PR2 user manual

Willow Garage

April 5, 2010

Contents

Chapter 1

Introduction

This manual is intended to give you enough information to successfully install, use, and develop code on your PR2 robot. The software on the PR2 is based on ROS. The recommended source of information for learning about ROS and the higher-level software available for the PR2 is http://ros.org

If you want to get started running the PR2 as quickly as possible, please start with Chapter 2 on safety (seriously - the PR2 is a big machine and can cause serious injuries or death), and then you can skip to chapter 5 to learn how to start up and run the PR2.

1.1 Before you start

Space To use PR2 you will need to have enough room for it to drive around and move it's arms. The PR2 is designed to move through ADA-compliant spaces (Americans with Disabilities Act), so corridors should be at least 36" wide, doorways should be at least 32", and the ground should be flat and level. You will need enough space for the PR2 to move around and perform tasks.

Safe Environment The space where the PR2 operates should be free of hazards. Specifically, stairways or other fall hazards can pose an extreme danger and the PR2 should not be operated near any type of dropoff. You should also avoid hazardous objects, such as knives, sources of fire, hazardous chemicals, or furniture that could be knocked over. See chapter 2 for more details on making sure your environment is safe.

Electrical The PR2 recharges using a standard 120V American power outlet. The robot can draw 15A of current when plugged in, so we strongly recommend recharging the PR2 only on outlets with no other devices on the circuit breaker.

Development tools You will need at least one laptop or desktop computer to use to connect to the robot. The PR2 ships with a base-station computer, which is a desktop, but you will need to provide a screen, mouse, and keyboard. A laptop with wireless access is ideal.

Linux We highly recommend familiarity with the Linux command-line. The PR2 computers both run Ubuntu, and since they don't have attached displays, all tasks on them have to be performed by logging in remotely (e.g. via ssh).

ROS Since all the PR2 software is based on ROS, running through the beginning tutorials available at ROS.org will help you understand the structure of the software on the robot and will give you tools to understand the software that's running and the data that is moving around in the system.

PR2 Safety Be sure to familiarize yourself with the contents of the following chapter on safety before using the PR2.

Chapter 2

Safety

Safety is a key goal at Willow Garage. It is important, it is challenging, and it is a continual process shared by the designer, user, and administrator of a robot. In the following we provide an overview of the issues, describe some safety-related design features, enumerate a set of basic usage guidelines to support safety, and finally detail the explicit safety program.

2.1 Overview

Safety is a vitally important concern whenever you are around a PR2. Its a heavy piece of equipment with many moving parts. It travels though the environment and can carry and manipulate a wide variety of objects. Its movements and actions are not completely predictable. It can cause significant damage if it falls on or runs over someone. There are several ways it can pinch, grab, and twist fingers or other parts of the body. It can wield dangerous implements and knock heavy things over. You must always be cautious and attentive when you are around a PR2.

Safety is neither absolute criterion nor a one-time event. Instead it is an appreciation that risks are inherent to any robotic endeavor and they must be minimized as well as weighed against benefits. It is a goal for which the entire community must continually strive.

We emphasize safety to avoid harm to any person, animal, or equipment. We also recognize that robots in general will not gain wide acceptance or fulfill their potential unless the community can adequately manage safety.

Managing safety is challenge when dealing with any complex engineering system. In the case of the PR2, consider also the open, extensible, programmable, experimental nature of the platform. The PR2's capabilities and behaviors change over time, with user interactions, and with reprogramming.

With this in mind, Willow Garage has chosen a three-fold approach. First, we have designed the PR2 to minimize potential risks and maximize inherent safety, cognizant of its uncertain uses. Second, we communicate to all users about how to minimize risk. And third, we have implemented an explicit safety program to ensure that the community continues to identify potential hazards, seek design mitigations, and communicate effective usage guidelines.

2.2 Design Features

We have designed both hardware and software to minimize risks, while retaining the power of an open platform. These aspects of the design are often described as inherent safety features. For example, PR2's arms are backdrivable. That means when an arm encounters an object, be it a table or a person, the interaction will drive the motors back and bring the arm to a stop. The PR2 arms can't "punch through" an object the way traditional industrial robots can.

We have further designed the PR2 arms with relatively small motors with respect to their payload. This is possible due to a spring counterbalance offsetting the gravity forces acting on the arms. That is, the arms do not need to hold their own weight against gravity. And so the motors need only be strong enough to hold the payload. The arms simply can't push very hard.

In software, we have incorporated low level checking to limit the current in a motor, to limit the velocity of a motor, and to limit the range through which a motor should travel. We obviously discourage users and developers from changing these configurations. High level applications also avoid obstacles in navigation and movement using the various on-board sensors.

These design choices also help make PR2 robust. However, a robot with the PR2s capabilities can never be absolutely safe. Your safety as well as the safety of others critically depends on your constant attention. You must be aware of the potential dangers, anticipate possible problems, and plan to prevent their occurrence.

2.3 General Usage Guidelines

While many guidelines for the safe use of a robot stem from common sense, we enumerate a basic set here. We stress the importance of following these guidelines but re-emphasize that these guidelines alone do not guarantee safety but reduce risk.

- Every organization that uses a PR2 must appoint a Safety Officer.
  - The Safety Officers contact information should be known by everyone in the organization who uses the PR2 (including designers, developers, programmers, and end-users).
  - Further details of the Safety Officers roles and responsibilities are described in section 2.4.
- Before operating or working with the PR2 you must do the following:
  - view the safety video
  - read this User Manual, particularly Chapter 2 on Safety
  - read and understand the latest list of potential hazards
  - know how to contact your organizations Safety Officer
- Supervise children, visitors, and anyone who has not followed the previous guideline. In particular, make sure they
  - do not come within range of the PR2 when active
  - are aware the robot could move unexpectedly and is potentially dangerous
  - are not alone with the PR2
  - do not operate the PR2
- Maintain a safe environment. Safety is not only an issue of how you operate the robot, but also the environment.
  - Make sure the robot has adequate and level space for any expected or unexpected operation
  - Make sure the the environment is free of objects that could pose a risk if knocked, hit, or otherwise affected by the PR2
  - Make sure no animals are the near the robot.
- The PR2 is designed to operate in an laboratory environment

  - It should not be operated outdoors
  - It should not come in contact with liquids
  - It should not be operated within 7 meters of the top of a stairway

- Anticipate potential problems and hazards. Always imagine what might happen if the robot malfunctions or behaves in a way different from the desired action. Be vigilant.

- The operator should always have immediate access to the run/stop and stop the robot at the first sign of a problem.

- Use common sense when operating the robot.

  - Do not allow the robot to grab or hit any person
  - Do not allow the robot to drive into contact with or over any body part.
  - Do not allow the robot to interact with any sharp or dangerous items

- Pay attention to warning labels on the robot.

- Do not remove the covers of a PR2 without prior and appropriate instruction by Willow Garage. There are high voltages and a variety of pinching and other mechanical dangers in the interior of the robot.

- Do not modify or remove any part of the software safety features.

2.4 Safety Program

Safety is a continual process, and in particular should include

- an awareness of risks

- a critical examination to expose risks, assess risks, discover mitigations, and evaluate trade-offs

- any deliberate actions needed to minimize and mitigate current and future risks.

To facilitate this process and communication within the community, Willow Garage has implemented a safety program.

2.4.1 Willow Garage Safety Board

The Willow Garage Safety Board identifies hazards related to Willow Garage products and ensures that appropriate actions are taken to mitigate those hazards. The Board maintains a database to keep track of the hazards and mitigation actions. To identify additional hazards for the database, the Board commissions Hazard Brainstorming Meetings and reviews incident reports from the field. To initiate actions that reduce the severity and/or likelihood of specific hazards, the Board commissions Hazard Response Projects. To ensure that appropriate actions remain in force, the Board commissions Hazard Response Audits. The Board works with Safety Officers in external organizations to improve safety across all users of Willow Garage products. The core Safety Board includes senior members of the Willow Garage management. Additional employees serve for term appointments. Members of the Board spend at least one day a month on safety related work.

2.4.2 Hazard Database

The Hazard Database includes three types of entity: Hazards, Incidents, and Responses. Hazards describe things the system might do that can cause damage, for example, running into something or falling down stairs. Each Hazard includes an estimate of its severity and likelihood of occurrence. Based on these estimates, the Hazard is assigned a priority for action. Incidents describe specific examples of Hazards, either things that have actually occurred or hypothetical occurrences. Each Incident is associated in the database with the Hazard(s) that it exemplifies. Responses describe actions that reduce the severity and/or likelihood of a hazard. These can involve changes to the design or documentation of the hardware and software, additional warnings to the community, or changes to the safety training and video.

The information in the Hazard Database is summarized in several reports that document the current understanding of potential hazards and the associated mitigating actions. Chief among these is the Prioritized Hazard Report, which lists the Hazards in the database in order of their priority.

2.4.3 Safety Officers

Each organization that uses Willow Garage products will appoint a Safety Officer who is responsible for all aspects of safety in the use of those products. The Safety Officer will:

- remain informed of all known safety hazards and mitigations,
- ensure that all known mitigations are implemented in their organization,
- ensure that everyone involved with the products receives safety training,
- report any safety incidents to Willow Garage in a timely fashion, and
- work with the Willow Garage Safety Board to improve safety.

Chapter 3

PR2 hardware

3.1 What's in the box

THe PR2 ships in 3 packages:

Robot crate	Large wooden crate which contains the PR2 itself
Accessory crate	Crate which contains accessory kit, tool kit, and robot base-station
Calibration target	Large checkerboard for accurate calibration of stereo cameras

3.1.1 PR2

See later sections for more detail on the PR2 robot itself and how to get it out of the crate.

3.1.2 Accessory kit

Wireless Joystick

The PR2 ships with a bluetooth joystick for teleoperating the robot. The bluetooth joystick is a Sony DUALSHOCK3 (Figure 3.1) wireless controller. It can be charged using any standard USB A to mini-B USB cable. For more information, see the ps3joy package at ros.org.

Wireless run-stop

The PR2 comes with an OMNEX T50 wireless run-stop transmitter. When stopped or out of range, the wireless run-stop transmitter will halt the motors and put the power system in standby mode.

Figure 3.1: The PR2 bluetooth joystick.

Figure 3.2: The PR2 wireless run-stop.

To start the run-stop, press the green start button (Figure 6.1); if this works properly, you will see a light flashing on the run-stop. While transmitting, the run-stop has a range of approximately 800 ft. The run-stop is powered by four AA batteries; the battery light will flash when the battery charge is low, which is an indication that you should change the batteries.

15 FT Robot Power Cord and 3-foot self plug-in Robot Power Cord

TODO: Include images of both power cords To recharge the PR2's batteries, it must be plugged into a 120V electrical outlet. Use only the provided power cords - many power cords and power strips have thinner conductors and cannot safely supply the current required by the PR2. Using inappropriate cables is hazardous and may cause fire. When plugging the robot is to charge by hand, you may use either cable. When attepting to have the PR2 plug itself in, you must use the shorter self-plug-in cord with attached checkerboard. The self-plug-in cord has magnets on the side opposite the checkerboard which should be used to attach it to the base as shown TODO: Include image of self-plug-in cable attached to magnets

Sensorized fingertips and boots

TODO: Include image of fingertips and boots The PR2 ships with sensorless fingertips attached. We have also developed fingertips which have an integrated 22-element pressure sensor, but they are easy to damage and are not as robust as the rest of the PR2. We provide 5 fingertip sensors (2 for each gripper, plus 1 spare), as well as 20 rubber protective "boots" which prevent the sensor from coming into direct contact with the environment. When using the fingertip sensors, you should always have an undamaged boot in good condition installed over the sensor. Continuing to use the fingertip sensors after the rubber boot is damaged greatly increases the chance that you will damage the sensors themselves. See section TODO: Create maintenance section for replacing fingertip sensors for more information.

Small calibration target

TODO: Include image of small checkerboard This calibration target, which looks like a checkerboard, is used for the PR2 self-calibration. The robot holds the checkerboard in one gripper and uses it to calibrate the arms, the head, the cameras, and the tilting laser together.

User Manual

Printed version of this document

3.1.3 Toolkit

TODO: Include image of open toolkit, with callout labels to all tools by proper name

TODO: TBD: The robot toolkit is not yet fully defined

3.1.4 Large calibration target

The large calibration target which ships with the robot looks like a checkerboard 1" thick and approximately 3 feet on a side. This is the recommended calibration target to use for calibrating the intrinsics of the stereo cameras on the robot. The robot ships with stereo cameras already calibrated, but you may need to re-calibrate occasionally as vibration and thermal effects change the parameters over time.

3.1.5 Base-station computer

TODO: Include picture of base-station here The base-station computer ships without a monitor, keyboard, or mouse. See section ??? for more information on how to configure your base-station.

3.2 Mechanism

PR2 is a 32-dof mobile manipulator with a mobile base, two arms, and a variety of sensors on a pan-tilt head.

3.2.1 Robot Anatomy

3.2.2 Terminology

TODO: From below, define link, joint, frame, motor, etc as well as basic naming conventions.

3.2.3 Links and Joints

TODO: Pulling from below, describe each component of the robot. Include updated tables of joint limits, torques, home positions, etc.

TODO: Provide links to more information (URDF, etc.)

3.2.4 Drivetrains

TODO: For each type of drivetrain, start with a list of joints and provide key information (invert mapping from what Tully did)

Kinematically speaking, the PR2 can be broken down into **links** by individual degrees of freedoms. Each **link** consist of one rigid body component that can be represented by a single collision model and a single lumped inertial property. All links are interconnected together by **joint** constraints in the form of revolute, prismatic or fixed **joints**. In this literature, a revolute joint without limits is also called a continuous joint. In general, the entire PR2 mechanism is represented by a tree where each node is a **link** and the edges connecting individual nodes are **joints**. In the tree representation of the PR2, each node (link) can be connected (via joints) to only one parent node but can have arbitrarily many child nodes.

A link without collision or inertial components is referred to as a **frame**. Frames are always rigidly attached to parent links via fixed joints and cannot have children links. A frame serves an important function of facilitating transform record keeping through our Transform and Frames Library (TF) for propagating information such as reference coordinates of an optical image, origin of range sensor generated point cloud data or the reference pose for the grasp point of a gripper.

A CAD approximated kinematic and inertial description of the PR2 can be found in the pr2_description package. This package uses a robot specific Extensible Markup Language (XML) and Document Object Model (DOM) representation called Uniform Robot Description Format(URDF) created here at Willow Garage.

Rather than diving into all 32 DOF's of the PR2 in detail right away, the next section examines the robot by its major components: head, torso, arms, base and casters. This overall breakdown of the PR2 robot is depicted in Figure **??**.

Head

The PR2 head consists of two links: **head_pan_link** and **head_tilt_link**. The **head_tilt_link** is attached via a revolute joint called **head_tilt_joint** to the **head_pan_link**. The **head_pan_link** is attached to the PR2 torso (**torso_lift_link**) by another revolute joint called the **head_pan_joint**.

The individual head sensors are detailed separately in the section 3.2.4

Arms

The PR2 arms can be further subdivided into shoulder, upper arm, forearm and gripper. Each components is separately described in detail below:

- Shoulers
  - **l_shoulder_pan_link r_shoulder_pan_link**
  - **l_shoulder_lift_link r_shoulder_lift_link**
  - **l_upper_arm_roll_link r_upper_arm_roll_link**
- Upper Arms
  - **l_upper_arm_link r_upper_arm_link**

  - **l_elbow_flex_link r_elbow_flex_link**
  - **l_forearm_roll_link r_forearm_roll_link**

- Forearms
  - **l_forearm_link r_forearm_link**
  - **l_wrist_flex_link r_wrist_flex_link**
  - **l_wrist_roll_link r_wrist_roll_link**

- Grippers
  - **l_gripper_palm_link r_gripper_palm_link**
  - **l_gripper_l_finger_link l_gripper_r_finger_link r_gripper_l_finger_link r_gripper_r_finger_link**
  - **l_gripper_l_finger_tip_link l_gripper_r_finger_tip_link r_gripper_l_finger_tip_link r_gripper_r_finger_tip_link**

Torso

The PR2 **torso_lift_link** refers to the torso part of the PR2 that moves up and down with the elevator actuator.

Base

The PR2 **base_link** is the lower part of the PR2 body, housing the PR2 computers and is also rigidly attached to the base Hokuyo laser scanner.

Casters

The PR2 has 4 caster units. Each caster contains one actuated steering degree of freedom and two individually actuated wheels.

Home Pose

In order to describe the PR2 robot pose and joint positions in a consistent manner, a **home pose** of the robot has been defined. The **home pose** is described in detail in section 3.2.4.

Sensors

Details of the PR2 sensors are described here. The PR2 contains the following sensors:

- Image Sensors
  - Dual Stereo Cameras
    - Narrow Stereo Camera Pair
    - Wide Stereo Camera Pair
  - Prosilica Camera
  - Forearm Cameras
- Range Sensors
  - Tilting Hokuyo Laser Scanner
  - Base Hokuyo Laser Scanner
- Accelerometers
  - IMU
  - Gripper Palm Mounted Accelerometers

PR2 Coordinate System

The reference **home pose** of the PR2 robot is defined as the robot pose with all the joint angles at zero, with the PR2 robot facing the positive x-direction, positive z-axis pointing upwards and positive y-axis pointing to the *robot-left* (see Figure **??**).

3.2.5 Drivetrains

The PR2 drivetrains have been designed to be low inertia and backdrivable with minimal backlash. There are three types of drivetrains on the PR2, gear drives, belt drives, and lead screw drives.

Belt Drives

The most common drivetrain on the PR2 is the belt drive.

The typical link drivetrain is a DC motor with attached encoder. It runs through a planetary gearbox then is attached to a belt drive. These drivetrains are fully backdriveable and have low backlash.

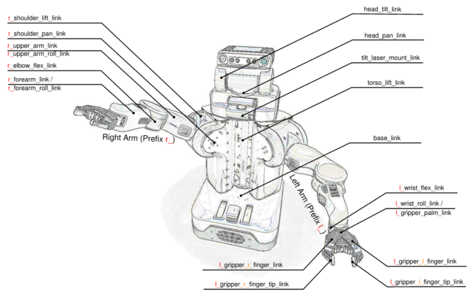


Figure 3.3: The PR2 URDF Link Naming Scheme.

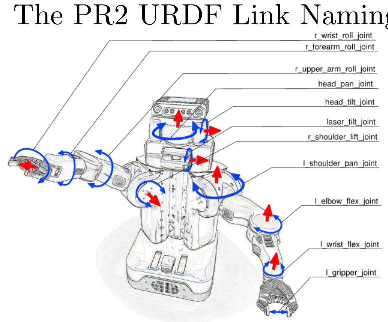


Figure 3.4: The PR2 URDF Joints Naming Scheme.

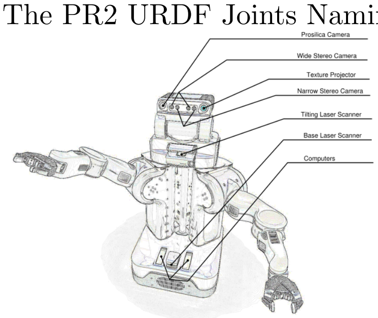


Figure 3.5: The PR2 Sensors.

Standard Belt Drives The following motors use belt drives as described above: Caster Wheel, Caster Turret, Head Tilt, Shoulder Pan, Shoulder Lift, and the Upper Arm Roll.

Semi-Standard Belt Drives The Upper Arm Flex is standard except that there is an additional 90 degree bevel gear pair on this joint between the gearbox and the belt drive. The Laser Tilt is standard except that there is no gearbox, there is a flex coupler in it's place. And the encoder is on the mechanism side of the flex coupler. The Head Pan is standard except that the gearbox is replaced by gears, with the encoder on the mechanism side of the gears.

Gear Drives

The wrist, head pan and forearm roll use gear drives.

Wrist The wrist utilized a differential drive between two motors. If they are turned in the same direction the wrist rolls, if they are turned oposite the wrist flexes. The differential bevel gears are attached after a standard encoder, motor, gearhead assembly.

TODO: wrist diagram

Forearm roll The forearm roll has a traditional encoder, motor, gearbox assembly with a pinion gear attached to the output shaft. The pinion gear directly drives a nylon gear built into the forearm body.

Lead Screw Drives

There are two drive trains which use lead screws as their primary method of movement, the torso lift and gripper.

Torso Lift The torso lift is driven by a vertical lead screw. The motor is augmented by a gas spring vertically such that the motor only needs to lift a portion of the weight of the torso. The torso is also mounted such that the lift mechanism cannot pull downward.

Gripper The gripper is a pair of 4 bar linkages which are coupled through a gearing on one of the piviots. The mechanism is actuated using a lead screw between the two fourbar linkages. Between the actuator and the gearing the movement of the grippers is limited to a single degree of freedom.

The gripper is backdriveable when off, however challenging. Backdriving the gripper when torque is applied is very hard, but possible.

Arm Counter Balance Although not an actuated drivetrain the arm counter balance is one of the more complicated mechanisms on the PR2. Inside of the shoulder turret there are two springs. These springs are attached to belts which run over cams and provide a uniform downward force.

In the upper arm there is a four-bar linkage which transmits the torque through to the elbow flex joint. As the arm moves through its configuration space the linkage changes the amount of leverage with which the counter-balance is pulling which compensates for the different configurations.

Drivetrain Limitations

The drivetrains are high fidelity however there are still limitations. Below are the most observable limitations.

Belt Stretch On all the belt driven joints. The belts do stretch when under load. The stretching is small but measureable.

Encoder Discritization As the arm is moving slowly the discrete nature of the encoders can become evident. It usually manifests itself as a jitter in the velocity. It can be filtered out to some extent, but not too much for it will cause lag.

Discuss the drive-train approach, how/why things work, what types of errors we expect to see and don't expect to see

3.2.6 Motion control

Motor controller boards

Each motor/encoder on the PR2 has its dedicated *Motor Controller Board* (MCB). The MCB detects and counts transitions in the encoder signal, measures the current the motor is using, and commands the voltage going to the motor. Each MCB runs a PI-control loop to control the motor current to a desired value, by commanding the motor voltage. This control loop is executed at 100 kHz on an FPGA. A shared etherCAT link allows all MCB's in the PR2 to communicate with the main computers.

Controller manager

The *Controller Manager* (CM) is the lowest level software component that directly talks to the MCB's. The CM runs in a hard realtime process and is guaranteed to be executed at 1 kHz. This means the CM sends desired

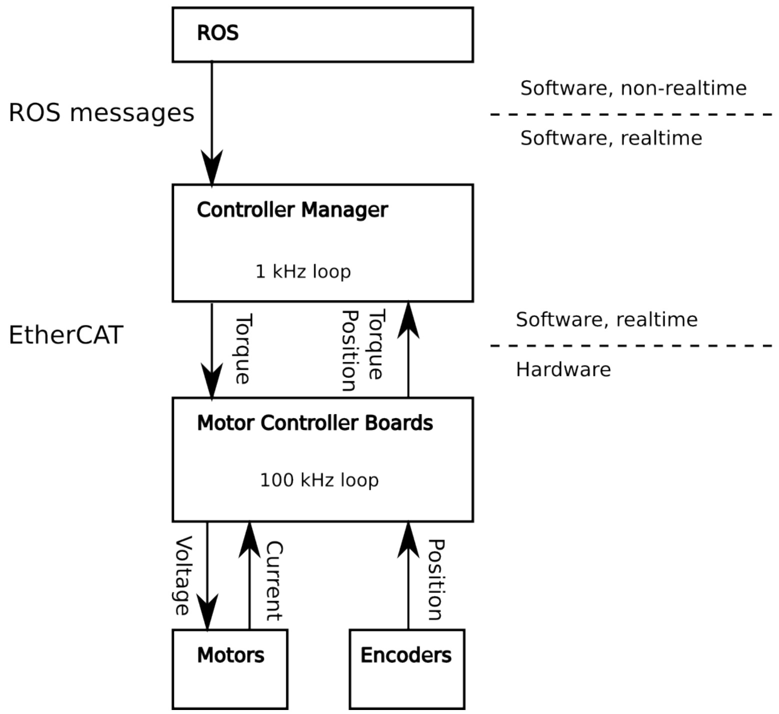


Figure 3.6: Motion control layout.

motor torques to the MCB's every milisecond, and receives measured motor torques and positions in the same milisecond.

The CM has a dynamic plugin loading mechanism that allows controller algorithms to get inserted and executed inside the 1 kHz realtime loop.

3.2.7 Mechanical specs

Before undertaking any new or risky operations with your PR2, please consult these mechanical specifications. Do not operate the PR2 outside of these specifications. If you have any questions about your PR2's specifications, please contact Willow Garage Support.

Environmental specs

The PR2 is an indoor, household robot. Operating outside this type of environment could cause damage to the PR2, or injury or death to operators.

Water The PR2 has not been tested for any type of contact with water or any other liquid. Under no circumstances should the PR2 come in contact with water from rain, mist, ground water (puddles) or liquid handling. Water contact can cause damage to the electrical circuitry and the mechanism.

Temperature and Humidity Temperature testing of the PR2 has allowed the unit to run between 15 and 35C. Temperatures outside of this range can cause malfunctions in the PR2 power system and instruments. The PR2 has been tested in high humidity, but under no circumstances should condensation be allowed to form on the vehicle.

Drive Surface The drive surface of the PR2 must be capable of supporting the entire weight of the PR2, about 450 pounds (220 kgs). If the surface is too soft, the PR2 can get stuck and fail to drive. A commercial carpet or tile is recommended.

Incline Surface The PR2 is ready for ADA compatible ramps, which specifies a 1/12 slope. Ramps that are steeper than 1/12 slope are unsafe and may be a tip over hazard.

Other Environmental Specs

- UV exposure should be minimized. UV radiation can damage the PR2's skin
- Dust and dirt can clog air filters

Forces and torques

Joint position, velocity, and force limits are implemented in the PR2's URDF file, in the "/etc/ros/urdf/robot.xml" file on your PR2. These joint limits control the range of travel of the mechanism, the allowable velocity to prevent overtravel. These limits are enforced by pr2_controller_manager, and are designed to prevent poorly designed controllers from damaging the PR2 and harming operators.

The limits below are from the PR2 URDF file. If a velocity or torque limit is not specified, no value is enforced by pr2_controller_manager.

Joint	Velocity (rad/s or m/s)	Torque (Nm or N)
*_caster_rotation_joint	-	-
*_caster_wheel_*_joint	-	-
torso_lift_joint	0.013	10000
laser_tilt_joint	10.00	0.65
head_pan_joint	6.00	2.65
head_tilt_joint	5.00	15.00
*_shoulder_pan_joint	2.10	30.00
*_shoulder_lift_joint	2.10	30.00
*_upper_arm_roll_joint	3.27	30.00
*_elbow_flex_joint	3.30	30.00
*_forearm_roll_joint	3.60	30.00
*_wrist_flex_joint	3.10	10.00
*_wrist_roll_joint	3.60	10.00
*_gripper_joint	0.20	1000

The PR2 motor controller boards (MCB's) will not allow a current command greater than the maximum continuous current specified for the joint's actuator. This means that maximum joint effort may be lower than the maximum effort specified above. Below are the actuators for each joint (Maxon part number), and their maximum allowable commanded current.

Joint	Motor	Power (W)	Max Current
*_caster_rotation_joint	236672	20	0.655
*_caster_wheel_*_joint	236672	20	0.655
torso_lift_joint	148877	150	3.12
laser_tilt_joint	310009	60	1.72
head_pan_joint	310009	60	1.72
head_tilt_joint	310009	60	1.72
*_shoulder_pan_joint	148877	150	3.12
*_shoulder_lift_joint	148877	150	3.12
*_upper_arm_roll_joint	148877	150	3.12
*_elbow_flex_joint	148877	150	3.12
*_forearm_roll_joint	310009	60	1.72
*_wrist_flex_joint	310009	60	1.72
*_wrist_roll_joint	310009	60	1.72
*_gripper_joint	222057	11	0.204

More information about each actuator may be found in Maxon datasheets.

Joint Limits and Types

The position limits for the PR2 are specified below. These "hard limits" are the maximum travel for the mechanism.

Joint	Type	Limit (+)	Limit (-)
∗_caster_rotation_joint	continuous	-	-
∗_caster_wheel_∗_joint	continuous	-	-
torso_lift_joint	prismatic	310 mm	0 mm
laser_tilt_joint	revolute	85°	45°
head_pan_joint	revolute	168°	168°
head_tilt_joint	revolute	60°	30°
r_shoulder_pan_joint	revolute	40°	130°
l_shoulder_pan_joint	revolute	130°	-40°
∗_shoulder_lift_joint	revolute	80°	30°
r_upper_arm_roll_joint	revolute	44°	-224°
l_upper_arm_roll_joint	revolute	224°	-44°
∗_elbow_flex_joint	revolute	133°	0°
∗_forearm_roll_joint	continuous	-	-
∗_wrist_flex_joint	revolute	130°	0°
∗_wrist_roll_joint	continuous	-	-
∗_gripper_joint	prismatic	86 mm	0 mm

Modifying Joint Limits

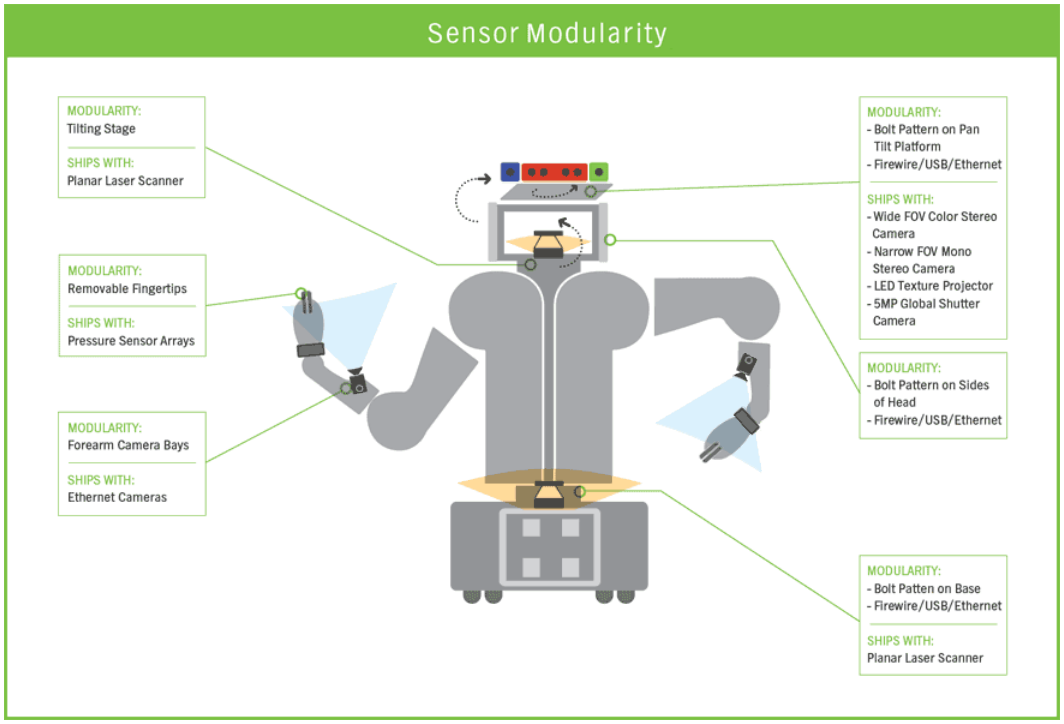
On the PR2, "soft limits" stop the joints from reaching the full range of motion to prevent damage to the mechanism. These soft limits, similar to a virual spring, are specified in the robot's URDF file. For an explanation of their implementation, see http://www.ros.org/wiki/pr2_controller_manager/safety_limits.

The soft limits have been carefully implemented and validated by Willow Garage. Under no circumstances should they be changed without prior written authorization by Willow Garage Safety. Unauthorized and unvalidated modification of these limits could cause mechanism damage and injury or death to PR2 operators.

Changing maximum allowable effort or maximum actuator current could cause serious damage to the PR2 and injury or death to operators.

3.3 Sensors

The PR2 has a variety of sensors spread out over it's body:



3.3.1 Base Laser

The base laser of the PR2 is Hokuyo Top-URG (UTM-30LX) scanning range finder that is located on PR2's base. This laser has a 30 m and 270° scanning range. For more information, see the hokuyo_node package at ros.org.

3.3.2 Tilting Laser

In addition to the base laser, the PR2 has a Hokuyo Top-URG (UTM-30LX) mounted on a tilting platform that is located just below the pan-tilt head. The tilting platform can sweep the scanning laser through 135° (+90° and −45° from level) and can be controlled using the default laser_tilt_controller. For more information, see the hokuyo_node and pr2_default_controllers packages at ros.org.

3.3.3 Head Cameras

The PR2 pan-tilt head has three cameras and a textured light projector:

Wide Stereo Camera The wide stereo camera of the PR2 is part of the dual stereo pair and is a 100Mb color ethernet camera. The wide stereo uses the Aptina MT9V032C12STC imager chip and has a maximum resolution of 752 x 480 pixels at 15 fps. The camera has a field of view (FOV) of approximately 90° and a 2.5mm F2.5 Marshall V-4402.5-2.5-HR lens. For more information, see the TODO package at ros.org.

Narrow Stereo Camera The narrow stereo camera of the PR2 is part of the dual stereo pair and is a 100Mb monchrome ethernet camera. The narrow stereo uses the Aptina MT9V032C12STM imager chip and has a max resolution of 752 x 480 pixels at 15 fps. The camera has a FOV of approximately 55° and a 5.6mm F2.0 Marshall V-4405.6-2.0-HR lens. For more information, see the TODO package at ros.org.

Gigabit Ethernet Camera The PR2 has a gigabit ethernet camera located to the left of the dual stereo pair on the pan-tilt head. The gigabit ethernet camera is a Prosilica GC2450C, which uses the Sony ICX-625AQ imager chip and has a maximum resolution of 2448 x 2050 pixels at 15 fps. Additionally, the gigabit ethernet camera has a 8mm F1.4-F16 Kowa LM8JC lens. For more information, see the prosilica_camera package at ros.org.

Textured Light Projector The PR2 has a textured light projector located to the (robot's) left of the dual stereo pair on the pan-tilt head. The projector has a FOV of approximately 55° and a 5.6mm F2.0 Kowa LM12JC lens. For more information, see the TODO package at ros.org.

3.3.4 Forearm cameras

Each forearm of the PR2 is equipped with a 12V 100Mb color ethernet camera. The forearm camera uses the Aptina MT9V032C12STC imager chip and has a maximum resolution of 752 x 480 pixels at 15 fps. Additionally, the forearm camera has a 2.5mm F2.0 lens. For more information, see the wge100_camera package at ros.org.

3.3.5 Gripper Sensors

Accelerometer The gripper of the PR2 is equipped with a Bosch BMA150 digital triaxal accelerometer. The measurement range (±2g, ±4g, or ±8g) and bandwidth (25Hz - 1500Hz) of the accelerometer can be selected in software. For more information, see the TODO package at ros.org.

Fingertip Pressure Sensors The default fingertips of the PR2 are sturdy aluminum blocks with non-slip rubber covers, for added friction and compliance in grasping. However, the aluminum tips can be swapped out for an (included) set of RoboTouch tactile sensing pads made by Pressure Profile Systems, each with 22 tactile sensing elements: 15 in a 5x3 array on the front surface, 2 on top, 2 on each side, and 1 in back, near the top. Each tactile element has a pressure range of 0-30 psi (0-205 kPa) and sensitivity of 0.1 psi (0.7 kPa). The sensors connect to the robot via an SPI ribbon cable and two screws, and have a maximum scan rate of 35 Hz.

The tactile sensors are highly fragile when not protected by the rubber covering, and so care should be taken not to damage the rubber covering when the tactile sensors are being used on the robot. For more information, see the fingertip_pressure package at ros.org.

Calibration LED

3.3.6 Inertial measurement unit

The PR2 has an inertial measurement unit (IMU) located next to the tilting laser. The IMU is a MicroStrain Inertial-Link 3DM-GX2 which has an accelerometer range of ±5g and a gyro range of $300°/s$. For more information, see the microstrain_3dmgx2_imu package at ros.org.

3.3.7 Speaker

The PR2 has one Logitech V20 notebook speaker that is located under the pan-tilt head next to the tilting laser. For more information, see the sound_play package at ros.org.

3.4 Power system

3.4.1 Overview

The PR2 has a Lithium-ion (Li-ion) battery system that is charged off of 120V wall current. The batteries power the computers, motors, and sensors on the robot. Power distribution is controlled by the *Power Board*, which communicates over Ethernet.

3.4.2 Power Busses

The robot has four internal power busses:

120v The robot has a 3-prong IEC320 plug in the back of the base, which is connected through a circuit breaker to the inputs of the four AC-DC converters that are used to charge the battery packs. The system is designed to draw less than 15A, a standard 15A or 20A outlet will suffice. The load is capable at times of approaching the 15A limit, you should make sure there are no other devices which draw significant power on the same circuit (e.g., computers or heaters). Ideally, add image of back panel once we have the PR2 betas done w/ Ryan's stickers on t

Motor Power Bus The motors get power through 3 descrete circuit breakers on the power board. These create three independent motor power busses: left arm, right arm, and base/head. The motor power bus can be in one of three states. *Enabled*, provides a direct connection to the unregulated battery power. The voltage will ranges from 52V when the batteries are fully discharged up to 72V when connected to wall power. *Standby*, the power board provides a low-power 18V supply that is used for communication and maintaing encoder position counts. *Disabled*, total shut-down of the circuit (e.g., in the event of a major power-system problem or manually disabled through the control panel).

12V system bus 12V power for sensors supplied by the power board. This is also the recommended source of power for user supplied accessories.

12V computer power 12V power for the computers supplied by the power board.

3.4.3 Batteries

The battery system has four battery bays comprised of four Ocean Server BA95HC-FL 14.4V Li-ion batteries, a V-INFINITY VF-S320-18A-CF 18V AC-DC Power Supply, and a Ocean Server XP-04SRW four-channel high current battery controller. This provides the PR2 with approximately two hours of countinous operation after a full recharge. For more information, see the ocean_server package at ros.org.

3.4.4 Power board

The power board is central to the PR2's power system. This is where the power from the batteries passes through to the rest of the system. The power board monitors total power drawn from the batteries by measuring the voltage and current at the power board input.

The battery power is used to provide regulated twelve volts to the computers and the system bus through DC converters. Three unregulated circuits are also provided with a unique electronic circuit breaker functionality. All outputs are ultimately controlled by the processor located on this board.

Here is an image of the power board, with e-stop receiver, removed from a robot.

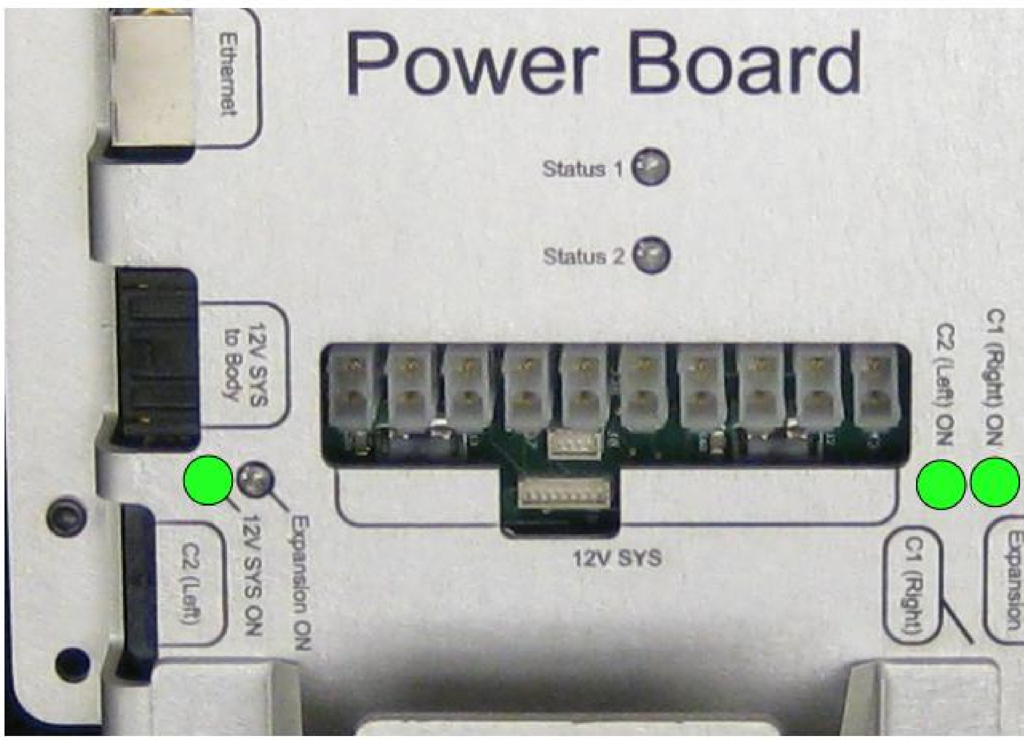
The power board contains an ARM processor with an ethernet interface. The processor is responsible for performing a self test at power-on, monitoring the state of all busses, providing information about the electrical system and responding to commands. The power board is the first thing to turn on and can not be turned off without switching off the DC circuit to the whole robot.

Besides supplying power the power board also interfaces with the emergency stop radio receiver and the emergency stop button. The emergency stop system is designed to reduce the power on the motor bus to a level where only the board communications and their encoder positions are maintained. The red button and radio system are described as providing a physical run/stop control over the motors in the robot. This description is used because this mechanism is implemented in hardware and can not be altered by software.

Please note that the run/stop system only reduces power to the motor power bus and nothing else. If there is an electrical problem or power needs to be shutdown for service, the red circuit breaker on the rear lower panel is the only way to totally shut off the power board.

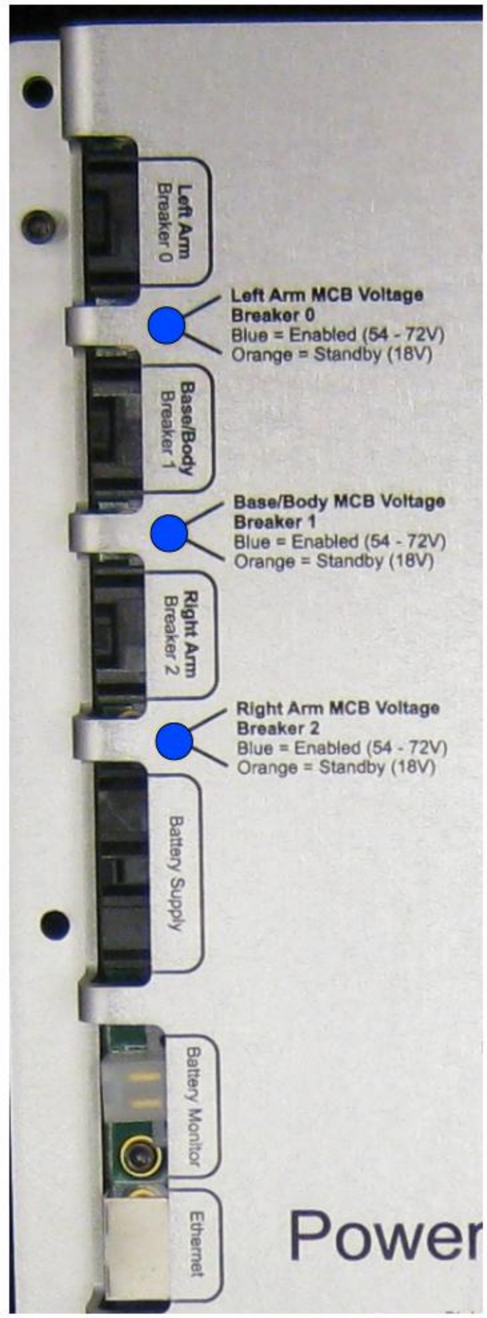
The power board contains many LEDs used to indicate status. I will show their locations and briefly describe their meanings.

The LEDs in green indicate the 3 regulated 12V outputs are enabled and



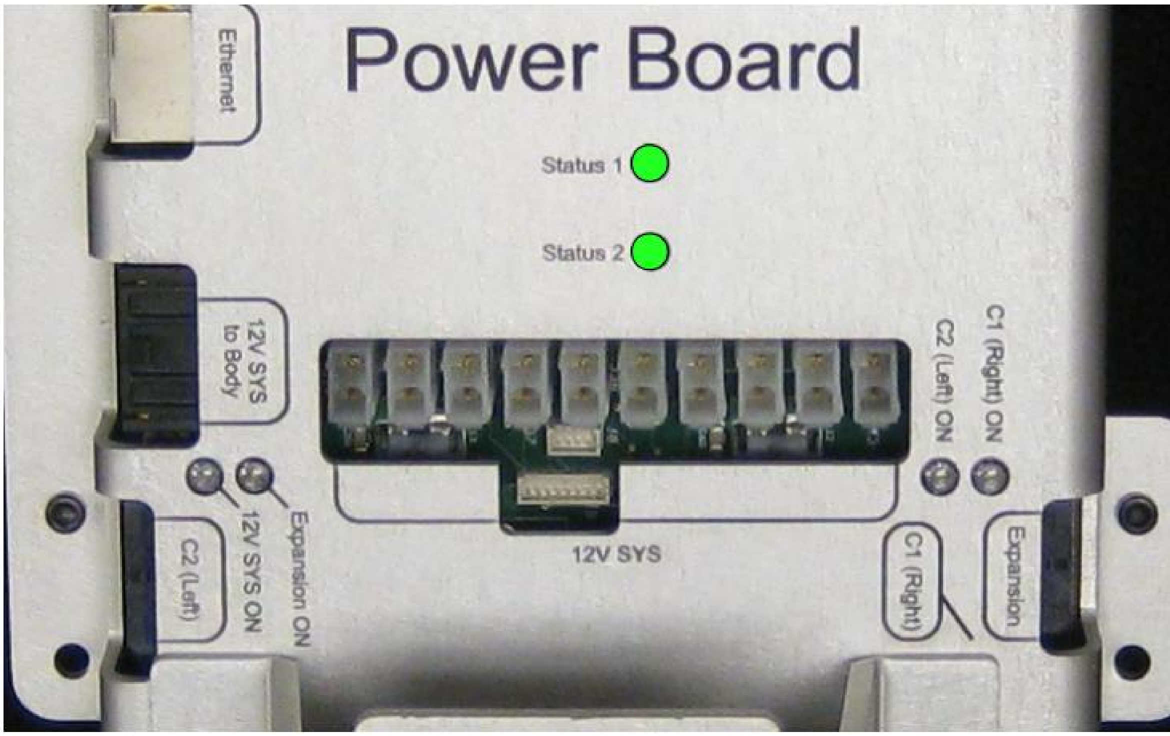
functional.

The Blue LEDs indicate the motor bus power is enabled. Blue indicates a voltage higher than the 18V standby. These LEDs can also be orange if the



output state is in the standby state.

The status LEDs are used by the processor for debugging. In the event of a failure a status code will display on these LEDs. Someone must count the number of rapid blinks followed by a slight pause. Status1 indicates the location where the fault occured and Status2 the cause of the fault.



The following table provides some meaning for the error codes. For further diagnosis please provide the status codes in bug reports.

count	location
1	Self Test
2	Main Loop
3	Breaker 0
4	Breaker 1
5	Breaker 2
6	Calibration Test

Main Loop Message	
count	message
1	Under Voltage Lockout
2	Over Temperature Lockout
3	Under Current condition
4	Over Current condition
5	Failed Self Test
6	18v Failure
7	Fan tachometer failure
8	calibration CRC failure
9	Battery voltage low
Circuit Message	
1	Disable via software command
2	Failed during turn on cycle
3	Shutdown by E-stop
4	Circuit breaker tripped

3.4.5 Power for sensors

As mentioned previously, the power board provides a *12V system bus*, that provides a well regulated 12V supply. The regulated 12V supply is the only supply designed for additions and expansion. This 12V supply is essentially always on once the Power board completes a self test. It is also monitored for undervoltage conditions and is fairly robust.

A few limitations should be followed when adding loads to the 12V user supply. This supply is already supporting the 2 Hokuyo scanners, 2 ethernet switches, the wireless AP and the E-stop receiver. Considering the loads and the supplies capabilities with some headroom this leaves 5A for expansion.

5A limit on 12V expansion

Access to the 12V accessory bus is available in several locations:

Inside the top head assembly.

insert picture

In the upper back.

insert picture

In the lower back, directly connecting to the power board.

insert picture

Before something is connected to this bus, strongly consider if hot plugging is safe. If you are uncertain, the safest choice is to shut the power system off completely.

All 12V connectors in the robot use the same connector, a 2x1 Molex Mini-Fit Jr. The mating connector part number is **39-01-3022** or from Digikey **WM1021-ND**. Pin one is negative and pin 2 is positive. Pin 2 is the pin closest to the latch.

Chapter 4

PR2 Computers

Covers the configuration of the computers and software that comes installed on the robot

4.1 Computer hardware

PR2 has two computers, each has 24 Gb of random-access memory (RAM), 2 quad-core Nehalem processors, and two attached hard-drives. The motherboards are XXX from Rackable systems. For more information about the computers themselves, please see the users's manual and documentation at XXX (link to Rackable information).

Additionally, the PR2 ships with a basestation computer which facilitates seamless communication with the PR2 when transitioning between wired and wireless networks and additionally serves a role in a number of maintenance tasks.

4.1.1 Computer 1 (Master)

Computer 1 is the computer physically located on the right side of the robot. It is referred to as the master computer because it serves a number of key roles for the computer infrastructure:

- Computer 1 stores the operating system for both computers. Computer 2 cannot boot unless computer 1 has booted first.

- Computer 1 is connected to the PR2 ethercat network, and is the only computer that can perform motor control.

- Computer 1 provides routing for the rest of the robot when it is plugged in via the WAN port .
- Computer 1 provides routing for the rest of the robot when connected to another network via an openVPN tunnel
- Computer 1 provides DHCP services for other devices connected to the robot internal network.

Computer 1's PCI slot is used for a 4-port ethernet card, giving it a total of 6 ethernet ports. These are:

- `lan0 - lan4`: connection to internal robot network
- `wan0`: connected directly to WAN port on back of robot
- `ecat0`: connected to robot ethercat network

4.1.2 Computer 2 (Slave)

Computer 2 is the computer physically located on the left side of the robot. It is referred to as the slave computer because it netboots from computer 1.

Computer 2 only has 2 ethernet ports, `lan0` and `lan1`, both of which are connected to the internal robot network.

4.1.3 Basestation

The basestation is a zareason xPC. It is intended to be a dedicated point of contact through which network traffic to the robot can be routed. It has 2 ethernet ports, `eth0`, on the motherboard, is the primary ethernet port. It is intended to be plugged into your local network. `eth1`, on the pci card is a dedicated port for servicing the robot. When necessary, this port should be plugged directly into the robot service port.

When configured properly, it is important that the basestation is visible on port `1194` both via your local wired network as well as via your local wireless network. This will likely require assistance from your local network administrator.

4.2 Networking

The majority of communication between components onboard the PR2 happens via an onboard ethernet network. This onboard network can be accessed directly via either wired or wireless connections. Additionally, the robot can be accessed via the basestation through a VPN tunnel.

4.2.1 Network segments

- `10.68.0.0`: Robot Internal Network 1. The `10.68.0.0` subnet is the primary network used internally by the computers on the robot. Both computers have addresses on this subnet. Many of the ethernet-based devices, such as the power board, are given addresses on this subnet. The robot Service Port, and the cradlepoint ctr350 Wireless Access Point are directly connected to this network. The master computer, if booted, will give out IP addresses via DHCP in the range `10.68.0.100-10.68.0.199`. Important addresses on this network:
  - `10.68.0.1` - c1 (master computer)
  - `10.68.0.2` - c2 (slave computer)
  - `10.68.0.5` - wifi-router (linksys router)
  - `10.68.0.6` - reserved for basestation on local network
  - `10.68.0.91` - c1-esms (master computer enterprise server management system)
  - `10.68.0.92` - c2-esms (master computer enterprise server management system)
  - `10.68.0.250` - wap (wireless access point)
- `10.69.0.0`: Robot Internal Network 2. The `10.69.0.0` network is a second subnet internal to the robot. The cameras are given addresses on this subnet to partition traffic accross the two network interfaces of the computers. This way heavy network utilization by the cameras is less likely to interfere with more critical computer functions such as NFS.
- `10.68.X.0`: Robot VPN Network. The primarily role of the basestation is to function as a VPN server for the robot. Each robot can be given a unique VPN subnet to facilitate the operation of multiple robots using a single VPN server. Under normal operation the basestation will automatically forward relevant traffic into the VPN network, hiding this from the end-user. However, if greater security is desired, the basestation can instead be configured to require users to be assigned a vpn key to access the robot VPN network. Important addresses on this network:

- `10.68.X.1` - c1 (master computer)
- `10.68.X.2` - c2 (slave computer)
- `10.68.255.1` - basestation

4.2.2 Service Port

The robot service port is the bottom ethernet port on the back panel of the robot. It connects directly to the robot internel network, allowing you to connect to the computers directly. *DO NOT PLUG THIS PORT INTO YOUR LOCAL NETWORK.* The master computer serves DHCP for this network and if it conflicts with another DHCP server this will most likely cause problems both on your local network and on the robot network depending on which DHCP server takes precedence.

4.2.3 Wireless access point

The robot comes configured with a cradlepoint ctr350 configured as a wireless access point . The ESSID of this network defaults to PRLAN, and allows direct access to the Robot Internal Network.

4.2.4 WAN port

The robot WAN port is the top ethernet port on the back panel of the robot. This connects directly to `wan0` on the master computer. This port is intended to be plugged into your local network. The robot will attempt to acquire an IP address via DHCP, and then attempt to contact the basestation at a known IP address.

4.2.5 Wireless router

The PR2 has a Linksys WRT610N dual-N band wireless router. This router can be configured to connect to your local wireless network. In the absence of a WAN connection, the robot will attempt to contact the basestation through the wireless router instead.

4.3 OS

The operating system running on the PR2 computers is an extended version of Ubuntu 9.04 (Jaunty Jackalope). It depends on a number of additional packages for system configuration, but should otherwise be familiar to for

Ubuntu users. If you run into a computer problem not covered by the PR2 documentation, the Ubuntu Documentation is the next place to look.

4.3.1 NFS and Unionfs

The single largest difference between a normal Ubuntu installation and the PR2 configuration is that the Slave computer mounts nearly its entire filesystem via NFS. The exceptions to this are the directories `/etc`, `/var`, and `/pr2bin` which are mounted via unionfs-fuse. In short, unionfs allows one to specify an additional overlay on top of the underlying filesystem. The contents of this overlay can be found in the directory `/slave` on the master machine. Files added here will show up in the appropriate location on the slave machine. For more information on how this is set up, see the man page for `unionfs-fuse` and look at the init-script `unionfs-fuse-nfs-root`.

New software and configuration changes should only be made on the master machine. Since the slave machine mounts most of its filesystems read-only, this will usually be enforced for you. If you are using a non-standard piece of software that attempts to write to the filesystem outside of your home directory you will likely need to make accommodating changes to either the computer configuration or the software you are trying to run.

4.3.2 autofs

Both of the computers have automount configured for mounting the `/home` partition of the other computer in a computer-independent way using autofs. These automounts are located in the directory `/pr`. To get to the home partition on computer 1 you can use the path `/pr/1/` and to get to the home partition on computer 2 you can use the path `/pr/2/`. You should rarely need to do this explicitly, but it is necessary to make sense of home directory locations.

4.3.3 Home directories

The default configuration for user home directories (as given in `/etc/passwd` is `/u/username`. Instead of a directory, this location is by default a symlink to `/pr/1/username`. To place a users home directory at a different location, such as the disk on the slave computer, an admin can simply move (or copy) the home directory and update the symlink accordingly.

4.3.4 Kernel

Discuss RT_PREEMT and anything else non-vanilla about the kernel. (BLAISE HAS AGREED TO FILL THIS IN).

4.3.5 Storage

Each computer has two hard-drives - one internal 2.5" 500Gb Seagate Momentus drive, and a slot for a removable 3.5" SATA hard-drive that is exposed on the top of the robot base.

By default the only hard-drive used is the internal drive on the Master computer. There are 3 relevant partitions created during installation:

- `c1:/dev/sda1` – `/` – holds the root filesystem for the OS
- `c1:/dev/sda5` – `c1:/home` – stores user home directories (linked to by `/u` by way of `/pr/1`)
- `c1:/dev/sda6` – `/hwlog` – stores hardware logs generated by the pr2

The internal hard-drive on the Slave computer has a single partition, which is generally used as extra user storage:

- `c2:/dev/sda1` – `c2:/home` – stores user home directories (linked to by `/u` by way of `/pr/2`)

Finally, both computers are configured to make use of the additional removable drives. Any drive loaded into the removable bay will always show up as `/dev/removable`.

- `/dev/removable1` – `/removable` – stores temporary data users may want to move off the robot, primarily used for large bag files. This NOT mounted by default. Users must explicitly

  ```
  $ mount /dev/removable
  ```

  to use it, and should

  ```
  $ umount /dev/removable
  ```

  when done.

Formatting Drives

If you want to partion / reformat one of the removable drives, or the the internal drive on the slave machine, yuo can use the `formatdisk` helper utility. Simply pass it the name of the drive you want to format, for example:

```
$ sudo formatdisk /dev/sda
```

4.3.6 User accounts

The expectation is that each person logging into the robot to develop code will have their own user account. Any robot admin can create a new user using the `adduser` command (NOT the `useradd` command).

Some examples:

```
$ sudo adduser bob
$ sudo adduser bill --shell /usr/bin/tcsh --uid 2000
```

Note: it may be helpful to assign users the same UID used on your local network so the UIDs are consistent when mounting shares or moving around the removable drives.

Moving the home directory and creating the symlink is handled automatically by the script: `/usr/local/sbin/adduser.local`.

4.3.7 User groups

There are a couple of important groups on the pr2:

- `admin` – Members of this group have full root privileges when using the sudo command
- `rosadmin` – Members of this group have access to change pr2-specific configuration settings
- `apt` – Members of this group can install new software

To add a user to a group, use the `usermod` command. The most common invocation is `-a` (append) `-G` (group). For example:

```
$ sudo usermod -a -G admin bob
```

4.3.8 Backing up and restoring users

Before reinstalling the robot operating system you will likely want to back up the user accounts. This can be done with the command: `pr2-usermigrate`. To save users:

```
$ sudo pr2-usermigrate save myrobot.users
```

Move this file off the robot before reinstalling. Then, to restore users:

```
$ sudo pr2-usermigrate load myrobot.users
```

4.3.9 Clock synchronization

Consistent time-stamping of data from the two computers is important for interpreting sensor data on a moving system. As a result, keeping the system time on the two computers synced together requires some attention. The system that is used for this is chrony, and the general strategy is to have the two computers tightly coupled to one another, but loosely coupled to an external time source to prevent the robot time from drifting too far from the outside world.

4.4 Basestation Setup and Pairing

Before you can use the robot and basestation, it needs to be configured. This configuration can be done by logging into the basestation. The default account is "pr2admin" with password "willow". The following instructions assume that the basestation is plugged into the robot via the service port.

4.4.1 Requirements

For the basestation to work properly, it needs to be given a static IP. This IP can be given to the basestation either statically or assigned via a DHCP server, but it needs to be constant. the IP address you assign it must be visible on both the wired and wireless networks on port `1194`. This is the port that VPN uses when the robot makes its connection back to the basestation. Depending on how secure your network is, you may need assistance from your local sysadmin to configure this properly.

Additionally, for the robot to be able to find the basestation, both the wired and wireless networks must provide a DHCP server. The operation of the system is that the robot will acquire an IP address via DHCP and then contact the basetation at the known IP address that it is paired to.

4.4.2 Initialize the Basestation VPN Server

One of the primary functions of the basestation is to host a VPN server that the robot can connect to. The server needs an appropriate key. To generate the key for the VPN server run:

```
$ sudo /etc/openvpn/gen_server_key
```

It will prompt you for some information. It is good practice to get the information correct, but will not actually impact the performance of the robot.

4.4.3 Configure the Basestation Network

There are two ways to set up the basestation, using DHCP or static IP.

DHCP

If you are using DHCP, you must make sure that your DHCP server respects the client-id specification and is configured to assign a consistant IP address to the client-id of the basestation. The default client-id for the basestation is "basestation." If you need to change this, you must edit the file `/etc/dhcp3/dhclient.conf`, and change the line:

```
send dhcp-client-identifier "basestation";
```

To use a different client-id of your choice.

Static IP

To configure the basestation with a static IP, you must edit the file `/etc/network/interfaces` and change eth0 to use a staic ip address, for example:

```
iface eth0 inet static
      address 192.168.1.100
      netmask 255.255.255.0
      gateway 192.168.1.1
      post-up robot_forward start
      pre-down robot_forward stop
```

You will also need to update `/etc/resolv.conf` to contain the appropriate nameserver, e.g.,

```
domain school.edu
search school.edu
nameserver 192.168.1.1
```

Hostname

If you want to change the hostname of the robot to something other than "basestation," you will need to edit the files: `/etc/hostname`, and `/etc/hosts`, replacing instances of "basestation" with your new name as appropriate.

Applying Settings

Once you have reconfigured the network, you should run:

```
/etc/init.d/networking restart
```

for the changes to take effect.

4.4.4 Configure the Robot Wifi

The Linksys wrt610n on the robot is located at the IP address: `10.68.0.5`. To configure it, open a web-browser on the robot and go to: `http://10.68.0.5`. Click on the "Wireless" tab. The default login should be "root" and "willow," although after a factory reset the router may end up with login, "root" and "admin". Make sure one of the radios is in client mode and enter the SSID for your network. If you need to change other settings, make sure that under "Setup," the WAN connection type is "Automatic Configuration - DHCP", the local IP Address is `10.68.0.5`, and the DHCP Server is Disabled.

To test the setup of the wireless, you can temporarily add a route on the basestation which will route a particular IP address through the wifi router.

```
$ sudo ip route add 157.22.19.22 via 10.68.0.5 dev eth1
```

Then point your web-browser to: `http://157.22.19.22`. This should take you to the willowgarage home page. Once that is working, you should remember to disable the route so that it works if the robot is not plugged into the service port.

```
$ sudo ip route del 157.22.19.22 via 10.68.0.5 dev eth1
```

4.4.5 Pairing with the Robot

Robot pairing is performed with the the `robot_setup` command on the basestation. It takes 3 arguments:

- name – This is the name your robot will be given. The Master will be "name1" and the slave will be "name2".

- VPN subnet – Unless you are using multiple robots, this should be 10.68.1.0. The 3rd field of the IP, however, can be any value other than 0 and 255, which are reserved for the robot local network and the basestation server respectively.

- Basestation IP – This is the IP address where the robot can reach the basestation.

```
$ sudo robot-brand <ROBOT_NAME> <C1_NAME> <C2_NAME> <VPN_SUBNET> <BASESTATION_IP>
```

For example, if your robot were going to be named "prx", c1 was going to named "prx1", c2 was going to be named "prx2", and your basestation was located at `192.168.1.100`, you would run:

```
$ sudo robot-brand prx prx1 prx2 10.68.1.0 192.168.1.100
```

During the pairing, you will be given the option to reformat the slave disk and removable drives. If you choose yes to this, all data will be removed.

When the script finishes, you should be able to connect to the robot over the VPN:

```
$ ssh pr2admin@10.68.1.1
```

4.4.6 Forwarding IPs to the robot

To make contacting the robot from your local network more convenient, the basestation can be configured to forward IP addresses (assigned statically or via DHCP) to the robot.

To set this up, edit the file: `/etc/robot_forward.conf` and add the lines:

```
<NAME>1 10.68.1.1 <ROBOT_IP1>
<NAME>2 10.68.1.2 <ROBOT_IP2>
```

For example

```
prx1 10.68.1.1 192.168.1.101
prx2 10.68.1.2 192.168.1.102
```

Alternatively, you can also have these IPs assigned via DHCP:

```
prx1 10.68.1.1 dhcp
prx2 10.68.1.2 dhcp
```

If this is the case the IP addresses will be acquired using the client-id's "prx1" and "prx2" respectively.

Robot forwarding is enabled whenever the networking starts, but to manually start forwarding, just run:

```
sudo robot-forward start
```

4.5 Software Maintenance

When the robots ship, the most recent version of ROS and the PR2 software stacks is installed in /opt/ros. By default, new users will have a ROS installation which references the pre-installed code. Users who wish to modify or replace a part of the system are recommended to install a development version of the stacks or packages which they wish to modify, but to continue to use the base installation for most of the robot functionality.

Debian packages exist for all of the custom software and configurations on the robot. The package which pulls in all the other requirements is called `pr2-environment`. `pr2-environment` depends on `pr2-core`, which in turn depends on `ros-boxturtle-pr2`, the meta-package which contains all of the ROS software for the PR2.

To upgrade the software on your robot, you can simply use `apt`:

```
$ sudo apt-get dist-upgrade
```

Which will update all packages on your system, pulling in security fixes for the pr2 configuration as well as for the appropriate ros-release.

4.6 Reinstall the Robot

To reinstall the robot, first back up anything you need off of the master machine. Next, shutdown the comptuers using `pr2-shutdown --netboot`.

You must then plug the service port directly into the basestation. You can then run the robot installer:

```
$ sudo robot-installer
```

Boot the Master computer by pushing the power button with a paperclip.

After N minutes, the netconsole on the basestation will show the isntall dialog. Hit enter to start the installer and wait for it to complete.

4.6.1 Forcing netboot

If the robot BIOS is not configured to netboot, you will need to be able hit f12 on the keyboard. You can do this in one of two ways.

The easier way to to connect up to the KVM. (See section XYZ on connecting to the KVM.)

Alternatively, plug a keyboard into the PR2. This can be done using the aux usb port under the back cover. To remove this cover, follow the instructions in section XYZ.

When the robot is booting you will hear 2 beeps followed by a short pause, and then another 2 beeps. After the 4th beep, hit f12 on the keyboard (either physically or through the KVM) to force the Master computer to netboot.

Chapter 5

Setting up PR2 in your lab

Taking the robot out of the crate and setting it up for the first time involves XXX.

5.1 The accessory kit

What to do with each item in the accessory kit

5.2 Taking PR2 out of the crate

Physically taking it out and setting it up to charge (charge while in the crate?)

5.3 Setting up PR2 for your network

Setting up the base-station, setting up DHCP and host-names, setting up the wireless network to allow base-station access.

5.4 Creating initial user accounts

What accounts exist initially. Changing the password. Who should get root access.

If you have done all the above steps properly, you're ready to start running and developing on the PR2.

Chapter 6

Running the PR2

Running the PR2 requires a basic understanding of ROS (http://www.ros.org), the BSD-licensed Robot Operating System. A ROS system consists of multiple processes running on multiple computers. If you are not familiar with ROS, it is highly recommended that you follow some of the beginner tutorials on ros.org. Familiarity with ROS tools will make using the robot much easier. In particular, you should understand what a launch file is and how to run it, and how to run ROS with nodes on multiple computers. This chapter will walk you through starting up and running a PR2, using ROS.

6.1 Getting set up

6.1.1 Out of the box

If you are starting your PR2 for the first time at your institution, please read the previous chapter, **??**. This will give you advice on setting up the network, setting the administrative password, and picking a safe location for charging the PR2. This chapter assumes that the PR2 is already set up for your lab.

6.1.2 Batteries and power

Before running the robot, you need to make sure it has power, which is easiest to do by making sure the robot is plugged in to a wall outlet. You can follow the instructions in this chapter to start up the PR2 while it is plugged into the power outlet, which will keep the batteries charged. The battery life of the PR2 is approximately two hours, so it is a good idea to keep the PR2 plugged in when not in use.

6.1.3 Run-stop

Before running the robot, you should also have the wireless run-stop nearby (Figure 6.1), so that you can shut down the motors if you need to.

The PR2 has two run-stop buttons: the red push-button on the middle of the PR2's back, and the yellow wireless run-stop transmitter. If either one is in the stop position, the motors are disabled so the robot cannot move. **In an emergency, press the stop button on the wireless transmitter, or hit the push-button in the back**. Putting either run-stop button in the stop position does not damage the robot or turn off the computers, it simply stops the motors.

Figure 6.1: The PR2 wireless run-stop.

The wireless run-stop is also powered by batteries, which can run down. Therefore, it is a good idea anyway to turn the wireless run-stop off (by pressing the "STOP" button) when not in use. If the wireless run-stop runs out of battery charge, then the battery light (the light next to the battery symbol in the lower half of the wireless run-stop face) will flash. To change the battery, you will need a slotted screwdriver to open the wireless run-stop case (Figure 6.2).

6.1.4 Getting an account

Before using the PR2, you will also need an account on the robot computers. Ask the robot administrator to create an account for you, using the instructions in section 4.3.6.

Figure 6.2: Opening the PR2 wireless run-stop to change the batteries.

6.2 Turning on PR2

If the robot is already running, you can skip this step, but you should still put the wireless run-stop in the stop position.

To turn the PR2 on, you should first verify that the wireless run-stop is in the stop position. If it is not yet in the stop position, then press the wireless run-stop's red "STOP" button; the lights on the wireless run-stop should turn off.

Second, switch the red run-stop button on the back panel of the robot to the "on" position; turn it clockwise and it will pop out.

Third, if the DC breaker switch at the back panel of the robot is switched off, then flip the DC breaker switch (Figure 6.3) to the "on" position.

Figure 6.3: The PR2 DC breaker switch in the ON position.

You should see the red lights on the computers turn on, hear the fans spin up, and hear several rising-tone beeps from the computers as they boot. The process of booting the computers will take about 5 minutes. You will know that the computers are on when you see a red light illuminate on each of the computers in the robot base.

6.2.1 Logging in

Once the robot is on, ssh into the main computer using the account created for you by the administrator, using the steps below.

In the following tutorial, we will use pr<x>1 to refer to the first computer on robot <x> and <username> to refer to your account username. Ask your robot administrator what name you should use for the robot you want to run.

To log in to the robot, open a terminal window and type

```
$ ssh <username>@pr<x>1
```

You should already have access to all the ROS tools. For instance, to see what packages are available to you, type

```
$ rospack list-names
```

and you should see a list of the ros packages currently in your path.

6.2.2 Checking for other users

If you logged in to a robot that was already running, you should check to see if anyone else is using the robot. To find out who else is using the robot, type

```
$ ckill list
```

This will show you what programs are currently running on the robot. If nothing shows up, then no one else is logged into the robot. If other people have processes running on the robot, you will need to find out if you can interrupt their work. You should find out who is on the robot and ask them to allow you to kill their processes so that you can work on the robot. If you cannot find the people running processes on the robot, ask the robot adminstrator for guidance on what the policy is in your lab. If it is fine for you to kill the other processes running on the robot, you can type

```
$ sudo robot stop
```

and all the processes that are being run on the robot will be killed.

6.3 Starting the software

The launch file /etc/ros/robot.launch is used to start up the basic functionality of the robot. This includes drivers for the sensors, motors, speakers, projector, power system, and joystick as well as the default set of realtime controllers, processing and logging of diagnostics information, and monitors for various types of problems your robot could experience. On a new robot, this launch file is standard, but if the robot you are working on has been altered (e.g., has additional sensors, has only one arm), then it is likely that someone administering the robot has also updated the /etc/ros/robot.launch file to work with your robot's configuration.

There are two ways to start the software on the PR2. The first way is easier than the second way.

6.3.1 Starting the software the easier way

Once you are logged in and ready to start the robot, you can run the robot by typing

```
$ sudo robot start
```

Running robot start will start the roslaunch for you in the background as a system user. This will allow you to continue to use your current terminal and the robot will keep running after you log out.

To see what topics are being published in ROS now, you can type

```
$ rostopic list
```

This is a good way to check to see if you started up ROS successfully.

6.3.2 Starting the software the manual way

As an alternative to starting the software the easier way, you can also start the software manually. If you prefer to see everything that is going on, you can also start the robot software manually, but you will need to keep that window open until you are done using the robot. Closing that the terminal in which you roslaunched will terminate the roslaunch.

```
$ roslaunch /etc/ros/robot.launch
```

to start up the pr2.

If you used "roslaunch" to start the robot, then you will need to open a new terminal and type

```
$ ssh <username>@pr<x>1
$ rostopic list
```

to see what topics are being published by the system in the new terminal while the roslaunch continues to run in the original terminal. Remember that <x> is a placeholding variable to be replaced with the ID of the robot you are using.

Either way, the robot should be running, but since you have the wireless run-stop in the stop position, the motors will not be moving.

6.4 Running the dashboard

When running the robot, the pr2_dashboard should always be up on your screen. This is how the robot software will let you know if something is going wrong, and is also how you turn the motors and power on and off.

On a computer with a built ROS installation (e.g., the base-station desktop computer that ships with the robot), not on the robot itself, set your ROS_MASTER_URI to point at the master running on the robot and launch the dashboard by typing

```
$ export ROS_MASTER_URI=http://pr<x>1:11311
$ rosrun pr2_dashboard pr2_dashboard
```

You should see the pr2_dashboard control panel (a graphical user interface) appear and provide you with information about the state of the robot. It is ok if not all of the icons are green. In fact, the run-stops should be "warning" you that the wireless runstop is not on.[1]

Take a moment to review the state of the robot. You can get a sense for the health of your robot by looking at the diagnostics information; click on the wrench on the far left. Since you have the run-stop in the stop position, you see that the motors are giving you a warning because the robot was just turned on and the encoders on the joints have not been calibrated yet.

If you see warnings or errors in any other sections, you should read the error messages and try to figure out what the problem is. Ask your robot administrator for help if there are errors that you do not understand.

[1] If you do not see the pr2_dashboard control panel appear, then there is a chance that you have not yet built the pr2_dashboard. To do this, type `rosmake pr2_dashboard` to build pr2_dashboard. Once this is done, you can try launching the dashboard again.

6.4.1 Understanding pr2_dashboard

When running the PR2, the most important piece of software for you to understand and control the state of the system is the pr2_dashboard. pr2_dashboard, Figure 6.4, is a GUI for debugging and controlling low-level state of the PR2. The dashboard displays the diagnostic, circuit breaker, run-stop, and battery status. The robot will not usually start up with all green lights. In fact,

Figure 6.4: The PR2 dashboard when starting up the robot.

if you bring up the robot properly, then you should see a dashboard display that looks more like Figure 6.4. This is because the wireless run-stop should be in the stop position. You can tell that the wireless run-stop is in the stop position by mousing over the dashboard to see the tooltips, which describe what each icon means. See Figure 6.5.

Figure 6.5: The PR2 dashboard when starting up the robot, showing tooltips.

The following sections describe the other information displayed in the dashboard. Some of the indicators in the dashboard can have a status of "stale." This means that some component of the robot has stopped reporting status information. A stale status is usually caused by a network problem (e.g., bad wireless connection from the base station to the robot).

Diagnostic Status The state of the robot is shown by the diagnostic indicators in the pr2_dashboard.

Component	OK	Warn	Error	Stale
Diagnostics: Clicking pops up the Robot Monitor				
Rosout: Clicking pops up rxconsole				
Motors: Clicking allows you to halt or reset motors		N/A		

Circuit Breaker Status The circuit breakers are labeled L/B/R, which stand for Left Arm, Base/Spine, and Right Arm. Each breaker can be in one of four states, and clicking on any of the breakers will pop up a menu (Figure 6.6), allowing you to change the state of one or all of them:

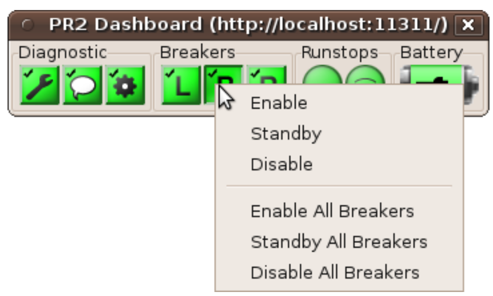


Figure 6.6: The circuit breaker options menu.

	Enabled	Standby	Disabled	Stale
Breaker Status	L B R	L B R	L B R	L B R

Run-stop Status The runstop indicators display the status of the PR2 runstops and their states cannot be changed using the dashboard. There are two run-stops on the robot, a wireless run-stop, see 3.1.2, and a red push button run-stop on the back of the robot.

	OK	Physical Stop is Off	Wireless Stop is Off
Runstop Status			

Battery Status The battery will change its color and % filled based on the amount of battery remaining. It will also show a power-plug symbol if the robot is charging.

	> 50%	30 - 50%	< 30%	Charging
Battery Status				

6.5 Calibrating the joints

If you do not discover any problems in the diagnostics, and you want to continue with robot bringup, first check the physical area around the robot to ensure that the robot will not hit anything around it when it calibrates. The robot's arms will be fully extended and the spine will move up and down during calibration. Once the physical area is cleared, press the green "START" button on the wireless run-stop. This should change the state of the run-stop indicator on the dashboard to a green circle, and will allow the robot to turn power on for the motors.

If the robot circuit breakers are not yet enabled, the next step is to enable the circuit breakers (marked L, B, R). When you click on the circuit-breaker icon, select "enable all" from the menu to turn on all three breakers.

If the robot motors are not yet reset, then click on the picture of a motor, and select "reset motors."

When you have started the wireless run-stop, enabled all the robot circuit breakers, and have reset thr boto motors, the robot will move its joints to find the absolute reference positions of each joint so that it can calibrate the mechanism. When calibration is finished, you should see all the icons in the dashboard reading OK (Figure 6.7).

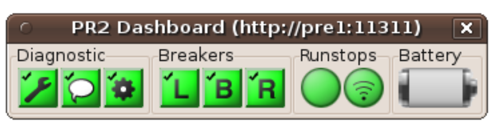


Figure 6.7: The PR2 dashboard when the wireless run-stop is in the start position.

If this calibration step fails, then you can use the pr2_dashboard to diagnose the problems with the robot. See your robot administrator if you need assistance in understanding and addressesing the issues.

6.6 Tucking the robot arms

Before driving a robot around, it is best to tuck the robot arms. If you do not tuck the robot arms before driving, then the arms are likely to swing around while the robot is moving.

To run the tuckarm package, type into the robot computer (pr<x>1):

```
$ rosrun pr2_tuckarm tuckarm.py b
```

This will tuck both robot arms. The feedback from tuckarm should display something like this:

```
[INFO] 1264099125.710039: Waiting for controller manager to start
[INFO] 1264099125.718283: Tucking both left and right arm
```

For more information about how to drive the robot around, see the tuckarm instructions at ros.org.

6.7 Driving with the joystick

To move the robot, you can now drive it around with the joystick. To do this, you will need to open a new terminal window with a new ssh connection to the robot and run the teleop_joystick launch file

```
$ ssh <username>@pr<x>1
$ roslaunch pr2_teleop teleop_joystick.launch
```

Be sure to check to see if the robot is plugged in (e.g., ethernet cables or power cables)! If it is plugged in, then be very cautious to avoid pulling the cables out improperly.

Once the teleop_joystick node is running, press the pairing button (Figure 6.8) in the middle of the joystick to pair it with the robot. Be sure to unplug the robot from wall outlets if you are going to drive the robot around.

Figure 6.8: Pairing the joystick with the PR2.

Then to control the robot, use L1 (Figure 6.9) as the "deadman" switch. L1 should be held down with your left pointer finger (like a trigger) whenever you wish to control the robot. If you let go of the L1 button, then commands will no longer be sent from the controller to the robot. While pressing L1, use the other buttons for the following movements:

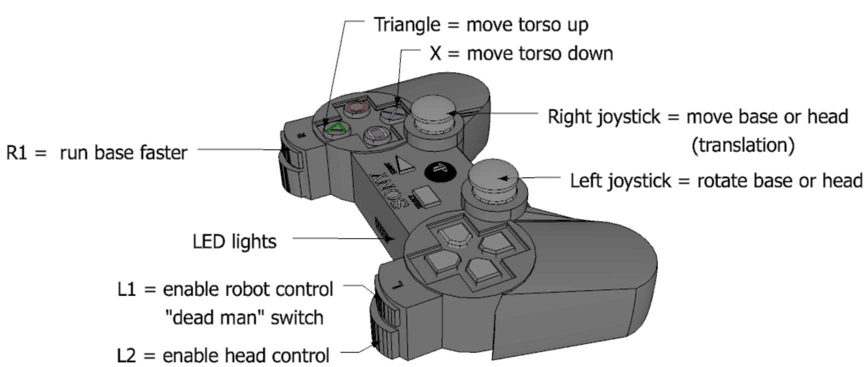


Figure 6.9: The PS3 controller mapped to robot movements.

1. L1 (left pointer finger): The "dead man" switch allows the controller to send commands to the robot
2. L2 (right pointer finger): The head switch allows the joysticks to con-

trol the head instead of the base

3. Right joystick (right thumb): Translates the base or head (i.e., front, back, left, right)

4. Left joystick (left thumb): Rotates the base or head (i.e., spin clockwise or counter-clockwise)

5. R1 (left pointer finger): The “run” switch commands the base to move faster

6. Triangle and X (right thumb): Moves the torso up and down

To drive the robot around (i.e., driving the base), hold down L1 and use the two joysticks to control its direction of movement (right joystick) and rotation (left joystick). To drive the robot around at faster speeds, also hold down R1.

To move the robot head, hold down L1 and use the two joysticks to control the head movements.

To move the robot torso, hold down L1 and press the triangle button to move the torso up or the X button to move the torso down. The torso moves slowy.

For more information about how to drive the robot around, see the ps3joy tutorial at ros.org.

6.8 Visualizing sensor data

To see the sensor data from the robot, you can use rviz on the offboard computer (e.g., base station). Again, you will need to open a new terminal window and configure the ROS_MASTER_URI to point at the robot, so run

```
$ export ROS_MASTER_URI=http://pr<x>1:11311
$ rosrun rviz rviz
```

and you should see rviz launch with a visualization of the robot.[2]

For more information about how to view different types of data coming from the robot, see the rviz documentation at ros.org. (TODO - WITH THE CURRENT LAUNCH FILE, YOU WILL NOT BE ABLE TO SEE CAMERA DATA)

[2] If you do not see rviz appear, then there is a chance that you have not yet built rviz yet. To do this, type `rosmake rviz` to build rviz. Once this is done, you can try launching rviz again.

6.9 What next?

From here, you can do what you want on the robot. (NOTE TO AUTHORS: Point to instructions for writing code and to a list of other applications and things to do with the robot.)

6.10 Putting Away the PR2

To properly put away the PR2, you should drive the robot to the appropriate location in your lab, plug in the robot to recharge the batteries, press STOP on the wireless run-stop, stop the processes on the robot, and log off of the robot.

First, put away the robot. Drive the robot to the appropriate location in your lab and plug it in to recharge.

Second, stop all the processes running on the robot. If you want to stop all of the processes running on the robot and log off of the robot, but not turn off the robot computers, then open a new terminal and type:

```
$ ssh <username>@pr<x>1
$ sudo robot stop
```

This logs you into the robot and stops the robot processes.

As an alternative to simply stopping the proceeses running on the robot, you can also turn off the robot computers. To do this, open a new terminal and type:

```
$ ssh <username>@pr<x>1
$ sudo pr2-shutdown
```

If you turn off the robot, you will hear four sets of descending beeps from the robot computers. Then the red lights on each of the computers will turn off. If this does not happen, then the robot computers might not be completely shut down properly.

Finally, to exit from the robot computer terminal, type

```
$ exit
```

If someone was waiting to use the robot after you, be sure to let them know that you are done with the robot.

Chapter 7

Writing code on PR2

The PR2 comes with the PR2 varient of the latest released distribution or ROS installed from binary debian packages in `/opt/ros` . It is recommended to use this installation for it saves compile time and disk space.

Start by getting rosinstall There is more documentation on the wiki package page rosinstall.

```
wget --no-check-certificate http://ros.org/rosinstall -O ~/rosinstall
chmod 755 ~/rosinstall
```

7.1 Installing code in user-space

To setup your environment with the installed packages type:

```
. /opt/ros/boxturtle/setup.sh
```

7.1.1 Installing another repository in user-space

To add another repository an overlay rosinstall file should be created like the following:

```
- VERSION-CONTROL-TYPE:
    uri: URI
    local-name: WORKING-COPY-NAME
```

Recreate this as /custom.rosinstall substituting appropriately for the all caps secionds.

Then type:

```
~/rosinstall -o ~/local_dir ~/custom.rosinstall
```

Setting environment to use local_dir

```
. ~/local_dir/setup.sh
```

7.1.2 Creating a package

This is documented on the wiki in the roscreate package

7.1.3 Using a local version of a stack

If a different version of a stack is desired instead of an installed one, add it to your overlay rosinstall file. Stacks in overlays are prepended to the path and thus will take priority over an existing stack. Note be careful, if changing a lower level stack, for all things which depend on it's changes must be rebuilt. This can be an issue if the dependent stacks are also installed.

7.2 Where to Start

7.2.1 Documetation and Tutorials

Documentation for ROS and the PR2 can be found on ros.org. An overview of the relevant libraries for filtering, navigation, cooridinate tranforms, etc., can be foung at http://www.ros.org/APIs. Below is a listing of several tutorials for learning and using ROS:

ROS stack tutorials These tutorials cover the basic ROS concepts and tools for writing and using ROS nodes.

tf tutorials These tutorials cover using the tf library for cooridinate transofrms using the python and C++ APIs.

navigation tutorials These tutorials cover configuring and using the navigation stack on the PR2 and other robots.

laser_pipeline tutorials These tutorials cover processing laser data and converting laser data into 3D representations.

image_common tutorials These tutorials cover working with images in ROS.

actionlib tutorials These tutorials cover implementing actions, a standard interface preemptible highlevel tasks, using the python and C++ APIs. More example tutorials can be found at turtle actionlib.

7.2.2 Mailing Lists

ros-users This mailing list provides the latest ROS news and is a forum for posting questions to the ROS community for help.

PR2-users This mailing list provides the latest PR2 news and is a forum for posting questions to the PR2 community for help.

Chapter 8

Maintaining PR2

8.1 Installing upgraded stacks

Instructions on using the pr2_admin user to install new version of stacks and apps next to what is already on the system.

Instructions on how users set up their accounts to be either using the latest code all the time or to be locked to a particular version. Discussion of the advantages of each.

8.2 Installing updated disk images

Instructions on installing an updated disk image from WG on the PR2 for, e.g. OS changes or robot configuration updates. We need to figure out what is maintained during updates (e.g. network configuration?), and what is over-written. This needs to be crisp and well-documented.

8.3 Resetting PR2

Sometimes, you may want to reset PR2 to factory defaults. This is the process. This can either be done while preserving user data, or can be done as a full factory reset.

Essentially, this is a full update to the last-installed disk image.

8.4 Diagnostics Logs

Since the PR2 is currently in beta, it is important for us to learn as much as we can about robot hardware usage and how problems develop. To address this, we are requiring all PR2 beta sites to regularly transfer diagnostics logs to Willow Garage.

These logs are gathered automatically by pr2_core in the /hwlogs partition. This directory should not be used for anything else, since everything there will be periodically transfered to Willow Garage.

8.4.1 Setting up automatic transfer

To make this process easier, we have provided a set of scripts to perform this log transfer automatically. To enable those, just add the line XXX to the pr2_admin crontab via these instructions:

8.4.2 Manual transfer

If you would prefer to not have the robot transfer logs back to Willow Garage automatically, you can transfer them manually instead. To do this, either run the script "transfer_logs", or look inside it to understand what it's doing and perform the transfer yourself.

8.5 Solving hardware problems

8.5.1 Figuring out what's wrong

The first step if you suspect hardware problems should be to check the diagnostics display in pr2_dashboard. If that doesn't help you resolve the problem, you can run the full robot self test. This is a GUI which allows you to select the portions of the robot you wish to qualify, and runs them through a variety of tests to diagnose problems. Instructions on using it and interpreting the data it outputs go here.

8.5.2 Asking Willow Garage for help

For hardware issues, please send email either to pr2_users or to the maintnenance list at XXX.

8.5.3 Documentation

For more detailed documentation, see the hardware maintenance manual at XXX. This includes drawings and step-by-step instructions for replacing all user-servicable parts, as well as guidelines and checklists you can use to figure out if what you're seeing is normal for the system. After this list of chapters, we should have a set of other tutorials which have been referenced from the text. Only tutorials that are on the critical path should be in-lined in the text. Everything else should be referenced and then collected at the end.