] TO NEW NAME ___*
6. [NEW PROGRAM NAME] <RET>

Program names can be up to 256 characters long including spaces. For convenience, keep them short and DOS-like or your SHOW TASKS screen can become very confusing. Remember that you cannot have two programs with the same name on the same RCL disc.

When you're done with RCL and want to return to DOS, enter

BYE <RET>

Program Size and Memory Limits

BEEPLITE was a small program (38 bytes) that barely scratched the surface of RB5X's potential. Individual programs up to 1000 bytes (≈72 lines) in size can be created, saved and run as needed. Even larger programs can be written by exploiting the hierarchical command structure of RCL.

Perhaps you recall reading that RB5X has 8000 bytes (8k) of standard RAM, and were wondering what the other 7k is doing. The short answers is: waiting for more programs. The 1k size limit on programs is built into RCL itself, and has nothing to do with physical memory space. If you need more than 1k, RCL provides a solution.

We explained earlier how the term "task" can mean either "program" or "command." This is the essence of RCL's hierarchical structure. Programs you have already created and saved to disk can be used as commands within other programs.

Using this capability of RCL is simple. All you need to do is include, as an RCL program line, the name of a previously created program. When that line executes, the program it names will run like a subroutine. When complete, it will return control to the following program line. Of course, the "called" program must be present on the RCL disk currently in use. In addition, it must not use any line names (as defined by the CALL THIS ROBOT TASK LINE ... command) already in use by any of the programs preceding it.

In other programming languages this is known as "linking" or "chaining." It provides a way to utilize most (but not all) of RB5X's 8000 byte memory — there's a few hundred bytes taken up by internal tasks relating to the robot's self-diagnostic startup routine.

Here's an example of how to do this. Suppose you have already created a program called MATH that performs several calculations needed for determining whether RB5X will move. Suppose also that during the execution of MATH the variable A is assigned some value.

```
1. PREPARE THE ROBOT
2. MATH
3. BEGIN A LOOP
4. COMPILE THE BASIC STATEMENT [IF A>10 GOTO 37] WHICH MEANS [CHECK A]
5. REPEAT THIS LOOP
6. MOVE DISTANCE FORWARD FOR THIS MANY FEET [3]
```

Note that the variable A needs to be reserved in the program MATH, but not in the program that calls MATH as a subtask. Note also that the statement IF A>10 GOTO 37 refers not to an RCL program line, but to the NSC Tiny Basic program line corresponding to RCL line 6. Enter this RCL program, BUILD it, and then list it in NSC Tiny Basic (using LTP) to see the line numbers.

To paraphrase the ever-prescient Mr. William Gates, 8k should be enough memory for anybody. But for extremely complex programs you may still run into the 8000 byte memory limit. If this starts to occur too often in your own robot applications, consider purchasing the 16k RAM option offered by General Robotics. 24k should be enough memory for anybody.

Appendix 11: Calibrating Your RB5X

About Calibration

There are a few initial calibrations that must be done before you can program RB5X to perform specific tasks. "Calibration" is the process of setting values or states for internal processes that are normally invisible to the user during operation. It controls how RB5X, RCL and NSC Tiny Basic respond to input and generate output.

Calibration settings are saved to the RCL floppy disc and are reapplied every time you BUILD AND LOAD a program. They are, however, specific to whatever serial number RB5X is currently being used. If you are fortunate enough to have more than one RB5X, and require different calibration settings for each, you should have and use separate "RCL for DOS" discs for each robot.

See your Reference Manual for a complete listing of calibration parameters. For most purposes, only a few are important. If you have students advanced in technology skills, you might want to assign them the responsibility of calibration.

To begin calibration, connect RB5X to your computer and go to the RCL command mode. If you don't understand what this means, return to the section titled Connecting RB5X to Your Computer and read it start to finish. Then come back here, empowered by your new-found fluency in RCL.

Calibrating the Com Port

If your computer has a serial (15 pin) mouse it is probably connected to COM1. Even though the mouse is not used while working in RCL, this can cause conflicts. RB5X must be connected to a free serial communications port on your computer. Since the default port for RCL is COM1, you will probably need to change it to COM2 and connect the robot to that port. Enter:

1. CALIBRATE <RET>

RCL will respond with

2. *CHANGE THE CALIBRATION PARAMETER NAMED __*
3. COMMPORT <RET>

RCL will respond by telling you which port it is currently using (1,2,3 or 4) and prompt:

4. *SET THE VALUE TO __*
5. 2 <RET> (or whatever port you need to use)

RCL will verify your setting and return to the command mode.

If your computer has no available serial communication ports you will need to make one available and set it for 1200 baud. Consult your operating system's documentation for instructions.

Calibrating the Bumper-Start Delay

This calibration setting will save you some programming steps later. It requires a program loaded into the robot to be activated by a bumper press, giving you time to disconnect the cable before motion begins. It also creates a three second delay (default value) between the bumper press and program execution. With this parameter set to Y (yes), an initial RCL program line of WAIT FOR ANY BUMPER PRESS is no longer required in programs that involve robot motion.

(In programs that do not involve robot motion this parameter has no effect. To create a delay in these cases, use the WAIT FOR ANY BUMPER PRESS command at the start of the program.)

To enable this "bumper start" feature for programs involving robot motion, enter

1. CALIBRATE <RET>

RCL will respond with

2. *CHANGE THE CALIBRATION PARAMETER NAMED* __
3. BUMPER.START <RET>

RCL will respond with

4. *BUMPER.START IS N* or *BUMPER.START IS Y*
5. Y <RET>

If you want to change the three second delay between bumper press and program execution, enter

6. CALIBRATE <RET>
7. *CHANGE THE CALIBRATION PARAMETER NAMED* __
8. DELAY.START <RET>
9. *DELAY.START IS 3*
   SET THE VALUE TO __

at which point you can specify the exact delay time.

There is another calibration parameter called BUMPER NEEDED, but I don't recommend using it as its value Y or N is not saved. All it seems to do is generate an onscreen reminder about disconnecting the cable. It does not create the bumper press requirement or program execution delay that BUMPER.START so conveniently provides.

Setting Other Calibration Parameters

Most of the other parameters have acceptable default settings. But if you need RB5X to execute very accurate motions, some of these parameters may need to be tweaked slightly. One motion that sometimes needs adjustment is rotation. This can be done by changing the parameters:

```
CALIBRATE SPIN.DEG/SEC = [62]
CALIBRATE SPIN.SECS/90DEG = [1.38]
CALIBRATE SPIN.SECS/TURN = [4.95]
```

For the first parameter, decrease the setting if the robot doesn't turn far enough, and increase it if the robot turns too far. Reverse this logic for the second two parameters. Note that each of these parameters affects specific rotation commands. The first affects commands that involve specified rotation angles, such as SPIN CLOCKWISE FOR [45] DEGREES. The second affects commands that invoke right-angle turns, such as SPIN RIGHT 90 DEGREES. The third affects only the SPIN AROUND CLOCKWISE (or COUNTERCLOCKWISE) [VALUE] TIMES command.

For some robot applications you may want to use metric values in commands involving distances, e.g., the command

```
MOVE DISTANCE FORWARD FOR THIS MANY FEET [VALUE]
```

This can be done by setting the following parameter:

```
CALIBRATE MOVE.SECS/FT = [9.84]
```

Note: If you make this change, movement commands will still prompt you for distances in FEET (as shown above). This text is built into RCL. Ignore it, and treat all distance prompts as if they actually read METERS.

If your intended use of RB5X does not include calculations, or if you just feel more comfortable working with imperial units, leave this parameter at its default setting of 3. If your intended use requires very accurate movements, further fine-tuning of this parameter may be necessary.

For sonar applications, the loop exit command

```
EXIT IF SONAR DISTANCE IS [LESS or MORE] THAN [VALUE] FEET
```

also expects the VALUE to be in units of FEET. To use a VALUE in units of METERS instead, change the following parameter from its default setting of 22 to:

```
CALIBRATE SONAR.COUNTS/FT=[72.6]
```

Again, after making this change, the command will still prompt for distances in FEET. If your intended use requires very accurate movements, further fine-tuning may be required. See the following section for detailed instructions on sonar calibration.

Calibrating the Sonar

If your robot applications require very accurate sonar range measurements, fine tuning of the sonar offset may be necessary.

```
CALIBRATE SONAROFFSET = [73]
```

Note that this command does not require a period after the word SONAR.

The default setting of 73 is the "usual" minimum value of the variable D returned by a series of sonar range measurements. However, we have found this value can vary from one robot to the next. To check your robot, you need a program that reports D as a series of horn beeps or other easily counted signals. Enter the following program:

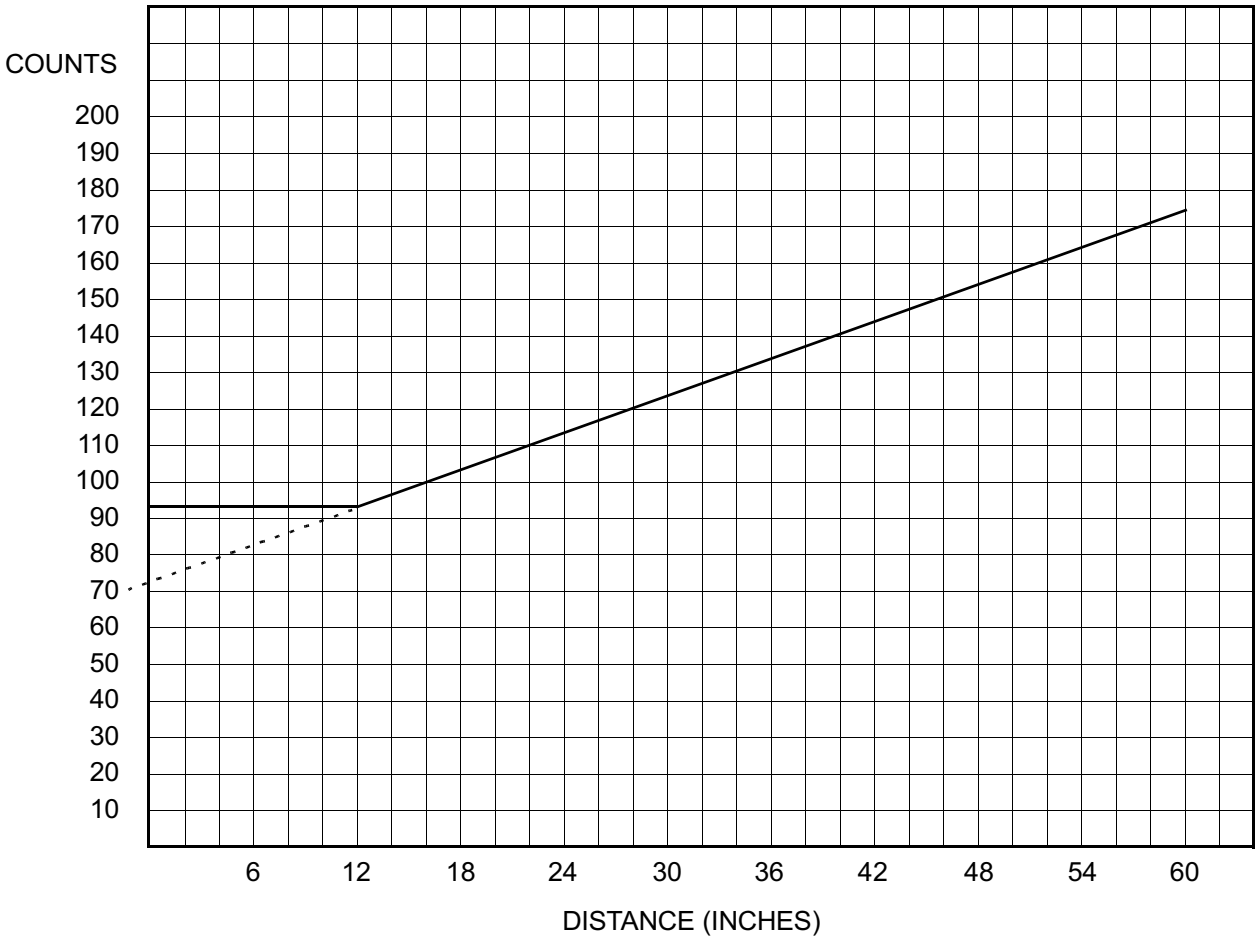
D-TEST: An RCL Program to Measure Sonar Offset

```
1.  PREPARE THE ROBOT
2.  RESERVE THE TINY-BASIC VARIABLE [A]
3.  BEGIN A LOOP
4.  EXIT IF SONAR DISTANCE IS [MORE] THAN [0.1] FEET
5.  REPEAT THIS LOOP
6.  CALL THIS ROBOT TASK LINE [START]
7.  COMPILE THE BASIC STATEMENT [LINK#1100:DELAY20] WHICH MEANS [GET D]
8.  SET THE TINY-BASIC VARIABLE [A] EQUAL TO [0]
9.  BEGIN A LOOP
10. HONK THE HORN FOR [0.1] SECONDS
11. WAIT FOR [0.2] SECONDS
12. COMPILE THE BASIC STATEMENT [A=A+1] WHICH MEANS [INCREMENT A]
13. COMPILE THE BASIC STATEMENT [IF A=D GOTO 47] WHICH MEANS [EXIT LOOP]
14. REPEAT THIS LOOP
15. WAIT FOR ANY BUMPER PRESS AND THEN WAIT [1] SECOND MORE
16. JUMP TO THE LINE CALLED [START]
```

Note: The initial loop in lines 3-5 are needed to simply activate the sonar; it isn't the most elegant method, but none of the LINK codes work by themselves. For convenience, you can write a program called SONARON consisting of these three lines. In future programs, the sonar can then be activated by the simple one-word command SONARON.

Set the robot in an open area where the test can be run. Place a flat, smooth vertical barrier to reflect the sonar 60 inches from the transmitter. Load D-TEST and get ready to count beeps. Record data from 60 inches to 12 inches in 6 inch increments. Run each distance a few times to make sure you get a representative value — spurious readings sometimes occur.

Next, graph your data with "counts" on the vertical axis and "distance" on the horizontal axis. Although details will vary from one robot to the next, you should end up with a plot that looks something like the following.



Finally, calculate the slope and Y-intercept of your plot.

Note that the data bottoms out near 90 counts, due to limitations in the sonar response time. The Y-intercept is not this bottomed-out value, but the one extrapolated from the plot itself (either mathematically or graphically). In this case the correct Y-intercept is about 72.

Likewise, when you calculate the slope, ignore this bottomed-out data. Express the slope in units of "counts/foot" (not counts/inch) to three significant figures.

You are now ready to recalibrate the sonar. Set the following parameters:

```
CALIBRATE SONAR.COUNTS/FT=[SLOPE]
CALIBRATE SONAROFFSET = [Y-INTERCEPT]
```

RB5X's sonar can now be used to measure distances with errors of 1-5%. Residual errors are due to variations in the air through which the sonar must propagate.