

Gripper & Experimenter's Module Manual

Copyright 1997, *ActivMedia*, Inc. All rights reserved.

Under international copyright laws, this manual or any portion of it may not be copied or in any way duplicated without the expressed written consent of *ActivMedia*, Inc.

The Saphira libraries and Pioneer software on disk and on the Pioneer server EPROM that accompany the robot and accessories and which are available for network download by Pioneer 1 customers are solely owned and copyrighted by Kurt Konoldige and SRI International. Pioneer 1 developers and users are authorized by revocable license to develop and operate custom software for personal, research, and educational use *only*. Duplication, distribution, reverse-engineering, or commercial application of the Pioneer software and hardware without the expressed written consent of SRI International and *ActivMedia*, Inc. is explicitly forbidden.

REAL WORLD INTERFACE Robots for the Real World

Real World Interface (RWI), Inc. manufactures the Pioneer 1 Mobile Robot and Gripper/Experimenter's Module accessories. The various names and logos for products used in this manual are registered trademarks or trademarks of their respective companies. Mention of any third-party hardware or software constitutes neither an endorsement nor a recommendation.

Contents	Page
1. INTRODUCTION	1
1.1 Gripper and Experimenter's Package	1
1.1.1 Package Components	1
1.1.2 User Supplied Components	1
1.2 Basic Pioneer Platform	2
1.3 The Gripper & Experimenter's Module	2
1.3.1 Gripper Description	2
1.3.2 Experimenter's Module Description	4
1.4 Additional Resources	5
1.4.1 Pioneer Web Software Archive	5
1.4.2 Pioneer and Saphira Newsgroups	5
1.4.3 Support	6
2. INSTALLATION	7
2.1.1 Step 1: Remove the Console	7
2.1.2 Step 2: Remove Microcontroller from Console	7
2.1.3 Step 3: Replace the PSOS EPROM	8
2.1.4 Step 4 Reattach Microcontroller to Console	9
2.1.5 Step 5 Remove Nose	9
2.1.6 Step 6 Connect Cables	9
2.1.7 Step 7 Attach Gripper/Experimenter's Module to Pioneer	9
2.1.8 Step 8 Re-Attach the Console Cables	10
2.1.9 Step 9 Re-Attach the Console	10
2.1.10 Step 10 Test the Assembly	10
3. GRIPPER OPERATION AND PROGRAMMING	11
3.1 Gripper Self Test	11
3.2 PSOS Server Information Packet	11
3.3 Gripper I/O	12
3.4 Gripper Programming	12
3.4.1 Direct Control via Digital Ports	13
3.4.2 State Commands	13

4. EXPERIMENTER'S MODULE	15
4.1 I/O Ports	15
4.1.1 Digin: Self-Test	17
4.1.2 Digout: Self-Test	17
4.2 Programming the Digital I/O Ports	18
4.3 Programming the A/D Ports	18
4.4 Programming the Speaker	18
5. MAINTENANCE & REPAIR	19
5.1 Drive Lubrication	19
5.2 Gripper Belt Adjustments	19
5.3 Factory Repairs	19
6. APPENDIX A	20
Example C Program Demonstrates Gripper and Speaker Functions	
7. APPENDIX B	24
Pioneer Server Information Packet	
8. INDEX	25
WARRANTY & LIABILITIES	28

1. Introduction

Congratulations on your purchase and welcome to the rapidly growing community of researchers, developers, and enthusiasts of the Pioneer 1 Mobile Robot.

This *Pioneer 1 Gripper & Experimenter's Module Manual* provides the general and technical details you will need to install and operate your new Gripper and to develop your own attachments and enhancements for the Pioneer 1 Mobile Robot with the integrated Experimenter's Module.



We also encourage you to use these companion resources that come with your Pioneer:

- ✓ Pioneer Operation Manual
- ✓ Saphira Software Manual
- ✓ Pioneer Registration & Account Sheet
- ✓ Personal account for the <http://css.activmedia.com> Internet server
- ✓ Pioneer- and Saphira-users@rwii.com support newsgroups

2.1 Gripper and Experimenter's Package

Our experienced robotics manufacturing staff put your Pioneer 1 Gripper and Experimenter's Module through a "burn-in" period and carefully tested them before we shipped the hardware and software to you. Our care extends beyond: Besides the companion resources listed above that bring the whole community of Pioneer to you, we warranty the Gripper and Experimenter's Module against mechanical and electronic parts and labor defects for 90 days. All of these precautions ensure that you have many years to enjoy your new accessories for the Pioneer 1 Mobile Robot.

Even though we've made every effort to make your package complete, please check the components once again after you unpack it from the shipping crate.

2.1.1 Package Components

- ✓ Pioneer 1 Gripper & Experimenter's Module Assembly
- ✓ 32K EPROM containing latest PSOS
- ✓ Gripper & Experimenter's Module Manual

2.1.2 User Supplied Components

- ✓ Pioneer 1 Mobile Robot
- ✓ Set of hex wrenches that accompanied original robot
- ✓ Small, flat-bladed screwdriver
- ✓ Optional "chip-puller" tool

2.2 Basic Pioneer Platform

Pioneer 1 is a small, mobile robot developed by Kurt Konolige of SRI International and Grinnell More of Real World Interface, Inc., and is available exclusively through *ActivMedia, Inc.*

The basic Pioneer 1 Mobile platform contains all of the components for sensing and navigation in a real-world environment, including battery power, drive motors and wheels, position encoders, and ultrasonic sonar transducers—all managed via an MC68HC11-based microcontroller board (Figure 1-1).

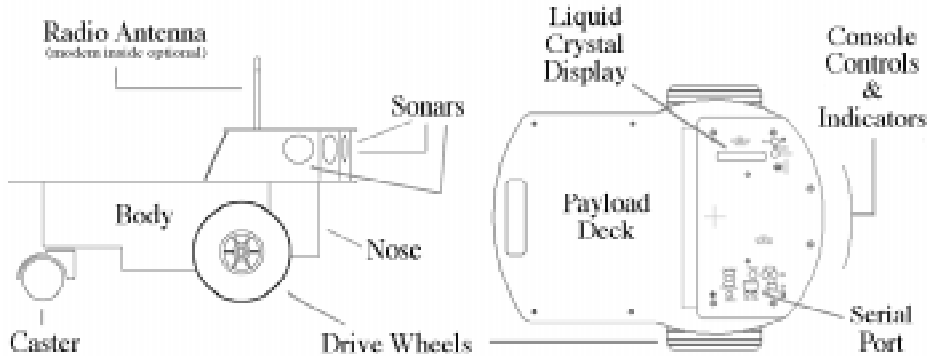


Figure 1-1. Basic components of the Pioneer 1.

The Pioneer 1 also comes with the Pioneer Server Operating System (PSOS) software on EPROM for easy access and control of the robot's systems, an RS232 serial port for communication between the robot and other computers, as well as a variety of expansion I/O ports for optional and custom attachments.

2.3 The Gripper & Experimenter's Module

The Pioneer 1 Experimenter's Module essentially is a circuit board and accessory electronics that extend the capabilities of the Pioneer 1 microcontroller's standard I/O ports, including electrical support for the Pioneer Gripper. The Experimenter's Module is part of the Pioneer Gripper assembly, which together replace the nose of the basic robot (see Figures 1-1 and 1-2 and Chapter 2, *Installation*).

2.3.1 Gripper Description

The Pioneer Gripper is a simple, yet powerful 2-DOF robotic manipulator and sensor (Figure 1-2 and Figure 1-3). Driven by a single, reversible-DC servo-motor under control of the Pioneer microcontroller, it operates between two states: down/open and up/closed.

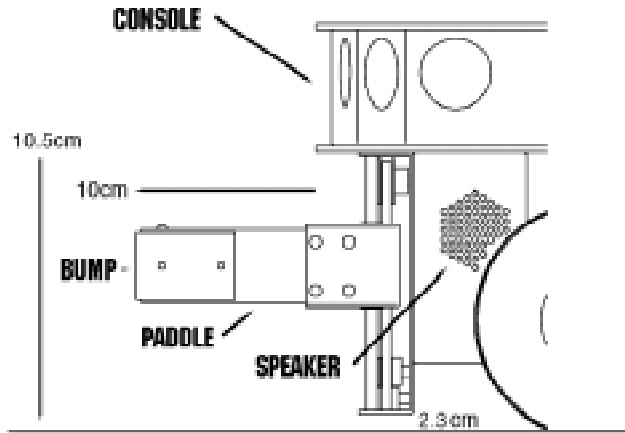


Figure 1-2 Pioneer Gripper side view.

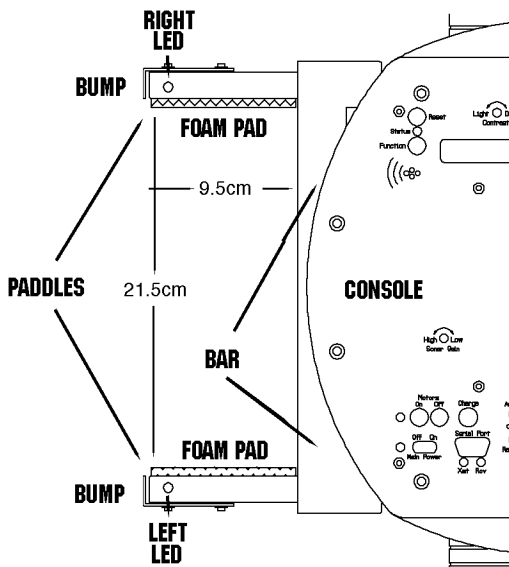


Figure 1-3. Pioneer Gripper top view

In its fully down/open state, the two 2.5 inch (5cm) tall by 3.5 inch (9.5cm) deep Gripper paddles (each with 0.5cm foam pads on the inside) are 8.5 inches (21.5cm) apart and ride 1 inch (2.3cm) off the floor. The paddles extend out from the front robot's main body 3 inches (10cm) beyond the console edge.

Overview

In transition from the fully down/open position to their up/closed state, the Gripper paddles close together horizontally until they pinch an object or close on themselves. At that point, the entire Bar and Gripper Assembly rises up vertically—with or without an object in its grasp. Under software control, the Gripper then rises to a special “carry” position or all the way to the top of the Gripper case 4 inches (10.5 cm) off the floor to its closed/up position. It is not normally possible to open the Gripper at this state.

The reverse cycle lowers the Gripper Bar and paddles to the bottom of the assembly, whereupon the paddles separate, dropping any contents, and open to their fully down/open position.

Two “bump” switches—one on each paddle tip—sense contact with objects. IR-based “break beams” embedded at the front and back of each paddle sense objects between and within the Gripper’s grasp.

Three other switches sense the Gripper’s paddle positions. One switch inside the Bar is on when the Gripper paddles are in their fully open position. Another at the top of the Bar toggles on when the Gripper reaches its fully up position. A third “carry” switch, located behind the Bar, indicates when the Gripper is between 1.5 and 2 inches (4-5cm) off the floor.

2.3.2 Experimenter’s Module Description

The Experimenter’s Module includes several additional features that users may enable through the Pioneer I/O controls: There are green “directional” LEDs embedded in the tip of each paddle and at the top of each side, facing front on the Gripper’s back plate. On the left side of the Gripper, there is a speaker, an SPST on-off slide switch, a momentary push-button switch, and another LED (amber).

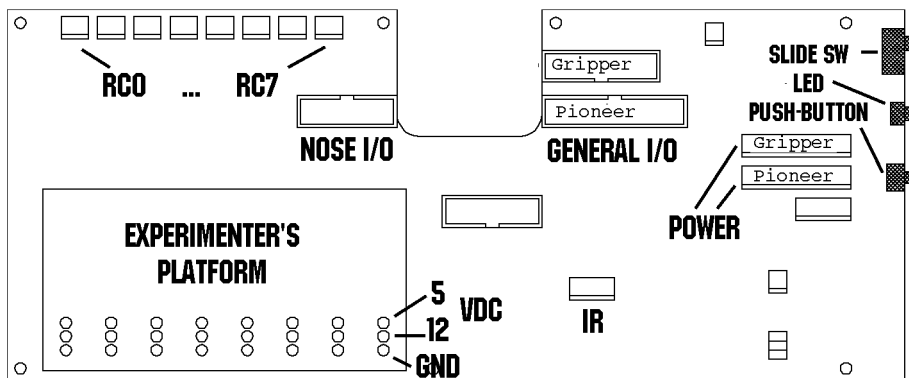


Figure 1-4. Pioneer 1 Experimenter’s Module

Inside and on the Experimenter’s Module is a bank of eight RC-servo controller ports. Layout/solder pads with DC power and I/O port connections supplied by the Pioneer microcontroller also are on the Module card, as well as eight multiplexed analog-to-

digital ports. There also is circuitry and a connector for an as-yet-implemented remote-controller IR.

A complete list of available I/O ports and connections can be found in Chapter 4, *Experiment's Module*.

2.4 Additional Resources

Every Pioneer 1 customer gets three additional and valuable resources: a private account on *ActivMedia's* Internet server for downloading Pioneer software, updates, and manuals, access to RWI's private robotics newsgroups, and E-mail access to the Pioneer 1 support team.

2.4.1 Pioneer Web Software Archive

We have a World Wide Web server connected continuously to the Internet where customers may obtain Pioneer 1 software and support materials. Point your favorite Web browser to:

<http://css.activmedia.com>

Some access areas are restricted to licensed customers, including Pioneer owners. To gain restricted access, use the username and password that are written on the Pioneer 1 *Registration & Account Sheet* that accompanied your robot.

2.4.2 Pioneer and Saphira Newsgroups

RWI also maintains a special E-mail-based newsgroup for Pioneer 1 owners and Saphira programmers to share ideas, software, and questions. We strongly encourage you to sign up. For details, send E-mail:

```
To: majordomo@rwii.com
From: <your return E-mail address goes here>
Subject: help (Subject: always ignored)

(body of message—choose one or more commands:)
help      (returns instructions)
lists    (returns list of newsgroups)
subscribe <list name here>      (waddayatink?)
unsubscribe <list name here>    (ditto)
end
```

Overview

2.4.3 Support

Have a problem? Can't find the answer in this or any of the accompanying manuals? Know a way that we might improve Pioneer 1? Share your thoughts and questions directly with us:

`pioneer-support@wii.com`

Your message goes to our team of Pioneer developers who will help you directly or point you to where you may find help. Because this is a support option, not a general-interest newsgroup like Pioneer-users described above, we must reserve the option to reply only to questions about bugs or problems with Pioneer 1 and its accessories.

3. Installation

Please read through this chapter carefully before you attempt to attach the Gripper & Experimenter's Module to your Pioneer 1 Mobile Robot. If for any reason you do not wish to perform the work yourself, contact RWI and make arrangements to have the assembly installed at the factory.

On the other hand, skip this chapter altogether if your Pioneer 1 Mobile Robot came with the Gripper & Experimenter's Module Assembly already attached.

Turn OFF the Pioneer robot's Main Power *before* installing the Gripper & Experimenter's Module.

3.1.1 Step 1: Remove the Console

The Gripper & Experimenter's Module Assembly's various ports and accessories connect to the Pioneer microcontroller's power and expansion I/O. Also, software support is provided through a special Pioneer Server Operating System (PSOS), which is stored on a 32K EPROM mounted on the microcontroller. Accordingly, to install the Gripper/Experimenter's Module and new PSOS, you must first disassemble the Pioneer Console.

With one of the hex wrenches that came with your robot, remove the six outermost screws that secure the top plate to the Console—two on the front (including camera, if attached) and two on each side.

If one is attached, carefully unplug and remove the Pioneer camera from the Console and back panel. Also unscrew and remove the radio modem antenna, if attached.

Twist *counter-clockwise* to remove, and twist *clockwise* to insert the various screws.

Gently lift the Console top plate to expose the microcontroller board, which is mounted to its underside, and the various wire harnesses attached to it. Being careful to not handle the attached microcontroller card, remove the wire connectors on both sides of the card from their sockets, freeing the Console top plate with the microcontroller card attached.

3.1.2 Step 2: Remove Microcontroller from Console

Lay the Console top plate face up, with the Pioneer 1 microcontroller card down, onto a clean, soft surface (first section of the Sunday Times is a great pad). Locate and remove the six hex screws—one in each side and two in the middle—which secure the microcontroller board to the Console top plate. Separate the parts. The microcontroller board should be exposed now, face-up on the table (Figure 2-1).

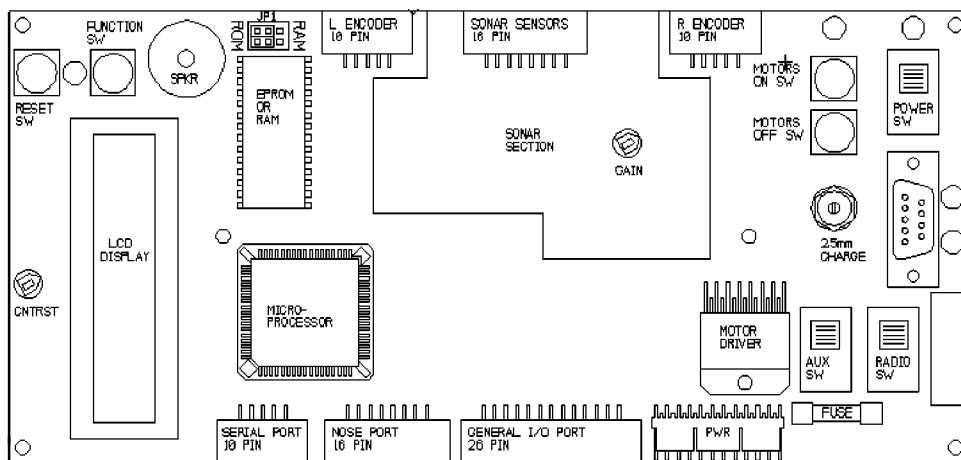


Figure 2-1. Pioneer 1 microcontroller card

3.1.3 Step 3: Replace the PSOS EPROM

Locate the EPROM/SRAM socket (Figure 2-2) on the microcontroller card and note that the Pin 1 locator notch on its case is towards the outer edge of the board—you'll want to similarly position the new PSOS EPROM.

Pry or pull out the socketed EPROM chip. We'd prefer that you use a special "chip-puller" tool, but if one isn't handy, use a thin, flat-bladed screwdriver and gently pry up each end of the chip until it lifts out of the socket freely.

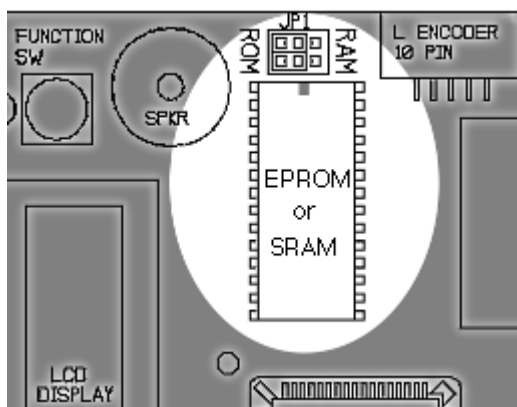


Figure 2-2. Location of the EPROM socket

EPROM chips are sensitive to static electricity. Be careful to handle them by their case and store them on conductive foam. Don't touch the pins with your fingers, and be particularly careful not to bend any of their 28 delicate connector pins.

Now carefully remove the new PSOS EPROM that came with the Gripper/Experimenter's Module from its protective foam and insert it into the EPROM socket. Be sure to line up the Pin 1 notch and insert the pins in their respective socket holes. Then

press carefully, but firmly down on the chip with your thumb or forefinger to seat it tightly in the socket.

3.1.4 Step 4 Reattach Microcontroller to Console

By reversing the procedures you took in Step 2 and using the same tools and screws, reattach the Pioneer microcontroller to the Console top plate. Set the entire assembly aside for the moment in a safe place.

3.1.5 Step 5 Remove Nose

The Gripper/Experimenter's Module replaces the Pioneer's nose. Use the hex wrenches that came with your robot to remove the six screws that secure the Nose to the Pioneer Deck and Body: four on top of the Deck and one on each side at the bottom. Store the Nose away.

3.1.6 Step 6 Connect Cables

Place the Gripper/Experimenter's assembly on top of a table close to and in front of the Pioneer, with the Module card and attached flat ribbon cables facing the robot's Nose. Push those three cables' free connector ends through the opening in the nose and up into the Console. Similarly, push the Pioneer's power cable through and out the Nose.

Plug the power cable attached to the Pioneer robot into its new 12-pin socket on the Experimenter's Module just below the accessory's power cable and socket. The cable's yellow wires should point towards the center of the Module.

CAUTION: HOT POWER CABLE!

The Pioneer power cable is connected, unswitched, directly to the battery. Be careful to align the connector with its socket. Misalignment may cause damage to the Pioneer microcontroller or to the Experimenter's Module.

You may find that first unplugging the new power cable connector from its socket on the Module will expose the lower socket for easier attachment of the robot's power connector. Do re-insert the new power cable from the Module back into its socket before proceeding with the installation. Do not switch the connectors, plugging the robot's power cable into the upper socket—otherwise, no damage will occur, but the Gripper will activate without provocation.

3.1.7 Step 7 Attach Gripper/Experimenter's Module to Pioneer

Use the hex wrenches that came with your robot and the screws you removed from the robot's Nose to attach the Gripper/Experimenter's Module to the Deck (4 screws; no screws attach to the Pioneer Body, as they do for the Nose).

Installation

3.1.8 Step 8 Re-Attach the Console Cables

Carefully handling the Console top plate with attached microcontroller by its edges, reattach the various cables, front and back. Although the order is not critical, for convenience we recommend attaching the front cables first—the sonar and drive (left and right) cables.

Then, to the sockets on the rear of the microcontroller, attach the 4-pin serial connector, the new 16-pin Nose I/O and 26-pin General I/O connectors, and finally the new power connector to their respective sockets.

3.1.9 Step 9 Re-Attach the Console

Align the Console top-plate with microcontroller on top the Console, and using the tools and the screws from Step 2, reattach it to the robot body. Reattach the camera to the front of the Console as well, if it is with your robot. Then plug the camera power and signal cables back into their respective sockets on the Console back plate, and screw in the antenna through the hole in the Console top plate and into the radio modem beneath.

3.1.10 Step 10 Test the Assembly

Time to switch on the Pioneer's Main Power and sniff for blue smoke. If sparks don't fly and smoke is absent, the Pioneer will automatically exercise it's new Gripper and finish with three loud and firm beeps through the newly attached speaker—not that weak, sniveling old “beep, beep” of the Gripper-less Pioneer, mind you, so keep sensitive babies and weaklings clear.

You've completed assembly and now are ready to run your Pioneer's brand new gadgets. And for your troubles, you get two loose screws left over (they are the two round headed ones that came out of the bottom of the Nose).

4. Gripper Operation and Programming

The Pioneer Gripper comes fully integrated with the robot's systems and software. The latest versions of PSOS (4.2+) contains support for direct control of the Pioneer's onboard I/O that run the Gripper functions, as well as state-based software control routines that manage the Gripper functions for you. In turn, these PSOS functions are supported in Saphira (version 5.3+), the Pioneer Application Interface (PAI), and Pioneer-LOGO. Consult the various Pioneer-related Operation and Software Manuals for more details about PSOS, Saphira, PAI, and P-LOGO.

4.1 Gripper Self Test

Gripper-enabled versions of PSOS come with test routines for all the main functions of the robot. This section describes the Gripper's self-test. Consult *the Pioneer 1 Operations and Software Manual* for a detailed description of the location and operation of the various Console function buttons and switches and for how to test the drive and I/O systems, sonars, and processor.

Make sure the Pioneer motors are *disengaged*, if you perform the Gripper self-test on a table top.

The Gripper self-test happens whenever you turn the Main Power on or press the Reset button. The exercise first drives the Gripper Bar and paddles to their fully down and open state, and then back up to their fully up and closed state.

Always place Pioneer 1 on the floor and have everyone step back *before* engaging the motors.

You may also test the Gripper's open, up, and carry state switches. See the Experimenter's Module, Digin I/O test mode section in Chapter 4 for details.

4.2 PSOS Server Information Packet

PSOS regularly sends Pioneer server information to a connected client over the serial communication line. Included in that Server Information Packet are the various states and readings from the Gripper and the Experimenter's Module I/O. A summary of the PSOS Server Information Packet is in the Appendix B. These various readings may be captured and individually read by the client, or you may use the various Saphira, PAI, or P-LOGO functions to manage the packet and read the values for you.

4.3 Gripper I/O

The Gripper's single drive motor is controlled through two digital output lines and under control of the microcontroller CPU: Output port OD0 controls the direction of rotation and OD1 enables/disables the motor.

The act of gripping and raising objects is mechanical and dependent on the Gripper's position—you cannot independently operate the gripper paddles. Rather, you drive the motor in one direction or the other. The paddles automatically open when the Gripper is being lowered and reaches the bottom of the assembly. In reverse, the paddles automatically close upon themselves or an object and then rise.

Switches on top (“up”), inside at one end of the Bar (“open”), and behind the Bar (“carry”) indicate the various Gripper (G) positions (Table 3-1). In addition, bump switches on the tips of each paddle indicate collisions, and break beams (b-b) inside the paddles indicate the position of an object within the grasp. We identify the various I/O ports and connections by name in Table 3-1. Consult the Tables in Chapter 5 for their pin locations on the General and Nose I/O connectors.

Table 3-1. Gripper motor controls and state switches

Name	I/O port	Meanings
G.motor	OD0	rotate 1=down; 0=up
G.motor	OD1	1=enable
top	ID0	0= G at top
carry	ID1	0=paddles closed, G 4-5cm above floor
open	ID2	0=paddles fully open
inner b-b	ID3	1=object inside
outer b-b	ID4	1=object inside
bumps	ID5	0=paddle tip touching object

4.4 Gripper Programming

You can directly control the Gripper motor through the PSOS `sfCOMDIGOUT` command and byte parameters. PSOS also will accept a new state command, `sfCOMGRIPPER`, to achieve a gripper state, and sets the gripper motor bits appropriately. There also are some Saphira convenience functions defined for getting and setting the gripper state. For details on programming PSOS directly and through Saphira, consult the Saphira Software and Pioneer Operations and Software Manuals.

sfCOMDIGOUT control of Pioneer digital output ports has been changed in PSOS version 4.2 and later.

4.4.1 Direct Control via Digital Ports

The revised PSOS 4.2 (and later) `sfCOMDIGOUT` command has a two-byte argument: The high byte is a mask of those output bits you want to change, and the low byte is the bit pattern for the bits to change. This way, you don't need to know the state of the port(s) you want to change; just the state in which you want them to be, and you won't change the state of any of the other ports.

The Saphira function, `sfRobotCom2Bytes`, packages the `sfCOMDIGOUT` command and values for you. For example:

```
sfRobotCom2Bytes(sfCOMDIGOUT, 0x20, 0x00)
```

resets bit 5 (OD0=0), and leaves everything else alone.

Accordingly, make the Gripper go up:

```
sfRobotCom2Bytes(sfCOMDIGOUT, 0x3, 0x3)
```

Go down:

```
sfRobotCom2Bytes(sfCOMDIGOUT, 0x3, 0x2)
```

And stop:

```
sfRobotCom2Bytes(sfCOMDIGOUT, 0x2, 0x00)
```

You have to monitor the various Gripper state switches to determine if and when it achieves a particular state—fully down with paddles open, for instance—when setting the Gripper's control I/O ports directly (see below).

4.4.2 State Commands

A more intelligent means of controlling the Gripper is through PSOS `sfCOMGRIPPER` command and state parameters. These automatically manage the Gripper so to achieve some pre-defined state. PSOS also monitors and reports the Gripper's progress towards that state, saving you some coding time and effort. And the state commands use a timeout feature to protect against unfortunate stalls or hang-ups.

To operate the Gripper this way, you simply issue a state command to the robot. You don't need to, but you may monitor the Gripper's progress towards the state via bits 0 and 2 of the digout part of the server information packet ("sfROBOT.digoutput"). During transitions, the state returned by the robot is set to `sfGRIPMOVING`. After successfully achieving a state, the state returned is that state. If there is a time-out (6 seconds) without achieving the state, the return state is set to `sfGRIPOFF`.

The various states and values are included in the script at the end of this chapter and that you may including in your own C programs.

There also are several Saphira convenience functions defined for gripper states:

```
int sfGripperGetState (void)
```

Returns the robot gripper state setpoint

```
void sfGripperSetState (int state)
```

Sets the robot gripper state setpoint

Table 3-2. Gripper functions

```

/*****
* Gripper functions --- For setting and retrieving gripper states
* States are sfGRIPUP, sfGRIPDOWN, sfGRIPMIDDLE, sfGRIPPOFF,
*   and sfGRIPMOVING
* sfGRIPMOVING occurs only during transitions between states
* sfGRIPPOFF can be set by the user, which means no state control
* sfGRIPPOFF is set by the robot to indicate a timeout (6
*   seconds) if a state is not achieved
*
*****/
/* states */

#define sfGRIPPOFF  0
#define sfGRIPUP    1
#define sfGRIPMOVING 2
#define sfGRIPMIDDLE 4
#define sfGRIPDOWN  5
#define sfCOMGRIPPER 33

/* gripper control bits (DIGOUTPUT) */

#define sfGRIPSTATEMASK 0x05 /* picks out state bits in DIGOUTPUT */
#define sfGRIPMOTORBIT 0x02 /* motor bit, 0 is off, 1 is on */
#define sfGRIPDIRBIT 0x01 /* direction bit, 0 is down, 1 is up */
#define sfSPEAKERBIT 0x04
#define sfRIGHTLEDBIT 0x08 /* 0 is off, 1 is on */
#define sfLEFTLEDBIT 0x10

/* gripper sense bits (DIGINPUT) */

#define sfGRIPTOPBIT 0x01 /* top limit switch, 0 if hit */
#define sfGRIPMIDBIT 0x02 /* carry switch, 0 if hit */
#define sfGRIPBOTBIT 0x04 /* open limit switch, 0 if hit */
#define sfGRIPINIRBIT 0x08 /* inner break beam, 1 is beam break */
#define sfGRIPOUTIRBIT 0x10 /* outer break beam, 1 is beam break */
#define sfGRIPBUMPBIT 0x20 /* gripper bumpers, 0 is bump */
#define sfGRIPMOMBIT 0x40 /* momentary switch, 0 is pressed */
#define sfGRIPSPSTBIT 0x80 /* SPST switch, 0 is up */

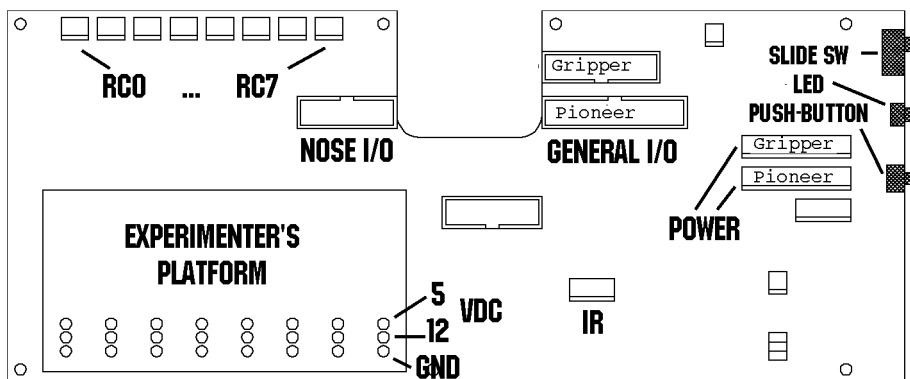
```

5. Experimenter's Module

Besides hosting the Gripper, the Experimenter's Module brings several new features to the Pioneer 1 Mobile Robot, and provides you with a platform for adding your own robotics options and features (Figure 4-1):

- ✓ Easy access to Pioneer digital I/O, A/D, and timer functions
- ✓ Speaker with new PSOS "say" command
- ✓ User-controlled LEDs and switches
- ✓ 8 RC-servo ports (RC0-7)
- ✓ 8 Analog-to-Digital Input Ports (AN0-7)
- ✓ Range-finding IR connector

Figure 4-1. Experimenter's Module.



5.1 I/O Ports

By cables connected to the Pioneer microcontroller, the Experimenter's Module brings both the Nose and General I/O expansion ports onto a single, open platform. All of ports now appear as solderable connections alongside a components pad, complete with power connections (Figure 4-1).

The I/O ports and their uses are summarized in Table 4-1. The Vcc ports supply logic power, so use them for that purpose only. The Pioneer microcontroller has a single A/D port that appears on both the General I/O connector and Nose I/O connector. This port is multiplexed into eight A/D ports on the Experimenter's platform. And special circuitry on the Experimenter's board lets you control up to eight separate RC servos.

Experimenter's Module

Note also that the Nose (N) 16-pin and the General (G) 26-pin I/O connectors are numbered odd pins on top and even pins on the bottom; not top, left to right, then bottom left to right. The pinouts for the Nose connector, for example, are:

1	3	5	7	9	11	13	15
2	4	6	8	10	12	14	16

Table 4-1. Pioneer I/O ports and connections

Label	Connector/Pin	Use
Digital Inputs		
ID0	G7	Gripper top switch
ID1	G9	Gripper carry switch
ID2	G11	Gripper open switch
ID3	G13	Paddle inner break beam
ID4	G15	Paddle outer break beam
ID5	N5	Paddle bump switches
ID6	N3	User push-button switch
ID7	N1	User slide switch
Digital Outputs		
OD0	G17	Gripper motor rotation direction
OD1	G19	Gripper motor enable
OD2	G21	User LED (amber)
OD3	G23	Right paddle and front LEDs
OD4	G25	Left paddle and front LEDs
OD5	N13	Speaker
OD6	N11	User servo motor logic
OD7	N9	User servo motor logic
Miscellaneous		
A/D	N7,G1	Analog-to-digital input
IDT	G3	Timer input
ODT	N15	Timer output
Logic Power		
Ground	N2,4,6,8,14,16; G2,4,10,12,14,16,18,20,22,24,26	
+5 VDC	N10,12; G6,8	

5.1.1 Digin: Self-Test

You may investigate the operation of the digital input and output ports through their PSOS self-test modes.

After starting up your Pioneer, press the black Function button once to enable self-test mode. The LCD message should now display:

```
BREAK, Boot to r
eturn      vv.vV*
```

indicating that Pioneer has entered self-test mode. (Note that the voltage V reading will vary.) You may press the Reset button at any time to disable the self-test, and you may press the Function button to skip any of the self-tests.

Skip to the digital input test by pressing the Function button 3 more times. The LCD should now read:

```
Digin:   E6  BOOT
to skip      vv.vV*
```

The E6 may vary—it is the hexadecimal representation of the various digital input ports, described in Table 4-1 and in the Gripper Chapter 4. Decode that indicator according to the bit rule in Table 4-2.

For example, if you press the user push-button switch, the E6 value would change to A6. See why?

Table 4-2. Bits and meanings of Digin: self-test value:

bit	7	6	5	4	3	2	1	0
1=	user slide-sw up	user push- button released	paddles bump free	outer b-b clear	inner b-b clear	paddles not open wide	gripper below/ above 4-5cm	gripper not at top
0=	user slide-sw down	user push- button pressed	paddles bump pressed	outer b-b obstruc- ted	inner b-b obstruc- ted	paddles wide open	gripper up 4- 5cm gripper at top	gripper at top

5.1.2 Digout: Self-Test

Press the black Function button on the Console once more to set the Pioneer into a self-test of its digital outputs. The test is pretty braindead: It simply toggles the odd and even output ports alternately off and on: AA, 55, AA, 55, and so on. Watch the LEDs flash on and off, the speaker click, and the Gripper twitch.

5.2 Programming the Digital I/O Ports

Use the sfCOMDIGOUT PSOS command and the related Saphira convenience command to manage the various digital output ports. They are described in detail in the previous chapter. (PAI and P-LOGO support similar commands, too.)

For example, to turn the right LED on, use:

```
sfRobotCom2Bytes(sfCOMDIGOUT, sfRIGHTLEDBIT,
sfRIGHTLEDBIT)
```

Or alternatively, specify the state of an LED:

```
void sfGripperSetLed(int which, int state)
```

for which=sfLEFT or sfRIGHT, and state=sfON or sfOFF.

5.3 Programming the A/D Ports

The eight analog-to-digital (A/D) ports on the Experimenter's Module (AN0-7) are multiplexed from a single A/D port on the Pioneer 1 microcontroller. Hence, only one port may be read at a time, although all eight ports may connect to individual sources.

Each AN port accepts a 0-5VDC analog input and outputs a proportionally related, single byte value 0-255. Use the digital output ports OD5-7 to address which port PSOS will read and return as the A/D value in the Server Information Packet (section 3.2, *PSOS Server Information Packet*).

For example, the following Saphira command addresses the AN4 port by masking the OD5-7 ports, setting OD7, and resetting OD5 and OD6: (remaining output ports remain untouched:

```
sfRobotCom2Bytes(sfCOMDIGOUT, 0xE0, 0x80)
```

5.4 Programming the Speaker

The latest PSOS implements a new command called SAY, which plays a sequence of up to 20 notes through the Experimenter's Module speaker. Now your Pioneer will be humming a different tune!

The SAY command (number 15) has two arguments: a string up to 40 bytes long, and the integer number of actual bytes in the string:

```
sfRobotComStrn(sfCOMSAY, str, n)
```

The string is comprised of duration and tone pairs; the duration byte is the number of 20 millisecond increments the following tone value half-cycle times—in 150 microsecond increments—should be played through the speaker.

For example, to play two short, high beeps using the Saphira equivalent command:

```
void sfPlayToneString ("\010\002\010\000\010\002", 6)
```

6. Maintenance & Repair

The Pioneer Gripper is built to last a lifetime and requires little maintenance.

6.1 Drive Lubrication

An occasional drop or two of oil on the guide rails is a very good idea. Place some thin, household oil on a Q-Tip or similar applicator, and rub along the rails. Then start up the Pioneer to exercise the Gripper and spread the lubricant.

6.2 Gripper Belt Adjustments

The Gripper's drive belts will stretch with use and may need to be tightened occasionally. We designed them to make that process easy.

Put the Gripper into its fully up position—its position after completing the self test. At the bottom of the Gripper assembly, you can see an aluminum gear with a vertical belt wrapped around it. Notice that it is fastened through a curved slot in the Gripper casing, and that it has a flat-bladed screwdriver slot?

Do not remove the screw, but loosen it enough so that it will slide in that slot. (You may need to hold the fastening bolt behind the slot with pliers or a nut-driver so you can loosen the gear.) With your thumb, press down on the gear inside the belt until the belt is fairly taut, but not “high strung”, then tighten the screw to hold the gear in place. Look inside the Gripper Bar. On the left side facing you (right side of the robot) is another belt and gear, similar to the up/down belt. This gear, too, has a screwdriver slot, which you loosen to tighten the belt.

6.3 Factory Repairs

If you are having *hardware* problems with your Pioneer 1 and, after reading this manual, you are satisfied that it needs repair, here's who to contact:

pioneer-support@rwii.com
(603) 532-6901 fax

In the body of your E-mail or fax message, describe the problem in as much detail as possible. Also include your name, E-mail and mail addresses, as well as phone and fax numbers, and when and how we can best contact you (we will assume E-mail is the best manner, unless otherwise notified).

We will try and resolve the problem through communication. If the robot must be returned to the factory for repair, obtain a shipping and repair authorization code and shipping details from us first. We are not responsible for shipping damage or loss.

7. Appendix A

Example C Program Demonstrates Gripper and Speaker Functions

```
/* #####  
* gripper.c --- Saphira-based functions for controlling the gripper and speaker  
*#####  
** Copyright 1997 by Kurt Konolige  
**  
** The author hereby grants to SRI permission to use this software.  
** The author also grants to SRI permission to distribute this software  
** to schools for non-commercial educational use only.  
**  
** The author hereby grants to other individuals or organizations  
** permission to use this software for non-commercial  
** educational use only. This software may not be distributed to others  
** except by SRI, under the conditions above.  
**  
** Other than these cases, no part of this software may be used or  
** distributed without written permission of the author.  
**  
** Neither the author nor SRI make any representations about the  
** suitability of this software for any purpose. It is provided  
** "as is" without express or implied warranty.  
**  
** Kurt Konolige  
** Senior Computer Scientist  
** SRI International  
** 333 Ravenswood Avenue  
** Menlo Park, CA 94025  
** E-mail: konolige@ai.sri.com  
**  
**/  
#include "saphira.h"  
/*  
* New comm function --- writes two bytes as argument  
*/
```

(continued)


```

void
sfRobotCom2Bytes(int com, int high, int low)
{
    sfRobotComInt(com, ((high & 0xff)<<8) + (low & 0xff));
}
/*****
* Speaker functions -- play a tone string
*
* Format of string is L1 T1 L2 T2 .... Ln Tn
* where Li is length of tone i in 20ms increments
* and Ti is tone half-cycle time in 150us increments
*
* Max length of string is 40 bytes (20 tones)
*
*****/
void
sfPlayToneString(char *str, int n)
{
    if (n > 40) n = 40;
    sfRobotComStrn(sfCOMSAY, str, n);
}
/*****
* Gripper functions --- set and retrieve gripper state
*
* States are sfUP, sfDOWN, sfMIDDLE, sfOFF, and sfMOVING
* sfMOVING occurs only during transitions between states
* sfOFF can be set by the user, which means no state control
* sfOFF can be set by the robot to indicate a timeout
*
*****/
/* states */
#define sfGRIPPOFF 0
#define sfGRIPUP 1
#define sfGRIPMOVING 2
#define sfGRIPMIDDLE 4
#define sfGRIPDOWN 5
#define sfCOMGRIPPER 33

```

(continued)

```

/* gripper control bits (DIGOUTPUT) */
#define sfGRIPSTATEMASK 0x05 /* picks out state bits in DIGOUTPUT */
#define sfGRIPMOTORBIT 0x02 /* motor bit, 0 is off, 1 is on */
#define sfGRIPDIRBIT 0x01 /* direction bit, 0 is down, 1 is up */
#define sfSPEAKERBIT 0x04
#define sfRIGHTLEDBIT 0x08 /* 0 is off, 1 is on */
#define sfLEFTLEDBIT 0x10
/* gripper sense bits (DIGINPUT) */
#define sfGRIPTOPBIT 0x01 /* top limit, 0 if hit */
#define sfGRIPMIDBIT 0x02 /* mid-level, 0 if hit */
#define sfGRIPBOTBIT 0x04 /* bottom limit, 0 if hit */
#define sfGRIPINIRBIT 0x08 /* inner IR, 0 is beam break */
#define sfGRIPOUTIRBIT 0x10 /* outer IR */
#define sfGRIPBUMPBIT 0x20 /* gripper bumpers, 0 is bump */
#define sfGRIPMOMBIT 0x40 /* momentary switch, 0 is pressed */
#define sfGRIPSPSTBIT 0x80 /* SPST switch, 0 is up */

```

```

int /* returns the robot gripper state setpoint */
sfGripperGetState(void)
{
    if (flakey.digoutput & sfGRIPMOTORBIT)
        return sfMOVING;
    else
        return flakey.digoutput & sfGRIPSTATEMASK;
}

```

```

void /* sets robot gripper state setpoint */
sfGripperSetState(int state)
{
    sfRobotComInt(sfCOMGRIPPER, state & sfGRIPSTATEMASK);
}

```

```

void /* sfLEFT, sfRIGHT, sfON, sfOFF */
sfGripperSetLed(int which, int state)
{
    switch(which)
    {
        case sfLEFT:
            which = sfLEFTLEDBIT;
            break;

```

(continued)

```
case sfRIGHT:
  which = sfRIGHTLEDBIT;
  break;
default:
  which = 0;
}
switch(state)
{
case sfOFF:
  sfRobotCom2Bytes(sfCOMDIGOUT, which, 0);
  break;
case sfON:
  sfRobotCom2Bytes(sfCOMDIGOUT, which, which);
  break;
}
}
```

8. Appendix B

Pioneer Server Information Packet

Name	Data Type	Description
Header	int	Exactly 0xFA, 0xFB
Byte Count	byte	Number of data bytes + 2; must be less than 201 (0xC9)
Status	byte = 0x3S; where S =	Motors status
	<i>sfSTATUSNOPOWER</i>	Motors power off
	<i>sfSTATUSSTOPPED</i>	Motors stopped
	<i>sfSTATUSMOVING</i>	Robot moving
Xpos	unsigned int (15 ls-bits)	Wheel-encoder integrated coordinates; platform-dependent units—multiply by
Ypos	unsigned int (15 ls-bits)	<i>DistConvFactor</i> in the parameter file to convert to mm; roll-over ~ 3 m
Th pos	signed int	Orientation in platform-dependent units—multiply by <i>AngleConvFactor</i> for radians
L vel	signed int	Wheel velocities (respective Left and Right) in platform-dependent units—
R vel	signed int	multiply by <i>VelConvFactor</i> to convert to mm/sec.
Battery	byte	Battery charge in tenths of volts
Bumpers	2 bytes - L and R	Motor stall indicators
Bumpers	unsigned int	
Control	signed int	Setpoint of the server's angular position servo—multiply by <i>AngleConvFactor</i> for radians
PTU	unsigned int	Pulse width of position servo
Say	byte	verbal/sound clues
Sonar readings	byte	Number of new sonar readings included in information packet; readings follow:
Sonar num	byte	Sonar number
Sonar range	unsigned int	Sonar reading—multiply by <i>RangeConvFactor</i> for mm
Input timer	unsigned int	User input timer reading
User Analog	byte	User analog input reading
User Input	byte	User digital input pins
User Output	byte	User digital output pins
Checksum	int	Checksum (see previous section)

9. Index

- A**
- A/D, 18
 - A/D port, 15
 - ActivMedia, Inc., 2
 - Additional Resources, 5
- B**
- break beams, 4
- C**
- carry switch, 4
 - COMDIGOUT, 12, 13
 - COMGRIPPER, 12
 - components
 - package, 1
 - Pioneer 1**, 2
 - user-supplied, 1
 - COMSAY, 18
- D**
- Digin, 17
 - digital-to-analog ports, 18
 - Digout, 17
- E**
- EPROM location, 7
 - Experimenter's Module
 - Description, 4
 - speaker, 18
 - Experimenter's Module, 15
 - components, 15
 - I/O ports, 15
 - self-test, 17
- F**
- FTP archives, 5
- G**
- General I/O, 16
 - GRIPMOVING, 13
 - GRIPPOFF, 13
 - Gripper
 - Bar, 4
 - belt adjustments, 19
 - carry switch, 4
 - Description, 2
 - direct control, 13
 - I/O, 12
 - installation. *See* installation
 - lubrication, 19
 - maintenance, 19
 - motor controls, 12
 - OD0, 12
 - OD1, 12
 - operating range, 3
 - operation, 11
 - paddles, 3
 - self-test, 11
 - state commands, 13
 - state switches, 12
 - states, 2
 - switches, 4
 - transition cycle, 4
 - Gripper Bar, 4
 - Gripper I/O, 12
 - Gripper maintenance, 19
 - Gripper motor controls, 12
 - Gripper switches, 12
 - GripperGetState, 13
- I**
- I/O
 - General, 16
 - Nose, 16
 - ports and connections, 16
 - programming, 18
 - I/O ports, 15
 - installation, 7
 - cables, 9

console removal, 7
EPROM location, 7
EPROM replacement, 7
microcontroller removal, 7

K

Konolige, Kurt, 2

L

LEDs, 18

M

majordomo, 5
microcontroller, 8
More, Grinnell, 2

N

newsgroups, 5
Nose I/O, 16
pinouts, 16

O

OD0, 12
OD1, 12

P

paddles
break beams, 4
positions, 4
paddles positions, 4
PAI, 11
Pioneer 1, 2
Pioneer Application Interface, 11
Pioneer Operations and Software Manual, 1
Pioneer Server Operating System. *See*
PSOS
Pioneer-LOGO, 11
PlayToneString, 18
P-LOGO, 11
PSOS, 11. *See* PSOS
EPROM, 7
PSOS EPROM, 7

R

RC-servo controllers, 4
Real World Interface, ii
Real World Interface, Inc., 2
RobotCom2Bytes, 13
RWI, ii. *See* Real World Interface, Inc.

S

Saphira, 11
sfCOMDIGOUT, 12, 13
sfCOMGRIPPER, 12
sfCOMSAY, 18
sfGRIPMOVING, 13
sfGRIPOFF, 13
sfGripperGetState, 13
sfGripperSetState, 13
sfPlayToneString, 18
sfROBOT.digoutput, 13
sfRobotCom2Bytes, 13
Saphira Software Manual, 1
SAY, 18
self-test, 11
Digin, 17
Digout, 17
Server Information Packet, 11
sfGripperSetState, 13
speaker, 18
SRI International, 2
state commands, 13
state switches, 12
support, 6

W

warranty, 28

Warranty & Liabilities

Your Pioneer 1 Mobile Robot and its accessories are fully warranted against defective parts or assembly for 90 days after it is shipped to you from the factory. This warranty explicitly *does not include* damage from shipping or from abuse or inappropriate operation, such as if the robot is allowed to tumble or fall off a ledge, or if it is overloaded with heavy objects.

The developers, marketers, and manufacturers of Pioneer 1 shall bear no liabilities for operation and use of the robot or any accompanying software except that covered by the warranty and period. The developers, marketers, or manufacturers shall not be held responsible for any injury to persons or property involving the Pioneer 1 Mobile Robot in any way. They shall bear no responsibilities or liabilities for any operation or application of the robot, or for support of any of those activities. And under no circumstances will the developers, marketers, or manufacturers of Pioneer 1 take responsibility for or support any special or custom modification to Pioneer 1.