

NUTS AND VOLTS

www.nutsvolts.com
April 2013

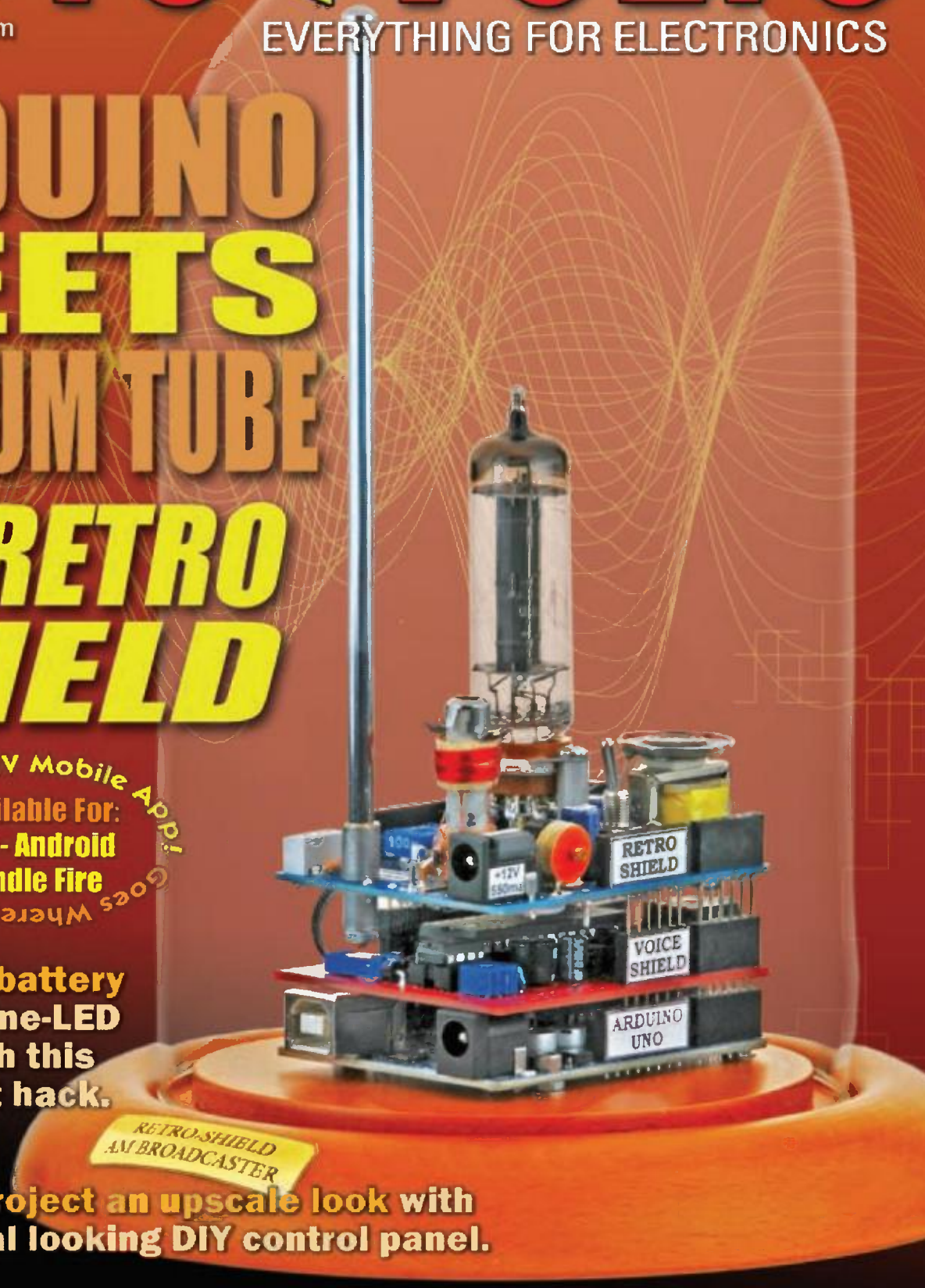
EVERYTHING FOR ELECTRONICS

ARDUINO MEETS VACUUM TUBE THE RETRO SHIELD

The New NV Mobile App!
Available For:
iOS - Android
Kindle Fire
Goes Where You Go

◆ **Extend the battery life of your nine-LED flashlight with this simple circuit hack.**

◆ **Give your project an upscale look with a professional looking DIY control panel.**



Easy-to-use Serial to Ethernet Solutions

➤ SBL2e 100

2-Port 3.3V TTL serial device support with analog to digital converters & digital I/O
Part No. SBL2E-100IR: \$21.95 Qty. 1K

➤ SBL2e CHIP (80-pin LQFP)

2-Port 3.3V TTL serial device support with analog to digital converters & digital I/O
Part No. SBL2ECHIP-236IR: \$12.50 Qty. 1K

For as low as
\$12⁵⁰

➤ SBL2e X

1-Port RS-232 serial device support
Part No. SBL2EX-100IR: \$79.00 Qty. 1K

➤ SBL2e XA

1-Port RS-232 and 3.3V TTL serial device support with analog to digital converters & digital I/O
Part No. SBL2EXA-100IR: \$79.00 Qty. 1K

Access serial, analog, or digital I/O data over Ethernet!



The goal:

Control, configure, or monitor digital I/O, analog inputs, or a serial device using Ethernet



The method:

Connect serial port, digital I/O, or analog input to a SBL2e device



The result:

Access device from the Internet or a local area network (LAN)

NetBurner SBL2e Devices network-enable serial, analog, or digital I/O devices out of the box - no programming or development is required. The included virtual COM port Windows utility enables you to easily access serial devices located on a remote NetBurner device as if they were plugged into your own computer. The hardware is pre-programmed to convert your serial data to Ethernet, enabling communication with the serial device over a network or the Internet.

Need a custom solution?

NetBurner Development Kits are available to customize any aspect of operation including web pages, data filtering, or custom network applications.

➤ **Development Kit for SBL2e**
Part No. NNDK-SBL2E-KIT: \$299.00

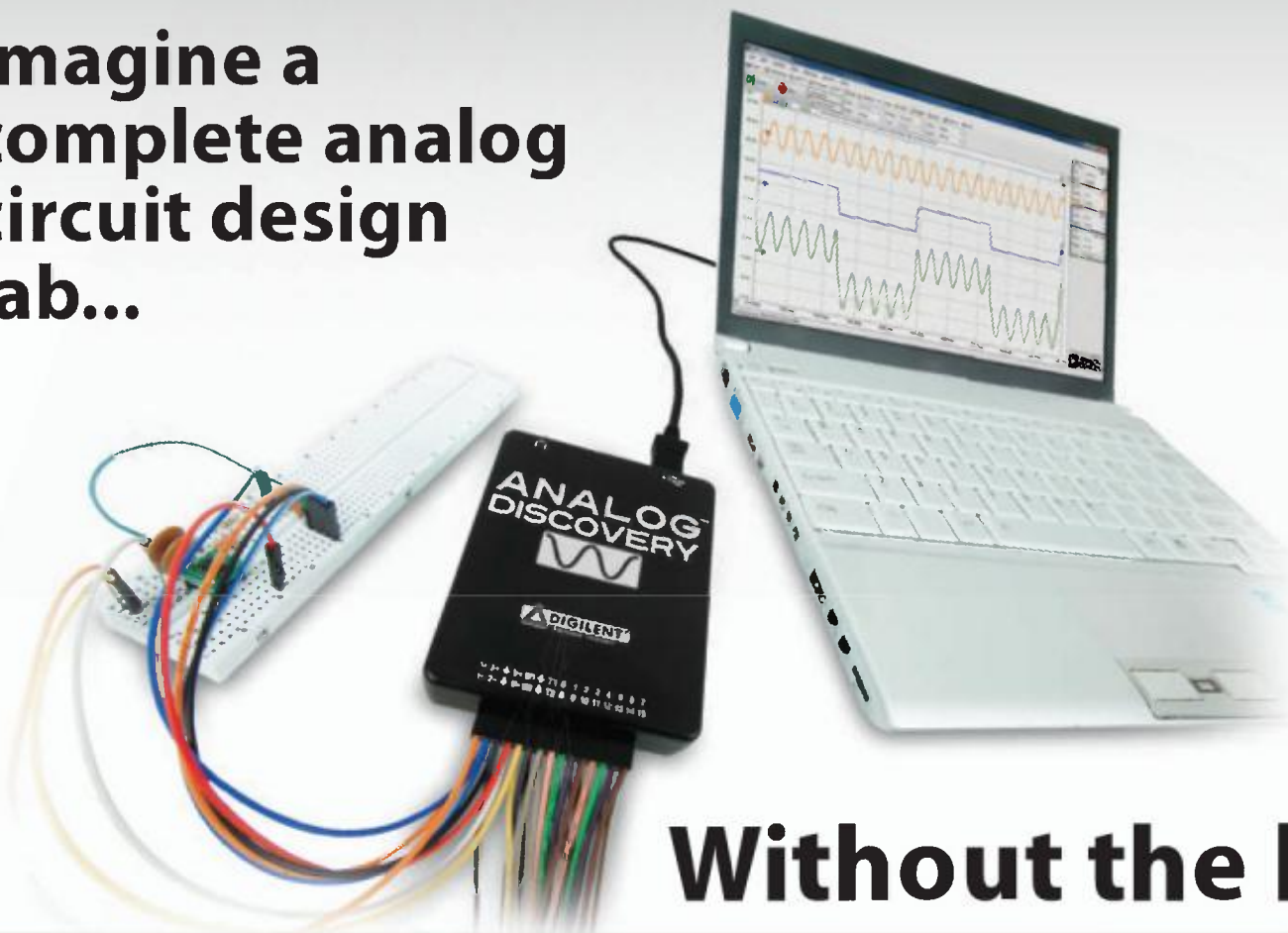


www.mikroe.com

Every embedded developer knows at least one of these words:
mikroC, mikroBasic, mikroPascal, Visual TFT, Visual GLCD, EasyPIC,
EasyPIC PRO, EasyPIC Fusion, EasyMx PRO, EasyAVR, Easy8051,
mikromedia, click, MINI, mikroBUS, mikroProg, Libstock...

How about you?

Imagine a complete analog circuit design lab...



Without the lab!

Dual Channel Oscilloscope

- Fully differential inputs, up to 100 MSPS, 14-bit ADC, up to 16KSa/channel memory
- 1M Ω /24pF inputs, $\pm 20V$ Max
- Real-time FFTs, X-Y plots, complex math and measurements on all channels
- 5 MHz analog signal bandwidth

Dual channel Function Generator

- Single-ended outputs, up to 100 MSPS, 14-bit D/A, up to 16KSa/channel memory
- Arbitrary and pre-defined waves up to $\pm 5V$
- Sweeps, envelopes, AM and FM modulation
- Bode plot feature using standard, Nyquist & Nichols coordinates
- Can drive 50 Ω

Digital I/O

- 16 signals, 100MSPS, 4KSa/channel memory
- Any signal can be input (logic analyzer) or output (pattern generator or virtual I/O)
- Supports cross-triggers with scope channels

Power Supplies

- Two fixed $\pm 5V$, 50mA



Turn your PC into a powerful circuit learning platform that can measure, visualize, analyze, record and control mixed signal circuits! The Analog Discovery is small enough to fit in your pocket and costs less than a textbook, but it is powerful enough to replace a stack of lab equipment. Powered by a Hi-Speed USB port and the free WaveForms software, the Analog Discovery lets you build and test analog and digital circuits outside of the lab.

\$199.00
(List price)

For courseware & more information, please visit:
www.digilentinc.com/AnalogDiscovery

Designed in cooperation with:



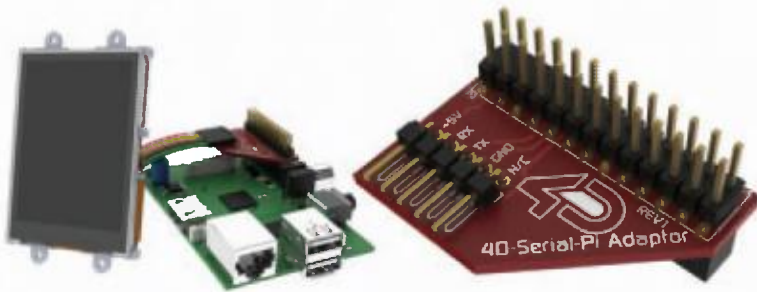
Intelligent Display Modules

4D Systems designs and manufactures intelligent and cost effective display modules. Our intelligent display modules offer a simple yet effective interface to any host processor, and our comprehensive workshop software suite is suitable for almost any level of developer.

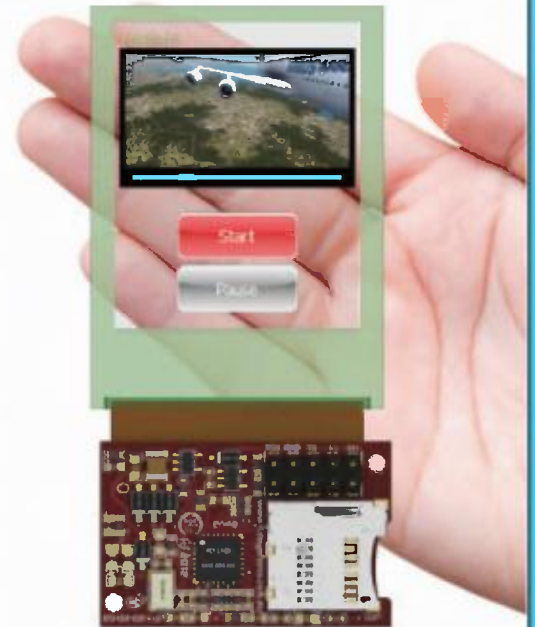
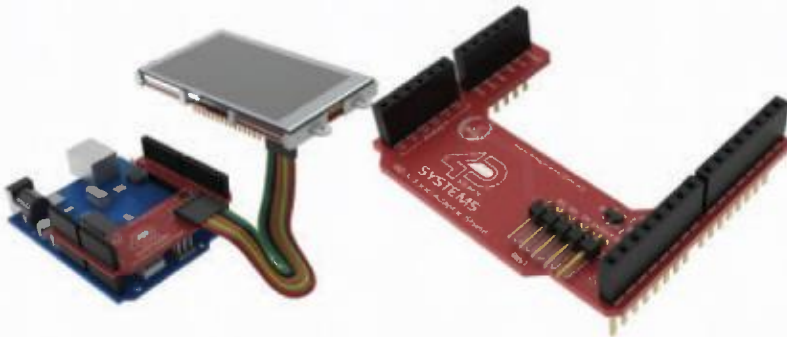
- ✔ Application Ready in Minutes
- ✔ Free Software Tools and Support
- ✔ Easy to Program
- ✔ Plug & Play Experience

Display Solutions for your favorite Embedded Platforms

All new **Raspberry Pi*** compatible display modules



All new **Arduino*** Shield compatible display modules



All new **TRANSPARENT OLED**
perfect for Head Up Displays

All that and many products you
can browse on our website :

www.4dsystems.com.au
TEL : +61 2 4721 7786



April 2013

Projects & Features

30 The Retro-Shield: Where the Past Meets the Present

Combine a vintage vacuum tube with an Arduino for an updated version of an AM broadcaster.

■ By David Goodsell

38 Hacking the Nine-LED Flashlight

The simple circuit in these now common flashlights actually eats through batteries pretty fast. This hack adapts the flashlight to use an external power source, and adds a pulsed driver circuit for better efficiency.

■ By Dan Gravatt



Page 38

43 How to Make Professional Looking Control Panels and Enclosures — Part 1

Don't you hate it when you've worked really hard on a project, only to have the finished product look like a six year old put it together? This tutorial will show you a cheap and easy way to make great looking panels and enclosures when you aren't able to utilize professional services.

■ By Gordon McComb

Columns

10 TechKnowledge 2013

Events, Advances, and News

Tapping into tap water, a super "duper," and core memory's 60th anniversary are just a few of the things you'll read about this time.

14 PICAXE Primer

Sharpening Your Tools of Creativity

Experimenting With Parallel Task Processing. All M2-class processors include this new software feature that allows them to simulate a multi-core processor by rapidly switching between two or more separate tasks.

21 Q & A

Reader Questions Answered Here

Topics include a pet trainer, bandpass filters, a push on/push off circuit, a DC servo, a voltage reducer, plus Mailbag.

54 Smiley's Workshop

Programming • Hardware • Projects

The Arduino Proto Shield Alarm Clock — continued some more.

We're winding down our discussion on how to use Fritzing to take an Arduino design from prototype to production.

60 The Design Cycle

Advanced Techniques for Design Engineers

Walking the USB Bridge into Androidville. FTDI gave us the power to easily embed USB functionality into our projects with the FT232RL. The FT311D USB Android host IC enables us to walk across the USB bridge that FTDI built into Androidville.

68 Open Communication

The Latest in Networking and Wireless Technologies

60 GHz Wireless Transports HDTV. Transmitting uncompressed digital high definition television signals is no easy task. Learn about the technologies that are currently being used to accomplish these transmissions.

Departments

08	DEVELOPING PERSPECTIVES	72	NV WEBSTORE
26	NEW PRODUCTS	75	CLASSIFIEDS
27	SHOWCASE	78	TECH FORUM
67	ELECTRO-NET	80	AD INDEX

Data Hold DMM with Backlight

A handy digital multimeter with lots of features at a great price. Has all the features you would expect including holster and probes included. See online for more info.

- 2000 count
- DC & AC voltage
- Transistor & Diode test
- 10A DC current
- Size: 145(L) x 65(W) x 35(H)mm

Cat. QM-1523

\$9.50*



Digital Multimeter Kit

Learn everything there is to know about component recognition and basic electronics with this comprehensive kit. From test leads to solder, everything you need for the construction of this meter is included.

- Size: 67(W) x 123(H) x 25(D)mm

Cat. KG-9250

\$15.75*



USB Power Monitor Kit

Plug this kit inline with a USB device to display the current that is drawn at any given time. Check the total power draw from an unpowered hub and its attached devices or what impact a USB device has on your laptop battery life. Displays current, voltage or power, is auto-ranging and will read as low as a few microamps and up to over an amp. Kit supplied with double sided, soldermasked and screen-printed PCB with SMD components presoldered, LCD screen, and components.

- PCB: 65 x 36mm

Cat. KC-5516

\$43.25*

Laptop not included



Jacob's Ladder MK3 Kit

A spectacular rising ladder of bright and noisy sparks for theatre special effects or to impress your friends. This improved circuit has even more zing and zap than it's previous design from April 2007 and requires the purchase of a 12V ignition coil (available from auto stores and parts recyclers).

- Kit supplied with silk-screened PCB, diecast enclosure (111 x 60 x 30mm), pre-programmed PIC, PCB mount components and pre-cut wire/ladder.
- Powered from a 12V 7Ah SLA or 12V car battery.

Cat. KC-5520

\$36.00*

Battery not included



Low Capacitance Adaptor for DMM Kit

Many modern multimeters come with capacitance ranges, but they're no good for very small values. This kit is a nifty little adaptor that allows a standard digital multimeter to measure very low values of capacitance from less than one picofarad to over 10nF. It will allow you to measure tiny capacitors or stray capacitances in switches, connectors and wiring. The kit is complete with PCB, components and case. All you'll need is a 9V battery and just about any modern DMM.

- PCB: 51 x 90mm

Cat. KC-5493

\$22.00*



High Energy Ignition Kit for Cars

Use this kit to replace a failed ignition module or to upgrade a mechanical ignition system when restoring a vehicle. Use with virtually any ignition system that uses a single coil with points, hall effect/lumenition, reluctor or optical sensors (Crane and Piranha) and ECU. Features include adjustable dwell time, output or follow input option, tachometer output, adjustable debounce period, dwell compensation for battery voltage and coil switch-off with no trigger signal.

- Kit supplied with silk-screened PCB, diecast enclosure (111 x 60 x 30mm), pre-programmed PIC and PCB mount components for four trigger/pickup options

Cat. KC-5513

\$36.00*



Transistor Tester Kit

Have you ever unsoldered a suspect transistor only to find that it checks OK? Troubleshooting exercises are often hindered by this type of false alarm. You can avoid these hassles with the In-Circuit Transistor, SCR and Diode Tester. The kit does just that, test drives WITHOUT the need to unsolder them from the circuit! VERY HANDY!

Cat. KA-1119

\$17.50*



DC Relay Switch Kit

An extremely useful and versatile kit that enables you to use a tiny trigger current - as low as 400µA at 12V to switch up to 30A at 50VDC. It has an isolated input and is suitable for a variety of triggering options. Kit includes PCB with overlay and all electronic components with clear instructions.

Cat. KC-5434

\$12.25*



12/24VDC 20A Motor Speed

Controller Kit

Control the speed of 12 or 24VDC motors from zero to full power, up to 20A. Features optional soft start, adjustable pulse frequency to reduce motor noise, and low battery protection. The speed is set using the onboard trimpot, or by using an external potentiometer (available separately, use RP-3510 \$1.50).

- Kit supplied with PCB and all onboard electronic components
- Suitable enclosure UB3 case, HB-6013 \$2.50 sold separately

Cat. KC-5502

\$29.00*



'The Champion' Audio Amplifier Kit with Pre-Amplifier

Suitable for general-purpose audio projects and supports microphone and electric guitar input. It uses the AN7511 audio IC to deliver 2W music power into 8 ohms from a 9 to 12V supply. Features low distortion, two inputs (mixed 1:1), mute and standby control. Power from 4 - 13.5VDC. See website for specifications.

- Kit supplied with silk-screened PCB, heatsink and PCB mount components
- PCB: 101 x 41mm

Cat. KC-5519

\$14.50*



HOW TO ORDER

PHONE: 1800 784 0263*
FAX: +61 2 8832 3118*
EMAIL: techstore@jaycar.com
POST: P.O. Box 107, Rydalmere NSW 2116 Australia

*Australian Eastern Standard Time (Monday - Friday 09.00 to 17.30 GMT + 11 hours)

*US Eastern Standard Time (Monday - Friday 3.00pm to 11.30pm)

*Prices valid until 30/04/2013

*ALL PRICES IN USD & EXCLUDE POSTAGE & PACKING

NOW SHIPPING VIA DHL

5 - 10 day working delivery

• FAST DELIVERY • TRACK SHIPMENT

Note: Products are dispatched from Australia, so local customs duty & taxes may apply.



Arduino Corner

Arduino Experimenters Kit

Servo motor, lights, buttons, switches, sound, sensors, breadboard, wires and more are included with a Freetronics Eleven Arduino compatible board in this extensive hobby experimenter and starter kit.

\$65.00*

- Comprehensive instructions included
- Size: 340(W) x 165(H) x 36(D)mm

Cat. XC-4262

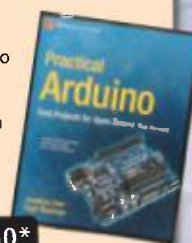
Practical Arduino

Describes 14 practical Arduino projects. It takes you beyond basics quite quickly and shows you how to make up a typical application/design.

- Soft cover, 422 pages. 235 x 190mm

Cat. BM-7132

\$38.50*





by Bryan Bergeron, Editor

DEVELOPING PERSPECTIVES

Kickstart Your Project

Okay. You've invented the perfect mouse control device. Let's say it senses a rodent's heat signature and sterilizes the critter — otherwise unharmed — with a blast of radiation from a cavity magnetron. Your prototype is based on a repurposed microwave oven, a PIR sensor, and an Arduino microcontroller which seems to do the job. If only you had, say, \$50K for parts and printed circuit board (PCB) design and fabrication, you could produce a few hundred units and just maybe change the world.

Problem is, \$50K is too little to interest a venture capitalist, too much to put on your credit cards, and too much to borrow from the bank.

If you're motivated, have a clear vision for your product, and have a good handle on production cost, then there's another, relatively risk-free option: Kickstarter (www.kickstarter.com).

Basically, it's an eBay for DIY funding in which you pitch your product plan to the world. Anyone or any company can back your project. The cost is 5% of the funding, but only if the project is fully funded. Unlike a venture capitalist or silent partner, you get to keep all of the intellectual property.

I recently backed a Kickstarter project that transforms a smartphone into a realtime IR camera. Sure, I could just buy a more expensive stand-alone unit off the shelf and not worry about warranty, customer support, or wonder if the designer will be around in two years, but that isn't the point. It's fun to support someone's dream and — at least

vicariously — be part of the action.

Back to *your* dream. Let's say you need that \$50,000 to produce 100 mouse control units at \$500 each. As on eBay, you set up an account so that you can get paid. You've got to describe your project (including a video or at least photos) and set up categories of backing. Support isn't an all-or-nothing proposition.

Typically, the first level of support is a thank you email for a pledge of \$1. At, say, \$25, a backer receives an official project t-shirt. At \$100, you might offer a bare-bones PCB with full schematic and instructions.

At \$250, you might offer an unassembled kit with all the parts. A pledge of \$500 gets you the full product (shipping extra). A pledge of \$750 gets one of 10 custom units, in dusty blue, signed by you.

At \$1,200, you offer to hand deliver and install a unit at the customer's site.

You can use Kickstart to fund just about any reasonable project. I've seen projects ranging from \$1,500 to use drones to search for Sasquatch to \$120,000 for a speaker system, to \$7,000 for a ghost detector that plugs into an iPhone.

I've seen projects funded at up to 1,500% over target. For their 5% cut, Kickstart hosts your product website and handles the financial transactions.

In addition to raising money to pay for your dreams, Kickstart is a great test marketing tool. Let's say that after 30 days your rodent control device has only \$3 in pledges. That's great marketing feedback for a relatively small investment in time and effort.

Perhaps you can redesign the unit so that you can offer it at, say,

NEW! DSA815 Spectrum Analyzer

Our new analyzer with all-digital IF technology redefines the product category!

- 9 kHz to 1.5 GHz Frequency Range
- Typical -135 dBm Displayed Average Noise Level (DANL)
- -80 dBc/Hz @ 10 kHz offset Phase Noise
- Total Amplitude Uncertainty <1.5 dB
- 100 Hz Minimum Resolution Bandwidth (RBW)

Get a Spectrum Analyzer at oscilloscope prices!

Buy online @ RigolAnalyzer.com

Starting at
\$1,295

RIGOL
Beyond Measure



NUTS AND VOLTS

EVERYTHING ELECTRONIC

Published Monthly By

T & L Publications, Inc.

430 Princeland Ct.

Corona, CA 92879-1300

(951) 371-8497

FAX (951) 371-3052

Webstore orders only 1-800-783-4624

www.nutsvolts.com

Subscriptions

Toll Free 1-877-525-2539

Outside US 1-818-487-4545

P.O. Box 15277

North Hollywood, CA 91615

FOUNDER/ASSOCIATE PUBLISHER

Jack Lemieux

PUBLISHER

Larry Lemieux

publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING

Robin Lemieux

display@nutsvolts.com

EDITOR

Bryan Bergeron

techedit-nutsvolts@yahoo.com

CONTRIBUTING EDITORS

Jeff Eckert

Russ Kincaid

Joe Pardue

Fred Eady

Ron Hackett

Lou Frenzel

Gordon McComb

David Goodsell

Dan Gravatt

CIRCULATION DEPARTMENT

subscribe@nutsvolts.com

SHOW COORDINATOR

Audrey Lemieux

MARKETING COORDINATOR

WEBSTORE

Brian Kirkpatrick

sales@nutsvolts.com

WEB CONTENT

Michael Kaudze

website@nutsvolts.com

ADMINISTRATIVE ASSISTANT

Debbie Stauffacher

PRODUCTION

Sean Lemieux

Copyright © 2013 by T & L Publications, Inc.

All Rights Reserved

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

Printed in the USA on SFI & FSC stock.



\$200. Or, perhaps you need to invest time in making a quality video that will attract more backers.

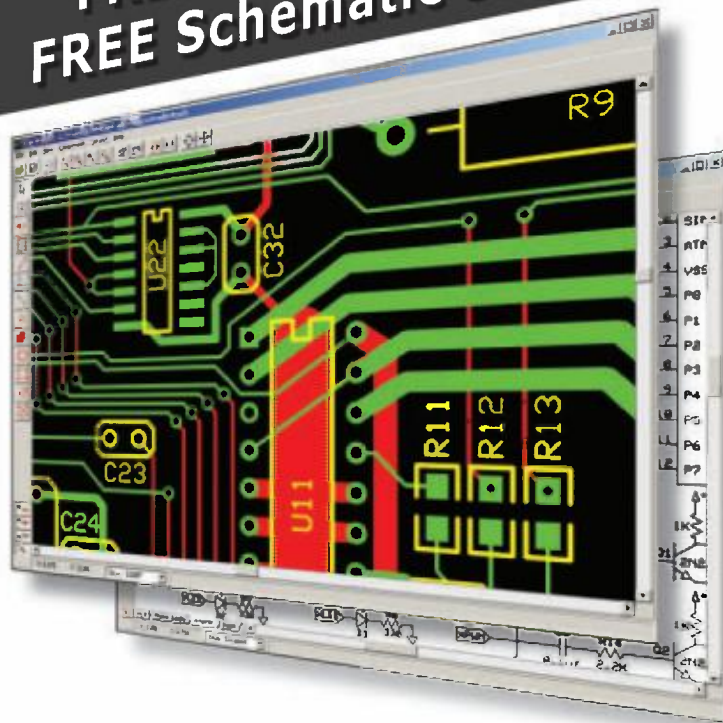
Even if you opt to bypass the funding and mortgage your home, the Kickstart site is worth visiting and studying. Take a look at what's selling and what isn't.

Most importantly, study the how-to guides on how to put together a killer video, how to put

the best marketing spin on your DIY project, and how to plan for the details that tend to fall through the cracks — such as the hidden cost of bubble mailers.

Finally, be careful for what you wish for. If you find you project funded at 300%, then you'd better have time (and space) to build a few hundred of those mouse control units. **NV**

For 3
\$51 PCBs
FREE Layout Software!
FREE Schematic Software!



01 DOWNLOAD our free CAD software

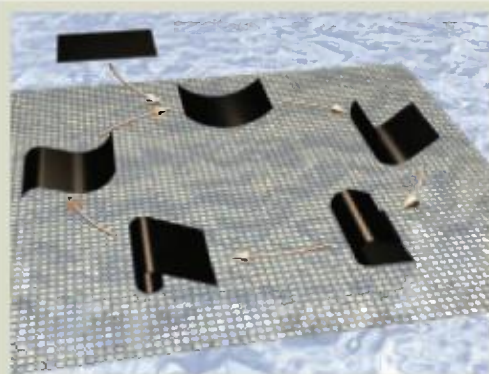
02 DESIGN your two or four layer PC board

03 SEND us your design with just a click

04 RECEIVE top quality boards in just days

expresspcb.com

ADVANCED TECHNOLOGY



■ Shape-shifting phases as MIT polymer absorbs and releases water vapor.

TAPPING INTO TAP WATER

For many years, people have proposed various schemes to convert water into fuel. After all, water is the most abundant compound on the Earth's surface, and is therefore

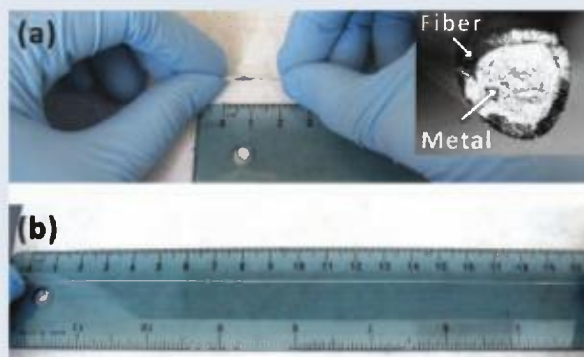
essentially free. The problem is that separating it into its components by hydrolysis eats up more energy than you can derive from burning the resulting hydrogen. However, some MIT (www.mit.edu) engineers have figured out a way to use water to generate electricity without including any kind of combustion in the process. In a paper recently published in *Science* magazine, postdoctorate student Mingming Ma describes the use of a new type of polymer film to harvest energy from water vapor. The film is analogous to a bimetal strip as used in thermostats and such, but in this case, it is made up of a layer of polypyrrole which is hard but flexible, then interlocked with a layer of polyol-borate, which is softer and swells up when it absorbs water vapor. The film curls during the absorption process and then flattens out as the water evaporates. The cycle is repeated, thus deriving continuous mechanical energy from the gradient between wet and dry environments.

Obviously, we're not talking about a lot of energy here, but on a small scale it can be significant, and, in fact, the film can act either as a mechanical actuator or a generator. The researchers demonstrated that 25 mg of film can lift a stack of glass slides weighing 9.5 g (380 times its own weight) or drag a 250 mg load of silver wires when acting as a "mini tractor." They also state that "the mechanical energy generated by the material can be converted into electricity by coupling the polymer film with a piezoelectric material which converts mechanical stress to an electric charge. This system can generate an average power of 5.6 nW which can be stored in capacitors to power ultra-low-power microelectronic devices such as temperature and humidity sensors." Scaling the process up to compete with hydroelectric dams seems unlikely, but the developers have speculated that the material could be incorporated into clothing so that, for example, you could power your exercise monitor with sweat. ▲

WIRES STRETCH 8X

Once upon a time back in the Dark Ages, telephones, headphones, and other devices incorporated coiled wires to allow users to flexibly connect transducers to base units. You don't see things like that much anymore, but there are still some times when it would be nice if you could stretch a wire out beyond its normal length. Such a stretchable wire has been devised by Dr. Michael Dickey, an assistant prof at North Carolina State University (www.ncsu.edu). Previous attempts to create stretchable conductors have focused on embedding metals or other conductive substances into elastic polymers, but the problem with that approach is that the more conductive material you include, the less elasticity you get, and vice versa. Dickey's approach is to fill a tube of elastomer with liquid metal (an alloy of gallium and indium), allowing it to be stretched up to eight times its original length without compromising its conductivity.

In addition, Prof. Dickey has demonstrated another type of stretchable wire that employs a self-healing polymer insulator, allowing you to snip it in two and reconnect it by just squashing the ends together with no twisting or soldering required. He noted, "We're excited about this work because it allows us to create more complex circuits and rewire existing circuits using nothing more than a pair of scissors by cutting and reconfiguring the wires so that they connect in different ways." For videos, just search YouTube for "ultra stretchable wires" and "self-healing stretchable wires." ▲



■ A one inch wire still functions when stretched to eight inches.

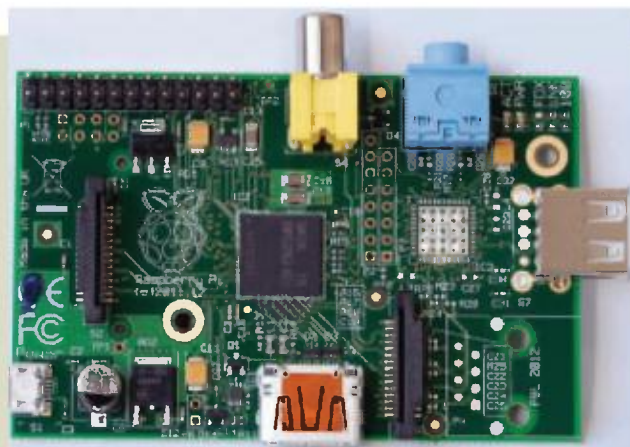
Discuss this article in the Nuts & Volts forums at <http://forum.nutsvolts.com>.

COMPUTERS AND NETWORKING

\$25 COMPUTER HITS THE MARKET

Back in August of 2011, we described the Raspberry Pi Foundation (www.raspberrypi.org) and its efforts to develop a functional computer that would sell for \$25, thus making it affordable to virtually everyone, everywhere. The result was the Raspberry Pi Model B which actually comes in at \$35 instead. However, with the recently announced Model A, the \$25 goal has been met. Both models are based on an ARM1176JZFS processor with floating point, running at 700 MHz, but apparently you can overclock it to 800 MHz if you want. They also include a Videocore 4 GPU which is capable of 1 Gpixel/s, 1.5 Gtexel/s, or 24 GFLOPs of general-purpose computing, giving it graphics capabilities that are "roughly equivalent to an Xbox 1 level of performance."

Debian (www.debian.org) is the default distribution OS. The only apparent difference is that the Model A is a stripped-down version of the Model B with no Ethernet, only one USB port instead of two, and 256 MB of RAM rather than 512 MB. This means that the Model A draws only about one third of the power, so it might be a better choice for "those of you wanting to run projects from a battery or



■ The Raspberry Pi Model A computer — a stripped-down version of the Model B.

solar power: robots, sensor platforms in remote locations, Wi-Fi repeaters attached to the local bus stop, and so forth." As of this writing, the Model As are available only in Europe, but that is slated to change "very soon."

Why did the Model B come before the Model A? Who knows. Why did the Model T Ford come before the Model A? Life is a mystery. ▲

GET OFF OF MY CLOUD!

Also, "cloud computing" is basically just a trendy marketing term for the use of shared hardware and/or software resources over the Internet. (However, if you want a government-generated seven-page document that makes it sound much more complicated, just go to www.nist.gov and search for "special publication 800-145.") Among the pros of cloud computing are cost reduction, simpler collaboration, and off-site storage of sensitive data. Among the cons, there are big questions about the ability to access your files if the server goes

down or you have a dispute with the provider, and even bigger questions about privacy.

The host has full access to your data at all times, and communications are not necessarily secure from outside parties. Recall, for example, that little incident in which a formerly secret program by the National Security Agency (with cooperation from AT&T and Verizon) recorded more than 10 million US citizens' domestic telephone calls.

Now, you can dump your backup services and set up your own private secure cloud service using a Transporter device from Connected Data (actually located at www.filetransporter.com). According to the company, "What sets Transporter apart from all other solutions is the ability to communicate and share files with other Transporter devices located anywhere in the world. This ability eliminates all complexity associated with syncing files and delivers fast local copies of data without having to move it into or out of the cloud."

All stored files are available to any computer or mobile device with Internet access, and files can be synced for offline access. In addition, there are no monthly fees, so you're just looking at a one-time cost. Three versions that are essentially the same are available. One comes without a hard drive, so you can just install your own (\$199), another includes a 1 TB SATA drive (\$299), and the third comes with a 2 TB drive (\$399). They are available from the usual sources on the web. ▲



■ Your own computing cloud starting at \$199.

CIRCUITS AND DEVICES

DSLR OFFERS 2D AND 2D/3D LENSES

For those of us who grew up using heavy bulky 35 mm SLR film cameras, the recently introduced NX300 from Samsung (www.samsung.com/us) looks like a familiar old friend. Of course, it's purely digital inside, and at 9.9 oz (280 g), it won't be as much of a pain in the neck strap as my trusty old Minolta SRT 101, which came in at 1 kg (2.2 lb). A central feature is its new 20.3 megapixel APS-CMOS sensor which offers an ISO range from 100 to 25,600, making it useful even in low light conditions. It also includes Samsung's proprietary DRIMe IV imaging engine, designed to provide better color reproduction and greater noise reduction, as well as support for full 1080p HD video capture in both 2D and 3D modes.

Did I say 3D? Yeah, the NX300 can be fitted with the company's new 45 mm f/1.8 2D/3D lens, billed as "the first one-lens 3D system capable of capturing both still images and full HD movies." Rather than using twin lenses to achieve 3D, the lens puts two LCD screens in the optical path and they take turns blacking out each other's images. The NX300 captures the slightly different alternating frames and later assembles them into 3D stills or movies. The camera includes Wi-Fi connectivity, so you can instantly share your shots via the SMART CAMERA app which is available for both Android and iOS smartphones and tablets. All of this doesn't come cheap, as the street price for an NX300 with the standard 20-50 mm lens is about \$750 as of this writing, and the 2D/3D lens will cost you another \$500. But this ain't no Brownie Starflash. ▲

■ Samsung's NX300 mixes a retro look with well-bred digital features.



SUPER DUPER

Anyone who has burned a few CDs or DVDs on their home computer knows that it's a boring, time-consuming process, even at 16X. For most of us, it's a minor annoyance, as we seldom use that medium

anymore. If you are involved in large-scale duplication jobs, however, you might want to take a gander at Aleratec's new 1:3 RoboRacer DVD/CD MultiDrive SA Duplicator – a fully automatic unit that features three drives and a large blank disc bin for hands-free operation. The machine can operate in stand-alone mode, so you don't even have to connect a computer to it.

According to Aleratec, "The user simply loads the duplicator with a source disc or source image via the USB connection and up to 150 blank discs. Simple, pushbutton controls start the duplication process, and no further user intervention is required." The MultiDrive duplicator is fitted with a 2 TB hard drive and employs the company's Smart Image Management System to maximize the number of images that can be stored thereupon. For details, visit www.aleratec.com.

Potential buyers should, of course, be aware that duplication of copyrighted material is forbidden by law, and so anyone who might be tempted to show gratitude by sending yours truly a complete set of music from, oh, let's say the Stax recording studio would not be officially encouraged to do so. If you can afford to shell out \$1,699 for the dupe unit, it would be theoretically possible. ▲



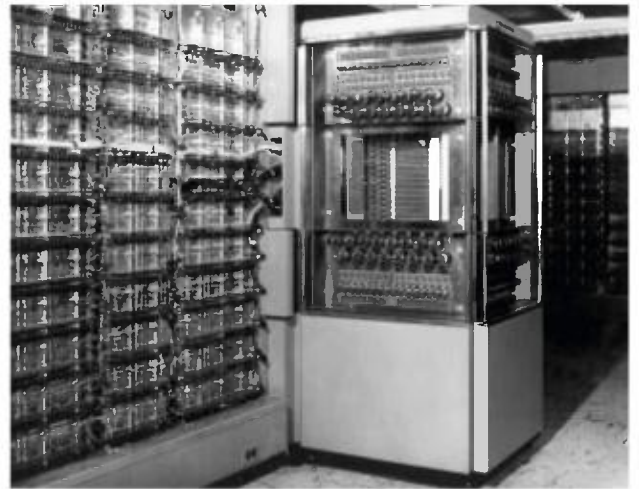
■ Aleratec's RoboRacer MultiDrive duplicator handles up to 150 discs with no human intervention.

INDUSTRY AND THE PROFESSION

CORE MEMORY'S 60TH ANNIVERSARY

With today's hard drives providing up to 2 TB in a single compact inexpensive unit, it's easy to forget just how far the technology has come. As it happens, it was 1953 when the first magnetic-core memory was installed on the Whirlwind computer in a joint project between MIT and the US Navy. The project entailed a real time interactive simulator and stabilizer analyzer for Navy flight training. The core consisted mostly of a wire mesh and a bunch of ferrite rings, but it allowed binary information to be recorded and randomly retrieved magnetically, which was a pretty novel idea at the time. This paved the way for the IBM 305 Random Access Memory Accounting Machine (RAMAC), which in 1956 became the first commercial computer that used a moving-head hard drive. One drive stored 5 MB on 50 24 in platters and was about the size of two telephone booths. **NV**

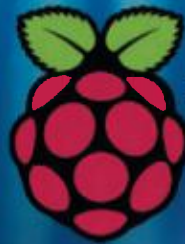
Photo courtesy of IBM.



■ This early memory device offered 8,192 words of core memory and a speed of about 75,000 single address instructions per second.



1-800-543-4330



Celebrating the
1 year
anniversary
of the Raspberry Pi Board!

MCM Electronicssm Is Your Raspberry Pi Specialist

MCM has a complete selection
of Maker/DIY accessories from
all your favorite manufacturers



**ONLY
\$35⁰⁰**

MCM Part #83-14421

Raspberry Pi Exclusives:



Gertboard

MCM Part #83-14460



Pi Face

MCM Part #83-14472



WiPi

MCM Part #831-3058



Pi View

MCM Part #83-14407

MCMelectronics.com/Maker

EXPERIMENTING WITH PARALLEL TASK PROCESSING

In the previous installment of the Primer, we constructed a multiplexed two-digit LED display based on the Lumex LDD-N514RI-RA display, and developed a program (`LEDmultiCount99.bas`) that counted seconds from 0 to 99. We also discussed the problem that any program modifications would involve a fair amount of "re-balancing" of the code so that the two displayed digits would appear equally bright. This month, we're going to explore a new M2-class feature called Parallel Task Processing which will greatly simplify the development of any project that includes our multiplexed display.

UNDERSTANDING PARALLEL TASK PROCESSING

All PICAXE processors are single-core devices which means that they can only execute one instruction at a time. The only exception occurs in situations where an instruction uses an on-chip component that functions independently of the main processor core. For example, the `pwmout` command utilizes an internal timer that allows the PWM waveform to be output simultaneously with whatever instruction the processor happens to be executing from moment to moment.

However, all M2-class processors include a new software feature called *Parallel Task Processing* that allows them to simulate a multi-core processor by rapidly switching between (among) two or more separate tasks in a "round robin" fashion. The task switching occurs at a very high speed, so the separate tasks appear to be executing simultaneously, even though the processor is only executing one instruction at a time in rapid succession.

Actually, this concept may sound familiar; we often do something very similar in many of our programs. In order to explain what I mean by that, let's consider a simple example that illustrates the point. Suppose I were

to give you the following programming challenge:

Hardware Setup: An 08M2 processor with an LED (and current-limiting resistor) attached to pin C.0; another LED (also with a current-limiting resistor) attached to pin C.4; and a momentary pushbutton switch attached to pin C.3.

Software Requirements: The LED on pin C.0 should blink very briefly once every five seconds. (The timing does not need to be precise.) The LED on pin C.4 is normally off, but pressing the switch immediately turns it on. It remains on as long as the switch is pressed, and turns back off as soon as the switch is released. No matter how often (or how long) the switch is pressed, the LED on pin C.0 continues to blink very briefly once every five seconds.

A photo of my breadboard setup for the above challenge is shown in **Figure 1**. (If you don't have an available 08M2, any M2-class processor will do.) In the photo, both LEDs have internal current-limiting resistors, and there's a 1K current-limiting resistor connected to ground in the switch circuit.

In my program for the challenge, I enabled the internal pullup resistor for pin C.3. On the 08M2 processor, pin C.3 is fixed as an input, so the 1K resistor isn't really necessary. However, if you connect a switch to

any PICAXE I/O pin that can be configured as an output, omitting a resistor in the circuit creates a potential source of damage. If the pin were accidentally configured as an output and set to a high state, pressing the switch would create a direct short to ground which would probably damage or destroy the pin, and possibly the processor as well.

If you decide to accept my challenge, you can stop reading at this point, implement your breadboard setup, and develop your software solution. When you're ready, continue reading to see my solution to the challenge.

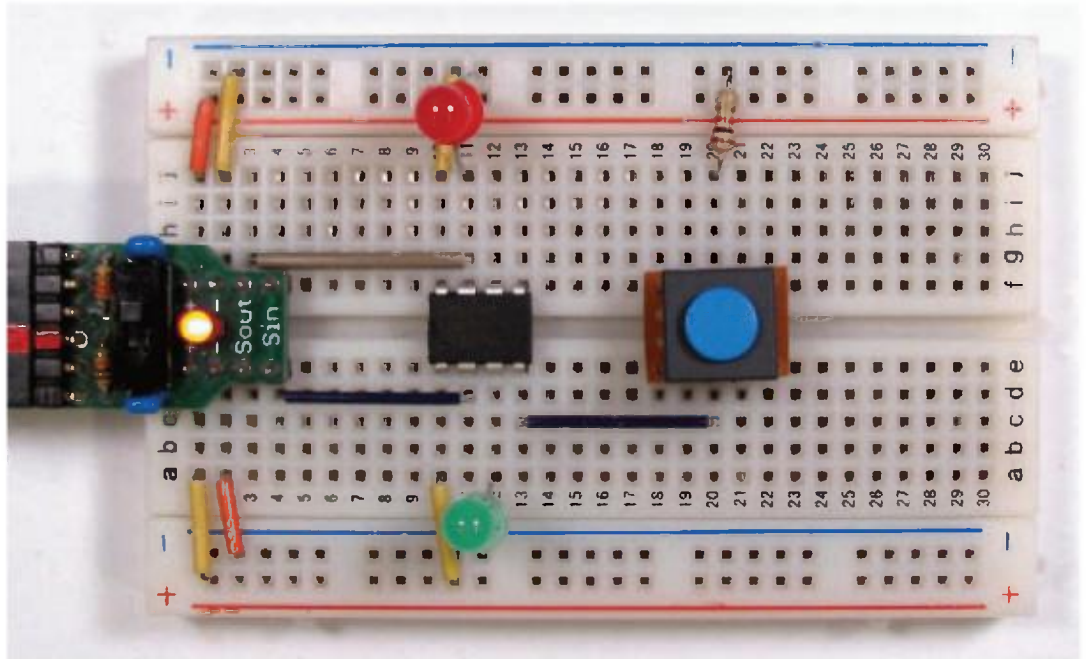
One possible solution to the challenge (*SwitchChallenge.bas*) is available for downloading at the article link, along with all the other programs we'll be using this month. The following code is the main *do/loop* from the program (without the comments that are included in the downloadable program):

```
do
  high LED0
  pause 100
  low LED0

  for index = 1 to 100
    if sw3 = 1 then
      low LED4
    else
      high LED4
    endif

    pause 49
  next index
loop
```

The *for/next* loop is the key to understanding how the above code functions. For each blink of the LED that's attached to pin C.0, the *for/next* loop checks the state of the switch (pressed or not pressed) 100 times, and appropriately sets the state of the LED that's attached to pin C.4.



■ FIGURE 1. Breadboard setup for my programming challenge.

Because the state of the C.4 LED is updated about 100 times in five seconds (i.e., 20 times per second), it appears to us that LED4 is responding “immediately” to our switch presses.

The problem with this approach to programming multiple tasks is that it isn't very easy to understand. For example, the inclusion of the *pause 49* statement may be somewhat confusing at first. Here's how I arrived at that value.

Since the timing doesn't need to be exact, I only considered the time used by the *pause* statements, so 49 ms * 100 iterations of the *for/next* loop = 4,900 ms “off-time” for LED4. Add that to the 100 ms “on-time” and you get 5,000 ms which is the same as five seconds. Of course, the other statements do take some time to execute, so each blink actually takes slightly longer than five seconds, but let's not quibble!

Now, let's take a look at how the challenge could be met by using a parallel task program. The complete downloadable program (*Switch2Tasks.bas*) includes relevant comments. All the declarations are the same as they were in the

previous program, so let's focus on the main portion of the code:

```
start0:
  pinsC = %00010111
  pullup %00001000

do
  high LED0
  pause 100
  low LED0
  pause 4900
loop

start1:
do
  if sw3 = 1 then
    low LED4
  else
    high LED4
  endif
loop
```

To begin with, note the *start0:* and *start1:* labels which — along with *start2;* *start3;* etc. — are pre-defined in the PICAXE compiler. Whenever we include one or more of these pre-defined labels, the compiler automatically switches the program into parallel task mode. The *start0:* label is optional because the first program statement is always the beginning of Task 0 — whether or not a *start0:* is included. However, if you do include a *start0:* label, it must be on the first line of program code.

If you examine the above code, you will see that there are two tasks. Task 0 configures the port C pins, and then blinks LED0 once every five seconds (or so) in an infinite loop. Task 1 is responsible for continually checking the state of the switch and appropriately updating the state of LED4.

The compiler executes the commands in the following round robin order: the first statement in Task 0, the first statement in Task 1, the second statement in Task 0, the second statement in Task 1, etc. Whenever a *pause* or *wait* statement is encountered, the compiler simply updates the time remaining in the delay and then moves on. In other words, when one task is pausing (or waiting), the statements in the other tasks continue to be executed.

As a result, parallel task processing is not only easier to understand, it also responds much faster than the traditional approach we used earlier. For example, in the present program, the state of the switch is tested much more frequently than it was in the earlier version.

In order to see how easy it is to develop a parallel task program, let's "up the ante" on our programming challenge by adding a third LED on pin C.2. (In order to keep life simple, let's declare this LED as LED2.) This time, the challenge is to modify the *Switch2Tasks.bas* program so that LED2 briefly blinks once every second. In other words, LED2 should blink five times as often as LED0 does. Also, one out of every five LED2 blinks should exactly coincide with the LED0 blink. Of course, LED4 should still immediately respond to any switch presses.

Before you continue reading, see if you can add a third parallel task that meets these requirements. When you have your program working correctly, save it as *Switch3Tasks.bas*. If the code that you added looks something like the following, you win the prize!

```
start2:
do
  high LED2
  pause 100
  low LED2
  pause 900
loop
```

A FEW IMPORTANT DETAILS OF PARALLEL TASK PROGRAMS

Section 1 of the PICAXE manual includes a thorough explanation of the details of implementing parallel task programs. However, I realize that not everyone likes to read manuals, so I want to include a few of the most important details to keep in mind:

- All constants, variables, and RAM/EEPROM storage are shared among all tasks.
- Tasks can be any length, as long as they fit within the processor's total program memory space.
- If necessary, you can suspend (and resume) a task during program execution. (See the documentation in Section 1 of the manual for details.)
- The *sleep* and *nap* commands temporarily suspend all tasks.

LIMITATIONS OF PARALLEL TASK PROGRAMS

Again, see Section 1 of the manual for more details, but the following are the most important limitations of parallel task programs:

- Parallel task programs are not suitable for programs in which multiple tasks simultaneously use different advanced features, or features that require critical timing accuracy, e.g., serial I/O, I²C, *irin*, *readtemp*, etc.
- The *setfreq* command cannot be used in parallel task

programs. The compiler automatically switches among available clock frequencies as necessary, in order to maintain an "apparent" clock speed of 4 MHz for all tasks. As a result, servo and PWM pulse generation may be adversely affected.

- Overall timing accuracy may be slightly reduced. Any program that requires a high degree of timing accuracy should be written in the traditional "single task" approach.

In spite of these limitations, parallel task programming is a powerful new feature that can greatly simplify many M2-class projects, which brings us to the main purpose of this installment.

USING PARALLEL TASKING TO MULTIPLEX THE N514RI LED DISPLAY

In our previous Primer, we experimented with a multiplexed two-digit LED display (the N514RI), and implemented a two-digit count-up timer with the *LEDmultiCount99.bas* program. At this point, we're ready to modify that software so that it functions as a two task parallel task program. (You may want to print out the *LEDmultiCount99.bas* program listing so that you can compare it to the software changes we're about to discuss.)

In the original single task program, we only needed to define two variables (*dig01* and *dig10*) because we were able to use two *lookup* commands to immediately update the port B pins with the appropriate segment values in order to display the digits as they changed in real time. However, in our two task program, we're going to update the segment values in Task 0 and

multiplex the LED display in Task 1. As a result, we need two additional variables in which to store the segment values; let's declare them as `segs01` and `segs10`.

Task 0 — which updates the segment values in real time — will look like this:

```
start0:
dursB = %11111111
dursC = %10111111

do
  if time > 99 then
    time = 0
  endif

  dig01 = time // 10
  lookup dig01, (63,6,91,79,
    102,109,125,7,127,
    111),segs01
  dig10 = time / 10
  lookup dig10, (63,6,91,79,
    102,109,125,7,127,
    111),segs10
loop
```

If you compare the above code to the main `do/loop` in the `LEDmultiCount99.bas` program, you can see that the `start0:` code essentially consists of all the statements from the original program that relate to updating the `time` variable and the corresponding segment values. The only difference (which I mentioned above) is that we need to store the segment values in separate variables so they can be accessed by the Task 1 loop, which we're about to discuss. (Don't forget, all program variables are available to all running tasks.)

Now, let's take a look at Task 1 which simply multiplexes the display:

```
start1:
do
  pinsB = 0
  low A.0
  pinsB = segs01
  'here, we need
  'a brief delay to
  'balance brightness
  pinsB = 0
  high A.0
  pinsB = segs10
loop
```

Again, if you compare this code to the main `do/loop` in the `LEDmultiCount99.bas` program, you

will see that the `start1:` code essentially consists of all the statements from the original program that relate to multiplexing the display. The only difference (which we'll discuss in detail below) is the missing code that we need to use in place of the `pauseus 320` statement in the original program.

For now, just think of that code as a `pause` statement that balances the brightness of the two displayed digits.

As usual, the fully commented version of the parallel task program (`Count2Digits2Tasks.bas`) is available at the article link. Download the program to the same hardware setup that we implemented in the previous Primer column; it should display the same 0 to 99 second counter that we observed before. If not, you will need to check the wiring of your breadboard setup.

TIMING CONSIDERATIONS FOR PARALLEL TASK PROGRAMS

Reading the parallel task documentation in Section 1 of the PICAXE manual had prepared me for the possibility that balancing the brightness of the two display digits might not be easy. However, it still turned out to be more of a challenge than I anticipated.

When I first started developing the program, I couldn't get the two display digits to look equally bright. At that point, I was using a `pause 1` statement, but any pause between 1 ms and 19 ms resulted in almost no visible change; the one's digit appeared a little dimmer than the ten's digit, regardless of the length of the pause I tried. Even worse, when I



EARN MORE MONEY

Get your dream job!

Be an FCC Licensed Radio Technician!

Make up to \$100,000 a year and more with NO college degree

Get your "FCC Commercial Radio License" with our proven Home-Study Course!

- No costly school. No classes to attend.
- Learn at home. Low cost!
- No experience needed!
- **MONEY BACK GUARANTEE:** You get your FCC License or money will be refunded!

Turn your interest in electronics into a high paying career in Communications, Radio, Television, Avionics, Marine, Radar, Safety, Municipal and more... even start your own business!

Call now for FREE info: 800-932-4268

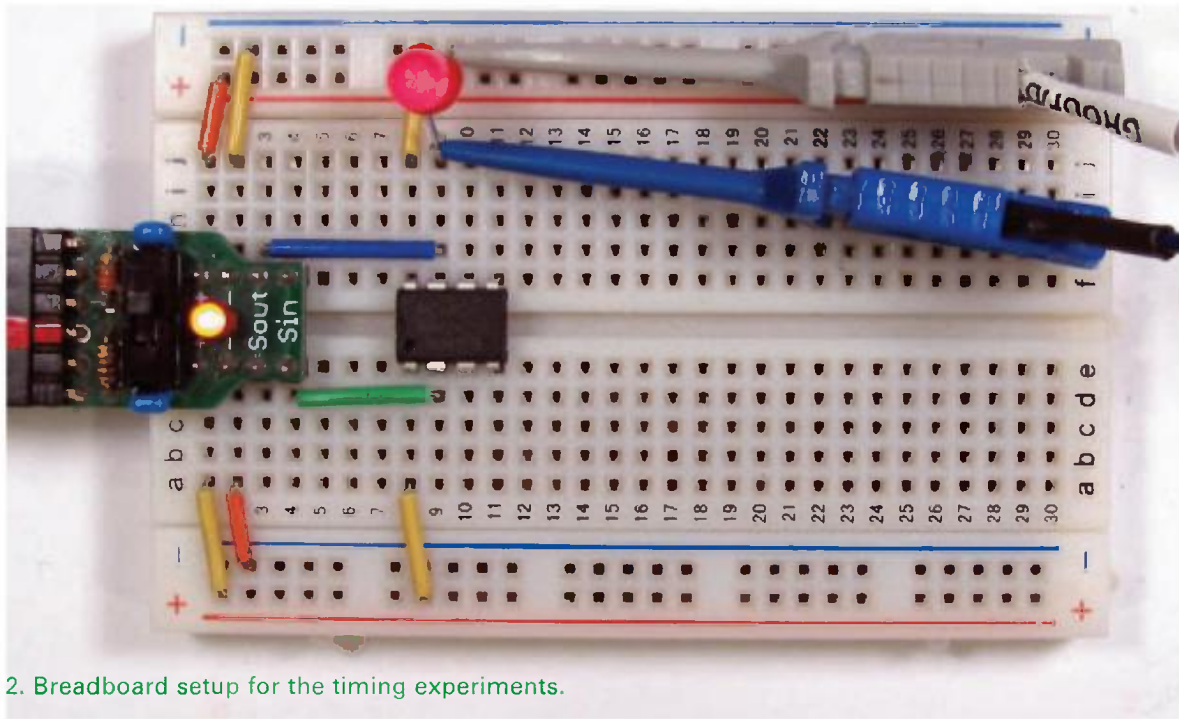
Or, email us: info@LicenseTraining.com

www.LicenseTraining.com

COMMAND PRODUCTIONS, Warren Weagant's FCC License Training
Industrial Center, 480 Gate Five Rd., PO Box 3000, Sausalito, CA 94966-3000



ext. 109



■ FIGURE 2. Breadboard setup for the timing experiments.

tried a 20 ms or larger pause, the one's digit suddenly became very bright, and the ten's digit flickered noticeably.

In an attempt to understand what was happening, I connected my digital logic analyzer to the A.0 output pin so that I could view the waveform that the program was outputting.

For pauses anywhere between 1 ms and 19 ms, the output wave remained essentially the same, with no change whatsoever in the duty cycle. Of course, that just confirmed the fact that I had a big problem.

In order to try to figure out what was happening, I decided to simplify the hardware and software as much as possible. I also decided to investigate the *pauseus* statement as well as the *pause* statement, in order to see if it worked better than the *pause* statement.

I don't think we've used the *pauseus* statement before, so let's take a quick look at its syntax: *pauseus microseconds*.

The *microseconds* parameter is a word variable or constant, so it can accommodate values between 0 and 65535. It's also a bit of a misnomer,

in that it specifies the time duration of the pause — not in microseconds, but in units of 10 microseconds. In other words, a *pauseus 100* statement produces a 1,000 μ S pause, and since 1,000 μ S = 1 ms, it produces the same delay as a *pause 1* statement. At least, that's what it does in single task programs, but read on ...

To carry out my experiments, I set up a simple 08M2 breadboard circuit with a resistorized LED on pin C.0 and my logic analyzer connected to the pin C.0 output as shown in Figure 2.

I ran each of the following two code snippets twice: first (as shown below) with the *pause 10* statement, and again with a *pauseus 10* statement (in place of the *pause 10* statement). Using the logic analyzer, I recorded the duty cycle of the waveform on pin C.0 for each experiment.

• Single Task

```
do
  low C.0
  pause 10
  high C.0
loop
```

• Two Tasks

```
start0:
do
  low C.0
  pause 10
  high C.0
loop

start0:
do
  for b0 = 0 to 99
    b1 = b0 / 10
    b2 = b0 //10
  next b0
loop
```

The results of the four experiments are presented in Figure 3. First of all, note that the single task results are "rock-solid" — for both the *pause* statement and the *pauseus* statement; each iteration of the *do/loop* resulted in an almost identical duty cycle. However, the two task results were somewhat more variable; the duty cycle varied slightly (within a range of 3%) with each iteration of the *do/loop*.

	Pause 10	Pauseus 10
Single Task	9%	55%
Two Tasks	49%-52%	47%-50%

■ FIGURE 3. Results of the timing experiments.

As I mentioned earlier in the "Limitations" section, overall timing accuracy may be slightly reduced in parallel task programs, and the present result is an example of that reduction in accuracy.

The duty cycles that resulted from the two single task snippets are also not surprising. A *pause 10* statement is 100 times as long as a *pauseus 10* statement, which greatly lengthens the low portion of the C.O output and, in turn, lowers the duty cycle (as compared to that of the *pauseus 10* statement).

However, the duty cycle that resulted from the two task snippet for the *pause 10* statement (47% to 50%) was very surprising (and troubling). Since the corresponding single task snippet produced a duty-cycle of 9%, I expected the two task results would also be reasonably close to 9%.

In hopes of shedding some light on my problem, I posted the details on the PICAXE forum. As usual, the responses were very helpful — especially one from "Technical," who is a member of the RevEd support staff. He pointed me to a brief paragraph in the documentation for the *pause* command in Section 2 of the manual: "During M2 part multitask programs, the accuracy of *pause* is reduced due to the parallel processing. The minimum resolution is around 20 ms in multitask programs. For greater accuracy, use single task mode."

I had been so focused on the "parallel task" documentation in Section 1 that I hadn't thought to thoroughly read the *pause* documentation in Section 2. Fortunately, Technical's heads-up confirmed that the problems I encountered are the result of

compiler limitations when a *pause* statement is used in a parallel task program.

He also explained that — unlike the *pause* statement — a *pauseus* statement is "blocking" which means that it finishes completely before the processor moves on to the next programming task. That, of course, means that it functions more predictably in a parallel task program.

Consequently, I decided to abandon the *pause* statement altogether, and replaced it with a *pauseus* statement. After a few trial-and-error attempts, I found that the digit brightness was evenly balanced when I used a *pauseus 100* statement in the N514R1 parallel task program.

However, we're not yet at the end of the story. For some reason, I decided to re-check the timing accuracy of the resulting program, and I discovered that it was losing

Superior Quality Soldering Tools from

HAKKO

NEW

FX-951

HIGH PERFORMANCE SOLDERING STATION

- Digital temperature display
- Superior heat transfer
- Fast thermal recovery
- 400° - 840°F (200° - 450°C)
- Low temperature alarm
- Sleep mode for longer tip life

FX-601

TEMP-ADJUSTABLE SOLDERING IRON

- Electronic "closed loop" control
- 464° - 1004°F (240° - 540°C)
- Ceramic heating element
- Lightweight (2.4oz w/o cord)
- Extra long 3-plug, 6 ft. cord
- UL/CSA Approved

FX-888D

LOW COST, DIGITAL SOLDERING STATION

- Digital display (°F or °C)
- Adjustable temperature
- 120° - 899°F (50° - 480°C)
- Idle temp within 1.8°F (1°C)
- Ergonomic, handpiece
- Ceramic heater

For complete information, visit HakkoUSA.com/NutsVolts



more than six seconds for every minute that it was running!

Evidently, in a parallel task program, the *pauseus* statement negatively impacts the accuracy of the M2-class built-in *time* variable to a point where it's practically useless. (To make matters worse, I can't even claim that the documentation didn't warn me about the timing problems.)

Needless to say, I was becoming more than a little discouraged, so I resorted to a tactic that has worked well for me in the past: I temporarily gave up! That may sound somewhat defeatist, but I've found that by distracting myself from what seems to be an intractable programming problem, a solution often occurs to me later when I'm not thinking about the problem at all.

In this case, I woke up the next morning with the realization that every programming statement takes a brief amount of time to execute, so maybe I could balance the digit brightness without using either a *pause* or a *pauseus* statement; instead, I could try a "dummy" statement whose only purpose was to "waste" a little time.

The very first statement that I tried (*low A.0*) improved the brightness balance of the two display digits, but I could still see a slight difference.

Next, I tried using a second *low A.0* statement, and that worked like a charm; the two digits were evenly balanced. The following code snippet demonstrates the use of the dummy statements:

```
start1:
do
  pinsB = 0
  low A.0
  pinsB = segs01
  low A.0 'dummy
  low A.0 'dummy
  pinsB = 0
  high A.0
  pinsB = segs10
loop
```

The second and third *low A.0* statements have no effect on the A.0 pin, because that pin is already in a

low state due to the first *low A.0* statement. However, the two additional *low A.0* statements do take a brief amount of time to execute. In fact, they take roughly the same amount of time as the *loop* statement at the end of the *do/loop*.

As a result, the brightness of the two display digits is evenly balanced, and (most importantly) the timing accuracy is also significantly improved.

I let the resulting program run for an hour, and it only lost about two seconds. In comparison, the earlier *pauseus* version would have lost six minutes in an hour. The final version of the parallel task program is *Count2Digits2Tasks.bas*. Download it to the same setup we used in the last Primer, and see what you think.

LIAR LIAR, PANTS ON FIRE!

In the beginning of this column, I mentioned that parallel task processing would greatly simplify the task of developing a project that includes our multiplexed display. If you remembered those words, it's understandable if you found yourself reciting that popular children's chant. I certainly wouldn't blame you if you did; getting to this point has been anything but simple!

However, now that we have the basic program functioning correctly, it's very simple to modify. (Really, it is!)

As an example, let's add an LED and current-limiting resistor to pin C.0 of our 20M2 setup, and modify the program so that it blinks the LED in time with the N5145RI count as it progresses.

All we need to do is add a third task to the program:

```
start2:
do
  lastSec = time
  high C.0
  pause 20
  low C.0
  gosub waitAsec
loop
```

```
waitAsec:
  if lastSec = time then
    waitAsec
  return
```

Of course, we also need to declare the *lastSec* variable; other than that, no additional program changes are required. The fully commented version of the program is *Count2Digits3Tasks.bas*.

The only slightly tricky part of Task 2 is how we're using the *lastSec* variable, but that has nothing to do with parallel task programming. In each iteration of the *do/loop*, *lastSec* is updated to the current value of *time*, and then the LED is briefly lit. Next, in the *waitAsec* subroutine, the program just loops until the *time* variable increments. At that point, the subroutine returns and Task 2 loops again, so *lastSec* is updated to the new value of the *time* variable, etc., etc. The result is that the LED blinks almost exactly in time with the incrementing count.

There is one other difference in program function that results from the addition of the third task; the LED display's update frequency is slightly lowered due to the additional commands that are being executed.

However, the program is still more than fast enough to avoid any visible display flickering. I don't know (yet) how many tasks can be added, but when I reach the limit, you will be the second to know.

ARE WE DONE YET?

Nope, but we have run out of space this month! Next time, we'll take a look at a stripboard circuit that includes the N514RI and its multiplexing transistor circuitry.

As a result, we'll be able to simply insert the stripboard's 10-pin header directly alongside the 20M2 in any project that needs a two-digit display.

Hopefully, I'll also get to the update on my 4.5V battery-powered 20M2 project. See you next time ...

NV

QUESTIONS

WITH
RUSSELL KINCAID

& ANSWERS

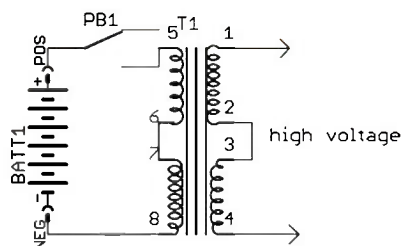
In this column, Russ answers questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. Send all questions and comments to: Q&A@nutsvolts.com

PET TRAINER

Some years ago, I found a published article that discussed a low current /high voltage circuit that I built, and that a friend used to discourage his pet dog from peeing on his flower bed. The circuit was battery powered, used a filament power transformer inserted backwards, and an SCR to generate a mild shock on an attached wire. The rest of the circuit details have faded in the passing years. I was wondering if you could come up with a suitable replacement circuit.

— Larry

FIGURE 1.



I tried the circuit in **Figure 1** and it worked, but if you hold the pushbutton down too long it will kill the battery. A capacitor in series doesn't work because the animal gets shocked when the pushbutton is released and the energy stored in the inductance is released.

Figure 2 includes a 555 timer producing a 20 millisecond pulse. The current through the transformer increases during the pulse time until the core saturates. I don't know if the core will saturate in 20 ms or not; you don't want the core to saturate because the current will be very high. You can improve the efficiency by reducing R2 until the shock becomes less. You can experiment on yourself if you are brave. I tried it on myself; it is a mild shock.

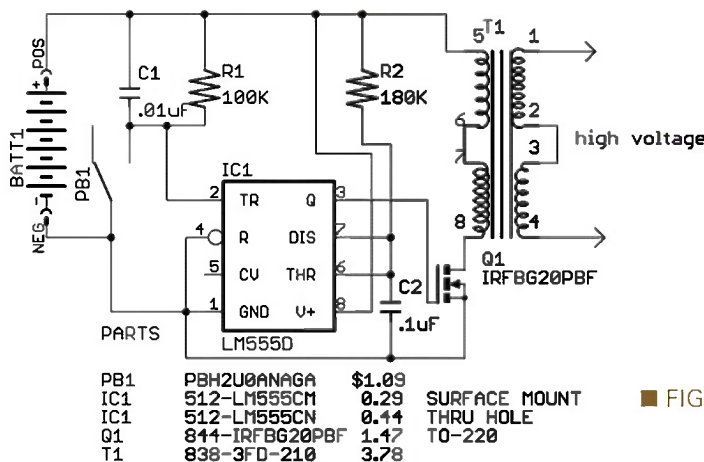


FIGURE 2.

BANDPASS FILTERS

I need a set of filters for upper and lower sidebands; the frequencies are as follows:

- The carrier is either 484.3 kHz or 494.3 kHz
- 484.3 kHz - 10 kHz = 474.3 kHz lower SB
- 484.3 kHz + 10 kHz = 494.3 kHz upper SB
- 494.3 kHz - 10 kHz = 484.3 kHz lower SB

- 494.3 kHz + 10 kHz = 504.3 kHz upper SB

I was thinking of using a 26A6 tube. The plate goes through an LC tuned circuit followed by the filter, then to another 26A6 grid. I don't know what the plate resistance or the grid resistance is. The 26A6 is useful because it can operate with 26.5 volts on the plate, as well as the filament. I am not able to find the data on the 26A6 but the 6BH6 is similar.

— Craig Kendrick Sellen

Figure 3 is the 484 kHz sideband filter schematic and **Figure 4** is the response curve. The 494 kHz filters will be the same except $L = 210 \mu\text{H}$ for the upper sideband. The lower sideband is the same as the 484 upper sideband where $L = 218 \mu\text{H}$. The components need to be 1% or better ordinarily, but I think it will work to tune each LC to the designated F_o .

For example: Use $L = 220 \mu\text{H}$ and tune the cap for resonance at F_o . F_o for each filter is listed as follows:

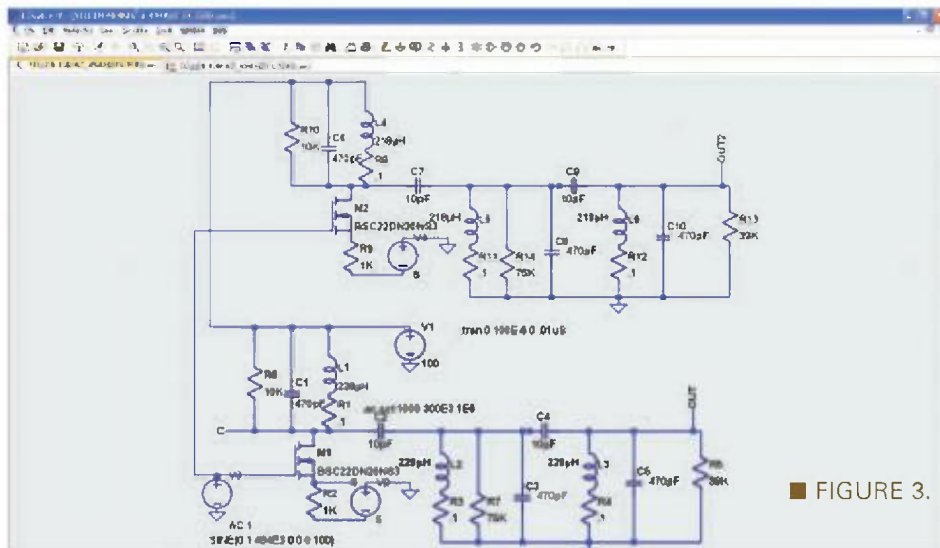


FIGURE 3.

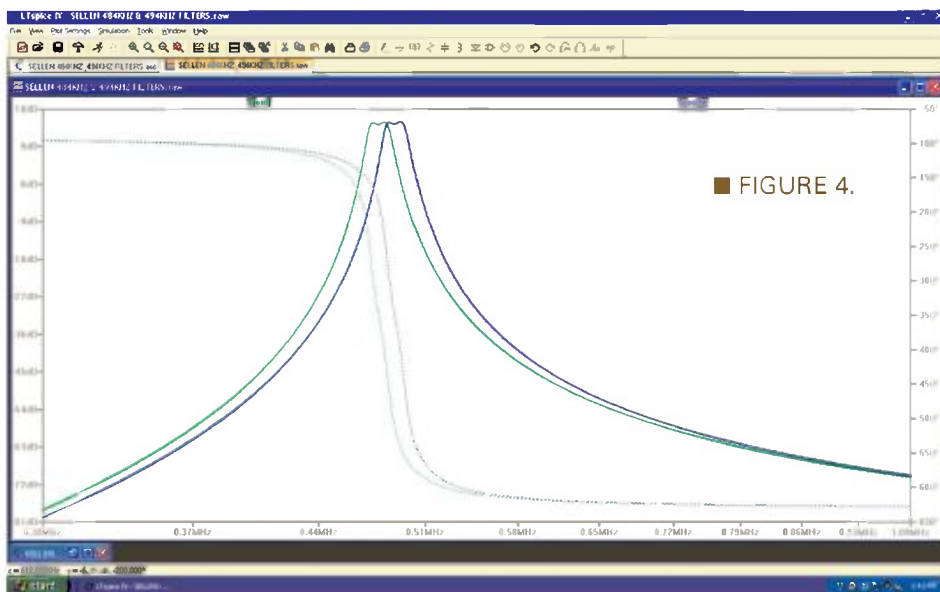


FIGURE 4.

- 484 lower sideband $F_o = 486.2 \text{ kHz}$
- 484 upper sideband and 494 lower sideband $F_o = 497.2 \text{ kHz}$
- 494 upper sideband $F_o = 506.6 \text{ kHz}$

The coil Q has to be high – above 50 and preferably 100; if it is not, you may need to adjust the resistor values to compensate.

I started with a stagger tuned filter but did not want to use that many tubes (or transistors), so I converted it to a top coupled design by fiddle and fudge.

A pentode tube should be

equivalent to the MOSFET that I used for simulation.

A PUSH ON/PUSH OFF CIRCUIT THAT WORKS

Q I have a Christmas music circuit with an existing push on/off (momentary contact) which I attempted to use to repair a snowman. I ended up using a slide switch and just energizing the circuit as needed. I still question why the transistor circuit would not have worked.

I built this circuit with

RadioShack 2N4401 and 2N4403 transistors. It worked only in that SW1 would start the circuit. Shorting it again would not stop the circuit. Shorting the junction R3, R6, and R7 to ground would turn the circuit off. The demo for it on YouTube showed the circuit working with a .1 cap between the output and +9 volts which was the author's fix for instability in the circuit.

I placed an indicator LED with a pull-up 330 ohm resistor in series at the output and – as stated above – was able to get it on/off using two different points for the switch. I placed his fix (the .1 cap) with no change. The coup de gras to its use was that loading the output left the circuit on permanently. I tried one meg loads on only the music circuit. No change with anything I tried.

To simplify: What I was attempting was to use a push on/off circuit to fire a 1.5 volt DC music module (a blob on a board circuit) designed for holiday cards or something similar to play music. It all worked except I was never able to find another simple circuit to do the work I needed.

A slide switch is okay, but given a circuit that works, I will modify the unit to work as I had intended it to. Any light you can shed on this project would be helpful, or do you have a better circuit?

All I need is a circuit that will apply 1.5 volts to the music circuit and push off using the same switch. There are several designs using two switches. The snowman has a built-in push on/push off switch. It is momentary, not latching. Thank you.

— Tim Edwards

Remember!

Send any questions
and/or comments to:

Q&A@nutsvolts.com

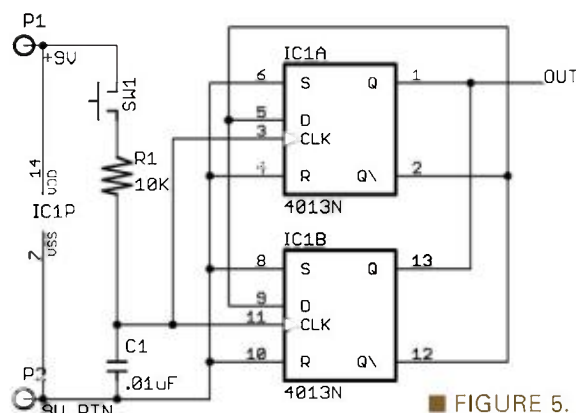
A I am sure you are aware that there are mechanical push on/push off switches; Mouser part 112-R13-512B2-BR is an example. An IC that comes to mind for the job is the D flip-flop (CD4013); see **Figure 5**.

The NOT-Q output is connected to the data input, and the data is transferred to the Q output on the rising edge of the clock input. Each time the button is pushed, the output toggles. I paralleled the two flip-flops to give more drive capability, then had to do something with the second flip-flop.

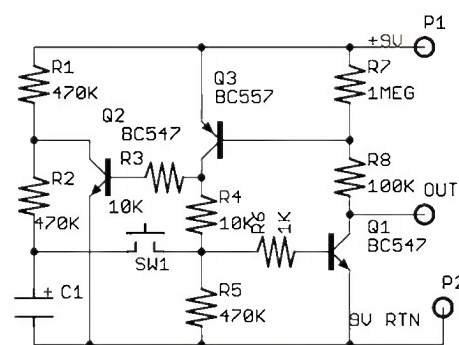
I looked at your circuit in **Figure 6**, and it ought to work. C1 is initially charged to nine volts so when the switch is closed, Q1 is turned on which turns on Q3 which, in turn, turns on Q2 and Q1. The Q1-Q2 loop is latched with the output low.

With Q2 on, C1 discharges through R2 with a time constant of 1/2 second. If the switch is held closed for more than one second, Q1 is turned off and the circuit goes back to the beginning.

I can't imagine that saturation storage time is the problem, but if you replace the bipolar transistors with enhancement mode MOSFETs that problem will go away.



■ FIGURE 5.



■ FIGURE 6.

DC SERVO QUESTION

Q I want to control my 3 HP DC motor using PWM with a microcontroller. The motor is Baldor CD1803R and I will be controlling it with knobs and switches. I want to do these things:

- Control the direction of rotation (CW and CCW)
- RPM control
- LCD display (all of the above information)

It will not decrease speed when loaded, hence I want to use speed regulation, standard feedback, and closed-loop control.

— Selahattin ŞADOGLU

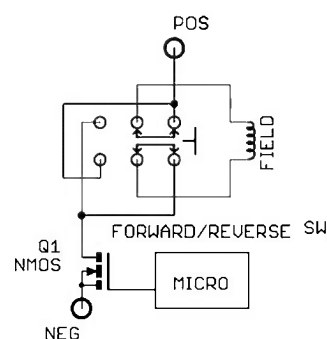
A I can't produce a detailed circuit for free, but I may be able to suggest a method. It has been nearly 60 years since I studied servos and the industry has changed a lot. The datasheet only tells me that the

motor is rated 3 HP, 180 volts; the field is rated 100/200 volts (I don't know what that means); and the field current is nominal one amp. That is not nearly enough to do a proper design, but I can offer some generalities. I have not mastered LCD displays, so I won't talk about that.

I assume this will be operated from full wave rectified single phase voltage; you don't have to filter the DC, but efficiency and power will be greater if you do. Unregulated DC will be fed to the armature and the control current will be applied to the field. For feedback, I recommend a segmented disk attached to the shaft with an opto sensor to produce a frequency proportional to RPM. For the reference RPM input, a keypad or rotary coded switches could be used. Four rotary BCD switches could set the RPM from 1 to 9999. The PWM will be applied to the field with a power MOSFET and diode to catch the backswing; 25 kHz will be high

enough to not produce an annoying sound. All you need to do to reverse the direction of rotation is reverse the field voltage; refer to **Figure 7**. Some means of removing power if the field current drops to near zero will be needed to prevent a runaway motor.

If it were me, I would do it all analog; a generator on the shaft to supply the field and a rheostat on the generator field to control the



■ FIGURE 7.

MAILBAG

Re: Burned Parts on Weed Whip Charger, December 2012, page 20.

In your December column, you analyzed a small battery charger in a weed whip. While your circuit analysis was correct (as always) based on a TO92 NPN transistor for the pass element, I believe the original part was an SCR. The give-away was the remaining letters "MCR." Motorola used to make a line of SCRs in TO-92 packages. The MCR100 series were .8A, and the MCR22 series were 1.5A. These are sensitive gate types requiring only 200 microamps of gate drive.

Looking at the circuit again with an SCR this time, it certainly looks more plausible. The zener clamps the gate to 15 volts. There are two diode drops (V_{GK} and the pass diode) between this reference and the battery. When the battery voltage drops to this level (approx 13.6V), current flows through the gate-cathode junction and fires the SCR for the remainder of the cycle. This continues at line frequency, until the battery voltage is pushed up closer to the reference and no gate current flows. The SCR remains off until the battery voltage falls again. Now would be a good time to mention that the circuit will only function with a pulsating DC input! The input must approach zero to commutate the SCR off every cycle. So, NO filter capacitors in the wall wart! The SCR has an on-state voltage drop of around one volt at 400 mA, combined with around a 90° conduction angle; dissipation should not be an issue.

The main drawback to the original circuit is the backward (positive) temperature coefficient. The reference zener voltage rises with temperature on one side, and V_{GK} with the pass diode's drops becomes smaller on the output side. The result is the charging voltage goes UP with increasing temperature, leading to overcharging. The opposite scenario leads to undercharging — exactly opposite of what a lead acid battery wants for a long life.

I have actually built similar circuits as drop-in additions to small unregulated chargers, and given them to friends for their classic muscle cars. They are used to float a charged battery between infrequent drives; see the **schematic**. The main difference in my circuit is the substitution of a TL431 "adjustable" zener for the fixed zener diode. This allows for a precise voltage adjustment to match the battery type (i.e., wet, gel, maintenance free, etc.). Temperature compensation is accomplished with the TL431 being essentially flat, along with D2 in series with the reference terminal. R5 provides a small current in D2. The battery voltage is divided down to the reference by R2, R3, and R4. R1 provides gate drive and operating current for the TL431. S1 and R6 were added to raise the output voltage to

an absorption charge level. The optional "charged" indicator lights up when the battery is at the set point. Calibration should be done close to 25°C/77°F — that is where many battery manufacturers specify their charge/float voltages. The tempco isn't exactly textbook, but it is about negative 10 mV/°C — about half of what it really should be. An interested reader could add another diode in series with D2 (and fiddle R3), and it should bring it in.

I hope this helps Mr. Stenlund and the many other readers of my favorite N&V column.

Tim Young

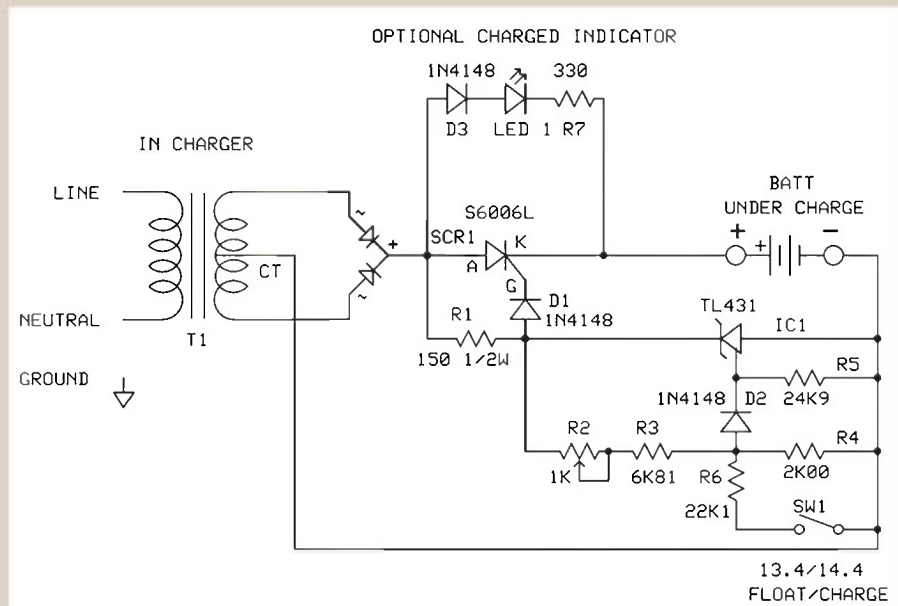
Thanks for writing, Tim. I had not considered that Q1 was an SCR, but it makes a lot of sense as you describe it. I am sure readers will appreciate your schematic.

Re: Voltage Converter question, January 2013, page 24:

The reason Q1 shows a four volt drop is that it is not turned on all the way. The transistor as shown in Figure 1 is an NPN transistor that is located on the high side of the load. To turn on, it needs more voltage than is on the collector. By putting it on the low side of the load (i.e., the emitter to negative and the load between positive and the collector), the transistor will be turned on. It will also need a 1K or so resistor on the base. It would be easier to use an N-channel MOSFET and not have to worry about biasing.

Mark Arnold

I suspect you are right, Mark. I would expect a 1.2 volt drop across Q1 in the best case, and more if the drive were not sufficient. The circuit is switching, so I stand by my original answer, but your suggestion to connect Q1 from the load negative to negative voltage is a good one. The MOSFET idea is also good. Thanks for writing.



generator output and set the RPM. You might want some amplification to tighten the loop.

VOLTAGE REDUCER

Q The voltage converter/reducer in my golf cart is malfunctioning. This is a sealed unit with no access for repairs. A commercially available replacement costs \$180. I need to take 48 volts DC from the battery pack down to 12 volts at 10 amps so I can operate my headlights and taillights. Can you provide a reasonably priced circuit?

— Michael Herman

A Certainly. A 556 dual timer and power switch should do it. I don't think it will hurt the filaments to put 48 volts on them for 10 μ S, but if — heaven forbid — the lights burn out

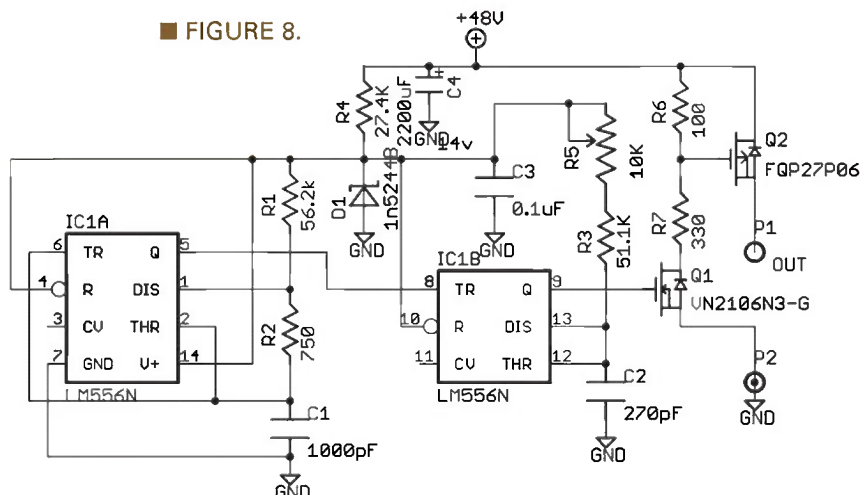
too soon, you can put a 10 amp 33 μ H choke in series. More inductance would be better, but is hard to find.

In **Figure 8**, the first section of the 556 runs at 25 kHz and clocks the second section which has a nominal pulse width of 10 μ S.

I could provide some feedback

to the control input of the second section to regulate the output voltage but I think that is overkill. The variable resistor will allow you to adjust the output to 12 or 12.6 volts. You can read the voltage with an analog meter; a digital meter probably won't work. **NV**

■ FIGURE 8.



We don't just say
our Geiger counters are accurate,
we can NRC certify it.



Also Available

- Uranium Ore
- Radioactive Isotopes
- Geiger Counter Tubes
- DIY Kits (Geiger Counters)
- Sled and Isotope holders
- Alpha Particle Spark Detector



The GCA-07 digital Geiger Counters make it easier to teach nuclear physics with its proven accuracy!

The only digital Geiger counter in its price range accurate enough for NRC certification.

NRC
Certification
Available

Applications

- Education – Classroom demos and experiments. Government Institutions
- Emergency Services – EMT, HAZMAT
- Domestic Preparedness
- Compliance Verification

Features

- Secondary Indicators – audio (clicks) and visual (LED). Also includes TTL serial/pulse data port, headphone and external power jack
- Low Battery Indicator
- Windows Radiation Graphing Software & TTL Serial to USB Cable Included



Images Scientific Instruments, Inc.
800-230-4534 • www.imagesco.com

RF Specialists

Affordable RF Modules

Introducing the LMX-ISM-242-SR and LMX-ISM-242-LR



Data Loggers

Stand Alone and Wireless Mesh Networking Logger



Industrial Bluetooth

OEM, Modules, Wireless Device Servers, RS-232 Long range options, low cost

ZigBee Pro

OEM Modules and USB ZigBee Sticks. Mesh Networks



Low Power, High Performance Semiconductors

Excellent Sensitivity, Very Efficient Power Amplifiers

Narrow Band, General Purpose, TSSOP, RF Microcontrollers
Typical Sensitivities are in the range of -130dBm to -126 dBm for 1.2 kbps

RF Design Services

Prepared to work with your in-house engineers, or support your RF project from initial design to implementation.

idw • Military • Space • Medical • Smart Grid Metering • SCADA • Lighting Control



LE MOS
INTERNATIONAL

www.lemosint.com
866.345.3667
sales@lemosint.com

"Making your RF ideas into profitable products."

NEW PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS

USE HUMAN BODY AS SECURE COMMUNICATION CHANNEL

Microchip Technology, Inc., has announced its BodyCom™ technology which provides designers with the framework for using the human body as a secure communication channel. Compared to existing wireless methods, BodyCom technology provides lower energy consumption, while further increasing security via bidirectional authentication.

Because no RF antennas are required, BodyCom technology allows for simpler circuit-level designs and a lower bill of materials. All of this is enabled by the BodyCom Development V1.0 Framework which is supplied through free software libraries that work on all of Microchip's eight-, 16-, and 32-bit PIC® microcontrollers.

BodyCom technology is activated by capacitively coupling to the human body. The system then begins communicating bidirectionally between a centralized controller and one or more wireless units. There are a broad range of applications where secure wireless communication is essential, and the human body can provide a secure channel. This is especially true when you add bidirectional authentication that supports advanced encryption, such as KeeLoq® technology and AES. For example, BodyCom technology helps prevent the "Relay Attack" problem that is typical in automotive passive-keyless-entry security systems.

BodyCom technology significantly increases battery life by eliminating the need for a wireless transceiver or high power inductive fields. It also simplifies development and lowers costs by not only making antenna design unnecessary, but also by using a low frequency framework with a common microcontroller and standard AFE frequencies (125 kHz and 8 MHz); no external crystals are



needed. Because it complies with FCC Part 15-B for radiated emissions, BodyCom technology eliminates the cost and complexity of certification.

Some example applications include access control (security systems, home/industrial door locks, pet doors); personal safety and security (equipment access/disable, power tools, firearms, computer systems); and consumer (profile management for gaming consoles and exercise equipment).

To further enable development and speed time to market, Microchip now has the BodyCom Development Kit (part # DM160213) available for \$149. It comes with a central controller unit and two wireless mobile units.

There is also a free BodyCom Development V1.0 framework which comes with a communication library, application code examples, and a development GUI for use on PCs.

For more information, contact:

Microchip Technology

Web: www.microchip.com

16 REED RELAY OUTPUT DIGITAL INTERFACE

The new 8003e PCI Express digital I/O interface from Sealevel

features 16 reed relay outputs that provide high quality, long life dry contact switch closures suitable for low current applications up to 10VA. Reed relays normally open and close when energized. This board is PCI Express X1 compliant and is

compatible with any PCI Express slot.

Sealevel SealIO Classic software drivers and utilities make installation and operation of the 8003e simple by using Microsoft Windows or Linux operating systems. Software for standard PCI boards will also work

with PCI Express boards. The 8003e is available for \$339. Also included is a 72" cable with a DB78 male connector to DB37 male connector, and optional terminal blocks simplify testing and field wiring. Standard operating temperature range is zero to +70°C. The 8003e is backed by a lifetime warranty.

For more information, contact:
Sealevel
 Web: www.sealevel.com



HIGH PERFORMANCE ELASTOMER SOCKS

Ironwood Electronics has introduced a new high performance QFP socket for 0.4 mm pitch 128-pin QFPs. The SG-QFE-7011 socket is designed for a

14 mm x 14 mm x 1.6 mm package size with 16 mm x 16 mm lead tip to tip, and operates at bandwidths up to 10 GHz with less than 1 dB of

Continued on page 41

SHOWCASE

Recycling & Remarketing High Technology

WEIRDSTUFF[®]

WAREHOUSE

Software, Computers, Electronics, Equipment, Doodadicks

WE BUY/SELL EXCESS & OBSOLETE INVENTORIES

FREE COMPUTER AND ELECTRONIC RECYCLING

GIANT 10,000 SQ. FT. AS-IS SECTION

384 W. Caribbean Dr.
Sunnyvale, CA 94089
 Mon-Sat: 9:30-6:00 Sun: 11:00-5:00
 (408)743-5650 Store x324
WWW.WEIRDSTUFF.COM

SDPSI sdp-si.com
 800-819-8900

FREE CATALOG

Small Mechanical Components

SETTING IDEAS INTO MOTION

QKITS.COM

36 LED Electronic Roulette

Great for holiday displays!

All 36 LEDs will light up and run in a circular manner. LEDs will run in high speed motion when pressing the switch, and gradually slows to a stop after release.

~~\$18.95~~
\$29.95

Quantity discounts available

1-888-601-4-KITZ

NKC electronics

THE OFFICIAL ARDUINO STARTER KIT!

NKCElectronics.com/starterkit
 Includes the Arduino Project Book (170 pages)

Purchase Orders are accepted from Educational Institutions
 US Government and Research Centers

MAKERplot[™]

Data Acquisition and Graphical Plotting Software for Microcontrollers

- Works With Popular Microcontrollers
- Plots & Logs Analog & Digital Data
- 3D Directional Data Acquisition & Control
- Interfaces With Thermocouple Standard
- No proprietary hardware required
- Completely Customizable
- "Make" Your Own Custom Interfaces
- Supports Full Color Monitors, Relays, Switches, LEDs and Sensor Controls To Choose From

Got Data? - MakerPlot It!

\$ 39.00 - Single User License
 \$199.00 - Lab Pack License

www.makerplot.com

www.Primecell.com

Battery rebuilding service

Dead Batteries? Don't toss them. Send them to us - our rebuilds are better than original specifications.

Tools

Hilti Skil Milwaukee Panasonic 180 DeWalt Makita All 2-36 Volts

Electronics

Bar Code Scanners Surveying Printers Laptops Photography

Radios

APELCO UNIDEN G.E. ICOM KENWOOD MOTOROLA MIDLAND MAXON YAESU ALINCO

Visit www.primecell.com for important details

24 Hr Secure recorder tel-fax (814) 623 7000
Quotes email: info@primecell.com
 Cunard Assoc. Inc. 9343 US RT 220 Bedford PA 15522

ROBOGAMES



Compete at RoboGames!

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

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

Be a RoboGames Sponsor!

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

Rent a Booth!

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

World's Largest Robot Competition
-Guinness Book of Records

North America's Top Ten Geek Fests
-Wired Magazine

SportCenter's Top Ten
-ESPN SportsCenter

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

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

Events:

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

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

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

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

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

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

Tetsujin (ExoSkeleton): Lifting, Walking, Carrying

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

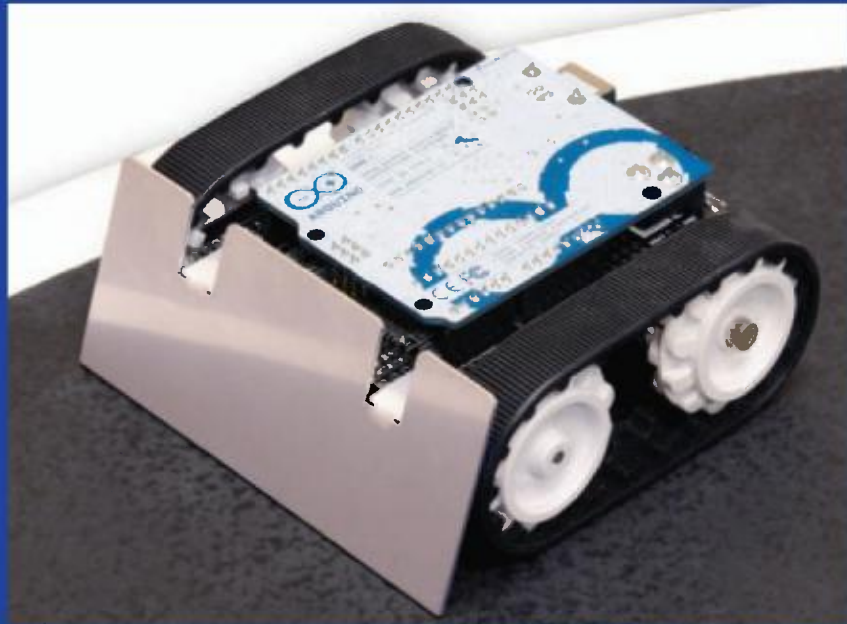
BEAM: Speeder, Photovore, RoboSapien Hacker



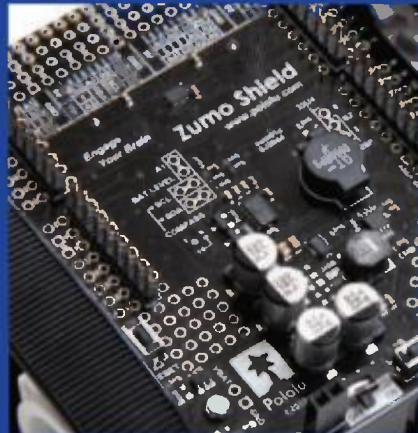
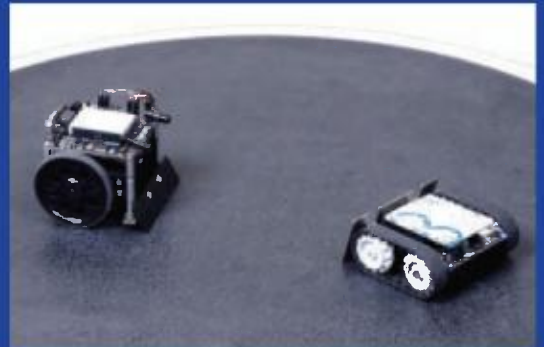
April 19-21, 2013

San Mateo, CA

<http://RoboGames.Net>



Build and customize your Zumo!



Small enough for Mini-Sumo; flexible enough to make it your own.



Put your Arduino or compatible controller on the right tracks with the Zumo chassis and Arduino shield! The Zumo is a small, tracked robot platform that works with a variety of micro metal gearmotors to allow for a customizable combination of torque and speed. Add a Zumo shield, which includes a dual motor driver, buzzer, and three-axis accelerometer and compass, to make an Arduino-controlled robot that can really throw its weight around!

 **Pololu**
Robotics & Electronics

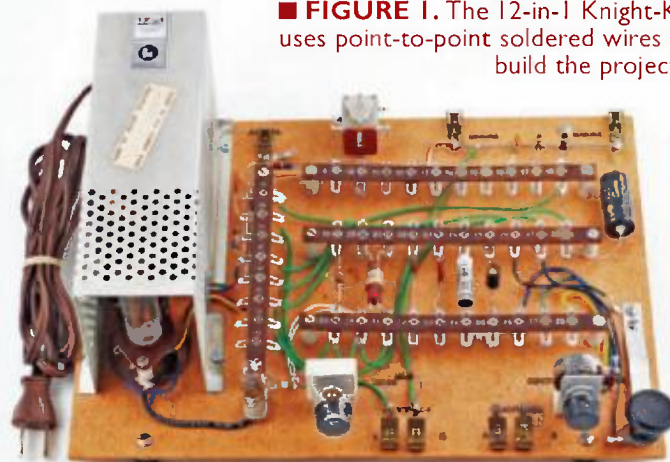
Find out more at www.pololu.com/zumo

THE RETRO-SHIELD: WHERE THE PAST MEETS THE PRESENT

By David Goodsell

Several months ago, I bought a vintage Allied Radio Knight-Kit 12-in-1 Electronic Lab on eBay. It used a 12K5 low voltage vacuum tube instead of transistors, as shown in **Figure 1**. As I was wiring up my "wireless AM broadcaster" project, it hit me. Why not combine the past with the present? Put an Arduino with a vacuum tube shield — a Retro-Shield!

■ **FIGURE 1.** The 12-in-1 Knight-Kit uses point-to-point soldered wires to build the projects.



The Vision

Actually, this idea didn't just come out of the blue. I had been puttering around with an Arduino Uno from RadioShack, and since the 12K5 vacuum tube only needed 12 volts to operate, I made the connection.

In my head, I could see the end result. It would be a stack of shields with the vacuum tube on the top board, glowing in the dark. I wasn't sure yet what it was going to do, but I knew it would be cool.

Fast Forward

Before I get into the details, I want to say how much fun this project turned out to be. First, I successfully dissected a vacuum tube and learned how it worked. Secondly, I had fun broadcasting silly messages around the house, along with Christmas music that at the time made our holiday very special. The talking clock drove my wife nuts, but I loved it. And finally, I built my first Arduino shield and managed to get it working just the way I wanted. What more could I ask for?

An mp3 sample of my broadcasts can be found at the article link. The first portion of the sample is a sentence generated by the talking clock; the second section is from a music player; and the last part is my voice, using a

microphone. The sequence is repeated twice. I think you'll enjoy it.

Back to Business

Now, back to the project. How could I possibly wed the broadcaster to an Arduino? I remembered some ads for sound-generating shields that produced voices and weird sounds for robots and talking clocks. A GinSing Sound Synthesizer shield looked interesting, but the demos sounded a little too unnatural for me.

The SpikenzieLabs VoiceShield had a different approach. It simply stored digitized audio, then played it back when triggered by an Arduino. The VoiceShield's memory could hold four minutes of audio. One application for it was a talking clock. You could load it with 80 prerecorded words, such as: the, time, is, now, one, o'clock. Or, you could put in any audio you wanted.

So, at last I had a vision of the final configuration; take a look at **Figure 2**. It would be three boards: an Arduino on the bottom, a VoiceShield in the middle, and the broadcaster on top. I figured that once it was finished, I would program the Arduino to announce the exact time — once every five minutes — to all the radios throughout the house. I was sure my wife would love it.

■ **FIGURE 2.** A glass dome is great for display and the inkjet printed label gives it a nice title.

12K5 Low Voltage Vacuum Tube

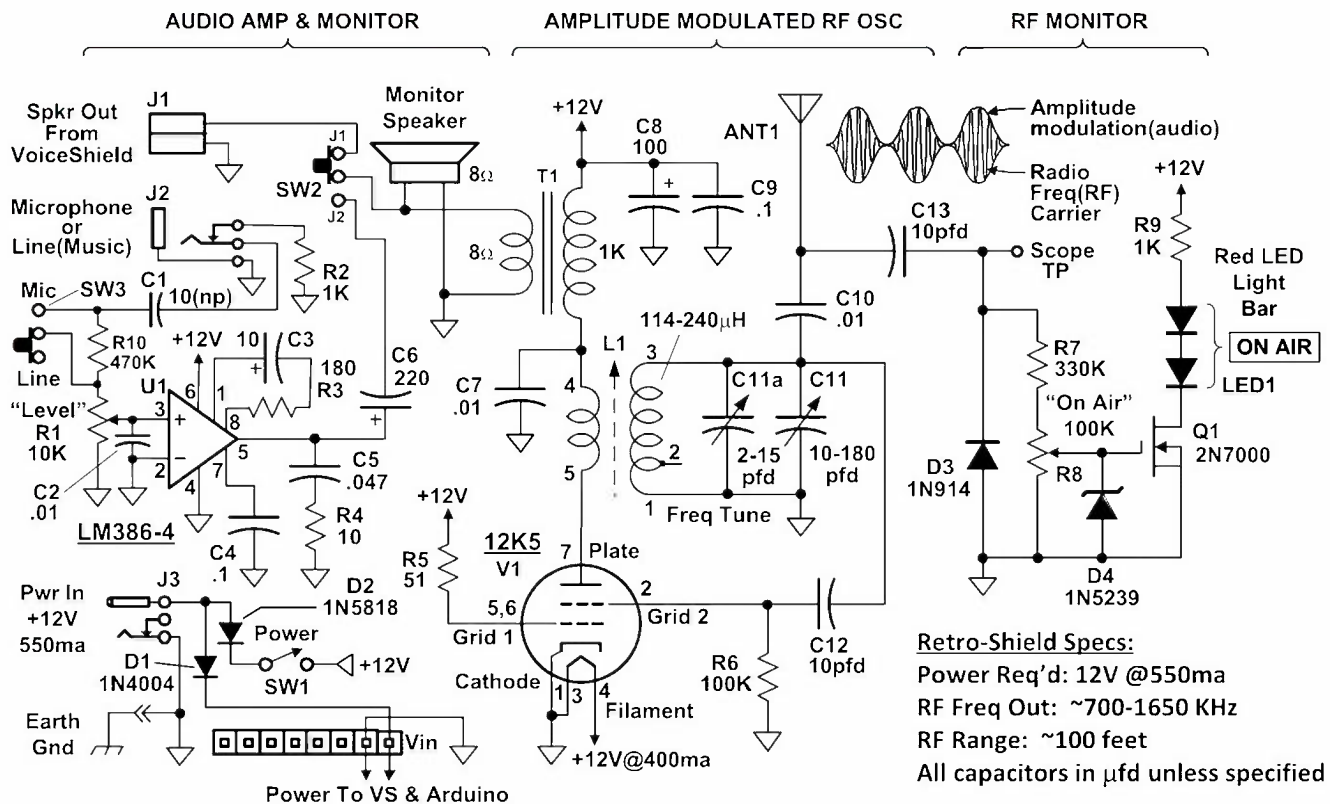
Most vacuum tubes require 100-300 VDC to operate, which sometimes presents a hazard for experimenters — young and old. If you aren't careful, you could get a nasty shock if a finger brushed against a 300 volt terminal.

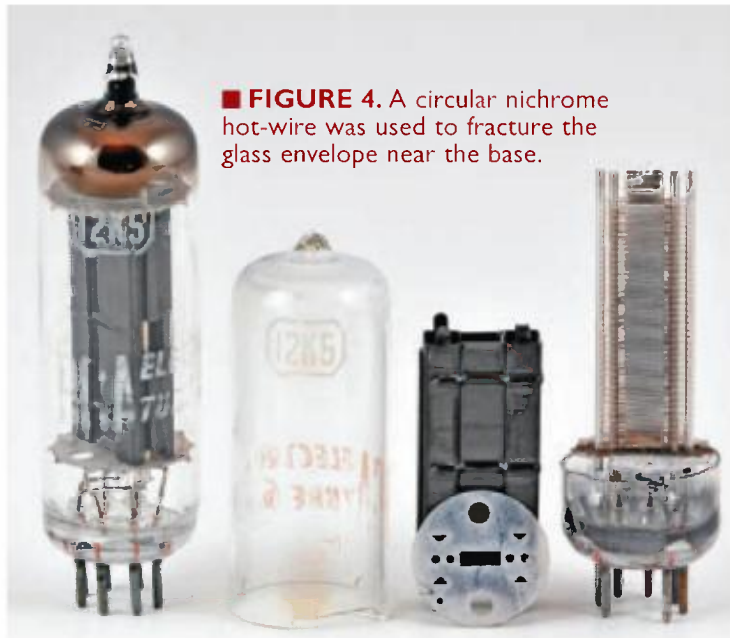
The 12K5 tube used in the Knight-Kit was different. It only required 12 volts for the filament, grids, and plate. The schematic in **Figure 3** shows how the elements of the tube are connected in the Retro-Shield. Please note the element names: filament, cathode, grid 1, etc.

Back in the 1950s, a company named Tung-Sol developed a line of "Space Charge" tubes just for car radios. They only needed the low voltage from the battery to operate — no high voltage. The tubes were only sold for a few years, however, because transistors were rapidly improving, and soon car radios were completely transistorized.



■ **FIGURE 3.** The Retro-Shield can accept audio inputs from three different sources.

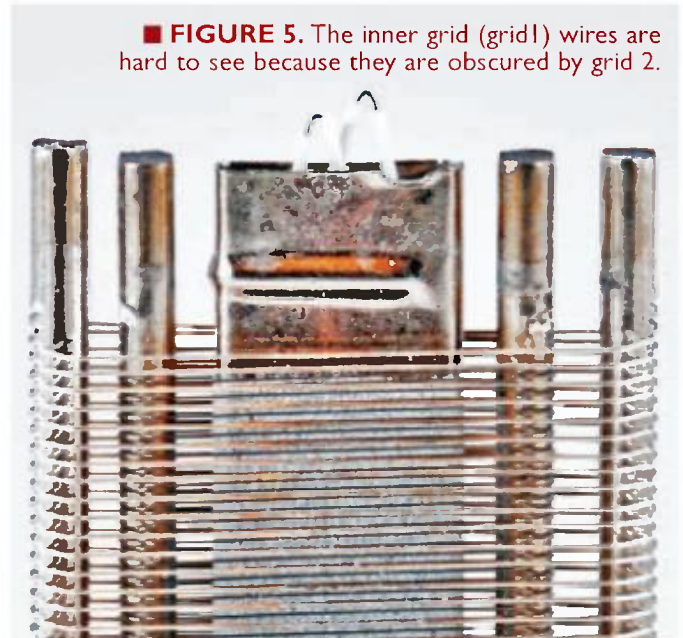




■ **FIGURE 4.** A circular nichrome hot-wire was used to fracture the glass envelope near the base.

In case you're not familiar with the parts of a tube, please take a look at **Figures 4-6**. In **Figure 5**, the cathode is the silvery hollow tube in the middle, with a folded alumina coated tungsten filament tucked inside, but peeking out the top. The two grids are the fine wires which are spiral-wrapped around the shiny support rods and nested inside of each other. In **Figure 4**, the plate is the hollow black ribbed rectangular structure and the round piece in front of it is a piece of thin mica — an insulator. During assembly, the plate is carefully slipped down over the grids and held in place by mica disks at the top and bottom. The whole thing is an intricate structure all nested inside of each other, but not touching.

Each of the elements is spot-welded to the pins in the glass base, and a glass envelope is slipped down from the



■ **FIGURE 5.** The inner grid (grid 1) wires are hard to see because they are obscured by grid 2.

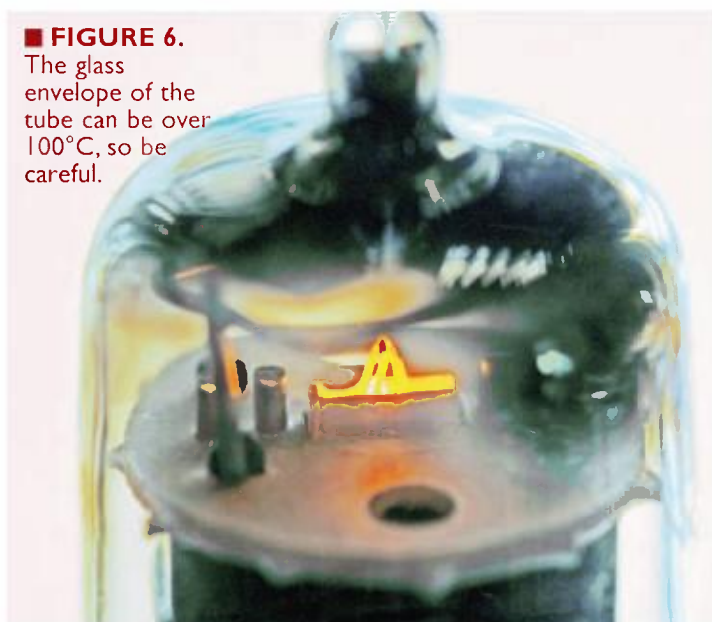
top. The envelope is sealed to the base with a torch, and then the air is sucked out through a small glass tube at the top. Another torch melts the small tube closed, sealing the vacuum inside. You can see the melted tip in the upper left of **Figure 4**.

In operation, 12 volts is applied to the filament and heats it up to over 1,000 deg C, as seen in **Figure 6**. The hot filament, in turn, heats the oxide-coated cathode which boils off a cloud of negative electrons — called the space charge. Grid 1 — which is hooked to +12 volts — is located very near to the space charge region and attracts most of the electrons to it. However, some do get through.

The accelerated electrons that get through grid 1 continue on through grid 2, and are attracted to the plate which is also hooked to +12V. The net result is that more electrons get to the plate as a result of grid 1 than if there was just the plate to attract them. Plus, these space charge tubes run the filament at a hotter temperature and have an increased cathode area in order to produce even more electrons than usual.

Now, if a small negative voltage (-4V) is applied to grid 2, it will repel the electrons back towards grid 1 and the cathode, shutting off some or all of the flow to the plate. In other words, a tiny change in the negative voltage on grid 2 makes a big change in the current flowing to the plate. This is how a tube can amplify a signal. However, in a practical amplifier circuit, additional components are needed to convert the plate current into a voltage and to bias the tube into the desired operating region.

The bottom line is the 12K5 is suitable for certain applications, but it's not as powerful as tubes that use high voltage. In the Retro-Shield, the 12K5 serves a dual purpose. L1, C11, C12, and R6 form a radio frequency (RF) oscillator tuned to the AM radio band. In addition,



■ **FIGURE 6.** The glass envelope of the tube can be over 100°C, so be careful.

the audio frequency signal coming into T1 modulates the plate voltage, causing the amplitude of the RF output to go up and down with the audio.

Searching for Parts

Before getting too excited about this project, I needed to find sources for the critical parts. The Knight-Kit was over 50 years old, and I wondered if the 12K5 and oscillator coil (L1) were still available. A good friend turned me on to a company called Antique Radio Electronics. Zowie! It was tube heaven! They had all kinds of cool things. A 12K5 was only \$3.95 and the coil was \$8.95. I'm not sure how many of these items they keep in stock, so you might make sure they are available before you get started. I noticed that the 12K5 also showed up on eBay and MDB Ventures.

Improving Output Power

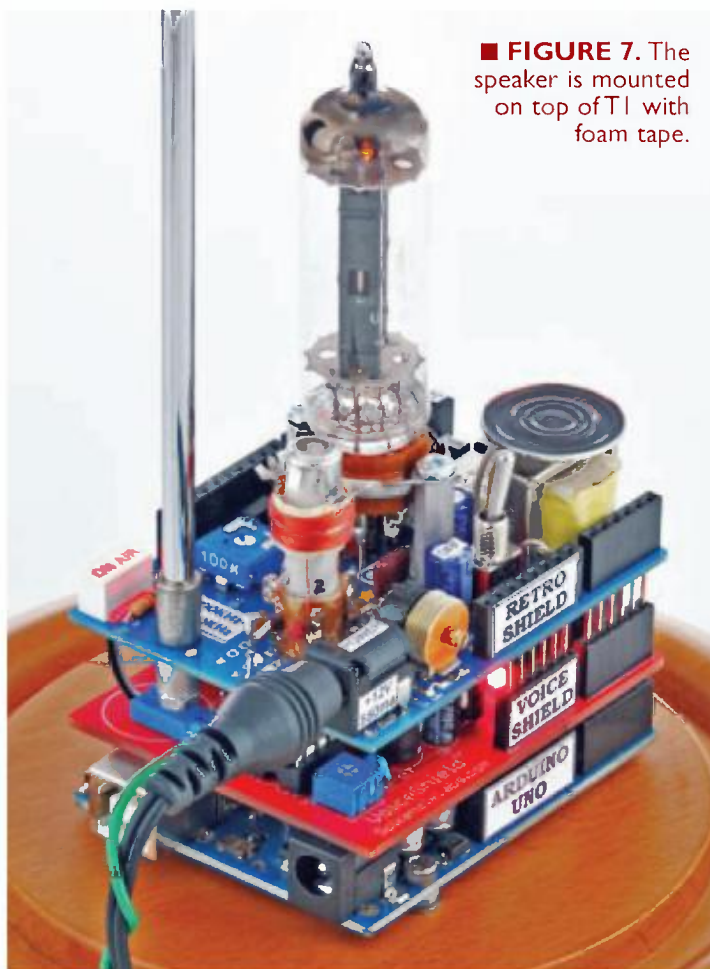
The only hitch with the Knight-Kit broadcaster was range. It didn't transmit very far. Of course, the little telescoping antenna was woefully short. I built a proper loading coil but it was quite large and not practical, so I gave up on that approach. Boosting the current through grid 1 to 56 mA raised the plate current to 4.1 mA and improved the RF output. I found that a good earth ground also helped reduce the 60 Hz pickup and added a ground plane. The ground (green) wire is shown in **Figures 7** and **9**. Finally, I changed the output capacitor from the original 100 pF to 0.01 μ F, which raised the output considerably. The improved signal was strong enough to reach all the radios in the house, but only if they had decent receiving antennas. I was unable to get an output power measurement that I believed, but I compared its output to a 100 mW Ramsey AM Broadcaster Kit and both seemed about the same.

FCC (Federal Communications Commission) Part 15 specifies a limit of 200 feet for an unlicensed AM transmitter. I also saw a limit of 100 mW in the FCC OET Bulletin No. 63. Either way, the range of the Retro-Shield should not interfere with your neighbor's radios. If you think there is a problem, simply reduce the value of C10.

Now, a word about AM radios. Most home stereos definitely need an external AM antenna to receive signals from the Retro-Shield. Some stereos come with a goofy little AM loop antenna, but it's better than nothing. Failing that, a 10' hank of wire would probably be enough. I used a small transistor radio to check the reception throughout the house, and the signal was pretty good everywhere.

Building the Retro-Shield

Of course, you don't have to put together a stack of



■ **FIGURE 7.** The speaker is mounted on top of T1 with foam tape.

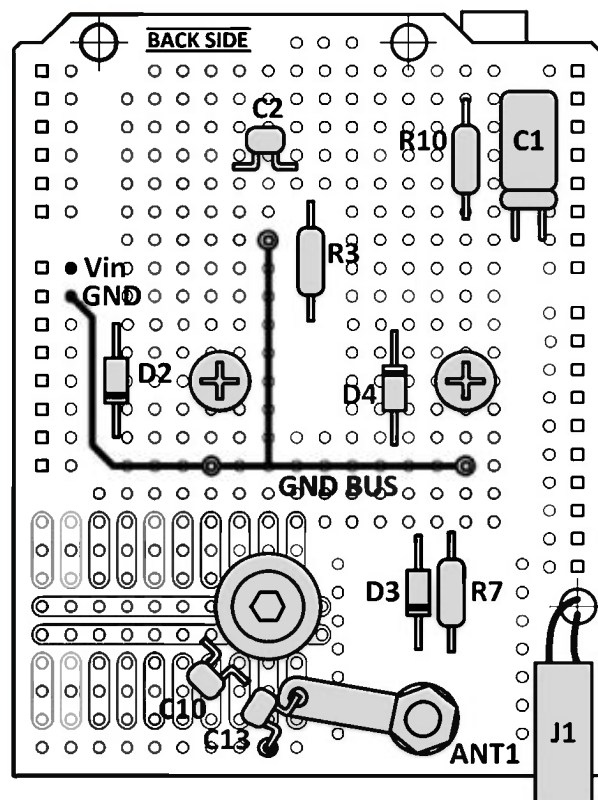
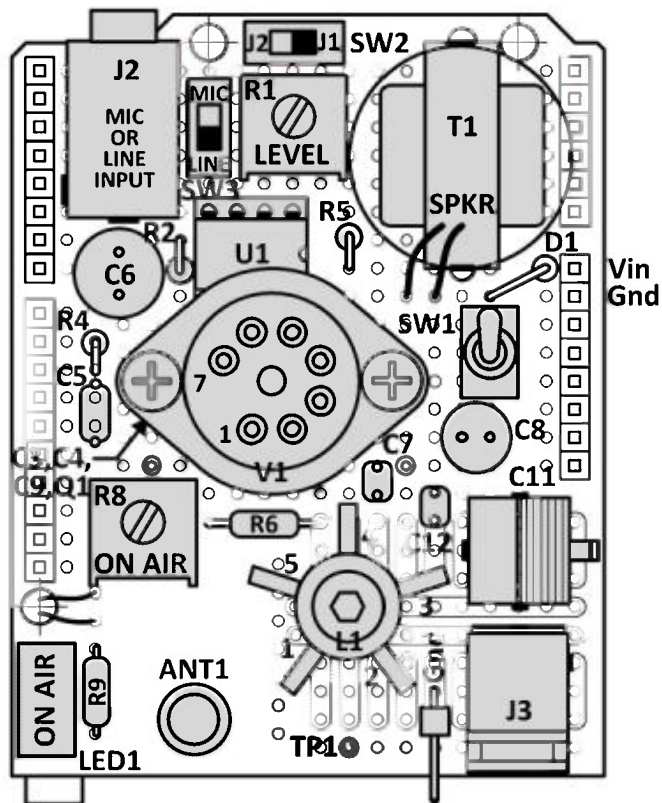
three boards as I did; you can build the broadcaster as a stand-alone unit. I sometimes use it to send iTunes from my PC to the radio in the garage. Next Halloween, I plan to pipe some scary music to a portable radio outside.

This project is not a slam dunk. You have to use your imagination and expertise to build and test the system. The Retro-Shield has both high (1 MHz) and low frequency (audio) signals running around. Try to keep the RF signals as isolated as possible. The LM386 IC had some pickup problems at first, but a friend suggested adding C2 and it solved the problem.

I selected the Arduino proto shield for the top broadcaster board, just to keep it in the Arduino family. A piece of perf board would be fine too, since only two pins (Vin and GND) are needed to carry the 12 volt power down to the shields below. A ground plane perf board would probably be even better. J1 routes the audio up from the VoiceShield.

The proto shield board has a small number of tiny traces that carry signals and power to various places. All these traces should be cut so they do not interfere with the broadcaster circuitry. The layout is shown in **Figure 8**. I didn't show individual connections because it would be much too busy. Just use point-to-point wiring; the components are grouped. The parts on the bottom of the

■ **FIGURE 8.** The ground pin next to J3 connects the board to earth ground.



board are shown in their approximate positions. Some holes need to be enlarged for V1, T1, L1, J1, and ANT1. Extra holes are also needed for some leads. If you decide to include C11a for fine-tuning, solder it in as a piggyback on top of C11.

Final Testing

First of all, L1 is shipped with the ferrite tuning slug at the very bottom. It needs to be screwed upward until it is roughly centered in the coil. Be sure to use the 0.100" nylon hex tool (Tool2). Allen wrenches are not the right

size and can crack or chip the slug. Later on, the slug can be used to fine-tune the RF output frequency.

Set SW3 at "line," SW2 at "J2," level pot (R1) to minimum, and the On Air pot (R8) to max. Next, position an AM radio within a few feet of the Retro-Shield and set it at 1,200 kHz or a quiet spot near there. Connect the 12 VDC adapter to J3 and turn on the power switch (SW1). After several seconds, you should see the filament start to glow like in **Figure 7**.

Please note that the glass envelope gets pretty toasty – 100 deg C – so don't touch it until it cools off. Now, you are ready to get serious.

To test the broadcaster, you'll need a source of music – like an mp3 player or the headphone output of your computer or stereo. Plug the music source into J2. Slowly turn up the level pot until you hear the music coming out the tiny monitor speaker on the Retro-Shield. If you don't hear anything, turn the level pot back to zero and check that music is truly coming through the cable.

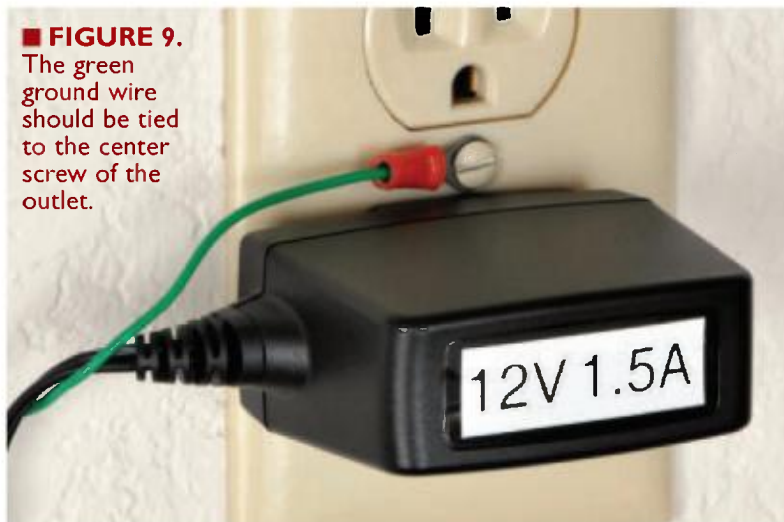
Be careful not to overdrive the LM386 because it will draw several hundred milliamps and could go up in smoke. The trick is to turn the music up just loud enough to hear it, but without any distortion.

Be sure to use the LM386-4 which is rated for 5V to 18V. The gain of the LM386 is set at 200 for the microphone and attenuated by a factor of 48 for the line input.

Here's a cool thing. It takes about 20 seconds

■ **FIGURE 9.**

The green ground wire should be tied to the center screw of the outlet.



for the 12K5 tube to warm up, so don't expect the RF to immediately come blasting out like with a transistorized unit. Just be patient. After warmup, the On Air LED should light, indicating that RF is being generated. If you have a scope, clip it on the test point to see the RF waveform.

During development, I had trouble with the 2N7000 FET (Q1) in the RF monitor section. It kept blowing out. It might have been static electricity or too much RF. After I added a zener (D4), the problem went away. I suggest that you keep a spare FET on hand.

Next, adjust the tuning capacitor (C11) slowly until you hear the music coming out of the radio. The AM broadcast band is from 530 to 1,710 kHz. C11 tunes between 720 and 1,710 kHz. Turn the level pot up slowly until the music starts to overmodulate the RF, then back it down a bit. A scope is real handy for setting the modulation level.

Now that the broadcaster is working, let's look at the VoiceShield.

Power and Programming Cables

First, let me give you a quick explanation about how to connect the USB and power cables.

Retro-Shield Only: Connect the 12 VDC at 1.5A AC adapter to J3. The 12K5 has a surge current of about two amps, which drops to under 500 mA after warmup.

Arduino Uno and VoiceShield: Connect a USB cable to the Uno for programming and power.

Stack of Three Boards: Connect a USB to the Uno for programming. Connect the AC adapter to the Retro-Shield to provide power to all three boards, down through the Vin pins.

The VoiceShield

On the VoiceShield board, you'll need to solder a two-pin right angle header to the speaker out terminals,

PARTS LIST

ITEM	DESCRIPTION	DIGI-KEY (unless otherwise noted)
C1	10 μ F, 25V, non-polarized	P1176-ND
C2,7,10	0.01 μ F, 200V	399-4261-ND
C3	10 μ F, 25V	P5148-ND
C5	0.047 μ F, 50V	399-4246-ND
C6	220 μ F, 25V	P5153-ND
C8	100 μ F, 35V	P5165-ND
C4,9	0.1 μ F, 50V	P4525-ND
C11	10-180 pF trimmer	SG3019-ND
C11a (optional)	2-15 pF fine-tune trimmer	SG3025-ND
C12,13	10 pF, 200V	399-4140-ND
R1	10K pot	3386F-103LF-ND
R2,9	1K, 1/4W	1.0KQBK-ND
R3	180 ohm, 1/4W	180QBK-ND
R4	10 ohm, 1/4W	10QBK-ND
R5	51 ohm, 1/4W	51QBK-ND
R6	100K, 1/4W	100KQBK-ND
R7	330K, 1/4W	330KQBK-ND
R8	100K pot	3386F-104LF-ND
R10	470K, 1/4W	470KQBK-ND
D1	1N4004 diode	1N4004FSCT-ND
D2	1N5818 Schottky diode	1N5818FSCT-ND
D3	1N914 diode	1N914BCT-ND
D4	1N5239, 9.1V zener	1N5239BFSC-ND
J1	Two pos, 0.1", housing	609-1262-ND
PIN	(2) 22-30 AWG female pins	609-4453-1-ND
J2	3.5 mm mono audio jack	CP-3536N-ND
J3	2.1 mm x 5.5 mm power jack	CP-102A-ND
L1	Osc coil, P-C70-OSC	Antique Electronic Supply
T1	Output xfmr, 1K to 8 ohms	Mouser, 42TM013-RC
T1 alternate	Output xfmr, 1K to 8 ohms	RadioShack, 273-1380
ANT1	Telescoping antenna	Mouser, 43AR104
NUT	3 mm nut for antenna	Hobby People, others
SW1	SPDT toggle	EG2447-ND
SW2,3	SPDT slide	450-1578-ND
LED1	LED bar, red	516-1243-5-ND
SPKR	20 mm dia, 8 ohm	102-2502-ND
V1	12K5 vacuum tube	Various Internet vendors
Q1	2N7000 FET	2N7000FS-ND
U1	LM386N-4 pwr amp, 5V-18V	LM386N-4/NOPB-ND
SOC1	Seven-pin tube socket, P-ST7-201MXB	Antique Electronic Supply
SOC2	Four-pin DIP socket	ED3308-ND
TOOL1	Var cap and pot adj tool, plastic	SPTOOL-ND
TOOL2	Osc adj tool set, S-T9304 (\$0.95)	Antique Electronic Supply
SPACER	Two 3/16 x 3/4 x 4-40 spacers	1895K-ND
PWRSUP	12 VDC @ 1.5A plug-in AC adapter	Mouser 826-DA18-120US-M
PCB1	Arduino Uno	Various vendors
PCB2	VoiceShield, SPL-005011	SpikenzieLabs
PCB3	Arduino proto shield	Various vendors
DOME	4.5" dia x 8" high dome and base	Collecting Warehouse

which mates with J1. The SpikenzieLabs website (www.spikenzielabs.com) has a tutorial on how to download and use the software for the Arduino and VoiceShield. I have embellished on the tutorial and my write-up is included at the article link. Using the VoiceShield really isn't all that complicated, but there is a learning curve.

After I got the VoiceShield working, I made up an mp3 sample of the talking clock and other broadcasts, as mentioned at the beginning of this article. For the sample, the broadcaster was located in the house and an AM radio was in the garage. It picked up the transmitted signal on 1,480 MHz; the headphone output was stored in an mp3 file, which is what you will hear on the link.

Final Thoughts

If you are not into vacuum tubes, Ramsey Electronics (www.ramseykits.com) has a transistorized AM transmitter kit that you can build. It uses six 2N3904 transistors to generate the modulated RF signal.

As for the future, the broadcaster board could be

housed in a nice metal case for shielding, and the controls could be changed to ones that have knobs. Perhaps the tube could even be replaced with a transistorized circuit so it doesn't draw so much power. I would sure hate to see the tube go, however, since it's so quintessentially "Retro." **NV**

HOW TO PROGRAM THE VOICESHIELD

VoiceShield Basics

The ISD4003 voice chip on the SpikenzieLabs VoiceShield holds 240 seconds (four min) of sound, sampled at 8 kHz. The sounds enter the chip through the "audio in" jack as analog audio. Audio files cannot be directly downloaded to the chip. They must be "played" into the jack. The chip then samples and stores them.

Typically, the user specifies 80 three second slots, numbered 0 to 79. The Arduino can call the slots in any order. If a sound is longer than three seconds, it will extend into the next slot; therefore, the very next recorded sound needs to start at least two slots down (or whatever). You have to keep track of the length of your sounds so the correct slot(s) gets recorded and played back without any overlap. It's tricky, but it works if you are patient.

If you have problems, there are two areas on the SpikenzieLabs website that may answer your questions:

1. VoiceShield, Software: Page 4 — Voice Shield FAQ and Troubleshooting
2. Forum, VoiceShield

Installation

1. The following installation was for a Windows XP machine. Other platforms are also supported by the vendor.
2. Download the Arduino IDE if not already loaded.
3. Go to the SpikenzieLabs/VoiceShield website and click on Software.
4. Download "Arduino VoiceShield Library" into the Arduino/libraries folder.
5. Create a VoiceShield folder and download "VSProgrammerLite."
6. Locate the already downloaded SoundScore.txt file and download "SoundBytes" into the same folder.
7. Downloading "VoiceShield Programmer" is optional; the Lite version is fine.

Hookup

1. Connect a USB cable from your computer to the Arduino Uno.
2. Locate a 3.5 mm audio splitter so you can listen to the audio going into the VoiceShield.
3. Connect the input of the splitter to your computer's headphone output.
4. Connect headphones to one output of the splitter.
5. Connect an audio cable from the other splitter output to the VoiceShield Audio In jack.
6. Set your computer's audio output volume at a normal listening level.

Record 80 Prerecorded Words and Do a Playback

1. Open the Arduino IDE and load VS_Loader (in the Arduino/examples folder) into the Arduino Uno.
2. Run VS_Loader, then close the Arduino (because the port needs to be released).
3. Open VSProgrammerLite and select the port.
4. Press Program. Wait for 80 words to load. You should hear the words in the headphones as they are loading. If not, recheck the wiring.
5. Close VSProgrammerLite.
6. Move the headphone plug over to the VoiceShield Audio Out jack.
7. Open the Arduino again, load File/Examples/VoiceShield/BasicPhraseTalk, and run it.
8. You should hear a number of phrases being played from the VoiceShield memory, such as "How much is the train?"

Make Up Your Own Sentence and Play It Back

Load and run this simple sketch into the Arduino:

```
//Title: Play_VS_Slots
//
#include <VoiceShield.h>
VoiceShield vs(80); // create 80 slots
void setup()
{
}
void loop()
{
  vs.ISDPLAY_to_EOM(56); // "Do" (play slot 56)
  vs.ISDPLAY_to_EOM(68); // "You" (play slot 68)
  vs.ISDPLAY_to_EOM(44); // "Want" (play slot 44)
  vs.ISDPLAY_to_EOM(42); // "The" (play slot 42)
  vs.ISDPLAY_to_EOM(50); // "Hot Dog" (play slot 50)
  delay(2000); // delay 2 secs, then loop again
}
```

Now, you can write a simple program to say any of the 80 words listed in Table 1.

Audacity

I love Audacity. It's a free piece of audio editing software that lets you see exactly what the audio waveform is doing. It has a very intuitive learning curve. It is simple to control the timing of clips by inserting "silence" or cutting out portions. You can amplify or reduce certain sections in order to match overall levels. In fact, I haven't even scratched the surface regarding its capabilities, but I did have a lot of fun.

Recording Your Own Sounds, Music, or Voice

I had a number of problems putting in my own audio, but here's what I found that worked.

1. Load your audio clips into Audacity — whether they

- are from a microphone, music player, or even Arduino-triggered words from the VoiceShield. Edit each clip to the desired length. If the clip is over three seconds long, you may have to add a few seconds of extra silence on the end which may or may not get cut off when VSProgrammerLite plays it into the VoiceShield.
- After editing, save each clip as a wav or mp3 file with a unique name. Note any names that have capital letters. Also, note the running time of each clip in seconds, and calculate how many three second slots will be required.
 - Now, put copies of all your wav or mp3 files into the VoiceShield/.../SoundBytes folder. The folder should already have a bunch of AIF files in it.
 - Open the VoiceShield/.../SoundScore.txt file in Notepad.
 - In the SoundScore.txt file, type in the desired slot numbers and file names that you want to record. Start with slot 1, not 0. Make sure capital letters are observed. For example:
1 (tab) FileName1.wav (return)
6 (tab) FileName2.mp3 (return)
15 (tab) FileName3.wav (return)
Finally, save the SoundScore.txt file back under the same name.
 - Open the Arduino IDE and load VS_Loader (in Arduino/examples/VoiceShield folder) into the Arduino Uno.
 - Run VS_Loader, then close the Arduino (because the port needs to be released).
 - Open VSProgrammerLite and select the port.
 - Press Program. Wait for your files to play (record). You should hear the words in the headphones as they are recording. If not, recheck the wiring.
 - After your files have "played," try pushing the stop button in the Lite window. If nothing happens, then just close VSProgrammerLite.
 - Now, move the headphone plug over to the VoiceShield Audio Out jack.
 - Load and run this simple sketch into the Arduino:

```
//Title: Play_VS_Slots
//
#include <VoiceShield.h>
VoiceShield vs(80);
    // create 80 sound slots
//
// the setup routine runs once
void setup()
{
}
// the loop routine runs over and
// over again forever:
void loop()
{
vs.ISDPLAY_to_EOM(1);
    // (play slot 1)
vs.ISDPLAY_to_EOM(6);
    // (play slot 6)
vs.ISDPLAY_to_EOM(15);
    // (play slot 15)
delay(2000); // delay 2 secs then
    // loop again
}
```

- If the playback sounds distorted, you'll need to experimentally determine the optimum audio level from the headphone output going to the VoiceShield. I did multiple records and playbacks of the same clip to find the best level, and then noted the headphone volume control settings for all further recordings. If you need to do this, go back to step 5, turn down the volume a bit, and repeat steps 6-11.
- Hopefully, all these steps were successful and VoiceShield played back your files.
- Now, it's time to connect the VoiceShield to the broadcaster.
- Note: The VoiceShield has a small volume pot which will be used to set the modulation level of the RF output. Set it at ~20% initially.
- Connect J1 of the Retro-Shield to the Speaker Out of the VoiceShield and stack the Retro-Shield on top. Fire up your AM radio.
- Set SW2 to "J1;" SW3 to "Line."
- Connect the wall wart to J3 and turn on SW1. The audio should immediately start playing through the tiny speaker.
- When the tube warms up and the On Air LED comes on, your recording should come blasting out of the radio. If not, tune the radio until you find the signal.
- If the broadcasted message sounds distorted, turn down the volume on the VoiceShield a bit.
- You can choose between a VoiceShield recording, an mp3 music player, or a live microphone to broadcast a message to your waiting fans.

The List of 80 Prerecorded Words

I located the files for the prerecorded 80 words on my computer at VoiceShield/application.windows/SoundBytes/Zero.aif, One.aif, Two.aif ...

Slot	Word	Slot	Word	Slot	Word	Slot	Word
0	Zero.aif	20	Twenty	40	Store	60	North
1	One.aif	21	Thirty	41	Take Me	61	Oh
2	Two	22	Forty	42	The	62	Volts
3	Three	23	Fifty	43	Train	63	Right
4	Four	24	Sixty	44	Want	64	Point
5	Five	25	Seventy	45	Water	65	Next
6	Six	26	Eighty	46	What	66	Car
7	Seven	27	Ninety	47	When	67	Front
8	Eight	28	Hundred	48	Where	68	You
9	Nine	29	Thousand	49	Who	69	East
10	Ten	30	How	50	Hot Dog	70	Left
11	Eleven	31	How Much	51	Hello	71	Stop
12	Twelve	32	South	52	Have	72	Turn
13	Thirteen	33	I	53	Hamburger	73	Celsius
14	Fourteen	34	Is	54	Goodbye	74	Degrees
15	Fifteen	35	It	55	Does	75	Readings
16	Sixteen	36	Now	56	Do	76	Back
17	Seventeen	37	O'Clock	57	Could	77	Your
18	Eighteen	38	Please	58	West	78	We
19	Nineteen	39	Some	59	Can	79	Kee

Table 1.

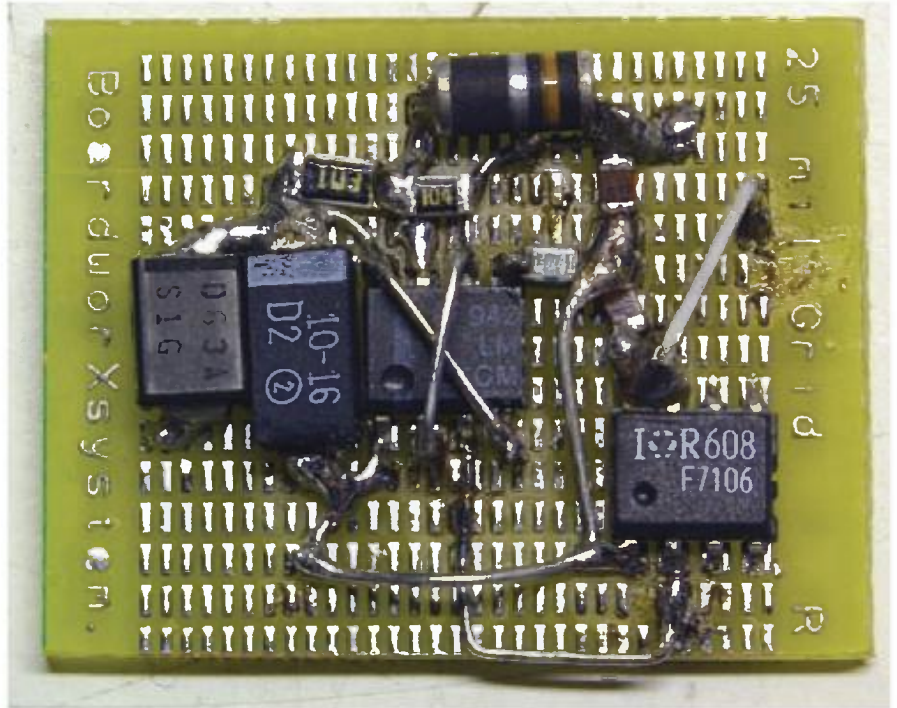
HACKING THE NINE-LED FLASHLIGHT

By Dan Gravatt



By now, just about everyone has seen or used the ubiquitous nine-LED flashlight. They are cheap, fairly bright, and run on three AAA batteries. However, this dirt-simple circuit eats through batteries pretty quickly and is hard on the LEDs — I've had an LED fail in mine already. The hack I'll discuss here adapts the flashlight to use an external power source, and adds a pulsed driver circuit for better efficiency and LED life without compromising brightness. The hacked flashlight can be repurposed as a task light, bicycle headlight, helmet lamp, or accent light.

■ **PHOTO 1.**
Driver circuit on
the SMT
breadboard.



In order to fit the pulsed driver circuit into the battery holder of the flashlight, surface-mount parts are a must. I used a SMT-400 surface-mount "breadboard" made by BoardworX, trimmed to slide into the three AAA battery holder (Photos 1 and 2).

The board is pretty easy to use — place each part where you want it, hold it in place with a toothpick, and tack the leads to the tinned bumps on the board with a fine-tipped soldering iron.

The LED connections for the circuit tap into the positive and negative terminals of the battery holder, so installing the driver circuit into the flashlight is simple.

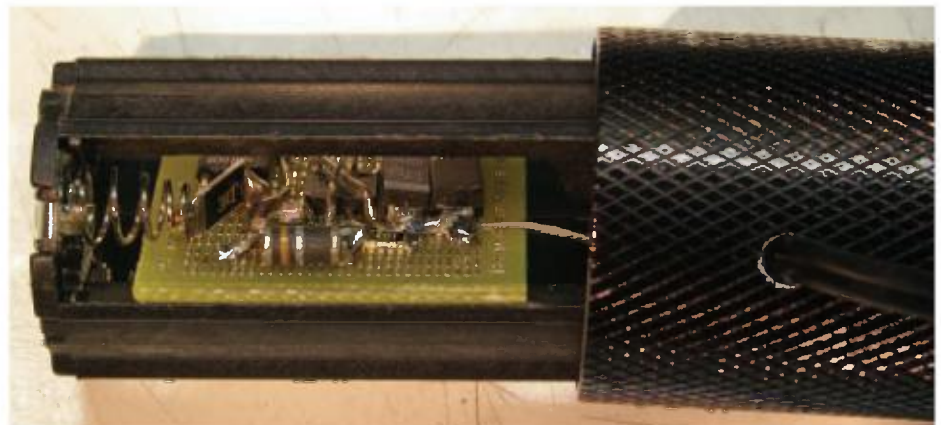
The wires to the external power source pass through a small hole in the side of the flashlight which can be sealed (if needed) to preserve the flashlight's water resistance.

The stock on-off switch in the end of the flashlight still works to turn the LEDs on and off, but does not cut power to the pulsed driver circuit. The circuit draws about 4 mA with the LEDs turned off when powered by a two-cell lithium-ion battery pack (7.4V nominal).

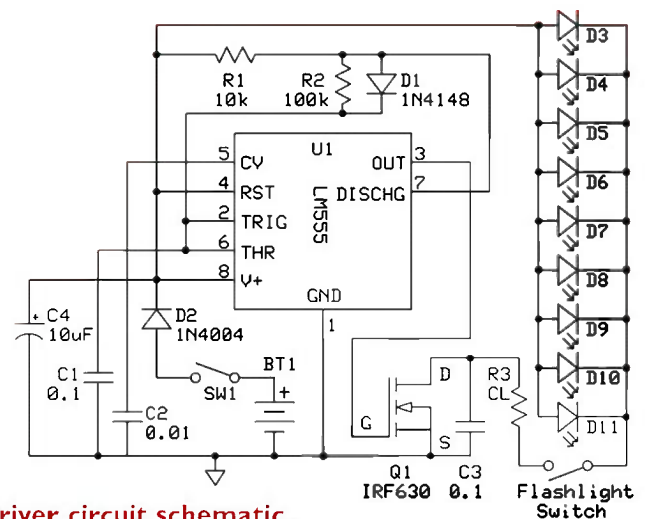
The pulsed driver circuit (Figure 1) is based on the venerable LM555 timer wired as an astable multivibrator and drives an N-channel power MOSFET transistor to switch the LEDs on and off.

Don't substitute a CMOS 555 in this circuit, as it doesn't have the "oomph" to effectively drive the MOSFET (see the article *Power MOSFETs, Part 1: Theory*, in the January 2009 *Nuts & Volts* for an explanation).

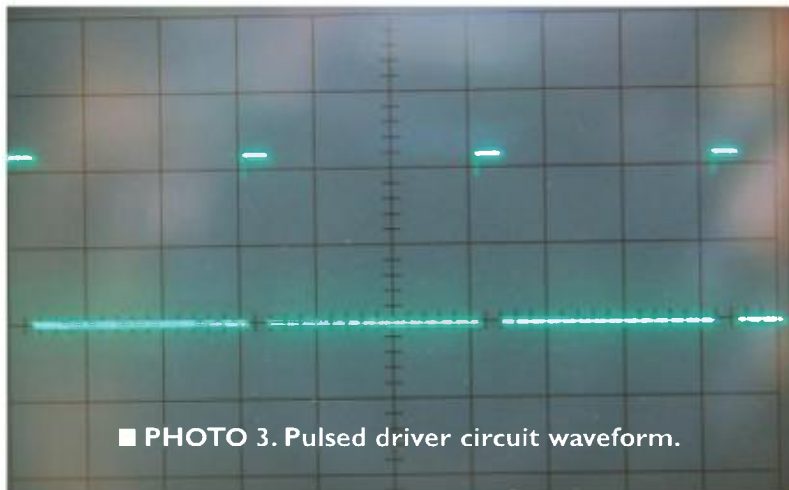
Diode D1 allows the circuit to have a duty cycle below 50% for better power efficiency; with the component values shown, the duty cycle is about 10% (Photo 3). Diode D2 is for reverse voltage protection. Capacitor C3 smooths voltage spikes across the LEDs due to the MOSFET switching and may further increase their lifespan. Switch SW1 is an external power switch (if desired), not the flashlight's stock on-off switch.



■ **PHOTO 2.** SMT breadboard installed in the flashlight's battery holder.



■ **FIGURE 1.** Pulsed driver circuit schematic.



■ PHOTO 3. Pulsed driver circuit waveform.

PARTS LIST

ITEM

LM555 timer
 IRF630 N-channel power MOSFET or equivalent
 1N4004 diode or equivalent
 1N4148 diode or equivalent
 10K ohm resistor
 100K ohm resistor
 Current limiting resistor (see text)
 10 μ F capacitor
 (2) 0.1 μ F capacitors
 0.01 μ F capacitor
 Two-cell lithium-ion battery pack
 On-off switch



■ PHOTO 4. The "business end" of a flashlight adapted as a task light.



■ PHOTO 5. Through-hole circuitry inside the task light.

Select a value for current limiting resistor R3 that meets your needs for battery life and brightness at whatever battery voltage you use. Some models of the flashlight may already have a small current-limiting resistor installed.

Technically, the circuit can be run at any voltage within the specifications of the LM555 and the MOSFET, but voltages above 12 volts or so might be detrimental to the LEDs in the long run, even with the pulsed drive and current-limiting resistor. Voltages below about 7.4 volts won't be sufficient to fully turn on most MOSFETs, leading to reduced LED brightness.

Observant readers will already have noticed that the component types in **Photos 1** and **2** do not match the schematic. The schematic components are all through-hole devices, in case you want to build it that way (**Photos 4** and **5**).

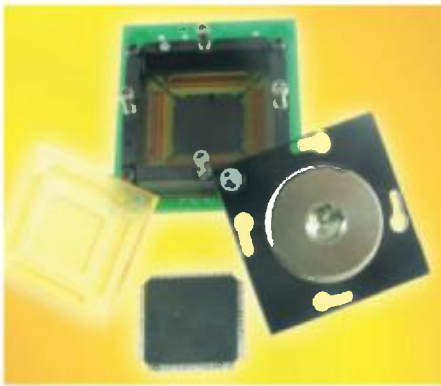
In the finest traditions of circuit hacking, I used all scavenged parts for both versions of this circuit, so I have no idea what their catalog part numbers would be. SMT resistors are coded with the first two digits of the value and a multiplier; "103" means 10×103 or 10,000 ohms. SMT ceramic capacitors often have a letter and a number such as A1 or E3 that indicates their value; go to <http://www.gfaelectronics.com/electronics/smtcapacitormarks.htm> for an explanation. A5 indicates a 0.1 μ F capacitor.

When selecting a MOSFET for your circuit (scavenged or new) make sure it can handle the voltage and peak current you expect to deliver to the LEDs.

However you decide to use your flashlight, happy hacking! **NV**

Dan Gravatt can be reached at dgravatt@juno.com.

Continued from page 27



insertion loss. The contact resistance is typically 20 milliohms per I/O.

The socket connects all pins with 10 GHz bandwidth on all connections. The socket is mounted using supplied hardware on the target PCB (printed circuit board) with no soldering, and uses one of the smallest footprints in the industry.

This footprint allows inductors, resistors, and decoupling capacitors to be placed very close to the device for impedance tuning.

The socket also incorporates a swivel lid with compression screw so that ICs can be changed out quickly.

The socket features a floating compression plate to force down the QFP leads on to elastomer. There is also a hard stop feature built into the compression mechanism.

The SG-QFE-7011 sockets are constructed with high performance and low inductance gold plated embedded wire on elastomer as the interconnect material between the device and PCB.

The temperature range is -35°C to +100°C. The pin self-inductance is 0.15 nH and mutual inductance is 0.025 nH. Capacitance to ground is 0.01 pF. Current capacity is two amps per pin.

Pricing for the SG-QFE-7011 is \$474, with reduced pricing available depending on the quantity required.

For more information, contact:
Ironwood Electronics
 Web:
www.ironwoodelectronics.com

HANDHELD RF SPECTRUM ANALYZER

Saelig Company, Inc., has introduced PSA Series II RF spectrum analyzers. Available in 1.3 GHz and 2.7 GHz versions (priced at \$1,395 and \$1,695, respectively), these new instruments are smaller, lighter, and have longer battery life than other more expensive handheld RF products.

PSA analyzers incorporate a 4.3" (11 cm) backlit TFT color touch-screen display, with a high capacity rechargeable Li-Ion battery to give more than eight hours operation per charge. The PSA1302 has a frequency range of 1 MHz to 1,300 MHz, while the PSA2702 operates up to 2,700 MHz. Dynamic range is 80 dB with a noise floor at -100 dBm. Resolution bandwidth is selectable



down to 15 kHz.

The PSA Series II's advanced features include sweep modes (continuous, single, peak hold, and sweep average), AM/FM audio demodulation with built-in speaker,

Wi-Fi Module & Dev Kit

Industry's First

Embedded HTML5 WebSocket Server

Control hardware with a web browser on iPhone, iPad, Android, PC or Mac

Module

- WebSocket server enables real-time bidirectional socket communication between module and browser
- Embedded web server
- 8MB USB flash drive for web files
- Industry's leading low power Wi-Fi SoC
- 2 DACs, 8 ADCs and 14 Digital I/Os
- Program 100% in JavaScript
- Small Size: 2.5"x1.6". Low volume-pricing



Development Kit

\$129



- LEDs, buttons and pots for digital and analog I/Os
- Temperature and light sensors
- Battery, antenna & speaker
- Javascript source code to demo all hardware features

Compatible with current browsers:



Bluetooth Keyboard Adapter

- Turns any USB/PS2 keyboard into a Bluetooth keyboard
- Works with any phone, tablet or PC that supports HID Bluetooth keyboard
- High-capacity Rechargeable battery



\$59

Battery Capacity Tester



\$99

- PC software controls and records discharge of battery, plots curve and calculates total mAh and energy
- USB based. Green PC software (no installation necessary. Just download and run)

Telephone Caller ID Manager

- PC software records caller ID with notes typed in by user
- When a call comes in, pops up window with call history and notes previously recorded
- USB based. Green PC software



\$99



handheldsci.com
support@handheldsci.com

Santa Clara
 California

408-694-3782
 Distributors Welcome

Continued on page 77

ALL ELECTRONICS

C O R P O R A T I O N

QUALITY Parts
FAST Shipping
DISCOUNT Pricing

CALL, WRITE, FAX or E-MAIL
for a FREE 96 page catalog.
Outside the U.S.A. send \$3.00 postage.

HOME CAMERA

Color, wide angle, CCD board camera. 5.5" diameter black plexiglass dome. Operates on 12Vdc (2.1mm, center + coax jack). BNC video connector. Power supply not included. Recommend our supply, CAT# PS-1251.

Note: Originally designed as a computer accessible, network camera, there are cables and circuitry for LAN and digital input/output as well as a CD with a downloadable instruction manual and software for camera's original purpose. We have been unable to utilize the camera's extra features. **We are selling this as a new, functioning video camera. We do not guarantee features enhanced by software or extra circuitry.**

CAT# VC-210

4 for \$25.00 each

\$27⁹⁵
each

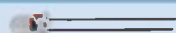


ULTRAVIOLET LED

390-400nm UV LED. 5mm (T1-3/4). 20° viewing angle. Water-clear in off-state. 3.4-3.6V. 20mA. Ideal for counterfeit bill detection, detection of fluorescence in minerals, black light posters. CAT# ULED-3

\$110
each

100 for 90¢ each
1000 for 75¢ each



21MM PIEZO ELEMENT

50mm wire leads.

CAT# PE-56

\$100
each

10 for 85¢ each



33,000UF 50V COMPUTER GRADE CAPACITOR

Nippon Chemi-con U32D 105C. 35mm dia. x 123mm high (excluding terminals). Screw terminals.

CAT# EC-335

10 for \$4.50 each

\$4⁷⁵
each



24VDC 2.8A SWITCHING POWER SUPPLY

Sino-American # SA165E-24U. Input: 100-240V 50-60Hz 1.7A. Output: 24Vdc 2.8A 67.2W.

Table-top, switching supply. 6' output cord with 2.5mm coax power plug, center positive. Detachable 2-conductor power cord. cULus, TUV, CE.

CAT# PS-2428

10 for \$14.00 each

\$14⁷⁵
each



PHOTOELECTRIC SWITCH FOR OUTDOOR LIGHTING

Area Lighting Research Model SPT-15. Photocell control to turn lights on at dusk/ off at dawn.

Raintight for outdoor use. For 120Vac operation. 16.6A 2000W Tungsten. 15.8A 1900VA Ballast. 6" pigtail leads.

3" x 2" x 1.5" sealed plastic box mounted on adjustable swivel with threaded bushing and nut for standard electrical knock-out box. UL, CSA.

CAT# PCL-4

\$9⁰⁰
each



LINEAR ACTUATOR FOR AUTO FUEL DOOR

12Vdc push/pull actuator for auto fuel door. 12mm push or pull

depending on polarity. Nylon housing and plunger. Housing dimensions 43 x 59 x 29mm. Two mounting holes on 50mm centers. Terminals are inside of nylon connector attached to housing. 5" nylon hooked extension rod can be easily twisted off end of plunger.

CAT# DLA-3

\$4⁷⁵
each



20 CHARACTER X 4 LINE LCD

Tianma # TM204ADB6. Module size: 99 x 60 x 10mm. Display area: 25 x 76mm. Gray mode, STN Positive, Transflective LCD module.

CAT# LCD-2040

\$6⁷⁵
each



HIGH-TORQUE SERVO

Z590M racing servo. Metal gears. 85 oz-in torque @ 6Vdc. Speed: 0.18 sec/60° @ 4.8V, 0.15 sec/60° @ 6V. Standard body size: 1.50" x 0.74" x 1.37" high. 7" leads w/ 3-pin JR connector.

CAT# DCS-107

10 for \$10.65 each

\$10⁹⁵
each



3-28VDC PIEZO BEEPER

Continuous tone. 103 db sound pressure. 1.14" diameter x 0.93" ABS plastic housing. 5" wire leads.

CAT# SBZ-100

\$3⁷⁵
each



MINI-MOTOR, VIBRATOR

Tiny motor with offset weighted shaft. Designed to vibrate in mobile communications devices. Operate on 1.5-4.5Vdc. 15mm - length including shaft. 4mm diameter without rubber.

CAT# DCM-371

\$1²⁵
each

10 for \$1.10 each
100 for 95¢ each



12 VDC XENON FLASHER

Bright strobe assembly with horseshoe flashtube. Approximately 80 flashes per minute. 12 Vdc operation. 65 x 75mm

CAT# FSH-13

\$5⁵⁰
each



Shop ON-LINE www.allelectronics.com
ORDER TOLL FREE 1-800-826-5432

MAIL ORDERS TO:

ALL ELECTRONICS CORP.

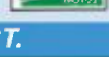
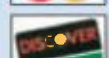
14928 OXNARD ST., VAN NUYS, CA 91411-2610

FAX (818) 781-2653 • INFO (818) 904-0524

E-MAIL allcorp@allcorp.com

NO MINIMUM ORDER • All Orders Can Be Charged to Visa, Mastercard, American Express or Discover • Checks and Money Orders Accepted by Mail • Orders Delivered in the State of California must include California State Sales Tax • NO C.O.D. • Shipping and Handling \$7.00 for the 48 Continental United States - ALL OTHERS including Alaska, Hawaii, P.R. and Canada Must Pay Full Shipping • Quantities Limited • Prices Subject to change without notice.

MANUFACTURERS - We Purchase EXCESS INVENTORIES... Call, Write, E-MAIL or Fax YOUR LIST.



How to Make Professional Looking Control Panels and Enclosures

Part 1

By Gordon McComb

www.nutsvolts.com/index.php?/magazine/article/april2013_McComb

Discuss this article in the *Nuts & Volts* forums at <http://forum.nutsvolts.com>.

Don't you just hate it when you spend days — maybe even weeks — on a great new electronics project, only to have it turn out looking like a six year old cobbled it together in half an hour?

What's on the outside of your projects is almost as important as what's inside. Control panels and enclosures are the first thing anyone sees of your build, so make sure it looks tip-top and professional.

Sound difficult or expensive? It's neither!

In fact, it's easy to make good looking control panels and enclosures for your projects. All you need is a computer, graphics program, and a laser or inkjet printer. It's inexpensive, too. Cost of materials is just a dollar or two per panel.

In this two-part article, I'll show you simple techniques for designing, printing, and finishing functional and aesthetic control panels of various shapes, sizes, and layouts. Not everyone can utilize professional computer designing tools, so I'll demonstrate all the ideas using the open source — and free! — Inkscape vector graphics program. It's available for Windows, Macintosh OSX, and Linux. Of course, don't forget about great services from companies like Front Panel, who can do this for you if you don't have the time.

Overview of the Process

First, a quick definition: A control panel is anything that provides a human interface to the electronic guts of your project. The panel can be comprised of a simple sheet of plastic, metal, wood, or even foam board. Switches, dials, indicator lights, and other control circuitry that provide interactive feedback go on the panel.

Often, but not always, the active circuitry and wiring of your project are directly attached to the back side of the control panel. In some cases, the panel serves as a kind of wired remote control. You hold the panel in your hands. Wires — or even wireless signals — connect the panel with the rest of your project.

For the sake of simplicity, I'm lumping all the various ways you can bundle your project into a package and

calling the front of it a "control panel." For this article, it doesn't matter whether the panel is just by itself or part of a box or other enclosure, or even if it's attached to a 19" wide rack in your recording studio.

With that part out of the way, let's look at what's involved in making functional and attractive control panels. The basic steps are:

1. Decide what controls are needed for your project interface. Does it need just an on/off switch and a power light? Or, does it also require some additional function switches, control dials, LEDs, LCDs, bar graphs, or mechanical meter movement?

Laying Out the Design

To demonstrate how to use Inkscape to design a control panel, I'll show the process for a fictitious project that uses the three most common control components: toggle switch, LED, and potentiometer. The panel is for an *Intergalactic Interrogating Interocitor* — a device that surely has something to do with reading mind pulses, and perhaps erasing them and melting the entire brain.

The Interocitor panel measures 3" x 4", with holes at the corners for mounting directly to a printed circuit board. It just so happens that the 3" x 4" form factor is

common in many microcontroller project boards, including the Parallax BASIC Stamp Board of Education, so it's a good size to practice with.

Figure 1 shows the finished control panel, with holes drilled and ready for completion. **Figure 2** shows the completed layout created in Inkscape.

The following assumes you already have Inkscape installed on your computer, and that you are at least somewhat familiar with its overall features, menus, and on-screen interface. If not, take the time now to download and install it to your PC; see the **Sources** box for where to get Inkscape. Take a few hours following the various tutorials for Inkscape. You can access them in the Inkscape program by choosing Help>Tutorials.

Begin the process of creating the layout for the Interocitor control panel by opening a new document. For those in the US and other areas that use letter size paper, select File>New>Letter; for those in countries that use "A" size papers, choose File>New>A4. (The document size is only really relevant for printing. If you choose the wrong size initially, you can always change it later.)

Note! I'll be showing simple methods for using Inkscape to create physical layouts. I will not be showing how to use grids, snaps, guides, or other more advanced features. These you can learn on your own, and use whatever additional tools Inkscape offers to help speed up your work.

Adjust the zoom control to 75% or more to make the workspace a little larger in your monitor. Then:

1. Make sure that both rulers and the Tool

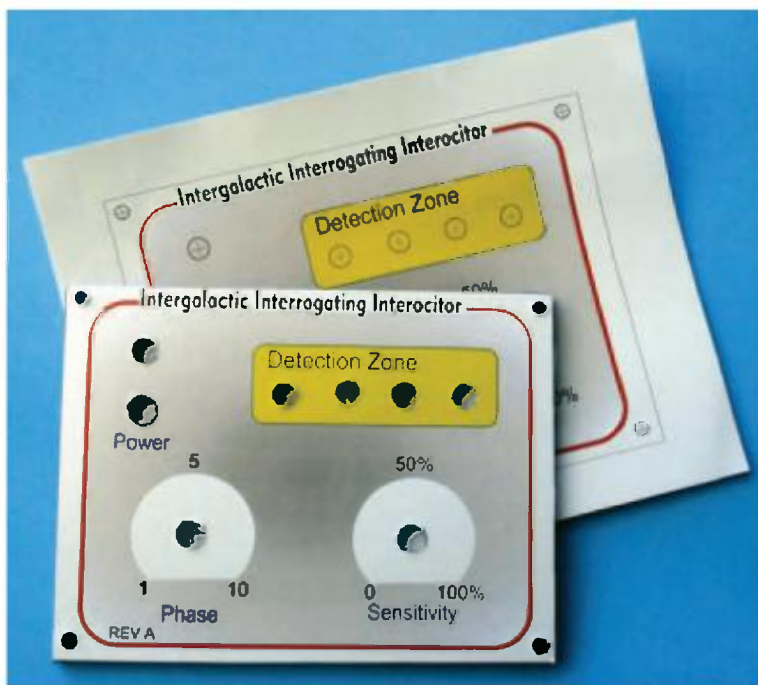


Figure 1. The finished Intergalactic Interrogating Interocitor panel, ready for mounting on switches, potentiometers, and LED indicators.

2. If you're placing everything into an enclosure, select a box that will hold not only the necessary internal parts of your project, but also provides ample space for all the control's components. If you're making a flat panel, pick a dimension for the panel so that it adequately supports all the components, including extra space for mounting your project boards, wiring harnesses, and other parts.
3. Use Inkscape (or other vector graphics tool) to lay out the design of the panel components. Draw the pieces to actual scale so that you can visualize how they'll appear when the panel is finished. You don't have to maintain engineering tolerances here. Programs like Inkscape provide simple scale and measurement tools that are more than

adequate for this job.

4. Print the panel onto paper, label stock, or other appropriate sheets. I'll get more into the printing aspect in the next article, but one of my favorite methods is to use polyester labels run through a color laser printer. After printing, you just cut the label sheet to size, peel off the backing, and apply to the control panel. (There are many other methods, and I'll cover those next time around.)
5. Using the graphic elements you provided in your layout, drill and cut to make holes, slots, and other shapes for the switches, LEDs, potentiometers, and other components.

With the panel completed, you can mount the control parts and finish your project.

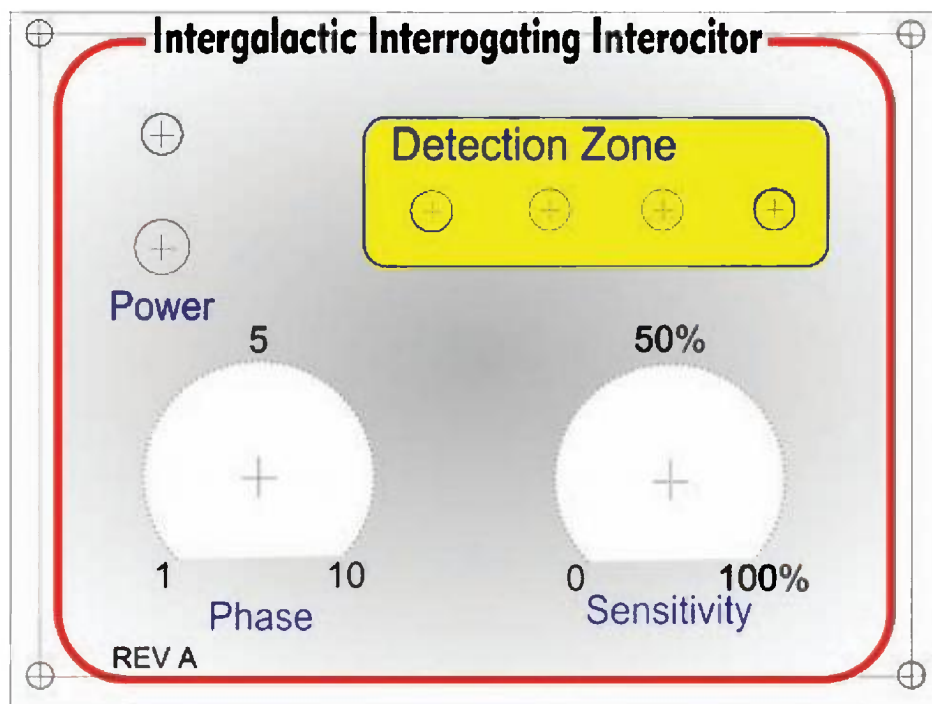


Figure 2. Layout template for the Interocitor panel, as produced by the Inkscape vector graphics program.

Controls bar are visible by selecting them in the View->Show/Hide menu. I like to turn on everything, just to save time in having access to all the main features from the main workspace.

2. Change the units of measure to inches in the Tool Controls bar. (You can use other units of measurement in your own project, but for this tutorial

I'm using inches, so it helps if your workspace is set up the same way.)

3. Select the Rectangle tool, and draw a rectangle three inches by four inches — as you create the shape, the current dimensions are shown in the status bar at the bottom of the screen. Don't worry right now if you can't get the dimensions exact.

4. Click on the rectangle to select

The Difference Between Vector and Bitmap Graphics

Take note that Inkscape is a vector graphics program. This means it deals with discrete shapes called objects that are constructed out of lines, circles, squares, and other vector shapes.

This is as opposed to a bitmap graphics program — such as Windows Paint or Adobe Photoshop — which creates images by manipulating individual pixels. A vector program makes it much easier to come back at a later time and make changes to the individual shapes (er, objects) that make up the drawing. While vector programs require a larger learning curve, they are very well suited for the work required to make control panels.

Sources

Inkscape
Open Source Vector Graphics Program
inkscape.org

Robotoid
Resources for control panel design
and construction
www.robotoid.com/panels

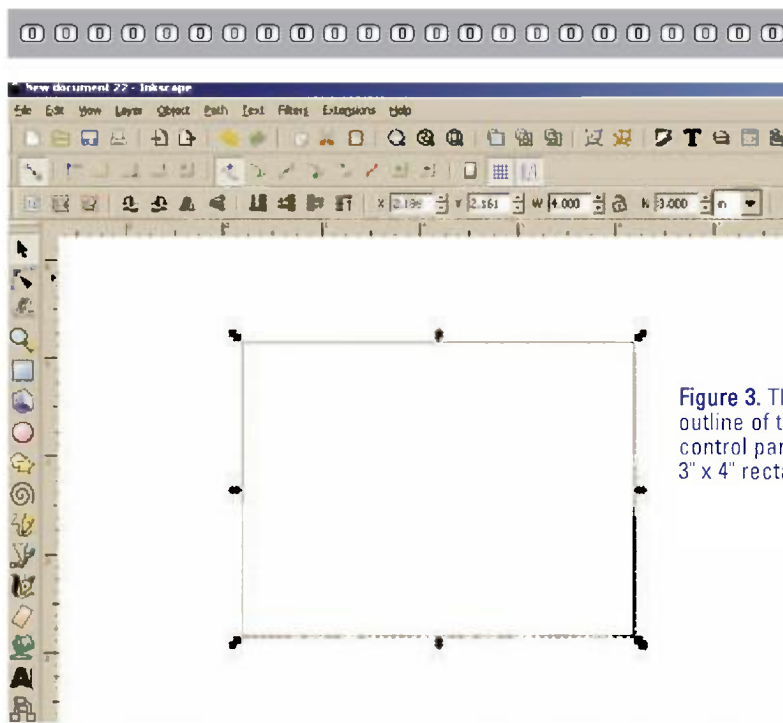


Figure 3. The basic outline of the control panel is a 3" x 4" rectangle.

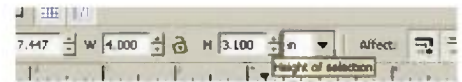


Figure 4. Use the Inkscape measurement tools to ensure accurate dimensions.

it, place it at the top middle of the document space, and choose Object>Fill and Stroke. Click the Fill tab and choose to remove any fill (refer to **Figure 3**). Click the Stroke paint tab and choose the color black. Finally, click the Stroke style tab and choose 0.5 pixels for the stroke thickness.

5. With the rectangle still selected, specify the exact width and height dimensions in the Tool Controls bar: 4.000" for the width and 3.000" for the height. Refer to **Figure 4**.

Note! Don't set the exact size of the rectangle until after you have altered the stroke width. Otherwise, the rectangle will not be sized correctly.

The rectangle you just created is the outside dimension of the control panel. Before printing, we'll actually make it a little larger, but for now we'll use the 3" x 4" dimensions as a reference for fitting on all of the control elements.

Next, create the mounting holes which, in our case,

are in the four corners of a 2-3/4" x 3-3/4" rectangle.

1. Create a new rectangle as in steps 1-4 above, this time measuring 2.750" x 3.750".
2. Select the 3" x 4" panel outline rectangle, then shift-select (hold down the Shift key) to also select the hole mounting rectangle.
3. Choose Object>Align and Distribute.
4. Click both the Center Objects Horizontally and Center Objects Vertically icons (refer to **Figure 5**).

Note: For these next steps, you may want to increase the zoom to at least 200% so you can make out all the fine detail.

5. Select the Ellipse shape tool and create a circle

Figure 5. The inner rectangle forms the footprint for the four mounting screws.

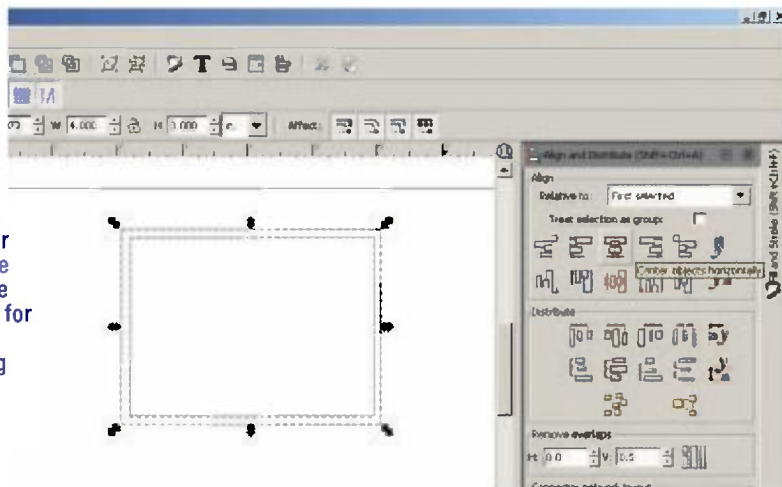
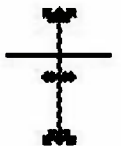


Figure 6. Make cross-hairs for accurate hole drilling by combining two lines, centering them, and then adjusting their size.



that's about .125" in diameter. Use the fill and stroke tools to remove the fill, and set the stroke to 0.5 pixels (px). Use the width and height boxes in the Tool Controls bar to resize the circle to 0.120" in diameter.

6. Click the Beziers curves tool, and while holding down the Control key to constrain the direction of movement, make a short line of about 1" in length. (Double-click at the end of the line to finish it.)

7. Make a copy of this line (File>Copy, then File>Paste; see **Figure 6**), and choose Object>Rotate 90° CW. Select both line segments, open the Align and Distribute panel, and align the lines both horizontally and vertically. Group these two lines by choosing Object>Group (or press the Ctrl+G keys).

8. Resize these line segments to 0.20", select both the lines and the circle made in step 5, and align the objects both horizontally and vertically. Group these two objects so it's easier to manipulate them.

9. Make three additional copies of the circle with cross-hairs, and manually position all four into the corners of the inner rectangle from step 1. The cross-hairs will help you find the center of the hole when drilling. You'll want to zoom in real close (see **Figure 7**) so that you can accurately position the cross-hairs for the mounting holes.

Laying Out Switches and LED Holes

So far, so good. It's now time to lay out the hole patterns for the switches and LED indicators used on the panel. These are made just like the four mounting holes, so you can use the same techniques. For the sake of visualization, I made the switch and LED holes for my control panels at about the size of the actual hole, though what really matters is the cross-hairs in the middle for setting the drill.

To make a hole for any component (switch, LED, etc.):

1. Begin by copying one of the four mounting holes made in the previous section. Ungroup the circle and cross-hairs objects.

2. Adjust the diameter of the hole to just under the actual dimension of the part. For example, for a switch with a 1/4" shaft, make the circle no larger than 0.240". For a 5 mm LED, the closest fractional inch size is 0.197", so make the holes for your LEDs no larger than about 0.18".

3. After resizing the circle, select both circle and cross-hairs, and re-align on the vertical and horizontal centers as shown in **Figure 8**. Group the objects together to make it easier to manipulate them. Make copies as needed.

The Intergalactic Interrogating Interocitor contains one power switch and indicator light. I've positioned them in the upper left corner and "eyeballed" their location. As shown in **Figure 9**, four additional indicator LEDs are placed within a box and are along the upper right of the

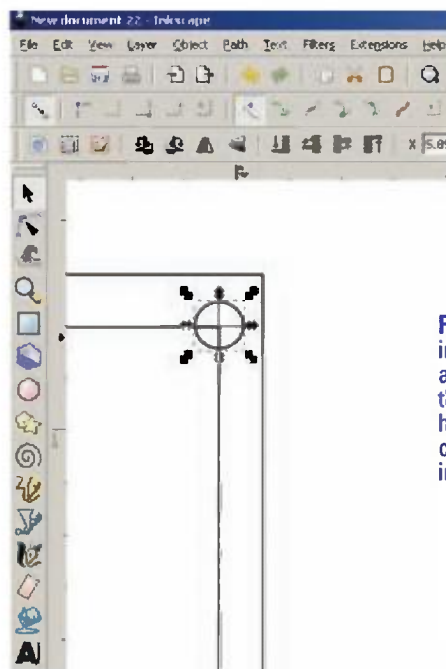


Figure 7. Zoom in closer to accurately place the mounting holes in the corners of the inner rectangle.

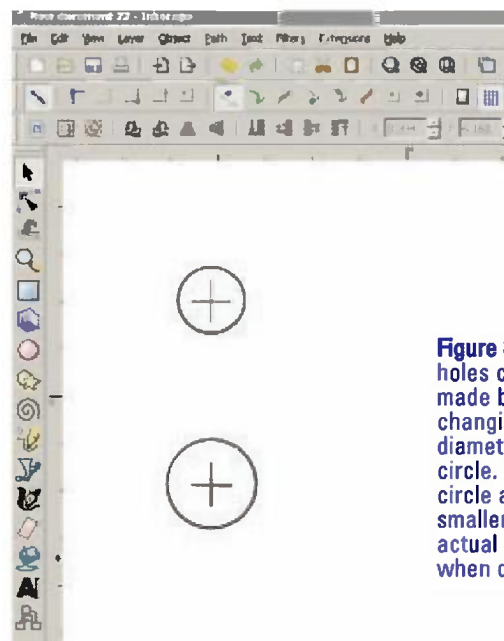


Figure 8. Other holes can be made by changing the diameter of the circle. Make the circle a little smaller than the actual hole when drilled.

pane. For your control panels, you can position the components wherever they look and fit best. Just make sure not to overcrowd the parts. The area taken up by the top (visible) portion of the component may be — and usually is — less than the area taken up by the underside portion. This is particularly true of things like toggle switches, as they can have very fat bodies even though the toggle handle takes up little space.

Be sure to leave enough space for wiring and other termination requirements, as well. If a component uses screw terminals, be sure there's ample room for

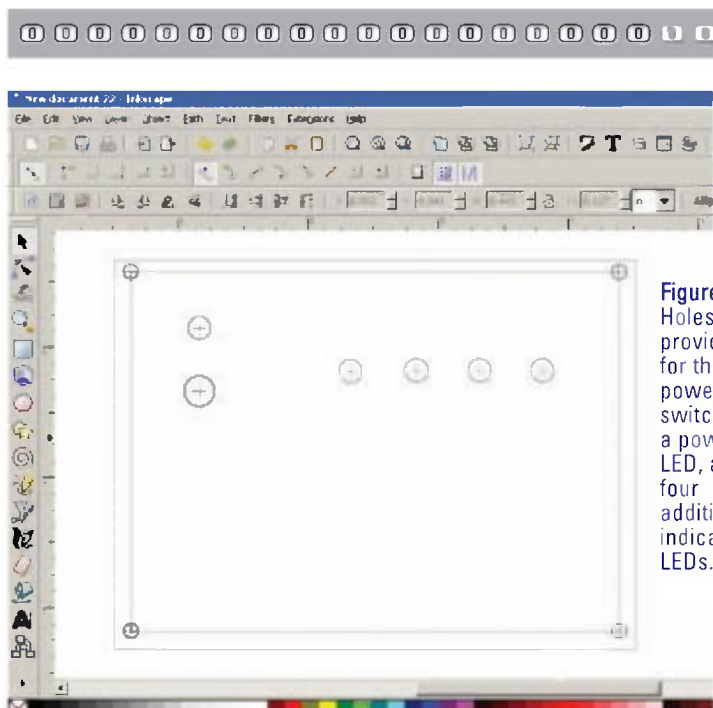


Figure 9. Holes provided for the power switch, a power-on LED, and four additional indicator LEDs.

the wires, as well as the screw driver to tighten and loosen the connection.

Laying Out Holes for Potentiometers

Many projects include one or more pots for making finely tuned adjustments or calibration. Potentiometers have both a mounting hole and an optional (but recommended) outer graduation. The Interocitor uses both.

1. Use a circle and cross-hairs like you made above, then ungroup them so you can work with them separately. Enlarge the circle to 1" in diameter – this will work with dials up to about 3/4" of an inch.

Note! When resizing objects, Inkscape may also scale

the stroke width. We want to turn this behavior off for this project, so locate the four Affect icons in the Tool Controls bar and unselect the one for scaling the stroke width.

2. Align the cross-hairs and circle both horizontally and vertically.

3. Select the circle and click on the Edit Paths tool. Drag the white handles to create an arc as shown in **Figure 10**. (If the circle is shown where all line segments are connected, keep the circle selected, then click on the Ellipse tool. In the Change bar that appears, click on the "Switch to arc (unclosed shape)" icon.

4. Show the Fill and Stroke panel, and select white as the fill color. Click on the Stroke style tab, specify a stroke width of 0.5 px, and set the style to one of the dotted lines (**Figure 11**). This is just for looks, so pick one you like best.

Note! A sometimes irritating "feature" of Inkscape is its penchant for remembering the last tool setting you used. This often comes in handy, but it can also be a source of frustration. Once you change the behavior of some tools – like the arc setting for the Eclipse tool – it stays until you change it again.

If you find all the new circles come out as arcs, simply revert back to the original setting by clicking on the Ellipse tool and click on the "Make the shape a whole ellipse" icon in the New bar that appears.

Adding Text

Most panels have explanatory text added here and there, even for control functionality that is obvious. Inkscape lets you add text anywhere, in any size and font you have installed on your system. For the Interocitor example, I've used standard Arial – one of the standard



Figure 10. Create a potentiometer dial face by creating a circle with the Ellipse tool, then dragging by the white handles to produce an arc.

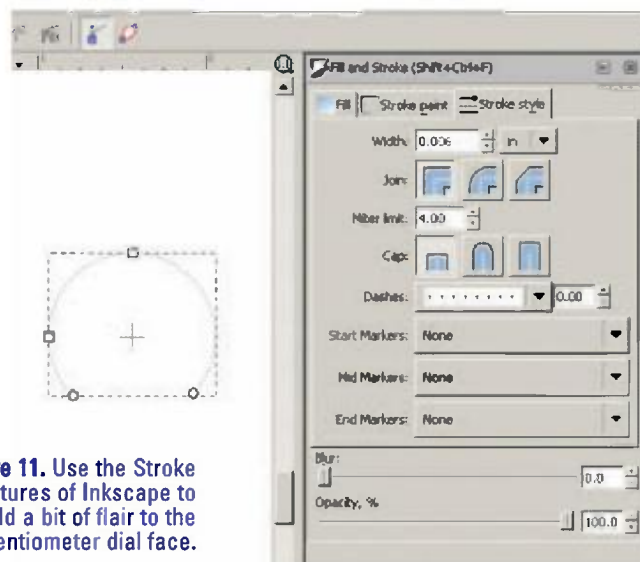
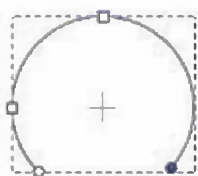


Figure 11. Use the Stroke style features of Inkscape to add a bit of flair to the potentiometer dial face.

fonts included in Windows. To add text:

1. Click on the Text tool, then click anywhere on the workspace.
2. Type the text you want, select it, and move it into position. You can change the color of the text by altering its Fill color. Do this either in the Fill and Stroke panel, or by clicking one of the colors in the color swatches that appear along the bottom of the Inkscape window.
3. Change the font and size by clicking on the Text icon in the Commands bar.
4. Find Arial and select it. Change the font size to 15 point (refer to **Figure 12**). Close the Text and Font box.

Repeat these steps for additional text. You can also copy an existing piece of text and just edit it as needed.

Aligning Elements

One of the benefits of using a graphics program for designing your control panels is that you can use automatic tools to align all the components. This greatly improves the look of the panels and you don't have to rely on guesswork.

To align two or more graphic objects (circle, text, etc.):

1. Select the first object you wish to serve as the "anchor." This object will not be adjusted; rather, all the other objects will be aligned around this one.
2. Open the Alignment panel (Object>Align and Distribute). Choose Relative to: First Selected.
3. Click on the Alignment icon(s) you wish to use. For example, to align the objects in the same vertical plane, click the Center Objects Horizontally icon as shown in **Figure 13**.

You can also use Inkscape to help you place multiple objects along a vertical or horizontal plane using the Distribute icons. Use this feature, for example, to make sure the four indicator icons are spaced exactly apart.

Adding Group Boxes

The larger and more complex your control panels, the more you'll want to group similar components together. This is for aesthetic reasons only, and doesn't affect the overall functionality of the panel.

Group boxes are made with the Rectangle tool. You can keep the standard sharp corners or round them off as I've done for the sample Interocitor panel. To round the corners:

1. Make a rectangle with the Rectangle tool.
2. Select it, then choose the Edit Paths tool.
3. Drag the white handle on the right side downward to increase the radius of the corner (**Figure 14**).

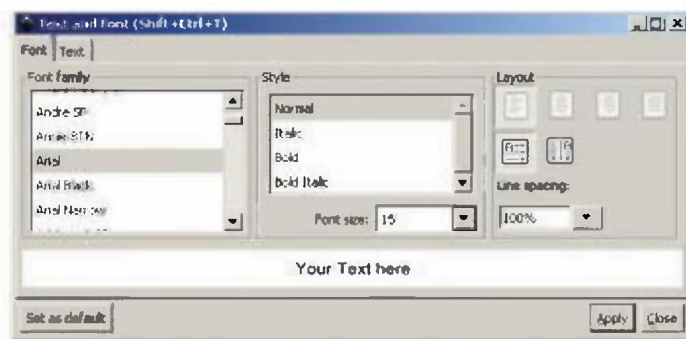


Figure 12. Add text with the Text tool. Type the text, then open the Text and Font dialog box to select the font and font size.

I've made two group boxes for the Interocitor panel: one to group together four LED indicators, and another larger one to enclose all the components of the panel.

Begin by creating the rectangle in the size and shape you want. It's okay to manually adjust the size, but be sure that for the large group box you don't accidentally cover up something you want — like a mounting hole. The large group box for the Interocitor panel is adjusted to fit neatly within the four corner mounting holes.

Change the fill and stroke color as needed. For the large main group, I've selected a gray gradient (more about this below) and a thick red stroke that's 3.5 pixels wide. For the four indicators, the group box is made to contain the four LEDs plus descriptive text. The fill color is yellow, the stroke is 0.75 px green. These were just subjective choices; you can choose anything you like.

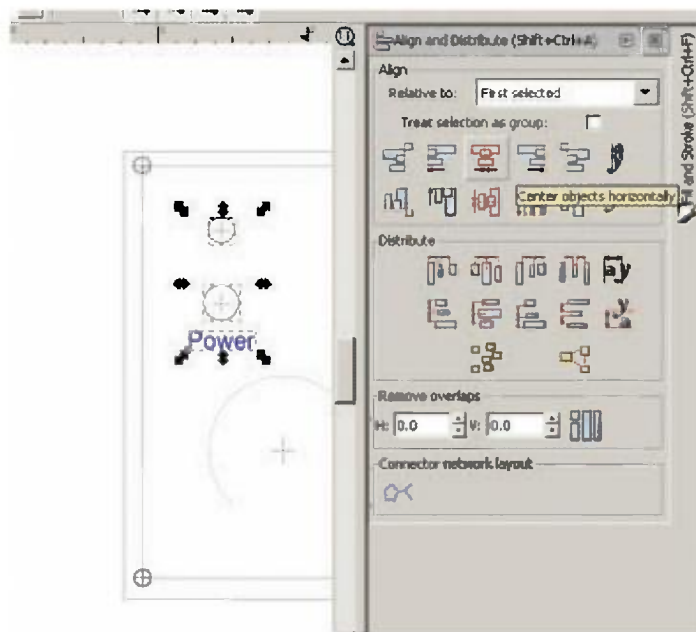


Figure 13. Align objects by selecting them (the first selected is the "anchor") and then choosing an option in the Align and Distribute panel.

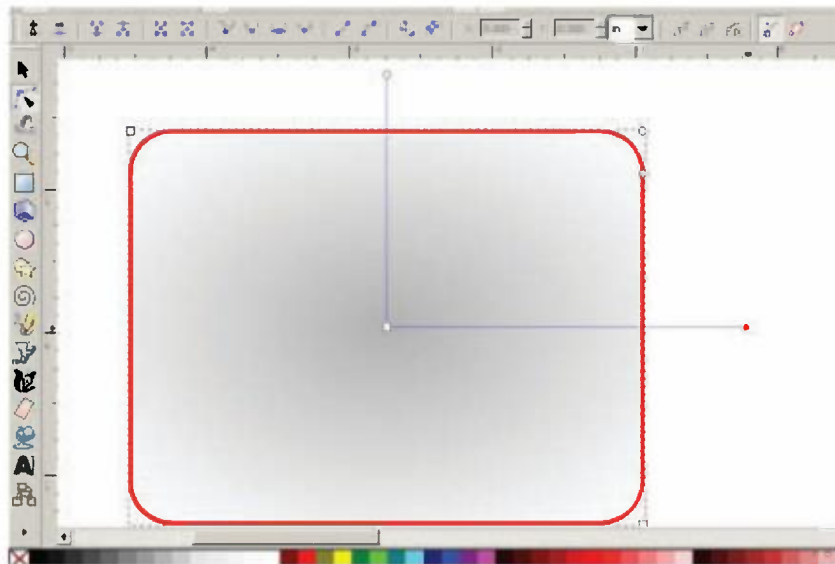
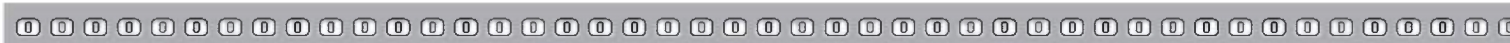


Figure 15. Adjust the scale of the gradient by dragging the horizontal and vertical control handles.

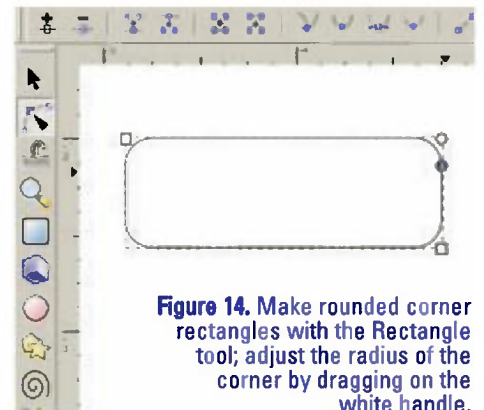


Figure 14. Make rounded corner rectangles with the Rectangle tool; adjust the radius of the corner by dragging on the white handle.

Now about that gradient. Rather than use a solid fill color, Inkscape can display a gradient of color — either fading off to white, or to one or more other colors. I’ve opted to keep it simple, and did this:

1. Select the main group box rectangle and fill with 30% gray.
2. In the Fill and Stroke panel, click on the Radial Gradient icon.
3. Click the Edit Paths icon, and re-select the rectangle. Handles for adjusting the scale of the gradient appear. As shown in **Figure 15**, drag the handles to increase or decrease the scale. I’ve increased the scale by moving both the horizontal and vertical handles outward.

The main group box has its own label along the top. I did mine in Arial Black, which is kind of a bolded-bold version of Arial. I then manually resized the text and place a small white-filled rectangle behind it to block out the red line underneath.

Note! Inkscape remembers the order of objects as you create them and places the most recent object over the

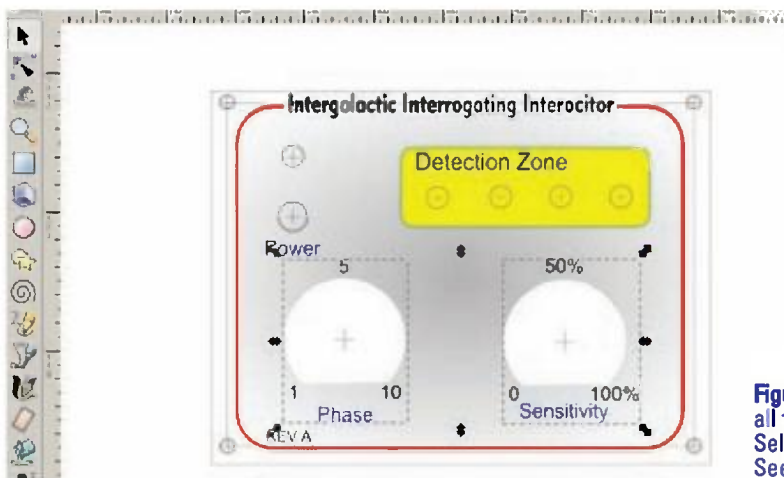
previous one. You can alter this “stacking” order by selecting the object you want to change, and choosing Object->Raise or Object->Lower. As a shortcut, you can also press the Page Up and Page Down keys.

So, for placing the white-filled rectangle behind the label text, position the rectangle where you want it to go, then click Page Down until it appears behind the text but over the red border of the main group box.

Finishing Up

Complete the layout for the Intergalactic Interrogating Interocitor by adding in the remaining text and aligning it as needed. Text is provided for the function and graduation scale of both pots, the power switch, and the four LEDs.

- Label text is Arial 15 point, blue color.
- Potentiometer graduations are Arial 14 point, black color.
- Main group label Arial Black, black color; size as noted above.



I’ve also added a small piece of text at the lower left to mark the revision number of the panel, in case it changes over time as the project is reworked — and what project isn’t! This text is Arial, black color, and 11 point.

Align the elements both visually and using Inkscape’s alignment tools. Text often needs to be adjusted visually because the invisible outlines of the text do not always match with the visible outlines of rectangles and other

Figure 16. Align the potentiometer dial faces by first selecting all the objects in each set. Then, group each set of objects. Select the two groups, and make one group of the pair. See Figure 17 to complete the task.

shape objects. For my sample panel, I've manually tweaked the position of the graduation numbers that circle the potentiometer dials by selecting the text and pressing the arrow keys. (Hint: To make finer adjustments, press and hold the Alt key when use the arrow keys.)

For the two potentiometer dials:

1. Group all the objects together for each dial (see **Figure 16**).
2. Group the two dials together.
3. Align the dials to the panel by first selecting the panel outline (remember, this makes it the "anchor"), then selecting the pair of dials. Align the objects horizontally (**Figure 17**).

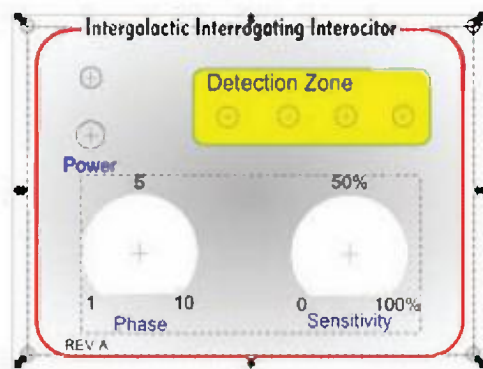


Figure 17. Align the dial faces by first selecting the panel outline, then the grouped potentiometer dials. Click on the Center Objects Horizontally icon.



Make a test print onto regular paper and confirm that the control panel dimensions are correct. Measure the outer rectangle with a ruler. It should be exactly 3" x 4". If it isn't, your printer may be scaling the image to fit. Look through the printer's options and turn off any selection that tells the printer to scale.

Next time around, I'll complete the how-to on making

professional looking control panels by detailing the ways to make colorful prints onto paper, vinyl, polyester, fabric, and other materials, plus ways to cut out customized control panels using a laser cutter or CNC machine. As this is a large field of discussion, I've set up some pages on my **Robotoid.com** website to help provide additional resources for you. **NV**



AP CIRCUITS
PCB Fabrication Since 1984

As low as...

\$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

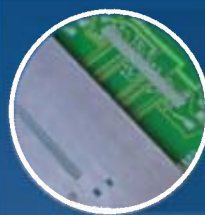
Unmasked boards ship next day!

www.apcircuits.com





THE ORIGINAL SINCE 1991
PCB-Pool
Beta LAYOUT



FREE Stencil
with every prototype order



Embedded RFID
authenticate, track & protect
your product

www.magic-pcb.com

Call Tyler: 1 707 447 7744
sales@pcb-pool.com

Motor Controllers • Arduino • Signal Generators • FM Transmitters • Timers • Audio Amplifiers

200 KHz Function Generator



\$49.95

ASSEMBLED

UK307

4-Port USB Relay Controller with
6-Channel Analog/Digital Interface



\$59.95

ASSEMBLED

UK1104

5A Motor Speed Controller (PWM)

\$22.95

ASSEMBLED



UK1102

Stereo Audio Amplifier (10W)



\$35.95

ASSEMBLED

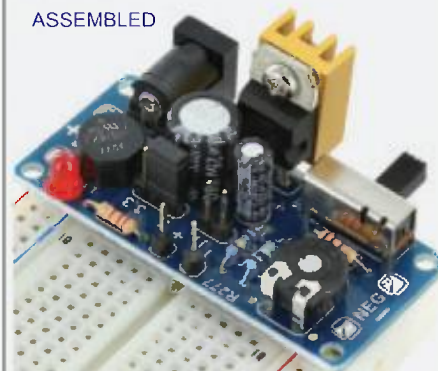
UK560

Plug-In Breadboard Power Supply

\$19.95

ASSEMBLED

UK277



7A Bi-Directional
Motor Speed Controller (PWM)

\$39.95

ASSEMBLED



UK1125

USB PIC Programmer
(MPLAB Compatible)



\$69.95

ASSEMBLED

UK1300

Digital LED Timer (Up/Down)

UK258



\$46.95

ASSEMBLED



SMILEY'S

#57 WORKSHOP

C PROGRAMMING - HARDWARE - PROJECTS

Part 4

ARDUINO PROTO SHIELD ALARM CLOCK

Can't see the forest for the trees? Yeah, it is kind of hard to follow these Workshops month after month without an occasional step back to look at where we've been and where we're going.

Back in the August 2012 *Nuts & Volts*, I began a four part series on Fritzing (Workshops 49-52). Then — still using Fritzing — I introduced the Arduino proto shield in December 2012.



■ Arduino Alarm Clock (AAC) PC Application.

Now, at Workshop 57, we are at Part 4 of the proto shield alarm clock series — this time mainly discussing software. Believe it or not, these nine Workshops are all related. What we have here is a series that could have been titled *Arduino Fritzing Prototype to Production*.

What we have done is to learn how to use the Arduino and Fritzing to take a concept from the first stages: a breadboard design, through a PCB (printed circuit board) prototype, to a production PCB. We've used an Arduino alarm clock as the demonstration project around which to hang the entire prototype to production learning.

So, naturally along the way, it got a bit confusing as to what we were really doing.

Were we learning how to use Fritzing or learning to use the proto shield, or learning to design an alarm clock?

The answer is yes — we were doing all of that. Don't worry, however, we are coming to the end of the tale of how to use Fritzing to take an Arduino design from prototype to production — and don't forget, we've also learned a lot about computer based alarm clocks.

I assure you the leaves fit the trees that fit the forest that we are trudging through. So, if while following me I moved a branch out of my way, then let it go and it smacked you in the face — sorry about that. Next time, you can lead the way.

PROCESS_ALARMS MODULE

This fourth module in our project processes the alarm functions. Recall that last month, we discussed the first three modules: Alarm_Clock, Commander, and Date_Time. We usually think of an alarm clock as having a big button that we can hit to shut off the buzzer in the morning. Our alarm clock, however, is much more capable than a simple wake-up device (besides, the button is tiny and hitting it might not be so easy when you are trying to wake up.)

We are using the example of having an alarm output a piezo tone and a button to shut it off, but this alarm can be adapted to pretty much control anything that an Arduino can control, and respond to pretty much anything an Arduino can respond to. We can adapt this code for use with a datalogger or an industrial controller or whatever we might imagine that needs to keep track of dates and times.

The go_alarm#() Functions

We've already seen that the Arduino *loop()* function in the Alarm_Clock module calls the alarm functions once per second when the *check_alarm* variable is set to 1 by the timer. When all the alarms have been checked, the *check_alarm* variable is set to 0 so that *loop* can skip checking the alarms. Each of the alarm functions is named *go_alarm#()* where # is 1 to 5. Now, let's look at these *go_alarm* functions.

Each alarm has seven eight-bit variables (*uint*_t*) associated with it: four are used to store the 32-bit Unix datetime variable; one holds the type of the alarm; one tells if the alarm is set; and the last tells if the alarm is tripped.

Check to See if the Alarm has Tripped

Each *go_alarm* function first checks the *alarm#_is_set* variable. If it is set, then it compares the alarm time with the current time, and if the current time is equal to or greater than the alarm, then the alarm is 'tripped' and the *alarm#_is_tripped* variable is set to 1. The user can then decide to handle the alarm immediately in the *go_alarm* function or leave this to another function in the *loop()* cascade that checks the *alarm#_is_tripped* variable.

I would recommend only using the *go_alarm* function for the alarm if you want to do something short and quick. You would not want to do something like run a tune on the piezo that continues until the button is pushed. For that, you might turn on the piezo tone generator, then return to the *loop()* where you'd loop until the

button has been pressed and then turn off the piezo.

Running the Piezo Buzzer in the Background

We created our piezo buzzer part using Fritzing in Smiley's Workshop 52, then we learned how to use it in the December 2012 Workshop 53. We found that the piezo is loudest at 4,300 Hz, and we saw how to make alarm patterns using the *delay()* function. However, the *delay()* function blocks the processor and we don't want to miss anything while waiting around for a beep to complete. So, we need to do something different.

We already have a timer interrupt that trips once per second, and we could use that to turn the piezo on or off in one second intervals. That is pretty long, though, so let's redo our timer interrupt to trip every 250 milliseconds so we can have shorter beep patterns.

Redo the Interrupt Timer

We were using the one second interrupt to set the *check_alarm* flag in the *loop()* function. We can continue to do this by keeping the new variable *check_alarm_count* that we increment every 250 ms and set the *check_alarm* flag when it has incremented four times. Then, we set it back to 0. We add the count variable at the top of the Alarm_Clock module:

```
// A timer is used to set the check_alarm flag
int check_alarm = 0;
int check_alarm_count = 0;
```

Then, we change the *myTimer* interrupt to keep the count:

```
if(check_alarm_count++ >= 3){
    check_alarm = 1;
    check_alarm_count = 0;
}
```

This causes the *check_alarm* flag to be set once per second as before. Now, we are free to use the timer for 250 ms events, so let's design some piezo beep patterns.

The Arduino proto shield alarm clock kit lets you build an alarm clock circuit on a breadboard and port that circuit to a PCB. This kit is the basis for my presentation of how to do a complete Arduino design cycle using Fritzing to go from a breadboard prototype, through schematic creation and breadboard layout, and finally producing your own printed circuit board. You can get the kit or materials that support this learning activity from the *Nuts & Volts* webstore.

Making Piezo Alarm Patterns

For this demonstration, we will create four alarm beep patterns.

1. Three short beeps (250 ms on, 250 ms off).
2. Three long beeps (500 ms on, 250 ms off).
3. SOS (three short, three long, three short);
4. Warble3 — two tones, 250 ms of 4,000 Hz, 250 ms of 4,300 Hz, 250 ms off, repeated three times.

[If you are a regular C programmer, this code is moderately cringe worthy, but since it is educational and for relative novices, I choose to make it easy to understand and not C efficient.]

```
int three_short_beep = 0; // 0 for off, not 0 for on
int three_short[] = { 1, 0, 1, 0, 1, 0, 2};
int three_long_beep = 0; // 0 for off, not 0 for on
int three_long[] = { 1, 1, 0, 1, 1, 0, 1, 1, 0, 2};
int sos_beep = 0; // 0 for off, not 0 for on
int sos[] = { 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 2};
int warble3_beep = 0; // 0 for off, not 0 for on
int warble3[] = { 1, 2, 0, 1, 2, 0, 1, 2, 0, 3};
int beep_count = 0;
```

```
// Rather than use the loop to check the alarm
// continuously and thereby adding quite a load
// on the processor, we will use a timer set
// to 250 ms.
```

```
void myTimer1()
{
    if(check_alarm_count++ >= 3){
        check_alarm = 1;
        check_alarm_count = 0;
    }

    if(three_short_beep){
        if(three_short[beep_count] == 0) noTone(8);
        else if(three_short[beep_count] == 1)
            tone(8,4300);
        else if(three_short[beep_count] == 2){
            three_short_beep = 0;
            beep_count = -1;
        }
        beep_count++;
    }

    if(three_long_beep){
        if(three_long[beep_count] == 0) noTone(8);
        else if(three_long[beep_count] == 1)
            tone(8,4300);
        else if(three_long[beep_count] == 2){
```

```
            three_long_beep = 0;
            beep_count = -1;
        }
        beep_count++;
    }

    if(sos_beep){
        if(sos[beep_count] == 0) noTone(8);
        else if(sos[beep_count] == 1) tone(8,4300);
        else if(sos[beep_count] == 2){
            sos_beep = 0;
            beep_count = -1;
        }
        beep_count++;
    }

    if(warble3_beep){
        if(warble3[beep_count] == 0) noTone(8);
        else if(warble3[beep_count] == 1)
            tone(8,4000);
        else if(warble3[beep_count] == 2)
            tone(8,4300);
        else if(beep_count >= 3){
            warble3_beep = 0;
            beep_count = -1;
        }
        beep_count++;
    }
}
```

These alarms continue as long as the variable (*three_short_beep*, *three_long_beep*, *sos_beep*, or *warble3_beep*) in the first *if()* statement is not 0. We will use the button detection to set all these variables to 0, thus turning off any alarm.

Once per second, the *myTimer* interrupt checks to see if an alarm is set and, if so, it calls the alarm handling function. Below we see the alarm function for alarm 1:

```
void goAlarm1(){
    #if defined(DEBUG)
        Serial.println(F("go_alarm1"));
    #endif

    if(alarm1.is_set){
        DateTime now = RTC.now();
        if(now.unixtime() >= alarm1.datetime.unixtime()){
            #if defined(DEBUG)
                Serial.println(F("Alarm1 tripped!"));
            #endif
            alarm1.is_tripped = 1;
            if(alarm1.type == 0){// If it is one shot,
                // unset it
                unsetAlarm('1'); // this also clears
                // the EEPROM data
            }
            else if(alarm1.type == 1){// if daily,
                add 24 hrs to the alarm
                #if defined(DEBUG)
```



```

Serial.print(F("Old alarm date time =
"));
showDate("alarm1",alarm1.datetime);
#endif
// Add one day, 86400 seconds
DateTime temp(alarm1.datetime.unixtime()
+ 86400L);
alarm1.datetime = temp;
#if defined(DEBUG)
Serial.print(F("New alarm date time =
"));
showDate("alarm1",alarm1.datetime);
#endif
// Write new alarm data to EEPROM
writeAlarm(1,alarm1);
} // if alarm1 is daily
} // if now > alarm1

// Do what you wanted to do with this alarm
// if it is tripped
if(alarm1.is_tripped){
  // if the button state is not pressed,
  // run the alarm
  if(!buttonState) warble3_beep = 1;
  else {
    alarm1.is_tripped = 0;
    // turn this alarm off
    warble3_beep = 0; // turn the beep off
  }
} // if alarm1.is_tripped
} // if alarm1._is_set
}

```

This function first checks to see if the alarm is set and if so, it sets the `alarm#_is_tripped` global variable so that other sections of the code can know that the alarm has been tripped. It then checks the alarm type. If the alarm is type 0 — a one-shot alarm — it “unsets” the alarm which also clears the EEPROM data for the alarm. If it is type 1 — daily — then it adds 24 hours (in seconds) to the alarm `datetime` variable so that it will trip again in a day.

Next, it checks to see if the alarm is tripped and, if so, it then runs the process for the alarm. In the above case, that process is to activate the `warble3_beep` and turn off the alarm (this one doesn't use the button). The piezo will then run the `warble3_beep` sounds. The whole process will repeat in a day. If we had not set the `alarm1.is_tripped` to 0, the piezo sound would repeat until the button is pressed.

ACCURACY AND CALIBRATION

DS1307 Accuracy

If you've been following along and built one of these Arduino alarm clocks, in among all the excitement of

learning new things you may have had a moment of disappointment when you realized that this thing isn't keeping time as accurately as you'd like. Mine gains about eight seconds a week. Well, bah! That's 416 seconds in a year — almost seven minutes. Don't panic yet! We can fix this — more or less — sort of. First, let's see what the DS1307 datasheet has to say about this issue:

The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast.

Our crystal is rated at ± 20 ppm (parts per million), so this would give ± 20 seconds in a million seconds (a million seconds is 16666.7 minutes, which is 277.7 hours, which is 11.6 days). Run the math and you get ± 1.7 seconds per day (about ± 10.3 minutes per year). Mine is gaining 1.23 seconds per day, so it is well within that specification. So, the RTC (real time clock) may gain or lose up to 12.3 seconds per week.

The amount that it gains or loses may vary depending on the ambient temperature. If the RTC is kept at a relatively constant room temperature, for example, then the gain or loss will be pretty constant over a long time period (years). In a moment, we'll see how to use an alarm to help keep the clock more accurately calibrated.

Maybe You Don't Want to Use It Outdoors

If the RTC is kept outdoors, then it will be subject to daily and seasonal temperature variations. The daily variations might average out over time, meaning that in any given week one might expect fluctuations to yield some sort of weekly mean that will change slowly from one week to the next.

For example, if a given week has highs mostly in the 70s and lows mostly in the 40s, then the mean will be in the mid 50s for that week. One might expect that the preceding week and the following week would also be more or less in the same range, and thus a calibration covering one week will be much the same as the following week.

When seasons change, however, the weekly mean temperature will rise or fall enough so that a calibration may be very different from that in another season. For a weekly calibration in the summer, it could be +10 seconds, while a weekly calibration in the winter could be -5 seconds.

For this reason, if you are going to design an RTC for the outdoors or anywhere with drastic temperature variations, I would recommend using an RTC with a built-in thermometer for automatic temperature compensation like the DS3234.



FIGURE 1. Calibration panel.



FIGURE 2. Valid calibration value.

More Accurate RTC With Built-in Thermometer

Our DS1307 costs about \$4 in singles from Mouser. They have another RTC — the DS3234 — with a built-in thermometer for about \$8.35 in singles. It has an accuracy of ± 2 minutes per year. Although I personally think it is worth the price, it only comes in surface-mount, so we'll stick with the through-hole DS1307 and provide a calibration technique that *might* make it nearly as accurate as the expensive one.

Calibrating Your Arduino Alarm Clock

I ran my AAC for a couple of weeks and checked it more or less daily to determine that it is gaining 1.23 seconds per day. We calculate that a day has 86,400 seconds and since 1.23 seconds per day is the same as 0.813 days per second, we can multiply the days per second times the seconds per day and get 70,244 seconds (70244.9 rounded up).

This is the number of seconds it takes for our AAC to gain one second. So, if we set an alarm to go off every 70,244 seconds and subtract one second from our 'real' time, we correct for the gain. I'm not sure how accurate this will be over the long term, but I'd speculate that it would be at least as accurate as the more expensive part.

Now, what we need to do is add a command in the PC AAC application to send the Arduino the calibration seconds, and whether to add or subtract a second at each interval. Then, in the Arduino, we need to build an alarm in the AAC that goes off every 'calibration' second, and either adds or subtracts a second depending on what it was told. [Yes, we could further automate this process so that we wouldn't have to manually calculate the calibration seconds, but I figure we'll only need to do this once so for me, it isn't worth the extra effort.]

Let's review how we set and check alarms so that we can see how to add the calibration alarm. In the Commander module, we parse the input bytes from the PC in the `commandArray[]` and see that the first byte is

the character 'A.' Then, we call the `ACase()` function where we further parse the input bytes, noting that the second byte in the `commandArray[1]` is the number of the alarm (1 to 5).

We then call the `setAlarm()` with the parameter being the pointer to the `DateTime` class instance for that alarm. The `setAlarm()` function then reads the rest of the `commandArray` bytes to get the date and time to set the alarm, and loads that into the specific alarm instance.

Once the data is loaded, then we can retrieve the *unixtime* for that alarm (which you may

remember is the number of seconds since January 1, 1970). Now that the alarm is set, our timer interrupt sets the `check_alarm` flag once per second, and the `loop()` function looks to see if that flag is true.

If it is, then it checks each alarm to see if it is set. If it is, then it calls the function specific to that alarm. That specific alarm function then does whatever it is programmed to do and finishes off by setting the `check_alarm` flag to false.

We will create a new command 'C' to handle the calibration alarm process, but we will try to use as much of the regular alarm process as makes sense. We will look at the Arduino side of this next month. For now, let's see how the PC application is modified to send the data.

Modify the PC Application

We need to send a calibration number and whether to add or subtract a second to the AAC. So, let's use a text box for the number and two radio buttons for the add or subtract. We'll also need a send button when we've input the data so that data is sent to the AAC.

The calibration number will need to be 32 bits to accommodate the value, so to simplify things we'll break the number into four bytes and send them separately. We'll also send a fifth byte with either 0 for subtract or 1 for add. The Arduino will then parse this input packet and set the calibration alarm.

Note in **Figure 1** that we've added a Calibration Seconds panel to the PC application that lets the user enter the number of seconds to use as an alarm to either add or subtract a second, depending on which radio button is pressed.

When you press the Send button, you get the message shown in **Figure 2** that shows the AAC returns that it was in CCase (if you are in the debug mode). It returns the number you sent along with the add/subtract state so you can verify that what it thinks it got is what you think you sent.

The C# Calibration Seconds Function

This wasn't quite as simple as some of the other additions we've made, but the following C# code for the PC should be somewhat self-explanatory.

It was a bit of a hassle figuring out how to translate the user input into bytes that can be sent to the AAC, and the methods chosen may not be the most efficient.

However, they work, so let's move on.



FIGURE 3. Calibration value too large.

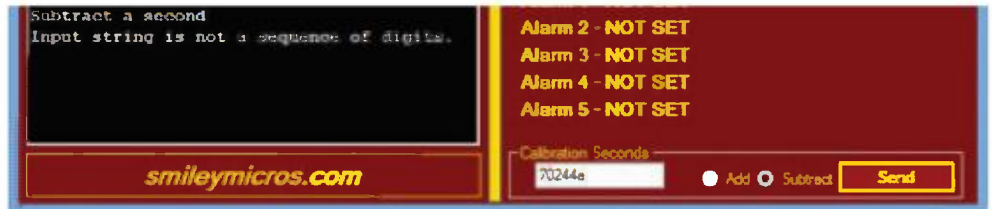


FIGURE 4. Calibration value not a number.

```
// Algorithm copied from MSDN and modified for
// this use
private void buttonSendCalibration_Click(object
sender, EventArgs e)
{
    string calibration;
    UInt32 numVal = 0;
    var val = new Byte[7];
    Byte addsubtract = 0;

    if (radioButtonAdd.Checked == true)
    {
        richTextBoxReceive.Text += "Add a
second\n";
        addsubtract = 0;
    }
    else if (radioButtonSubtract.Checked
== true)
    {
        richTextBoxReceive.Text += "Subtract
a second\n";
        addsubtract = 1;
    }

    calibration = textBoxCalibrationSeconds.Text;

    // ToInt32 can throw FormatException or
    // OverflowException.
    try
    {
        numVal = Convert.ToUInt32(calibration);
    }
    catch (FormatException)
    {
        richTextBoxReceive.Text += "Input string
is not a sequence of digits.\n";
        return;
    }
}
```

```
catch (OverflowException)
{
    richTextBoxReceive.Text += "The number
cannot fit in an Int32.\n";
    return;
}

val[0] = Convert.ToByte('C');
val[1] = (Byte)(numVal & 0x000000FF);
val[2] = (Byte)((numVal & 0x0000FF00) >> 8);
val[3] = (Byte)((numVal & 0x00FF0000) >> 16);
val[4] = (Byte)((numVal & 0xFF000000) >> 24);
val[5] = addsubtract;
val[6] = Convert.ToByte('!');

serialPort1.Write(val, 0, 7);
}
```

I got the PC application algorithm from MSDN and it had a couple of error catches that help the user to get the calibration number correct. **Figure 3** shows catching the error that the number is too large, so it isn't sent to the Arduino.

Another error catch in **Figure 4** shows what happens if you send something that isn't a number like 70244a. This won't catch every kind of error, but we can assume some level of good sense in the user, so it should suffice.

Next month, we will continue learning how to calibrate the AAC by looking at the Arduino code for this process. We need to add or subtract a second after a specified period, then repeat the process. I bet you think it will be easy?

Well it wasn't for me, but then nothing ever is — tune in next month! Remember all this mix of software and hardware is part of our long term learning about *Arduino Fritzing Prototype to Production*. **NV**

WALKING THE USB BRIDGE INTO ANDROIDVILLE

If you follow the Design Cycle, you will recall a discussion based on the FTDI Vinculum microcontroller. As microcontrollers go, the Vinculum embodies all of the things that make silicon a microcontroller plus logic that implements the FTDI USB host protocol. With a little bit of programming, the Vinculum easily interfaces directly to any device that adheres to the FTDI USB protocol standard. That means the Vinculum can directly interface to FTDI-based USB memory sticks and RS-232 conversion cables. Extending that thought a bit further, our ability to design in a downstream FTDI USB IC allows the attachment of project-specific USB interfaces.

The Vinculum is a great answer to USB host-mode interfacing with FTDI-based devices. When it comes to the Android, the Vinculum can also romp in that playground with a bit of programming. If hardware is your forte and Android programming is a necessary evil, you will want to keep reading.

THE FT311D USB ANDROID HOST IC

The FT311D is a USB host IC that stands between an Android USB port and an external device. The FT311D is actually a bridge that supports a host USB interface on the Android side, and GPIO, SPI Master/Slave, PWM, I²C Master, and UART interfaces on the device side. Like most FTDI USB ICs, the FT311D does all of the complex USB work on the chip. The FT311D can work with any Android platform that supports Android Open Accessory Mode. The external downstream device can also be identified as an Android accessory.

Internally, the FT311D multiplexes the six peripheral's I/O via three configuration pins (CNFGx). The FT311D peripherals are technically called peripheral interface modules. All of these modules share the common FT311D I/O pins. The GPIO module supports seven I/O pins. The SPI master/slave module serves up the standard four-wire interface (CLK, MOSI, MISO, SS). The UART module is full featured, including RTS/CTS modem control

lines and a TX-active line for RS-485 operation. There are four PWM outputs and a standard I²C master implementation. The eighth GPIO pin is dedicated to USB error indication.

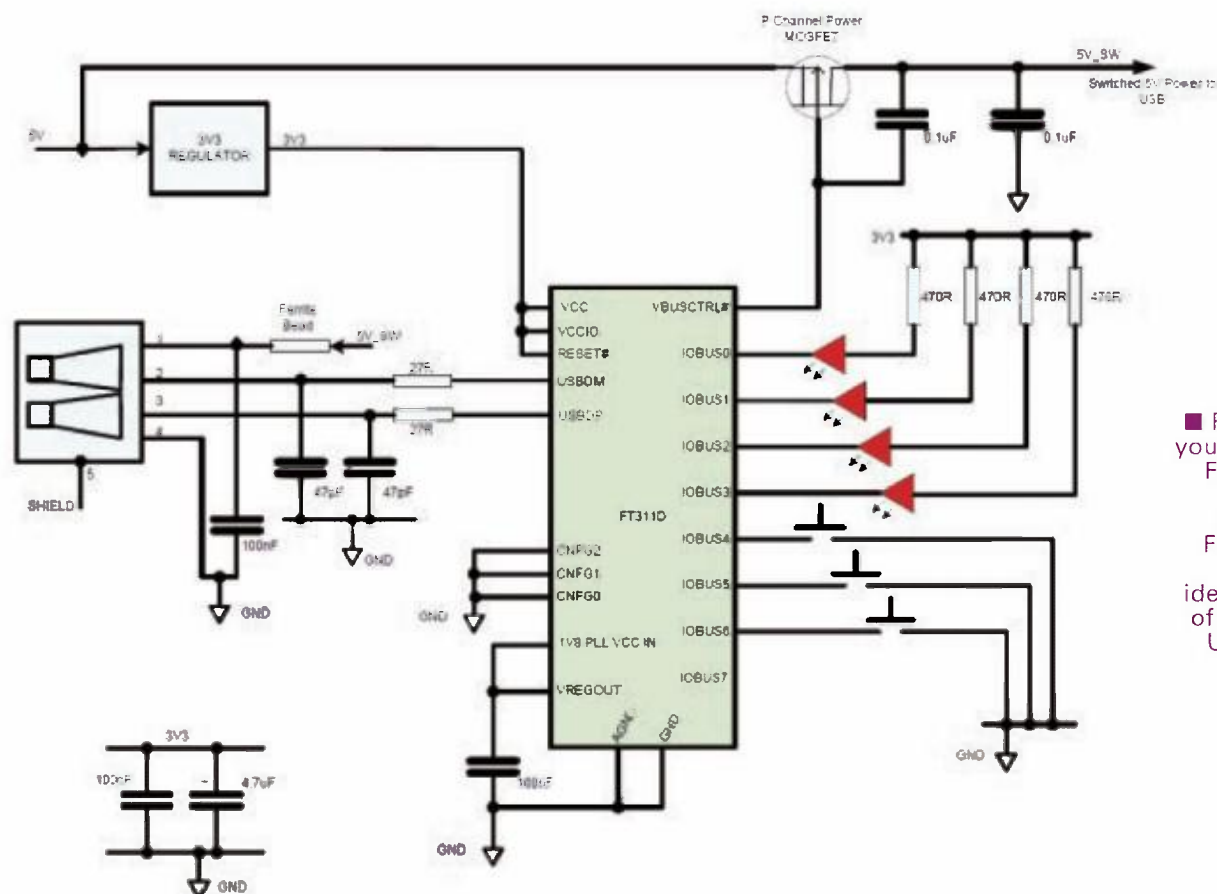
All of the physical 1.1 and 2.0 USB interfacing is handled by the FT311D. It can run at low speed and full speed USB data rates. The USB host interface feeds a +3.3 volt I/O device interface. The incoming +3.3 volt power rail is internally regulated to +1.8 volts to provide power for the FT311D's PLL.

As you probably already know, a USB host must supply +5.0 volts on its USB interface. The incoming +5.0 volt power that supplies the FT311D's external +3.3 volt voltage regulator is also routed to the source pin of a P-channel MOSFET. The FT311D's active-low VBUSCTRL pin is attached to the MOSFET's gate. Power to the USB host portal is controlled by the logic level of the FT311D's VBUSCTRL pin. A typical FT311D GPIO implementation is shown in Figure 1.

FT311D HARDWARE

The FT311D is available in 32-pin QFN and LQFP packages. The 33rd pin you see in the schematic is actually the exposed pad area under the QFN package. The exposed pad area should be grounded. We're not going to hand-fabricate any FT311D hardware this time around. Instead, we'll use the Android USB host evaluation kit shown in Photo 1.

In any case, if you're planning on



■ FIGURE 1. As you can see, the FT311D needs very little support. The FT311D's USB interface is identical to that of the FT232RL USB-to-UART bridge.

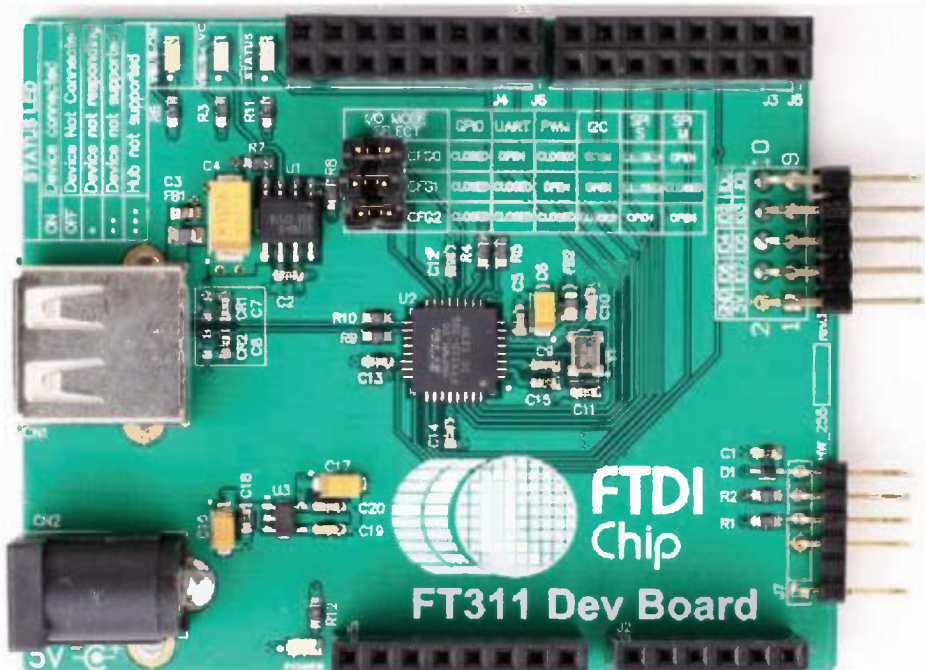
using the FT311D, you'll have to download its documentation. So, instead of duplicating the full-blown host evaluation kit schematic in this text, we will examine the FT311D electronics in print. You can follow the description flow using your downloaded schematic package. You can find the schematic in the datasheet. Just click on the Android link on the FTDI home page.

Let's begin by taking a look at the FT311D's power subsystem. **Photo 2** is a fly's eye view of the voltage regulator and its supporting components. There is no rocket science here. However, since the FT311D is a host USB device, it must provide +5.0 volts to the host USB connector's VBUS pin. The VBUS voltage originates from a regulated five volt wall wart. The FT311D needs a 3.3 volt power rail and that is provided courtesy of U3, which happens to be a 3.3 volt LDO voltage regulator.

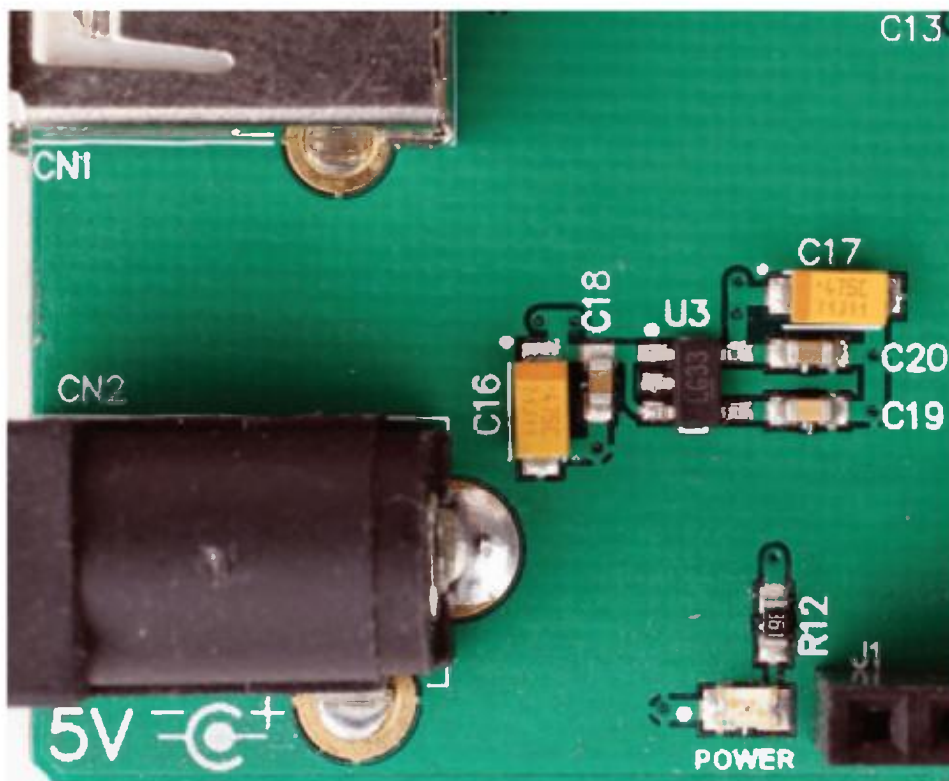
A downstream USB device that is being serviced by a USB host can

draw a maximum of 500 mA. The job of the active device (U1) you see in **Photo 3** is to insure that the downstream USB device does not

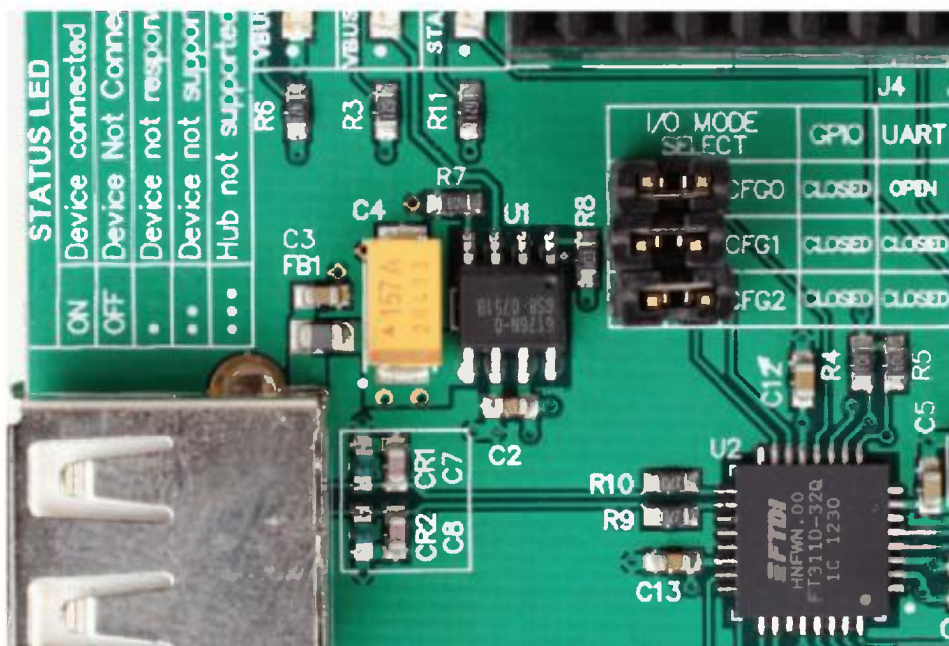
put an excessive load on the host USB power subsystem. U1 — an AIC1526 dual USB high-side power switch — is a specialized IC designed



■ PHOTO 1. We can't scratch-build this for less than the out-of-the-box price.



■ PHOTO 2. There is nothing new and exciting here. It looks to be just another 500 mA LDO voltage regulator with shutdown, surrounded by the normal complement of filter capacitors.



■ PHOTO 3. The AIC1526 dual USB high-side power switch is a better choice here than a no-frills P-channel MOSFET.

for use in USB power systems. The AIC1526 can switch 500 mA continuously on each of its two channels, and both channels include safeguards such as current limiting, short-circuit protection, thermal

shutdown, and undervoltage lockout.

The USB host evaluation kit utilizes the B portion of the AIC1526. The B-side MOSFET is permanently activated by grounding the B-side enable pin. The A-side is disabled by

tying the A-side enable pin logically high. If a USB power fault occurs, the LED attached to the B-side error flag pin is illuminated. Otherwise, the VBUS power LED glows bright green.

The business end of this evaluation kit can be seen in the aerial reconnaissance shot represented by **Photo 4**. If I had to guess, I would say that the FT311D is a chunk of Vinculum specifically programmed to perform the USB/Android bridge function. The reason I think this way is swayed by the presence of the 12 MHz crystal.

If you examine the USB host portal hardware, it smells heavily of the FT232R USB-to-RS-232 converters. The FT311D's internal identification EEPROM can also be modified in a similar fashion to that of the FT232R.

There is a programming port (J7) that is used to load the FT311D's operational ROM. The FT311D's similarity to the FT232R takes much of the pain out of the hardware design process.

HELPER HARDWARE

Take another look at **Photo 1**. Note that there aren't any superfluous LEDs or pushbuttons. Everything on the kit main board means business. The resident LEDs are used to display power and operational status. The jumper blocks are used to select the desired I/O mode. It's easy to see that the FT311D's I/O pins are feeding the female headers, as well as the double row of male headers just above the programming port.

If you are thinking Arduino, you're close but no cigar. The female header interface is intended to support a Vinco shield. If Vinco is not ringing any bells, pull out your September 2011 issue of *Nuts & Volts* and check out the Design Cycle column. While you're reloading on Vinco, take a look at the Vinco programmer interface. It looks a lot like the FT311D programming interface. Hmmmmmm ...

The cat flailing around in the bag

is showing its claws in **Photo 5**. The FT311 GPIO board can be used with the USB host evaluation kit, as well as the Vinco. When coupled to this kit, the FT311 GPIO board's 300 mA LDO voltage regulator is powered from the same wall wart that supplies the +5 VDC to the kit.

An Android demo program is available on the FTDI site that interfaces directly with the FT311 GPIO board's pushbuttons and LEDs. However, running the Android GPIO demo program isn't as much fun as seeing your home-grown Android app blink those LEDs.

A HOME-GROWN FT311D ANDROID APP

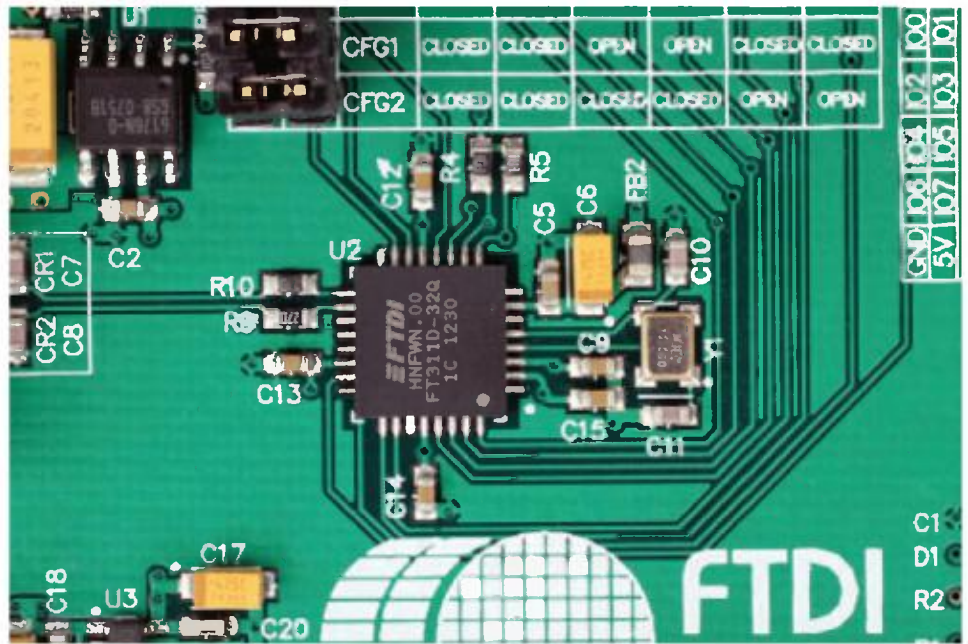
Our FT311D Android application begins in this manner:

```
Sub Process_Globals
    Dim i As Int
    Dim manager As UsbManager
    Dim packetOut(4) As Byte
    Dim outFT311 As
OutputStream
    Dim inFT311 As InputStream
    Dim FT311Acc() As
    UsbAccessory
End Sub
```

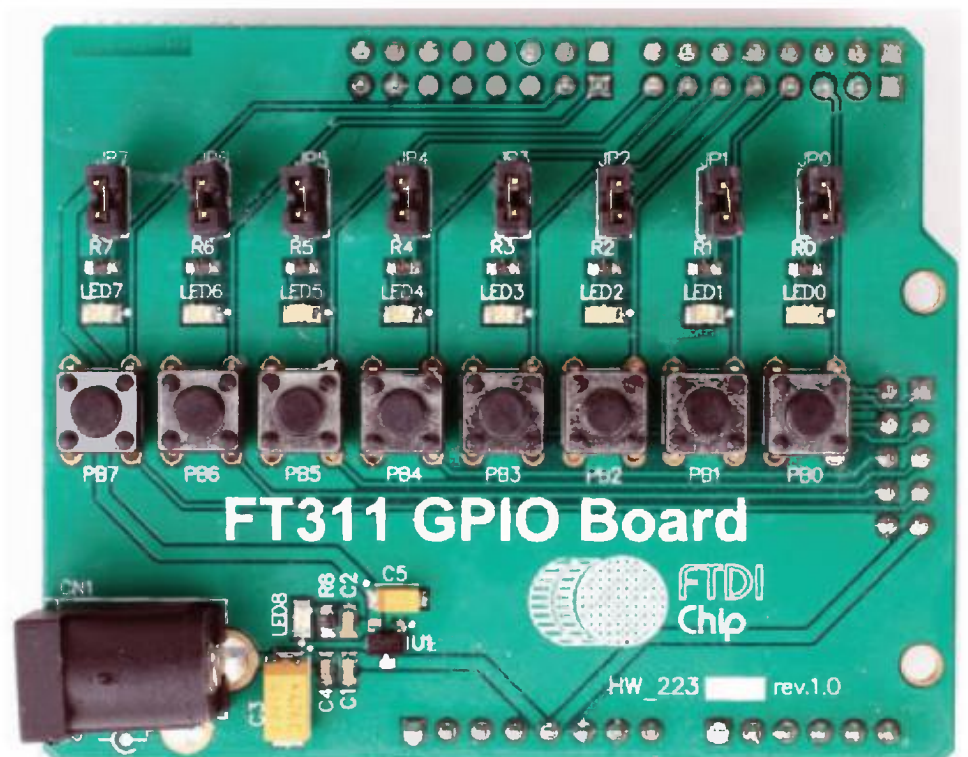
With the exception of the *"i"* variable and the *packetOut* array, the rest of the declarations are spawning instances of objects. The referenced USB objects are realized in Basic4android's USB library. Basic4android is a rapid application development environment aimed solely at the development of Android applications.

If you've delved into classic Android programming, you know that Android application development is based on Java and the Android SDK. Basic4android uses the same aforementioned resources to produce native Android APK files that do not require any runtime assistance.

In that Basic4android is a combined IDE/compiler, it eliminates the need for an external IDE such as Eclipse. The *Dim* declarations suggest that programming with Basic4android is very similar to programming a Visual Basic application. In fact,



■ PHOTO 4. Like everything from FTDI, the FT311D supporting component layout is nice and clean.



■ PHOTO 5. Are these enough pushbuttons and LEDs for you? Plus, they come with their very own power supply.

Basic4android looks very much like Visual Basic.

For those of you that are Visual Basic programmers, your Basic4android learning curve has been shortened considerably.

Let's get back to the Basic4android code. We will use a

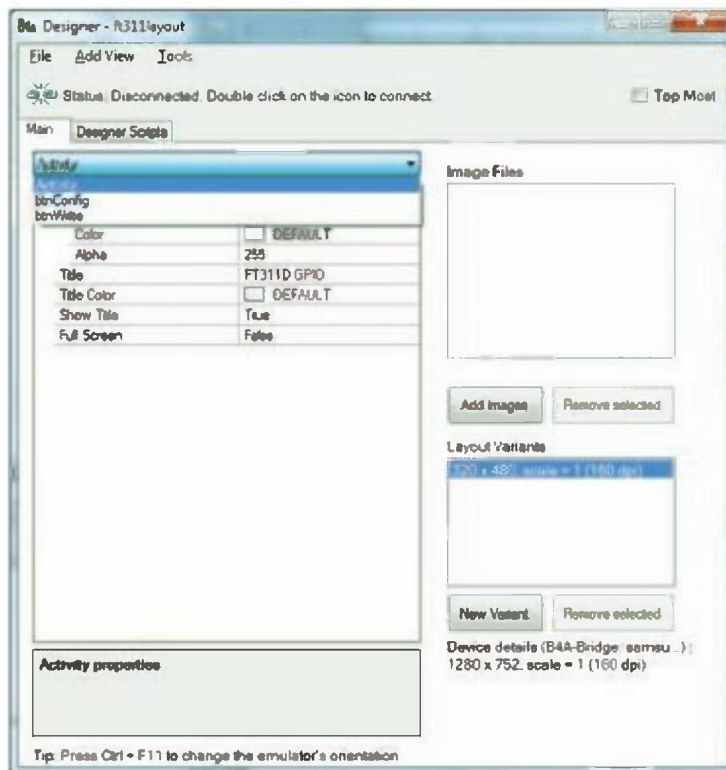
Samsung Galaxy Tab to communicate with our FT311D accessory. At a minimum, we'll need to wake up the *UsbManager*, *InputStream*, *OutputStream*, and *UsbAccessory* objects. The *UsbManager* object provides access to the connected USB devices. USB class constants are

also defined within the `UsbManager` object. `InputStream` and `OutputStream` are members of the `UsbAccessory` object. Each attached USB accessory will contain a unique `InputStream` and `OutputStream`. Other unique members of the `UsbAccessory` object include a description of the accessory, the accessory's serial number, and the version of the accessory.

The FT311D accessory communicates with the host device in four-byte chunks. That explains the four-byte `packetOut` array declaration. For debugging purposes, the `"i"` variable is declared as a global variable and is used in standard indexing algorithms. All of the objects and variables declared within the `Process Globals` source code area can be accessed by any of the Android application's modules.

PUSHING BUTTONS OVER THE BRIDGE

Like Visual Basic, Basic4android is event driven. So, we'll need some event generators that we can control.



■ SCREENSHOT 1. Designer works just like the properties editor in Visual Basic.

That equates to adding buttons to our Android application. This is done within Basic4android using Designer. I've added a configuration button and write button to our Android FT311D application. This is evident in **Screenshot 1**. The Basic4android Designer allows the placement of the buttons in real time on the Samsung Galaxy Tab's display. The Tab is physically attached to the FT311D USB host evaluation kit via a USB

cable. To use Designer's resources, we must somehow connect the Basic4android environment to the Tab. That is accomplished using the B4A bridge which can be crossed in a couple of ways: We can use Bluetooth or Ethernet. Ethernet is the best choice since all of my Windows and Android devices share the shop LAN.

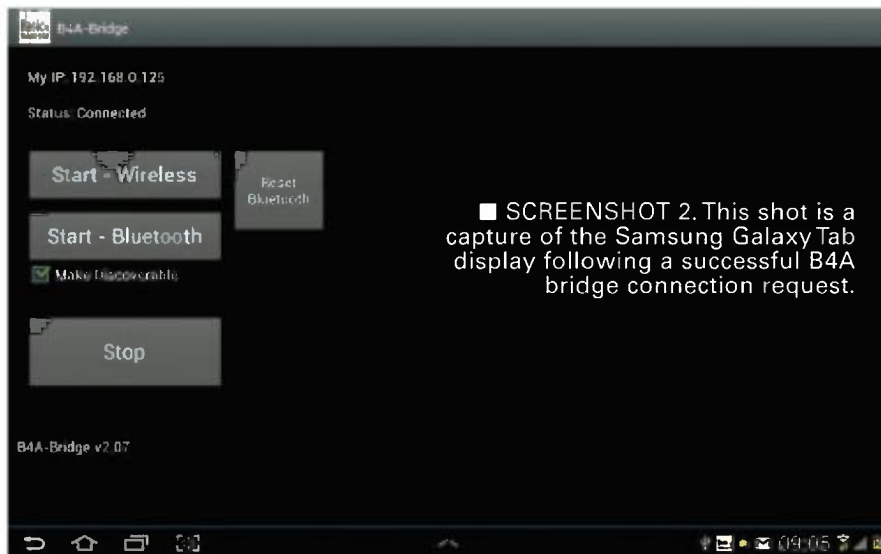
So, at this point the host evaluation kit with the piggy-backed FT311 GPIO board are connected to the Galaxy Tab by a USB cable. Once I tap on the Designer's Status icon, the Basic4android programming environment will be in touch with the Tab via Wi-Fi.

The B4A bridge allows us to load and debug the FT311D Android application over the Wi-Fi connection. **Screenshot 2** depicts a successful connection.

With a B4A bridge session established, we can call upon the resources of Designer to lay down our buttons. The port configuration and port write buttons are under the Designer's control in **Screenshot 3**. At this point, I can simply touch the buttons and drag them into the desired positions.

Designer also allows me to choose the button sizes and colors. I have control of what is displayed in the banner area of the display. As you can see in **Screenshot 3**, I've displayed "FT311D GPIO." This was done by altering the Activity properties member title in the Designer.

Once the buttons were to my liking, I accessed the Tools area of Designer to generate the button members. The member generation process placed the button declarations in the `Globals` source code area. Items declared as global can only be accessed by the module



■ SCREENSHOT 2. This shot is a capture of the Samsung Galaxy Tab display following a successful B4A bridge connection request.

they reside within. Here is what Designer placed in the FT311D Android application source code:

```
Sub Globals
    Dim btnConfig As Button
    Dim btnWrite As Button
End Sub
```

The member generation process also allows us to specify event types associated with the member. We want to simply touch the buttons to generate an event, so I chose to generate Click events for both buttons. The resulting code looks like this:

```
Sub btnWrite_Click
End Sub

Sub btnConfig_Click
End Sub
```

CODING THE BUTTONS

As it stands, the buttons will jump to their respective Click routines, but nothing will happen. We need to perform some useful work between the Sub and End Sub declarations.

Let's begin by issuing a Configure Port command with the btnConfig_Click event. First, we will write the actual command code:

```
Sub ConfigPort(outMap As
Byte,inMap As Byte)
    outMap = Bit.AND (outMap,0x7F)
    ' GPIO pin 7 Not Valid
    inMap = Bit.AND (inMap,0x7F)
    ' GPIO pin 7 Not Valid
    packetOut(0) = 0x11
    'configure port command
    packetOut(1) = 0x00
    'not used
    packetOut(2) = outMap
    'output bit map 1 = output
    packetOut(3) = inMap
    'input bit map 1 = output

    outFT311.WriteBytes(packetOut,
0,4)
End Sub
```

Our code configures the FT311D GPIO port as full output. The most significant bit of the GPIO port is the USB error indicator which cannot be used as a GPIO pin. Any presence of a 1 bit in the inMap or ourMap variables assigns the corresponding pin as an input or output pin,



■ SCREENSHOT 3. The process of generating touchable Android components is almost identical to generating and customizing objects in Visual Basic.

respectively. The ConfigPort routine is called from the btnConfig_Click event. Thus, we simply add the filler as shown here:

```
Sub btnConfig_Click
    ConfigPort(0x7F,0x00)
End Sub
```

Now that the GPIO pins are all output pins, we can extinguish the FT311 GPIO board's LEDs by writing a logical 1 to that bit position. Conversely, writing a logical low will illuminate the associated LED.

Writing binary data to the FT311D GPIO port is almost identical to configuring the GPIO port:

```
Sub WritePort(portData As Byte)
    portData = Bit.AND
(portData,0x7F)
    ' GPIO pin 7 Is OUT
    packetOut(0) = 0x13
    'write port command
    packetOut(1) = portData
    'data to write
    packetOut(2) = 0x00
    'not used
    packetOut(3) = 0x00
    'not used

    outFT311.WriteBytes(packetOut,
0,4)
End Sub
```

To extinguish all of the FT311 GPIO board's LEDs, we would write 0x7F to the FT311D's GPIO pins via the btnWrite_Click event:

```
Sub btnWrite_Click
    WritePort(0x7F)
End Sub
```

I don't think I need to tell you how to illuminate all of the LEDs. However, you're probably wondering why I have failed to explain away the last source code line in the ConfigPort and WritePort routines.

PUTTING THE CART BEFORE THE HORSE

We must do some homework before we can perform any of the aforementioned I/O operations against the FT311D. The Basic4android Activity_Create routine must be run before anything else can occur. The Activity_Create routine only runs one time and performs similar tasks such as those found in the Arduino setup function:

```
Sub Activity_Create(FirstTime
As Boolean)
    Activity.LoadLayout
("ft311layout")
    manager.Initialize

    FT311Acc = manager.
GetAccessories
    For i= 0 To FT311Acc.Length
        - 1
        Log( i & " Accessory " &
FT311Acc(i))
    Next
    Log (FT311Acc(0).Description)
    Log (FT311Acc(0).Manufacturer)
    Log (FT311Acc(0).Model)
    Log (FT311Acc(0).Serial)
    Log (FT311Acc(0).Uri)
    Log (FT311Acc(0).Version)

    If manager.HasAccessory
Permission(FT311Acc(0))
= False Then
manager.RequestAccessory
```



```

** Activity (main) Create, isFirst = true **
0 Accessory anywheresoftware.b4a.objects.usb.UsbManagerWrapper$UsbAccessoryWrapper$414779d0
Vinculum Accessory Test
FTDI
FTDIGPIODemo
VinculumAccessory1
http://www.ftdichip.com
1.0
Permission Granted
Connected
** Activity (main) Resume **

```

■ **SCREENSHOT 4.** The Basic4android logging function is handy in that it allows you to debug using program flow instead of breakpoints.

```

Permission(FT311Acc(0))
End If

```

```

If manager.HasAccessory
Permission(FT311Acc(0))
= True Then

```

```

    Log("Permission Granted")
    manager.OpenAccessory
    (FT311Acc(0))
    Log("Connected")
    outFT311 = FT311Acc(0).
    OutputStream
    inFT311 = FT311Acc(0).
    InputStream
Else
    Log("Permission Denied")
End If
End Sub

```

Recall that we used Designer to place our buttons and generate the skeleton code to support them. Our Designer button layout was saved as *ft311layout*. The very first thing we do is pull this layout file to the display. Once our buttons are displayed and the banner is visible, we are ready to connect to the FT311D accessory. The only way to make any type of Android USB connection is to invoke the members of the *UsbManager* object which we instantiated as *manager*.

We only have one physical

accessory attached which is our FT311D USB host evaluation kit. So, our accessory list – which is kept in *FT311Acc* – will only contain an entry at array slot zero. The accessory list is collected invoking *manager.GetAccessories*.

Basic4android includes a very useful built-in logging functionality. Take a look at **Screenshot 4**. From top to bottom, the log states that this is the first time through *Activity_Create*. We know that only one accessory is attached which is reflected by the 0 accessory entry. The rest of the accessory log entries fall into place like this:

```

Log (FT311Acc(0).Description)
Vinculum Accessory Test
Log (FT311Acc(0).Manufacturer)
FTDI
Log (FT311Acc(0).Model)
FTDIGPIODemo
Log (FT311Acc(0).Serial)
VinculumAccessory1
Log (FT311Acc(0).Uri)
http://www.ftdichip.com
Log (FT311Acc(0).Version)
1.0

```

Screenshot 5 appears at this point of execution within the

Activity_Create routine. A touch on OK grants permission to access the FT311D accessory and a positive acknowledge message is logged. With the granting of access, we can now initialize the input and output streams to the accessory:

```

outFT311 = FT311Acc(0).
OutputStream
inFT311 = FT311Acc(0).
InputStream

```

With the I/O streams registered, we can now issue the following output command:

```

outFT311.WriteByte(packet
Out,0,4)

```

We can also get the status of the FT311 GPIO board's pushbuttons by declaring an input buffer (*inPacket*) and issuing a Read Port command (0x12):

```

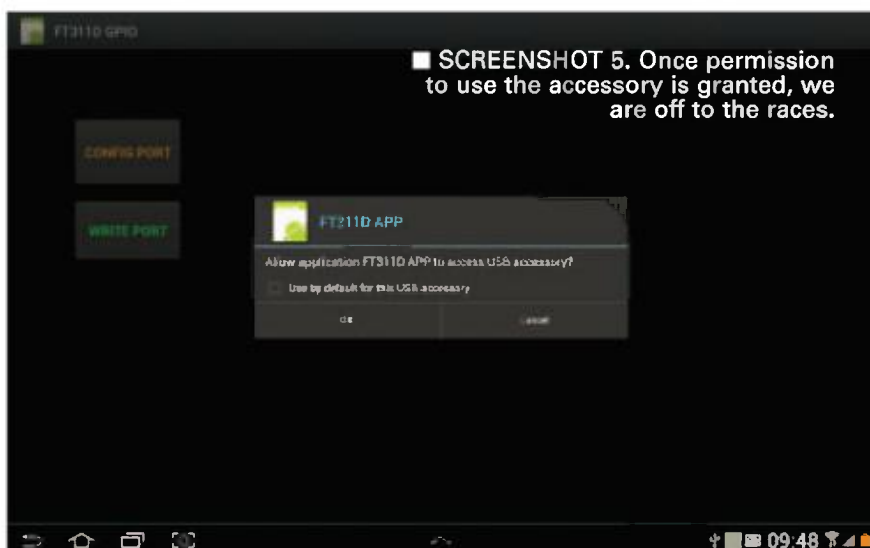
Sub btnReadPort_Click
Dim a As Byte
outPacket(0) = 0x12
outPacket(1) = 0x00
outPacket(2) = 0x00
outPacket(3) = 0x00

outPort.WriteByte(
outPacket,0,4)
a = inPort.ReadBytes
(inPacket,0,4)
End Sub

```

FLY BY WIRE

Believe it or not, all of the Basic4android FT311D code is contained within the words and graphics of this edition of Design Cycle. You just learned how to use your favorite Android device's USB port to control devices using FT311D and Basic4android without any previous knowledge of Java, Eclipse, or XML. Your Design Cycle now includes the Android SDK by way of Basic4android. **NV**



■ **SCREENSHOT 5.** Once permission to use the accessory is granted, we are off to the races.

SOURCES

Anywhere
Software
Basic4android
www.basic4ppc.com

FTDI
FT311D
Android USB
Host Evaluation Kit
FT311D
FT311 GPIO Board
www.ftdichip.com

ELECTRONET

GET THE NUTS&VOLTS DISCOUNT!
Mention or enter coupon code **NVRMZ12**
and receive 10% off your order!

ramsey
www.ramseykits.com
AM/FM Broadcasters • Hobby Kits
Learning Kits • Test Equipment
...AND LOTS OF NEAT STUFF!



THE ORIGINAL SINCE 1994
PCB-POOL
Beta LAYOUT

- Low Cost PCB prototypes
- Free laser SMT stencil with all Proto orders

WWW.PCB-POOL.COM

INVEST in your BOT!



HiTEC
12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecrod.com

ALL ELECTRONICS CORPORATION
Electronic Parts & Supplies
Since 1967

For the ElectroNet online, go to
www.nutsvolts.com
click **Electro-Net**

Got electronics?
www.nutsvolts.com

Beginner's Guide Complete Combo!



Combo Price \$229.95
VOL3 Experiment Component Pack

For complete details, visit our webstore @ **www.nutsvolts.com**.

www.canakit.com

- Over 200 Kits & Modules to choose from!
- Easy Worldwide Online Ordering
- Fast Delivery & Same Day Shipping

FM Transmitters • Timers • Audio Amplifiers • Motor Controllers



USB Add USB to your next project--
It's easier than you might think!

DLP Design
USB-FIFO • USB-UART • USB/Microcontroller Boards
RFID Readers • Design/Manufacturing Services Available
Absolutely NO driver software development required!
www.dlpdesign.com

superbrightleds.com

Component LEDs • LED Bulbs • LED Products
St. Louis, Missouri - USA Fast Online Ordering
superbrightleds.com



Instrument Shop Supervisor Needed
Department of Engineering
Initial Deadline: 11:59pm, 3/7/13
For qualifications/application info, see Job ID #30386 at **www.asu.edu/asujobs/**
ASU college of technology & innovation

Pololu
Robotics & Electronics
WWW.POLOLU.COM



Serial LCD & LED
www.melabs.com



Proto Buddy Book & Kit Combo



Includes an article reprint!
Only \$59.95

For complete details visit our webstore @ **www.nutsvolts.com**

OPEN_{COMMUNICATION}

■ BY LOU FRENZEL W5LEF

THE LATEST IN NETWORKING AND WIRELESS TECHNOLOGIES

60 GHz Wireless Transports HDTV WiGig and WirelessHD

Transmitting uncompressed digital high definition television (HDTV) signals from set top box to DVD or DVD to large screen display is no easy task. Depending on what HD format you are using, this video could have a data rate up to 1.485 Gbps. That is a super high rate that only a few cables can handle. The best cable connection is probably fiber, but a copper HDMI cable will also do it. That is what most HDTV systems use today.

Wouldn't it be great to do this wirelessly? No cables, just the freedom of open space. The question is what kind of wireless technology can deliver that kind of data rate affordably? Wi-Fi wireless systems using the 802.11n or the newer 802.11ac standards can easily transport compressed video at a few hundred Mbps rate (which is commonly done). Unfortunately, video compression does compromise the detail in the video to some extent, and many can see the difference.

So, what we need is a faster wireless technology to handle the uncompressed HDTV signal. We now have that. There are two wireless technologies competing to be the uncompressed video transport champions: WiGig and WirelessHD. Both use the unlicensed spectrum at 60 GHz.

60 GHz IS FAR OUT

Far out is right – 60 GHz is in that region referred to as extremely high frequencies (EHF) or what we also call millimeter waves. The more familiar microwave range runs from 1 GHz to 30 GHz. That part of the spectrum is widely used for radar, satellites, wireless LANs, cellular

systems, and a myriad of other short-range wireless technologies.

So, 60 GHz is inside the millimeter (mm) wave spectrum that covers the 30 GHz to 300 GHz frequency range. That's pretty high. It lies between microwaves and infrared (IR) light if that puts it into perspective for you. That territory is the outer limits of wireless radio.

It has only been within the past decade that we have had transistors and circuitry that could be used to build millimeter wave radios. The latest semiconductor materials and process technologies are now in place, and we are beginning to see some integrated circuits that can work in this range.

A popular worldwide frequency range is the 57 to 64 GHz band available in most countries for unlicensed use. Frequencies in this range are ideal for very high speed serial data that exceeds 1 Gbps. In fact, rates to about 10 Gbps plus are possible.

While such high frequencies are ideal for transmitting high speed data, they do have their drawbacks. One of the most limiting is a very short transmission range. Physics tells us that for a given power level, the range of a radio signal is determined by the signal wavelength. The shorter

the wavelength, the more limited the transmit distance in free space.

Higher power extends the range, of course, but high power (> 1 watt or so) is really hard to come by at these frequencies. Therefore, the typical range is mostly limited to about 10 meters (or roughly 33 feet), and that assumes direct line of sight (LOS) from transmit to receive antenna. Millimeter wave signals are easily reflected and blocked, and absorbed by walls and other objects which further limits the range.

Those designing radios for these frequencies have found that this power and range limitation can be overcome by using phased array antennas. A phased array is a collection of small antennas arranged in an array or matrix. An 8 x 8 array, for example, uses 64 antennas that focus the radio wave into a narrow beam. This focusing has the same effect as raising the actual transmit power.

At lower frequencies, 64 antennas would be an imposing physical object. However, at 60 GHz, that array may only be a fraction of an inch wide and long. Most 60 GHz radios use such an antenna to extend the range, and focus the signal to prevent interference to and from other nearby radios using the same band.

WiGig

WiGig is the trade name of one of the 60 GHz technologies getting lots of attention lately. It is actually now an official IEEE standard called

802.11ad. It uses the same protocol as other Wi-Fi radios, just at the higher frequency. This means that the so-called media access control (MAC) layer of the protocol is compatible with the other popular Wi-Fi standards 802.11a/b/g/n.

The 802.11ad standard divides the 57 to 64 GHz band into four 2.16 GHz channels to support four signals.

The basic modulation scheme is orthogonal frequency division multiplexing (OFDM).

OFDM is a technique that uses multiple (usually hundreds) subcarriers in the allotted band. The high speed data is then divided into many slower bit streams, and each is modulated on to one of the subcarriers. The result is a very robust and reliable signal transfer, despite noise and multipath reflections that are common to millimeter waves. With OFDM, data rates to 7 Gbps can be achieved. A single carrier mode is also supported by the standard, and with this arrangement data rates can be up to 4.6 Gbps. Either mode easily supports the transmission of uncompressed HDTV.

One neat feature of the WiGig standard is its incorporation of a protocol adaption layer (PAL) in the networking structure. This is a software feature that communicates with the MAC layer and permits other standard interfaces to be transmitted. These include USB 2.0 and 3.0, PCI Express, DisplayPort,

Thunderbolt, and HDMI. WiGig permits the cables to go away.

One of the first companies to make WiGig chips is Wilocity. Their single chip 60 GHz transceiver

Since the wavelength defines the antenna size in wireless, antennas are usually half a wavelength or less; or, for 60 GHz, only 2.5 mm or less. That is a big advantage.

incorporates the phased antenna array. The IC is usually packaged along with a standard 802.11n Wi-Fi radio to enable laptops, tablets, and other devices. One available product is the module shown in **Figure 2**. It uses a Qualcomm Atheros' AR9642 802.11n chip and the Wilocity chip to make a 2.4 GHz, 5 GHz, and 60 GHz module that will implement 802.11n and 802.11ad. A newer version also incorporates the more recent 5 GHz 802.11ac standard.

The WiGig capability is just now showing up in some products, but you will be seeing it in laptops and tablets as well as smart TVs, set top boxes (STBs), DVD players, PC docking stations, and any other product that uses uncompressed video.

WirelessHD

WirelessHD is another 60 GHz standard developed by a company

Wavelengths

The term millimeter (mm) waves comes from the wavelength (λ) of a radio signal. Wavelength is the physical length of the actual signal of varying magnetic and electric fields in space. It is also the distance that the wave travels in one cycle of the wave. You can calculate the wavelength with the formula:

$$\lambda = 300/f_{\text{MHz}}$$

Wavelength is expressed in meters.

30 GHz is 30,000 MHz, so its wavelength is: $\lambda = 300/f_{\text{MHz}} = 300/30000 = .01$ meter or 10 mm

300 GHz is 300,000 MHz, so its wavelength is: $\lambda = 300/f_{\text{MHz}} = 300/300000 = .001$ meter or 1 mm

The wavelength at 60 GHz is: $\lambda = 300/f_{\text{MHz}} = 300/60000 = .005$ meter or 5 mm

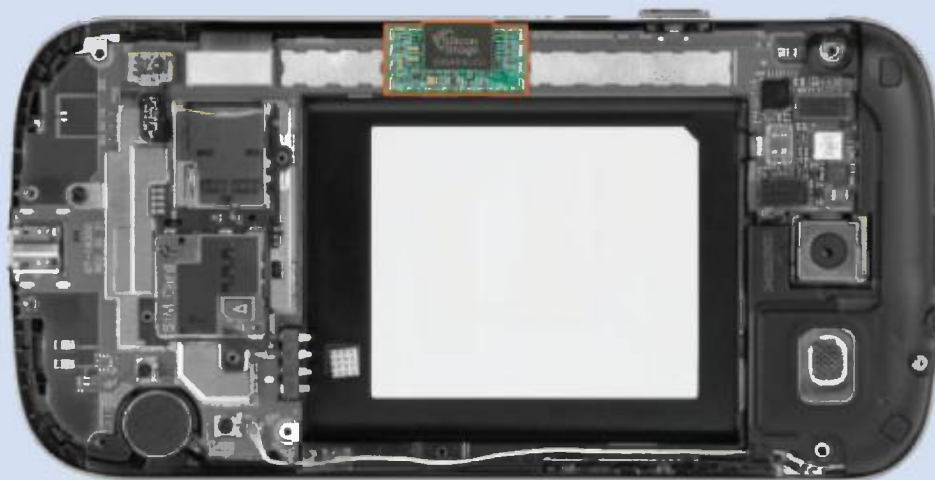
Since the wavelength defines the antenna size in wireless, antennas are usually half a wavelength or less; or, for 60 GHz, only 2.5 mm or less. *That is a big advantage.*

called SiBEAM. SiBEAM became part of the chip provider Silicon Image in 2011. They have further developed and fine-tuned the standard that is now promoted and managed by the WirelessHD Consortium.

The WirelessHD standard also uses 2.16 GHz channels in the 57 to 64 GHz spectrum. Like WiGig, it uses OFDM to achieve high data rates. One version uses 512 subcarriers and QPSK or 16QAM to achieve a data rate to 3.8 Gbps. A slower version uses 128 subcarriers for data rates to 40 Mbps. Silicon Image claims that even faster versions are possible with data rates of 10 to 28 Gbps. This makes the WirelessHD standard attractive for wireless HDMI and other video cable standards like MHL and DVI.

WirelessHD also uses the phased antenna array for gain that produces an effective radiated power (ERP) up to 10 watts. The resulting narrow beam is also steerable to avoid interference and obstacles. Typical maximum range is 10 meters.

The latest WirelessHD chip from Silicon Image is the UltraGig 6400. This transceiver is housed in a 7 x 10 mm chip that is easy to embed in portable gear. **Figure 1** shows its potential incorporation into a smart phone.



■ **FIGURE 1.**

This illustration shows Silicon Image's UltraGig 6400 60 GHz transceiver and how it may be incorporated into a smartphone.

OPTIONAL CHOICES

If you are okay with compressed HDTV video, you do have some other cable replacement choices. For example, the IEEE 802.11n standard with its MIMO feature can crank out a data rate to 450 Mbps in the 2.4 and 5 GHz Wi-Fi bands. Some TV sets, STBs, and DVD players already use this technology.

Another alternative is the more recent 802.11ac standard. It too

uses MIMO, but works in the 5 MHz band with wider bandwidths of 40, 80, and 160 MHz to produce data rates over 1 Gbps, with 900 Mbps being the most common.

One more choice is ultra wideband (UWB). This has been around for a while and works in the 3.1 to 10.6 GHz unlicensed band using OFDM. It can achieve a data rate from 53 up to 480 Mbps. It is already used in laptops, docking



PBP3
PICBASIC PRO™ Compiler

Experimenter - \$49.95
Silver Edition - \$119.95
Gold Edition - \$269.95

The industry-standard BASIC compiler for Microchip PIC® microcontrollers.

Multi-Seat Licensing
for Educational Institutions

Upgrade from
PICBASIC™ Compiler (PBC)

Download a FREE trial version now.

www.PBP3.com

microEngineering Labs, Inc.

www.melabs.com

888-316-1753

PICBASIC and PICBASIC PRO are trademarks of Microchip Technology Inc. in the USA and other countries. PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.

■ FIGURE 2.

This tri-band module is designed to be embedded into laptops and other products. It uses the WiGig 802.11ad standard, as well as a Qualcomm Atheros AR9642 802.11n chip for 2.4 and 5 GHz operation, and the Wilocity 60 GHz chip.



stations, and digital cameras.

A final choice is the Wireless Home Digital Interface (WHDI). It uses an unusual 5 x 4 MIMO system in the 5 GHz band with 40 MHz wide channels to achieve data rates up to 3 Gbps – good enough for uncompressed video. Its big benefit is that it can more easily penetrate walls than 60 GHz signals, and it has a longer range of up to 30 meters.

NV

Resources

If you want more gory details on the 60 GHz technologies, go to the following sources:

Silicon Image

www.siliconimage.com

Wilocity

www.wilocity.com

Wigig Alliance

www.wirelessgigabitalliance.org

Wireless HD Consortium

www.wirelesshd.org

**Need a small and cost-effective device?
Look no further.**



PoKeys

- Smallest and worlds most flexible USB HID or ETHERNET I/O interface
- Keyboard and joystick simulator (USB)
- 55 digital I/O (configurable)
- LCD and LED matrixes driver
- 6 PWM outputs, 26 encoder inputs
- Support up to 10 I2C, up to 27-Wire sensors, up to 7 analog sensors
- Expandable with PoNET
- Free software and web interface (ethernet)



PoScope Mega1+

- Smallest USB 2.0 portable 1MS/s oscilloscope
- Data acquisition of analog and digital signals
- Data recording
- Export to CSV, XLS, PDF and HTML
- Simple usage of advanced features
- Examples for C++, VB, Delphi and LabView
- Free software and updates

PoLabs



Visit us at
www.poscope.com

NUTS AND VOLTS *Free Shipping!*

2012 CD-ROM

Order yours today!

Only \$24.95

800 783-4624

www.nutsvolts.com

NUTS AND VOLTS 2012

Volunteers 2012, No. 1-12

May NOT be posted on a public or corporate internet site.

For personal use only
Copyright 2012-2013
L. Publications, Inc.
All rights reserved.

The Nuts & Volts **WEBSTORE**

GREAT FOR DIYers!

Build Your Own Transistor Radios

by Ronald Quan

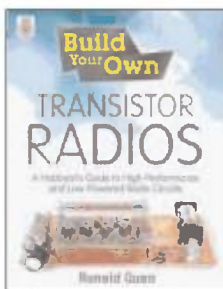
A Hobbyist's Guide to High-Performance and Low-Powered Radio Circuits

Create sophisticated transistor radios that are inexpensive yet highly efficient. Inside this book, it offers complete projects with detailed schematics and insights on how the radios were designed. Learn how to choose components, construct the different types of radios, and troubleshoot your work.

*Paperback, 496 pages

Reg Price \$49.95

Sale Price \$40.95



NEW!

How to Diagnose and Fix Everything Electronic

by Michael Jay Geier

Master the Art of Electronics Repair!

In this hands-on guide, a lifelong electronics repair guru shares his tested techniques and invaluable insights. *How to Diagnose and Fix Everything Electronic* shows you how to repair and extend the life of all kinds of solid-state devices, from modern digital gadgetry to cherished analog products of yesteryear. **About the Author:** Michael Jay Geier began operating a neighborhood electronics repair service at age eight that was profiled in *The Miami News*.

\$24.95



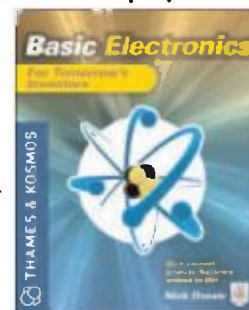
Basic Electronics for Tomorrow's Inventors

by Nick Dossis

Learn all about electronics with hands-on experiments and projects.

This great book features a variety of simple, solder-free projects, including an LED reading light, electronic security keypad, IR target practice game, a real working telephone, temperature and moisture sensors, spy gadgets, and other neat stuff. Best of all, these experiments require only plug-and-play "breadboards" and other commonly available parts.

\$19.95

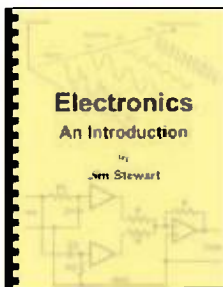


Electronics An Introduction

by Jim Stewart

This book is designed as an in-depth introduction to important concepts in electronics. While electronics can be highly mathematical, this text is not about calculations. It is about how electronic equipment is able to extract, process, and present information held in electrical signals. If you are in — or studying to be in — a profession that requires the use of electronic equipment, then this book will provide the insight necessary to use such equipment effectively.

\$33.95*



Build Your Own Electronics Workshop

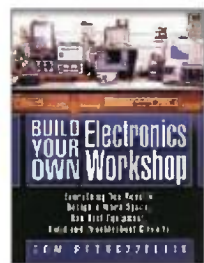
by Thomas Petruzzellis

BUILD YOUR OWN DREAM ELECTRONICS LAB!

This value-packed resource provides everything needed to put together a fully functioning home electronics workshop! From finding space to stocking it with components to putting the shop into action — building, testing, and troubleshooting systems. This great book has it all! And the best part is, it shows you how to build many pieces of equipment yourself and save money, big time!

Reg Price \$29.95

Sale Price \$26.95



Beginner's Guide to ... Programming the PIC24/dsPIC33

by Thomas Kibalo

Kibalo takes you step by step through the fundamentals of programming the PIC24H which can equally be applied to the dsPIC33. His clear explanation of the inner workings make learning the PIC24H/dsPIC33

16-bit architecture easy. His code examples demonstrate how to perform the functions most applications require. The hardware is shown in a simple breadboard setup so even a beginner can build it, along with very few extra components needed.

\$39.95*



HTML: A Beginner's Guide

by Wendy Willard

Create highly functional, impressive websites in no time. Fully updated and revised, *HTML: A Beginner's Guide*, Fourth Edition explains how to structure a page, place images, format text, create links, add color, work with multimedia, and use forms. You'll also go beyond the basics and learn how to save your own web graphics, use Cascading Style Sheets (CSS), create dynamic web content with basic JavaScript, and upload your site to the web. By the end of the book, you'll be able to build custom websites using the latest HTML techniques.

\$29.95



Programming PICs in Basic

by Chuck Hellebuyck

If you wanted to learn how to program microcontrollers, then you've found the right book! Microchip PIC microcontrollers are being designed into electronics throughout the world and none is more popular than the eight-pin version. Now the home hobbyist can create projects with these little microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software language. Chuck Hellebuyck introduces how to use this development setup to build useful projects with an eight-pin PIC12F683 microcontroller. **\$14.95**

Master and Command C for PIC MCUs

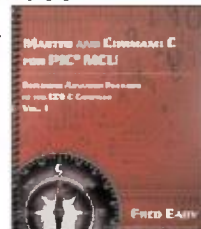
by Fred Eady

Master and Command C for PIC MCU, Volume 1

aims to help readers get the most out of the Custom Computer Services C compiler for PIC microcontrollers.

The author describes some basic compiler operations that will help programmers particularly those new to the craft create solid code that lends itself to easy debugging and testing. As Eady notes in his preface, a single built-in CCS compiler call (output_bit) can serve as a basic aid to let programmers know about the "health" of their PIC code.

\$14.95



Order online @ www.store.nutsvolts.com
Or CALL 1-800-783-4624 today!

BOOK & KIT COMBOS

Proto Buddy Book & Kit Combo



Only
\$59.95

For complete details visit our webstore @ www.nutsvolts.com

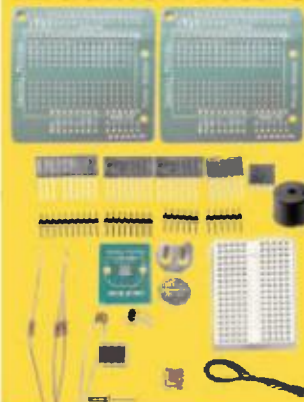
Beginner's Guide Complete Combo!



Combo Price
\$229.95

For complete details, visit our webstore @ www.nutsvolts.com.

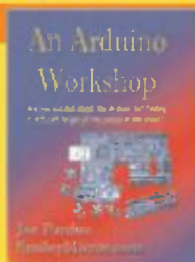
Follow Smiley's Workshop: Fritzing the Arduino Arduino Proto Shield Alarm Clock Kit



Learn to do a complete Arduino
project development cycle.

Begin by building a Real Time Clock
on a breadboard shield. Next, transfer
the design to a PCB shield. Finally,
learn how to produce a custom printed
circuitboard using Fritzing.

\$24.95



Puzzled by the Arduino?

An Arduino Workshop \$44.95

Projects Kit \$84.95

Book+Kit Combo \$124.95



From Smiley's Workshop

CD-ROM SPECIAL

Nuts & Volts
9 CD-ROMs
& Hat Special!
That's 108 issues.
Complete with supporting
code and media files.



Only \$189.95
or \$24.95 each.



The Nuts & Volts Pocket Ref

All the info you need at your fingertips!

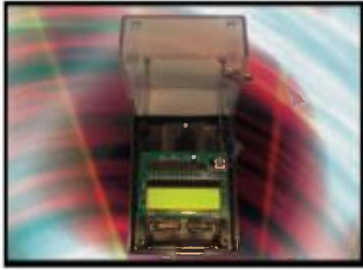
This great little book is a concise
all-purpose reference featuring
hundreds of tables, maps,
formulas, constants &
conversions.
AND it still fits in
your shirt pocket!

Only \$12.95

Visit <http://store.nutsvolts.com> or call (800) 783-4624

PROJECTS

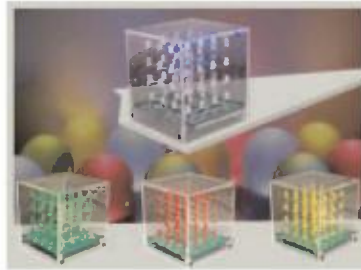
Gravimeter Kit



Gravity is that mysterious force that is hard to feel when it changes. This kit is an inexpensive way of measuring the changes in gravity due to a wide variety of circumstances, such as the position of the moon or even the density of the terrain you're on.

Subscriber's Price \$61.95
Non-Subscriber's Price \$64.95

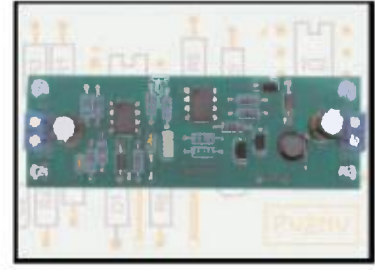
3D LED Cube Kit



This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes. Colors available: Green, Red, Yellow & Blue

Subscriber's Price \$57.95
Non-Subscriber's Price \$59.95

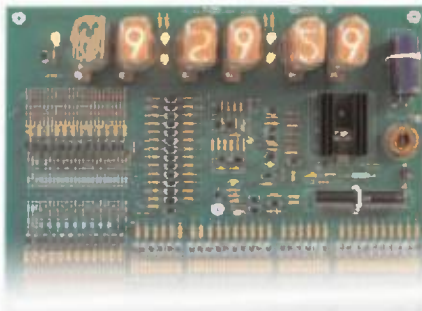
Inverter DC-DC Converter Kit



If you need +12V and -12V, but all that's available is +12V, then this project is for you. The inverted DC-DC converter gives you -V out when you put +V in, and works over a range of voltages without adjustment. It's an easy to build voltage mirror that can supply over 100 mA without a significant drop in voltage.

Subscriber's Price \$29.95
Non-Subscriber's Price \$32.95

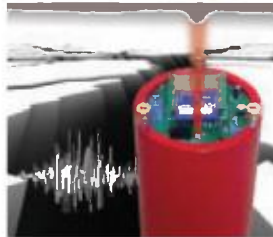
Neon Transistor Clock Kit



Add HIGH VOLTAGE to your clock!
 This is a Nixie Tube display version of the Transistor Clock. It uses only discrete components — no integrated circuits. For more info, see the April 2012 issue.

Subscriber's Price \$245.95
Non-Subscriber's Price \$249.95

Seismograph Kit



As seen in the May 2012 issue.
 Now you can record your own shaking, rattling, and rolling.

The Poor Man's Seismograph is a great project /device to record any movement in an area where you normally shouldn't have any. The kit includes everything needed to build the seismograph. All you need is your PC, SD card, and to download the free software to view the seismic event graph.

Subscriber's Price \$79.95
Non-Subscriber's Price \$84.95

Magic Box Kit



As seen on the April 2007 cover.

This unique DIY construction project blends electronics technology with carefully planned handcraftsmanship. This clever trick has the observer remove one of six pawns while you are out of the room and upon re-entering you indicate the missing pawn without ever opening the box.

Subscriber's Price \$39.95
Non-Subscriber's Price \$45.95

FOR BEGINNER GEEKS!

The Learning Lab 1 Fundamental Concepts



\$59.95

The Learning Lab 2 Basic Digital Concepts and Op-Amps



\$49.95

The Learning Lab 3 Basic Electronics: Oscillators and Amplifiers



\$39.95

The labs in this series — from GSS Tech Ed — show simple and interesting experiments and lessons, all done on a solderless circuit board.

As you do each experiment, you learn how basic components work in a circuit, and continue to build your arsenal of knowledge with each successive experiment.

For more info and a promotional video, please visit our webstore.

CLASSIFIEDS

KITS/PLANS

Electronic Kits

- Fun and Educational
- Discount White and Blue LEDs
- Quantity Discounts

www.bakatronics.com

ENGINEERING/DESIGN SVCS

FLEXREV China Direct Manufacturing

Turn Key or Subassemblies
Components, Enclosures, Keypads

\$0.007 per soldering point
2 layer PCBs \$0.14/in²
Small jobs welcome

FLEXREV.COM

www.nutsvolts.com

ROBOTICS

Ultrasonic Distance Sensing Made EZ

HRUSB-MaxSonar[®]-EZ™

- Multi-Sensor operation
- USB interface
- Easy integration
- 1 mm resolution
- Calibrated beam pattern
- Starting at \$42.95



HRXL-MaxSonar[®]-WR™

- High noise tolerance
- IP67 rated
- 1 mm resolution
- Multi-Sensor operation
- Calibrated beam pattern
- Starting at \$109.95



I2CXL-MaxSonar[®]-EZ™

- Great for UAV's and robotics
- Incredible noise immunity
- I2C interface
- 1 cm resolution
- Automatic calibration
- Starting at \$38.95



www.maxbotix.com

QKITS.COM

AVR Treasure Finder Robot kit detects metal, stops and beeps!



\$46.95
USD

1-888-GO 4 KITS

COMPONENTS



RF PARTS™ CO.

Tubes, Transistors, Power Components
Email: rfp@rfparts.com • Web: www.rfparts.com

800-737-2787 Fax 888-744-1943

uM-FPU64

64-bit Floating Point Coprocessor



DIP-28, SOIC-28, TQFP-44

64-bit and 32-bit IEEE 754 compatible
SPI or I2C interface, 3.3V Supply, 5V Tolerant I/O
Extensive floating point support, FFT, matrix operations
GPS input, local device support, Flash functions, IDE

Robotics Navigation Sensor Modules Embedded Systems
Additional products...
uM-FPU V3.1 uM-PWM1 32-bit FPU Servo Coprocessor

www.micromegacorp.com

SURPLUS

SURPLUS ELECTRONIC PARTS & ACCESSORIES



Over 20,000
Items in Stock

Cables
Connectors
Displays
Fans

Hardware
LEDs
Motors
Potentiometers

Relays
Semiconductors
Service Manuals
Speakers
Switches
Test Equipment
Tools
VCR Parts

Surplus Material Components
SMC ELECTRONICS
www.smcelectronics.com

No Minimum Order.
Credit Cards and PAYPAL Accepted.
Flat \$4.95 per order USA Shipping.

HARDWARE WANTED

DEC EQUIPMENT WANTED!!!

Digital Equipment Corp.
and compatibles.
Buy - Sell - Trade

CALL KEYWAYS 937-847-2300
or email buyer@keyways.com

JOB OPENING

Instrument Shop Supervisor Needed

Department of Engineering
Initial Deadline: 11:59pm, 3/7/13
For qualifications/application info, see Job
ID #30386 at www.asu.edu/asujobs/

ASU is an Equal Opportunity Institution

ENERGY SOLUTIONS



LIGHTING



OSCILLOSCOPES

LOW COST, VERSATILE 30MHz SCOPE



ONLY \$299

WWW.SAELIG.COM

www.servo
magazine.com

NUTS AND VOLTS

2012 CD-ROM

Order yours today!

NUTS AND VOLTS 2012
Volume 35, No. 1-62

Only \$24.95

Free Shipping

www.nutsvolts.com

Always wanted to be a Rock Star, but music just wasn't your thing?

OK, so maybe it turns out you're better at soldering up an overdrive circuit for an old Fender guitar amp than playing the 4-1/2 minute guitar solo from Freebird.

Well, your geekish tendencies could *still* propel you to stardom and satisfy your need for fame and fortune (*mostly fame*) in the pages of *Nuts & Volts*!

Did you know that much of what you read in Nuts & Volts comes from readers just like you?

Why not share your awesome project or story with our tech-savvy readers. If we like what we see, you could be famous!

Well, you'll be a star to the Nuts & Volts audience anyway ... or your mom ... or kids ... or dog.

Anyway, we're always looking for:

- Projects Built From Scratch (*Analog or Microcontroller based*)
- Electronics Tutorials (*How Tos*)
- Programming Shortcuts, Tips
- Electronics Fundamentals
 - Product Reviews
 - Tips and Tricks
 - Hardware Hacks
 - New Technologies
- Cool Circuit Designs

If you've been in the electronics hobby for more than a week, you probably have something to share. **Run it past our editors!** If it's not quite ready for prime time, we'll let you down easy, *not like those mean American Idol judges*. We **CAN** promise *one thing*, you won't get rich, but you'll have the respect and admiration of your peers. (*er... two things?*)

If this sounds like a dream come true, send an email to editor@nutsvolts.com and we'll send you back a copy of our new updated writer's guidelines, with all you need to know to get started.

Continued from page 41

and data logging for traces, data points, or screen images (with storage for 10,000 entries per file triggered from a key press, internal timer, external trigger, or the limits comparator).

Traces or complete screen images can be saved to files, and compensation tables for antennae or other external transducers can be created and loaded. USB host and device connectors allow the use of USB Flash drives or direct connection to a PC.

Control of the PSA analyzers is by finger-operated touch-screen soft keys in a hierarchical menu system that gives rapid access to menu functions. All functions can be operated using just the hard keys, if desired.

The ruggedized casing incorporates a rubber protection buffer, a bench stand, and screen protection. For bench-top use, the instrument can be operated continuously from its AC charger. For portable use, the battery life of eight hours can be further extended by selecting auto-off which turns the instrument off (retaining all data) after a selectable delay of five to 60 minutes from the last key press.

The PSA Series II provides a cost-effective and highly portable tool, placing a spectrum analyzer in areas that are difficult to access with bench top instruments. The PSA analyzers weigh 560 grms (20 oz) and are 9.2 cm (3.6") wide, making them suitable for any portable RF service kit.

For more information, contact:

Saelig

Web: www.saelig.com

If you have a new product that you would like us to run in our New Products section, please email a short description (300-500 words) and a photo of your product to:

**newproducts@
nutsvolts.com**

ALUMINUM HUB MOUNTS

ServoCity's 0.770 hub pattern has now been incorporated into their new line of aluminum hub mounts. Integrating both 6-32 tapped and .140" through holes (in the hub pattern), and 6-32 tapped base mounting holes provides for countless attachment possibilities.

Constructed from 6061-T6 aluminum for superior strength, these mounts won't easily flex or bow. The 1/2" bore makes it simple to incorporate shafting and tubing into a project design. Starting at \$4.49, they are offered in many styles and configurations.



traction and quiet running. The 3" and the 5" wheels are offered in gray, blue, and orange; the heavy-duty and foam wheels are currently offered in black.

All wheels are compatible with Servocity's line of aluminum hubs and adaptors. Starting at \$2.59, these wheels can be used in applications such as designing a mobile robot, a camera slider, or a wheeled platform.



For more information, contact:

ServoCity

Web: www.servocity.com

HEAVY DUTY ROBOT WHEELS

ServoCity also recently added a new line of robot wheels to their expanding product selection. The hard surface wheels offer combined high load capacity with low rolling resistance. The foam wheels work well for applications that require high



Sabertooth 2x60

**Dual 60A Continuous
Brushed Motor Controller**

Specifications:

- 60A Continuous, 120A peak current
- 6V - 33.8V input voltage range (8s lipo max)
- Analog, RC (PWM), and Serial Interface
- Overcurrent, overvoltage, undervoltage, and overheat protection built in
- Fan cooling keeps the controller cool
- Heavy duty terminals take up to 2 gauge wire
- 1 year warranty on every Sabertooth
- DEScribe Software Suite Compatible

**ONLY
\$189.99**



• www.dimensionengineering.com/Sabertooth2x60.htm • 899 Moe Dr. #21 Akron, OH 44310 • 330 634-1430

>>> QUESTIONS

I have two Fuji Film digital cameras (FinePix 2650 and FinePix A210). Each came equipped with a 16 MB xD-Picture card (image memory card) which is a semiconductor memory (NAND-type Flash memory) to record digital image data.



I have a micro-SD adapter (orange) with a 1 GB card which fits EXACTLY into the xD-Picture card slot. Even though I formatted the card INSIDE the cameras several times, I'm still getting "Card Error - Card Not Initialized" messages on the camera's LCD monitors.

Why can't I use this 1 GB micro-SD adapter with these digital cameras? Under what circumstances can I use this type of adapter?

#4131 Don
via email

I have tried connecting a color security camera to a Maxent (MX-42 HPM20) plasma TV without great results. Either the color hues are bad or it goes from color to monochrome and back again, all under great lighting conditions. The camera's composite out goes directly into the TV's AV1 or AV2 input. The TV's other inputs work fine and the camera looks great on another monitor. What's going on here?

#4132 Radio Mike
via email

I'm trying to use an SDX01G2 pressure sensor to log CPAP breathing patterns. The circuit uses an LM324N op-amp, an ADC0831 CCN eight-bit A/D converter, and a BASIC Stamp 2

processor. The problem is the sensitivity. The output goes from 0 up to 191, and then back to 0. I have tried several op-amp configurations with little success. I am a newbie, and could sure use some help.

#4133 Henry
Visalia, CA

I'm trying to replace the four-port IC mic with a two-port mic since the mic IC is very expensive. When I try to do it, I don't know where the continuity of the mic port is — especially in the Samsung mobile.

#4134 Sankar Rameswaran
Tuticorin / TAMIL NADU

>>> ANSWERS

[#1132 - January 2013]
iPhone / X10 Control

Is there a way to operate my home X10 devices using my iPhone through my Wi-Fi network (preferred) or Bluetooth?

Try the Insteon hub. It gives you smartphone and web access to your Insteon and X-10 modules at home or away, plus text alerts from sensors, if you install them. It's a free app and there are no fees. Or, use the Smartlinc unit — an older version of the above — except no text alerts.

Mike Carland
Santa Clarita, CA

[#2133 - February 2013]
Low Audio on Dish Receiver

I'm looking for a schematic for a Dish Network satellite receiver model DISH311. The audio is too low and I'm wondering if there was any adjustment that could be made to increase it.

Check the menu settings for the Volume Control options.

Robert
Fresno, CA

[#3132 - March 2013]

I lost my hearing at a very young age and have always struggled with listening to the audio out phones, computers, MP3 players, etc. When I have my hearing aids in and try to listen, it's actually worse!

I would like to build an audio amplifier with a programmable equalizer, to be connected to the 3.5 mm audio output jack. I will connect +5V separately. The device will be connected to a PC temporarily through the USB jack for programming, and then be removed.

Here are my questions:

Is a one stage amplifier enough?

Is it better to amplify first then equalize, or equalize then amplify?

Who makes the best audio chips?

What software is needed to program the EQ?

Any suggested readings, websites, software, forums, etc.

Specifications:

Send all questions and answers by email to forum@nutsvolts.com or via the online form at www.nutsvolts.com/tech-forum

All questions AND answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. All submissions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted

by readers and **NO GUARANTEES WHATSOEVER** are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals.

Always use common sense and good judgment!

>>>YOUR ELECTRONICS QUESTIONS ANSWERED HERE BY N&V READERS

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

Easy to build and program using a chip like Monolithic Linear integrated circuit LA3600; www.electronic-lab.com/blog/?tag=equalizer.

Input voltage: 5V

Gain: ?

Equalizer: Seven band or better

Output: 3.5 mm

And for the hard way, using a DSP or FPGA to build an advanced equalizer/gain headset that could be used in any computer, phone, or MP3 player!

If you're generally new to audio processing, I would stick with analog for a while. It's not as cool or widely configurable as a DSP, but it usually does a pretty good job when done right. For your project, you'll probably want:

1. Input gain
2. Three-band or more *parametric* EQ
3. Output gain
4. Power amp

I included two gain stages so that the EQ's working level can be adjusted independently of both the input and output levels. This allows you to over-power the analog noise in that stage without fear of clipping and still get the output level that you want. (Professional analog mixing consoles are basically that on steroids.) The gain stages are the easiest parts of the circuit: Use a dual-gang linear potentiometer (for stereo, single-gang for mono) with an extra resistor between the center-tap and ground. This loading resistor should be roughly 10% to 15% of the pot's resistance, and makes a better audio taper than most audio pots.

The power amp can be a single IC for small speakers or headphones.

They're sometimes called power op-amps. The EQ is a little different than I think you had in mind. I think you're imagining a many-band *graphic* EQ, which is easier to understand at first glance, but I think a more useful tool for you would be a *parametric* EQ. Parametric means that you can adjust the frequency of each band, as well as the gain. For example, if you have a problem at, say, 3 kHz, and you have a graphic EQ with adjacent bands at 2.5 kHz and 3.5 kHz, it's going to be hard to adjust that satisfactorily. (This is why professional graphic EQ's have up to 31 bands. Even so, they still don't work well for everything.) With a parametric EQ, you could adjust a frequency knob to put one band exactly where you need it for each problem area. A semi-parametric EQ is easier to build than a full-parametric, and just has the two knobs per band — frequency and gain. A full-parametric EQ has a third knob that adjusts the width of each band. Despite missing a knob though, the semi-parametric version is usually sufficient because it automatically adjusts the bandwidth as a function of gain. Large gain adjustments typically make a narrow spike or notch, while smaller adjustments make a wide hill or valley.

This website (www.sound.au.com) is a great resource for analog audio projects. It has circuits for almost anything you might want to do to your sound,

with schematics, detailed descriptions, and usually a PCB for purchase. One minor sticking point for you might be that these circuits are designed for $\pm 15\text{V}$ supplies (30V total, with signals referenced to a center-tap). If all you have is 5V, you might be able to make a switching power supply that provides that, or you might be able to adapt the circuits to run on 5V with a higher noise floor. (Actually, the noise level stays the same, but you're using a smaller signal.)

Aaron Duerksen
via email

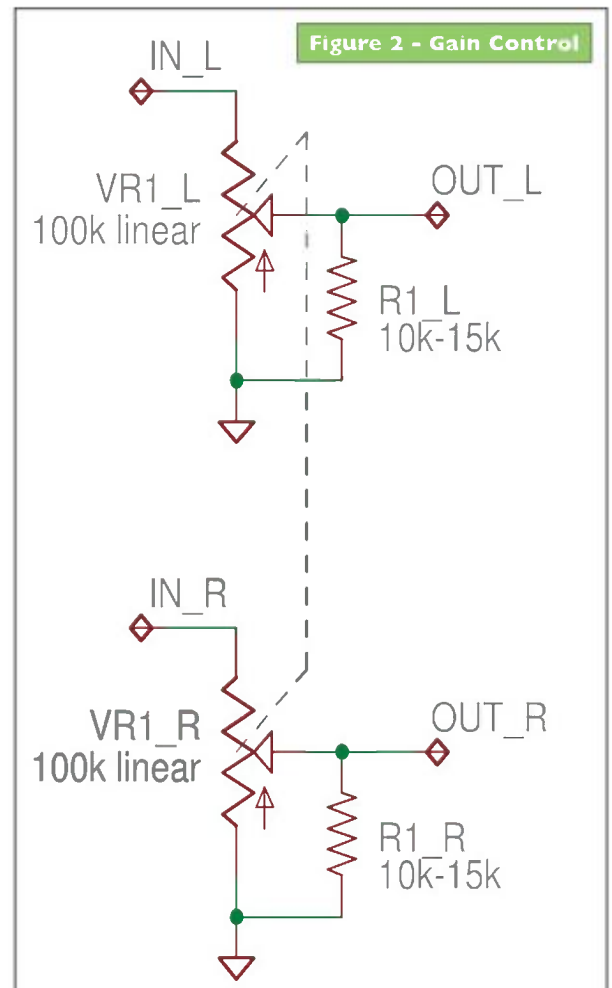
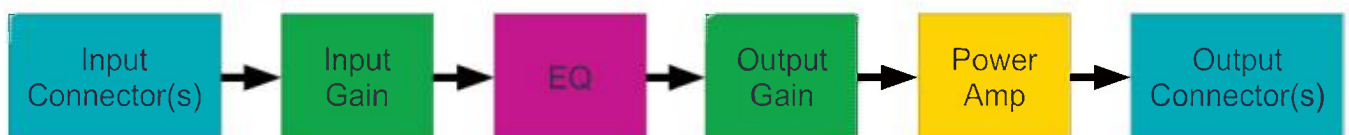


Figure 1 - Block Diagram



AMATEUR RADIO AND TV

Ramsey Electronics82-83

BATTERIES/CHARGERS

Cunard Associates27

Handheld Scientific Inc.41

BUYING ELECTRONIC SURPLUS

Jaycar Electronics7

Weirdstuff Warehouse27

CCD CAMERAS/VIDEO

Ramsey Electronics82-83

CIRCUIT BOARDS

AP Circuits51

Cunard Associates27

Digilent4

Dimension Engineering.....77

ExpressPCB9

Front Panel Express LLC52

MCM Electronics13

PCB Pool52

COMPONENTS

Cana Kit Corp.53

MCM Electronics13

SDP/SI27

COMPUTER

Hardware

EZSBC52

Weirdstuff Warehouse27

Microcontrollers / I/O Boards

Abacom Technologies51

Images Co.25

microEngineering Labs70

MikroElektronika3

Pololu Robotics & Electronics.....29

Software

MakerPlot.com27

DESIGN/ENGINEERING/

REPAIR SERVICES

Cana Kit Corp.53

ExpressPCB9

Front Panel Express LLC52

Images Co.25

PCB Pool52

DIGITAL

OSCILLOSCOPES

Rigol Technologies8

DRIVE COMPONENT CATALOGS

SDP/SI27

EDUCATION

Command Productions17

Digilent4

NKC Electronics27

Poscope71

EMBEDDED TOOLS

NetBurner2

ENCLOSURES

Front Panel Express LLC52

EVENTS

Maker Faire81

RoboGames28

GRAPHICAL PLOTTING

SOFTWARE

MakerPlot.com27

GUI FOR

MICROCONTROLLERS

MakerPlot.com27

KITS & PLANS

Cana Kit Corp.53

Jaycar Electronics7

NetBurner2

NKC Electronics27

QKITS27

Ramsey Electronics82-83

MISC./SURPLUS

All Electronics Corp.42

Front Panel Express LLC52

Weirdstuff Warehouse27

PROGRAMMERS

microEngineering Labs70

MikroElektronika3

RF TRANSMITTERS/

RECEIVERS

Abacom Technologies51

ROBOTICS

Digilent4

EZSBC52

Lemos International Co.25

Pololu Robotics & Electronics.....29

SDP/SI27

SATELLITE

Lemos International Co.25

SOLDERING STATIONS AND IRONS

HAKKO19

TEST EQUIPMENT

Dimension Engineering.....77

Images Co.25

Jaycar Electronics7

MCM Electronics13

NKC Electronics27

Pico TechnologyBack Cover

Poscope71

Rigol Technologies8

TOOLS

HAKKO19

Handheld Scientific Inc.41

MikroElektronika3

NetBurner2

PanaVise52

Poscope71

VIDEO/DISPLAY

MODULES

4D Systems5

WAVEFORM

GENERATORS

Rigol Technologies8

WIRELESS PRODUCTS

Handheld Scientific Inc.41

4D Systems5

Abacom Technologies51

All Electronics Corp.42

AP Circuits51

Cana Kit Corp.53

Command Productions17

Cunard Associates27

Digilent4

Dimension Engineering.....77

ExpressPCB9

EZSBC52

Front Panel Express LLC ...52

HAKKO19

Handheld Scientific Inc.41

Images Co.25

Jaycar Electronics.....7

Lemos International Co.25

Maker Faire81

MakerPlot.com27

MCM Electronics13

microEngineering Labs70

MikroElektronika3

NetBurner2

NKC Electronics27

PanaVise52

PCB Pool52

Pico Technology ...Back Cover

Pololu Robotics &

Electronics.....29

Poscope71

QKITS27

Ramsey Electronics82-83

Rigol Technologies8

RoboGames28

SDP/SI27

Weirdstuff Warehouse27

Maker Faire®

8th Annual Bay Area
SAN MATEO EVENT CENTER

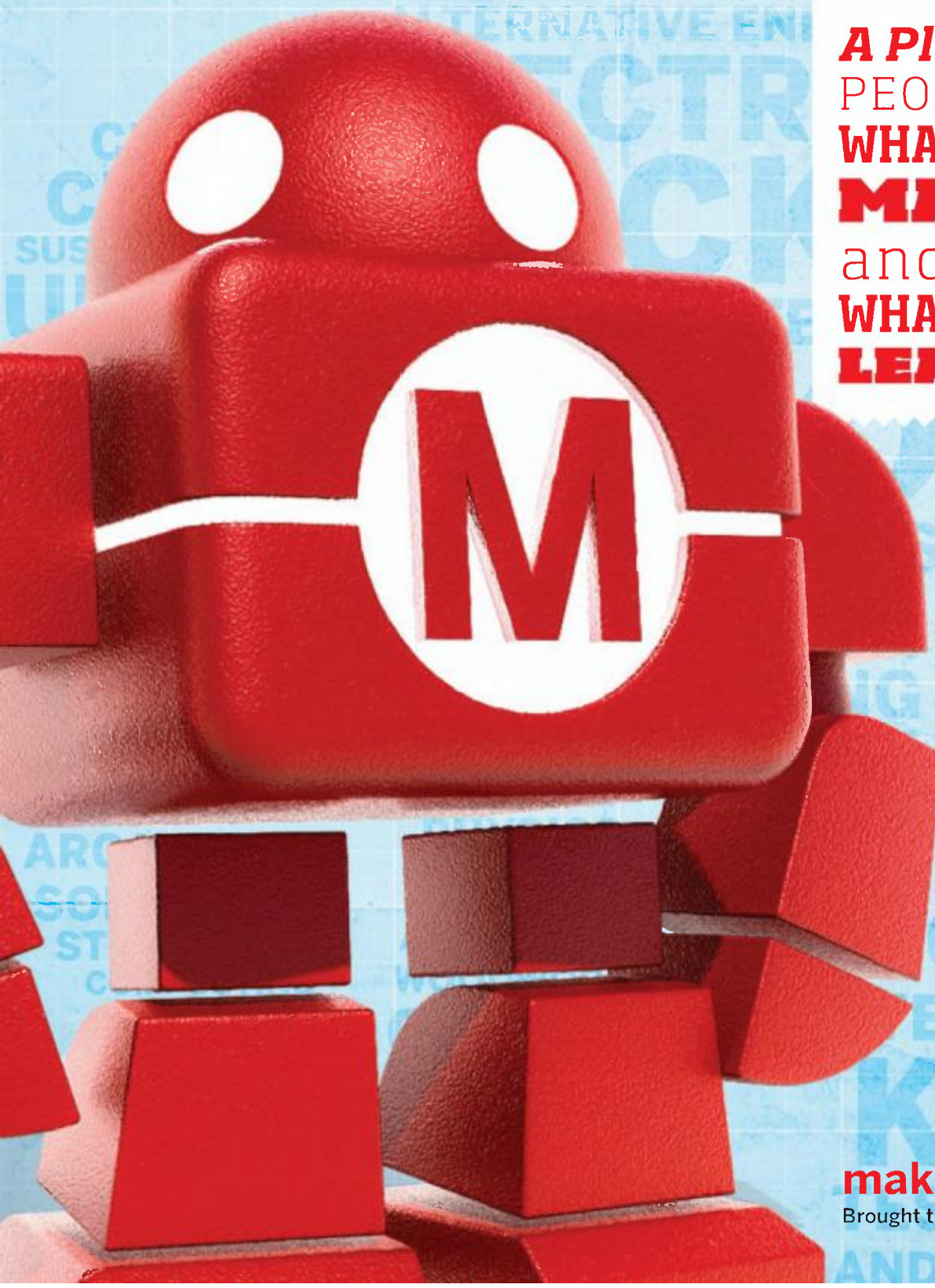
MAY 18–19



A Family-Friendly
SHOWCASE OF
**INVENTION
CREATIVITY &
RESOURCEFULNESS**



A Place Where
PEOPLE SHOW
WHAT THEY ARE
MAKING
and SHARE
WHAT THEY ARE
LEARNING



THE
GREATEST
SHOW
(and TELL)
ON EARTH

makerfaire.com

Brought to you by **MAKE** magazine



Stereo Audio Platform Gain Controller

- ✓ Stereo audio processing while preserving audio dynamics!
- ✓ True stereo control keeps virtual sonic source location intact!
- ✓ Auto-bypass restores original levels when power is turned off!
- ✓ Unbalanced RCA and 3.5mm stereo input and output line level jacks!
- ✓ Built-in bar graph indication of signal level with display mute!

The SGC1 is one of our latest kits, and provides a great solution to the age-old problem: how can we easily correct inconsistent audio levels without negatively affecting the dynamics of the audio signal? The SGC1 circuit implements a principle known as the "Platform Gain Principle," which was originally developed by CBS Labs (what we now know as CBS in the TV and radio world) to allow transmitted audio levels to be automatically adjusted to keep them within a desired range.

Think of it like an audio engineer, constantly adjusting the output level in order to limit highs that would be too loud while boosting lower levels so that they can still be heard. You may think "oh, this is just another limiter/compressor!" Not so! Here's the real trick: keeping the full dynamic range ratio of the output signal the same as the original input - something the typical limiter/compressor can only dream of doing! The SGC1 can be placed in just about any standard analog stereo line level audio circuit (the red and white RCA connectors or the mini-phone connector) to keep the audio level within the desired range. It's also the perfect addition to any of our hobby kit transmitters, allowing you to match levels between different audio sources while keeping lows audible and preventing the highs from overdriving.

The SGC1 makes a great addition to any audio system where you need to keep levels from different sources under control, but still make sure they all sound great! In addition to its useful basic function and great audio performance, the SGC1 also boasts a front panel LED meter to give an indication of the relative level of the input signal, plus a level control (also on the front panel) that allows you to adjust the controller to the min/max center point of your desired level range. And yes, it is a **Stereo Gain Controller!** Meaning that the levels of both the left and right channels are monitored and adjusted equally, thereby maintaining the relative virtual position of things like instruments, singers and speakers! The entire unit is housed in a slim attractive black textured aluminum case that is sure to complement your studio or home theatre. If you're looking for perfect audio levels, hire a broadcast audio engineer, but if that doesn't fit your budget, the SGC1 is the next best thing! Includes 15VDC world-wide power adapter.

SGC1 Stereo Audio Platform Gain Controller Kit

\$179.95

FM Stereo Transmitter Kit

The FM25 has been the hobbyist's standard for FM synthesized stereo transmitters for more than two decades! Just plug in the stereo left/right audio from your MP3 player, CD player, or computer, and broadcast it on any frequency in the standard FM broadcast band.

The sound quality and stereo separation of this little transmitter will keep the pickiest audiophile happy. The FM25B features a PIC microprocessor for easy frequency programming through board mounted DIP switches. The transmit frequency is PLL controlled for unparalleled stability making frequency drift a thing of the past. The RF output level is adjustable from 5uW to 25 mW. Includes power supply & stereo patch cable.

FM25B FM Stereo Transmitter Kit

\$139.95

Voice Activated Switch

Voice activated (VOX) provides a switched output when it hears a sound. Great for a hands free PTT switch or to turn on a recorder or light! Directly switches relays or low voltage loads up to 100mA. Runs on 6-12 VDC.

VS1 Voice Switch Kit

\$9.95

Passive Aircraft Monitor

The hit of the decade! Our patented receiver hears the entire aircraft band without any tuning! Passive design has no LO, therefore can be used on board aircraft! Perfect for airshows, hears the active traffic as it happens! Available kit or factory assembled.

ABM1 Passive Aircraft Receiver Kit

\$89.95

Laser Trip Sensor Alarm

True laser protects over 500 yards! At last within the reach of the hobbyist, this neat kit uses a standard laser pointer (included) to provide both audible and visual alert of a broken path. 5A relay makes it simple to interface! Breakaway board to separate sections.

LTS1 Laser Trip Sensor Alarm Kit

\$29.95

Digital Voice Changer

This voice changer kit is a riot! Just like the expensive units you hear the DJ's use, it changes your voice with a multitude of effects! Features a built-in microphone and both a speaker and line output. Runs on a standard 9V battery, not included.

MK171 Digital Voice Changer Kit

\$14.95

Pink Noise Generator

Pink noise is a great tool for testing and equalizing rooms. Unlike white noise, which has equal power per Hz, pink noise is filtered to give equal power per octave or per 1/3 octave, making it ideal for acoustic equalization! Can also be modified for white noise. Runs on 12VDC.

K4301 Pink Noise Generator Kit

\$19.95

Logic Interface Module

Interface your digital output to the real world with an on-board SPDT relay rated at 240V at 10A! It takes a digital low (.5VDC or less) or a high (+1 to +12VDC) and provides your choice of an active low or high closure! It's that simple! Runs on 12VDC at 60mA.

RI1 Logic Interface Kit

\$17.95

4-Channel USB Relay Control

This professional quality USB relay controller allows computer controlled switching of external devices, plus full bi-directional communication with the external world using the USB port of your computer!

The controller features four onboard relay outputs with a current rating of 10A each. Also onboard is a 6-channel Input/Output interface, with each channel individually configurable as Digital Input, Digital Output, Analog Input (10-bit Resolution). In Digital Input/Output modes, each channel can support a TTL compatible or ST input or a 5V output signal. In Analog Input mode, each channel can convert a voltage of between 0 to 5V into a 10-bit digital representation.

UK1104 4-Ch USB Relay Interface Kit

\$59.95

Ultimate 555 Timers

This new series builds on the classic UT5 kit, but takes it to a whole new level! You can configure it on the fly with easy-to-use jumper settings, drive relays, and directly interface all timer functions with onboard controls or external signals.

All connections are easily made through terminal blocks. Plus, we've replaced the ceramic capacitor of other timer kits with a Mylar capacitor which keeps your timings stable over a much wider range of voltages! Available in through hole or surface mount versions! Visit www.ramseykits.com for version details.

UT5A Through Hole 555 Timer/Osc Kit \$21.95
UT5AS SMT 555 Timer/Osc Kit \$29.95

USB PIC Programmer

Finally, a compact USB PIC Programmer with a 20 pin ZIF socket for easy programming of most Microchip PIC Flash devices that does not require low voltage programming. Plus it uses USB, therefore no more RS232 compatibility blues!

CK1301 USB PIC Programmer Kit \$34.95

Doppler Direction Finder

Track down jammers and hidden transmitters with ease! 22.5 degree bearing indicator with adjustable damping, phase inversion, scan and more. Includes 5 piece antenna kit. Runs on 12VDC vehicle or battery power.

DDF1 Doppler Direction Finder Kit \$169.95

Air Blasting Ion Generator

Generates negative ions along with a hefty blast of fresh air, all without any noise! The steady state DC voltage generates 7.5kV DC negative at 400uA, and that's LOTS of ions! Includes 7 wind tubes for max air! Runs on 12-15VDC.

IG7 Ion Generator Kit

\$64.95

HV Plasma Generator

Generate 2" sparks to a handheld screwdriver! Light fluorescent tubes without wires! This plasma generator creates up to 25kV at 20kHz from a solid state circuit! Build plasma bulbs from regular bulbs and more! Runs on 16VAC or 5-24VDC.

PG13 HV Plasma Generator Kit

\$64.95

Speedy Speed Radar Gun

Our famous Speedy radar gun teaches you doppler effect the fun way! Digital readout displays in MPH, KPH, or FPS. You supply two coffee cans! Runs on 12VDC or our AC121 supply.

SG7 Speed Radar Gun Kit

\$74.95

Broadband RF Preamp

Need to "perk-up" your counter or other equipment to read weak signals? This preamp has low noise and yet provides 25dB gain from 1MHz to well over 1GHz. Output can reach 100mW! Runs on 12 volts AC or DC or the included 110VAC PS. Assmb.

PR2 Broadband RF Preamp

\$69.95

Van de Graaff Generators

Create your own lightning with these time tested student favorites! Produces low current static currents that can be "shocking" but perfectly safe! Draw sparks to a screwdriver, grab hold and watch your hair stand straight up! Two models produce from 200kV to 400kV!

VG Van de Graaff Generators from \$139.95



Get the brand new 2013 Ramsey Hobby Catalog today! Filled with some of the neatest goodies around! Order yours today... Or download the PDF at www.ramseykits.com/catalog!

Follow Us and SAVE \$\$\$

Follow us on your favorite network site and look for a lot of super deals posted frequently... exclusively for our followers!



8-Channel Remote Ethernet Controller

Now you can easily control and monitor up to 8 separate circuits via the standard Ethernet network in your home or office. Connection wise it couldn't be simpler. The controller functions as an IP based web server, so it can be controlled by any internet browser that can reach your network! There are no drivers or proprietary software required, just access the controller like any web page from your PC, laptop, or even your smartphone! Security is assured allowing up to 4 separate user credentials. The controller can be set to a specific static IP within your network subnet or can be set to DHCP (auto negotiate). The controller can even be programmed to send you an email to notify and confirm power up and status changes!

To simplify the connection of your equipment to the controller, 8 separate and isolated relay outputs are provided! This gives you internet or network control of up to 8 separate functions. No need to worry about interfacing a logic high or logic low, or burning up the interface! The applications are endless! From something as simple as turning on and monitoring lights at your house with a normal latched closure to advanced control of your electronic gadgets, radio equipment, or even your garage door! Each relay contact is rated at 12A at 30VDC or 16A at 230VAC. Each of the 8 channels has built-in timer and scheduler programs for day, weekend, working days, every day, and every day except Sunday. Relay control functions are programmable for on, off, or pulse (1-99 seconds, 1-99 minutes, or 1-99 hours). In addition to control functions, the web interface also displays and confirms the status of each channel. Each channel can be custom labeled to your specific function name. The controller operates on 12VDC or 12VAC at 500mA or our new AC121 global 12VDC switching power supply below. Factory assembled, tested, and ready to go! Even includes a Cat-5 cable!

VM201 8-Channel Remote Ethernet Controller, Factory Assembled & Tested

\$169.95

Laser Beam Audio Communicator

Now you can talk to your friends over one of the most secure ~~communication systems~~ available, a laser beam! The transmitter uses a microphone or external audio to modulate a laser beam on and off at a rate of more than 16kHz so the audio fidelity is much better than that of a standard 3kHz telephone line!

The receiver includes filtering to remove the 16kHz carrier and leave behind the high quality audio, and then boost its level for use with earphones. Transmitter audio AGC keeps your level perfect! Includes transmitter, receiver and laser pointer. Each runs on a 9V battery (not included).

LBC6K Laser Beam Audio Communicator Kit

\$59.95

5A PWM Motor Speed Controller

This handy controller uses a pulse width modulated output to control the speed of a motor without sacrificing torque! Handles a continuous current of 5A and includes LED to indicate speed as well as an oversized gold heatsink! Also available factory assembled.

CK1102 5A PWM Motor Controller Kit \$14.95

Digital LED Thermometer

This handy thermometer reads Celsius or Fahrenheit on an eye-catching .56" LED display! Based on the DS18B20 sensor and controlled by a PIC, it has a range of -67°F to 257°F (-55°C to 125°C) with a wired remote range of 325 feet!

CK127 Digital LED Thermometer Kit \$29.95

Tone Encoder/Decoder

Encode and decode with the same kit! This little mini-kit will simultaneously encode and/or decode any audio frequency between 40Hz and 5,000 Hz! Precision 20-turn trim pot adjustment! 5-12VDC.

TD1 Tone Encoder/Decoder Kit \$9.95

Tickle-Stick Shocker

The kit has a pulsing 80 volt tickle output and a mischievous blinking LED. And who can resist a blinking light and an unlabeled switch! Great fun for your desk, "Hey, I told you not to touch!" Runs on 3-6 VDC.

TS4 Tickle Stick Kit \$9.95

Laser Light Show

Just like the big concerts, you can impress your friends with your own laser light show! Audio input modulates the laser display to your favorite music! Adjustable pattern & speed. Runs on 6-12VDC.

LLS1 Laser Light Show Kit \$49.95

Optically Isolated Module

The hobbyist's headache solver! Converts any AC or DC signal to logic level. The beauty is that the input and output are totally isolated from each other! Output can drive up to 150mA at 40VDC.

OM2 Optically Isolated Module Kit \$16.95

Water Sensor Alarm

This little \$7 kit can really "bail you out"! Simply mount the alarm where you want to detect water level problems (sump pump)! When the water touches the contacts the alarm goes off! Sensor can even be remotely located. Runs on a standard 9V battery.

MK108 Water Sensor Alarm Kit \$6.95

12VDC Regulated Switching Supply

Go green with our new 12VDC 1A regulated supply. Worldwide input 100-240VAC with a Level-V efficiency! It gets even better, includes DUAL ferrite cores for RF and EMI suppression. All this at a 10 buck old wallwart price! What a deal!

AC121 12VDC 1A Regulated Supply \$9.95

12VDC Worldwide Supply

It gets even better than our AC121 above! Now, take the regulated Level-V green supply, bump the current up to 1.25A, and include multiple blades for global country compatibility! Dual ferrite cores!

PS29 12VDC 1.25A Global Power Supply \$19.95

Touch Switch

The ultimate touch switch! Touch once - it's on, touch again - it's off, or use the momentary outputs that stay on only as long as touched. Two switch circuits on each board. Drives loads up to 100mA. Runs on 6-12VDC.

TS1 Touch Switch Kit \$9.95

Tri-Field Meter Kit

"See" electrical, magnetic, and RF fields as a graphical LED display on the front panel! Use it to detect these fields in your house, find RF sources, you name it. Featured on CBS's Ghost Whisperer to detect the presence of ghosts! Req's 4 AAA batteries.

TFM3C Tri-Field Meter Kit \$74.95

Electronic Watch Dog

A barking dog on a PC board! And you don't have to feed it! Generates 2 different selectable barking dog sounds. Plus a built-in mic senses noise and can be set to bark when it hears it! Adjustable sensitivity! Unlike my Saint, eats 2-8VAC or 9-12VDC, it's not fussy!

K2655 Electronic Watch Dog Kit \$39.95

Sniff-It RF Detector Probe

Measure RF with your standard DMM or VOM! This extremely sensitive RF detector probe connects to any voltmeter and allows you to measure RF from 100kHz to over 1GHz! So sensitive it can be used as a RF field strength meter!

RF1 Sniff-It RF Detector Probe Kit \$27.95

Electret Condenser Mic

This extremely sensitive 3/8" mic has a built-in FET preamplifier! It's a great replacement mic, or a perfect answer to add a mic to your project. Powered by 3-15VDC, and we even include coupling cap and a current limiting resistor! Extremely popular!

MC1 Mini Electret Condenser Mic Kit \$3.95

Classic Nixie Tube Clocks **HOT SELLER!**

Our next generation of classic Nixie tube clocks perfectly mesh today's technology with the Nixie era technology of the 60's. Of course, features you'd expect with a typical clock are all supported with the Nixie clock... and a whole lot more!

The clocks are programmable for 12 or 24 hour mode, various AM/PM indications, programmable leading zero blanking, and include a programmable alarm with snooze as well as date display, 4 or 6 tube, kit or assembled!

We then jumped the technological time line of the 60's Nixie displays by adding the latest multi-colored LEDs to the base of the Nixie tubes to provide hundreds of illumination colors to highlight the glass tubes! The LED lighting can be programmed to any color and brightness combination of the colors red, green, or blue to suit your mood or environment.

Then we leaped over the technological time line by integrating an optional GPS time base reference for the ultimate in clock accuracy! The small optional GPS receiver module is factory assembled and tested, and plugs directly into the back of the clock to give your Nixie clock accuracy you could only dream of!

The clocks are available in our signature hand rubbed Teak & Maple, polished aluminum, or clear acrylic bases. You also have your choice of IN-14 or highly sought after IN-8-2 nixie tubes (for the 6-tube clock).

NIXIE Classic Nixie Tube Clock Kits From \$229.95

GET THE INUTS AND VOLTS DISCOUNT!

Mention or enter the coupon code **NVRMZ12** and receive **10% off your order!**

800-446-2295
www.ramseykits.com

Prices, availability, and specifications are subject to change. We are not responsible for typos, stupid, printer's bleed, or confusion that April showers bring May flowers! Robin thinks winter is over, just because she lives in CA! Wrong! Visit www.ramseykits.com for the latest pricing, specials, terms and conditions. Copyright 2013 Ramsey Electronics Inc. so there!

RAMSEY ELECTRONICS®

590 Fishers Station Drive
Victor, NY 14564
(800) 446-2295
(585) 924-4560

PicoScope[®]




PC OSCILLOSCOPES

THE WORLD'S FIRST USB 3.0 OSCILLOSCOPE



PicoScope	3207A	3207B
Bandwidth	250 MHz	250 MHz
Sampling	1 GS/s	1 GS/s
Memory	256 MS	512 MS
Signal generator	Function generator	AWG
Price	\$1813	\$1978
Power supply	From USB port	
Compatibility	USB 2.0 & 3.0	

NEW

PicoScope 3200 Series	PicoScope 3400 Series	PicoScope 4000 Series	PicoScope 6400 Series
	 NEW		 UPDATED 2012
2 Channels + EXT and AWG 60 to 200 MHz Bandwidth 500 MS/s Sampling 8 bits Resolution (12 bits enhanced) 4 to 128 MS Buffer memory Price from \$658	4 Channels + EXT and AWG 60 to 200 MHz Bandwidth 1 GS/s Sampling 8 bits Resolution (12 bits enhanced) 4 to 128 MS Buffer memory Price from \$988	2 Channels + EXT and AWG, or 4 Channels 20 to 100 MHz Bandwidth 80 to 250 MS/s Sampling 12 bits Resolution (16 bits enhanced) 32 MS Buffer memory Price from \$823	4 + External trigger and AWG or Func. Generator 250 to 500 MHz Bandwidth 5 GS/s Sampling 8 bits Resolution (12 bits enhanced) 128 MS to 1 GS Buffer memory Price from \$3300