Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

SumoBot® – Mini-Sumo Robotics

Assembly Documentation and Programming

VERSION 2.1

PARALLAX

WARRANTY

Parallax Inc. warrants its products against defects in materials and workmanship for a period of 90 days from receipt of product. If you discover a defect, Parallax Inc. will, at its option, repair or replace the merchandise, or refund the purchase price. Before returning the product to Parallax, call for a Return Merchandise Authorization (RMA) number. Write the RMA number on the outside of the box used to return the merchandise to Parallax. Please enclose the following along with the returned merchandise: your name, telephone number, shipping address, and a description of the problem. Parallax will return your product or its replacement using the same shipping method used to ship the product to Parallax.

14-DAY MONEY BACK GUARANTEE

If, within 14 days of having received your product, you find that it does not suit your needs, you may return it for a full refund. Parallax Inc. will refund the purchase price of the product, excluding shipping/handling costs. This guarantee is void if the product has been altered or damaged. See the Warranty section above for instructions on returning a product to Parallax.

COPYRIGHTS AND TRADEMARKS

ISBN 1-928982-26-3

DISCLAIMER OF LIABILITY

Parallax Inc. is not responsible for special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, or any costs of recovering, reprogramming, or reproducing any data stored in or used with Parallax products. Parallax Inc. is also not responsible for any personal damage, including that to life and health, resulting from use of any of our products. You take full responsibility for your BASIC Stamp application, no matter how life-threatening it may be.

INTERNET DISCUSSION LISTS

We maintain active web-based discussion forums for people interested in Parallax products. These lists are accessible from www.parallax.com via the Support → Discussion Forums menu. These are the forums that we operate from our web site:

- BASIC Stamps – This list is widely utilized by engineers, hobbyists and students who share their BASIC Stamp projects and ask questions.
- Stamps in Class® – Created for educators and students, subscribers discuss the use of the Stamps in Class educational program in their courses. The list provides an opportunity for both students and educators to ask questions and get answers.
- Parallax Educators –Exclusively for educators and those who contribute to the development of Stamps in Class. Parallax created this group to obtain feedback on our curricula and to provide a forum for educators to develop and obtain Teacher's Guides.
- Translators – The purpose of this list is to provide a conduit between Parallax and those who translate our documentation to languages other than English. Parallax provides editable Word documents to our translating partners and attempts to time the translations to coordinate with our publications.
- Robotics – Designed exclusively for Parallax robots, this forum is intended to be an open dialogue for a robotics enthusiasts. Topics include assembly, source code, expansion, and manual updates. The Boe-Bot®, Toddler®, SumoBot®, HexCrawler and QuadCrawler robots are discussed here.
- SX Microcontrollers and SX-Key – Discussion of programming the SX microcontroller with Parallax assembly language SX-Key® tools and 3rd party BASIC and C compilers.
- Javelin Stamp – Discussion of application and design using the Javelin Stamp, a Parallax module that is programmed using a subset of Sun Microsystems' Java® programming language.

ERRATA

While great effort is made to assure the accuracy of our texts, errors may still exist. If you find an error, please let us know by sending an email to editor@parallax.com. We continually strive to improve all of our educational materials and documentation, and frequently revise our texts. Occasionally, an errata sheet with a list of known errors and corrections for a given text will be posted to our web site, www.parallax.com. Please check the individual product page's free downloads for an errata file.

Table of Contents

PREFACE

Like its human counterpart, robot Sumo was born and thrives in Japan. It was introduced to the United States in the early 1990's by Dr. Mato Hattori. One of the early American adopters of robot Sumo was noted Seattle Robotics Society member, Bill Harrison, who organized some of the first U.S. robot Sumo tournaments.

While things started out very slowly, robot Sumo eventually caught on. Bill created a "lightweight" class that matched the Japanese physical dimensions of 20 cm by 20 cm, but reduced the mass from three kilograms (6.6 pounds) to one kilogram (2.2 pounds). The intention was to reduce the sophistication of the components required to construct a working Sumo robot. Those early contests didn't have much in the way of corporate support with prizes, so Bill resorted to offering 30 hours of his own machine-shop services to the winner.

As luck would have it, Bill's friend Robert Jorgensen won that first contest prize. Since Robert already had a winning Sumo robot, he suggested that they build a smaller version, about half the size and weight of the lightweight class to be used as a robot Sumo demonstrator. The result of their work was a very small Sumo robot that measured just 8 cm by 8 cm and mass about 240 grams. Bill took that first small Sumo to a contest in San Francisco and actually won the lightweight competition – against bigger and heavier robots. The Mini-Sumo robot class was born.

The Mini-Sumo dimensions (10 cm x 10 cm) and mass (500 grams) were formalized and Bill published adapted Japanese robot Sumo rules on his Sine Robotics web site (mirrored on many other sites, and reprinted with permission in this document). Through Bill's tireless efforts and nearly ten years of travel – often toting more than 20 robots in his bags – Mini-Sumo robotics has grown to a favorite activity among robot clubs all across the United States.

RECOGNITIONS

Many Mini-Sumo designs – especially the dual-wheel-and-scoop concept – can be traced back to Bill Harrison's early efforts to promote Mini-Sumo robotics competition. Parallax also recognizes Bill Boyer of the Dallas Personal Robotics Group for his version of the dual-wheel-and-scoop design that was refined and developed into the Parallax SumoBot robot described in this text.

This text was authored by Jon Williams of Parallax, and contains additional material by several contributors, including Andy Lindsay and Ken Gracey of Parallax, as well as Bill Wong of Pennsylvania. Bill is an editor with *Electronic Design* magazine and a serious BASIC Stamp® robotics enthusiast. Bill enjoys creating BASIC Stamp powered robots with his daughter, who has gone on to win several county and state awards with her maze solving robotics projects.

AUDIENCE

SumoBot was written for ages 12+ as a complimentary text to Parallax's *Robotics with the Boe-Bot* and *Advanced Robotics with the Toddler* student guides. Like all Parallax texts, this series of experiments teaches new techniques and circuits with minimal overlap between the other publications. The general topics introduced in this series are: basic SumoBot locomotion under program control, edge avoidance, and opponent detection based on a variety of sensor inputs, as well as navigation opponent hunting using programmed artificial intelligence. Each topic is addressed in an introductory format designed to impart a conceptual understanding along with some hands-on experience. Those who intend to delve further into industrial technology, electronics or robotics are likely to benefit significantly from initial experiences with these topics.

If your experience with the SumoBot® robot differs from our expectations, please let us know at support@parallax.com.

EDUCATIONAL CONCEPTS FROM THE SUMOBOT

Educators frequently ask us at Parallax what can be learned from our different texts and application notes. The SumoBot is considered an intermediate robotic project and generally will instruct the following concepts:

- Interaction between mechanical and electrical systems, and the ability to tune hardware or adjust software to obtain desired results.
- Intermediate programming skills with the BASIC Stamp 2 microcontroller. An efficient SumoBot program makes use of efficient BASIC Stamp programming techniques with **`BRANCH`** and **`LOOKDOWN`**, variable aliasing, general sound programming practices (constant/variable definitions that allow for program customization in just a few places rather than throughout an entire program).
- A step-wise process which starts with the basics and builds to something more complex and ultimately more useful.

Chapter 1: Assemble the SumoBot

There's an old axiom among robot enthusiasts that states, "It's harder than it looks...." Speaking from experience, we know this to be true. That said, the purpose of this statement is not to alarm or dissuade the new robot builder, but simply to remind him or her that robotics – even on a small scale – is a serious endeavor and shouldn't be taken lightly. Patience is indeed a virtue. Follow the construction steps carefully and you'll have your SumoBot running and ready to compete in about an hour or so.

The SumoBot is capable of doing any of the things other rolling robots can do. As you learn to program the SumoBot for competition, you'll become a more proficient – and *efficient* – programmer and will learn to exploit the BASIC Stamp microcontroller's capabilities. The SumoBot demonstrates the importance of a PBASIC program that uses constants and variables, as well state-oriented design. A well-designed program means you can easily tune the software for the right mechanical control in just a few places rather than rewriting your entire program.

A surface-mounted BASIC Stamp 2 microcontroller provides the intelligence for the SumoBot. The BASIC Stamp is used throughout the *Stamps in Class* educational series, and provides plenty of program space, speed and memory for use with a SumoBot.

The SumoBot is a purpose-built rolling robot, much like its general-purpose cousin the Parallax Boe-Bot. While they share the same differential drive mechanism and the use of sensors, the SumoBot design meets the specific criteria defined by Mini-Sumo competition rules:

- Maximum [width and depth] dimensions of 10 cm by 10 cm
- Maximum mass of 500 grams

The standard SumoBot comes with two sets of sensors: two QTI line sensors to keep the SumoBot on the playing surface and two sets of infrared emitters/detectors used to locate its opponent. Advanced users may expand on the standard SumoBot design by adding ultrasonic or IR distance measuring, tilt sensing and motor current sensing.

LET'S BUILD THE SUMOBOT

The SumoBot chassis design leaves little room for mechanical alteration; a requirement to stay within standard Mini-Sumo competition rules. Where the student is encouraged to explore changes is in the types of sensors used to detect the Sumo ring border and the opponent and the software algorithms used to control the SumoBot robot's behaviors. The demonstration code provided with this text will focus on the standard sensors provided in the SumoBot kit. Future supplements may be published that deal with advanced sensors and techniques for incorporating them into the SumoBot robot's control logic.

TOOLS REQUIRED

A Parallax screwdriver is included in your kit. You may find a pair of needle-nose pliers and a wire stripper to be useful (not included).

ABOUT PARTS IN THE SUMOBOT KIT

Appendix A includes a parts listing for the SumoBot robot kit. These instructions refer to different pieces of hardware. If your SumoBot kit is missing a piece, Parallax will replace it free of charge. Replacement Parallax Continuous Rotation servos and infrared emitters and detectors are available to purchase online from the Parallax Component Shop (www.parallax.com → Component Shop). If you need other parts replaced, please contact sales@parallax.com or call toll free in the United States: 1-800-512-1024.

If you have trouble identifying the type of part referred to in these instructions, see the color back cover of this text that shows each part with a colored picture and Parallax stock code.

Step #1
Install the Battery Box

Parts Required:

- Battery Box
- (2) 4/40 3/8" long flat-head countersunk machine screws
- (2) 4/40 nuts
- SumoBot chassis

Stand the SumoBot on its PCB mounting ears. Install the plastic battery pack using two 4/40 3/8" flat-head screws and nuts. The screws will be countersunk into the battery pack when tightened and should be out of the way of the batteries.

Step #2
Install the Servo Motors

Parts Required:

- (2) Parallax Continuous Rotation Servos
- (8) 4/40 3/8" long pan-head machine screws
- (8) 4/40 nuts
- SumoBot chassis

Using four 4/40 3/8" pan-head machine screws and 4/40 nuts, attach each servo motor to the chassis. The easiest way to do this is to hold the nut with one finger while turning the screwdriver with the other hand.

Step #3
Install the Rear SumoBot PCB Stand-offs

Parts Required:

- (2) 5/8" round standoffs
- (2) 4/40 3/8" long pan-head machine screws
- SumoBot chassis

Using a 4/40 3/8" pan-head machine screw, attach each stand-off to the rear of the SumoBot chassis.

Step #4
Install the Front SumoBot PCB Stand-offs

Parts Required:

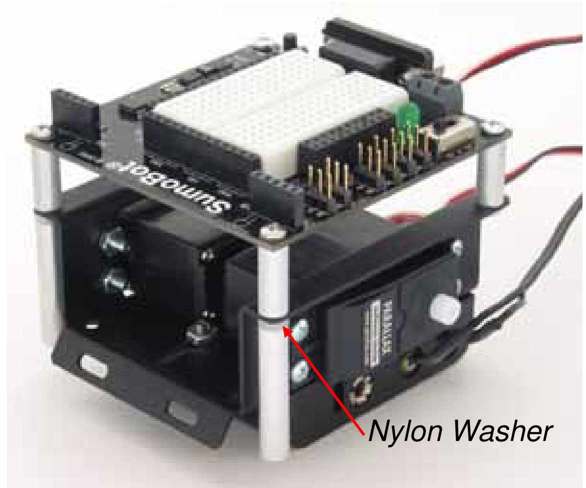
- (2) 5/8" round standoffs
- (2) 4/40 1" long pan-head machine screws
- SumoBot PCB

Using a 4/40 1" pan-head machine screw, attach each standoff to the front mounting holes of the SumoBot PCB.

Step #5
Mounting the PCB

Parts Required:

- SumoBot PCB
- (2) 4/40 3/8" long pan-head machine screws
- (2) 1-1/4" round stand-offs
- (2) Nylon washers
- SumoBot chassis



Feed the ends of the 1" long pan-head machine screws through the front mounting holes on the SumoBot chassis. Secure the rear side of the SumoBot PCB to the 5/8" standoffs with two 3/8" pan-head machine screws. Holding the chassis upside-down, place a nylon washer onto the end of each 1" long pan-head machine screw, then secure by threading on the 1-1/4" round standoff.

Step #6
Prepare the Wheels

Parts Required:

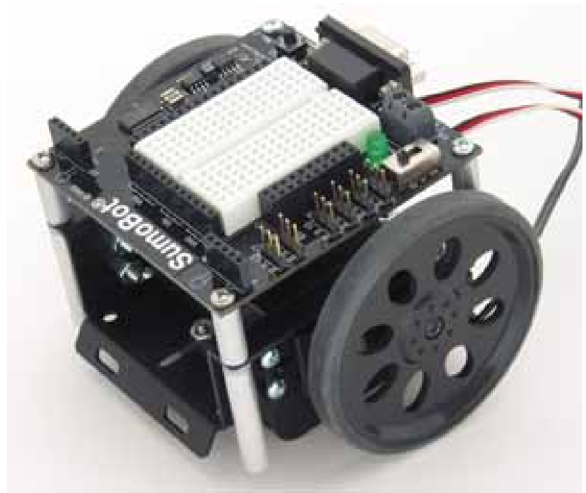
- (2) SumoBot wheels
- (2) SumoBot rubber tires

Stretch a "tire" of each wheel and adjust so that the "tire" is centered across the wheel.

Step #7
Mount the Wheels

Parts Required:

- (2) Prepared wheels/tires
- (2) Black servo-horn screws
- SumoBot chassis

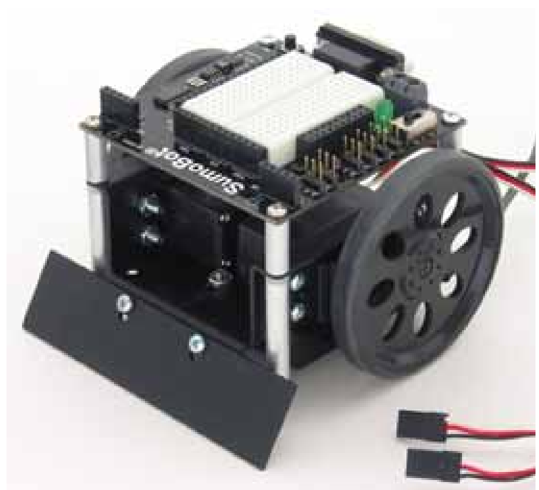


Carefully press each prepared wheel onto the servo splines. Secure each wheel with the small black Phillips head screw.

Step #8
Mount the Scoop

Parts Required:

- SumoBot scoop
- (2) 4/40 1/4" long pan-head machine screws
- (2) 4/40 nuts
- SumoBot chassis



Using two 4/40 1/4” pan-head machine screws and 4/40 nuts, attach the scoop to the SumoBot chassis. Carefully center the scoop before tightening the screws and nuts.

Step #9
Install Line Sensor Wires

Parts Required:

- (2) 10" 3-pin extension cables
- SumoBot chassis

Carefully feed each 10" 3-pin extension cable through the center chassis slot.

Step #10
Install the QTI Line Sensors

Parts Required:

- (2) QTI line sensors
- (2) 4/40 1/4" long pan-head machine screws
- SumoBot chassis

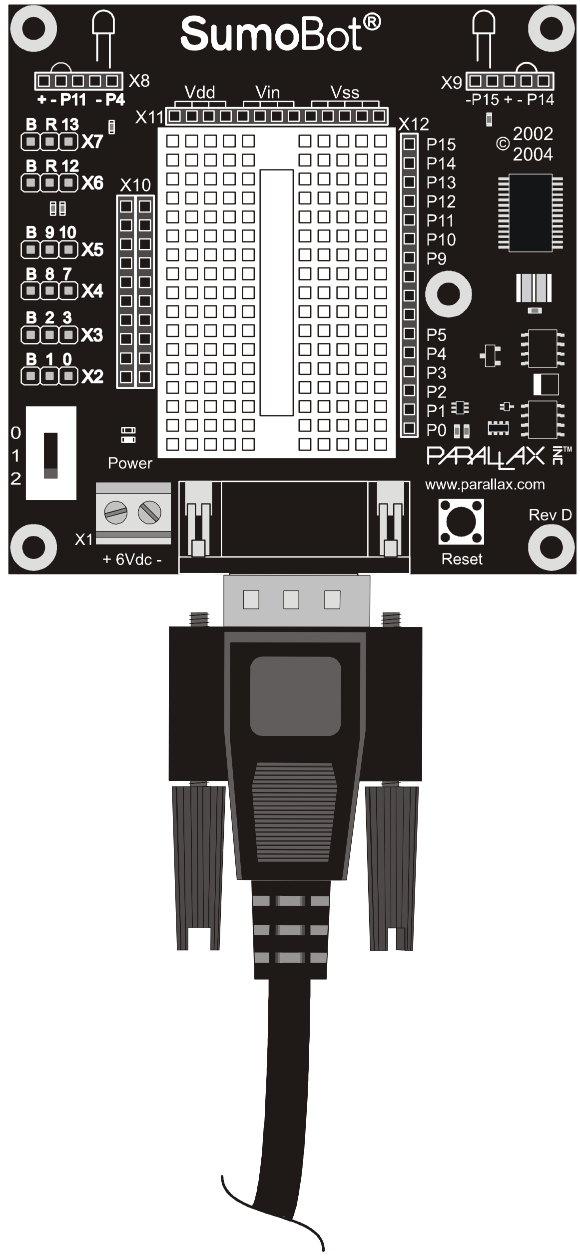
Using two 4/40 1/4" pan-head machine screws, attach the QTI line sensors to the 1-1/4" round stand-offs. Connect the ends of the 10" 3-pin extension cables to the QTI line sensors, noting the polarity markings B[lack]-R[ed]-W[hite] on the QTI sensors.

Step #11
Make the Connections

Plug the servo motors and QTI sensors into the SumoBot PCB connectors as indicated below. Note that the "B" pin on each connector is for the black wire.

X7 = Left Servo Motor
X6 = Right Servo Motor
X5 = Left QTI Line Sensor
X4 = Right QTI Line Sensor

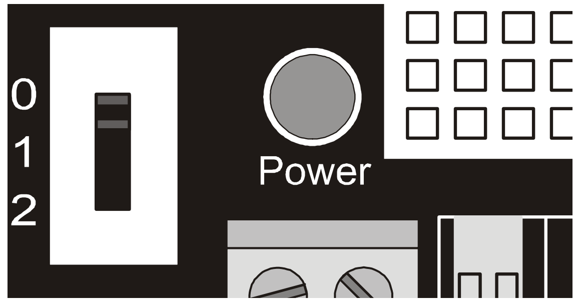
Connect the battery pack wires to SumoBot PCB connector X1. The battery pack's white-striped lead connects to the **+** terminal.



Note: Previous versions of the SumoBot PCB were labeled "SumoBoard" instead of "SumoBot." These boards are electrically identical to the SumoBot PCB illustrated.

When using SumoBot PCBs with a revision code of C or earlier, the Vs1 and Vs2 (servo ground) connections must be jumpered to Vss for proper servo operation.

Step #12
Powering the SumoBot



The SumoBot PCB has a three-position power switch. The state of each position is shown below. The three-position switch has a middle position that powers the entire circuit except the servos. A complete schematic of the SumoBot PCB is included in Appendix D.

Position 0 – No Power
Position 1 – Power PCB
Position 2 – Power PCB & Servos

Chapter 2: SumoBot Locomotion

The first task of any Mini-Sumo robot is to move – most competition rules do not allow the robot to stop (without competitor contact) for more than a few seconds. In this experiment you will learn how to get the SumoBot moving and learn to take control over its motion.

HOW A SERVO WORKS

Normal (un-modified) hobby servos are very popular for controlling the steering systems in radio-controlled cars, boats and planes. These servos are designed to control the position of something such as a steering flap on a radio-controlled airplane. Their range of motion is typically 90° to 270°, and they are great for applications where inexpensive, accurate high-torque positioning motion is required. The position of these servos is controlled by an electronic signal called a pulse train, which you'll get some first hand experience with shortly. An un-modified hobby servo has built-in mechanical stoppers to prevent it from turning beyond its 90° or 270° range of motion. It also has internal mechanical linkages for position feedback so that the electronic circuit that controls the DC motor inside the servo knows where to turn to in response to a pulse train.

SumoBot motion is controlled using two pre-modified Parallax Continuous Rotation servo motors using a process called differential drive. The modification "tricks" the feedback circuitry so that the servo will stop only when it receives a centering command; it also allows the servo to continuously rotate in either direction. When both motors are turning in the same direction, the SumoBot will move in that direction. When the SumoBot servo motors turn in different directions, the chassis will rotate. The rate of movement or rotation is determined by motor speeds.

TIME MEASUREMENTS AND VOLTAGE LEVELS

Throughout this text, amounts of time will be referred to in units of seconds (s), milliseconds (ms), and microseconds (µs). Seconds are abbreviated with the lower-case letter "s". So, one second is written as 1 s. Milliseconds are abbreviated as ms, and it means one one-thousandth of a second. One microsecond is one one-millionth of a second. Figure 2.1 shows how Milliseconds and Microseconds equate in terms of both fractions and scientific notation.

Figure 2.1: Milliseconds and Microseconds Details

$$1\,\text{ms} = \frac{1}{1000}\,\text{s} = 1 \times 10^{-3}\,s$$

$$1\,\mu\text{s} = \frac{1}{1{,}000{,}000}\,\text{s} = 1 \times 10^{-6}\,s$$

A voltage level is measured in volts, which is abbreviated with an upper case V. The SumoBot PCB has sockets labeled Vss, Vdd, and Vin. Vss is called the system ground or reference voltage. When the battery pack is plugged in, Vss is connected to its negative terminal. Vin is unregulated 6 volts (from four AA batteries) and it is connected to the positive terminal of the battery pack. Vdd is regulated to 5 volts by the SumoBot PCB's onboard voltage regulator, and it will be used with Vss to supply power to circuits built on the SumoBot PCB's breadboard.

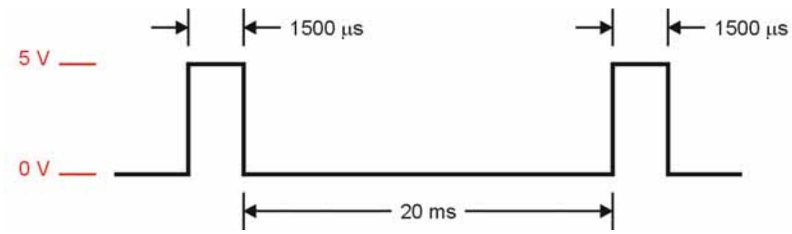
Figure 2.2: SumoBot PCB Voltage Labels

Vss = 0V (ground)
Vdd = 5V (regulated)
Vin = 6V (unregulated)

> **!** Only use the Vdd sockets above the SumoBot PCB's breadboard for the activities in this workbook. Do not use the Vdd on the 20-pin App-Mod header.

The control signal the BASIC Stamp sends to the servo's control line is called a "pulse train," and an example of one is shown in Figure 2.3. The BASIC Stamp can be programmed to produce this waveform using any of its I/O pins. In this example, the BASIC Stamp sends a 1500 µs pulse to P13 (left servo) and P12 (right servo). When the pulse is done being executed the signal pin is low. Then, the BASIC Stamp creates a 20 ms pause.

Figure 2.3: Servo Pulse Train Analysis



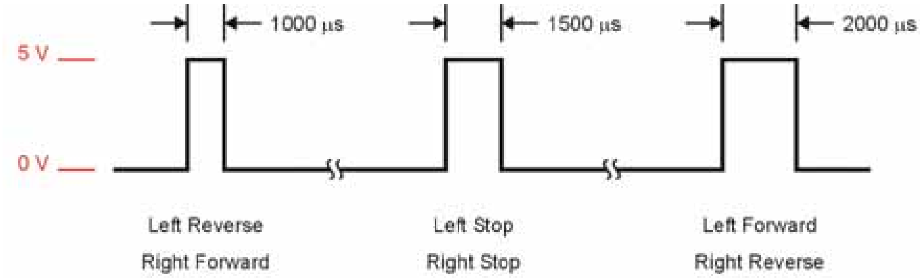
This pulse train has a 1500 µs high time and a 20 ms low time. The high time is the main ingredient for controlling a servo's motion, and it is most commonly referred to as the pulse width. Since these pulses go from low to high (0V to 5V) for a certain amount of time, they are called positive pulses. Negative pulses would involve a resting state that's high with pulses that drop low.

The ideal pause between servo pulses is 20 milliseconds, but can be anything between 10 and 40 milliseconds without adversely affecting the servo's performance.

> The BASIC Stamp 2's **`PULSOUT`** instruction works in increments of 2 micro-seconds. For example, the following snippet of code creates a 1500 µs pulse:
>
> ```
> PULSOUT P13, 750                    ' 1500 us pulse on pin 13
> ```

A pulse width of 1500 µs (normally, the centering command) will cause the modified servo to stop. To make the servo turn we must give change the pulse width toward either end of the standard control range of 1000 to 2000 µs. Since the right side servo motor is physically mirrored from the left, its control signals are as well. Figure 2.3 shows the control signaling for the SumoBot servos.

Figure 2.4: SumoBot Servo Control Pulses



For pulses between the 1500 µs stop point and the extremes on either end of the control range, there is a degree of speed control. This range is not linear, however, and at pulse widths just outside the stop band, servo current increases dramatically. At some points in the control range, the servo current can go high enough to cause an excessive load on the BASIC Stamp's regulator circuitry, causing it to reset or behave erratically. For Mini-Sumo competition, precise speed control is not a requirement. The goal is to find the opponent and move quickly toward him.

Open the BASIC Stamp Windows Editor.[1] Load the following program that will be used to align the SumoBot motors.[2]

---

1 The Parallax BASIC Stamp Manual 2.x includes a "Quick Start" section that details how to open and launch the BASIC Stamp Windows Editor.

2 Source code for this text is available in a zipped file for download from www.parallax.com.

```
' SumoBot_2.1_Motor_Align.BS2
' {$STAMP BS2}
' {$PBASIC 2.5}

' -----[ I/O Definitions ]-------------------------------------------------

LMotor          PIN     13                      ' left servo motor
RMotor          PIN     12                      ' right servo motor

' -----[ Constants ]-------------------------------------------------------

LStop           CON     750                     ' left motor stop
RStop           CON     750                     ' right motor stop

' -----[ Initialization ]--------------------------------------------------

Reset:
  LOW LMotor                                    ' initialize motor outputs
  LOW RMotor

' -----[ Program Code ]----------------------------------------------------

Main:
  DO
    PULSOUT LMotor, LStop                       ' stop left
    PULSOUT RMotor, RStop                       ' stop right
    PAUSE 20
  LOOP

  END
```

Move the SumoBot power switch to position 2, and then download the code using the Run command from the Run menu, or by pressing the ► button on the toolbar. As soon as the program is downloaded, watch for wheel movement. If either motor turns, insert a small screwdriver into the adjustment port of the servo and adjust the centering potentiometer until the motor stops. Figure 2.5 shows the location of the servo adjustment ports.