Covering The World Of Personal Robotics

SERVO MAGAZINE

FOR THE ROBOT INNOVATOR

www.servomagazine.com

August 2010

5 Demo Platforms

Built Using LabVIEW Robotics From National Instruments

✦ **The Evolution Of Ethel**
A journey from pile of parts to well-dressed automaton.

✦ **The NXT Big Thing**
Learning about robotics and prototyping with NXT is as easy as it gets.

✦ **Basics Of Using C**
No well-skilled roboteer should be without programming skills in their arsenal of tools.

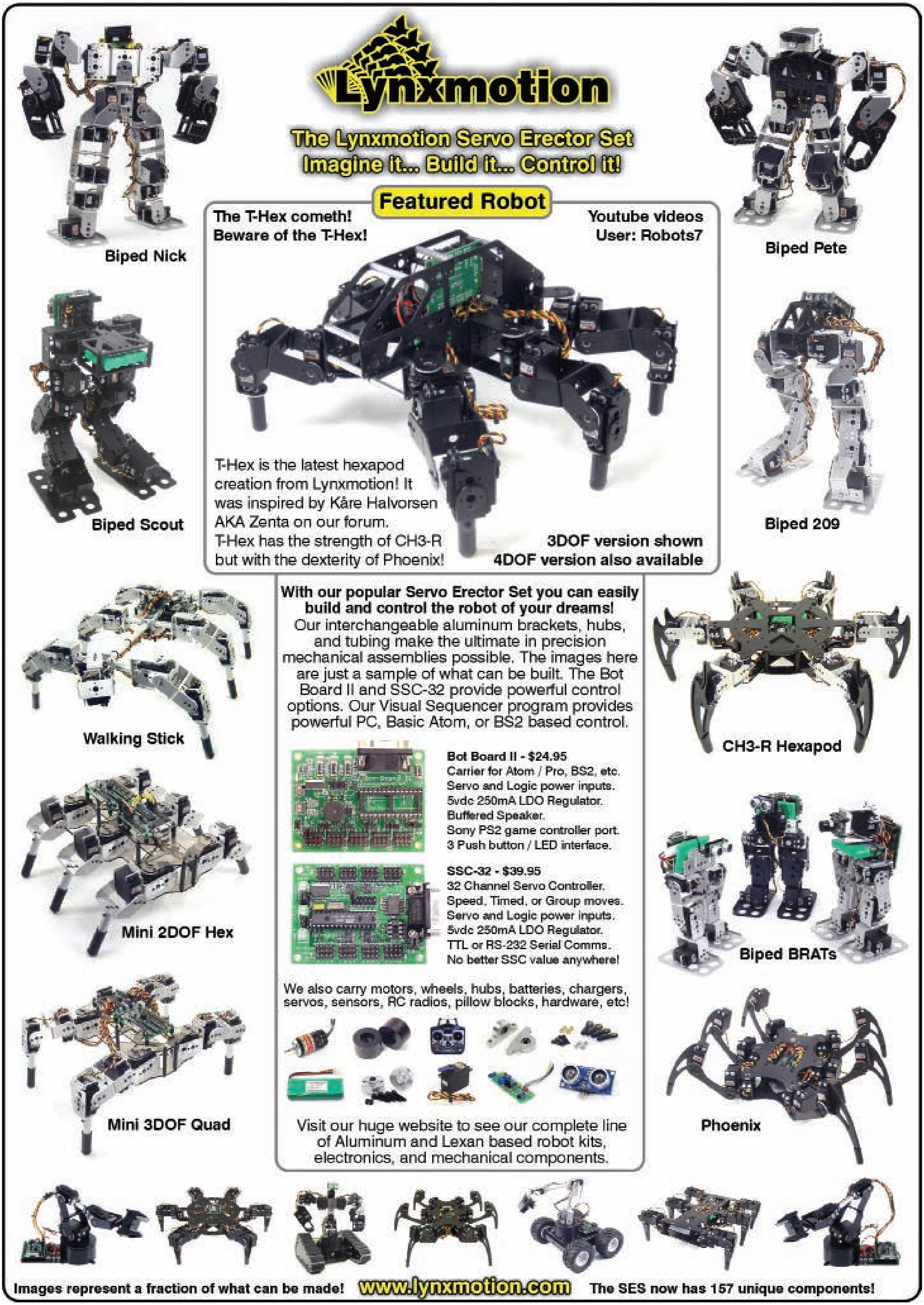
✦ **Combat Zone**
Modifying NPC drive motors for combat.

U.S. $5.50 CANADA $7.00

08>

0 71486 02422 4

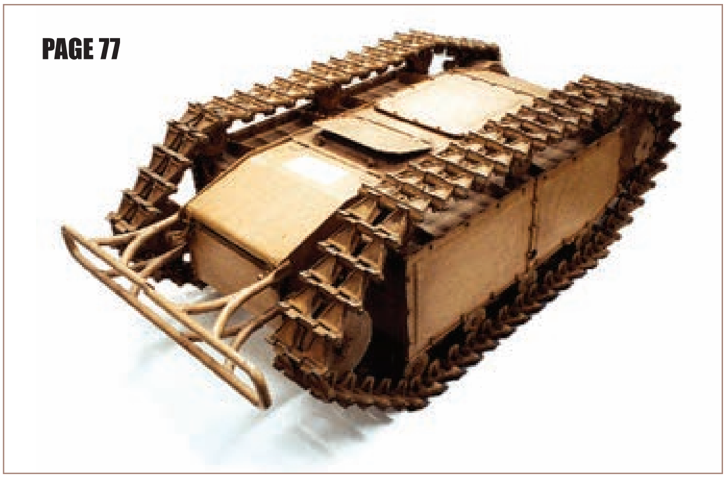

Let your geek shine.
Meet Jen Lewin, creator of The Pool. Commissioned for the Burning Man Project, Jen's creation features a combination of XBees, Arduinos, and LEDs fitted in custom enclosures. When the pads sense weight, they respond with a beautiful display of light and color that mimics rippling water.
Whether you're searching for a new microcontroller, or building a room-sized interactive environment, the tools are out there. Create a work of art, and let your geek shine too.
sparkfun ELECTRONICS
Sharing Ingenuity
WWW.SPARKFUN.COM
©2010 SparkFun Electronics, Inc. All rights reserved. All other trademarks contained herein are the property of their respective owners. Read more about Jen Lewin's Pool at http://tinyurl.com/232m54u or check out Burning Man at www.burningman.com. Have fun out in the desert, Jen!

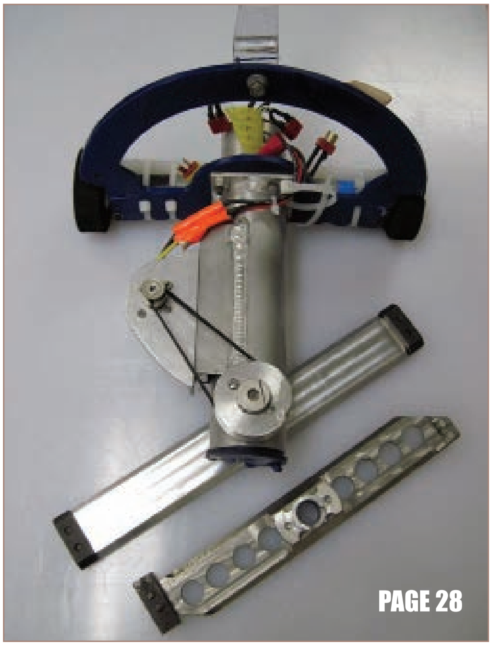

Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
Featured Robot
The T-Hex cometh!
Beware of the T-Hex!
Youtube videos
User: Robots7
T-Hex is the latest hexapod creation from Lynxmotion! It was inspired by Kåre Halvorsen AKA Zenta on our forum.
T-Hex has the strength of CH3-R but with the dexterity of Phoenix!
3DOF version shown
4DOF version also available
Biped Nick
Biped Pete
Biped Scout
Biped 209
Walking Stick
CH3-R Hexapod
Mini 2DOF Hex
Biped BRATs
Mini 3DOF Quad
Phoenix
With our popular Servo Erector Set you can easily build and control the robot of your dreams!
Our interchangeable aluminum brackets, hubs, and tubing make the ultimate in precision mechanical assemblies possible. The images here are just a sample of what can be built. The Bot Board II and SSC-32 provide powerful control options. Our Visual Sequencer program provides powerful PC, Basic Atom, or BS2 based control.
Bot Board II - $24.95
Carrier for Atom / Pro, BS2, etc.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
Buffered Speaker.
Sony PS2 game controller port.
3 Push button / LED interface.
SSC-32 - $39.95
32 Channel Servo Controller.
Speed, Timed, or Group moves.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
TTL or RS-232 Serial Comms.
No better SSC value anywhere!
We also carry motors, wheels, hubs, batteries, chargers, servos, sensors, RC radios, pillow blocks, hardware, etc!
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
The SES now has 157 unique components!

SERVO MAGAZINE

08.2010

VOL. 8 NO. 8


PAGE 77


PAGE 28

Columns

Departments

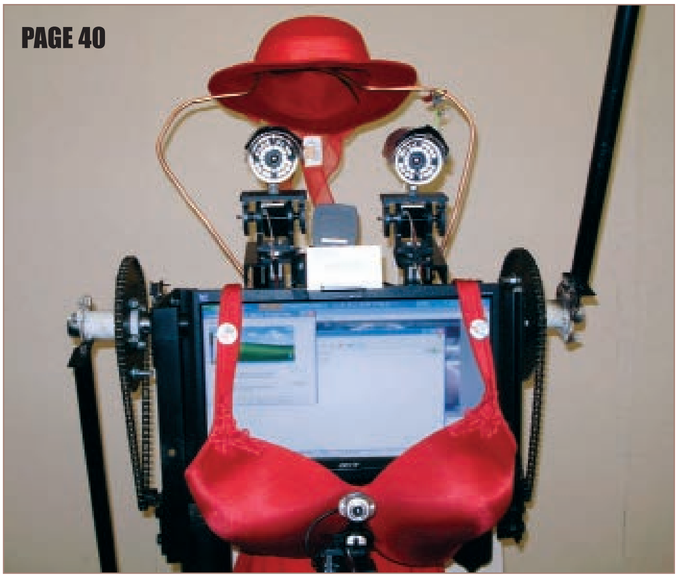
The Combat Zone...

Features

Events

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

In This Issue ...


PAGE 40

PAGE 56

Mind / Iron

by Bryan Bergeron, Editor

Dull, Dirty, Dangerous, and Often the Only Option

The recent oil spill in the Gulf of Mexico put the spotlight on the use of robots in deep-sea operations. It was immediately evident from the video posted on the web that remotely operated vehicles (ROVs) and other robots were at the center of attempts to contain the spill. As with most commercial robotics applications, the task at hand certainly fits the criteria of dull, dirty, or dangerous. Moreover, in this working environment, robots are the only option. Miles below the surface of the gulf, humans are unable to interact directly with the containment equipment.

As with the US DoD's Predator pilots, the operators of the ROVs sit comfortably in an air-conditioned command room. And like the large drones, the ROVs aren't something that can fit in your living room – think large truck. Fly-by-wire interfaces enable operators to operate the hydraulic manipulators and aim cameras at various areas of the leak.

Daily footage of the spill area also underscores the current status of robotics. Modern ROVs might be sophisticated, powerful machines, capable of operating at great depths, but even with a human calling the shots, they're not invincible. And just because ROVs allow an operator to be virtually present doesn't mean that a quick fix is possible. If a skilled human operator can't use an ROV to stop the leak, imagine how an AI would respond to the situation. Clearly, we're not to the point commonly depicted in Sci-Fi movies where robots autonomously handle the routine chores around the house or space ship, repair holes caused by meteor damage, and other disasters.

Remotely operated robots — while not as sexy as the self-aware androids of the movies — are where the action is in practical robotics. And, not surprisingly, the capabilities of remotely operated robots, whether in the battlefield or operating room, are a direct function of the skills of the operator. For example, if you find yourself a candidate for robotic surgery, better check the record of the surgeon. Just because he or she uses a robot instead of directly manipulating a scalpel doesn't guarantee you a better outcome. As studies have shown, results of robotic surgery — in terms of recovery time and number of complications — can be significantly greater than with traditional surgery if there's a klutz at the controls. It turns out that there is a practical benefit from spending all those hours playing video games.

Onto a related subject, if you're a regular reader of my editorials, you know that I keep tabs on the bleeding edge of robotics R&D by following the quarterly DoD SBIR grant program (**www.dodsbir.net**). The most intriguing title this round is 'Hands-Free and Heads-Up Control of Unmanned Ground Vehicles," put forward by the US Army. It's a little too close to Avatar for my taste. The object of the funding is to develop a system for controlling an unmanned ground vehicle without requiring the use of the operator's hands, while allowing the operator to maintain situational awareness of his surroundings and that of the robot. The DoD solicitation lists brain-computer-interface (BCI), eye tracking, sub-vocalization, and speech as allowable user interfaces. Check out this and other opportunities on the DoD website. SV

FOR THE ROBOT INNOVATOR SERVO MAGAZINE

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

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

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

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

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Tom Carroll, Dennis Clark, Fred Eady, Steven Nelson, Daniel Ramirez, Ray Billings, Nick Martin, Jenn Eckert, David Geer, R. Steven Rainwater, Kevin Berry, Greg Intermaggio, Pete Smith, Sean Canfield

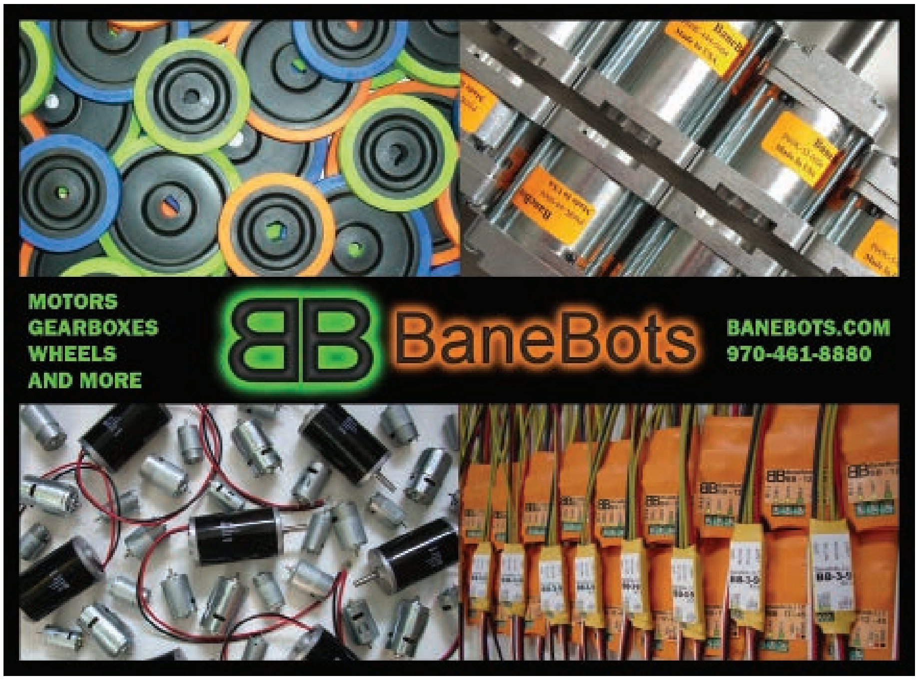
CIRCULATION DIRECTOR
Tracy Kerley
subscribe@servomagazine.com

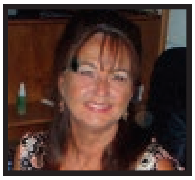
MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Shannon Christensen

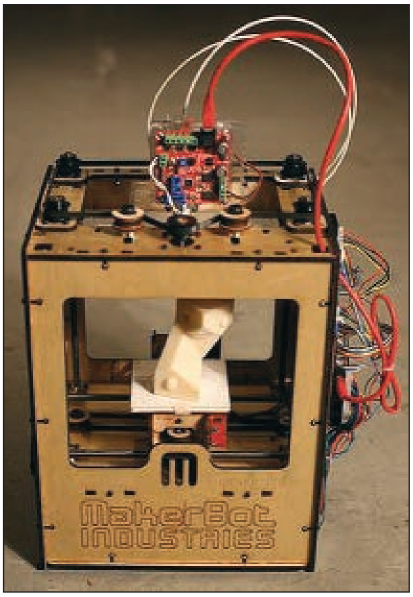




Robytes

by Jeff and Jenn Eckert

Build a Bot that Makes Things



MakerBot Industries' CupCake CNC machine.

In case you haven't heard about it, there is a process out there known as "3D printing" that is commonly used for prototyping and some other applications. The term is a bit misleading, as the objective is not to put ink or toner on paper. Rather, it is an additive manufacturing process in which successive layers of material are laid on top of each other to produce a solid, 3D object. With 3D printing, you can design a part in your CAD program, send the data to the printer, and watch it generate the part. There are quite a few vendors selling the equipment, but you're generally looking at $20,000 or more, so owning your own can be cost prohibitive, especially given that you can have a printing service (e.g., **shapeways.com**) do it for you for a few bucks. However, Makerbot Industries has broken the $1,000 price barrier with its CupCake CNC rapid prototyping machine.

The main drawbacks are (1) You have to assemble it yourself (average time 11 hours); (2) It can't make anything larger than about 4 x 4 x 6 inches; and (3) It only makes plastic parts. Within those boundaries, you can create pretty much anything you want.

The basic unit sells for $750 and the deluxe for $950. The latter includes a USB2TTL cable and other wiring, a power supply, assembly tools, and other extras. The ABS plastic can cost you as little as $10 per pound, so if you have the urge to operate your own desktop robotic manufacturing facility, check out **www.makerbot.com**.

Exploring Cybernetic Leadership

We generally think of bots as subservient entities that exist only to provide services or entertainment to their owners. Maurizio Porfiri, of the Polytechnic Institute of New York University (**www.poly.edu**), however, is designing some mechanisms that are destined for leadership positions. Specifically, he is working on robotic fish that he

Dr. Porfiri's ichthyological Pied Piper.

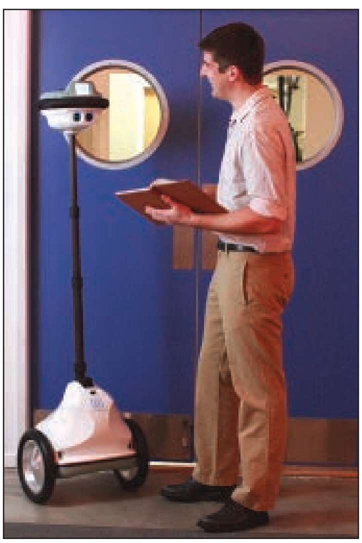
hopes will be able to attract schools of the real thing and lead them around. You might wonder why anyone would want to do that, but Dr. Porfiri's paper on the subject won the Best Robotics paper award at a 2009 American Society of Mechanical Engineers Conference, so there must be something to it. Apparently, leaders of fish schools tend to "beat their tails faster, mill about and accelerate to gain attention, gather a school, and lead it," so they somewhat resemble politicians and Hollywood celebrities. The objective is to use such fish — which would power themselves by tapping into the energy in eddies and underwater vibrations — to lead entire schools away from things like power plant turbines and maybe even oil spills. Eventually, the concept might be adapted to lead flocks of birds to better migratory destinations, lemmings back to land, or even shoppers back into K-Mart.

Travel Back in Evolutionary Time

It's been pretty well documented in these pages that mixing robotics and art is a precarious undertaking, and that has

SiliFulin, the robotic tail for evolutionary reactionaries.

Anybots QB, a "telepresence robot" for homebodies.



been reaffirmed by Japanese "device artist" Ryota Kuwakubo. Designed for people who for some reason have a longing for the look and feel of having a tail, the SiliFulin unit employs a series of servo motors to wiggle tail segments in reaction to your hip movements. We couldn't find a website for Ryota or a place to buy one (well, we didn't try very hard), but you can see Silifulin in action if you search for it on **youtube.com**. You may be amazed. But probably not.

Anybot Anybody?

You might remember Bob Christopher, former CEO of Ugobe — the company that brought us the Pleo robotic dinosaurs. He left Ugobe in 2008, and the company filed for Chapter 7 bankruptcy the following year, but Bob is back as the CEO of Anybots (**www.anybots.com**). His latest crack at the robot market is the Anybots QB, a "telepresence robot" that is essentially a stand-in for people who want to go to work without actually going to work.

What you get is a mobile device that looks like a cross between a cartoon Martian and a Segway that can wander around the office, attend meetings, and annoy people in various ways. If your boss has one, he can look over your shoulder to make sure you're not sipping from a beer under your desk or visiting the adult sites again.

The QB sports a microphone, speaker, camera, and small video screen, so people can communicate with you remotely and even see your smiling (or otherwise) face. You control it via a web browser, so even though it can move around, you can't. One of the complaints about Pleo was that it was a cute idea, but no one wanted to pay $350 for a toy. The Anybot QB — slated to hit the market soon — is priced at $15,000. Could there be a pattern here?

USB Module Retrofits Old Boards

Okay, it's not sexy or high tech, but it's worth a mention. If you happen to have any legacy robotic (or other) equipment that communicates via a DB9 RS-232 connector, you now can upgrade it with the DB9-USB-RS232 modules from Future Technology Devices International (**www.ftdichip.com**). They feature a standard USB mini-B-type connector in a module that fits the footprint of a standard nine-pin DB9 connector. A built-in converter IC handles the protocols, and the devices come with royalty-free drivers for Linux, Windows, and Mac OS. Both male and female versions are available, starting at about $22 in single units. **SV**

The DB9-USB-RS232 modulues allow conversion from RS-232 to USB.

GEERHEAD

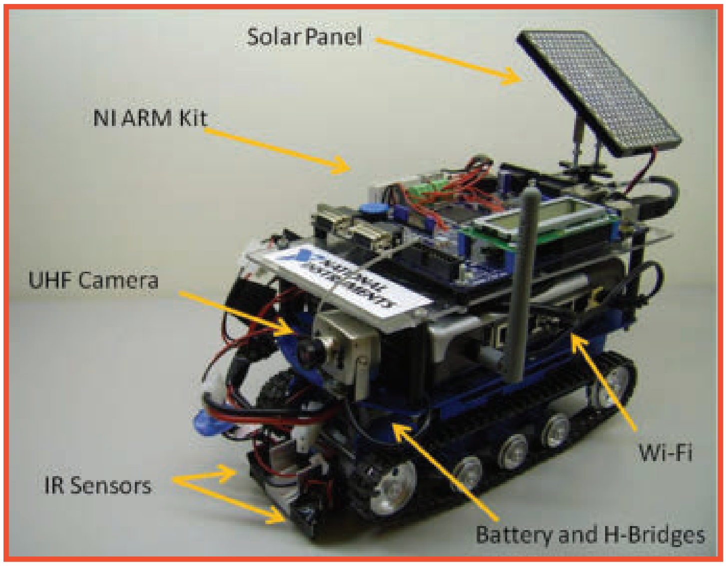
by David Geer

Contact the author at geercom@windstream.net

NI Robot Platforms

National Instruments LabVIEW graphical programming combined with powerful embedded hardware such as the NI Single-Board RIO enable engineers and scientists to develop sophisticated autonomous systems and is used extensively in robot creation for a variety of applications. Resulting robots perform sensor-based exploration and reconnaissance, and serve as demonstrations of what roboticists can achieve with the platform.

Let's examine the basic elements of a few of NI's robotic platforms and demonstration robots, and other NI based submissions.

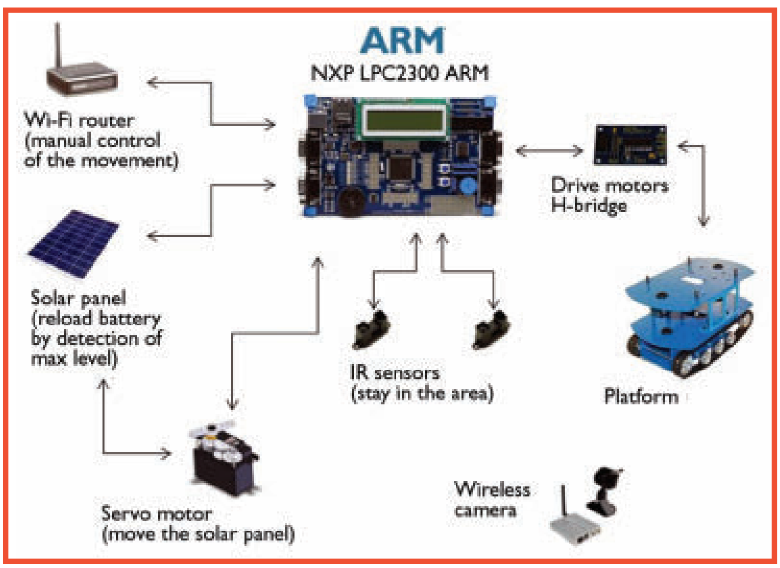


The Charsol robot with its components identified in their respective places, including the solar panel on top, the NI ARM processor kit in front of that, the UHF camera in front, the IR sensors below, the battery and H-bridges beneath the camera, and the Wi-Fi router under the processor board.

The Charsol Robot

Charsol (French for solar tank) is an autonomous robot that senses its environment via video camera and two infrared sensors from Radiospares. Operators can use the robot to explore the environment while resting assured it will continually replenish its battery (a Conrad) by seeking out strong light sources for its solar panel (from **Econologie.com**).

Sensor communications are traced from the camera to a UHF transmitter, then via Wi-Fi (wireless) to a PC. "The host PC contains an interface to capture and convert the UHF signal to



A better view showing individual Charsol hardware components.

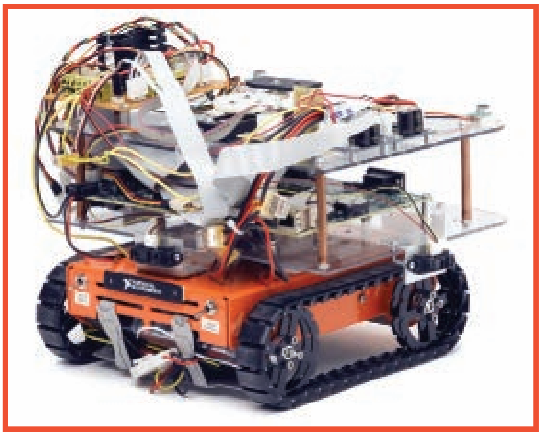
USB using integrated Wi-Fi communications," according to NI. The robot uses its sensors to track a predefined area that it must remain in to receive possible transmissions from an operator using teleoperation.

This robot makes good use of the LabVIEW embedded module for ARM microcontrollers. This LabVIEW component is useful for routing and interpreting sensor data, making decisions, controlling motors or actuators, moving solar panels for optimum light exposure, controlling Wi-Fi, and it uses an LCD display to report the robot's status and the amount of light hitting the solar panel.

A servo controls the angle of the panel which is the battery's only means of recharging. An H-bridge (from Personalize par transistors) alters current polarity to rotate the motors forward and backward.

Accurate estimates and controls around power usage are crucial to making the robot autonomous; this gives it enough power to use all its technologies and perform all its tasks on its own without running out of energy. The H-bridge controls the current to the motors to prevent spikes which could occur during motor activation and could drain too much power. To take some load off the battery, two regulators step the battery voltage down for the ARM processor board and the wireless router which each require 5V. Ultimately, a larger solar panel can be added.

The existing panel only powers the analog-to-digital input to the ARM Keil MCB2300 card which needs about 3.3V. Within the Wi-Fi network, the robot-based router automatically assigns IP addresses for the onboard ARM MCB2300 that functions as the server, and the offboard PC that serves as the client.



NIRo from the opposite angle.

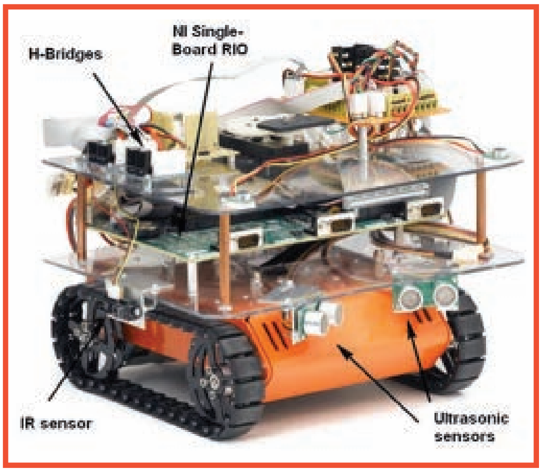
NIRo

NIRo is a small UGV programmed by LabVIEW Real-Time and FPGA Modules, and built on the Single-Board RIO embedded platform using off-the-shelf components exclusively. The robot applies three infrared sensors and two ultra-sonic sensors for object avoidance and navigation.

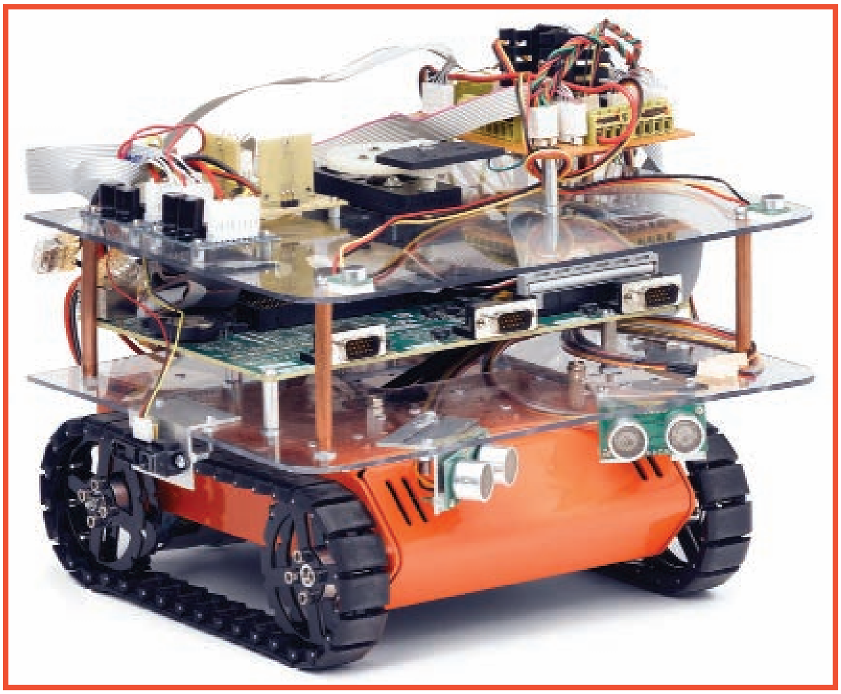
The robot's embedded controller (sbRIO-9632) manages its sensor data, motor control, and intelligence. The infrared sensors (Sharp GP2D12) detect objects via distance calculations based on the time it takes for the infrared light to reflect back to the sensor.

The robot's ultrasonic sensors (Devantech SRF05) detect objects based on the time it takes sound to return to the sensors. The robot uses an H-bridge (Solutions Cubed) to move current to the motors from the battery.

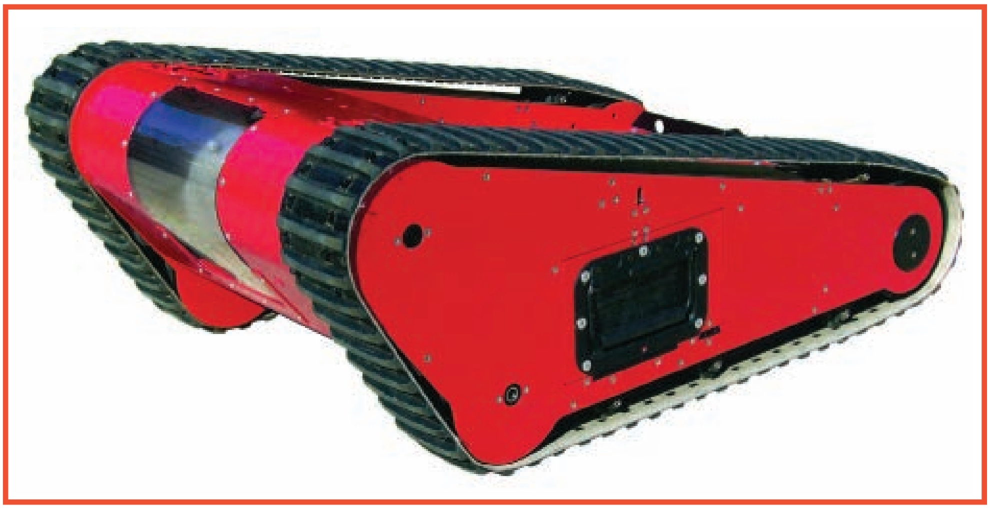
All the robot's communications and networking are internal, with the robot only operating in an autonomous mode. First, the robot's controller manages obstacle avoidance tasks. "The obstacle avoidance algorithm uses



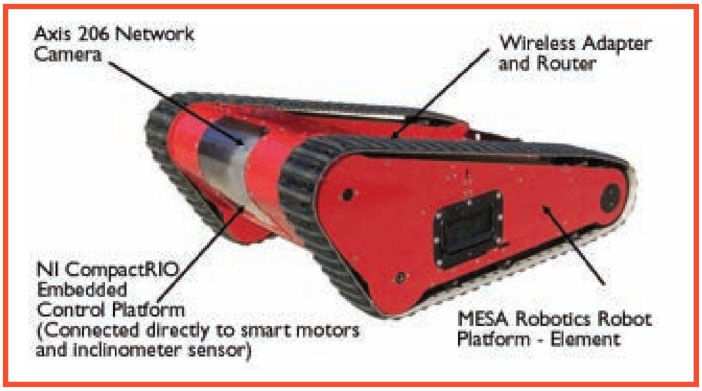
See the H-bridges up top, the Single-Board RIO below, the IR sensor to the left of the tread, and two pairs of ultrasonic sensors out front.



This is an iteration of the NIRo robot — an in-house demonstration platform built by NI engineers. The robot showcases what roboticists can do with the platform using off-the-shelf components from a local hobby store. Engineers based the UGV (Unmanned Ground Vehicle) on the Single-Board RIO embedded platform.



Note the window out in front where the Axis 206 network camera resides, and the locations of the router and the CompactRIO embedded controller.



All of the Element robot's components are internalized.

the data from the IR and ultrasonic sensors to make decisions about how to navigate its environment," per NI.

Second, the robot controls the motors via its FPGA (Field Programmable Gate Array) using outputs from the obstacle avoidance algorithm. The FPGA outputs pulse-width modulated signals through the digital I/O lines which control the motors.

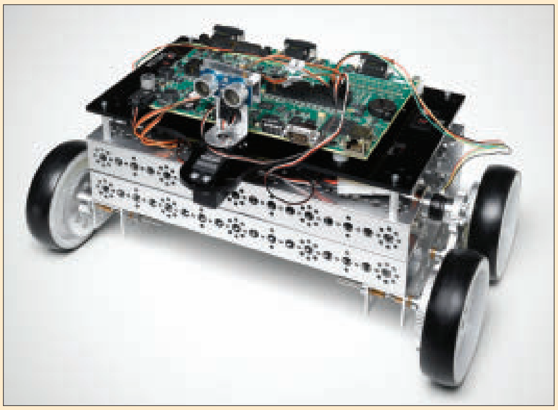
To perform obstacle avoidance, NIRo takes the values from the infrared and ultrasonic sensors, and when the values show something close to the robot, it turns away from that object.

"NIRo will turn at a rate proportional to the distance the obstacle is away from him. The closer an obstacle is to NIRo, the faster he will turn away from it," says NI.

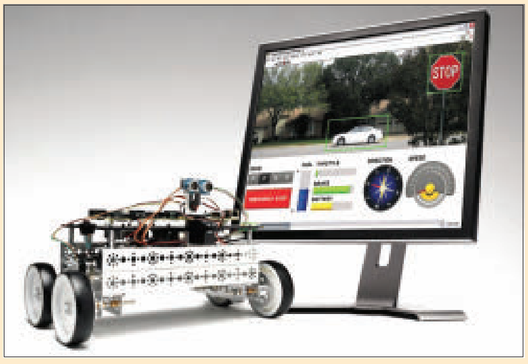
The robot's creators designed NIRo to stay a "desired" minimum distance from objects. NIRo will veer away from any obstacle. NIRo uses the output data from its obstacle avoidance algorithm to control the PWM signals for each motor. Depending on the output, it may tell the motors to move the robot evenly on both sides which moves it straight forward. If the output says to turn one way or the other, the signals to each motor will be different so that one motor will move its side forward faster than the other motor does. This will cause the robot to turn.

The signal that turns the robot comes from the real-time processor to the FPGA. The FPGA interprets the commands that it sends to the H-bridges. The H-bridge

DaNI



If you are new to the LabVIEW and NI RIO platform, the LabVIEW Robotics Starter Kit, "DaNI," is available as an evaluation or prototyping platform. This pre-built platform includes an NI Single-Board RIO, a PING))) Ultrasonic distance sensor, motors, encoders, and a mechanical frame from Pitsco Education. Out-of-the-box, DaNI uses the LabVIEW Robotics Module to perform obstacle avoidance and can navigate autonomously.



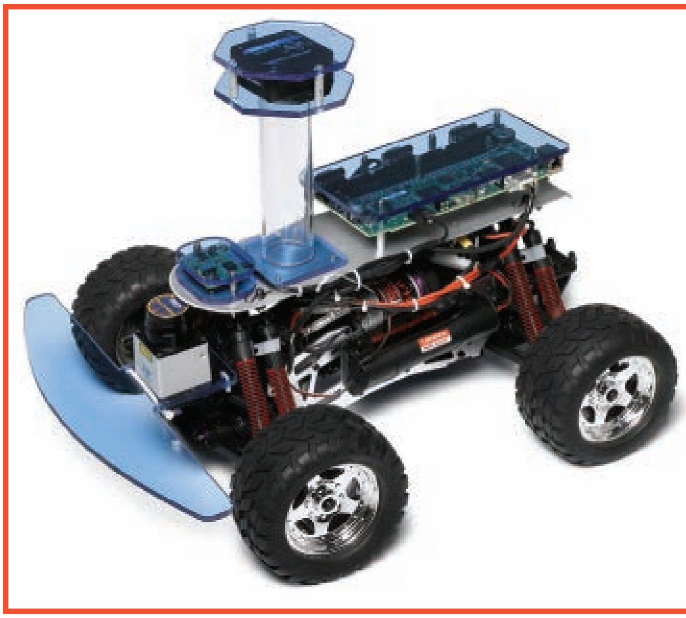
This expandable platform features built-in digital and analog I/O, connectors to NI C Series modules, in addition to an on-board Ethernet and serial port allowing for the connectivity of additional third-party sensors and actuators. DaNI includes a 180-day evaluation of the LabVIEW platform in addition to its real-time module, FPGA module, and Robotics module.

What Is LabVIEW?

LabVIEW is a graphical programming environment used by millions of engineers and scientists to develop sophisticated measurement, test, and control systems using intuitive graphical icons and wires that resemble a flowchart. LabVIEW offers integration with thousands of hardware devices and provides hundreds of built-in libraries for advanced analysis and data visualization. The LabVIEW platform is scalable across multiple targets and operating systems.

signals the motors so they turn the robot, move it forward or backwards, or even keep it stationary.

NIRo is a good showcase of what the NI platform can do to provide the foundation for large, complex robots that do object avoidance and exploration. "The use of LabVIEW Real-Time, LabVIEW FPGA, and Single-Board RIO enables developers to easily integrate hardware and software, and to rapidly design, develop, and deploy algorithms," explains NI.



The NIcholas demo robot is a smaller scale autonomous UGV. Nicholas uses a LIDAR sensor to visualize its surroundings by sensing distance/depth as the radar bounces off of surfaces.

The Element

The Element is another demo built for teleoperated search and rescue. The tank style treads and ability to build everything inside the robot stems from a platform from MESA Robotics. Two motors mobilize the robot controlled by an NI CompactRIO controller. A webcam provides the operator with a view of Element's line of vision. A Windows PC communicates with the robot via 802.11 wireless. The operator enters the robot commands via a 360 wireless controller. Teleoperated from a distance of 200 feet, the human operator uses the controller to maneuver the robot within its intended environment. Thanks to the treads, this includes difficult terrain (indoors and outdoors) and stairs.

The controller sends signals via Bluetooth wireless to a laptop running LabVIEW, then on from there to the controller on the robot. The controller processes the commands in real time and forwards them to the smart motors for action.

Element uses a sensor called an inclinometer to read its level of elevation to keep track of the terrain. The data is useful in constructing a 3D model of the robot on its current terrain.

NIcholas

NIcholas is a much smaller scale UGV that uses autonomous navigation techniques to explore its surroundings. The robot gathers a visualized impression of its environment using a Hokuyo Laser Rangefinder (LIDAR). A Vector Field Histogram (VFH) algorithm helps it to avoid obstacles. NIcholas uses an inertial measurement unit to sense velocity and acceleration.

The LIDAR sensor is connected to a serial port on the processor, while the IMU is connected to digital lines on the Single-Board RIO. The motors are also connected to this.

Obstacle avoidance and drive topologies are much the same as with the NIRo robot. The LIDAR uses a laser to measure the distances and angles to nearby objects. This information is sent to the VFH algorithm to compute a path which best avoids these nearby obstacles. Once the path is computed, the FPGA is used to send PWM signals to the drive and steering motors.

Conclusion

The NI robots make great experimental platforms to develop and test algorithms on. Robot recipe code is available as a free download to the community. **SV**

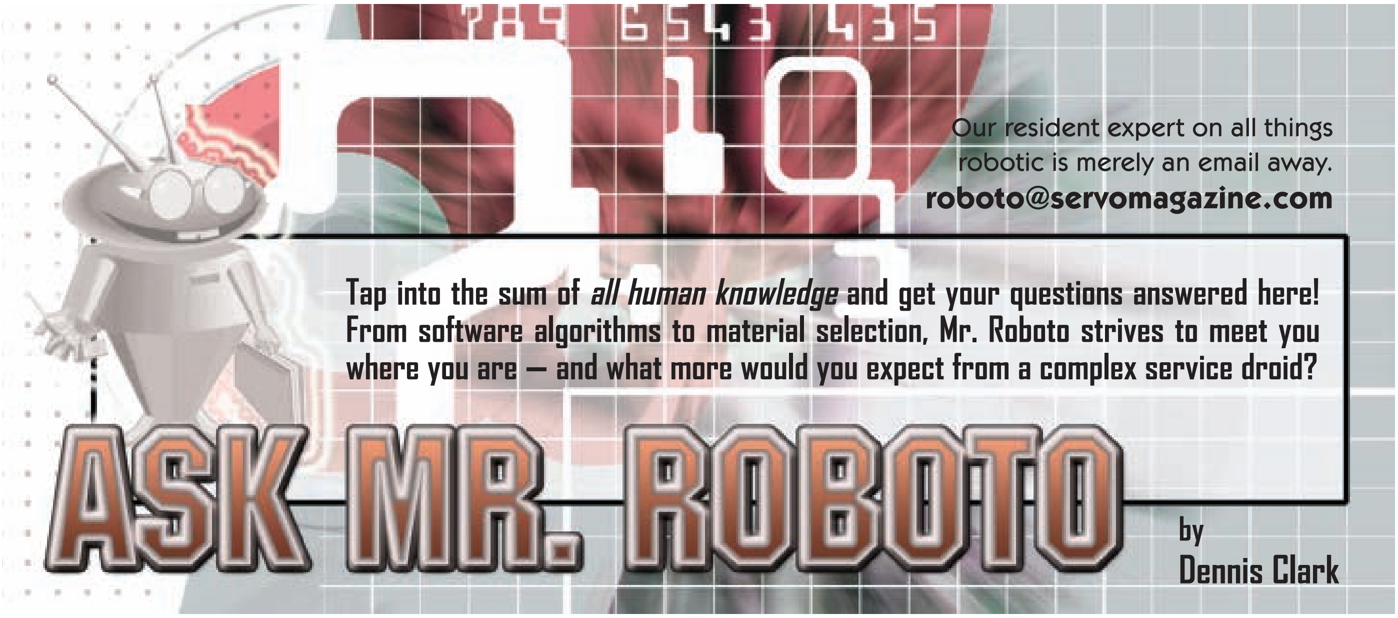
Resources

National Instruments
www.ni.com/robotics

LabVIEW for ARM Microcontrollers
http://sine.ni.com/nips/cds/view/p/lang/en/nid/205040

LabVIEW Robotics Software Bundle
http://sine.ni.com/nips/cds/view/p/lang/en/nid/208170

Other LabVIEW Data
www.ni.com/labview/optin



It's quiet here in my town right now, so it's the perfect time to answer some questions!

Before I get started, I do have an addendum to an answer I gave in my June '10 column dealing with creating a custom servo controller that could control the span of a servo. I noted the Pololu Serial Servo Controller ROB-08897. Pololu informed me that this is a bit dated and that they have a much better solution: the ***Micro Maestro six-Channel USB Servo Controller*** *found here:* ***www.pololu.com/catalog/product/1350***.

This is the replacement for the older board that I was familiar with and it has better specs, too.

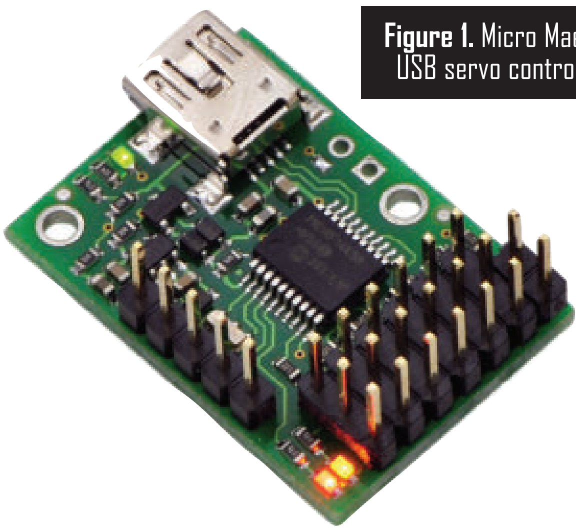


Figure 1. Micro Maestro USB servo controller.

Q. I have heard of a method of using your printer to mask off where the traces go on copper clad boards. Are there any good books or literature where I can learn more about this and what is the most cost-effective method? Any ideas would be greatly appreciated,

— Shawn

A. I have done this process a few times and I considered writing about my adventures, but I realized that there is a better place to get good details on how to use a laser printer to make iron-on board transfers ... **www.instructables.com**! This link looks exceptionally well detailed: **www.instructables.com/id/DIY-Printed-circuit-board**.

Be sure to read the comments about how to dispose of the etchant solution so you won't damage your plumbing (or your septic). One of the things that constantly plagued me when using the laser printer iron-on method was the iron temperature, force, and time I needed to get the transfer to work well. It really depended on the iron and the paper used. Your best bet is to use shiny paper (like photo paper) and a heavy or "high quality" print so you have lots of toner on your traces. Don't press too hard either; do multiple passes with only the weight of the iron on your board. Make sure that your surface is flat and sturdy so that you can apply even pressure across the entire surface.

I'm sure that you can find other sites and examples to help you out. The key to the process is to be patient and consistent. If you don't like the transfer, then scrub it off and try again. You aren't committed until you start etching the board! Good luck and have fun!

Q. I have tried to get an Eclipse AVR plug-in configured twice now. Last month, it was on a Ubuntu 9.10. The other day after installing Ubuntu 10.04, I tried it again. I get as far as having Eclipse installed and running. It is the addition of the AVR plug-in that I cannot seem to get accomplished. Could you elaborate or go over in more detail how to install the plug-in?

A. Here is a detailed way to install this plug-in using the IDE. I was going to show the alternate method of downloading the plug-in and unzipping the files, but all of the procedures I found on the Internet to do this were hopelessly outdated and won't work. Also, there doesn't seem to be a canonical install location on Linux, so everything that I found simply didn't work with my install. *So*, with that in mind, I'll pretend I didn't waste my time there and we'll just discuss the way to do this from within the Eclipse IDE. My current install is Eclipse Galileo; newer versions no doubt exist now, so hopefully they kept the same menu organization.

1: Start Eclipse.

2: Click on **Help->Install New Software**.

This will bring up the window shown in **Figure 2**.

The **Name** is just a title to let you know what you are doing when you look at your plug-in list later on. The **Location** field needs to be typed *exactly* as I show it: **http://avr-eclipse.sourceforge.net/updatesite**.

When you have all of this typed in, press **OK**. To find the plug-in on the site, you need to type in "avr" into the unnamed window shown in **Figure 3**. I've found that the AVR Eclipse plug-in will show up as soon as you type the "A." Don't forget this step or nothing will show up in the selection screen (**Figure 3**). Now, just click the **Next** button and follow the instructions.

Make sure that you select the plug-in by clicking on the selection box to the left of the ***AVR Eclipse Plugin*** tree view. After you click **Next**, you'll get a warning about an unsigned content; don't worry, just accept it. Then, you'll get a license screen that you have to accept to get your plug-in. After these steps, you'll get the screen shown in **Figure 4**; you are loading your plug-in. Click **Finish** and you're done.

3: Enjoy the view.

To bring up the AVR Device Explorer — which gives you all of the register names for your chosen part — navigate to: **Window->Show**

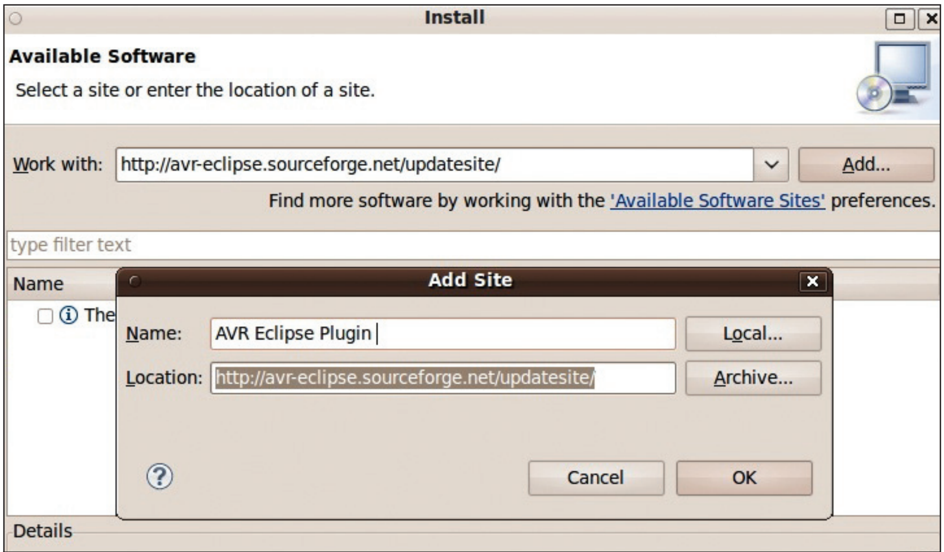


Figure 2. Locating the plug-in.

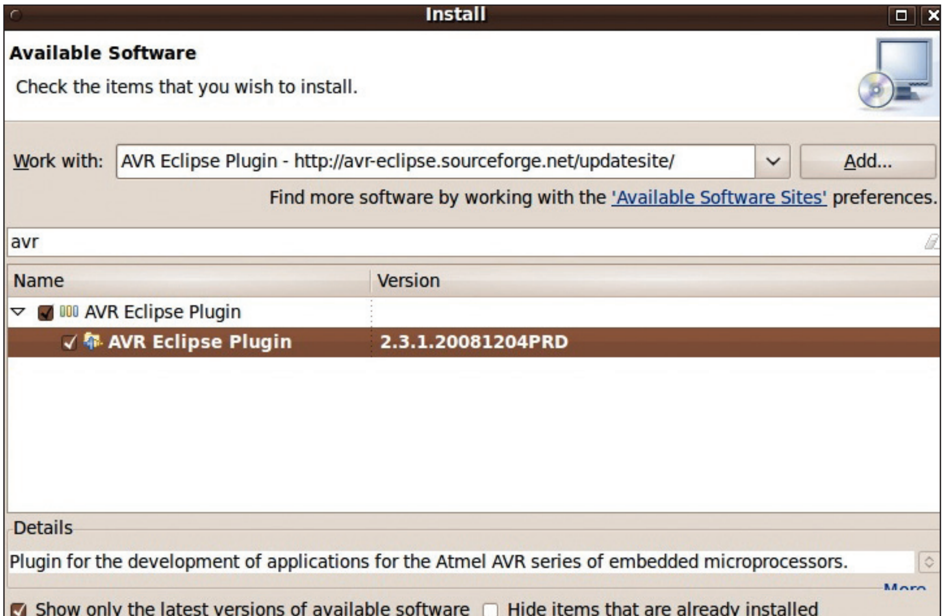


Figure 3. Selecting the plug-in.

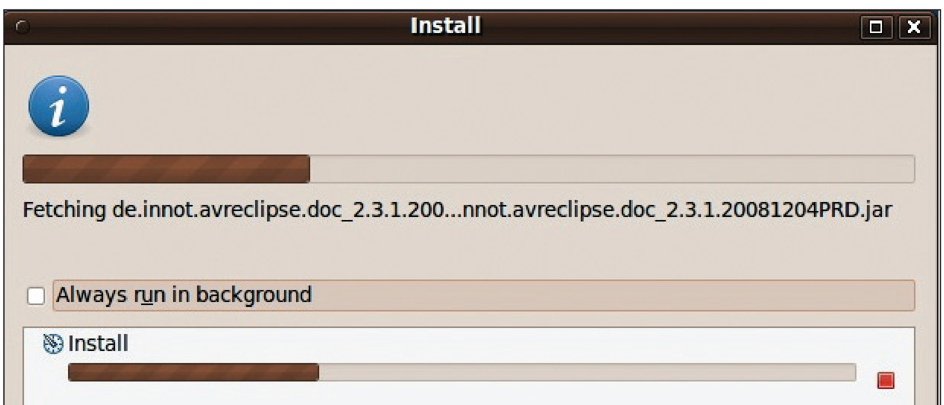


Figure 4. Your plug-in is being installed!

View->Other. Then, click on the AVR drop-down and select **AVR Device Explorer**. Experiment with this and the **AVR Supported MCUs** view, as well.

If you don't want the tab to be down at the bottom (the default), simply drag the tab to the top, left, or right side and it will reside there for you.

Again, type everything exactly (as before) and you will get your plug-in. Oddly, I've found that sometimes when I clicked on **OK** the button would highlight but not select. I'd have to hit **Enter** to actually select the button. This may be something odd in my install, however. I'm a Mac OS X kind of guy so I'm familiar with its quirks. I'm only now learning the Linux and Windows "issues."

That wraps up my column for this month. I hope you find something that you can use in your robotic endeavors. As usual, if you have a robotics related question, please drop me a line at roboto@servomagazine.com and I'll do my best to answer it. Have fun! **SV**

NEW PRODUCTS

THRUSTERS

Low Cost ROV/AUV Thrusters

CrustCrawler is releasing a lower cost version of their popular "Hi-Flow" series ROV/AUV thrusters. At just 6.25" (158.75 mm), they are compact, lightweight, and powerful. The Hi-Flow Model 400HFS will feature all aluminum construction, 85 mm four blade propeller, machined aluminum propeller adaptor, and a type 37 Kort nozzle. The 400HFS is targeted for the light industrial, education, and hobby market where a low cost, high quality, low maintenance thruster is required. Target price will be $459/thruster and will be sold in single units, three-packs, and six-packs.

CrustCrawler's new 400 watt thrusters were engineered to be rugged, customizable, and provide maximum thrust in a compact, easy to integrate package for all types of ROV/AUV's. With the optional programmable brushless speed controller, the throttle and brake curves can be customized to provide the programmable flexibility needed to set specific thrust parameters for different underwater conditions. These units are machined from top-of-the-line materials, embedded with a gear reduced, powerful brushless motor, and sealed with a custom engineered, replaceable fluoroloy spring jacket seal with a life span of 7–10 million cycles.

Some key features include:

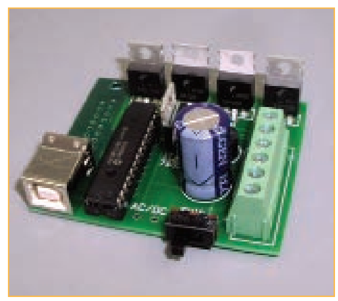
- 5,000+ ft depth rating (oil compensated).
- Propeller is efficient in both forward and reverse thrust.
- Type III hard anodized finish for maximum corrosion resistance in salt water.
- Rugged 6061 aluminum thruster housing with weld re-enforced Kort nozzle adaptor.
- Powerful 600W brushless motor with a 4.28:1 gear ratio.
- Operates in all of the popular UAV/ROV voltage ranges from 0 to 50 volts.
- Field serviceable components for low cost ownership.
- High reliability.
- Complete field replaceable thruster heads available for quick field repair.

For further information, please contact:

CrustCrawler	Website: **www.crustcrawler.com**

MOTOR CONTROLLERS

USB Stepper Motor Kit



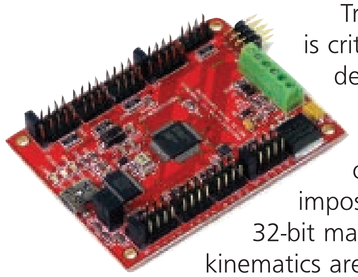
A USB stepper motor controller kit is now available from Images Co. that allows the user to control a five volt or 12 volt uni-polar stepper motor from Windows. The exclusive Windows interface is designed to teach fundamental stepper motor control. Automatic and manual control of the stepper motor is available through the Windows interface. Windows controls available include speed (steps/second), direction (Clockwise [CW] or Counterclockwise [CCW]), full step, half step, and manual step. A tutorial booklet explains all the basic functions of the uni-polar stepper motor.

For further information, please contact:

Images Scientific Instruments	**Website: www.imagesco.com**

Servo Controller

The ARC32 from Lynxmotion is a state-of-the-art robotics and servo controller. It utilizes a hybrid hardware solution to drive 32 servos with less than 1% CPU time. In addition to being based on a blazing fast 32-bit microcontroller, it has an integrated 32 channel servo control.



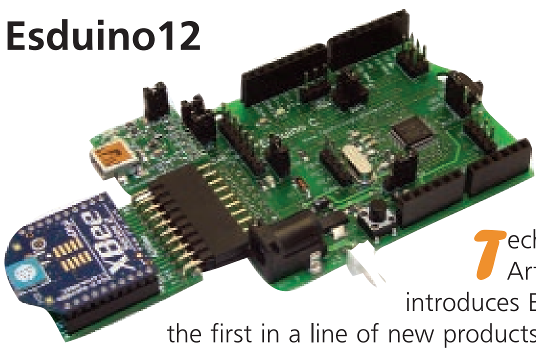
True 32-bit math hardware is critical for the most demanding robotics application. Creating position data for 32 servos on the fly is nearly impossible without high speed 32-bit math. Things like inverse kinematics are now possible using the ARC32. The ARC32 is programmable in Basic, C, or ASM using Basic. Price is $99.95 USD.

For further information, please contact:

Lynxmotion	Website: **www.lynxmotion.com**

CONTROLLER BOARDS

Esduino12



Technological Arts introduces Esduino12, the first in a line of new products conforming to the popular open-source Arduino hardware form-factor. Based on the popular 16-bit Freescale 9S12C microcontroller, Esduino12 offers enhanced capability and programming flexibility over existing Atmel-based designs, yet retains hardware compatibility with the growing number of available Arduino application shields.

Esduino12 is programmable in assembler, BASIC, and C. For assembler and C programming, Freescale's free Special Edition of CodeWarrior for Windows is a popular choice. For those who prefer BASIC, the object-based Windows-hosted nqBASIC language is available free of charge from **www.nqBASIC.com**.

With a base price of only $39, Esduino12 offers plug-in options for both USB and wireless XBee communications (no shield required), and provides simple three-way routing that enables the user to choose USB-to-XBee, USB-to-MCU, or XBee-to-MCU communications. Both 3.3V and 5V regulators are provided on-board, giving added utility to the design.

While a standard BDM connector is provided for advanced users, the Freescale Serial Monitor (i.e., bootloader) comes factory-programmed into the 9S12C chip so that a BDM pod is not required for basic erasing and programming operations.

Two product configurations are being offered: #ESD12C32 ($39) with 32K Flash and 2K RAM and #ESD12C128 ($49) with 128K Flash and 4K RAM. The optional USB-to-TTL interface (#USB2MCU) is $15.50, and a plug-in carrier for XBee modules (#ADXB-RA) is $7.50.

For easy implementation of custom applications, a selection of four new Arduino-compatible $10 prototyping shields has also been announced. These shield designs maximize use of the available board area, and are optimized for applications involving narrow and wide DIP packages, LED displays, and discrete through-hole components.

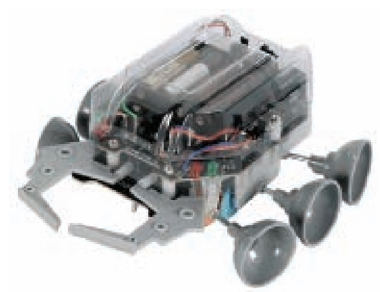
For further information, please contact:

Technological Arts	Website: **www.technologicalarts.com**

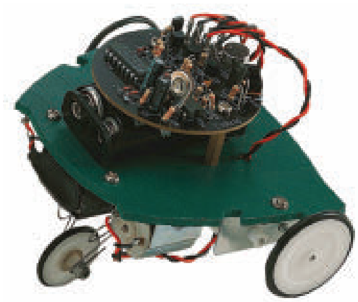
KITS

Four New Robot Kits

The Escape Robot Kit from Ramsey Electronics uses basic artifical intelligence to navigate, and it never fails to find its way out of a maze. Using three IR emitting diodes and one IR receiving module, it sends and receive signals to detect obstacles in its path. The Escape Robot's built-in microprocessor enables it to "think" on its own: It gathers and processes information on its environment. The Escape Robot crawls around on six legs and comes with another set of differently designed legs that move in their own unique way. It's powered by four AAA batteries (not included) and measures 5.5 x 5.9 x 3.9." Price is $34.95.



Ramsey's Scarab beetle-inspired robot uses two touch sensors to detect obstacles. When its sensitive antennae detect something blocking its path, the scarab will first step back and then automatically execute a two-step maneuver to skirt the obstacle. The maneuver is a combination of "left turn," "right turn," "reverse," or "stop." The robot can also be configured with different sets of movements. It's powered by four AAA batteries (not included) and measures 6.9 x 5.7 x 3.3." Price is $34.95



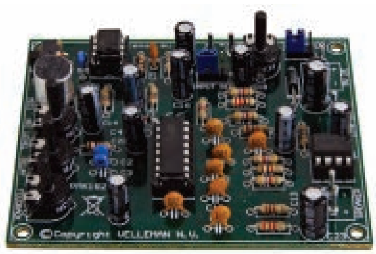
The Auto-Turning Frog Robot kit is one of the more interesting kits available from Ramsey. Once soldered together, the fun begins. When it detects sounds, it will move and repeat the following steps sequentially: Start (move forward), Stop, Left Turn, Stop, Right Turn, Stop. These kits offer an introduction for students to electronics and gives them an interesting novelty once they are done. It comes complete with detailed instructions and is recommended for ages 12 and up. It uses two AA batteries (not included) and one 9V battery (not included). It measures 3.5 x 4.9 x 3.9" and is priced at $23.95.

Ramsey's Ladybug Robot walks on six legs and uses infrared emitting diodes for "eyes" in order to avoid obstacles. The bot will walk straight ahead until it "sees" an obstacle. Then, it automatically makes a left turn and goes along its way. It comes complete with detailed instructions and is powered by four AAA batteries (not included). It measures 4.7 x 5.9 x 3.3" (feelers length: 1.6") and is priced at $29.95.

For further information, please contact:

Ramsey Kits	Website: **www.ramseykits.com**

Digital Echo Chamber Kit



QKits now has a digital echo chamber kit that can be used for various animatronics projects, audio systems, karaoke, P.A. toys, or just for fun.

Features include:

- Four adjustments for MIC or input level, volume or output level, echo delay time, and feedback.
- 40 Kb SRAM for low distortion and good sound quality.
- Onboard microphone and audio amplifier.
- Line-In and Line-Out connection.
- 9V battery snap (included).

Specifications are:

- Power supply: 9V battery (6LR61 not incl.) or 9-12V DC regulated.
- Power consumption: 150 mA max.
- Delay time: 80 and 200 ms.
- Speaker output: 500 mW (8 ohm, 10% THD).
- Dimensions: 3.35 x 2.76 x 0.78."

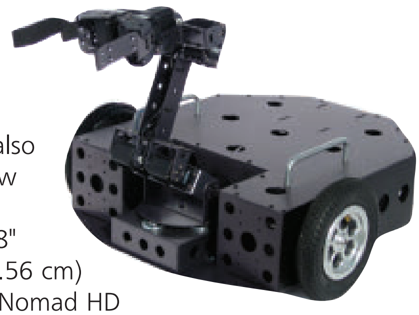
Price is $21.95; or 10-49 items at $19.76 each

For further information, please contact:

QKITS	Website: **www.qkits.com**

PLATFORMS

Nomad Heavy Duty Rover



CrustCrawler has also released their new Nomad Heavy Duty Wheeled Rover. At 18" (45.72 cm) x 14" (35.56 cm) x 4" (10.16 cm), the Nomad HD rover can carry everything from laptops to any assortment of custom electronics, batteries, and hardware accessories. The Nomad also features Parallax's wheel kit with all aluminum wheels and inflatable treaded tires; a "robotic arm deck" which accepts CrustCrawler's SG5-UT, SG6-UT, and AX12 smart robotic arms; and the S3 Tilt/Pan system for cameras and a variety of sensors. Machined from .063, .090, and .25" solid aluminum, the rovers can heft a payload capacity at more than 10 pounds.

For further information, please contact:

CrustCrawler	Website: **www.crustcrawler.com**

EVENTS Calendar

ROBOTS.NET

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

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

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

— R. Steven Rainwater

AUGUST

6-8 **Rescue Robot Contest**
Kobe, Japan
Autonomous and teleoperated search and rescue robots.
http://rescue-robot-contest.org

9-13 **AUVS International Aerial Robotics Competition**
Puerto Rico
Autonomous aerial robots and sub-vehicles compete in a series of tasks.
http://iarc.angel-strike.com

16 **ROBOMO Competition and Expo**
Missouri State Fair, Sedalia, MO
ROBOMO (the St. Louis Area Robotic Group) will be joining with various other local robot groups for a day of robot competitions of all types for all ages.
www.robomo.com

SEPTEMBER

3-6 **DragonCon Robot Battles**
Atlanta, GA
Autonomous and remote control robots battle at the popular science fiction convention.
www.dragoncon.org

5-12 **Microtransat Challenge**
County Kerry, Ireland
Autonomous sail boats start on a race across the Atlantic ocean; this will take three to four months to complete.
www.microtransat.org

15-19 **FIRA Robot World Cup**
Bangalore, India
This is the big annual robot soccer competition which includes RoboSot robot soccer, MiroSot micro robot soccer, HuroCup single humanoid robot soccer, AndroSot multiple humanoid robot soccer, and AMiRESot minirobot soccer. If your robot likes to play soccer, this is the place to be. Don't forget your vuvuzela!
www.fira.in

18 **RobotCup Junior Australia**
Canberra, Australia
Students and mentors create robots that compete in challenges such as robot soccer, robot rescue, and robot dance.
www.robocupjunior.org.au

18 **Robotour**
Bratislava, Slovakia
Autonomous robots must perform a navigation task in a park.
http://robotika.cz

OCTOBER

3-10 **Devyanin Mobile Robots Festival**
Moscow State University, Moscow, Russian Federation
Autonomous robot race.
www.mobilerobots.msu.ru/en

22-24 **Critter Crunch**
Hyatt Regency Tech Center, Denver, CO
Robot combat — 2 lb and 20 lb event categories. Autonomous and remote control.
www.milehicon.org

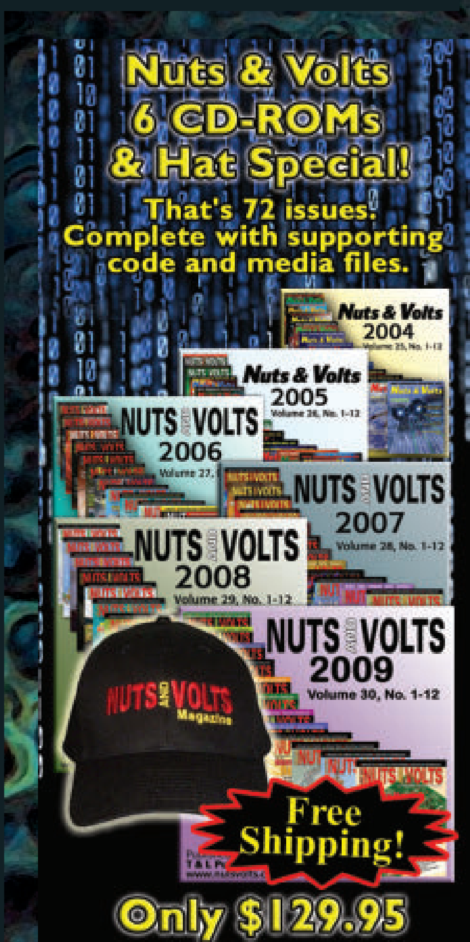
22-24 **SRS Robothon**
Seattle, WA
This year's Robothon include Robo-Magellan, Robot Combat, Double-Cross, and Brickheap Wars.
www.robothon.org

NOVEMBER

2-3 **Junior Robotics Challenge**
Singapore
Line-following can collection.
http://jrc2009.webs.com

7 **International Micro Robot Maze Contest**
Nagoya University, Japan
Micro Robot Racer (1 cm cube), Climbing Competition (1 cm cube), Maze Solver (1 inch cube), and Two Leg Robot Competition (2 inch).
http://imd.eng.kagawa-u.ac.jp/maze

Robotics Showcase



bots IN BRIEF

SKATING AROUND THE ISSUES

Hitachi's EMIEW2 humanoid robot uses wheels on its legs instead of feet. This approach gives the robot efficiency and stability while moving, along with the potential for it to make its way over obstacles (sort of).

EMIEW2 has adaptive suspension, but it's unclear as to whether the robot can actually balance itself on one wheel which would be necessary for it to "step" over an obstacle larger than its wheels are capable of negotiating. It can do precision speech recognition and sound localization which is handy, plus it's also kinda cute.

As with most Asian humanoid robots with fancy plastic shells, Hitachi suggests many ways in which EMIEW2 might be commercially useful (the obligatory guide/guard duties), but doesn't offer any sort of cohesive commercial details.

EMIEW2 stands about 32 inches tall and weighs about 30 pounds. It can move around on either two or four wheels depending on its leg position, and navigates via a scanning range finder. Aided by its active suspension system, incorporating a linear actuator movement mechanism and conventional wheel legs, it can roll around obstacles on uneven floors without losing its balance and it is especially mechanized to detect wheel slip. This makes it efficient to overcome the threshold of elevator doors and steps with more stability.

Its top speed is reported to be 3.7 MPH and it has 25 axes of movement.

It separates the noise out using three-dimensional sound localization with a 14-channel microphone in its head.

SHEDDING SOME LIGHT

LuminAR, designed by Natan Linder, is not the first robotic desk lamp to come out of MIT. In 2007, recall AUR, a five DoF robotic lamp designed to follow your movements around a desktop, changing the color, intensity, and focus of its light to help keep you organized and productive. LuminAR is similar in concept, except it's significantly more complex, incorporating a pico projector along with a vision system to allow for a large degree of interactivity.

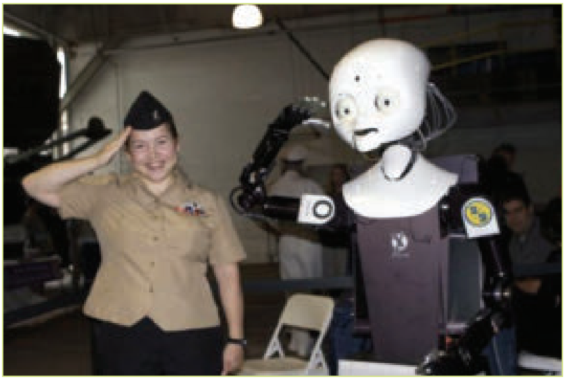
Fortunately, all this interactivity looks to be intuitive rather than overbearing.

QUADRUPLE THE CLEANING POWER

People use Roombas (and Creates) as bases for all kinds of robots. They're not often used for bigger (heavier) bots, but if you stick four of them together, apparently you can support a payload of up to 20 kg or one Robotinho android. The Roomba QuadDrive was developed by the University of Bonn's Autonomous Intelligent Systems Lab, and consists of four Roomba 530s bolted together. As far as mobile bases go, it's probably pretty cheap and reliable. The top speed of 0.5 MPH isn't going to amaze, but it's fine for tooling around a museum and scaring little kids.

Of course, the floors of the museum are pretty damn clean.

bots IN BRIEF



I'M IN THE NAVY NOW!

About a year ago, Nexi took part in a somewhat silly (but reasonably effective) demonstration of a robotic role on a US Navy vessel. Guess the Navy was impressed, because they now own one of their very own. This Nexi is named Octavia.

The Navy hopes to use Octavia to explore how humans and robots interact, with the aim of minimizing the amount of time that humans spend dealing with a robotic interface and maximizing the amount of information that can be communicated. Octavia specifically is good for things like this because of her intensely expressive (and somewhat uncanny) face and head.

SOCCER IT TO ME

In human World Cup soccer watch humans have pretty much peaked. Really, the only thing about the game that evolves reliably from cup to cup is the ball. Robots, on the other hand, have no such limitations. Carnegie Mellon's small-size robotic soccer team (CMDragons) has taken another step towards robot domination by teaching these small size soccer bots the physics of ball movement. A robot that knows physics is pitted against a robot that doesn't. The robots are autonomous and are controlled by a computer that watches the action on an overhead camera.

Without modeling the physics of the ball, the computer just tries to position the robots on the ball without taking the movement of the ball into account. A physical model allows the computer to move the robot predicatively, greatly improving its skill.

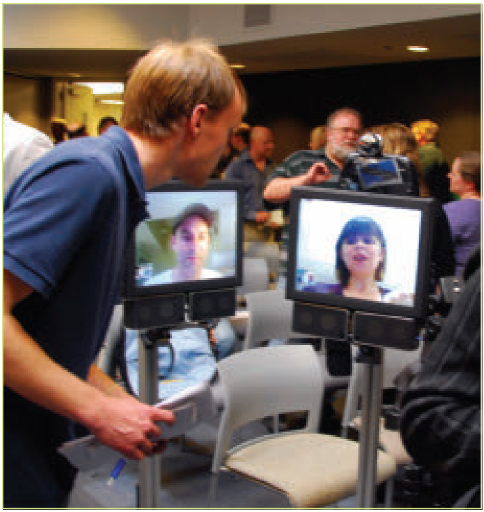
OPEN QBO

The Corpora has started up a YouTube channel devoted to their open source robotics project, Qbo. Their first video shows the stereo camera calibration method.

GET YOUR ZEN ON

Among the robots that were presented at the Shanghai World Expo 2010 was Aldebaran Robotics' NAO (representing France). A large group of them performed Tai Chi motions in unison much to the thrill and amazement of attendees.

Cool tidbits herein provided by Evan Ackerman at www.botjunkie.com, www.robotsnob.com, www.plasticpals.com, and other places.



ANYBODY THERE?

Telepresence is great in concept, but it's a strange combination of being somewhere and not being somewhere, and interactions with people are different in ways that range from subtle to drastic. Willow Garage has been using Texai in their office for quite a while. One of their employees, Dallas Goecker, 'commutes' daily from Indiana to California via Texai. So, they've been figuring out some of these social rules as they go, to the point where some things are now a part of the Texai communication software:

A few built-in bits of etiquette include:

Texai Rule #1: If you see me, I see you. Explanation: It's about two-way communication. Implications: The cameras face forward because the screen faces forward. The pilots are only allowed to drive the Texai once they've shared their video stream.

Texai Rule #2: Texai does not record audio or video. Explanation: It's about face-to-face communication.

MOWED OVER

Summer is here and the grass is growing, so this is the year to get a robot to do the chore. This 28 x 28 x 15" Electric Robomower can move 360° and up a 30° slope. It can be programmed to run automatically or by remote, and has a built-in rechargeable battery. The two blades produce a cutting height of 1.6" at a speed of 6,000 RPM and the company promises a relatively quiet operation.

ROBOTS FOR $200

IBM has been hard at work for the past three years developing their latest Question Answering computer. Watson — named after founder Thomas J. Watson — is supposed to duplicate the human brain's ability to understand the meaning of words, know the difference between relevant and irrelevant content, and deliver final answers.

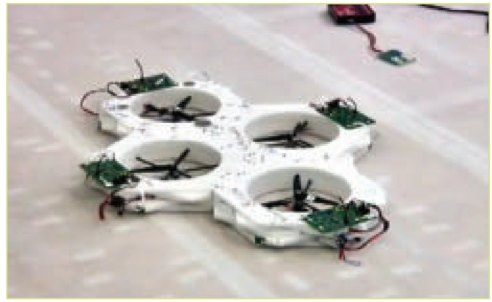
The company is so sure of the ability of their computer they are working on an avatar so that it can compete on Jeopardy later this year. The research has gone as far as getting the computer to answer in the form of a question.

An array of server racks have been endowed with linguistic algorithms allowing it to not only recognize oddly phrased or implicative questions, but to answer them in kind, with direct and accurate responses. Stuffed with encyclopedic knowledge of the world around it, it answers on the basis of information stored within its data banks.

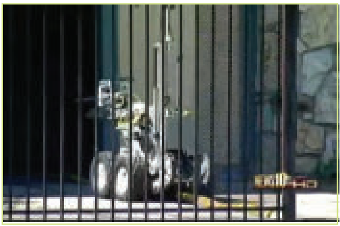
ROOMBAS ON STEROIDS

If you happen to travel to Tokyo, you may come across Narita Airport's cleaning robots. It is rumored that Narita-kun and Epo-chan do a bang-up job and are more fun to watch than your basic Roombas.

FLY UNITED

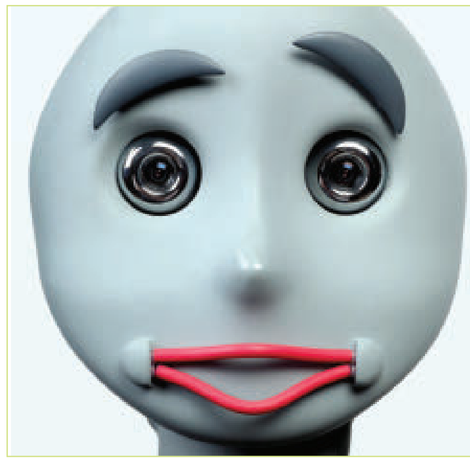


Scientists at the Swiss Institute of Technology in Zurich have developed a Distributed Flight Array that consists of individual bots that fly together in a swarm. The drones have infrared beams to find each other and then connect with magnets. While each quadrocopter is somewhat clumsy alone, when they dock they become a "sophisticated multi-propeller system." Even if one fails, the others can still function and, if attacked, can detach and later regroup.



SEND IN THE BOTS

During the recent two-day standoff in Sacramento, between police and an armed 25-year old man accused of robbing three banks and firing at a police officer, a robot was sent in to peek in through a window after attempts to force the suspect out with firehoses failed. Alavarez shot at the bot. It looks like the bot was okay as it was recently seen again.

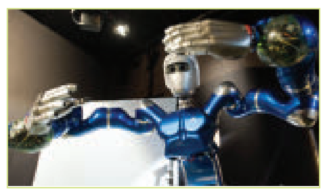


BANDIT HAS A VISION

Bandit was created to assist autistic children when family members may have problems in communication. Designed by researchers at the University of Southern California, the bot has camera eyes and has been programmed to make simplified facial expressions and movements. The hope is that he will help the kids become more socially active.

LOOKING FOR A ROMEO?

This is Romeo — the latest entry into the service bot category for assisting the disabled and the aged. Only a concept at this point, the 10 million Euro project will be able to do simple chores and interact with speech and gestures. Aldebaran Robotics should have a prototype by the end of the year and the real thing by 2011.



DANCING WITH THE STARS

SpaceJustin was designed to head into the final frontier and assist his human counterparts. The musical bot made his debut at the recent International Aerospace Show in Germany and did a bit of break dancing via telepresence. His other moves will include repairing and upgrading satellites. Wonder if he'll create his own dance channel ...

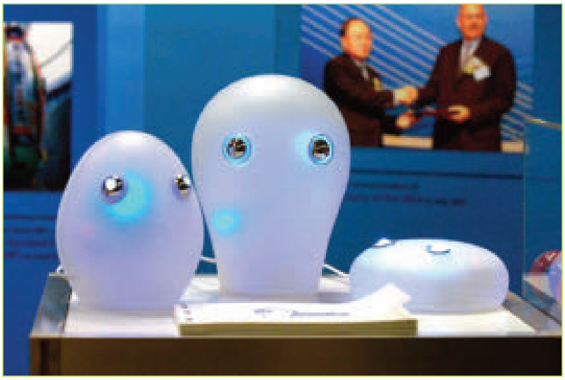


Image credit: HelloDD

BRIGHT IDEA

Students at KAIST and Yonsei University in South Korea have created a robot called Mung that they hope will instill the spirit of sportsmanship in players at events like the World Cup 2010. Mung was originally conceived as a "language purification" robot which would change color upon detecting profanities in elementary schools and at home — similar to the idea of a "swear jar" only without the monetary penalty. It was also used to promote the clean and safe use of nuclear energy.

SPARKFUN ANNOUNCES ANTI-ASIMOV CONTEST

Do you love robots? Do you love destruction? Are you tired of the same old robot competitions? SparkFun has just the thing for you: the SparkFun Antimov competition. Based on Isaac Asimov's Three Laws of Robotics, this competition challenges you to design a robot which *breaks* the laws of robotics (well, except for hurting humans, of course). A robot is traditionally designed to perform a complex task as efficiently as possible. Building a competent and efficient robot that completes its task unharmed is so last year, according to contest organizers.

SparkFun wants you to build a robot that completes a trivial task in the most inefficient and laborious way possible. Oh, and it needs to destroy itself doing so.

This competition focuses less on engineering abilities and more on creative ingenuity.

There is a $300 cash first prize, but this event is definitely more about the glory.

This is an incomplete list of the rules and is subject to change.

1. Robot must be autonomous. The only external control is a required kill switch which can be wired, but if so, must prove to not be a control mechanism.
2. Robot must complete at least one task performed upon an object external to itself.
3. Robot must destroy itself after the completion of the task in Rule 2.
4. Robot may use any prop, structure, or external device that can be placed in and removed from the exposition area. It can also use any of the existing features in the exposition area.
5. Robot cannot violate Asimov's first law, i.e., no hurting humans, and must have a reliable kill switch if something goes more wrong than intended.
6. Robots will be given three chances to complete their performance. If at any point during the performance the designer wishes to stop the robot, she/he can do so without penalty, up until the third run.

Scoring: Scoring will be based on a set of four criteria, listed below. Each category is 25 points for a possible total of 100 points.

1. Inefficiency (0-25 points) - the inefficiency of the initial task. The more complicated and inefficient, the higher the score.
2. Pointlessness (0-25 points) - the pointlessness of the performed task. The more trivial the task, the higher the score.
3. Drama (0-25) points) - the dramatic element to the robot's self-destruction. The more poetic, creative, and dramatic the death, the higher the score.
4. Destruction (0-25 points) - the completeness of the robot's self-destruction. The more in-operable and devastated the robot is at the end of its performance, the higher the score.

Prohibited Items: This is a partial list only. Each applicant will be required to send in a project proposal for review by a third party safety panel. If the delivered project is not as safe or safer than the proposed project, the competitor will be disqualified. SparkFun reserves the right to prohibit any item or items it feels will compromise the safety of the competition, its participants, or spectators.

This event is slated for October 16th, 2010. Go to **SparkFun.com** for more details and updates.

MONKEY BUSINESS

This robot arm controlled by a monkey brain has a whopping seven degrees of freedom, but that doesn't seem to phase the monkey much, as it deftly uses brain control to grasp a knob with the arm and receive a tasty reward.

The monkey relies on two brain implants (in the arm and hand areas of its motor cortex) to interpret nerve impulses and uses them to control the arm. The hope is (as least, as this technology applies to people with disabilities) to make the controller non-invasive, and some of the technology is sort of there. Sort of. Perhaps more importantly, this experiment shows just how capable and adaptable a brain is. Well, most brains.

A LEG UP

AIST's bipedal humanoid robot HRP-2 Promet was one of the robots on hand at the Japanese pavilion at the Shanghai World Expo 2010. (The robot shows its balance by standing on one foot.) A big part of Japan's display at the expo involved robots designed to help people; robots like Toshiba's ApriPoco that interact with home appliances through verbal commands, to robots that can move heavy objects.

CARS FOR BOTS

This 10 m (32 ft) tall robot statue reminiscent of the Transformers films was built using an old truck and other recycled parts (total cost was $44,000 USD).

HEAD CASE

Vstone designed this robot called RYOMA (Robot Yell Ohasuta Miracle Action) specifically for an appearance on Ohasuta (The Super Kids Station). It's based on the VisiON 4G which won Team Osaka its fourth consecutive victory at RoboCup in 2007, with one key difference: the head has been super-sized! The head alone weighs 1 kg and the omni-directional sensor on top isn't actually functional, but the eyes do light up with colorful LEDs. With three degrees of freedom in its neck "Ryo-chan" is capable of natural and cute expressiveness as it looks around. Despite its extra large head and the added weight, the standard walking gait works fine.

COMBAT ZONE

Featured This Month:

BUILD REPORT:

Bigger Better Beetle Blade

● by Pete Smith

The first blade I had created for our 3 lb beetleweight, Pure Dead Brilliant (seen in **Figure 1** with the original and new blades), had been done in a big hurry to meet weight for its first competition. As a result, it was shorter and lighter than I would have liked. The bot did reasonably well in competition, and there seemed to be no great hurry to replace that original blade. The 2010 Motorama event changed that when our blade lost yet another tooth, plus Maniac Kathy irreparably damaged it. This gave me the impetus to create a batch of blades that could be used on any PDB kit (available at **www.kitbots.com**) or indeed any beetleweight designed around it.

The blade was based on that of Maniac Kathy, but with some improvements. The original had been machined from 6061 aluminum and had simple flame


FIGURE 1

FIGURE 2


FIGURE 3

hardened teeth secured with 10-24 screws. The new blades would be in the much tougher 7075 aluminum and have professionally hardened S7 teeth secured with bigger 12-24 screws.

I designed the new blade assembly in SolidWorks® and used the "Mass Properties" tool to check that the weight was the same as the original Maniac Kathy version.

I like working with aluminum so I decided to make the blades myself. I'm not so keen on working with S7 steel, needing multiple teeth per blade, and with them also requiring heat treatment, so I sent my drawings to Team Whyachi (**www.teamwhyachi.com**) and ordered a batch of 10.

A lot of the time spent machining something like these blades is in setting up the machines for each operation. The time to make four blades is probably only 2X the time to make a single blade. With this in mind, I decided to make a batch of four blades.

Caution: Mills are potentially very dangerous machines. Do not wear any rings, a wristwatch, or jewelry; remove any loose clothing; tie back long hair; and wear eye protection at all times!

The first task was to cut four blanks (**Figure 2**) out of the 1/2" x 1-1/2" x 48" 7075 aluminum bar. Each blade was to be 11" long so each blank was cut to about 12"


FIGURE 4


FIGURE 5

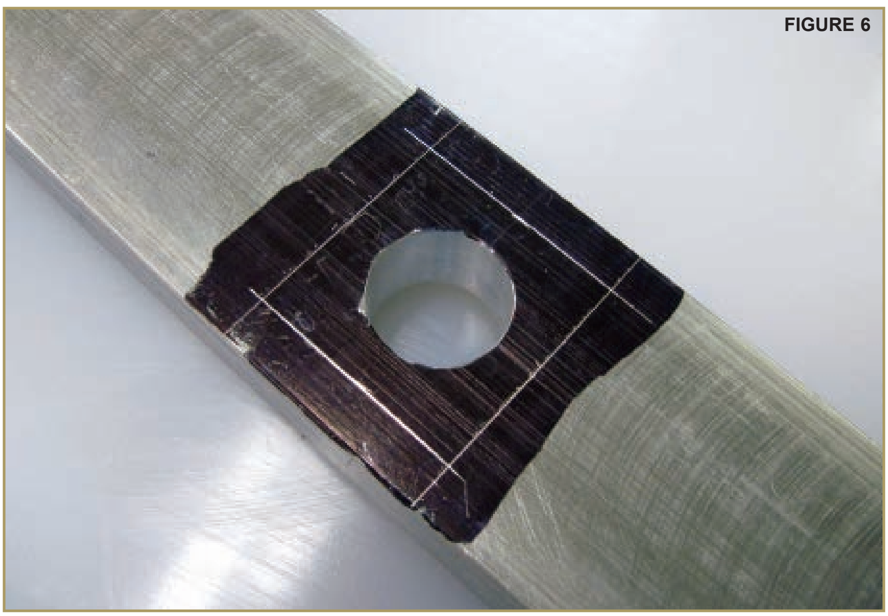


FIGURE 6

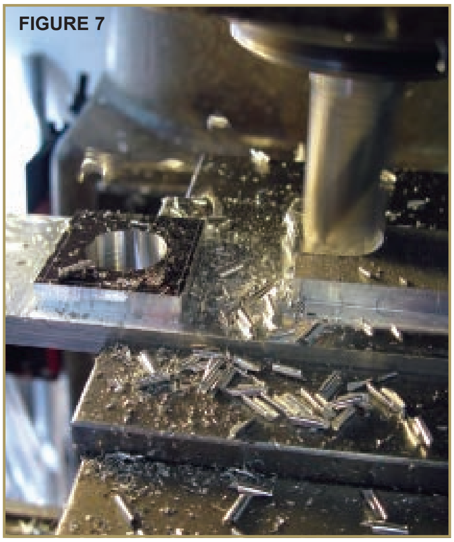


FIGURE 7


FIGURE 8


FIGURE 9

using a cutoff saw. Accuracy was not important as I would be machining the blades to their exact lengths later.

The blades are attached to the drive axles using a Trantorque™ bush. These work well but require the mounting hole to be accurate within a few thousandths of an inch. This can be done using a boring bar but requires considerable patience and careful measuring to make sure one does not go oversize.

To avoid this problem, I decided to use a three step process to quickly and accurately create the holes. The diameter required is 5/8" so after carefully positioning the mill head over the center of the bar, I milled a 1/2" diameter hole using a 1/2" end mill (**Figure 3**), followed by a 19/32" drill (**Figure 4**), followed by a 5/8" reamer (**Figure 5**) at low RPM and with plenty of lubricant. The table of the mill remains locked in position throughout the process.

The rationale behind this method is that the original milled hole will be straight; the drilled hole will be to within 1/32" of the required diameter and still reasonably straight; and then the reamer removes the last 1/32" to give a straight, very accurate hole. (A quick rule of thumb is that a reamer 1/2" or less requires a hole 1/64" smaller, while anything over that requires a hole 1/32" smaller than the reamer.)

With the table locked in place, the holes were added to the other three blade blanks in the same way. All were within 0.002" of the required diameter.

The next step was to reduce the thickness of the majority of the blade. The Trantorque requires the full 1/2" thickness but the rest of the blade would be reduced to 1/4." This thickness had worked well with the softer 6061 aluminum used in Maniac Kathy's blade so it should be fine with the much stronger 7075.

I first scribed out the material I wanted to leave around the hub

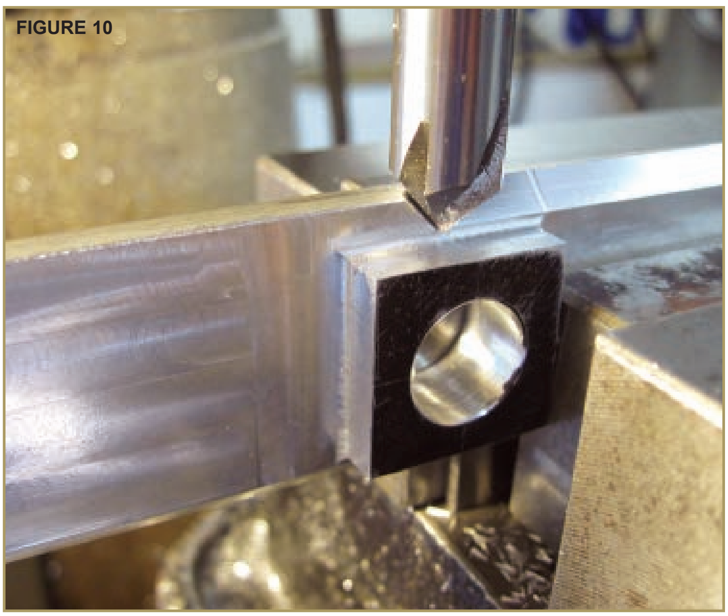


FIGURE 10


FIGURE 11

(**Figure 6**) and then rough-cut 0.2" in a single pass with a big 3/4" end mill from one end of the blade. I then moved the blade in the vise and repeated this on the other end (**Figure 7**). Ideally, one would not move the part but my vise is too small to adequately support the whole cut at once. However, with the knee of the table locked (the Z axis), the part can be moved and the cut depth remains constant. Note I also left about 0.050" around the boss, as well. I repeated this again for the remaining three blades.

The 0.050" finish cut is made with a 1/2" end mill that has a 0.060" radius on the cutter (**Figure 8**). This tool leaves a nice 0.060" chamfer on any internal corner and thus reduces stress risers that might allow a crack later.

The blades are then trimmed to length after carefully measuring and scribing the correct distance from the center hole and machining off the excess (**Figure 9**).

To reduce weight and improve the aerodynamics of the blades, I beveled the edges using a 1/2" 45° chamfer mill (**Figure 10**). I could have tipped the head of my mill to do the same thing, but this method is quicker and easier.

The teeth fit into pockets in the end of the blade and these were carefully measured, scribed, and then machined using the same 0.060" radius mill used earlier (**Figure 11**). The teeth had arrived by this time and after ensuring they fit okay, I machined the pockets on the other blades. Next, the two holes for the screws were added and countersunk (**Figure 12**).

I recommend transfer punching and then drilling one hole first, fitting a tooth, and then using it as a guide for positioning the second hole to ensure that they line up correctly. If you try to just transfer punch and drill both at the same time, it is too easy for the drill to wander, and you'll end up with the holes incorrectly spaced. I didn't fully countersink the holes so I could leave more metal around them which will make the mounting stronger. The tooth mounting screw heads remain slightly proud of the surrounding metal (**Figure 13**). You can see that the teeth also have 0.060" radii on all internal corners and are symmetrical (allowing them to be reversed to give a fresh sharp edge if required).

A completed blade can be seen in **Figure 14**. I need to order slightly longer screws for securing the teeth and a 12-24 tap to clean out a few of the threads them, but the new blades will be ready for their first competition later this summer. Pure Dead Brilliant will hitting harder than ever! **SV**

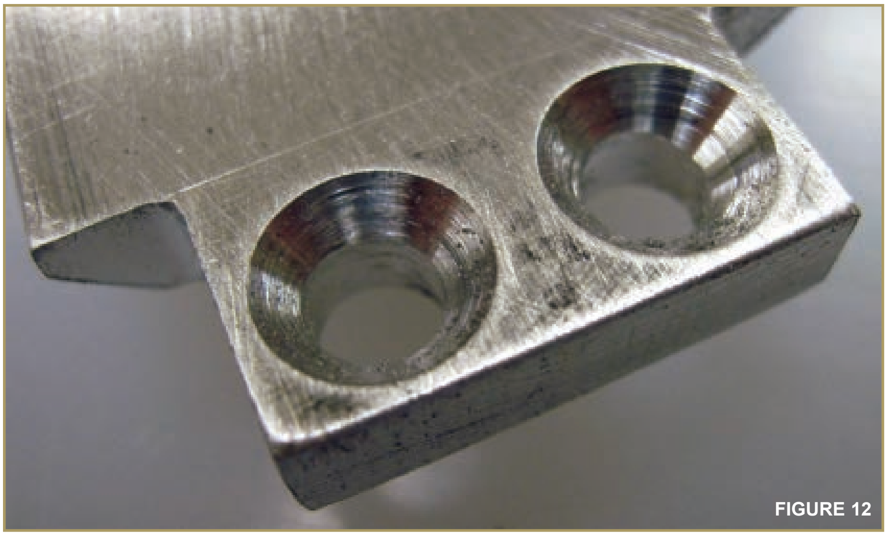


FIGURE 12

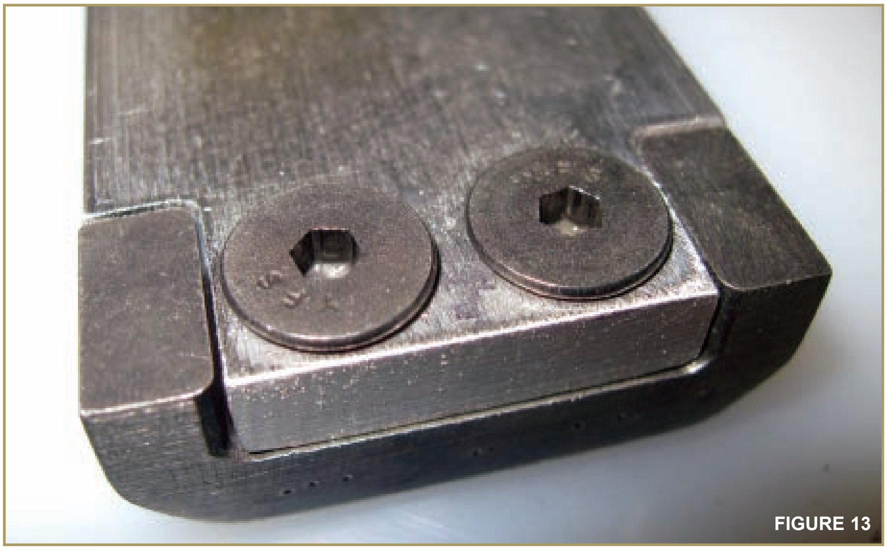


FIGURE 13


FIGURE 14

MANUFACTURING: Modifying NPC Drive Motors for Combat

● by Ray Billings

National Power Chair (or NPC) has been in business for many years, rebuilding and selling electric wheelchair motors and parts to individuals and wheelchair repair facilities throughout the country. This depth of knowledge gives them a lot of experience in dealing with geared electric motor applications — something the combat robotic community has certainly made use of. A quick review of the top performers in any of the heavier weight classes over the years will show that a large percentage of builders rely on these motors for their robotic creations.

NPC (**www.npcrobotics.com**) offers a variety of motors — both geared and ungeared — but the focus of this article is around the most commonly used motor: the T-64. The T-64 has outstanding torque (a single pair of these motors can work as drive in a 340 pound super heavyweight!), is relatively lightweight (I have used these as a drive platform in a 60 pounder), and is versatile — operating anywhere from 12V to 36V. Standard gear reduction on the T-64 is 20:1, giving an output of 230 RPM at 24V. There is also an optional 14:1 gearset (part number NPC-G1410) that can be used for increased speed.

This is a new unmodified NPC T-64 motor. You can see the plastic plate on the end of the brush housing that will need to be removed.

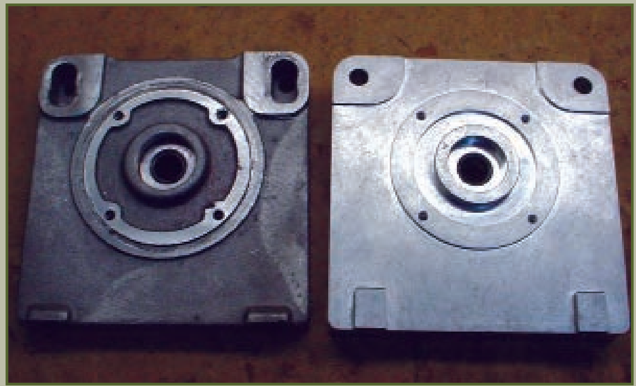
With a wide variety of potential applications and its simple "bolt and go" construction, it is no wonder that the motors have such appeal. However, this is still a motor that was originally designed for a non-combat application. There are a few modifications that can be done to make a great motor even better.

The motor (as originally designed for wheelchairs), has a brake assembly built into the brush housing. This makes sense in a wheelchair but not in a combat robot, so NPC sells these motors

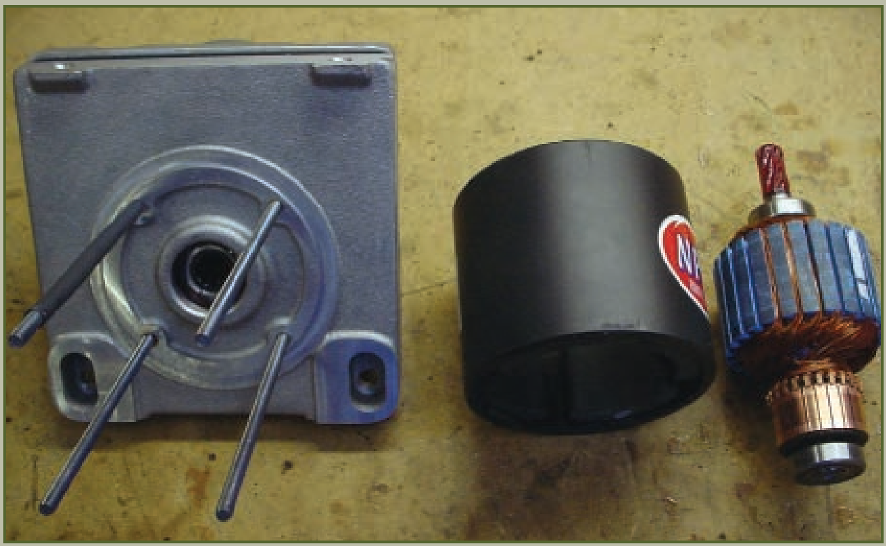
Remove the brushes and these four small nuts, and remove the brush housing from the motor assembly.

Here are two brush housings: one original and one modified for combat. The size difference between the two is fairly substantial.

Here we see an original motor, and a stage 1 modified motor that has been assembled using the coupling nuts.



The cast aluminum mounting plate on the left is usable for combat, but the machined 6061 plate on the right is substantially stronger.



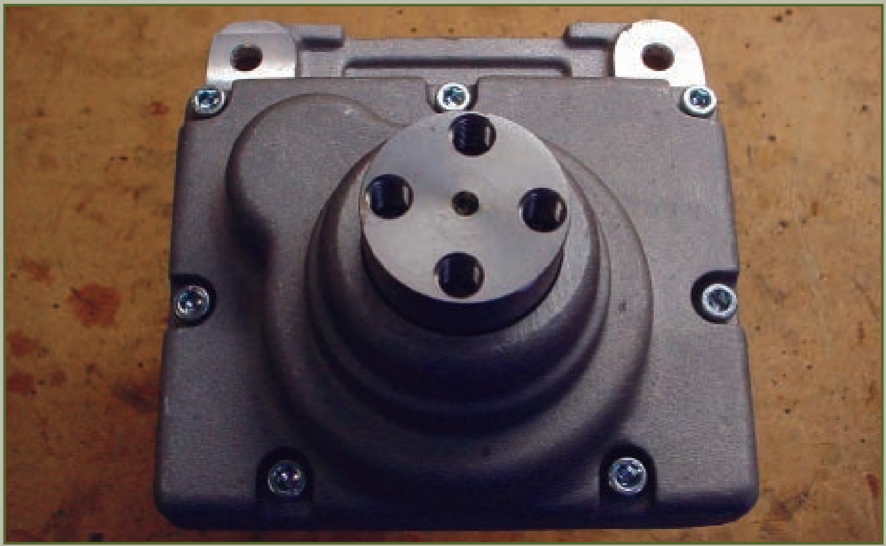
To assemble a stage 2 setup, you will need to disassemble the motor completely and reassemble using the heavier duty mounting plate. You can see the protective sleeve over one of the studs that is used to protect the wire leads from chaffing on the stud.

without the brake assembly installed. The brush housing, however, is still designed for this brake assembly, making it physically larger than is needed for combat. So, the first action needed is to shorten this housing.

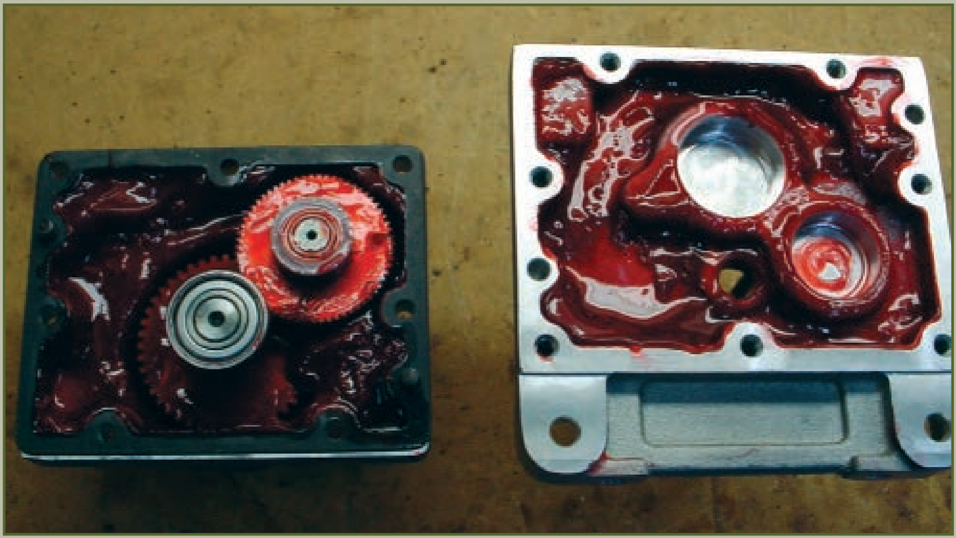
The brush housing has a plastic cover over the end — some older versions had a metal cover. In either case, this cover needs to be removed and discarded. Remove the four brushes from the housing, and then remove the four small nuts holding the brush housing to the back of the motor. The brush housing should be able to be pulled straight off the back of the motor at this time. Be careful not to pull the armature and magnet can from the mounting plate.

As you can see in the **photos**, there is about an inch of brush housing that is not needed. You can cut this off in several different ways. I have used band saws, jigsaws, a hacksaw, and even a chop saw once. By far, the best method has been with a lathe. Whatever you do, make sure you do not cut into any of the machined area in the bottom of the cavity, and try to make the edge of the cut as smooth as possible. This is just a cast piece and it can crack under strain, so it is best not to give any potential stress risers.

I usually leave about an 1/8" of an inch overhang when done. This is enough to add some strength to the edge of the brush housing, but short enough to make mounting easier in the robot. Make sure all debris has been removed from



This is the output shaft of the motor. Removing the seven socket head screws will allow you to remove the gear assembly from the mounting plate.



This is the output shaft of the motor. Removing the seven socket head screws will allow you to remove the gear assembly from the mounting plate.

inside the brush housing, and then reassemble.

The orientation of the brush housing is not mandatory — if you need the wire leads to come out at any specific angle, just reassemble the motor with the leads pointing in the direction you would like. The only thing to watch for here is the protective sleeve over one of the studs. This sleeve needs to be on the stud that corresponds to where the leads enter the motor, since this sleeve protects the wires from chaffing on the stud.

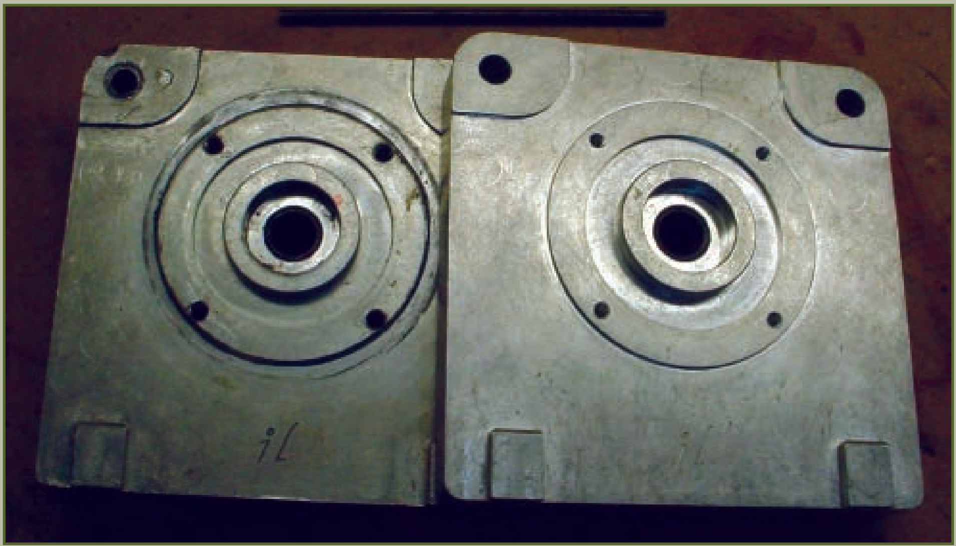
One source of problems with this motor design is the smallish studs that hold the motor assembly to the gearbox. In some older models, these were only 8-32 studs. Current design has been improved to 10-32, but it is still quite easy for the four small nuts on these studs to strip under the big hits seen in a combat robot. For this reason, I replace the four small nuts with longer coupling nuts (like part number 90264A430 from McMaster). This gives a much longer area of engagement on the stud, and has almost completely removed the potential of stripped nuts here.

I would call this a stage 1 modification (the various stage numbers I use here are something I came up with to keep track of the modifications I make, and are not anything official from NPC) and would consider this highly recommended for any combat robotic use. This is the lightest setup that can be used at ~12.4 pounds, and is the current arrangement I use in our middleweight robot, The Mortician.

Stage 2 modification would require some physical strengthening of the cast parts of the gearbox. The mounting plate for this motor is a cast aluminum piece, and although it is strong enough for combat use, replacing this part with a machined billet part is far stronger. NPC sells an upgrade mounting plate machined from 6061 specifically made for this purpose (part number NPC-64GBPT). Inertia Labs (**www.inertialabs.com**) also sells an aftermarket plate for these motors, similarly machined from 6061. The machined plates shown here are all Inertia Labs versions. All the mounting holes are identical to the cast piece, so it is really just a matter of disassembling the motor and reassembling with the new mounting plate. The new mounting plate is heavier than the original cast piece, so be aware of this when planning your weight budget.

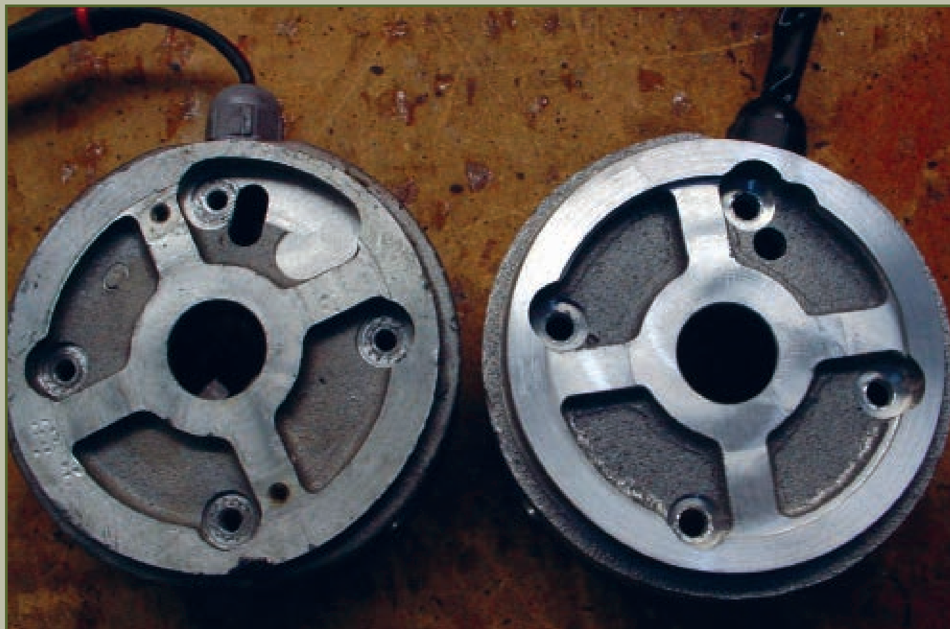
Weight of a stage 2 T-64 is ~13.5 pounds.

Stage 3? You bet! Under really extreme shock loading (anyone who has ever seen my heavyweight Last Rites fight knows what extremes I am talking about), it is possible to stretch the studs that hold the back of the motor together, thereby loosening the whole



For stage 3 modification, we increase the mounting hardware from the 10-32 studs to 1/4-20 cap screws. The plate on the right has the original 10-32 tapped holes for the studs, while the left plate has been drilled and tapped for 1/4-20. You can see the size difference in the holes, and the fasteners above.

Increasing the screw size means the hole in the brush housing that the screws pass through also needs to increase. Be careful drilling the hole where the leads come through, as it is easy to damage the wires.



assembly. One of the assets of the T-64 motor is that it is already geared down to the appropriate output speed in a fairly compact form factor.

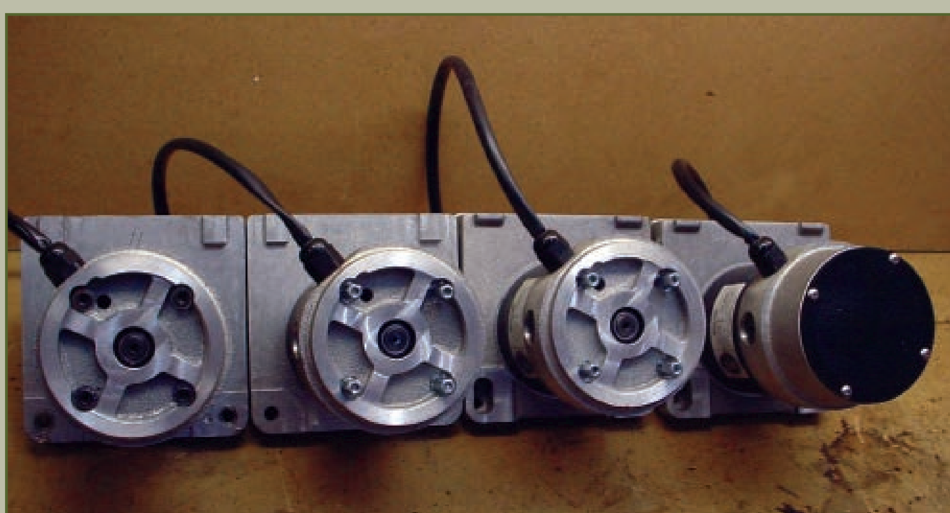
However, to accomplish this compactness, the motor is designed with the output shaft of the armature mating directly to the first stage gear. Any change to the orientation of the armature in relation to this first stage gear can damage the armature and gearset. So, in order to ensure that this assembly stays intact and tight, I machine the mounting plate to accept a 1/4 20 thread, and use a set of graded cap screws (McMaster part number 91251A090) to assemble the motor. The larger and higher quality fasteners resist stretching, and the durability has been greatly improved. There is virtually no weight change from stage 2 to stage 3.

For combat use (with whatever stage modification you have on your motors), you should frequently check the motor assembly for tightness on all the fasteners. I check mine after every match. This may seem extreme, but should be considered really for every motor from any manufacturer. Some of this is due to vibration, and some due to stretching, but it can be astonishing how much things can loosen in just one match.

A final modification that is cheap and easy is to wrap electrical tape around the outside of the brush housing to confine the brushes. It is extremely rare to actually see the brushes pop out (I have only seen this a few times, and never on an NPC), but it is certainly possible and, of course, the fix is really simple.

NPC sells a variety of hubs and wheels that bolt directly to the output shaft of these motors. The standard hub from NPC (part number NPC-PH448) has the bolt pattern used for most go-cart and utility rims, greatly increasing the available tire types and rim styles you can choose from. The wheels that NPC sells are foam filled rubber and get fantastic traction, but if you are looking for a lighter weight option you can use the carefree style tires available from The Robot Marketplace (**www.robotcombat.com/products/wheels_main.html**).

So, there you have it — all the modifications that Team Hardcore uses on the NPC T-64 drive motors. These motors have proven themselves to be extremely versatile, and have plenty of power in a fairly compact design. They can also make your design work much simpler, since a complete drive system can be purchased from "off-the-shelf" components. If you have any questions, feel free to contact me at ray@hardcorerobotics.com. **SV**



Here we see all the versions. Right to left we have an unmodified new motor, stage 1 with shortened brush housing and coupling nuts; stage 2 with machined 6061 mounting plate; and stage 3 with larger cap screw fasteners.

PARTS IS PARTS: Bantam E-Station BC8HP Charger Review

● by Nick Martin

The BC8HP charger is an update to Bantam's already popular BC8 model. In short, Bantam has kept all the good features, increased the power rating, and improved the ergonomics. The main feature is in the "HP" part of the name — a power increase from 150 to 280 watts and a maximum charge current of 12 amps. This will definitely interest builders in the larger weight classes or users of the latest 5C charge rate Lipo packs.

Compared to the Original BC8

The new charger has the same dimensions of the BC6 and BC8 models. The backlit LCD screen is virtually identical, but looks smaller due to the garish yellow front panel (see **Photo 1**). The buttons are now sealed and labeled better than earlier models.

The cabling is vastly improved over previous models; the charging and balancing outputs are now both on the righthand side, which is far more convenient. The connectors are labeled better than before, but could still be improved.

In the Box

The BC8HP package comes with the charger, software CD, USB adapter, USB cable, battery cable, and manual. Some sites list the USB adaptor as an extra, so check when ordering. The manual is well written and jinglish free; most users will be up and running in 10 minutes. The box is also supposed to include a LiPo balancing adaptor; some shops include a standard model with Thunderpower connectors while others let you pick your own model (see **Photo 2**).

Capabilities

The BC8HP will charge almost any battery chemistry: up to eight Lithium Polymer, Lithium-Ion, or A123 cells; up to 27 NiMH or NiCD cells; and lead acid batteries up to 36V. Don't underestimate this last feature! Fast charging a dead car battery has gotten me out of trouble several times!

Keep in mind that the maximum rating of 280 watts and Ohm's law limit your maximum charge current at higher voltages; after about 22 volts, the charger will not be able to supply the full 12 amps. To get an idea of your top charging current, divide 280 by your pack voltage. I found that in practice, the current never quite reaches this value, so take off about 10%.

The LiPo balancing works very well. After a charge, my packs are usually balanced to below the resolution of my multimeter, and you can monitor individual cell voltages on the display while charging.

Experts and tinkerers can use the included software and USB adaptor to analyze their packs. The PC program graphs the voltage, current, capacity, and temperature curves. It also shows the individual voltage of each cell in the Lithium battery pack.

The USB connection uses an annoying and somewhat delicate adaptor; take care not to break the shrink-wrap coating!

PHOTO 1. The charger front panel.

The LiPo balancing adaptors come in many configurations and should cover virtually any brand of battery on the market. Bantam also makes multi pack adaptors that allow you to charge two or three smaller packs in series which makes the charger very versatile.

Operating the Charger

Power supply: The charger will accept from 11 to 18 volts, and it is suggested that you use a power supply capable of at least 15 volts and 35 amps to avoid thermal shutdown or slower than expected charging. The fan will kick in even at low charging currents and the heatsink can get quite warm, so be sure to allow plenty of ventilation.

PHOTO 2. The charger and its main accessories.

Closest Competitors

In this price range, the Triton 2EQ and the Thunder Power TP1010C have similar features and are worth looking at. The Trition charger's main selling point is a built-in AC power supply; it can charge and balance up to 6S Lithium packs and up to eight amps. It has a power rating of 120 watts. Its interface is possibly the best in the business and it is a great choice for builders wanting a single all-in-one charger for smaller bots.

The TP1010C charger's best feature is the high cell count; it can charge up to 10 Lithium cells and up to 10 amps with a total capacity of 220 watts. The interface is similar to the Bantam units and this charger is only let down by the absence of internal balancing. This would be a leading choice for builders still using NiMH or Nicad batteries.

Summary

This is a well made and capable charger for larger packs. The quality is somewhere between cheap Hobby King chargers and the expensive European models; my other Bantam chargers have lasted for years and this new model should too.

Expect to pay around $195 for the basic charger. Balancing adaptors are $12 to $15, depending on the model. **SV**

Sources

- RC Accessory
  www.rcaccessory.com/BC8HP.aspx
  They are direct importers and service the Bantam chargers.

- The Robot Marketplace
  www.robotcombat.com/products/0-BAN-BC8HP.html

EVENTS

Completed Events

Completed Events: May 10 - June 1

Maker Faire Bot Gauntlet was presented by California Insect Bots in San Mateo, CA, on May 22nd and 23rd.

Maker Faire Bot Gauntlet

HORD Spring 2010 presented by Ohio Robot Club in Brecksville, OH, on May 15th. **SV**

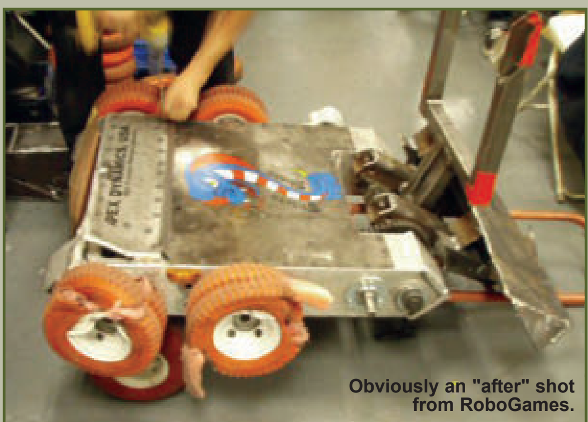
O.R.C. Ohio Robot Club

COMBAT ZONE'S GREATEST HITS

● by Kevin Berry


Last Rites and Sewer Snake at ComBots 2009.


"With your shield, or on it" — Sewer Snake pays the usual price for exposed wheels.


Obviously an "after" shot from RoboGames.

Ray Billings sent in this great selection of damage shots. All of these are results of fights against his HW Last Rites. He says: "The Sewer Snake fight at ComBots shows a good hit in a string of big hits that eventually disabled him. I managed to pull off the wedge and all but one of the tires before Matt finally tapped out."

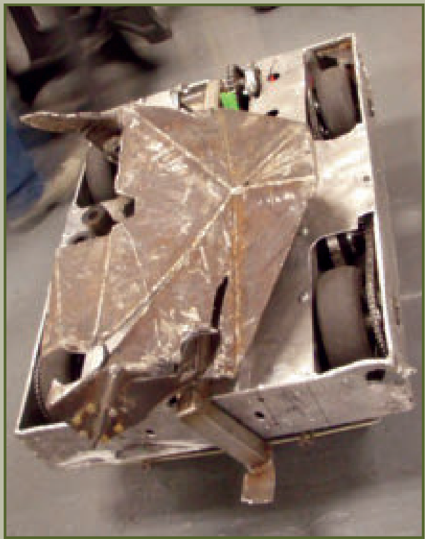
At RoboGames, the fight against Sewer Snake was just as epic, although this time Matt held on for the win even with all of his damage.

The fights against SJ and Original Sin at RoboGames were both brutal, with Last Rites winning each of those.

The ComBots photos were taken by Michael "Fuzzy" Mauldin, and the RoboGames


More Original Sin before meeting up with Last Rites at RoboGames 2010.

Original Sin before meeting up with Last Rites at RoboGames 2010.

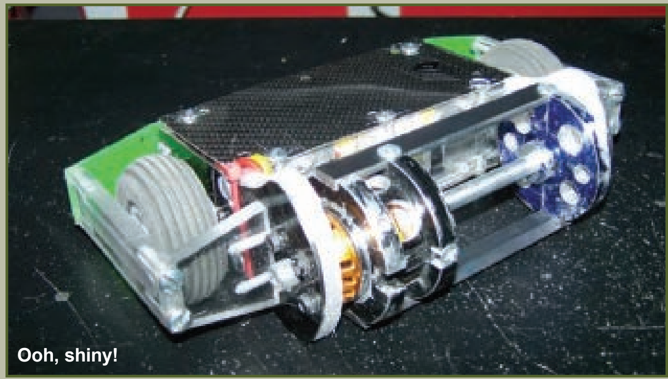


Wedges only work when they stay attached! SJ absorbed some big hits.

photos are by Ray Billings.

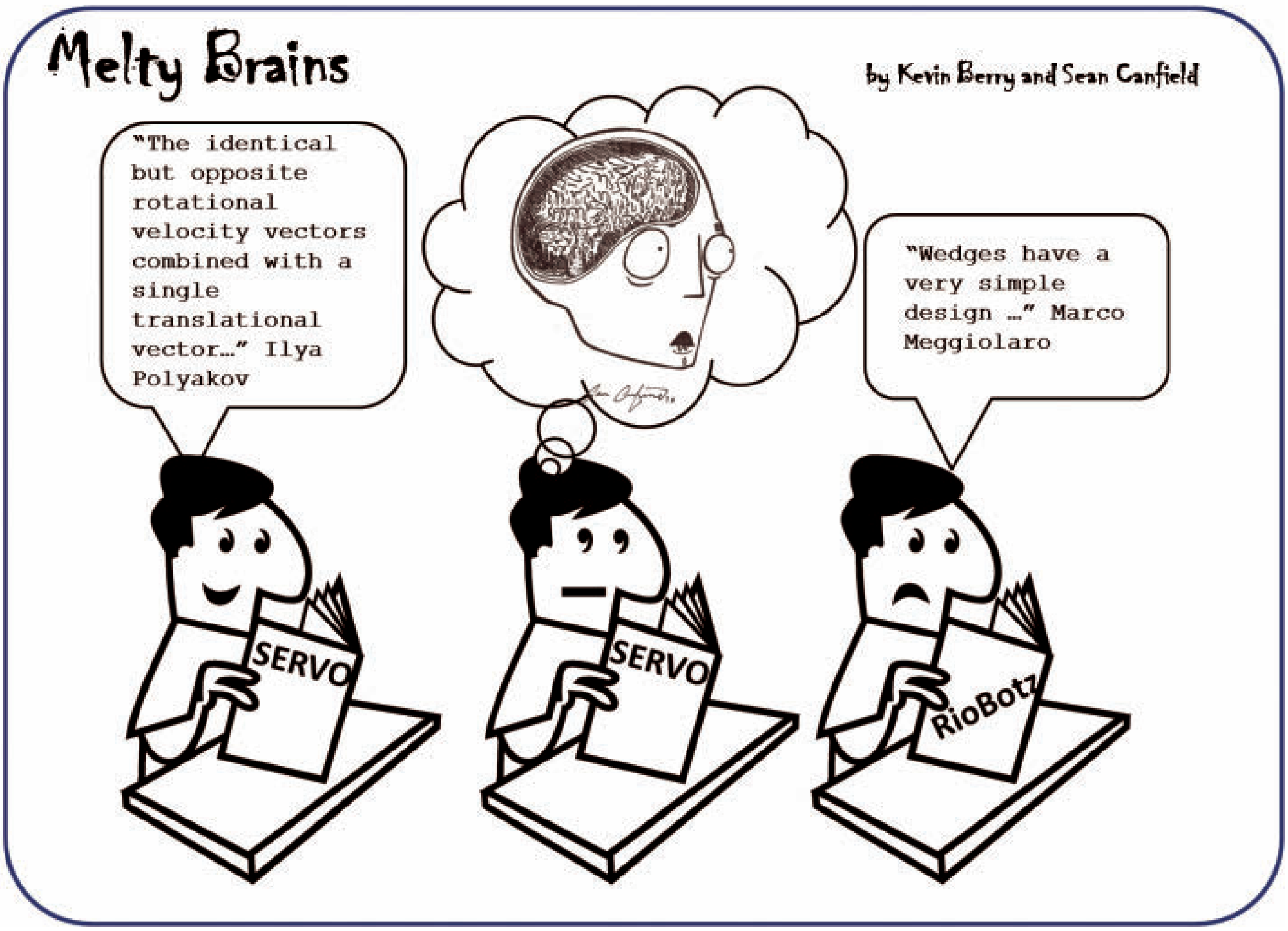
At Pa Bot Blast 2009, Kyle Singer's bot Fangus 2.1 faced Maelstrom in both the semi-finals and finals.

Before and after photos, brief description of the fight, and builder's name can be submitted to me at LegendaryRobotics@gmail.com. Or, if you have an action shot that clearly shows what's going on, those are welcome too! These don't have to be current; anything you can (legally) submit, clear back to the good old days of wooden bots and iron builders is fair game. SV



Ooh, shiny!

Somehow, it doesn't hurt so badly when you win!



The Evolution of Ethel

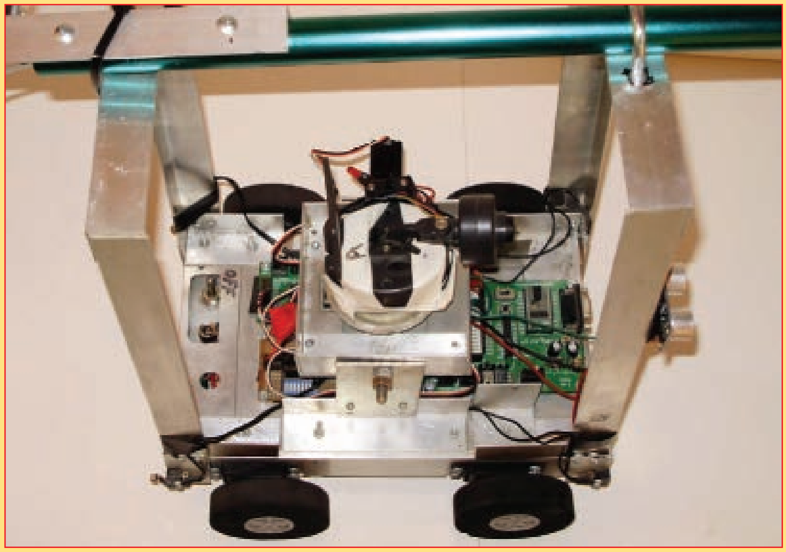
By Steven Nelson

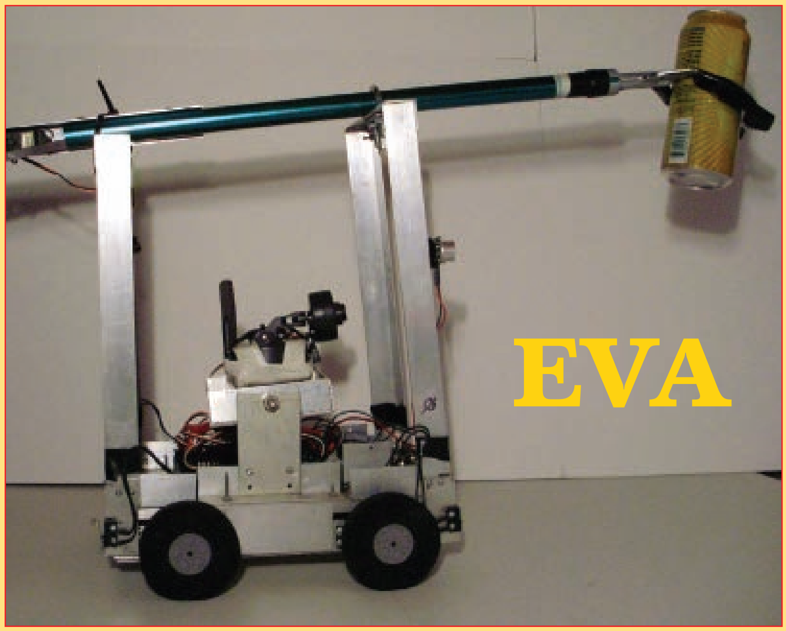
I have been experimenting with kit robots like the Parallax BoeBot and the Toddler and various infrared, ultrasonic, and vision sensors since 2004. All of these great teaching machines got me back into programming and taught me about interfacing microcontrollers with electronics and software. Once I understood how to use these components, I built a slightly larger 6 lb 4WD radio/BASIC Stamp II controlled robot of my own design called EVA. EVA allowed me to experiment with both autonomous and radio control, plus a mixture of both. EVA also allowed me to add a wireless video camera to a mobile machine and perform tasks like getting a beverage from the refrigerator. EVA used her claw to open the door and grab a can while I was sitting on the couch watching and controlling the action on TV. EVA also showed me that I needed to build a bigger machine to do larger tasks since anything weighing much over 1 lb was too heavy for her to pick up and carry reliably. I really needed a more powerful friend ... enter Ethel.

Ethel is Born

It's amazing the collection of parts you compile after building combat robots for several years. Of course for some of us, it is difficult to part with a part because you might find a new use for it one day. After four years of waiting, in October '08 I went to the Robo-Development convention and was given a VIA Nano-ITX 1000G microcomputer as a door prize. A month later, I pulled out a large pile of pillow block bearings, one inch diameter axle shafts, wheels, and sprockets, and laid them out on the welding table. In this collection of very used hardware was the essence of 12 previous 220-340 lb combat machines my father and I had built and the seeds for a new machine called Ethel were sown.

Ethel's drivetrain is mounted to a welded steel frame made from one inch square steel tubing and 1.5 inch angle iron that is used for the 12 pillow block bearing mounting surfaces. The overall frame is 22 inches long and 20 inches wide. Ethel is laid out using a 4WD skid steer configuration with two Bosch GPA 750 watt motors (one for each side). Each side consists of a motor that drives a jack shaft with a 4.8-to-1 chain reduction. The jack shaft then drives the two wheels using another 4.8-to-1 chain reduction. This gives Ethel a 23-to-1 total gear reduction and about 368 ft lbs of total available torque. I call this chain layout a bow tie configuration since it resembles the shape of one when you look at it from the side.

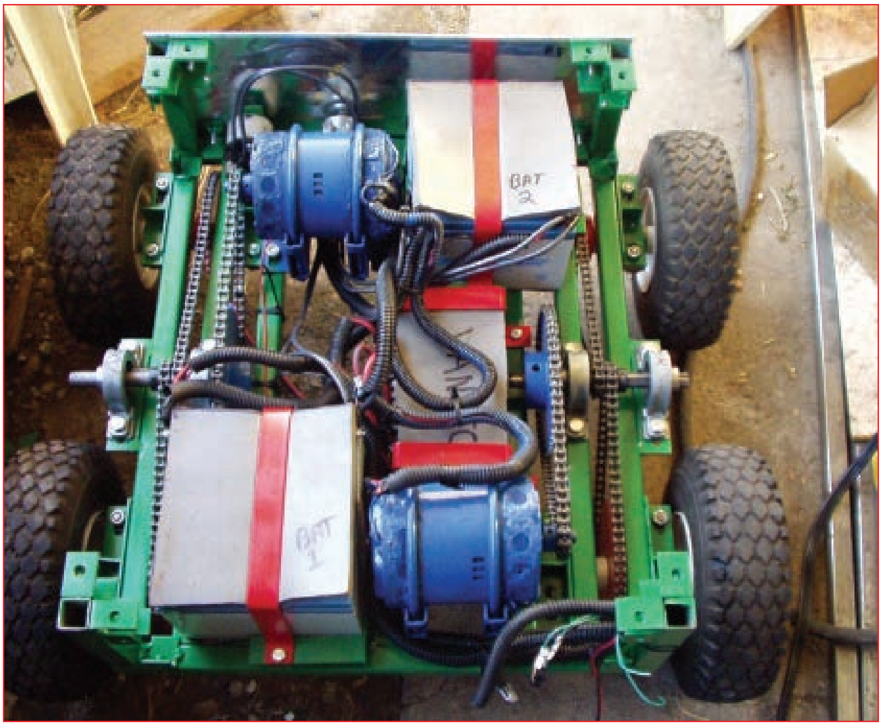




With her 10.5 inch diameter studded pneumatic tires, she can carry about 814 lbs. Ethel is capable of about 4.6 mph. I figure that's a pretty safe maximum speed for her to use around humans. Ethel has 6.5 inches of ground clearance. Her tires are overhung outside of the frame and also extend past the frame front and rear for maximum obstacle climbing abilities. She can easily climb over a four inch tall board.

Ethel is built for off-road use and rough surfaces. Her drivetrain motors are powered by two 26 Ahr Powersonic sealed lead acid (SLA) batteries wired in series to provide 24 VDC to a Vantec RDFR38E dual motor speed controller. I had the chassis rolling and running under standard RC (radio control) in about four days. Basically, Ethel was just another proven super heavyweight (340 lb) combat robot drivetrain design at this point. However, I intended for her to play nice. Now it was time to build her computer system, control electronics, and write the software to tame her down a bit.

I had become aware of the VIA Pico microcomputer boards from an article in the September '08 *SERVO Magazine* about a Johnny 5 robot and I was quite happy to finally have a similar board of my own to play with. The 1000G Nano-ITX board Ethel uses is pretty small. It measures 4.7 x 4.7 inches — about the size of a DVD — and contains a 1 GHz Luke processor and supports 1 GB of RAM. The board is capable of 1024 x 768 VGA graphics plus stereo sound, and has a microphone input and aux audio out. The board also supports six USB jacks (if you purchase extra cables). I found a good source for VIA parts and accessories at an online company called Logic Supply and purchased a 160 watt ATX computer power supply that runs off of 12 VDC for automotive use. I also added a Western Digital 160 GB IDE hard drive and a Sony DVD burner that I purchased locally at Fry's Electronics.



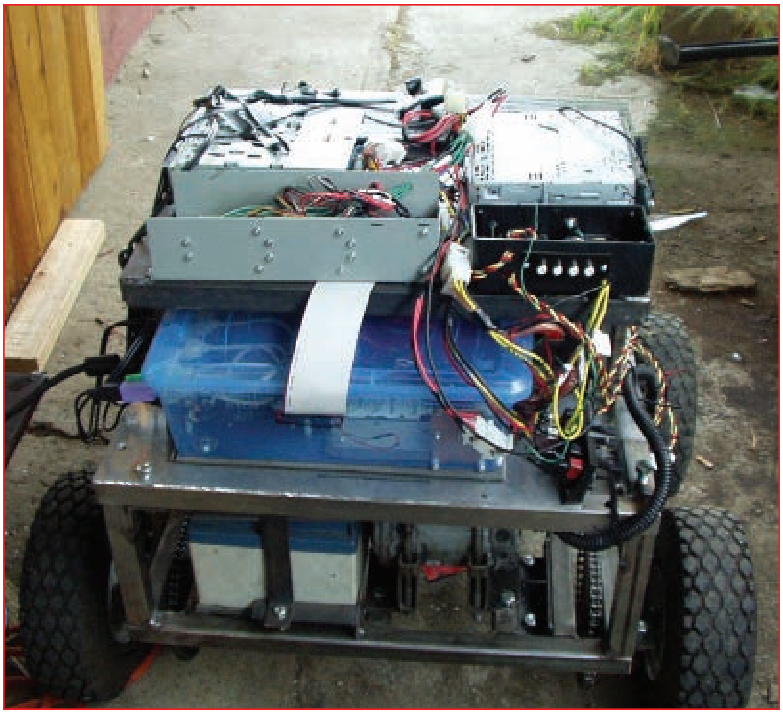
How It All Works

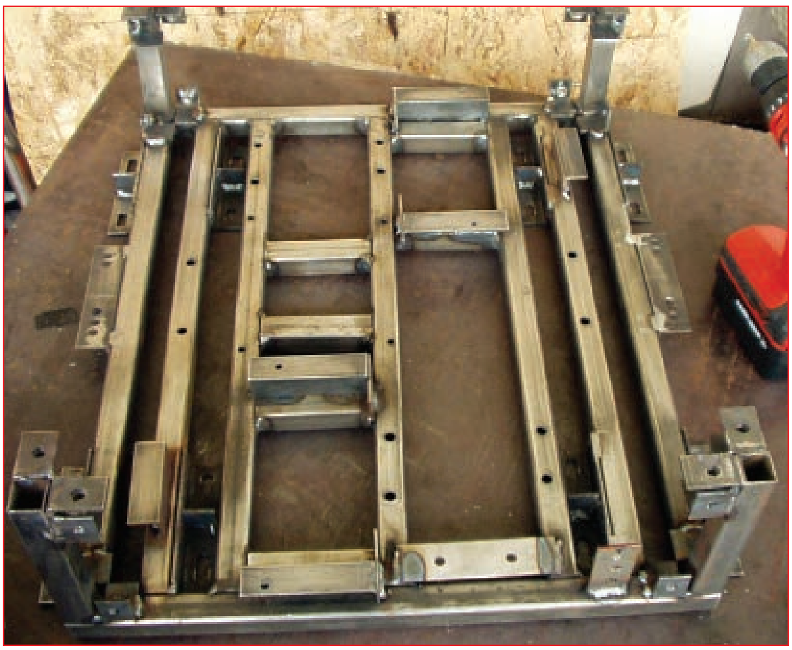
For basic wireless joystick control, Windows first reads the joystick, then that data is sent to RoboRealm. This software runs in an endless loop or pipe to constantly read a USB camera and process video. This makes it possible to add different built-in software modules inside of this pipe to perform different functions. For basic teleoperations in Ethel, I added the joystick module, a VBscript module, then a display variable module "for debugging," and finally an SCC-32 board module.

The joystick module reads the x and y position of the Logitech wireless joystick and the condition of all 10 of its switches. This data is stored in 12 variables that are passed to the VBscript module where the custom code I wrote makes decisions on what actions to take based on joystick moves. Turning is controlled by reading the "X" right or left position of the joystick. The values are sent to a math formula to provide mixing and proportional speed control. These values are passed using a USB-to-serial connection to the actual SCC-32 servo control board and its Atmel 128 processor continually converts this data to PWM signals which are sent to the motor controller which, in turn, feeds highly excited electrons to the dual 750 watt Bosch motors. It is almost too simple.

However, I found that getting the speed and acceleration mix right was actually a bit painful. Originally, Ethel would accelerate instantly to full speed in one body length. When I first tested her, she was really quick and too touchy to drive safely around humans. I actually ran myself down and almost broke my leg. After that incident, I had to de-tune her software a bit.

All of the other digital joystick button functions (that are on/off controlled) work in a similar manner depending on what is being controlled. My VBscript code just sends a full on (servo full clockwise), full off (servo center), or proportional PWM (servo) commands with a serial data link to the SCC-32 board, then PWM position commands to





the respective controllers. For basic on/off control, I use Team Delta RCE 220 dual-ended switches since they provide proven robotic combat fail-safe functions. Basically, they always default to off (or servo center commands) in case of a power or control signal loss.

Ethel currently uses two of these to control her arms in an H-bridge mode for motor forward/backwards with the optional position-limiting switch function enabled for up/down control. There are two more of them to control the solenoid valves for her fire effects, plus one more to control video switching for her multiple video camera feeds.

Ethel's frame was designed to be modular, and stack vertically like a wedding cake. Her base contains the drivetrain motors, batteries, and speed control. Her next layer contains the VIA computer and two other 26 Ahr batteries that power the computer, the Team Delta controls, the 250 watt AM/FM/CD Sony stereo and speakers, plus the DVD player and her custom electronics. Ethel also has a custom-built 4 Ahr 6 VDC NiCad battery pack for her various servos, as well as several independent power supplies to feed her various devices.

Ethel's Poofers

Having been involved with Fire Art and the Flaming Lotus Girls, I decided Ethel needed a set of fire effects. I mounted two propane vapor effects — known as poofers — on another frame layer above her electronics. Each poofer consists of a 2.5 gallon propane cylinder that acts as a gas vapor accumulator and a 12 VDC electronic pilot-operated gas valve. A separate fuel line feeds a continuous burner flame on the end of each arm for ignition. The accumulator vapor is fed through a 1/2 inch propane hose to a 3/4 inch pipe nozzle mounted on each of Ethel's arms. The gas is released and ignites when it passes the burner's open flame. The whole system is fed from a five gallon cylinder of propane fuel. Each arm can rotate 270 degrees using windshield wiper motors and a 6-to-1 reduction chain drive. Each motor is controlled by an RCE 220 switch with position limiting switches installed and a hard stop mechanical fail-safe device with an electronic fuse shutdown for total safety. The dual flames can both reach 30 feet in height and six feet in diameter.

Going Soft

Now it was time to load some software ... so I thought. I hit the power switch and applied 12 VDC to the ATX power supply ... nothing happened. After two hours of reading manuals, probing with a multimeter, and searching the Web for an answer, I found a pin on the VIA board that triggers the ATX power supply's start-up when you apply ground to it. This wasn't clearly mentioned in the online manual. I added a switch to trigger that pin and the computer started to boot up.

After that bit of builder-enhanced drama, the first program I loaded into the VIA was Windows XP Pro. Next, I downloaded and installed what would become the core programming environment for Ethel — a vision processing and robot control system called RoboRealm. I had been experimenting with RoboRealm on my desktop computer (with a web cam) and on my small robot EVA for a couple of months and was dying to try it on a big robot. Once RoboRealm was running and seeing the web cam, I did a stress test to see what the little computer was capable of. I added i-Tunes and started playing songs and videos while running RoboRealm; the computer didn't seem to notice. So, a parts buying spree was in order. Ethel soon received a Logitech wireless keyboard and mouse, a Logitech wireless joystick, and a D-link WiFi card. Even after all that, the software drivers were installed and everything was still playing nice.

With the VIA computer up and running, I built an enclosure for it using a blue Tupperware container (since all good robot computers are contained inside of blue Tupperware). I mounted the logic board, the power supply, and double shock-mounted the 160 GB hard drive using foam rubber and some angled aluminum brackets. With all of the computer's boards safely mounted and wired, I was ready to start on the output electronics. One thing that becomes obvious when you build robots with small microcontrollers is you eventually run out of processor speed for reading sensors and driving motors.

Earlier experiments with EVA had taught me this lesson, and I did find a solution for her. I added a Parallax servo control board (PSC) to remove the burden of constantly updating motor positions with the Stamp. Once I did that for EVA, the BASIC Stamp II was free do more important things like reading sensors or other control inputs like commands from an RC receiver. Since this proven strategy worked so well in EVA, I also used it in Ethel's more powerful computer system. I figured the VIA computer had enough to do running Window's XP Pro, RoboRealm, iTunes, and sometimes a web browser plus Skype.

Ethel uses a Lynxmotion SCC-32 servo control board to send PWM signals to all of her control electronics and motor speed controllers, and even the servos that pan and tilt her cameras and operate her other mechanical toys. The signals from the SSC-32 are compatible with all of the usual RC electronics I have tested so far in my combat robots, and the interface board has performed perfectly.

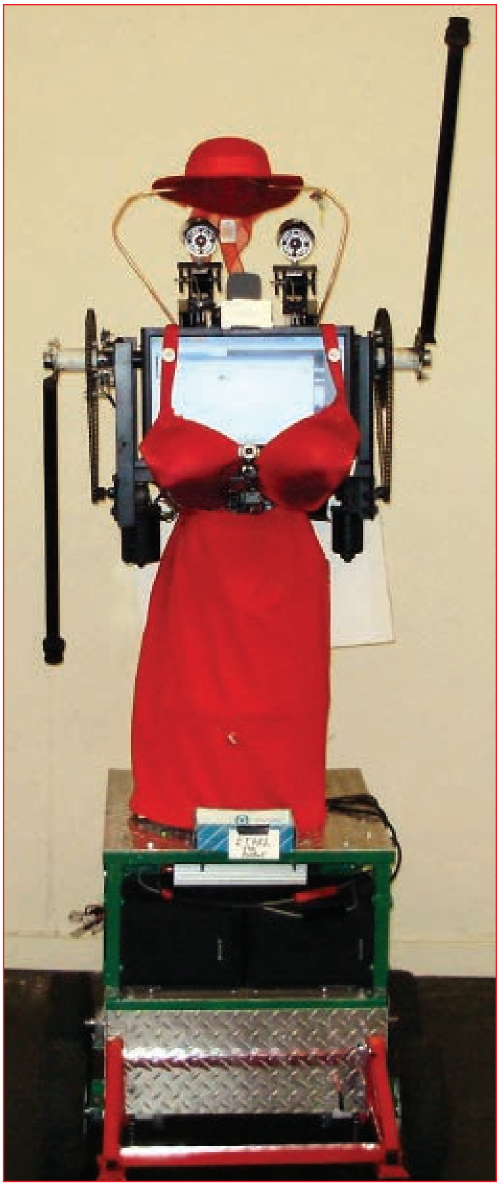
Running With the Humans

Ethel was really only tested for about 10 minutes before I had to leave for the '09 Maker Faire in San Mateo, CA. When we arrived, her flame units were not totally plumbed and some of the wiring to control them was not finished. She had a pending inspection with the Fire Marshal, so fortunately we got her parts that day and assembled it that night, and passed the fire safety inspection the next morning. Ethel was assigned a fenced off area where she was allowed to demonstrate her warm and cozy nature. After the flame demos, her propane fuel cylinder was removed and she was allowed to roam around the event for the next two days.

I operated Ethel as an ROV (remote operated vehicle) and I will say driving a 300 lb machine around 80,000 humans packed into a venue can be a challenge. People are totally unpredictable and very hard not to run over because of their actions around the machine. Thankfully, all went well and Ethel had a great time interacting with everyone. I also discovered a hidden talent Ethel possessed: She can dance. If I control her just right by putting her in a skid steer turn and slowly advance the throttle on a hard surface, the skidding of her high traction tires can get her hopping off the ground. She can get up to three tires off the ground.

After that Maker Faire, I took Ethel to RoboGames 2009 and tested her video tele-presence. Ethel has two video cameras (one in front and one in back). The video signal was sent through a custom switch (basically a DPDT relay) that could be operated by a button on her joystick. When the relay is off, the front camera video is sent to her 5.8 GHz video transmitter. When the joystick button is pressed, the rear camera video is sent to the transmitter. This worked out okay for most driving tasks. I made my first attempt at having Ethel retrieve a beer. Watching only the wireless video feed and using Ethel's right arm with a cup holder, I purchased an open cup of beer from the snack bar. Although she spilled it three times on the trip back (mostly because of stopping suddenly for people who stepped into the camera view unexpectedly), she did manage to get about half of a cup full back to my control station.

The next day, Ethel went on another beverage run and purchased an unopened can of beer. This test run was much more successful (although there were a few tense moments when I lost sight of a small child that was very close to the robot). The most important lesson I learned from this real world testing is the need for multiple cameras and pan and tilt mounts. You can never have too much video information when tele-operating around people.



Ethel Gets a Makeover in 2010

In her original configuration, Ethel was working very well mechanically, so that hasn't changed much. Since she would be going to RoboGames again, I removed the flame units and decided to concentrate on tele-presence and video. I added an 800 watt Coleman power inverter to provide 120 VAC to a 15.7 inch Acer video monitor attached to her upper chest. I had noticed some shortcomings when using single camera video tele-presence, so to address this, two Servo City pan and tilt camera mounts and two Swan Infrared enhanced security cameras were added. These faced forward (similar to a human's eye placement) and another IR enhanced security camera was moved to the rear. This was a bit of a problem since Ethel only had one video transmitter, so I added a Quad video processor to combine multiple camera images into one video feed that could be transmitted and displayed on a single monitor in various split screen modes.

Since the forward cameras can both pan and tilt, they can either be synched to move together like human eyes or they can move independently in any direction. (I like to call this "iguana vision.") This mode really helps to improve her peripheral vision and remove those pesky blind spots that were originally experienced at RoboGames.

Ethel also received a servo mounted USB web cam and microphone on her chest. This way, I can tilt the USB camera up and down to adjust to the height of people she meets. By using Skype, Ethel has audio/video tele-presence to anywhere in the world (as long as she is near a WiFi connection). While running RoboRealm, the video from the USB camera can be displayed live on her monitor so people can see themselves staring back from Ethel's chest. Most everybody — especially children — seemed to really like this camera mode. In fact, it can make driving forward very difficult because they just won't move away!

Probably the smartest thing I did so far was to add a bubble gun to the rear of Ethel. With this simple toy, Ethel can blow a stream of soap bubbles and the kids almost always move to the rear of the robot to play with them. This clears the path in front of her USB camera so she can keep moving forward.

Of course, no makeover for a woman would be complete without a new wardrobe. So, I went shopping at the local Goodwill store and found a nice red skirt and lovely red hat. The only thing that was missing was a top. I found some large soup ladles and bras at the local fleamarket. The "helpful" vendor told me "Well, if they don't fit you can bring them back."

I built a frame for the dress out of a steel wire tomato planter hoop. The soup ladles were mounted just under Ethel's monitor to fill out her bra cups, and off we went to RoboGames 2010. Nora Judd helped me with the hat. To my surprise, the new outfit (which started off as more of a joke) was a big hit. The bubbles kept both kids and adults happy.

Ethel even won a Silver medal for Best in Show in the ArtBot category.

More importantly, I was able to test the tele-presence with the multiple cameras and found the increased viewing area to be a marked improvement in operation and public safety.

After RoboGames 2010, Ethel got a new squeeze bulb-type bicycle squeaker horn which is activated by a servo. I also added a 110 db Ooo-Gaah horn which is activated by a Team Delta switch and a 30 amp relay. I took Ethel next to Maker Faire 2010 where she was a big hit playing her music, waving her arms, blowing bubbles, and, of course, dancing with the kids.

Vision for Ethel's Future

After about a year of testing, I believe that I understand Ethel's capabilities well enough to take her to the next level by using RoboRealm's vision processing modules and addimg some more ultrasonic and infrared sensors to let her begin to roam free. My hope is to someday get her to take out the trash.

Operating a large machine in public does have its risks, but that's really how you learn where the problems and/or weaknesses are. Of course, after 12 years of surviving robotic combat, power tool drag racing, fire art, and large robot building, I feel prepared and experienced. However, if you see me start to run, you should really try and pass me. **SV**

Parts and Software Suppliers

Grainger Industrial Supply
1-888-361-8649
www.grainger.com/Grainger/wwg/start.shtml

MSC Industrial Supply
1-800-645-7270
www1.mscdirect.com/industrial-equipment.html

Northern Tool
1-800-221-0516
www.northerntool.com

Harbor Frieght
1-800-444-3353
www.harborfreight.com/

The Robot Marketplace
877-762-6899
www.robotmarketplace.com/store.html

Lynxmotion
1-816-512-1024
www.lynxmotion.com

Logic Supply
1-802-821-2300
www.logicsupply.com

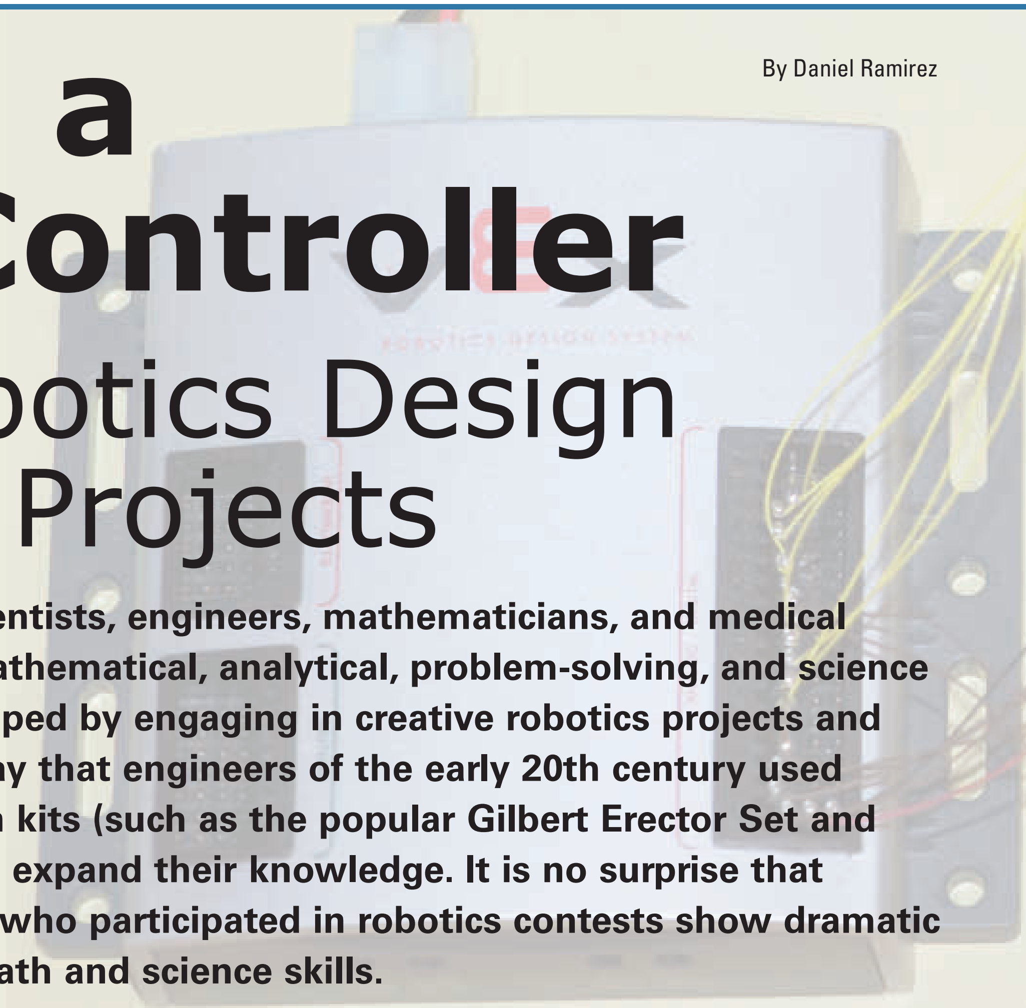
Parallax
1-888-512-1024
www.parallax.com

RoboRealm
Machine vision processing and robot control software.
www.roborealm.com

By Daniel Ramirez

Using a VEX Controller

VEX Robotics Design System Projects



Twenty-first century scientists, engineers, mathematicians, and medical researchers need the mathematical, analytical, problem-solving, and science skills that can be developed by engaging in creative robotics projects and contests in the same way that engineers of the early 20th century used mechanical construction kits (such as the popular Gilbert Erector Set and Meccano Erector Set) to expand their knowledge. It is no surprise that students in high school who participated in robotics contests show dramatic improvement in their math and science skills.

One area that has been neglected over the years is the electronics field that once helped to make the US the leader in electronics, computers, and consumer devices such as telephones, TVs, and radios in the 20th century. Now that lead has dramatically eroded due in part to globalization but also due to the lack of course curriculum provided by public and private schools in the US. Electronics courses are generally not available to secondary and high school students.

About the only exposure to electrical theory is in Physics classes. It's really at the college or university level that students can study electronics if they choose. Most students who learn electronics from an early age are either self-taught or have parents in the electronic field that provide assistance to them. Other resources include DIY electronics projects published in magazines such as *Nuts & Volts* and *SERVO Magazine*.

The VEX construction system (which is similar to the Gilbert and Meccano erector sets) is now widely used by both elementary and high school level students who compete regularly in national comptetions like the VEX Robotics Contest and FIRST. These contests have brought together international students from Canada, China, Mexico, Brazil, and other countries from around the world. The VEX microcontroller provides the motor control and also provides feedback from various sensors including bumper switches, limit switches, sonar rangers, IR rangers, and quadrature optical encoders for VEX-based robots. One only has to go to the VEX forum (**www.vexforum.com**) and the VEX Gallery (**www.vexforum.com/gallery/index.php**) to see the hundreds of models featured. Other independent VEX forums exist such as **www.vexfan.com**, which I highly recommend since they also have many models as examples.

The VEX microcontroller is the "silicon brain" that makes an excellent, low cost learning platform that can be used for carrying out both analog and digital electronic experiments, as well as ones for science, robotics, and animatronics (as I have tried to demonstrate in the last three articles). It can be used with the original VEX starter kit, the Vexplorer kit, and additional new kits that IFI now sells. These include educational classroom bundles, along with various robot accessories and contest props.

The VEX Keypad Experiment

In this article, I will show you how to develop a user interface for the VEX microcontroller using the DIY LCD display information described in my previous article (June '10 issue) with one additional component: an external 4 x 4 numeric keypad. I will detail how you can create your own custom keypad using standard pushbutton switches or even bumper switches and limit switches. I will also describe how I developed a practical user interface written in PIC18 C that demonstrates how you can enter the speed for a motor and special robot commands. It may be used as a terminal or menu-driven user interface for the VEX controller that you can use to enter numeric data necessary to run your applications in a portable manner, while being unplugged from the PC or laptop, using only the VEX microcontroller, a VEX motor, an LCD, and a keypad.

You will be pleasantly surprised that you will not need any other external components other than the keypad, LCD, and pin headers since the VEX controller already has pull-ups and resistors in series to protect the digital input ports even though the schematic shows them. The keypad works together with an LCD display by assigning the I/O pins in such a manner that it works as few pins as possible.

With the information presented here, you should be able to write C applications to make menu selections using a simple user interface so that you can select various robot behaviors without the need for a laptop or PC. Imagine all the VEX applications that could use a keypad as an input device when a laptop or PC is not available. It also opens up the VEX field to rapid prototyping machines such as calculators, vending machines, appliances, etc.

Pushbutton and toggle switches can be used as a convenient way to select various autonomous modes that run with the microcontroller mounted on a robot or prop (without having to use a PC). Pushbutton switches can also be used for menu selections for an embedded user interface. They can be wired to emulate a 4 x 4 keypad, providing up to 16 independent inputs using the same firmware provided with this article.

The pushbutton switches can be replaced with bumper switches or limits switches which is handy to sense objects and to check that mechanical stops or limits are not exceeded when running in autonomous modes.

There are many kinds of switches and pushbuttons. The normally open pushbutton (NO), normally closed pushbutton (NC), the single pole single throw toggle switch (SPST), the single pole double throw switch (SPDT), and others handle low voltages and currents to high voltage home appliance switches (120 volts AC). These switches include momentary switches.

You should also be able to connect up to 16 pushbutton switches and scan their states using only eight VEX I/O pins. If you think of each key as a pushbutton switch and wire it in a similar manner to the keypad matrix, you will be able to connect up to 16 individual pushbutton switches and use our firmware to scan and read their states using only eight I/O pins. Think of all the bumper switches and limit switches that you could sense for your next robot or prop.

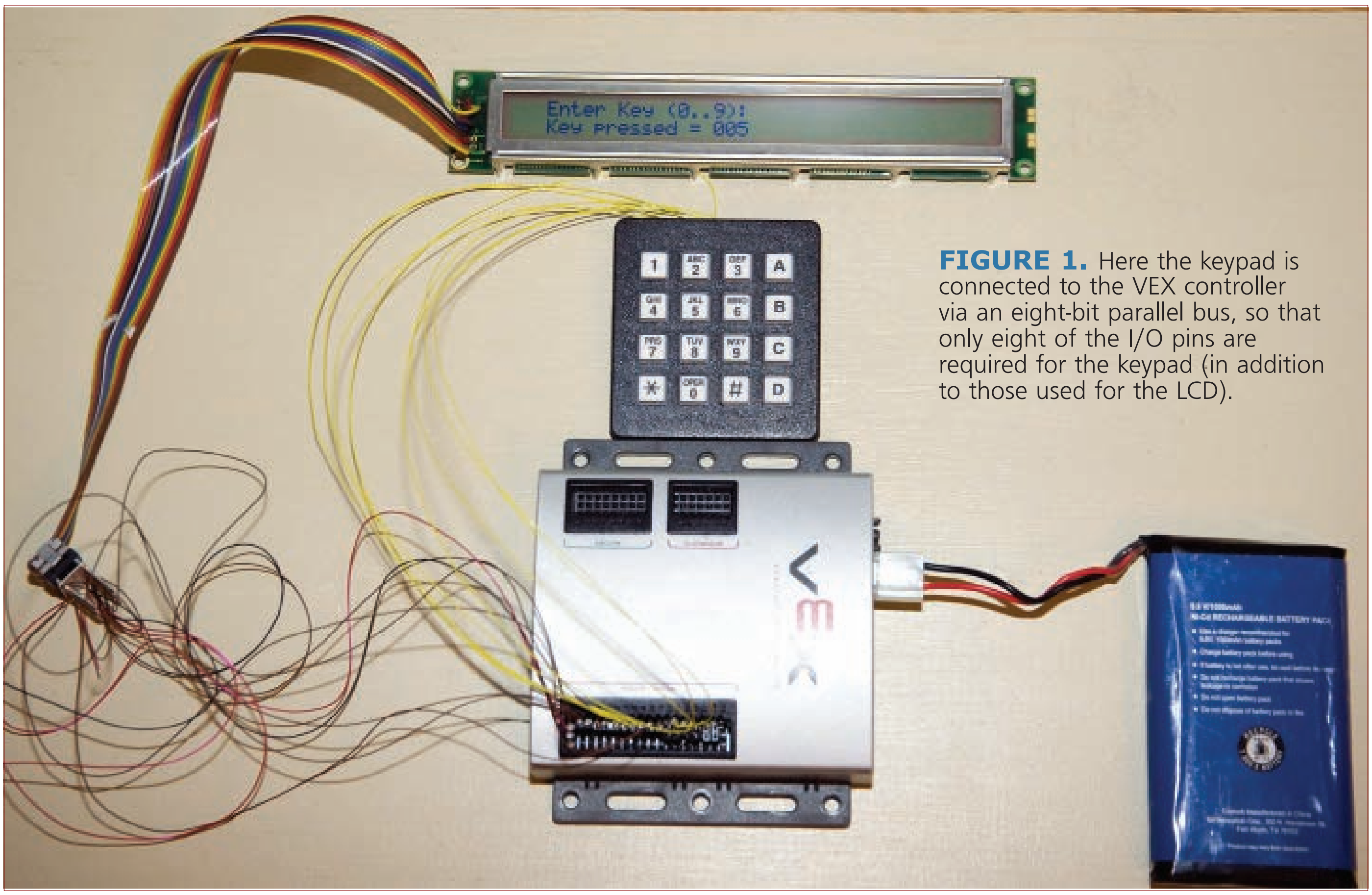


FIGURE 1. Here the keypad is connected to the VEX controller via an eight-bit parallel bus, so that only eight of the I/O pins are required for the keypad (in addition to those used for the LCD).

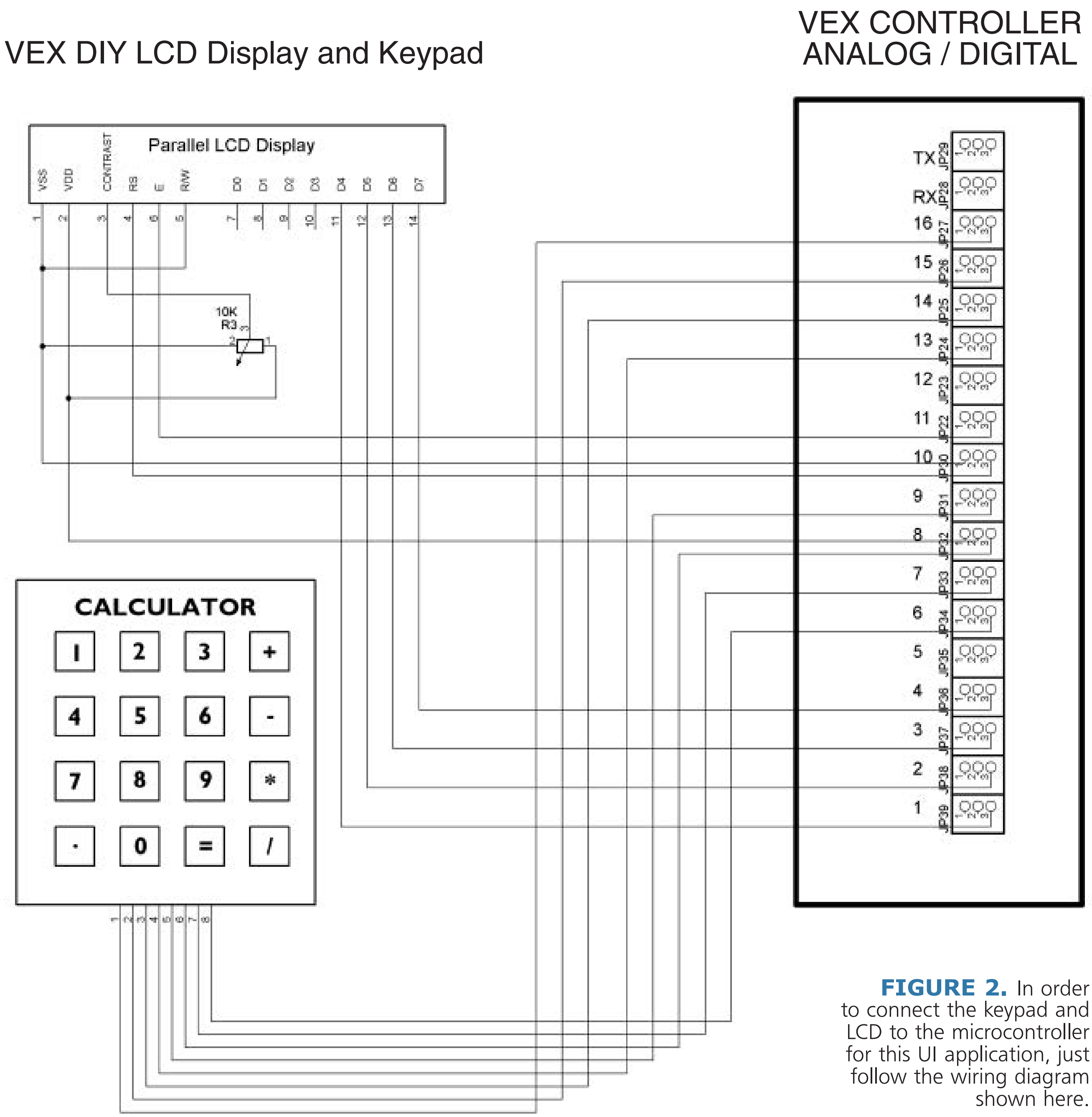


FIGURE 2. In order to connect the keypad and LCD to the microcontroller for this UI application, just follow the wiring diagram shown here.

Keen on Keypads

A keypad allows the user to enter data and commands to an embedded controller or the data entry for the user interface as shown in **Figure 1**. The keypad may also be used to enter text and numeric data like in a calculator application. The keypad and LCD combination makes an excellent portable data input device so robot behavior can be changed on the fly. The keypad is connected to the controller using an eight-bit bus — similar to the four-bit bus used on the LCD display.

Connecting a 4 x 4 Keypad

Connecting the keypad is very simple. The only parts required are the keypad, some pin headers, and wire-wrap wire as shown in the **figure**. Connections to other kinds of microcontrollers usually require four 10K to 470K pull-ups and eight 100 ohm resistors in series (or having weak pull-ups enabled on their input pins if this feature is available). The VEX controller has these components already connected to it internally. Again, our keypad is connected to the controller via an eight-bit parallel bus, so that only eight additional I/O pins are required (for the keypad) in addition

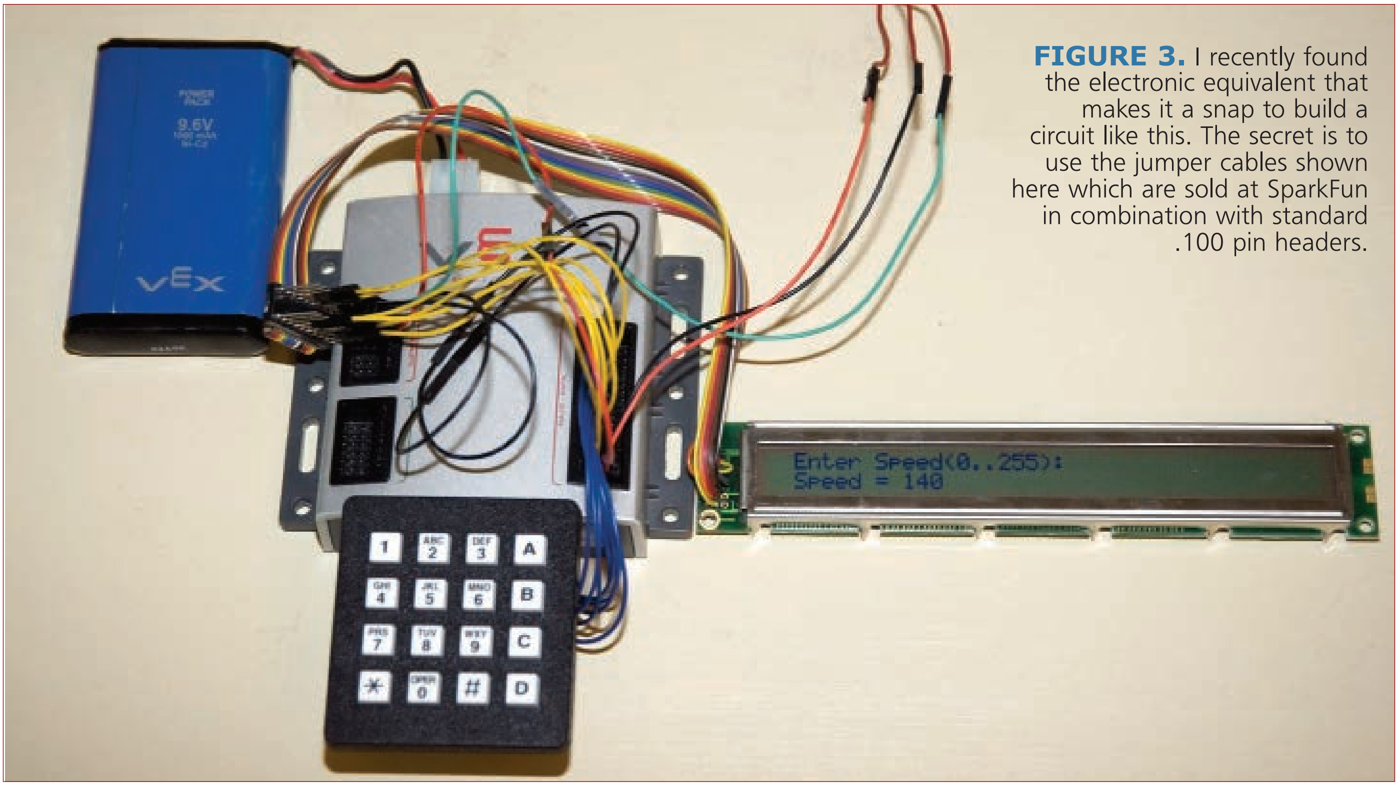


FIGURE 3. I recently found the electronic equivalent that makes it a snap to build a circuit like this. The secret is to use the jumper cables shown here which are sold at SparkFun in combination with standard .100 pin headers.

to those used for the LCD as shown in **Figure 2**.

A VEX User Interface

Data entry can be either in decimal, floating point, or hexadecimal format since the keypad has 16 keys. In fact, we can edit the values by assigning the unused keys from the keypad as a backspace, insert or delete key, cursor position, or tab key. Our PIC18 C User Interface (UI) application uses the keypad scan routine to scan for the user's keystrokes and uses the digits entered to build an ASCII representation of the motor speed entered. We'll slightly alter the firmware from the June '10 article to work with the keypad.

This application also uses the WPILIB routines to drive the VEX motor to the selected speed which prevented me from using a modified version of printf to display data since the WPILIB could not be modified. Instead, I had to resort to making specific function calls to position the cursor and format the data for the LCD.

Although the example provided is very rudimentary, the reader can build on this to develop custom user interfaces using our source code as a starting point.

Some Assembly Required

In order to connect the keypad and LCD to the microcontroller, just follow the wiring diagram shown in the schematic in **Figure 2**. The motor is connected to motor block, pin 1, located on the microcontroller. Please note that two of the LCD signals (RS and E pins) have been relocated from IO7 and IO6 to IO10 and IO11 on the microcontroller (this is different from the schematic in the May issue). I had to change the LCD wiring to accommodate the keypad firmware. The 10K potentiometer controls the contrast to the LCD and should be adjusted for ambient lighting conditions.

Locating pin 1 on the keypad (see **Figure 2**), check the orientation. If it is connected backwards, then the key scan routine will return the values incorrectly when pressed. Care must be taken when connecting the keypad since there are many signals and wires used. The resistors in series with the keypad wires are there to protect the microcontroller from Electro-Static Discharge (ESD). Check for shorts using a

TABLE 1. Bill of materials required to build the User Interface using a keypad and LCD. One component not shown in the schematic is a VEX motor. I used one to demonstrate how to change the motor speed using this interface.

ITEM	QTY	DESCRIPTION	SOURCE
1	1	VEX Controller	Innovation First, Inc. **www.vexforum.com**
2	1	4 x 4 Keypad	RadioShack **www.radioshack.com**
3	1	16 x 2 LCD Display	SparkFun **www.sparkfun.com**
4	40	.100 Pin Headers	Digi-Key **www.Digi-Key.com**
5	1	10K ohm Trim Potentiometer	Digi-Key **www.Digi-Key.com**
6	1	Wire-wrap Cable	RadioShack **www.radioshack.com**
7	1	Package of Jumper Cables	SparkFun **www.sparkfun.com**
8	1	VEX Motor	Innovation First, Inc. **www.vexforum.com**

DVM. Connections should be tested using a continuity tester (or DVM).

Rapid Prototyping the Keypad

I recently found something that makes it a snap to build our circuit. The secret is to use the jumper cables shown in **Figure 3** which are available from SparkFun (**www.sparkfun.com**) in combination with standard .100 pin headers. I was able to build the complete UI (including the keypad, LCD, and motor) in less than two hours using the parts shown in **Table 1**. I took my time to check the connections against the schematic, but one problem I encountered was that I accidently connected the wire leading from the 10K potentiometer wiper to a ground pin which caused smoke to come from the POT when it was first powered up. Fortunately, I was able to disconnect power quickly so no apparent damage was done. It is also always a good idea to check to see if the jumper cables are plugged into the correct pin sockets. These jumper cables are ideal electronic rapid prototyping materials since no soldering (other than pin headers) is required and the parts are easily taken apart and reused. The cables include assortments of various lengths and colors that just plug into the I/O blocks. It is faster than wire-wrapping but not as permanent since the jumper cables can be pulled out (unless they are fastened with tape or hot glue). I do not suggest using these for moving robots or props, or for making permanent circuits. Instead, consider using wire-wrap or point-to-point wiring for those applications. One idea I had is to build a three-row pin header connector so that the combined keypad and LCD modules could just be plugged into the microcontroller for a plug-and-play solution.

VEX Firmware

The PIC18 C example shown in **Listing 1** demonstrates how to scan the keypad for a particular key that has been pressed by the user and then displays it on the LCD. De-bouncing each keystroke is necessary so that repetitive key entries are not accidently entered as valid data.

Listing 1 provides all these necessary functions. To use it, just compile and link it using the PIC18 C tools, and then download it with the IFI loader.

```
// Display each keystroke entered from the
// keypad to the LCD.  Works!!!
// Clear the LCD and move the cursor to
// home position
lcd_clear();
Wait(2);

//Format the Message to be displayed on the LCD
Buffer[0] = ' ';
Buffer[1] = 'E';
Buffer[2] = 'n';
Buffer[3] = 't';
Buffer[4] = 'e';
Buffer[5] = 'r';
Buffer[6] = ' ';
Buffer[7] = 'K';
Buffer[8] = 'e';
Buffer[9] = 'y';
Buffer[10] = ' ';
Buffer[11] = '(';
Buffer[12] = '0';
Buffer[13] = '.';
Buffer[14] = '.';
Buffer[15] = '9';
Buffer[16] = ')';
Buffer[17] = ':';
Buffer[18] = 0;      // String terminator (null
                     // character)

// Position the cursor to the first line
lcd_goto(0,2);

Wait(1);

// Send message to the LCD Display
lcd_puts((char *) Buffer);

// Send message to the Serial Terminal
// if available
   printf("Enter Key ('0'..'9') \r\n",
   KeyValue);

//Format the Message to be displayed on the LCD
Buffer[0] = ' ';
Buffer[1] = 'K';
Buffer[2] = 'e';
Buffer[3] = 'y';
Buffer[4] = ' ';
Buffer[5] = 'p';
Buffer[6] = 'r';
Buffer[7] = 'e';
Buffer[8] = 's';
Buffer[9] = 's';
Buffer[10] = 'e';
Buffer[11] = 'd';
Buffer[12] = ' ';
Buffer[13] = '=';
Buffer[14] = ' ';
Buffer[15] = 0;      // String terminator
                     // (null character)

// Demo loop which reads values from keypad and
// sends them to the LCD. It Waits for press to
// indicate a keypress, then displays the key
// on the LCD Display.  Note that this code
// clears the press bit when done in order to
// prepare for the next press.
while(1)
{
   KeyValue = GetKey();     // Convert the key
                            // value from ASCII
                            // to binary


      // Check the range of the Key Value to make
      // sure is a digit between 0 and 9
      if ((KeyValue >=0) && (KeyValue <=9))
      {
         // Position the cursor
         lcd_goto(1,2);
         Wait(1);

         // Display formatted text to the LCD
        lcd_puts((char *) Buffer);
         lcd_printdec(KeyValue);
      }
}
```

LISTING 1. The PIC18 C example shown here demonstrates how to scan the keypad for a particular key that has been pressed by the user and then displays it on the LCD.

Using the WPILIB Library

WPILib was initially developed as a framework for programming robots used in the *FIRST* competition. There is a version that also works with standard VEX microcontrollers. This is an expanded version of the library that is used by easyC Pro; another version is available for PIC18 C and MPLAB. For more information on WPILib, go to **http://users.wpi.edu/~bamiller/WPILib/**.

There are PIC18 C, Easy C, and Easy C Professional functions that enable the VEX user to read digital and analog inputs which are common to all three C compilers. They all share the WPILIB framework that provides these functions. In Easy C and Easy C Professional, they are automatically included but in PIC18 C applications you need to include the C header file API.h and specify the WPILibVex.lib in the MPLAB project as shown in the screen capture in **Figure 4**. Programming details will be clarified in future articles with C examples, but it's a good idea to get familiar with the programming statements that relate to digital and analog inputs and outputs. (I used these functions to set the motor speed entered from the UI as demonstrated in **Listing 2**.) The PIC18 C example shows how you can change the speed of a motor and display the current speed value on the LCD. It also shows you how to use the WPILIB functions to set the actual motor speed.

Figure 4 shows all the necessary C modules including

```
// VEX User Interface (UI)
// This is a practical example that allows you
to enter the speed for a specific VEX Motor
// Display and display it on the LCD.

// Define a robot two wheels using WPILIB
TwoWheelDrive(1, 2);   // This code works!!!

// Clear the LCD and move the cursor to home
// position
lcd_clear();

//Format the Message to be displayed on the LCD
Buffer[0] = 'E';
Buffer[1] = 'n';
Buffer[2] = 't';
Buffer[3] = 'e';
Buffer[4] = 'r';
Buffer[5] = ' ';
Buffer[6] = 'S';
Buffer[7] = 'p';
Buffer[8] = 'e';
Buffer[9] = 'e';
Buffer[10] = 'd';
Buffer[11] = '(';
Buffer[12] = '0';
Buffer[13] = '.';
Buffer[14] = '.';
Buffer[15] = '2';
Buffer[16] = '5';
Buffer[17] = '5';
Buffer[18] = ')';
Buffer[19] = ':';
Buffer[20] = 0;       // String terminator
                      // (null character)

// Position the cursor to the first line
lcd_goto(0,2);

// Send message to the LCD Display
lcd_puts((char *) Buffer);

// Send message to the Serial Terminal
// if available
   printf("Enter Speed (0..255) \r\n",
KeyValue);

//Format the Message to be displayed on the LCD
Buffer[0] = 'S ';
Buffer[1] = 'p';
Buffer[2] = 'e';
Buffer[3] = 'e';
Buffer[4] = 'd';
Buffer[5] = ' ';
Buffer[6] = '=';
Buffer[7] = ' ';
Buffer[8] = 0;        // String terminator
                      // (null character)

Speed = 0;
i = 0;
while(1)
{
       KeyValue = GetKey();
              // Convert the key value from
              // ASCII to binary

    // Check the range of the Key Value to make
    // sure is a digit between 0 and 9
    if ((KeyValue >=0) && (KeyValue <=9))
    {
       // Enter the Speed one digit at a time

       if (i<3)
       {
          Speed += (int)KeyValue*PowerOfTen[i];
          i++;

          // Position the cursor
          lcd_goto(1,2);

          // Display formatted text to the LCD
         lcd_puts((char *) Buffer);
           lcd_printdec(Speed);
       }
       else
       {
          // Set the VEX Motor # 1 Speed
             SetPWM (1,Speed);

          // Reset digit count and Speed
          i = 0;
          Speed = 0;
       }
    }
    else if (KeyValue == ENTER)
    {
       // Process the speed value entered
              // Send the scanned key to the
              // host serial terminal also
    printf("Current Motor Speed = %d \r\n",
Speed);

       // Set the VEX Motor # 1 Speed
          SetPWM (1,Speed);

       // Reset digit count and Speed
       i = 0;
       Speed = 0;
    }
}
```

LISTING 2. This PIC18 C example shows how you can use a VEX UI to change the speed of a motor and display the current speed value on the LCD. It also shows you how to use the WPILIB functions to set the actual motor speed.

keypad.c and lcd.c that are required to implement a simple VEX UI. These are conveniently provided as a download from the *SERVO* website at **www.servomagazine.com**. To test the keypad and LCD interface, you only need to download the keypad.hex application. To customize it, you will need to modify the code in the main function of the WPIkeypad.c routine and re-make the project using MPLAB. The functions used in this application include the following:

- The Get_Key function scans the keypad and returns the keystroke that was pressed from the user after de-bouncing it.
- The lcd_goto function positions the LCD cursor to the specified position.
- The lcd_puts function sends the text contained in the buffer to the LCD.
- The lcd_printdec displays numeric data to the LCD.
- The TwoWheelDrive function called from the WPILIB initializes the microcontroller to run the VEX motors in two wheel drive mode (2WD).
- The SetPWM function called from the WPILIB is used to set motor #1 speed to the value entered from the keypad.

To run the application, download the WPIKeypad.hex file to the microcontroller using the IFI bootloader and start entering a speed for the motor with up to three digits for a speed value between 000 and 255. Notice that the first digit entered is the most significant digit (unlike calculators which enter the least significant digit first). This can easily be modified in the WPIKeypad.c application by storing the digits into a small buffer. The motor will start turning at the selected speed once an extra digit between 0 and 9 is entered from the keypad; the cycle repeats, overwriting the current motor command.

I will be referring to more WPILIB functions in future articles and showing you some advanced coding examples that would be much more difficult if done without WPILIB since controlling motors requires waveform generation (PWM) and reading sensors may require interrupt support. **Figure 4** also shows where I added my own routines to scan the keypad. This MPLAB project contains all the necessary modules to compile and link the keypad application that generates the keypad.hex file that is used to program the controller once the keypad has been wired to it. As a side note, I do prefer using the DDT tools instead of WPILIB when interrupts are required (such as scanning pushbutton switches or bumper switches since it allows access to most of the PIC18F8520 registers, timers, and peripherals).

Going further, an interesting experiment is to use the keypad to enter floating point or hex numbers, and display them using the numeric LED or LCD display, or develop custom menus for selecting various robot behaviors while disconnected from a PC.

Until next time, when I show you how to run stepper motors from the microcontroller. **SV**

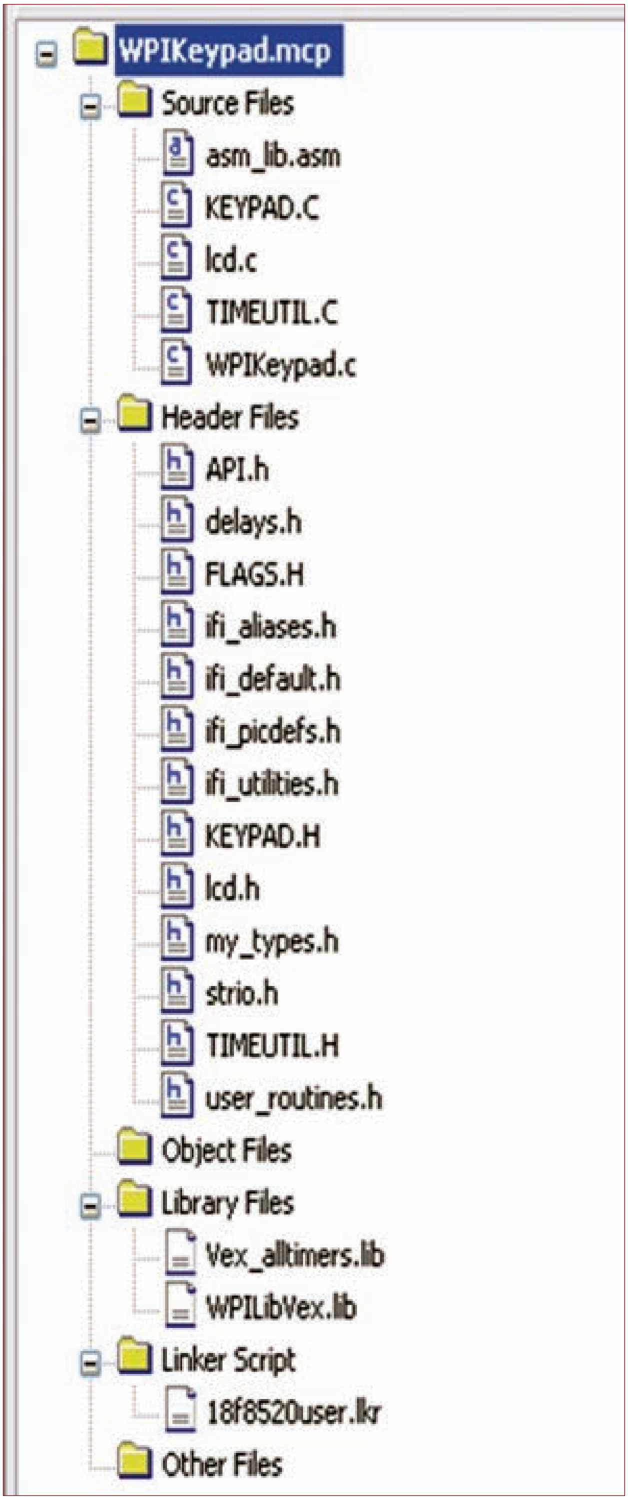


FIGURE 4. The MPLAB project file shows where I added my own routines to scan the keypad. This contains all the necessary modules to compile and link the keypad application that generates the keypad.hex file that is used to program the VEX controller once the keypad has been wired to it.

The NXT Big Thing

By Greg Intermaggio

We will be taking a closer look at the LEGO MINDSTORMS NXT kit and all the features that make it great. If you don't already have one, go online and get one from any of the various vendors selling them.We'll start by creating a simple driving chassis and doing some basic programming, then move on to some more complex projects (in future articles) like creating a Bluetooth remote control and using additional sensors available from **HiTechnic.com** to create some very cool, very unique projects.

The LEGO MINDSTORMS NXT is an amazing piece of technology in more ways than one. With the combination of the power of the NXT and the ability to rapidly prototype mechanisms with LEGO elements, the MINDSTORMS NXT kit really brings together all the components you need to learn the basics of robotics which would otherwise seem like a daunting task. First, we'll look at what exactly the NXT is good for; where it falls short; and just how far we can push the software. Then, we'll build a driving chassis, learn some basic programming, and talk about gear ratios.

What exactly is the NXT good for?

One of the most common responses I get when I tell people I teach LEGO robotics is "Wait — you can build real LEGO robots? I always thought the LEGO Star Wars kits were cool but I had no idea that there was a kit for building real robots!" The fact is, most people think of LEGO as the brick and have no concept of just how far LEGO has come over the years. The LEGO MINDSTORMS NXT is in every way more advanced than LEGO Star Wars, and it's important that people make a distinction between a 'model' and a 'robot.' The NXT is a robotics development kit — not a static model kit — so if you're looking to build the next big movie re-enactment in LEGO, look elsewhere.

Because the LEGO TECHNIC pieces included in the MINDSTORMS kit are made of plastic and the snap-in connections are relatively weak compared to screws or bolts, the NXT is primarily useful for learning about robotics and prototyping — rather than building the next big ComBot. In the kit, you get a set of three motors, several sensors, a brain, and a bin full of LEGO pieces to build your robot with — as well as software with which to program it.

Where does the NXT fall short?

The biggest shortcoming of the NXT is its inability to deal with extra weight. The pieces are not designed to handle much stress beyond their own weight. So, as soon as you add a few pounds of cargo to your bot, you'll probably notice sagging and strain on the motors. For this reason, it's important to try to keep your LEGO robots lightweight; even then, make sure to use reinforced connections wherever feasible. Scale is another issue that sometimes comes into play with LEGO. LEGO pieces are only so big and only so small. So, your dreams of building a life-sized robotic LEGO car or a miniature 500 g Sumo robot out of LEGO are sadly ill-founded. Besides that, the NXT is capable of some very impressive things.

Just how far can you push the software?

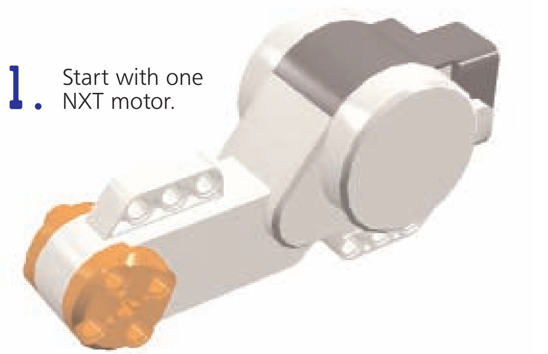
The awesome thing about the NXT is that it's open-source, and there are many languages and programming environments you can choose from to up the performance of your robot. For this reason, the NXT software is capable of some extremely impressive things. Notably, software has been created for NXT robots that balance on a ball, solve a Rubik's Cube, and even draw digital pictures on paper automatically!

For those looking to go beyond using the included software, you might check out Lejos — a Java-based programming language for the NXT — or Robot C — a C-based programming language also compatible with the NXT.

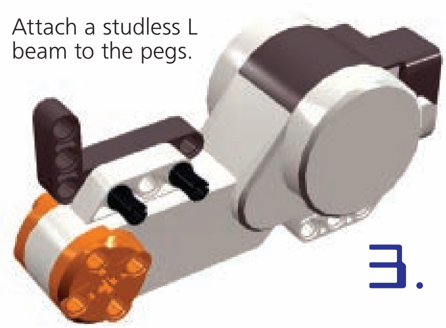
Building Eddie

Now that we've seen some of what's possible with the NXT, let's get on to our project: building a basic, expandable driving platform. The robotic driving platform we're building today is named Eddie 1.0 (the "ED"ucational robot). Eddie has two drive wheels and uses a plastic ball as a caster in the rear. Eddie is very expandable with lots of points to affix attachments, and some other great mechanical features that we'll talk more about later! So, let's get building!

1. Start with one NXT motor.



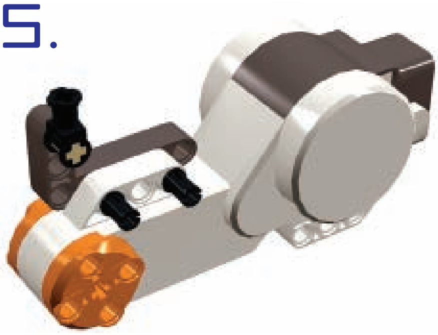
2. Add two "long" friction pegs.



3. Attach a studless L beam to the pegs.

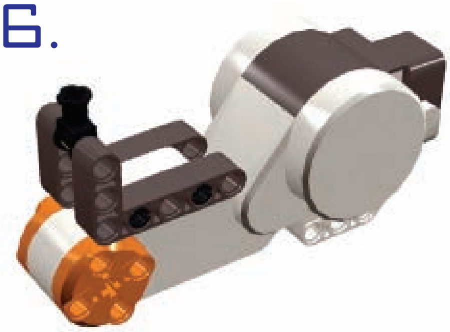
4. Add a tan axle-pin.

5.



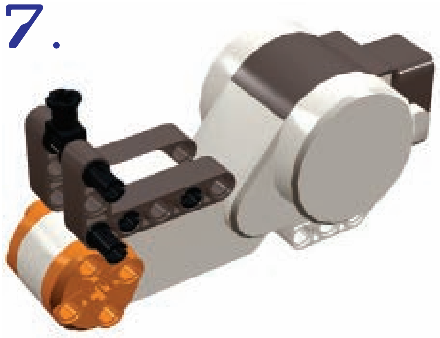
Now, slide a piston joint onto the axle.

6.



Attach a studless L beam to the other side of the pegs.

7.

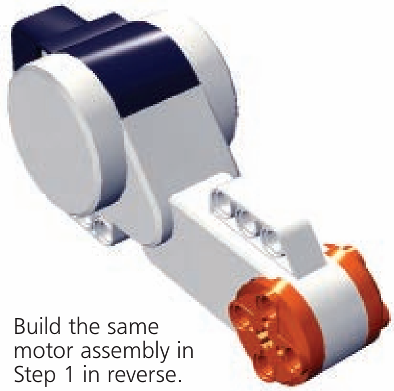


Snap in two standard friction pegs.

8.

Connect a studless beam with nine holes to the two friction pegs.

9.



Build the same motor assembly in Step 1 in reverse.

10.

Build the same motor assembly in Step 2 in reverse.

12.

Build the same motor assembly in Step 5 in reverse.

11.

Build the same motor assembly in Steps 3 and 4 in reverse.

13.

Build the same motor assembly in Step 6 in reverse.

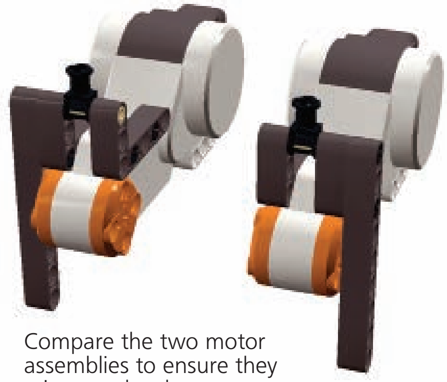
14.

Build the same motor assembly in Step 7 in reverse.

15.

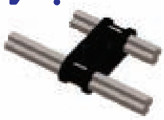
Build the same motor assembly in Step 8 in reverse.

16.



Compare the two motor assemblies to ensure they mirror each other.

17.



Stick a three-stud axle and a five-stud axle through a perpendicular axle-joiner.

18.

Slide one nine-hole studless beam onto either side of the axle-joiner.

19.

Line this assembly up between the two motor assemblies.

20.

Slide them together to ensure everything fits (at this time, the connection will be loose).

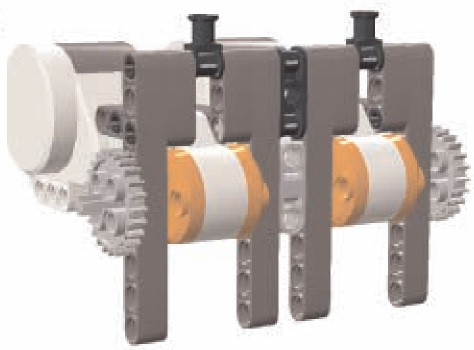
21.

Slide a seven-stud axle through each motor.

22.

Separate the motor assemblies for a moment to attach a 3 x 3 double peg.

23.



Attach a 24-tooth gear to each motor axle.

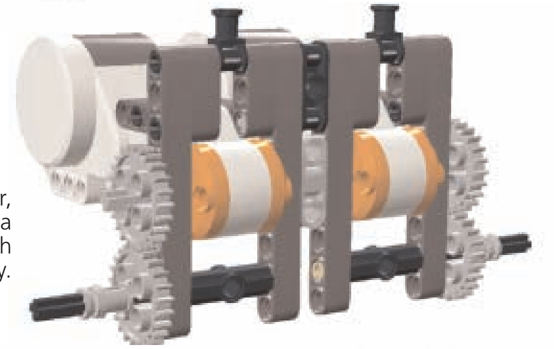
24.

Using two six-stud axles, two angle connector #2s, and two tan axle-pins, create the two assemblies indicated.

25.

Attach the axle assemblies one hole from the bottom of each motor assembly.

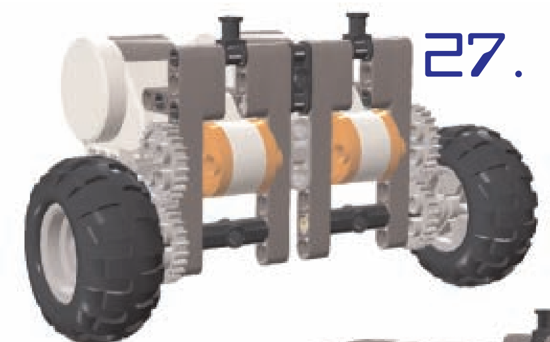
26.



Add a 24-tooth gear, a bushing, and then a half-bushing to each axle assembly.

Slide on the wheels.

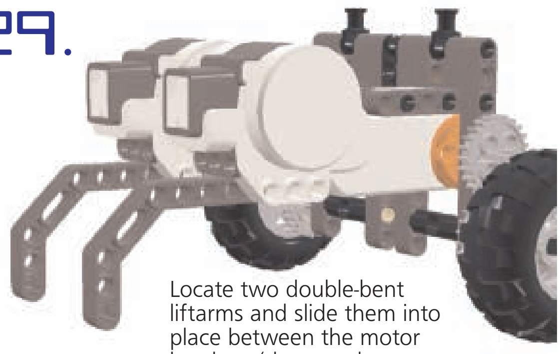
27.



Orient the robot as indicated to continue.

28.

29.

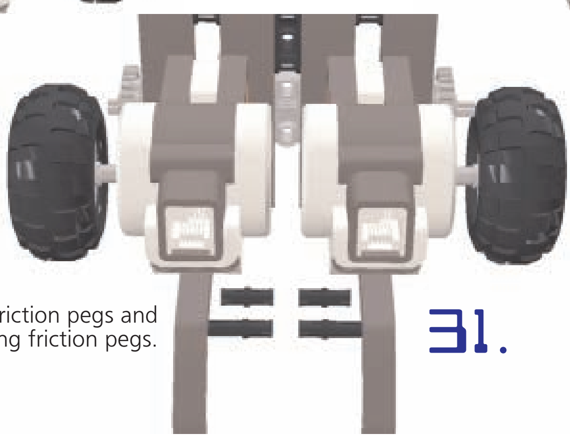


Locate two double-bent liftarms and slide them into place between the motor brackets (they aren't secured yet).

Re-orient the robot.

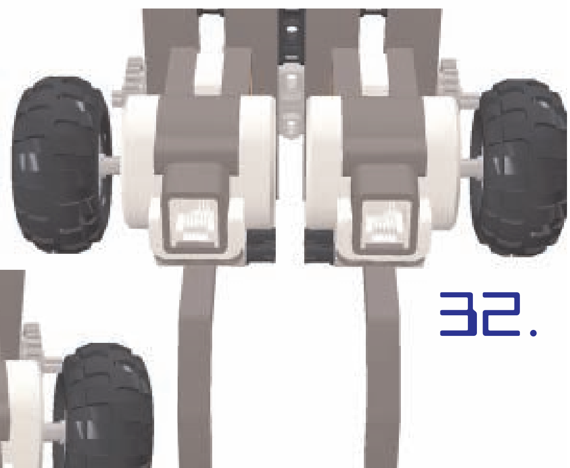
30.

31.



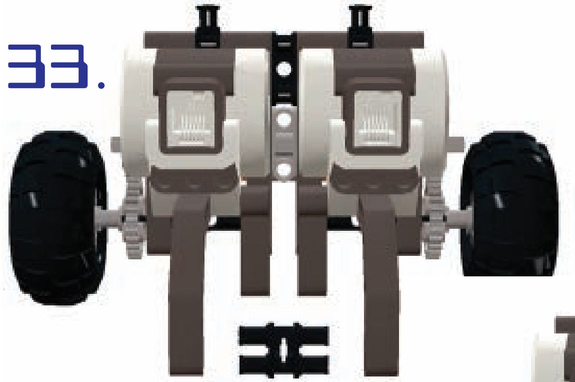
Locate two friction pegs and two long friction pegs.

32.



Attach them to the robot as indicated (shorter friction pegs go in the center brackets; longer friction pegs go in the brackets closest to you).

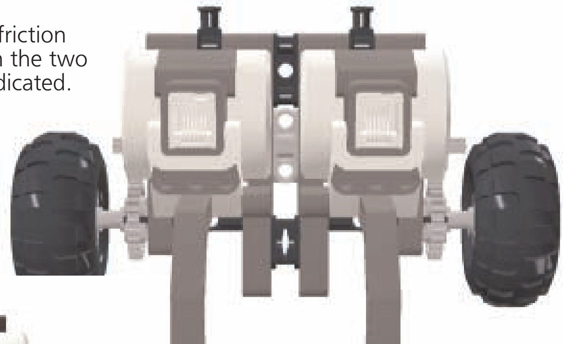
33.



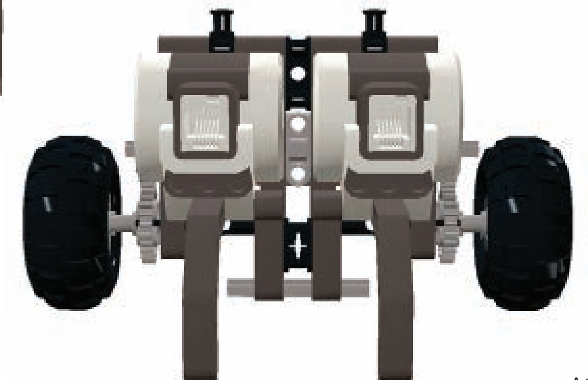
Now attach a mini L beam to the pegs you just attached, with the longer arm closer to the center of the robot. Also, find a double friction connector.

34.

Attach the double friction connector between the two mini L beams as indicated.

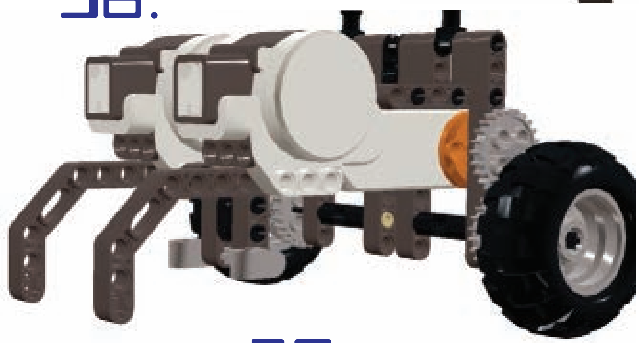


35.



Slide a five-stud axle through the bottom holes of the mini L beams.

36.



Now orient the robot like this.

37.

Find one normal friction peg and one long friction peg.

38.

Attach the normal one closest to the wheels, and the long peg in the center hole as indicated. Add similar pegs on the opposite side of the robot.

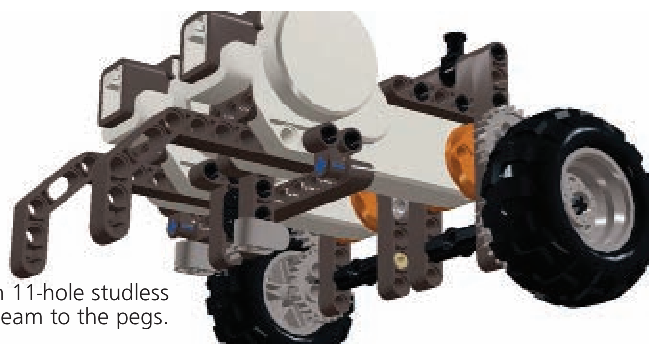
39.

Snap in double perpendicular axle-joiners on each of the two groups of two pegs.

40.

Slide blue axle-pins into each perpendicular axle-joiner, facing the wheels.

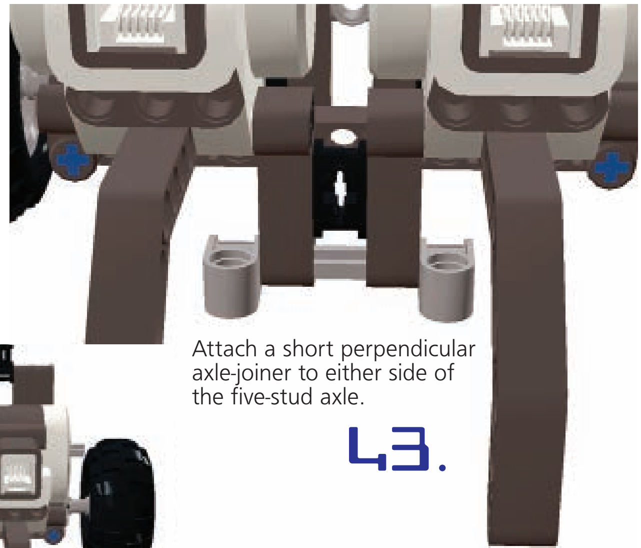
41.



Attach an 11-hole studless beam to the pegs.

Take a closer look at what will become the rear of the robot.

42.



Attach a short perpendicular axle-joiner to either side of the five-stud axle.

43.

44.

Some perspective.

45.

Now, we'll add the parts that will hold the rear caster ball in place.

46.

Add two tan axle-pegs, two three-stud axles, and two bushings as indicated.

47.

Slide in perpendicular axle-joiners onto each set of axles and add a friction peg to each.

48.

Snap on a nine-hole studless beam.

49.

Now, we're almost ready to attach the NXT.

50.

Attach a blue axle-peg on either side of the caster assembly.

51.

Slide in perpendicular axle-joiners on each one with a friction peg facing away from the wheels.

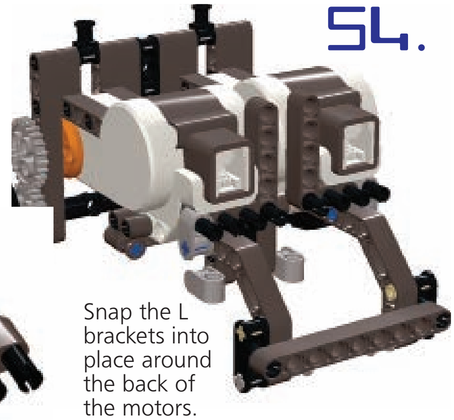
52.

Find two L brackets and snap two long friction pegs into each as indicated.

Connect one normal friction peg to each, as well.

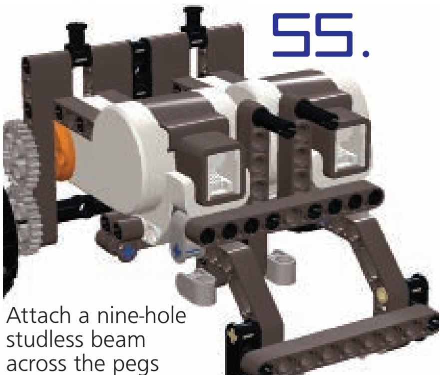
53.

54.



Snap the L brackets into place around the back of the motors.

55.

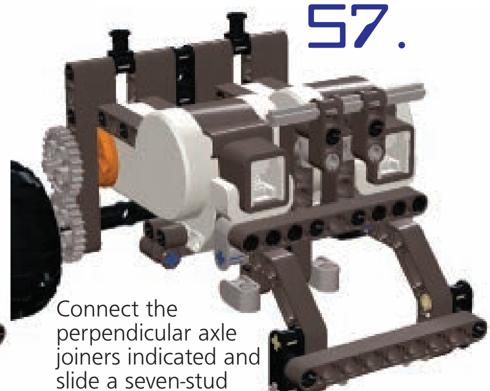


Attach a nine-hole studless beam across the pegs and add two long friction pegs atop the L brackets.

56.

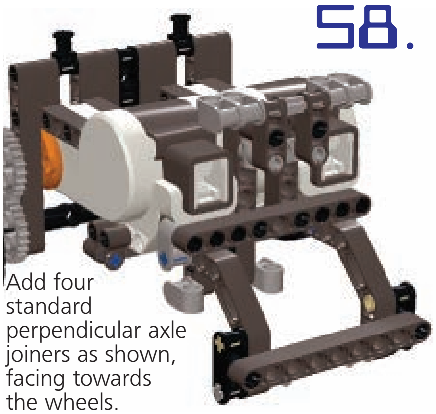
Attach a three-hole studless beam to the friction pegs and snap in a 3 x 3 double peg below the studless beam.

57.



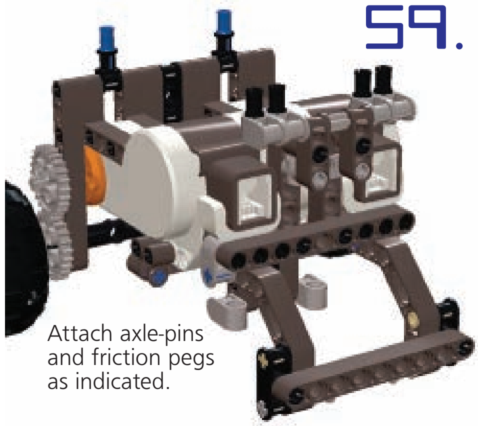
Connect the perpendicular axle joiners indicated and slide a seven-stud axle through them.

58.



Add four standard perpendicular axle joiners as shown, facing towards the wheels.

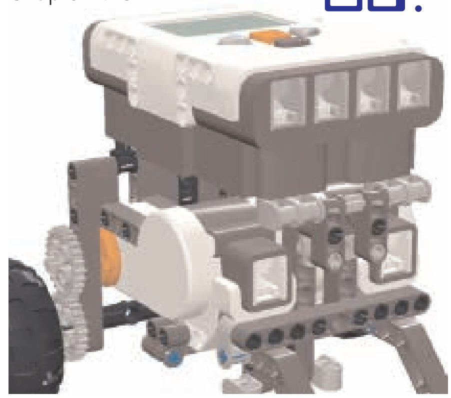
59.



Attach axle-pins and friction pegs as indicated.

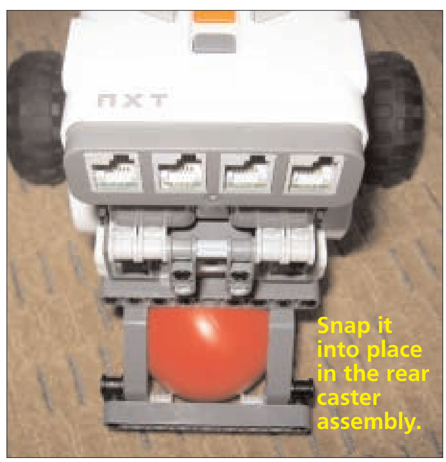
60.

Snap on the NXT.



At this point, your robot should look something like this.

Find the red ball included in your NXT Educational Kit.



Snap it into place in the rear caster assembly.

The final product.

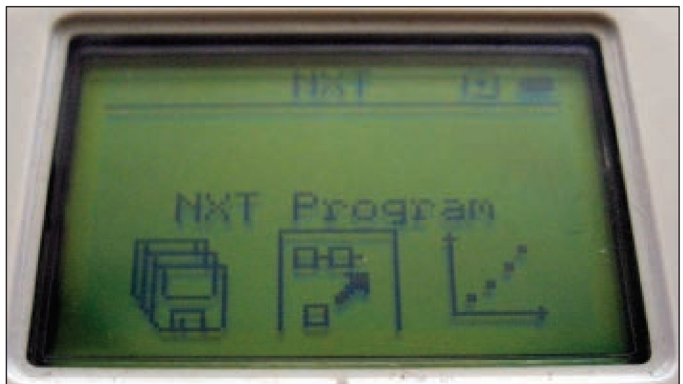
A view from above (note: in this orientation, the robot's front is at the top, and the rear is at the bottom).

Your First Program

Now that the chassis is built, it's time to experiment with a program or two. Plug the left drive motor into port B and the right drive motor into port C. Then, turn on the NXT and navigate to the NXT Program menu. This feature will allow you to create very simple programs for your robot, as opposed to the more complex computer-based software we'll be looking at next time.

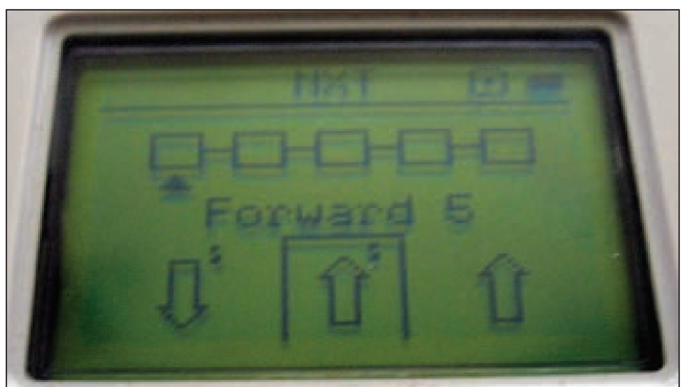
The very first thing to try is to make the robot move forward and backward in a straight line. This is the robotic equivalent of a program that says "Hello World!" As your first step in the NXT Program editor, select the "Forward 5." Then select "Empty" for step 2; "Backward 5" for step 3; and "Empty" for step 4. Finally, choose "Loop" instead of "Stop," and press "Run" to see Eddie move for the first time.

Because we consider the drive wheels to be the front of the robot and the caster to be the back, Eddie actually drives backwards when you tell him to go forwards, and forwards when you tell him to go back. The reason for this is his gear train.



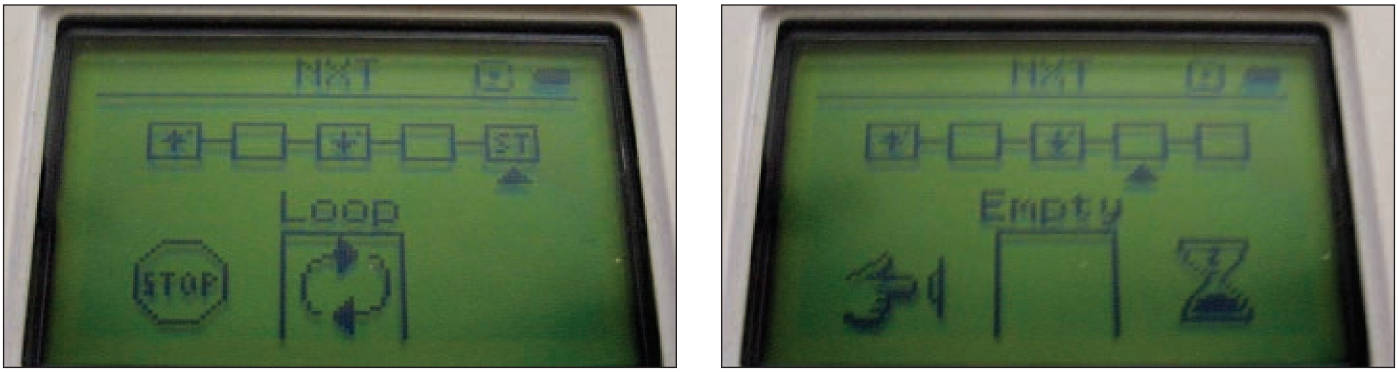
NXT Program icon.

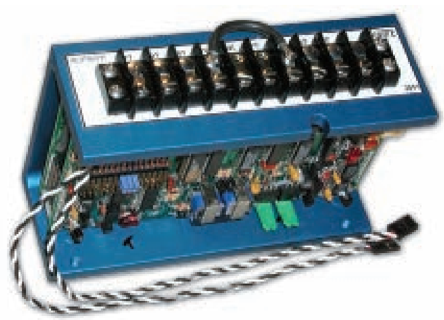
NXT menu screen after turning it on.



Programming on the NXT.

More programming on the NXT.





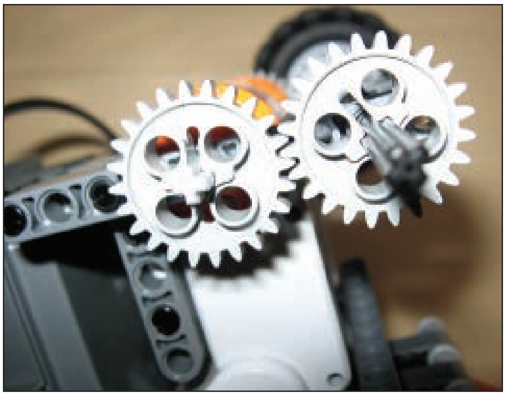
Gear Trains and Ratios

Gear trains are very important in the world of robotics because they provide not only a transfer of energy, but a translation of energy. In other words, speed can be translated into torque and torque to speed. The amount of speed and torque output of a gear train is measured in a "gear ratio." In our case, Eddie is running a 1:1 (24:24) gear ratio.

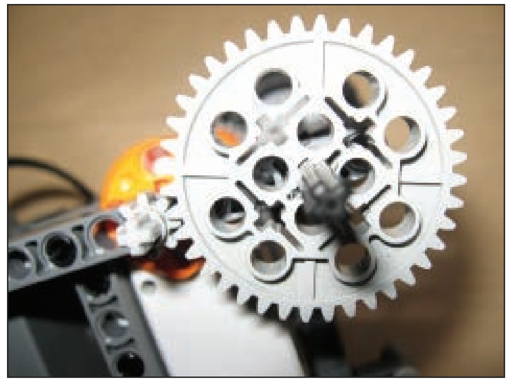
To understand gear ratios, just remember that the bigger the number on the left is compared to the number on the right, the faster and weaker the output will be. Conversely, the bigger the number on the right of the ratio is compared to the left, the slower and stronger the output of the gear train will be.

Right now, because we have a 1:1 gear ratio, our output is the same as our input (minus friction and other nominal forces). Experiment with Eddie by replacing the gears on the motors with tiny eight-tooth gears, and the gears on the drive axles with the large 40-tooth gears. This will give Eddie an 8:40 or 1:5 gear ratio, meaning that he'll have a lot more pushing power. You can also try the opposite — put 40-tooth gears on the motors and eight-tooth gears on the drive wheels, and watch the wheels fly!

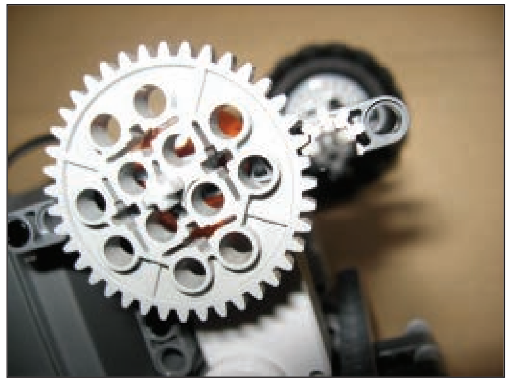
Remember, you can also move the drive axle position to fit even more gear ratios.



A 1:1 (24:24) gear ratio.



A 1:5 (8:40) gear ratio, geared for torque.



A 5:1 (40:8) gear ratio, geared for speed.

Experiment!

Now that you understand the basics, do some experimentation. Here are some ideas:

- Get Eddie to move forward, then stop.
- Try making Eddie turn to the right, then move forward.
- Make Eddie drive in an inward spiral.
- Try gearing Eddie to win a drag race — what's the optimal speed: torque ratio?
- Gear Eddie to pull a clipboard with a six-pack of soda on it.
- Create your own experimental program or gear train.

Wrapping Up

We've taken a look at the LEGO MINDSTORMS NXT kit as a whole, and built a basic driving chassis. Next time, we'll take a look at how to program the NXT on a computer, and learn all about touch sensors. From there, we'll continue to add to our knowledge of the NXT, eventually working with all the sensors from LEGO, and some third party sensors as well, so stay tuned! SV

Getting Down to Some Basics of Using C

Modern roboteers have unlimited access to innovative mechanical tools and development devices. However, many of you scientifically-inclined mechanics don't realize that you also have a similarly powerful tool chain on the software side.

By Fred Eady

In today's electromechanical environment, it is becoming commonplace to see a person lay down a mechanically sound weld joint and then turn to his or her laptop and write some code with the same dexterity. In the past, writing embedded electromechanical programs in a microcontroller's native assembler was normally beyond the average robotic machinist. In the infant days of microcontrollers and microprocessors, writing pure assembler applications with the development tools of the day was just as heavy a task for those who proclaimed to know what they were doing.

Assembler programming is still as important today because there are times when a tight group of hand-picked assembler mnemonics is the best way to generate or capture a high speed signal or event. However, in the early days of melding microcontrollers with machined parts, machinists that could also wear the embedded programmer hat began to jump ship on assembler programming only to be rescued by the BASIC programming language. These electromechanical engineers needed to switch power, spin motors, and sense limits. BASIC was a very easy way to meet these requirements and was just as easy to learn.

Today's mechatronic engineers share their task lists with those that came before them. As a person that is capable of creating useful metallic objects, you also need to switch power sources, spin and control the rotational direction of AC or DC motors, and sense the position of "things." Those "things" are usually the rotational position of a motor shaft or the extent of the linear extension of a mechanical device. A major advantage the modern roboteer has over his forefathers is the luxury of choosing from a pool of embedded programming languages like BASIC, Pascal, Forth, and C.

As you would imagine, I participate in bunches of embedded mechatronic conversations and more than once I've had a BASIC-using, mechanically-inclined person say, "I don't use C because it's too hard to learn." That statement always takes me back to my childhood and being introduced to my very first piece of pizza. It was one of those very early 1960's grocery store frozen jobs and the lady burned it. We didn't have a pizza joint in my hometown of Fayetteville, TN back then, but that didn't stop me from trying pizza again.

With that, if you're hungry to learn C in a way that complements your mechanical talents, you've picked up the right magazine and thumbed to the correct page.

BASIC is a Great Language

This month's discussion will not be a BASIC language bash session. In fact, in a future *SERVO* discussion we'll give BASIC the same treatment as we're about to give to the C language. We're simply going to concentrate on C this time around.

PHOTO 1. This linear actuator is a superior alternative to a hobby servo and a push wire for most robotic linear thrust applications.

Dennis Ritchie was rolling out the C programming language at about the same time I was graduating from high school. My very first C encounter occurred in the mid 1980s by way of a 5150 IBM PC. A few years later, the first serious PIC C project I embarked upon was supported by the CCS C compiler. The CCS C compiler is totally oriented towards the PIC microcontroller. A multitude of PIC-friendly built-in functions makes life easy for the PIC C programmer as canned library routines can be called at will to perform tasks related to RS-232, I²C, and GPIO (General Purpose Input Output), just to name a few. Although the CCS C compiler has lots of code in a can, that doesn't mean you can't use it to reach out and take control of your program and the resources provided by your target PIC. You can still write your own functions that get down to the PIC register and bit level with the CCS C compiler. If the need arises, the CCS compiler will also allow you to mix in a bit of assembler with your C.

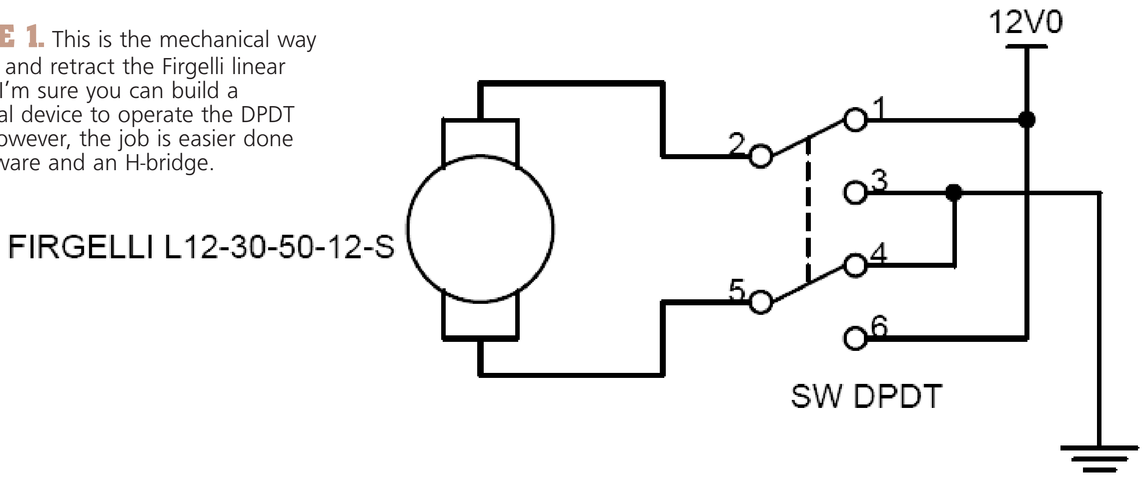
Driving a Firgelli L12 Linear Actuator with CCS PIC C

Let's do some real work with some real C source instead of wading through a textbook introduction to the CCS C compiler. This particular project we are about to code involves extending and retracting the business end of the Firgelli linear actuator shown in **Photo 1**. The L12 Firgelli actuator we'll be working with is designed to push and pull loads along the full length of its stroke. Our particular actuator has a gear reduction ratio of 50 which allows it to move relatively quickly. However, the price for faster actuator movement is paid for with reduced actuator force. Thus, an L12 with a gear reduction ratio of 210 moves slowly at full speed but produces more than twice the force of a quick-moving linear actuator geared at 50.

An L12 actuator will stop and hold its position when power is removed. A load that exceeds the actuator's holding force will drive it backwards. I've backdriven many a linear actuator without inflicting damage on it. Stalling of the actuator is allowed for short periods of time since a few seconds of stall will not damage it.

Let's use the model designators you see stamped to our Firgelli linear actuator in **Photo 1** to determine the type of L12 we will be working with. (Looking at the label, the model designator is interpreted as model number (L12): stroke length in millimeters (30); gear reduction ratio (50); voltage (12); controller type (S); and custom features (C). So, to reiterate, we can see that our L12

FIGURE 1. This is the mechanical way to extend and retract the Firgelli linear actuator. I'm sure you can build a mechanical device to operate the DPDT switch. However, the job is easier done with software and an H-bridge.



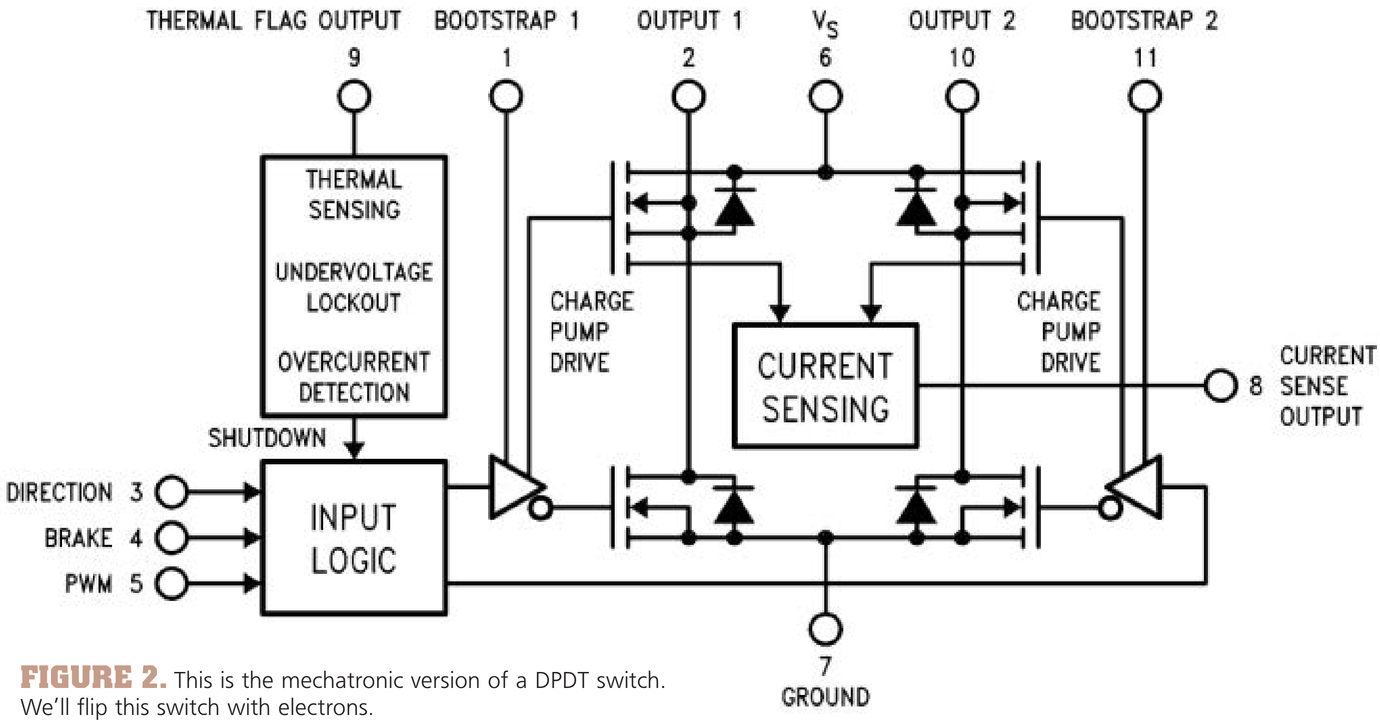


FIGURE 2. This is the mechatronic version of a DPDT switch. We'll flip this switch with electrons.

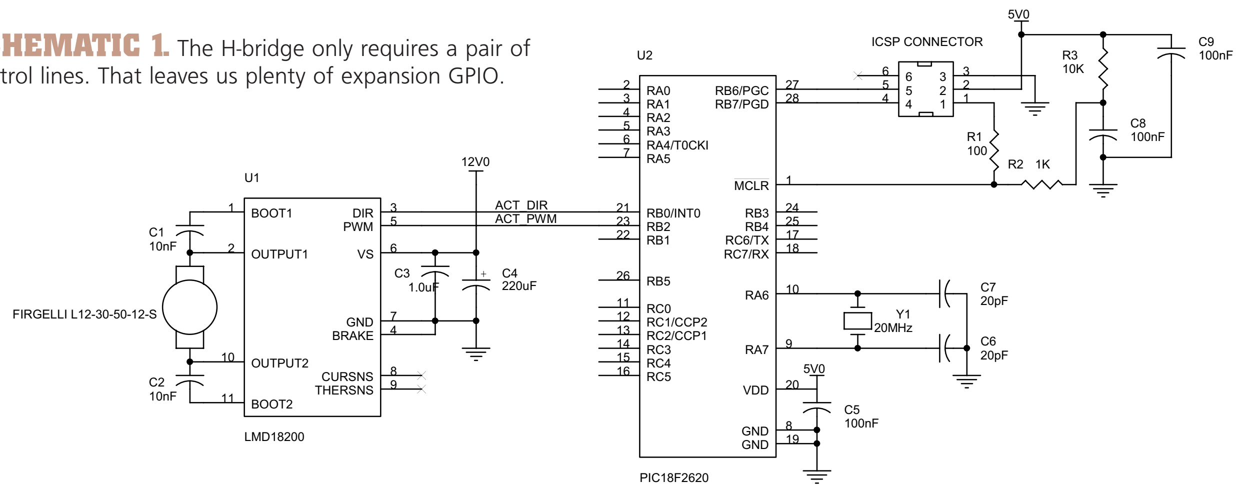
actuator has a 30 mm stroke with a gear reduction ratio of 50. The actuator's working voltage is 12 volts and the controller is a two-wire, open-loop interface with limit switching at the stroke endpoints. Actually, this actuator does not house an electronic "controller" since power is applied directly to its motor. The only control we have over this actuator is to reduce the voltage to the motor which will result in slower actuator movement and reduced load handling capability. Motor voltage reduction is usually only considered in battery-powered systems.

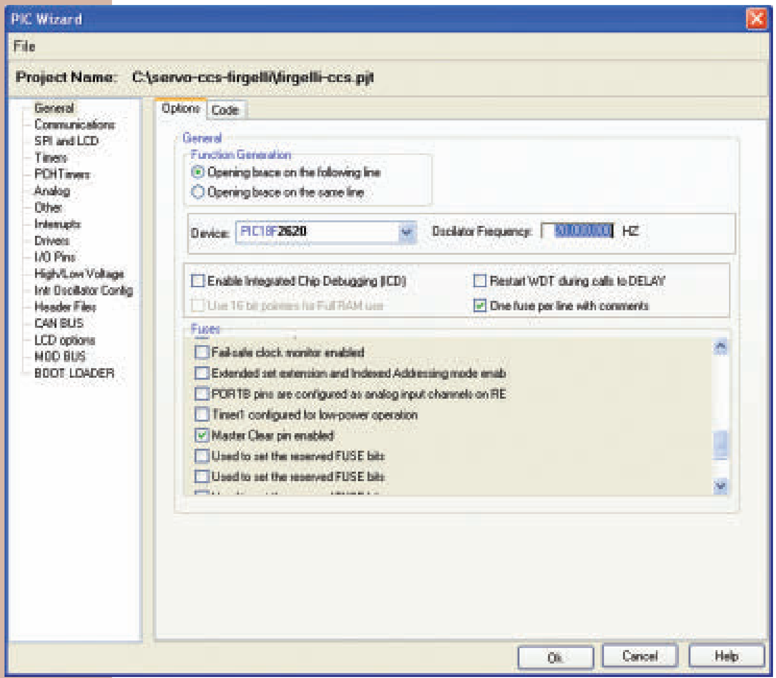
The two-wire interface is actually a power interface with the red wire representing *MotorV+* and the black wire acting as the *Motor ground* input. An internal limit switch will remove power to the motor when the actuator is within 0.5 mm from either stroke extent. Once an extent is reached and power is removed from the actuator by the limit switch, the only way to move the actuator is to reverse the polarity of the voltage applied to the actuator power interface. The actuator will drive in the opposite direction until it is within 0.5 mm of the extent it is approaching.

A purely mechanical way of extending and retracting the actuator is depicted in **Figure 1**. A DPDT mechanical switch is all we need to reverse the polarity at the motor interface. We can write the firmware behind the mechanics you can build to throw the switch, or we can convert mechanics in **Figure 1** to mechatronics.

The mechatronic alternative is drawn up in **Figure 2**. Now we have something to hang some code around as the H-bridge device has three logic inputs. We need to rough out an H-bridge/PIC

SCHEMATIC 1. The H-bridge only requires a pair of control lines. That leaves us plenty of expansion GPIO.





SCREENSHOT 1. The CCS C compiler PIC Wizard utility allows a programmer to twiddle the bits on all of the selected PICs peripherals and I/O. Once you have things configured the way you want them, the Wizard writes the code for you.

design before we can begin our C coding. Since our requirements are small, let's keep the microcontroller design small. I've chosen the venerable 28-pin PIC18F2620 to manipulate the H-bridge. You can see how I attached the PIC18F2620's I/O to it in **Schematic 1**.

CCS C Compiler Primer

I used the CCS C compiler's Project Wizard to generate the skeleton code for our project here. **Screenshot 1** is a capture of the PIC Wizard window I used to configure the PIC18F2620's clock and fuses. In reality, we can write all of the code that the PIC Wizard generates manually. However, it is easier and faster to use the Wizard. Here's the code that the PIC Wizard produced from my clock and fuse inputs:

```
#include <18F2620.h>
#device adc=8

#FUSES NOWDT                //No Watch Dog Timer
#FUSES WDT128               //Watch Dog Timer uses 1:128 Postscale
#FUSES HS                   //High speed Osc (> 4mhz for PCM/PCH) (>10mhz
                            //for PCD)
#FUSES NOPROTECT            //Code not protected from reading
#FUSES NOIESO               //Internal External Switch Over mode disabled
#FUSES NOBROWNOUT           //No brownout reset
#FUSES BORV21               //Brownout reset at 2.1V
#FUSES NOPUT                //No Power Up Timer
#FUSES NOCPD                //No EE protection
#FUSES STVREN               //Stack full/underflow will cause reset
#FUSES NODEBUG              //No Debug mode for ICD
#FUSES NOLVP                //No low voltage prgming, B3(PIC16) or B5(PIC18)
                            //sed for I/O
#FUSES NOWRT                //Program memory not write protected
#FUSES NOWRTD               //Data EEPROM not write protected
#FUSES NOEBTR               //Memory not protected from table reads
#FUSES NOCPB                //No Boot Block code protection
#FUSES NOEBTRB              //Boot block not protected from table reads
#FUSES NOWRTC               //configuration not registers write protected
#FUSES NOWRTB               //Boot block not write protected
#FUSES NOFCMEN              //Fail-safe clock monitor disabled
#FUSES NOXINST              //Extended set extension and Indexed Addressing mode
                            //disabled (Legacy mode)
#FUSES NOPBADEN             //PORTB pins are configured as digital I/O on RESET
#FUSES NOLPT1OSC            //Timer1 configured for higher power operation
#FUSES MCLR                 //Master Clear pin enabled

#use delay(clock=20000000)
#define ACT_DIR    PIN_B0
#define ACT_PWM    PIN_B1
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
```

Note the pair of *#define* commands that associate logical names with the PIC18F2620's PORTB I/O pins. The use of this Wizard to create these associations is the reason that the *#define* commands are in this code segment. In that the fuse and clock code is automatically encapsulated into a *#include* file and is not displayed by default in the CCS PIC C IDE, I prefer to have the *#define* commands lie in the main C source file where they are easily referenced during the coding process. In our case, we can simply cut and paste them into the main C source file. We could have manually coded the *#define* commands into our main C source file. **Screenshot 2** shows how I used the PIC Wizard to generate the ACT_DIR and ACT_PWM *#define* commands.

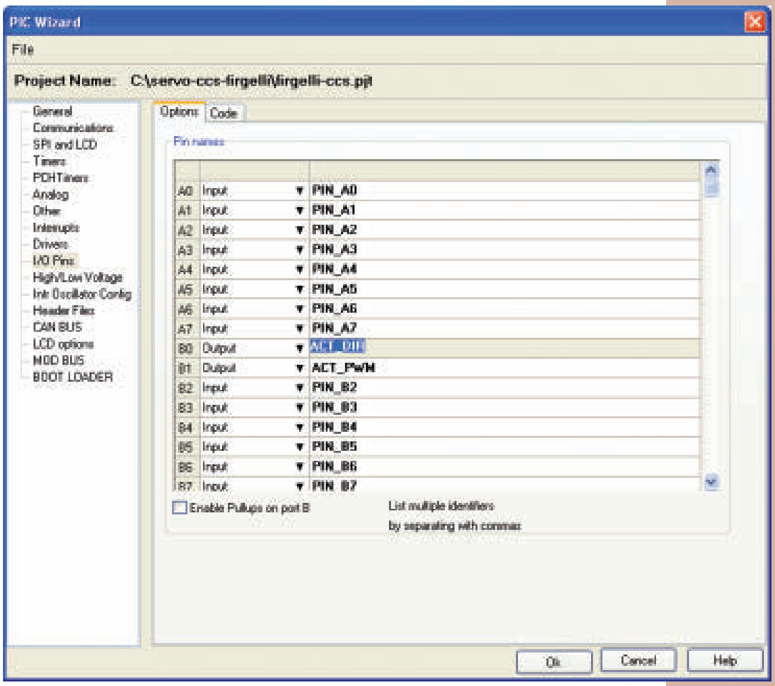
In addition to generating the fuse, clock, RS-232, and *#define* command code, the PIC Wizard will generate main C source code to initialize the target PIC's peripheral set according to the programmer's configuration input:

```
#include "C:\servo-ccs-firgelli\firgelli-ccs.h"

#define ACT_DIR    PIN_B0
#define ACT_PWM    PIN_B1

void main()
{

    setup_adc_ports(NO_ANALOGS|VSS_VDD);
    setup_adc(ADC_OFF|ADC_TAD_MUL_0);
    setup_spi(SPI_SS_DISABLED);
    setup_wdt(WDT_OFF);
    setup_timer_0(RTCC_OFF);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
//Setup_Oscillator parameter not selected from Intr Oscillator Config tab

    // TODO: USER CODE!!

}
```

All of the PIC18F2620 fuse, clock, RS-232, and *#define* command code gets stuffed into the *firgelli-ccs.h* file. As you can see in the code snippet, I cut the *#define* commands out of the *firgelli-ccs.h* file and pasted them into the main C source created by the PIC Wizard utility. Note that all of the PIC18F2620 peripheral configuration code inside of the C *main* function is composed of CCS compiler built-in functions. These functions eliminate searching through the PIC datasheet for each peripheral's SFR (Special Function Register) bit definitions. Another advantage of using the compiler's built-in functions is that you can read the function arguments and immediately know what is enabled, disabled, on, and off.

We must take a peek at the LMD18200T H-bridge datasheet to understand how to write our Firgelli linear actuator driver code. The LMD18200T is a 3A H-bridge that is specifically designed for motion control applications. All of the necessary H-bridge safeguards are built into the LMD18200T to prevent thermal or transient initiated damage to the actuator motor. This H-bridge is an excellent robotic motion part as it is inexpensive, available, and rugged.

The LMD18200T's DIRECTION input determines which pair of OUTPUT MOSFETs is active. The BRAKE input is used to quickly stop a motor attached to the LMD18200T H-bridge's OUTPUT pins by shorting like-powered MOSFETs in the H-bridge. Our actuators won't be moving fast enough to need breaking. So, we'll tie the BREAK input to ground which disables the LMD18200T's braking function. Normally, the H-bridge's PWM input would be used to control the speed of a motor being driven by the LMD18200T. We'll use the PWM input as a power switch. When the PWM pin is driven logically low, the DIRECTION input logic level is ignored and only the source MOSFETs attached to V_S are biased ON. With no path to ground, the actuator motor is stopped and the biased ON MOSFETs act as a brake. Let's put a stake in the ground and standardize on the actuator's red



SCREENSHOT 2. The PIC Wizard allows the programmer to associate all of the target PIC's available I/O with a logical name. If you choose to use the Wizard to generate the #define commands instead of manually coding them, the associations will be placed in the #include file instead of your main C source file.

MotorV+ lead being attached to the H-bridge's OUTPUT 1 and the actuator's black *Motor ground* lead attached to the H-bridge's OUTPUT 2. To extend the actuator, we must drive OUTPUT 1 with V_S and provide a ground path via OUTPUT 2. To accomplish this, we must drive the DIRECTION and PWM inputs logically high. To retract the actuator, we reverse the current flow through the actuator's motor by driving OUTPUT 2 from V_S and grounding OUTPUT 1. Driving the H-bridge's DIRECTION logically low while holding the PWM input logically high will retract the actuator. Thus, we can start and stop actuator motion by driving the H-bridge's PWM input logically high and logically low, respectively. When the PWM input is being driven logically high, the DIRECTION input logic level determines the actuator's direction of travel.

Let's turn the H-bridge actuator motion logic into C source code. However, before we dive into the logic of driving the actuator, we need to do a bit of I/O port preparation. If you're familiar with the PIC, you know that we need to configure PIC I/O pins we will use for input or output operation. The CCS compiler does a great job of shielding the C programmer from having to determine if an I/O pin is an input or output, and writing code to that effect. Since we initially defined our H-bridge control pins using the PIC Wizard, there's no more I/O direction code that needs to be added for the application to run as designed. Here's one of those times that the C programmer has the desire to have control over a built-in compiler feature. I would feel better if I specified the way the PIC's H-bridge I/O pins operate. My first override is to tell the compiler that all I/O operations on PORTB will be fast:

```
#use fast_io(B)
```

I've just instructed the CCS compiler not to set the I/O direction before each I/O operation. With the compiler clear of generating I/O direction code, the programmer assumes the responsibility of setting the I/O port pin direction. We set RB0 and RB1 as output pins using this built-in compiler function:

```
set_tris_b(0b11111100); //RB0 and RB1 outputs - all other pins are inputs
```

Okay. Now we can turn our attention to writing the H-bridge driver code. We'll take advantage of our compiler's *output_bit* function. The *output_bit* function arguments consist of the desired port pin followed by the logic level to be driven out of the selected port pin. For instance, output_bit(RB0,1) will force I/O pin RB0 logically high. I don't think you'll have any problem deciphering this code snippet:

```
#define ACT_EXTEND           output_bit(ACT_DIR,1);   \
                             output_bit(ACT_PWM,1);

#define ACT_RETRACT          output_bit(ACT_DIR,0);   \
                             output_bit(ACT_PWM,1);

#define ACT_STOP             output_bit(ACT_PWM,0)
```

ACT_EXTEND and ACT_RETRACT are C macros we created using the C language *#define* command. Multiple lines of C source that make up the C macros are separated by the "\" character. The ACT_STOP definition is a simple text substitution that is called statement shorthand. The idea behind the C macros and the statement shorthand is to allow the C programmer to use human-readable logical definitions (instead of keying in the actual lines of C source code represented by the macros and shorthand into the main C source).

All of the supporting elements are in place. So, let's test our H-bridge driver code by putting together a simple application that extends, retracts, and stops the actuator in timed intervals. The CCS C compiler has a built-in millisecond delay function that we can call on to produce the timed intervals. Here's a look at the main C source:

```
#include "C:\servo-ccs-firgelli\firgelli-ccs.h"

#use fast_io(B)

#define ACT_DIR       PIN_B0
```

SOURCES

Firgelli
Firgelli L12 Linear Actuator
www.firgelli.com

National Semiconductor
LMD18200T H-bridge
www.national.com

CCS
CCS C Compiler
www.ccsinfo.com

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

```
#define ACT_PWM       PIN_B1

#define ACT_EXTEND          output_bit(ACT_DIR,1);   \
                            output_bit(ACT_PWM,1);

#define ACT_RETRACT         output_bit(ACT_DIR,0);   \
                            output_bit(ACT_PWM,1);

#define ACT_STOP            output_bit(ACT_PWM,0)

void main()
{
   setup_adc_ports(NO_ANALOGS|VSS_VDD);
   setup_adc(ADC_OFF|ADC_TAD_MUL_0);
   setup_spi(SPI_SS_DISABLED);
   setup_wdt(WDT_OFF);
   setup_timer_0(RTCC_OFF);
   setup_timer_1(T1_DISABLED);
   setup_timer_2(T2_DISABLED,0,1);
   setup_comparator(NC_NC_NC_NC);
   setup_vref(FALSE);

   set_tris_b(0b11111100);  //set RB0 and RB1 as outputs in binary

   ACT_RETRACT;             //make sure actuator is retracted
   delay_ms(3000);          //retract for 3 seconds
   ACT_STOP;                //stop the actuator

   do{
      ACT_EXTEND;           //extend the actuator
      delay_ms(2000);       //extend for 2 seconds
      ACT_STOP;             //stop the actuator
      delay_ms(2000);       //stop for 2 seconds
      ACT_RETRACT;          //retract the actuator
      delay_ms(2000);       //retract for 2 seconds
       ACT_STOP;            //stop the actuator
      delay_ms(2000);       //stop for 2 seconds
  }while(1);
}
```

I don't think you're having any trouble following the logic of the Firgelli linear actuator driver code we've assembled. Once all of the housekeeping is done, our C actuator driver makes sure that the actuator begins in a retracted position. The C code contained within the do/while loop executes forever as *while(1)* is always TRUE. The actuator will extend for two seconds, stop for two seconds, retract for two seconds, and stop again for two seconds before repeating the sequence.

C(ing) the Light

The real beauty of our C project is that anyone can cold-read our C source code and immediately know what the application is doing. Our liberal use of *#define* commands, statement shorthand, and C macros makes our Firgelli linear actuator driver application easy to read and understand. The code we've written is almost self-commenting. Who says C is hard to learn? **SV**

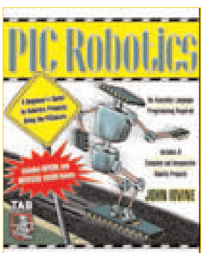
The *SERVO* Webstore

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

CD-ROM SPECIALS

ROBOTICS

PIC Robotics
by John Iovine

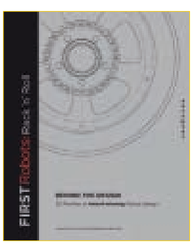


Here's everything the robotics hobbyist needs to harness the power of the PICMicro MCU!

In this heavily-illustrated resource, author John Iovine provides plans and complete parts lists for 11 easy-to-build robots each with a PICMicro "brain." The expertly written coverage of the PIC Basic Computer makes programming a snap – and lots of fun.
$24.95

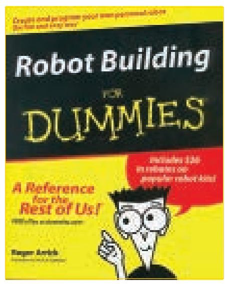
FIRST Robots: Rack 'N' Roll: Behind the Design
by Vince Wilczynski, Stephanie Slezycki

More than 750 photographs!



The second annual book highlighting the creativity and process behind 30 winning robot designs from the 18th annual international FIRST Robotics Competition. The FIRST organization, founded by Dean Kamen (inventor of the Segway), promotes education in the sciences, technology, and engineering.
Reg $39.95

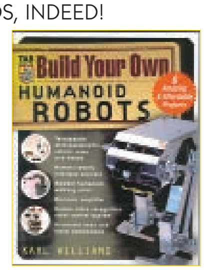
Robot Building for Dummies
by Roger Arrick / Nancy Stevenson



Discover what robots can do and how they work. Find out how to build your own robot and program it to perform tasks. 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.
$24.95

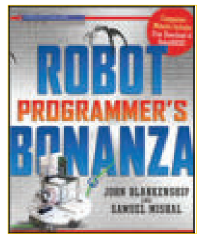
Build Your Own Humanoid Robots
by Karl Williams

GREAT 'DROIDS, INDEED!



This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot. **$24.95***

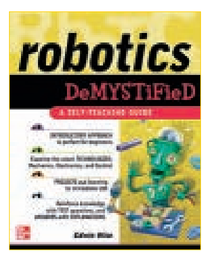
Robot Programmer's Bonanza
by John Blankenship, Samuel Mishal



The first hands-on programming guide for today's robot hobbyist!
Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.
$29.95

Robotics Demystified
by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!

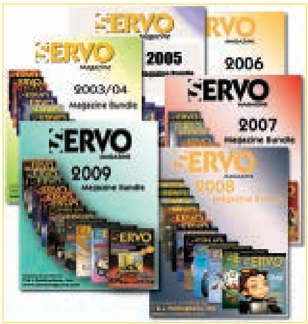


Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots!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

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

To order call 1-800-783-4624

SERVO Magazine Bundles



Published by T & L Publications, Inc.

$57 per bundle

Save $10 off the normal price!!

Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, and 09.

Kickin' Bot

by Grant Imahara

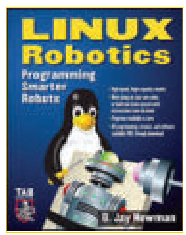
Enter the arena of the metal gladiators!

Do you have what it takes to build a battle-ready robot? You do now! Here are the plans, step-by-step directions, and expert advice that will put you in competition — while you have a heck of a lot of fun getting there. Grant Imahara, the creator of the popular BattleBot Deadblow, shares everything he's learned about robot design, tools, and techniques for metal working and the parts you need and where to get them.

$24.95

Linux Robotics

by D. Jay Newman

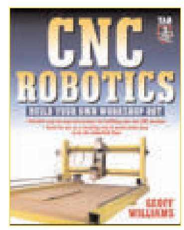


If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any 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.

$34.95

CNC Robotics

by Geoff Williams

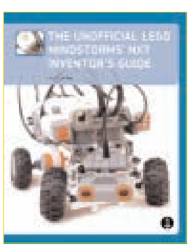


Here's the FIRST book to offer step-by-step guidelines that walk the reader through the entire process of building a CNC (Computer Numerical Control) machine from start to finish. Using inexpensive, off-the-shelf parts, readers can build CNC machines with true industrial shop applications such as machining, routing, and cutting — at a fraction of what it would cost to purchase one. Great for anyone who wants to automate a task in their home shop or small business. **$34.95**

SPECIAL OFFERS

The Unofficial LEGO MINDSTORMS NXT Inventor's Guide

by David J. Perdue



This book was written for the first version of the NXT set (#8527), and its projects are only compatible with the first version. In other words, because of piece differences between the NXT 1.0 and 2.0 sets, the projects in this book can only be built with an NXT 1.0 set. However, much of the other information is still helpful, and the building, mechanical, and programming details are still applicable.

Reg $29.95

Sale Price $25.95

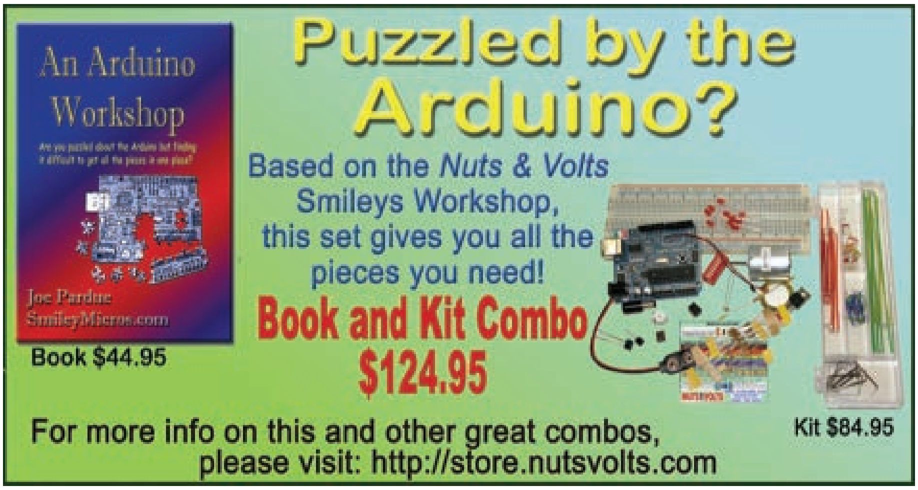
Technology Education Package for Everyone Starting in Electronics

This lab – from the good people at GSS Tech Ed – will show you 40 of the most simple and interesting experiments and lessons you have ever seen on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit. Along with the purchase of the lab, you will receive a special password to access the fantastic online interactive software to help you fully understand all the electronic principles. For a complete product description and sample software, please visit our webstore.

Regular Price $79.95 **Subscriber's Price $75.95**

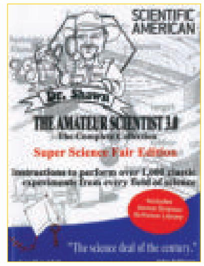
Or order online www.servomagazine.com

SPECIAL OFFERS



The Amateur Scientist 3.0 The Complete Collection

by Bright Science, LLC

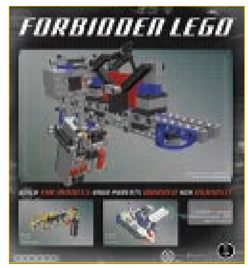


There are 1,000 projects on this **CD**, not to mention the additional technical info and bonus features. It doesn't matter if you're a complete novice looking to do their first science fair project or a super tech-head gadget freak; there are enough projects on the single CD-ROM to keep you and 50 of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

Reg $26.95 Sale Price $23.95

Forbidden LEGO

by Ulrik Pilegaard / Mike Dooley



Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse. **Reg $24.95 Sale Price $19.95**

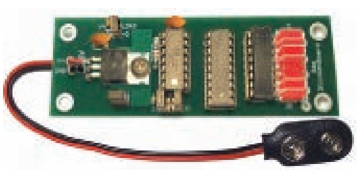
PROJECTS

Das Blinkenboard Kit

As seen in *Nuts & Volts* June issue Personal Robotics by Vern Graner

This kit includes a pre-programmed ATtiny84 microcontroller that sports eight software PWM channels to control motor speed and light brightness. Jumper selectable patterns can be used to operate motors, solenoid valves, relays, or any DC load up to 24V/500 mA per channel!

Expand your board with the "Der Magnetfelder Detektor" component pack.



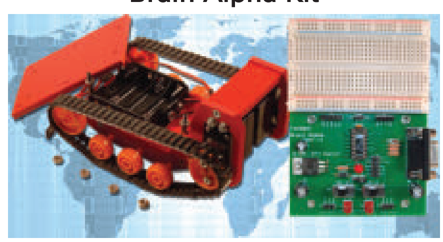
Subscriber's Price **$32.45**
Non-Subscriber's Price **$35.95**
PCBs can be bought separately.

16-Bit Micro Experimenter Board

Ready to move on from eight-bit to 16-bit microcontrollers? Well, you're in luck! In the December 2009 *Nuts & Volts* issue, you're introduced to the 16-Bit Micro Experimenter. The kit comes with a CD-ROM that contains details on assembly, operation, as well as an assortment of ready-made applications. New applications will be added in upcoming months.

Subscriber's Price **$55.95**
Non-Subscriber's Price **$59.95**

Tankbot Kit & Brain Alpha Kit



Tankbot/Brain Alpha originally by Ron Hackett
Now with New Columnist Calvin Turzillo
A series filled with projects and experiments to challenge you through your learning process while you grow your fully expandable Brain Alpha PCB!
The brain is a PICAXE-14A!

For more info & pictures, visit the *SERVO* Webstore.
Tankbot and the Brain Alpha Kit can be purchased separately.

Combo Price $ 138.95

Then and NOW

ROBOT APPLICATIONS

by Tom Carroll

Robots have come a long way since the first industrial robots of the early '60s. George C. Devol obtained a patent in 1954 for 'a programmable method for transferring articles between different parts of a factory.' Later, he met up with a younger engineer, Joseph Engelberger; they observed that '50 percent of the people who work in factories are really putting and taking.' After wondering, "Why are machines made to produce only specific items?" They asked, "How about approaching manufacturing the other way around, by designing machines that could put and take anything." Thus was born the first industrial robot that could pick up something at one location in a manufacturing operation and transfer it to another location. Robotics had now become a new branch of manufacturing. This was the sole robot application for a while, but manufacturers quickly saw ways that these new machines could be used in other ways. Since that original Unimate robot had a limited semi spherical range of motion, manufacturers soon designed robots with the more familiar articulated arm configuration to perform more tasks.

Robotics in Modern House Construction

There are many new houses still being built, and I have always enjoyed watching this building process. For example, one of the most interesting ways of pouring concrete for foundations, patios, and walkways is the use of a concrete pumper that reaches out and places the wet concrete virtually any place on the lot. Instead of the traditional method where a cement mixer truck deposits small amounts of concrete into wheel barrows that are then pushed to the places where they're needed, the truck delivers many yards of mix into a hopper on the pumper truck that is then pumped to where it's needed. **Figure 1** shows a remotely-controlled pumper's boom. Note the man with the wireless remote control. Moving one of two joysticks, the operator easily moves the hose at the end of the boom to any place where cement needs pouring. As I've watched several of these operations, it's quite obvious that the pumper's arm is — in reality — a huge multi-axis robot arm.

FIGURE 1. Concrete pumper remote control.

At another building site, I saw another type of crane system (**Figure 2**) deftly scooping up about 500 pounds of drywall sheets and inserting them into an upstairs window of a new house — all under control by a person with a wireless RC panel similar to the concrete pumper. The operator scoops up a bundle of drywall sheets with an end-effector similar to a forklift, but the forks move up vertically as the drywall is presented at the window where a person easily removes each sheet. As anyone who's done any drywall installations knows, these sheets are not only very heavy but are extremely fragile if they are dropped or even bumped into a sharp object. Carrying them up a partially completed stairwell one at a time is a dangerous and tiring task that used to be 'part of the job' just a few years ago, but modern robotic technology has taken over the hardest part of a drywall installation job.

FIGURE 2. Drywall boom crane.

At another house under construction, I saw a roofing shingle truck delivering bundles of shingles to the peak of the house via a long conveyor belt system. A person on the truck would place an eight pound bundle of shingles at the bottom of the inclined conveyor and a person at the peak would remove them one at a time and stack them into position. It wasn't too long ago when roofers would carry a heavy bundle over their shoulder as they climbed a ladder to the roof. When I talked to the guy at the bottom of the conveyor, he told me that he had heard that a remotely-controlled conveyor system was 'in the works' where a single person on the ground could operate it to place bundles all across the peak, using just a single RC panel.

Roofing and drywall workers have always been at the top of the list of serious on-the-job injuries. Decades ago, these robotic applications didn't exist, though I am sure that workers in the various construction categories wished for some sophisticated machines (like a robot) to make their jobs easier and safer. Microcontroller technology coupled with powerful motor controllers and inexpensive and capable wireless modules have allowed robotic technology to be moved from the factory floor to other locations for numerous applications.

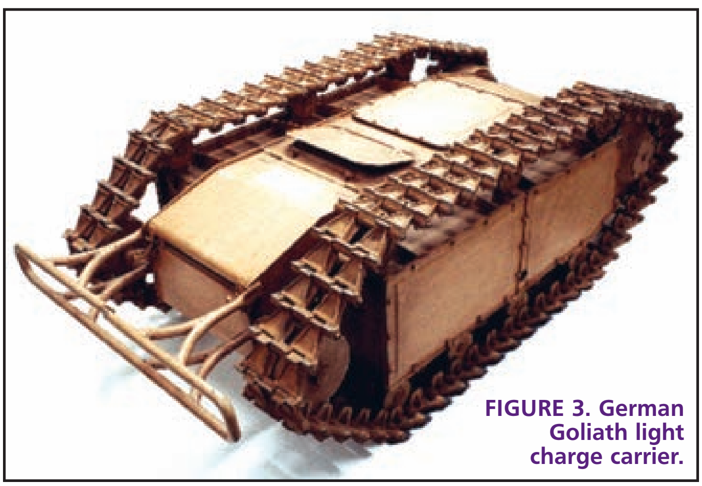


FIGURE 3. German Goliath light charge carrier.

Industrial Robot Applications

While watching these basic construction jobs, I got to thinking about all the other applications that robotics technology has encompassed since its beginnings — from commercial, agricultural, military, security, space, R&D platforms, oceanographic, educational, service, personal, medical, and industrial robots. However, it is the industrial robot category that seems to have the widest variety of applications. This stands to reason as these robots were the first commercial robots sold and manufacturing operations subsequently found numerous applications within the factory environment.

Dedicated CNC machining centers are faster and more efficient for machining *individual* intricate metal parts, but robots can handle a vast array of manufacturing tasks within the factory environment outside of basic machining. These tasks can include, among others: spot welding,TIG, MIG, wire welding, resistance welding, paint spraying, parts transfer, pick and place, machine loading, palletizing, dangerous parts handling, deburring, torch cutting of metal, circuit board assembly, glass handling, order picking, and food/product packaging, plus numerous other categories and sub categories of robotics in manufacturing.

Robots and factory automation go hand-in-hand with automatic conveyor systems, vision systems, dedicated machining centers, and warehouse systems to create very sophisticated robot applications within the industry. Virtually every robot company offers something for industrial applications. Modern industry the world over has found that it must implement robots in their manufacturing processes to stay competitive. There are basically four possibilities for completing any factory task that engineers can consider: humans, robots, dedicated automation, or a combination of all three. They must also look at the Four Ds of Robotics: *"Is the task dangerous, dirty, difficult, or dull?"* Paint spraying is a good example of all four of these considerations and was quickly implemented into automobile assembly lines around the world; same with spot welding of car bodies.

Service Robot Applications

Service robots didn't exist at the beginning of the industry, but inventive robot designers soon discovered that the basic advantages to robotic technology existed outside the manufacturing sector. The Four Ds apply to *all* areas being considered for using robots, not just industrial applications. Service robot applications are usually mobile; a gray area of this category might be AGVs (automated guided vehicles) that move parts around a factory floor.

Most of the noteworthy robots in the service category are in military applications, especially in these days of war in the Middle East. Typical military tasks call for a mobile robot with some sort of arm or gripper, and television feedback to a remote operator. Long before iRobot came on the scene with their Packbots, the Germans used wire-controlled 'tanks' that were actually bombs with tracks to destroy enemy tanks and vehicles. This gas engine powered tank proved to be ineffective, but unique nonetheless, as its control cable could be easily cut or shot through. **Figure 3** shows the Goliath light charge carrier that could carry up to a 100 KG explosive charge.

We usually think of robot arms used in industrial applications within factories. **Figure 4** shows a unique, but rapidly-growing use of robots: in farming, agriculture, and, in this case, sorting and packaging produce. This plant in Murcia, Spain uses 69 Fanuc robots to process over 400,000 heads of lettuce a day — greatly increasing production over the previous manual methods. An outside robot integrator company studied the needs and developed a vision system with a series of robots. The process starts where a robot empties the trays of picked lettuce heads into a conveyor where other robots position each head and spread them evenly so they enter the cutting station to have the root part removed. The vision system not only reads their position, but their size so heads that are too large or too small are rejected. The robot again inspects the heads under another camera and the lettuce is packed in trays for shipment. Personnel have been reduced by 80%, and accuracy and cleanliness has increased with this unique robotic application in place.

FIGURE 4. Lettuce sorting robots in Spain.

Other Service Robot Applications

Other agricultural applications include automated pickers and reapers, combines, animal feeders, crop care, and pest control.

Other military uses include autonomous and remote-controlled weapons platforms, intelligent bombs and torpedoes, autonomous and R/C flying vehicles, autonomous and R/C ground surveillance vehicles, intelligent gun platforms, and weapons handling systems.

Cleaning uses include home vacuums (Roomba style), window cleaning, hard floor cleaning, swimming pool sweeping, gutter cleaning, grass cutting, leaf collecting, and also clean room maintenance.

Inspection robots can be used to visually inspect pipelines, sewers, human-inaccessible areas in buildings, rubble from disasters, and other remote sites such as damaged undersea oil wells like we're experiencing in the Gulf.

Oceanographic ROVs are used to explore extreme depths, retrieve plant, animal, water and geological samples, and photograph remote sites.

Security uses include both civilian (police and home) and military surveillance. Civilian and government agencies use both surface and aerial robots to observe border areas, with police agencies using robots for bomb verification and disposal, suspect observations, and in disaster operations.

Medical robotics applications can be as small as complete robots that enter the body to retrieve tissue samples or dispense medications, or as complex as the da Vinci HD surgical system robot made by Intuitive Surgical and shown in **Figure 5**. This system allows surgeons to perform intricate and minimally-invasive operations by remote control. The surgeon is sitting at the console to the left and can perform delicate laparoscopic surgeries with full HD vision and multiple surgical tools. Patients recover much faster and there is little blood loss. Doctors in a hospital a continent away can actually operate on an injured soldier in a hospital tent next to a battle area — all by remote control. The newest da Vinci system has four arms with seven degrees of freedom.

Personal Robot Applications

This category interests most robot experimenters as it entails robots within the home and is a sub-category of service robots. The previously mentioned iRobot Roomba has been very popular with robot enthusiasts, as well as typical home owners, and has sold millions around the world.

FIGURE 5. da Vinci surgical robot.

However, it is the more versatile home-built mobile platforms with various appendages and sensors that pique our interests. iRobot realized this when they saw that many perfectly good Roombas were being hacked to bits. So, they came out with their Create platform shown in **Figure 6.** The Create platform features 32 built-in sensors, a cargo bay showing the optional green Atmel ATMega 168 microcontroller command module installed, and a 25-pin expansion port for additional hardware.

FIGURE 6. iRobot Create experimenter's platform.

I have found the Create to be an excellent experimenter's platform. It comes ready-made, rather than as a kit. The Parallax BASIC Stamp Boe-Bot is another popular robotics platform, as are the VEX and LEGO NTX kit series. Factory implementations of industrial robots may be fine for the manufacturing engineer, but robot hobbyists want their creations close and personal in their homes.

Aside from what's currently available on the market and events such as maze solving, RoboMegellan, MicroMouse, line following, Sumo, and combat robots, most experimenters want a basic home robot that can serve them in some way. Fetching items seems to be at the top of the list, as in "Bring me a Pepsi or Coors Light." A more serious application may be a robot to assist the elderly in independent living — a category that has been very close to my heart for a number of years. Household tasks such as entertaining guests, watering plants, household security and monitoring, home systems management, and basic house cleaning, are all desirable. Sometimes, a robot may be desired as a companion, just like the old Sony Aibo or Paro the seal.

The Versatility of Robots

I've just briefly touched on some of the many hundreds of applications that robots can perform. It is the versatility of these machines that make them so appealing to experimenters and hobbyists. In homes, factories, on battlefields, in space, roving on the surface of planets, operating on patients, undersea, or in the air, robots are amazing machines. Television programs (such as *MythBusters)* and science shows have introduced viewers to unique mechanical solutions that solve or prove (or disprove) varied concepts or problems bringing more awareness to robotic technology.

I was recently invited to Willow Garage in Menlo Park, CA to give a talk on the use of robots to assist the elderly in independent living and to view their facility and the amazing PR2 robot we've discussed in previous columns. **Figure 7** shows the robot projecting a pattern on the wall in order to locate and identify a power outlet for re-charging. Founded by Scott Hassan (best known for his eGroups start-up) and run by President and CEO, Steve Cousins (who previously was a manager at Xerox's PARC), the company's recently offered 11 PR2 robots to qualified researchers at some of the world's top institutions and companies to encourage robot-specific applications.

FIGURE 7. Willow Garage's PR2 searching for a power outlet.

The PR2 is not just your everyday robot but a highly sophisticated and capable machine. The robot has unlimited application possibilities. As Steve says: *"Willow Garage is organized to encourage spin-out companies. We intend for spin-outs to participate in the industry, enabled by projects developed at Willow Garage. These spin-out projects should have broad applications that deliver significant contributions to the robotics community and beyond."*

Scott and Steve's generous attitude towards the robot community is also shown in their development of ROS, their software platform which stands for Robot Operating System (much like a computer operating system). The software is completely open source and free for others to use.

Final Thoughts

I've discussed just a few of the many categories and applications for robots, and robot experimenters will certainly be coming up with many more in the future. It is this versatility that makes our machines and our hobby so interesting. **SV**

Tom Carroll can be reached at TWCarroll@aol.com.

ROBO-LINKS

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

To Advertise: Call 951-371-8497

Advertiser Index