



# Balancing Bot Tutorial

Last Updated Wednesday, May 30, 2007 by Tony Le.

- **Contents**

- [Overview](#)
- [Parts/Components](#)
- [Configure Hardware](#)
- [Configure Software](#)
- [Test Components](#)
- [Final Test Results](#)
- [Final Thoughts & Useful Links](#)

- **Overview**

*Keywords:* Balancing, Inverted pendulum, ~~Baby Orangutan~~, SparkFun 5DOF

This tutorial will go over my implementation of a balancing bot based on the article from Nuts&Volts magazine titled "Building a Balancing Bot on a Budget" by Phil Davis and Brandon Heller. The major difference in my implementation is the hardware components. As stated in their article, the components that they use are not regularly available. As a result, I chose the following components specifically the Baby Orangutan as the on board controller for the robot. The choice was mainly based on its size and use of the Atmel Atmega168 microcontroller. The B-Bot code that Davis uses an AVR port of the FreeRTOS architecture. To stay in lines with this I chose to use the Atmega 168 vs. the originally specked Atmega32 that was used on the unavailable Brandon board. However, the other key component used that did remain the same is the Sparkfun 5 DOF IMU. Its small size and accuracy level made it a key component and also the most expensive individual component for this robot.

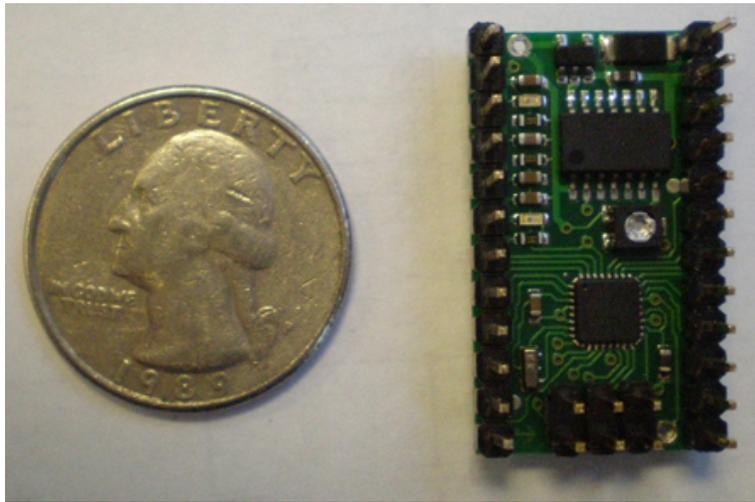
- **Parts/Components**

Table 1. Parts List

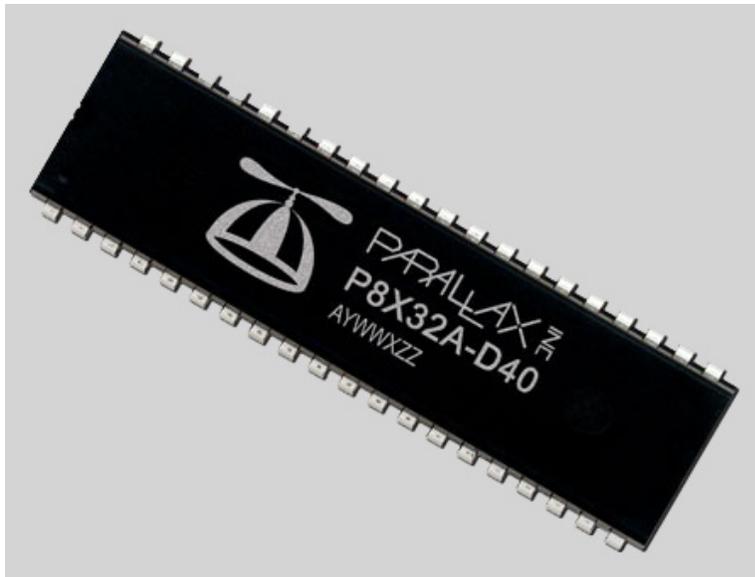
Part Description	Vendor	Part#	Price (2007)
<a href="#">Propellor PropStick</a>	<a href="#">Parallax</a>	P8X32A-D40	\$79.95
<a href="#">Baby Orangutan Controller</a>	<a href="#">Pololu</a>	0216	\$29.95
<a href="#">5DOF IMU</a>	<a href="#">Sparkfun</a>	Sense-5DOF	\$109.95
<a href="#">LK202-25-WB</a>	<a href="#">Matrix Orbital</a>	LK202-25-WB	\$59.95
<a href="#">Continuous Rotation Ball-bearing Servo Motor</a>	<a href="#">Junun</a>	GWS S03N 2BB	\$16.00
<a href="#">SumoBot Wheel &amp; Tire</a>	<a href="#">Parallax</a>	721-00002	\$3.95
<a href="#">Circuit Board</a>	<a href="#">RadioShack</a>	276-150	\$1.79
<a href="#">Atmel AVRISP</a>	<a href="#">DigiKey</a>	ATAVRISP2-ND	<del>\$34.00</del>

- **Configure Hardware**

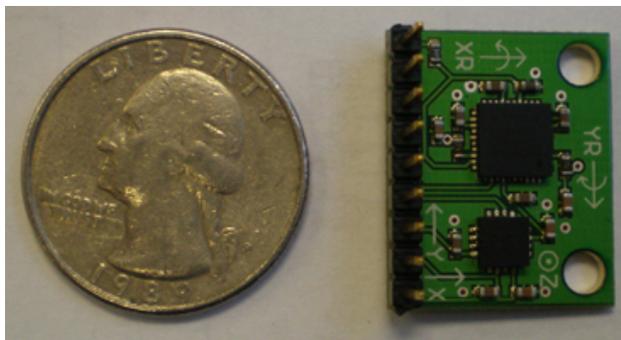
- Required Hardware Components
  - **\*EDIT\*** The Baby Orangutan was not cutting it as the primary controller so a decision was made to switch to the Parallax Propellor microcontroller. ~~Baby Orangutan Robot Controller~~ *Note: When this item was purchased the pin headers were not surface mounted so there will be some light soldering required.*

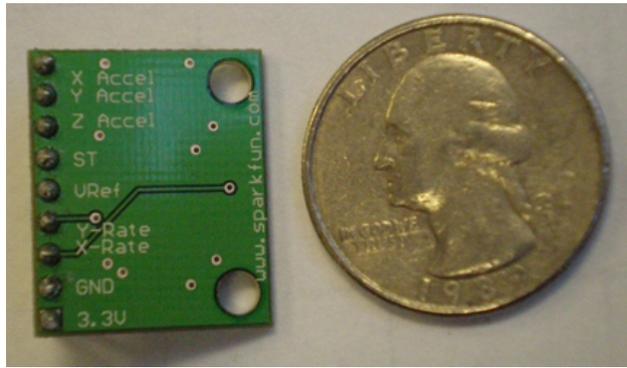


- Parallax Propellor - P8X32A-D40 (DIP) Chip:

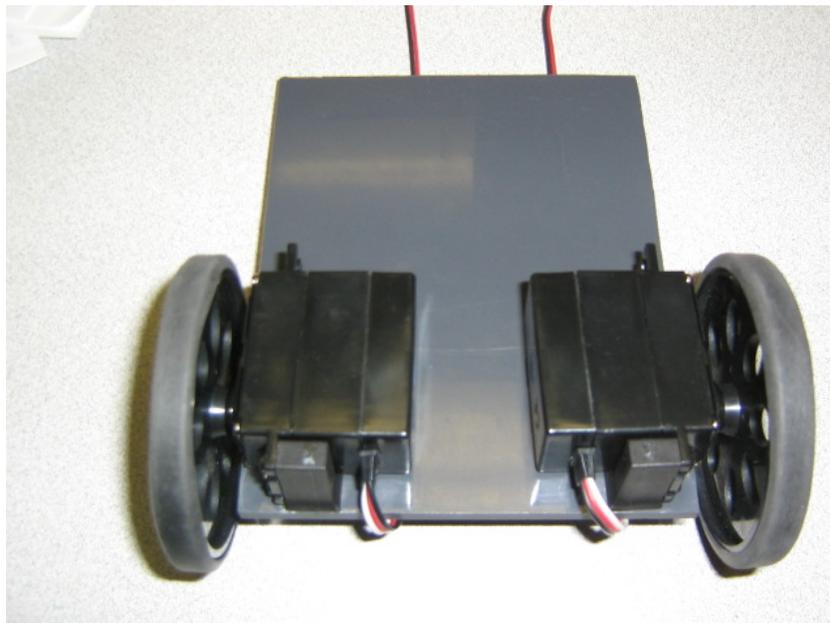


- SparkFun 5DOF IMU - *Note: When this item was purchased the pin headers were not surface mounted so there will be some light soldering required.*



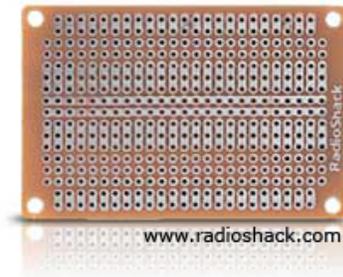


- Chassis, Servo, and Wheels: The chassis can be composed on any sturdy material. Here I chose to use my own make-shift PVC chassis instead of purchasing the laser cut white board that was described in the Nuts&Volts article. I mounted the servos and MarkIII Sumo wheels to the PVC board as seen below.

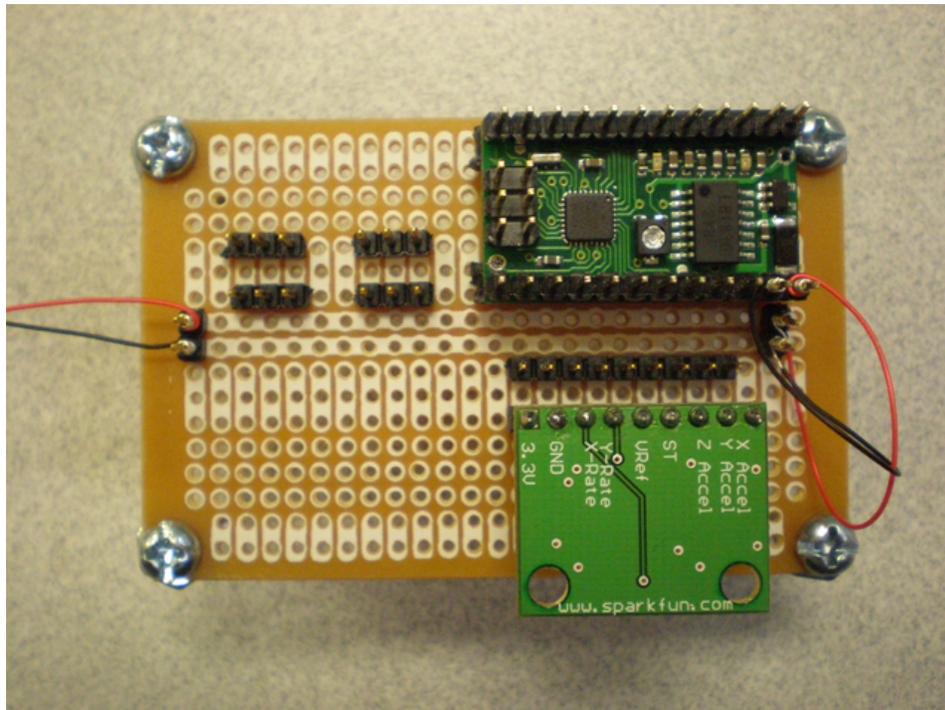


- Construction

- With the servos mounted to the chassis the next step is to add on the other hardware components. I chose to mount the LCD screen last and to the top of the robot. It isn't pictured here because it will be mounted last after programming it to work with the controller. To make mounting the components easier, I used a Radio Shack multipurpose circuit board.

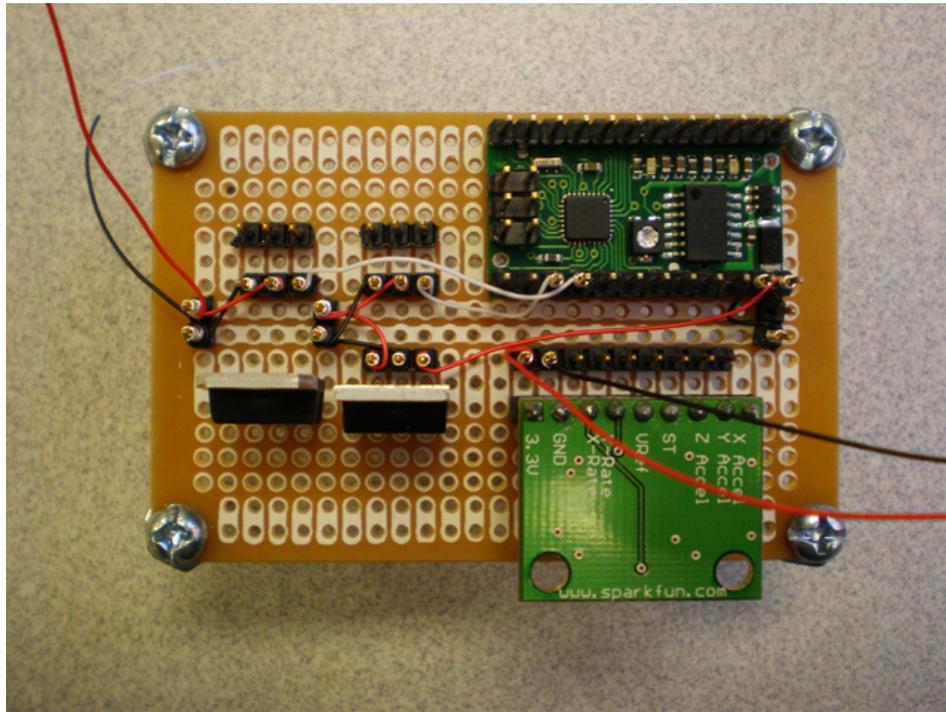


- Using assorted pin headers and extenders I mounted space the controller and the IMU:

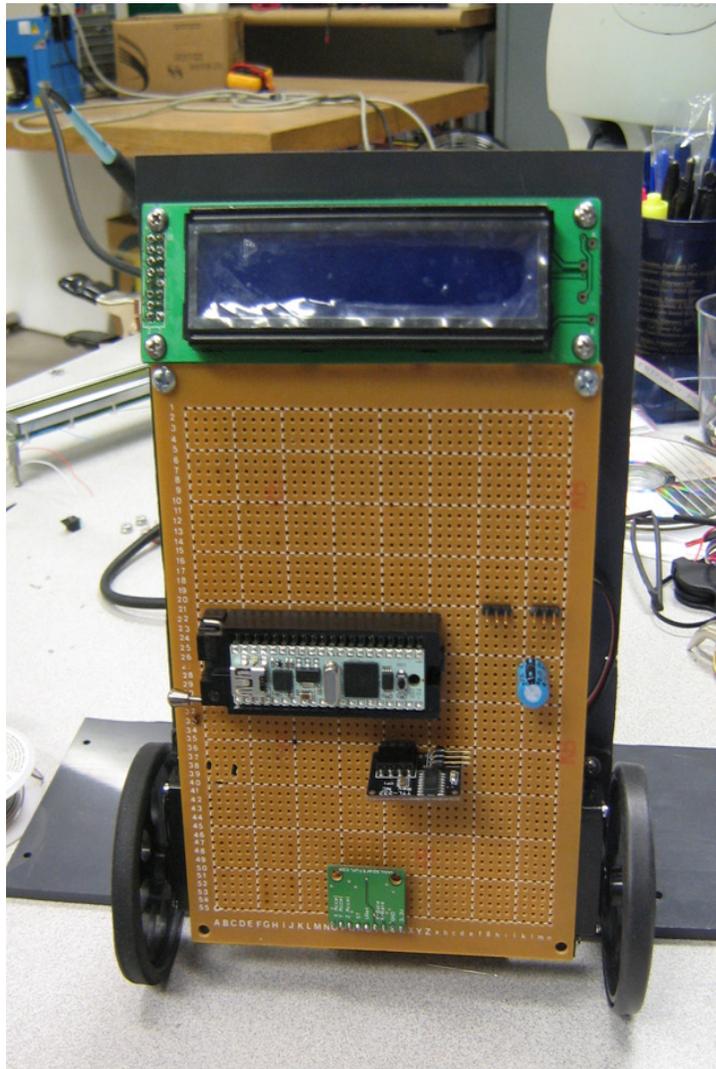


*Note:* I left room in the bottom corner for a voltage regulator to step down the 5 V needed to power the Baby Orangutan to the 3.3 V needed to power the IMU. The area to the top left is to hook the servo motors. The power that is going to be supplied to them will need to be seperated from the PWM signal which will be hooked up to the controller. As you can see the middle line across will be where the DC voltage will be attached. Before moving on, you should check that you can use all the individual components.

- **UPDATE: March 19th, 2007:** I added the voltage regulators. The left one will regulate a DC voltage to 3.3 V for the IMU and the right one regulates to 5V for the Baby Orangutan.



- **UPDATE: May 30, 2007:** Despite that each individual component was working combining the new design of potentiometers for PID gain tuning, the LCD screen, and the IMU did not combine well with the Baby Orangutan. With some help from Bryan Kobe, the decision was made to switch to the PropStick using a Propellor Chip to be the new micro-controller. Instead of using the parallel LCD screen, we used the Matrix Orbital serial LCD screen. Since there are two of us, we have quickly completed a new chassis to accommodate the LCD screen and everything else. The target date for completion is still June 1st. The new chassis is seen here:



- **Configure Software**

- To insure that the controller works, we will want to compile and try to program the controller with a PWM signal and insure that we can control the servos and then check the connection with the IMU.
- First thing is to download and install the [WinAVR software](#) for coding and compiling. After installing the software, download the [B-Bot source code](#) and compile it.
- To test the PWM signal, [download](#) and program the controller with the code I ported and implemented. The program should rotate both servos in one direction (forward) for 2 seconds and then the other (backwards) for another 2 seconds, and then stop.
- **UPDATE: March 4th, 2007:** I was using theCableAVR ISP programmer to program the Atmel Atmega168 microcontroller to no prevail. I established the programmer that I am using is faulty and am requesting to purchase an Atmel AVRISP programmer. The advantage of doing this is that it will be able to program the entire Atmel family of microcontrollers.
- **UPDATE: March 11th, 2007:** With the new Atmel AVRISP mkII programmer, I programmed and tested out my PWM generating signal using the ATmega168 microcontroller. I initially had an issue with the 20 MHz external oscillator mainly because the software for the mkII doesn't automatically support the Baby Orangutan. After reconfiguring the clock cycle internally in my program, the PWM signal generated only moves the servos in one direction only and I have been unable to move it in the other direction.
- **UPDATE: March 13th, 2007:** Implementing new code, I successfully have a working PWM signal i.e. working motors, next step is to add the SparkFUN IMU data and begin doing some balancing.
- **UPDATE: March 19, 2007:** I attached the IMU and started implementing the balancing bot using the PID controller designed by Davis's code. The robot doesn't balance at all yet, what it does is basically completely overshoots as spins out of control. This is probably due to the data coming from the IMU. Since there is no

LCD, I have no idea what readings are being read so the next step should be adding that to view the real time data. The issue basically lies in the IMU and the control code. The plan now is to add the LCD, voltage regulators, and lastly add a longer chassis. Software wise, I need to modify the PID controller to handle the data being received by the IMU.

- **UPDATE: May 30, 2007:** The code thus far is pretty much completed. We are waiting an ADC 0838 so that we can hook up the pots and IMU to the prop and begin balancing. We get the parts today so stay tuned we may have something in a next coming day.

- **Test Components**
- **Final Test Results**
- **Final Thoughts & Useful Links**

- *Useful Links*

- [WinAVR Sourceforge](#)
- [AVRFreaks](#)
- Original Nuts&Volts Articles
  - [Part 1](#)
  - [Part 2](#)
  - B-Bot Source [Code](#)
- Contact [me](#)