ROBOT MUSCLES • VAPOR BOT • GENETIC ALGORITHMS

SERVO MAGAZINE

FOR THE ROBOT INNOVATOR

www.servomagazine.com

May 2008

THE R2 BUILDERS CLUB AND THE JEDI CODE

THE SERVO BUDDY

Intro to Servo Motors And A Circuit To Control Them Without A Microcontroller

TEAR DOWN - THE iROBOT LOOJ

Get Inside The Latest Offering From iRobot And Explore Its Potential As A Robotics Platform


Let your geek shine.
Meet Leah Buechley, developer of LilyPad—a sew-able microcontroller—and fellow geek. Leah used SparkFun products and services while she developed her LilyPad prototype.
The tools are out there, from LEDs to conductive thread, tutorials to affordable PCB fabrication, and of course Leah's LilyPad. Find the resources you need to let your geek shine too.
sparkfun ELECTRONICS
»Sharing Ingenuity
SPARKFUN.COM
©2008 SparkFun Electronics, Inc. All rights reserved.

Hitec Goes The Distance!

Don't let your robot take a fall, make sure it can go the distance by using one of Hitec's high powered robotics servos. From the sport level HSR-8498HB to the stump pulling torque of the Titanium geared HSR-5990TG, Hitec has a servo for your robotics project.

HSR-8498HB High Torque Karbonite Gear | *HSR-5498SG High Torque Steel Gear* | *HSR-5980SG Coreless Mega Torque Steel Gear* | *HSR-5990TG Coreless Mega Torque Titanium Gear* | *HSR-1425CR Non Proportional Continuous Rotation*

Model	Gear Type	Torque(oz) 6V / 7.4V	Speed(sec) 6V / 7.4V	Bearing	Dimensions L" x W" x H"	Weight (oz)	Protocol
HSR-8498HB	Karbonite	103 / na	0.20 / na	Dual BB	1.57 x .78 x 1.45	1.75	*HMI/PWN
HSR-5498SG	Steel	153 / 188	0.22 / 0.19	Dual BB	1.57 x .78 x 1.45	2.10	*HMI/PWN
HSR-5980SG	Steel	333 / 417	0.17 / 0.14	Dual BB	1.57 x .78 x 1.45	2.36	*HMI/PWN
HSR-5990TG	Titanium	333 / 417	0.17 / 0.14	Dual BB	1.57 x .78 x 1.45	2.39	*HMI/PWN
HSR-1425CR	Nylon	na / 57	16 rpm	Dual BB	1.59 x .77 x 1.44	1.6	PWM

*HMI Is Hitec's Multi Protocol Interface which allows the programming of our servos via a PC using the optional interface kit (Part No. 78206) PWM is the standard R/C protocol and allows the programming of the robotics servos using the HFP-20 field programmer (Part No. 44430).

ROBOTICS HITEC CUP
1st US Competition

JOIN US AT ELECTRIC Flight EXPO

April 25-27, 2008
University of Phoenix Stadium
Glendale Arizona
Visit EFExpo.com For Details

HITEC ROBOTICS

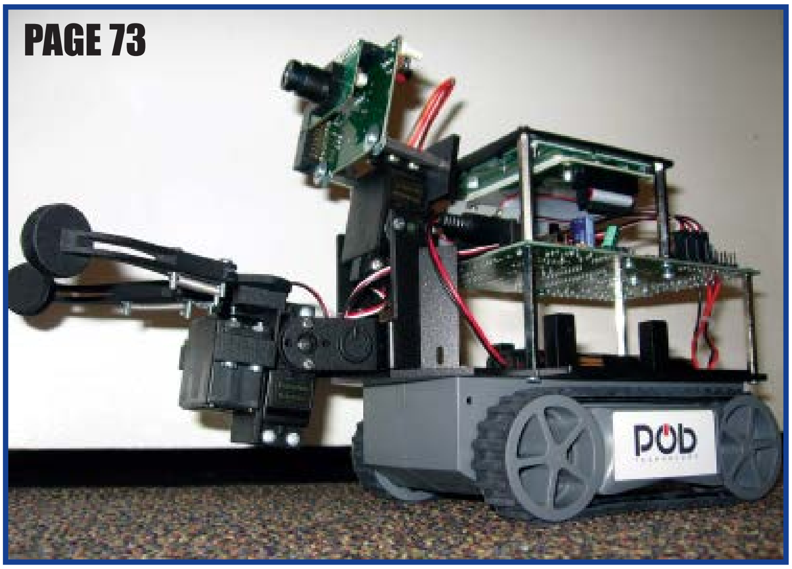
12115 Paine Street . Poway CA 92064 . 858-748-6948

Check out the RoboNova-1 and all the other Hitec Robotics products at www.hitecrobotics.com

SERVO MAGAZINE

Columns

PAGE 73



Departments

This Month In

THE COMBAT ZONE ...

Features

Events

Robot Profile

PAGE 10

05.2008
VOL. 6 NO. 5

Features & Projects

***SERVO** Magazine* **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

Mind / Iron

by Bryan Bergeron, Editor

Tough Enough

Although not always labeled as such, a lot of the articles featured in *SERVO* address the issue of fragility. For one robot to battle another head-on, or to compete in an arena where jarring contact is inevitable, it must be hardened against both the environment and attack from opposing robots. Weapon-wielding battle bots are the best example of the advantage of being tougher than the competition and, to a lesser extent, the environment. However, if you take a champion battle bot out of a clean arena and run it outside in the rain, the water will likely destroy the electronics and mud clog the drive mechanism long before the opposing battle bot has a chance to land a blow.

I'm not suggesting that the robotics community should focus on creating Terminator robots. However, if we're going to create truly useful service robots that can cook, clean, carry, and look after us in our old age, then we need to move past fragile robots to more sturdy designs. Most of my robots — whether from a kit or designed from scratch — couldn't survive a fall from a tabletop or an accidental dowsing with a cup of coffee. My relationship with these robots is as their caretaker. I can't yet imagine being taken care of by a robot.

Hardening a robot against the environment and other robots or even ill-meaning humans is no mean task. Companies like iRobot spend millions on R&D to harden their robots against the heat and humidity of the environment and still be capable of carrying out a military mission. Fortunately, as with many military advances in robotics, methods of hardening have percolated through to the consumer market.

The iRobot Looj (torn down in this issue) illustrates how an inexpensive consumer robot can be useful by virtue of its ability to perform in a hostile environment. The Looj features water-tight seams, structural integrity that allows it to survive a fall from a dozen feet onto a grassy surface, and power that approaches that of a battle bot. Even so, it's light enough to be clipped to a utility belt.

The typical environmental challenges faced by designers of service robots can be just as threatening as a rain gutter. Prototype robots designed to rescue soldiers from the battlefield must be able to handle a significant, unbalanced load while remaining impervious to mud and debris. Similarly, assistant robots for home and hospital use must be able to operate when contaminated with body fluids and, more significantly, withstand the rigors of decontamination. Unlike a surgical instrument that can be autoclaved, nurse proxy robots must be able to withstand frequent spray-downs with antiseptics.

As illustrated by the prototypes of service robots used to retrieve wounded soldiers from the battlefield or to help patients in and out of bed, service robots must have sufficient environmental resistance while posing

Mind/Iron Continued →

Published Monthly By
T & L Publications, Inc.
430 Princeland Court
Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277
North Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Tom Carroll, Gordon McComb, David Geer, Dennis Clark, R. Steven Rainwater, Fred Eady, Kevin Berry, Brian Benson, Robert Doerr, Kate Howard, Jim Stewart, Charles Guan, Rob Farrow, Bryce Woolley, Evan Woolley, Heather Dewey-Hagborg

CIRCULATION DIRECTOR
Tracy Kerley
subscribe@servomagazine.com

MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

PRODUCTION/GRAPHICS
Shannon Lemieux
Joe Keungmanivong

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

Copyright 2008 by
T & L Publications, Inc.
All Rights Reserved
All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *SERVO Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *SERVO*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *SERVO*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: **430 Princeland Court, Corona, CA 92879.**

minimal threat to humans. A robot capable of scooping up a soldier from a field like a forklift poses obvious threats to the downed soldier. Without considerable thought to scooping mechanism design, the scoops could further injure the soldier.

Similarly, an elderly patient could be bruised or seriously harmed by either a hard exoskeleton or exposed rotating parts. Some designers have addressed the issue of potential human injury by encasing their creations in a thick skin of foam. However, doing so introduces issues of reduced cooling efficiency, increased bulk, increased resistance to movement, and additional challenges of disinfecting the surface.

So, as you work to advance the state of robotics through higher-order AI functions such as navigation and auto-calibration of sensors, it's critical not to lose sight of environmental threats. If your goal is to develop robots that physically interact with humans, consider how you would harden the robot against the environment while presenting an interface that won't tear or bruise fragile skin and muscles.

If you think about it, human anatomy presents a perfect archetype for a service robot — a tough, weight bearing, and protective endoskeleton encased in a soft, protective skin that can interface with humans without causing them harm. Perhaps the Terminator model isn't far off after all. **SV**

Robot Psychologist: A New Era in Mental Health Services

Two clinical psychologists associated with the Institute for Eclectic Psychology in Holland (Jaap Hollander and Jeffrey Wijnberg) have developed the first robot psychologist, named "MindMentor." MindMentor is an online computer program that helps people solve problems and achieve goals. It has the unique quality (as compared with other online psychological help systems) of requiring no live human intervention and being completely automated. Said Hollander in a recent interview: "What made this whole endeavor exciting was that we suddenly saw a possibility to create an unlimited amount of psychological help."

Some psychologists have responded positively to their robot colleague. Said David Van Nuys, Ph.D. (Emeritus Professor of Psychology Sonoma State University): "At the end of the hour-long session, I have to say my outlook and spirits were lifted considerably. It was smart, supportive, fun, and funny, and helped me to focus in on the central issue. I find the blend of artificial intelligence, NLP, and other goal-directed therapeutic techniques effective."

How effective is the robot psychologist? Hollander, interviewed by a Dutch radio program, explained: "We did some research into the effectiveness of this system in 2006. We had a much more primitive version then, and with that we performed a test-run with 1,600 clients from all over the world. Our data show that MindMentor was able to solve the problems for 47% in just one session. When people were asked afterwards to what extent they had solved the problem with the help of the robot psychologist, 100% meaning totally solved and 0% meaning absolutely no change, the average result was 47%. We believe that this is a success percentage that any real-life psychologist would be satisfied with, especially given the fact that this was after just one session."

Check out MindMentor for yourself at **www.mindmentor.com**.

www.robotis.com

Why DYNAMIXEL?

High-performance networked actuators for robots

EX-106

NEW EX-106
- 200% upgraded POWER with Dual Action
- 400% enhanced ACCURACY with Encoder
- For +1 meter HUMANOID at 10~18V

DC Motor + Reduction Gear + Feedback Sensor
DYNAMIXEL
Controller + Driver + Network

AX-12+

- Engineering plastic gear
- Control network : TTL
- Gear Ratio : 254
- Postion Sensor : POT

AX-12+	at 7V	at 9.6V
Holding torque(Kgf.cm)	12.5	17.1
Speed(sec/60deg)	0.269	0.196
Weight(g)	53.5	

RX-10

- Full metallic gear
- Control network : RS-485
- Gear Ratio : 193
- Postion Sensor : POT

RX-10	at 10V	at 12V
Holding torque(Kgf.cm)	10.1	12.1
Speed(sec/60deg)	0.143	0.119
Weight(g)	64.5	

RX-28 / DX-117

- Full metallic gear
- Control network : RS-485
- Gear Ratio : 193
- Postion Sensor : POT

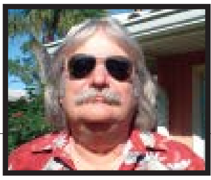
	RX-28		DX-117	
	at 12V	at 16V	at 12V	at 16V
Holding torque(Kgf.cm)	28.3	37.7	28.3	37.7
Speed(sec/60deg)	0.167	0.126	0.167	0.126
Weight(g)	72		67	

RX- 64

- Full metallic gear
- Control network : RS-485
- Gear Ratio : 200
- Postion Sensor : POT

RX-64	at 15V	at 18V
Holding torque(Kgf.cm)	64.4	77.2
Speed(sec/60deg)	0.188	0.157
Weight(g)	125	

ROBOTIS Visit www.robotis.com for more information on parts and accessories.
For business partnership contact us at contactus2@robotis.com



Robytes

by Jeff Eckert

Chatty Cathy Reincarnated

Classic Chatty Cathy (photo courtesy of www.joyndolls.com) vs. iCub (photo courtesy of RobotCub).

The concept of a mechanical talking adolescent isn't exactly new, with the terminally cute Chatty Cathy dating back to 1960. But while Cathy — at her peak —could only speak 18 phrases, the iCub, designed by the RobotCub Consortium (**www.robotcub.org**), may soon be generating complex conversations on its own.

An international group, led by the University of Plymouth (**www.plymouth.ac.uk**), began its Integration and Transfer of Action and Language Knowledge in Robots (ITALK) program on March 1. ITALK seeks to teach iCub to speak by employing the same methods parents use on their children (presumably skipping over "goo goo" and "dah dah"). The training will include "experiments in human and robot language interaction to enable the robot to converse with humans."

According to Profs. Chrystopher Nehaniv and Kerstin Dautenhahn of the University of Hertfordshire, typical experiments will include such activities as inserting variously shaped objects into corresponding holes, stacking wooden blocks, and aligning cups. iCub will then be asked to name the objects and actions, thus learning phrases like "robot puts stick on cube." Prof. Dautenhahn noted, "iCub will take us a stage forward in developing robots as social companions. We have studied issues such as how robots should look and how close people will want them to approach and now, within a year, we will have the first humanoid robot capable to developing language skills." The next challenge will be to make it shut up and go to bed.

Robots to Replace Animals

A new chemical safety test program was recently announced by the National Institutes of Health (NIH, **www.nih.gov**) and the Environmental Protection Agency (EPA, **www.epa.gov**). It aims to employ robots to reduce researchers' reliance on animal testing and improve the overall process. The collaborative effort will use the NIH Chemical Genomics Center's high-speed automated screening robots to analyze suspected toxic compounds "using cells and isolated molecular targets instead of laboratory animals. This new, transagency collaboration is anticipated to generate data more relevant to humans; expand the number of chemicals that are tested; and reduce the time, money, and number of animals involved in testing."

Full implementation is probably years away, so you won't run into many lab rats in the unemployment line any time soon. But when the proper procedures have been validated and put in place, it should be possible to test thousands to hundreds of thousands of chemicals per day to assess their possible toxic effects. More info is available from the National Human Genome Research Institute (**www.genome.gov**).

Automated Octopus Balls

Proving again that industrial robots can be programmed for more creative activities than welding automobile fenders, a recent exhibition at the Osaka Museum of Creative Industries revealed a Toyo Riki (**www.toyoriki.co.jp**) robot that was set up

This robot from Switzerland's Stäubli Group is part of an NIH/EPA program to improve toxicity testing.

A robotic chef prepares a seafood treat. Photo courtesy of Oriental Physical Machine Industry Co., Ltd.

to grill octopus balls (insert your own joke here). The exact ingredients were not revealed, but the bot does follow "a strict recipe for making the delicacy, crafting each octopus ball individually from scratch." It also flips each piece while it's grilling, then sticks it on a bamboo skewer, puts it on a plate, dribbles some sauce on it and, finally, shakes on some spices. Soon to be available at Long John Tentacle's.

Bot Bugs Bums

On a different level of the comestibles business is "Bum Bot," created to drive trespassers away from a section of Atlanta that includes O'Terrill's Irish pub (**www.oterrills.com**) and the Beacon of Light Daycare Center. Pub owner Rufus Terrill created the mechanism from a three-wheeled scooter and a barbecue smoker, equipping him with a spotlight, IR camera, PA system, and water cannon, the latter of which is capable of hosing down urban outdoorsmen from a distance of 20 ft. The 400-lb Bum Bot is operated by remote control, and a walkie-talkie provides him with the

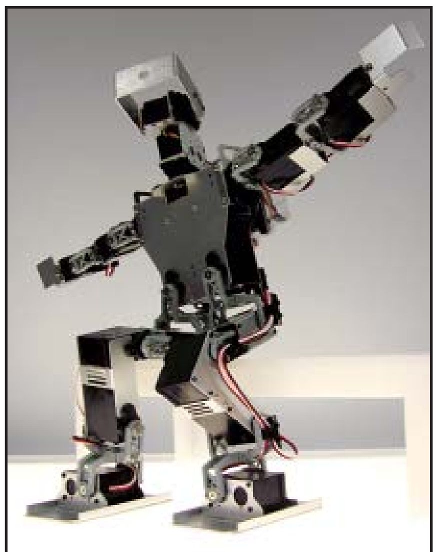
Photo courtesy of R. Terrill.

ability to address undesirable visitors in an authoritative voice. You can see him in action at **www.youtube.com/watch?v=y4mRJY6NEwI**.

According to Beacon operator Lydia Meredith, the whole square is "enveloped with homeless people and drug dealers, defecating, urinating, prostituting — the whole nine yards." And a nearby playground is habitually littered with used syringes and condoms. While most of the locals have nothing but praise for Bum Bot, a representative of the Atlanta police has warned Terrill that he "would be committing an assault if he intentionally sprays water on someone when in control of the robot." As they say, you just can't please everyone. But if you're in town, say "hi" to the Bummer and try O'Terrill's award-winning fish and chips. At only $11.99 for a 10-oz haddock filet, hand-cut chips, and homemade slaw, you won't get hosed — at least while you're inside.

Automaton of Your Dreams

Fernando Orellana and Brendan Burns, a Ph.D. roboticist, have created "Sleep Waking" (that's Waking, not Walking), as an experiment in linking REM sleep to robot behaviors. The interesting thing is that it was inspired by the concept of tapping into your dreams and programming a robot to reenact them. The disappointing thing is that, of course, it doesn't actually do that. But it does log a sleeping subject's eye position and translates that into the bot's head movement; if your eyes move to the left, so does the robot head, and so on. It also uses brainwave activity as detected by an EEG readout, assigning preprogrammed behaviors to a set of recognized brainwave patterns. Sleep Waking is meant to be viewed primarily as a metaphorical vision of the future, so it is pretty much art for art's sake. The robot's movements have even been set to music by San Francisco's spaced-out Ade Lun Sec (**www.adelunsec.net**). However, the creators managed to get partial funding from Union College and the Albany Regional Sleep Disorder Center, so someone apparently believes that it has some practical potential. You can view a performance at **www.youtube.com/v/1RkM1Bt2b3k&amp**. SV



Brainwave-inspired robot in flying stance. Photo courtesy of fernandoorellana.com.

GEER HEAD

by David Geer
Contact the author at geercom@alltel.net

The Northern Bites RoboCup Team

RoboCup was born to call attention to artificial intelligence and intelligent robot research, according to RoboCup.org. The subsequent contests and competitions challenge roboticists from various colleges and universities around the world to build the best AI robots and prove their achievements by winning all-robot soccer meets.

Playing soccer (football outside the US) requires robots to demonstrate many of the emerging technological capabilities that AI must rely on. According to **RoboCup.org**, these include teamwork, certain real-time intellectual properties, and advanced motion control. In 2007, teams of hacked Sony Aibo robot dogs competed in the Four-Legged League of RoboCup for the championship prize. The overall winner was the Northern Bites team from Bowdoin College.

Parents weekend at Bowdoin College; families gathered to watch a Northern Bites Soccer match.

The Northern Bites

The Northern Bites team uses a total of 12 ERS-7 Aibos, including four from each of the models a, b, and c, according to Professor Eric Chown, the team advisor at Bowdoin College.

While the RoboCup rules strictly prohibit physical modifications to the robots, the Aibo's necks are not strong enough for the soccer competitions. When a German team discovered a way to make the necks tougher and more resilient, RoboCup decided to permit all teams to make the same physical improvements in the robots.

Except for these minor physical adjustments, each team's main objective is to program the robots so they can play soccer. The best programming for the task should naturally lead a team to become the ultimate champs in the Soccer competition. In 2007, the best programming would come from Bowdoin College.

The Northern Bites team uses "SVN" as their version control software, and "Trac" as their overall project manager, according to Professor Chown. While the Northern Bites programmers could also have used existing development environments like Tekotsu, they decided to write the rest of their tools and software for the Aibos from scratch. "We're computer scientists! We can write our own tools. Now, we have a whole suite of tools (that we are currently integrating into one system called 'the tool')," says Chown.

While the Northern Bites team and competing RoboCup teams use Aibos as they come — with their own operating system software — this has been an obstacle rather than an aid. Sony created the Aibos as toys and with limited functionality, not as full-on Soccer athletes. Each team has had to adapt the existing software by adding code and other programming to make them Soccer-ready.

Fetching Aibo's Memory Stick is Tedious

One of the limitations of the Aibo software involves the robot's physical infrastructure. "Programming the robots normally requires compiling a program onto a Sony memory stick," explains Chown. Here's how this leads to trouble ... "Anytime you want to debug, you have to turn the robot off, remove the memory stick (not an easy task), put it into a reader, re-compile your program, and replace the memory stick. This is extremely time consuming."

In addition to the memory stick issue, Sony writes the Aibo programs in C++, which is a harder language to work with, according to Professor Chown. To resolve both problems, the Northern Bites team ported the Python programming language to the Aibos so they could work with that instead of C++.

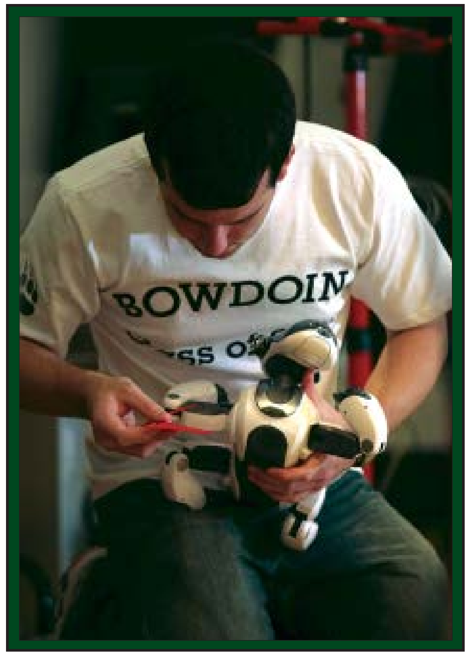
Python is easier to work with and it enables the Northern Bites programmers to load their Python-based programs into the dogs "on the fly," without the arduous process of removing and re-installing the memory stick every time. "We can update the robot wirelessly," smiles Chown.

To do remote command and control with the Aibos, the Northern Bites team had to build a wireless framework. "We call this system AiboConnect," says Chown. AiboConnect allows the team's programmers to easily debug the software they create for the Aibos. AiboConnect is a foundation for other software tools used by the Northern Bites team.

One such tool calibrates the Aibo's robot vision for differing environments. The tool streams images from the dog wirelessly and sends commands back to the robot, directing it to move into different areas of the playing field to collect different images to calibrate against.

Command and control communications between the Aibos during competition are also accomplished wirelessly. There are some not so simple obstacles to using wireless in this type of competition. Because many leagues of robots are using wireless in the same venue at the same time, there is a lot of interference across the wireless spectrum, according to Chown. The use of wireless in the Northern Bites robots also tends to slow the robots down. To avoid these complications, the Northern Bites have a bandwidth limit of 500 Kbps on wireless transmissions.

Northern Bites in scrimmage match.



One of the Northern Bites soccer robots in the hands of a Bowdoin student.

Team operators use a software game controller to direct the actions of the robots. "It takes the dogs through various game states like playing, penalized, ready, and kick-off," says Chown. "When that isn't working, such as when wireless isn't working, the robots need to respond to button pushes (the robots have three buttons on their back that can be used as a substitute for commands from the game controller)."

Technology Gets Northern Bites to the Ball First

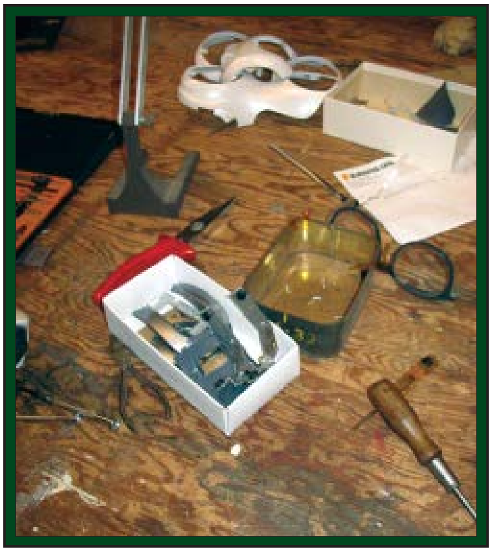
According to Chown, about four times per second, each Northern Bites Aibo sends out information about its current position, whether it sees the ball, where it thinks the ball is, and how long it thinks it would take it to grab the ball.

"Each robot uses a combination of what it knows and what it heard from its teammates to assess what role it should be playing. Attacker, defender, supporter, and goalie are the basic roles. At any given time, the closest robot to the ball (that isn't the goalie) becomes the attacker. The Aibo robots decide the roles for the

FIXING AIBOS FOR ROBOCUP COMPETITION

Many of the images in this article show a disassembled Aibo in the process of a neck joint fix. This joint permits the neck to turn in relation to the rest of the body. The Aibo's head comes with three additional joints. One joint enables the head to turn left to right. Another enables the head to tilt up and down. And finally, a third joint enables the mouth to open and close.

This particular neck joint became loose and gave out due to a design flaw related to the gear housing. A loose neck joint would make it difficult for the Aibos to shoot the ball because the team relies on the accuracy of the pressure on the ball from that joint in order to quickly shoot the ball.



This image shows two boxes with parts from a disassembled Aibo and additional parts in white in the background. The Northern Bites team had to remove all these to get at the robot's neck, to remove it.

Not from Sleepy Hollow, but a headless Aibo none-the-less. On its back are silver touch-sensitive LEDs. Inside are its motors and microprocessor.

rest of the team based upon relative positioning. Our positioning and teamwork is what won us the world championship. Though we were no faster than any other team, we got to loose balls about two out of three times against the best teams (and far more against weaker teams)," says Chown. When the ball goes out-of-bounds, the referee places it in-bounds based on where it went out and which team knocked it out, Chown explains. This is because the robot dogs can't actually throw the ball in, as regular Soccer athletes would.

The Northern Bites dogs use an Extended Kalman Filter (EKF) to estimate the ball's position. This becomes especially important when determining position after the ball has been placed back in bounds.

For the dogs to recognize the ball's new position when placed by the referee, the programmers added rules to the EKF that basically tell the dogs that the ball is able to teleport to standard field locations once it has gone out-of-bounds. This enables them to easily locate the ball again after it has been moved by the referee. This feature enabled the Northern Bites robots to get to balls positioned by referees 85 out of 127 possible times.

Soccer Moves

The basic moves for a Soccer-playing Aibo include walking, kicking, and running. The Northern Bites team uses a walk engine to create the variety of joint angles needed to get the robot to walk or run as desired. The Northern Bites programmers looked at walking or running as the movement of the robot's foot through space. In that way, they can think of the movement as a shape like a trapezoid. By using an inverse kinematic system, the walk engine tells the programmers how to set up the dog's joints so it can make those shapes and walk or run accordingly. The roboticists feed this information into a machine learning system. This enables the robots and programmers to learn the best shape of the curve for the foot to make and how fast it should move through that curve for optimal walking or running.

To kick, the dogs trap the ball under their chins and draw it back to their chests. "From there, most kicks consist of the dog using its two front legs in a kind of chopping motion to knock the ball forward," says Chown. The team uses a method in which the dog runs at the ball as fast as it can and grabs the ball under its chin in one motion.

This is another disassembled Aibo with the neck removed. The gearbox was at issue here, which the team was trying to repair. It is lying left of the Aibo's head. A workbench with tools and instructions for dismembering Aibo lay in the back.

Here you can see the neck joint under repair. In the middle of the neck, on the light gray plastic is a fitting that connects the gear above the neck. The fitting has begun to erode. The team used epoxy to strengthen the connection between the gear and fitting.

"For the actual kick, we then have a kick engine based on a keyframe idea. Each kick is specified as a series of frames where each frame contains the joint locations for all of the robot's joints, as well as the time elapsed to the next frame," explains Chown. "At its heart, our goal is to tell the robot 60 times a second what its joint angles should be. Meanwhile, 30 times a second it is giving us a vision frame."

To see the ball, the Aibos do repeated scans of every image using a basic vision algorithm, first scanning from top to bottom then from left to right. This scanning collects and differentiates between runs of color. The software then places the runs together into blobs using run-length encoding.

Conclusion

It is unique to see a relatively new undergraduate team from a liberal arts college beat the top graduate schools in the RoboCup competition. And that's exactly what the Northern Bites did. The Northern Bites team will compete at the US Open in Pittsburgh at the end of May 2008 and at the world championships in China in July. **SV**

RESOURCES

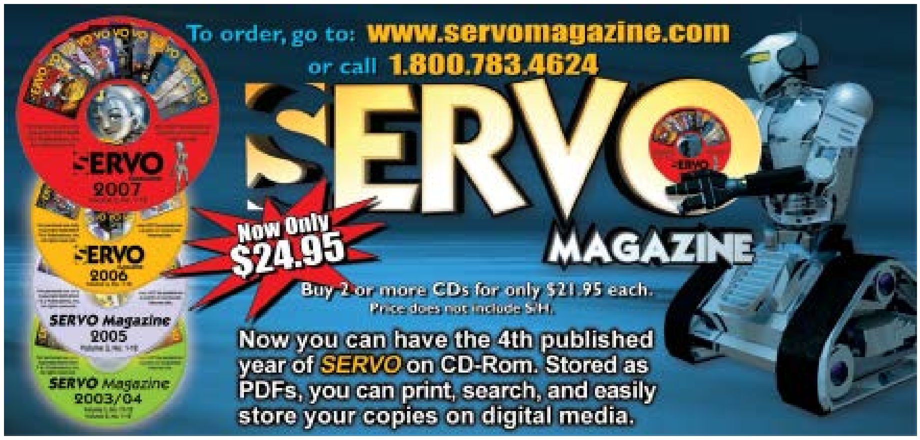
Bowdoin College
www.bowdoin.edu

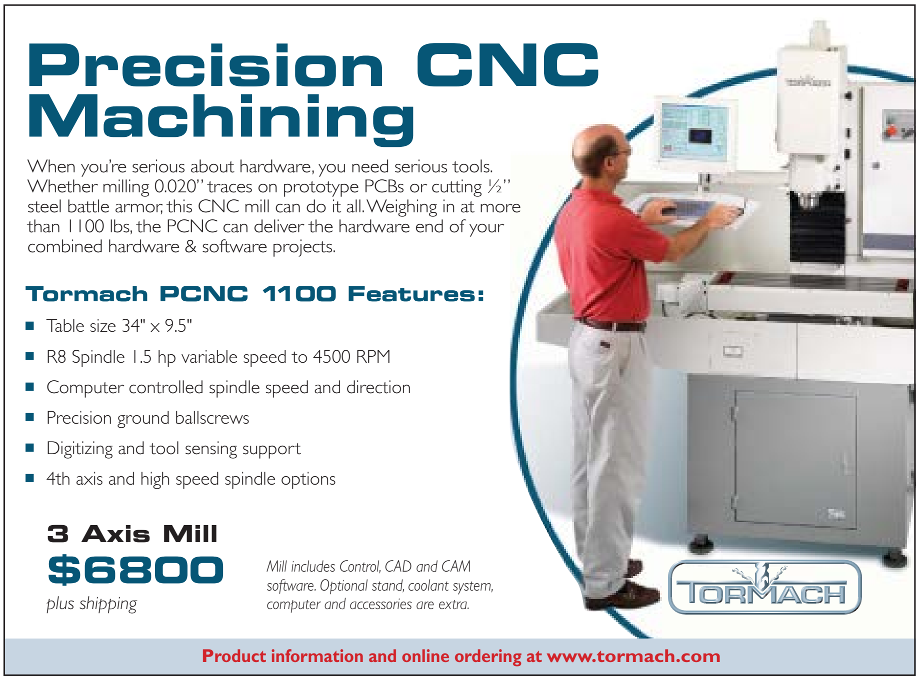
Bowdoin College, RoboCup
www.bowdoin.edu/computer-science/robotics/robocup/

Bowdoin College Robotics
www.bowdoin.edu/computer-science/robotics

Procedures for repairing Aibo necks as published by Jörg Zimmer from the University of Darmstadt, Germany, and approved by RoboCup
www.student.informatik.tu-darmstadt.de/~zimmer_j/case-legs.htm

www.student.informatik.tu-darmstadt.de/~zimmer_j/neck-tilt.htm"





Our NEW resident expert on all things robotic is merely an email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

by
Dennis Clark

Q. When building a homemade robot controller, how do I separate logic and analog ground? I've read that all grounds need to be tied together, even if you are using more than one battery. If this is the case, don't these need to be separate?

— Mike Bonner

A. This question comes up a lot. I'm going to answer the second part of the question first since that will make the first part easier to understand. The biggest confusion comes from the concept of "ground." Because our robots are mobile (most of the time), they really aren't "grounded," which means they aren't connected to Earth ground. The minus (-) side of the battery is really just the current return — the completion of the circuit. It is often called common because all signals are referenced to this "common" electrical level. In other words, 5 volts DC is only 5V DC in reference to the common. When more than one battery is in a system, this common point becomes very important. How would we know what the voltage is between parts of the circuit if there is no agreed upon common point? Why is this important? Here is why.

Let's say that we have an H-bridge that gets its power from a 12V battery. Let us also say that we have a 5V battery powering our micro. If the bridge needs a logic "high" of 3.5V to turn on the power to the motor, that has to be 3.5V in reference to the common. If our batteries do not have their returns (or grounds if you wish) tied together, what does that 3.5V mean? I can see the questioning look in your eyes already ... Why would the return be different for one battery than the other? It can be, and it can be hugely different.

In order to measure a voltage, we measure a "potential difference" of electrons between two points. If those points are isolated from each other, there is no current path. We have all heard that "current follows the path of least resistance." This is partly true, but I won't get into the details of that right now. Most of the current follows the path of least resistance, but all of it travels somewhere. When all of the paths are high resistance — like air — you could get any value at that measured point, even hundreds of volts (I've seen that happen on an improperly grounded power supply.) There is just no way to know what will be seen at that control pin. Your operation will be erratic, and most likely non-existent. With the minus (-) leads all tied together, the common reference is defined and, if you will, agreed upon by all the components in the circuit. This is why you must have all batteries' minus (-) leads tied together.

Now that I've told you that, I'll contradict myself a little bit. If you need to have positive and negative voltages in the circuit, how would you do that? Let's say that you need to have plus and minus 12V in your robot. Remember that I said that the ground really isn't ground? That it is really the common reference? (Can you guess what I'm going to write?) Yup, if you need a -12V, you would tie the plus (+) side of one battery pack to the common and take the voltage from the minus (-) side of that pack. I'm really not contradicting myself here — that the common point is the reference — that battery is providing -12V with respect to the common reference. Hopefully, that will make sense to everyone. Okay, now that I've answered that part of the question, let's answer the first part of the question. Why are the logic and analog (or high power circuits, as well) kept separate, and how can they be separate if they have to be tied together (as said in the first part of my answer)?

Well, there is separate and there is separate. We know that all of the grounds/commons must be tied together, but there are ways to do this that are good and ways that are bad. First some background: The traces on a circuit board are copper, and because copper is not a perfect conductor, it has resistance. The longer the trace, the higher the resistance. This makes the trace look like a resistor to current flow. Ohm's Law states that V = IR, which means that there is a voltage drop (V) across any resistor (R) through which current (I) flows. So, while our DVM set to beeper mode tells us that any point in the ground plane (or return path if you want) is the same, in reality while the devices on the board are running, the voltage at various points in that ground/common may not be at the same 0V. The space between two chips or even two pins on the same chip will have resistance and current flow will cause a small voltage difference. This difference is usually called ground bounce and it is considered to be a "bad" thing; most especially for analog circuits. It is bad for analog circuits because the difference between 0.1V and 0.15V may be of great value to us (think of the analog output from a Sharp IR rangefinder.) We want the power lines to analog circuits to be "quiet." One of the ways to keep this power quiet is to have minimal ground bounce. The other is to have a nicely filtered voltage on it, but this is another topic. Digital circuits are more tolerant of small ground bounce because there is a fixed logic low maximum and logic high minimum that allows a range of correct values.

When we have high powered devices in the circuit like a motor H-bridge, this could affect even the digital circuitry, so even more caution is required in how we connect up the power and common. Figure 1 shows an example of a bad ground/common bus and Figure 2 shows an example of a good ground common bus. In these figures, the fat lines are high current paths and the skinny lines are low current paths. You want to avoid putting devices that have low current needs in the middle of paths flowing between devices that have high current needs.

In the bad ground bus, we have connected battery and device grounds anywhere we please. This means that high currents and low currents can be seen anywhere on the board. If that H-bridge is putting out two amps of current, then lots of different ground current loops in the board and ground bounce is going to affect chips. In the good ground bus, we see that the digital, analog, and high current grounds are kept separate and that all grounds are tied together at a single point — what we usually call a star configuration. Note that the microcontroller has both analog and digital grounds. Keep those separate until you tie to the common return point. Note also that the H-bridge has a logic and a power ground; again, keep those separate until you tie into the common return point. Multiple digital components can be tied together along the way to the common return, as well as multiple analog devices. The secret is to keep the ground/common paths of the various signal types isolated from each other until they tie together at the common return point. I think that I've stressed this point enough now. You can have lots of problems that are difficult to understand with a controller board if you don't follow these basic guidelines.

Q. How can I set up the PWM on an Atmel ATMEGA part? What frequency is the best to use?

— Tran Pham

A. The first part of this question is fairly simple; I'll use the Timer/Counter1 timer which creates the OCR1A and OCR1B PWM outputs. I have no idea what compiler you are using so I'll pick two popular ones: BASCOM/AVR® and gcc-avr. To keep this simple and easy to use, I will also use what Atmel calls Fast PWM and eight-bit mode. (Please see their datasheet for what this means.) I've chosen a frequency around 1 kHz (976 Hz) since this will work with even (ahem) inexpensive DC brushed motors.

This assumes that your clock frequency is 16 MHz; adjust your prescale value accordingly to get as close to 1 kHz as you can. Both 500 Hz and 2 kHz will typically work well if you can't get to 1 kHz. You have a limited number of prescale options to choose from (1, 8, 64, 256, and 1024.) Your PWM pins are dependent upon the actual ATMEGA device that you are using.

BASCOM/AVR

```
Config Timer 1 = Pwm, Pwm = 8, Prescale = 64, Compare A
Pwm = Clear down, Compare B Pwm = Clear down
```

If this doesn't work correctly, you can bypass the hardware abstraction call above and punch numbers directly into the associated registers like this:

```
Tccr1a = &HA1    'Set type and mode of PWM and turn it on
Tccr1b = &H0B    'Set mode of PWM and prescale value

Ocr1a = 0        'set the PWM duty cycle for OCR1A output
Ocr1b = 0        'set the PWM duty cycle for OCR1B output
```

gcc-avr

```
TCCR1A = 0xA1;          //Fast PWM, 8 bit
TCCR1B = 0x0B;          //8 bit 976 Hz
OCR1A = 0;              //No signal out when = 0
OCR1B = 0;
```

The key is to get the right configuration bits set and a usable PWM frequency. This configuration uses the Fast

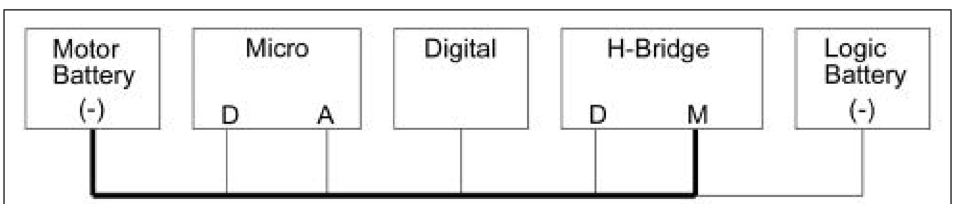


Figure 1

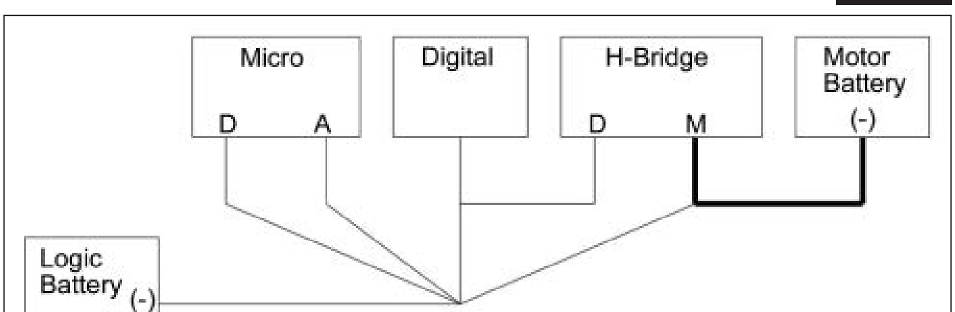


Figure 2

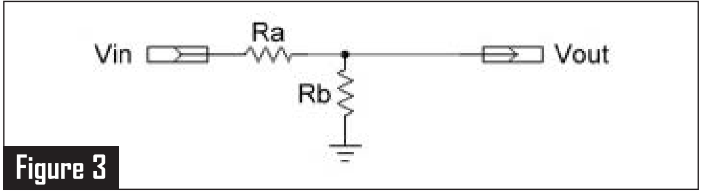


Figure 3

PWM settings, which means that the PWM counts down to the match number (OCR1A or OCR1B) and then clears the output. When the count rolls back over to 255, it sets the output again and starts over. The Phase Correct method that you can choose ramps down to the match number and clears, then when it reaches zero it does the opposite, ramping back up. I've not noticed any performance difference between these two modes.

The frequency you should use for your motors depends entirely upon the motor quality and the ability of your H-bridge. The motors used in automobile window drives, toys, and such work best at 500 Hz to 1 kHz. An expensive Escap motor will PWM very smoothly at 20 kHz. Check your datasheets for your H-bridge, as well. A 754410 and L293D dual H-bridge specifies their tests at 5 kHz. I've blown the top off of the chip at 20 kHz, so I'd respect that number.

Q. I have a battery that is being used to power a very high powered set of motors in my robot and it can get very hot. This bothers me and I want to measure the temperature of the pack while I'm running it and shut down the motors if it overheats. How can I do this with a TC77 temperature sensor?

— Tom Bartlett

A. A lot of people just don't pay attention to that little detail. Heat is a killer for all battery packs and it can be REALLY bad if you are using lithium polymer packs. The TC77 gives its temperature in straight Celsius or Fahrenheit, depending upon the part. It uses an SPI-ish synchronous serial protocol that is easy to use. If you let it run in the default mode, you can simply read it and the most recent value will be sent back.

This is handy because the TC77 has a single data input/output line that is kind-of weird to use, so we'll simply read from it and save time and confusion. Because this part is synchronous serial, you can bit-bang the communication even if you have interrupts in your system disturbing your code; it will tolerate the clock bit stretch.

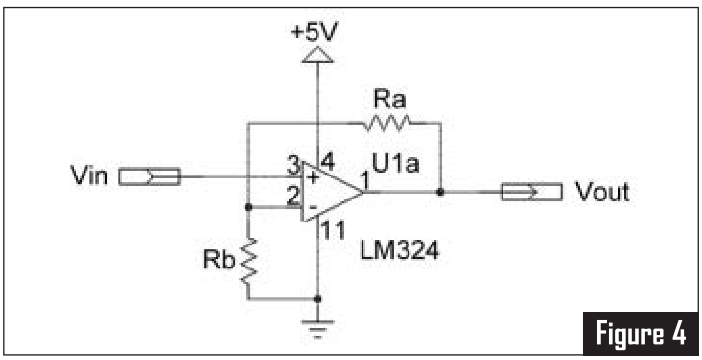


Figure 4

My example for this part is using Microchip's C18 C compiler for a PIC18F252 microcontroller to handle this bit banging. You can easily convert to any other compiler; the code is very simple. I chose to only get the integer part of the temperature and ignored the fractional value. I suspect that this is adequate for your use.

My code waits the minimum amount of time for each bit clock. The part is quite fast and I'm running my PIC18F252 at the full 40 MHz.

PIC C18

```
unsigned char GetTemp(void)
{
      unsigned char temp;
      unsigned char i;

      TC77_SCK = 0;      //Set all lines to a known state
      TC77_CS = 0;
      temp = 0;

      for (i=0;i<9;i++)  //Yes, 9 bits, discard the first one
      {
           TC77_SCK = 0;
           temp <<=1;     //Get ready for the next bit
           TC77_SCK = 1; //clock in on the rising edge
           temp |= TC77_SIO;     //Get the next bit
      }
      TC77_CS = 1;       //disable communications

      return (temp);     //This is the integer temperature
}
```

The three TC77 defines are assigned to their respective I/O pins going to the TC77 sensor. Make sure that you put the sensor directly on the pack, preferably right in the middle.

Q. How do I scale down a voltage to read it using an ADC pin on my microcontroller if it is too high to read?

Rowen Cowley

A. The simple answer to the first part of your question is to use a resistor divider to scale down the input voltage. Most microcontrollers require that you keep the impedance of the analog circuit under some value. With a PIC thats value is 10K ohms; the Atmel AVR parts also recommend 10K ohms as a maximum impedance. Other microcontrollers may require a different maximum impedance. Figure 3 shows the resistor network circuit that you would use.

You will need to choose resistors that will step the voltage down to what you need. Choose the resistor set that will keep the maximum voltage that you will experience in the range that your micro can handle. The formula for determining this voltage is:

$$Vout = Vin(Rb/(Ra + Rb))$$

This answer just begs to be enhanced with a way to scale a voltage up to get better resolution. If you have an ADC of eight bits (maximum of 255) and you have a voltage coming in that is only 1.2V maximum and your

ADC has a 5V reference, then you will only get to use about 1/5th of your total resolution to describe the signal. But, if you could amplify that signal to a higher voltage, then you will get much better resolution of that signal.

This means you can use more numbers to represent the signal if the signal is larger. The way to get this amplified signal is by using an op-amp. The one that works best — in my opinion — for hobbyists is the LM324. The LM324 requires only one voltage supply; many op-amps require a positive and negative supply. Figure 4 details how you would use the LM324 to scale a voltage up for your microcontroller. The gain of an op-amp is determined by the bias resistors using this formula:

$$V_{out} = V_{in}(1+(R_a/R_b))$$

You can use a variable resistor in series with Ra to tweak your amplifier to get the value that is just right. **SV**

NEW PRODUCTS

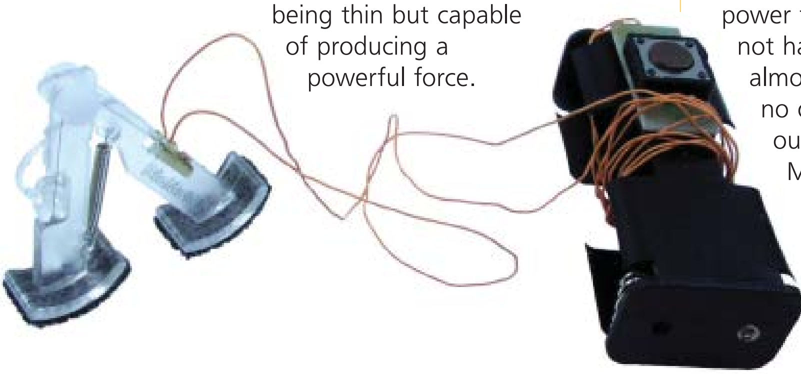
CONSUMER ROBOTS

BioMetal Inchworm

This walking robot, available from Images Co., uses a small length of BioMetal Fiber. Place the walker on a non-slippery flat surface. Press the switch on the battery case for about two seconds to widen the leg angle. Release the switch to make the legs return to their original position. Make the robot move forward by repeatedly pressing and releasing the switch. Price is $24.95. (A movie is available on the website.)

BioMetal Fiber

The BioMetal Fiber (BMF) is a fiber-like actuator (drive unit) designed to contract (tense) and extend (relax) like muscles. The key feature of this product is its flexible, smooth movements, like those of real life. Although soft and pliable like a nylon thread under normal conditions, it becomes stiff like a piano wire and sharply contracts when a current is fed through it. If the passage of a current is stopped, it will soften and extend to its original length. The BMF can also be moved by changing its temperature. It begins to contract when heated to about 70°. If it is cooled to below the temperature, it will return to its original length. Because of its stable internal structure, the BMF has very high durability and exhibits stable operating characteristics, being thin but capable of producing a powerful force.



The key difference between this material and standard Nitinol wire, for example, is this material extends back to its original length when cooled down (whereas Nitinol wire must be stretched back to its original length by a biasing weight or spring).

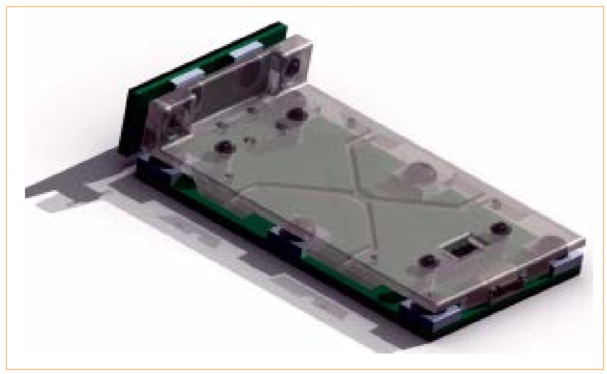
Price is $29.95 and includes one BioMetal Fiber (one meter long), four spectacle terminals, four eyelet terminals, crimp sleeves, spring, and instructions.

For further information, please contact:

Images Scientific Instruments, Inc.	Website: **www.imagesco.com/nitinol/inchworm.html** *or* **www.imagesco.com/nitinol/biometal-fiber.html**

SENSORS

SkewlZone



RoadNarrows introduces SkewlZone™, a suite of products designed to augment legged robots. Essentially, SkewlZone is an electronic brain and sensor pack. By design, SkewlZone products are a generalized system that can be used on many legged robots, providing these robots with sensors and computing power that walking robots on the market currently do not have. Many of the popular legged robots operate almost entirely in open-loop. That is, the robots have no or limited sensory input from the environment outside of servo position and speed data. Manufacturer-supplied robot controllers are usually dedicated for the real-time control of 17 or more servos. They have little access capacity to process more complex sensory input from the environment and integrate these input streams with the current set of desired robot behaviors and high-level goals. Augmenting a

Carnegie Mellon University's Robot Hall of Fame® Inducts Four Robots

Carnegie Mellon University has inducted four robots into its Robot Hall of Fame®. A ceremony, featuring actors Anthony Daniels and Zachary Quinto, highlighted the contributions and significance of each of the new inductees — the Raibert Hopper, NavLab 5, LEGO® Mindstorms, and the fictional Lt. Cmdr. Data.

In addition to the induction of the robots, Carnegie Science Center and Carnegie Mellon University announced that the Robot Hall of Fame will have a permanent home beginning in spring 2009 at the Science Center when it opens Roboworld, the nation's largest and most comprehensive permanent robotics exhibition. The Robot Hall of Fame — created in 2003 by the Carnegie Mellon School of Computer Science — recognizes excellence in robotics technology worldwide and honors the fictional and real robots that have inspired and embodied breakthrough accomplishments in robotics. Each year a jury of scholars, researchers, writers, designers, and entrepreneurs select the robots for recognition and induction into the Robot Hall of Fame.

Daniels, who played C-3PO in all six Star Wars movies, served as master of ceremonies and Quinto, a Carnegie Mellon alumnus who will play Spock in the upcoming Star Trek movie, attended on behalf of Data — an android with super strength and super memory that was portrayed by actor Brent Spiner during the 1987-1994 television run of "Star Trek: The Next Generation."

Also scheduled to attend was Marc Raibert, president of Boston Dynamics, who led development of the one-legged Hopper in his Leg Laboratory, first at Carnegie Mellon and later at MIT. The Raibert Hopper explored principles of dynamic balance that are central to agile movement by bipedal and quadrapedal robots.

Lars Nyengaard, director of Innovation and Education Projects for LEGO Education, was on hand for the induction of Mindstorms, a robotic kit that made robots accessible to the masses.

Todd Jochem, a PhD graduate of Carnegie Mellon's Robotics Institute, spoke at the ceremony on behalf of NavLab 5, one of a series of autonomous vehicles developed at the Robotics Institute. Jochem, who later founded Applied Perception, Inc., and is now group director of Foster-Miller, Inc., was one of two students who rode in NavLab 5 in 1995's "No Hands Across America" tour, during which NavLab 5 steered itself coast-to-coast on public highways.

The four robots inducted this year were announced last year at the RoboBusiness Conference and Exposition in Boston. The induction ceremony at Carnegie Science Center was held in conjunction with this year's RoboBusiness event at the David L. Lawrence Convention Center in April.

legged robot with the SkewlZone endows it with far greater potential. An easy-to-program single board Linux computer (SBC) connects to SkewlZone add-on sensors, such as hand and foot tactile sensors, an inertial measurement unit, and a USB color camera. The Linux SBC is also connected to the manufacturer's servo controller to issue commands and receive timely servo state updates. Taken altogether, a robot with SkewlZone provides a platform with capabilites of motion, touch, balance, and eyesight. Finally, a Wi-Fi connection is included for monitoring and for even more intense off-target AI applications.

Highlighted in the product suite is the SkewlZone Humanoid Foot. It gives the roboticist the ability to continuously measure the location of the center of gravity of the robot, the center of pressure on each foot, and the magnitude of force on the bottom of the foot. The toe also gives force feedback making it ideal for soccer competitions. Its onboard LEDs give the user instant feedback on the foot's current status and high visual appeal. An onboard Atmel® micro-controller comes preprogrammed with sensor calibration and operation controls, however, an I^2C interface offers programmable control over all of the foot's functions for advanced users. The feet can be used as stand-alone components, but their true value really "stands" out when integrated with RoadNarrows' Linux SBC brain pack. Currently, it is compatible with Kondo and Manoi Robots. For more information, please contact:

Road Narrows
1151 Eagle Dr. #140
Loveland, CO 80537
Tel: **800 • 275 • 9568**
Email: **oneway@roadnarrowsrobotics.com**
Website: **www.roadnarrowsrobotics.com**

COMBAT ZONE

Featured This Month:

If you'd like to see more photos of kinetic energy weapons from the April Combat Zone article by Mike Jeffries, go to ***www.servomagazine.com/downloads****. There just wasn't enough room to print all of them!*

BUILD REPORT

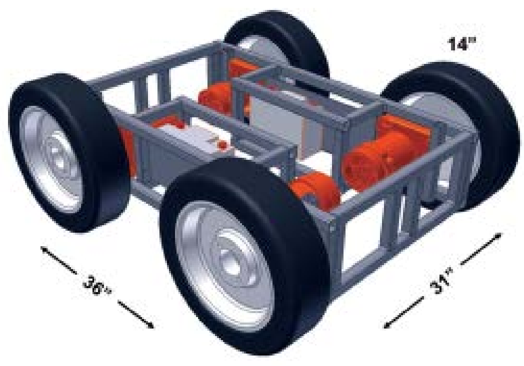
Vapor Bot Build Report

● by Kevin Berry and Charles Guan

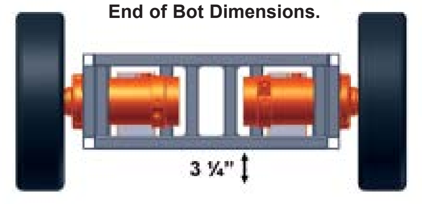
Come on, we've all done it. Bot inspiration hits, and we jump on parts sites and toss together a vapor bot. Sometimes it's in our heads, sometimes sketched on the back of a random piece of paper in a meeting or classroom, and often it makes it to the drawing stage. Honest builders will tell you they've produced a dozen vapor bots for every one they've actually fielded.

My latest venture into vapor bot land was prompted by a post to the SouthEastern Combat Robotics forum by a firefighter in a small rural department. Their budget won't allow them to buy a $30,000 off-the-shelf public safety machine, so they are hoping to build something themselves. Well, combat builders are the kings of build-it-yourself, so a collaborative vapor bot project quickly ensued.

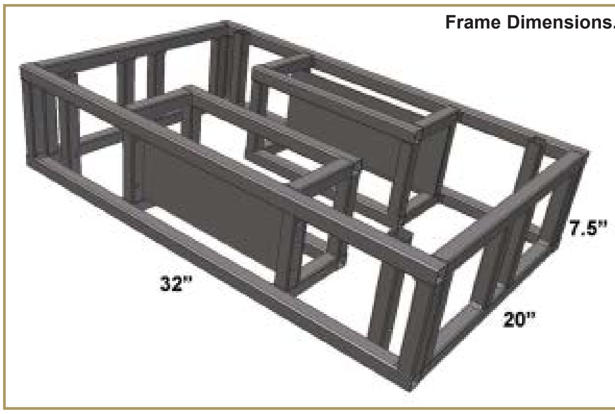
We came to understand the basic requirement was for something that could crawl out to an accident site, carrying some stand-alone monitors and a camera (items they already have), possibly shoving debris like a car door out of the way or climbing over it. It might also carry spare breathing bottles or beefy tools for rescue personnel into the danger zone, so fewer people are exposed to potential hazards. A



Bot Dimensions.



End of Bot Dimensions.



Frame Dimensions.

higher end requirement still under discussion would be for the bot to grasp an unconscious person and pull them out of an extremely dangerous situation, to a spot where rescue workers could intervene.

Being a near-farm boy myself, (we lived in town, but it had 800 people and only one hardware store), I thought about what a rural farm shop might have available to build something like this. Out of that came some general design requirements:

1) Must be able to be built in a medium sized farm or school shop, or a welding shop by generally handy people.

2) Must be able to function in an explosive atmosphere. This means the electrical devices must be fully enclosed.

3) Any functions done in the field (on/off switch, changing payloads, carry bars) can be done in full turnout gear with breathing mask (gloves, full face mask, coat, boots, helmet). Radio can be operated with bare hands.

4) Most parts available at a chain hardware, home improvement, or farm supply store. Rest easily ordered on-line.

5) Power by common motorcycle, ATV, jet ski, snowmobile sealed lead acid battery; 120 VAC battery charger on board.

6) Service and major component replacement in firehouse using only common hand tools.

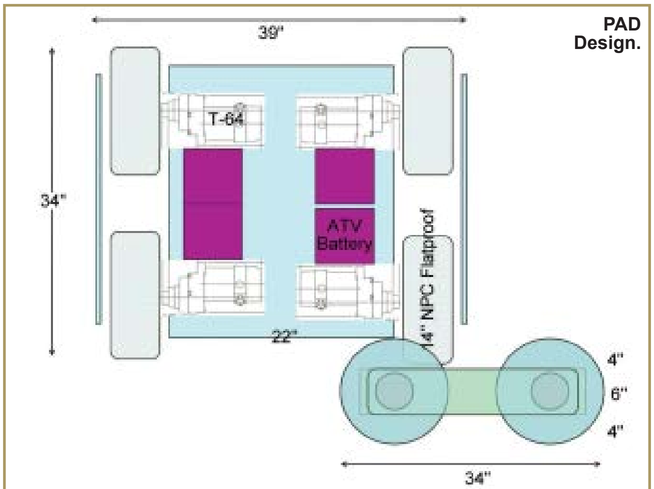
7) One failure tolerant for critical functions (Multiple motors each side? Parallel ESCs?).

8) Indicators provided (LEDs) to assist in troubleshooting and operational verification. Critical indicators must be visible through binoculars from several angles.

After some preliminary back and forth brainstorming, Chewy from Team Tiki popped in with a rock solid basic parts list:

- Four NPC T-64 gearmotors
- Four Victor 883 ESCs
- Two Hawker/generic SLA batteries
- Steel frame
- Cheap, simple, and robust

Within 15 minutes of seeing this list, I was on The Robot MarketPlace (**www.robotmarketplace.com**) — my favorite vapor bot "pick list" site. I recognized that buying new might not be possible, given the limited resources of a rural fire department, but it seemed like a good starting point. Out of that exercise came the following design "concept." These are ugly as original sin and twice as smelly, but remember we're being honest here. We've all done it, even if we don't show them in public. So here, using PAD software (Powerpoint Aided Design) is what I came up with. (I added two batteries, a master switch, hubs and wheels, and radio system to Chewy's basic list). I then



PAD Design.

CAD Design.

used my version of CAD (Clip Art Design) to build a really cool graphic, since I thought it might look better in *SERVO*! (Actually, I swiped all the graphics off the Robot MarketPlace also).

If you do the math or want to check mine, you'll see the soft creamy inside parts that need to be steel encased, plus the sticky-outey parts (technical term for hubs and wheels) come to a total of $2,240. Adding in the radio system, it rounds out at about $2,470.

On the one hand, that's not bad. On the other, cheaper is better. The gang threw around several cost reduction options. The first is to just equip the bot with two motor setups instead of four, with gear/chain connections to the other wheels, or have two driven wheels and two idlers. This would knock about $800 off the top. Another, of course, is to buy used not new, or get donations. Most combat robot motors in the big classes are really hardened wheelchair motor/gearbox combos. A quick scan of eBay showed that various types can be had for under $50 a pair, again pulling $800 out of the equation for a four motor bot. It should also be fairly easy to obtain donations from local wheelchair service stores for this kind of motor, gearbox, and hub. Another cost savings is to go with hardware store wheels and tires, saving at least $200.

A different point of view was expressed, noting that the T-64 gearmotors have an offset shaft. Since bot fighters tend to think invertable, I had defaulted to mounting the axles dead center in the box, requiring 14" wheels. Rotated 90 degrees, all of a sudden 10" wheels, lighter and cheaper, become a possibility. Another vigorous discussion was held about using tracks instead of wheels. This is certainly an option, with both home brew and commercial systems possible. I decided for this first pass to just go with simple wheels, until a track hero emerges to help. Also, for this first pass, we used the expensive components and centerline axle locations.

The next job — doing up a simple CAD (using a real program) — was grabbed by Charles Guan, a previous contributor to Combat Zone. He took my "drawings" and converted them into a simple tubular steel frame, able to be welded by any shop, including a good high school or home workshop.

He designed it using 1" square tubing, 1/8" walled. There are only a few holes to drill to make the basic frame, the rest is all welded. He figures the basic bot with frame, motors, batteries, and wheels to be around 150 pounds. Besides the weight reductions mentioned above, he also figures that 1/16" wall tubing could be used instead, reducing everything to about 110-120 pounds if durability isn't affected.

Each battery bay can house one sealed lead acid battery and the drive control electronics for that side. The center of the bot has a clear 6" channel (or more, if the end user wants to make the end frames wider) for mounting things or for additional structure. The width could also be reduced by nearly that amount, if it becomes an issue. To meet the needs of our rural fire department for hazmat purposes, the whole bot would need to be skinned in steel or aluminum, sealed against vapors and liquids, and provided with hard points for mounting sensors or equipment.

The next step in this project is to refine the design, including wiring and component mounting, chase down the parts, hook up with builders, and make it happen! The thought was to produce a basic bot design that could be built and adapted by public safety agencies anywhere, for 1/10th to 1/4 the cost of commercial units. We plan to post the plans as open source information, and work with national fire and law enforcement organizations to spread the word about this capability. We would love to hear from other small public safety departments about their needs, and their interest in participating in this project. **SV**

Please email the author at Legendaryrobotics@gmail.com if you are interested in helping turn this vapor bot into something real.

EVENTS

Results and Upcoming Events

Results Feb 11 — Mar 10, 2008

Motorama 2008 was held by North East Robotics Club, in Harrisburg, PA on February 15-17, 2008. Results are as follows:

NERC www.robotconflict.com

● *Fairy* — 1st: P150, Overvolted robots.

● *Ants* — 1st: Black Death, Enigma Robotics; 2nd: Dolla Dolla Bot, Mad Scientist; 3rd: Absolutely "Naut"-VDD, Team Anarchy Robotics.

● *Beetles* — 1st: Yeti, Mad Scientist; 2nd: Aggravator, Dreadfully Wicked Robots; 3rd: Pressure Point, Team JandA.

- *12-pounders* — 1st: Surgical Strike, Team Rolling Thunder; 2nd: Mephit, NovaRobots; 3rd: Blunt Instrument, Team Rolling Thunder.

- *30-pounders* — 1st: Power Of Metal, Team EMF; 2nd: Gloomy, Terror; 3rd: Sloth, Massacre Robotics.

- *Sportsman's* — 1st: Mangi, Team Half Fast Astronaut; 2nd: Upheaval, Mad Scientist; 3rd: Gnome Portal, Robotic Hobbies.

- *Special Awards* — Best Driver: John Durand, Team Anarchy Robotics; Most Aggressive Driver: Al Kindle, Team Half Fast Astronaut; Coolest Robot: Herald, Robotic Hobbies; Best Engineered Robot: One Fierce Low Ryda, Fierce Robots.

The BotsIQ Boston Regional was presented by BotsIQ Boston in Plymouth, MA on March 8, 2008. Results were not available at press time.

Roaming Robots held an event at the Barnsley Metrodome on February 17, 2008. "For the first time, we headed up north to Barnsley at their Metrodome Leisure Complex to put on two action-packed, fun-filled shows. Once again, we had it all from robots being flipped out of the arena, fire in the arena, arena hazards doing their duty, and also an added bonus of Star Wars popping in to keep the audience in check in between the fights." Well done to the two heavyweight winners Envy and Kan-Opener, and the two feather weight winners Pain in the Asp and Cyberon.

Upcoming Events for May-June 2008

ComBots Cup III will be presented by ComBots in San Mateo, CA May 2-4, 2008. Go to **www.robogames.net** for more details. This is the Heavyweight Championship!

Roaming Robots will hold events at Fenton Manor, in Stoke On Trent on May 4, 2008, in Kingsbury at the Aylesbury Town Centre, on May 11, 2008, and at the Guilford Spectrum in Parkway, Surry on June 15, 2008. Go to **www.roamingrobots.co.uk** for more details.

Robots Live will hold London MCM Expo on May 24-25, 2008 at the London ExCeL Centre. Go to **www.robotslive.co.uk** for more details.

Carolina Combat Robots will hold their CCR Memorial Day Qualifier in Greensboro, NC on May 24, 2008. Go to **www.carolinacombat.com** for more details. This will be an East Coast Nationals Qualifier! Carolina Combat Robots is having its third event in Greensboro, NC. The arena is a 16 foot x 32 foot steel structure with 1/4" steel floor and 1/2" of Lexan for the walls. The event will include Robots from 150 g Fairyweight to the 120 lbers.

RoboGames 2008 will be presented by ComBots in San Francisco, CA, June 12-15, 2008. Go to **www.robogames.net** for more details. The fifth annual International RoboGames invites you to join thousands of robots from around the planet. ComBots and The Robotics Society of America (RSA) invite you to the largest robot show in the world. All major robot competitions will be held over the four-day event. Last year's event included teams from 20 countries — this year, we want 50 countries! **SV**

EVENT REPORT

WAR in Seattle

● by Rob Farrow

In February 2008, Western Allied Robotics (WAR) made its second appearance at the North West Model Hobby Expo. Racing R/C cars, along with plane and helicopter demos, competed for attention with a particularly destructive and memorable robot combat competition. WAR's arena was expanded to 16' x

Itsa and MeltyB(eetle).

12' allowing plenty of room for the robots to maneuver and for fans to watch the action. Teams from California, Oregon, Washington, and British Colombia attended the event. The core of the competition was in the 3 lb (beetle weight) and 12 lb (hobby weight) classes with a few 1 lb and 30 lb robots rounding out the show.

Heavy hitters in the 3 lb class included WAR's returning champ, "Hurty Gurty" by Team Death by Monkeys, the always dangerous "Altitude" by Team Velocity, and "Itsa," a home-plate shaped robot with a giant titanium blade created by Mike Daniels of Team Bad Bot and driven by Greg Schwartz of team LNW. In its first competition, Rich Olson's one-wheeled robot MeltyB(eetle) was sporting its unique drive system that allows it to use 100% of its mass in its spinning body, giving it the potential of hitting harder than any of the other bots at the event.

Early in the competition, Melty B(eetle) fought Hurty Gurty. As the robots came together, Hurty Gurty was thrown to the ceiling as Melty B(eetle) bounced off the walls. MeltyB(eetle) appeared to be getting the best of Hurty Gurty but started to smoke and was knocked out. Hurty Gurty won the fight but was effectively knocked out of the competition with its 1/4" thick aluminum sides being badly bent, disabling the weapon.

After moving up the losers bracket, MeltyB(eetle) went up against the under-cutter Itsa. Bright white sparks shot from the front end of Itsa as its titanium blade skidded across the steel floor. Like a boxer with a long reach, Itsa kept MeltyB(eetle) from getting close enough to deliver a knockout blow. Itsa chipped away at MeltyB(eetle)'s one wheel for the knock-out win.

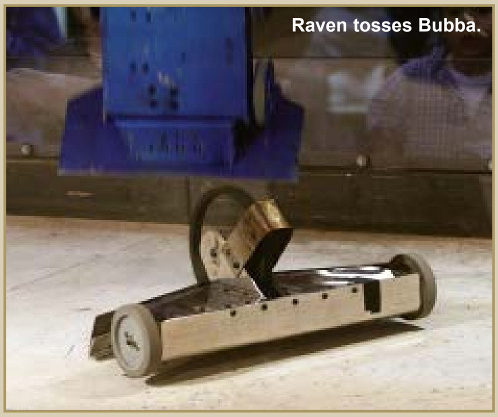
In the beetle weight final, Itsa was matched up against Altitude. The two bots fought earlier with Altitude getting the win. Altitude has a large vertical spinning disk with its sides made of heavy aluminum and titanium. Again, Itsa used its long reach to an advantage. Itsa took out the support skids holding Altitude's weapon off the ground. With no way to keep the weapon off the floor, Altitude threw itself around the arena until time expired in the match. With a unanimous judge's decision, Itsa took home 1st place.

The returning champs from the Canadian team DMZ brought two robots in the 12 lb class. Their powerful wedge Death Dealer driven by Amir Marvasti went up against the always destructive horizontal spinner Fiasco driven by Kevin Barker. Death Dealer used an effective anti-spinner attachment on the front of its wedge. The composite spring steel and rubber material damped Fiascos blows enough to give the wedge the victory.

Shag, a plow-bot driven by Rob Purdy of Team Guasswave pushed its previous opponents around like toys with the aid of a powerful magnet to increase its pushing power. Shag's opponent Raven has a wide wedge with a powerful vertical spinning disk. In Raven's previous fight against the titanium wedge Bubba, driven by Joe Murawski of Team X-Bots, it had some spectacular hits as its spinning disk tore up Bubba's armor. Shag dominated Raven early on but a strong hit by Raven's weapon knocked

Mission Control lifts an opponent.

Raven tosses Bubba.



Event Results

- *12 lb Hobby Weight* — 1st: Death Dealer, Team DMZ, driven by Amir Marvasti; 2nd: Raven, Team DMZ, driven by Michael Clift; 3rd: Shag, Team Guasswave, driven by Rob Purdy.
- *3 lb Beetle Weight* — 1st: Itsa, Team Bad Bot, driven by Greg Schwartz; 2nd: Altitude, Team Velocity, driven by Kevin Barker; 3rd: Hurty Gurty, Team Death by Monkeys, driven by Rob Farrow.

Shag upside down making its plow useless, giving Raven the win. The 12 lb final ended on a bit of a down note as Death Dealer and Raven were slated to fight but due to technical difficulties due to a battery being plugged in backwards, Raven forfeited giving Death Dealer its second competition win in a row. For info on upcoming events and the WAR organization, check out **www.westernallied robotics.com**. SV


Team DinoBots.

ROBOT PROFILE

TOP RANKED ROBOT THIS MONTH

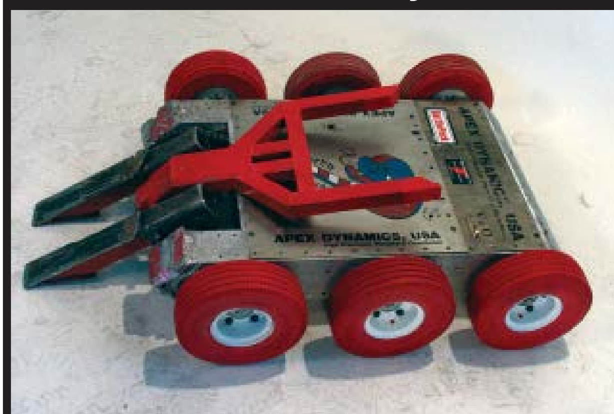
● by Kevin Berry

Top Ranked Combat Bots

History Score			Ranking		
Weight Class	**Bot**	**Win/Loss**	**Weight Class**	**Bot**	**Win/Loss**
150 grams	VD	26/7	**150 grams**	Micro Drive	7/1
1 pound	Dark Pounder	44/5	**1 pound**	Dark Pounder	28/3
1 kg	Roadbug	24/10	**1 kg**	Underkill	11/5
3 pounds	3pd	48/21	**3 pounds**	Limblifter	12/1
6 pounds	G.I.R.	14/2	**6 pounds**	G.I.R.	8/2
12 pounds	Solaris	42/12	**12 pounds**	Rants Pants	11/2
15 pounds	Humdinger	26/4	**15 pounds**	Humdinger	26/4
30 pounds	Helios	31/6	**30 pounds**	Billy Bob	9/2
30 (sport)	Bounty Hunter	9/1	**30 (sport)**	Bounty Hunter	9/1
60 pounds	Wedge of Doom	43/5	**60 pounds**	Texas HEAT	8/3
120 pounds	Devil's Plunger	53/15	**120 pounds**	Touro	5/0
220 pounds	Sewer Snake	35/9	**220 pounds**	Brutality	4/0
340 pounds	SHOVELHEAD	39/15	**340 pounds**	Psychotic Reaction	4/1
390 pounds	MidEvil	28/9	**390 pounds**	MidEvil	3/0
History Score is calculated by perfomance at all events known to BotRank			Current Ranking is calculated by performance at all known events, using data from the last 18 months		

Rankings as of March 8, 2008

Sewer Snake – Currently Ranked #5



Historical Ranking: **#1**
Weight Class: **220 lb Heavyweight**
Team: **PlumbCrazy**
Builder: **Matt and Wendy Maxham**
Location: **Sacramento, CA**

BotRank Data	Total Fights	Wins	Losses
Lifetime History	44	35	9
Current Record	5	3	2
Events	16		

Sewer Snake has competed in ROBOlympics/RoboGames 2007, ComBots Cup II/Maker Faire, Game Developers Conf, February Fun Fest, ComBots Cup I, 2005 RFL Nationals, Robotic Revolution — New Orleans, Battle Beach 3, RoboGames 2005, War-Bots Xtreme Premier, 2004 RFL Nationals — Open, 2003 Triangle Series Nationals, Steel Conflict 4, Steel Conflict 3 , and RC Expo. It also fought as a Super Heavy at ROBOlympics 2006. Details are listed below:

- *Configuration*: Six-wheel drive, modular lifter.

- *Drive ESC*: IFI Thor 883s (two per drive motor).

- *Drive motors*: S28-400 Magmotor on Apex Gear box (one per side).

- *Drive batteries*: 3,000 Ah NiCad Battlepacks (5-6 for both drive and weapon).

- *Weapon*: Shock-mounted modular lifting system including: 3/16" Abrasion Resistant (AR400) steel wedge for spinners; two steel spikes for wedges on the front, and a secondary lifting fork on top of the robot.

- *Weapon power*: Chain driven, utilizing ANSI 50 chain.

- *Weapon motor*: S28-150 Magmotor on Apex gearbox.

- *Weapon ESC*: IFI Thor 883.

- *Armor*: Three bright red tires per side, Al on back, lifting weapon up front, .060" titanium on top.

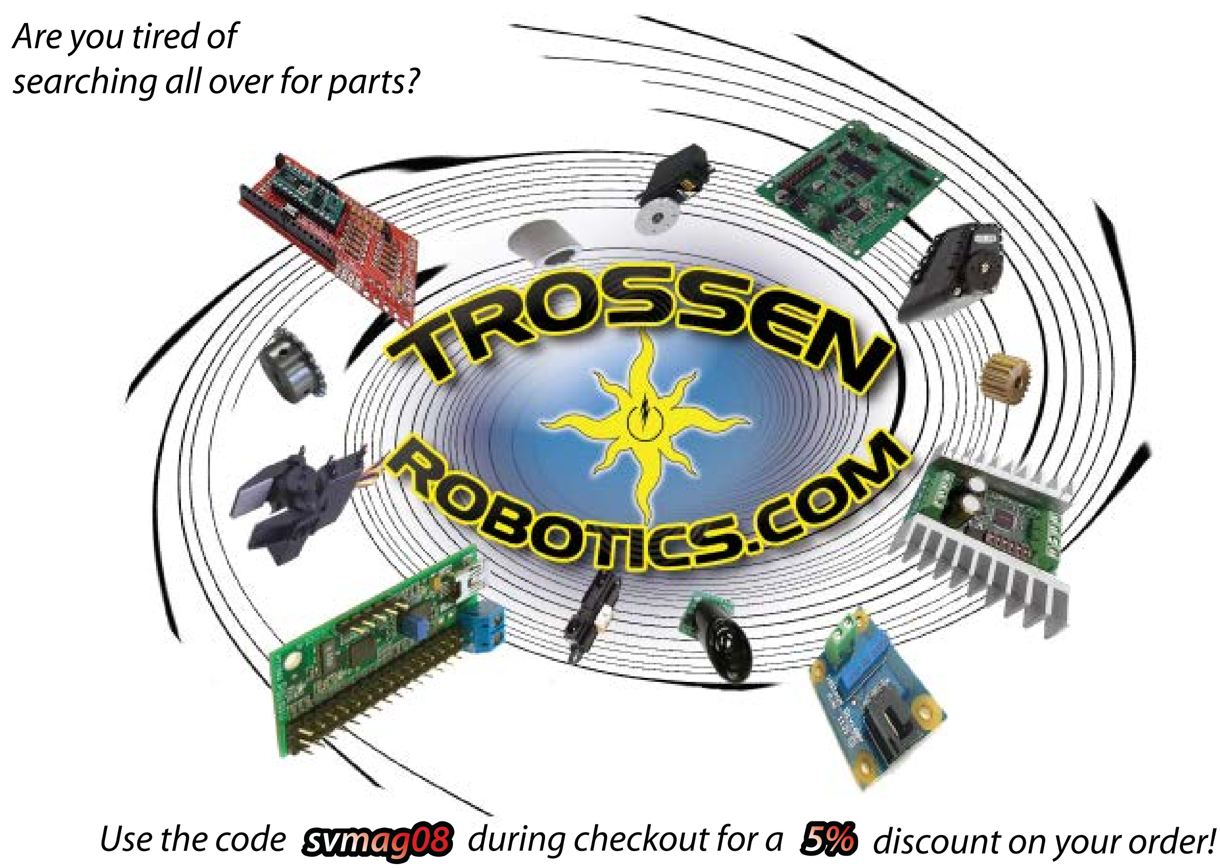
- *Radio System*: 2.4 GHz Spektrum.

- *Future*: Keep running this frame until it falls apart ... then, who knows?!?!?

- *Design philosophy*: Kiss! "Keep It Simple Stupid." We like a bot that will keep going, and is easy to fix when it stops going!!! Don't try to kill your opponent, just try to control them and redirect their energy. Let your opponent kill themselves on the arena.

- *Other tidbits worth noting*: Inducted into the Combat Robot Hall of Fame in 2007 **http://members.toast.net/joerger/hall_of_fame.html**; BotRank's historically ranked #1 HW since 2005; the six-wheel drive version of Sewer Snake debuted at the HW Open tournament of the 2004 Nationals and has been in 45 competition matches with only seven losses in nine tournaments; Sewer Snake has also been in 11 exhibition matches in both the US and UK. **SV**

Photos and information are courtesy of Team PlumbCrazy. All fight statistics are courtesy of BotRank (**www.botrank.com**) as of March 8, 2008. Event attendance data is courtesy of The Builder's Database (**www.buildersdb.com**) as of March 8, 2008.



Introducing Our New Hardware.

Due to overwhelming demand from customers, we've created yet another new part for the VEX Robotics Design System.

However, it is not for sale.

The only way to get this rare item is to earn it through competition. When we created the VEX Bridge Battle game and offered it for free on our website, we never imagined that events would spring up all over the globe. Much less that event organizers and competitors would insist on a playoff to determine the best in the world.

Now, through the generous support of Autodesk, NASA, FUTURE Foundation, CREATE Foundation and others, we're pleased to present the VEX Robotics Competition World Championship to be held May 1-3, 2008 on the campus of California State University Northridge in Los Angeles.

If you're concerned about missing your only chance to take home our special new hardware, don't worry. A coalition of partners are launching hundreds of local, state, national, global and even virtual low cost VEX Robotics Competitions.

Visit www.RobotEvents.com and www.VexRobotics.com for more information on organizing or participating in an event.

Perhaps you have what it takes to get your hands on the newest VEX hardware.

VEX ROBOTICS WORLD CHAMPIONSHIP

Autodesk

NASA

VEX EDUCATIONAL ROBOTICS EDR

California State University Northridge

Innovation First A product of Innovation First.

www.vexrobotics.com

Copyright 2008. Innovation First, Inc.

TEAR DOWN

by Bryan Bergeron

The iRobot Looj

The treaded, weatherproof, remotely controlled Looj from iRobot is intended to facilitate the dull, dirty, and often dangerous job of clearing gutters of leaves, pine cones, twigs, and other light debris. At $99, I couldn't resist exploring the potential of the Looj as a robotics platform. Following is a tear down of the Looj and wireless controller.

SPECIFICATIONS

The Looj is an elongated tank geared for power, as opposed to speed. Unlike most inexpensive robotics platforms, the Looj is weather-resistant. You can run it through a wet gutter or puddle and then hose it off. The motors, electronics, and removable 7.1V NiCad battery pack are sealed from the environment. On the downside, turning is out of the question. Movement is limited to linear forward and reverse and the powerful front auger doesn't seem obviously useful for tasks other than clearing or drilling a path. Given these caveats, the Looj has definite potential.

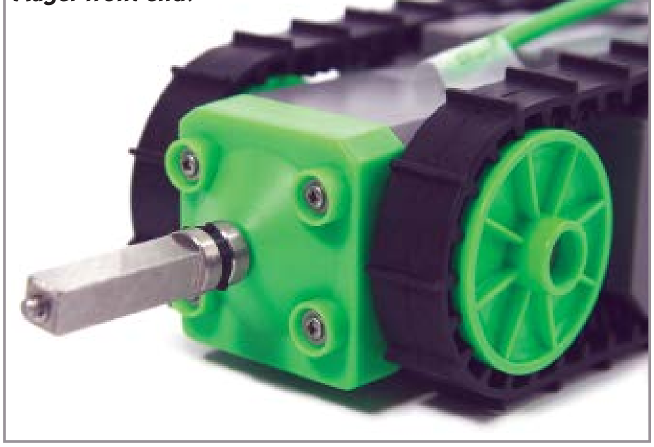
The basic specifications of the Looj, based on my measurements, are summarized in Table 1. A third of the total unit weight of about three pounds is due to the battery. Despite the weight, the robot moves at a respectable nine inches per second. It isn't a stealth platform, but sounds like a power drill when either the auger or drive motor is engaged. Current drain — 300 ma with drive activated and an additional 500 ma for the auger — was measured with a bench power supply set to 7.1V, connected to the Looj using a hand-held cord with the battery pack installed in the compartment. Radio control range — a very modest 57 feet — was measured with the Looj in an empty parking lot with the controller held at waist height. The indoor range test was conducted with the robot and controller separated by a sheetrock wall.

Unfortunately, charging the battery involves removing the water-tight compartment lid (two screws), removing the battery pack, and connecting it to the micro-sized wall charger. Given the 'dumb' charger provides only 150 mA, charging the NiCad pack requires 15 hours, and the charger doesn't have an auto-shutoff feature.

CONSTRUCTION

Not obvious from the specifications is the ruggedness of the Looj. This isn't a typical carpet roamer robot. I didn't attempt it, but I'm convinced that it would survive a fall from roof height onto a grassy surface. In addition — although I didn't measure the torque of the auger tip or drive — I couldn't stop the auger tip or the rear wheels with my bare hands. Think cordless power drill for both the auger tip and the drive mechanism.

In other words, while this power may be a boon to your development plans, small fingers should not be



Auger front end.

Dimensions	15.5" L x 3.25" W x 2.2" H
Drive	Rear (opposite end of auger)
Tread	Rubber, one piece, 1/2" wide
Length, auger tip (total)	1.9"
Dimensions, square auger shaft	1.2" L x 0.3" square
Length, antenna	13.5"
Weight, body (no battery)	23 oz
Weight, NiCad pack	12 oz
Charger	9 VDC wall module, 150 mA
Speed (forward/reverse)	9 in/second
Control range, outdoors	57 feet
Control range, indoors	30 feet
Battery	7.1V NiCad; capacity not listed
Current drain @ 7.1V	Drive only — 300 mA Auger only — 500 mA Auger and drive — 800 mA
RC frequency	49.86 MHz
Controller weight (no battery)	5 oz

TABLE 1. Looj robot specifications.

allowed around the rear wheel mechanism. Because the bright green plastic design is inviting, it's a good idea to treat this as you would any of your power tools — up and away from children.

THE TEAR DOWN

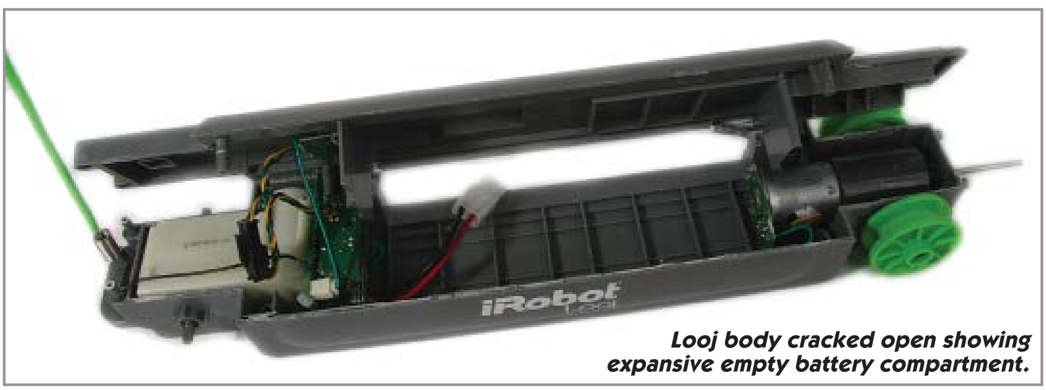
The rubber tread peels off easily, and the rear wheels are attached with Phillips head screws. After that, peering inside the Looj takes a bit of work. Unfortunately, iRobot elected to glue the cover in place. I used a sharp screwdriver to break the seal and then remove the cover. The most challenging section of the cover is over the thin bridge framing the battery compartment.

Once the cover is removed, the electronics and drive assemblies are readily accessible. The front auger motor and gearbox are easily extracted without tools. The rear compartment is more complicated. The electronics — located adjacent to the rear drive gearbox — are also easily removed with a gentle tug. Fortunately, connectors are used for power, motors, and the on-off switch. The antenna is the only wire that must be cut or unsoldered to extract the electronics unit.

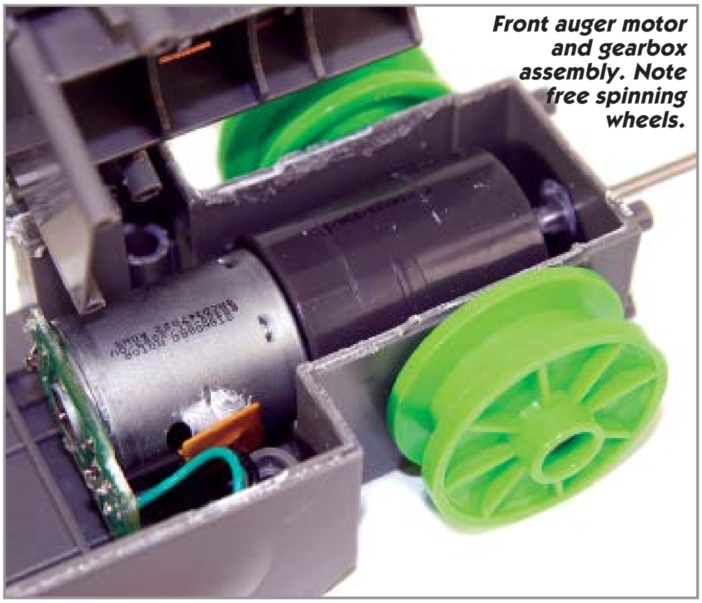
The electronics assembly consists of a motherboard and two solder-on boards. The motherboard appears to be primarily for power management and input/output. One board is obviously the RF receiver, given the RF coils and the antenna connection. The other board is apparently the motor controller. One of the two 14-pin chips on the controller board is a 74HC74D flip-flop and the other is an LM339DG quad comparator. The balance of semiconductors are three-lead devices and diodes.

The rear drive motor assembly is best left inside the Looj body. The axel runs through the gearbox and out of the Looj body, making extraction problematic. I used modest force in an attempt to drive the axel through the gearbox, but encountered stiff resistance. Instead of possibly destroying the gearbox, I used a Dremel to saw through the body of the Looj to release the gearbox. The complete electromechanical system is

Rear drive mechanism and antenna mount.



Looj body cracked open showing expansive empty battery compartment.

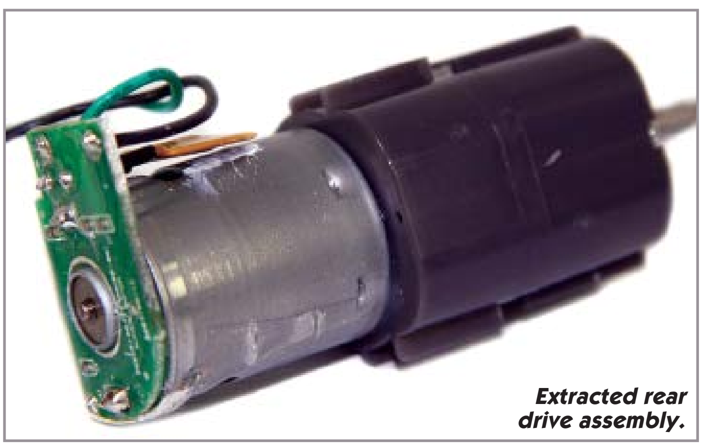


Front auger motor and gearbox assembly. Note free spinning wheels.

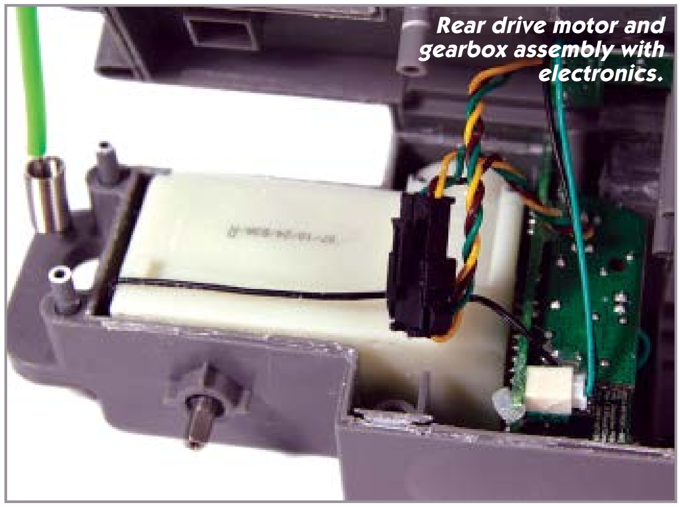
compact and relatively lightweight. The largest and heaviest component of the system is the NiCad battery pack.

The controller features an on-off switch and momentary on buttons for forward and reverse and auger direction. Like most power tools, the forward/reverse switch requires constant pressure for activation. The electronics of the controller — accessible by removing a couple Phillips

Extracted auger drive assembly.



Extracted rear drive assembly.



Rear drive motor and gearbox assembly with electronics.

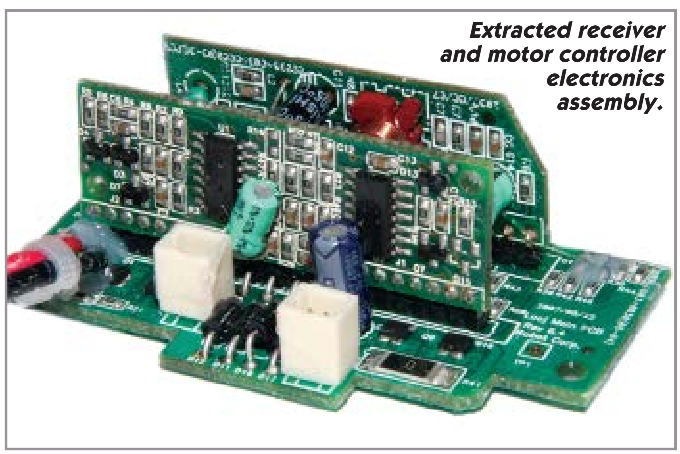
head screws — feature a 49.86 MHz crystal and 26 inches of antenna stuffed into the head of the controller. This unorthodox antenna design may account for the relatively short range of 57 feet. An uncoiled antenna, similar to that used on the robot body, would certainly provide better range.

FROM HERE

The Looj robot presents several opportunities for the robotics innovator. One is to leave the unit intact and build on it. Fortunately, there's a lot of unobstructed real estate on the top of the Looj. I noticed that the body happens to be just wide enough to hold a BASIC Stamp 2pe motherboard — my latest favorite controller from Parallax.

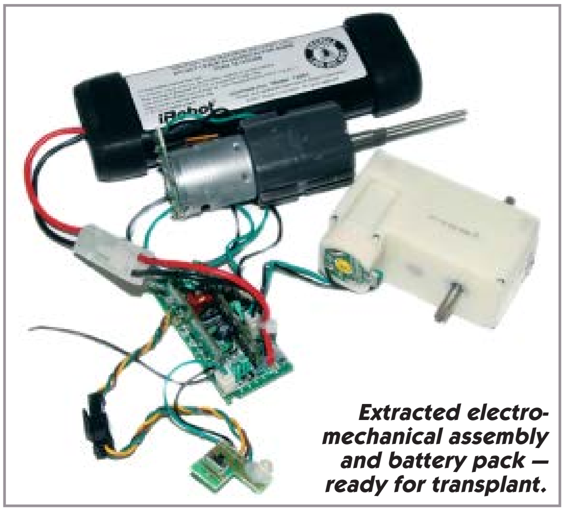
If you decide to use an unmodified Looj as a development platform, then consider replacing the NiCad battery pack with a more significant NiMH pack; 3.8 amp-hour NiMH packs (that's 12 hours of running time at 300 mA) with the same connector used on the Looj are available at numerous hobby shops on the web. If you opt for a NiMH pack, then replace the 15 hour NiCad charger with a quick NiMH charger.

An obvious question is whether the Looj is more valuable as parts than as an intact platform. For $99,

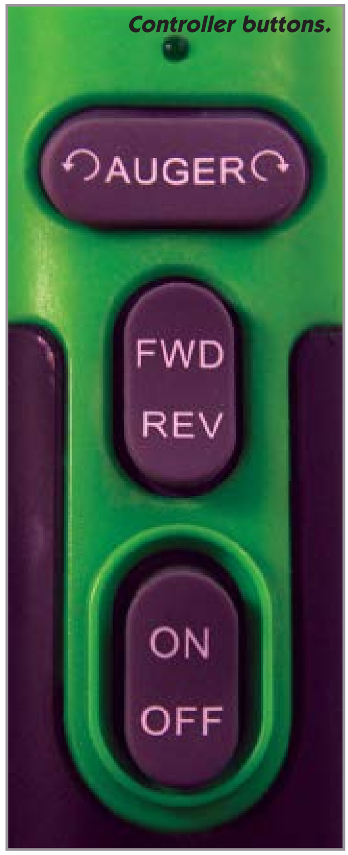


Extracted receiver and motor controller electronics assembly.

you get an R/C unit, controller, two motor and gearbox assemblies, and a modest battery power supply. You could replace the auger assembly with a steering servo to create a more nimble platform. However, given the rugged, weatherproof construction, I've opted to keep the base intact — after repairing the plastic frame with epoxy — and replace the front end with electronics. If you decide to use the Looj as a platform for your projects, please send in your results to share with other readers. **SV**

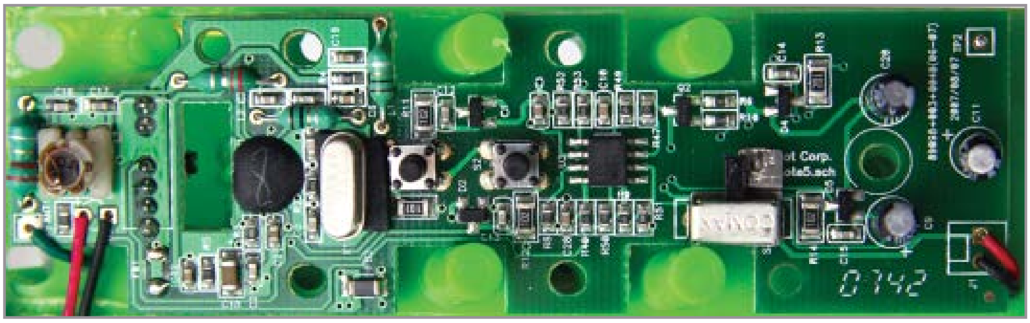


Extracted electro-mechanical assembly and battery pack — ready for transplant.



Controller buttons.

Controller electronics.



The Servo Buddy

by Jim Stewart

FIGURE 1

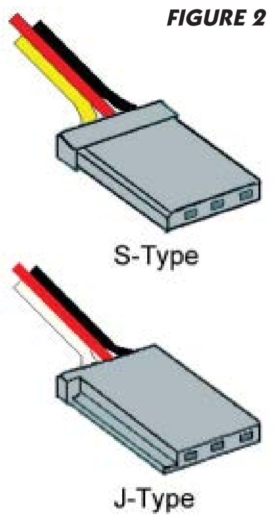
This article introduces servo motor construction and operation, and describes an inexpensive circuit you can build to control a servo without a microcontroller. When I first started to build projects with R/C servo motors it became clear that, during construction, I needed a way to set the position of a servo manually. You can't just grab the shaft and turn it, and writing software for a micro was overkill. Just a simple little circuit would do the job. That was the birth of the Servo Buddy. First, let's review some basics.

RC Servos

An R/C servo (or just servo) is an electro-mechanical device used to rotate an actuator to a precise position and hold it there, even if the actuator is pushing back. R/C stands for Radio Control since originally these servos were used for radio control of model airplanes. Standard ranges of rotation are 90 degrees and 180 degrees. Figure 1 shows a Hitec model HS-5645MG servo. Note the three wires near the actuator. They are power, ground, and input. While servos made by Hitec and Futaba are very popular, you will find servos from other companies, especially from China.

Servos come in two basic types: the original analog type and the newer digital type. Both types look similar from the outside and also have the same basic parts on the inside. The difference between analog and digital servos is in the electronics. Digital servos contain a microprocessor.

FIGURE 2



Connectors

Figure 2 shows a Hitec S-type connector and a Futaba J-type connector. They are almost the same except for a polarizing key along the edge of the Futaba connector. They mate with standard 0.025 inch square pins on 0.1 inch centers. The sequence of wires is the same, but the colors differ. Voltage (typically +5V) is the center red wire. Ground is the black wire. The input signal is either yellow (S-type) or white (J-type). For other servo brands, the colors may differ.

Parts of a Servo

Figure 3 shows a simplified view of what's inside a servo. A small DC motor is connected to an output shaft through a set of speed-reduction gears. The power of a motor is P = kwG, where k is a constant, w is the rpm, and G is the torque. If power is fixed, then reducing speed will increase torque on the output shaft. The motor is controlled by the electronics. A position command is the input, while a potentiometer on the shaft provides position feedback. The actuator — commonly called a horn — has grooves in its mounting hole that mate with the spline at the end of the output shaft. The spline prevents the horn from slipping under torque. A screw attaches the horn to the shaft. Horns come in various shapes: arms, bars, crosses, discs, etc. Note that the spline on a Hitec servo has 24 grooves while the spline on a Futaba servo has 25. Their horns aren't interchangeable.

Input Signal

The control input is a pulse width modulated (PWM) signal as shown in Figure 4. For an analog servo, the pulses are typically 20 milliseconds apart for a repetition rate of 50 Hz. Digital servos use the same PWM widths but can use a higher repetition rate, up to 300 Hz. The pulse widths shown are common, but other widths are also used. Check the particular manufacturer's datasheet.

Analog Servo Electronics

Figure 5 shows a block diagram of the electronics of an analog servo. The local pulse generator (triggered by the input pulse) generates a pulse width proportional to the current position. The local pulse and the input pulse go to a comparator which subtracts one from the other. The difference is the error pulse. The direction signal depends on which pulse was wider. The error signal goes to a pulse stretcher which, in effect, is an amplifier. So, a 1% difference in width from the comparator can generate a 50% drive to the H-bridge. The H-bridge sets the polarity of the voltage going to the motor according to the direction signal. The percent drive to the H-bridge decreases as the position approaches the command point. To prevent "hunting" (an oscillation around the final position), there is a small dead-band. Once the difference between the command point and position is within the dead-band, the motor drive goes to zero.

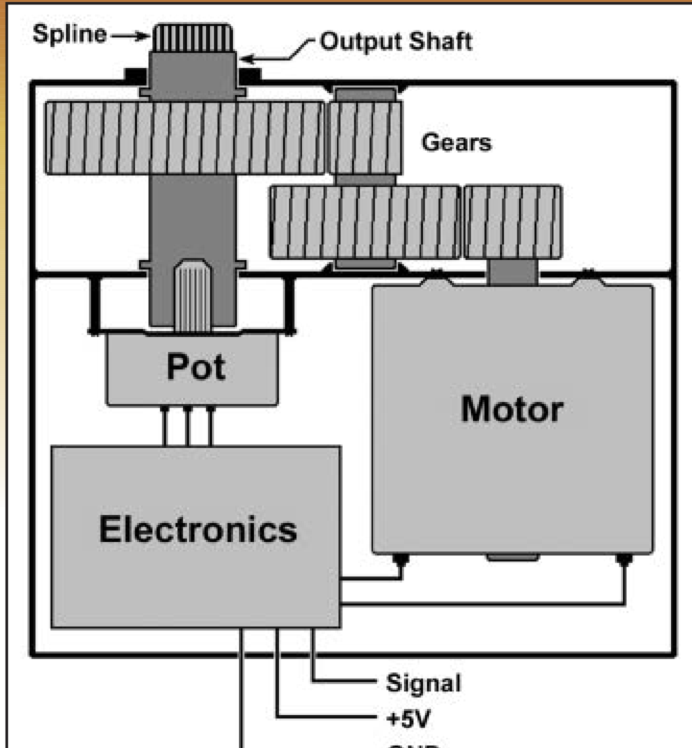


FIGURE 3

Gear Material

Three types of material are used to make gears for RC servos:

• *Nylon*: the most commonly used. Nylon gears are lightweight and run smoothly with low wear, but are at the low end in durability and strength.

• *Metal*: the strongest material. Metal gears are heavy and, due to wear on the teeth, will develop "slop" (looseness) in the gear train. Slop causes loss of position accuracy and sometimes causes a mechanical instability or oscillation under certain types of load. If you have the money, titanium gears offer superior wear resistance.

• *Karbonite*: carbon reinforced plastic (not to be confused with carbonite, which is highly explosive). Hitec's Karbonite™ gears are stronger and more durable than plain nylon, but just as lightweight. They wear better than metal gears, but metal is still the strongest.

Servo Specs

Two important specs for a servo are speed and torque. Speed is specified as how long it takes to rotate through a given angle, such as 0.15 seconds for 60 degrees. Torque is given in ounce-inches (oz-in) or kilogram-centimeters (kg-cm). Speed and torque are given for specific voltages, usually 4.8V and 6V. One factor that affects speed and torque is the bearing surface for the output shaft. Possibilities are plastic, metal sleeve, or ball bearings. Servos vary in size and weight with more powerful servos being bigger and heavier. There are micro, mini, and standard sizes, as well as some "maxi" sizes.

Digital vs. Analog

While digital servos are more expensive than analog models, they offer a lot of advantages. Because they contain a microprocessor, some can be programmed for parameters such as speed, direction of rotation, range of rotation, and dead-band. Or, you can skip the programming and use them as they are right out of the box.

Because they can receive input commands faster than analog servos, digital servos can update the motor position faster. That means faster response, higher starting torque, a tighter dead-band, and more holding torque. The trade-off is that digital servos can draw a lot more current than analog servos. They can kill a battery quickly.

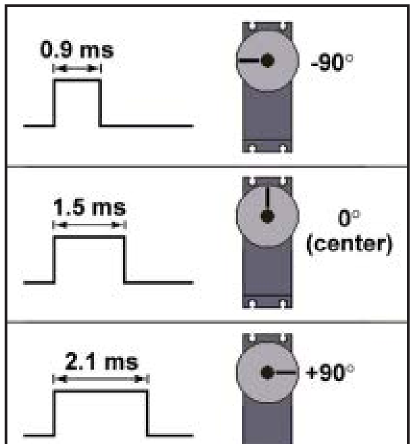


FIGURE 4

Continuous Rotation

Because of the built-in H-bridge and gear train, people sometimes modify servos to rotate continuously to become, in effect, inexpensive gear-head motors. But servos were

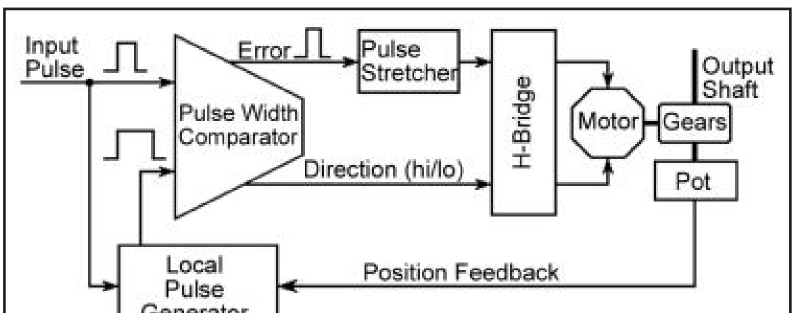
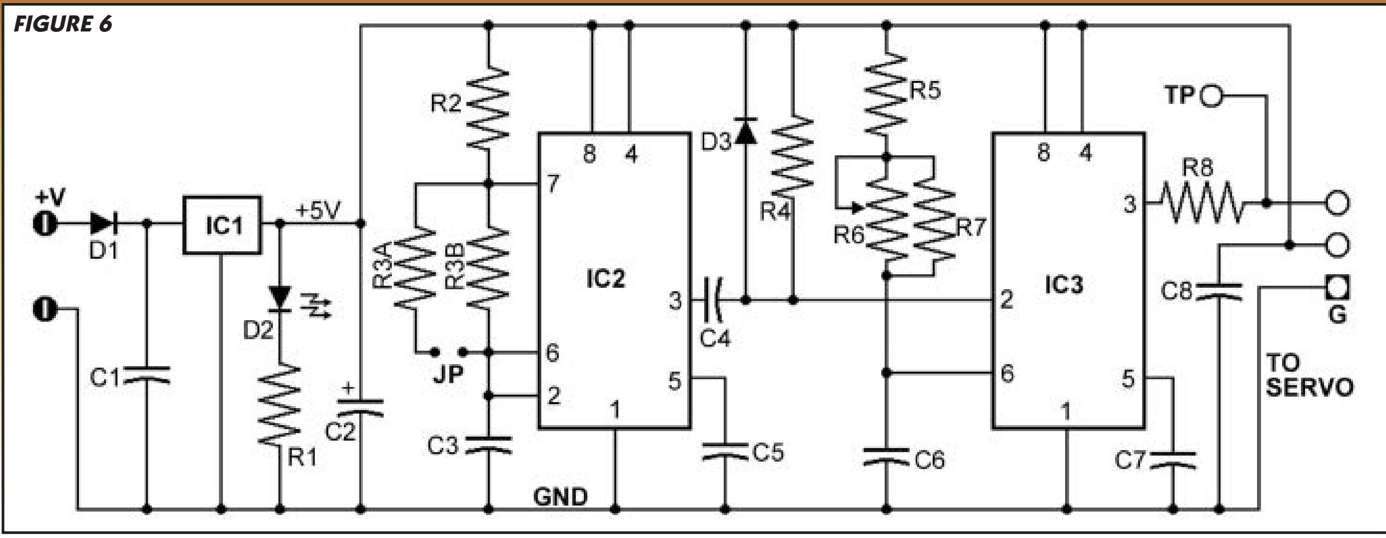


FIGURE 5

FIGURE 6



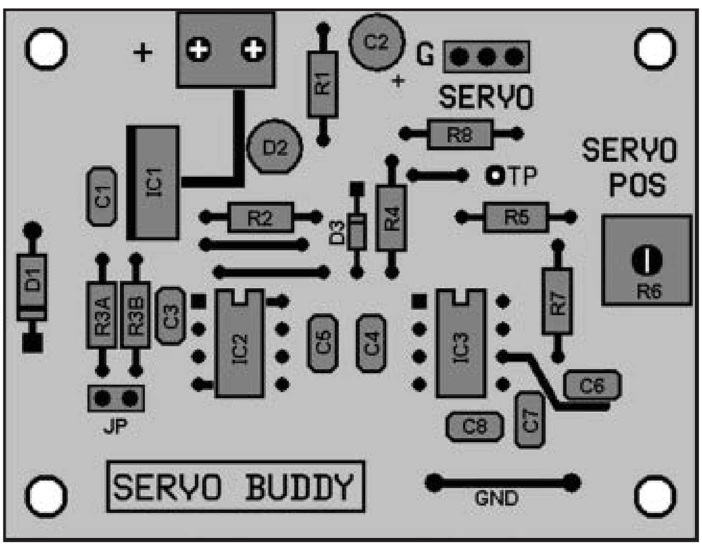
not designed for continuous rotation, and using them that way can shorten their lives. The modification involves rewiring the potentiometer and snapping off the mechanical stops that prevent 360° rotation. If you're interested, there are several sites on the Internet that will show you how to do it.

Now, let's build our buddy.

The Servo Buddy Circuit

Figure 6 shows the schematic. It uses two CMOS 555 timer ICs. The first is an oscillator running at 50 Hz to generate the 20 ms spacing between pulses. The second is a one-shot timer with its output pulse width set by a potentiometer. I used CMOS because bipolar 555s often have a transient internal short of power to ground when they switch. Such transients generate noise in the form of high current spikes. I used 555s because I had a lot of them. You can also use a 556, which is a dual 555 in a 14-pin DIP. Figure 7 shows the parts placement on a printed circuit board (PCB), and Figure 8 is a photograph of the finished unit. Note the loop of wire along the bottom edge. It's soldered to ground as a place to attach the ground lead of an oscilloscope or voltmeter.

FIGURE 7

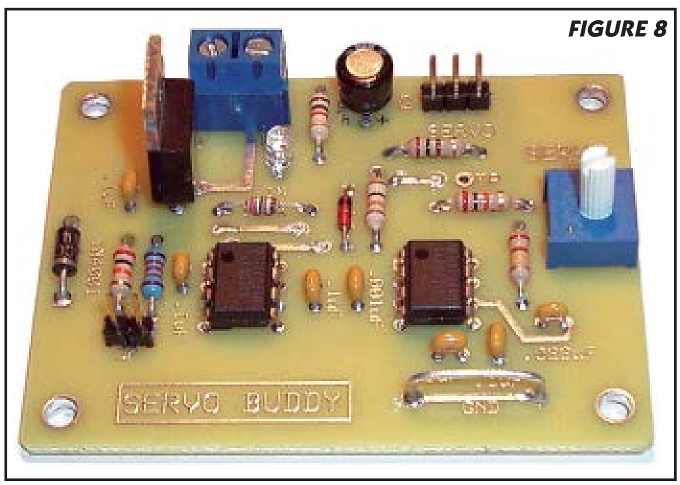


Circuit Operation

The board is designed to be powered by an unregulated DC wall-wart supply; a 9V @ 1A unit should do the job. Power is connected to a two-position terminal block, and D1 protects against reverse polarity. An LM7805 regulates the supply down to five volts and is hefty enough to handle one amp slugs of current. C2 is a 100 µF cap, also to handle high current draw. The servo connector attaches to a three-pin header on the board edge.

The oscillation frequency of IC2 is set by C3, R2, R3A, and R3B. R3A and R3B are connected by a removable jumper (JP). With the JP out, IC2 runs at 50 Hz for analog servos. With JP in, IC2 runs at 250 Hz for digital servos. Digital servos will work at 50 Hz, but having the higher frequency allows them to be tested with a typical input signal. Using 1% resistors for R2 and R3B, the exact frequency depends on the tolerance of C3. Monolithic ceramics can vary ±20% depending on temperature so

FIGURE 8



Need a PCB?

The PCBs and/or a complete kit for this project can be purchased through the *SERVO Magazine* Webstore at www.servomagazine.com.

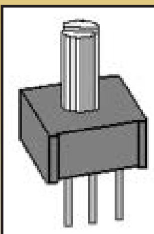


FIGURE 9

either purchase a higher tolerance X7R or temperature compensating NPO or COG types.

Since the output of IC2 is a pulse train, components C4, D3, and R4 form a differentiator to get the negative-going edge required to trigger IC3 in one-shot mode. The output of IC3 goes to the three-pin header, so R8 is there in case the signal pin gets shorted to ground or +5V. The output pulse width of IC3 is set by R5, R6, R7, and C6. R6 is a single-turn pot with a shaft for easy adjustment (see Figure 9). The pot is a no-name brand I bought from Electronix Express (part #18STS100K). It's similar to a Bourns type 3386T.

If you need more precise control of servo position, use a multi-turn pot. R5 in series with the pot and R7 across the pot allow you to set the minimum and maximum pulse widths. The values used here give a range of 0.8 to 2.5 ms. The pulse width can be observed at test point TP.

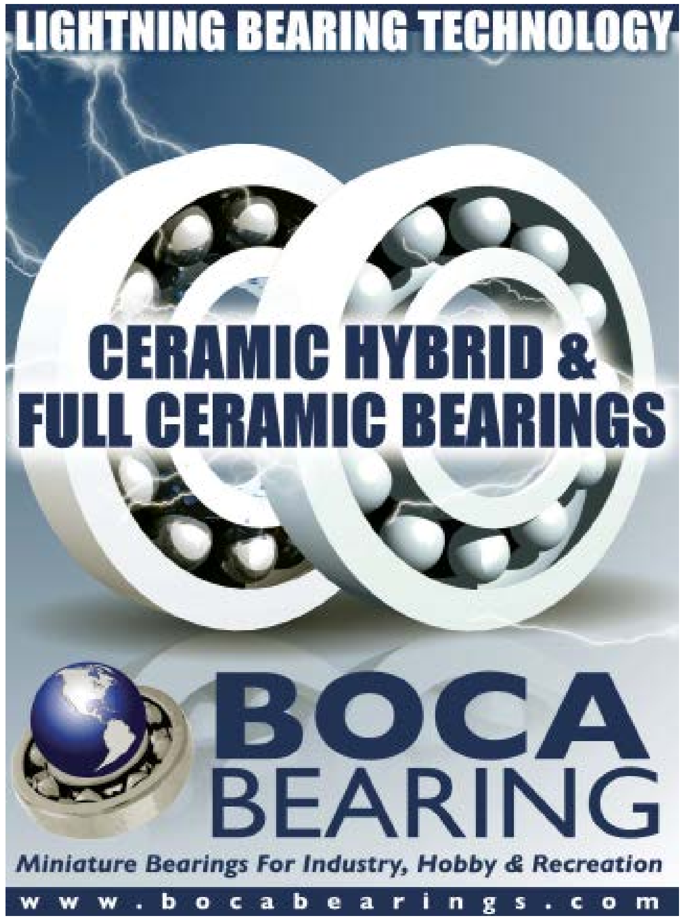
Construction

Construction is straightforward. You will need a breakaway ("snappable") type male header with at least five pins. For the jumper JP, you need two pins and for the servo connector you need three pins. Such header strips are available from distributors such as Jameco, Electronix Express, and others. The circuit is simple enough to build on a pre-drilled proto board like the RadioShack 276-150. **SV**

Parts List

ITEM	DESCRIPTION	ITEM	DESCRIPTION
R1	1K, 1/4W, 5%	C1	0.1 µF, 50V monolithic
R2	22.0K, 1/4W, 1%	C2	100 µF, 10V electrolytic
R3A	20K, 1/4W, 5%	C3	0.1 µF, 50V monolithic
R3B	133K, 1/4W, 1%	C4	0.001 µF, 50V monolithic
R4	10K, 1/4W, 5%	C5	0.1 µF, 50V monolithic
R5	33K, 1/4W, 5%	C6	0.022 µF, 50V monolithic
R6	100K pot (*see text*)	C7	0.1 µF, 50V monolithic
R7	220K, 1/4W, 5%	C8	0.1 µF, 50V monolithic
R8	100 Ω, 1/4W, 5%		
IC1	LM7805	D1	1N4001
IC2,IC3	LMC555	D2	Green LED
		D3	1N914

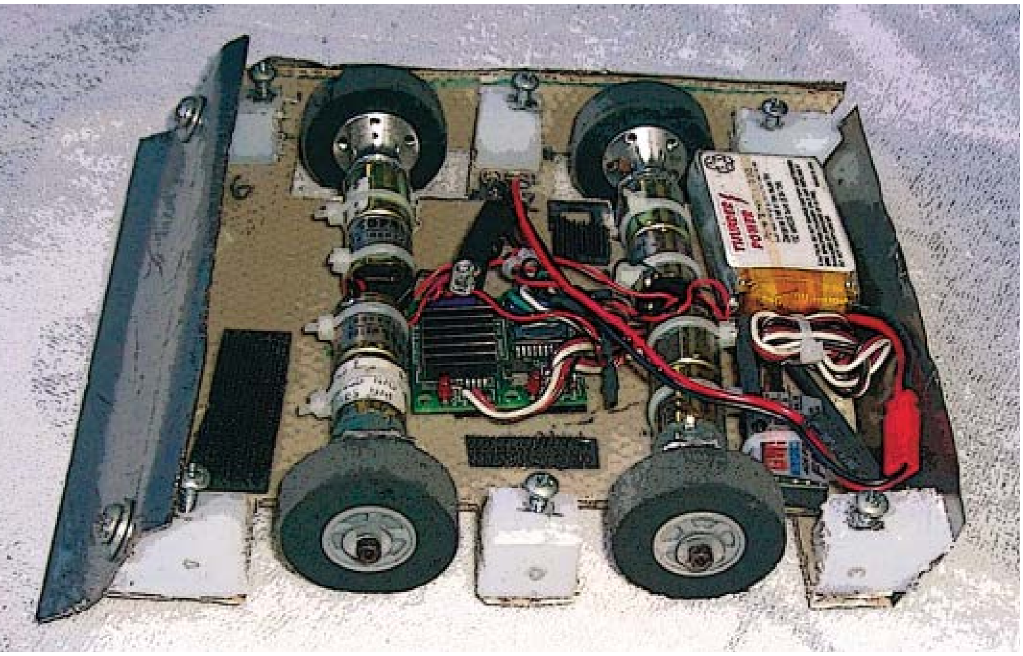
- PCB or proto board
- Removable jumper (Jameco p/n 112416; Electronix Express p/n 24SHRTBAR, or equivalent)
- Male header strip on .100 inch centers, break-away type (AMP type 103185 or equivalent)
- Two-position terminal block (RadioShack p/n 276-1388 or equivalent)



Designing and Building a ROBOT from SCRATCH

Part 3

So far in this series of articles, we have worked our way through the complete design phase. Starting with "I want to build a robot," we developed a goal and created a set of design specifications. We then brainstormed and used decision matrices to decide on a combat robot with a horizontal bar spinner as the weapon — much like an upside-down lawn mower. Next, we chose a frame type and the components along with determining the ideal layout. We completed the design of the robot (shown in Figure 1) and now we are ready to discuss final preparation before the actual construction phase!



by Brian Benson

Before we dive into the actual build of our designed robot, there are a number of things we need to discuss. This will allow us to be on the same page when we get to the actual construction. First, I will explain some common tools and techniques that are useful in the shop. Then, I'll talk about some tricks of the trade that will make your building process easier!

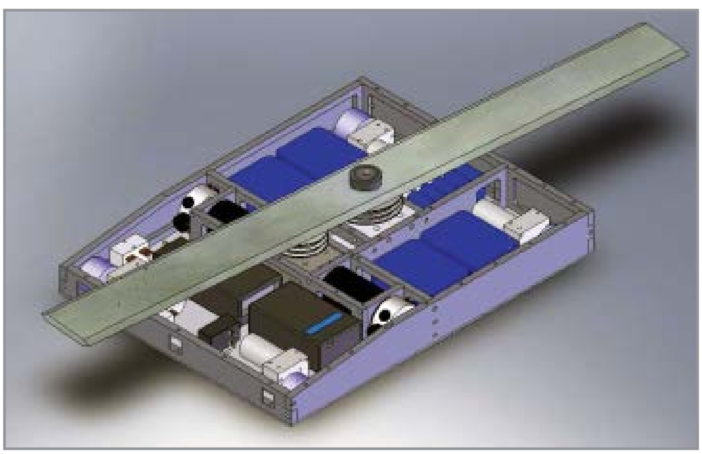


FIGURE 1. Last month, we finalized the design of the 120 lb combat robot using computer aided design (CAD).

Tools and Techniques

Every robot is different, therefore making the build process of each robot unique. It would be impossible to squeeze everything you need to know to build every kind of robot into a book, nevermind a short series of articles. Luckily, there are a few tools and techniques that apply to a wide variety of robotic projects.

Cutting

Nearly every robot requires cutting materials to size in order to build it. There are a number of tools that can be used to do this. Following is a brief overview of different tool options you have and when to use them. These are tools that people tend to have in their workshop. Also, whenever using a power tool, be sure to wear proper eye protection!

Nearly all metals will benefit from using some sort of coolant when cutting. WD-40 will work, but specific chemicals such as TapMagic will work better in most cases. Most plastics cut perfectly fine dry, just be sure not to cut at too high a speed as to melt the plastic (caused by overheating). If you are having a problem with the plastic melting, water makes a good coolant. However, mixing water and metal tools usually leads to rust, so be cautious.

When you are preparing to cut your material, there are a number of things you can do to help make accurate cuts. If you are cutting metal, spray it down with layout fluid, which is a bluish paint-like coating that allows scribe marks to be very visible. It also comes off easily with the correct chemical remover. Then you can use a machinist square and scribe to draw the line. After scribing the line, mark a large "X" on the excess material. Now when you go to cut the material, place the cutting blade such that the blade width is on the side with the X; this will keep your material from being a blade widths' shorter than planned. It will also serve to remind you which part is the excess and which is the actual piece.

Cutting sheet stock is best done with a jig saw or a vertical band saw. Both allow you to make long straight cuts in addition to complicated curves. Just be sure you are using the appropriate blade type and tooth count for what you are cutting. These tools can be used to cut most metal, plastic, and wood.

Bar stock can be cut using a variety of tools but a horizontal band saw is usually the easiest. You can also use a vertical band saw, jig saw, sawzall, and chop saw. Just be aware that an abrasive disk on a chop saw should not be used for anything but ferrous metals (iron/steel). Also, these disks can fragment and explode, so always stand outside the path of the disk and wear full face protection.

Carbon fiber, Kevlar, fiberglass, and other composites require great care when cutting. The dust created when cutting them is very harmful and should not be inhaled. If you decide to work with these materials, be sure to use a proper mask to filter the air you breathe, do it in a properly ventilated area, and wet it to minimize the dust.

Drilling

Drilling a hole can be very straightforward and easy. However, doing it accurately and correctly is a different matter. A hole generally needs to be perpendicular to the surface of the material, placed correctly, and made deep enough. The following is a procedure I use when drilling a hole.

1) First, measure where you want your hole to be and mark it with an X using a scribe.

2) Next, use either a center punch or a center drill to create a pilot hole for the drill bit:

2a) A center punch will make a small dimple at the center of the X by deforming the material under impact. There are two types of center punches. You can use a spring-loaded, automatic center punch which you simply have to line up and push down until it pops, or a manual center punch which you line up and then hit the top with a hammer. Both work equally well but the automatic center punch can be more convenient.

2b) If you require high accuracy, a small center drill is the best way to start — and countersink — the hole you wish to drill. Center drills are unique in that they are very rigid with very little flex so they do not wander.

3) If you are drilling a large number of holes that need to be the same depth, an easy method is to put a "shaft collar" on your drill bit. Usually, this is a round collar that you set at the correct distance up the bit and lock into place via a set screw. If you don't have one of these, then a thin strip of duct tape wrapped around the bit at the correct height will work also. This will give you a clearly defined stopping depth when drilling your holes.

4) Now you are ready to drill the actual hole. If you are drilling a hole with a diameter of 1/4" or less, you can simply use that size drill bit. For larger sizes, I recommend you start with a 1/4" drill and then work your way up in 1/4" increments to the size hole you need. When you drill these holes, be sure that the drill bit is perpendicular to the material. This can be done easily with a drill press as seen in Figure 2.

If you are using a handheld drill, then there are a few tricks. The first is to use two levels or machinist squares to keep the drill upright. Use one for keeping it upright front to back, and the other for left to right. An assistant is a big help while doing this because while you are controlling the drill, they can be continuously checking the orientation of it. If you have many holes to drill, then this method can be very time-consuming, in which case you can make a guide. A piece of metal with the right size hole

FIGURE 2. A drill press is used to make perfectly perpendicular holes in a frame rail.

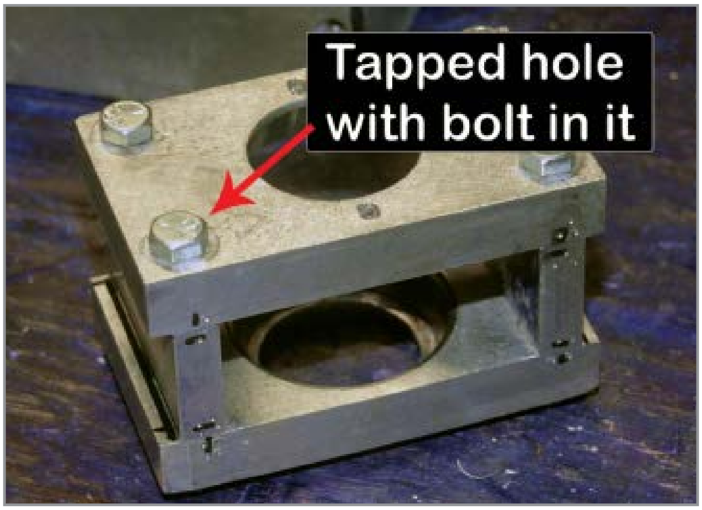


FIGURE 3. This is a very good example of appropriately using a tapped hole. Notice the top and bottom plates both screw into the uprights. This allows for quick assembly and disassembly.

in it (the same size as you are trying to drill) will serve as a guide in keeping the drill straight up and down. To build this guide, find a small flat piece of aluminum or steel about 1/2" thick and drill a perpendicular hole using the methods we just talked about. Now you can slide the guide over the drill bit so that it spins freely and doesn't bind. Next, line the drill bit up with your dimple and then slide the guide down flush with the material. Now you can drill the hole, using the guide to keep the drill straight up and down.

Drilling a hole properly involves turning the drill bit at the appropriate speed. You may have a drill press with markings that show what rpm it is spinning at for different settings. If so, then the rule of thumb is that for a 1/2" diameter bit in steel, spin at 500 revolutions per minute. For different drill sizes, simply apply the same proportion such that when the diameter in inches and the speed are multiplied together, the result is 250. For example, for a 1/4" diameter bit I would spin at 1,000 rpm or for a 1" diameter bit, I would use 250 rpm. When working with aluminum, simply double the speed that you would use with steel.

This is only a rule of thumb, so if a speed isn't working, play around and see if something else works better for your situation. The softer the material you are drilling, the faster you can run the drill bit. If you can't tell what rpm your drill is spinning at, then just remember that the smaller the drill bit, the faster you want to go, and vice versa. (Again, remember to always where safety glasses!)

Tapping

Tapping is the process of forming or cutting threads using a tap in a predrilled hole (of the appropriate size) so that you can screw into it using common machine screws or bolts. Usually, this is done in metals and plastics. Tapping is very useful when you want to connect two things so that they can be assembled and disassembled repeatedly. It is the equivalent of drilling through two pieces of material and using a bolt with a nut, but easier because you don't have to have access to the backside to hold the nut in place. You can also use this when a bolt and nut aren't possible, such as connecting two frame members at a right angle as shown in Figure 3.

For every size screw, there is a matching tap and drill bit. Screw sizes are designated using two numbers: the diameter and the number of threads per inch. For example, a 1/4-20 screw has a 1/4 inch major diameter (the diameter of the outermost part of the threads) and 20 threads per inch along the length of the screw. Also, for each major diameter there is typically a fine and coarse thread size, for example, 1/4-20 and 1/4-28. I recommend coarse threads for most applications because it is less likely the screw will pull out from excessive force. There is also a metric screw system, but it doesn't work with any other screw types.

A tap looks like the combination of a drill and a screw. However, do not attempt to drill a hole with a tap — it will break! There are two basic types of taps: spiral point and bottoming. The spiral point tap is used for through-holes, or holes that go all the way through the material. They are stronger and easier to use, but they push away all of the chips that they cut down into the hole. So, if you try to use one for a blind hole (a hole with a bottom), it will fill the bottom with chips and you will not be able to tap the full depth of the hole. Don't try to force the tap to push the collected chips down because you will risk

FIGURE 4. A t-handle and 1/4-20 spiral flute bottoming tap is used to create a blind hole in aluminum.

breaking the tap.

A bottoming tap is used for — you guessed it — getting to the bottom of blind holes. This type requires more care when using than a spiral point tap because it draws all the chips up out of the hole. So, you need to worry about chip build-up which can cause binding and tap breakage. I recommend spiral flute bottoming taps.

You will also need a tap handle, which come in two styles. The t-handle looks like a "T" as its name suggests, and is good for smaller tap sizes such as a 1/4" and below. Figure 4 shows a t-handle tap being used to bottom tap aluminum. There is also the straight handle tap wrench, which is often larger and consists of a single straight handle. This is good for larger taps where more leverage is required. The third option for tapping is using a hand tapper as shown in Figure 5. This allows you to easily keep the tap straight when tapping. Now that we know the tools necessary to tap, we can go over the actual tapping procedure!

1) Obtain the correct size and type drill, tap, and tap handle. Drill the hole to the appropriate depth, ensuring that it stays perpendicular to the material. Next, add some oil to the tap and hole; WD-40 works well but there are chemicals specifically meant for tapping (such as TapMagic mentioned previously). If you are tapping plastic, water will work fine.

2) Start tapping the hole by setting the tap into the hole and slowly turning the tap handle. Be careful to keep the tap perpendicular to the material and in line with the hole. This is critical because if this is not straight, it will cause your tap to break as you get deeper into the hole.

3) As you begin to cut threads, you will feel a little resistance. The amount of resistance varies for every tap size and material type, so judging how much is too much comes with experience. However, if you feel like it is surprisingly difficult to turn, stop immediately and back the tap out of the hole to check things out. You may not be in straight, the hole may be too small, or you may not have enough cutting oil.

4) Assuming that everything is going smoothly, you now want to turn the tap to start cutting threads in increments. For every two turns, you must go backwards at least one quarter turn. This will break up the chips and prevent binding.

5) If you are tapping a through-hole, then make sure to tap past the edge of the hole, because the end of the tap doesn't cut a full size thread. If you are tapping a blind hole, stop when you feel the tap hit the bottom of the hole; you can tell by a dramatic increase in resistance.

6) When you are sure your threads are fully formed, unscrew the tap from the hole. Be careful because this is the stage when tap breaks are easy. The tap is weakest going backwards and if a few chips get in the way, it will snap the tap off in the hole. If you encounter any resistance when backing out, simply go back down a turn or so and try to come back up.

7) Now you can clean the hole out by blowing into it with compressed air; just be sure to wear safety glasses.

As a beginner, you should only be tapping by hand. Power tapping can be done using a reversible cordless drill but unless you want to risk breaking taps off in the material, stick to hand tapping.

They say there are two kinds of people in this world: those that have broken a tap and those that will. This is because even with the proper techniques, right tools, and extensive experience, it is very easy to break a tap! So, what do you do when you break your first tap? One option is to use a dremel tool and small rotary cut-off disk to grind a slot into the broken tap. Then use a flat head screwdriver to try and unscrew it. You can also use a center punch and hammer to get the tap to rotate out, however, this is easier said than done. Tap removers do exist but only work well under perfect conditions and tend to be expensive. I bought one a few years ago and have yet to use it successfully. There are also chemical dissolvers you can buy that will, over days, etch away enough of the steel tap body allowing you can crack it out. (Obviously they only work on aluminum, plastic and other non-ferrous metals.)

The final and most effective solution is to simply move on and drill a hole next to it to try again. Just remember to grind down any sharp edges from the broken tap!

Welding

In smaller robots, welding usually does not have much

FIGURE 5. A hand tapper is used so that the tap always remains perfectly upright and in line with the hole.

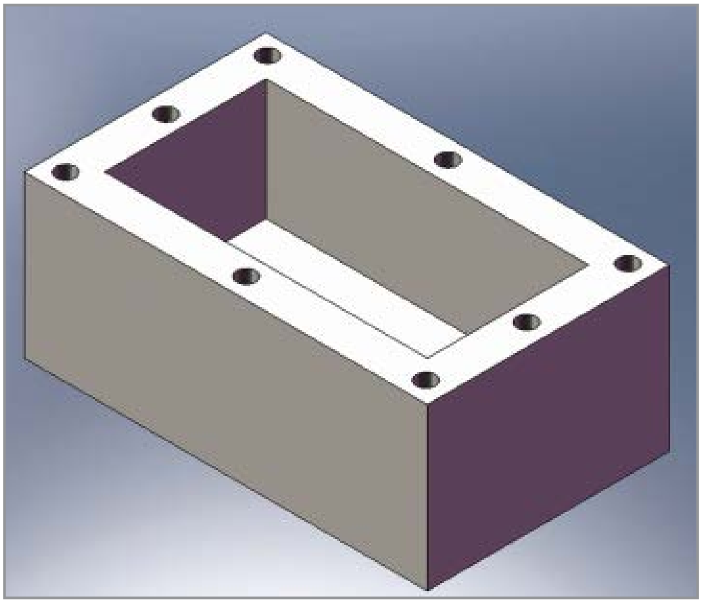


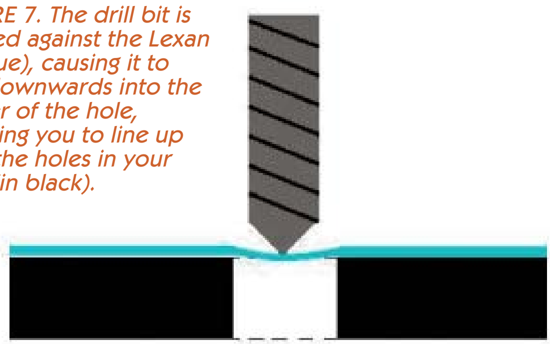
FIGURE 6. This type of part is a perfect example of when to make a hole template using a thin piece of Lexan.

of a role, but for larger bots it can be extremely useful. There are many types of welding but generally, there are three things that are required: an electrode which causes an arc which heats up the metal you are welding to a molten state; a shielding gas which protects the molten metal from contamination; and filler material that you add to the molten puddle to form the weld. Three specific types of welding that you will find useful are: stick welding, MIG welding, and TIG welding.

Stick welding involves using a consumable welding rod or electrode with a flux coating. The flux coating creates a shield of inert gas to protect the weld from contamination. A high electric current is passed through the electrode which causes an arc, thereby heating up the metal. The electrode also acts as the filler material as it melts from the arc. This method is one of the most simple and common welding methods. It is typically used to weld steel but also can be used to weld raw iron, aluminum, nickel, and copper alloys.

MIG (metal inert gas) welding is perhaps the easiest type of welding. In this method, a welding "gun" is used, through which a wire electrode is fed through the gun along with an inert gas. This type of welding is usually compared to using a hot glue gun. This method can be used for welding steel, aluminum, and other metals.

FIGURE 7. The drill bit is pushed against the Lexan (in blue), causing it to flex downwards into the center of the hole, allowing you to line up with the holes in your part (in black).



TIG (tungsten inert gas) welding is a more versatile type of welding but is also more difficult. It uses a non-consumable tungsten electrode to produce the arc. The welder controls the electrode with one hand, while they add filler metal from a welding rod with the other. There is also a foot pedal that controls the power going through the electrode. This type of welding requires the most experience, but it allows you to make much cleaner and controlled welds. Materials that can be welded using this method include steel, titanium, aluminum, and more.

Tricks of the Trade

After almost six years of building combat robots, there are a number of tricks that I have learned that help in the robot building process. These allow me to work more quickly with less precision, yet still obtain the same results. There are better ways that involve a lot more time and effort to set up, but I've found these tricks will get you about the same results.

Hole Templates

There are many times when you need to drill a pattern of holes into a sheet of metal or plastic (such as a base plate) so that they line up with holes in a frame or other part such as shown in Figure 6. If the sheet isn't clear — and often times it isn't — this can lead to a very long and arduous process of measuring out the placement of each hole. This can take hours to do properly. There is a trick, however, that I have used many times that — if done carefully — will result in all of your holes lining up perfectly. It requires a piece of sacrificial Lexan, usually about .063" thick, to be used as a template.

1) First, cut out the Lexan so that it will cover all of the holes in the part. Next, clamp it to the part that the holes need to match up to. Mark where the part is in relation to your Lexan; I usually do this by making a pen mark around the perimeter of the part on the Lexan.

2) Use a hand drill with a drill bit slightly smaller than the holes in the part. Place the drill bit over the center of the hole, so that the bit is on top of the Lexan and the hole is under the Lexan. Slowly push down and start the drill so that the Lexan deflects into the hole and the bit auto-centers. See Figure 7 for an illustrated example.

3) Drill through the Lexan being careful not to hit any threads in the part underneath. If you are mounting to something you made, then just don't tap it before this step to avoid messing up the threads, then you can use the same drill size. Repeat this procedure for all of the holes; you should finish with something that looks like Figure 8. With all of the holes drilled, you can unclamp your completed template.

Now that you have your template, you can clamp it onto your sheet material and use the template as a guide to drill all of your holes. This method can be used for a variety of situations. I have used it for mounting gearboxes to a base plate, along with mounting base plates to frames.

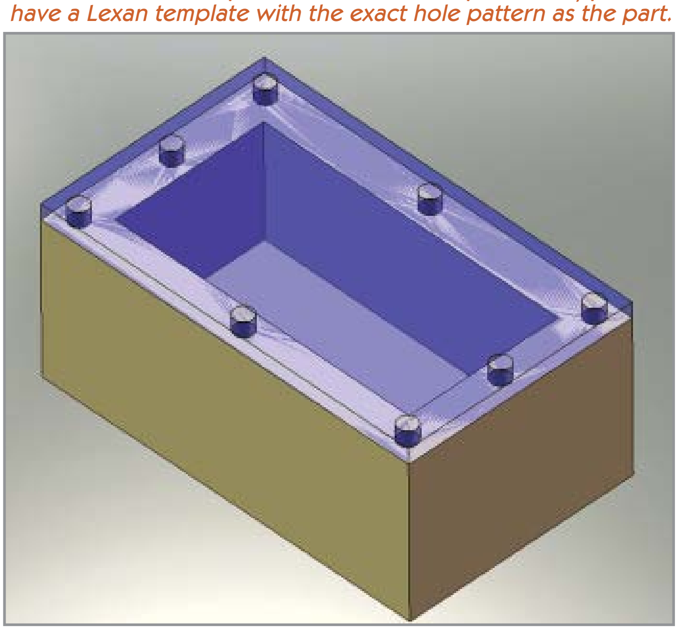
Printed Templates

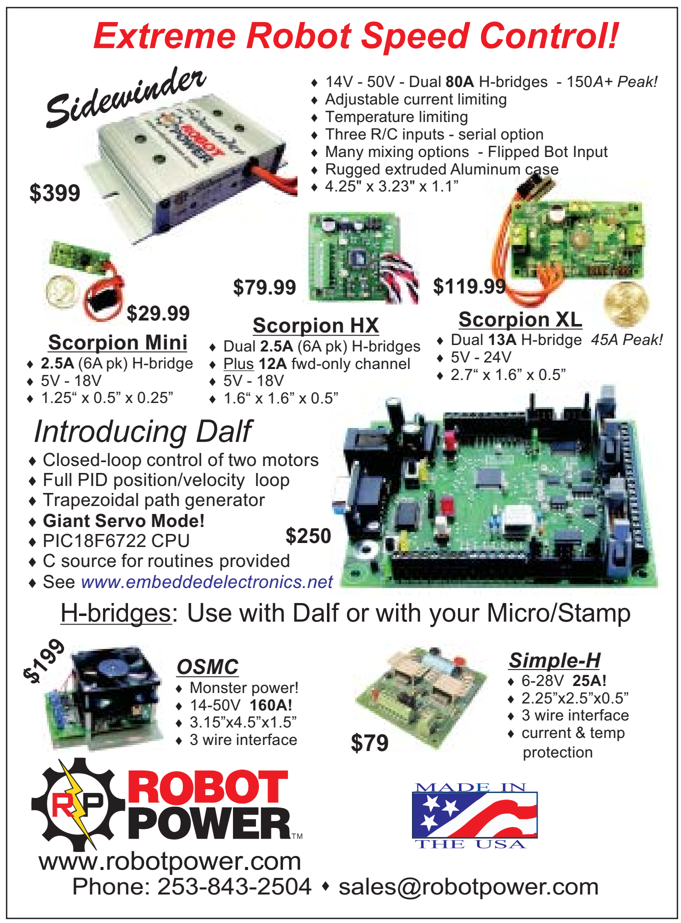
Oftentimes, you need to cut out a complicated part or drill a complicated hole pattern in situations where the previous trick wouldn't work. For situations like this, office supply stores carry a very technologically advanced and useful product: sticker paper. (Okay, so it's not that advanced, but it is very useful!) Simply buy a pack of the full sheet sticker paper, print out your template on it, and stick it to your material. When you print out your template, be sure it is scaled as 1:1. You can check this by measuring some known dimensions on it after printing it out. Also, mark all hole locations with a set of cross hairs or an X. You can then use a center punch to accurately dimple where each hole needs to go.

In Review

With the final design completed, we've been able to take a look at some of the tools and techniques that will be useful in building a robot. We explored cutting, drilling, tapping, and welding. You should now have enough knowledge to begin drilling and tapping. You also know a little bit more about cutting and welding so that you can choose a method and learn even more! We discussed a few of my favorite tricks of the trade, which should help to make your future builds easier. Next month — in the final segment of this series — we will be going through the entire build process of our designed robot using our newly acquired knowledge! For more information on myself and my robots, visit **www.robotic-hobbies.com**. **SV**

FIGURE 8. After you have drilled all of your holes, you will have a Lexan template with the exact hole pattern as the part.





The R2 Builders Club

From left to right: Disneyworld's R2-D2, R2-KT, and ILM's R2-D2.

by K. Stiles Howard

and the Jedi Code

"Adherents to the Jedi way closely follow an ancient code that guides their actions in the service of the Republic. It reads, in part:

There is no emotion; there is peace.

There is no ignorance; there is knowledge.

There is no passion; there is serenity.

There is no death; there is the Force."[1]

About nine years ago, in this galaxy, in a continent far, far away from North America, Australian Dave Everett began an Internet-based *Star Wars* fan club, which he named the R2 Builders Club (R2BC). Established for people interested in building 1:1-scale R2-D2s, the R2BC has evolved into an online community of more than 6,600 members worldwide (**http://movies.groups.yahoo.com/group/r2builders/**). Once created, these R-series astromech droids do not merely sit around gathering dust. Quite the contrary, they live very active lives, joining their humans in performing service projects that raise money for charitable organizations and educate children about science, computer technology, engineering, and robotics.

"It is from the ranks of the Jedi Masters that the High Council is chosen ..."[1]

Committed to creating a community of R2-D2 Builders, as well as

NOTE

Star Wars, R2-D2, R2-R9, the Jedi Order, and the Jedi Code are registered trademarks of Lucasfilm Ltd. R2-KT is a fan-built project and not authorized by Lucasfilm as a recognized character. Star Wars, its characters, creations, and all associated items are the intellectual property of Lucasfilm. (c)2006 Lucasfilm Ltd. All rights reserved.

fostering the construction of high-quality replica robots, founder Everett was determined that the club put measures in place to ensure parts compatibility and also safeguard against profiteering. This prompted a brilliant and inspired move: the assemblage of a Builders Council. This five-person council approves each and every one of the club's parts suppliers. These suppliers — who are also members of the group — work from R2BC's "official blueprints" and offer all their robot parts at cost.

"It is not a venture to be undertaken lightly. As such, Jedi instruction is rigidly structured and codified to enforce discipline and hinder transgression."[1]

The official blueprints for building R-series astromech droid replicas were developed by early members of the R2BC. This research and design phase lasted a couple of years. Everett recruited a small group of Builders to assist him in this endeavor. They collected measurements from Star Wars exhibits and dimensions from actual droids. These Builders then used this information to draft blueprints and perform test builds. They created their initial R2-D2 replicas by hand and adjusted the blueprints as needed. This process enabled them to make sure all the pieces fit together. The accuracy of the resulting blueprints provided the group with standardized plans, which ensures the uniformity of the robot parts made by Builders, as well as parts suppliers. So, if an R2 Builder in Paris, TX bought a pair of aluminum outer ankle brackets from an approved supplier in Paris, France, those French brackets would be compatible with that Texan's other aluminum robot parts. (To read more about how the R2BC official blueprints benefited a German fellow who was creating an R2 on his own, visit **www.r2-d2.de/index5.htm/**.)

When enough R2BC members post to the group expressing interest in a particular robot part, an approved supplier will offer a "parts run." First, the supplier collects payment from all interested parties. Then, the supplier pays a shop (for example, a machine shop) to make the part that has been requested. The supplier also submits contact information for the chosen shop and that run's buyers to the Builders Council. Longtime R2BC member Craig Smith explains: *"The R2BC does not offer complete R2-D2s for sale. Group parts that are available from time to time are made by Builders for Builders as a hobby. The parts are not mass produced for profit, and no kits are available."*

Once a particular batch of parts is created, the supplier ships those parts to the Builders who ordered them. The duration of a parts run can vary from weeks to months — even years, in some cases. Soon after the shipments are made, cheers of glee are emitted around the globe as Builders collect their newly crafted parts from their porches. At approximately the same time (particularly if the shipped parts are aluminum), UPS and Fed-Ex carriers the world over


A father and daughter meet R2-KT.

R2-KT in all her glory.


can be heard grumbling about lower back pain.

"As the Jedi mature, the apprentice is paired with a master to continue the next phase of the training."

So, how does the R2 Builders Club online community work? When Builders have difficulties — say, locating a part, figuring out the blueprints, deciding which foot motor to use, or simply wondering where to begin — they post to the group, and they get answers. Of the group's 6,600+ members, only 100 or so actively post to the group.

Longtime members delight in coaching new members on the process of building these replica robots. Master astromech Builder Jerry "JAG" Greene enjoys helping out fellow club members. One of the first dozen members of the group, founder Dave Everett recruited JAG to assist him in creating the official R2BC blueprints. Needless to say, Jerry has a wealth of knowledge and experience in all R2 related topics. JAG's advice to new R2 Builders is *"Have fun. Don't stress out about it. The R2BC includes design engineers, helicopter pilots, and machinists. If you don't know how to do something, someone in the group will."*

"To become a Jedi requires the deepest commitment and most serious mind."

R2-D2 may be diminutive in size (with a height of 0.96 meters or 3'2"), but building a 1:1 scale replica of its likeness is no small feat. It requires an enormous time commitment, some disposable income, meticulous research, and tireless dedication. It also helps to have supportive family and friends. Another necessary component is the realization that this enterprise cannot be done alone. Each member of the R2BC needs this group — its standardized blueprints, its expertise, its parts runs, its camaraderie — to see this task to completion.

Craig Smith advises prospective members about the amount of time it takes to create a droid: "R2 built from scratch or built from club parts is not a quick project. It will take months to complete a static, non-moving display — even years for a motorized version. When I look at the parts on my droid, I recall the week it took to make the shoulders, the days it took to make the ankle covers, the days it took to re-make the ankles themselves because I did not offset them correctly. And there are dozens more parts on the droid with similar memories. A project such as this is a huge commitment!"

> R2 built from scratch or built from club parts is not a quick project. It will take months to complete a static, non-moving display — even years for a motorized version ..."

I spoke with Jerry Greene by phone one recent Saturday afternoon. The first droid he built was a replica of R2-R9, which is red and silver. According to JAG's website (**www.r2-r9.com/**), R2-R9 made a brief appearance in *Star Wars Episode I: The Phantom Menace*. While "serving aboard Queen Amidala's Royal Starship as a repair droid, he was shot off the hull during a repair." It took Jerry three years to build his first droid. I asked him about the average time and financial investment involved in building an all-aluminum R2-D2. He responded, "About two years and around $9,000 — about the cost of a nice, mid-sized car." Needless to say, if you're going to spend that much time and that much money building a droid, it doesn't hurt to have a healthy sense of humor.

However, it doesn't have to cost an arm and a leg to build — well, an arm and a leg. Astromech droids can be built using any sort of material imaginable. Whatever medium one can afford, has available, and feels comfortable using can be fashioned into a lovable R2-D2. Many R2 Builders pride themselves in building on the cheap. The club has even offered workshops on "Droid Building on a Budget." Here's what longtime R2 Builders Club member Craig Smith has to say on the range of astromech-building media that he's seen used: *"The most impressive droids I've seen are almost 100% home built. For the body, I've seen skins on frames in both aluminum and plastic — even wood. Rolled aluminum cylinder, fiberglass and, of course, 18" PVC pipe have been used. The method of building a droid directly reflects what materials a Builder is comfortable working with."*

As for what an astromech droid can do, that is a matter only its Builder can decide. According to the R2 Builders Club brochure, some people start out by making a static droid to display at home or at work. After building one droid and gaining some confidence, a Builder might get more adventurous. Some club members create remote control droids that emit sounds and have blinking lights. An experienced R2 Builder like Smith can create a droid that can do very impressive tricks indeed. *"I have three astromechs that are R/C and do the show-stopping 2-3-2 leg transformation. Lucasfilm had one filming unit that could go into the three-leg mode via an air piston with locks at the end of the function, but it could not go back into the two-leg position. My systems use electric motors that are geared down to provide much-needed torque to control the transitions. Limit switches at the ends of the functions stop the transitions where they need to. My design has improved from one droid to the next. So, naturally, I left my latest droid without coverings so people* *can see how the final design works."*

As for how to power an R/C astromech, apparently, there is some very fancy footwork involved. Smith offers the following recommendations: *"I cannot say enough good things about the Robot Power Scorpion XL dual-speed controller for the foot drives. This little $120 unit learns your radio, shuts down in signal loss, has dual or single-stick control, direct output throttle, or sweep exponential control (and has a) battery eliminator that powers the receiver. I have not heard of one fried controller in the field as of yet."*

"For foot motors, many (R2 Builders) were using surplus motors and finding ways to adapt them to a good gear ratio wheeled drive. But since the electric scooters have become popular, one can afford to buy a pair of scooters and hack/modify the frame to fit in the foot. Scooter parts are also now hitting the surplus and aftermarket parts suppliers. One can get a pre-fab drive system that goes much faster for way cheaper today than the options we had just a few years ago."

Members of the R2 Builders Club enjoy meeting other Builders in person, and there are many opportunities for them to do so. There are regional groups, such as the New England (NE) Builders (**www.r2-r9.com/Gallery_R2-MA.html**) and the Midwest Builders (**http://stevesr2.blogspot.com/2007/08/so-this-saturday-was-annual-mid-west-r2.html**), who gather from time to time. Regional R2 Builders groups may meet up at one area Builder's shop to make parts or just spend time together. The NE Builders get together every six months or so. Someone will have a barbecue and the Builders will swap stories and catch up with each other's news. For larger gatherings of Builders, there are always comic book conventions and fan-based conferences (for example, *Star Wars* Celebrations (**www.starwars.com/celebration/**), San Diego Comic Con (**www.comic-con.org/cci/**), and Dragon Con (**www.dragoncon.org/**)). At such events, R2BC members exhibit their droids, lead panel discussions, and hold workshops. Celebration III featured 50 fan-built droids on display. George Lucas even viewed the R2 replicas at that event, much to the delight of the Builders. Over the years, the R2 Builders Club has also developed a unique relationship with Lucasfilm Limited — George Lucas' production company — and the creative force behind the *Star Wars* movies. When I asked Jerry Greene to characterize Lucasfilm's relationship with R2BC, he said, "As long as we play nice, they leave us alone." Well, apparently they do play nice, because Lucasfilm has called on an R2 Builder or two from time to time to ask a favor — and, on at least one occasion, to grant a favor.

According to Wookiepedia, the *Star Wars* wiki (**http://starwars.wikia.com/wiki/Main_Page**), the R2BC's "official club logo was adopted by the Lucasfilm R2-D2 Unit for their crew gear during the filming of *Star Wars Episode II: Attack of the Clones* in Australia." Needless to say, the club's members were thrilled at this nod to their work. Greene also informed me that Lucasfilm has contacted R2 Builders in the New York City and Los Angeles areas to request that their astromech droids make appearances at movie premieres. For a *Star Wars* fan and R2 replica builder, attending a premiere at the behest of Lucasfilm must be a dream come true. A red carpet event must prove a most exciting venue in which to show off an astromech droid that was years in the making.

Greene also commented that Don Bies, a Modelmaker and R2-D2 Operator for Industrial Light and Magic (ILM) who has worked on several *Star Wars* movies, is a member of the R2 Builders Club (see **www.donbies.com/bio.htm**). When Bies needed a spun aluminum, laser-cut, R2 dome to use in *Star Wars Episode III: The Revenge of the Sith*, he knew who to call: Master R2-Dome Creator and fellow R2BC member Ron Barkley. Ron's dome appears in a scene in which an R2 unit's dome is ripped off of a ship. No matter how brief the scene, I have little doubt that R2 Builders the world over are immensely proud of it.

The Jedi are "a noble order of protectors unified by their belief and observance of the Force."[2]

To say that Albin Johnson is an active member of the *Star Wars* community would be a vast understatement. Albin is the founder of Vader's Fist: 501st Legion (The World's Definitive Imperial Costuming Organization). His 501st Legion not only entertains *Star Wars* fans during their appearances, this group works year-round to raise money for charitable organizations worldwide. Albin is greatly admired and respected for his devotion to his family, as well as his

R2BC droids on display at Celebration IV.

desire to make a positive impact in the world. And so it was devastating to all who know Alvin and his family when they learned that his six-year-old daughter Katie had a brain tumor. A bright and shining star to all who knew her, Katie's friends and family were deeply saddened by this news.

While attending a church service with his family shortly after his daughter's diagnosis, Albin had a lightbulb moment. According to the R2-KT website (**www.r2kt.com/**): *"Albin noticed something funny about the sanctuary's windows. Call it a sign, call it Al not paying attention in church, but the window looked eerily like an R2 unit and it gave him an idea: Why not build an R2 to watch over Katie as she slept (just like R2-D2 watched over Padme in Episode II)? Katie's older sister Allie went one step further: Why not paint it pink and name it after Katie: R2-KT? An idea was born ..."*

In April of 2005, Albin discovered the R2 Builders Club. Upon hearing of Albin's intentions to build his courageous and wonderful daughter a pink R2-D2, the group was eager to help him realize this goal. Given that an R2 Builder's first droid can take years to complete, it was clear to the group that something unprecedented would need to occur in order for R2-KT to take shape.

Greene suggested that the R2 Builders Club build an all-aluminum R2-KT and donate it to the Johnson family. Greene offered to orchestrate this group build and assemble the droid himself. Albin graciously and humbly gave the club permission to proceed. Shortly thereafter, Jerry began creating sketches of R2-KT and posting requests to the group to donate the parts he would need to build this one-of-a-kind, pink astromech droid. The response was overwhelming. While the group set about building an all-aluminum R2-KT, R2BC member Andy Schwartz did something quite extraordinary, as well. He disassembled his own R2-D2, painted all of the blue sections pink, put it back together again, and arranged for this original R2-KT to be delivered to the Johnsons' home. This act of generosity allowed Katie to have her very own pink astromech droid to keep her company and lift her spirits.

After a brave fight, Katie Johnson passed away on August 9, 2005. Her family, her friends, and the *Star Wars* community mourned her loss.

The R2 Builders Club decided to continue building R2-KT as a memorial to this wonderful girl. It was an emotional process for everyone involved. Eager to express their compassion for this family, dozens of R2 Builders from around the globe donated astromech parts to the R2-KT project. Over the next year, box after box of donated parts arrived at Greene's Rhode Island home, where he worked tirelessly in his basement studio to assemble this unique droid. (Visit JAG's "Project R2-KT" website at **www.r2-r9.com/Project%20R2-KT.html** to read a chronology of the build and see images of all this robot's parts.)

By July 2006, R2-KT was complete. Greene and the R2 Builders Club constructed this adorable, circus pink and white astromech droid in record time, taking little more than one year from start to finish. Once the robot was complete, Greene and his girlfriend, Lisa, loaded R2-KT in their minivan and took it on its first adventure. Jerry and Lisa drove


More R2BC droids on display at Celebration IV.

"The R2 Builders Club decided to continue building R2-KT as a memorial to this wonderful girl."

seven hours from Rhode Island to the annual Shoreleave *Star Trek* convention in Hunt Valley, MD. The reason for this journey: to deliver R2-KT to the Johnson family.

Schwartz and other members of the R2 Builders Club joined Jerry and Lisa at the convention, where they met Albin, his wife Kathy, and their daughters Allie and Emily. On behalf of the R2 Builders Club, R2-KT was presented to the Johnson family as a gift. Albeit an emotional meeting, it was also a beautiful, loving celebration of young Katie's life.

And that was only the beginning of R2-KT's adventures. Like her fellow R-series replica robots and the members of the 501st Legion, this little droid has a mission. Not only that — she has a mission statement. According to the R2-KT website (**www.r2kt.com/**): *"R2-KT's mission is to entertain children, raise awareness of pediatric cancer, and raise money for such charities as Make-A-Wish and Children's Cancer Fund."*

In November 2006, R2-KT participated in her first toy drive. Appearing alongside the 501st Legion and the United States Marine Corps at a Toys-for-Tots drive at a Toys R Us store in Columbia, SC, R2-KT was a huge hit. In February 2007, R2-KT brightened the day for patients and staff at Palmetto Richland Children's Hospital, where Katie Johnson received treatment. And it seems that this lovable droid is not only cute, she's smart, too. In March of that year, R2-KT went to college. She accompanied the Johnson clan at Albin's alma mater for the University of South Carolina's FIRST Robotics Competition. She proved vastly entertaining to the children in attendance, and she also (with Albin's help, no doubt) offered instruction on robotics. *"For his courage, Artoo was personally thanked and recognized by Queen Amidala."*[3]

In early 2007, the Hasbro toy company contacted Albin. Hasbro and Lucasfilm had heard of R2-KT, and they were partnering to create a limited edition action figure in her likeness. Lucasfilm invited the Johnson family and R2-KT to attend the Celebration IV *Star Wars* convention in Los Angeles, CA. Hasbro and Lucasfilm's official announcement of the R2-KT action figure occurred on May 25, 2007, the 30th anniversary of the theatrical release of *Star Wars*. Available exclusively at the 2007 San Diego Comic Con, **hasbrotoyshop.com**, and **starwarsonline**, all proceeds from R2-KT sales were donated to the Make-A-Wish Foundations of San Diego and South Carolina. The projected benefits exceeded $100,000.

Since the R2-KT project, the R2 Builders Club continues building accurate replicas of R-series robots. And even though engineering and computer technology are major components of these models, these creations are also gorgeous works of art. Although, technically, they are reproductions of robot models made for a series of blockbuster movies, they represent much more than that. These R2s represent the joys and complexities of childhood. They celebrate the seemingly limitless expanse of the

R2BC droids line up to be admired at Celebration IV.

Resources

http://movies.groups.yahoo.com/group/r2builders/

http://starwars.wikia.com/wiki/R2_Builders_Club

www.r2-r9.com

www.r2kt.com

www.carolina501st.com/content

R2BCBrochure.pdf

www.astromech.net

www.starwars.com/databank/updates/news20010613.html

www.starwars.com/databank/organization/thejediorder

www.starwars.com/databank/droid/r2d2

http://starwarsmovies.net/star-wars-news/the-jedi-code

http://starwars.wikia.com/wiki/R2-KT

http://starwars.wikia.com/wiki/501st_Legion_%28 organization%29

R-Series, Issues 1, 2, and 3 — edited by Cory Pacione and Dan Baker

human imagination. And these adorable little astromech droids offer us a glimpse into what is possible when human beings work diligently, share knowledge, and strive for excellence. The excellent craftsmanship of the R2 Builders Club's robot replicas, the group's success in thwarting profiteering, as well as its contributions to educational outreach programs and charitable organizations may have played a role in opening the door to a mutually respectful relationship with Lucasfilm. We'll never know for certain. When founder Everett established the Builders Council, he had no way of knowing that he was setting the stage for a project in which dozens of builders from around the planet would donate robot parts to create a unique astromech droid to honor one very special girl.

"During the restructuring of the Jedi Order by the now Jedi Grand Master Luke Skywalker, a new code was established for easier interpretation for the newer generation of Jedi. The code retained the same core beliefs as the millennia old code, rewritten for better understanding.

Jedi are the guardians of peace in the galaxy.

Jedi use their powers to defend and protect, never to attack others.

Jedi respect all life, in any form.

Jedi serve others rather than ruling over them, for the good of the galaxy.

Jedi seek to improve themselves through knowledge and training."[4]

At a recent educational outreach event, Jerry Greene had the honor of introducing his R2-R9 to a blind *Star Wars* fan. Having often wondered what an R2 unit looked like, this 12-year-old boy was overjoyed to be in the presence of an astromech droid, be allowed to explore its entire surface, and be able to experience all of its tactile qualities. For Greene, having an opportunity such as this — to quietly observe as a fellow human being embraces the opportunity to experience his world in an entirely new way — all the time, research, money, and effort invested in building an R2 unit just melts away into space.

To read a profile of an R2BC member who is crafting an all-aluminum astromech droid and see images of hand-milled robot parts this Builder has created, check out Vern Graner's "Personal Robotics" column in the May issue of *Nuts & Volts Magazine*. SV

REFERENCES
[1] **www.starwars.com/databank/updates/news20010613.html**
[2] **www.starwars.com/databank/organization/thejediorder/**
[3] **www.starwars.com/databank/droid/r2d2/**
[4] **http://starwarsmovies.net/star-wars-news/the-jedi-code/**

Acknowledgements

I would like to thank Vern Graner, Jeff Green, Craig Smith, and Chris Simonds for their inspiration and guidance; Guy Vardaman, Bob, and all the R2 Builders who supplied me with images; James Delaney and Jennifer for help with graphics; Anna and Lici for their encouragement.

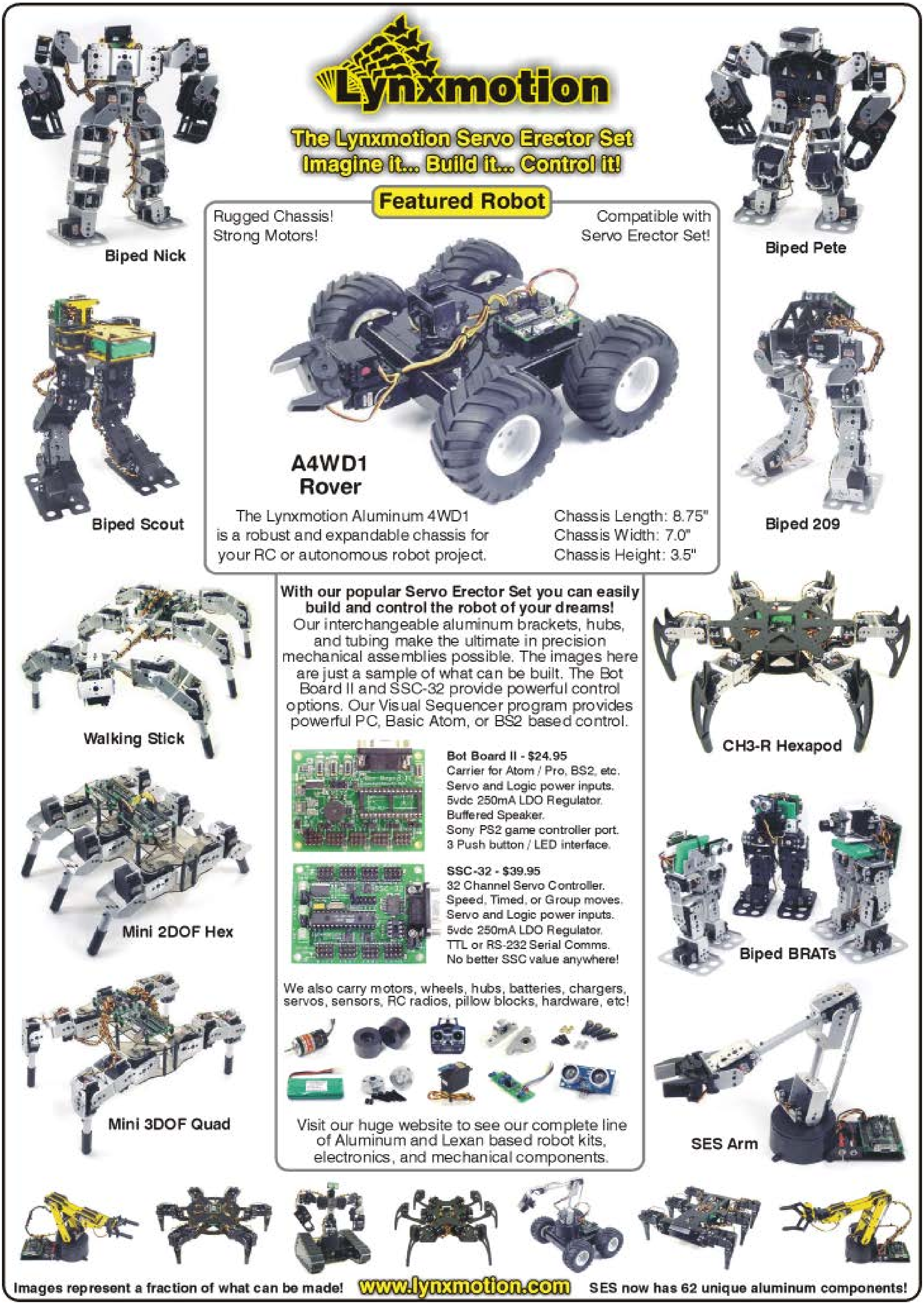
YOU CAN BE COOL, TOO!

Get your very own *SERVO* t-shirt and hat!

Hats $12.59

Shirts $9.95

Call:
800-783-4624
or order online at:
www.servomagazine.com


Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
Featured Robot
Rugged Chassis!
Strong Motors!
Compatible with
Servo Erector Set!
A4WD1
Rover
The Lynxmotion Aluminum 4WD1
is a robust and expandable chassis for
your RC or autonomous robot project.
Chassis Length: 8.75"
Chassis Width: 7.0"
Chassis Height: 3.5"
Biped Nick
Biped Pete
Biped Scout
Biped 209
With our popular Servo Erector Set you can easily build and control the robot of your dreams!
Our interchangeable aluminum brackets, hubs, and tubing make the ultimate in precision mechanical assemblies possible. The images here are just a sample of what can be built. The Bot Board II and SSC-32 provide powerful control options. Our Visual Sequencer program provides powerful PC, Basic Atom, or BS2 based control.
Walking Stick
CH3-R Hexapod
Bot Board II - $24.95
Carrier for Atom / Pro, BS2, etc.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
Buffered Speaker.
Sony PS2 game controller port.
3 Push button / LED interface.
SSC-32 - $39.95
32 Channel Servo Controller.
Speed, Timed, or Group moves.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
TTL or RS-232 Serial Comms.
No better SSC value anywhere!
Mini 2DOF Hex
Biped BRATs
We also carry motors, wheels, hubs, batteries, chargers, servos, sensors, RC radios, pillow blocks, hardware, etc!
Mini 3DOF Quad
SES Arm
Visit our huge website to see our complete line of Aluminum and Lexan based robot kits, electronics, and mechanical components.
Images represent a fraction of what can be made!
www.lynxmotion.com
SES now has 62 unique aluminum components!

MANAGING YOUR MOBILE MONKEY

by Fred Eady

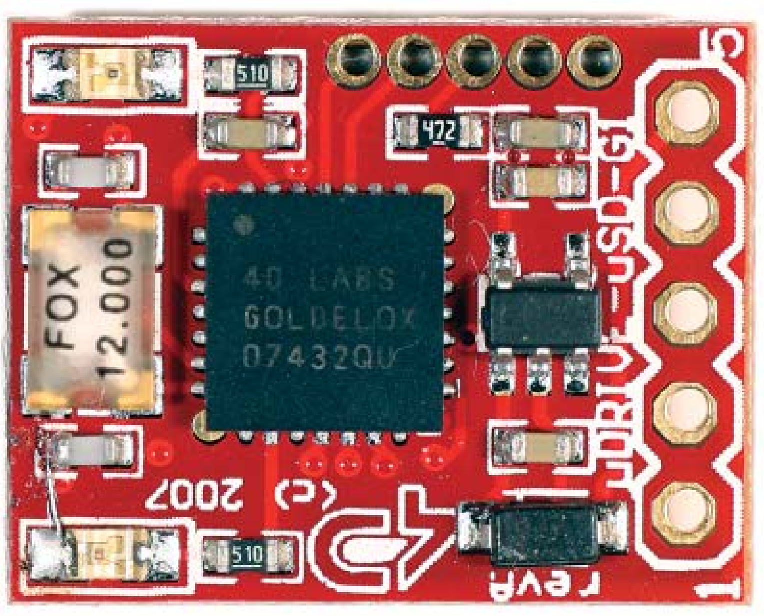


PHOTO 1. The 2 GB microSD memory card fits neatly into an automatic release microSD socket on the other side of the uDrive-uSD-G1 printed circuit board. You may also use a 64 MB or 1 GB microSD memory card if 2 GB is overkill for your application.

As a robotician, think about all of the neat things that you have direct access to by way of the pages of *SERVO*. For instance, Parallax offers a variety of sensors that include a PIR (Passive Infrared) sensor, a color sensor, and a combination temperature/humidity sensor to name just a few. Digging deeper into a *SERVO* magazine I'm reading at the moment reveals a company called Maxbotix that is offering an ultrasonic sensor they call MaxSonar. If you want to sense with IR, you have the resources of HVW Technologies at your disposal. As a reader of *SERVO*, you know that I've just scratched the surface when it comes to innovative gadgets offered by *SERVO* advertisers.

Obviously, sensing and gathering data is a robot thing. However, what do you do with all of those bits of information your little aluminum monkey finds? Sure, you may be able to process most of it immediately but there may be a time when you need to store the data bananas your mobile metallic simian has gathered for analysis or use at a later time.

Most modest robotic designs are based on small microcontrollers and as a rule, small microcontrollers don't pack a punch when it comes to available data SRAM. The average PIC microcontroller can only promise about 3 KB or less of on-chip data memory. If your application is based on the new PIC32MX, you can count on a bit less than 32 KB of data memory and that's only if you use the largest PIC32MX variant. You can also choose a PIC microcontroller that will allow you to attach a big chunk of SRAM to its I/O subsystem. If money is not an issue and your robot doesn't have a power, weight, or size limitation, you can choose to run your robotic data collection agency with a full-blown Intel-based or AMD-based embedded computer and a regulation spinning disk drive. Regardless of your robot's size or power source, would you be interested in reading the rest of this article if I could show you how to add 2 GB of direct access storage to your robot with a PIC and a micro-SD memory card?

uDrive-uSD-G1

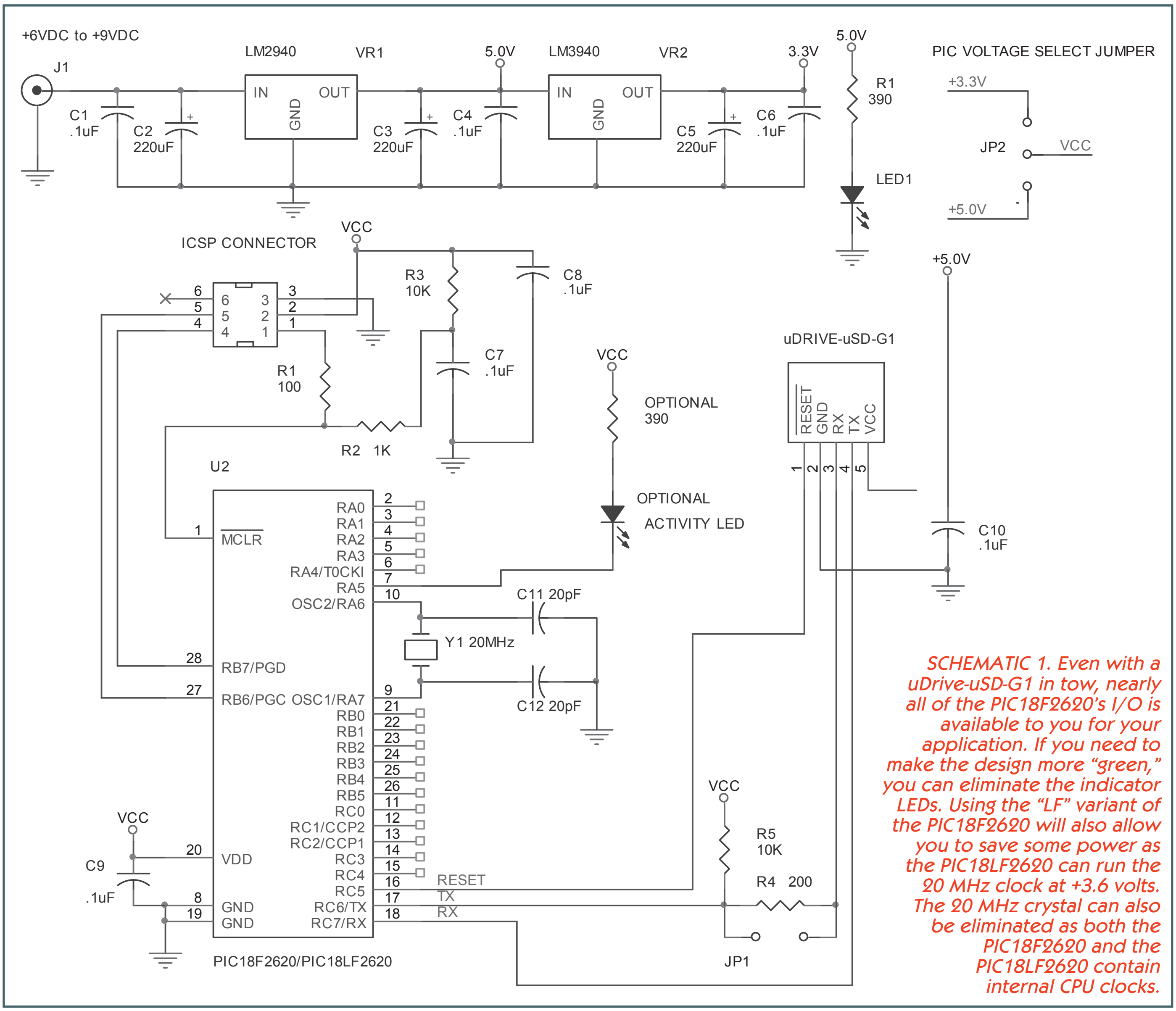
My uDrive-uSD-G1 and its storage element can be seen in Photo 1. As you can imagine from the view, the uDrive and its micro-SD memory card are very compact and don't tip the scales to any great extent. It's also apparent that the uDrive is designed to be a holding tank for a very large amount of data. The cool thing about the uDrive is that is has a humongous data capacity and a very tiny I/O interface. All you need are two I/O pins to facilitate movement of data between it and your PIC.

The uDrive requires a power source between +3.6 VDC and +6.0 VDC but only a paltry 23 mA of current. I actually used a +6.0 VDC AA battery pack to power my entire design, which includes a PIC18F2620 microcontroller.

Take another look at Photo 1. The component just above pin 1 of the five-pin header is a diode, which is in series with pin 5 — the uDrive's incoming power source portal. The diode is there to prevent damage if the power is applied with the wrong polarity. If power is attached correctly, the blocking diode feeds the input of a +3.3-volt linear regulator. How do I know this? The letters "LORA" are embossed on the regulator. A datasheet search exposed the voltage regulator as a National Semiconductor LP2985. I revealed the identity of the LP2985 voltage regulator with a clue provided by the uDrive's I/O specifications.

The uDrive communicates serially with a host processor using its TX and RX pins. The RS-232 voltage levels for the serial I/O are at 3.3 volts according to the user manual. The 3.3 volt RS-232 levels set me on the path of a +3.3 volt, five-pin SOT-packaged linear voltage regulator. I found a package description of "LORA" in the LP2985 datasheet which, according to the datasheet, can be found only on the National Semiconductor LP2985AIM5-3.3. With this information, I was able to match the uDrive's voltage regulator input and ground pins to the LP2985 datasheet connection diagram.

The LP2985 is a low-dropout linear regulator, which explains the minimum input voltage of +3.6 volts. The inclusion of the +3.3-volt linear voltage regulator in the uDrive circuitry allows it to be used with +5.0 volt microcontrollers and their +5.0 volt peripherals. The only external component that is recommended for +5.0 volt systems is to add a 100Ω to 220Ω resistor in series with the uDrive RX I/O pin. The uDrive can be part of any +3.3 volt microcontroller system, as well. If you absolutely have the need for a +3.3 volt system, you can choose to bypass the uDrive's onboard voltage regulator. I don't



SCHEMATIC 1. Even with a uDrive-uSD-G1 in tow, nearly all of the PIC18F2620's I/O is available to you for your application. If you need to make the design more "green," you can eliminate the indicator LEDs. Using the "LF" variant of the PIC18F2620 will also allow you to save some power as the PIC18LF2620 can run the 20 MHz clock at +3.6 volts. The 20 MHz crystal can also be eliminated as both the PIC18F2620 and the PIC18LF2620 contain internal CPU clocks.

recommend that as the user manual never mentions that possibility. You've already figured out that the uDrive's I/O pins include a VCC pin, a GND pin, a TX pin, and an RX pin. The fifth and remaining I/O pin is an active-low RESET pin, which is tied high internally.

The uDrive "talks" to us by way of a pair of LEDs. When power is applied, a green LED is illuminated. A red LED tells us that the memory element mounted on the uDrive is being accessed. In order for it to talk to us, we must first talk to it.

Talking to the uDrive-uSD-G1

The device uses standard start-bit-data-stop-bit RS-232 protocol to communicate with a host microcontroller. However, instead of using true mark (negative voltage = one) and space (positive voltage = zero) RS-232 voltage levels, the uDrive's TX and RX pins tie directly to the serial port of the host microcontroller it is to support. No RS-232 converter IC is included in the uDrive-uSD-G1-to-microcontroller communications link. Thus, a simple three-wire (TX,RX, and GND) TTL serial interface is all that is required for talking to the uDrive. The serial data format used is also a defacto standard: (8N1) eight data bits, no parity, and one stop bit. The uDrive's and PIC18F2620's physical communication link connections can be seen in Schematic 1. The uDrive is designed for easy implementation. A native Command Set is part of the internal programming. The Command Set consists of a General Command Set and a Disk Drive Operation Command Set.

Commands and data are transferred without any delimiting characters. In other words, there are no carriage returns, commas, or spaces in the command and data transfer sequences. The uDrive does not check for correct command syntax. Therefore, we must make sure that each command is sent correctly. For instance, a command that is made up of five bytes will not return an error if only four of the five bytes are sent. In addition, there are no command timeouts. Considering our example with four of five command bytes sent, the uDrive will wait silently — and forever — for that fifth byte.

With the exception of the version command, the uDrive will always return an ACK byte when a valid command is received. The ACK is represented by a hexadecimal 06 (0x06). Upon receipt of an unknown command, the uDrive will respond with a hexadecimal 15 (0x15). The 0x15 represents a NAK, or negative acknowledgement.

We want to always try to do things that force us to process ACK characters and eliminate the need for the uDrive to pass NAK characters back to the PIC. So, we will write our application program interface (API) to insure that complete and recognized commands are always transmitted to the uDrive by our PIC18F2620.

Building a PIC-Based uDrive-uSD-G1 System

Before we can write and test our API, we'll need to build up some supporting hardware. I've already turned your head towards Schematic 1. So, swivel your noggin to Photo 2 for now. Working our way left to right, the pair of voltage regulators you see in Photo 2 are a matched set. The LM2940 is a +5.0 volt low-dropout linear regulator. The voltage regulator sitting at the LM2940's output pin is a low-dropout +3.3 volt linear regulator. The LM3940 is designed to stand at the LM2940's output to provide +5.0 VDC to +3.3 VDC voltage conversion. The complementary pair of voltage regulators can supply up to 1A of current without the need for an external heatsink. We won't get anywhere near the 1A current consumption figure in our design and no formal heatsinks have been added to the hardware design.

As per the schematic, we will power our uDrive with +5.0 VDC. Since I have chosen the PIC18F2620 instead of the PIC18LF2620, we must jumper the +5.0V portion of the PIC voltage select jumper (JP2). The PIC18F2620 cannot reliably operate at voltages below +4.5 VDC.

At power-up or at forced reset, the uDrive-uSD-G1 will look for the host to attempt an autobaud detect operation. To assure that the uDrive does not sense a false start bit, we must pull the RX line logically high at power-up. I have done this with pullup resistor R5. Note that I pulled up the PIC18F2620's TX I/O pin on the PIC side of the 200Ω limiting resistor (R4). The placement of R5 insures that we won't destroy the integrity of the uDrive's +5.0 volt I/O tolerance that is provided by resistor R4. If this were a +3.3 volt system, we would properly jumper JP2 for +3.3 volt operation and place a shorting jumper across JP1 as the +5.0 volt I/O tolerance would not be required. Regardless of the position of JP2, the device is always powered by the

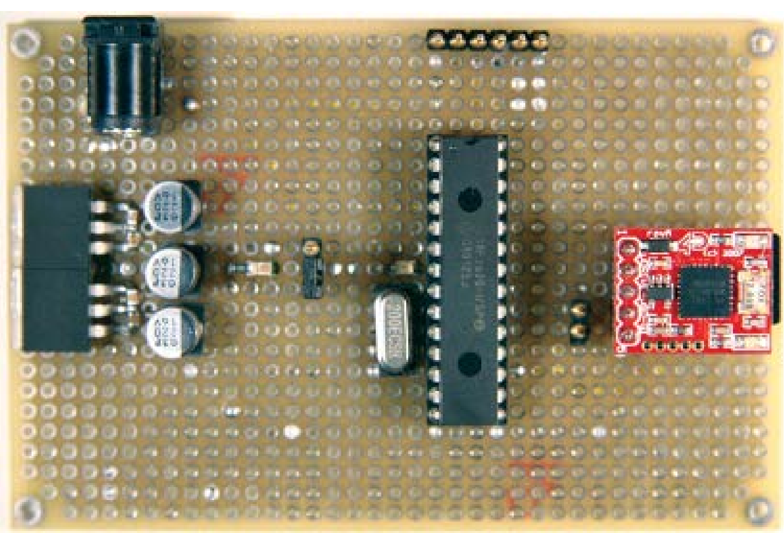


PHOTO 2. What you see here is a simple dual-voltage linear power supply feeding a minimal PIC18F2620 microcontroller configuration. The uDrive-uSD-G1 stands out at the far right of the shot. The six-pin header at the top of the shot is the PIC ICSP programming/debugging interface. A three-pin voltage selector jumper (JP2) positioned between the power supply and the PIC18F2620 is jumpered for +5.0 volt operation. The two-pin jumper (JP1) that lies between the PIC and the uDrive-uSD-G1 is open, allowing the 200Ω RX series resistor to come into play as this is a +5.0 volt system.

+5.0 volt rail as its onboard +3.3 volt regulator is always engaged. The PIC18F2620 is the major player in a minimal PIC microcontroller configuration. There are just enough supporting components to allow us to clock, program, and debug the PIC18F2620. We could make the uDrive design a bit leaner by using the PIC18F2620's on-chip internal CPU clock instead of the 20 MHz crystal (Y1) and its associated capacitors. The PIC18F2620 can run at speeds between 31 kHz and 32 MHz by utilizing its internal oscillator and companion PLL (Phase Locked Loop) clock multiplier. I incorporated the optional activity LED into the hardware you see in Photo 2. The activity LED is triggered by the Timer3 interrupt and blinks at 1 Hz when the PIC is percolating as it should. If you think the activity LED is just eye candy, the inactivity of the activity LED helped me pinpoint and repair a cold solder joint in my circuitry during the hardware build phase of this project.

The uDrive attaches directly to the PIC18F2620's EUSART (Enhanced Universal Synchronous/Asynchronous Receiver Transmitter) I/O pins. Using the uDrive's RESET pin is optional. I chose to connect the RESET pin as I like the host PIC to have full control of the peripherals within its domain.

I assembled my uDrive project on a custom plated-through breadboard, which I had manufactured by ExpressPCB. As I alluded to earlier with my reference to a cold solder joint, the hardware you see in Photo 2 was constructed using point-to-point wiring techniques. You can use your connector of choice for the Microchip ICSP programming/debugging interface. I used a six-pin configuration here as I have a RJ-12 dongle that I use as an adapter for my Microchip REAL ICE in-circuit emulator. I'll provide the ExpressPCB file for the dongle just in case you want to use it, as well. You can get the ExpressPCB dongle file from the *SERVO* website (**www.servomagazine.com**).

Coding the uDrive-uSD-G1 API

Since the uDrive communicates serially with the PIC18F2620's EUSART, we'll need to assemble a serial driver for the PIC18F2620. Our PIC18F2620 serial driver is interrupt driven and consists of a pair of 256-byte transmit and receive buffers. The buffers are holding points for incoming and outgoing data. Data transfers to and from the buffers are triggered by the EUSART interrupt mechanism. A single-statement function called *CharInQueue* is the EUSART interrupt trigger. Incoming serial characters from the uDrive are processed by the *EUSART_RxBuf* buffer and *recvchar* function while the *sendchar* function and *EUSART_TxBuf* handle outgoing serial traffic. You can see all of the buffer definitions and the actual EUSART code by downloading the API source code from the *SERVO* website.

The EUSART code is universal and you can apply it in your other PIC projects. When you get your copy of the uDrive API source code, you'll see that I defined the PIC18F2620 TX pin that is normally under total control of the PIC18F2620's EUSART. We need to control the logic level of this TX I/O pin when initializing the uDrive. The device needs some time alone before we attempt to talk to it. That "alone" time is 400 ms immediately following power-up. We need to make sure that the uDrive's RX line is pulled logically high and the PIC18F2620's RX line is inactive within 100 ms after power-up as the uDrive will be mumbling incoherently over its TX line during this time. The pullup resistor makes certain that the RX line is pulled high at power-up. We must disable the PIC18F2620's EUSART to take control of its pins. So, at uDrive power-up, the PIC18F2620 EUSART is not enabled and the PIC18F2620's TX line is commanded logically high by the API code. Pullup resistor R5 is logical high insurance as it maintains a logically high level on the uDrive RX pin during the entire power-up sequence.

The PIC18F2620 timing is also interrupt-driven. Timer1 and Timer3 are set up as real time clocks with each timer ticking away milliseconds, seconds, minutes, and hours. Timer1 is utilized by the second and millisecond delay functions *sdelay1* and *mdelay1*, respectively. With the exception of blinking the activity LED, the Timer3 resource is free and can be applied at your discretion.

The uDrive API is written in C using the HI-TECH PICC-18 C compiler. The first function called from the *main* function is the *init* function. Following the invocation of the *init* function, the PIC18F2620's I/O pins are initialized, the analog-to-digital converter is disabled, the timers are configured, and the interrupts for Timer1 and the EUSART are activated. The activation of the PIC18F2620 interrupt mechanism allows us to use the EUSART and timer resources, as they are interrupt-driven. Following a one second delay, the EUSART is initialized and enabled. The one second delay gives the uDrive the "alone" time it needs to properly set itself up and prepare to perform the auto-baud detection sequence. Once the PIC18F2620's EUSART is operational, an ASCII "U" (0x55) is transmitted to the uDrive. It uses this character to determine the PIC18F2620's EUSART baud rate. If all goes as planned, the uDrive returns an ACK (0x06) character to the PIC18F2620.

The Timer3 interrupt is activated if an ACK is returned for the autobaud operation and a five second delay period is invoked. The blinking of the activity LED indicates that the autobaud operation was a success. No blinky LED after five seconds means that the autobaud detection process failed. The activity LED will remain illuminated to indicate an autobaud error has occurred. The activity LED on my uDrive driver hardware is blinking. So, we have a GO to move on and initialize the disk drive memory card.

The uDrive-uSD-G1 Initialize Disk Drive Memory Card API Function

If a memory card is mounted when the uDrive powers up, the disk drive memory card is automatically initialized. If the memory card is inserted after power-up, the initialize Disk Drive Memory Card command must be invoked. For the purposes of demonstration, our API code always initializes the memory card programmatically. Here's the source code for the initialize Disk Drive Memory Card function:

```
//*******************************************************
//*     INITIALIZE DISK DRIVE
//*******************************************************
char init_disk(void)
{
        sendchar('@');
        sendchar('i');
        do{
                ++scratch8;
        }while(!(CharInQueue()));
    return (bytein = recvchar());
}
```

The initialize Disk Drive Memory Card command consists of two bytes: "@i." Now you see why the "i" in initialize is not capitalized in the command name. Once the disk drive memory card is initialized, an ACK is returned by the uDrive. All of the API functions report back to the calling function with a return code. The return code will be either an ACK or a NAK. This allows you to optionally error check each API call your application makes. For instance, here's how we invoked the *init_disk* function:

```
        rc = init_disk();
```

The return code (rc) can then be used to make a GO or NOGO decision like this:

```
        If(rc == uACK)
        {
                Your GO code here.
        }
        else
        {
                Your NOGO code here..
        }
```

Coding the Set Memory Address API Function

We'll need 32 bits of address information to read and write a byte at a time over the 2 GB of disk drive space we have. Those 32 bits of address information are exactly what the uDrive's Set Memory Address command needs to help us do those byte-wise reads and writes. The Set Memory Address command is called with "@A" followed by four bytes of address information. This command must be executed before any byte-wise read or write operation is performed. Our Set Memory Address API function looks like this:

```
rc = set_mem_addr(0x00000000); //call the API function

//*******************************************************
//*     SET MEMORY ADDRESS
//*******************************************************
char set_mem_addr(unsigned long addr)
{
        sendchar('@');
        sendchar('A');
        sendchar((addr & 0xFF000000) >> 24);
        sendchar((addr & 0x00FF0000) >> 16);
        sendchar((addr & 0x0000FF00) >> 8);
        sendchar((addr & 0x000000FF));
        do{
                ++scratch8;
        }while(!(CharInQueue()));
    return (bytein = recvchar());
}
```

I added the C statement to call the *set_mem_addr* function to show you what the address argument of the function looked like. In my example, we're setting the read/write memory address to 0x00000000. I chopped up the 32-bit address into eight-bit chunks and sent them along their way in order using the *sendchar* function. Once all of the six bytes of the Set Memory Address command have been sent, we loop and wait for an ACK or NAK to be returned to us by the uDrive. Let's use the Set Memory Address command and its API function code to build up a write-a-byte API function.

Writing and Reading a Byte to the Disk Drive

We only need to send three bytes to write a byte to a memory location within the disk drive memory card: "@w" + data. The format of the write-a-byte API function is identical to the Set Memory Address API function code:

```
//*******************************************************
//*     WRITE BYTE
//*******************************************************
char write_byte(char data,unsigned long addr)
{
        set_mem_addr(addr);

        sendchar('@');
        sendchar('w');
        sendchar(data);
        do{
                ++scratch8;
        }while(!(CharInQueue()));
    return (bytein = recvchar());
}
```

You can see the write-byte command sequence in the *write_byte* function source code. Note that we stuffed a *set_mem_addr* function call in at the beginning of the write-byte function code. The address information is included as an argument of the write-byte API function.

I'm positive that you've already logically deduced the command string for a byte read: "@r." So, without further ado, here's the read-byte API function source code:

```
//*******************************************************
//*     READ BYTE
//*******************************************************
char read_byte(unsigned long addr)
{
        set_mem_addr(addr);

        sendchar('@');
        sendchar('r');
        do{
                ++scratch8;
        }while(!(CharInQueue()));
    return (bytein = recvchar());
}
```

There's nothing in the read-a-byte code that you haven't seen before, other than the read- byte command string. Here's all it takes to read and write a byte to address 0x00000000 using the API code we've just built:

```
//*******************************************************
//*     MAIN SERVICE LOOP
//*******************************************************
void main(void)
{
        init();
        rc = init_disk();
        rc = write_byte(0x42,0x00000000);
        rc = read_byte(0x00000000);


        while(1);
}
```

The results of our write-a-byte and read-a-byte operations can be seen in the MPLAB IDE debug memory and Watch views shown in Screenshot 1.

Tossing Big Chunks of Data with the uDrive-uSD-G1

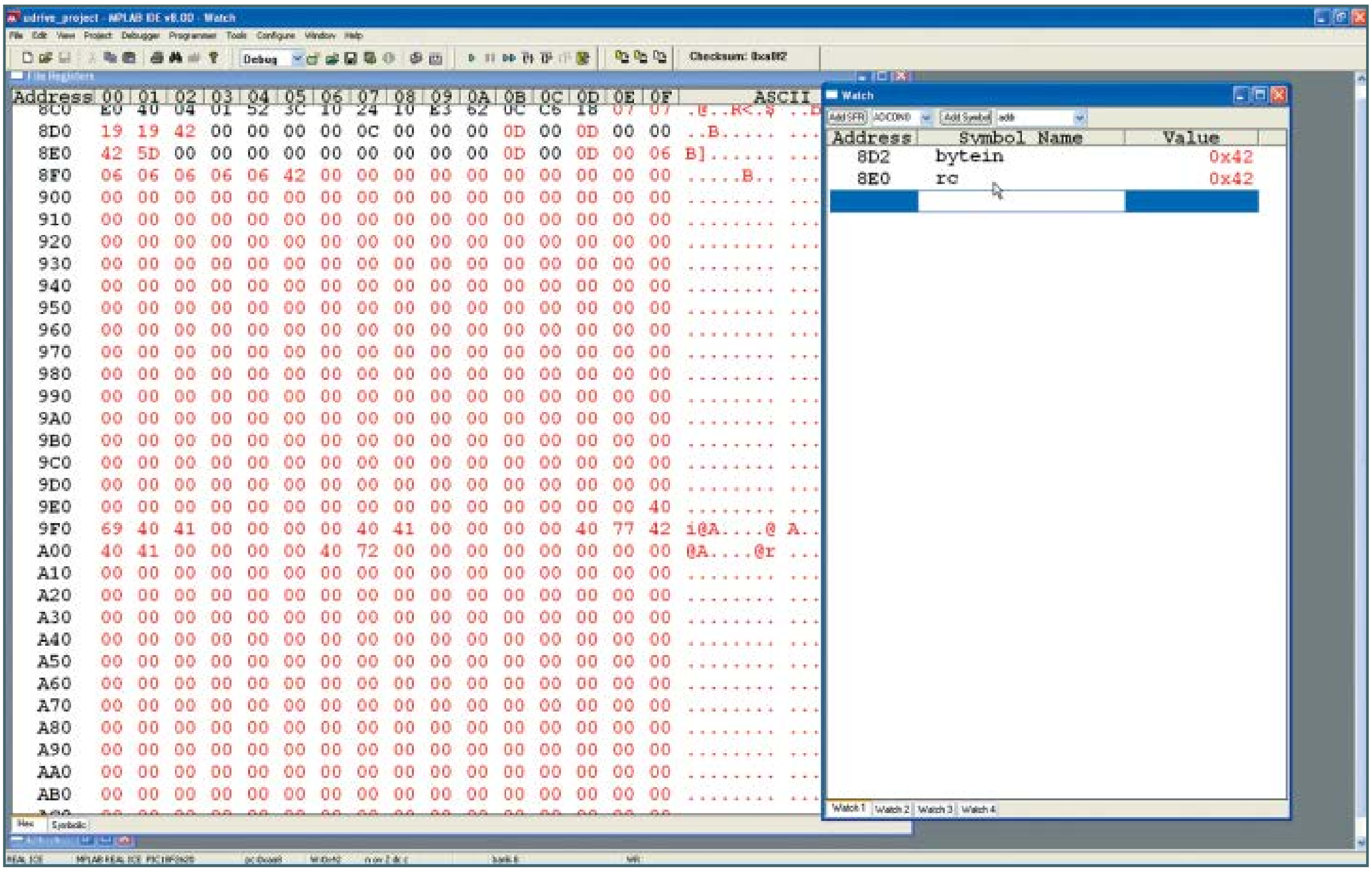
Throwing around single bytes of data can be useful, depending on the application. However, there will be times when you'll need to move large amounts of data. So, let's write two more API functions to do just that.

The command set contains a pair of commands that allow the user to read and write 512-byte sectors in a single operation. We'll use these commands (Read Sector Block Data and Write Sector Block Data) to build our sector-read and sector-write API functions. Let's put some data on the drive and see if we can read it back. Here's the sector-write API function source code:

```
//*******************************************************
//*     WRITE 512-BYTE SECTOR
//*******************************************************
char write_sector(unsigned int sector,char *buffer)
{
        char *byteptr;
        unsigned int index;

        byteptr = buffer;
        buffer = 0;
        sendchar('@');
        sendchar('W');
        sendchar((sector & 0x00FF0000) >> 16);
        sendchar((sector & 0x0000FF00) >> 8);
        sendchar((sector & 0x000000FF));
        for(index=0;index<512;++index)
        {
                sendchar(*byteptr++);
        }
        do{
                ++scratch8;
        }while(!(CharInQueue()));
    return (bytein = recvchar());
}
```

Note that we only need three bytes of sector address information, which is pushed out just like the address information in the Set Memory Address API function. The

SCREENSHOT 1. This is a shot of the MPLAB IDE debug memory area and the MPLAB IDE Watch window. In this case, the return code of an ACK or NAK is replaced by the byte that was received.



major difference in the API function from our byte API functions is the use of a buffer and a pointer that roams within the buffer. A 512-byte sector_buffer and a 512-byte read_buffer are allocated within the API source code. The *write_sector* function dips into the specified 512-byte buffer and sends the buffer's contents out to the uDrive byte by byte using the *sendchar* function. Before we can send any data, let's fill one of the buffers with something we can recognize. Here's some code that fills the sector_buffer with data:

```
scratch8 = 0;
for(scratch16 = 0;scratch16<512;++scratch16)
{
        sector_buffer[scratch16] = scratch8++;
}
```

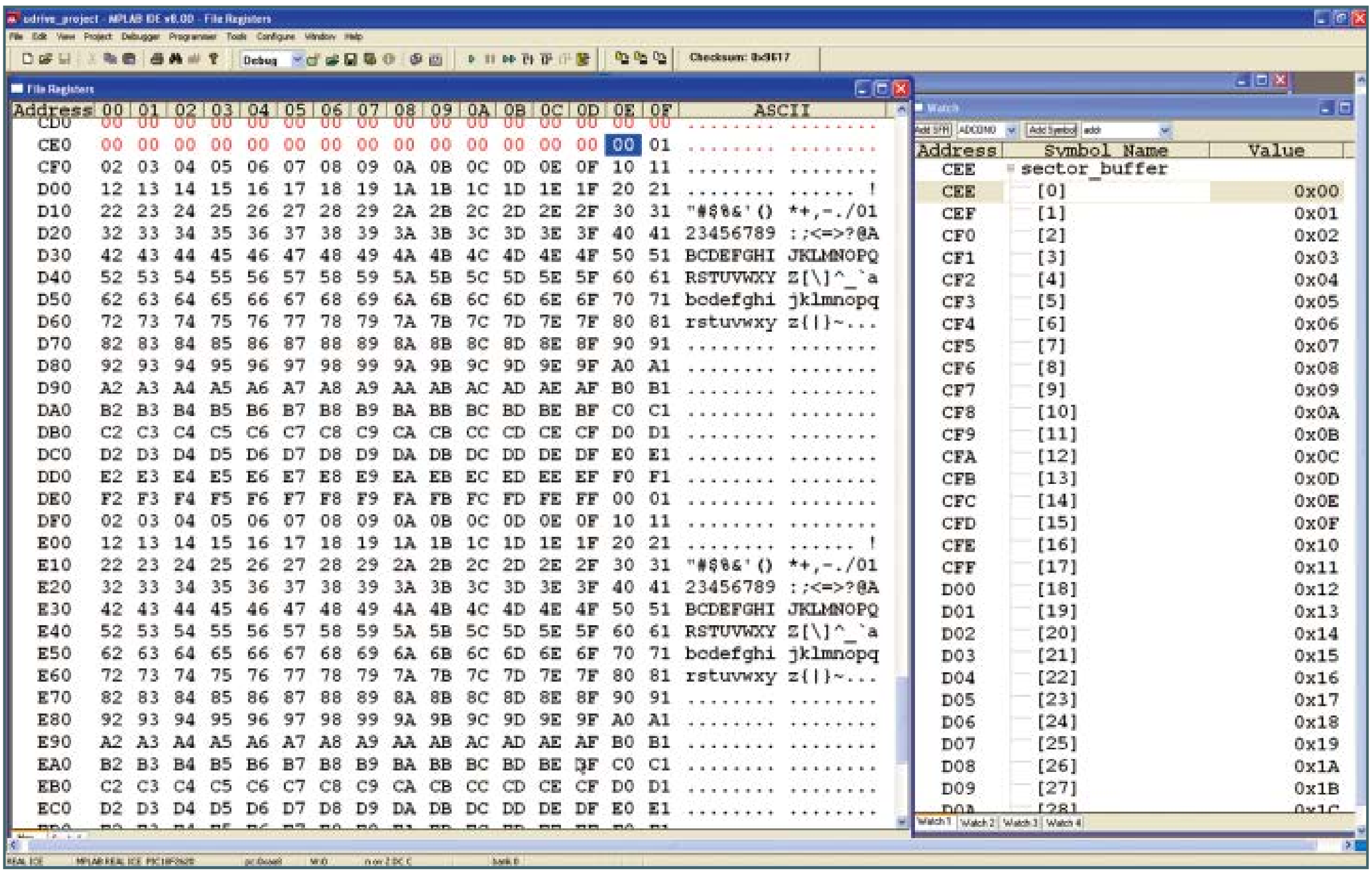
The sector_buffer is now filled with data that counts from 0x00 to 0xFF twice. That's equivalent to counting from 0x00 to 0x1FF, or 512 including the zero byte. The results of our sector_buffer-fill-up code are shown in Screenshot 2.

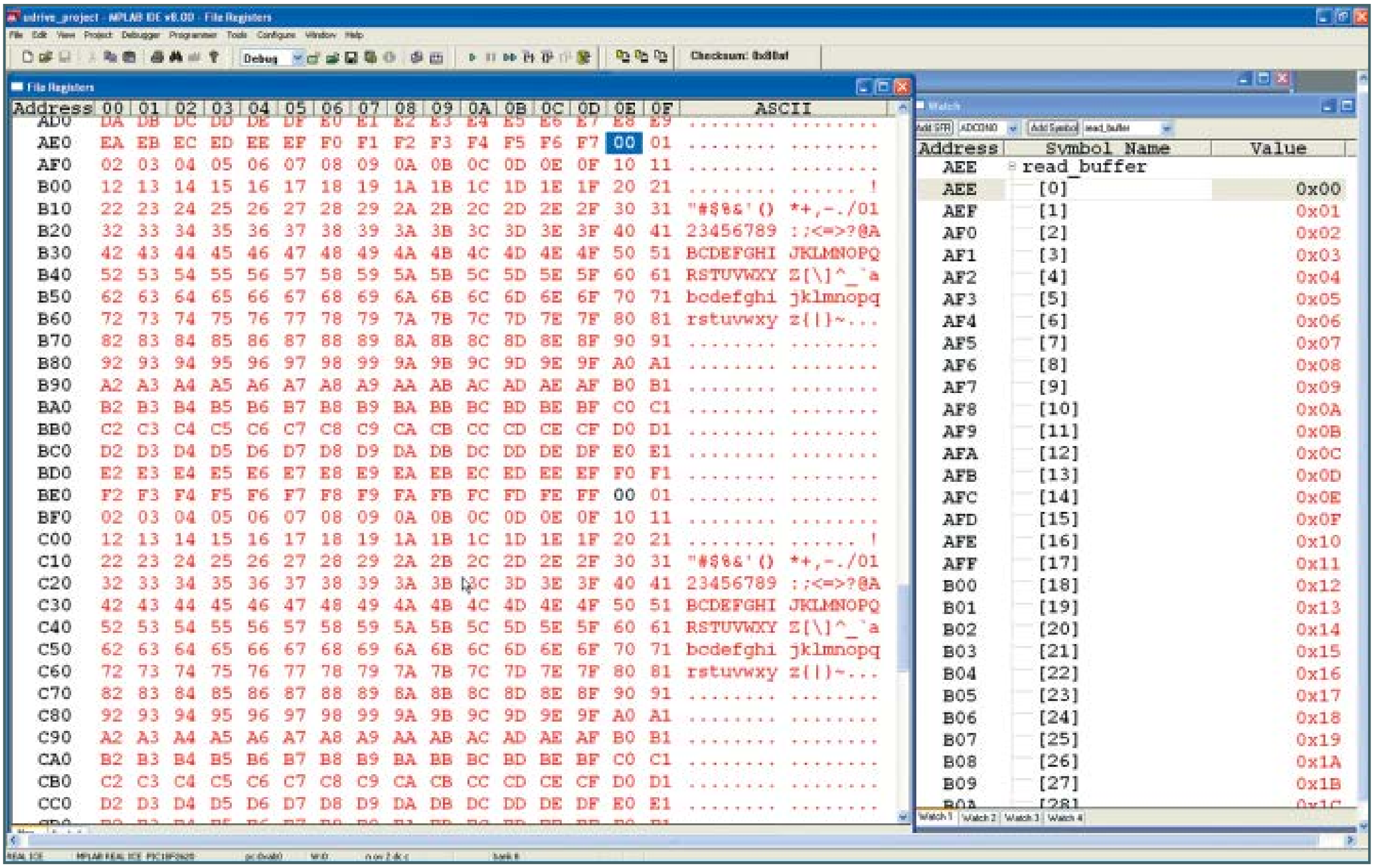
Okay ... now that we have something to write, let's put together a plan to read it when we're done. A read-sector API function would be nice, huh? So, here it is:

```
//*****************************************************
//*     READ 512-BYTE SECTOR
//*****************************************************
char read_sector(unsigned int sector,char *buffer)
{
        char *byteptr;
        unsigned int index;

        EUSART_RxTail = 0x00;   //flush receive buffer
        EUSART_RxHead = 0x00;

        byteptr = buffer;
        buffer = 0;
        sendchar('@');
        sendchar('R');
        sendchar((sector & 0x00FF0000) >> 16);
        sendchar((sector & 0x0000FF00) >> 8);
        sendchar((sector & 0x000000FF));
        do{
                *byteptr++ = recvchar();
                }while(*byteptr < 512);
    return (bytein = recvchar());
}
```

We've used the *write_sector* function as a model for the *read_sector* API function. The difference in the API functions is that the buffer pointer is used to place incoming bytes into position in memory rather than read them from a memory location. Note also that we cleared the EUSART receive buffer before issuing the call to the Read Sector Block Data command. Let's do some sector reading and writing:

SCREENSHOT 2. We simply counted from 0x00 to 0xFF twice to form the data pattern in the sector_buffer memory area of the PIC18F2620.





SCREENSHOT 3. The sector_buffer started at address 0xCEE. You can see that our read_buffer begins at offset 0xAEE. The sector_buffer and read_buffer are identical, indicating that the sector read and write API functions worked as designed.

```
//**********************************************************
//*      MAIN SERVICE LOOP
//**********************************************************
void main(void)
{
        init();
        rc = init_disk();

        scratch8 = 0;
        for(scratch16 = 0;scratch16<512;++scratch16)
        {
                sector_buffer[scratch16] = scratch8++;
        }

        rc = write_sector(0x0000,sector_buffer);
        rc = read_sector(0x0000,read_buffer);

        while(1);
}
```

The write_sector API function is called after filling the sector_buffer with our data to be stored. We saw that the sector_buffer data we generated did actually get written in Screenshot 1 to an area in the PIC18F2620 SRAM beginning at offset 0xCEE. Screenshot 3 shows us the read_buffer memory area that begins at offset 0xAEE. Note that the same ascending hex pattern you see in Screenshot 2 is shown in Screenshot 3.

Mission Accomplished

We have completed the assembly of a complete set of API functions for the uDrive-uSD-G1. Our API function calls will insure that each of the commands is transmitted correctly every time.

You have everything you need to build up your own uDrive-uSD-G1 system. If you have any questions or problems with your design, just tap out an email to me. I've posted the API source code and the ExpressPCB file on the *SERVO* website for your viewing pleasure. See you next time! **SV**

Fred Eady can be reached via email at fred@edtp.com.

Resources

Distributor – Saelig Company (**www.saelig.com**): UDrive-uSD-G1

Support – 4D Systems (**www.4dsystems.com.au**): uDrive-uSD-G1

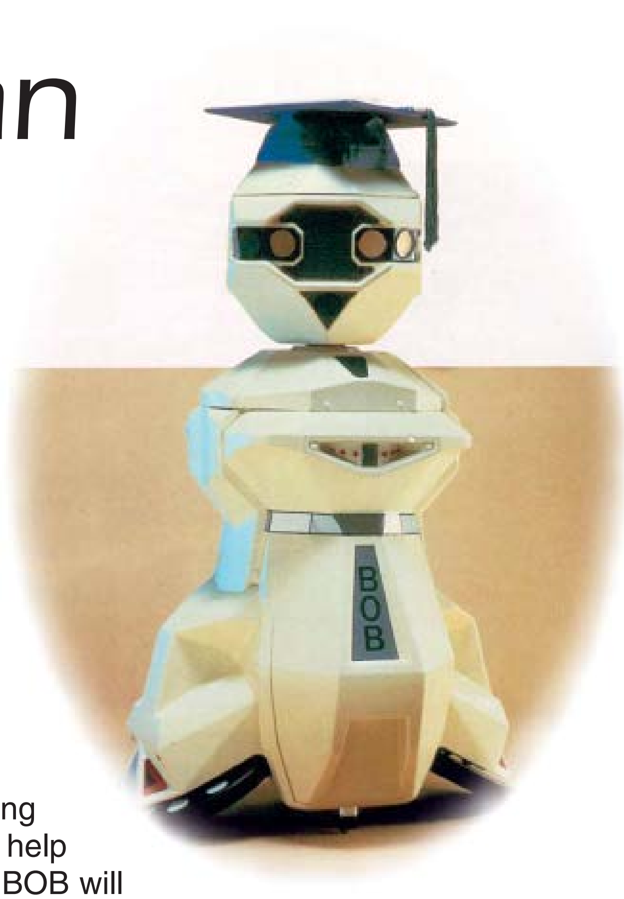
HI-TECH Software (**www.htsoft.com**): HI-TECH PICC-18 C Compiler

National Semiconductor (**www.national.com**): LM2940; LM3940

Microchip (**www.microchip.com**): PIC18F2620; MPLAB IDE

ExpressPCB (**www.expresspcb.com**): ICSP Dongle Printed Circuit Board

Reviving an
Androbot BOB



PART 4 — Extra sensors, original sensors, wireless operation, autonomy, and programming

by Robert Doerr

I hope that those of you following this series have learned a few new things! The previous articles have primarily focused on the hardware aspect of reviving BOB. They have covered taking an empty robot shell, adding a new brain with larger H-bridge drivers, working with the sonar sensors, and adding a co-processor to help offload tasks from the main processor. This article on BOB will cover adding some extra sensors and the programming of the main controller itself.

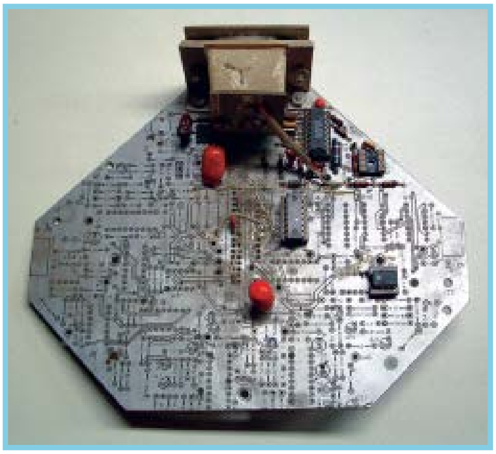
Some of the types of sensors we'll use to enhance BOB will be a couple of light sensors, a temperature sensor, a sound sensor, and a compass. The Handy Board firmware has low-level code to control the drive motors, the LCD, and read the analog/digital ports. There are also several custom add-on libraries that are useful, but they are still generic. To supplement these, the lower level BOB specific routines are built which will use these new sensors and talk to the co-processor. Finally, some higher level functions will be built upon those which can start to add some autonomy to BOB so he can move around and interact with his world.

A Step Back Before Going Forward

Last month covered the co-processor that controls the head and the lights on BOB. Their control all work great. However, I got an un-expected surprise when I went to wire up the footlights. This is one of those things that can cause plans to change along the way. To test everything on the bench and help debug the code for the co-processor, I used an extra pair of footlight boards from a TOPO II. These footlight boards have a pair of NPN transistors which handle switching each set of three LEDs on and off. These are meant to run at 12V for the LED power and are controlled with a normal 5V logic signal. When I pulled out the footlight boards that were on BOB, I discovered these were based on the earlier TOPO I footlight boards, but were customized for BOB. Instead of having an onboard transistor for driving the LEDs, these boards just had two sets of three LEDs wired in series. What is unique about these compared to the TOPO I board is that the center LED for each set of three was just shorted across with a solder bridge and short length of wire. I suspect the reason for this was so it could be directly driven from a 5V supply. If all three LED were left in series, then 5V wouldn't be enough to turn them on.

Well, this wasn't really a problem. I had run out of room anyhow for any new circuitry on the SX48 co-processor board and my test harness can still be used on the bench. The solution was to make a small driver board to drive the BOB footlight boards. It plugs into the existing connector on the co-processor where I had plugged in the cable I used on the bench. The driver board has the current limiting resistors and driver transistors for the LEDs.

A lot of this BOB project has been about re-using odd parts, so I used some unmarked NPN transistors for the LEDs. Without knowing the exact specs, there aren't too many places I would have used them but they are perfect for switching LEDs like this. Since we're only turning on/off some LEDs, just about any NPN transistors will do. The connection from this driver to the footlight boards is accomplished with a pair of eftover CD-ROM audio cables. Each cable has three wires inserted into four pin housings. It just took moving one of the pins in the connector and then trimming off the connector housing to make it into a three position housing. With this new driver in place and wired up, BOB now has some more lights on the body under our control.



IR sensor and original electronics.

Both IR sensors and sonars (before cleaning).

Bob Needs More Input

Currently, BOB has a limited amount of input he can get from his surroundings. He can get readings from his internal encoders to verify that his head or wheels moved a certain amount. He can also get distance by reading the five sonar sensors in his head to get an idea of what his surroundings are. It would be useful for BOB to get some more details of the world around him. Today, there are a huge variety of sensors available. We're going to add some of the basics here to get started. I'm not going to add any that would require cutting the shell or altering the original robot. That limits the selection a bit, but there are some we can easily add and I expect that BOB will eventually get a lot more down the road.

To start, there are some tried and true sensors that are easy to add to the Handy Board. These would be an analog temperature sensor, a sound level sensor to detect ambient sound levels, and a pair of simple photo resistors to provide ambient light levels. Another sensor that is a good choice is a Vector compass module or Dinsmore compass sensor. Either one of these will give BOB an idea of which way he is heading. I've had a Dinsmore digital sensor sitting on the shelf for years and BOB will be the perfect project to finally use it.

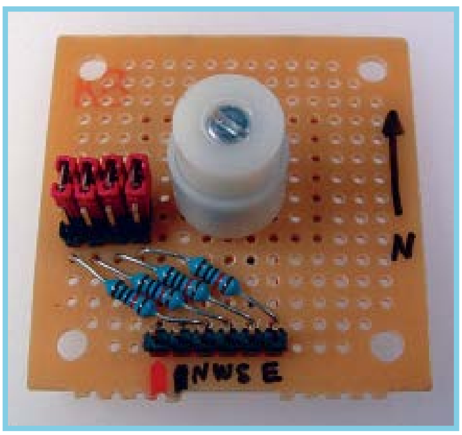
One of the first sensors I added was an LM34 temperature sensor. This can normally be wired directly to the analog port on the microcontroller. The Handy Board has 47K pull-up resistors on the analog ports which prevented me from just plugging the sensor directly into one of those ports. One solution was to cut the trace for the pull-up resistor on the analog port I wanted to plug the sensor into. This trick would have worked for analog ports 2 through 6 since each one has a dedicated pull-up resistor. The ones on the expansion board share a common pull-up for eight inputs, so if it was disabled it would have affected a group of analog ports instead of just one.

A much easier solution presented itself. The Handy Board inside BOB never had the thumbwheel installed. The thumbwheel is just a 20K adjustable trim pot that goes to an analog input without a pull-up resistor. I just soldered the LM34 temperature sensor in that spot and can read it directly with the built-in **knob()** function which returns the analog value read. This made use of an analog port I wouldn't have normally used and kept one of the regular ones open for something else later on.

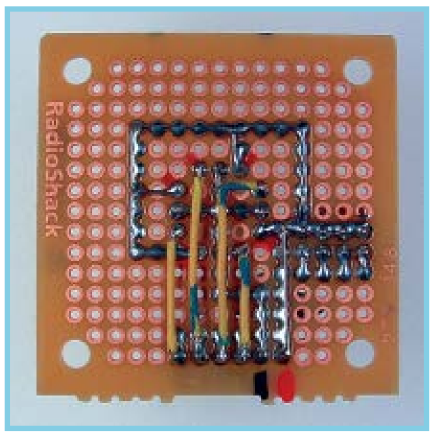
A pair of photo resistors for light level sensing is the next type of sensor BOB is going to get. The photo resistors are easy to connect to the Handy Board. Each one directly plugs into an analog port on the Handy Board. One lead goes into ground and the other to the analog input. That's it! These will be mounted in the head; one near the right side and the other near the left. With these, BOB can look around to find the brightest light source and then follow it by keeping track of the difference between the two photo resistors. This is a very common experiment that you've probably seen used on other robots like the Parallax BoeBot.

Adding a compass module will definitely help give BOB

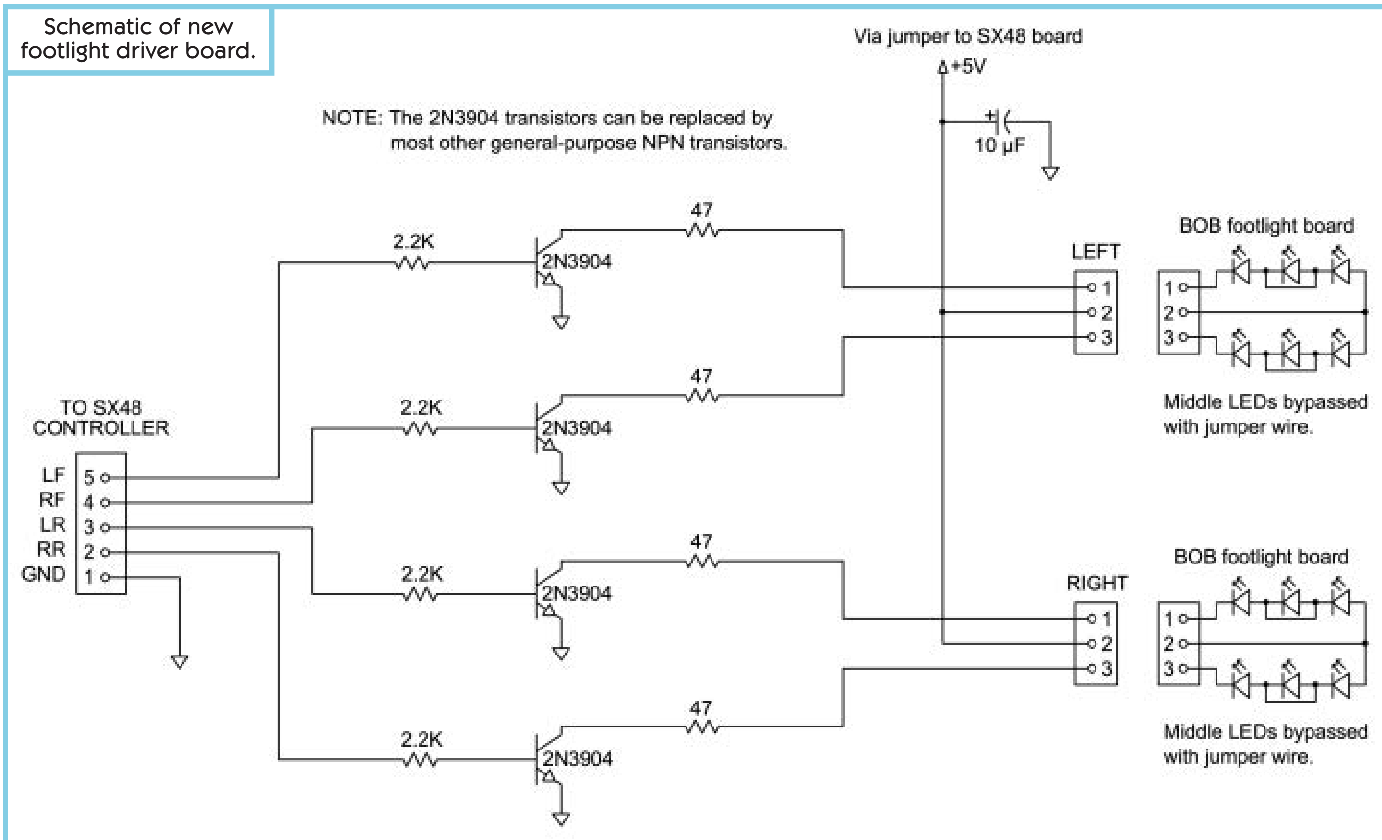
Dinsmore compass module (component side).



Dinsmore compass module (solder side).



Schematic of new footlight driver board.



a sense of direction. In this instance, we're installing the Dinsmore compass sensor. The Dinsmore sensors come in a couple different variants. The 1490 sensor can output eight digital compass positions (N-NE-E-SE-S-SW-W-NW) while the 1525 sensor is analog based and outputs a continuous analog sine/cosine signal. This is capable of higher resolution than the digital sensor. The actual accuracy will depend upon the resolution of the ADC (analog-to-digital converter) it is connected to.

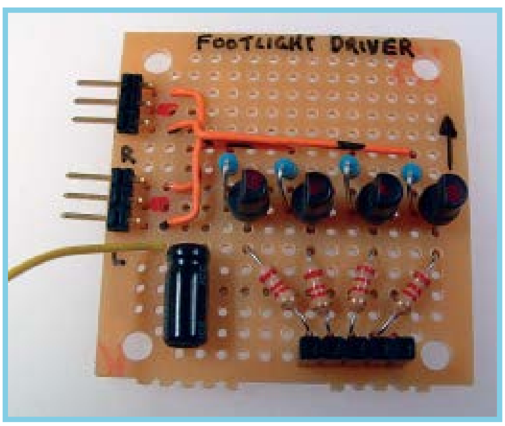
In this project, BOB will be using the 1490 digital version since that is what I had on hand. The placement of the actual sensor is important since it should not be located too close to the main drive motors, head motor, or speaker. Basically, the sensor should not be near anything that may generate stray magnetic fields which may affect the readings of the compass module.

The wiring of the digital compass is straightforward as it just has four digital outputs; one output each for North, South, East, and West. Depending upon direction, either one or two of these will be active. This will give us a total of eight possible valid combinations. These would normally just be connected to digital inputs but they've already been used up. No worries though as we can just connect these four lines to the analog ports instead and treat the values we see as digital ones.

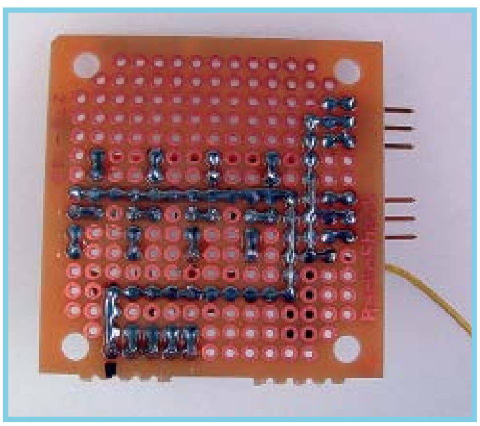
This may seem odd at first, however, this works just fine as long as we take the analog reading and treat any low value (less than 128) as a 0 and any value over 128 as a 1.

With only eight different readings, BOB will only get a general direction of which way he is facing. There is a way we can improve the accuracy, however. For example, say we want BOB to point as close to North as possible. If we just go by watching the North bit to see when it is the only one on, we'll know we're pointing in that direction but we could be right

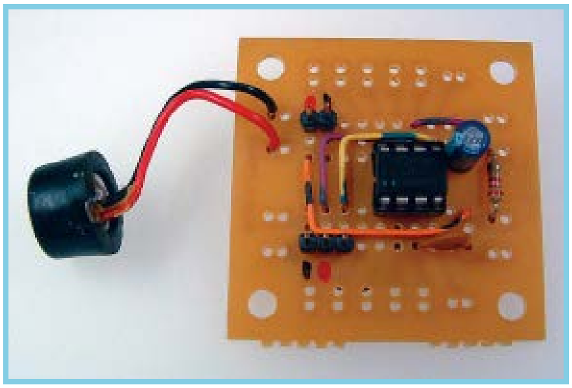
Footlight driver board (component side).

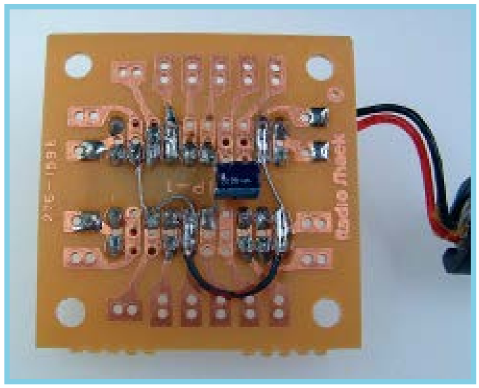


Footlight driver board (solder side).



on or almost 22.5 degrees off in either direction. To get a better idea of where North really is, BOB can turn until either the East or West bits toggle on. From there, he can keep track of how far he has to turn until the opposite side toggles on. By splitting the difference and turning back, he should be fairly close to North. To keep track of the turning, we can either time the movement or use the encoders on the main drive motors. This is a good way to start combining the data from multiple sensors and improve BOB's view of the world.


Audio amp (component side).


Audio amp (solder side).

BOB is also going to get some rudimentary sound level detection. It will give BOB an idea of what the ambient sound levels are. This is done using a simple condenser microphone and an amplifier based on an LM386 chip. The microphone came out of an old cordless phone I had. The output of the amplifier is connected to the analog 6 input on the Handy Board. Nothing fancy here. We can watch the value on this port to look for drastic changes, which could be things like hand claps, etc.

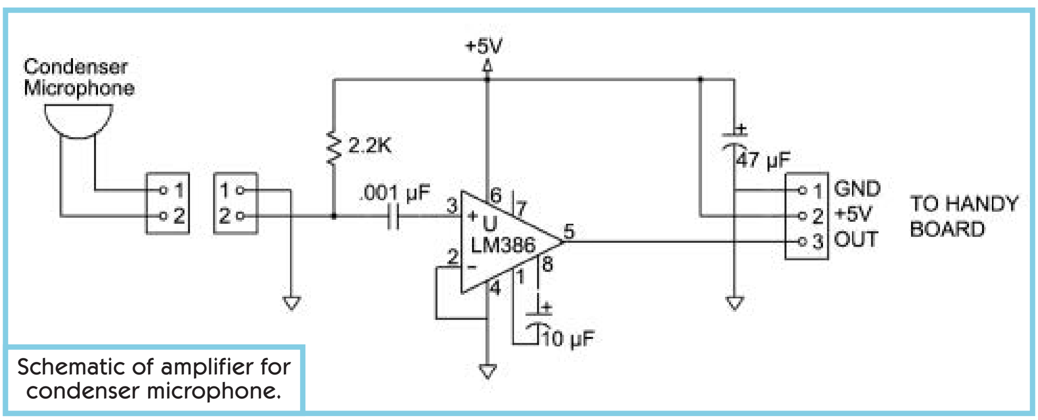
There are two more sensors that I haven't yet had a chance to connect. These are the two original IR sensors that I found inside BOB's head. One seems to be an IR motion sensor and the other an IR heat sensor. These two infrared sensors were supposed to be a human detector. The original Androbot flyer for BOB says "Infrared sensors, attuned to the frequency of human body heat, enable BOB to locate human beings." Another section says, "His infrared sensors then 'recognize' people (while differentiating between human body temperature, fireplace heat, and incandescent light heat)." Wow! That sounds like it would be really slick to get wired up and working again.

I still need to figure out exactly how these sensors were supposed to work. I started by gathering up all the datasheets for the chips used on the board. With those in hand, I am tracing out the schematic for the hand wired circuit board in the head. It is taking a while to sort out and I hope to get some sort of data from them again. In practice, I don't know how well they ever worked.

Years ago, I had talked to an Androbot engineer who remembered BOB and said that part was always troublesome. It was a bit too sensitive and supposedly could tell the difference between a white and black file cabinet. Seeing the difference in radiated heat, he would stop and think it was a person. It would have been pretty funny to see BOB trying to strike up a conversation with a file cabinet.

HOW DID COLUMBUS DO IT?

As a test to make sure I had the Dinsmore compass module wired up and working correctly, I used a real compass as a sanity check to make sure they agreed. Before I could do this, I had to buy a compass since the one I had was lost over the years. You'd think just buying a compass would be an easy task — Not! I was shocked at the poor quality and choices available. The first one I picked up didn't seem to move as freely as I would have expected so I questioned just how accurate it would be. I didn't need to be within a degree or two but at least wanted to be in the ballpark. To my dismay, when I looked at all of the compasses sitting on the shelf most of them all pointed in different directions! Needless to say, I tried another store that carried more brands. It seemed that everywhere I checked many of the compasses were off, too. Finally, I found several different compasses from various manufacturers that all agreed and picked one of them. During the trip home, the compass agreed with the one in my car and also with the direction of the local roads. It makes me wonder just how many people purchase a compass without checking them and assume they are okay. The important point here is to double-check things like this before you buy and not to assume they will be right. It is critical if you are using it as a reference to check other devices.

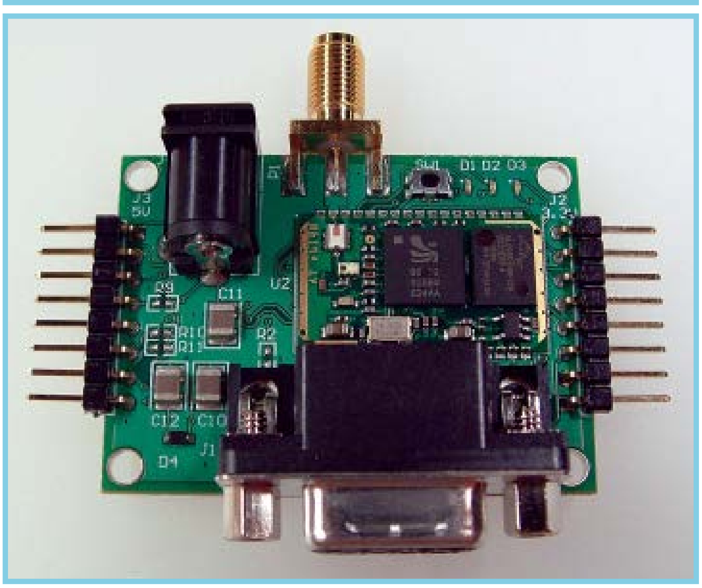


Schematic of amplifier for condenser microphone.

New firmware loaded on co-processor (also mounting of footlight driver).

Getting New Code Into BOB

Whether the Handy Board is a new one and the firmware needs to be downloaded, you're just sending down the latest Interactive-C program, or you're just using the Interactive portion of Interactive-C, you need to have a connection to the Handy Board. This normally requires connecting a serial cable to a small charging adapter which then connects to the Handy Board via a four-pin RJ-11 cable. The manual firmware download is a bit finicky in that the timing requirements are fairly strict and it uses a slower rate (1200 baud) than the normal communication speed (9600 baud) used by Interactive-C. Since it is rare that the firmware needs to be refreshed (thanks to the addition of the NVRAM mod), this isn't much of a problem. We only need to do that again if the NVRAM gets corrupted somehow or we want to download a newer image of the firmware.



Close-up of A7 Engineering eb501-SER Bluetooth adapter.

Since BOB has his own brain on board, I wanted him to be autonomous like he was originally intended to be. I needed to easily send down new code as new features and routines were constantly created. The ability to use the interactive portion of Interactive-C helped speed development along. The solution was just to cut the cord and go wireless. Bluetooth works well to create this wireless link. On BOB, I am using an eb501-SER Bluetooth module from A7 Engineering that has both RS-232 level and logic level serial signals available. The connection to the Handy Board is pretty straightforward and only requires +5V, ground, RX, and TX. It just requires a custom four-pin adapter made from an old phone cable.

One end was cut off and a six-pin connector was installed to hook up to the 5V logic level connector on the A7 Bluetooth module. The four-pin modular connector on the Handy Board has the ground, RX, and TX already present. However, it was lacking the +5V supply since the power lead on the RJ-11 jack is normally used to charge the Handy Board. Since the Handy Board in BOB has been modified to run off a 5V DC-DC converter, another modification was done to route this +5V to the RJ-11 so it can power the Bluetooth module. The adapter for the hardwired connection was also modified so that it can be powered from the Handy Board, as well.

Many newer systems already have built-in Bluetooth adapters. However, these may be a Class 2 device with a range of only 10 m (33 ft). The adapter I used on the PC was a ZOOM Bluetooth USB adapter, model 4310. This is a Class 1 device with a small antenna so it is rated at 100 m (330 ft) to match the A7 Bluetooth adapter on the robot. This provides an excellent range of operation. The Bluetooth adapter looks like a serial port on the PC. The first time the adapters are used they have to be paired up with each other. It is a fairly straightforward process on the A7 module, but may be different for other Bluetooth adapters.

Once this is done, it should not need to be done again unless one of the adapters is swapped out. Now whenever BOB is powered up, it shows as an available serial port. It works great for downloading new programs into BOB and also working with him interactively. The only portion that poses a problem is the rare occasion for new firmware. It hasn't been a problem yet, but I do plan on wiring up a DPDT switch to toggle between the Bluetooth serial device and the hardwired serial connection. If I ever do have to send down the low level firmware again, I can just flip a switch, plug in the Interface board, and send down the code. Once there, just flip the switch back and go back to wireless operation

Building up BOB's Code Base and Adding Autonomy

The code for the little co-processor doesn't change too often. If that code is revised, it is a manual process of plugging in an SX-Key on the co-processor board and flashing the new code onto the SX48 chip. About the only other features I may add is to allow the LEDs to fade in and out and perhaps add a maximum speed for head movements. Other than that, all of the focus will be on building up the custom library of routines specific to BOB and higher level code upon that.

Photo resistors for BOB.

With the exception of the original TOPO robot, the original programming for both the TOPO II/III and BOB was done in Forth. There is good documentation on the Forth routines in the TOPO II/III robots which provides a good reference for the original function names and coding style. Although most of the original code for BOB has been lost, a good friend of mine (Bob Wind) helped me uncover some of the routines for the BOB/XA robot. Although we are writing all the new code for BOB in Interactive-C, we can certainly name many of the functions and routines with names like the original ones. This way, BOB will be programmed a bit like it was originally. At least as close as it can be considering we are using a different language! We'll start out with the lower level functions specific to BOB then we'll add a bit of smarts and autonomy to make him act more like he was supposed to.

Besides the description of the functions from the TOPO manual and the fragments uncovered from BOB/XA, we can get other clues as to what BOB was supposed to do from the advertising brochure: "A list of routines from the Standard cartridge are: random walk, random speech, obstacle avoidance, and self diagnostic check. He can also flash lights while singing and check his battery level." This should all be possible with the new electronics on board! (I'm sure that he'll be able to do all this and more.)

A while back, I received an email from an Androbot engineer who remembered BOB and said, "It rolled along a wall using ultrasonic sensor data to maintain a constant distance from it, and emitted speak-and-spell phrases at random when its forward-pointing sensors got an echo from less than x feet. A bit of showmanship can do much with such a machine, but the supposing happens in more intelligent creatures." With the new electronics on board, BOB should easily be able to do this again.

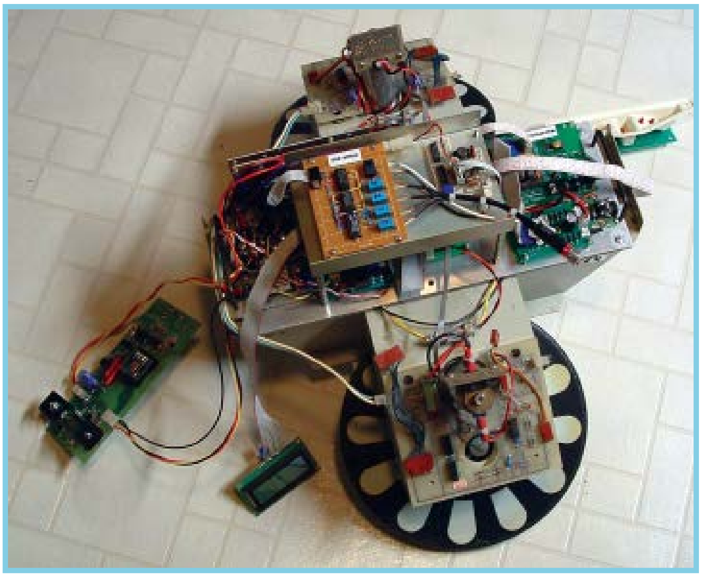
Since the Handy Board already has a good library of low level routines built in, the next step is to add some low level code that is BOB-specific. Some examples of this start with controlling the LEDs on the body. We can either display a pattern on the chest of BOB or we can control the footlights near the wheels. These new commands are:

Both sides of Bluetooth link.

```
// where value is the 8-bit pattern to
displayCHESTLIGHTS(value);

// where value is the pattern to display
//   (in lower 4-bits)
FOOTLIGHTS(value);
```

Both of these are very simple routines. First, they just use the **select_SX();** to make sure that the serial output

BOB at Consumer Electronics Show (CES) show years ago!



Chassis with electronics installed.

will go to the SX48 co-processor. Then, it sends the appropriate command ("L" for the chest lights or "F" for the footlights), followed by the bit pattern to display.

Next would be functions for controlling the head. The co-processor makes this fairly easy since we just tell it what to do and don't have to worry about the details. Some of the commands dealing with the head movement are:

```
HEAD_MOVE(hdegrees);     // signed position in half
                         // degrees (90 = 45 degrees)
HEAD_HOME();             // return head to home position
                         // (pointing forward)
HEAD_CLEAR();            // Clear HdErr flag on copro
                         // (in case head moved);
HEAD_BUSY();             // Return the status of the head
                         // (busy moving or not)
```

These each use the **select_SX();** to ensure the commands go to the SX48 co-processor and then send the command and any required parameters. If we want to make the head move 45 degrees to the right, we just use **HEAD_MOVE(90);** where 90 is the number of degrees times 2. A value of -90 would send it 45 degrees to the left. If we are moving the head to take sonar readings, we can use the **HEAD_BUSY();** function which can tell us if the head is still moving or not. The **HEAD_CLEAR();** function will clear the flag which we use to determine if someone moved the head manually.

There are also some low-level routines for the SpeakJet which send out a phonemes string for the speech, check to see if the speech is active (still speaking), and another that will make random R2-D2 sounds:

```
SAY(stringpointer);       // Say the string pointed to
                          // (terminated by 255)
SPEECH_DONE();            // returns 0 if speaking or
                          // 1 if done
RESET_SPEECH();           // sends code to SpeakJet to
                          // reset speech parameters
SET_SPEECH_RATE(speed);   // Sets the speed of the speech
SET_VOLUME(volume);       // Sets the SpeakJet volume
R2();                     // Make a few beeps like R2-D2
```

All of these start by using the **select_SJ();** function to make sure the phonemes all go to the SpeakJet instead of the SX48 co-processor.

Next would be some core routines for using the main drive motors. These use the built-in motor control features of the Handy Board. Currently, they use a time delay to determine distance, but they will eventually be revised to incorporate closed loop control using the encoders for more accurate movements. Some of the BOB-specific movement routines are:

```
FWD(distance);        // distance is float value in inches
BACK(distance);       // distance is float value in inches
LEFT(degrees);        // turn left in degrees
RIGHT(degrees);       // turn right in degrees
GO_ANGLE(degrees);     // uses LEFT(); or RIGHT();
                      // depending upon sign of angle
```

With these building blocks, we can now combine the built-in Handy Board libraries and the BOB-specific code to start building more interesting routines which will give BOB some character. Below are some examples:

```
SAY_NUM(value);  // Say the signed integer value
SAY_COMP();      // Say the compass position
SAY_SONAR(sensor); // Take sonar reading and speak
                   // the value
TALK_FAST();     // Make following speech fast
TALK_SLOW();     // Make following speech slow
SCAN_TEST();     // Test that turns head and takes sonar
                 // readings along the way
CHESTDEMO();     // Cycle the lights back and forth on
                 // chest lights
```

As mentioned previously, one of the aspects of Interactive-C that I really like is the interactive part. It makes it easy to try out new routines and see how they work before adding them into the main program. It is also nice for demonstrating the robot as new routines can be written on-the-fly. Some examples of this scripting:

Continually flash the two inner LEDs on chest panel:

ENGINEERING A POLITE CONVERSATION

Interactive-C on the Handy Board can support multiple processes. It is important to ensure that only one process talks to the SpeakJet and SX48 co-processor at a time. Otherwise, the processes may end up talking to these at the same time and clobber the communications. To avoid this, you can talk to them through a single process. Another option is to have some global flags defined in your code so that each process can check to see if it is safe to communicate with the SpeakJet and SX48 co-processor.

ANDROBOT BOB HANDY BOARD PORT ASSIGNMENTS

The Handy Board is currently set up as the main controller for an early prototype BOB robot. Here is a list of the assignment and meaning of each I/O pin on the Handy Board. Also listed are the built-in ports for motor control, etc., which show how they are wired into the robot. Some notes about each one are shown as a reference to explain how they are used within BOB.

BUILT-IN HANDY BOARD PORTS:

***RJ11 Connector**: Serial connection (direct downloading of firmware/wireless Bluetooth operation)*

Pin #	Description
1	NC
2	RX - Input
3	+5V from Handy Board (this is customized for BOB) Can power Bluetooth or Serial Interface
4	Ground
5	TX - Output
6	N/C

***Motor Ports:** Used for main drive motors*

Port	Description
0	Left Drive Motor (to LMD18200 H-bridge)
1	Right Drive Motor (to LMD18200 H-bridge)
2	Open – TI75540 H-bridge on Handy Board
3	Open – TI75540 H-bridge on Handy Board

Analog In:

Port	Description
0	Pass Through to Expansion Board
1	Pass Through to Expansion Board
2	Open
3	Open
4	Open
5	Open
6	Ambient Sound Level

Digital In:

Port	Description
7	Sonar ECHO (from J12 on expansion board)
8	SpeakJet Status (buffer half full)
9	PA7 – Serial Out to SX48 and SpeakJet
10	Left Encoder – Main Drive
11	Left Encoder – Main Drive
12	Right Encoder – Main Drive
13	Right Encoder – Main Drive
14	SpeakJet Status (Speaking)
15	SpeakJet Status (Ready)

EXPANSION BOARD PORT CONNECTIONS:

Analog In:

Port	Description
16	Head Busy (from SX48 controller)
17	Head Error (manually moved)
18	Compass – North
19	Compass – West
20	Compass – South
21	Compass – East
22	Smoke Detector
23	Battery Voltage/Level
24	Ambient Light Level – Left
25	Ambient Light Level – Right
26	Motion Detect (reserved for original I/R)
27	Motion Detect (reserved for original I/R)

Digital Out:

Port	Description
0	Bit 0 – Sonar Select
1	Bit 1 – Sonar Select
2	Bit 2 – Sonar Select
3	Serial Select – Low (0) SX48 Co-processor, or High (1) SpeakJet
4	Open
5	Open
6	Open
7	Open
8	Open

LEGO Sensor In:

Port	Description
28	Open
29	Open
30	Open
31	Open

Servo Outputs:

Port	Description
0	Open
1	Open
2	Open
3	Open
4	Open
5	Open

Miscellaneous Handy Board connections:
START Button: Tap Switch on BOB's Head
STOP Button: Hidden Switch Inside (used for Firmware upgrades)
Thumbwheel: Ambient Temperature (LM34)
I/R Output: Unused
I/R Input: Unused (may be used for I/R remote control)

TOPO BATTERY MONITOR BOARD (USED ON PROTOTYPE BOB):

***J1:** Power in from batteries*

Pin #	Description
1	Ground
2	+12V
3	+24V

***J2:** Switched power out to logic/drive*

Pin #	Description
1	+12V
2	+24V

WEB REFERENCES

RobotWorkshop (Author's website)
www.robotworkshop.com

Norland Research: Small TOPO clone and Androbot info
www.smallrobot.com

Main Handy Board site
www.handyboard.com

Botball website (keeper of the latest Interactive-C code)
www.botball.org

Botball webpage for Interactive-C
www.botball.org/educational-resources/ic.php

A7 Engineering (Bluetooth modules)
www.a7eng.com

Zoom Technologies (USB Bluetooth adapters)
www.zoom.com

Wikipedia article on Bluetooth
http://en.wikipedia.org/wiki/Bluetooth

Dinsmore Compass Sensors
www.dinsmoresensors.com

LM34/35 Temperature Sensors Application Note
www.national.com/an/AN/AN-460.pdf

Datasheets for ELTEC Pyroelectric Detectors
www.silverlight.ch/eltec/eltec_download.php

{while(1) {CHESTLIGHTS(1); sleep(.5); CHESTLIGHTS(2); sleep(.5);}}

Display random LED patterns on chest panel:

{while(1) {CHESTLIGHTS(random(255)); sleep(.5);}}

Move head 45 degrees right, go forward three inches, turn head 90 degrees left, back up two inches, home head, and then move forward a couple inches:
{HEAD_MOVE(90);FWD(3.0);HEAD_MOVE (-90);BACK(2.0);HEAD_HOME();FWD(2.0);}

Turn on all footlights, BOB says his name, move right 35 degrees, wait a bit, turn off the footlights, BOB says he'll do your bidding, move head 45 degrees right, turn left 70 degrees:
{FOOTLIGHTS(255);SAY(sp_bobnm); RIGHT(35); sleep(2.0); FOOTLIGHTS(0); SAY(sp_servant); HEAD_MOVE(90); LEFT(70);

As you can see, having the ability to call the C functions on-the-fly is a really nice feature and can help with development and demos.

Conclusion

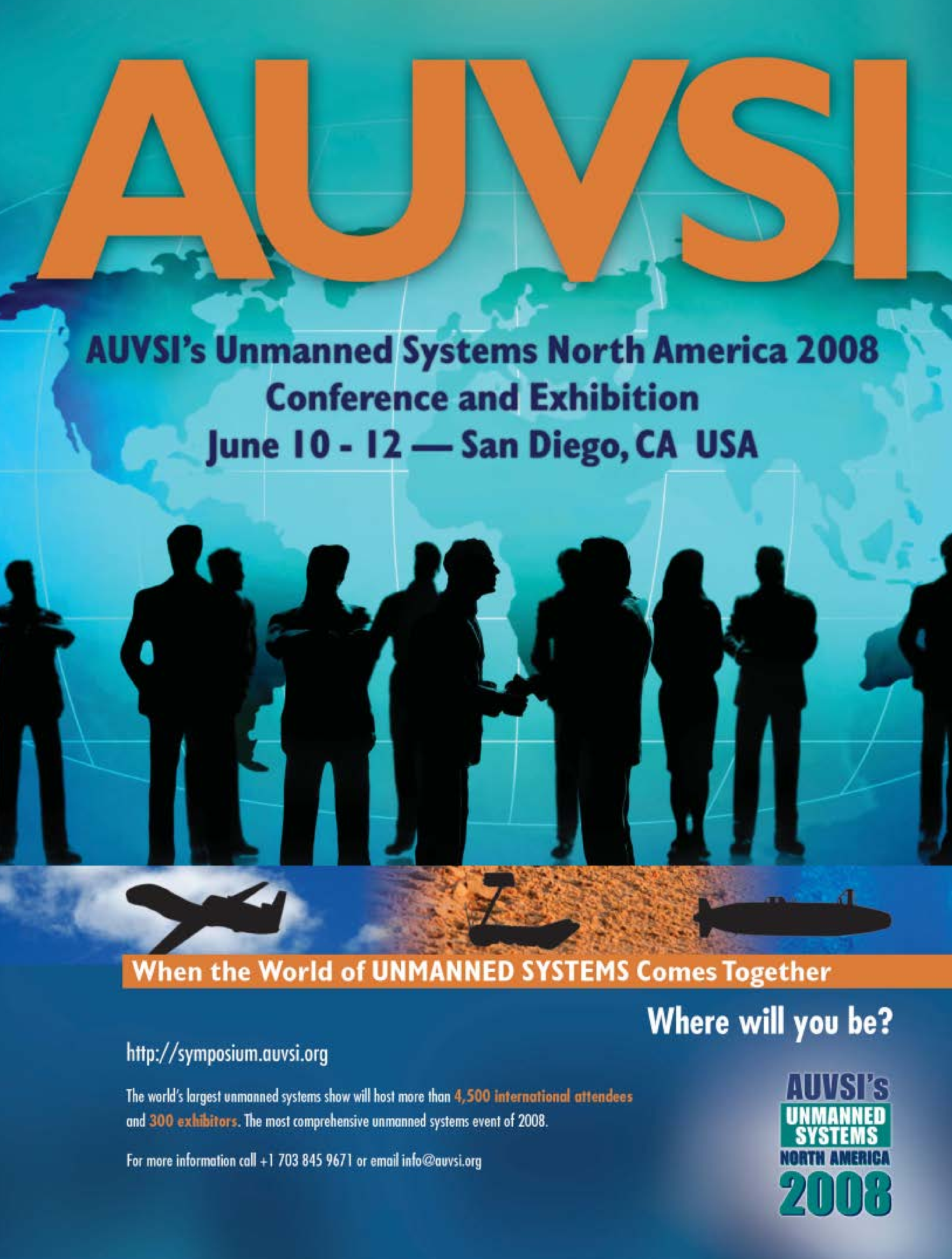
Hopefully, this has provided a good overview of how BOB is programmed. I would have liked to go into detail on every aspect but there is only so much you can put in each article and this is already running long. This whole project came out better than I could have expected and leveraged a lot of idle robotics parts that needed a purpose. It also recycled many used parts, as well. Since it all just bolted in place and plugged into the existing wiring, it was all done without any modifications to the robot itself. If the original electronics are ever found, all of this could be moved into another robot. Eventually, he will be able to do a lot more once some other sensors are installed and routines created. At any rate, it is good to see BOB moving again. I think he is glad to be back, too.

Keep those old robots alive! **SV**

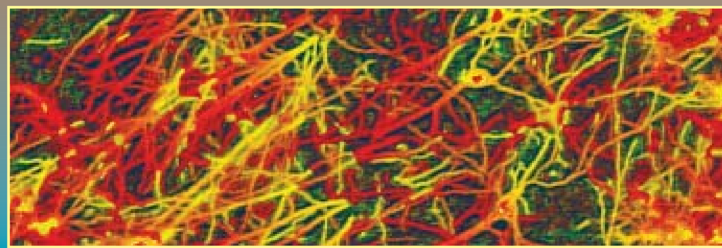
Robert Doerr can be reached via email at rdoerr@bizserve.com.

NOTE

For supplemental reading on Interactive-C and the Handy Board, there are two books I would suggest. The first is *Robotic Explorations A Hands-On Introduction to Engineering* by Fred G. Martin (ISBN 0-13-089568-7) and the second *MOBILE ROBOTS Inspiration to Implementation* by Joseph L. Jones and Anita M. Flynn (ISBN 1-56881-011-3). These are both excellent books. They will cover some of the programming techniques used on the Handy Board and also provide tons of other useful robotics information.


AUVSI
AUVSI's Unmanned Systems North America 2008
Conference and Exhibition
June 10 - 12 — San Diego, CA USA
When the World of UNMANNED SYSTEMS Comes Together
Where will you be?
http://symposium.auvsi.org
The world's largest unmanned systems show will host more than 4,500 international attendees and 300 exhibitors. The most comprehensive unmanned systems event of 2008.
For more information call +1 703 845 9671 or email info@auvsi.org
AUVSI's
UNMANNED
SYSTEMS
NORTH AMERICA
2008

DIFFERENT BITS



by Heather Dewey-Hagborg

ARTIFICIAL LIFE:

PART 2 – GENETIC ALGORITHMS: HELLO WORLD

Last time, we talked about the theory behind genetic algorithms and looked at an abstract example of how they work. In this month's article, we will implement a simple example on the PIC to understand the details and difficulties of doing this kind of programming on such a limited processor.

Like many of the topics we have covered in this column, the circuits are simple but the code is fairly advanced, so you should tackle this project with a good amount of microcontroller programming (and more specifically troubleshooting) under your belt. This article will build upon the ideas and the basic algorithm discussed in March, so if genetic algorithms are new territory for you, it would be wise to read that article before jumping into this project.

I recently decided to test drive Microchip's C18 C language compiler for 18 series PICs. The software is free to try for 60 days and is available on the development tools section of their website. I determined that this project would be a good assessment of the environment, so all of the code you will see here is written for the C18 compiler. I will say, it is not the most user-friendly coding environment to work in, and it is not intuitively documented at all, but with a little leg work it is a nice transition for the intermediate coder looking to move beyond assembly. The commands and libraries directly stem from the processor architecture making it a very transparent coding experience if you are already well acquainted with the PIC framework.

To refresh your memory, genetic algorithms abstract the natural process of evolution to solve difficult computational problems by evolving populations of possible solutions. They simulate aspects of the biology behind evolution, and are interesting as implementations of life as it could be (in the words of Christopher Langton). In this article, we will focus on the computational aspects and pragmatics of implementation on a microcontroller. Next time, we will delve into what I consider to be the beauty of the algorithm by looking at how it can simulate a biological system, and we will also explore interactive applications of the technique.

SIDE NOTE

Microchip C18 compiler is freely available to try from their website. Look for the "Student Edition" download: **www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en010014&part=SW006011**.

Search Space

If you look at any computational problem, the collection of all the possible answers to that problem comprise what is known as the search space. For example, the classic traveling salesman needs to visit a set of cities around the country and doesn't want to visit any city twice. The list of *every possible* sequence of cities to visit that meet the specifications would comprise the search space. In another example, an artificial chess player would describe the list of all possible moves they can make during a game as a search space. The term is important because the larger the search space is, the more difficult it becomes to use brute force, or classic heuristic techniques to find an answer.

To give you an idea, imagine the traveling salesman has to visit 30 cities. The number of possible routes he could take is equal to 30! or $2.65 * 10^{32}$. Add one more city to the mix and the number of possible solutions increases exponentially to $8.22 * 10^{33}$ in what is called a combinatorial explosion. Meanwhile, the chess player is even

SIDE NOTE

The example which originally inspired mine was written by Daniel Shiffman for the Processing computer programming environment. He wrote a program to evolve the classic Shakespearean phrase, "To be or not to be." You can find his code and an explanation of it at **www.shiffman.net/teaching/nature/ga**.

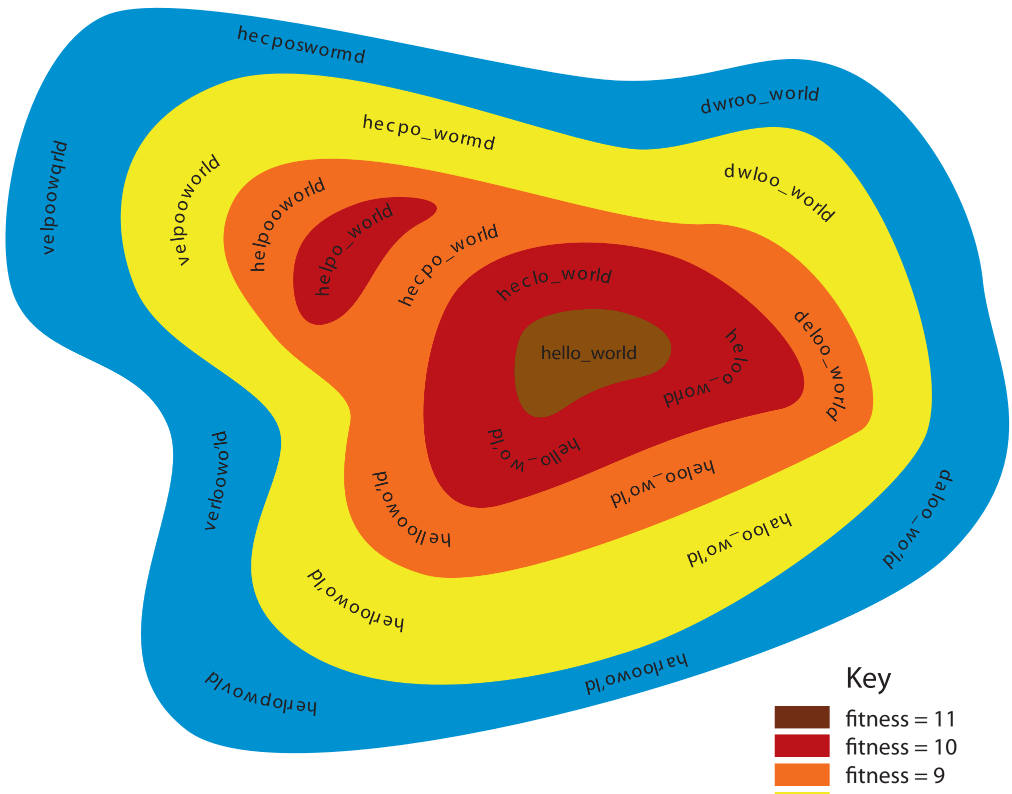


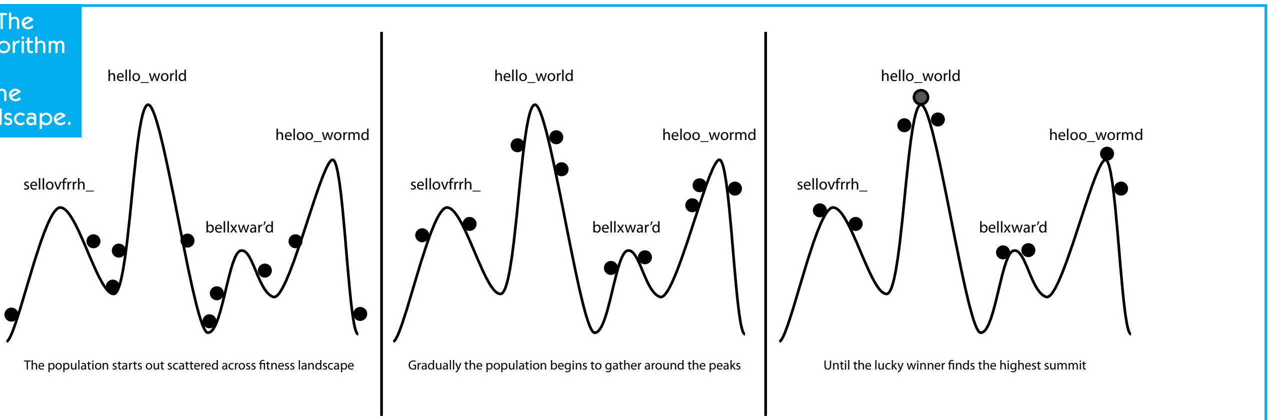
FIGURE 1. A section of the fitness landscape for the hello world search space.

worse off. They need to sift through 10^{120} possible moves, or approximately the number of molecules in the entire universe. The example we will implement on the microcontroller is evolving a familiar text of 11 characters long: "hello_world." This problem is similar to the ones previously described in that it has a large search space, though admittedly smaller than the chess game. The hello world search space contains 11^{27} or 13,109,994,191,499, 930,367,061,460,371 possible solutions — to be exact. This query would take quite a while to solve using brute force search techniques, especially on a 20 MHz processor. But with a genetic algorithm, it takes only about 150 iterations and completes in 30 seconds.

Fitness Landscape

Genetic algorithms are often described as traversing a fitness landscape and I want to give you a clear explanation of what this often-confusing term means before we jump into the code. It will help to visualize exactly what the algorithm is doing and how it traverses the search space. If we imagine all the possible solutions in a search space — for example, every possible combination of 11 characters spread out on the ground — some of these solutions are going to be *more fit* than others; "hellobgorld" is a lot closer to our goal of "hello_world" than "jubopttttreh" is. Now imagine that all of these possible solutions are connected to each other by how similar they are, and have a certain elevation based on their fitness level. This picture is the basic idea behind a fitness landscape (see Figure 1). It is a series of hills and valleys of fitness woven together by a fabric of similarity.

FIGURE 2. The genetic algorithm population navigates the fitness landscape.



BILL OF MATERIALS

- PIC18F4539
- 20 MHz ceramic resonator or crystal + capacitors
- 51K pull-up resistor
- 2 - .1 µF decoupling capacitors
- Push button (for reset)
- Serial LCD (available online from SparkFun: **www.sparkfun.com/commerce/product_info.php?products_id=461**)
- 2N2222 transistor (for inverting serial output)
- 10K resistor
- 1K resistor

Genetic algorithms create populations that begin as randomly scattered across the fitness landscape. Gradually, generations of agents jump around the hills and valleys until they begin to converge around one or more peaks. From here, the mutation operator becomes very important as a fine tweaking parameter, gradually moving one lucky solution to the top of the highest crest in sight (see Figure 2).

Hello World

The Circuit

The circuit is nice and simple. Hook the PIC up as usual, connecting a capacitor across each VDD/VSS pair; the 51K resistor from MCLR to VDD; and the reset button from MCLR to VSS. Connect your clock source and standard power connections. If you want to read the output of the PIC on a computer as well as the LCD, connect the transistor and remaining resistors as an inverter. I recommend doing this to start out with because you will probably want to add more print statements to the code for debugging and it is hard to read all the activity on a little LCD with no history display. Finally, connect the serial LCD to VDD, to VSS, and run a line between the PIC TX pin and the LCD RX pin. See the schematic in Figure 3 for details.

The Code

The program is written in Microchip C18 and is available from the *SERVO* website (**www.servomagazine.com**). I am going to walk through each section of the code by reference to the line numbers because it is too long to reprint here in full. If you open the .c file with any standard compiler, you should be able to scroll through and follow along with count.

- *Lines 1-17*

We start off with basic initialization routines. We include standard C libraries and the USART library for our serial output. We define some constants and global variables to hold our program data. The key variables here are pop_size, the population arrays, the mutation rate, and the metric array; "pop_size" is a constant which declares how many members the population will consist of. Due to PIC memory and array storage constraints, I limited the population to 10. The algorithm will run faster with a larger population and a size of 100 is standard in desktop computing, but an external

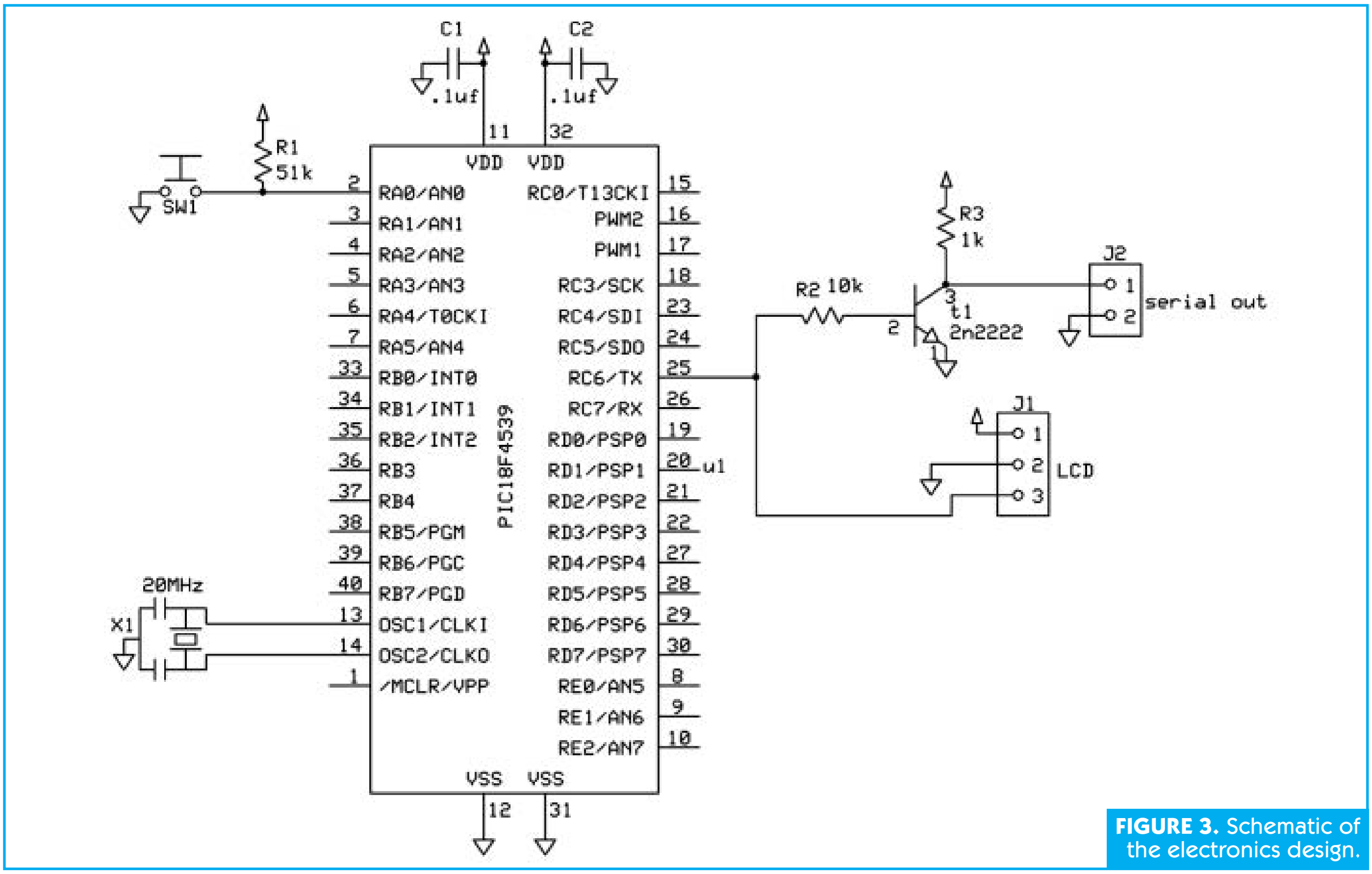


FIGURE 3. Schematic of the electronics design.

RAM chip would be necessary to accommodate a much larger array on a PIC. "pop" and "pop_new" are two-dimensional arrays that hold the DNA strands for each member of the population. We need two of them because we are going to copy, slice, and mutate the old generation stored in "pop" to generate the new generation stored in "pop_new."

"mutation_rate" is a variable which determines what percentage of bits will be randomly mutated during the breeding cycle. I made this a variable rather than a constant definition because it is often advantageous to adjust the mutation rate during evolution. For example, you can start out with a high mutation rate at the beginning and pare it down to a low mutation rate for final tweaking at the end. Finally, the metric array contains the ASCII coded letters of our solution, "hello_world."

• <u>*Lines 19-47*</u>

There are three peripheral functions in the program: delay(), quit(), and evaluate(index). The delay function creates a 1,000 cycle pause which is used in conjunction with the serial commands to avoid overflowing the LCD serial data receiving buffer. The quit command simply moves the program execution into an infinite loop, freezing the last message on the LCD. The evaluate function takes an index as an argument and evaluates how closely that index in the "pop" array matches against the "metric" array; in other words, how many letters in the DNA are correct.

• <u>*Lines 49-64*</u>

The main function implements the genetic algorithm framework we looked at last time:

1) Create an initial population of candidate solutions.
2) Evaluate the initial population.
3) While no member of the population meets the criteria for success (Breeding loop):
   a. Select individuals into a mating pool.
   b. Create a new population using crossover and mutation.
   c. Evaluate the new population.

The first step is initializing our variables, configuring the USART, and clearing the LCD screen.

• <u>*Lines 66-78*</u>

The second section is the first step of the genetic algorithm, creating a random initial population. Now, I should be clear here in saying that the microcontroller does not generate *truly* random numbers; rather it starts with a number as a seed and uses that number as the basis of mathematical calculations which generate a pseudo-random sequence of integers. If the seed is the same, the random integers will be the same and the genetic algorithm program will unfold exactly as before. I wanted to see a lot of different variations of the algorithmic solution, so I chose to seed the random number generator with the current value of timer1 which is just a quick way to get a bunch of different starting numbers for testing.

To create the DNA for the initial population, a random number between 95 and 122 was chosen by scaling and offsetting the output of the 15-bit random number generator. This represents ASCII characters "_," "`," and the lower case alphabet.

• <u>*Lines 81-98*</u>

The next step is to evaluate our initial population and check to see if any of the DNA managed to be 100% correct on the first try.

• <u>*Lines 100-136*</u>

Then we enter our breeding loop, which will terminate only when the solution is found. For a more difficult problem, you could add additional clauses here to end the breeding loop after a certain number of iterations, or when a different fitness criteria is met.

The breeding loop contains a lot of code, but is actually quite simple. First, we find the fittest member of the population and then we find the total fitness of the population. With this information, we go back through and normalize the fitness values of each member to reflect what percentage they contribute to the whole population's fitness. With normalized fitness values, we can create a weighted breeding pool which allows us to pick more fit parents more often for breeding.

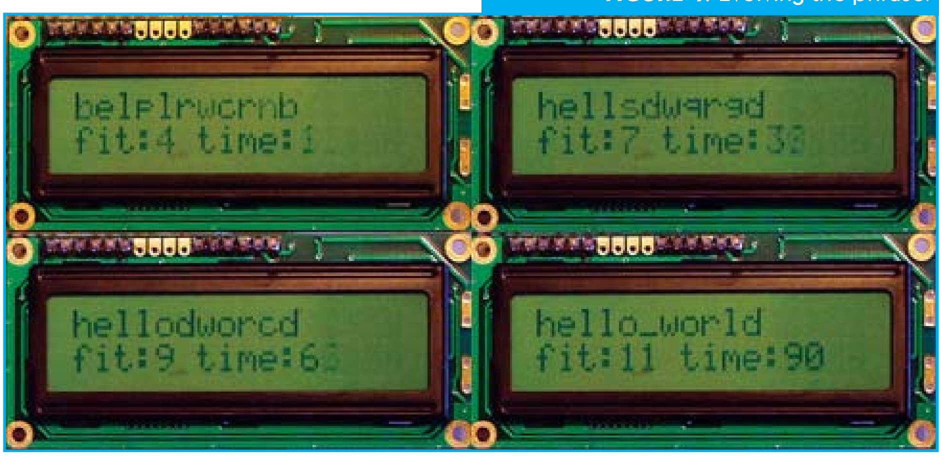
• <u>*Lines 137-199*</u>

From here, we start picking parents from the breeding pool to mate. Two parents are randomly chosen, and their DNA is broken into three pieces and interchanged to create two new children with the recombined information of their parents. The final step in breeding is to randomly mutate a percentage of bits in the DNA based on our "mutation_rate" parameter above.

• <u>*Lines 200-209*</u>

To maintain a certain level of

FIGURE 4. Evolving the phrase.



fitness in the population and to ensure we don't end up losing our genetic progress though crossover and mutation, we copy the most fit member of the original population to the new. After all these breeding steps, we can dispense with the old generation by copying the "pop_new" array into the original "pop" array.

- *Lines 210-252*

Finally, we repeat our earlier steps to evaluate the new population and we display the DNA of the most fit member of the population on our LCD screen. The last part of the loop progressively decreases the mutation rate after we hit a threshold fitness level of 5.

So that is the genetic algorithm, implemented in 5.6K of program space and 551 bytes of RAM on the PIC. There is still plenty of memory and most of the I/O pins are left to allow you to make this code one part of a larger program. Don't feel limited to evolving the phrase "hello_world;" get creative! For instance, I can imagine a digital clock display that is evolving the word representations of the time, or a robot which evolves different greetings for different people and objects it encounters. Or, how about a display without a specific goal set ahead of time that tries to evolve new words with a fitness factor based on how closely they resemble existing words?

One way you could implement this would be to make it interactive and utilize a human judge of the fitness of each word. This is called *interactive evolution*, and we will get into the details and try out a practical example of this technique next time.

Although the hello world problem is a good illustration of genetic algorithm techniques, it is a bit of a toy problem because we know the answer in advance. Typically, the reason for employing a GA in the first place is because the answer is difficult to estimate, and this makes determining a fitness measure a bit trickier. For example, in the traveling salesman problem we don't know how short the shortest route between all the cities should be, so we can't be positive that we have found the best solution. What genetic algorithms give us is the humbling quality of mother nature who demands only that her children are good enough. We don't have to be perfect, we just have to survive. Similarly, with the traveling salesman problem fitness would be measured as a function of how long the route was. The algorithm can be run and the settings can be tweaked until a satisfactory solution is found, a path with a reasonable length, or it can be run until it plateaus and the results don't seem to be getting any better.

Happy coding! **SV**

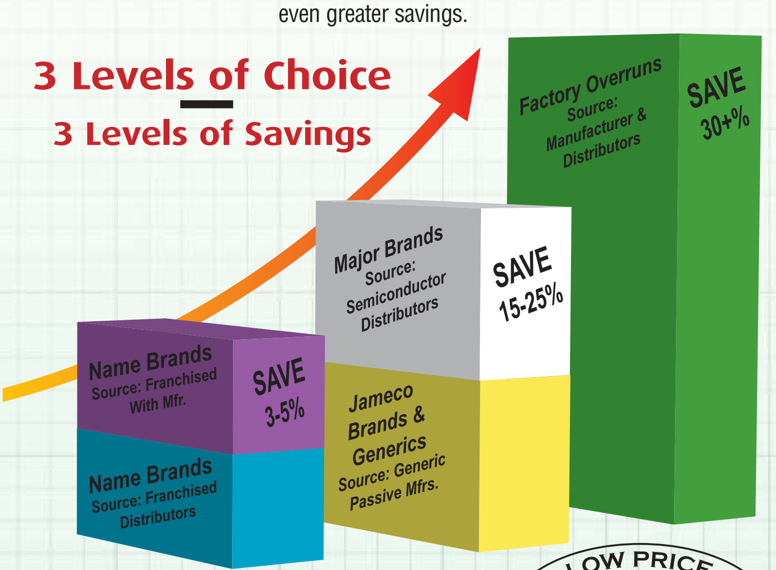
Exclusively at Jameco...
The Lowest Prices...
Guaranteed!

Save 3% to 5%
Looking to save on thousands of Name Brand components? Jameco offers these popular products for 3% to 5% below market price, and we guarantee it!

Save 15%-25%
Save more when a specific manufacturer is not required. Major Brand semi's are sourced from 5-6 specific major manufacturers, plus we also offer Jameco Brand and Generic passives for even greater savings.

Save 30% or more!
Looking for killer deals? Jameco buyers often find Factory Overruns from some of the industry's biggest names. That means savings of 30% and more to you!

3 Levels of Choice
3 Levels of Savings



Other Jameco Advantages:
- We offer over 300 Major Manufacturers
- ***99%*** of catalog products ship the same day.
- Lowest prices guaranteed, or we pay 10%.

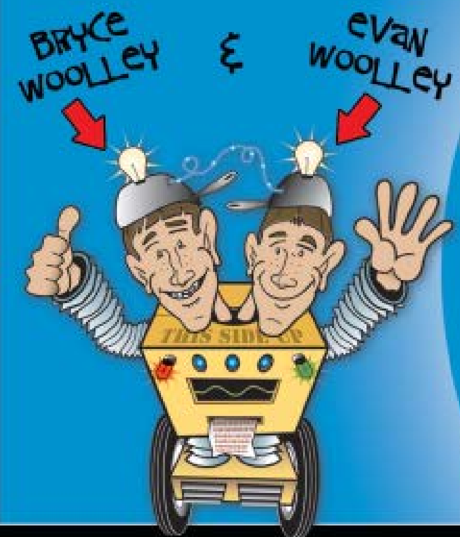
LOW PRICE GUARANTEE
If you find a lower print-advertised price for a product we offer, just let us know. We'll offer you that product for 10% less than the price you found, right down to our cost!

Order 24 hours a day, 7 days a week
www.Jameco.com
Or call 800-831-4242 anytime

JAMECO® ELECTRONICS

SEMICONDUCTORS • PASSIVES • INTERCONNECTS • ELECTROMECHANICAL • POWER

Twin Tweaks



THIS MONTH:

Geekings From France

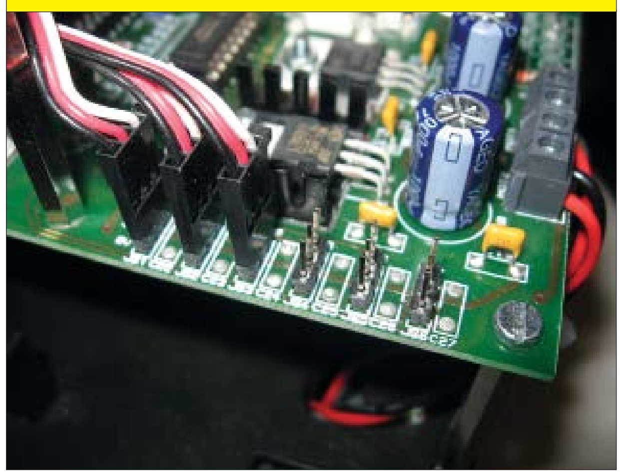
This month, we have the privilege of introducing an intrepid robotic envoy all the way from France, courtesy of POB Technology. We were lucky enough to receive the POB Golden Kit, which includes a fully assembled robot, a software CD with an electronic manual, and a serial cable for programming. The robot is equipped with tank treads, a camera, and a claw for manipulating objects. The easily accessible circuit board boasts plenty of open terminals for the addition of sensors and mechanisms, and the multicultural bot is truly an accomplished linguist that is fluent in a wide variety of languages for programming.

Cameras, hacker ports, and a creative software platform make the POB robot sound like an ambitious project, and it most certainly is. The POB bot is meant to appeal to a broad audience of tinkerers including everyone from university researchers to hobbyists. The POB bot even attempts to target that most elusive and sought after of demographics — kids. While many kits may find their appeal to the younger crowd hindered by too much complexity, too much simplicity, or any other unfavorable medium in between, we think the biggest obstacle to getting kids interested in robotics (and as an extension, science and engineering in general), is the stiff competition presented by other, more favored leisure time activities like video games and television. The POB robot has a few tricks up its sleeve to surmount that very obstacle.

Here's Looking at You, Kid

The POB bot presents immediate entertainment by offering some preprogrammed behaviors. As soon as the bot is turned on, the LCD screen comes to life and users are able to choose from one of three options: Look, Follow, and Test. A joystick and push button allow the user to make selections on the screen, and we think such a setup will appeal to the elusive younger demographic that has gained a degree of technical literacy by playing countless video games, but often go without a more productive pursuit to fill up their free time. The Look option will quite literally let the user see the world through the eyes of the POB robot, and while that world is pixilated and monochrome, it is also undeniably cool. Half of the LCD screen shows the output from the camera and the other instructs the user to press Enter to return to the menu. One might be hesitant to do that, however, because there is something enigmatically fascinating about seeing the world distilled down to a few pixels, and

POB PWM Ports.



something even more entertaining about waving your hand in front of the camera and seeing part of yourself become a piece of that mysterious low resolution world.

The Follow option is much like the Look command, but instead of languidly observing its surroundings, the POB robot will follow moving objects. The prowling style of the robot often seems to be cautious, slow, and a bit choppy, but in well lit rooms it becomes a bit more fluid. It is truly amazing how the POB bot can so quickly exude that sense of mysterious machine intelligence, even before its true talents can be uncovered.

The last option is entitled "Your Test," and it is a very useful and easily accessible troubleshooting program that also doubles as an easy way to test any mechanical additions to the bot without having to go through all the programming rigmarole. Once this option is selected, the screen will show six plus and minus signs and two minimalist representations of the motors for the drive train. Pressing the plus and minus signs will order the elected motor to move forward or backwards, with the motion lasting for as long as you have the button pressed. The first three sets of buttons are already dedicated to the existing mechanisms on the bot — the pan and tilt of the camera and the opening and closing of the beetle-like mandibles. The three other sets of buttons correspond to the open PWM ports on the main board of the robot, which are perfect for expanding the mechanical functionality of the bot. We happened to have some extra Futaba servos lying around, which were even of the same brand and model as the servos already on the robot. After simply connecting the PWM cables, the extra servos respond with just the touch of a button.

Many kits present themselves as an effective way to do rapid prototyping, but very often the speed of your prototyping is limited by the swiftness of your programming. Depending on the hospitality of the interface or the experience of the programmer, this can be a discouraging step. With the test feature on the POB robot, you can truly get some instant vindication for simple design ideas, and we think this feature will just encourage more creativity.

Ultimate RISBEE

The POB robot came with a software CD, and in it is so much more than the average quirky GUI or C programmer. (It comes with both, actually.) As we alluded to earlier, the POB robot is a multilingual creation. The software CD comes with a variety of programming interfaces, and everything is also available on the website. Lots of cool stuff that will allow users to program their robots in C, Java, or Basic, or even to program their bots using Linux or a Mac operating system are available, as well. All of the compilers and any other software needed to write download programs using any of these languages is all online, and generous documentation is sure to answer any questions one might have, whether you're a novice just starting to get your feet wet or an expert becoming oriented to a new system.

Lots of robot kits have ample documentation and sample programs that seek to shepherd newbies through the ins and outs of C programming and even Basic, but to someone whose fanciest computer trick is a firm command of Microsoft Word even simple sample programs can be daunting, and disembodied commands can seem like gibberish. Many robot kits aimed specifically at beginners — particularly the younger crowd of budding roboticists — try to demystify the challenge of programming by providing graphical, object oriented programming interfaces. Classic examples include the older, pre-LabView LEGO Mindstorms programming and Easy-C for the Vex Robotics design system. POB Technology has come up with its own entry into the pantheon of introductory programming interfaces, and it goes by the endearing name of RISBEE.

RISBEE stands for Robotics Initiation Software for Budding Engineers and for Education, and the animated building blocks (reminiscent of early LEGO Mindstorms programming) certainly look capable of enticing and educating novice roboticists. The RISBEE interface presents the user with a grid on which to construct programs, with the different available

POB RISBEE.

commands divided up into three tabs. The General Orders tab includes commands to begin and end the program, begin and end subroutines, and to define and compare variables. The Basic Orders tab includes commands to move the robot forward, backward, left and right; wait commands to prolong the movements; commands to manipulate servos; and other commands to set digital input values or to save mysterious multimeter values. The Special Orders tab includes commands that deal with the POB eye; like ones to center the robot in front of a pattern. Commands chosen from these tabs are simply dragged and dropped onto the grid.

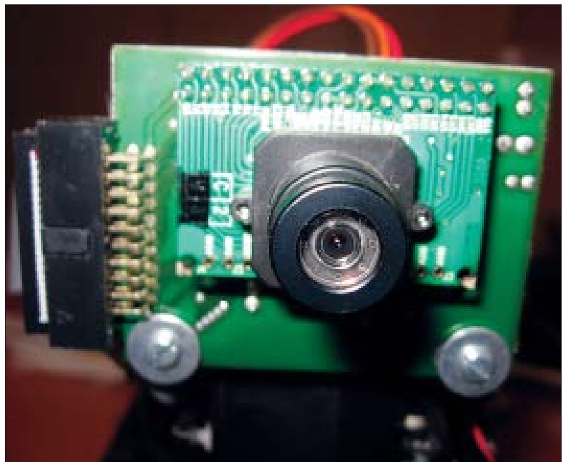
Programs are assembled from left to right, in something that generally resembles ladder logic. RISBEE dodges some of the disadvantages of ladder logic by offering special commands to form and execute subroutines, and subroutines can even be repeated multiple times. Programs in RISBEE can also include sensor input and conditional activities, like having the robot execute a certain activity when it sees a certain pattern with the POB eye, or even execute a command for the entire time that it sees the pattern. While some real deal programming terms like "goto" and "subroutine" populate RISBEE, even with the absence of a glut of official terminology, beginning programmers are effectively introduced to a lot of essential programming concepts that will serve them well in their later programming endeavors. We are confident that the folks at POB Technology have created a winning interface in RISBEE that strikes the balance between being informative and being fun. With the interactive LCD screen and joystick — and even the RISBEE programming itself that can feel more like a game of Battleship with its animated icons and innovative structure — the POB robot is sure to appeal to novice roboticists.

The Great Pink Litten

With all of that attention being paid to RISBEE and to the beginners targeted by it, some more experienced folks might be wondering what the POB robot has in store for them. The answer is — as you might have hoped — plenty. Once again, POB Technology has an excellent website that is choc full of resources that are sure to answer every need for hobbyists and serious roboticists. In addition to the RISBEE software that can be downloaded from the site, users can get their hands on a number of other programming tools, and also documentation on the modification and physical construction of the company's impressive bots. The other programming interfaces deserve a mention because they are a step up in complexity from RISBEE, but still good for those who want to do crazy things with the bot but are reluctant to program. The POB Builder, for example, is another programming interface for use with the POB bot. While not quite as graphical as RISBEE, it still offers easy to use commands assembled in a ladder logic form. It may not offer the level of sophistication some users might demand, however, the POB Builder is still a useful program for rapid prototyping.

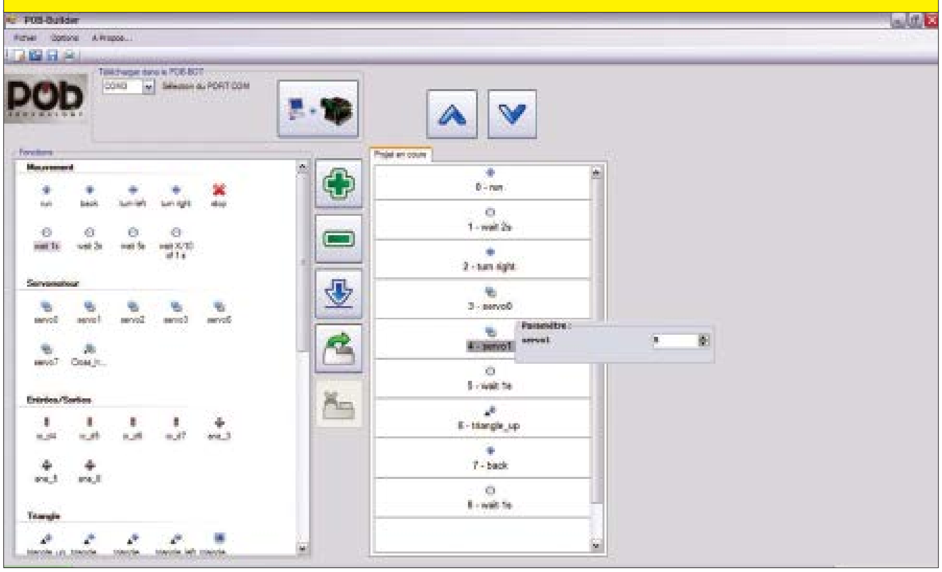
The POB Tools program offers an easy way for users to add their own patterns to the POB Eye's repertoire of recognized shapes. Users can import their own bitmap files for the robot to learn, so these patterns could ostensibly come from programs like Paint or from scanned pictures saved in the right format. The documentation on the POB Eye provides ample pointers on what the Eye can and cannot recognize, so users should be able to have fun with it in no time.

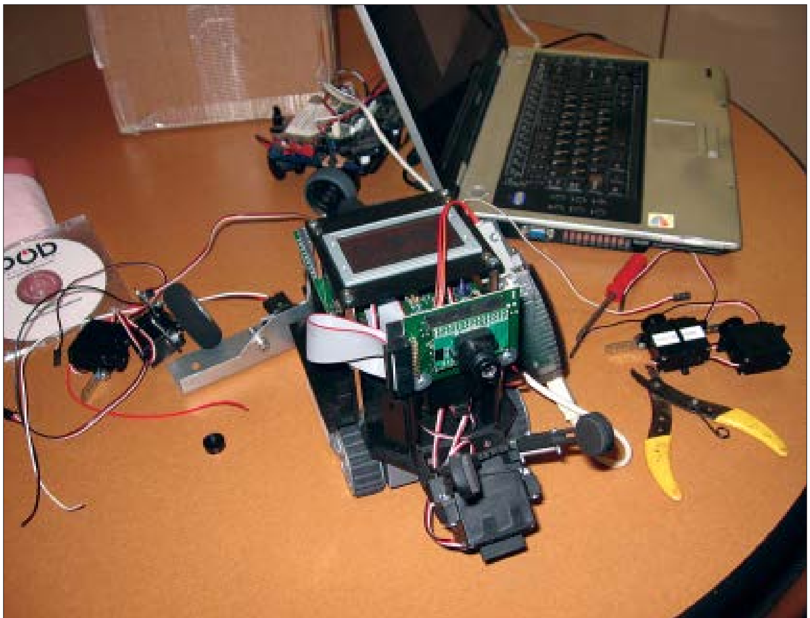
And, of course, the programming experts always have the straight up, no nonsense, no graphics C programming interface to use if they prefer something a bit more sophisticated. Once again, though, ample documentation and sample programs make the C interface more than just a playground for the experts,



POB EYE ("EYE" SEE YOU!).

POB BUILDER.





POB WORK.

POB TOOLS.

but also a realistic goal for the beginners that start out with RISBEE.

No Subtitles Required

With RISBEE showing some similarity to other graphical programs (like LEGO Mindstorms programming), we were curious about what the philosophy was behind developing this new system. Such an ambitious robot, in general, was also bound to have an interesting design philosophy behind it, and to learn more about RISBEE and the POB mind set, we contacted Philippe Kervizic at POB Technology.

Woolley Brothers: What audience is the POB bot meant to appeal to? Is it intended for hobbyists, serious roboticists, or both?

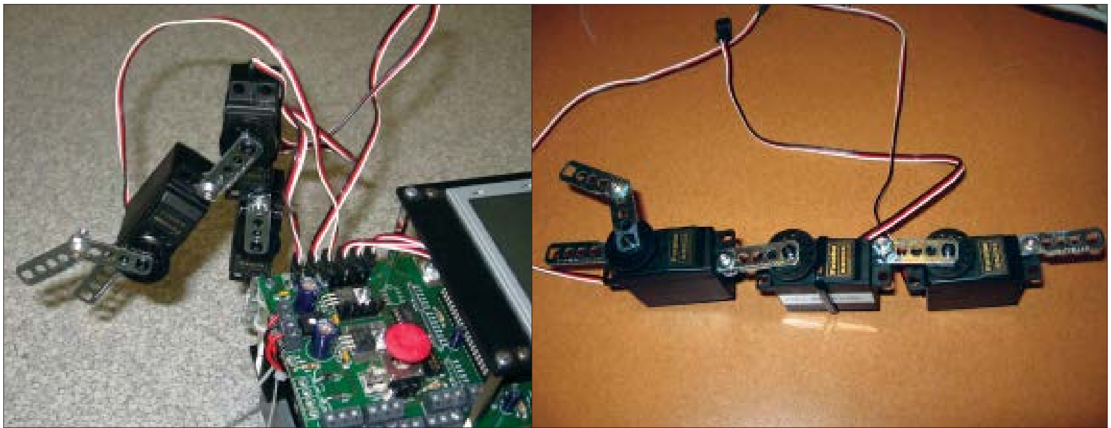
Philippe Kervizic: This is a good question. Most of our users are educational, for teaching robotics and automation from junior high schools (introducing the concepts) to higher education (improvement of the engineer sciences). Thanks to the design of the robot and software, the skill level of the users are from beginners to serious users. Lost of hobbyists use the robot, even quite young hobbyists. Thanks to the calculating power of the POB Eye and the I/O possibilities, serious roboticists also use the robot or parts, programming in C. For example, a researcher runs a neural network on the robot to study the action selection for the robot behavior, like animals do. I am myself another example. I met POB 18 months ago while I bought a POB bot to build a robot which turns the pages of a book. I am computer science oriented and I have very few electronics or mechanical knowledge. I looked for existing electronics to program. Then, I built a robot with parts of printers and scanners, managed by the POB Bot parts. (Two months later, I joined POB to manage the sales activities.)

POB SCREEN FIREPLACE.

POB SCREEN HAND.

WB: What is the philosophy behind the design of the POB robot? Is it meant to be educational, accessible, expandable, or all of the above?

PK: The philosophy behind the robot is to give tools to users, based on an open robotic architecture, to build their own projects (robotic project, educational project, etc.). As robotics is computing plus electronics plus mechanics, and only few users know all of these fields, we wanted to give tools to allow the user to focus on what he likes. Then, the robot is designed to be able to forget one or two of these fields. Of course, computing basis are needed. This way, we do our best to help users, giving lots of downloadable examples and providing personalized support. Open architecture also means an expandable and customizable robot. Lots of robot makers want to reuse some parts they already have (arm, I^2C stuff, etc.). With POB, they can. This open approach is one of the keys of our success in education, because educators have a lot of possibilities to focus on what they want.



POB Arm.

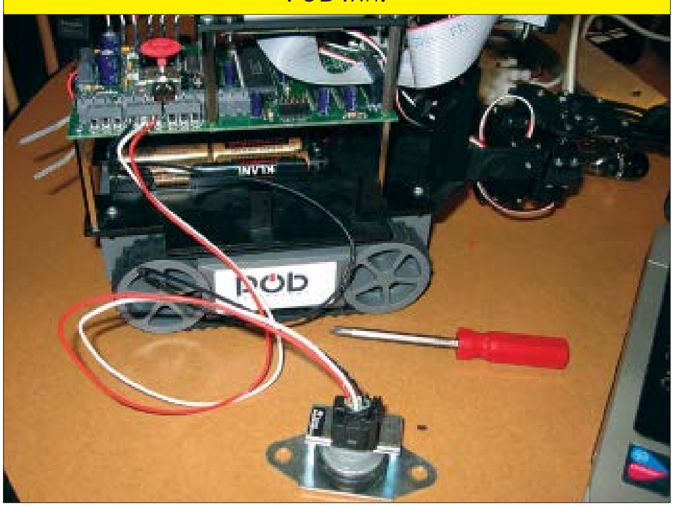
WB: Could you tell us more about the development of RISBEE and its goals?

PK: RISBEE follows the same philosophy: 1) To make the robot accessible for people who don't know programming. Historically speaking, POB users were mainly from higher education and a little from secondary schools. With RISBEE, we target junior high schools and secondary schools. We had the surprise to see that highly skilled users also sometimes use RISBEE for quick sensor tests or quick idea validation. 2) Keeping the "open" philosophy, RISBEE is designed to work with different robots from the market. We are negotiating with well known robot makers to adapt RISBEE to their robots. The goal is to have only one development environment to know for several robots. But it is a bit early to tell more about it.

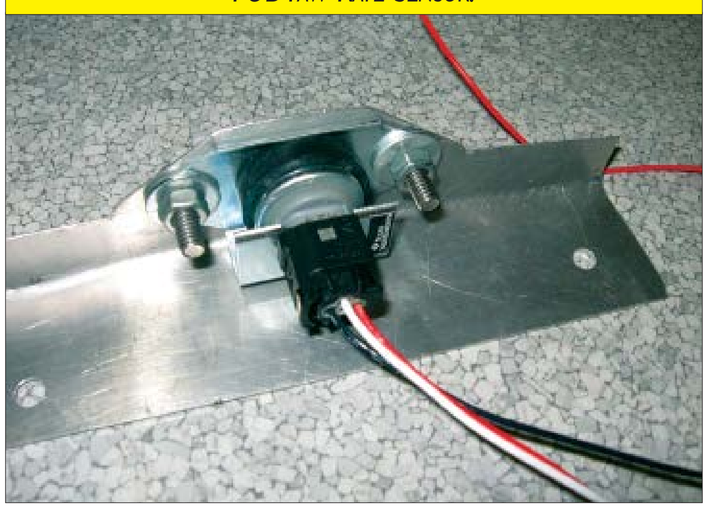
Good Sharky, Colonel God!

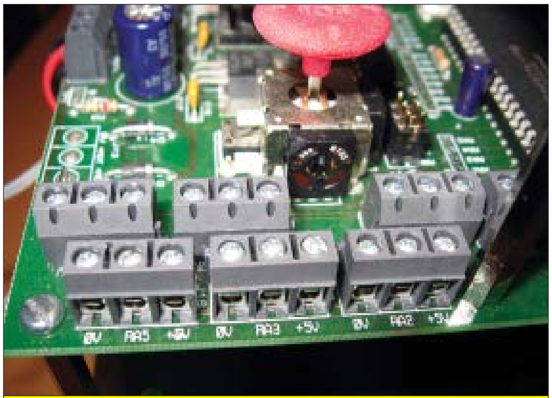
POB Technology terms the hobbyist use of their products as "innovative leisure," and we certainly agree with such a moniker. The POB robot is a great educational tool, but its accessibility and expandability also make it a great way to spend a casual afternoon tinkering. We are more of the mechanical predilection, so we wanted to see how easy it would be to add on a mechanical addition to the POB robot. As we alluded to earlier, we had a few extra servos to mess with. Even though the POB bot was already equipped with a capable gripper in the front, we thought of adding another robotic arm to the back (or perhaps it was more of a scorpion tail, depending on how you look at it). After using the test function to see that all of our servos (modified for full rotation) were fully functional, we were ready to go. Our additional arm was assembled in a bit of slapdash fashion, and we're sure that many folks could come up with something much more elegant than the zip ties we used to attach the arm

POB Yaw.



POB Yaw Rate Sensor.





POB Terminals.

POB Board.

what we've come across in other projects.

Why did we put on the yaw rate sensor, anyway? Well, it was convenient, but we're sure that there would be plenty of applications for a yaw rate sensor on an intrepid robot explorer. The yaw rate sensor measures the rate of change of the yaw angle, which is the angle of rotation about the vertical axis. This has obvious applications for navigation, but usually it's seen in aeronautics and not on land-bound French robots. That is not to say it wouldn't be useful — it could also potentially be used for things like feedback on the physical movements of the bot to ensure physical stability if you're, say, carrying a cumbersome load around some corners. But hey, we're just spitballing here.

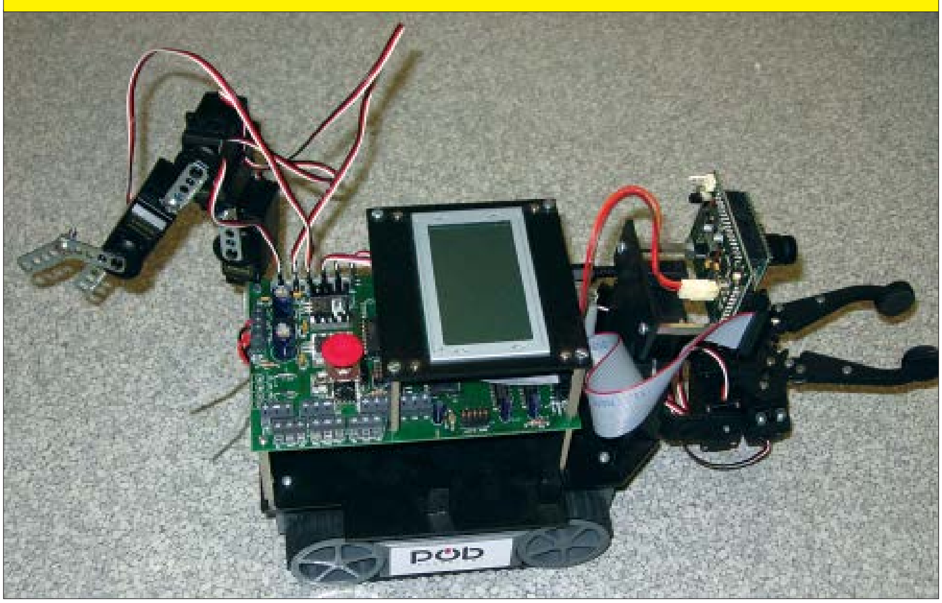
to the back of the robot. In our endeavor to attach the arm is actually where we stumbled upon our first criticism of the POB bot — the extra PWM ports may seem to invite the addition of extra servos, but the board itself does not provide ample places to physically attach them.

A little ingenuity is easily able to overcome this problem, but it just stuck out to us just because these good folks have made every other thing about the robot so easy. Even integrating the mechanical additions into the programs was easy, especially in RISBEE and POB Tools — all you have to do is choose the appropriate port number from a dropdown list and you're in business.

We also got our hands on a yaw rate sensor, which we used mainly because it had the unsocketed wires that would be easiest to incorporate into the free terminals on the bot's board. The POB Board includes both digital and analog sensor inputs for your hacking pleasure, and both are conveniently labeled in the documentation.

All it took to hook up the yaw rate sensor was to place the power, ground, and signal wires in the appropriate slots and to tighten the screws. Commands in C to read digital and analog input are provided in the documentation, but even the simple RISBEE program had commands capable of reading sensor input. Much like for the motors, the appropriate sensor would simply need to be chosen from a dropdown list and you were ready to go — a much easier process for integrating sensors than

POB Modified.

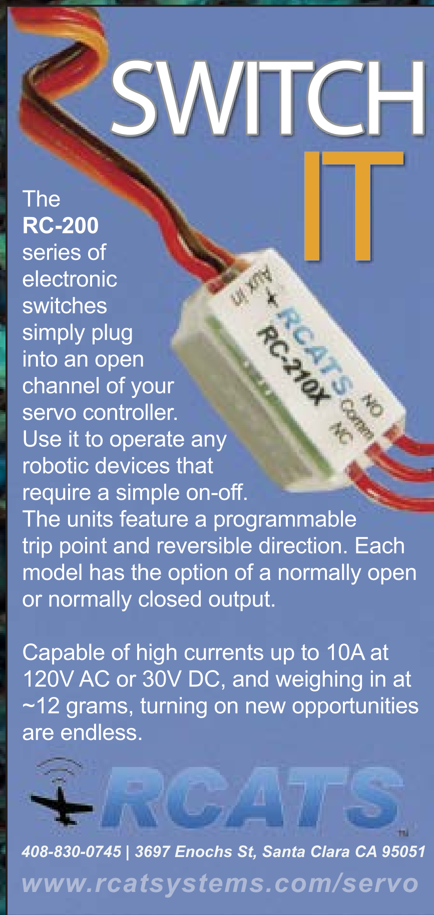


Open Source, Open Mind

Overall, we think the POB bot is one of the most ambitious robotics kits that we have ever received, but instead of becoming a jack of all trades and master of none, the POB robot is a comprehensive robotics development platform that can appeal to a wide demographic of tinkerers. Its open source philosophy is perfectly executed, and the robot provides a versatile platform for programmers, electronics whizzes, mechanical gurus, or any combination thereof. And not only does the POB robot make itself available to tinkerers of all skill levels and interests, but it also provides a great way for everyone from hobbyists to professionals to broaden their technical education. Whether it be learning C, trying out Linux for programming a robot, adding your own custom sensor modules, or just playing around, the POB robot literally has something for everyone. **SV**

For more information, go to: ***www.pob-technology.com****. We would like to extend a special thanks to Philippe Kervizic for all his help.*

www.RobotShop.ca is upgrading...
Uploaded
-New Microsoft.Net technology
-New optimized product search engine
-New user account: historic, order status,etc...
-New automatic quantity discount
-New real-time PDF price list generator
-New quick shipping cost evaluator
-New real-time stock status
Uploading...
-New one page checkout
-New autonotify B/O feature
-New B/O shipping options
-New order modification capability
-...



Robotics Showcase

Give your Robot the power it deserves...

Introducing the

proSeed

microcontroller

only $89

- 16 general purpose I/O's
- 16 analog inputs
- HUGE 128KB RAM
- 128KB EEPROM
- Super-Fast 70MHz SX CPU
- 24 pin compatible
- Programming languages
  16-bit ANS FORTH.
  C and Assembly available by third party.
- Visit our website for more...

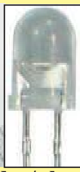
LORAX WORKS

www.loraxworks.com

The **Escape Robot**'s built-in microprocessor enables it to "think" on its own. (KSR4) **$29.95**

Soldering required

The robot frog moves forward when it detects sound and repeats: start (move forward) -> stop -> left turn -> stop -> right turn -> stop. (KSR2) **$19.95**



5mm White water clear **LED** 3.5V 10,000 mcd (AB287) **$0.56**

20 second voice recorder/playback module. The electret microphone is on the board. One button records, the other button is momentarily pressed to replay the message. (pre-assembled) (A96010) **$6.60**

The Velleman **Personal Scope** is not a graphical multimeter but a complete portable oscilloscope at the size and cost of a good multimeter. (HPS10) **$146.**

Thousands more items on our web site!

Electronics123.com
Tel: 1-888-549-3749
102 E. Park Ave, Columbiana OH 44408

Robotic Football!
www.roboticfootballleague.com

RFL

Throw! Catch! TACKLE! Hack!

The AI-01 Robot

- Throws and catches ball
- Runs plays
- QB and lineman/receiver arm attachments included
- Video game style controller
- HACKABLE! Programming and serial port gives access to PIC flash microcontroller

active innovations

for more information
www.active-innovations.com

Robotics Showcase

ALL ELECTRONICS CORPORATION

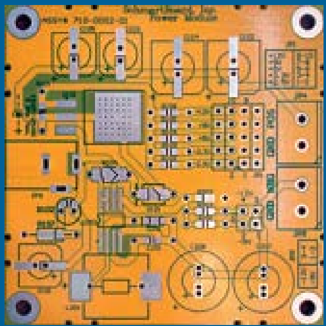
THOUSANDS OF ELECTRONIC PARTS AND SUPPLIES

VISIT OUR ONLINE STORE AT
www.allelectronics.com

WALL TRANSFORMERS, ALARMS, FUSES, CABLE TIES, RELAYS, OPTO ELECTRONICS, KNOBS, VIDEO ACCESSORIES, SIRENS, SOLDER ACCESSORIES, MOTORS, DIODES, HEAT SINKS, CAPACITORS, CHOKES, TOOLS, FASTENERS, TERMINAL STRIPS, CRIMP CONNECTORS, L.E.D.S., DISPLAYS, FANS, BREAD-BOARDS, RESISTORS, SOLAR CELLS, BUZZERS, BATTERIES, MAGNETS, CAMERAS, DC-DC CONVERTERS, HEADPHONES, LAMPS, PANEL METERS, SWITCHES, SPEAKERS, PELTIER DEVICES, and much more....

ORDER TOLL FREE
1-800-826-5432
Ask for our FREE 96 page catalog

The Power SchmartModule™ will power up your circuits with your choice of -9, -12, +2.5, +3.3, +5, +9, and +12 volts.



$15

Very Schmart!

www.schmartboard.com

> The First Full Silicon
> Neural Network-based
> Pattern Recognition System
> RNR RoboSight Lite™

I^2C and Lego® NXT Compatible

RoadNarrows

RoadNarrows Robotics
Loveland, Colorado
800.275.9568
www.roadnarrows.com

EVENTS CALENDAR

Send updates, new listings, corrections, complaints, and suggestions to: **steve@ncc.com** or FAX **972-404-0269**

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**

— R. Steven Rainwater

May

1 **UNI Mini-Sumo Robotics Challenge**
Student Union, University of Northern Iowa, Cedar Falls, IA
There are two notable features about this Mini-Sumo event. The first novelty is that it allows "ship-in" competitors. Robot builders from other states and even outside the country shipped in robots to compete in the 2007 competition. The second interesting feature is that the event is streamed live over the web, so don't forget to tune in and watch.
http://list.dprg.org/archive/2008-March/031554.html

3 **The Tech Museum of Innovation's Annual Tech Challenge**
Parkside Hall, San Jose, CA
A different robot challenge is designed each year. Check the rules on the website for the details of this year's challenge.
http://techchallenge.thetech.org

5-9 **Alcabot and Hispabot**
University of Alcala, Madrid, Spain
Both competitions are being held together this year. Events include Sumo, maze solving, line following, and robot soccer.
www.depeca.uah.es/alcabot

10 **CybAiRBot**
Poznan, Poland
Robot Sumo using standard Japanese rules.
www.sumo.put.poznan.pl

10 **DPRG RoboRama**
Fair Park, Dallas, TX
All new events this year including two contests for indoor robots that involve navigating a course delineated by orange cones. Also, four contests for outdoor robots including Out-and-Back, Borenstein Squares, an obstacle course, and the Long Haul.
www.dprg.org/competitions

17 **PDXBot**
Doubletree Hotel, Lloyd Center, Portland, OR
This year's PDXBot will include "robot competitions, exhibits, vendors, raffles, and more." Past events have had an impressive array of robot events, so be sure to attend if you can.
http://portlandrobotics.org

17-18 **Swiss Eurobot**
Yverdon-les-Bains, La Marive, Switzerland
Regional event for Eurobot competition.
www.swisseurobot.ch

19-23 **ICRA Robot Challenge**
Pasadena, CA
This contest includes three events with intriguing names: Autonomous Planetary Surface Exploration; Teleoperated Robot Emergency Mission; and Human-Robot Interaction
http://icra.wustl.ed

21-25 **Eurobot**
Heidelberg, Germany
Even though the name implies Europe, the contest is open to anyone. Previous Eurobots have had entries from as many as 350 teams from 26 countries. The 2008 event is called Mission to Mars and involves simulated Mars robots looking for signs of life.
www.eurobot.org

23 **NATCAR**
UC Davis Campus, Davis, CA
Very high speed autonomous line following.
www.ece.ucdavis.edu/natcar

30 **AUVS International Ground Robotics Competition**
Oakland University, Rochester, MI

EVENTS CALENDAR

Send updates, new listings, corrections, complaints, and suggestions to: **steve@ncc.com** or FAX **972-404-0269**

The contest continues through June 2nd. "A fully autonomous unmanned ground robotic vehicle must negotiate around an outdoor obstacle course under a prescribed time while staying within the 5 mph speed limit, and avoiding the obstacles on the track."
www.igvc.org

June

1-6 **CIG Car Racing Competition**
Hong Kong Convention and Exhibition Centre, Hong Kong
Autonomous robot cars compete against each other in this racing event. **www.wcci2008.org**

21-22 **International Autonomous Robot Contest**
Del Mar Fairgrounds, Del Mar, CA
Includes three events for autonomous robots including the Urban Challenge, in which robots must find their way through a maze; the Gold Rush Challenge, in which robots cross a desert filled with rocks and cacti; and a technical presentation, where contestants explain the robot's hardware and software.
www.iaroc.org

26-28 **MATE ROV Competition**
Scripps Institution of Oceanography, San Diego, CA
Underwater robot contest for high school and university students.
www.marinetech.org/rov_competition

28 **UK National Micromouse Competition**
Technology Innovations Center, Birmingham, United Kingdom
Micromouse robots compete for the coveted Brass Cheese award.
www.tic.ac.uk/micromouse

July

8-11 **Botball National Tournament**
Norman, OK
Educational robot contest for middle and high school students designed to use science, technology, engineering, and math to solve real world problems.
www.botball.org

13-17 **AAAI Mobile Robot Competition**
Chicago, IL
This year's competition will take the form of exhibits that demonstrate either robot creativity or mobility and manipulation. Expect to see robots that dance, paint, play musical instruments, and much more.
www.aaai.org/Conferences/conferences.php

14-18 **K'NEX K*bot World Championships**
Las Vegas, NV
Events for two-wheel drive autonomous K*bots, four-wheel drive autonomous K*bots, and the remote control Cyber K*bot Division.
www.livingjungle.com

STEER WINNING ROBOTS WITHOUT SERVOS!

Perform proportional speed, direction, and steering with only two Radio/Control channels for vehicles using two separate brush-type electric motors mounted right and left with our **mixing RDFR dual speed control**. Used in many successful competitive robots. Single joystick operation: up goes straight ahead, down is reverse. Pure right or left twirls vehicle as motors turn opposite directions. In between stick positions completely proportional. Plugs in like a servo to your Futaba, JR, Hitec, or similar radio. Compatible with gyro steering stabilization. Various volt and amp sizes available. The RDFR47E 55V 75A per motor unit pictured above.
www.vantec.com

VANTEC

Order at (888) 929-5055

ROBOTICS RESOURCES

Tune in each month for a heads-up on where to get all of your "robotics resources" for the best prices!

BY GORDON MCCOMB

Robot Builder's Bookshelf Redux

Despite fancy multimedia, Flash animations, and free Internet tutorials, the lowly book remains the most popular way of learning about a new subject. Books are portable and can be read when traveling, while waiting for your hamburger, or at home beside the fireplace.

If you own the book, you can mark it up with a highlighter, or paste Post-It Notes on your favorite pages so you can return to them quickly later. And, unlike Web pages and many other Internet resources, your book won't catch a computer virus, pop up ads on you, or blurt out annoying MIDI songs through your PC's speakers at 3 AM in the morning!

Maybe because I am an author of books — and robot books at that — I tend to gravitate toward the bound written word when I want an authoritative guide. It doesn't even have to be a new book; there are hundreds of useful and educational books on robotics that are 10+ years old, many of them no longer in print.

So, here are some books — some old, some new — you may want to consider adding to your reading list. These are available for purchase, new or used, but don't forget about your local library. Most towns have several library branches, so ask the librarian (or consult the library's card catalog) if the book you want is available at another location.

Where to Get Your Books

A great place to look for the printed word is in the online world! First, consider the *Nuts & Volts* hobbyist bookstore, provided by the publishers of *SERVO* and *Nuts & Volts*. The books have been carefully selected by the editors for their content and value. Check out this month's ad toward the back of the magazine, or go online to **www.nutsvolts.com** or **www.servomagazine.com** and click on the Store link. You can choose from a number of robot-related books and products.

Online bookstores boast large inventories and fast shipping, but unfortunately, finding a technical book can be a real chore because of the way topics are organized. However, you can usually find what you're looking for if you know the actual book title or author.

Local book stores are great if you want to browse through books, especially if you want to see if the book is right for you. Most of the large bricks-and-more bookstores have a technical section. Ask a store clerk to help you find it. Often, the robot books are mixed in with other books about electronics and science.

Of course, there's no rule you have to purchase every book you read. Local libraries are a rich source of excellent books on any subject. While the library may not have the absolutely latest title on cutting edge technology, for many aspects of robot building you don't need it. The core concepts of robotics haven't changed much in the last 20 or so years; some of the robot books from the early and mid '80s still contain valid information. (And besides, they're fun to read!)

I like to browse the book section of local used bookstores and thrift shops. It's amazing what people ell or discard! I once found an entire set of books on learning how to use metal machinist tools, the lathe, and mill. These appeared to have been part of a correspondence course and were printed in the early 1970s. I got perhaps $150 in books for under $5!

While on the subject of older books, online sellers like Amazon and eBay are a great source for books that may no longer be available in print. When getting a used book through Amazon, you're really buying it from a third party. Amazon merely acts as a go-between, collecting the money and sending orders to the seller. Many books — including specialized technical ones — are available used for less than their original selling price. It's a good way to beef up your personal library without having to go into debt.

Sources

Following are some books on amateur robotics you may wish to consider. Some are a few years old, but still contain a good deal of relevant information. Since I can't list all the robot books available, consider this list a springboard for further research. I have provided the name of the author, the publisher, the publishing date, and the ISBN. If you're interested in ordering a book online or through your bookstore, all you really need is the ISBN (and the title to verify that you've gotten the right one).

General Books on Robot Building

Something for everyone.

123 Robotics Experiments for the Evil Genius
by Myke Predko
McGraw-Hill, 2004
ISBN: 0071413588

Large format book is presented like a lab journal.

Absolute Beginner's Guide to Building Robots
by Gareth Branwyn
Que, 2003
ISBN: 0789729717

Engaging book that teaches the elementals of robot building while practicing with three "junkbots" built from stuff around the house.

Applied Robotics
by Edwin Wise
Delmar Learning, 1999
ISBN: 0790611848

Grab-bag of amateur robotics, with a strong emphasis on hardware design using an Atmel AVR microcontroller.

Bionics for the Evil Genius
by Newton Braga
McGraw-Hill/TAB Electronics, 2005
ISBN: 0071459251

Like the robotics book above, but skewed toward "bionics" experiments.

Build Your Own Combat Robot
by Pete Miles, Tom Carroll
Osborne McGraw-Hill, 2002

Older book on building battling robots. Still a great resource if you want to construct a heavy-duty bot.

Build Your Own All-Terrain Robot
by Graham McGowan
McGraw-Hill, 2004
ISBN: 007143741X

Construct a vehicle-class, remotely controlled robot. Plans for two video controlled robots.

Build Your Own Humanoid Robots
by Karl Williams
McGraw-Hill, 2004
ISBN: 0071422749

This has complete construction plans for bi-pedal robots, typically using metal construction. Be sure to also check out Karl's other robot construction books, *Insectronics: Build Your Own Walking Robot* (ISBN: 0071412417), *Amphibionics: Build Your Own Biologically Inspired Reptilian Robot* (ISBN: 007141245X), and *Build Your Own Humanoid Robots* (ISBN: 0071422749).

FIRST Robots: Aim High: Behind the Design
by Vince Wilczynski, Stephanie Slezycki
Rockport Publishers, 2007
ISBN: 1592533663

Inspiration and how-to in building and competing in the FIRST Robotics contests.

JunkBots, Bugbots, and Bots on Wheels
by David Hrynkiw, Mark Tilden
McGraw-Hill Osborne Media, 2002
ISBN: 0072226013

All about BEAM robotics: Biology, Electronics, Aesthetics, Mechanics.

Kickin' Bot: An Illustrated Guide to Building Combat Robots
by Grant Imahara
Wiley, 2003
ISBN: 0764541137

Tools, techniques, and how-to on building heavy-duty combat robots. Grant is a co-host on the popular Myth Busters television show.

Linux Robotics
by D. Jay Newman
McGraw-Hill, 2005
ISBN: 007144484X

Constructing a larger-scale robot

The *Evil Genius* series of books is presented in a large format that looks like a lab journal.

using a PC-style motherboard and the Linux operating system.

Mobile Robotic Car Design
by Pushkin Kachroo, Patricia Mellodge
McGraw-Hill, 2004
ISBN: 007143870X
How to build a fairly sophisticated scale model robotics car.

Muscle Wires Project Book
by Roger G. Gilbertson
Mondo-Tronics, 2000
ISBN: 1879896141
Book and kit about "Muscle Wires," a brand of shape memory alloy available from various sources such as The Robot Store. Projects include Boris, a walking robot.

PIC Robotics: A Beginner's Guide to Robotics Projects Using the PICMicro
by John Iovine
McGraw-Hill, 2002
ISBN: 0071373241
Robot project book using the PIC microcontroller as the central brain.

Practical Robotics: Principles and Applications
by Bill Davies
CPIC Technical Books, 1997
ISBN: 096818300X
General book on different mechanical construction ideas, with practical details. Not much actual robot building in this one, but the pieces can readily be put together to make one.

Robot Builder's Bonanza, Third Edition
by Myke Predko
McGraw-Hill, 2006
ISBN: 0071468935
The third edition of my original *Robot Builder's Bonanza* (first edition 1987, second edition 2000). I didn't have any involvement with this edition, which was updated by author Myke Predko.

Robot Builder's Cookbook
by Owen Bishop
Newnes, 2007
ISBN: 0750665564
Projected oriented plans for several robots.

Robot Builder's Sourcebook
by Gordon McComb
McGraw-Hill, 2003
ISBN: 0071406859
Over 700 telephone book-sized pages on what it is and where to buy robotics parts. A little long in the tooth, and some of the sites no longer exist, but still a valid resource.

Robot Building for Beginners
by David Cook
APress, 2002
For raw beginners with little or no experience. Very good for those just starting out. Be sure to also take a look at the author's *Intermediate Robot Building* (ISBN: 1590593731), which discusses more in-depth concepts.

Robotics Demystified
by Edwin Wise
McGraw-Hill, 2004
ISBN: 0071436782
Simplifying the concepts of robotics. Includes electronics, mechanics, programming, and other robot-related technologies.

Robot DNA (series)
McGraw-Hill; 2002-2004
Series on robotics construction, consisting of the following titles: *Building Robot Drive Trains* (ISBN: 0071408509), *Programming Microcontrollers* (ISBN: 0071408517), and *Constructing Robot Bases* (ISBN: 0071408525).

Robots, Androids and Animatrons, Second Edition
by John Iovine
McGraw-Hill, 2001
ISBN: 0071376836
Several entry-level projects, including a small six-legged walking robot.

LEGO Robotics and LEGO Building

LEGO Mindstorms continues to be one of the best ways to get into the field of robotics.

Building Robots With Lego Mindstorms
by Mario Ferrari, Giulio Ferrari, Ralph Hempel
Syngress, 2001
ISBN: 1928994679
Highly recommended guide to intermediate- and advanced-level LEGO Mindstorms robotics.

Building Robots with LEGO Mindstorms NXT
by Mario Ferrari, Guilio Ferrari, David Astolfo
Syngress, 2007
ISBN: 1597491527
Updated discussion centering around the Mindstorms NXT set.

Creative Projects with LEGO Mindstorms
by Benjamin Erwin
Addison-Wesley Pub Co, 2001
ISBN: 0201708957
Using color illustrations, this book demonstrates over a dozen fun and unusual LEGO Mindstorms creations. A great book for the classroom.

Unofficial LEGO MINDSTORMS NXT Inventor's Guide
No Starch Press, 2007
ISBN: 1593271549
Construction plans for various NXT vehicles and robotics, many of them very unusual and eclectic.

Technical Robotics, Theory, and Design

For the more scholarly inclined.

Mobile Robots: Inspiration to Implementation
by Joseph L. Jones, Anita M. Flynn, Bruce A. Seiger
AK Peters Ltd, 1999
ISBN: 1568810970
Hands-on guidebook to constructing mobile robots.

Robot Evolution: The Development of Anthrobotics
by Mark Rosheim
John Wiley & Sons, 1994

ISBN: 0471026220

Technical overview on robots and robot systems. For the hard-core roboticist.

Robotic Explorations: An Introduction to Engineering Through Design
by Fred Martin
Prentice Hall, 2000
ISBN: 0130895687

This is not an inexpensive book, but it's one of the best for teaching the fundamentals of robotics in the classroom. The author provides explicit examples and exercises using LEGO Technic pieces and the MIT Handy Board controller, but you can apply what you learn to most any robotics platform.

Sensors for Mobile Robots: Theory and Application
by H. R. Everett
AK Peters Ltd, 1995
ISBN: 1568810482

Overview book designed to help familiarize you in the role of sensors in robotics, and the available technologies.

ISBN: 0262611376

This book is a review of functional systems and techniques in artificial robots used by researchers and companies.

Behavior-Based Robotics
by Ronald C. Arkin
MIT Press, 1998
ISBN: 0262011654

Overview text of various behavior-based robotics techniques. The book contains references to robots that have used the various AI techniques described.

Cambrian Intelligence: The Early History of the New AI
by Rodney Allen Brooks
MIT Press, 1999
ISBN: 0262522632

In these pages you'll find the collection of Rodney Brooks' (he's a professor at MIT) earlier works on robotics intelligence.

Robot: Mere Machine to Transcendent Mind
by Hans Moravec
Oxford University Press, 2000
ISBN: 0195116305

Speculative but informed look at one possible future with intelligent robots.

Robot Programming: A Practical Guide to Behavior-Based Robotics
by Joe Jones, Daniel Roth
McGraw-Hill, 2003
ISBN: 0071427783

Robotics maven Joe Jones provides practical examples of implementing behavioral programming. Great follow-on book to this author's *Mobile Robots*.

Vehicles: Experiments in Synthetic Psychology
by Valentino Braitenberg
MIT Press, 1986
ISBN: 0262521121

Highly influential book on how simple machines can mimic living organisms. **SV**

CONTACT THE AUTHOR

Gordon McComb can be reached via email at robots@robotoid.com

Artificial Intelligence and Behavior-Based Robotics

These books detail various approaches to endowing your robotic creations with artificial intelligence.

An Introduction to AI Robotics
by Robin R. Murphy
MIT Press , 2000
ISBN: 0262133830

This book is a round-up of several popular approaches to artificial intelligence in robotics.

Artificial Intelligence and Mobile Robots
by D. Kortenkamp, R.P. Bonasso, R. Murphy
MIT Press, 1998

Robotics maven Joe Jones provides practical examples of implementing behavioral programming.

ROBOGAMES

World's Largest Robot Competition
-Guinness Book of Records

North America's Top Ten Geek Fests
-Wired Magazine

SportCenter's Top Ten
-ESPN SportsCenter

"If you are a robot enthusiast, I would definitely encourage you to attend the RoboGames... Take a plane, train, space elevator, but definitely go!"
-SERVO Magazine

"Impossible to Imagine, Impossible to Forget!"
-Robot Magazine

Compete at RoboGames 2008!
Last year, over 1000 builders from around the world brought over 800 robots to San Francisco, in the 4th annual international event. This year, we expect even more robots and engineers to compete. Be one! With 80 different events, there's a competition for everyone - combat, androids, sumo, soccer, Lego, art, micromouse, BEAM, or Tetsujin! More than half the events are autonomous. Even if you just come to watch, you'll be overwhelmed with the diversity.

Last year, RoboGames hosted teams with over 800 robots from Argentina, Australia, Austria, Brazil, Canada, China, Colombia, Czech Republic, Denmark, Germany, India, Indonesia, Iran, Japan, Korea, Mexico, Netherlands, Peru, Singapore, Slovenia, Sweden, Taiwan, UK, and the USA.

Be a RoboGames Sponsor!
RoboGames is the world's largest open robot competition - letting people of any age, gender, nationality, or affiliation compete. Sponsoring RoboGames not only helps more people to compete, but also gets your company unrivaled press coverage and visibility. The event has been covered by CNN, ESPN, Fox, CBS, ABC, NBC (live), EBS Korea, NHK Japan, BBC, and countless print and web companies. Your logo can be everywhere the cameras turn!

Rent a Booth!
Booth spaces are at the front of the venue, ensuring lots of traffic. With 3000-5000 people each day, you're company will get amazing traffic!

Events:

Combat: 340 lbs, 220, 120, 60, 30, 3, & 1 lbs

Androids: Wrestling, Demonstration, Stair Climbing, The Eagle, Door Opening, The Toss, BasketBall, Lift and Carry, Marathon, Obstacle Run, Penalty Kick, Dash, 3:3 Soccer, Weight Lifting

Open Events: Fire-Fighting, Robomagellan, Maze/MicroMouse, Walker Challenge, Biped Race, Robot Triathlon, Line Slalom, Ribbon Climber, Vex Open, Lego Challenge, Lego Open, Aibo Performer, Balancer Race, Best of Show, Bot Hockey

Sumo: 3kg - Auto & R/C, 500g , 100g , 25g, Humanoid

Robot Soccer: Biped 3:3 & 5:5 , Mirosot 5:5 &11:11

Junior League: Lego Challenge, Lego Open, Lego Magellan, Woots & Snarks, Handy Board Ball, BotsketBall, 500 g Sumo, 120 lb combat, Best of Show, Vex Open

Tetsujin (ExoSkeleton): Lifting, Walking, Carrying

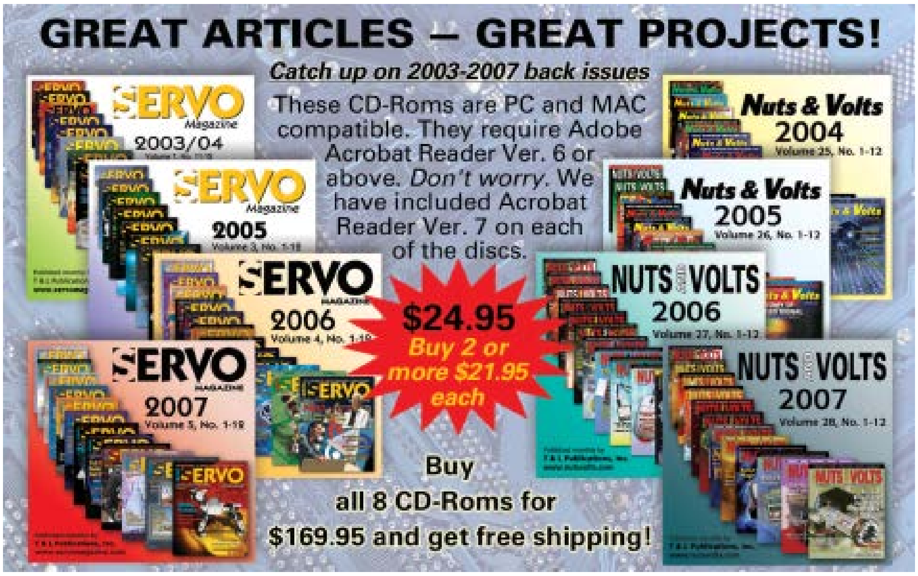
Art Bots: Static, Kinetic, Bartending, Musical, Drawing

BEAM: Speeder, Photovore, RoboSapien Hacker

June 12-15th, 2008
San Francisco, CA
http://RoboGames.Net

The *SERVO* Webstore

Attention Subscribers ask about your discount on prices marked with an *



SERVO Magazine Bundles

Published by T & L Publications, Inc.

$57 per bundle

Save $10 off the normal price!!

Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, and 07.

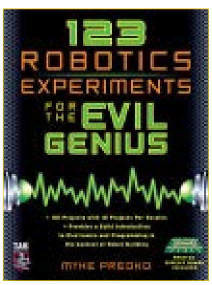
FIRST Robots: Rack 'N' Roll Behind the Design

by Vince Wilczynski / Stephanie Slezycki

The second annual book highlighting the creativity and process behind 30 winning robot designs from the 18th annual international FIRST Robotics Competition. The FIRST organization, founded by Dean Kamen (inventor of the Segway), promotes education in the sciences, technology, and engineering in collaboration with sponsors including Motorola, Xerox, NASA, Delphi, General Motors, and other companies invested in science education. **$39.95**

123 Robotics Experiments for the Evil Genius

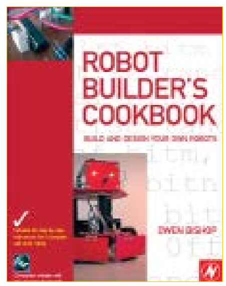
by Myke Predko



If you enjoy tinkering in your workshop and have a fascination for robotics, you'll have hours of fun working through the 123 experiments found in this innovative project book. More than just an enjoyable way to spend time, these exciting experiments also provide a solid grounding in robotics, electronics, and programming. Each experiment builds on the skills acquired in those before it so you develop a hands-on, nuts-and-bolts understanding of robotics — from the ground up. **$24.95**

Robot Builder's Cookbook

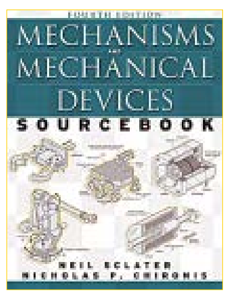
by Owen Bishop



This is a book for first-time robot builders, advanced builders wanting to know more about programming robots, and students in further and higher education tackling microcontroller-based practical work. They will all find this book a unique and exciting source of projects, ideas, and techniques to be combined into a wide range of fascinating robots. **$29.95**

Mechanisms and Mechanical Devices Sourcebook

by Neil Sclater / Nicholas Chironis

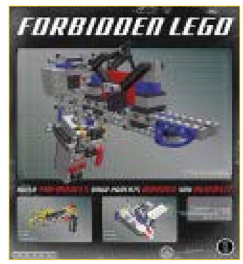


The fourth edition of this invention-inspiring engineering resource covers the past, present, and future of mechanisms and mechanical devices. You'll find drawings and descriptions of more than 2,000 components that have proven themselves over time and can be incorporated into the very latest mechanical, electromechanical, and mechatronic products and systems. Overviews of robotics, rapid prototyping, MEMS, and nanotechnology, along with tutorial chapters on the basics of mechanisms and motion control, will bring you up-to-speed quickly on these cutting-edge topics. **$89.95**

Forbidden LEGO

by Ulrik Pilegaard / Mike Dooley

Build the Models Your Parents Warned You Against.

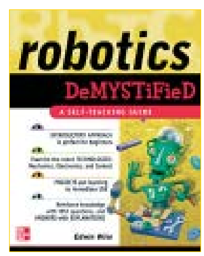


Forbidden LEGO introduces you to the type of free-style building that LEGOs master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse. **$24.95**

Robotics Demystified

by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!

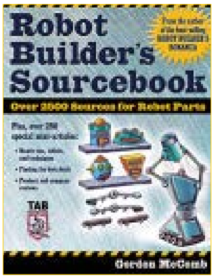


Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **$19.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

To order call 1-800-783-4624

Robot Builder's Sourcebook
by Gordon McComb

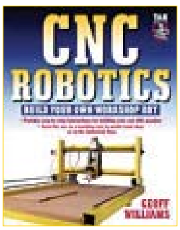


Fascinated by the world of robotics but don't know how to tap into the incredible amount of information available on the subject? Clueless as to locating specific information on robotics? Want the names, addresses, phone numbers, and websites of companies that can supply the exact part, plan, kit, building material, programming language, operating system, computer system, or publication you've been searching for? Turn to *Robot Builder's Sourcebook* — a unique clearinghouse of information that will open 2,500+ new doors and spark almost as many new ideas. **$24.95**

Oil prices getting to you, too?! Stay home, read a book.

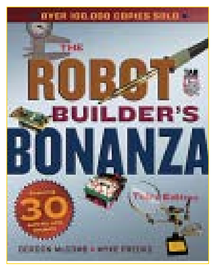
The online store @ www.servomagazine.com

CNC Robotics
by Geoff Williams



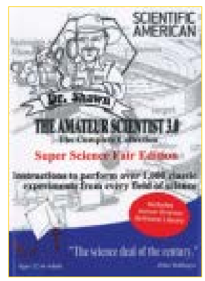
CNC Robotics gives you step-by-step, illustrated directions for designing, constructing, and testing a fully functional CNC robot that saves you 80 percent of the price of an off-the-shelf bot — and that can be customized to suit your purposes exactly, because you designed it. Written by an accomplished workshop bot designer/builder, this book gives you everything you need. **$34.95**

Robot Builder's Bonanza Third Edition
by Gordon McComb / Myke Predko



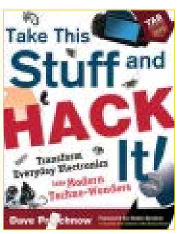
Everybody's favorite amateur robotics book is bolder and better than ever — and now features the field's "grand master" Myke Predko as the new author! Author duo McComb and Predko bring their expertise to this fully-illustrated robotics "bible" to enhance the already incomparable content on how to build — and have a universe of fun — with robots. **$27.95**

The Amateur Scientist 3.0 The Complete Collection
by Bright Science, LLC



There are 1,000 projects on this **CD**, not to mention the additional technical info and bonus features. It doesn't matter if you're a complete novice looking to do their first science fair project or a super tech-head gadget freak; there are enough projects on the single CD-ROM to keep you and 50 of your friends busy for a lifetime! **$26.99**

Take This Stuff and Hack It!
by Dave Prochnow



Transform common household items into really cool stuff. You don't need to be an electronics genius to get started turning everyday items into high-performing wonders. With how-to guru Dave Prochnow's step-by-step directions and fully illustrated plans, even beginners can hack their way to a high-tech home, cooler toys, and less yard work. Certain to fire your imagination and start you plotting new, original, and even more creative wonders you can make from ordinary household items, *Take This Stuff and Hack It!* is the perfect gift for your inner inventor. **$27.95**

Blow Out Special $9.95!

From HomoSapien to RoboSapien

SERVO Magazine T-Shirts
For men and women

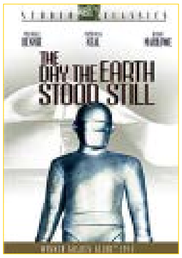
Before R2D2 there was R1D1

Get your very own limited-edition *SERVO Magazine* T-shirt. Shirts come in sizes S, M, L, and are available in either black or white.
All shirts are 100% preshrunk cotton. Please check availability on our website.

Or order online www.servomagazine.com

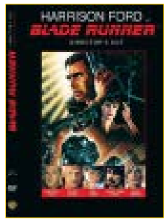
CLASSIC SCIENCE FICTION DVDS!

The Day the Earth Stood Still



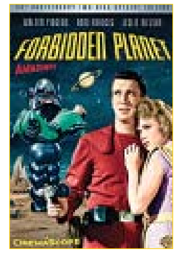
An alien (Klaatu) with his mighty robot (Gort) land their spacecraft on Cold War-era Earth just after the end of World War II. They bring an important message to the planet that Klaatu wishes to tell to representatives of all nations. *Directed by: Robert Wise*
Actors: Michael Rennie, Patricia Neal
Run Time: 92 minutes **$14.95**

Blade Runner (The Director's Cut) (1982)



In a cyberpunk vision of the future, man has developed the technology to create replicants, human clones used to serve in the colonies outside Earth but with fixed lifespans. In Los Angeles, 2019, Deckard is a Blade Runner, a cop who specializes in terminating replicants.
Directed by: Ridley Scott
Rated: R — Violence **$19.95**

Forbidden Planet: 50th Anniversary Edition

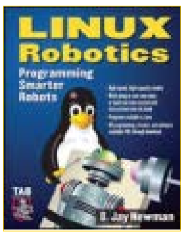


A starship crew goes to investigate the silence of a planet's colony only to find two survivors and a deadly secret that one of them has. Dr. Morbius and his daughter Altaira have somehow survived a hideous monster which roams the planet.
Directed by: Fred M. Wilcox
Run Time: 98 minutes **$26.95**

BACK ROOM SPECIALS

Linux Robotics

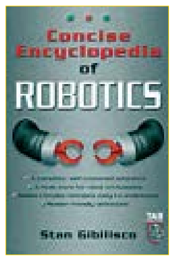
by D. Jay Newman



If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need, plus complete training in doing original programming. **$34.95** **Sale Price $ 29.95**

Concise Encyclopedia of Robotics

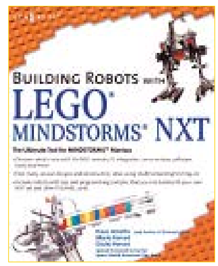
by Stan Gibilisco



Written by anaward-winning electronics author, the *Concise Encyclopedia of Robotics* delivers 400 up-to-date, easy-to-read definitions that make even complex concepts understandable. Over 150 illustrations make the information accessible at a glance and extensive cross-referencing and a comprehensive bibliography facilitate further research.
$19.95 **Sale Price $9.95 *"Only 2 Left"***
Pub date: November 12, 2002

Building Robots with LEGO Mindstorms NXT

by Mario Ferrari, Guilio Ferrari



The Ultimate Tool for MINDSTORMS® Maniacs, the new MINDSTORMS kit has been updated to include a programming brick, USB cable, RJ11-like cables, motors, and sensors. This book updates the robotics information to be compatible with the new set and to show how sound, sight, touch, and distance issues are now dealt with. **$39.95**
Sale Price 35.95 *Phone orders only.*

SERVO'S CUSTOM EMBROIDERED HAT

This is one good looking hat, with its stone washed blue cap and bright yellow/gold stitching. There's no better way to show everyone that you're well endowed with a bigger - than - average brain then by proudly displaying your affiliation with SERVO Magazine. **$ 14.95***

Hydrogen Powered Racer and Hydrogen Station

With this kit, you now can see and feel the future of energy generation in your own hands!

Recently named as one of the Best Inventions of 2006 by *Time Magazine*, the H-racer is now the best selling fuel cell product in the world. The H-racer is a micro-version of what engineers and scientists have been dreaming about for real cars: combining hydrogen with oxygen to generate a DC current to power an electric motor. Unlike a gas-powered car engine, the only byproducts of this electrochemical process are electricity, heat, and pure water. With the H-racer, you can witness the power of new energy technology in the palm of your hand. Horizon has created this unique, patented miniature fuel-cell car and hydrogen refueling station. All you need to do is add water. ***Only* $115.00 plus S/H**

Building Character through ... Robot Building!

by Robin Lemieux

I Know Absolutely Nothing About Robots ... Well, Almost!

For the last two decades, I've been surrounded by robots and electronics. Could I recite Ohm's Law? Could I tell you what a capacitor does? Do I know the difference between servo and stepper motors? Is there a difference? Do I know how to design and build an autonomous robot?

This last April marks 20 years of working at T&L Publications for me — publishers of *Nuts & Volts* and *SERVO Magazine*. I have proofread literally thousands of technical articles, but probably couldn't tell you what most of it was about. (I just have to remember to keep my Ps and Qs straight ... literally!)

We have been going to robot events since 1997, when we took our three-year-old son to watch Robot Wars in San Francisco, CA. It was one of the top three best behavior moments for him because he was completely hypnotized watching the sparks and flying carnage. It was the first combat event we had ever been to and we all thought it was pretty darn cool.

After that, we attended and sponsored Battle Bots in Long Beach, CA and continue to be a part of many events, ranging from larger ones like FIRST to smaller club competitions that are held across the US. (I even judged the Robot Bartender competition at RoboGames last year.) I've also watched a lot of women participate in combat robotics, both as drivers in the ring and welders in the pits. (All the really hot chicks know how to weld.) The different competitions really bring families together.

The impact we've seen on kids, from participating in these different events, is phenomenal. Two of my top writers — Evan and Bryce Woolley of Twin Tweaks fame — came up through the ranks of robot competitions and are now attending college, armed with a plethora of information that they wouldn't have started with otherwise. Read their column in the March 08 issue to see what I mean.

I tell you, these two writers are some sharp kids that have it together. It's obvious in the style and tone of their writing. They have truly benefited from the challenges that arise when participating in this sport. And, in our current climate of political correctness where "everyone's a winner, no one's a loser" which only breeds more losers (in my opinion), it is downright refreshing to see that having to deal with a time-sensitive goal, hard work, frustration, and disappointment actually builds character. What a powerful message it is for kids to see other kids work towards something they might otherwise feel is impossible and then obtain some level of success.

I grew up with the message from my folks that you have to work hard to get the things you want in life. That not everyone is the same and those who are willing to put the time in will reap the most benefit.

Nuts & Volts monthly Personal Robotics columnist, Vern Graner, wrote an excellent Appetizer in the August 07 issue of *SERVO* about recovering our technical literacy and the lack of school curriculum in areas like metal/machine shop, wood shop, electronics, or ham radio clubs. Check the number of US college students studying to be engineers as compared to Asian countries.

Just a couple months ago, we visited the Bay area in northern California and stopped in to see the guys at TechShop. In case you're unfamiliar, TechShop is "a fully-equipped open-access workshop and creative environment that lets you drop in any time and work on your own projects at your own pace." A monthly or annual membership buys you time on their wide variety of machinery and tools. Even if you have no experience using a mill, press, or plasma cutter, you can attend a short instructional session and be qualified to use them in about an hour. You can brainstorm with fellow builders and even rent a storage space to keep your stuff there. We have some photos from our visit on the website at **www.nutsvolts.com**, so be sure and check those out. We also visited our good friends at Make Magazine, who have helped to rekindle the DIY spirit, not just in the technical arts but in many areas, from crafting to green transportation.

Attending events like RoboNexus, RoboGames, RoboDevelopment, and Maker Faire has really opened my eyes to a unique and united sub-culture. All of these different events and venues are helping to bring basic technical knowledge and principles back to the forefront and encourage actual, hands-on participation. A social network is being created and nurtured to the point where being a geek totally has the cool factor.

In the next few months, we're going to be working with Schmartboard and their new system called "Soldering By Numbers." This is kind of like paint-by-numbers except you follow a color-coded schematic to assemble a circuit board to build an electronic project. It should be the perfect way to get non-techies started. You don't have to have a clue about what resistors, capacitors, or trimpots are. We're going to put the system to the acid test on someone who really doesn't have a clue: Me. If I can do it, anybody can. If you happen to be at the Bay Area Maker Faire, come by our booth to see a demo.

We are also putting together a robotic starter kit for all those parents who have come up to us at shows and asked us how to get their kid started in robotics (not sure if they meant their 10-year-old or their husband). We're hoping to have these ready in the next month or so, and plan to have them at our booth at RoboGames.

While we're big advocates here at *SERVO* of learning by doing, another good way to get started or to upgrade your skills is to read. This month, Gordon McComb's Robotics Resources is all about great books on robotics — some old, some new — that cover everything you need to know about automatons but were afraid to ask. Armed with a little information, a few basic tools, and some inspiration, you too can learn by doing and have fun in the process. I know I'm going to!

Be sure and stay tuned for updates on my progress with the soldering (hopefully, there will be progress!). And try to attend some of these excellent events with your kids. There's so much to see that will inspire and amaze you. **SV**

Then and NOW

ROBOT MUSCLES — ELECTRIC MOTORS

by Tom Carroll

What would a robot be without their version of muscles? Well, maybe it would be just a computer. Now, that's not so bad if all you want to do is input data and information via a keyboard, CD-ROM, voice or some other means, have the computer do something with that information, and then have the computer store or output that data onto a screen, printed page, a speaker, or some other passive device. Most of us enjoy our computers quite a bit, but we expect some sort of movement from them. Yes, hard drives and CD-ROMs are spinning, arms move across the spinning discs to search for data, discs are ejected, and spinning fans cool the hot components. Printers zip printheads across a page that is moved by another motor, but this is not the movement we want as robotics enthusiasts. We want our computer to move across the table or floor — intelligently. We want a robot!

FIGURE 1. Hero's ÆOLIPILE.

Early Mechanical Power

This movement is going to require some sort of linear mechanical actuator or rotary motion producing device. We immediately think of electric motors for this type of movement, but mechanical power has long been supplied by simple steam power. Hero of Alexandria in 150 BC built what is known as the first steam engine. Figure 1 shows his simple ÆOLIPILE that spun when an enclosed boiler fed steam up the two side tubes into the sphere and out the two jets. Hero (also known as Heron) is credited with designing (not actually building) several ingenious automatons, or robots as we now know them. It certainly was inefficient as the steam just spouted out into the air and not against a piston in a cylinder, as came later with James Watt's engines of 1769. The spinning sphere amazed people of Hero's time, but Englishman Thomas Savery developed the first engine to be used for a task — removing water from a mine in 1698. In 1712, Thomas Newcomen used some of Savery's ideas to make a vast improvement on the mine pump. It was Watt's piston and cylinder design connected to a rotating flywheel that made the Industrial Revolution a reality.

FIGURE 2. Mag beam engine.

Train locomotives and even the old stern wheelers used steam power to create a reciprocating motion that was, in turn, changed to rotary motion. Today's steam turbines are even more efficient and are the power source for all nuclear-powered ships and submarines, as well as coal, nuclear, and oil-fired electric power plants.

A couple of years ago, I wrote about a steam powered robot called Steam Man and other steam-powered robots in the June '06 issue of *SERVO*. Some great mechanical things have used steam — and even wound springs for power — but it is the electric motor that has proven to be the most adaptable as motive power for so many mechanisms in the world, especially robots.

Electric motors have the distinct ability to be able to utilize an external power source — whether battery or AC power line — that can be easily turned on or off or modulated to control the motor. Figure 2 is a photo of a very early electric engine that I saw on Jerry Everard's blog that reminds me of a combination steam engine mechanism hooked to electric solenoids. It converts reciprocal motion to rotary motion like a steam engine. Figure 3 is a motor that I saw advertised on eBay. It is a Curtis & Crocker bipolar motor from the late 1880s that I believe is a fan motor. It was being sold as a collectable. Many electric motors of that era had open structures that exposed the internal wiring to errant fingers. Today's liability lawyers would salivate over seeing such a motor on today's market.

Motors for Robots

Steam engines, hydraulics, pneumatics, spring-wound motors, shape-memory actuators (Muscle wires), solenoids, and even twisted rubber bands have powered experimental, industrial, service, and toy robots for years, but we really rely on the good old electric motor for robot power. And that 'good old motor' was usually a DC permanent magnet (PM) motor. I won't go into detail how each type of electric motor works, as I am sure the vast majority of *SERVO*'s readers are quite familiar with the repulsion and attraction of the motor's armature poles and the field poles. We all know how the commutator at the end of the armature of a DC motor allows the brushes to continually change the magnetic fields on the armature to keep it spinning. There are numerous links on the Internet that will go into depth on all types of DC and AC motors.

Back in the '70s, robot builders relied on surplus businesses, junk yards, car parts, and toys for their motors. Motors have always fascinated me and I had amassed over 500 electric motors and gearmotors in my collection several years ago; most were from surplus places. I got to thinking, "why does anybody need so many motors? Am I ever going to build a hundred different types of robots?" Since I couldn't come up with any good answers and with my wife's urging, I managed to cull my collection down to about 50 of the best ones and then gave a bunch away at Seattle Robotics Society meetings, sold a bunch, and donated some to schools.

Surplus Store Motors

The best of these motors came from a favorite surplus place of mine in Pasadena, CA that was about a 40 mile drive north from my home at the time in Long Beach, CA. C&H Sales had thousands of military surplus motors and the staff knew me well enough to let me wander in the back of the store, plowing through bins of complex military mechanisms to pull out the best motors. I could test them on-site for speed, torque, operating voltage range, current at various loads, and mounting methods. I almost always chose gearhead motors as these didn't require a separate attached gearbox or belt reduction.

Some of the best gearmotors were planetary drives made by TRW and other suppliers to the government. Something that I found out rather quickly is a person cannot go by the listed current draw or voltage to determine the actual torque and output speed of a particular motor. As an example, a particular gearmotor may draw one amp at 28 VDC at full load and 20 RPM with 100 oz-in of torque, and another only 500 mA at 28 VDC for the same output. Look at a surplus motor catalog and you'll see what I mean. Choose wisely.

Quite a few of these motors — especially the military variety — were 24 or 28 VDC, but would run nicely at a reduced voltage such as 12 VDC. Pittman made (and makes) a great series of gearmotors that operate nicely at 12 VDC, and I used many of these for robot appendages, grippers, and head motion applications. C&H has shrunk quite a bit in the past few years and now sells most of their stock on eBay. In the '80s, they had a great monthly catalog that went to quarterly several years later and was a great reference for their motors. Considering that these motors were surplus, they probably had been sitting on shelves or dark military warehouses for years and were not constructed using the latest technology.

AlNiCo and ferrite magnets seemed to be the technology for PM motors in those days. Universal motors were one of the most popular military surplus styles, though these motors were not particularly good for robot applications. They used an electromagnet winding for the field poles instead of permanent magnets and could operate on AC, as well as DC current. Vacuum cleaners still use this type of motor. Don't use these for robots.

FIGURE 3. Curtis & Crocker bipolar motor.

Another popular type of DC motor was the pancake motor or printed circuit armature motor. These motors have the advantage of compactness that is especially nice when coupled to a geartrain. They have low weight, are fairly energy efficient, and start and stop fast due to low inertia of the armature. I was given a box of PMI pancake motors that I used for all sorts of projects. I did find that they could be damaged when they were overloaded as the armature did not have an iron core to soak up and dissipate excess heat. I found this out the hard way when a pancake motor I was using had the copper-wound armature so cooked that it was warped.

Figure 4 from the University of New

FIGURE 4. Pancake motor.

FIGURE 5. Pancake motor armature.

South Wales, Australia is a typical pancake style motor. Figure 5 shows a side view of the thin armature. Notice the eight PM magnets and the black magnetizing wire interlaced around the magnets. There are also eight more magnets on the other side of the case that's not shown. The two brushes are placed 45 degrees apart and rub against the darker center commutator ring of the armature. Newer pancake motors use rare earth magnets. Other motors that I found to be excellent for large robots were A-BEC wheelchair motors, especially if they came with attached wheels and a hub that could be unlocked so the robot could be easily pushed along by hand. They were originally designed for a British electric wheelchair manufacturer and the wheels were made of a non-marring rubber that was kind to nice floors. I used these types of motors on the four robots that I built for the film, 'Revenge of the Nerds,' and also for a promotional robot that I built for a Beverly Hills dentist who used the robot to educate kids in proper dental health care.

Electric Motors of Today

The greatest improvement in today's PM DC motor is the use of rare earth field magnets in the place of the old ferrite and AlNiCo magnets. There are many varieties of motors used in robots, and various sub-configurations, but the most popular is the PM DC motor. This is the one used in the small model aircraft servos that are so popular with small robots to large drive motors for combat robots. coreless, brushless, and pancake style motors are also popular, as well as stepper motors. Linear motors are used in industrial applications but have found little use in experimenter's machines. AC motors also find little use in today's hobbyist robots.

The PM DC Motor

Quite often, the first electric motors that new robot builders are exposed to are radio control model airplane servos. The actual motor may be hidden in a plastic case and is the key component of the servo. HiTec, makers of popular servos dedicated to robots speak of their "neodymium rare earth magnets used in the coreless motor." Futaba's line of digital servos come with a brushless motor that uses samarium cobalt magnets as the field magnets." This is not just hype from maker's advertisements; this is the way motor technology is heading today. Yes, these servos are quite a bit more expensive than the standard servo of 20 years ago, but they will last far longer, draw less current, and react faster.

Rare Earth Magnets

I've tossed about the term 'rare earth magnets' a bit when describing the field magnets of these newer motors, but the use of these magnets was the turning point in DC motor technology. In 1966, the high energy compound, samarium cobalt ($SmCo_5$) was discovered that could be magnetized to an energy product of 18 megagauss oersteds (18 MGOe). In 1972, the ratio of the alloy was changed to SM_2Co_{17} that allowed a higher energy product of 30 MGOe. In 1983, a new formulation of neodymium-iron-boron ($Nd_2Fe_{14}B$) was developed that had an even higher product of 35 MGOe. Today's standard for these magnets seems to be 40 MGOe, or a grade N40 for commercially-available super magnets.

As you might expect, there are numerous patents on these types of permanent magnets that contain the above elements, and even lanthanum and cerium. Most of the rare earth permanent magnets are of the sintered or bonded alloy powder types. The more powerful sintered magnets are compressed powder formed in a die under heat and are similar to the earlier ceramic magnets and can be easily broken. These types are usually plated with a copper-nickel coating and are shiny. The bonded magnets use a polymer base to hold the powder together and are tougher but weaker. The US patents expire for most of them in 2014.

DC Motor Types

Most of us are fairly familiar with the popular brushed DC motor used in tools, toys, and many other lower-cost applications. This is not to say that these types of motors may not be best for a particular robot application as the following motors are of the brushed type. The most powerful type of robot motor is the Magmotor series. The chart in Table 1 shows the three models of Magmotors. Notice the torque at over 3,800 oz-in (that is without an attached gear train). Can you imagine the power and torque output with gearing and over-volting? I calculate 8.6 HP at 36 volts for the more massive C40 (which can take the heat build-up). Now that's robot power!

There are many motors that are great for the larger robot, such as the large variety sold by National Power Chair (NPC) Robotics. They remanufacture many brands and types of electric wheelchair motors. One great thing about using wheelchair motors is the built-in gearbox and wheel attachment. Considering that these things are used all day long to move a human-sized weight around a house and in public,

TABLE 1. Three Models of Magmotors.

C40-300	S28-400	S28-150
3.8 Horsepower	4.5 Horsepower	3 Horsepower
4" Diameter	3" Diameter	3" Diameter
6.9" Long	6.7" Long	4" Long
3,840 oz-in Torque	3,720 oz-in Torque	1,970 oz-in Torque
84% Efficiency	83% Efficiency	82% Efficiency
24 Volts	24 Volts	24 Volts
4,000 RPM	4,900 RPM	6,000 RPM
11.9 Pounds	6.9 Pounds	3.8 Pounds
Built-in Capacitors	Built-in Capacitors	Built-in Capacitors
Ferrite Magnets	Neodymium Magnets	Neodymium Magnets

Listings from Robot Book's website.

they are constructed to be quite efficient in current draw and operate very quietly — perfect for a large robot. Quite a few have been used in combat robots but they don't exactly have the brute force of the powerful Magmotors. NPC also offers a series of wheels to fit various motors.


FIGURE 7. S28-400 Magmotor.

FIGURE 6. C40-300 Magmotor.

Brushless DC Motors

Brushless motors are another variety that have found a lot of uses in all sizes of robots. They differ from the brushed types mentioned earlier in that they have a wound stator (field) and a series of permanent magnets on the rotor (armature) — exactly opposite of the brushed varieties. Since something has to produce the rotating magnetic field, the stator windings are electronically commutated instead of brushes wiping a spinning commutator. Internal circuitry produces a three phase trapezoidal wave form (though some motors have a four phase circuitry). Muffin style fans are a popular application and are usually two phase. Since there are no friction-producing brushes, they are more efficient and can operate at higher speeds. Figure 8 (from the UC Berkeley Engineering Department) shows a typical larger brushless motor. Notice the electronics board that produces the desired waveforms.

Coreless DC Motors

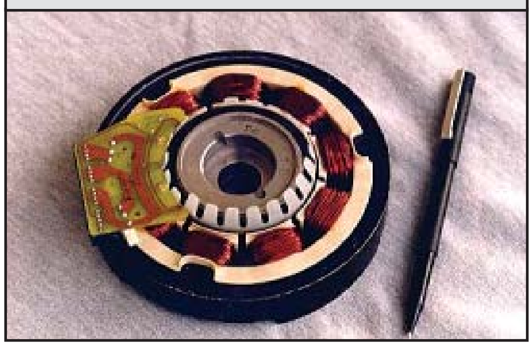
Coreless motors have a great advantage of low inertia in the moving rotor and a low mechanical time constant with the result being the ability to rapidly rev up to speed, and just as quickly come to a stop. Another nice aspect of the coreless motor is the absence of the annoying cogging as the rotor poles pass by the field magnet poles, allowing smooth slow speeds. Though brushed, the lack of the iron core dramatically reduces rotor inductance and commutator arcing. This, in turn, reduces EMI noise in the control circuitry. Coreless motors are usually small and are popular in the better lines of hobby servos. Figure 9 from GlobalSpec shows the unique basket weave winding of the rotor.

Winding Down

These are just a few of the most popular motors used in experimental robot construction. The selection of the proper motor for your particular design or application should be preceded by a thorough study of motor manufacturer's websites and some of the more popular robot reference books such as Gordon McComb's *Robot Builder's Sourcebook*. Motors can range from the tiny vibrator motors used in cell phones and pagers, to the huge Magmotors. All have their place in the design of experimenter's robots and are vital components that give our creations motion. SV

Tom Carroll can be reached via email at TWCarroll@aol.com

FIGURE 9. Coreless DC motor.


FIGURE 8. Brushless motor.


Advertiser Index

CrustCrawler Robotics

"Build Smarter."

AX12 Smart Arm!

Now with 3 gripper configurations!

Hardware Features:

- ✓ More than 1.5 pounds of lifting capacity
- ✓ Powerful 440 oz-in main joints
- ✓ Single cable network connections
- ✓ Precision, carbon steel, ball bearing turntable
- ✓ Integrated Pem nuts for quick and easy construction
- ✓ Rugged, type II anodyzed finish

Operating Features:

- ✓ Blazing 1,000,000 bps communication speed
- ✓ Full feedback on position, speed, load, voltage and temperature
- ✓ 300 degrees of movement in precise 1024 increments
- ✓ Full control over torque and speed in 1024 increments
- ✓ Servo movement range can be set by the user

Multi Sensor Gripper

Engineered for cameras, pressure sensors, Ultrasonics, and IR detection.

Optional "2 in 1" Gripper

Great for grabbing large objects!

Great for grabbing objects of all sizes!!

See our web site for pictures, videos and more!

www.CrustCrawler.com

Or call us at:

480-577-5557

Serial-to-Ethernet Converter

Available Now!
New 2008 Lower Pricing

PoE SitePlayer Telnet System
- $99.95

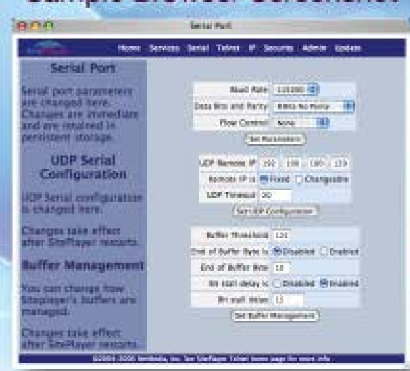
SitePlayer Telnet System
- $79.95

SitePlayer Telnet Modules
- $24.95

SitePlayer™ TELNET

SitePlayer Telnet Module

Sample Browser Screenshot



PoE Power Over Ethernet

- Configure and upgrade through web browser
- Access micro-controller or other RS232 serial device over The Internet
- Full RS232 handshaking
- 50 to 11500 baud data rates
- DHCP, Link Local, and static IP Addresses
- TCP/IP and UDP data transfers

siteplayer.com

NetMedia Inc. Tucson AZ 85737 t.520.544.4567 f.520.544.0800 netmedia.com

PROPELLER QUADROVER

The newest Parallax robot is coming your way! This giant gas-powered robot sports a 4-stroke 49 cc engine and hydraulic power train. It is controlled by a Propeller chip for the ultimate in customizable robots; eight 32-bit cogs at 20 MIPS each create endless programming possibilities. A 64 KB EEPROM leaves 32 KB for nonvolatile data storage, and ample expansion ports provide plenty of flexibility for added sensors.

Four solenoid valves allow for the hydraulic power to be independently enabled, disabled, or reversed for either side of its skid steer system. The robot can rotate in place or make traditional turns while traveling up to 10 miles per hour. Servo controlled throttle and disk brakes make for precise acceleration and deceleration even with a powerful 2.5 horsepower engine.

The Propeller QuadRover robot ships fully assembled and ready to program. The electronics include a Propeller-powered control board with connections for the included GPS, compass, and 3-axis accelerometer sensors. All you have to do is program, fill 'er up, and go. Measures approximately 29 x 19 x 21 inches. Robot weighs 89 lbs. (shipped weight 110 lbs.).

Propeller QuadRover (#28195; $3995.95)
Order now and receive FREE SHIPPING*! That's a $150 value! This offer expires 06/02/08.

*Free shipping offer valid for addresses in the contiguous United States only. Robots will ship via UPS Ground.

To order the **Propeller QuadRover** online visit www.parallax.com. Or call our Sales Department toll-free 888-512-1024 (Mon-Fri 7 a.m. - 5 p.m., PST).

Propeller, QuadRover, Parallax and the Parallax logo are trademarks of Parallax Inc.

PARALLAX INC
www.parallax.com