ARTBOTS • ROBOT SNAKES • CHART RECORDER BOTS

SERVO MAGAZINE

FOR THE ROBOT EXPERIMENTER

www.servomagazine.com

APRIL 2007

Get On The *Dean's List* Do Your Homework *FIRST!*

• Build The BioCrab

• Failsafe Design and Implementation For RC Robots

• A Simple Inertial Navigator

U.S. $5.50 CANADA $7.00

04>

0 74470 58285 4

JAMECO® ROBOT STORE

Perfect projects for kids of all ages!

robotic kits
chassis
servos
passives
optos
integrated circuits
semiconductors
muscle wires
connectors
motors
test equipment
software
books

Robotic Kits *and* Components...

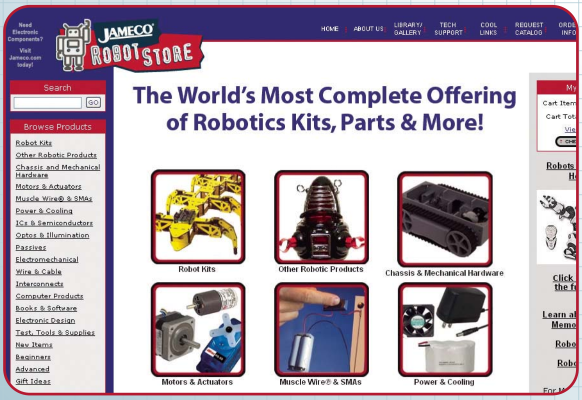
...there's something for everyone!

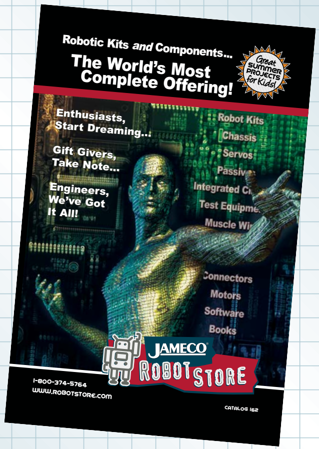
Robotic kits help you and your child to experience and learn about perception and control using a variety of sensors and actuators. Challenge yourself with Jameco's selection of fun and interactive kits! You or your child can assemble the kits and then enjoy endless hours of discovery.

Check out our unique selection of robotic kits at **www.RobotStore.com!**

- **Robot Insects & Animals**
- **Solar Robots**
- **Listening, Touching & Seeing Robots**
- **Hackable Robots**
- **Programmable Robots**
- **Educational Kits**
- **Legged and Wheeled Platforms**
- **OctoBot Survivor Kit**

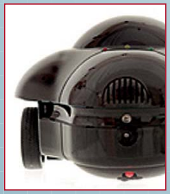
At Jameco's RobotStore you can get the world's most complete robotic offering—*all in one place!* You'll find kits for all ages and experience levels, along with gear boxes, servos and chassis, for those who are ready to take on the extreme.

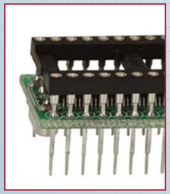


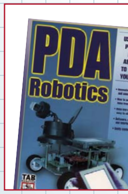


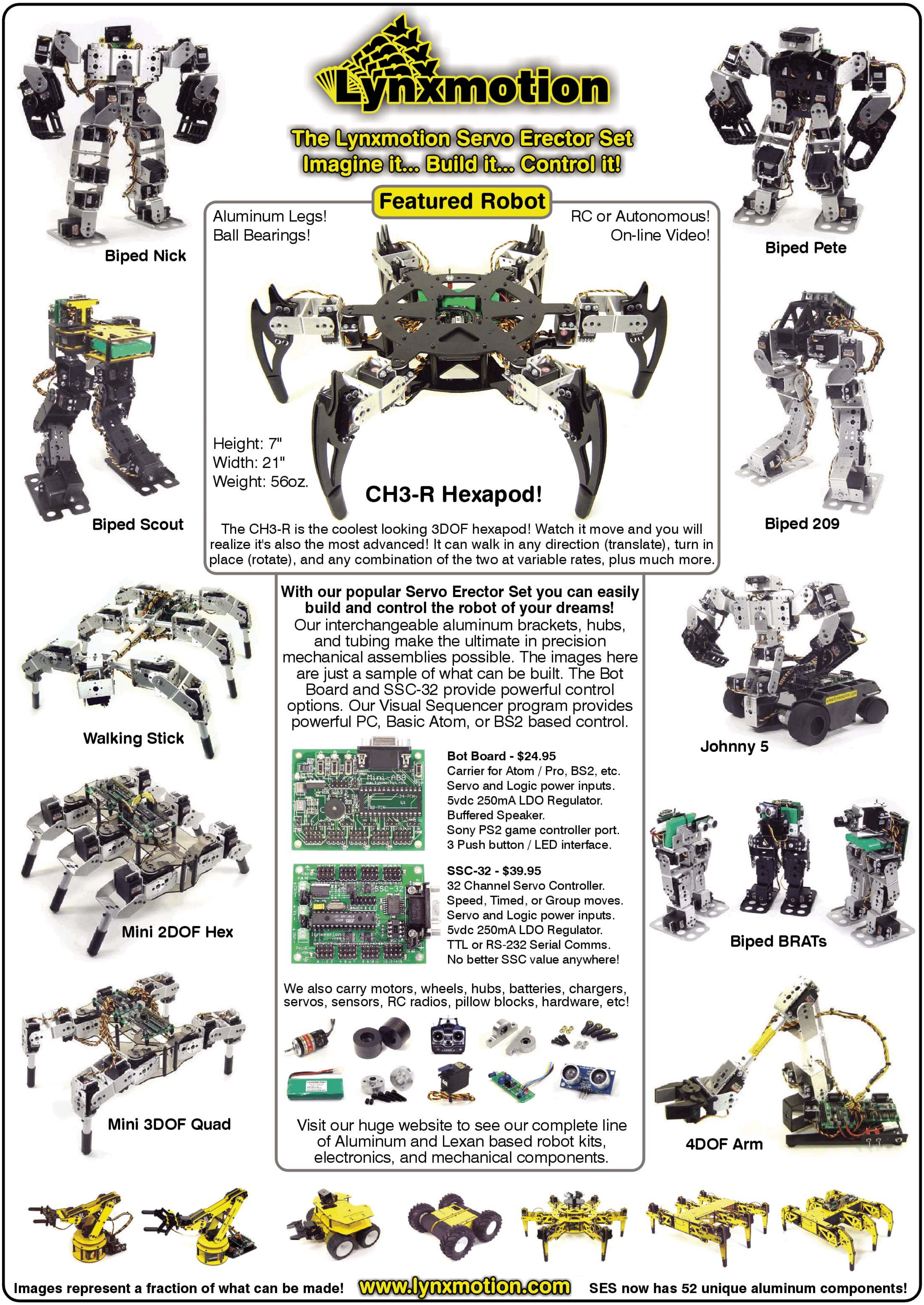
WWW.ROBOTSTORE.COM/SRD
1-800-374-5764

Call for your free catalog today!





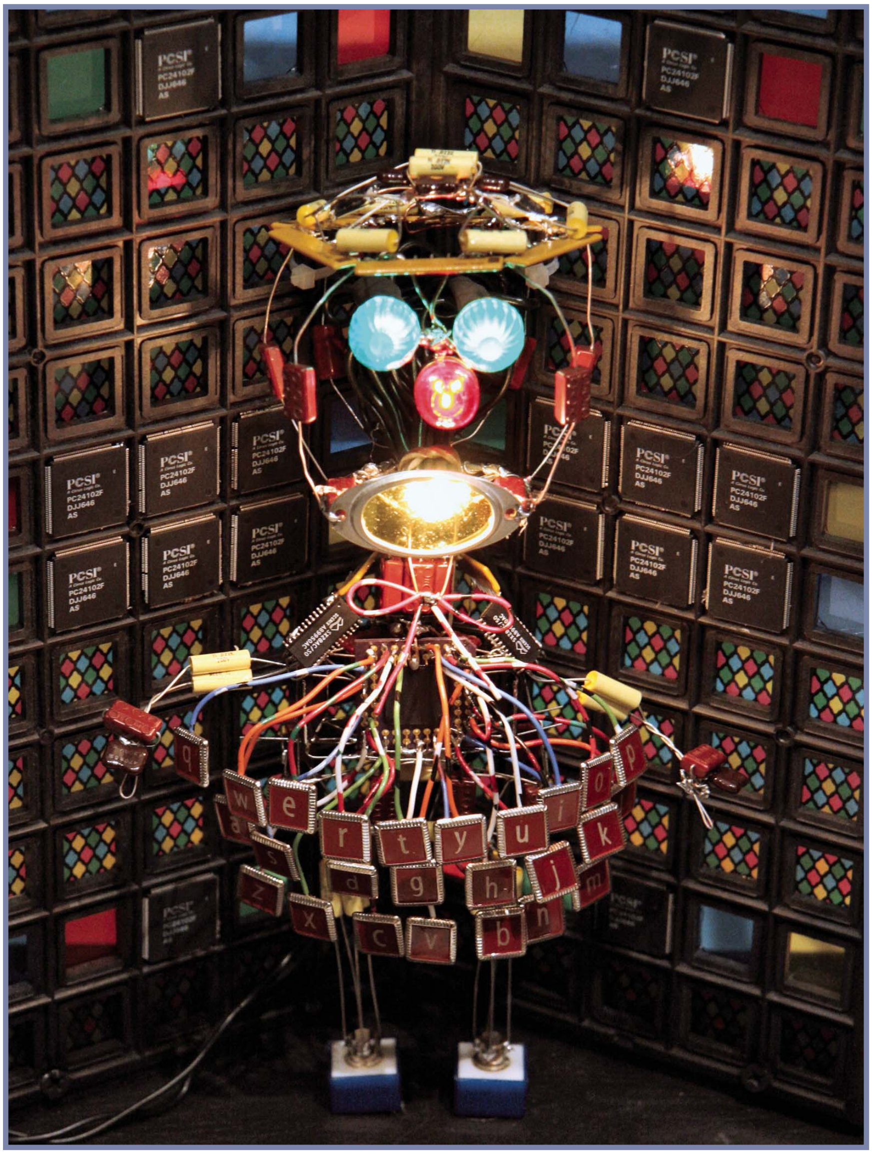
Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
Featured Robot
Aluminum Legs!
Ball Bearings!
RC or Autonomous!
On-line Video!
Height: 7"
Width: 21"
Weight: 56oz.
CH3-R Hexapod!
The CH3-R is the coolest looking 3DOF hexapod! Watch it move and you will realize it's also the most advanced! It can walk in any direction (translate), turn in place (rotate), and any combination of the two at variable rates, plus much more.
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 and SSC-32 provide powerful control options. Our Visual Sequencer program provides powerful PC, Basic Atom, or BS2 based control.
Walking Stick
Johnny 5
Bot Board - $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
4DOF 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 52 unique aluminum components!

SERVO MAGAZINE

Features & Projects

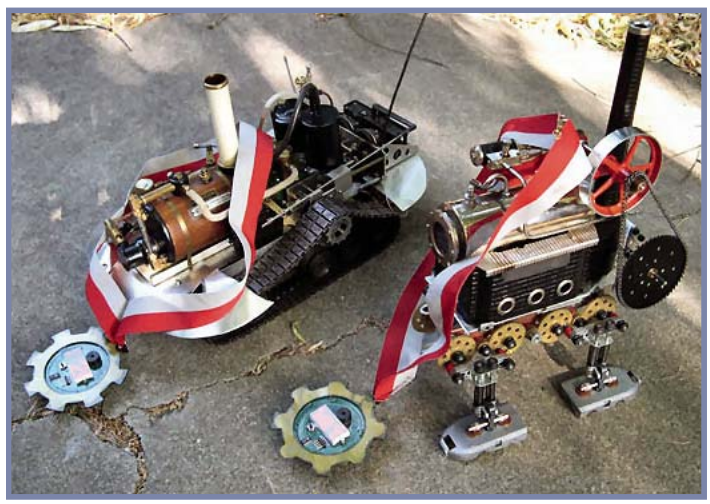
Columns

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



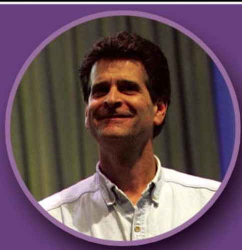
04.2007
VOL. 5 NO. 4

Departments



ENTER WITH CAUTION!

Mind / Iron



by Dean Kamen

Every January, thousands of FIRST team members across the US and around the world, attend or view a webcast of the Kick-Off of the FIRST Robotics Competition. This year, as in the past, it was my pleasure to be part of the event as we revealed the new game, presented the kit of parts, and began another exciting season of intense planning, designing, and building.

There is another annual Kick-Off tradition with which the general public may not be familiar, but the FIRST family has come to expect — the part where I give my "homework assignment" to all the teams. In the past, the homework assignments have been consistently at a "retail" level: more teams, more sponsors, more community involvement — let's all work to grow FIRST. Well, it worked. FIRST has grown from fewer than 100 teams and one regional competition, to more than 1,000 teams and more than three dozen regional competitions. In fact, we have grown enough that it's time for a new kind of homework assignment — an assignment that will help us leave the "prototype" stage of our history, and move to a full blown "wholesale" production version of FIRST.

The new homework assignment: Every FIRST participant — students, mentors, and sponsors — should write to their senators, congressmen, and governors, to tell them about FIRST, and invite them to a Regional Competition and/or to the National Championship at the Georgia Dome in Atlanta, April 13th and 14th.

Our government officials recognize that there is a crisis in science and technology education, and it's time to help them recognize that FIRST is an ideal solution to that crisis. FIRST is ready to emerge from being a grass roots, experimental program to a truly expansive, accessible part of our culture. FIRST will inspire students from any background, from any type of school or any geographical location, to pursue careers in science and technology.

We have proven that FIRST is scalable — more than 130,000 students participated in FIRST programs this year. We have programs that reach students of all ages: FIRST Robotics and the Vex Challenge for high school students; FIRST LEGO League for 9 to 14 year olds; and Junior FIRST LEGO League for 6 to 9 year olds. We have also proven that FIRST has impact: Recently, Brandeis University's Center for Youth and Communities conducted an independent, retrospective survey of FIRST Robotics Competition participants and compared results to a group of non-FIRST students with similar backgrounds and academic experiences, including math and science. When compared with the comparison group, FIRST students are:

- More than three times as likely to major specifically in engineering.
- Roughly 10 times as likely to have had an apprenticeship, internship, or co-op job in their freshman year.
- Significantly more likely to expect to achieve a post-graduate degree.
- More than twice as likely to expect to pursue a career in science and technology.
- Nearly four times as likely to expect to pursue a career specifically in engineering.

Mind/Iron Continued →

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

Subscriptions
Inside US **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

CONTRIBUTING EDITORS
Jeff Eckert, Tom Carroll, Gordon McComb, David Geer, Pete Miles, R. Steven Rainwater, Monty Reed, Bryan Bergeron, David Calkins, Kevin Berry, Jay Johnson, Valerie Johnson, Pete Smith, Robin Hewitt, Michael Simpson, Christopher Gilleski, Schlomo Engelberg, Shachar Braver, David Avikasis, Robert Jacoby, Jeffrey Scholz, Steven Kirk Nelson, James Isom, Lester Davis, Dean Kamen

CIRCULATION DIRECTOR
Tracy Kerley
subscribe@servomagazine.com

WEB CONTENT/STORE
Michael Kaudze
sales@servomagazine.com

PRODUCTION/GRAPHICS
Shannon Lemieux
Michele Durant

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

Copyright 2007 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.**

I'd like to take this opportunity to let all the FIRST participants know that I am doing my homework, too. For example, I recently attended the National Governors Association Meeting, and I'm happy to report that Rhode Island Governor Donald Carcieri, New Hampshire Governor John Lynch, Michigan Governor Jennifer Granholm, and Hawaii Governor Linda Lingle have all made specific personal commitments of their time and resources to growing FIRST programs as soon as possible in their home states. Many other governors also indicated a significant level of enthusiasm for FIRST.

As I write this article, we are about halfway through the FRC Season. I am optimistic about the midterm results of the homework assignment, but we still have a few more weeks for all of us to contact our elected officials and do whatever it takes to get them to attend a FIRST event. We're confident that once they experience FIRST, they will become enthusiastic FIRST supporters. The following statements by a few governors and federal officials should give you confidence that if we can get their attention, we will get their support.

Governor Donald Carcieri, Rhode Island

"I am proud to make FIRST an important part of our plan for educating Rhode Island's next generation of science, technology, and engineering leaders. I commend you for creating a program that so clearly inspires young people to discover the excitement and rewards of science, technology, and engineering. Your work is an important step towards building a society that more deeply values a culture of invention and innovation."

Senator Michael Enzi, Wyoming

"I have attended both the regional competitions in Denver, CO, and the national competition in Atlanta, GA, for FIRST and was very impressed. The young participants do a spectacular job. The skills learned at FIRST will ensure that the United States remains competitive and a leader of innovation in the future."

Congressman James Langevin, Rhode Island

"FIRST encourages kids to go into science, technology, engineering, and math, and allows them to explore their gifts, talents, and dreams. FIRST creates programs where its participants are limited only by the bounds of their imagination and gives them a forum to begin to examine the world's problems by exploring science, technology, engineering, and math. It is my goal to have a FIRST program in every school in Rhode Island."

Governor Linda Lingle, Hawaii

"FIRST and its mentors and teachers are helping the world to be a better place by helping develop our global future leaders. FIRST delivers both hope and heroes for tomorrow who will help cure diseases, create clean drinking water, grow up happy and strong, and give back to our world."

Governor John Lynch, New Hampshire

"While FIRST is about science, math, technology, and engineering, it is also about innovation. It is about learning to dream and daring to invent that dream and make that dream come true. The challenges of the future demand these skills."

As you read this article, please redouble your efforts to inform your representatives and community leaders about FIRST so that we can create a majority of state and national leaders that will assure that FIRST, and its programs, are available to all students in all of our schools. **SV**

To get all the details, event schedules, and ways to get involved in FIRST, go to **www.usfirst.org**

Industrial Strength Motor Control for All

Get the DC Motor Controllers that are at the heart of many of today's world's most demanding Industrial, Military and Research Robots, and many other innovative Applications.

- RS232, RC, or Analog input
- Speed or Position Mode
- Dual channel output up to 140A
- Optional single channel at double Amps
- Optical Encoder Inputs
- Intelligent Current Limiting
- Thermally Protected
- Field Upgradable Firmware
- Enclosed and Board-Level versions
- and many more advanced features ...

Model	Amps	Features	Price
AX500	2x15A	B	$145
AX1500	2x30A	B	$275
AX3500	2x60A	O-R-B	$395
AX2550	2x120A	A	$495
AX2550HE	2x140A	A	$645
AX2850	2x120A	O-A	$620
AX2850HE	2x140A	O-A	$770

A=Aluminum Extrusion, B=Board-Level, O=Optical Encoder In, R= RC outputs. Qty1 price. Contact us for OEM Qty prices

RoboteQ

8180 E.Del Plomo Dr.
Scottsdale AZ USA 85258
(602) 617-3931 - info@roboteq.com

www.roboteq.com

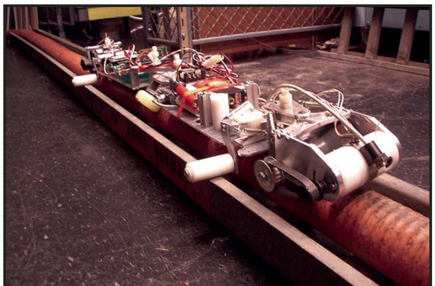
Robytes

by Jeff Eckert

Are you an avid Internet surfer who came across something cool that we all need to see? Are you on an interesting R&D group and want to share what you're developing? Then send me an email! To submit related press releases and news items, please visit www.jkeckert.com

— Jeff Eckert

Protecting the Power Grid



This prototype wire crawler can inspect miles of power cable autonomously. Photo courtesy of the University of Washington.

It's not sleek or pretty, it doesn't play music, and it can't mop your floor. But an engineering professor at the University of Washington (**www.washington.edu**) has created "the first robot built that can inspect power cables, autonomously looking for incipient failures," and it's already proving itself to be pretty useful.

Late last year, Professor Alexander Mamishev ran the first field test of the bot in New Orleans, chosen because there is still widespread Katrina damage to the city's power system. The test took place at Lockheed Martin's Michoud NASA Assembly Facility where it monitored the high-voltage lines that carry electricity from the distribution plant to substations. The UW robot pinpoints problem spots by analyzing the cable's surface and assessing what's underneath.

It uses three techniques: a heat sensor that detects heat dissipation; an acoustic sensor that listens for partial electrical discharge; and a special sensor that identifies "water trees," which are water filaments that have seeped into the insulation. The process is monitored via a wireless link, and engineers can check out the robot's environment via a video camera.

Riding along the cable, it can travel for miles and even negotiate tight turns and underground passages. No major damage to the Michoud lines was reported, but the successful test should help stimulate interest in building more of the units for use in the power industry, where sending technicians out with handheld instruments is time-consuming and expensive.

CMG-Driven Arms Going Zero-G

DARPA's SUMO spacecraft shown with two CMG-driven robotic arms. Photo courtesy of Cornell University.

An interesting collaboration has been announced in which some students from Cornell University's CMG Research Project (**http://mae.cornell.edu/cmg/**) will be testing a special type of robotic arm during a week of flights in NASA's zero-gravity aircraft. Cornell has been researching the use of control-moment gyroscopes (CMGs) as robotic arm actuators, which could save 90 to 99 percent of the power normally required for assembling and manipulating things in space, and NASA has offered them the opportunity to conduct a test of a two-link mechanism with four gyroscopic actuators. (A CMG is made up of a spinning rotor and gimbals that change the rotor's angle of momentum, thereby creating gyroscopic torque that can be used, in this case, to move the arm.)

Cornell built the first prototype arm last May and will complete the second one by the same time this year. Because simulations of the arm's effectiveness can be validated only in a microgravity environment, the upcoming test is particularly important. Several teams have been granted access to the Boeing KC-135 aircraft as part of NASA's Microgravity University program, details of which are available at **http://microgravityuniversity.jsc.nasa.gov/**.

Challenge Offers Big Bucks

Crossing the Urban Challenge finish line first can get you $2 million. Photo courtesy of DARPA.

If you missed the December 13 deadline to enter the Defense Advanced Research Projects Agency (DARPA) 2007 Urban Challenge, you may regret it. Just before the deadline, it was announced that the first, second, and third teams to cross the finish line will receive, respectively, $2 million, $1 million, and $500,000 awards. The goal of the program is to develop unmanned ground vehicles that can safely operate and maneuver in the presence of other moving vehicles and essentially revolutionize military ground operations.

Participants will need to demonstrate such operations in a mock urban area, executing simulated military supply missions while performing a range of functions. This is no simple feat, however. The first Grand Challenge event was held in 2004 over a 142-mile desert course, and none of the 15 participants completed it. In 2005, however, four vehicles made it through a 132-mile route within the 10-hr time limit, and DARPA forked over $2 million to a team from Stanford University.

This year's event will be held on November 3 at a location to be disclosed sometime before the National Qualification Event in October. For details, visit **www.darpa.mil/grand challenge**.

Nominations, Please

A gold medal and $250,000 await the next Bower Science Award Winner. Photo courtesy of the Franklin Institute.

Even if you missed out on the DARPA contest, you still have until May 31 to nominate yourself (or someone else) for the Franklin Institute's 2008 Bower Science Award, which will be awarded to "an individual who has played a seminal role in either the design and construction of robotic systems or the advancement of enabling technologies as related to robotics such as mechanical structure, sensors, and control algorithms." The award is available to any living individual, regardless of nationality, and includes a gold medal and $250,000. For details, check out **www.fi.edu/tfi/exhibits/bower**.

Digital Drive Introduced

The ION Digital Drive provides motion control for various motors. Photo courtesy of Performance Motion Devices, Inc.

If you haven't raked in a fortune lately, you might be able to afford to put the ION™ Digital Drive from Performance Motion Devices (**www.pmdcorp.com**) into your next project. It's a distributed-control module that offers networking, positioning control, and power amplification in one package.

The single-axis module, based on the Magellan Motion Processor, is available for DC brush and brushless motors and microsteppers, and is aimed at a range of robotic and other applications. You need one for each axis of motion desired, and up to 127 of them can be linked on a single network. Peak output is 15A, and 500W at 56V. Programming is achieved via your PC. Prices start at $223 in OEM quantities.

Pool Bot Introduced

This month's product for people who have entirely too much money is the Aquabot Turbo T4RC robotic pool cleaner from Aqua Products (**www.aquaproducts.com**). According to the manufacturer, it cleans everything from

The Aquabot Turbo T4RC will clean most pools in one hour. Photo courtesy of Aqua Products, Inc.

the waterline to the walls and floors, and it can handle pools up to 50 feet long, any shape or surface, in about an hour.

The unit is driven autonomously by two microprocessor-guided drive motors, but it also comes with a handheld remote if you want to take control. It microfilters up to 5,000 gallons per hour. The catch is that it lists for $1,900 (but as of this writing can be had for $1,799). If that's too hefty for your wallet, you can scale back to the Pool Rover Junior for a mere $349.

Bot Engineering Degree Offered

Citing a growing need for advanced robotics in defense, health care, and consumer products, Worcester Polytechnic Institute (WPI) has announced plans to offer the nation's first bachelor's degree program in robotics engineering beginning next fall. The program will be offered jointly by the Computer Science, Electrical, and Computer Engineering, and Mechanical Engineering departments, and students will receive a firm grounding in the fundamentals of three fields. For complete information, see **www.wpi.edu/Academics/Majors/RBE/** or drop by if you happen to be in Worcester, MA. SV

GEERHEAD

by David Geer

Contact the author at geercom@alltel.net

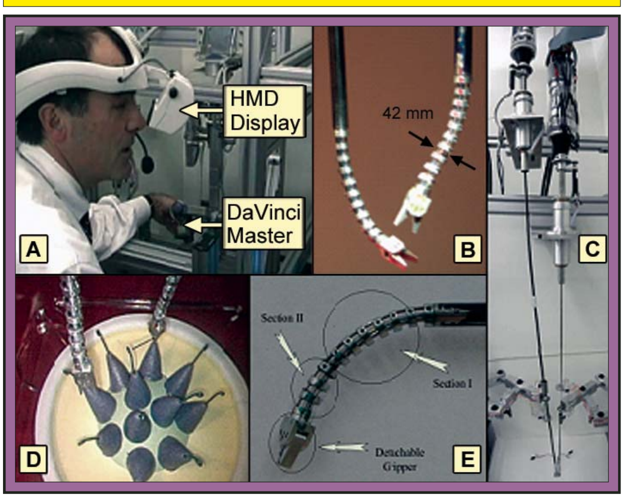
Robot Snakes "Side-Wind" to a Perfect Cut

Twin robotic snakes' pincer-like grippers are now an invaluable extension for surgeons. These snakes reach into the quadrants of the human body on which the physician is operating, extending the precision of the surgeon during "minimally invasive" surgery. The prototype is used only in the throat and upper air passage.

A surgeon manipulates hand controllers/manipulators from a daVinci surgical robot, of which the snakes are an extension, and the snakes mimic what the surgeon would do, adding several benefits not existent in human hands and minds, through algorithms and instilled hardware capacities. The interaction between the surgeon and the snakes is teleoperated from some distance while the physician observes the operation via a stereo, head-mounted computer display in 3D.

The snake surgical arms each possess nine degrees of freedom (DOF) consisting of a four degree-of-freedom "tool manipulation unit" that can, in turn, place their attached four-degree-of-freedom wrists and grippers where they need to be during a given surgical maneuver. The algorithm was developed at Johns Hopkins to optimize the interaction between the surgeon, the snakes, and the patient.

Prototype "snake" robot for throat surgery. (A) Dr. Paul Flint of the JHU otolaryngology department controlling the robot by manipulating a daVinci master control arm while observing the surgical scene through a head-mounted stereo display. (B) Close-up of four degree-of-freedom snakes and grippers. (C) Our current two-handed prototype. (D) Surgeon's view using a standard training phantom. (E) Detail of one snake, showing two stages and gripper. Photo courtesy of Dr. Russell Taylor.



All photos are courtesy of Will Kirk/Johns Hopkins University unless otherwise noted.

How to Control a Surgical Snake

In order to control these snake-like surgical arms, there is a need to write better control algorithms, according to Nabil Simaan, assistant professor, The Fu Foundation School of Engineering and Applied Sciences, Department of Mechanical Engineering, Columbia University, who designed the snakes.

"The problem is," explores Simaan, "that our snakes have some actuation redundancies, meaning that they are more accurate than they need to be, so the load between their very small backbones can be redistributed." This also enables researchers to figure out a way to sense the forces of the snake,

SURGICAL SNAKES THAT FEEL FOR THE DOCTOR

According to video from Johns Hopkins, the manipulators and arms connected to the daVinci intuitive surgical robot (reported elsewhere) enable the surgeon to feel what the snakes are doing as though he was operating with his own hands.

The snakes enable the doctor to work precisely in a confined space, such as the throat. Not only are the snakes accurate, they also enable the added safety precaution that, as with the daVinci surgical robot itself, the technology prevents the doctor from making mistakes by defining the space within which the tools may operate. Using the current control algorithm, the snake robotic arms can also pass sutures and sew up patients.

The stereo head mounted display provides a three-dimensional view into the patient via two cameras looking down inside the patient's throat. Each camera brings back an image to each eye of the physician.

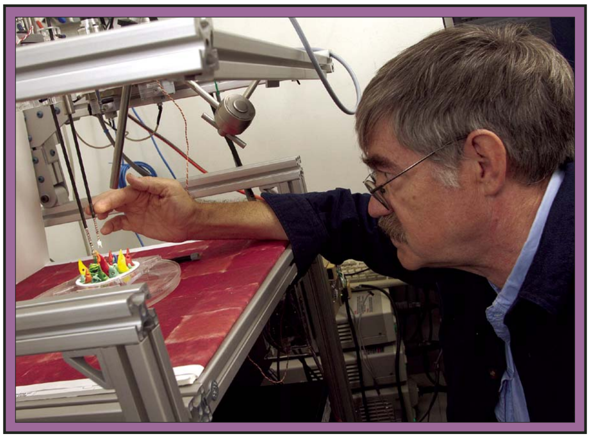
according to Simaan. This is ongoing research for Simaan and his staff at Columbia, though clearly the original control algorithms allow for a certain level of control, enabling the snakes' current capabilities.

The snakes are also very flexible, too flexible for their own good. There is the possibility of deflection when a force is applied to the snakes.

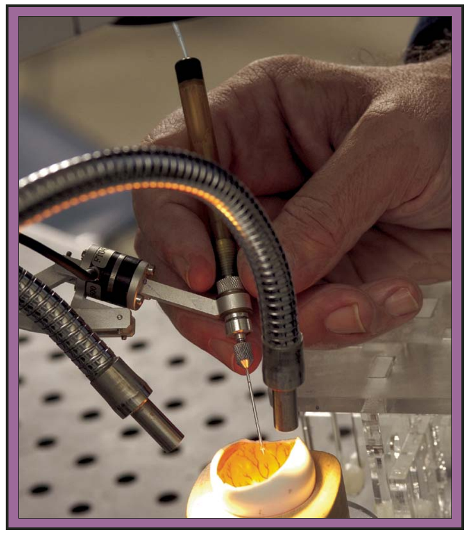
"There are two ways of compensating for that," notes Simaan, "either [by using] a very accurate model of the flexibility for the device, or by having a very smart control algorithm that allows one to compensate for the flexibility." These are also areas of current research for Simaan.

The pros behind the snakes' construction are that, through the lack of any gears or pulleys, they are very easy to make — consisting of only a bunch of flexible rods connected together inside a housing/shaft — with few mechanical joints.

But, there is a much more important attribute: "The advantage is eliminating backlash from the system," notes Simaan. In a traditional design, he explains, one has gears and wires that normally have much more backlash than in these snakes.



Dr. Russell Taylor of Johns Hopkins working with the surgical arms. Note how small they are.



Surgical snake aparatus as equipped (potentially) for eye surgery.

How These Snakes Measure Up

The snakes are made of nickel-titanium tubes and wires — nickel-titanium is highly elastic — that are both the actuators and the structure of the arms. Their dimensions are just over 4 mm in diameter and about 40 mm in length.

"Super elastic nickel titanium [backbones were] very carefully designed and selected to support a certain threshold of maximum compression force that would allow one to do the job without buckling or damage to the backbone," says Simaan.

The two-stage throat surgical snakes begin with four nickel-titanium tubes about .6 mm in diameter. These four tubes pass through aluminum disks that space them apart. There is also a primary "backbone" tube.

There are three secondary backbone tubes that offer two degrees of freedom via push-pull actuation. The next stage employs three .4 mm nickel titanium wires inside the secondary backbones, which attach to a small hand via an aluminum backbone and finally to the grippers.

This construction and actuation method offers stronger force actuation than mechanical pivots and joints and is scalable. According to Simaan, the hollow backbones allow for multiple modes of operation like surgical tool manipulation, aspiration and suction, and imaging and light delivery.

Goals

"The aim of the project was to define a general architecture for a snake-like device that would support several applications, meaning that [the snakes] could manipulate a surgical instrument, could provide drug delivery, aspiration, light delivery, and some laser treatment or imaging. So the goal was a simple design that could be miniaturized," says Simaan.

Simaan is currently working on smaller versions of the snakes with diameters of .5 mm; these are for eye surgery. Diameters of less than .2 to .3 mm are necessary for neurosurgery, he explains. "One of the things proposed is a brain snake," says Dr. Russell Taylor of Johns Hopkins. "The goal of the brain snake," he explains, "is to build a [...] snake robot that will go through a neural endoscope to give the surgeon working way down inside a brain more dexterity in trying to say, take some

"STICK A NEEDLE IN YOUR EYE"

In his video about the snake-like robot arms, Dr. Russell Taylor, director, the National Science Foundation's Engineering Research Center for Computer-Integrated Surgical Systems and Technology, (NSF ERC CISST), Johns Hopkins, theorizes that if the snakes could be equipped with a needle small enough to pierce a blood vessel in the eyeball, an injection could be used to dissolve a blood clot. This would be practicable because other methods of clot removal could cause loss of sight. This is a project in research now, which will be facilitated through even smaller snake-like robotic arms.

The challenges to this research include the fact that the surgeon's actual hands have tremors when working at that scale, according to Dr. Taylor. The surgeon also has a limited ability to feel at this scale, at which point their vision is also limited.

Dr. Taylor believes that each of these human limitations can be augmented by developing robots with hands that don't shake, a sense of touch, and improved haptic feedback.

RESOURCES

Surgical robotics video, snake-like surgical arms
http://mfile.akamai.com/7111/wmv/www.jhu.edu/news_info/realmedia/surgicalrobotics.wmv

Medical robotics news, Johns Hopkins
www.jhu.edu/news/audio-video/medical_robotics.html

Nabil Simaan's site (he engineered the robotic arms)
www.columbia.edu/cu/mece/arma

ERC CISST (NSF supports this research)
www.cisst.org

NIH also supports this research
www.nih.gov

Russell Taylor's site
www.cs.jhu.edu/~rht/

Johns Hopkins School of Medicine
www.hopkinsmedicine.org/som/index.html

pressure off a blood vessel or clean away a tumor without doing some damage that would hurt the patient."

"To use this kind of system effectively in a clinical application," notes Dr. Taylor, "requires integration of the software with the hardware. So, the computer accepts inputs from the human sur-

Two surgical snake-like arms and grippers working with various sizes and shapes of objects. Specifically, as in the video, the snakes can pick up rings on one pointed object and place them on another.

geon that tell the computer what the surgeon wants the hand of the robot to do."

Next, the computer algorithms must translate the surgeon's decisions and movements into servo commands that the snakes' motor controllers can properly interpret.

Dr. Taylor and colleagues have developed a way of doing this that involves solving a mathematical optimization problem every 30 milliseconds, according to Taylor, who identified this capability as attending to all the redundancies of the robot, as well.

In their work, Dr. Taylor and colleagues are integrating current and future surgical robots with high-level software that enables the combination of inputs from the surgeon with patient information. This, in turn, improves the ability of the surgeon to perform as desired while ensuring greater safety and efficiency.

An example of a built-in safety precaution is the "no fly zone," alluded to earlier. "Having made a computer representation of some part of the patient's anatomy, from the images, [one might program in information that], there is the aorta, please keep the knife away from there unless the surgeon overrides it," says Dr. Taylor.

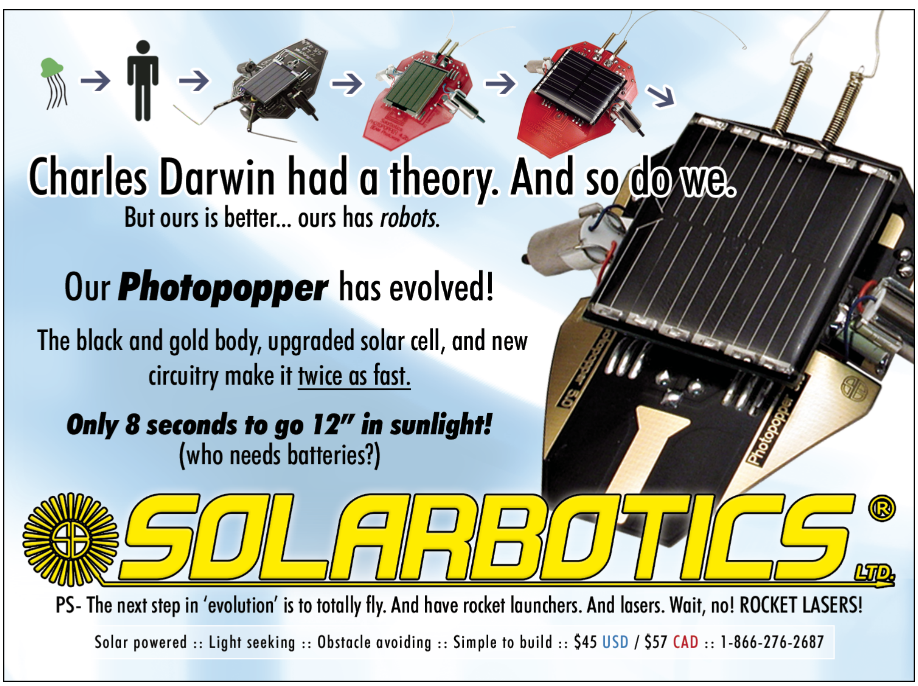
Anecdotes

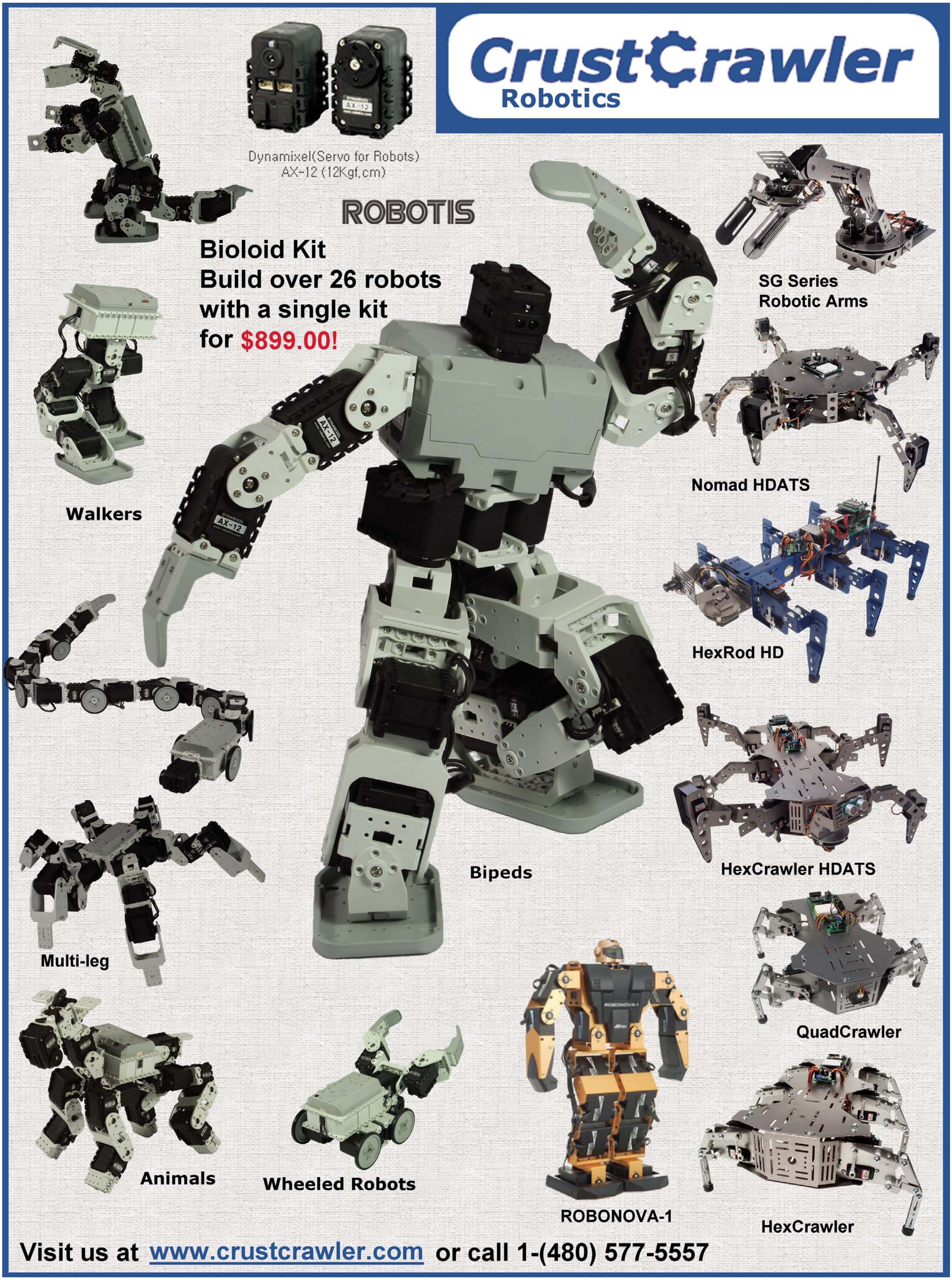
Dr. Taylor, recalling the satisfaction he and his colleagues have reaped from the snake surgical arm work, notes the strong interest from clinicians who see what they are working on and express high interest and enthusiasm toward potential uses for the snakes' surgical dexterity in their own operating rooms.

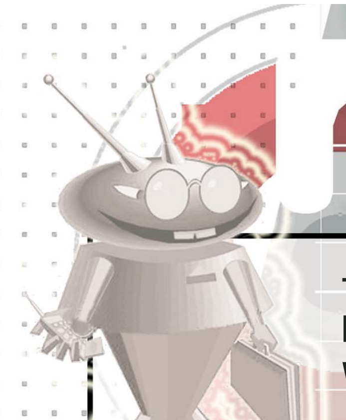
Another source of satisfaction has been the robust nature of the basic control formula, which enables the researchers to combine various surgical requirements from the high-level control hierarchy, the task levels, and the snakes' properties.

Ultimately, the snake is highly redundant and optimized to efficiently employ many degrees of freedom. The general challenge, according to Dr. Taylor, when trying to control high degree-of-freedom redundant systems is how does one exploit the redundancy to help improve the performance of the desired task without making the system very, very difficult to deal with or inflexible?

With the control framework they are working with, they have already demonstrated the ability to change an element of the snakes' behavior by changing the constraints or weightings on the objective function. "And it just works!" exclaims Dr. Taylor. **SV**




CrustCrawler
Robotics
Dynamixel(Servo for Robots)
AX-12 (12Kgf.cm)
ROBOTIS
Bioloid Kit
Build over 26 robots
with a single kit
for $899.00!
SG Series
Robotic Arms
Nomad HDATS
Walkers
HexRod HD
HexCrawler HDATS
Bipeds
Multi-leg
QuadCrawler
Animals
Wheeled Robots
ROBONOVA-1
HexCrawler
Visit us at www.crustcrawler.com or call 1-(480) 577-5557



Our 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 Pete Miles

Q. I have a BASIC Stamp P40 and I would like to program it to play different ring tones, like the ones you hear on cell phones. Can this be done? Is there an easy way to do this without having to buy expensive parts? Any help would be appreciated.

— Kendra Allison

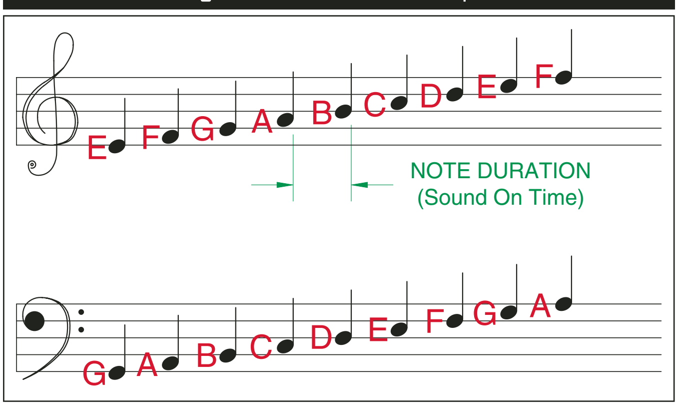
A. Ring tones on a BASIC Stamp? Now that sounds like a fun project! After doing a bit of research, I found that this really isn't that hard if you can read sheet music. All you have to do is use the BASIC Stamp's FREQOUT command to play each music note. The trick is getting the correct frequency for each note, and playing them for the correct amount of time.

Figure 1 shows a simple Treble Clef and Bass Clef music staff. For those people that are not familiar with reading sheet music, I have labeled some of the basic music notes: A, B, C, D, E, F, and G. In order to hear a note, it must vibrate the air at a unique frequency, and each labeled musical note corresponds to a different sound frequency.

To make things a little bit more interesting, each labeled note (i.e., C) has been assigned to several different frequencies, known as octaves. Most people really don't care about what the actual frequency sound each note makes, they only care about the sound the piano, flute, or guitar makes when a particular note is played. On the other hand, a microcontroller outputs a signal at a preprogrammed frequency, which is interpreted as a sound when played through a speaker. In order for a microcontroller to play music notes, each note needs to be mapped to a specific sound frequency.

Table 1 shows the different sound frequencies for each music note for the first six octaves. Each octave has 12 different notes. Not every note has a sharp (#) or flat (b) sound. The frequency for each note has a simple mathematical relationship to each other. The equation that follows shows this relationship. Here, f_o is the base frequency, f_n is the frequency of the note which is n note positions away from the base frequency. n is positive for frequencies greater than the base frequency, and is negative for all the frequencies below the base frequency.

Figure 1. Sheet music example.



Take for example, a base frequency, f_o, of 440 Hz which is the A note in the fourth octave, or commonly written as A4. The frequency for B4 (the B note in the fourth octave) is 493.88 Hz since it is n = +2 notes higher than A4. A note that is 20 positions below the A4 note (n = -20) is 138.59 Hz, or C3#.

Remember that the octave changes every 12th note. By definition (a strange one, that is), the frequency of the note A above middle C is defined to be equal to 440.00 Hz. Middle C is in the fourth octave. Most musical instruments play in the third, fourth, and fifth octaves.

$$f_n = f_o\left(2^{1/12}\right)^n$$

With this information, a table can be made to map the music notes to specific sound frequencies. Table 1 was generated using the Excel spreadsheet program and the frequency equation shown earlier.

FREQOUT *Pin, Duration, Freq1 {, Freq2}*

The output frequency of the BASIC Stamp's FREQOUT command is based on a number, *Freq1*, that ranges from 0 to 32767. Due to the internal timing differences for the different Stamp modules, the actual frequency that is heard through a speaker will be different. Table 2 shows the different frequency units for each of the BASIC Stamp modules. The actual frequency that is heard through a speaker will be the unit value for *Freq1* (from Table 2) multiplied by the *Freq1* number that is used in the FREQOUT command. For example, if a number 1000 is programmed into the FREQOUT function, the actual frequency heard through a speaker would be 1,000 Hz on a BASIC Stamp 2, 3,770 Hz on a BASIC Stamp P40, or 6,030 Hz on a BASIC Stamp PX.

Now, in order to play the notes shown in Table 1, the frequencies will need to be converted into the *Freq1* units for the Stamp's FREQOUT command. This is done by dividing the note's frequency by the *Freq1* unit for the particular Stamp module. For example, to get the BASIC Stamp P40 to play the A4 note, 440 Hz is divided by 3.77 Hz to get 116.7. Since BASIC Stamps only work with integers, this result is rounded to the nearest integer, in this case, 117. Thus, 117 would be used in the FREQOUT command to output a true 440 Hz sound.

Notes	1	2	3	4	5	6
C	32.70	65.41	130.81	261.63	523.25	1046.50
C#/Db	34.65	69.30	138.51	277.18	554.37	1108.73
D	36.71	73.42	146.83	293.66	587.33	1174.66
D#/Eb	38.89	77.78	155.56	311.13	622.25	1244.51
E	41.20	82.41	164.81	329.63	659.26	1318.51
F	43.65	87.31	174.61	349.23	693.46	1396.91
F#/Gb	46.25	92.50	185.00	369.99	739.99	1479.98
G	49.00	98.00	196.00	392.00	783.99	1567.98
G#/Ab	51.91	103.83	207.65	415.30	830.61	1661.22
A	55.00	110.00	220.00	440.00	880.00	1760.00
A#/Bb	58.27	116.54	233.08	466.16	932.33	1864.66
B	61.74	123.47	246.94	493.88	987.77	1975.53

Table 1. Musical note sound frequencies in Hz for different octaves.

Now to play a ring tone, all you need to do is sequentially execute the FREQOUT command for each note in the ring tone for a specific amount of time. There are many ways to go about doing this. I like to use a lookup table since it provides a compact and easy way to record large amounts of data (well, 256 pieces of data for each lookup table.) One way that this could be done is to create a simple subroutine that contains all the ring tone data, and have a main routine that sequentially calls the subroutine to play each note. The Get_Note subroutine shown here contains two pieces of information for each note: the actual note itself and the playing duration. Since playing music is more about playing notes and not being concerned about the actual sound frequency, the lookup table is using variables for the notes. At the beginning of the ring tone program, each note's variable is assigned the actual frequency value.

```
Get_Note:       'Ring Tone Note and Duration Information
    LOOKUP idx,[F5,2,F5,2,F5,2,F5,2,F5,2,D5,4,F5,2,F5,4,
               F5,2,f5,2,D5,4,0,1,F5,2,F5,2,F5,2,F5,2,
               F5,2,D5,4,F5,2,F5,4,F5,2,f5,2,D5,4,0,1,
               F5,2,F5,2,F4,2,F4,2,0,2,F5,2,F5,2,F4,2,
               F4,2,0,1], value
RETURN
```

The two elements for the first note in Get_Note are F5 and 2. Here, the F5 (Note F, fifth Octave — 698.46 Hz) is to be played for two units of time. The actual unit of time is arbitrary since it depends on the tempo that you want to play. The duration of each note with respect to each other is defined by the song. Generally, the duration of each note is an integer fraction of the song's beat (i.e., four notes per beat). For simplicity, I chose a time unit of one to be smallest note duration, and all the other notes are integer multiples of this small time increment.

The following loop will index through all 33 notes in the song. The *idx* value points to the note position and the tone duration position in the lookup table; the Get_Note subroutine is called twice. Once the note and duration are known, the results are played on a speaker using the FREQOUT command.

```
FOR i = 0 TO 33
   idx = 2*i
   GOSUB Get_Note            'Get the ring tone note
   note = value
   idx = 2*i+1
   GOSUB Get_Note            'Get the play duration
   duration = value * 1000   'Scale Duration to real time
   FREQOUT 0, duration, note 'Play the sound on Pin 0
NEXT
```

The program in Listing 1 will play a portion of a song we are all very familiar with. As you can see, there really isn't a lot to playing ring tones on a BASIC Stamp. Figure 2 shows a simple circuit for driving an eight-ohm speaker with a BASIC Stamp that I used to test this concept, which is the same circuit that is built into the Parallax Professional Development Board (**www.parallax.com**).

Q. Hello! I am an electronics teacher at the Orleans Career & Tech Center in Medina, NY. We have entered the Monroe Community College SUMOBOT competition for the past four years and have done very well.

	BS2 and BS2E	BS2SX	BS2P	BS2PE	BS2PX
Units in Duration	1 ms	0.4 ms	0.265 ms	1 ms	0.166 ms
Units in Freq1 and Freq2	1 Hz	2.5 Hz	3.77 Hz	1.51 Hz	6.03 Hz
Range of Frequency	0 to 32767 Hz	0 to 81.92 kHz	0 to 123.53 kHz	0 to 49.48 kHz	0 to 197.59 kHz

Table 2. Unit specifications for the FREQOUT command on a BASIC Stamp.

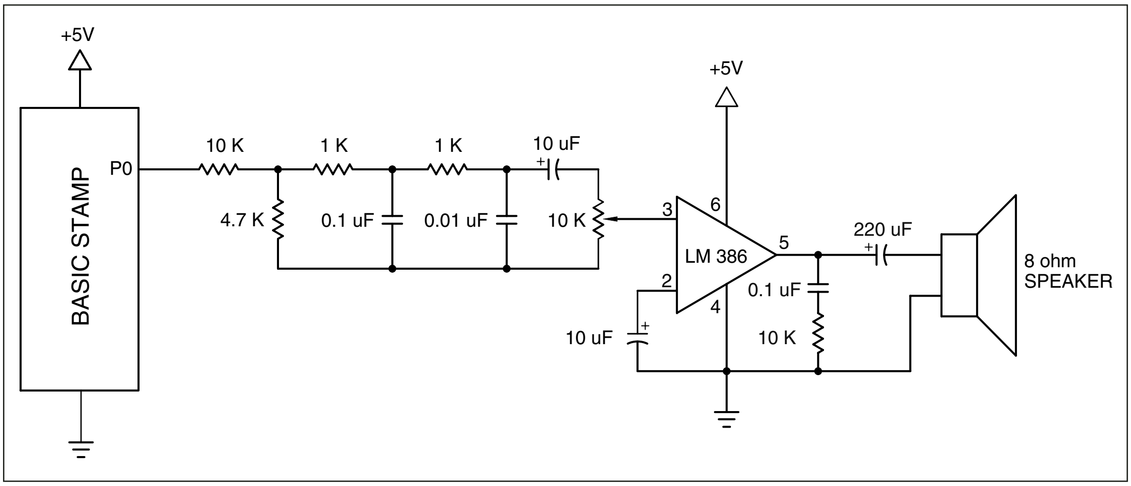


Figure 2. Simple schematic for playing sounds on a BASIC Stamp.

Listing 1. Program listing for demo ring tone.

```
' {$STAMP BS2p}
' {$PBASIC 2.5}

i        VAR Byte    'Counter Variable
j        VAR Byte    'Counter Variable
idx      VAR Word    'Pointer Variable
Note     VAR Word    'Sound Note
Duration VAR Word    'Time duration of Sound, in ms
value    VAR Word    'Temporary variable

E4   CON   54    '329.63 Hz, E-Note, 4th Octave
F4   CON   57    '349.23
F4S  CON   62    '369.99
G4   CON   65    '392.00
G4S  CON   68    '415.30
A4   CON   72    '440.00
A4S  CON   77    '466.16
B_4  CON   81    '493.88
C5   CON   86    '523.25
C5S  CON   91    '554.37
D5   CON   97    '587.33
D5S  CON  103    '622.25
E5   CON  109    '659.26
F5   CON  115    '698.46

main:
  FOR j = 1 TO 2                        'Repeat the song twice
    FOR i = 0 TO 33                     'Play each note
      idx = 2*i
      GOSUB Get_Note                    'Get ring tone note
      note = value
      idx = 2*i+1
      GOSUB Get_Note                    'Get the play duration
      duration = value * 1000           'Scale Duration to real time
      FREQOUT 0, duration, note         'Play the sound on Pin 0
    NEXT
  NEXT
  PAUSE 2000                            'Wait 2 seconds before repeating
GOTO main
END

Get_Note:       'Ring Tone Note and Duration Information
    LOOKUP idx,[F5,2,F5,2,F5,2,F5,2,F5,2,D5,4,F5,2,F5,4,
               F5,2,f5,2,D5,4,0,1,F5,2,F5,2,F5,2,F5,2,
               F5,2,D5,4,F5,2,F5,4,F5,2,f5,2,D5,4,0,1,
               F5,2,F5,2,F4,2,F4,2,0,2,F5,2,F5,2,F4,2,
               F4,2,0,1], value
RETURN
```

They have videos from the competition on their website at **www.monroecc.edu/depts/eng&phy/highschl.htm**.

I have run into a BRICK wall for parts! Specifically, compound gears that are Mod 1. That is to say that the number of teeth is equal to the diameter in millimeters +1 mm. I have used a gear that has a 40-to-10 tooth combination. They have an outer diameter of 41 mm; inner gear is 11 mm. I could use other combinations, but they must be Mod 1. By the way, LEGO gears and Kelvin ** gears P/N 990174 are Mod 1. My usual supplier is **NELNICK.COM**, but they don't answer the phone or emails and I fear they're out of business or at least out of town!! The website is **http://nelnick.com//nelnickrobotics/index.php?cPath=21&osCsid=d5f00cb4eedc88af3ebeb383ea78f5e4**.

So, here's the question ... Where can I find any Mod 1 compound gears? Or, better yet, how about molding my own? I have old gears or single gears I could make a mold from!

— Bill Leggett
Medina, NY

A. Thanks for the link to the sumo contest at Monroe Community College. I have not heard of this particular event, and congratulations

NAME THAT SONG!!

Can you identify the song that the program in Listing 1 is playing? Build the circuit, program the Stamp, or just play it on the piano. Name the song and the artist and email the answers to *Roboto@ServoMagazine.com* by June 1st. We will randomly choose one winner out of all the correct submissions. The winner will receive a one year subscription (or renewal) to *SERVO Magazine* and a *SERVO Magazine* t-shirt. Please include your postal address. Good Luck!!!

on your team's success.

This is a tough question. Compound gears are generally not a commercially off-the-shelf type of product because they are usually made for specialized applications. This is probably why you are having a hard time trying to find replacements.

First off, I need to make a small correction in the definition you provided for what a Mod (Module) 1 gear is. By definition, the pitch diameter, *D*, of a gear is the Module, *m*, multiplied by the number of teeth, *N*, on the gear ($D = mN$). The outside diameter, D_o, of the gear is equal to the pitch diameter, *D*, plus twice the Module, *m* ($D_o = D+2m$ or $D_o = m(N+2)$). For those people that have never heard of gear module before, it is the metric equivalent to gear pitch.

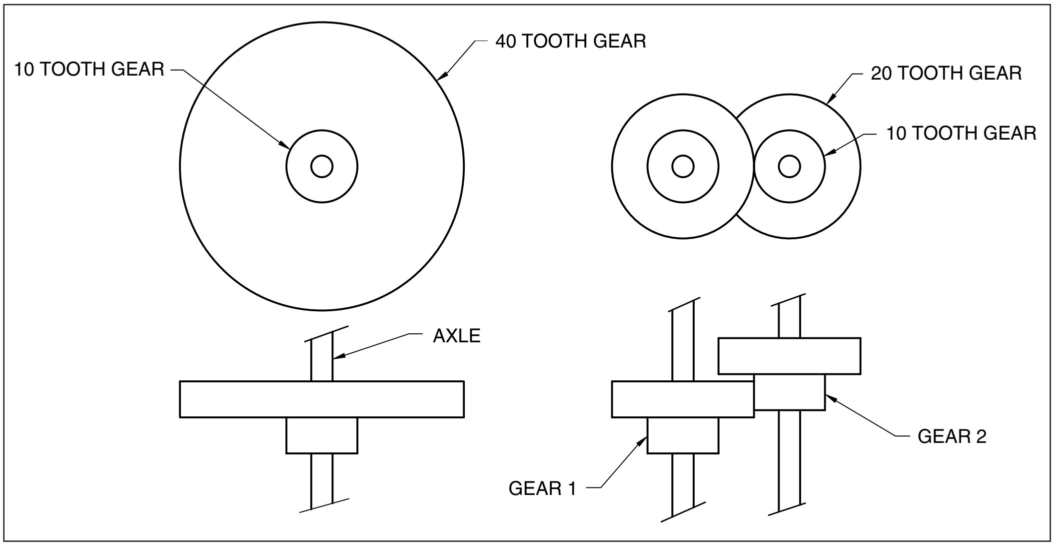


Figure 3. Using a pair of 20-to-10 tooth compound gears to obtain the same gear reduction as a single 40-to-10 tooth compound gear.

If you have been getting your gears from **NELNICK.COM**, then I would keep trying to call them. Phone calls are better than email requests because you never know if they actually get your email when they go unanswered, of if the email accidentally gets treated as spam. Looking at their website, it doesn't look like they are selling the 40-to-10 tooth compound gear anymore.

One alternative solution is using a pair of 20-to-10 tooth compound gears (which Nelnick.com sells) to obtain the same gear reduction as a single 40-to-10 tooth compound gear. Figure 3 illustrates this concept. To do this, you would have to add an additional axle in your gearbox.

Also, thanks for the link to Kelvin (**www.kelvin.com**). I am amazed at all of the different types of parts they sell. They are almost like a one-stop robot parts shopping store. It doesn't look like they sell 4:1 compound gears, either. One idea that might work is to make a compound gear by bonding a 10 tooth gear to a 40 tooth gear. The gear set P/N 990174 you mentioned from Kelvin has both of these gears. The key is making sure that you get a good bond between the two gears and that they are perfectly concentric on the same axle.

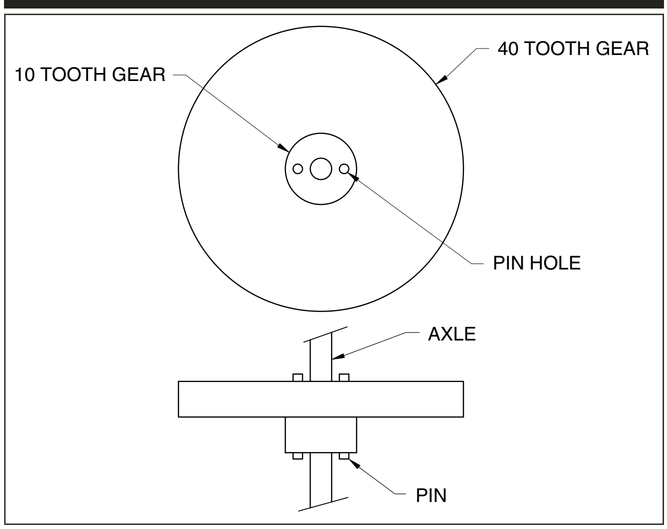
The problem that you may run into when doing this is that the torque between the two gears will shear the bond joint between them. One way to solve this problem is to pin the two gears together after they have been glued. Here, you will drill two small diameter holes on opposite sides of the axle through both gears. Then press a small diameter pin into both holes. The diameter of the pin needs to be slightly larger than the hole that you drilled.

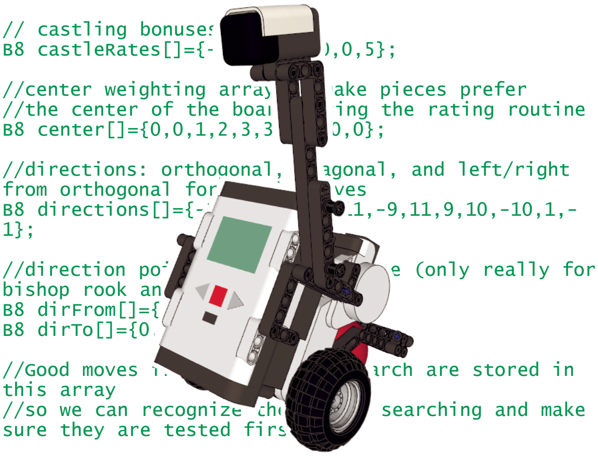
Music wire or a small paper clip should work. Since a 10-tooth gear is rather small, you will want to drill the holes near the axle, and use something like a 1/32 inch diameter wire. Figure 4 illustrates this concept. This is far from ideal, but it could help you out while you continue to search for an actual replacement.

If you want a much wider selection of individual gear types, sizes, and materials, I would suggest that you take a look at Stock Drive Products (**www.spd-si.com**). They don't have compound gears, but they will have all the other sizes that you need, and the huge variety of choices might help you with different design options.

As for casting a gear from one of your existing gears, yes, this can be done. This intrigued me enough to give it a try myself. Unfortunately, I ran out of time to test it for this article (I didn't receive the casting materials in time), but next month I will talk about casting a compound gear. **SV**

Figure 4. Illustration of pinning two gears together to make a compound gear.





LESSONS FROM THE LABORATORY

A bi-monthly column for kids!

The Runt and the Compass

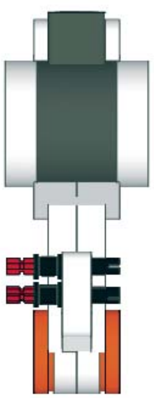
by James Isom

Spring is a time of renewal — flowers flowering, birds singing, new robot chassis glistening in the sun ... well, maybe for some of us. So, in this light, I present to you without further flourish a fresh robot design for spring — Runt. I developed Runt in my never-ending quest for a chassis that is sturdy and quick to build. It obviously takes some of its strengths from JennToo (see Lesson's from the Lab — October '06), but unlike JennToo, it can be built from a single LEGO Education NXT Base Set which I know many of you have, know, and love.

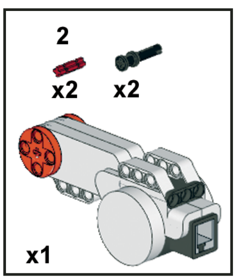
Towering above Runt is the new compass sensor from HiTechnic (**www.hitechnic.com**). This little guy senses direction by measuring fluctuations in the Earth's magnetic field. After we build Runt, I'll show you how to program a popular robot trick using the new Compass Block for the NXT-G software.

BUILDING RUNT

STEP 1:

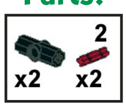


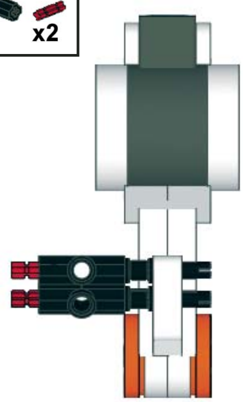
Parts:



STEP 2:

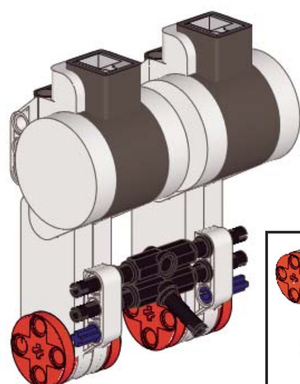
Parts:



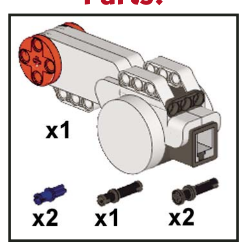


STEP 3:

Tilt the bottom connector as shown in the picture. It will hold the long pin that keeps the ball from falling out.

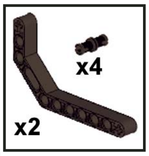


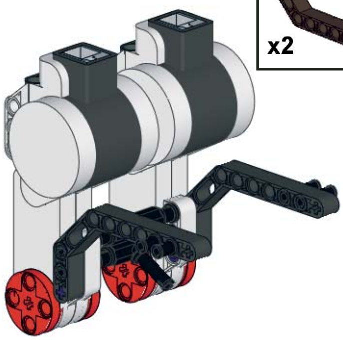
Parts:



STEP 4:

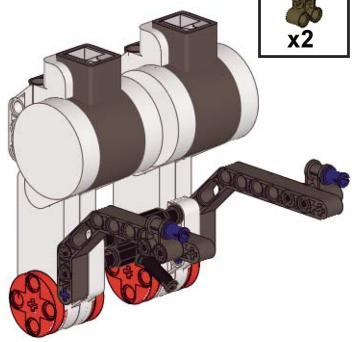
Parts:





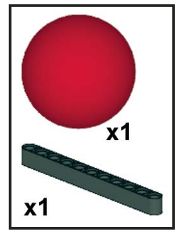
STEP 5:

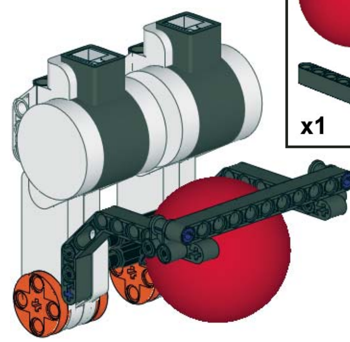
Parts:



STEP 6:

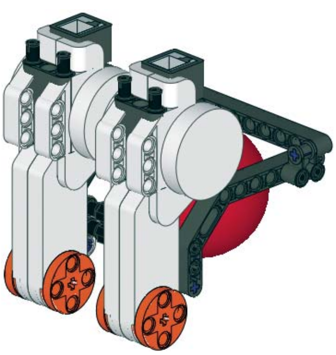
Parts:





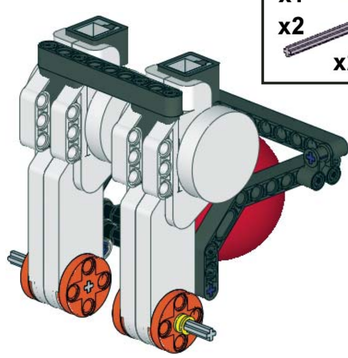
STEP 7:

Parts:



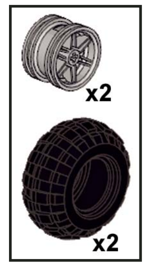
STEP 8:

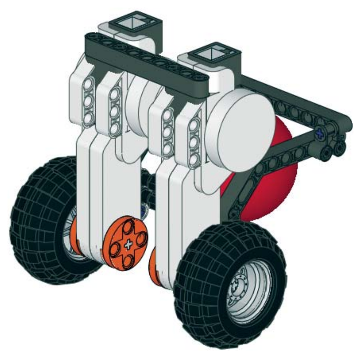
Parts:



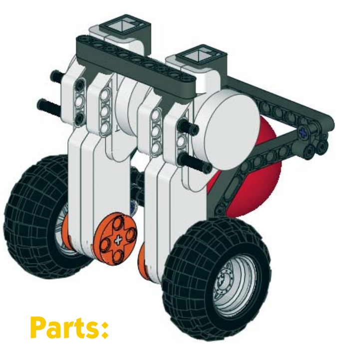
STEP 9:

Parts:





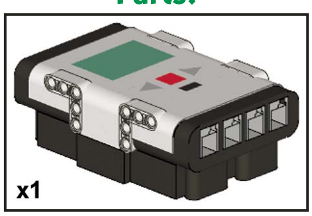
STEP 10:

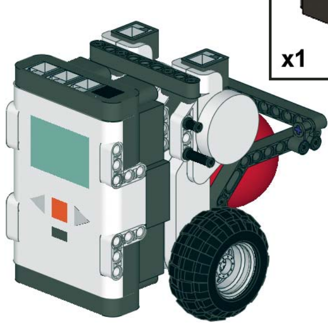


Parts:

STEP 11:

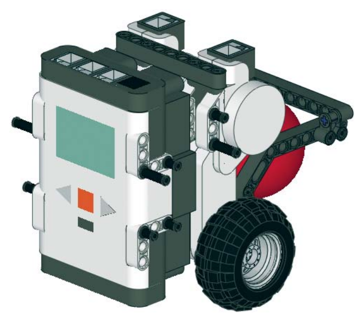
Parts:





STEP 12:

Parts:

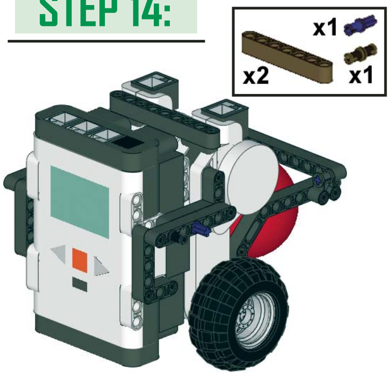


STEP 13:

Parts:

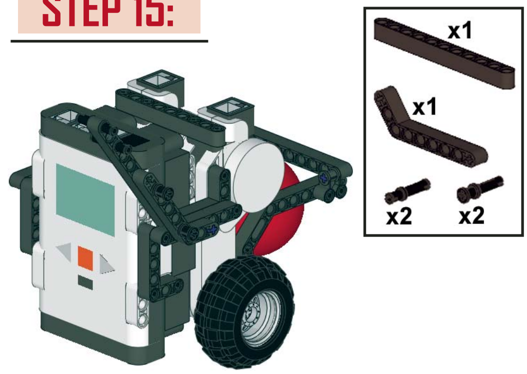
STEP 14:

Parts:



STEP 15:

Parts:

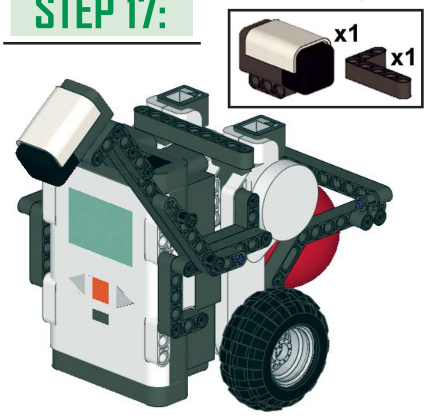


STEP 16:

Parts:

STEP 17:

Parts:



STEP 18:

Congratulations!
You're finished!

Wiring it All Together

Face the front of the NXT (ball in the back). Wire the left motor to Port B and the right motor to Port C. Connect the compass sensor to Port 2 (see Figure 1).

Use the nine-hole beam connecting the two motors together to wrap your extra wire around (see Figure 2).

FIGURE 1

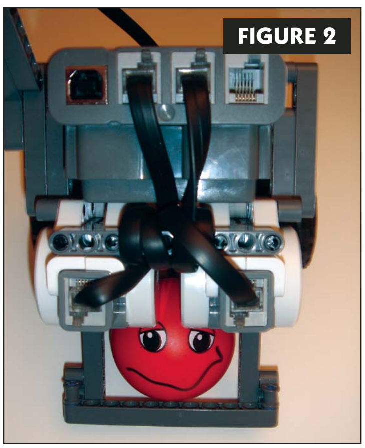


FIGURE 2

A Little Program for the HiTechnic Compass Sensor

This little south-pointing trick has been around awhile but I hadn't seen a version that used the new HiTechnic Compass Sensor Block, so I thought I would give it a whirl here. The program is a simple demonstration of how the compass sensor works. When the program starts, the robot will turn to face whichever direction it's programmed to face. The farther away the robot is from the desired direction, the faster it will turn to get back. Special thanks to Brian Davis for helping me work out a nice, efficient way to do this.

To begin, download and install the Compass Sensor Block. If you don't know how to do this, visit the resources section at **LEGOedwest.com** for a quick tutorial.

Once installed, drag the following programming blocks from the Complete Palette into a loop: Compass Sensor, Math (x2), and Move (see Figure 3).

Select the Compass Block to bring up the configuration panel.

We want the robot to turn to the south when the program runs. Setting the compass sensor action to Absolute Reading will return a heading from 0 to 359 degrees, with 0 representing north, 90 east, 180 south, and so on (see Figure 4).

Connect the wire from the Absolute Heading plug into the B plug of the first Math block. Run a second wire from the YES/NO Logic plug to the Motor Block's Direction plug. This will control which direction the robot turns when trying to return to south.

The Math Block shown in Figure 5 gives us the margin of error by taking the desired direction "A" (south = 180) and subtracting the current heading. The number the block spits out represents how far from south the robot is. The resulting number is used to control motor speed. The farther from south, the higher the number and the faster it goes, gradually slowing as it reaches south where it comes to a complete stop. For example, 180 (the desired heading) - 180 (the actual heading) = 0 (power).

Connect a wire from the Result plug of this block to the A plug of the second Math block.

The second Math Block shown in Figure 6 isn't really necessary but I thought it was a fun way to tune the robot's reaction time so it's been left in. Try dividing by 2 or more to see what happens to the robot's reaction time.

Connect a wire from this block's Result plug into the Power plug of the Motor block.

Finally, set up the Motor Block to the configuration in Figure 7. Then download your program and run it. Try putting your robot on a piece of paper so that you can turn it in place. You will find that no matter which way you try to turn, the robot always returns to face south. Furthermore, the faster you move the paper, the faster the robot turns.

What's going on? Two things really. The first we've already gone over, establishing a margin of error in order to send more or less power to the motors, depending on how far off the robot is from facing south.

The next thing is to watch the Logic plug. The Logic plug sends out a binary signal (0 or 1, off or on, true or false), depending on whether or not it is in the range of 180-359 that tells the robot which direction to turn to get back facing south the quickest. If the robot is off by a small amount, it doesn't have to turn all the way around; it knows which way is the quickest back to south.

Like most things made with LEGO, the Runt chassis is a work in progress. By the time this article runs, the design will probably have changed. For updates to the design, building instructions for other sensor attachments, and an additional program by Brian Davis that uses the Relative Heading of the Compass Sensor, visit **www.LEGOedwest.com**. **SV**

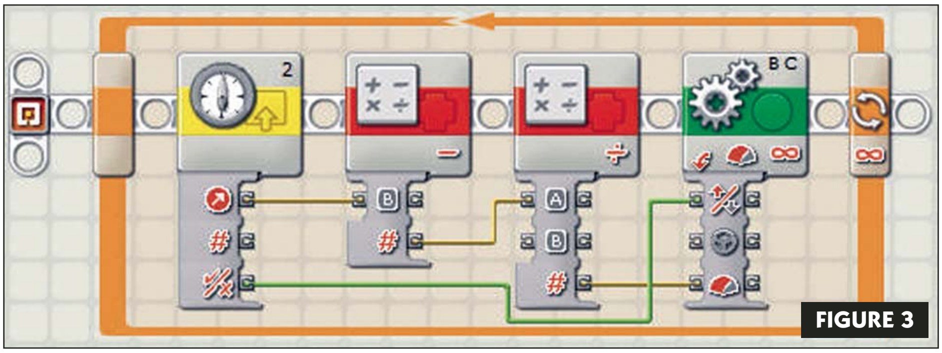


FIGURE 3

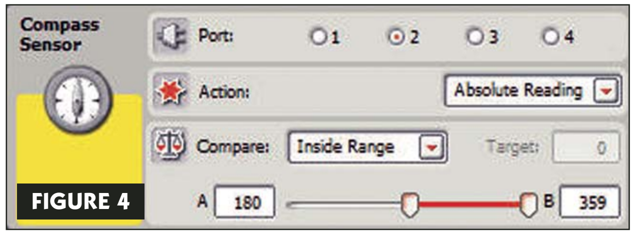


FIGURE 4

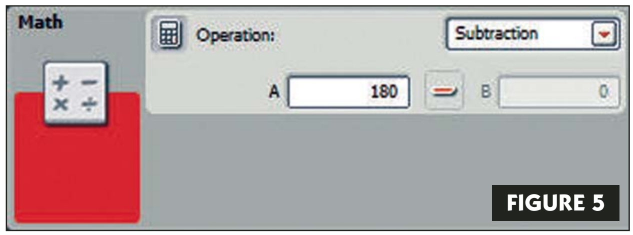


FIGURE 5

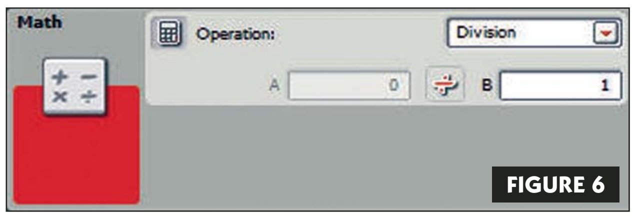


FIGURE 6

FIGURE 7

EVENTS CALENDAR

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

April and May are traditionally the busiest months of the year for robot competitions, and this year is no exception, with at least 17 events taking place in April and 11 more in May. This is a conservative estimate as there are at least a half-dozen others that hadn't set final event dates by *SERVO*'s publication deadline. This brings to mind a helpful hint to event organizers: If you want your event to be listed in print publications, plan and announce your event date well in advance. An announcement of the event date and venue should be made at least 90 days beforehand. Well-organized groups tend to plan a full year in advance.

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

April

1-8 **Africa Cup International Robotics Competition**
Pretoria, South Africa
Events include obstacle race, wall climbing, Sumo, and robot soccer.
www.nydt.org/home.asp?pid=713

12 **BattleBotsIQ**
Location TBA
Student-built, radio-controlled vehicles destroy each other for the educational value.
www.battlebotsiq.com

12-14 **FIRST Robotics Competition**
Atlanta, GA
National Championship for the regional FIRST winners.
www.usfirst.org

14 **CRAC Maisonneuve Sumo Competition**
College Maisonneuve, Montreal, Canada
Another autonomous Sumo event.
http://crac.cmaisonneuve.qc.ca

14 **CybAiRBot**
Poznan, Poland
This robot competition and exhibition will feature robot Sumo played by the traditional Japanese rules.
www.sumo.put.poznan.pl

14 **UC Davis Picnic Day MicroMouse Contest**
University of California, Davis Campus, CA
Standard micromouse contest.
www.ece.ucdavis.edu/umouse

14-15 **Trinity College Fire-Fighting Home Robot Contest**
Trinity College, Hartford, CT
The well-known championship event for fire-fighting robots.
www.trincoll.edu/events/robot

20 **Carnegie Mellon Mobot Races**
CMU, Pittsburgh, PA
The traditional Mobot slalom and MoboJoust events.
www.cs.cmu.edu/~mobot

21 **RoboRodentia**
California Polytechnic State University, San Luis Obispo, CA
A micromouse-like maze navigation contest for autonomous robot mice. In addition to navigating the maze, robot must pick up balls and place them in a nest.
http://ieee.ee.calpoly.edu/ieee_cs/roborodentia.php

25 **Istrobot**
Slovak University of Technology, Bratislave, Slovakia, EU
This one includes line-following, IEEE Micromouse, mini Sumo, and free-style events.
www.robotics.sk

27 **RobotRacing**
University of Waterloo, Ontario, Canada
Autonomous cars must race head-to-head on outdoor courses. A two-car drag race on a 20-meter straight course is followed by a multi-car, multi-lap race on a 150-meter circuit course. The circuit course is bounded by orange cones and GPS waypoints are provided.

www.robotracing.org

27-30 **Robotica National Festival of Robotics**
University of Minho, Portugal
The festival continues through May 1st and includes events for small autonomous robots that include RoboCup soccer and robot dancing.
www.ccvalg.pt/robotica2007

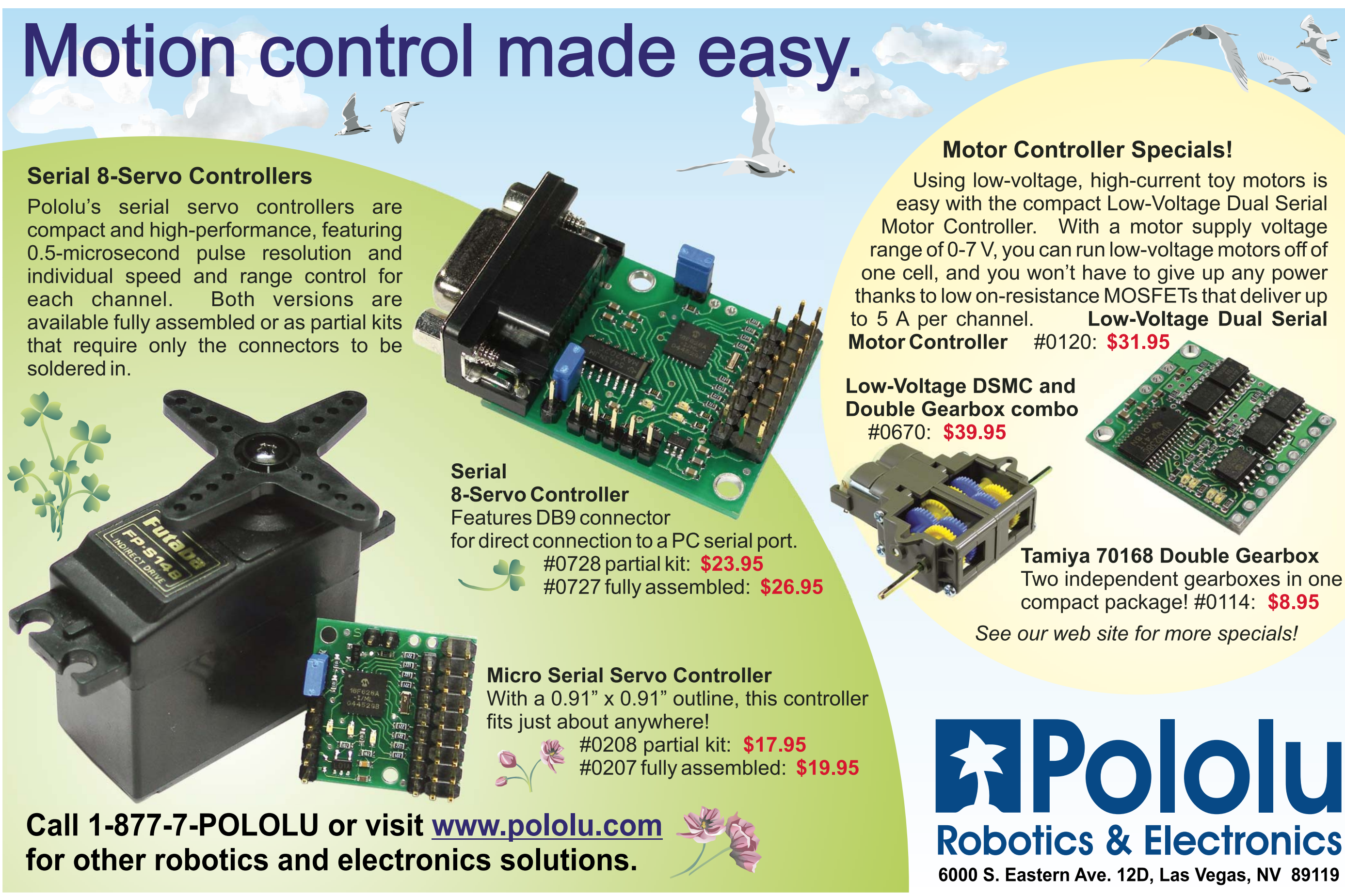
28 **Historical Electronics Museum Robot Festival**
Linthicum, MD
Events include fire-fighting, FIRST, robot Sumo.
www.robotfest.com

28 **RoboFest**
Lawrence Technological University, Southfield, MI
Events for LEGO robots and advanced robots.
http://robofest.net

28 **ROBOMO Maze Solving Competition**
Winfield VFW, St. Louis, MO
The St. Louis Area Robotics Group has designed a competition for autonomous maze-solving robots. The goal is to be the fastest robot to get from the maze entrance to the maze exit.
www.robomo.com

28 **The Tech Museum of Innovation's Annual Tech Challenge**
San Jose Civic Auditorium, 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

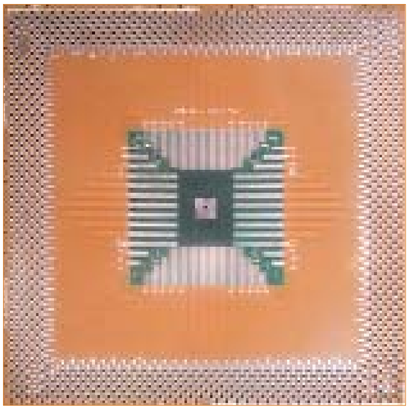
Continued on page 70



NEW PRODUCTS

CIRCUIT BOARDS

Three New SchmartBoards



SchmartBoard — the developer of a new technology that has significantly simplified the creation of electronic circuits for hobbyists, education, and industry — announces the release of three new boards for large QFP (Quad Flat Pack) devices.

SchmartBoard|ez prototyping boards allow virtually anyone to hand-solder surface-mount components easily, quickly, and flawlessly. Prior to the development of this technology in late 2005, few people had the dexterity to hand-solder surface-mount components with small pitches, such as .8 mm or smaller. The advent of SchmartBoards has allowed people to hand-solder components with up to 100 pins, which previously was inconceivable to most engineers, students, and hobbyists.

Based on demand from experienced users, SchmartBoard has launched several new boards with the ability to support larger and more complicated components.

"People who, a couple years ago, would not be able to hand-solder some small components with eight pins are now asking for us to support up to 256 pins," said Neal Greenberg, vice president of marketing at SchmartBoard. "There are many components that are this large for microprocessors, FPGAs, and other advanced ICs."

Three new SchmartBoards will now support these large QFP package components. The only other prototyping options today for large surface-mount components require expensive sockets.

The three new boards have been specially created for QFP components and are available with the following specs:

- .8 mm pitch with 120-144 pins
- .65 mm pitch with 112-160 pins
- .5 mm pitch with 128-240 pins

For further information, please contact:

SchmartBoard	Website: **www.schmartboard.com**

MOTOR CONTROLLERS

DC Motor Controller/Driver for Mobile Robots

Roboteq, Inc., introduces a microcomputer-based, dual channel DC motor controller capable of directly driving up to 15 amps on each channel at up to 24V. The AX500 is targeted at designers of mobile robotic vehicles including Automatic Guided Vehicles (AGV), Underwater Remote Operated Vehicles (ROVs), and robots for exploration, hazardous material handling, and military and surveillance applications. The motor controller is equally suitable to most traditional motion control applications in machines and factory automation.

Fitted on a compact 4.2" x 2.9" board, and targeted primarily to OEMs, the controller accepts commands from either a standard R/C radio for simple remote controlled robot applications or serial port interface. Using the serial port, the AX500 can be used to design fully or semi-autonomous robots by connecting it to single board computers, wireless modems, or wireless LAN adapters.

The controller's two channels can be operated independently or combined to set the direction and rotation of a vehicle by coordinating the motion on each side of the vehicle. The motors may be operated in open or closed loop speed mode. Using position sensors, they may also be set to operate as heavy-duty position servos.

The AX500 features intelligent current sensing and controlling that will automatically limit each channel's power output to a user preset value up to 15A per channel.

The controller supports a long list of features, including analog and digital I/Os for accessories and sensors, thermal protection, programmable acceleration, input command watchdog, and non-volatile storage of configuration parameters.

The AX500 can be reprogrammed in the field with the latest features by downloading new operating software from Roboteq's website.

The AX500 is offered as a board-level product. The board may be ordered as a single channel controller capable of driving up to 30A loads.

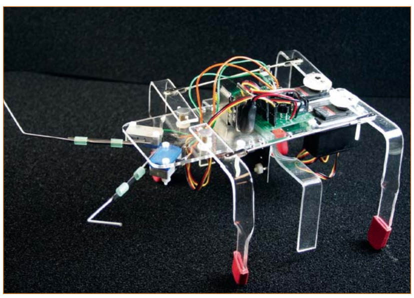
The AX500 is available now at $145 in single quantities, complete with cable and configuration software. Product information, application examples, and software can be downloaded from the company's website.

For further information, please contact:

Roboteq, Inc. Website: www.roboteq.com

ROBOT KITS

Hexapod Walker



Images SI, Inc., introduces the Hexapod walker robot kit. It has a plastic base and legs, (3) servomotors (42 oz), two switch sensors and whiskers, PC board, microcontroller, electronics, mounting hardware, and a booklet. The hexapod walker is available in two leg colors: transparent (shown) and black; the PIC-WALK-01T and PIC-WALK-01B, respectively.

The Hexapod Walker has the following features:

- Simple three servomotor construction.

- Standard alternating tripod gait (walking). It can turn left and right and walk forward and backward.

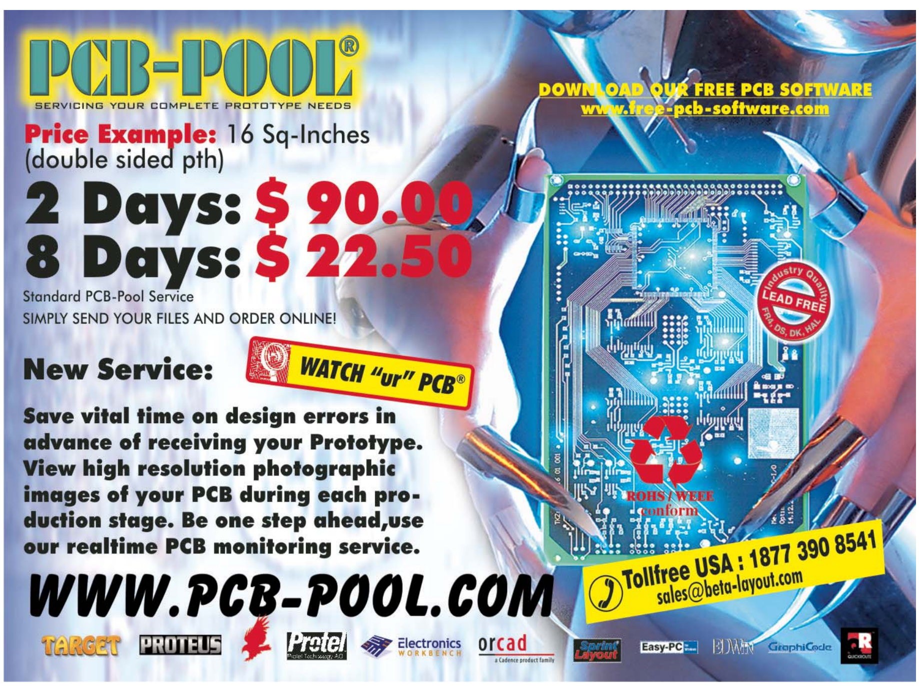
- Obstacle Avoidance: Front mounted sensors help hexapod avoid and move around obstacles.

- Power: Nine-volt battery.

- MSRP $149.95.

For further information, please contact:

Images Scientific Instruments, Inc.
109 Woods of Arden Rd.
Staten Island, NY 10312
Website: **www.imagesco.com/kits/hexapod-walker-kit.html**



COMBAT ZONE

Featured This Month

PARTICIPATION

The Safe Way to Test Spinners

● by Jeffrey Scholz

Here are five specific instructions to test your spinner style robot for safety.

First, ensure your robot properly responds to the radio. Do you still have control when operating it from 100 feet away? You do not want your drive train to glitch when you throttle up the weapon.

To test this, disconnect the weapon motor from the weapon drive and put a tape flag around the shaft. Drive the robot around for a minute and inspect it to see if the motor shaft rotated. Now, activate the weapon motor. Does the drive train twitch? Even if it doesn't, bear in mind it will turn a massive load after you reconnect the motor, which may cause fluctuations in the power supply.

Second, confirm the blade is balanced; it should not vibrate when it spins. Depending on how severely the blade shakes, the robot can drift or bounce around while the weapon operates. Don't test vibration by spinning up! Instead, place the bore of the weapon over a pool ball; if the blade has a heavier side, it will fall towards that side.

You *did* install a weapon lock, right? If not, go fashion a solid one immediately!

Now for the spin-up test. If size allows, clamp the robot in a vise. Otherwise, place the robot in the middle of an empty parking lot *and stand far back and behind something to protect you from*

Radio? Check. Blade balanced? Check. Weapon lock? Check. Face shield? Check.

shrapnel. Wear a face shield and push the throttle up slowly until the weapon barely turns. Now stop. Next, attempt spinning it a little faster, stop, and repeat. If you see the robot starting to drift or shake, stop testing and fix the problem immediately.

Did it safely reach full speed? Congratulations! Satisfied that the weapon works, proceed to test the drive and the weapon simultaneously. Do not let the weapon hit anything! Bear in mind that the blade might catch some debris and fling it towards you.

Finally, are you ready to hit stuff? Always remember to *stay out of the plane of the blade*. For horizontal spinners, stand on a roof. For vertical spinners, stay on the ground and along the robot's side.

Here are a few more things to keep in mind:

- The blade might bend back and hit the frame, causing it to explode.
- An overhardened blade or one with sharp internal corners will snap.
- Weapon teeth can come off.
- Concrete is no match for a spinning blade; it will shatter.
- Tighten down all the shaft collars and bolts. SV

Combat Brackets

● by Christopher Gilleski

Tournament brackets and the equipment used to run an event are issues nearly as important to robotic combat as the actual robots competing in a tournament. Robotic combat brackets must tread a fine line between competitor satisfaction, fan enjoyment, and time constraints. Each bracket presents several positives and negatives in these categories, making the event organizer's task even more difficult.

The most famously-used brackets in robotic combat are the Single Elimination (SE) tournaments run by Battlebots during its run on Comedy Central. Single elimination is a bracket most sporting fans are used to, used by the NFL for its football playoffs and by the NCAA basketball tournament. It is simple for the audience to follow: If a robot loses, it is eliminated and winners advance until only one winner remains.

Some fan confusion does erupt from the determination of third place which — due to the symmetric nature of the brackets — must be determined by an added "Third Place Match" similar to what is done in the FIFA Soccer World Cup.

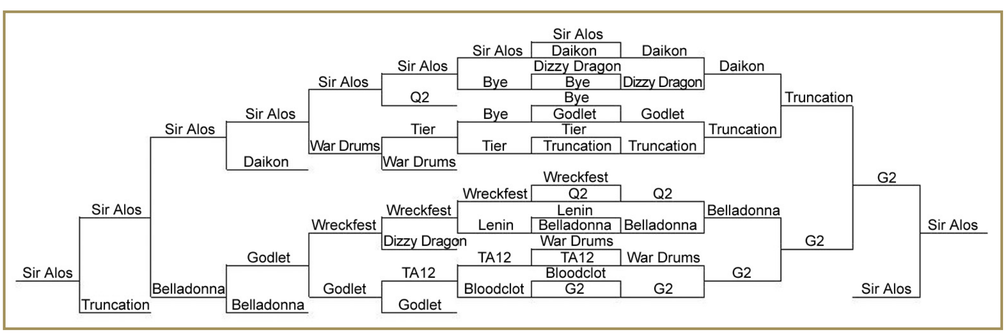
Single Elimination tournaments also provide the best option from a time management standpoint. By guaranteeing every entrant only one match, the number of matches in a tournament decreases significantly, shrinking the length of an event and the promoter's expenses. Negatives to this format exist, including the fact it is unpopular with competitors, and seemingly against the purpose of robotic combat.

Competitors build their machines and compete for a fair chance at determining the best robot in the tournament. Single elimination tournaments increase the likelihood of luck or unexpected mechanical failures determining a champion rather than the skill and ingenuity of a builder and driver due to the fact that a single loss will end months of work.

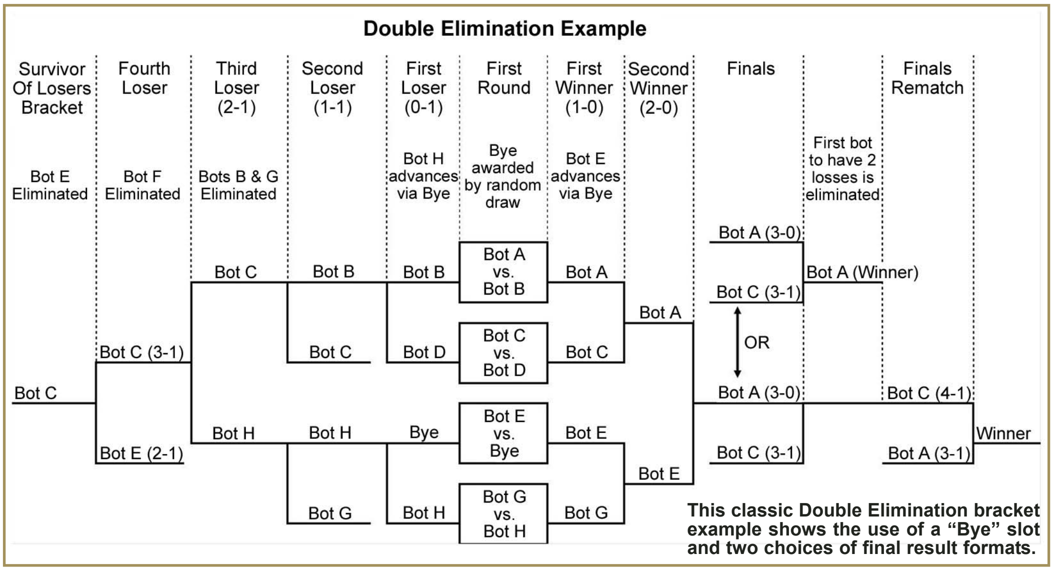
The opposite end of the spectrum is the round robin tournament. Round Robin (RR) tournaments consist of a field of competitors that will each face every other entrant in the tournament, with the best record in the tournament determining a champion, and second and third going to each corresponding record. The concept is simple for an audience to understand, but is less interesting to watch since it rarely yields a decisive exciting match for the championship.

Round robin tournaments are often employed in small robot events where the number of entrants is under eight and thus does not result in too many matches, which larger events may not be able to afford. In theory, this format is ideal for the competitor. In practice, this format sees increasing

A well-run small bot tournament board showing DE and RR formats. Photo courtesy of Bay Area Robot Fights.



This example of an actual DE bracket shows the complexity of running a tournament. Image courtesy of Gregory Dabkowski.



This classic Double Elimination bracket example shows the use of a "Bye" slot and two choices of final result formats.

repairs as the tournament continues, causing delays in matches and anticlimatic breakdowns.

A compromise adopted by most events is the double elimination tournament. Double Elimination (DE) tournaments consist of a double bracket, one for the winners of the first round matches and one for the losers. A robot that loses its first match is placed in the loser's bracket in a part of the bracket corresponding to where it lost in the winner's bracket. This continues throughout the tournament with losers from the winner's bracket being placed in the loser's bracket, and robots that suffer a second loss in the tournament being removed from the tournament

Finally, the only robot that survives the winner's bracket fights the robot that survives the loser's bracket in the finals of the tournament. The robot which loses to the survivor of the loser's bracket is awarded third place, which is more understandable to the audience than most of the double elimination format.

Audiences unaware of robotic combat are not used to double elimination brackets and often will not understand why a robot that has lost is still in the tournament. Furthermore, due to the separation of the brackets, it can often be difficult to tell how far the tournament has progressed for the uninitiated.

The finals of double elimination tournaments are the source of controversy in robot combat. In some events, the finals of the tournament are single elimination for the robot that has suffered no losses to this point, putting finalists on equal footing. It is argued that the winner of the winner's bracket has gone through fewer fights than that of the loser's bracket, so it is only fair that the final is single elimination. North East Robotics Club events continue to use double elimination rules in the finals of the tournament, meaning the robot that has suffered no losses to this point is not eliminated by a single loss in the finals. Instead, that robot is given the allotted time to make repairs and a second final is held. The latter method may seem more fair to the robot that has not lost, meaning it too gets the benefit of a loss that its opponent had earlier in the tournament, but this is a time-consuming process. Most events grant robots 20 or 30 minutes to make repairs in between matches, leaving the audience with nothing to do.

A compromise between single and double elimination tournaments has been created due to this controversy and is known as a Hybrid Bracket. Introduced recently by Combots, it begins as a double elimination tournament, and at a predetermined time, the tournament switches to single elimination. This shortens the length of an event while granting all robots at least two matches. A hybrid bracket also eliminates confusion in the finals over whether a rematch should occur.

Another unique option was presented at a "Pound of Pain" event. POP attempted to employ a Modified Swiss bracket. Robots were to be divided into two brackets, which would then fight each robot in the bracket round robin style. The winner of each bracket would fight each other to determine the champion of the tournament. Sadly, this novel approach is never used in an official event due to the number of entrants for the intended event.

In an event concerned with its competitor's needs, some form of double elimination tournament is a must. Giving competitors at least two matches rewards them for their effort building and money spent entering the tournament. A full double elimination tournament may

Round Robin Example

	Bot A	Bot B	Bot C	Bot D	Bot E	Record	Results
Bot A	x	Bot A	Bot C	Bot A	Bot A	(3-1)	1st Place
Bot B	x	x	Bot C	Bot B	Bot E	(1-3)	5th Place
Bot C	x	x	x	Bot D	Bot E	(2-2)	Tie
Bot D	x	x	x	x	Bot D	(2-2)	Tie
Bot E	x	x	x	x	x	(2-2)	Tie

Tie worked out by tournament rules such as:
- Further Round Robin among tied bots
- Coin toss
- Examination of records
  - Bot D beat Bots C & E so awarded 2nd place
  - Bot E beat Bot C but lost to Bot D so awarded 3rd place
  - Bot C lost to Bots D & E plus beat Bot B so awarded 4th place

A Round Robin format for five bots. The event organizer needs to consider and publish the rules for resolving ties as part of the event rules set.

not be possible for all events, but when possible, is the best compromise between tournament length and competitor satisfaction. If double elimination is impractical for an event, the new hybrid bracket presents a unique compromise that more events should explore if pressed for time. SV

THE MATH BEHIND SPINNING WEAPONS

● by Jeffrey Scholz

Spinning a piece of metal and ramming it into the opponent is currently the most common destruction technique. Designing these devices often raises questions like "is my motor big enough?" "Is the weapon powerful enough?" "How much time will it take to spin the weapon up to full speed?" The design process behind these weapons will be discussed here.

You're probably aware that a faster, heavier blade will deliver a harder hit than a slower, lighter one. Here is the exact relationship:

KE = 0.5*I*ω^2

where KE (kinetic energy) is in Joules, "I" is the moment of inertia, and ω is the angular velocity in radians/sec.

A Joule is a unit of energy sufficient to apply a force of one Newton (about 0.23 lbs) through one meter.

The moment of inertia (MOI) is a quantity describing how the mass of a spinning object is distributed; it is measured in kg*m^2. This means that the value goes up linearly as the object's mass increases, and exponentially as a greater portion of the mass moves away from the center of rotation. Figure 1 illustrates this principle.

The website **www.physics.uoguelph.ca/tutorials/torque/Q.torque.inertia.html** lists several formulas for finding the MOI of various shapes. Shapes with equations that have smaller fraction constants have more mass towards the center, and shapes with larger or no fractions have more mass towards the edge.

The other variable, ω, accounts for the weapon's speed. The unit for this measurement is radians/second, not RPM. One radian is equal to 180/π. To convert RPM to radians per second, multiply the RPM by 0.1047.

Let's add some meaningfulness with an example.

Sample weapon specifications:

Shape: Bar
Length: 6"
Width: 1"
Thickness: 1/8"
Material: Steel, which has a density of 4.512 oz/in^3
Speed: 10,000 RPM

How much energy will this weapon have? First, find the bar's mass:

Mass = volume*density = 6[in]*1[in]*1/8[in]*4.512[oz/in^3] = 3.384 oz = 0.09593[kg]

Length = 6", or 0.1524[m]

Width = 1" or .0254[m]

Bar MOI = (1/12)*mass*($length^2$ + $width^2$)

I = (1/12)*0.09593[kg]*(0.15242 + 0.02542) = 1.9083E-4[kg*m2]

ω = 10,000RPM*0.1047 = 1047[rad/s]

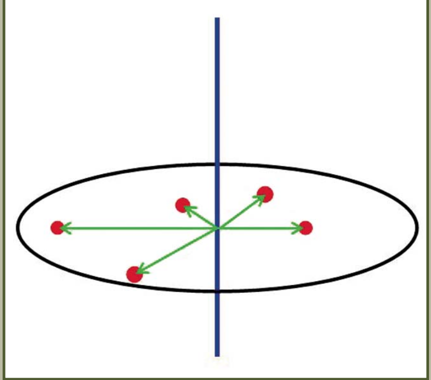


FIGURE 1. Moment of Inertia: Sums up all the little bits of mass (represented by the red dots) in a shape and factors in their distance, r (represented by the green arrows), from the center around which the object spins (the blue line).

Now, plug it into the formula:

KE = .5*I*ω^2 = .5*1.9083E-4[kg*m^2]*1047^2[rad/s] = 104.5948 [Joules]

The blade will store 104.5948 Joules. However, only a fraction usually no more than 2%) will actually get transferred into the opponent because running a spinning blade into a target is an inefficient way to transfer the energy. Spinners tend to have between 50 and 200 Joules for every pound of the robot's total weight.

Another piece of important information is the spin-up time. You'll need the specifications mentioned earlier along with specifications of the battery to calculate this. We'll do another example, this time with the 3S Polyquest 400 battery.

Specifications:

Voltage: 11.1[volts]
Milliamp-Hour rating: 400[mAH]
Maximum continuous current: 20C; 20*400[mAH] = 8000[mA]

or 8[A]
Maximum watts: 11.1*8 = 88.8 [Watts]

The formula is:

Seconds = (Iω)/Torque

Torque is in Newton-meters and ω is the desired angular speed. A brushless motor's torque is rarely specified. However, you can back-calculate this from the motor's voltage constant. The voltage constant, signified by *Kv*, tells you how many RPMs the motor will produce for every volt you feed it. The torque constant, signified by *Kt*, tells you how much torque the motor produces for every amp it draws. The *Kv* and *Kt* have a very important relationship:

$$Kv_{[rad/s]} * Kt_{[N \cdot m]} = 1$$

Thus,

$$Kt = Kv^{-1}$$

In order for the weapon to spin at 10,000 RPM at 11.1 volts, the motor must have a Kv of 900.9 RPM/volt or 94.3242 (rad/s)/volt. Thus, its Kt is 0.0106 N·m/Amp. The battery can source eight Amps, therefore ...

Torque = 0.0106[N·m/A]*8[A] = 0.0848[N·m]
ω = 1047[rad/s]
I = 1.9083E-4[kg*m^2]

Spin-up time = (1047[rad/s]* 1.9083E-4[kg*m^2])/0.0848[N·m] = 2.3561 seconds

Those computations work for *brushless motors*, not "brushed" motors. Brushless motors output maximum torque from 0 to no-load speed; with brushed motors, torque and RPM are inversely proportional. As the RPM approaches no-load, the acceleration drops. After the blade reaches 66.3% of the no-load speed, the torque diminishes rapidly. Determining exactly when the weapon stops accelerating will require calculus. However, if you only want to know how long it takes to spin the blade up to 66.3% of the motor's no-load speed at a given voltage, then you can calculate the spin-up time using the same formula. Since we targeted 10,000 RPM, we want a brushed motor with a no-load speed of 15,083 RPM at 11.1 volts (10,000 is 66.3% of 15,083). Bear in mind ω must be the no-load speed of the brushed motor, not the desired speed of the blade.

If a motor produces 15,083 RPM at 11.1 volts, it has a voltage constant (Kv) of 1358.8 RPM/volt, or 142.27 (radian/s)/volt. If we use the same components as in the previous examples, we can infer the torque will be 0.05623 [N·m] and the MOI will be 1.8701E-4. All together now:

(1579.1901[rad/s]* 1.8701E-4[kg*m^2])/.05623[N?m] = 5.252 seconds

You may wonder how a gear or belt reduction factors into the equation. Just treat the output stage of the reduction as the output of the motor. If you don't want to do all this math yourself, you can go to these websites:

- **www.teamcosmos.com/ke/ke.shtml** — Calculates the MOI and kinetic energy.
- **www.totalinsanity.net/tut/mechanical/spinuptime.php** — Calculates spin-up time.
- **www.onlineconversion.com** — Converts between metric and imperial units.

At this point, you can determine how much power the weapon holds, choose the right battery, select the correct motor, and predict the spin-up time. You will avoid having a weapon that spins up too slowly, or unintentionally buying an oversized motor or battery. Have fun and build safely! SV

EVENTS

RESULTS — January 15th - February 11th

Kilobots X, Spectrum 2007 was held on 1/20-21/2007 in Saskatoon, Saskatchewan, Canada. The 10th event in this series was presented by the Saskatoon Combat Robotics Club, and was an RFL Nationals qualifying event. Results are as follows:

- *Antweights (1lb)* — 1st: "Microrat," lifter, Rumble Robotics; 2nd: "Wizard Beard," lifter, GuavaMoment; 3rd: "Glitch 2," plow, Chaos Robotics.
- *Kilobots (1kg)* — 1st: "Roadbug," wedge, Chaos Robotics; 2nd: "Underkill," spinner, Inner Logic; 3rd: "Goliath," drum, FingerTech.
- *Beetleweights (3lb)* — 1st: "Limblifter," lifter, GuavaMoment; 2nd: "Wedgie," Inner Logic; 3rd: "LoBlow," spinner, Chaos Robotics.
- *Mantisweights (6lb)* — 1st: "Wedgely Brickleson," wedge, FingerTech; 2nd: "G.I.R.," Drum, Chaos Robotics; 3rd: "Pedro," wedge, Rumble Robotics. SV

TECHNICAL KNOWLEDGE

Combat Lifecycle — Design and Construction

● by Bob and Robert Jacoby

KillerV (pronounced "Killer Vee") was conceived and built to break the dominance of full-body spinners in the middle weight (120#) classes. The primary weapon is the open "V" in the front which is intended to trap full body spinners and effectively use their own kinetic energy against them.

The secondary weapon, a full body spinning mode, activated by the "Super Tasmanian Devil" button instantly puts one drive motor in full forward and the other motor in full reverse. Finally, the third line of defense is KillerV's wedge-like features, which allow it to get underneath its competitors.

Pushing power and speed were the main considerations so the robot was designed around two NPC-T74 motors (a slightly beefier version of the current NPC-T64 motors) coupled to NPC flat-proof tires. Other components included an Innovation First robot controller, Victor 883 speed controllers, and two SVR 12-volt, sealed lead-acid batteries wired in series to produce 24 volts.

Figures 1-11 chronicle the design and construction.

A small paper and construction board model was built to visualize the concept (see Figure 1). We actually used this paper model to "play fight" combat robot toys in order to spot any design errors or major vulnerabilities.

Next, look at Figure 2 — a full scale model, minus the V was built in plywood. This entire, fully-functional model was built in one weekend and was relatively inexpensive (relative to CNC plasma-cut aircraft grade aluminum!). The plywood prototype also allowed us to test all the electronics and finalize the onboard software. Two-wheel robots are a little tricky to drive straight at high speed. KillerV's proportional steering software was written in PBasic, and final software testing and debugging was completed in the wooden prototype. High speed stability and overall agility met all of our expectations.

Refer to Figure 3. The V was added to the plywood prototype. While we didn't do any combat testing with the wood version, we did play "robot soccer" with a basketball which we managed to puncture.

The individual pieces were designed and drawn in AutoSketch® (see Figure 4). Tolerances were specified to accommodate the CNC plasma cutting process. The weight of each individual component was calculated as it was designed, in order to keep the entire bot under its 120-pound weight limit. Finally, the file was saved in DXF format to facilitate

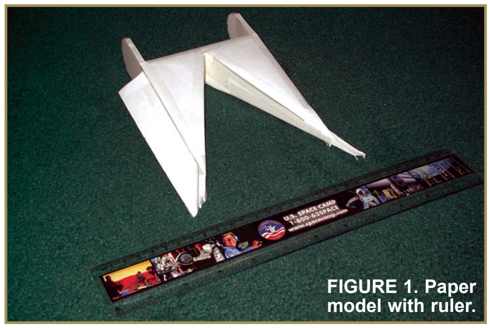


FIGURE 1. Paper model with ruler.


FIGURE 2. Wooden prototype. Robert (left — age 11) and Bob (right).

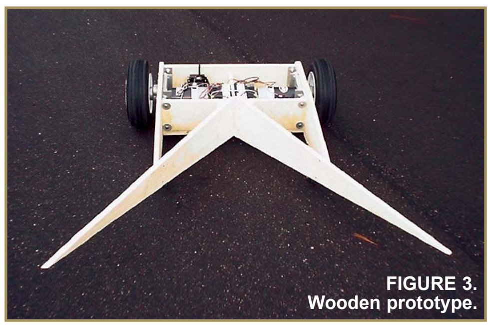


FIGURE 3. Wooden prototype.

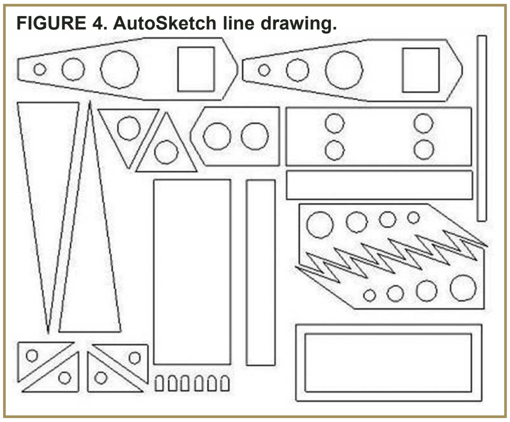
FIGURE 4. AutoSketch line drawing.



FIGURE 5. Plasma cutter.

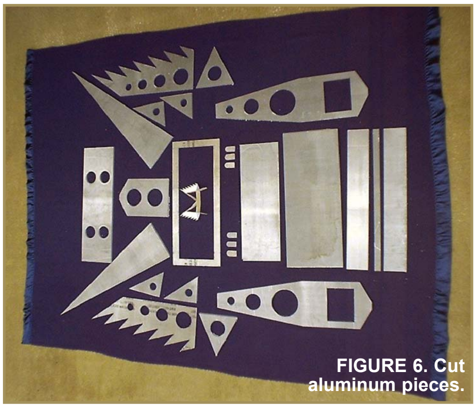


FIGURE 6. Cut aluminum pieces.

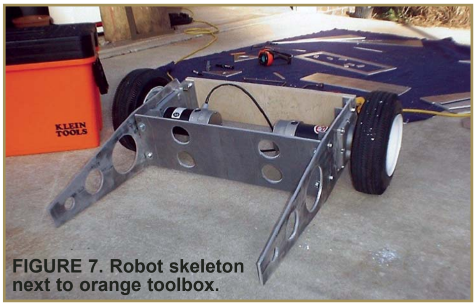


FIGURE 7. Robot skeleton next to orange toolbox.

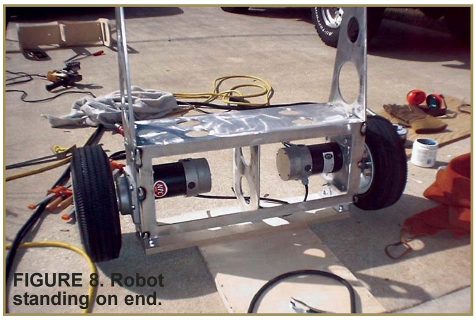


FIGURE 8. Robot standing on end.

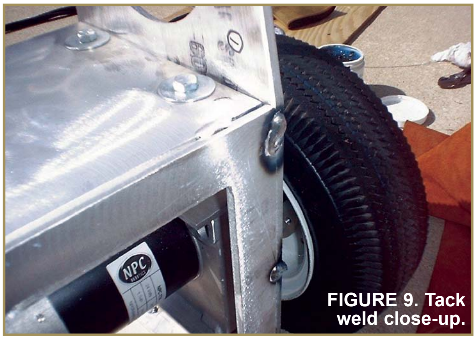


FIGURE 9. Tack weld close-up.

conversion into G-code — the language required by the CNC plasma machine.

KillerV was cut out of 1/4" 6061 T6 aircraft grade aluminum on a CNC plasma cutter (see Figure 5). If you look closely, you can see that the aluminum plate is semi-submerged in water. The water helps to keep the material cool and also captures dust created by the cutting process.

Look at Figure 6. Ordinarily, CNC plasma cutting produces much cleaner parts, but we had moisture problems with the torch head which severely degraded the cut quality. The rough edges on the aluminum parts were cleaned with coarse files, a hoof rasp (which is an extremely coarse file), and a belt sander. Here you can see the parts laid out and ready for assembly. The round holes were designed into the parts in order to stay under the 120-pound weight limit. Note the additional paper model set in among the parts.

KillerV is built around its drive train – NPC-T74 motors and gearboxes. The first step in assembly was to align the side panels and front bulkhead to the drive train. Once aligned, a plywood panel was bolted to the back of the bot to hold everything symmetricaly during welding (see Figure 7).

The transverse interior bulkhead (the piece between the motors) and top were the first to be tack-welded in place. The angle grinder in the background is equipped with a special stainless steel wire wheel specifically designed to clean aluminum prior to welding. KillerV's plywood prototype is visible in the upper left of Figure 8. Other items in this photo include: the ground clamp for the welder, a leather welding apron, a jar of "tip dip" for the welder, MIG welding gloves, and ear protection.

Figure 9 shows a close-up shot of the tack welds. KillerV was welded using a MillerMatic® 185 MIG welder equipped with a spool gun (a special attachment for welding aluminum and other hard-to-weld materials). Argon gas was used to shield the molten metal and 0.035" and 0.030" 4041 wire was used as a filler material.

KillerV is mostly tacked up in Figure 10. Large welding projects are always tacked first in order to limit the distortion caused by the welding process. The KillerV plywood prototype is sitting to the left of KillerV in this photo. The final Bot was reduced in size to keep KillerV within its weight limit. KillerV's final fighting weight was 119.5 pounds which is pretty close to the limit in the 120-pound class!

Figure 11 shows a final shot before fight day. You'll notice the small round objects at the front of the bot — these are known as "ball transfers" and allow KillerV to keep the points of the V slightly off the ground. The ball transfers also provide a great degree of lateral mobility.

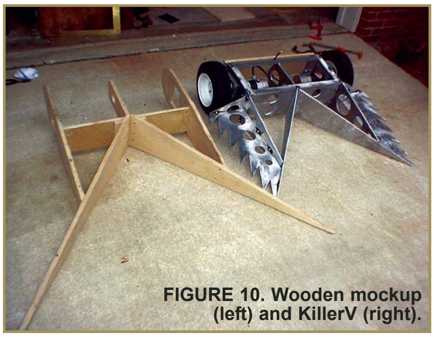


FIGURE 10. Wooden mockup (left) and KillerV (right).

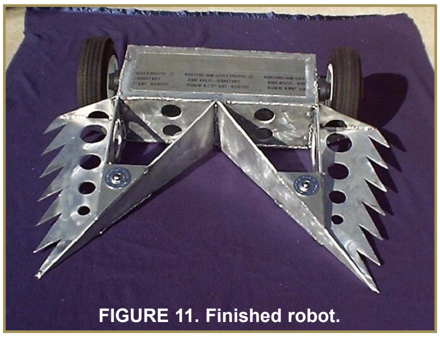


FIGURE 11. Finished robot.

Conclusion

KillerV was built on an extremely tight timeframe and was literally finished just hours before its first fight. The paper and wood prototyping process really shortened the development time. For instance, we were able to complete the control software using the plywood prototype while waiting for the aluminum plate to arrive.

In the final analysis, the KillerV concept worked well — it killed every full body spinner we ever fought ... well, we actually never drew a full body spinner in any competition!

The closest we ever came to was an overhead horizontal bar spinner which we did beat. Its performance against other robot designs was average, but not bad for our first combat robot! **SV**

PRODUCT REVIEW — Some Do it With Lead

● by Steven Kirk Nelson

When building larger robots, you run into larger problems. In any self contained machine, your run time is limited by the weight of the machine and the power required to get it moving. There are several types of battery technology — such as sealed lead acid, NiMH, and NiCad — to choose from. Depending on your parts budget, you have to decide on what you can afford. If you're on a low budget and can afford the weight, that leaves you with a simple choice. Do it with lead!

Sealed lead acid (SLA) batteries have been around for a long time and have proven to be mostly reliable. I've had pretty good luck with Powersonic SLAs and even better luck with Hawkers — which have a spill-proof construction and are safe to operate in any position. There is no need to add electrolyte and they have recombination systems to prevent hydrogen gas from escaping during recharging.

One of my favorite Powersonic models is the PS-12120 F2 which is a 12 volt, 12 amp hour model that weighs 8.5 lbs. It can be recharged at 2.4 amps. Another fav is the Powersonic PS-12180, also 12V with an 18Ah capacity, a weight of 13.1 lbs, and a charging amperage of 5.4A.

Both of these batteries work well in a middle to heavyweight robot when using two drivetrain motors up to one horsepower each. One of the limitations to these Powersonic batteries is that they are not designed to deliver much more than about 120 to 180A or so for more than a few seconds. This limits them to use with lower powered motors like wheel chair motors. Also, these batteries have to be charged slowly — around 20 to 30% of their Ah rating. Charging them faster than this shortens their lifespan or could even destroy them.

The Big Dogs

If you really want to use SLAs that can drive a powerful bot, you have a simple choice: Hawker Odyssey. Powersonics are probably the best standard SLAs I've tested, but the Odyssey batteries are a whole lot better when the current draws get high.

Hawker batteries are a unique type of SLA. They perform as both a high output cranking battery and a deep cycle battery. A couple of my favorite Odyssey models are the PC 535 (12V, 13 Ah, 535 cranking amps) for my heavyweight and the PC 925 (12V, 27 Ah, 925 cranking amps) for my superheavyweight.

The current you can source from a Hawker battery can be three times that of a normal SLA. I'll tell you what, I never believe anything until I do my own testing. So I tested some 13 Ah Powersonics in one of my heavyweight robots, running some high current drive motors — and it worked okay. I replaced the Powersonics with the Odyssey batteries and suddenly the robot would instantly smoke the tires almost constantly for three minutes. I've been using them ever since.

Another excellent feature of the Hawker batteries is you can *fast* charge them. They are designed to work with automotive charging systems, so they can take a lot of charging current from standard chargers. I wouldn't be afraid of feeding them 20-30 amps of charging current if I needed them pronto for a match — in fact, the manufacturer states there is *no* limit on charging current. And that's amazing!

Anyway, if you're on a budget, standard SLA batteries work pretty well. Odysseys work a whole lot better. What I'm saying, I'm saying with lead, partner! **SV**

Websites

www.enersysreservepower.com/default.asp

www.batterymart.com

EVENTS

UPCOMING — April and May

RattleBots Invitational — This event will take place on 4/14/2007 in Dorchester, WI. It's presented by WHRE. Trophies for all classes and cash prizes for all classes with three or more bots. Go to **www.rattlebots.com/** for more information.

Upstate NY Robot Battle V — This event will take place on 4/14/2007 in Syracuse, NY. It's presented by Upstate NY Robot Combat Club. Ants and Beetles fight in a 6' x 6' arena. Standard double elimination tournament; audience participation with house bots. Carousel Mall venue for lots of food and spectators. Go to **www.unyrcc.com** for more information.

Carolina Combat — This event will take place on 5/4/2007 through 5/5/2007 in Greensboro, NC. It's presented by Carolina Combat Robots. Robots from 150g Fairyweight to the 120lb Middleweights; big bot arena is a 16' x 16' steel structure with 1/4" steel floor and 1/2" Lexan for the walls. Go to **www.carolinacombat.com** for more information.

ComBots Cup 2007/Maker Faire — This event will take place on 5/19/2007 through 5/20/2007 in San Mateo, CA. It's presented by ComBots. Antweights through Superheavyweights, with a $10,000 Heavyweight prize pool and a $3,000 MiddleWeight pool! Go to **www.robogames.net** for more information. SV

Fill Your Basket With a Little Something Extra This Year!

DON'T WASTE ANY TIME!
Start your subscription to *SERVO Magazine* today!

Call 877-525-2539
or go online at
www.servomagazine.com

Give your Robot the power it deserves...
Introducing the
proSeed
microcontroller

only $89

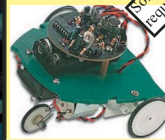
* 16 general purpose I/O's
* 16 analog inputs
* HUGE 128KB RAM
* 128KB EEPROM
* Super-Fast 70MHz SX CPU
* 24 pin compatible
* Programming languages
  16-bit ANS FORTH.
  C and Assembly available by third party.
* Visit our website for more...

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) **$129.**

Thousands more items on our web site!

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

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

ROBOT ON

The RC-100 series of electronic switches simply plug into an open channel of your servo controller. Use it to operate any robotic devices that require a simple on-off.

Capable of high currents up to 10A at 120V AC or 30V DC, and weighing in at 12 grams, turning on new opportunities are endless.

RCATS™

408-830-0745 | 3697 Enochs St, Santa Clara CA 95051
www.rcatsystems.com/servo

10 Year Olds Can Now Hand Solder ANY Surface Mount Component - Even BGA

AND SO CAN YOU!

Request a Free Sample Today

www.schmartboard.com

Seeing With OpenCV

Face Recognition With Eigenface

by Robin Hewitt

PART 4

Last month's article introduced Camshift — OpenCV's built-in face tracker. This month and next, this series concludes by showing you how to use OpenCV's implementation of eigenface for face recognition.

Face recognition is the process of putting a name to a face. Once you've detected a face, face recognition means figuring out whose face it is. You won't see security level recognition from eigenface. It works well enough, however, to make a fun enhancement to a hobbyist robotics project.

This month's article gives a detailed explanation of how eigenface works and the theory behind it. Next month's article will conclude this topic by taking you through the programming steps to implement eigenface.

What is Eigenface?

Eigenface is a simple face recognition algorithm that's easy to implement. It's the first face-recognition method that computer vision students learn, and it's a standard, workhorse method in the computer vision field. Turk and Pentland published the paper that describes their Eigenface method in 1991 (Reference 3, below). Citeseer lists 223 citations for this paper — an average of 16 citations per year since publication!

The steps used in eigenface are also used in many advanced methods. In fact, if you're interested in learning computer vision fundamentals, I recommend you learn about and implement eigenface, even if you *don't* plan to incorporate face recognition into a project! One reason eigenface is so important is that the basic principles behind it — PCA and distance-based matching — appear over and over in numerous computer vision and machine learning applications.

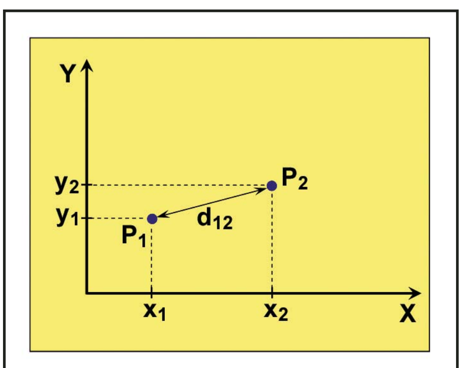
Here's how recognition works: Given example face images for each of several people, plus an unknown face image to recognize,

1) Compute a "distance" between the new image and each of the example faces.

2) Select the example image that's closest to the new one as the most likely known person.

3) If the distance to that face image is above a threshold, "recognize" the image as that person, otherwise, classify the face as an "unknown" person.

FIGURE 1. Euclidean distance, d_{12}, for two points in two dimensions.



How "Far Apart" Are These Images?

Distance — in the original eigenface paper — is measured as the point-to-point distance. This is also called Euclidean distance. In two dimensions (2D), the Euclidean distance between points P_1 and P_2 is

$d_{12} = \mathrm{sqrt}(\Delta x^2 + \Delta y^2)$,
where $\Delta x = x_2 - x_1$, and $\Delta y = y_2 - y_1$.

In 3D, it's $\mathrm{sqrt}(\Delta x^2 + \Delta y^2 + \Delta z^2)$. Figure 1 shows Euclidean distance in 2D.

In a 2D plot such as Figure 1, the dimensions are the X and Y axes. To get 3D, throw in a Z axis. But what are the dimensions for a face image?

The simple answer is that eigenface considers each pixel location to be a separate dimension. But there's a catch ...

The catch is that we're first going to do something called *dimensionality reduction*. Before explaining what that is, let's look at why we need it.

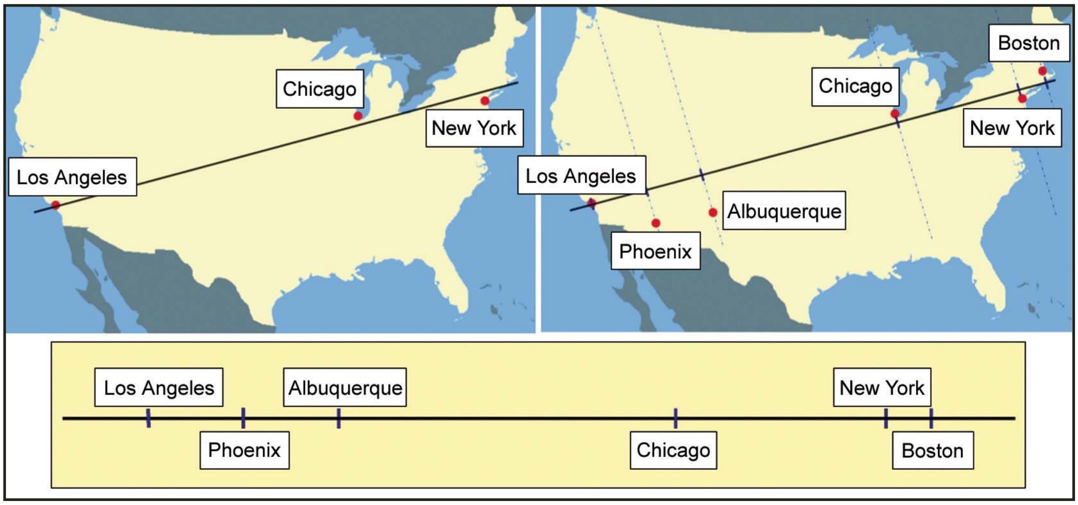
Even a small face image has a lot of pixels. A common image size for face recognition is 50 x 50. An image this size has 2,500 pixels. To compute the Euclidean distance between two of these images, using pixels as dimensions, you'd sum the square of the brightness difference at each of the 2,500 pixel locations, then take the square root of that sum.

There are several problems with this approach. Let's look at one of them — signal-to-noise ratio.

Noise Times 2,500 is a *Lot* of Noise

By computing distance between face images, we've replaced 2,500 differences between pixel values with a single value. The question we want to consider is, "What effect does noise

FIGURE 2. Right: Fitting a line to three points is a special case of PCA. Left: To project points from the 2D map onto the 1D line, locate the point on the line that's closest to the 2D point. Bottom: The 1D subspace, and the distances between points in this subspace.



have on this value?"

Let's define noise as anything — other than an identity difference — that affects pixel brightness. No two images are exactly identical, and small, incidental influences cause changes in pixel brightness. If each one of these 2,500 pixels contributes even a small amount of noise, the sheer number of noisy pixels means the total noise level will be very high.

Amidst all these noise contributions, whatever information is useful for identifying individual faces is presumably contributing some sort of consistent signal. But with 2,500 pixels each adding some amount of noise to the answer, that small signal is hard to find and harder to measure.

Very often, the information of interest has a much lower dimensionality than the number of measurements. In the case of an image, each pixel's value is a measurement. Most likely, we can (somehow) represent the information that would allow us to distinguish between faces from different individuals with a much smaller number of parameters than 2,500. Maybe that number is 100; maybe it's 12. We don't claim to know in advance what it is, only that it's probably much smaller than the number of pixels.

If this assumption is correct, summing all the squared pixel differences would create a noise contribution that's extremely high compared to the useful information. One goal of dimensionality reduction is to tone down the noise level, so the important information can come through.

Dimensionality Reduction by PCA

There are many methods for dimensionality reduction. The one that eigenface uses is called *Principal Components Analysis* — PCA for short.

Line Fitting and PCA

To get an intuition for what PCA does, let's look at a special case of PCA called a "least squares line fit." The lefthand side of Figure 2 shows an example of fitting a line to three points: the 2D map locations for Los Angeles, Chicago, and New York. (To keep the explanation simple, I've ignored 3D factors such as elevation and the curvature of the Earth.)

These three map points are almost — but not quite — on a single line already. If you were planning a trip, that relationship would be useful information. In that sense, a single line expresses something essential about their relationship. A line has only one dimension, so if we replace the points' 2D locations with locations along a single line, we'll have reduced their dimensionality.

Because they're almost lined up already, a line can be fitted to them with little error. The error in the line fit is measured by adding together the square of the distance from each point to the line. The best-fit line is the one that has the smallest error.

Defining a Subspace

Although the line found above is a 1D object, it's located inside a larger, 2D space, and has an orientation (its slope). The slope of the line expresses something important about the three points. It indicates the direction in which they're spread out the most.

If we position a rectangular (x,y) coordinate system so that its origin is somewhere on this line, we can write the line equation as simply

$y = \mathbf{m}x$,

where **m** is the line's slope: $\Delta y/\Delta x$.

When it's described this way, the line is a *subspace* of the 2D space defined by the (x,y) coordinate system. This description emphasizes the aspect of the data we're interested in, namely the direction that keeps these points most separated from one another.

The PCA Subspace

This direction of maximum separation is called the first principal component of a dataset. The direction with the next largest separation is the one perpendicular to this. That's the second principal component. In a 2D dataset, we can have at most two principal components.

Since the dimensionality for images is much higher, we can have more principal components in a dataset made up of images.

However, the number of principal components we can find is also limited by the number of data points. To see why that is, think of a dataset that consists of just one point. What's the direction of maximum separation for this dataset? There isn't one, because there's nothing to separate. Now consider a dataset with just two points. The line connecting these two points is the first principal component. But there's no second principal component, because there's nothing

FIGURE 3. Left: Face images for 10 people. Right: The first six principal components viewed as eigenfaces.

more to separate: both points are fully on the line.

We can extend this idea indefinitely. Three points define a plane, which is a 2D object, so a dataset with three data points can never have more than two principal components, even if it's in a 3D, or higher, coordinate system. And so on.

In eigenface, each 50 x 50 face image is treated as one data point (in a 2,500 dimensional "space"). So the number of principal components we can find will never be more than the number of face images minus one.

Although it's important to have a conceptual understanding of what principal components are, you won't need to know the details of how to find them to implement eigenface. That part has been done for you already in OpenCV. I'll take you through the API for that in next month's article.

Projecting Data Onto a Subspace

Meanwhile, let's finish the description of dimensionality reduction by PCA. We're almost there!

Going back to the map in Figure 2, now that we've found a 1D subspace, we need a way to convert 2D points to 1D points. The process for doing that is called projection. When you project a point onto a subspace, you assign it the subspace location that's closest to its location in the higher dimensional space. That sounds messy and complicated, but it's neither. To project a 2D map point onto the line in Figure 2, you'd find the point on the line that's closest to that 2D point. That's its projection.

There's a function in OpenCV for projecting points onto a subspace, so again, you only need a conceptual understanding. You can leave the algorithmic details to the library.

The blue tic marks in Figure 2 show the subspace location of the three cities that defined the line. Other 2D points can also be projected onto this line. The righthand side of Figure 2 shows the projected locations for Phoenix, Albuquerque, and Boston.

Computing Distances Between Faces

In eigenface, the distance between two face images is the Euclidean distance between their projected points in a PCA subspace, rather than the distance in the original 2,500 dimensional image space. Computing the distance between faces in this lower dimensional subspace is the technique that eigenface uses to improve the signal-to-noise ratio.

Many advanced face recognition techniques are extensions of this basic concept. The main difference between eigenface and these advanced techniques is the process for defining the subspace. Instead of using PCA, the subspace might be based on Independent Component Analysis (ICA) or on Linear Discriminant Analysis (LDA), and so on.

As mentioned above, this basic idea — dimensionality reduction followed by distance calculation in a subspace — is widely used in computer vision work. It's also used in other branches of AI. In fact, it's one of the primary tools for managing complexity and for finding the patterns hidden within massive amounts of real world data.

Picturing the Principal Components

In our definition of a line as a 1D subspace, we used both *x* and *y* coordinates to define **m**, its 2D slope. When **m** is a principal component for a set of points, it has another name. It's an eigenvector. As you no doubt guessed, this is the basis for the name "eigenface." Eigenvectors are a linear algebra concept. That concept is important to us here only as an alternative name for principal components.

For face recognition on 50 x 50 images, each eigenvector represents the slope of a line in a 2,500 dimensional space. As in the 2D case, we need all 2,500 dimensions to define the slope of each line. While it's impossible to visualize a line in that many dimensions, we can view the eigenvectors in a different way. We can convert their 2,500 dimensional "slope" to an image simply by placing each value in its corresponding pixel location. When we do that, we get facelike images called — guess what — eigenfaces!

FIGURE 4. Face images from two individuals. Each individual's face is displayed under four different lighting conditions. The variability due to lighting here is greater than the variability between individuals.

Eigenface tends to confuse individuals when lighting effects are strong.

Eigenfaces are interesting to look at, and give us some intuition about the principal components for our dataset. The lefthand side of Figure 3 shows face images for 10 people. These face images are from the Yale Face Database B (References 4 and 5). It contains images of faces under a range of lighting conditions. I used seven images for each of these 10 people to create a PCA subspace.

The righthand side of Figure 3 shows the first six principal components of this dataset, displayed as eigenfaces. The eigenfaces often have a ghostly look, because they combine elements from several faces. The brightest and the darkest pixels in each eigenface mark the face regions that contributed most to that principal component.

Limitations of Eigenface

The principal components that PCA finds are the directions of greatest variation in the data. One of the assumptions in eigenface is that variability in the underlying images corresponds to differences between individual faces. This assumption is, unfortunately, not always valid. Figure 4 shows faces from two individuals. Each individual's face is displayed under four different lighting conditions.

These images are also from the Yale Face Database B. In fact, they're face images for two of the 10 people shown in Figure 3. Can you tell which ones are which? Eigenface can't. When lighting is highly variable, eigenface often does no better than random guessing would.

Other factors that may "stretch" image variability in directions that tend to blur identity in PCA space include changes in expression, camera angle, and head pose.

Figure 5 shows how data distributions affect eigenface's performance. The best case for eigenface is at the top of Figure 5. Here, images from two individuals are clumped into tight clusters that are well separated from one another. That's what you hope will happen. The middle panel in Figure 5 shows what you hope *won't* happen. In this panel, images for each individual contain a great deal of variability. So much so, that they've skewed the PCA subspace in a way that makes it impossible for eigenface to tell these two people apart. Their face images are projecting onto the same places in the PCA subspace.

In practice, you'll probably find that the data distributions for face images fall somewhere in between these extremes. The bottom panel in Figure 5 shows a realistic distribution for eigenface.

Since the eigenvectors are determined only by data variability, you're limited in what you can do to control how eigenface behaves. However, you can take steps to limit, or to otherwise manage, environmental conditions that might confuse it. For example, placing the camera at face level will reduce variability in camera angle.

Lighting conditions — such as side lighting from windows — are harder for a mobile robot to control. But you might consider adding intelligence on top of face recognition to compensate for that. For example, if your robot knows roughly where it's located, and which direction it's facing, it can compare the current face image only to ones it's seen previously in a similar situation.

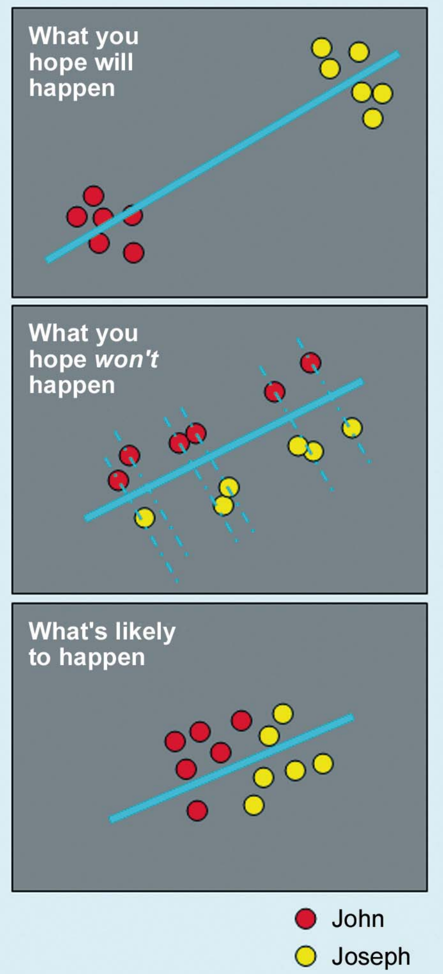
Even highly-tuned commercial face recognition systems are subject to cases of mistaken identity. In fact, part of the challenge of incorporating face recognition into any robotics application is finding ways to accommodate these.

Coming Up

Next month's article concludes this series by taking you step-by-step through a program that implements eigenface with OpenCV.

Be seeing you! **SV**

FIGURE 5. How data distributions affect recognition with eigenface. Top: Best-case scenario — data points for each person form tight, well separated clusters. Middle: Worst-case scenario — variability in the face images for each individual is greater than the variability between individuals. Bottom: A realistic scenario — reasonable separation, with some overlap.



References and Resources

- *OpenCV on Sourceforge* **http://sourceforge.net/projects/opencvlibrary**
- *Official OpenCV usergroup* **http://tech.groups.yahoo.com/group/OpenCV**
- Turk, M., Pentland, A., *Face recognition using eigenfaces*, Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1991.
- *Yale Face Database B* **http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html**
- A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, D.J.,*From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose*, IEEE Trans. Pattern Anal. Mach. Intelligence, 2001.

A SIMPLE INERTIAL NAVIGATOR

by David Avikasis, Shachar Braver, and Shlomo Engelberg

In 1963, Robert Heinlein published *Podkayne of Mars*. In *Podkayne,* Heinlein has his heroine use a portable inertial navigation system (which Heinlein refers to as an "inertial tracker"). Ever since reading about it, trying to build such a system has been a desideratum for author Shlomo Engelberg.

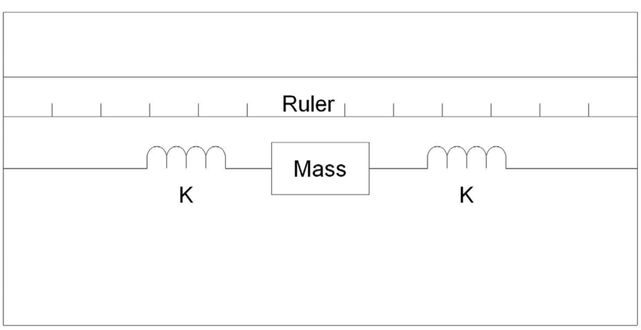
A year and a half ago, we started designing and building an inertial navigation system using simple and relatively inexpensive parts. As you will see, there is no "royal road" to building a simple and inexpensive inertial navigation system — even in 2006.

What Is Inertial Navigation?

From time, immemorial people have needed to measure their position without making use of landmarks. A classical example of such a measurement technique is *dead reckoning,* as practiced hundreds of years ago. In order to calculate where a ship had gone, you would mark down where the ship left from, and you would record the direction and speed of the ship at regular intervals. Once the directions and speeds were known, it was possible to work out the ship's position. (See the sidebar for more about shipboard speed measurements in the good old days.)

Dead reckoning requires a knowledge of the speed at all times. It has long been understood that it is impossible to measure a constant speed without an outside reference. (See the article on *Galilean invariance* in for more information about this topic.) Acceleration, however, is another story. Even a person sealed in a windowless box can detect acceleration — and so can electronic measurement devices.

FIGURE 1. The block diagram of a simple accelerometer.



How Do Accelerometers Work?

A spring-mass system is a simple accelerometer. If you take a box, connect two springs to a mass, and put a ruler alongside the mass, you have a simple (single-axis) accelerometer (Figure 1.) When the box is accelerated along the axis of the spring-mass system, it "drags" the mass with it. The springs connected to the mass deform to apply the necessary force. Assuming that the springs are linear — that they obey Hooke's Law — the displacement of the mass is proportional to the acceleration in the direction of the axis of the spring-mass system. If you want to measure the acceleration in the x, y, and z planes, then you need three accelerometers oriented 90 degrees to each other.

It is possible to implement tiny accelerometers as Microelectromechanical Systems (MEMS). These MEMS accelerometers are very small spring-mass systems that are similar to the systems described previously. Here, however, the measurement of the mass' displacement is made in an interesting way. Connected to the mass is a fin, and the fin's position alters a variable capacitor. Thischange in capacitance is measured and used to calculate the instantaneous acceleration. (See Doscher for more information on accelerometers in general, and MEMS accelerometers in particular.) The chip that we used — the ADXL202 — is a MEMS device that has two on-board accelerometers that are mounted at right angles to one another. The chip has quite a bit of on-board signal processing circuitry that takes care of outputting a useable signal.

Measuring Speed the Old-Fashioned Way

How was speed measured in ships in the days long before electronics? A sailor would throw a chip log attached to a knotted rope from the boat's stern. One sailor would count off knots while another sailor kept track of the time that passed. Because the knots were placed at regular, known intervals and because the chip log remained relatively stationary in the water, the sailors could use the number of knots per second to calculate the ship's speed. (This is thought to be the origin of the knot used to measure a ship's speed.) Casting the log is described in one of the books in C. S. Forester's *Hornblower* series — a wonderful series describing the life of Horatio Hornblower, a member of the Royal Navy in the late 1700s and early 1800s.

Distance From Acceleration

Now that we know how to calculate the acceleration in a particular direction, we need to see how we can calculate the distance and direction traveled from the acceleration. Displacement, $\vec{s}$, is the vector that gives the distance traveled and the direction in which the distance was traveled. Displacement is the double integral of acceleration, $\vec{a}$; and the single integral of velocity, $\vec{v}$. On the one hand, it is easy to write:

$$\vec{s} = \int_0^t \int_0^\tau \vec{a}(y)\,dy\,d\tau + \vec{v}(0)t + \vec{s}(0)$$

On the other hand, there are a host of practical problems connected to measuring the displacement of an object using the signal that is output by the accelerometer.

A given spring-mass system can only measure acceleration in one dimension. In order to measure acceleration in more directions, you need more accelerometers. Also, you need to know what direction the accelerometer is pointing in. (In a moving object, this can be quite a challenge.) There are a variety of ways of keeping track of this direction.

One way is to mount the accelerometers on a platform whose orientation you keep fixed. A traditional method of achieving this goal is to mount the platform on gimbals and to use gyroscopes and motors to control the position of the platform in such a way that the accelerometer is kept in a fixed orientation. (For more information on this topic, see [King].) In our project, we put the responsibility for maintaining the orientation of the accelerometer on the user. The user is expected to move the system without changing its orientation.

Our Implementation

The inertial navigation system we implemented measures the distance traveled in two dimensions. The system assumes that it is located on a level surface, and the system measures the distance traveled in the plane. In implementing this system, we made use of relatively inexpensive parts: a microcontroller, a dual-axis accelerometer, and a serial LCD display. The microprocessor we used — the ADuC841 — is a one-clock 8052-type microprocessor. It is capable of 20 MIPS; it has an on-board ADC that is capable of sampling at up to 400 thousand times/second; and it costs under $7 in quantity. It also has some internal/external RAM — RAM that is internal to the chip but is defined as external RAM from the point-of-view of the core; it is a nice chip to work with. Because of its

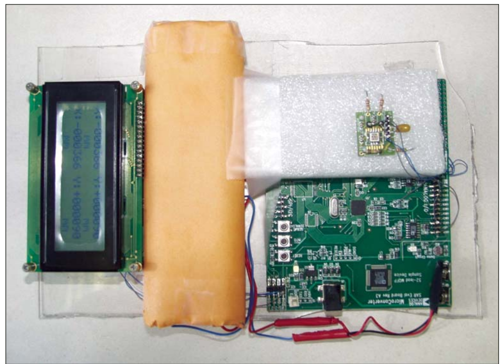
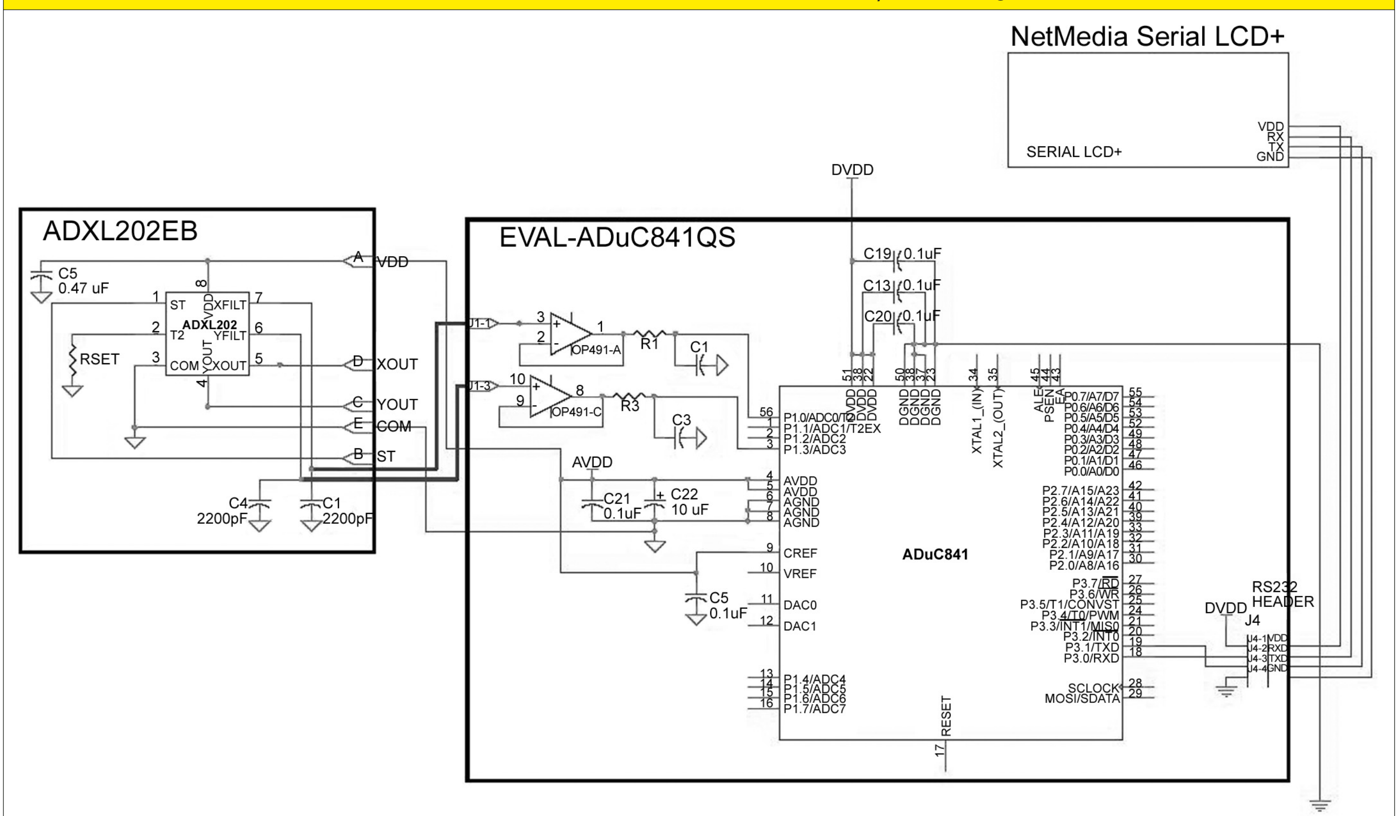


FIGURE 2. The final system.

FIGURE 3. The schematic diagram of the final system. Many of the connections on the EVAL-ADuC841QS board are not shown here. For the full schematic of the unmodified board, see Analog Devices' website.



Building a Digital Level

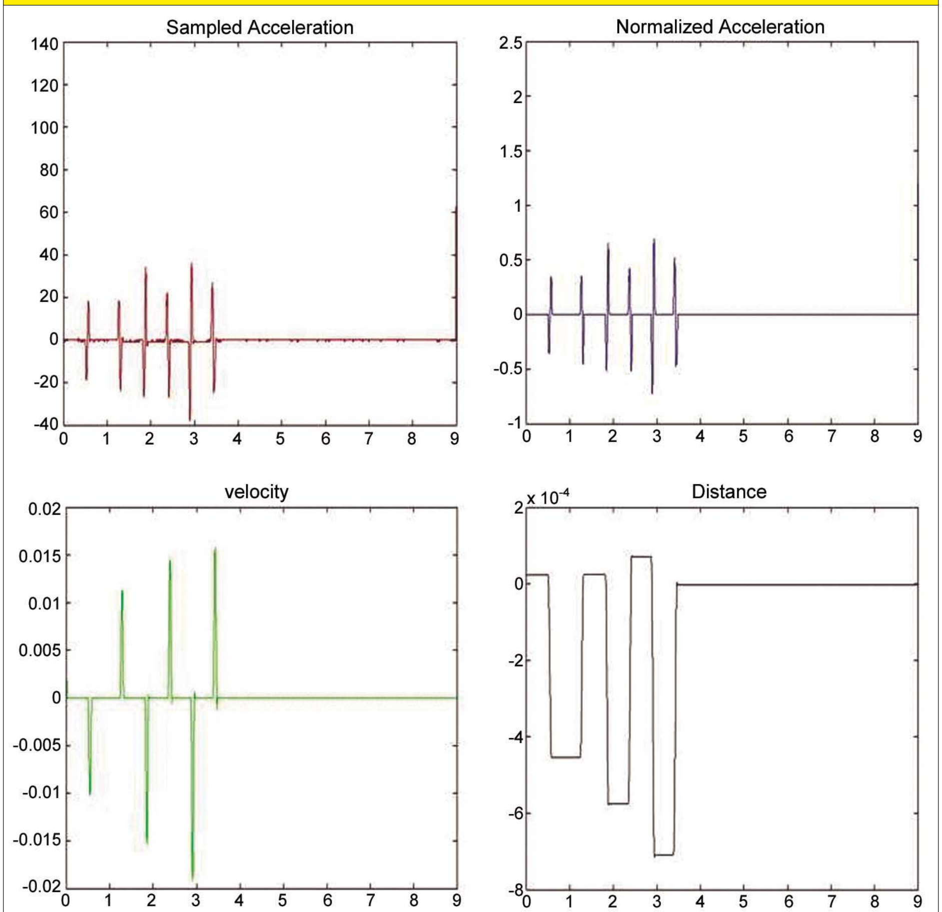
Because our accelerometer is sensitive to the slope at which it is placed, the accelerometer can be used as the basis for a "smart level." This is a very simple project to put together. In order to implement a level, all that you need to do is place the system on a level surface once, let the microcontroller record the values it reads from the two accelerometers, and then use these baseline values to determine whether the system is level at a later time. In fact, you can have the system tell you which way a surface must be tilted in order to make it level. We have implemented this idea using the same hardware that was used for the more complicated inertial navigation system. (The code necessary to implement the digital level is available on the *Nuts & Volts* website at **www.nutsvolts.com**.)

"blinding" speed (relative to earlier models like the ADuC812), it is possible to do some processing of the sampled values on the fly. (See the analog microcontrollers section on **www.analog.com** for more information about this family of processors.) Rather than using the bare bones microcontroller, we used Analog Devices' evaluation kit for the microcontroller (the EVAL-ADuC841QS). This board leads out most of the important pins on the microcontroller; the serial cable includes level shifting circuitry to allow you to make use of the UART with ease, and the board has buffers for many of the important analog signals.

Programming an LCD display can be somewhat tedious at times. We chose to take the easy way out by using NetMedia's Serial LCD+ (Ver. 1.5). (See their website at **www.netmedia.com** for more information on their product lines.) This LCD is controlled using simple commands sent by the 8052-standard UART.

We also made use of the ADXL202EB dual-axis accelerometer evaluation board (Analog Devices was kind enough to donate this to the Jerusalem College of Technology). It made it simple for us to interface to the accelerometer. See Figure 2 for a photo of the finished product and Figure 3 for a schematic of the system.

FIGURE 4. In the upper left-hand graph, the almost unprocessed measurements are shown. In the upper right-hand graph, the normalized acceleration is given. In the lower left-hand graph, the velocity is given (in arbitrary units). In the lower right-hand graph, the position is given (in arbitrary units).



Problems and Solutions

The ADXL202 produces a PWM signal as its natural output, but it also has an analog output. The evaluation board makes it simple to access the analog signal that is proportional to the acceleration. As the ADuC841 has an onboard analog to digital converter (ADC), it was more convenient and efficient to use the analog output. By carefully adding a wire to the development kit, we were able to pick off the voltage we needed from the ADXL202 chip and use the voltage as the input to the ADuC841's easy-to-use ADC.

In most places, very few surfaces are truly level. Though having a slight slope does not annoy people very much and may not even be noticeable, it can make an accelerometer quite confused. Given a slight slope, an

Bibliography

• [Doscher] J. Doscher, "Accelerometer Design and Applications," available at **www.analog.com/Analog_Root/static/library/techArticles/mems/sensor971.pdf**; last visited 6 July 2006.

• [Heinlein] R. A. Heinlein, *Podkayne of Mars*, Ace Books, 2005.

• [King] A. D. King, "Inertial Navigation — Forty Years of Evolution," *GEC Review*, Vol. 13, No. 3, 1998, available at **www.imar-navigation.de/download/inertial_navigation_introduction.pdf**; last visited 6 July 2006.

• [Wal] K. J. Walchko, M. C. Nechyba, E. Schwartz, and A. Arroyo, "Embedded Low Cost Inertial Navigation System," Proceedings of the 16th Florida Conference on Recent Advances in Robotics, 2003, available on-line at **www.mil.ufl.edu/publications/fcrar03/Walchko-2.pdf**; last visited 12 July 2006.

accelerometer will believe that it is undergoing a constant acceleration. (See the sidebar, however, for a way to make use of the accelerometer's sensitivity to slopes.)

A second problem we faced was that while moving — and even while moving with a constant velocity — the accelerometer's output is generally noisy. (Figure 4 shows a sample of the accelerometer's output and plots of the velocity and distance derived from the measured acceleration.)

The effects of the noise were mitigated somewhat by filtering the output of the accelerometer. Because of the speed of the ADuC841, we were able to sample the accelerometers output quickly and average 16 samples before doing any further processing.

The tilting problem was handled in two ways. First of all, when the system is turned on, it measures the average value of the acceleration. This average is stored and is used as a baseline. As long as the unit is placed properly to begin with, this calibration should take care of much of the problem.

There is a second problem, however. After being moved, the unit will not generally be perfectly aligned anymore. The slight angle leads to a small, constant acceleration. We made use of the noise that is seen when the object is moving in order to take care of this problem. We found that when in motion, the output of the accelerometer attains high values on a regular basis. We decided that if we did not see such high values, then the system was not in motion; when the measured acceleration is consistently small, we set $\vec{a} = 0$ and $\vec{v} = 0$. This technique works pretty well, though if the accelerometer is kept at a steep enough angle, the measured values of the acceleration will make it past the threshold and the system will think that it is being accelerated.

Another challenge we faced was performing all the calculations on a small 8052-based processor like the ADuC841. Because we did a fair amount of computing, we decided to do all of our programming in C. For this purpose, we use the Keil uVision3 development environment. (The uVision3 software can be downloaded from the Keil website at **https://www.keil.com/demo/eval/c51.htm**.) We tried to avoid floating point numbers because of the computational effort — the processor time — they require.

We performed the numerical integrations of the acceleration by calculating the iterated cumulative sum of the (already averaged) measured acceleration. Though this is not, in principle, the most accurate way of computing the acceleration, it is the simplest way (which is a consideration with a small microprocessor). Additionally, we know that there are many relatively large sources of error (whether it is the small slopes that abound or the noise that is produced by the motion of system). The numerical noise added by the simple method of integration is the least of our worries.

Our final unit measures distance in millimeters. We have found that under relatively optimal conditions, the system is accurate to within 10 to 15 percent. It is inexpensive, easy to understand, and fun, but it is probably not accurate enough for use as a stand-alone inertial navigation system. (The code for this project can be downloaded from the *Nuts & Volts* website at **www.nutsvolts.com**.) See the sidebar for places where somewhat more complicated — but still not terribly accurate — systems are used. SV

Inertial Navigation and the Global Positioning System (GPS)

As a rule, inexpensive inertial navigation systems do not give very accurate long-term results. With the advent of GPS-based systems, a need for short-term accurate systems has appeared. Using GPS, you can determine your position pretty accurately — most of the time. If for some reason you cannot receive the GPS radio transmissions, then you can no longer update your position. In such cases, an inexpensive — though not terribly accurate — inertial navigation system can take over. The system will continue to update your position for the (hopefully short) period that you cannot receive GPS signals. For more information on such systems, see [Wal].

About the Authors

• *David Avikasis is currently completing his Bachelor of Science degree in electronics engineering at the Jerusalem College of Technology — Machon Lev. He can be contacted at david.avikasis@gmail.com*

• *Shachar Braver is currently completing his Bachelor of Science degree in electronics engineering at the Jerusalem College of Technology — Machon Lev. He can be contacted at shachar.braver@gmail.com*

• *Shlomo is chairman of the electronics department of the Jerusalem College of Technology. He can be contacted at shlomoe@jct.ac.il*

CHART RECORDER BOTS

by Monty Reed


FIGURE 1. An assortment of chart recorders.

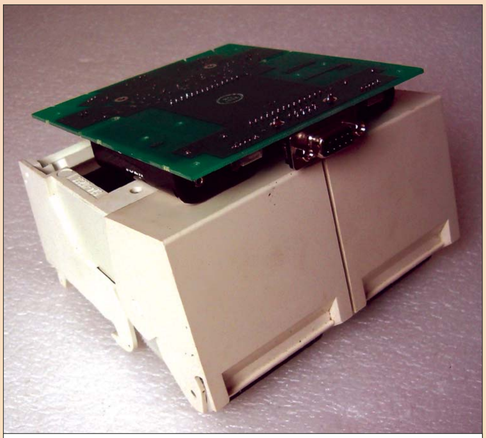


FIGURE 2. The charter Tabletop Charter Bot.

FIGURE 3. Close-up of the motor and gears that can be harvested from chart recorders.

FIGURE 4. Close-up of a 25 pin connector.

One of the first rules for hobby robotics is to save money. The best way to enforce that rule is remembering, the best price is free. While attending one of Bill Beaty's Weird Science Salon meetings on the first Friday of the month at Seattle's Museum of the Mysteries, I found treasure. Many of the science buffs that attend the monthly social event bring goodies to give away (see Figure 1).

Newcomers to the meeting may have looked at the piles of stuff and wondered, where are the goodies? Thing is, much of the wealth can only be discovered by looking inside each of the items. I found a few boxes of chart recorders. Everyone was taking a couple and when each one had taken all they wanted, some were left over and I was able to cart them home.

The box contained a few different types. Some were from types of medical equipment. A lot of medical equipment is taken out of service far before it's worn out for the safety of the people involved with the equipment. Bottom line, most surplus medical equipment still has a lot of life left for robotics use.

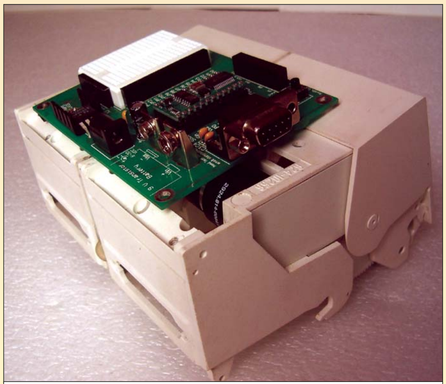


FIGURE 5. Tabletop Charter Bot with BASIC Stamp microcontroller.

Figure 2 shows Tabletop Charter Bot which was the first (or charter) chart recorder robot. Notice it is made from two chart recorders and a BASIC Stamp sumo bot board. Chart recorders have excellent motors and gears (see Figure 3). This lot came with nice 25 pin connectors (see Figure 4); one of my favorite connectors for robotics due to the abundance of printer cables in thrift stores and garage sales.

One of the simple ways to use these chart recorders as robots is to modify the paper roll door so that the paper roller becomes a wheel. Next, modify a second chart recorder and attach the two together. For a simple tabletop robot, this will allow for the tank-type steering I talked about in my article in the January '07 issue. Since the bot presented here will be a tabletop chart bot, no rear caster will be needed; the back of the robot will slide around on the table. If you choose to use the robot on different terrain (other than a tabletop), you can add a rear caster.

In the final step we add the BASIC Stamp control board to the top of the robot (see Figure 5) and run a simple program. I found some TAB Electronics sumo robot PCBs (printed

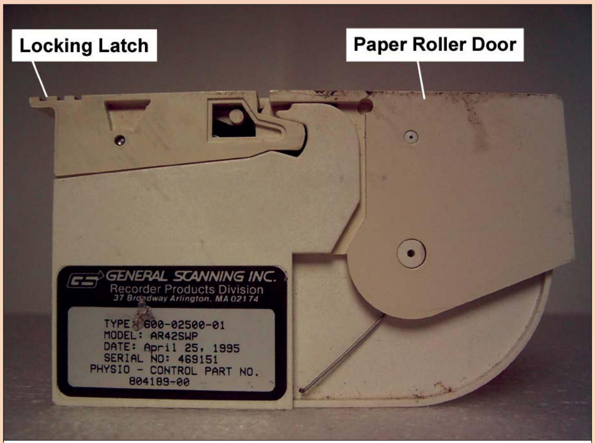


FIGURE 6. Profile of chart recorder with callouts: locking latch, door.

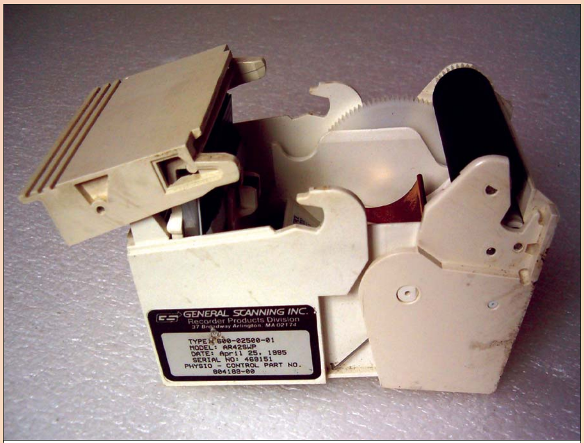


FIGURE 7. Chart recorder with door open callouts: roller (wheel), gears, motor.

circuit boards) #ISBN 0-07-141264-6 that were on sale a few months ago, and bought four to have on hand just for a project like this. These were the PCBs from a sumo kit that I had bought last year and it worked great. The PCBs have a BASIC Stamp with two motor control pinouts that just happen to fit the chart recorder motor connectors. If you do not have this PCB, you can use a Board of Education (BOE) from Parallax.

The chart recorder I chose for this project was made by General Scanning, Inc., type 600-02500-01, Model AR42SWP. By pulling up on the locking latch (see Figure 6) and opening the door, the paper compartment is exposed. If the spring is still working, the door will pop open revealing the gears, roller, and motor (see Figure 7). Push the locking latch back into the locked position and turn the chart recorder on its side.

To activate the wheel function, all you have to do is lock the door open. The three things I use to do this are: tape, set screws, and hot glue. Hold the door in the open position and place some tape on the side. Next, drill a setscrew hole in the top (see Figure 8), screw in a self-tapping screw (see Figure 9), and run a bead of hot glue along the top (see Figure 10). Let the hot glue cool.

Chart recorder motors can be run from the main connector or you can power the motors directly. On this chart recorder, pins 13 and 12 on the 25-pin connector power the motor. For this project, we will drive the motor directly so you will need to disconnect it from the chart recorder. Turn the chart recorder over (see Figure 11) and examine the motor connector. If it has a protective plastic sleeve over the connector, grab hold of it with needle nose pliers and slide it back off of the motor connector onto the wire. Disconnect the motor connector.

Repeat these steps with another chart recorder.

Place the two chart recorders side by side (see Figure 12). Connect the two together using some tape. After the two chart

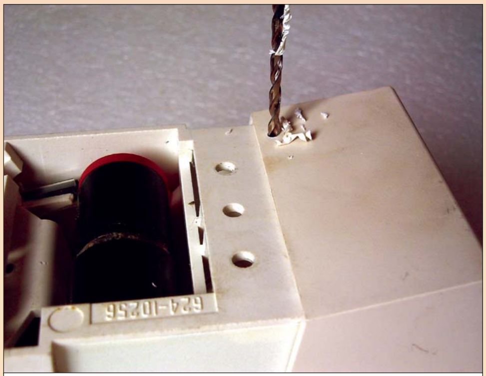


FIGURE 8. Setscrew hole.

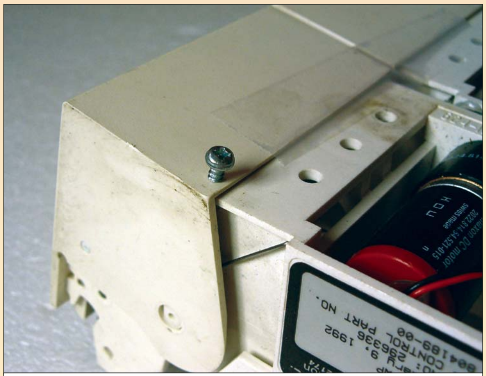


FIGURE 9. Self-tapping screw.

FIGURE 12. Two chart recorders side by side.

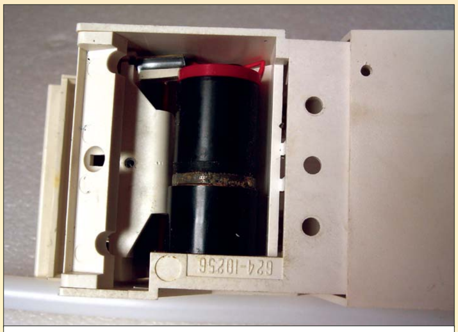


FIGURE 11. Chart recorder upside down showing the motor and the motor connector callouts.

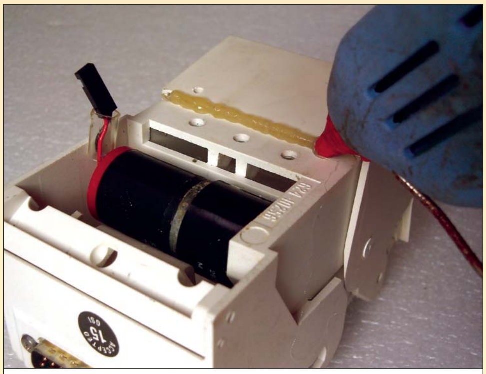


FIGURE 10. Bead of hot glue.

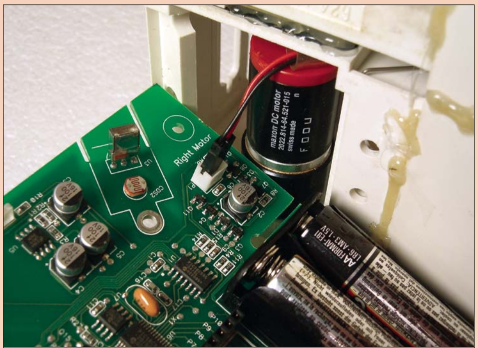


FIGURE 13. Charter Bot on its side with one motor connected.

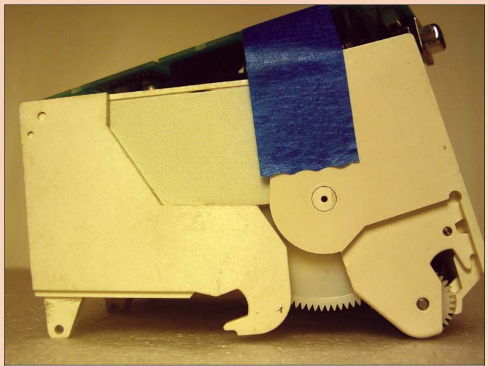


FIGURE 14. Charter Bot close-up profile.

recorders are secured together, heat up the hot glue gun and start running some beads to glue them together.

What You'll Need

- Hot Glue Gun
- Needle Nose Pliers
- Tape
- Drill and Drill Bit (matching self-tapping screws)
- Self-tapping Screws
- Surplus Chart Recorders
- TAB Sumo Bot Control Board or Parallax Board of Eduction.

You can remove the tape as you go or leave it in place. To make the connection more secure, you can add some metal brackets and self-tapping screws to be sure the two chart recorders stay connected.

Place the batteries into the control board. Turn the Tabletop Charter Bot on its side and connect the motor connectors to the motor pinouts marked "left motor" and "right motor"(see Figure 13). Using some tape, connect the control board to the sides of the Charter Bot. Plug in the controller cable and download a simple program available from Parallax or TAB.

From TAB Electronics website (**www.tabrobotkit.com/pdfs/robot anatomy.pdf**), I was able to download the schematics. I downloaded the software from TAB, as well (**www.tabrobotkit.com/code/robot092.asm**). **SV**

Contact the Author

For more information on this and other projects, check out ***www.servomagazine.com*** *or* ***www.theyshallwalk.org****. You can contact the author directly at lictonsprings@yahoo.com or through mail at Monty Reed, They Shall Walk, 6201 15th Ave. NW #542, Seattle, WA 98107.*

ROBOBusiness CONFERENCE & EXPO 2007

The International Business Development Event for the Mobile Robotics & Intelligent Systems Industry

May 15-16, 2007
Hynes Convention Center
Boston, MA

Conference Tracks

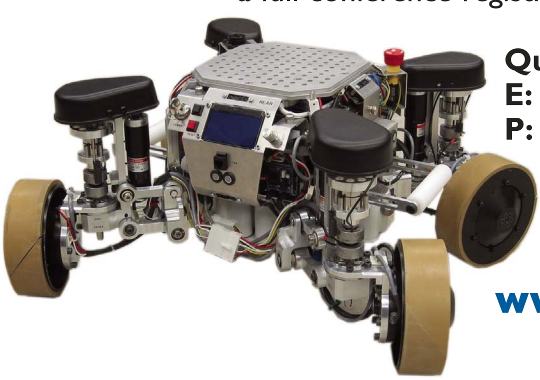
- Business Development and Investment
- Technology and Standards
- Applications and Products
- New Markets for Industrial Robotics
- Application and Products
  - First Responder Robotics
  - Security and Defense
  - Intelligent Transportation & Field Robotics
  - Healthcare Robotics
  - Consumer Robotics

Now in its fourth year, the RoboBusiness Conference and Exposition is the 'must attend' event for the mobile robotics and intelligent systems industry.

- Two full days of sessions and keynotes in 5 tracks featuring 50+ world class speakers from business, government and academia
- Expo floor featuring the latest robots and robotic technology from around the world – many never seen in North America
- Special events, networking receptions, Robot Hall of Fame and Microsoft Robotics Studio luncheons and more!

REGISTER EARLY AND SAVE

Register online at www.roboevent.com and use Priority Code RBCD200 to save $200 off the price of a full conference registration.



Questions?
E: info@roboticstrends.com
P: 508-663-1500 x329

www.roboevent.com

Founding Sponsor
iRobot

Produced by
roboticstrends

Premier Sponsors
THE TECHNOLOGY COLLABORATIVE
Accelerating Digital & Robotic Innovation.
MASS TECHNOLOGY LEADERSHIP COUNCIL

Gold Sponsors
CoroWare
evolution robotics
freescale semiconductor
Holland+Knight
Microsoft

Corporate Sponsors & Exhibitors
Advanced Digital Logic
ATI Industrial Automation
Autonomous Solutions
Boston Engineering
Carnegie-Mellon University
City of Osaka
Energid
Foster-Miller
Galileo Mobility Instruments
Gears Educational Systems
Global Emerging Trade Organization
GOSTAI
Hobby Manufacturers Association
Hokuyo Automatic
Honeywell
Hydroid
InterSense
InTouch Health
IZI Robotics
Japan External Trade Association
Kumotek
MachineBus
Magmotor
Merlin Systems
NeuroRobotics
NEXSTAR
Object Management Group
OC Robotics
OceanServer Technology
OLogic
Perrone Robotics
Robosoft
Robot Hall of Fame
RoboTech Center
Robotics Society of America
SCHUNK
Segway
Strategy Analytics
Valde Systems
Vecna Robotics.

Media Sponsors
SERVO MAGAZINE
Nuts & Volts

ROBOT SIMULATION: Systems-Level Applications

by Bryan Bergeron

A holistic view of robot design and development blurs the boundary between abstract behavior and the form, composition, and construction of the physical platform. While a single roboticist can often use this approach to develop a relatively simple robot, it soon becomes unwieldy as the level of complexity increases. At some point between a carpet roamer and a fully autonomous robot, most roboticists benefit from thinking of a robot as a collection of interdependent systems, and of robot design as the straightforward orchestration of these systems.

Working with robotics at the systems level may involve determining the optimal number and placement of sonar sensors on a hexapod, defining the bandwidth requirements of a robot-PC communication link, or tuning the motor controller of a dual drive robot. While trial and error may be an acceptable approach for simple challenges such as determining optimum rangefinder sensor placement, a more formal engineering approach is usually appropriate when the components are expensive, time is limited, or when simply 'acceptable' performance isn't good enough.

The cornerstone of modern robotics engineering is to use modeling and simulation to determine the best system-level configuration in software — before trying it on a real robot. This article, the third in a series on the uses of simulation in robotics, focuses on system-level applications of the technology. Using motion control as an example, it describes how you can use affordable software tools to prototype new robot systems and improve upon your existing designs.

Systems Simulation

A systems-level perspective is useful because it can often provide insight into improvements and help diagnose problems. Consider the challenge of designing a semi-autonomous, wheeled robot with a fused sensor network, motion control in hardware and software, onboard machine vision, voice recognition, a robot arm, a collision avoidance system, energy management hardware and software, and an RF communications link to a PC. Thinking in terms of systems not only partitions a potentially overwhelming challenge into manageable components, but the practice lends itself to modeling and simulation.

Simulation involves first identifying the key, quantifiable variables, such as the useable range, bandwidth, power, noise immunity, and sensitivity requirements of a robot-PC RF communications link. Once the variables have been identified, the next step is to develop the mathematical models that incorporate the variables in a description of the system. This step may involve using standard equations in a spreadsheet or working with a specialized simulation package that provides ready-built libraries of components and systems. Once the model has been constructed and tested, it can serve as the basis for a variety of 'what-if' scenarios that describe the functionality of a system before it is built and integrated into the overall robot design.

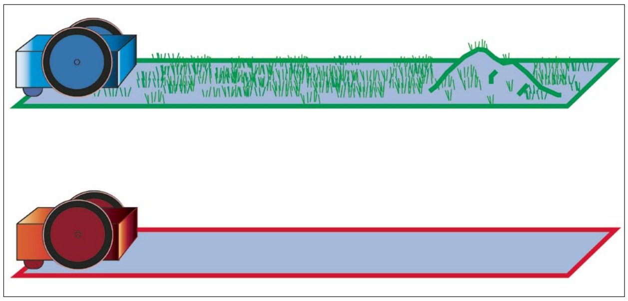


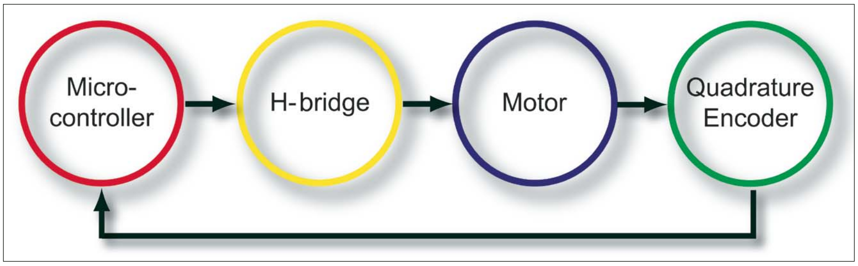
FIGURE 1. Two robots facing different environmental conditions that must maintain the same speed.

Robot Motion Control

To illustrate the application of modeling and simulation at the systems level, let's consider the motion control of a wheeled robot. Specifically, the task is to determine the best DC motor control algorithm for autonomous robots that must operate cooperatively with a swarm of similar robots. Consider the case of two operationally identical robots from the swarm that must arrive at a specific location at approximately the same time. The robots are equidistant from the target location, but one robot must travel on smooth asphalt while the other must traverse patches of grass and overcome a small hill.

In this scenario depicted in Figure 1, an open-loop speed control in which each robot is set to the same, fixed power level will result in the red robot arriving at the destination ahead of the blue robot. Not only does the grass present greater rolling resistance and greater overall drag than the asphalt, but at the preset power level, the blue robot may have insufficient drive to overcome the hill and may stall or even roll backwards.

FIGURE 2. Typical closed-loop motion control elements.



One approach is to set the blue robot's power level greater than that of the red robot. However, even if the blue robot doesn't flip over or collide with objects because of instability at the higher speed, an accurate estimation of the optimum power setting would have to consider the differences in motor efficiency, battery discharge rates, and other factors that might favor one robot over the other. Furthermore, the solution would be specific to the course and the two robots. If the swarm moved to a new environment, used different robot pairings, or added a third robot to accompany the current pair, the power setting estimates would likely have to be reworked.

A better solution is to replace the fixed drive level, open-loop control with a closed-loop system that dynamically adjusts drive level to compensate for environmental factors, variations in the robot power system, changes in payload, and other internal variables. Figure 2 shows the components of a typical closed-loop motor control system. In practice, a servo can be substituted for the H-bridge and motor, and a magnetic pickup or other type of feedback device can be used in place of a quadrature encoder. Note that in this model, the microcontroller transforms the desired speed to a DC voltage or PWM signal that instructs the H-bridge or other software/hardware how hard to drive the motor.

Following is a discussion of how modeling and simulation can be used to select the operating parameters for the most popular motion control techniques used in mobile robots — on-off control and various forms of PID (Proportional, Integral, Derivative) control.

On-Off Control

If you've ever driven a bumper car in an amusement park, then you've taken part in a form of on-off control. When you step on the power switch, a bumper car lurches ahead at full power. Release the switch and the bumper car rapidly decelerates to a stop. Maintaining a speed somewhere between the maximum possible and zero requires modulating the duty cycle of the motor — that is, intermittently pressing and releasing the foot switch.

With a little practice, most drivers can predict when to let up on the button instead of simply reacting to the current speed. This discussion considers the reactive component of on-off control — a task that a microcontroller can handle with relatively simple programming. For example, we can model on-off control with the following pseudocode:

```
If speed >= set point then motor off
If speed < set point then motor on full
```

where *speed* is the current speed, and *set point* is the desired speed. In practice, the set point is a narrow range of speed, such as 50 ±1 cm/sec, instead of a single value. Even so, the model illustrates the strengths and weaknesses of on-off control.

Converting the pseudocode to a more formalized model that includes the key variables, we have:

$$\text{Speed}_t = \text{Speed}_{t-1} + (\text{Speed}_{t-1\text{-dead time}} < \text{Set Point}) \times \text{Increment} - \text{Drag}$$

where *dead time* is the time required for the robot to respond to a 'go' signal, *increment* represents the speed increment due to acceleration from the motors, and *drag* embodies the deceleration due to forces working against robot forward motion.

Dead time encompasses robot inertia, the effects of static friction within the drive system, and the relatively minor delays caused by electrical components. The speed *increment* is a function of current battery charge, the efficiency of the drive system, and the power setting. For our wheeled robots in Figure 1, *drag* includes deceleration due to gravity and resistance due to the asphalt and grassy surfaces. A more complete model would consider the effects of ambient temperature on static and dynamic friction, battery efficiency, and motor/servo operation, as well as wind conditions, humidity, and rain or snow.

Assuming a 'tick' or time interval between calculations of one second, the speed of the robot is equal to the speed one second ago, plus the speed increment, less deceleration due to drag. If speed of the robot one second ago was less than the set point, then the speed increment is added to robot speed. For example, assuming dead time = 1 second, set point = 50 cm/sec, drag = 7 cm/sec, speed increment = 15 cm/sec, and speed at 4 sec = 48 cm/sec, the speed at 5 seconds is computed as:

$$\begin{aligned}\text{Speed}_{5\text{ sec}} &= 48\text{cm/sec} + (48\text{ cm/sec} < 50\text{ cm/sec}) \times 15\text{ cm/sec} - 7\text{ cm/sec}\\ &= 48\text{ cm/sec} + (1) \times 15\text{ cm/sec} - 7\text{ cm/sec}\\ &= 56\text{ cm/sec}\end{aligned}$$

Note that the *increment* and *drag* variables are expressed in terms of cm/sec because the time interval is one second. That is, the velocity increment is a measure of robot acceleration, as in:

$$\begin{aligned}\text{increment} &= \text{acceleration} \times \text{time}\\ &= 15\text{ cm/sec/sec} \times 1\text{ sec}\\ &= 15\text{ cm/sec}\end{aligned}$$

Following the on-off algorithm, speeds at six seconds and beyond, in cm/sec, are 49, 57, 50, 58, 50, 43, 51, and so on, as graphed in Figure 3. According to the model which is run in Microsoft Excel, the speed of each robot will vary between 36 and 65 cm/sec en route to the target location. This jerky movement would likely wreak havoc on an onboard accelerometer or a sensitive laser rangefinder, cause excessive drain from the battery pack, and result in premature failure of the drive train.

Actual robot speed would follow the plot in Figure 3 to the extent that the underlying model accurately captures the variables relevant to the motion control system. In this example, the model is simplified for illustration purposes. In reality, dead time isn't a binary event in which the robot is still and then moving at full speed one millisecond later, but the ramp-up in speed is more accurately described by a function that considers factors such as the instantaneous status of battery charge. Similarly, *drag* isn't a constant, especially for the blue robot. The drag on the blue robot will vary when dirt adheres to the tires, grass blades become entangled in the axles, and when the robot moves from the near side to the far side of the hill.

Proportional Control

Because on-off control can result in a stuttering robot and excessive wear and tear on the entire robot structure, more advanced closed-loop control systems are often used on all but the least expensive robots. The most popular alternative is some form of PID controller. A model of basic proportional control takes the form:

$$\text{Speed}_{t} = \text{Speed}_{t-1} + (\text{Set Point} + \text{Set Point Correction} - \text{Speed}_{t-1\text{-dead time}}) \times \text{Proportion Constant}$$

The new term, *set point correction*, is introduced into the equation because without it, the *set point* would never be reached. Think of it as a fudge factor to make up for the fact that only a proportion of the difference between actual and target speed — as opposed to the entire difference — is used to calculate the current speed.

Proportional control, when used with the appropriate proportion constant and set point correction, provides much smoother motion control than on-off control after an initial period of instability, as shown in Figure 4. The proportional control response illustrates a potential limitation of the technique — the initial speed may drastically overshoot the desired speed significantly before settling down to the set point.

Proportional control results in instability at specific combinations of set point, set point correction, dead time, and proportion constant values. In particular, as the proportion constant is increased, there is a tendency for the output level to oscillate. An advantage of working with a model of proportional control is that these problematic combinations can be identified more easily than by manipulating the variables on a real robot.

Proportional Integral Control

Proportional integral control attempts to address the instability of the purely proportional control scheme by adding a term that represents the sum of the difference between desired and actual speed over time. The basic proportional integral control equation is defined as:

$$\text{Speed}_{t} = \text{Speed}_{t-1} + (\text{Set Point} + \text{Speed}_{t-1\text{-dead time}}) \times \text{Proportion Constant} + \text{Integral}_{t-1} + (\text{Speed}_{t} - \text{Integral}_{t-1}) \times \text{Integral Constant}$$

where *Integral* is the sum of differences between *Set Point* and current *Speed*, and the *Integral Constant* determines the contribution of the integral to the current speed. Compare the response curve shown in Figure 5 with that of Figure 4. Note the absence of the *Set Point Correction* variable from the equation. The models used to generate the two graphs use the same set point, set point correction, dead time, and proportion constant.

Because the integral term reflects the difference between actual and desired speeds over time, proportional integral control is most useful when robots must travel relatively long distances at constant speeds — as in the two robots depicted in Figure 1. A robot that is constantly stopping, turning, and reacting to elements in the environment may not benefit from the addition of the integral term.

Proportional Integral Derivative Control

Like the purely proportional controller, the

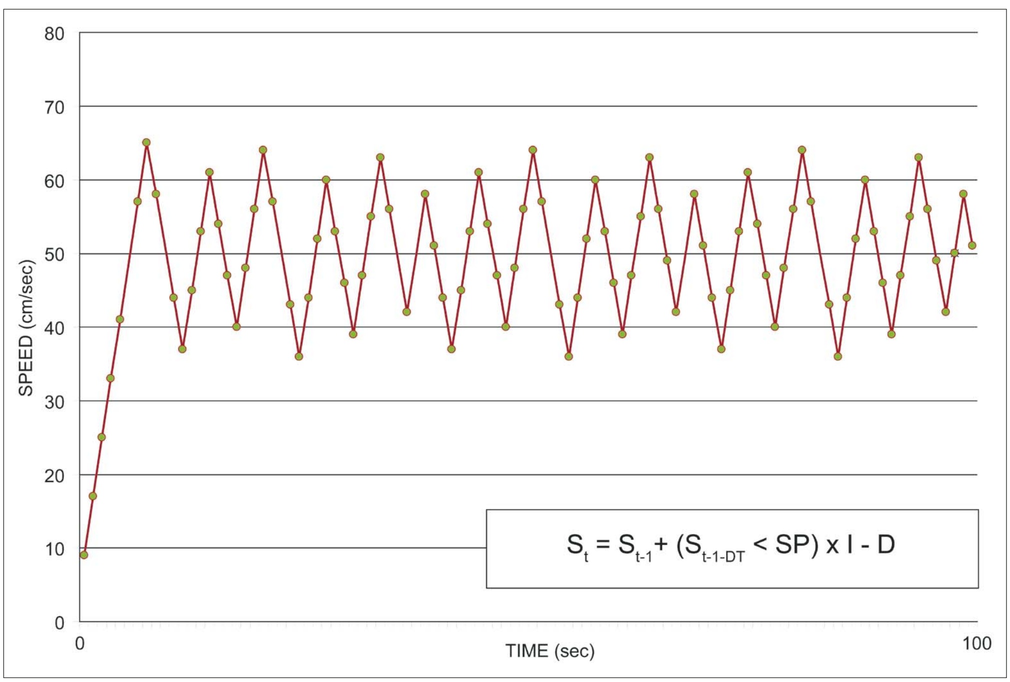


FIGURE 3. On-Off control response plotted in Excel. Set Point (SP) = 50 cm/sec; Increment (I) = 15 cm/sec; Drag (D) = 7 cm/sec.

proportional integral controller is predisposed to instability with some variable combinations. To counteract this instability, a derivative term — the equivalent of a smoothing capacitor in a power supply — can be used to suppress the rate of change. The full proportional integral derivative control equation is defined as:

$$\mathrm{Speed}_{t} = \mathrm{Speed}_{t-1} + (\mathrm{Set\ Point} + \mathrm{Speed}_{t-1-\mathrm{dead\ time}}) \times \mathrm{Proportion\ Constant} + \mathrm{Integral}_{t-1} + (\mathrm{Speed}_{t} - \mathrm{Integral}_{t-1}) \times \mathrm{Integral\ Constant} - (\mathrm{Speed}_{t-1} - \mathrm{Speed}_{t}) \times \mathrm{Derivative\ Constant}$$

where *Derivative Constant* determines the sensitivity of the model to second-to-second variations in speed. Note that, unlike the proportional and integral terms, the derivative term is subtracted from the value of speed one second ago. This tends to diminish the rate of change, as shown in Figure 6.

In addition to these basic control models, there are numerous variations and enhancements to PID controllers, as well as much more sophisticated algorithms used for motion control. However, regardless of the specific algorithm, modeling and simulation can be used to explore the benefits and limitations of a particular approach.

Tools

As illustrated by Figures 3-6, a spreadsheet can be used to define and run models of motor controllers. Spreadsheet applications are available free, as part of Open Office, and as a component of the Microsoft Office suite. A spreadsheet is a perfect modeling environment for systems that can be represented by a few algebraic equations. However, modeling with a spreadsheet quickly becomes difficult as the complexity and number of equations increases. Furthermore, applying models appropriately often requires low-level knowledge of the underlying math. For these reasons, specialized modeling and simulation tools are often used to implement complex system-level simulations.

Spreadsheet Add-ons

The rows and columns of numbers and static graphs produced by a spreadsheet aren't intuitive to everyone, and repeatedly modifying parameters and graphing the output can be tedious. For these situations, a variety of spreadsheet add-ons can be used to enhance visualization and enable

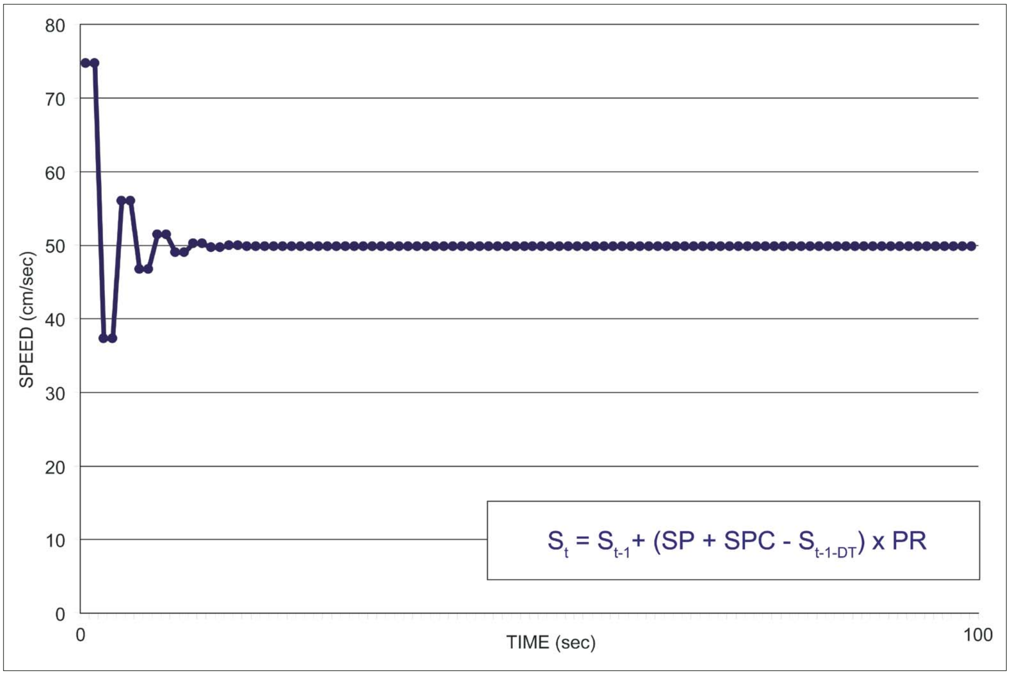


FIGURE 4. Proportional control response. Set Point (SP) = 50; Set Point Correction (SPC) = 100; Dead Time (DT) = 1; Proportion Constant (PR) = 0.5.

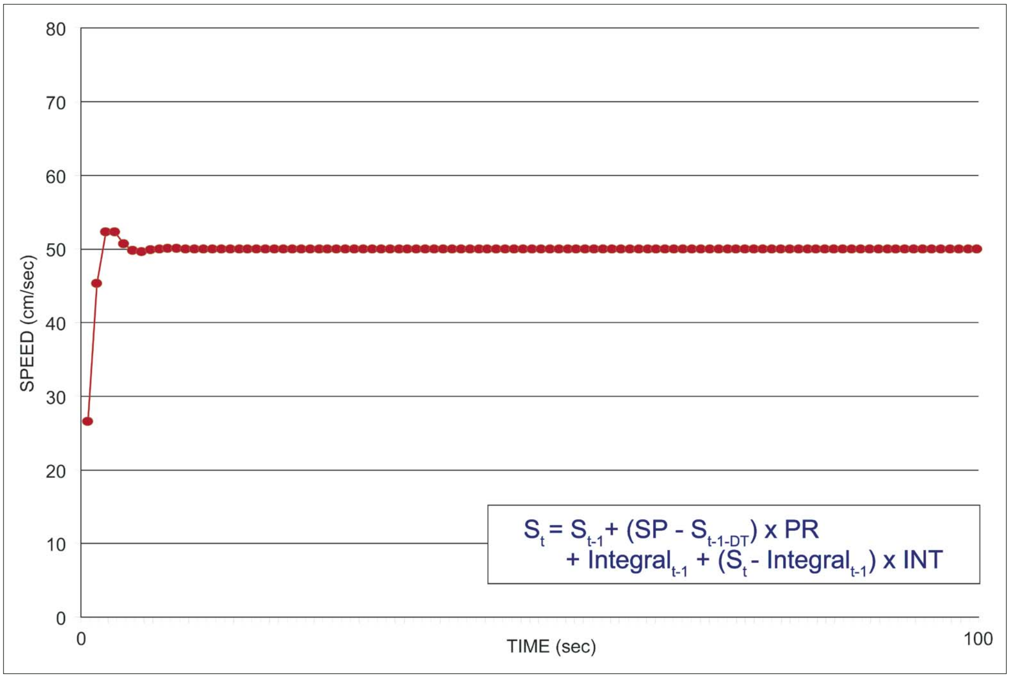


FIGURE 5. PI controller response. Set Point (SP) = 50; Set Point Correction (SPC) = 100; Dead Time (DT) = 1; Proportion Constant (PR) = 0.5; Integral Constant (INT) = 0.8.

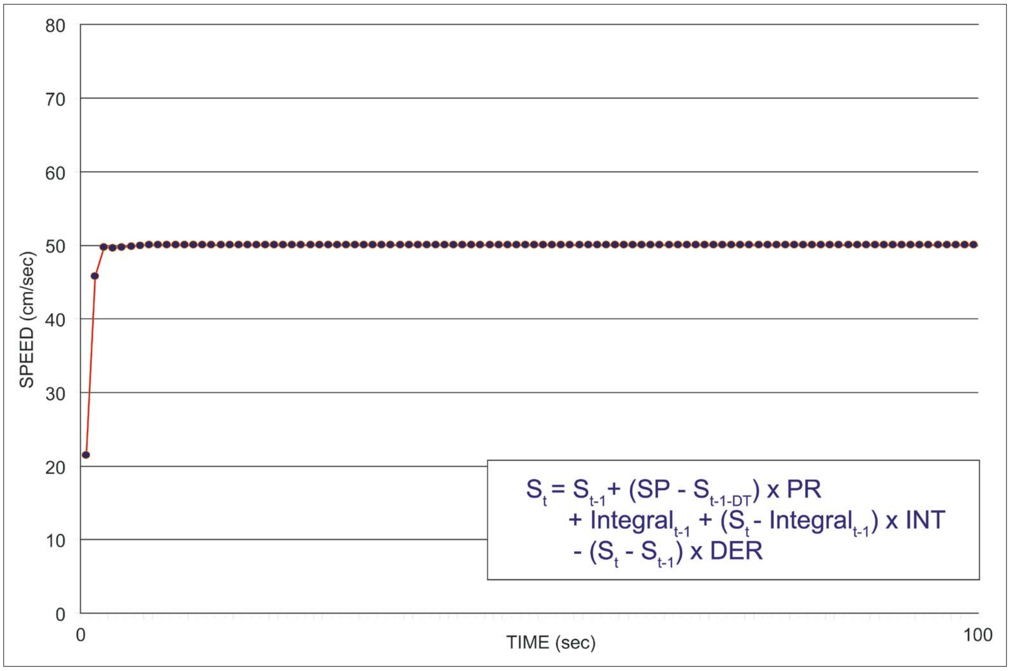


FIGURE 6. PID control response. Set Point (SP) = 50; Dead Time (DT) = 1; Proportion Constant (PR) = 0.5; Integral Constant (INT) = 0.8; Derivative Constant (DER) = 0.5.

more rapid data entry.

Figure 7 shows an example of a control system simulation I created with Business Objects Crystal Xcelsius, a Flash compiler for Microsoft Excel. Instead of entering numbers in an Excel spreadsheet, data are entered by dragging the mouse, and output appears on the dials, in approximately real-time.

The interactive simulation, which runs in a Flash-enabled browser, is available for download from the *SERVO* website (**www.servomagazine.com**). If you want to try your hand at creating simulation front-ends to your spreadsheet models, you can download a free, trial version of Xcelsius from the Business Objects website. You'll have to put up with a watermark on the Flash files.

Extend

General-purpose spreadsheets and language compilers can be used to create system-level simulations — given enough time and expertise with the language. For time challenged robot developers, several special-purpose modeling and simulation environments may be more useful. The advantage of a specialized environment is that the relevant models are likely already written and debugged. As a result, the simulation task becomes one of providing accurate values for variables instead of defining new equations.

Extend, from Imagine That!, is one of the easiest to use systems-level modeling and simulation environments. The program boasts an extensive library of electronic components and systems, including a ready-made PID controller, as shown in Figure 8.

Defining a system is as easy as drawing lines between icon input and output terminals, double-clicking on icons to specify variable values in pop-up dialog windows, and then running the simulation. The source code for each icon — which is written in a C-like script — can be modified and recompiled as a new icon or as a replacement for an existing icon. Furthermore, icons can be nested within each other so that the complexity of the display can be managed.

The output of Extend is similar to that of a spreadsheet, in that there is a table of data and the corresponding plot of variables over time. Although the plot is static, there are tools to zoom in to examine particular variables with a few clicks of the mouse. For example, the plot in Figure 8 shows the motor drive signal in red and the corresponding motor response in blue. You can download a free demo version of Extend from the Imagine That! website. The demo is limited to building (but not saving)

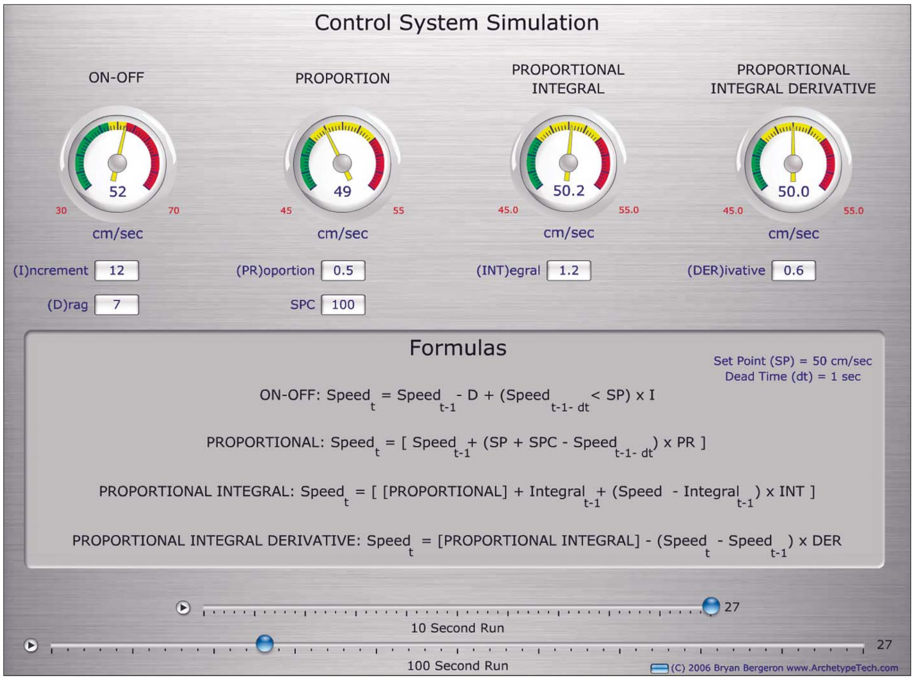


FIGURE 7. Crystal Xcelsius front-end to on-off, P, PI, and PID control models.

models with up to 25 icons.

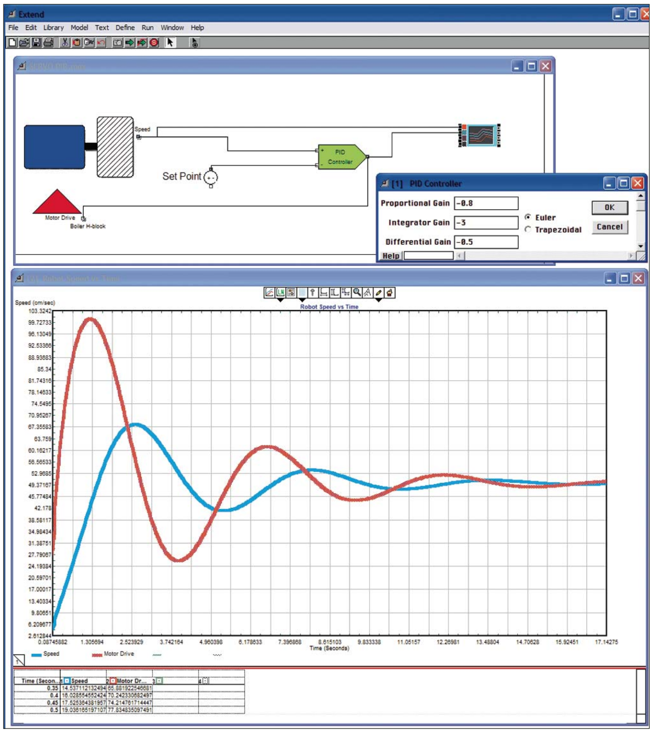
MatLab

MatLab, which has been discussed at length in the previous two articles in this series, offers the greatest flexibility of the solutions discussed here. With the addition of the Control System Toolbox — a collection of algorithms that implement common control system modeling and simulation techniques — MatLab is both powerful and easy to use. The graphical user interface shown in Figure 9 provides a veneer over the matrix operations, allowing you to specify variables and manipulate equations through forms and interactive displays.

As an example of the detail supported by MatLab, consider the motor DC response curve generated by the program for use in PID control simulation. Instead of specifying a delay time, of say, one second, the motor characteristics can be defined to achieve a more realistic control output. Support for this level of detail also enables you to specify a different motor with a different response curve and determine how the overall system will respond. Furthermore, MatLab has a built-in utility for automatically determining the optimum coefficients in a PID model. This automatic tuning feature can save hours of repeatedly manually manipulating variables and examining the output graph.

The downside of this ability to define a model with exquisite detail is that you can easily get into trouble if you don't understand the theory underlying the available options. Fortunately, MatLab is so popular in engineering that there are many academic texts that use MatLab and the Control System Toolbox to illustrate control system principles. *Modern Control Design with MATLAB and SIMULINK* [1] is an example of an approachable book applicable to robot motion control system design.

FIGURE 8. Extend showing a PID controller and a DC motor/wheel assembly.

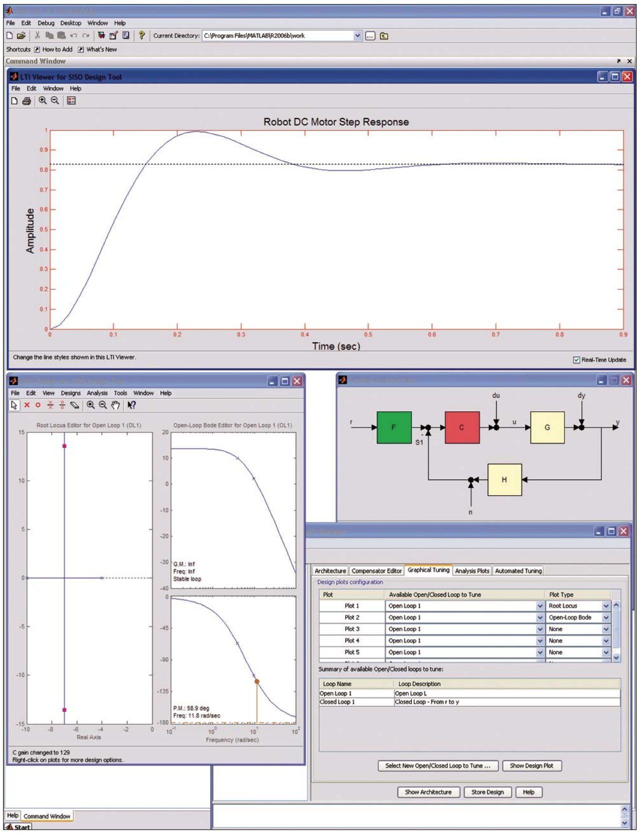


From Here

The texts by Holland [2] and Jones [3] both have introductory chapters on PID control systems. Dahlen's *Nuts & Volts* series on the PID controller [4-6] is another good reference for beginners. Moving on to other robot systems, *The Robot Builder's Bonanza* [7] discusses sensor arrays, communications, energy management, and other robot elements from a system-level perspective. *The Mechatronics Sourcebook* [8] offers an even lower level treatment of robotics, including sensor construction and electronic circuit design.

If you want to try your hand at robot circuit design, there are several simulation systems that can make the task more efficient and enjoyable. Spreadsheets, Extend, and MatLab can be used to simulate

FIGURE 9. Exploring the response of a DC motor under proportional control with the MatLab Control System Toolbox.



Resources

Crystal Xcelsius. Business Objects.
www.businessobjects.com

MatLab/Simulink. The MathWorks.
www.mathworks.com

Extend. Imagine That!
www.ImagineThatInc.com

Electronics Workbench Multisim.
National Instruments.
www.ni.com

TINA-TI. Texas Instruments
www.ti.com

TINA Professional
www.tina.com

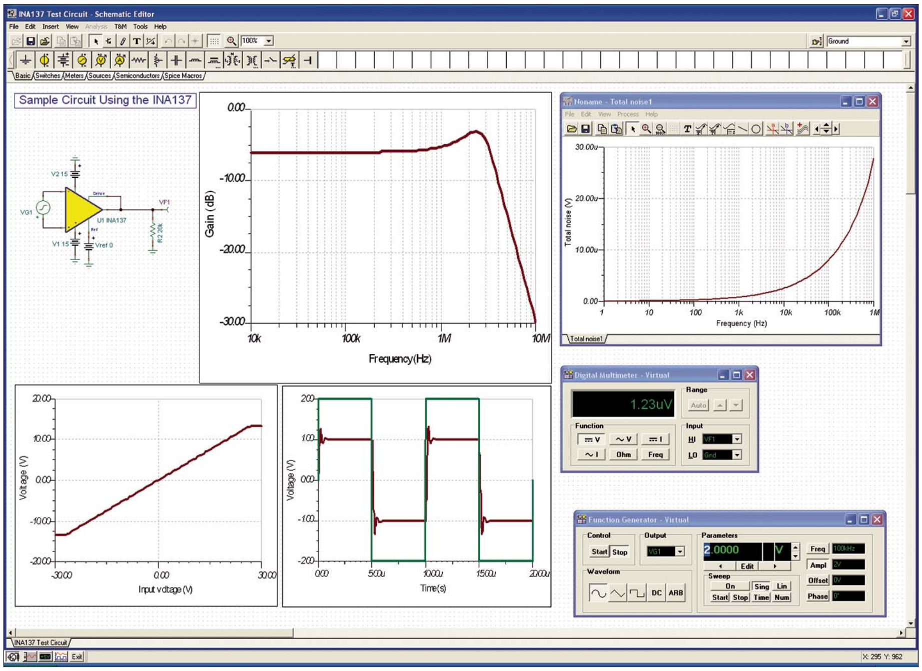


FIGURE 10. TINA-TI showing gain, DC transfer, transient response, and noise figures for the INA137 op-amp.

circuitry at the component level, but there are better options, such as the SPICE (Simulation Program with Integrated Circuit Emphasis) circuit simulator. Originally developed at UC Berkley, dozens of free and commercial versions of SPICE are available. I suggest TINA-TI, a free, downloadable, fully functional SPICE-based simulator preloaded with a library of passive components and digital and analog ICs.

Figure 10 shows TINA-TI simulating an operational amplifier, with graphs of gain, DC transfer, transient response, and total noise. In addition to the digital multimeter and signal generator shown in the figure, TINA-TI includes an oscilloscope, XY recorder, and signal analyzer. The only limitation of TINA-TI is that the library of ICs is limited to TI (Texas Instruments) products. While this isn't a major limitation with analog ICs, identifying equivalents to TI digital ICs is more problematic.

A commercial version of TINA — complete with a full component library — is available from DesignSoft. If you're considering a commercial version of SPICE, then the Electronics Workbench Multisim from National Instruments is also worth investigating. A free 90-day evaluation copy on CD-ROM is available from National Instruments. **SV**

References

[1] Tewari, A., *Modern Control Design with MATLAB and SIMULINK*. 2002: John Wiley & Sons.

[2] Holland, J., *Closed Loop Controls, Rabbits and Hounds*, in *Designing Mobile Autonomous Robots*. 2004, Elsevier: New York. p. 55-76.

[3] Jones, J., *Control Systems*, in *Robot Programming: A Practical Guide to Behavior-Based Robotics*. 2003, McGraw-Hill: New York. p. 25-47.

[4] Dahlen, A., The PID Controller: Part 1. *Nuts & Volts*, 2005. January: p. 65-70.

[5] Dahlen, A., The PID Controller: Part 2. *Nuts & Volts*, 2005. February: p. 62-68.

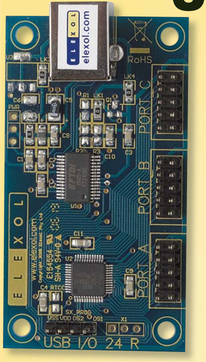
[6] Dahlen, A., The PID Controller: Part 3. *Nuts & Volts*, 2005. March: p. 56-62.

[7] McComb, G., *The Robot Builder's Bonanza*. 2001, New York: McGraw-Hill.

[8] Braga, N., *Mechatronics Sourcebook*. 2003, New York: Thompson Delmar Learning.

www.orteches.com www.orteches.com www.orteches.com www.orteches.com www.orteches.com www.orteches.com

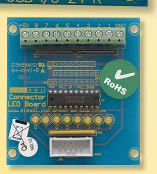
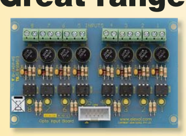
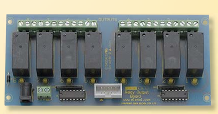
Elexol 3rd Generation Low Cost USB Data I/O Module



Need to get data into or out of a USB port? Here's what you need…

- 24 independently programmable Input/Output pins grouped into 3 ports.
- Single module high-speed digital Input/Output solution.
- Up to 128 modules can be connected to a single PC with capabilities of further expansion.
- Easy to connect by 0.1" pitch headers to suit standard IDC connectors.
- Integrated Type-B USB connector.
- On-board unique serial number in EEPROM and custom programmable FLASH microcontroller.
- Both USB enumeration information and microcontroller can be re-programmed to suit customer needs.
- Module powered by the USB from the PC.

Great range of I/O-24 peripherals, too…


CONNECTOR/LED BOARD Provides screw terminal connections plus optional either pull up or pull down resistors and LED port status indication.

OPTO INPUT BOARD Provides 8 Opto isolated inputs for each of the I/O-24 pins with LED indications.

RELAY OUPUT BOARD Provides 8 isolated relay contact outputs suitable for a variety of loads.

SWITCH/PUSH BUTTON BOARD Provides 8 switch/push buttons with LED indicators showing the status of the I/O pin.

Visit our website www.orteches.com

Ortech™ Education Systems

Ortech Education Systems
1119 - 9th Ave. S. • Moorhead, MN 56560
218.287.1379

www.orteches.com www.orteches.com www.orteches.com www.orteches.com www.orteches.com www.orteches.com

Build the BioCrab

Putting the AX-12 to Work

by Michael Simpson

Last month, I gave you some technical background into the AX-12 protocol, as well as a simple two-wheeled example bot. The AX-12 actuators used in this project are the future of robotics. With a fast and easy networked interface, we no longer have to worry about the control aspects of our project. Now it's time to really dig into robotics. I'm going to show you how to build a large crab — the BioCrab.

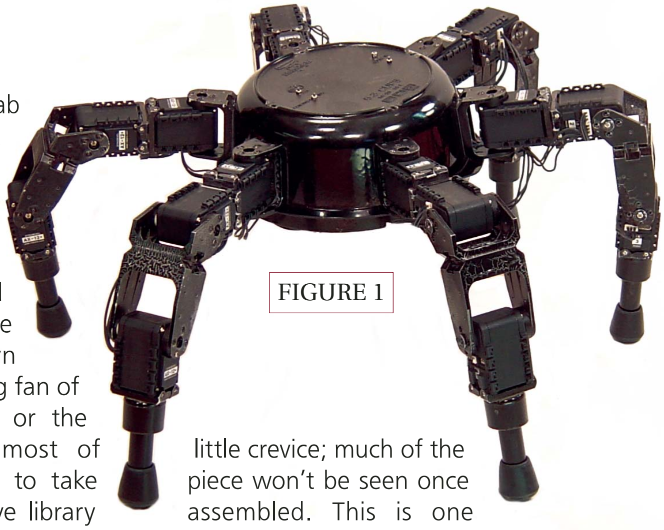


FIGURE 1

I call this bot the BioCrab because I used a Bioloid comprehensive kit for most of his components. I like to think outside the box, so you won't find this project in the Bioloid manual. The comprehensive kit comes with its own controller, but I'm not a big fan of closed system controllers or the graphical systems that most of them use. I also wanted to take advantage of the extensive library system of the DiosPro.

You can build the BioCrab by purchasing a frame kit and 18 Ax-12s. This will yield you several more frame pieces, as well as 18 extra #20 cables. The good folks over at CrustCrawler have several Bioloid options. They even carry the DiosPro and Dios Workboard Deluxe used in this project.

Assembly

Before I get into the assembly, let me talk a little about paint. I wanted the color of my BioCrab to be black. For the bowl and PVC parts, this was no problem. Simple plastic paint worked fine. The Bioloid frame pieces were a different story. Robotis calls the material engineered plastic. I'm not sure what that means, but they are almost impossible to color. I tried various paints and dyes but had little luck; the paint would just fall off and the dye just didn't affect the color. What did work, however, was a permanent marker.

I used both a fine and ultra fine point marker to color some of the pieces. If you decide to use paint, the markers are great for touch-up. The F10 piece shown in Figure 2 is probably the worst-case scenario because of all the tiny crevices.

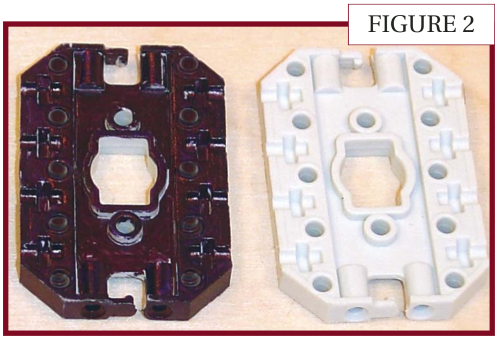
It's not necessary to cover every little crevice; much of the piece won't be seen once assembled. This is one reason that I recommend full assembly before painting. Once you have the BioCrab put together, you can remove the pieces you want to paint or color.

Assemble the Six Legs

The BioCrab is a very easy robot to build. First, you will need to assemble six legs. I recommend assembling a single leg, then once you have it down, it's a simple matter of duplicating the remaining five legs.

In the steps that follow, I will be referring to Figure 3. The frame pieces are marked F1 through F7 and the AX-12s are marked ID1-ID3. AX-12s have a small index located on the hub. There is also an index mark on the AX-12 body. These should be aligned before assembly. I recommend you download the AX-12 manual as it gives detailed instructions on how to

FIGURE 2



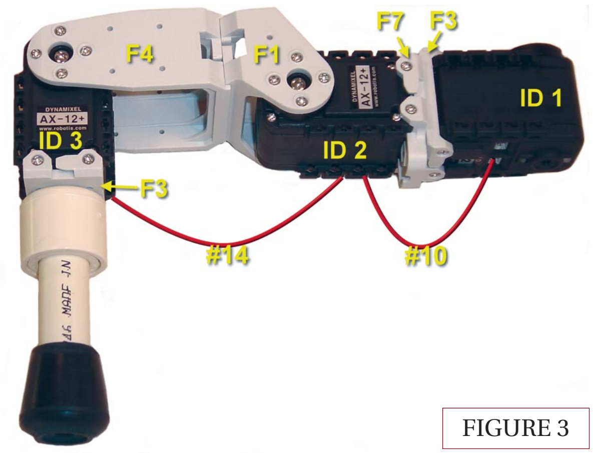


FIGURE 3

attach it to the various frame pieces.

STEP 1: Using four S1 screws and nuts, attach an F3 to an F7. Note that the F7 has a rounded lip. This lip should be facing down.

STEP 2: Using four S1 screws and nuts, attach an F1 to an F4. One side of the F4 has a lip. This lip should be facing up.

STEP 3: Attach an F3 to a 3/4" PVC cap. To do this, center the F3 on the top of the cap and mark two of the opposing holes with a fine point marker. Drill out the two holes with a 5/64" drill bit. Use two S2 screws and nuts to attach the pieces. This is done by inserting the screws through the two holes from the inside of the cap. The nuts are placed in the F3. Make the connections very tight because you won't be able to do so later.

STEP 4: Insert a 2-1/2" x 1/2" PVC pipe into a 3/4" x 1/2" PVC bushing. Tap it in place so it bottoms out.

STEP 5: Insert a 3/4" x 1/2" bushing into the cap.

STEP 6: Insert a rubber foot over the other end of the PVC pipe. I picked up the feet I used from my local home center. There were many options. Many department stores sell these feet, as well. Take one of the legs with you so that you can make sure you have a tight fit.

STEP 7: Attach the completed foot to an AX-12 with an ID of 3. Make sure the orientation is as shown with the hub facing up. You will need a bushing (BU) and a washer (WA) inserted into the F4 hole opposite of the hub. Insert an SB screw into the bushing and four S1 screws into the hub.

STEP 8: Attach an AX-12 with an ID of 1 to the F3/F7 assembly. This is done by inserting four nuts into the slots on the bottom of the AX-12, as shown in the AX-12 manual. Slip the F3 over the mounts and insert four S1 screws into those nuts.

STEP 9: Attach the F7 side of the F3/F7 assembly to an AX-12 with an ID of 2. Don't forget to insert the four nuts into the slots on the bottom of the AX-12.

STEP 10: Attach the other end of the AX-12 (ID 2) to the F1. Again, the hub is facing up and the bushings are on the bottom. Attach with S1 and SB screws as before.

STEP 11: Repeat steps 1-10 for each of the remaining five legs. The legs should have the AX-12 IDs as listed:

Leg 0 = ID 1,2,3
Leg 1 = ID 4,5,6
Leg 2 = ID 7,8,9
Leg 3 = ID 10,11,12
Leg 4 = ID 13,14,15
Leg 5 = ID 16,17,18


FIGURE 4

I will refer to the leg numbers later when we attach them to the base.

STEP 12: Attach a #14 cable between the AX-12 on the foot and the middle AX-12.

STEP 13: Attach a #10 cable between the middle AX-12 and the hip AX-12. Note that the comprehensive kit only comes with four #10 cables. To create a cable for the remaining two legs, attach two #6 cables with a three-pin header.

Assemble the Base

The BioCrab base is even easier to assemble than the legs. Start with a bowl that is at least 6" in diameter. When choosing the bowl, there are a few things to consider:

1) The bowl should have sides that are straight. If the sides are angled, it will affect the geometry of the legs.

2) The material should be as rigid as possible. If you purchase a thin plastic or vinyl bowl, the robot may flex and wobble when it walks.

3) The bowl should be between 6" and 8" in diameter.

The bowl shown in Figure 4 is a Stain Shield made by Rubbermaid. The bowl is clear but it can be painted. If you decide to paint your bowl, paint both the inside and outside.

STEP 1: You need to mark six equal points on the bowl. The easiest way to do this is to use a clock face as a guide. Place the bowl gently on the top of the clock face and — using a marker — make a mark at the 12, 2, 4, 6, 8, and 10 hour positions.

STEP 2: Place an F2 frame piece on one of the marks. Using an ultra fine point marker, mark four of the holes. Do this for all of the clock marks that you made in Step 1. It's not important where you place the F2 piece on the clock mark. What's more important is that you are consistent. In other words,

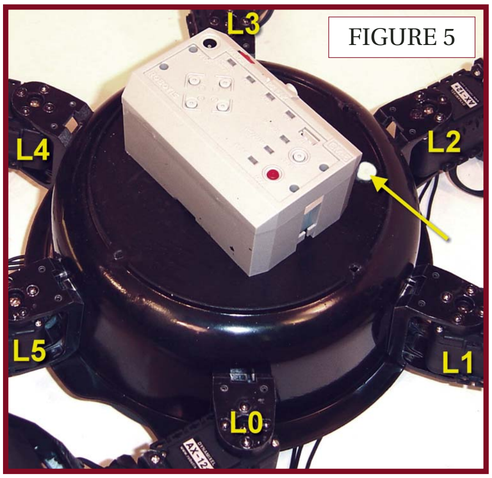


FIGURE 5

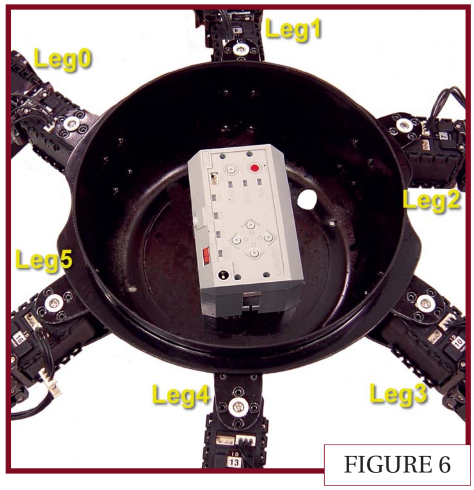


FIGURE 6

if you center the F2 on one clock mark, it's important that you do the same on all the other marks.

STEP 3: At this point, you can paint the bowl and F2 pieces.

STEP 4: Using four S1 screws and nuts, attach each F2 frame piece to the bowl (see Figure 4).

STEP 5: It's time to attach the CM5 to the base. The easiest way is to sit the CM5 on top of the base and place a thin Phillips screwdriver in the four mounting holes to score the marks.

Notice the orientation. The CM5 should be roughly in-line with leg 1 and leg 4, as shown in Figure 5. Also notice the small hole drilled in the top of the base. We will use this later to pass one of the AX-12 cables to our Dios Workboard. I drilled a 1/2" hole for mine, but you can use a rotary tool to create a smaller hole that is just large enough to pass the cable through.

Once the four CM5 holes are scored, drill them out with a 5/64" drill bit.

STEP 6: Attach the CM5 from the under side of the base as shown in Figure 6. Use four S2 screws and nuts. The power connector should be on the end closest to leg 4.

STEP 7: Attach the Dios Workboard Deluxe to the base (see Figure 7). The board should be positioned so that the hole is to the left of the board as shown. To mark the board, place the board on top of the base and mark or score the location of the four holes. The bowl I used had a small 1/8" lip right where the screws attach to the board. I used a small rotary tool to grind away a small portion of the lip where the holes were located.

First, attach four 5/8" standoffs to the base with four 3/8" #4 machine screws. Place a #4 washer on the screws, then pass the screws through the holes from the under side of the base. Leave the standoffs loose.

Next, attach the board to the standoffs with four 1/4" #4 machine screws. Tighten all screws at this point.

STEP 8: Attach a #18 cable to each of the CM5 ends as shown in Figure 8. These connect to leg 1 and leg 4 as shown. Attach the Expansion PCB to the CM5 with a #6 cable.

Attach a #20 cable from the Expansion PCB to each of the remaining legs as shown. The last #20 cable connects to the remaining connector on the Expansion PCB and through the hole.

STEP 9: Connect a three-pin header to the end of the cable that is protruding from the hole. The connector has a flat side and a rounded side. Take the rounded side shown in Figure 9 and mark the GND lead with a small black mark.

Wire ports 8 and 9 on the breadboard hole 28 as shown in Figure 10. Connect the Vin lead to hole 29 on the breadboard. Connect VSS to hole 30.

STEP 10: The final step is to connect the cable with the header to the breadboard (see Figure 10 again). I recommend that you remove both the DiosPro chip and the RS-232 chips from their sockets. Once the cable is connected, switch on the CM5. The CM5 LEDs should light up. The green LED on the Dios Workboard should also light up.

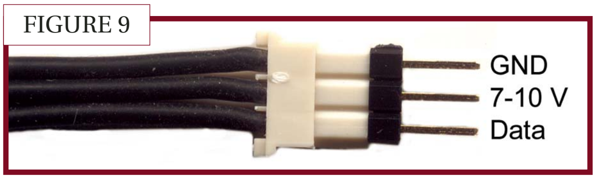


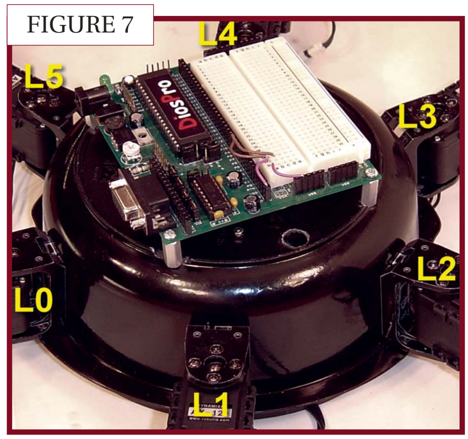
FIGURE 9

Now switch off the CM5 and plug the two chips into their sockets. Once you power on the CM5, the LEDs should again light up.

BioCrab Software

Test Software

You now have a complete BioCrab. It's time to start the legs moving. Just to

FIGURE 7


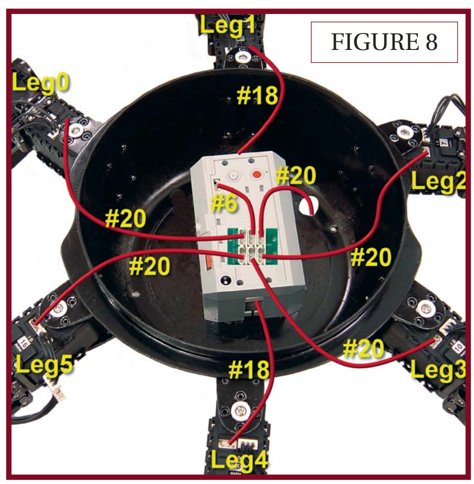


FIGURE 8

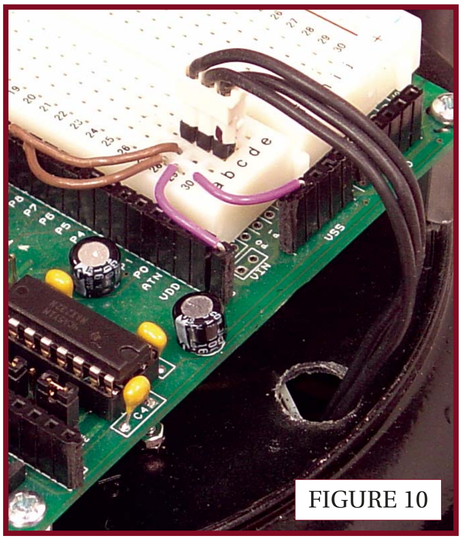


FIGURE 10

make sure everything is connected properly, let's run some test programs. Before you run any of the programs, though, you should build some sort of stand for your BioCrab. This stand support should be tall enough to keep the legs from touching the surface on which it is sitting when fully extended. A one-gallon paint can works well for this.

Connect a serial cable from your PC to the DiosPro. Power-up the DiosPro and AX-12 network. Load the first program called *TestHHip.txt*. This program will test all the horizontal hip joints in succession. The horizontal hip is the AX-12 that is connected to the base.

Click the program button and the program will compile and upload the test program. Verify that all the joints are moving in sequence.

Load the *TestVHip.txt* and the *TestKnee.txt* programs and test those joints, as well. If one or more of the joints don't move, there could be several reasons why this is happening. Power-down the BioCrab and make sure the legs can move freely without sticking. It's also possible that the AX-12 ID needs to be changed.

There are a couple of ways you can change the ID. You can use the CM5 and Bioloid software to change it. I have also created software that will allow you to communicate with the AX-12s from your PC. One of these programs will allow you to change the IDs of the devices. These programs can be downloaded from the Kronos Robotics website (see the Parts List).

WalkerAX Library

In order to move the 18 AX-12 devices in a synchronized manner, I have added a new series of commands to the DiosPro AX library that allows you to synchronize the movement of several AX-12s.

There are three commands:

- *AXclearsync* — Clears the sync memory and prepares for a new sequence of synchronized commands.

- *AXaddsync* — Lets you add a goal position and speed to a device.

- *AXsendsync* — The AX-12 devices will not begin moving until this command is sent.

These commands use one of the free memory banks of the DiosPro so you can send sync commands for up to 50 devices at once.

Here is an example of moving four devices with these commands.

```
AXclearsync
    AXaddsync 1,500,100
    AXaddsync 2,250,1000
    AXaddsync 3,350,400
    AXaddsync 4,500,100
AXsendsync
```

Back when I built the FaceWalker robot, I used a library called IK that used Inverse Kinematics to calculate the position of the legs. This program worked fairly well but had many issues. First, it relied on many trigonometry calculations. These worked fine but left little processing power left to do other functions. They were complicated and very difficult to modify. This made it almost impossible to use with other walkers.

I decided to write a set of state driven routines that would give me the same functionality. This is how the WalkerAX library was born. The WalkerAX library is included with the Dios compiler. To use the library, just add the following include statement to the end of your program:

include \lib\WalkerAX.lib

The WalkerAX routines were designed on the premise that three legs are always in contact with the ground. The other three legs are moving into position so that they may be lowered. Once one set is lowered, the other set is raised. For the sake of this discussion, legs 0, 2, and 4 are triangle A and legs 1, 3, and 5 are triangle B.

The main walker state machine has four states. Let's take a close look at what happens in each of those states. There is a download on the *SERVO Magazine* website (**www.servomagazine.com**) called *BCslowspin.mpg*. I used program BioCrabspin.txt as a quick gait demonstration.

I set the speed at the lowest setting and added a one-second delay between states. It should help to demonstrate the following state descriptions. In Figures 11 through 14, the red legs indicate legs that are on the surface supporting the BioCrab. The other legs are up and moving to the idle position.

<u>*STATE 1:*</u> In this state, we force triangle A (legs 0, 2, 4) down. Legs 1, 3, and 5 should already be down (see Figure 11). The triangle down routines take into account the global variable called *height* to allow you to change the walking height of the BioCrab on-the-fly.

Note that when the walker is initialized, all legs are set to the down state at a height of 150. The second time through, the B legs will have been in their moved position.

<u>*STATE 2:*</u> Here, we move the triangle B (legs 1, 3, 5) up. The amount of lift from the down position is set by the global variable called *leglift*. Also, at the same time we are moving these legs up, I am returning them to their center (idle) position as shown in Figure 12.

This is a busy state because I calculate the movement of each of the triangle A legs. Several variables are looked at here that I will go into later. This leg movement is what actually moves the BioCrab.

<u>*STATE 3:*</u> Remember that we were repositioning the B triangle so we know that once we get to this state, we simply only need to move them down. This is the B version of State 1. At this point, all legs are once again in the down

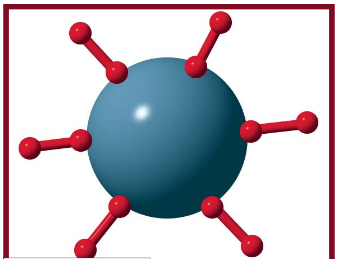


FIGURE 11

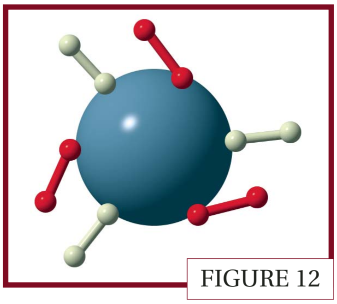


FIGURE 12

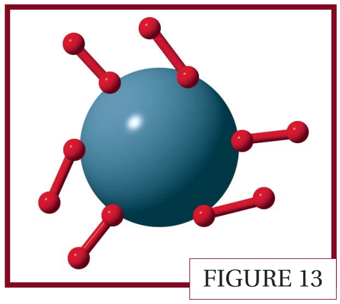


FIGURE 13

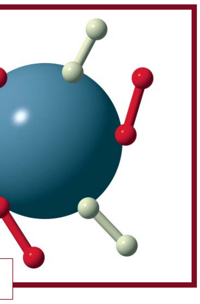


FIGURE 14

position as shown in Figure 13.

STATE 4: This is a mirror image of State 2. We move the A triangle up and set the now-down B triangle into motion as shown in Figure 14. Then we cycle back to State 1 and the whole thing starts over again.

Although it is hard to show, these states are not finite. For instance, the state only initiates the movement of the legs. There is a calculated delay in-between each state in which the legs are moving. I have played with a six state gait. It is a little less smooth but can handle a rougher terrain. In a six state gate, we add the additional states 2 and 4 into two parts. Before sending the B or A legs to their idle state, we first move them up.

The real calculations are done in the moveleg function. The moveleg function calculates the position it needs to move based on the following global variables:

- *State* — The current state the State Machine is in.

- *spin* — The amount of momentum clockwise or counterclockwise the BioCrab is performing. The range is in 100-500, with 300 indicating no momentum.

- *fwd* — The amount of momentum forward or backward the BioCrab is performing. This range is also in 100-500, with 300 indicating no momentum.

- *strafe* — The amount of momentum side to side the BioCrab is performing. Again, the range is in 100-500, with 300 indicating no momentum.

- *height* — The walking height of the BioCrab. This has the range of 0-200.

- *speed* — The walking speed with a range of 1-5.

- *leglift* — This is the amount the leg is lifted off the surface. It has a range of 0-200.

Once the new position is calculated, we can calculate the actual amount of movement. This is needed so that we can set the speed of the AX-12 we are moving. For the most part, we want all the legs put into motion to complete their range of movement at the same time. The AX-12's ability to set the speed is what allows this function. The BioCrab walks much smoother with this ability.

Walker Example

The program BioCrabDios.txt is a walker program. It will cause the BioCrab to walk forward at the slowexst speed, and then walk backwards. The BioCrab will then spin at speed 3 and start the cycle again.

Take a look at the code and you will notice that I created a variable called *statecounter* that increments each time the main state machine handler *walkerAX()* is called. This is used to determine how long the BioCrab does a particular movement. Notice that we make a call to the routine called *initwalkerAX()*. This is needed to set some of the default variable states, as well as initialize the AX network library.

My Old Favorite

Yes, once again I have included a small controller program that uses an IR remote to control your BioCrab. The program is called *BioCrabIR.txt* and while it does not give you total control over your little friend, it will allow you to put it through its paces.

Figure 15 shows how to connect the Vishay IR module. Use any Sony remote to control the BioCrab.

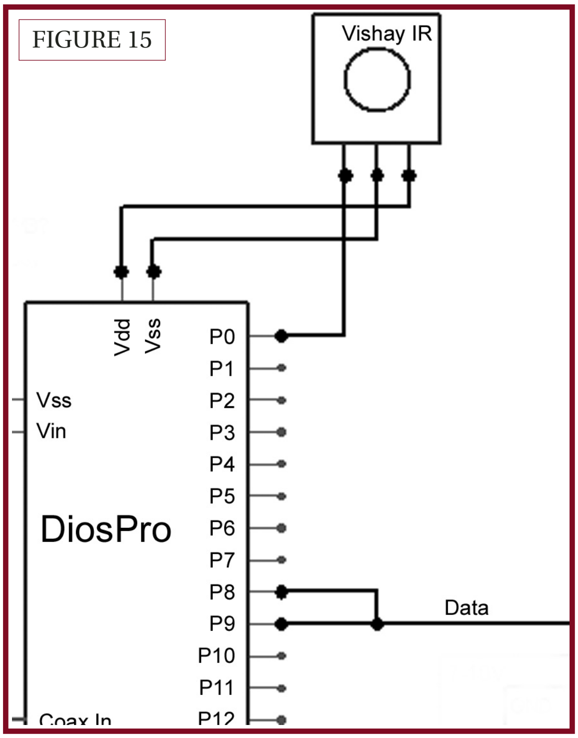
VOL = Spin
CH+/CH- = Fwd/Rev
Reverse/Forward = Strafe

Going Further

Are the WalkerAX routines perfect? No. I plan on doing a lot of tweaking. One thing I can see is instead of moving the legs back to the idle center position, I could set them up for a better stride.

Also, I haven't tested for any AX-12s that may have been shut down. There is plenty of processing time left to do this, as well as other functions. The program

FIGURE 15



PARTS LIST

AVAILABLE FROM CRUSTCRAWLER (www.crustcrawler.com)
- 18 AX-12 Smart Servos
- Bioloid frame set
- CM5 or equivalent (used for power distribution)

AVAILABLE FROM KRONOS ROBOTICS (www.kronosrobotics.com)
- DiosPro chip #16148 (also available from CrustCrawler)
- Dios Workboard Deluxe #16452 (also available from CrustCrawler)
- DiosCompiler free download from the Kronos website
- Vishay IR module #16226

MISCELLANEOUS
- Eight 3/8" #4 machine screws, Jameco #40969
- Four #4 washers, Jameco #106826
- Four #4 standoffs, Jameco #77519
- Plastic bowl (see text)
- Six 1/2" x 2-1/2" PVC pipes
- Six 3/4" PVC caps
- Six 3/4" x 1/2" PVC bushings. This is an adapter so that you can insert the 1/2" PVC pipe into the 3/4" cap.
- Six 1/2" rubber feet. These need to fit tightly over the 1/2" PVC pipe.
- Nine-pin serial cable available from various sources. Needed to program the DiosPro.

The PVC and rubber feet can be purchased at most hardware stores or home centers.

Find Jameco Electronics at ***www.jameco.com****.*

spends much of its time waiting in delay routines. One option I plan on providing is the ability to turn off all these delays so that you can do the delay yourself. This, in turn, will allow you to place AX-12 monitor routines in your main loop. You could then watch for too much load or lack of position updates. This would indicate that one of the legs has come in contact with an object.

Feel free to play with the example program or add some sensors. There are plenty of sonar and various other interfaces on the Kronos Robotics website. The AX network library also has support for the AX-S1 sensor.

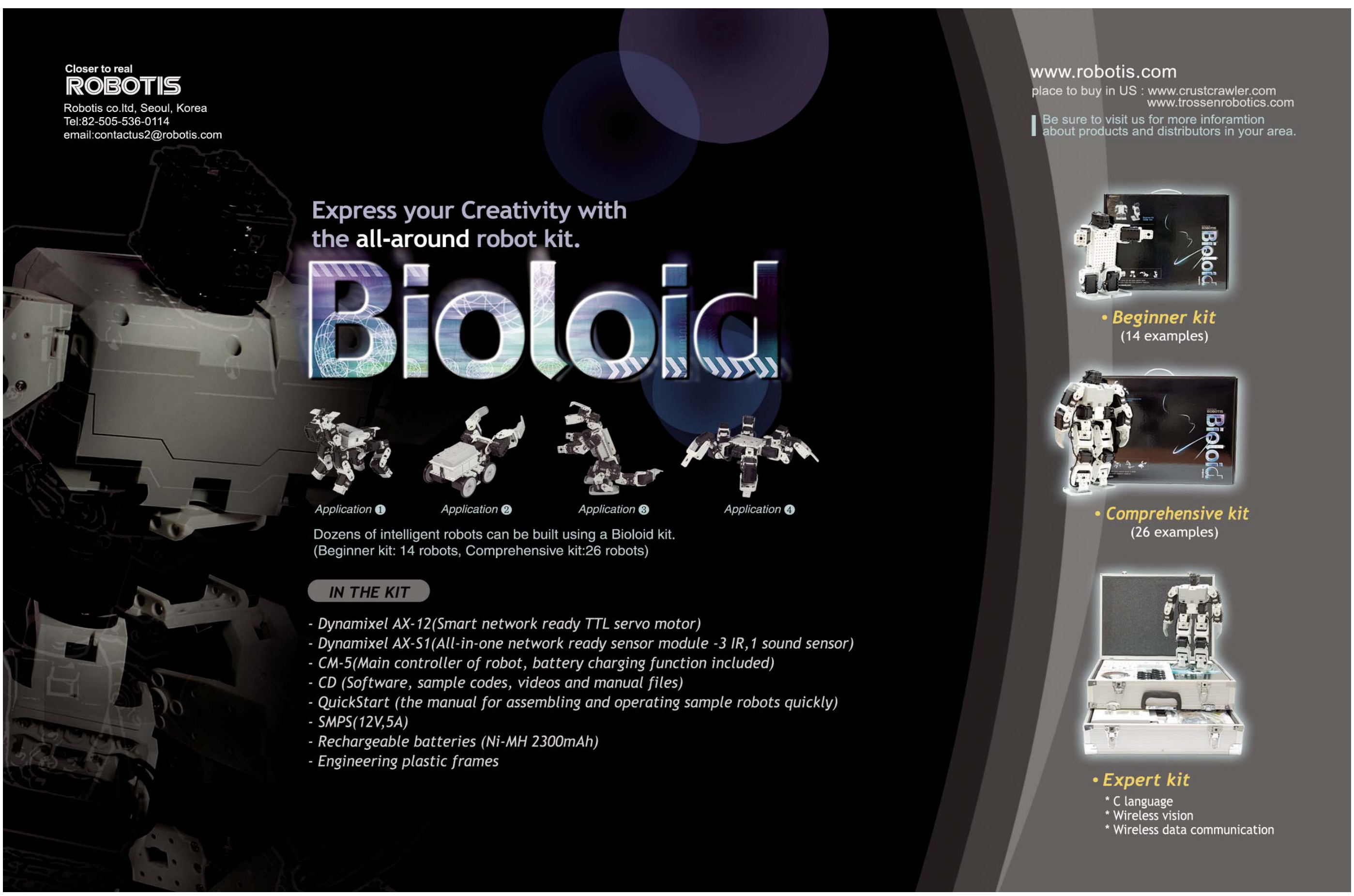
We used the CM5 for power distribution. Feel free to replace it with your own battery pack and cable harness. If you decide to do this, I recommend wiring in a main power switch.

What's Next?

I mounted the Dios Workboard on top of the base in order to do various experiments. One such experiment is a very cool remote control system I plan on interfacing in the next article.

Eventually, I will move the Dios Workboard to the inside of the base so that I can add some sort of graphical display to the top. I'm thinking of a giant animated mouth.

All the example programs, as well as the source, are available for download at **www.kronosrobotics.com/Projects/biocrab.shtml**. **SV**



FAILSAFE DESIGN *and* IMPLEMENTATION *for* RC ROBOTS

by Jay and Valerie Johnson

FIGURE 1

The image of dangerous robots run amok is rampant in popular entertainment, but with a little planning and some common-sense procedures, your robot can avoid falling into this stereotype. Whether you build line-followers, sumos, autonomous navigators, ribbon climbers, or hulking combat robots, the need to control your robot's actions reliably is a critical component of its overall operation. Not only is the ability to stop your robot when needed important for your own safety but, in fact, most events will require you to demonstrate proper failsafe behavior before you are allowed to compete.

CONTROL OR CHAOS?

While there are many ways to control your robot, one of the most common is through an FM or PCM radio link. If the link between the radio and the receiver is compromised, control may be lost — and that's the moment you need a failsafe to take control and shut things down safely. Basic conditions that can cause control failure in radio controlled systems are loss of radio signal and interference or "noise."

Loss of signal can have many causes, including physical blockage of the transmitter signal (a particular concern in enclosed arenas), poor receiver antenna placement or geometry, failure of the transmitter or receiver, or excessive range from transmitter to receiver. Interference similarly has no single cause, but a few of the potential ones include two operators transmitting on the same radio channel, inadvertent noise from nearby electronic equipment, or even the noise from motors or wiring within the robot itself. This article will focus on the procedures necessary for reliably and routinely ensuring proper failsafe behavior in your radio-controlled robot.

CLIMBING THE WALL

Our example for practical discussion of failsafe systems will be Heavy Metal Noise (HMN) — a two wheeled, 120 pound combat robot with twin vertical flywheels (Figure 1). HMN is a veteran of the combat circuit and is controlled using a Futaba 6XAS radio and R138DP PCM receiver. The 6XAS is a commonly used helicopter radio that has been refitted to use ground frequencies (75 MHz). Like many popular R/C radios, the Futaba comes with built in failsafe capabilities. However, it is important to understand the default failsafe setup on your particular radio and to learn how to alter this setup to work properly with your robot's actuated systems.

Our team found this out the hard way when we were just getting started in robot competitions. HMN had its first competition experience in Phoenix, AZ at BotBash 2001. During technical inspection, our inspector Dan Danknick (a long time gearhead who went on to claim his 15 minutes of fame as a "Robotica" commentator and former editor of *SERVO*), asked us to show forward movement with the robot and then to turn the radio off while the throttle stick was still pushed forward.

This is a commonly used test to demonstrate the performance of the failsafe. When we turned the radio off the robot continued to move forward on its own until it encountered a wall, which it promptly tried to climb (unsuccessfully). We had to quickly turn the radio back on to regain control of the robot. We were sent back to the pits with our tails between our legs to properly sort out our failsafe operation.

In fact, the failsafe had performed exactly as designed — for an airplane. The factory default failsafe behavior for the 6XAS causes the receiver to hold the last valid signal. If the throttle is on full

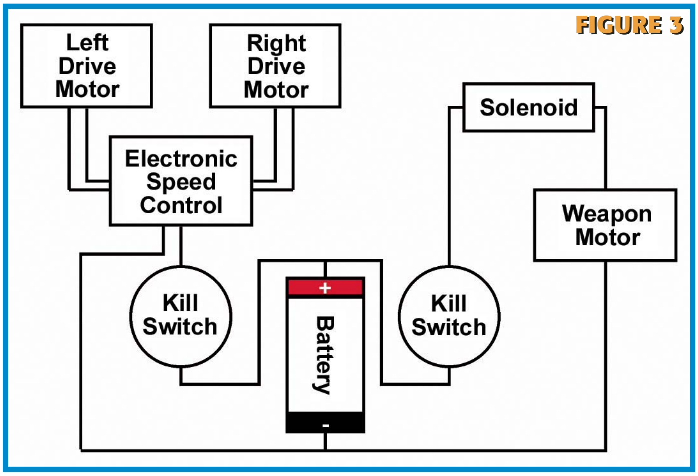


FIGURE 3

FIGURE 2

when the signal is lost, the throttle stays on full until it receives another valid command. In an airplane, this makes sense; it keeps the plane from falling out of the sky until a valid signal is regained. In a robot, this is not such a good idea! Back at our pit table in Phoenix, we read through the radio's user manual and used the in-radio menu system to reprogram the Futaba to return the throttle to zero whenever the failsafe engaged. With this small — but critical — change, we passed our technical inspection with no other glitches and went on to enjoy many successful fights.

Incorrect failsafe behavior in a drive train is bad enough, but imagine the hazard if the robot's weapon system similarly misbehaves! While most of the time you will want to set the failsafe behavior for all channels to return to a neutral or zero position, it is possible to set a different failsafe behavior for each of the channels used to control the robot.

If you were building a robot with a powerful flipping arm, you would want the channels controlling the drive train to return to zero but keep the channel responsible for triggering the arm in the "untriggered" position. Consult your radio's user manual for specific details on programming your radio failsafes.

GENTLEMEN, START YOUR ENGINES

Another important aspect of a failsafe system is establishing robust startup and shutdown procedures. In order to design and implement reliable procedures, it is important to understand how and when failsafe information is sent from the radio to the receiver. According to its user manual, the Futaba 6XAS radio failsafe settings are transmitted every 60 seconds. When the failsafe is first sent, the "PCM" icon on the transmitter blinks (Figure 2).

I have observed that the first failsafe transmission occurs fairly consistently about five seconds after the radio is turned on. Consider the practical implications of this — when you turn the receiver on, it may have an unknown failsafe behavior. In a worst-case scenario, someone else operating improperly on your channel could unintentionally send incorrect failsafe settings to your robot! All Robot Fighting League sanctioned events use frequency clips to minimize the possibility of two radios transmitting on the same frequency at the same time.

One of the best ways to design safe behavior into any robot, regardless of the control system used, is to incorporate two manual power switches — one power switch for the receiver and a second switch (often referred to as a "kill switch") for the robot's main power system. Providing a way to turn on only the receiver allows you to make sure that the proper failsafes are enabled before you power-up the drive or weapon systems on your robot.

Our team likes to go one step further and segregate our drive power system from our weapon power system. We route all drive power through one switch and all weapon power through another switch (Figure 3). We use the popular "Mondo" switches (Figure 4) available from Team Delta (part # RCE44) for both our drive train and weapon system kill switches and a Futaba SWH13 Switch Harness & Charge Cord (Figure 5) for our receiver power switch. This arrangement also has the advantage of making it much easier and safer to do "wheels up" testing of the drive system. We can power-up the drive train with no danger of the weapon powering up.

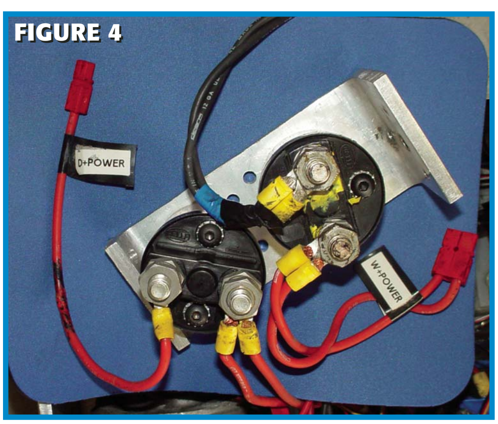


FIGURE 4

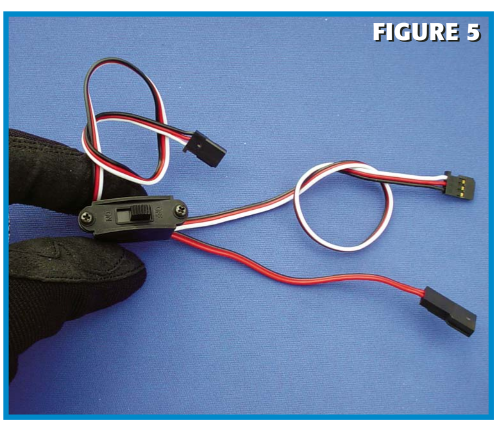


FIGURE 5

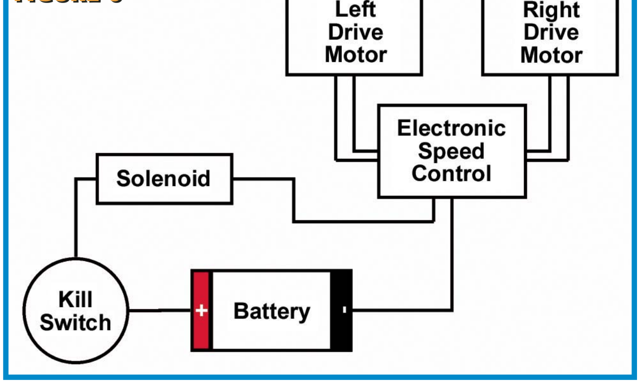
HMN STARTUP AND SHUTDOWN PROCEDURES

Let's take a look at the proper startup sequence for Heavy Metal Noise:

1) Turn on the receiver switch.
2) Turn on the radio transmitter.

3) Watch the radio until the PCM blinks. Failsafe has now been communicated to the robot's receiver.
4) Turn on main power to drive train.
Turn on main power to weapon system.
5) Remove physical safety restraints (a steel pin that prevents the weapon from spinning).

FIGURE 6



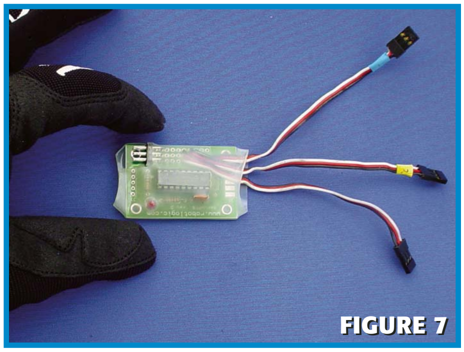


FIGURE 7

The shutdown procedure:

1) Turn off main power to weapon.
2) Turn off main power to drive.
3) Turn off receiver switch.
4) Turn off radio transmitter.
5) Insert physical safety restraint.

These startup and shutdown procedures continue to evolve over time and at the direction of safety staff for specific events. It is important to standardize your startup and shutdown procedures and follow them every time. If your process is complex, you may find it helpful to tape a copy of the steps to your robot or radio.

BATTERY FAILSAFE

Many radios — including the Futaba 6XAS — have an additional built in feature called "battery failsafe." This is a completely different type of failsafe than the signal loss or signal interference failsafe discussed previously. When the receiver's battery voltage drops below approximately 3.8V, the receiver will bring the throttle channel down to a high idle.

In the world of airplanes and helicopters, this would allow you to land before experiencing total loss of battery power. In the world of robots, this would likely appear as a serious loss of drive power. By moving the throttle to idle, you can temporarily reset the battery failsafe and normally control the throttle for about 30 seconds. To avoid encountering a battery failsafe condition, simply make sure your receiver batteries are well charged at all times.

REMOTE POWERDOWN OPTION

When you are designing a safety system for a specific robot, you must consider the relative hazard posed by that robot operating in an uncontrolled fashion. After all, a misbehaving RoboSapien is probably more of a hazard to itself than to others. On the other hand, an out-of-control bomb disposal robot could have lethal consequences. Though not required by most competitions, for the ultimate in safety, you could implement a remote powerdown system. In this type of system, a circuit is included in the robot's control system that permits power to be applied or removed from all the robot's active systems via remote control (Figure 6). Furthermore, the failsafes are set up so that if the signal is lost, the power is automatically interrupted.

DEDICATED FAILSAFE MODULES

In addition to the failsafes often available through the radio, there are other failsafe devices that can be added to the robot to ensure safe failure in subsystems. The IMX-1 from RobotLogic (Figure 7) is a common choice. This modular component installs between the receiver and electronic speed controllers (ESC). It was specifically designed to allow driving a "tank steer" robot with a single control stick. This can be particularly handy if your radio or ESCs don't have a tank steer mixing mode built in.

The IMX-1 also incorporates a handy invert feature that is controlled by an R/C channel to allow easier driving if the robot becomes inverted. But most importantly, the IMX-1 incorporates a failsafe on loss of radio signal.

READY, SET, FAIL!

It is important to understand that not all radio control systems exhibit the same behavior or even similar responses to identical situations. Many of the concepts presented here would translate well to IR-controlled bots or even autonomous robots.

The point of this article is not that you build the same failsafe system that our team uses for HMN. Instead, take the time to understand the joint behavior of your remote control system and your specific robot components to design a robust combination of programming and procedures to maximize the safety inherent in your robot's behavior. So, go ahead and fail! **SV**

LINKS

BotBash
www.botbash.com

Robot Fighting League (RFL)
www.botleague.com

Team Delta
wwww.teamdelta.com

RobotLogic
www.robotlogic.com

Big Bang Robotics
www.bigbangrobotics.com

ABOUT THE AUTHORS

Jay and Valerie Johnson are Big Bang Robotics and have been designing, building, and fighting combat robots since 2001.

by Dave Calkins

ROBOGames Prep:

Art Robots — Does it Always Need to Have a Reason?

A sad thing about many robots is that they don't look interesting. They can perform their programs and tasks beautifully, but they just don't stop you visually. That's why there are artbots — to give robot builders a good reference point on what robots should look like. We can't all be Cynthia Breazeal, but we can do a lot to improve how humans view our mechanical creations. There's nothing like a grinning aluminum smile to get people to warm up to your robot.

Some artists paint with pigment and texture, others sculpt with clay and stone, some perform with voice and instrument; we just happen to use microcontrollers, motors, and digital multimeters. Robotics is the major new art-form for the 21st century. Eventually, robots will be as commonplace as computers, but for now, they are rare — yet they should provoke strong emotional responses.

Robotics is fertile ground for artists; lots of social experiments, room for technical improvement, and opportunities to demonstrate creativity. You can't really learn this stuff in class, either. It is street-theater. The main pre-requisite passion — the outcome as diverse as any ecosystem.

Not only are robots "art" unto themselves, but an art-form with the potential to mirror our own humanity — not only in appearance, but in deed. Since the days of "Leonardo Da Vinci," robotics has quietly remained at the nexus point of art and science. Well, my friends, that brings me back to you ... why do we build homebrewed robots? Because they aren't going to build themselves ... yet.

One of the best things about art robots is the lack of constraints in the rules — you can make whatever you can dream up. You can be an experienced programmer or someone who's never picked up a soldering iron. The point of the five artbot classes (shown in the sidebar) is to shift from engineer to artist. Instead of a box on wheels, how about a box with wheels for eyes?

Al Honig, one of my favorite art-bot builders, has a workshop like most of us. And, like most of us, it's filled with junk and scrap (admit it!). Unlike most of us, it's not electronic. Coffee pots, forks, street lights, desk lamps, and sink drains are just a few of the things in Al's shop. To the untrained eye, it looks like a scrap heap. Yet here and there on workbenches, you see these lost items becoming found parts. Odd items come together to form beautiful works of art — a nose appears where

ARTBOT RULES

- All bots are judged by a team of three judges. Each judge scores the robot on several criteria. Scores are totaled and averaged. Bots with the highest score wins. Judging happens throughout the course of the event.

- Static Artbots are judged on three criteria: Perceived User-friendliness, Aesthetics, and Originality. Static art bots do not move. They just look cool.

- Kinetic Artbots move in some manner. This movement doesn't need to be locomotion. Bots are judged on four criteria: Aesthetics, Attractiveness of Movement, Functional Completeness, and Functional Longevity.

- Barbots have to make a mixed cocktail. Actual alcohol need not be used. The four judging criteria are: Aesthetics, Style, Delivery, and Versatility.

- Musical Robots have to play a musical instrument. This doesn't have to be a traditional instrument, but it can't be a MIDI device with speakers. Musical bots are judged on: Creativity, Quality of Sound, and Diversity of Sound.

- Painting Robots have to paint something — they can use markers, oil, acrylic, or other mediums so long as the robot is moving the brush/pen in a two-dimensional axis (printers do not qualify. Judging is based on creativity of movement, aesthetic quality of final painting, and diversity of colors/materials used.

> “Some artists paint with pigment and texture, others sculpt with clay and stone, some perform with voice and instrument; we just happen to use microcontrollers, motors, and digital multimeters.”

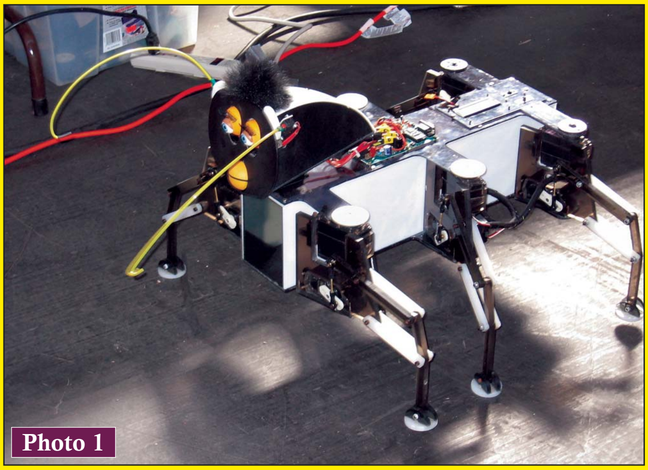


Photo 1


Photo 4


Photo 3


Photo 2


Photo 5


Photo 6

PHOTO 1. Hexapod’s should look cute!

PHOTO 2. “Arca Musarithmica 1” took Blue Man Groups’ improvised instruments one step further.

PHOTO 3. You don’t even need metal to make a kinetic artbot!

PHOTO 4. Johnny Five actually rolls around and can grab things — including your attention.

PHOTO 5. “Science is spectral analysis. Art is light synthesis.” — Karl Kraus

PHOTO 6. Deadly Necrobot. Sure, it’ll give you nightmares — unless you don’t oil it.

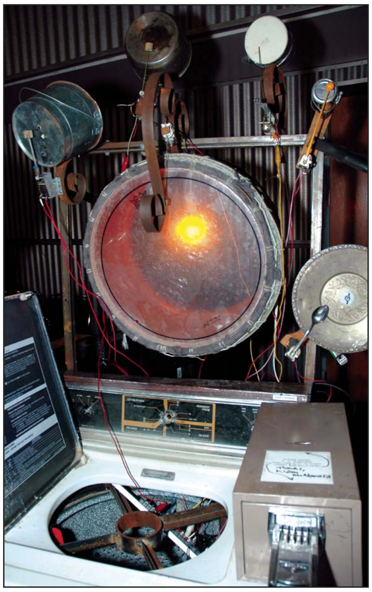


PHOTO 7. "Mama's Calliope," the gold medal winning musical artbot, used the holes in a washing machine like a player pianos. Cost? $0.00.

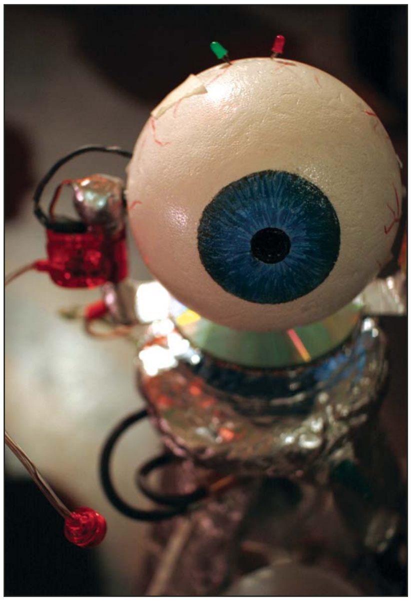


PHOTO 8. Not all that glitters is aluminum.

once there was only a faucet. Door handles are suddenly arms. And tea strainers become eyes.

As Camp Peavy points out, "The robot-artist is the one wondering through the aisles looking for God-only-knows-what. Do not ask if he can be helped ... he is beyond help. He's studying the mechanics of everything, closely. A little too closely. He asks about modifying any and everything; and nothing is safe if a screwdriver is within reach (and, of course, one always is). He waxes poetic about adhesives; tools and drools over anything autonomous. Yes, my friend, if you recognize these symptoms then you might be a robot-artist."

Your inner artist has many outlets — you can make a static artbot which has no motors, but mirrors creation, and gives the spectator new perspective on life. Kinetic artbots take it one step further, and make the robot move. It doesn't need to do any specific task, but it should shift from the dimensional to the fourth dimension, where its position changes with time.

Once you've got the robot moving, why not give it a purpose? And what better purpose than cocktails and music? Cocktail robots must make a mixed drink — Cosmo, martini, mohito — whatever you'd like.

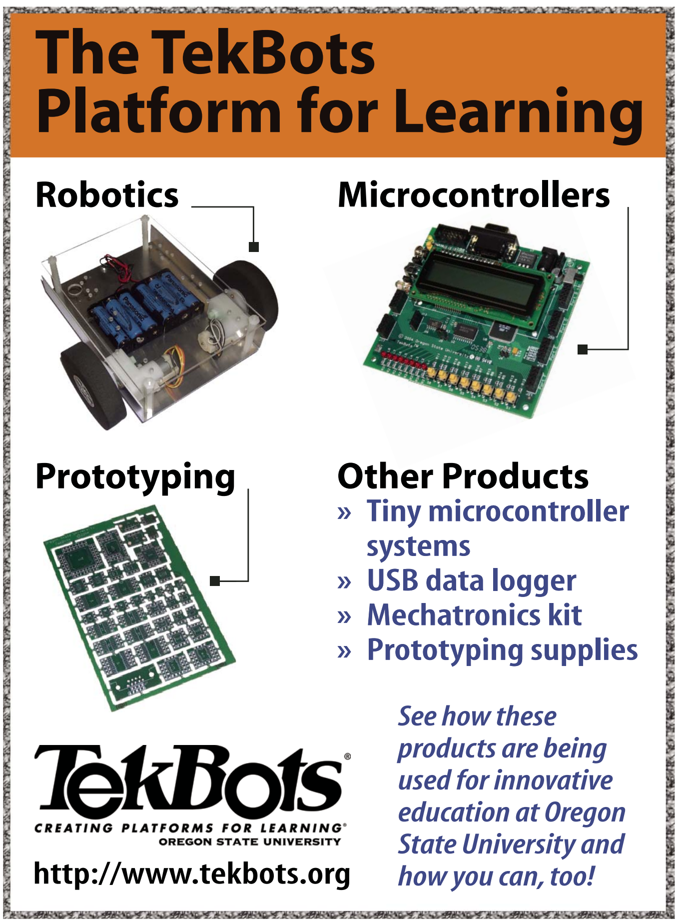
If bars aren't your bag, then switch to music — whether the bot plays the bagpipes or an instrument that you invented along with the robot. Finally, what happens when art imitates life imitating art? Well ... art. Painting robots make paintings or illustrations. Let your robot be as creative as you.

Al has won twice for his static robot, Sentry #2, which looks like a steel soldier guarding an unseen plastic princess. Last year's gold medal winner "Deadly Necrobot" looked like the unholy offspring between a Tyrannosaurus and a Cadillac.

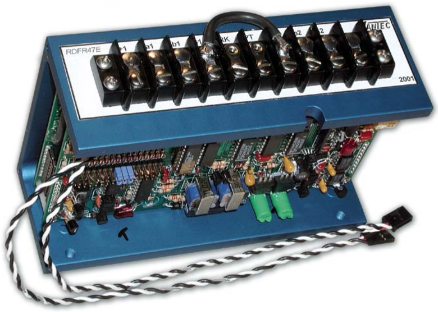
Camp's kinetic artbot, "Springy Thingee" is a medal-winning kinetic bot, because "since it doesn't actually do anything useful, it must be art! Springy is a robot I built for the Burning Man festival after Robot Wars folded. I billed her as an artbot as opposed to the fighting robots that I had been building. Springy is a regular at the HomeBrew Robotics Club meetings at Carnegie Mellon West, where she's usually remote control, but does have an autonomous feature where she follows an infrared beacon; as for being practical, she carries my toolbox and computer."

Of course, your robots don't have to look to the future. I Wei's walking robots rely on steam power to move their legs. And his steam punks took home two gold medals last year. Watching their water powered engines slowly turn tiny flat feet inspires both awe and minds.

The photos in this article should serve to inspire you, as well. Even if you don't make an artbot, I hope that your robots take on a new dimension. **SV**



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

Extreme Robot Speed Control!

Sidewinder

$399

- 14V - 50V - Dual **80A** H-bridges - 150A+ *Peak!*
- Adjustable current limiting
- Temperature limiting
- Three R/C inputs - serial option
- Many mixing options - Flipped Bot Input
- Rugged extruded Aluminum case
- 4.25" x 3.23" x 1.1"

$29.99

Scorpion Mini
- **2.5A** (6A pk) H-bridge
- 5V - 18V
- 1.25" x 0.5" x 0.25"

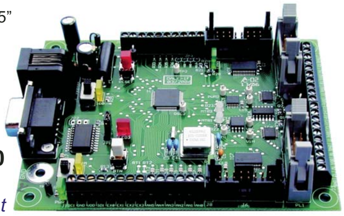
$79.99

Scorpion HX
- Dual **2.5A** (6A pk) H-bridges
- Plus **12A** fwd-only channel
- 5V - 18V
- 1.6" x 1.6" x 0.5"

$119.99

Scorpion XL
- Dual **13A** H-bridge *45A Peak!*
- 5V - 24V
- 2.7" x 1.6" x 0.5"

Introducing Dalf
- Closed-loop control of two motors
- Full PID position/velocity loop
- Trapezoidal path generator
- **Giant Servo Mode!**
- PIC18F6722 CPU
- C source for routines provided
- See *www.embeddedelectronics.net*



$250

H-bridges: Use with Dalf or with your Micro/Stamp

$199

OSMC
- Monster power!
- 14-50V **160A!**
- 3.15"x4.5"x1.5"
- 3 wire interface

$79

Simple-H
- 6-28V **25A!**
- 2.25"x2.5"x0.5"
- 3 wire interface
- current & temp protection

ROBOT POWER™

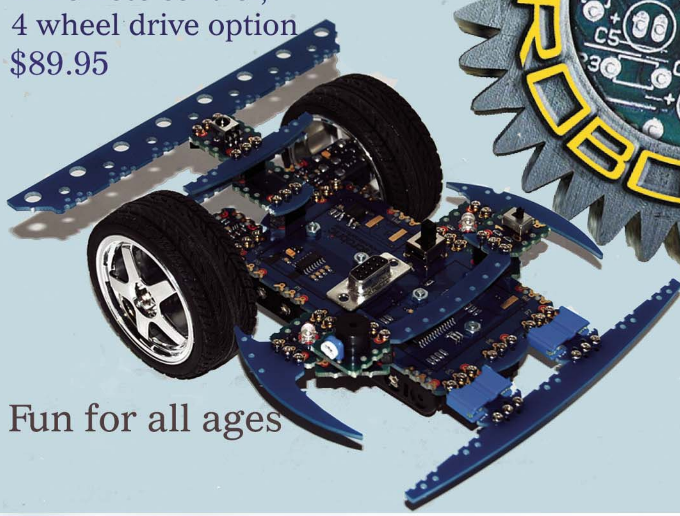
MADE IN THE USA

www.robotpower.com

Phone: 253-843-2504 • sales@robotpower.com

Programmable Robot Kits

MicroBric Viper
Basic Atom & Software
Dual motors & Drivers
Bolt together, no soldering
Dual Bump Sensors
IR remote control,
4 wheel drive option
$89.95



Fun for all ages

MAXIMUM ROBOTICS

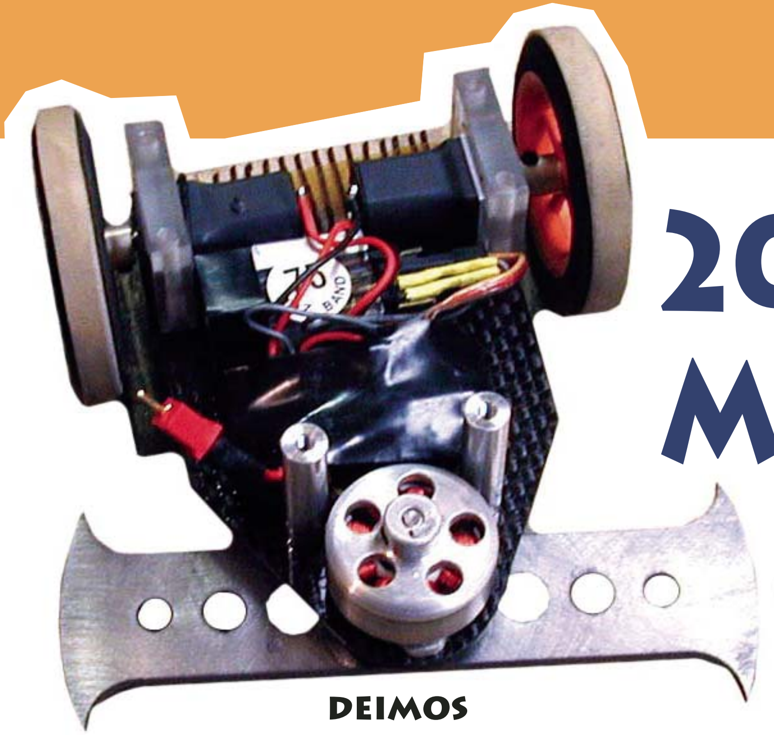
MicroCamp Mega8
Atmel Mega 8, Dual motors,
Bump & Line Following Sensors
No soldering, Extra Inputs
$59.95

Micro-Mouse Starter Kit

1-800-979-9130
MAXIMUMROBOTICS.COM

2007 NERC MOTORAMA

BY PETE SMITH



DEIMOS

This marked the fifth year that N.E.R.C. (Northeast Robotics Club, **www.nerc.us**) have held their big, annual competition as part of the Motorama motor sports event in Harrisburg, PA (**www.motoramaassoc.com**).

The competition ran February 16-18, with the Fairy and Ant weight events being held on Friday, while the Beetles, Hobby, and Featherweight rounds took place on Saturday and Sunday.

Harrisburg was tightly in the grip of winter with several inches of snow on the ground and bitterly low temperatures. The wide Susquehanna River was frozen all the way across and many of the minor roads were still being cleared. This appeared to have affected the Motor sports enthusiasts, as turnout was down in the other arenas. However, it had little, if any, effect on the numbers in the Robot competition.

There was a new class competing for the first time. The Sportsman class came about as an answer to the growing power of the kinetic energy weapons. Some felt that the enormous destructive power of bots like "Totally Offensive," "Relic," and "Kilobyte" was driving out the other innovative designs that could not afford the weight allowance to adequately protect themselves.

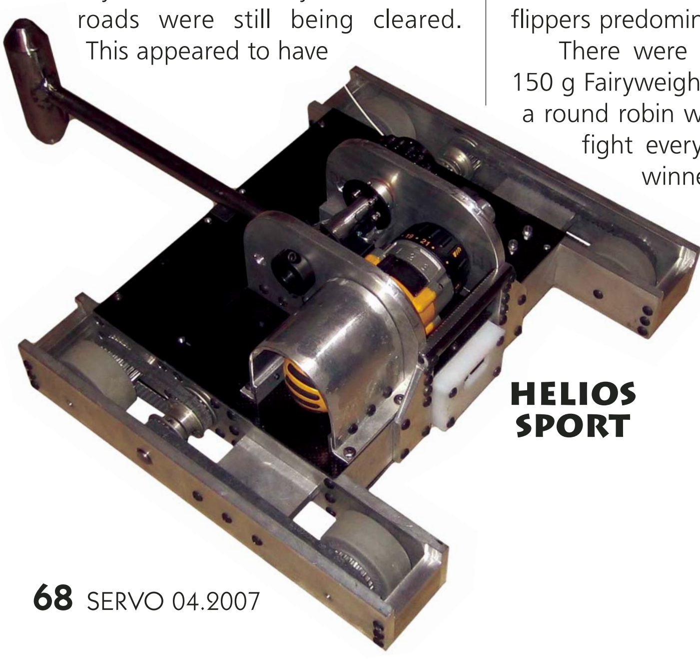
The new class strictly limits the rotational speed of any weapon and also bans ground-hugging wedges. There were 10 entries for this class with hammer bots and pneumatic flippers predominating.

There were only four of the tiny 150 g Fairyweight bots, so they fought a round robin where each bot had to fight every other bot and the winner was the one who won the most fights. "Deimos" from Team Cosmos dominated, winning all three of its fights taking the overall first place and repeating his success in winning in 2006. "Mr Bigglesworth" took second losing only to "Deimos."

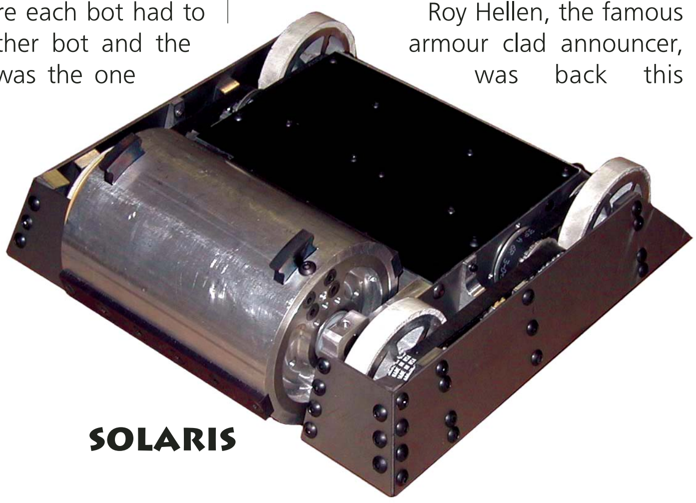
There was a bigger field of Antweights — 25 in all. The fight brackets were the usual double elimination where every bot has to lose twice before being eliminated. "Switchblade," built by Team Sawzall, used its Brushless motor driven "Eggbeater" blade to power through the brackets to take first place. A classic 2WD Wedge "Fender Bender" from Team JandA was second, losing only in their two fights against "Switchblade."

The bigger bots started going through safety on Friday evening as the big 16' x 16' arena was being assembled and continued Saturday morning with fights getting started early in the afternoon.

Roy Hellen, the famous armour clad announcer, was back this



HELIOS SPORT



SOLARIS

year with his inimitable style of commentary which always keeps the crowd and competitors entertained.

Fights alternated between the Beetles (3 lb), Hobbyweights (12 lb), Featherweights (30 lb), and the new Sportsman class thoughout the afternoon and early evening. There were teams from all over the country and at least a couple of teams from Canada. For some people, this was their first robot and first event, while others were regulars with improved versions of bots that have competed before.

The fights stopped for the day around 8.00 pm with over half the bots knocked out.

The competition restarted around 10:30 on Sunday. The crowds were much smaller (apparently the conflict with the Daytona 500 cuts numbers for the second day), but the fights were just as good if not better. Quarter, then semi finals cut the numbers left in each class, but the excitement was kept up by a number of Rumbles, Grudge, and other fun fights filling the gaps as bots took time to repair damage and recharge their batteries. A two-against-two battle was fought with a spinner and wedge on each team. The "British Commonwealth" team with drivers from Canada and Scotland handily beat their American rivals this time, but I can see this contest being re-fought next year!

In the Beetle class, HockeyRunner Robotics "Can of Nuns" took a surprise first. Its reliability and toughness beat out several more deadly looking rivals including Novarobots "Enemy" which had to settle for second place.

An old favorite "Solaris" built by Team Cosmos and relative newcomer Team Sawzalls "Darkblade" battered through the brackets of the 12-lbers before "Solaris" finally won first prize when its opponent could not self-right after being flung upside down.

A smaller than usual number of 30 lb Featherweights entered this year. Some flipper bots like Team Neubots "Dark Thunder" have moved over to the new Sportsman class and top spinners like "Helios" and "Totally Offensive" that have dominated the last few years did not attend. The eight bots that took part still put on a good show. Spinners again dominated with "Tripolar" having a good run, demolishing "Whamo" before a weapon drive problem resulted in a loss to the Canadian, vertical spinner "Sloth." The overall winner was another vertical spinner, Robotic Hobbies "Billy Bob" who beat "Sloth" twice in a row to grab the first prize.

The Sportsman class proved popular with the crowds and proved that a less destructive 30-pounder can still put on a good show. The flippers dominated over the other bots but "Helios Sport" did the best of the hammer bots to get third place.

The hammers didn't seem to be doing much damage but I think that will change by next year. The pneumatic flippers are better developed and I think have less headroom for improvements.

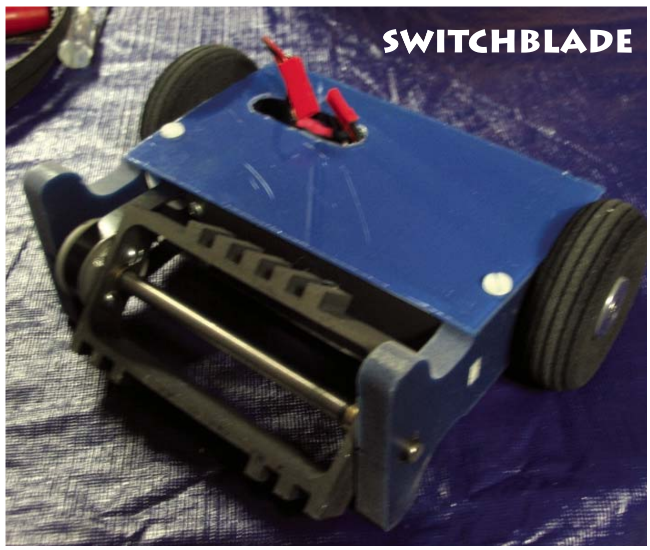
One thing required by both types is improved driving skills and that was well demonstrated by "P.I.T.R." and "Bounty Hunter." "P.I.T.R" was built to compete in the Featherweight class so used up a lot of its weight in armor, while "Bounty Hunter" was built for the new class so did not have to make the same compromises. This advantage probably helped "Bounty Hunter" to take first place.

The competition was over by late Sunday afternoon and the prizes provided by Team Wyachi, Titanium Joe, and KitBots were given out by Miss Motorama.

Several teams stayed after the competition to help dismantle the arena and pack it into the trailer. They were rewarded with a good meal at a local steak house.

Motorama is proving to be the premier event in the northeast and I can definitely recommend it for anyone wanting to compete in the smaller weight classes. SV

DARKBLADE



SWITCHBLADE

Photos courtesy of Team Cosmos and Team Sawzall.

EVENTS CALENDAR

Continued from page 23

May

4 Robot-SM
Sweden
Autonomous Sumo and mini Sumo event. There's no English version of the website, so if anyone can pinpoint the location a little more precisely, let me know.
www.robotsm.se

4 SPURT
Rostock-Warnemunde, Germany
Robots race on the official SPURT track.
http://spurt.uni-rostock.de

5 Eastern Canadian Robot Games
Ontario Science Centre, Ontario, Canada
This event includes mini Sumo, line-following, walker race, and a new event for beginners called Search and Rescue.
www.robotgames.ca

8-9 Haifa Robot Competition
Haifa, Israel
Autonomous robot events for university students, high school students, and amateurs.
http://math.haifa.ac.il/robotics/competition

11-12 SwissEurobot
Yverdon-les-Bains, La Marive, Switzerland
This is a regional version of the main Eurobot competition which will be held on May 16-20 in France.
www.swisseurobot.ch

12 Chibotica
Museum of Science and Industry, Chicago, IL
Line-following, maze solving, Sumo, and a robot talent show.
www.chibots.org

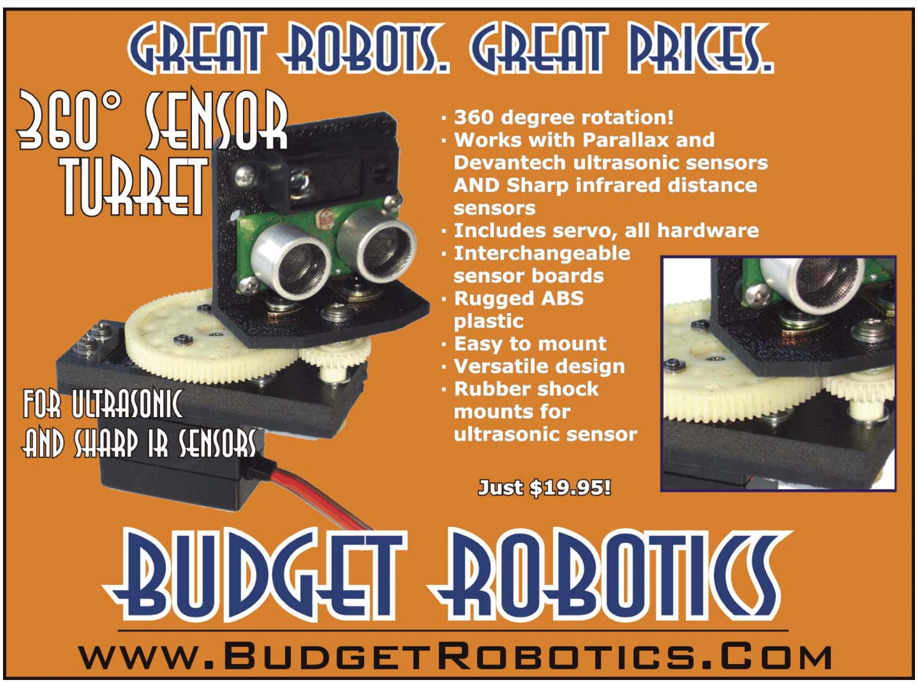
12 Western Canadian Robot Games
Alberta, Canada
This event includes robot Sumo (western rules), mini Sumo, walking robot triatholon, robot art contest, Atomic Hockey, and a full set of BEAM events.
www.robotgames.net/robot_games.htm

16-20 Eurobot
Ferté-Bernard, France
This year's event is called the Robot Recycling Rally. Autonomous robots must sort waste into appropriate recycling bins.
www.eurobot.org

19-20 Mechwars
Eagan Civic Arena, Eagan, MN
In this event, radio-controlled vehicles destroy each other in Minnesota.
www.tcmechwars.com

19-20 PDXBOT
Portland, OR
Includes lots of autonomous Sumo events, as well as line-following and Robo-Magellan.
www.pdxbot.org

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



ROBOLYMPICS

www.robolympics.net

San Francisco, California - June 15-17 2007

Robots from around the world invade San Francisco for the Third Annual International ROBOlympics!

Join us in 2007 for the only international combined-event robot competition. Compete in any of 60 different competitions, including no-cost junior league events (for kids 17 and under.) At the 2006 event, over 650 engineers from 13 different countries competed with more than 450 robots! No matter which event you compete in, you'll still meet hundreds of robot builders from around the world, see new robots you never knew existed, and learn more about robots than you imagined possible!

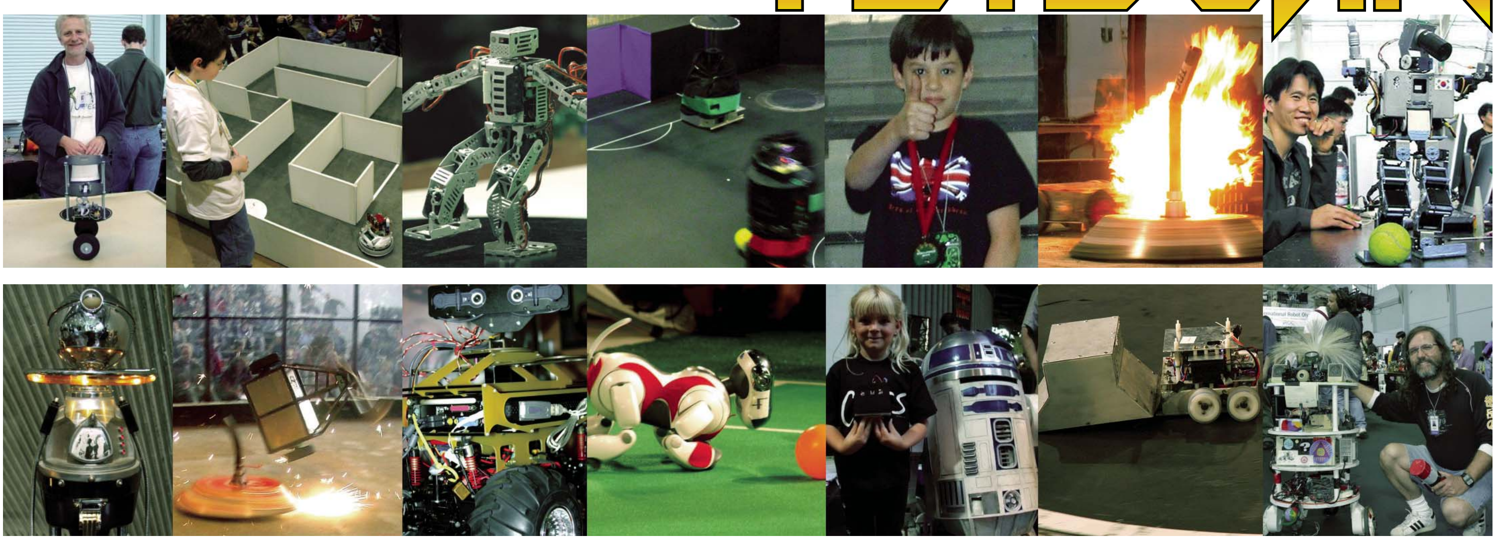
Don't miss this show!

Sponsorship opportunities are still available for the 3nd Int'l ROBOlympics! Take advantage of the tremendous publicity potential! Robotics is becoming the new lifestyle sport of the thinking age. The Renaissance had music, we have DC motors. ROBOlympics is the channel to a growing audience with both a voice and a high disposable income. Contact us for info on this exciting opportunity: sponsors@robogames.net

Events at ROBOlympics:

Combat: 150g-340lbs and 2 autonomus classes;
Robot Soccer: Humanoid, MiroSot, NanoSot, RoboSot, Aibo, Small League, Middle League;
Sumo: 25g - 3kg;
Open: Robot Triathlon, Ribbon Climber, Line Slalom, Mindstorms Challenge, Best of Show, Maze Solving, Aibo Performer, Balancer Race, Fire Fighting, Mindstorms Open, Table Top Navigation, Vex Open, Vex Challenge, Biped Race, Walker Challenge, Robomagellan
BEAM: Speeder, Photovore, Robosapien Hacker
Junior League: Lego - Woots and Snarks, 120 lb Combat (Jr), Mindstorms Challenge, Best of Show, 500 g Sumo, Handy Board Ball, Mindstorms Open, BasketBall Challenge, Vex Open, Vex Challenge
Robo-One: Wrestling, Agility, Stair Climbing, Door Opening, The Eagle, Toss
Art Bots: Musical, Static, Kinetic, Painting, Bartending
Bot Hockey: 3kg & 12 lb
and of course:

TETSUJIN



The Third International ROBOlympics is sponsored by

Robotics Society of America

COMBOTS

SERVO MAGAZINE

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

Joining an Online Robotics Community

I still remember the first time I went online. It was 1983 and I had just signed up with the CompuServe Information System. My trusty Osborne 1 computer was decked out with a 300 baud acoustic modem — the kind you plugged the phone handset into — and before long, I was riding the information superhighway. Only back then it was more like the information trail. Bumpy, and not yet fully paved.

As with the Internet today, one of the draws of CompuServe and the early online services was the user forums — places where like-minded people could congregate, ask questions, and share ideas. There are countless such user-to-user communities spanning interests from cake baking to child rearing to — wouldn't you know it! — robot building. In this month's installment of Robotics Resources, we'll look at Internet forums and newsgroups.

Stalking the Internet Newsgroups

As you may know, while the World Wide Web is the primary duct used to transport information around the Internet, there are other mechanisms, as well. Most go by acronyms that describe their protocol: HTTP is for the Web, FTP is for file transfer, and NNTP is for newsgroups. It's this last protocol that many folks — especially those new to the Internet — are unfamiliar with.

Internet newsgroups (also called Usenet) are composed of tens of thousands of public and private user groups. They are defined within a naming hierarchy, and many start with alt (alternative), comp (computer), or rec (recreation). From there, the groups are further divided into subgroups, and subgroups within those. For example, comp.lang contains many Usenet groups devoted to computer languages. Within this subgroup, you'll find **comp.lang.java**, **comp.lang.python**, and **comp.lang.perl**.

There are only a few robotics-specific Usenet groups — comp.robotics.research and comp.robotics.misc — and the former is not that active. But in the spirit of Usenet, topic coverage is well-defined. If you have a question about machining a part for your robot, you can post it in comp.robotics.misc, of course, but perhaps a better avenue is to ask the subject matter experts in the metalworking forums, such as rec.crafts.metalworking. Some web browsers, like Microsoft Internet Explorer, can't directly access Usenet groups. If your web browser does not support newsgroups, you still have a variety of ways to connect, including:

- If your PC has them installed, use Microsoft Outlook or Outlook Express to join and use newsgroups.
- Download a free or shareware newsreader. One of my favorites is Xnews (**xnews.newsguy.com**). For others try a Yahoo! or Google search using the term "newsgroup reader" (with the quotes).
- Use a web-based newsreader. One of the most popular is Google Groups. If you use a paid Usenet service, such as Supernews or Easynews, check to see if they have a web-based portal.

Finding and Joining Web-based Forums

The popularity of Usenet has dwindled somewhat in recent years. Perhaps it's the added complexity of newsreader software, or the inability to search through postings to find what you are looking for (though Google Groups and others provide this). Taking its place are web-based forums, made all the more popular thanks to cheap and low-cost web bulletin board programs, such as phpBB. These programs can be easily added to a website, often at the touch of a button. They're easy for website owners to add to their pages, and they run all on their own.

By their nature, web-based forums do not require any special software apart from your web browser. Just go to the main page of the forum, join or log in (if required), and read the posts.

A good example is the *Nuts & Volts/SERVO Magazine* bulletin board, found at **www.servomagazine.com/forum**. The forum is divided into five main groups, or headings. As you'd expect, one of the groups is dedicated to robotics. Click on the Robotics group, and you're greeted with hundreds and hundreds of topics, sorted in reverse chronological order (latest first). The bulletin board shows you the number of replies in a topic, who started the topic, and the last post. You can even get a general idea about the interest level of a topic by looking at the numbers of views it's had. A search feature lets you scan through topics to find a subject you're interested in.

Most — but certainly not all — web-based forums require registration before you can post a message, either a new topic or a reply to an existing topic. This is unlike Usenet groups, where you do not need to join (unless the group is private, in which case you'll probably need a user name and password).

Some web-based forums also require you to join just to read the topics. Personally, I tend to stay away from these forums, as more often than not, this policy is just a sneaky way for the site owner to collect email names. (Okay, there are some legitimate reasons to require joining just to read, but these reasons usually apply to sensitive and personal topics, and not for things like how to modify a servo for continuous rotation.)

In order to join a forum, you'll need to pick a user name and sometimes a password; some forums provide a random password, but usually let you change it to something more meaningful to you. Most require a working email address, to which the forum software sends an automated registration message. Registration is complete after you respond to this message.

Yahoo!, MSN, and several other Internet portals provide what's known as "managed user groups." They provide the system for running the group; the owner of the group is charged with starting it up, and the members with keeping it going. As an example, accessing a Yahoo! group depends on the forum settings, which vary from group to group. With some, you can browse the topics without having a Yahoo! ID; with others you need to have a Yahoo ID! and be signed in to access the topics. Yahoo!, MSN, and other portals do not charge for the ID.

Using Mailing Lists

An older but still quite valid method of sharing ideas and information with others is the mailing list (also called a list server). With these, members exchange group emails, which are received via your regular email address. Mailing lists are handy because you can easily delete the messages that are not interesting to you, and store away the ones that are.

A number of web-based forums, such as the Yahoo! groups, have an email list option. You can elect to use the web interface to read and reply to messages, or have all messages sent to your email address.

If you sign up to a mailing list, you'll want to make sure your mail software and/or ISP doesn't view the mail as spam. If you've signed up and you're not getting any email, it may be blocked. Add the originating domain of the email to your whitelist so that it will pass the spam filters.

Posting Rules and Etiquette

You don't have to be online for long to know that people are different when they're sitting behind a monitor, typing at their keyboard, and not talking to someone face-to-face. Online discourse has its own sociology. Most newsgroups have no "owner" per se; the group is a public resource, but contributors are still bound by expected rules of decorum and etiquette. Conversely, web-based forums are owned and operated by a company or individual, who has the last say in what goes on.

The basic rules are simple enough: Be courteous, stay on topic, avoid vulgarisms and profanity (kids use these groups, too), and avoid flame wars with other contributors. These rules not only make it a nicer place to visit, but it helps everyone get the most out of the group or forum. Staying on topic helps others avoid lengthy threads that are not cogent to the purpose of the group. Some off-topic discussions are to be expected, but user-to-user groups tend to go downhill quickly when members constantly veer off to discuss hot-button topics such as politics and religion.

Many forums have written or unwritten rules related to things like quotebacks (including all the text of the previous message), top-posting (putting your reply above the original message, rather than below it), and texting (using shortcuts like "u r" rather than "you are"). In the more strict forums, violating these rules may cause you to be banned, while in others, you simply risk not getting any help at all.

Too, you'll get a better response if you are concise in your questions. If you're having trouble getting something to work, explain enough detail about your situation so that others have enough information to give you a reasonable answer. Be especially careful of the "gimme gimme" post, typified by "Tell me everything there is to know about robots." If anyone responds at all, it'll usually be a stern retort about how to use Google.

Keeping Private Things Private

When you post through a newsgroup reader, you have the option of providing a real, valid email address so that others can communicate with you directly, rather than through the newsgroup. I found this is a surefire way of getting on every spam list in the universe. Consider one of the following three alternatives: providing no email address (some readers do not allow this), using a totally bogus email address (just be sure it doesn't actually belong to someone!), or obfuscate your own email address to make it harder for spam harvesters to understand. For example:

mynamepleasenospam@pleaseno spamdomainname.com.

At the end of your message, in a common "sig" signature file, you can provide a short explanation of how to remove "pleasenospam" from your email address. Most web-based forums do not reveal their user's email addresses, so this isn't as much of a problem.

Consider that messages you post on newsgroups and forums are, for the most part, stored forever. Be careful what you say before you click the Post button. Be especially careful about revealing personally identifying information, if you don't want that information to be made public. If you're under 18, it's best not to indicate your age. It's best to avoid the wackos to begin with.

Finally, before joining any newsgroup or web-based forum, be sure your

PC is outfitted with good virus protection software, and that it has been recently updated. Be sure your operating system has been patched with the latest security releases. Be wary of any newsgroup or forum where members can post binary files. Avoid downloading files, unzipping files, and running programs unless you know the poster, and unless the file has been thoroughly scanned.

Sources

Following is a selection of web-based forums, mailing lists, and Usenet groups specifically about robotics. Included are several forums provided by makers and resellers of robotics parts.

Acroname Forums
www.easierrobotics.com/cgi-bin/forum.cgi
Forums for Acroname products and robot kits.

Art & Robotics Group (ARG)
www.interaccess.org/arg
User group, discussion board, and latest news on the artistic side of robotics.

Central Illinois Robotics Club
http://tech.groups.yahoo.com/group/circgroup/
Yahoo! group for the Central Illinois Robotics Club.

Chicago Area Robotics Group
http://tech.groups.yahoo.com/group/chibots/
Yahoo! group for the Chicago Area Robotics Group.

Comp Robotics
comp.robotics.misc
Main robot-related Usenet group. Need a newsreader to access this group.

Crustcrawler Forums
forum.crustcrawler.com/index.php
Robotics forums at Crustcrawler.

Dallas Personal Robotics Group (DPRG)
www.dprg.org
Projects, tutorials, articles, and mailing list. You can browse the mailing list archives.

Denver Area Robotics Club
http://groups.yahoo.com/group/DenverRoboticsClub/
Yahoo! group for the Denver Area Robotics Club.

Google Groups
groups.google.com
Web-based portal for Usenet and specialty groups.

HomeBrew Robotics Club
www.hbrobotics.org
Offers a list server for email.

Kansas City Robotics Society
groups.yahoo.com/group/KCRS
Yahoo! group for Kansas City Robotics Society.

Laboratory Robotics Interest Group
lab-robotics.org
Look for the Discussion Group.

Lugnet
news.lugnet.com/robotics
LEGO-centric robotics discussions.

Lynxmotion
www.lynxmotion.net
User forum centering around the Lynxmotion robotics kits and parts.

MSN Groups Main Page
groups.msn.com
Main portal page to all MSN groups. Use the search feature to find subject-specific groups.

Nuts & Volts/SERVO Bulletin Board
www.servomagazine.com/forum
Forum sponsored by the publishers of *Nuts & Volts* and *SERVO Magazine*.

Ottawa Robotics Enthusiasts (O.R.E.)
http://tech.groups.yahoo.com/group/ORE_bits/
Yahoo! group for ORE.

Parallax Discussion Forums
www.parallax.com/html_pages/tech/forums/main.asp
Web-based forums centered around the popular Parallax microcontroller and robotics products.

Phidgets Forums
www.phidgetsusa.com/forums/index.php
Forums supporting Phidgets products and robotics parts.

Phoenix Area Robot Experimenters
http://tech.groups.yahoo.com/group/parex/
Yahoo! group for the Phoenix Area Robot Experimenters.

Pololu Forums
www.pololu.com/forum
Web-based forums centered around using Pololu products in robotics applications.

Portland Area Robotics Society
http://tech.groups.yahoo.com/group/PARTS/
Yahoo! group for PARTS.

RobotBuilders.Net
www.robotbuilders.net
Umbrella website for various specialty Internet-based robot building clubs; several with forums and mailing lists.

- B-9 Club
- Robot Club
- R2-D2 Builders Club
- The Drone Room (Silent Running)

Robotics Society of Southern California San Diego Robotics Society
http://tech.groups.yahoo.com/group/SDRS-List/
Yahoo! group for San Diego Robotics Society and Robotics Society of Southern California.

San Francisco Robotics Society of America
http://tech.groups.yahoo.com/group/sfrsa/
San Francisco, CA — Yahoo! group fgor The San Francisco Robotics Society of America.

Seattle Robotics Society
tech.groups.yahoo.com/group/

SeattleRobotics

Yahoo! group for Seattle Robotics Society.

Sparkfun Forums
forum.sparkfun.com

Forums supporting Sparkfun products, which include ones that are useful in robotics.

The Robot Group
www.robotgroup.org

Provides a mailing list and archive.

The Robotics Club of Yahoo! (TRCY)
groups.yahoo.com/group/theroboticsclub

Yahoo! group for TRCY.

Triangle Amateur Robotics
triangleamateurrobotics.org

Provides a web-based forum.

Twin Cities Robotics Group
http://tech.groups.yahoo.com/group/tcrobots/

St. Paul, MN

Yahoo! group for the Twin Cities Robotics Group.

Vancouver Island Robotics
www.vancouverislandrobotics.org

Provides a mailing list.

Vex Forums
www.vexforum.com/forum.php

User-to-user forums for the Vex line of robotics kits.

Yahoo Groups
tech.dir.groups.yahoo.com/dir/Science/Engineering/Mechanical

Master list of mechanical engineering groups at Yahoo!. See the Robotics specialized group. **SV**

ROBO-LINKS

Robotics Kits and Components... the World's Most Complete Offering!
WWW.ROBOTSTORE.COM

Give your Robot the power it deserves...
LORAX WORKS
proSeed microcontroller
www.loraxworks.com

ALL ELECTRONICS CORPORATION
Electronic Parts & Supplies Since 1967

CONTROLLERS
NEW ARM2103 Controller
ONLY $24.90
Includes LPC2103 Microcontroller, Direct In-Circuit Program Download and Update, Easy to connect Port Connections, Delivered Ready to Run, Ideal for Robots and Control Systems.
Also Available: ARM7024 Controller, ATMEGA8535 Controller, and PIC16F877 Controller
www.futurlec.com

Technological Arts
9S12 on a DIP!
from $30
freescale
Toll-free: 1-877-963-8996
www.technologicalarts.com

$29.95 (MSRP) MaxSonar-EZ1
High Performance Ultrasonic Range Finder
- serial, analog voltage & pulse width outputs
- lowest power - 2mA
- narrow beam
- very easy to use!
www.maxbotix.com

Pololu Robotics & Electronics
6000 S. Eastern Ave. 12D, Las Vegas, NV 89119
www.POLOLU.com 1-877-7-POLOLU
Motor & servo control
Robot controllers
Gearboxes & wheels
Robot kits
Custom laser cutting
Solder paste stencils

C Stamp™ Controller
• LOW COST - High Pin Count
• 48-Pin Module with Analog
• High Speed, Large Memory
• Professional SW Dev. Tools
• Easy to Connect Accessories
• Free Expert Technical Support
Toll-Free: 1 (800) 985-AWIT
www.c-stamp.com

IntelliBrain™-Bot
- Java™ programmable
- Example programs
- Tutorials
- IntelliBrain controller available separately
www.ridgesoft.com

RobotShop
www.robotshop.ca
Domestic Robots and Home Automation
Robot Toys - Robot Kits - Robot Parts

We have your parts! Kits too!
MAXIMUM ROBOTICS
maximumrobotics.com

Beta LAYOUT
TOLL FREE 1877 3908541
PCB-POOL
WWW.PCBPOOL.COM
2 Days: $ 90 8 Days: $ 22.5
Price Example: 16 Sq-Inches, (ds - pth)

The VR Stamp™ Module
Add AI Features like:
* Voice Recognition
* Sound Effects
* LipSyncing
* Voice Passwords
SENSORY®
The World Leader in Embedded Speech Technology
www.sensoryinc.com sales@sensoryinc.com

For the finest in robots, parts, and services, go to www.servomagazine.com and click on Robo-Links.

Mind Candy For Today's Roboticist

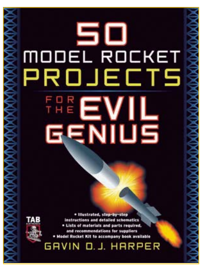
The *SERVO* Store

SERVO CD-ROM

Are you ready for some good news? Along with the first 26 issues of *SERVO Magazine*, all issues from the 2006 calendar year are now available, as well. These CDs include all of Volume 1, issues 11-12, Volume 2, issues 1-12, Volume 3, issues 1-12, and Volume 4, issues 1-12 for a total of 38 issues all together. These CD-ROMs are PC and Mac compatible. They require Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the discs. **$24.95 – Buy 2 or more at $19.95 each!**

50 Model Rocket Projects for the Evil Genius

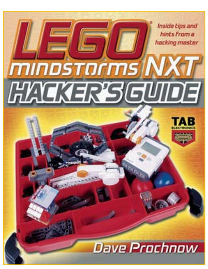
by Gavin D. J. Harper



Yes, as a matter of fact, it IS rocket science! And because this book is written for the popular Evil Genius format, it means you can learn about this fascinating and growing hobby while having run creating 50 great projects. You will find a detailed list of materials, sources for parts, schematics, and lots of clear, well-illustrated instructions. **$24.95**

LEGO MINDSTORMS NXT Hacker's Guide

by Dave Prochnow



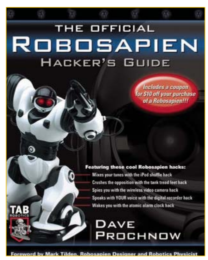
Here is an awesome next-generation collection of LEGO MINDSTORMS projects that enables you to build and program a real working robot in just 30 minutes! New technologies and expanded sensor capabilities make it easier than ever to add a level of sophistication to robotic and architectural creations. This cutting-edge guide describes new advances that make LEGO MINDSTORMS NXT such a great robotics resource. The book explains the all-new NXT intelligent brick ... the interactive servo motors with rotation sensors that align speed for precise control ... the ultrasonic sensor that allows robots to "see" by responding to movement ... the improved light and touch sensors that let robots detect color and feel ... and much more. **$24.95**

SERVO CD-Rom

Are you ready for some good news? Along with the first 14 issues of *SERVO Magazine*, all issues from the 2005 calendar year are now available, as well. These CDs include all of Volume 1, issues 11-12, Volume 2, issues 1-12, and Volume 3, issues 1-12, for a total of 26 issues all together. These CD-ROMs are PC and Mac compatible. They require Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the discs. **$24.95 – Buy 2 or more at $19.95 each!**

The Official Robosapien Hacker's Guide

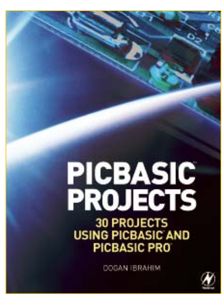
by Dave Prochnow



The Robosapien robot was one of the most popular hobbyist gifts of the 2004 holiday season, selling approximately 1.5 million units at major retail outlets. The brief manual accompanying the robot covered only basic movements and maneuvers — the robot's real power and potential remain undiscovered by most owners — until now! This timely book covers all the possible design additions, programming possibilities, and "hacks" not found anyplace else. **$24.95**

PIC Basic Projects

by Dogan Ibrahim



Covering the PIC BASIC and PIC BASIC PRO compilers, *PIC Basic Projects* provides an easy-to-use toolkit for developing applications with PIC BASIC. Numerous simple projects give clear and concrete examples of how PIC BASIC can be used to develop electronics applications, while larger and more advanced projects describe program operation in detail and give useful insights into developing more involved microcontroller applications. **$29.95**

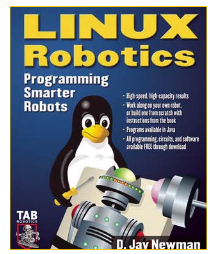
We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

SERVO CD-Rom

Are you ready for some good news? Starting with the first *SERVO Magazine* issue — November 2003 — all of the issues through the 2004 calendar year are now available on a CD that can be searched, printed, and easily stored. This CD includes all of Volume 1, issues 11-12 and Volume 2, issues 1-12, for a total of 14 issues. The CD-Rom is PC and Mac compatible. It requires Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the disc. **$29.95**

Linux Robotics

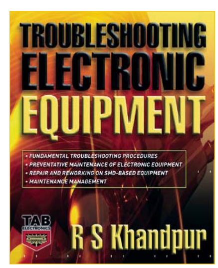
by D. Jay Newman



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**

Troubleshooting Electronic Equipment

by R. S. Khandpur

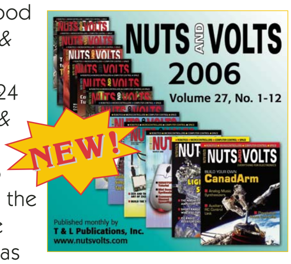


From cell phones to medical instruments to digital and microprocessor based equipment, this hands-on, heavily illustrated guide clearly explains how to troubleshoot, maintain, and repair all types of electrical equipment. The author covers all the essentials such as necessary tools, soldering techniques, testing, fundamental procedures, and mechanical and electrical components. **$49.95**

To order call 1-800-783-4624 or go to our website at www.servomagazine.com

Mind Candy For Today's Roboticist

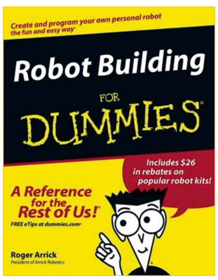
Nuts & Volts CD-ROM



Here's some good news for *Nuts & Volts* readers! Along with all 24 issues of *Nuts & Volts* from the 2004 and 2005 calendar years, the 2006 issues are now available, as well. These CDs include all of Volumes 25, 26, and 27, issues 1-12, for a total of 36 issues (12 on each CD). These CD-ROMs are PC and Mac compatible. They require Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the discs. **$24.95 – Buy 2 or more at $19.95 each!**

Robot Building for Dummies

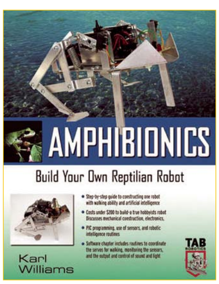
by Roger Arrick / Nancy Stevenson



Ready to enter the robot world? This book is your passport! It walks you through building your very own little metal assistant from a kit, dressing it up, giving it a brain, programming it to do things, even making it talk. Along the way, you'll gather some tidbits about robot history, enthusiasts' groups, and more. Do it the Dummies' way — explanations in plain English, "get in, get out" information, icons and other navigational aids, tear-out cheat sheet, top 10 lists, and more. **$21.00**

Amphibionics

by Karl Williams

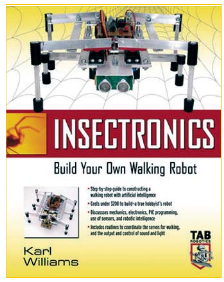


If you're a robotics hobbyist with a flair for creativity, here's your opportunity to join the revolution and advance robotic evolution. *Amphibionics: Build Your Own Biologically Inspired Reptilian Robot* leads you step-by-step through four amazing projects which, when completed, will leave you as master of a gang of robot reptiles that jump, slither, walk, swim, and respond to their environment! **$19.95**

Check out our online bookstore at www.servomagazine.com for a complete listing of all the books that are available.

Insectronics

by Karl Williams



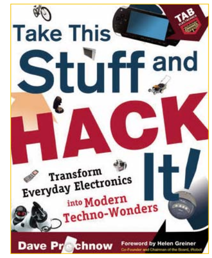
This complete project book delivers all the step-by-step plans you need to construct your own six-legged insect-like robot that walks and actually responds to its environment. Using inexpensive off-the-shelf parts, hobbyists can "build a better bug" and at the same time have loads of fun honing their knowledge of mechanical construction, programming, microcontroller use, and artificial intelligence. **$19.95**

10 Cool LEGO Mindstorm Dark Side Robots, Transports, and Creatures

The Dark Side Developer's Kit is targeted towards the young or novice LEGO MINDSTORMS designer, age 9 and up, although experienced MINDSTORMS fans will appreciate the unique possibilities offered by this kit, as well. *The Dark Side Developer's Kit* includes special MINDSTORMS pieces that allow the user to create a host of Star Wars themed robots, creatures, and vehicles. It also comes with the Micro Scout, a mini-computer with seven built-in programs, a motor, and a light sensor that brings the MINDSTORMS creations to life with a minimum of effort. **$24.95**

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**

10 Cool LEGO Mindstorm Robotics Invention System 2 Projects

The LEGO MINDSTORMS RIS 2.0 is the core set for all MINDSTORMS users, that lets users design and program working robots — limited only by their imagination. Initially designed for users 12 and up, LEGO MINDSTORMS has taken off with LEGO enthusiasts of all ages. *10 Cool LEGO MINDSTORMS RIS 2.0 Projects: Amazing Projects You Can Build in Under an Hour* provides step-by-step instructions and detailed illustrations for users of all skill levels and proficiencies. **$24.95**

$20 each
15% off if you buy two or more.

From HomoSapien to RoboSapien

Before R2D2 there was R1D1

SERVO Magazine T-Shirts

Get your very own limited-edition *SERVO Magazine* T-shirt. Shirts come in sizes S, M, L, and XL, and are available in either black or white. All shirts are 100% pre shrunk cotton.

FIRST

by Lester "Ringo" Davis

It's that time of year again. Flowers will be blooming, birds will be singing, robots will be rolling, and, of course, it's time for FIRST again. If you read this magazine at all, then you've heard of FIRST. There is never a shortage of kids wanting to compete, but what is usually needed is more adult volunteers. FIRST is always in need of inspectors, judges, mentors, etc.

I've been a safety inspector and a mentor for FIRST and VEX and it is a great experience. You don't have to be a robotics expert to contribute, either. As a safety inspector, I checked out various teams' robots before they could compete, inspecting things like height, weight, construction methods, etc. It is as simple as going down a checklist and looking for things like sharp edges that could hurt somebody, wiring that is done incorrectly, or the use of items not allowed like duct tape.

This is something that anyone can do. It's great fun to talk to all the kids that have built these things and to watch them compete. Before I go to an event, I try to imagine all the different ways I could construct a robot to accomplish the goal. I'm always amazed at the variations and the stuff I never thought of.

One of the first competitions I went to involved stacking large plastic boxes. I kept seeing robot after robot that would pick up a box and put it on top of the stack, then pick up another box and put it on top of the taller stack, so the height of the stack was limited by the reach of the robot. This made perfect sense to me until I saw a short little robot come out.

At first, I thought they just gave up on the stacking portion of the contest until I saw their strategy. Instead of picking up a box and putting it on a stack, they picked up the stack and put it on a box! So, their robot was much smaller, but it just had to be strong enough to pick up several at once. And since they only lifted one box high, it was faster than most of the others. I don't know if I would have thought of that, but I will never forget it.

If you don't want to be an inspector or a judge, then you can be a mentor. I mentored Vex team #8 for two years and everyone really enjoyed it. My team was motivated so it made my part very easy. I hosted most of the meetings and just checked on the kids occasionally to make sure they were on track. They designed, built, and programmed everything themselves. Some teams just need an adult to step up and just "be the adult" in the room so that the kids can have permission from the school to work on the project.

You don't have to do it full time, either. A team can have as many mentors as it needs, so if you can help out once every other week, that would still contribute to the team. It means a lot to the kids when adults show an interest in what they are doing.

One of the biggest things that amazed me about going to a competition is that this is the first time I've seen cheerleaders for nerds (a.k.a., young engineers). When a team is announced, there is often a roar from the stands as dozens of students in face paint or even costumes cheer on their school team. You would think you were at an NBA playoff game sometimes, when four popular teams are competing at the same time. I've seen banners, signs, even a 30-foot dragon at one event. There are TV cameras, reporters, photographers, people running around on Segways, and, of course, kids with enough energy to run a city.

The attitude of the kids competing is amazing, also. Everybody wants to win, of course, but if a robot breaks down and needs a part, there will often be several teams who immediately offer a replacement. My team broke a gear in the semifinals and had two minutes until the next round. Our gear box was in an adjacent building and there was no way to get there in time. A competitor of ours saw what happened and took a gear off of their robot and gave it to us so we could go on to the next round.

That is not something you would see in a sporting event. If you have never been to one of these events, you should attend one just as a spectator to see how excited everybody gets. Then you can decide on the best way to help out in the next competition.

I've often seen articles talking about how the numbers of engineers are declining in the US. The best way to prevent this is to get young people interested in engineering, and what better way to do that than to show them an event where they can learn, have fun, get recognition, and get to build cool robots. **SV**

Then and NOW

THE FUTURE OF ROBOTICS

by Tom Carroll

When people speak of the 'future' of anything, they generally go overboard. They take their fondest wishes and then extrapolate 'way out there.' I remember reading as a kid about the flying cars that we'd all have, and moon and Mars bases at the turn of the next century. Even today, we're prone to exaggerate. The future of robotics is no exception and has made big headlines in today's media. Mark David, Editor of *Electronic Design*, commented in the February 1st, 2007 issue " ... At CES (The large Las Vegas Consumer Electronics Show), robots steal the show." He had seen Honda's Asimo walking about and iRobot's new iRobot Create open platform for experimenters and was impressed with the growth in robotics.

"A Robot in Every Home" proclaimed Bill Gates in his January 2007 article in *Scientific American*. The subtitle went on to say "The leader of the PC revolution predicts that the next hot field will be robotics." Comments about his article showed up in a blog posted in the *Huffington Post* on January 4, 2007, "Robots Will Be A Nearly Ubiquitous Part Of Our Day-To-Day Lives In Near Future, by Bill Gates.

Other people jumped on his comments with other blogs in the *Huffington Post* that stated "Bill Gates is a fool with no vision for the future. Robots will not be involved in the future at all." (What a short-sighted comment!) As a very high-profile entrepreneur, Gates has taken far more undeserved flack than praise but I'm sure that he feels that comments like that 'are all part of the game.' I personally admire the guy, his robotics team, and his company, but I'll get back to Bill Gates later.

For years I have often been asked, "just what is the future of robotics?" You also may look at this title and wonder what is the difference between the future as we saw it 'then' and the future as we see it 'now.' Really, there is a lot. I have read different experts opinions on the subject and some were right on target while others were way out in left field — if 'on earth' at all. Before mentioning others who may have made predictions of the future that did not quite pan out, it's only fair that I mention one that I made back in early November, 1984 in a paper I delivered at the Robots West conference in Anaheim, CA.

"Close your eyes and let's dream a bit," I said. "It's January 1st, the year 2001, about 9:00 AM. You've recovered from last year's New Year's party and are lying in bed with your significant other, half awake, when you detect that wonderful aroma of fresh coffee, bacon, and eggs. Soft music is wafting through the air when you're suddenly startled awake by a synthesized voice:"

Good morning. Here is your breakfast, as programmed — coffee, orange juice, scrambled eggs, and toast.

"You look on the tray and see crumbs, but no toast. 'Where's the toast?'

Oh, the toast? It was on the tray when I left the auto galley.

Sure enough, you look on the floor over near the doorway and you see your toast lying there with a robot's tire track across it. You mutter to yourself, 'a week out of the crate and my new robot already has problems. Oh well, I'll call Universal Robots tomorrow and find out why Hal shakes so much.'

Hal dispenses the daily news-fax sheet and departs ...

I used this opening to loosen the attendees up a bit from the previous session that was probably about as interesting as a typical industrial robot stacking hot forgings on a pallet in a factory. It wasn't just about humor, though; it was to enlighten them about the possibilities of a personal robot in their home. I continued on with my talk, mentioning that the above scenario 'sounds a lot like science fiction.'

Yes, there were some pretty neat personal robots in those days, but that probing question remained to be asked, and is still asked of me today about personal robots: "but ... what can it do?" To say that I was a bit off in my prediction of what would be available in 2001 is an understatement, but conference planners love these predictions, accurate or not. In predicting the future, we take what knowledge we know at the moment and extrapolate into what we feel will be the future for the particular subject.

Yes, I took a bit from Arthur C. Clark's story, *2001, A Space Odyssey*,

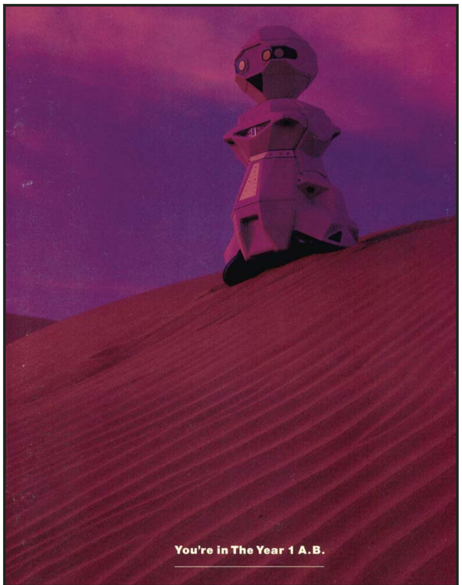


Figure 1. Androbot Topo.

using the date and the computer's name — Hal. I missed the mark, though, as it is now over 22 years after my paper and we still don't have a mass-produced personal home robot that can act as a true home servant. Yes, it's possible with today's technology, as I did a lot of research on a personal assistant robot to allow the elderly to enjoy independent living, but it's still not a reality.

Roombas and Scoobas sweep our floors, Aibos entertain us, and security robots patrol our factories, but only home experimenter's and university built robots truly serve a few of us in our homes.

One of the best predictions that I recall is about what it would take to make a real robot. I don't recall the time, place, or person who answered the question put to them, but it went something like this: "If I could place 640K of memory in my robot, find a microprocessor that could operate at 10 MHz or higher, and could afford a hard drive that had five megabytes of capacity — wow, could I build a fantastic robot!" This was back in the late '70s or early '80s, but the statement has always stuck with me. It happens to many a grant recipient when they are asked by the organization that funded them "When will we see some results?"

"Oh, just give me some more time. The solution is just around the corner; just over the horizon."

Nowadays, computers are available that cost around a thousand dollars and have a gigabyte (1,000 megabytes) or more of memory, a dual-core processor that operates at 3 GHz (3,000 MHz), and a gigabyte in hard drives. I still don't see super intelligent robots roaming about our streets or in our homes. Hmmm. Maybe it was that person's way of saying: "I know you people think that I am some sort of great robotics engineer. The reason I haven't built a great robot is because we just don't have the technology available. It's not my fault. That's my excuse and I'm sticking with it."

The Future of Robotics — As Envisioned in the '80s

Let me back up a little bit in the 'future' of personal robotics to the 1980s, probably the pivoting point in the new age of robots. I have a lot of literature in my files dating back to this period and some of the claims make me smile a bit. *The Journal of the Robotics Society of America* was called 'Robotics Tomorrow' implying that robotics is the future. Androbot, RB Robot, and Heath all made big splashes to the public at about this time. All were saying that their product was to 'usher in the age of robotics.'

Silver-tongued Nolan Bushnell of Pong and Atari fame formed Androbot, maker of Topo and Bob. Bushnell's company distributed some of the most polished literature and glossy posters touting "Topo as the world's first personal robot." "You're in the Year 1 A.B." touted his well-designed handout brochure at many robot exhibitions (see Figure 1). "Meet Topo. He's a mobile extension of your personal computer."

Yes, indeed, he was an extension of your computer as the basic Topo model could only be operated by an Apple II computer through an RF link transmitting on a 27 MHz CB frequency. Later models used an IR link and the Topo III actually had dual 8031 microprocessors on board to allow some autonomy.

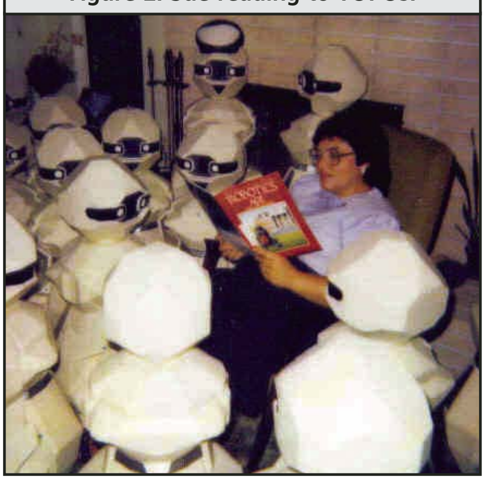
Androbot's top-of-the-line, BOBXA featured a single Intel 8088 processor with 128K of RAM, expandable to 256K. It sold for an astonishing $7,995, but, of course, that included 'four ultrasonic range-finder transducers, a flux-gate compas, and full documentation.' Remember, this is 1980's money and you could buy a new mid-size car for that same amount.

Androbot Wobbles into the Sunset

Androbot sadly folded several years later, much to the chagrin of Bushnell who truly saw the future in this company. It didn't fail for the lack of enthusiasm as Bushnell and his team probably spent more money on promotion and well-written flashy literature than they did on manufacturing the robots. He has said many times later that his greatest product line and most disappointing loss was Androbot.

Our Southern California Robotics Society was fortunate to be able to buy a truckload of Topos with a crate of manuals, batteries, and literature for about a hundred dollars each when Androbot was folding. My wife, Sue is shown in Figure 2 reading a robotics magazine to 17 Topos in our livingroom. My armless Topos have earned their keep as movie action props, Halloween door greeters, and experimental bases, but avid listeners to a robot story, they are not.

Figure 2. Sue reading to TOPOs.



RB5X — The *Intelligent* Robot

Figure 3 tells it all about the true feelings of Joe Bosworth, the founder of RB Robot Company. "The Future of Robots is Now! (September, 1982)" RB Robot did not hide the fact that their robot was more attuned to the educational market rather than the cute and wobbly Topo that could only roam about a home without any arms to do anything useful. Yes, you can see RB5X holding a newspaper with its fragile arm, but its main purpose was to teach robotics. Besides, the initial model did not have that arm that neatly tucked into a cavity in the robot's body.

RB Robot's brochure went on to say "Today, we are living the future that was only a vague image 20 years ago and our vision of robots is becoming more focused." Its Motorola 68331 processor and one-pound capacity articulated arm gave it some amazing capabilities for a robot in those days. A version of the RB5X is still being produced to this day (go to **megadroid.com**).

Heathkit Launches the Hero Series

Heath Company, long noted for their electronic kits, brought forth a series of robot kits (and already assembled robots) that were the answer to prayers of many a robot experimenter. These machines were even more focused on the educational aspect of robotics with a heavy emphasis on industrial robotics. (See my February 'Then and Now' about educational robots.)

If the first Hero 1 was a robot experimenter's delight, then the Hero 2000 was their absolute dream with its powerful articulated arm. Does the lightning above the Hero in Figure 4 indicate power or does the name 2000 (in the 1980s) indicate the future? Were the Heroes and these other robots of the past really the robots of the future? Certainly!

They set the framework for personal robots to come. They might have been a bit simplistic in their sensor suites, processing power, and a bit larger than today's machines, but the robots of the '80s were — as Joe Bosworth stated — quite a bit more than the 'vague images' that experimenters dreamed about in the '60s.

The Future is Now!

We're now well into the next century and millineum and tens of thousands of us call robotics their main hobby. As I mentioned in my February column on robotics education, we've taken a distinct turn in this new field. Most US industrial robot companies have faded into the sunset or have been purchased by others, but personal robotics has made leaps and bounds in progress. Industrial robotics groups once shunned experimenters, but these manufacturing-oriented people are now seeing the amazing results in personal robot development.

Bipedal robots walk up stairs, four-legged battlefield robots scramble across stone-covered slopes, airborne and undersea vehicles operate for days without a person at the controls, and autonomous vehicles navigate across hundreds of miles of desert or even city streets in these times. Yet, we'll look back at the decade of the 2000's and see how quaint our present technology was, but that has always been the case since the dawn of civilization. Progress marches onward.

Bill Gates and the Dawn of the Age of Robots

Getting back to Bill Gates, some may wonder just what this guy does know about robotics. He's a software entrepreneur, not a robotics expert. Many of us know that Bill tapped Tandy Trower as General Manager of Microsoft's Robotics Group to develop a scalable architecture to span a wide variety of robotics hardware using eight to 32 bit systems and various sensor suites. This Microsoft Robotics Studio includes a set of useful tools with a visual simulation environment, plus technology library services to help robot developers. Does this step make Microsoft a big player in this field?

I feel that this development by Microsoft is a great start and it tells me that Gates' organization is listening to the needs of personal robot experimenters. Virtually all of this group's efforts are concentrated on mobile and personal robotics, not the industrial varieties. Never underestimate the power of Microsoft. If they can be convinced that there is a significant market for robot software, Microsoft will make it happen.

It's interesting to note that man's first visions of a robot were anthropomorphic creations, and these human-like, bipedal forms followed fictional accounts and movies well into the 20th century. Then came Star Wars and R2D2, and

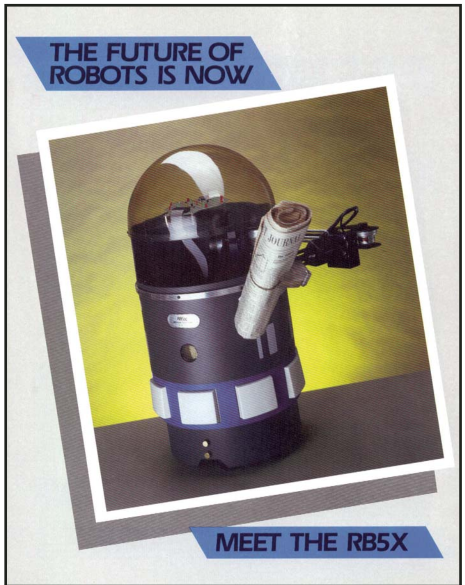
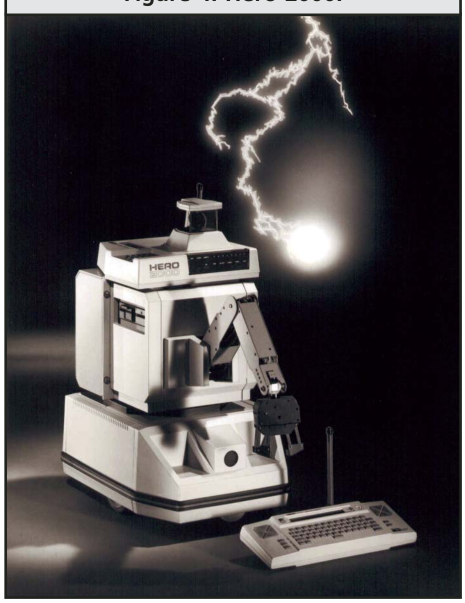


Figure 3. RB5X.

Figure 4. Hero 2000.


READER FEEDBACK

I would like to encourage many of you readers to email me your opinions of the 'future of robotics.' I am planning a large article to cover just what all of us feel is the true future of our avocation of robotics. In these short columns that I write, I wade through a lot of literature to find just a few nuggets of appropriate information and sometimes sidestep the real jewels in the rough. Please let me know if I've missed something of importance or have even got it all wrong. Also, I'd appreciate information on the old, classic robots that I have yet to cover in this column. John Boisvert, who goes by the name of Hotwing, has a site called Megadroid.com. He also has a company, RB Robotics, where he continues to sell the classic RB5X that has been around since the 80's. I've used some of his photos and I'd appreciate photos from others about these old machines.

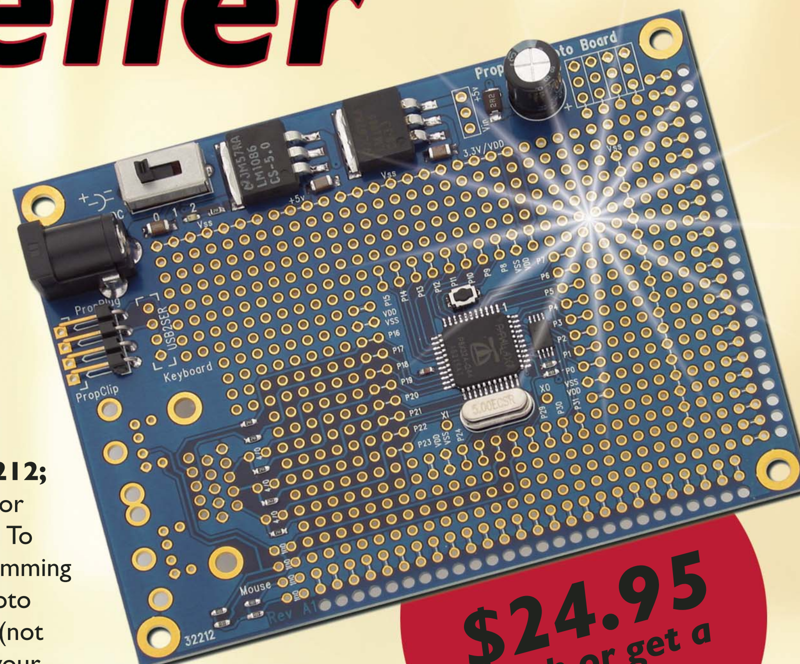
robots could resemble rolling, beeping trashcans or anything you could imagine. Robots now rolled on wheels.

We're starting a new millineum and, guess what, there are more and more bipedal, anthropomorphic robots gracing the pages of this magazine and many others. Honda's Asimo walks about entertaining us as experimenter-produced machines walk their way out of our workshops. Is this the new future of robots? That remains to be seen. **SV**

Advertiser Index


Protect the environment
Make the right choice
SerialLCD™
Easy to use!
Priced from $34.95
Complete info
at BasicX.com
ALAILABLE IN RED, GREEN AND BLUE.
SitePlayer
TELNET
PoE - Power Over Ethernet
Serial to Ethernet converter that allows your serial device to be controlled via ethernet or through the internet.
$129.95 www.siteplayer.com
RoHS 2002/95/EC
COMPLIANT PRODUCTS FROM NETMEDIA.
NetMedia
NetMedia Inc., 10940 N Stallard Pl, Tucson Arizona 85737 tel.520.544.4567 fax.520.544.0800 www.BasicX.com

Propeller™ Proto Board

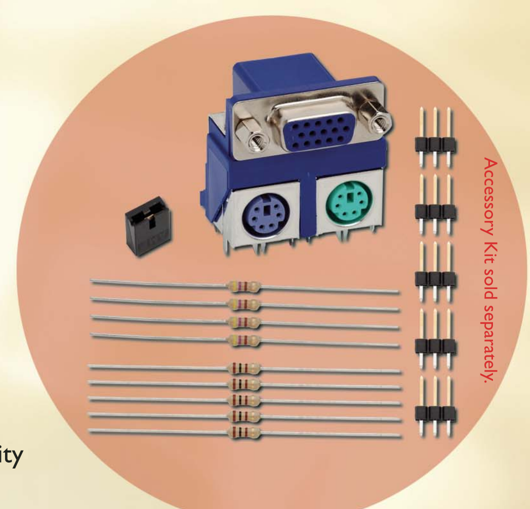


$24.95 each or get a 5-pack for $100.00!

Parallax's **Propeller Proto Board (#32212; $24.95)** is a low cost, high quality solution for permanent projects using the Propeller chip. To save the user some expense, the USB programming interface is not included on the Propeller Proto Board; we suggest you purchase a Prop Plug (not pictured; #32201; $29.95) for programming your board. *This board is ready for use with the Prop Clip (not pictured; #32200; $29.95), and is also compatible with the USB2SER Development Tool (not pictured; #28024; $29.95) with the addition of a 4-pin header.*

Features of the Propeller Proto Board include:

- P8X32A-Q44 Propeller chip
- 64 KB EEPROM for program and data storage
- LM1086 5V and 3.3V regulators provide up to 1.5 Amps with an input power supply of 6-9 VDC
- Accepts the optional VGA, mouse and keyboard interface available in the Accessory Kit (sold separately; #130-32212; $14.95)
- 3-position power switch (off, logic/power, power/logic and servo ports)
- Sockets for 4 servos
- Removable 5MHz crystal
- Access to all 32 I/O pins
- Unplated row of holes along perimeter to provide stress relief to off-board connections
- Same mounting holes as the Board of Education (3x4") for compatibility with the Boe-Bot® robot



Order the **Propeller Proto Board** online (#32212; $24.95) at www.parallax.com or call the Parallax Sales Department toll-free 888-512-1024 (Mon-Fri, 7am-5pm, PT).

PARALLAX

www.**parallax**.com

Propeller, Spin, Parallax, and the Parallax logo are trademarks of Parallax, Inc.