Why the Propeller Works
by Chip Gracey, creator of the Propeller chip and President of Parallax Inc.

I am the person who designed, debugged, tuned, and tested the Propeller chip. This project took eight years of my time, plus two years of a layout engineer’s time. An excruciating amount of attention went into every aspect of the Propeller’s implementation and testing, and I allowed no compromises.

The Propeller was an entirely “full-custom” effort. Every polygon of the Propeller’s mask artwork was made here at Parallax. We designed our own logic, RAM, ROM, PLLs, bus-drive references, oscillators, and even FDD-hardwired I/O pads. All these structures were first fabricated on test chips and then thoroughly exercised, often resulting in design changes. This yielded an ideal set of known-good blocks, which could be confidently applied to the overall design. Then the whole chip was fabricated and subsequently tested at many levels. This allowed us to fix any problems resulting from integration and to fine-tune the clocking systems that are key to the Propeller’s low-power consumption. The final chip, which is the only version we’ve ever sold, is the third iteration of this whole-chip process and has no known problems.

The Propeller was given the kind of thorough design treatment that almost no other chips receive today. It used to be that every chip was full-custom, and all of its transistors and wiring were designed by hand, for the point of application. As semiconductor technology shrunk, though, the prevailing design methodology shifted away from the kind of specialization, toward generalization and abstraction, so that designs of greater complexity could be practically realized. The modern design methodology centers around hardware description languages, IP block reuse, and the automated placement and interconnection of potentially billions of gates. The end silicon result is inevitably an incomprehensible sea of wiring, standard cells, and IP blocks, usually more of which were designed by the engineers applying them. This methodology is certainly a boon for very complex designs, but it has become the standard approach for designing almost any chip containing logic today. For chips, the old method means small dies, high speed, low power, and existing performance, whereas the modern method tends to generate bigger dies, lower speed, more heat, and sometimes bugs from IP which you have no control over. I’m sure you get the idea.

Through an exceptional effort, we made the Propeller as electrically robust and efficient as we could. Yet, the core quality of the Propeller really resides in its architecture. The architecture is what took the first six years of development to iron out, and the architecture is what engages people. All the effort that went into the silicon implementation was to ensure that this core quality was ideally housed.

The story of the Propeller’s architecture would be a book in itself but to get an idea of the point, you can visit the Propeller discussion forum [http://forums.parallax.com] and witness the excitement of people doing things they never thought possible. They are finding the Propeller to be a great vehicle for invention and discovery, as well as the means to realize complex embedded systems that are not possible with any other chip. A few forum members have even said that the Propeller has drawn them back into software and electronics after long absences.

The Propeller is tough, reliable, and low-power. It’s no illusion, and no accident.

We plan on a very long sales life for the Propeller and we have no intention of diluting the concept with many slight variants, for which you’d inevitably be getting end-of-life notices after a few years. This is good news for customers, because they are the ones who are going to be making investments in programming that will, in turn, dwarf the energy that we spent developing the Propeller. We made a platform that is, hopefully, deserving of their coming efforts.

Sincerely,

Chip Gracey
President
Parallax, Inc.

Learn more at www.parallax.com/propeller
The PM-128E is an enhanced version of our best selling PM-128A. The E version is a sophisticated PC based oscilloscope adaptor providing performance compatible to mid/high priced. Their unmatched performance can be set to work with either a 5VDC or 9V (7-11V) independent power source. It is a sophisticated PC based oscilloscope adaptor providing performance compatible to mid/high priced. Their unmatched performance can be set to work with either a 5VDC or 9V (7-11V) independent power source. It is suitable for crowded circuit components, as well as advanced packages, as well as being suitable for lead or lead-free soldering. It brings sophisticated trigger and record length expansion to the simplicity of a handheld and can be used easily for rework. 85BYGH350C-03, 85BYGH350B-03, 85BYGH207, 42BYGH404, 57BYGH207, 57BYGH303, 57BYGH405.

- 200MHz, 2Ch DSO
- 1 GS/s Real Time Sampling Rate & 25 GS/s Equivalent-Time Sampling
- 25 points Record Length Maximum
- Large 5.6-in TFT Color Display
- "F"T Function
- Standard USB Host and Device Interfaces; Optional GPIB
- Optional DC Power (Factory Installed)
- "Auto-Level, AUTO, NORMAL, SINGLE, TV, Edge, Pulse Width, Time-delay(2CH Only), Event-delay(2CH Only)
- RX232 and TTL Inputs (65 MHz 10:1, 100 MHz 5:1, 200 MHz 2.5:1)
- 2x16 and 4x20 Serial Displays
- Three simultaneous dual traces
- Shake proof design
- 3 wire hookup
- Manual 200/100/60 MHz Bandwidth
- 2 or 4 Input Channels
- ±1 GS/s Real Time Sampling Rate & 25 GS/s Equivalent-Time Sampling
- 25 points Record Length Maximum
- Large 5.6-in TFT Color Display
- "F"T Function
- Standard USB Host and Device Interfaces; Optional GPIB
- Optional DC Power (Factory Installed)
- "Auto-Level, AUTO, NORMAL, SINGLE, TV, Edge, Pulse Width, Time-delay(2CH Only), Event-delay(2CH Only)
- RX232 and TTL Inputs (65 MHz 10:1, 100 MHz 5:1, 200 MHz 2.5:1)
- 2x16 and 4x20 Serial Displays
- Three simultaneous dual traces
- Shake proof design
- 3 wire hookup
- Manual 200/100/60 MHz Bandwidth
- 2 or 4 Input Channels
- ±1 GS/s Real Time Sampling Rate & 25 GS/s Equivalent-Time Sampling
- 25 points Record Length Maximum
- Large 5.6-in TFT Color Display
- "F"T Function
- Standard USB Host and Device Interfaces; Optional GPIB
- Optional DC Power (Factory Installed)
- "Auto-Level, AUTO, NORMAL, SINGLE, TV, Edge, Pulse Width, Time-delay(2CH Only), Event-delay(2CH Only)
- RX232 and TTL Inputs (65 MHz 10:1, 100 MHz 5:1, 200 MHz 2.5:1)
- 2x16 and 4x20 Serial Displays
- Three simultaneous dual traces
- Shake proof design
- 3 wire hookup
- Manual 200/100/60 MHz Bandwidth
- 2 or 4 Input Channels
- ±1 GS/s Real Time Sampling Rate & 25 GS/s Equivalent-Time Sampling
- 25 points Record Length Maximum
- Large 5.6-in TFT Color Display
- "F"T Function
- Standard USB Host and Device Interfaces; Optional GPIB
- Optional DC Power (Factory Installed)
- "Auto-Level, AUTO, NORMAL, SINGLE, TV, Edge, Pulse Width, Time-delay(2CH Only), Event-delay(2CH Only)
- RX232 and TTL Inputs (65 MHz 10:1, 100 MHz 5:1, 200 MHz 2.5:1)
- 2x16 and 4x20 Serial Displays
- Three simultaneous dual traces
- Shake proof design
- 3 wire hookup
- Manual 200/100/60 MHz Bandwidth
- 2 or 4 Input Channels
- ±1 GS/s Real Time Sampling Rate & 25 GS/s Equivalent-Time Sampling
- 25 points Record Length Maximum
- Large 5.6-in TFT Color Display
- "F"T Function
- Standard USB Host and Device Interfaces; Optional GPIB
- Optional DC Power (Factory Installed)
- "Auto-Level, AUTO, NORMAL, SINGLE, TV, Edge, Pulse Width, Time-delay(2CH Only), Event-delay(2CH Only)
Go Wireless  
Go Rabbit®

Check out our new Wi-Fi and ZigBee® core modules — the latest additions to our pin-compatible family.

RCM4510W  
ZigBee RabbitCore®  
$72 (qty. 100)

RCM4400W  
Wi-Fi RabbitCore  
$99 (qty. 100)

Wireless Development Kits  
Only $199 (reg. $299)

Applications
- Industrial Control
- Remote Terminal Unit (RTU)
- Building Automation
- Data Acquisition

RCM4500W Family
- ZigBee/802.15.4
- 49 GPIO
- 6 Serial Ports

RCM4400W Family
- Wi-Fi/802.11
- 35 GPIO
- 6 Serial Ports

Order Your Kit Today
Development kits include everything to start your development, Dynamic C® software, wireless RabbitCore module, development board, programming cables and accessories.

www.wirelessrabbit.com

RABBIT Semiconductor
2900 Spafford Street, Davis, CA 95618  Tel 530.757.8400
36 KEEP THOSE COMPONENTS COOL!
Adjust fan speed by the temperature to keep things running smooth and quiet.
■ By Stuart Ball

41 FLIGHT RECORDER FOR MODEL ROCKETRY Part 2
Learn the embedded software required to record and replay three channels of acceleration. Plus, a plot of actual flight data is shown in Microsoft Excel.
■ By Mike Bessant

46 CONTROL YOUR WORLD
■ By Michael Simpson

52 BUILD A .01% ACCURATE VOLTAGE REFERENCE
So, just how accurate is your DMM?
■ By Doug Malone

58 FLOATING POINT MULTIPLICATION AND DIVISION WITHOUT HARDWARE SUPPORT
Using Horner's method to build a color organ.
■ By Craig Lindley

68 SWIFT, SMALL, “CELL” BROADBAND CHIP
These days, applications and experimentation with the Cell/B.E. technology abound. So, just how fast does the Cell chip cruise?
■ By David Geer

71 A LOGIC ANALYZER TUTORIAL Part 1
A logic analyzer is an indispensable tool if you design and troubleshoot digital circuits. Find out why.
■ By Vaughn Martin

Nuts & Volts (ISSN 1528-9885/CDN Pub Agree#0702530) is published monthly for $24.95 per year by T & L Publications, Inc., 439 Princeton Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL MAILING OFFICES. POSTMASTER: Send address changes to Nuts & Volts, P.O. Box 15277, North Hollywood, CA 91615 or Station A, P.O. Box 54, Windsor ON N9A 6J5; secretars@nutsvolts.com.
Digital Oscilloscopes

- 2 Channel Digital Oscilloscope
- 500 MSa/s max single shot rate
- 1Mpt sample memory
  - 250 MSa/S (Dual channel) 512 Kpts
  - 500 MSa/S (Single channel) 1 Mpts
- Only 9 oz and 7" x 3.5" x 1.5"
- Portable and Battery powered
- USB 2.0
- FFT Spectrum Analyzer

<table>
<thead>
<tr>
<th>Model</th>
<th>Channels</th>
<th>Frequency</th>
<th>Memory</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSO-8202</td>
<td>2</td>
<td>200 MSa</td>
<td>128 Kpts</td>
<td>$799</td>
</tr>
<tr>
<td>DSO-8502</td>
<td>2</td>
<td>500 MSa</td>
<td>1 Mpts</td>
<td>$950</td>
</tr>
</tbody>
</table>

Logic Analyzer & Pattern Generator

- 32 channels Logic Analyzer
- up to 32 channels Pattern Generator
- up to 400 MSa/s
- Variable Threshold
- 2 External Clocks
- SPI output and disassembly
- I2C output and disassembly
- up to 2Msamples/ch

<table>
<thead>
<tr>
<th>Model</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO-3208A</td>
<td>$750</td>
</tr>
<tr>
<td>IO-3232A</td>
<td>$899</td>
</tr>
<tr>
<td>IO-3232B</td>
<td>$1399</td>
</tr>
</tbody>
</table>

www.Link-instruments.com
(973) 808-8990

September 2007
The projects in this issue of *Nuts & Volts* address a range of interests, from rocketry instrumentation and weather, to thermal management and instrument calibration standards. Although the relevance of standards is most evident in Doug Malone’s article on building a voltage reference, each of the projects is a tribute to the necessity of standards in the design, construction, operation, and maintenance of electronic circuitry and instrumentation.

In electronics, standards — which are sometimes expressed in terms of compatibility — range from the thread and diameter of nuts and bolts, logic families, and computer busses, to communications and low-level signal levels. Fortunately for readers, the authors have assumed the responsibility of determining which hardware and software combinations provide the best results. You don’t have to decide between resistor-transistor logic (RTL), transistor-transistor logic (TTL), or complimentary metal-oxide semiconductor logic (CMOS) chip sets, Bluetooth or WiFi wireless communications, or between a PIC and a STAMP microcontroller.

A wireless weather station, for example, may incorporate sensors calibrated to international standards of barometric pressure, temperature, and humidity. These inherently analog signals are typically interfaced to digital hardware and associated software. The cascade of national and international standards involved — from the physical property measured to the signal levels in the microcontroller or microprocessor — is invisible to the casual experimenter. The IEEE alone is responsible for over 1,300 standards in telecommunications, information technology, and related fields (see standards.ieee.org). In addition to decreasing development time and costs, standards increase product quality and safety, and provide a modicum of protection against obsolescence.

Several important standards are readily accessible, even if only indirectly. For example, the National Institute of Standards and Technology (NIST) broadcasts standard time and frequency signals over the web, telephone network, and radio. NIST maintains the primary standard for frequency and time intervals with a Cesium Fountain Atomic Clock in Boulder, CO (see tf.nist.gov/cesium/fountain.htm). You can access time based on this standard over the Internet, accurate to within 0.2 seconds, at www.time.gov. The NIST radio station WWV, known to most radio amateurs and shortwave listeners, broadcasts time signals at 2.5, 5, 10, 15, and 20 MHz. If you have one of those ‘atomic’ clocks or watches, the synchronizing signal is from NIST station WWVB, which broadcasts continuously at 60 kHz.

The Cesium time standard is an example of an *intrinsic* standard, in that it is based on a constant of nature, as opposed to an experiment conducted in a specific environment. A voltage standard based on a battery, for example, is an extrinsic standard. For this reason, in 1972 NIST moved from a definition of the standard volt based on the Weston cell to the Josephson Volt Standard, which is based on electrical properties of the Josephson junction — two superconductors linked by a non-conducting barrier. While a standard based on a reproducible, solid-state, cryogenic superconductor has been a boon to industry, it is still beyond the reach of individuals. Hence, the value of Doug’s article. Doug’s article, albeit an extrinsic standard, is an affordable source for a standard volt.

Given the readily available, inexpensive digital multimeter, a standard volt may seem superfluous. After all, for $30, it’s possible to obtain a six-digit DMM with a built-in frequency counter. If you own a Fluke, HP, or other quality DMM, you know that part of what you paid for is accuracy over time and with changes in the operating environment. A $30 meter might be accurate to within a tenth of a volt out of box, but six weeks later, in the heat of summer or cold of winter, a measurement might be off by a half volt. Although all meters require recalibration, it’s more likely that the $30 DMM will require more frequent recalibration than a quality DMM. A handy voltage standard, even if it is extrinsic, enables you to recalibrate your inexpensive DMM or oscilloscope.

As you prepare to build one of the projects described in this month’s *Nut & Volts*, pause for a moment to consider the myriad standards involved, and the limitations of your multi-digit digital test equipment. Understanding the underlying standards can not only make your debugging more productive, but you’ll appreciate the lengths developers have gone through to insure component and system compatibility. NV
Announcing a complete hardware and software solution from NetBurner

The NetBurner MOD5234
ETHERNET CORE MODULE with eTPU

Only $99
Qty. 100

NetBurner MOD5234 Ethernet Core Module Features

INDUSTRIAL TEMPERATURE RANGE
-40°C to +85°C

PERFORMANCE AND MEMORY
32-Bit CPU | Freescale ColdFire MCF5234 147 Mhz
2MB Flash Memory | 8MB SDRAM

DEVICE CONNECTIVITY
10/100 Ethernet | 3 UARTs | 16-channel eTPU | I²C | SPI | CAN
47 Digital I/O | 16-bit Data Bus | SD/MMC Flash Card Support

Customize with NetBurner’s Royalty-free Software Suite

DEVELOPMENT SOFTWARE
NB Eclipse IDE | Graphical Debugger | Deployment Tools | Examples

COMMUNICATION SOFTWARE
TCP/IP Stack | HTTP Web Server | FTP | E-Mail | PPP | Flash File System

All hardware and software is included with the
NetBurner MOD5234 Development Kit for only $299!
The Development Kit features NetBurner’s Eclipse, an enterprise level professional IDE that provides editing, downloading and debugging in one environment.
Order the MOD5234 Development Kit: Product No. NNDK-MOD5234-KIT

Product No. | MOD5234-100IR
Information and Sales | sales@netburner.com
Web | www.netburner.com
Telephone | 1-800-695-6828
EVERYTHING FOR ELECTRONICS

MAGIC BOX KIT DOES THE TRICK

I recently completed the Magic Box featured in your April/May editions and must say it was without a doubt one of the best projects I have seen in your magazine. Since I don’t usually have all the spare parts, nor the time to gather them up, I usually opt for the kit version and go from there. When it arrived, I was pleasantly surprised to find all the parts, very well packaged in groups and labeled, rather than all thrown into one large bag like many other kits. The building instructions and documentation were expertly explained and diagramed. It was a real pleasure building the Magic Box, and it’s a blast watching people try to figure it out. I contacted Zone-masterskits.com to request a set of extra Coke™ bottle pawns, and their response was immediate and informative. I look forward to Nuts & Volts for just this type of project. I only wish all the projects were of this magnitude. Keep them coming.

Kimberly Hamel

EXPERIMENTING WITH PERSPECTIVE

Thank you for Bryan Bergeron’s editorial, Developing Perspectives, in the July ’07 issue of Nuts & Volts.

Your readers should try the American Radio Relay League’s online course on digital electronics (www.arrl.org). This is an excellent hands-on course on using CMOS ICs.

Also, an additional way to develop perspective is to try to invent a new way of doing a function, such as sending radio waves, adding numbers, steering a robot, storing electrical energy, etc. This inventive process will encourage one to really examine what is going on in existing electronics circuits and to consider what alternatives may be possible. You can use a morphological table to further stir up new ideas.

Nickolaus E. Leggett
Reston, VA

OP-AMP OPTIONS

I have some questions for Gerard Fonte concerning his article in the August N&V on the ARB. The op-amp he specified is an LMC6082 CMOS device. That particular op-amp seems to be difficult to find in a PDIP package in small quantities. Do you know where I can buy just a couple of them? Alternatively, what other op-amp would you recommend? How about the LMC6442 or the LMC6042?

Continued on page 97
DSX-4000LG
4 Channel DVR with Audio
- RS-485 port for PTZ
- TCP/IP Networkable
$375.95

REMOTE ZOOM!
AFL-22XHR
High-Resolution 1/4" Color CCD Zoom Camera
- 480 TVL, 0.7 LUX, 3.0-85.8mm VF AI
- 22x Optical / 16x Digital Zoom
$379.95

REMOTE PTZ!
IP-280
Color Network P/T/Z Camera
- 640 x 480@30fps, 90° Pan / 30° Tilt
- Browser-based TCP/IP Operation
$199.95

Built-in Microphone!
550 TVL!
HVS-D550
High-Res. 1/3" Color CCD UTP Camera
- 550 TVL, 0.15 LUX Color, 0.001 LUX DSS
- UTP Video Out, Audio Out, Dual Voltage
$159.95

Built-in Microphone!
TFT-10.4
10.4" TFT LCD - 640 x 480 Resolution
- RCA Video Input, VGA Input
- IR Remote & Headphones Included
$165.95

Built-in Microphone!
USB!
WRL-722USB
2.4 GHZ 4 Channel Receiver
- USB Interface Allows Video & Still Capture Directly to Your PC
$129.95

Call the experts at PolarisUSA for all of your video and security needs!
800-308-6456
Prices subject to change without notice!
nanotechnologists at the University of California, Riverside (www.ucr.edu) have come up with a way to control the color of a liquid by applying a magnetic field. The liquid is just a colloidal suspension of iron oxide nanoparticles in water, but the particles are “superparamagnetic,” meaning that they have magnetic properties only in the presence of an external magnetic field. This is in contrast to “ferromagnetic” materials, which retain their magnetism after the field is removed.

If you vary the field strength applied to the liquid, the particle arrangement is altered, thereby changing what happens when light passes through or is deflected by the solution. According to researcher Yadong Yin, “By reflecting light, these crystals (also called photonic crystals) show brilliant colors. Ours is the first report of a photonic crystal that is fully tunable in the visible range of the electromagnetic spectrum, from violet light to red light.”

Because the materials are cheap, nontoxic, and readily available, the technology could be used to create reflective color displays that use millions of the particles for pixels. It also has possible applications in erasable, rewritable electronic paper and might even be used to make ink that changes color electromagnetically.

**ANOTHER SOLAR CELL ADVANCEMENT**

Meanwhile, over on the other side of L.A., a research team has been working on a new type of plastic solar cell with a noteworthy boost in efficiency. Nobel laureate Alan Heeger — a physics professor at UC, Santa Barbara (www.ucsb.edu) — and associates have come up with a “tandem” organic device that, by virtue of its dual multilayer design, gathers a wider range of the spectrum, at both shorter and longer wavelengths. According to Heeger, the cells “... can be fabricated to extend over large areas by means of low-cost printing and coating technologies that can simultaneously pattern the active materials on lightweight, flexible substrates.”

At present, the tandem cells are working at only about 6.5 percent efficiency, which isn’t nearly as impressive as the 40.7 percent achieved last year by a Boeing Spectrolab (www.spectrolab.com) device. However, “This is the highest level achieved for solar cells made from organic materials. I am confident that we can make additional improvements that will yield efficiencies sufficiently high for commercial products,” Heeger noted. He expects the technology to be commercially available within about three years.

**COMPUTERS AND NETWORKING**

**NEW LOW-END COMPUTER LINEUP**

In July, Dell (www.dell.com) introduced a new lineup of PCs designed for small businesses on a budget. Consistent with market trends that saw notebook sales up by 23 percent in the first quarter but desktops pretty stagnant, the new brand at introduction included four laptops but only one desktop (although available in either mini-tower or slim case versions). Apparently, Dell has received a substantial number of complaints about PCs that arrive bloated with trialware (also known as “junkware” or “crapware”), so the Vostro™ (Latin for “yours”) machines come without it. They do include “simple to use tools that address top-of-
mind problems such as data back-up, PC performance and health, and specialized networking support for customers without dedicated IT staff.”

The notebooks start at $449, and you can get the Vostro 200 shown for about $319. The notebooks use either dual-core AMD or Intel Core™ 2 Duo processors and have displays ranging from 14.1 to 17 inches. The 200 uses the Intel chip. The company also introduced a new 19-inch widescreen display that lists for $229.

**AN MD IN YOUR PC**

One of the weirder programs out there is Desk Doctor, a product of Einspine Ltd. (www.einspine.com), a Malta-based entity. The basic concept is that you get a virtual Dr. Russ Hornstein (the inventor) in your PC, and he makes regular assessments of your upper body condition so as to avoid repetitive strain injuries (e.g., carpal tunnel syndrome and tendonitis.) The program monitors computer activity and calculates your health score, which is displayed onscreen. When your score drops to a risky level, the MD will determine that a workout is needed. You then call up your personal exercise trainer, who serves up a high-res video demonstration of the suggested routine.

According to the company, keeping a good average health score will ensure that your musculoskeletal health is maintained. The product has actually been around for a couple years, but version 1.3.1 has just been released and, because it uses universal binary, it runs on both Intel and PowerPC Macs in native mode. (Version 1.2.0 is available for Windows 98 through Vista.) The program sells for $129 on the website, but you can download a free 15-day trial.

**FREE EMAIL FOR iPHONES**

If you are among those who have actually shelled out as much as $600 for an iPhone and every month are forking over between $59.99 and $219.99 to AT&T (depending on the plan), money may not mean much to you. On the other hand, you may be living in a 1983 Oldsmobile to make up for it, in which case it’s nice to know that Cortado (www.cortado.com) is offering all iPhone users, anywhere in the world, a free mobile e-mail service (subject to the data flat rate from AT&T, of course).

The Cortado Free program allows users to automatically receive emails at any time, and it includes a personal Cortado email address with professional spam filter, virus protection, and 20 MB of storage. (Or you can use an existing email address.)

**NEW MEMORY SUBSYSTEM FOR EMBEDDED APPLICATIONS**

In July, Silicon Storage Technology (www.sst.com) began releasing sampling quantities of the model SST88VP1107, which is the first in its new “All-in-OneMemory” subsystem line. The concept puts code storage (NOR), data storage (NAND), and system RAM (PSRAM) functions on a single PSRAM bus, thus simplifying designs and reducing the time to market in mobile and embedded applications.

The device comes configured with 512 KB instant-on boot NOR, 128 MB execute-in-place (XIP) code storage, 120 MB data storage, and 12 MB system RAM, all in a 10 x 13 x 1.4 mm package. According to the company, “by intelligently managing all memory components with a resident 32-bit microcontroller, All-in-OneMemory offers a large and expandable XIP area, instant secure boot, memory demand paging, NAND Flash management, and ATA data storage protocol on a single PSRAM bus in a small footprint package; thereby reducing system complexity.

**CIRCUITS AND DEVICES**

**MP4 PLAYER/WATCH COMBINATION**

This month’s “neat, but probably useless gadget” award goes to the MP4 Player Watch from Etronicland (www.etronicland.com). Not only is it a digital wristwatch, it also plays MP4 videos on a built-in color screen and MP3 music through an included set of earphones.

The device comes with up to 2 GB of Flash memory, a 1.5-in color screen (128 x 128 pixels), and digital recording capability. It has five equalizer modes plus a “super bass 3-D” playback mode. Viewing options include thumbnail, slide show, and manual, and it even comes with a USB cable for file transfer and recharging. (You get eight hours of continuous music playback on a charge.)

Originally priced at $199.99, it is now offered at $159.99, so get your Christmas shopping over early and look forward to watching your friends and family walk into walls and drive off cliffs.
and lowering overall cost.”

**LED DESIGNED FOR SOLID-STATE LIGHTING**

Avago Technologies (www.avagotech.com) recently rolled out a new Moonstone™ warm white 1W power LED for high-brightness applications such as decorative architectural and garden lights, reading lights, and accent and marker lights.

The ASMT-MY00 LED features a smooth radiation pattern and the 110-degree viewing angle can be driven at 350 mA, delivering 56 lumens. The chip is encapsulated in a silicone compound to provide UV and heat resistance, is specified for operation over a range of -40 to +95°C temperature range, and can withstand electrostatic discharge levels of 16 kV. Color temperatures range from approximately 2,600 to 4,000 K. The price in manufacturing quantities is about $2.50.

**INDUSTRY AND THE PROFESSION SPammers/Scammers ARRESTED**

If your inbox has contained fewer spam emails promoting obscure penny stocks lately, it could be because the Securities and Exchange Commission (www.sec.gov) recently filed fraud charges against two Texans who ran a scam that hijacked PCs nationwide, used them to disseminate millions of spam emails, and cheated investors out of more than $4.6 million. Darrel Uselton and his uncle Jack, both described as “recidivist securities law violators,” allegedly illegally profited from a 20-month “scalping” scam by obtaining shares from some 13 different penny stock companies and artificially inflating the market for them through the email and other promotional activities. They subsequently sold the stocks at a profit.

Unfortunately for the Useltons, this is a violation of Section 10(b) of the Securities Exchange Act of 1934 and Rule 10b-5, and the SEC was not amused. In related enforcement actions, Texas authorities indicted the Useltons for engaging in organized crime and money laundering and seized more than $4.2 million from their bank accounts. The best part is that you can see a video of Darrel’s arrest at www.oag.state.tx.us/media/videos/play.php?image=070907uselton_arrest&id=235.

**LEON H. SIBUL, 1932-2007**

Earlier this year, the electronics industry lost Leon H. Sibul, a pioneer in the application of signal processing to underwater guidance and control systems. Sibul was born in Võru, Estonia in 1932, but fled to a German refugee camp in 1944 and finally emigrated to the Washington, DC, area in 1949. He served in the US Air Force as an electronics and radio technician from 1953 to 1957. In 1960, he was hired by Bell Labs, where he worked on electronic switching systems and Telstar, the satellite that ushered in the era of satellite communications. Finally, in 1964, he joined the Applied Research Lab at Penn State, conducting signal-processing R&D for undersea weapons guidance and control, sonar systems, and other naval applications, and also taught acoustics. Sibul retired in 2002. He held electrical engineering degrees from George Washington University (bachelor’s), New York University (master’s), and Penn State (Ph.D.) and was a life member in the IEEE.
Microchip's 20-pin, general-purpose PIC® microcontrollers are flexible and easy to use – saving you design time and development costs.

Whether you’re designing an MP3 player or a flight control system, Microchip’s low-pin count devices give you the features you need to get the job done, while making configuration easy. A simple architecture, a cost-saving integrated peripheral set, and easy-to-use development tools allow you to spend your time perfecting your application – instead of debugging it.

---

**20-PIN DEVICE OPTIONS**

<table>
<thead>
<tr>
<th>Device</th>
<th>Flash (words)</th>
<th>SRAM (bytes)</th>
<th>EEPROM (bytes)</th>
<th>10-bit A/D Ch</th>
<th>Timers 8/16-bit</th>
<th>Other Features</th>
<th>20-pin Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F631</td>
<td>1024</td>
<td>64</td>
<td>128</td>
<td>0</td>
<td>1/1</td>
<td></td>
<td>PDIP 50-pin 5x5QFN</td>
</tr>
<tr>
<td>PIC16F677</td>
<td>2048</td>
<td>128</td>
<td>256</td>
<td>12</td>
<td>1/1</td>
<td>SPI/PC™</td>
<td>PDIP 50-pin 5x5QFN</td>
</tr>
<tr>
<td>PIC16F685</td>
<td>4096</td>
<td>256</td>
<td>256</td>
<td>12</td>
<td>2/1</td>
<td>ECPP</td>
<td>PDIP 50-pin 5x5QFN</td>
</tr>
<tr>
<td>PIC16F687</td>
<td>2048</td>
<td>128</td>
<td>256</td>
<td>12</td>
<td>1/1</td>
<td>EUSART, SPI/PC™</td>
<td>PDIP 50-pin 5x5QFN</td>
</tr>
<tr>
<td>PIC16F689</td>
<td>4096</td>
<td>256</td>
<td>256</td>
<td>12</td>
<td>1/1</td>
<td>EUSART, SPI/PC™</td>
<td>PDIP 50-pin 5x5QFN</td>
</tr>
<tr>
<td>PIC16F690</td>
<td>4096</td>
<td>256</td>
<td>256</td>
<td>12</td>
<td>2/1</td>
<td>ECPP, EUSART, SPI/PC™</td>
<td>PDIP 50-pin 5x5QFN</td>
</tr>
<tr>
<td>PIC16F785</td>
<td>2048</td>
<td>128</td>
<td>256</td>
<td>12</td>
<td>2/1</td>
<td>Op Amps, CCP, Hardware PWM</td>
<td>PDIP 50-pin 5x5QFN</td>
</tr>
</tbody>
</table>

---

Get started today with the PICkit™ 2 Starter Development Kit for only $49.99!
When I’ve discussed driving an LCD module from a PIC MCU in the past, I received several emails asking where to buy one. I just took it for granted that everybody knew they were available just about everywhere. The secret is in the common driver chip used in the LCD modules — the Hitachi 44780 chip that handles all of the character generation. If you find an LCD module with that chip, then you are set. You can get that type of LCD module from the typical suppliers, such as Mouser, Jameco, or visit individual websites. One of my favorites is junun.org. They sell robot kits and parts, and also have a great deal on LCDs. You can even get surplus LCDs and LCDs removed from equipment, at Marlin Jones (MPJA.com) and others, at incredibly cheap prices. In fact, I use a surplus unit in this month’s project.

For this month’s article, I wanted to cover a little different feature of the Hitachi LCD chip’s character generator, which many of you may not know about. The first eight locations of character memory can be modified to make custom 5x8 characters. This month, I will load custom 5x8 characters into those eight locations and then use them to display large characters on a 4x20 LCD. I will drive the 4x20 LCD from a BasicATOM 28-pin chip, which is just a PIC16F876A with the custom BasicATOM self-programming bootloader installed. To make the connections simpler, I use my Ultimate OEM module with the BasicATOM 28 installed. However, wiring it directly to any PIC MCU will also work. If you want to use the PICBASIC PRO compiler or some other compiler, the software setup is very similar. Let me show you how it’s done.

**PROJECT DESCRIPTION**

To start, let me explain how the custom characters are set up in the LCD. The LCD control chip has eight locations at the beginning of its character memory, which can be modified to make custom 5x8 characters. Once those are stored in the LCD’s chip, these custom characters can be called in the same manner as any standard, pre-stored ASCII character. In Figure 1, a 5x8 character bitmap is shown with the first custom character the program will define. This custom character will be placed at memory location $00 of the LCD character memory. It then forms a small ramp that slopes downward.

Each row of the character has to be defined by a byte value. Since the characters are only five bits wide in size, the three most significant bits of the byte value are always zero. The highest bit value used is the fifth bit (bit number 4).

The key is to set or clear the proper bits to form the characters.
character you want. For example, if you look at the fourth line of the character box (offset row value 3); the byte to the right shows hex $10 or binary %00010000. This makes the fifth block solid black, but the rest of the row clear. The next line sets two blocks black, by using hex byte $18 or binary %00011000, and the rest you can see in Figure 1. By setting these bits and storing the data in the LCD’s character memory, we have established a new custom character that the software can call.

This doesn’t stop us from displaying standard, single-line words on the LCD screen. It does, however, allow us to also create large digits that span all four lines of a 4x20 LCD, by using the custom characters shown in Figure 2. It shows seven custom characters on line 2 of the LCD, but the eighth custom character is a blank so it doesn’t appear. The main project software loop (discussed later) will create the hexadecimal number system and display them in sequence, starting with the numbers 0–9 and then A–F, all in a large-character format that spans all four lines. The number “1” is displayed in Figure 3 to illustrate what I’m trying to describe. Using large numbers like this makes it very easy to read from across the room.

**PROJECT SETUP**

If you remember the 2x16 LCD project that I did in a previous column, then you will find the 4x20 LCD has all the same connections. A great advantage to using LCD modules is their common connection system, which makes it easy to change from 2x16 to 4x20. Figure 4 shows the connections to the Ultimate OEM BasicATOM module. You can easily connect the LCD to a PIC16F876A directly, by following the connection names in the schematic.

**SOFTWARE**

This software listing is kind of long, but most of the code deals with setting up the LCD to display the large custom characters. (The complete software listing is available on the Nuts & Volts website at www.nutsvolts.com.) In the code, you will notice the “|” pipe character at the end of several lines. This is for line continuation. This is a special character that the BasicATOM compiler recognizes as a continuation message. When the compiler sees that character, it knows the command line was too long for the editor window and continues on the next line. Setting up the characters takes a lot of space, so the line-continuation function is used often.

**HOW IT WORKS**

First, we establish a few variables and constants. The variables are just temporary storage locations labeled X and Char. The constants define the LCD “E” pin and “RS” pin.
The next section is the heart of this program. In this block of code, the custom characters are created and stored in the LCD character memory locations 0 through 7. Each character takes eight bytes of data, for a total of 64 bytes (eight characters times eight bytes).

To do this, we first have to point to location zero of the Character RAM. We do this with the LCDWRITE command, again by sending the “CGRAM” pointer. We don’t have to add an address value, since it defaults to the zero or first location.

*** Create Custom Characters in LCD memory locations 0-7 ***

`lcdwrite rspin\epin,outc,[CGRAM]`

Now, we send the custom characters to the LCD character RAM by using a FOR-NEXT loop and the LOOKUP command. The FOR-NEXT loop counts from 0 to 63, for a total of 64 loops, and it defaults to stepping one count per loop. The variable x stores the present loop count value. The LOOKUP command then takes the value of x and jumps to that many places, reads the byte value, and stores it in the “char” variable. For example, let's assume x = 5 or the sixth time through the loop since the count starts at zero. The value of “char” will equal $1C, since it is the sixth value listed.

```for x = 0 to 63
  lookup
  x,[$00,$00,$00,$10,$18,$1C,$1E,$1F,$00,$00,$00,$1F, |
  $07,$03,$01,$00,$00,$1F,$1F,$1F,$1F,$00,$00,$00, |
  $00,$00,$00,$1F,$1F,$1F,$1F,$1F,$1F,$1F,$1F,$1F, |
  $1F,$1F,$1F,$00,$00,$00,$00,$00,$00,$00,$00,$00], char
lcdwrite rspin\epin,outc,[char]
next```

After the code above executes, the custom characters are now in the LCD character-generator memory. The program can now call them to create the large characters on the LCD. The “main” label starts the central program loop. In the section below main, we use the LCDWRITE command to display a description of what this program will do, as shown in Figure 2. We display “Large Digits Using” by using the LCDWRITE command.

*** Initial screen with program description ***

```main
  lcdwrite rspin\epin,outc,[clear,home,scrram,"Large Digits Using"]
  lcdwrite rspin\epin,outc,[scrram + $40]
```

This next section will call up the custom characters just created, one at a time, using a FOR-NEXT loop, and will display them using the LCDWRITE command. The variable x holds a value from 0 to 7. LCDWRITE directs the LCD to display characters 0 through 7. See how easy it is to display custom characters, once they are created?

```for x = 0 to 7
  lcdwrite rspin\epin,outc,[x]
next```

We finish this block of code by displaying “Custom Characters” and “Demo in three seconds” to the display lines 3 and 4. SCRRAM +$14 is the beginning of line 3, and SCRRAM +$54 is the beginning of line 4.

`lcdwrite rspin\epin, outc,[scrram + $14, | "Custom Characters"]
lcdwrite rspin\epin, outc,[scrram + $54, | "Demo in 3 seconds"]`

Finally, we pause three seconds so you can read the display and then move on to the next section.

`pause 3000`

From here, the program creates the custom large characters using the custom 5x8 characters stored in CGRAM. I’ll just describe the digit 1 shown in Figure 3, but all the other large character sections below operate in the same manner.

The #1 character is created by placing custom characters 1 and 6 on line 1, character 6 on line 2, character 6 on line 3, and characters 5, 6, and 5 on line 4. We then pause one second, so the digit can be read. The SCRRAM is offset with values that center the 1 on the LCD.

```' *** “1” character
  lcdwrite rspin\epin,outc,[clear,home,scrram+$09,1,6, |
  scrram+$4A,6,scrram+$1E,6,scrram+$5D,5,6,5]
pause 1000```

The rest of the large digits (0 through F) are created in a similar fashion. Each is displayed, and then a final message is displayed. The final section of code displays, “Just imagine what you can do.”

```' *** Final message from program before looping back to the top ***
lcdwrite rspin\epin,outc,[clear,home,scrram, | "Just imagine what",scrram+$40,"you can do!"]
pause 3000
goto main```

With this custom character method, you can create just about anything on an LCD screen.

**NEXT STEPS**

The projects that can result from this are endless. Just remember that nothing stops you from redefining the custom characters in the middle of the program. Let’s say you want to display large characters, initially, and then later in the program want to create an animation using different custom characters.

After completing the large custom number characters, clear the LCD screen and then load new custom characters.
in CGRAM locations 0–7. From these new characters, you can create the animation. Since the custom characters load in CGRAM quickly, the person watching the display just notices a frame change from words to large digits to animation. I’ve seen custom characters that had the old Pacman character eating dots across the screen.

**CONCLUSION**

By no means am I declaring that I invented this custom-character method. In fact, there are numerous sites on the Internet that refer to creating custom characters on an LCD. Scott Edwards (www.seetron.com) even incorporates this type of large-character generation into his PIC MCU-driven serial LCD modules. If you are really creative, you can probably create a whole animated cartoon on the LCD by constantly changing the custom characters. It will potentially take a lot of memory, but most graphic programs do.

I once again used the BasicATOM chip, because of the simplicity of the software and the low cost for any reader that wants to follow along by doing the projects. I have received many emails from readers asking me to pick a platform and stick with it. Because Basic Micro (creator of the BasicATOM) also has a Basic compiler — which is called the MBasic Professional compiler — this platform offers the reader the option to start cheap with the BasicATOM software via a free download and a $20 BasicATOM chip (which should be cheaper by the time you read this). Eventually, most people will move to programming blank PIC MCUs, to save money on larger-volume projects.

Here’s a tip for readers: There is a book available in the Nuts & Volts Hobbyist Store (www.nutsvolts.com) called *Programming the PIC Microcontroller in MBasic* by Jack Smith (this has a similar title and exactly the same publisher as my PICBASIC compiler book). Jack does a great job of detailing how to use the PIC16F877A MCU with MBasic Pro. Best of all, his book’s CD includes a free version of the MBasic Pro compiler, which is limited to working with the PIC16F876(A). This offers you the option to take a 28-pin BasicATOM chip design and move it pretty easily to a PIC16F876A — all for the cost of a book and a programmer (to load the .hex file into the PIC16F876A). Check it out.

Feel free to email me with your comments at chuck@elproducts.com, and thanks for all of the feedback I continue to get. I do like reading the feedback and try to respond to all of the emails as quickly as possible. See you next month. **NV**
**It’s Back To School Time**

Providing Fun Electronics Learning Kits For Over 3 Decades!

---

**Did You Know...**

- **The Solution!**

Visit [www.ramsseykits.com](http://www.ramsseykits.com)

Free Giveaways and ALL the information you ever needed!

---

**Tickle-Slick Shocker**
The kit has a pulsing 40 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.

**Super Snoop Amplifier**
Super sneaky amplifier that will pick up a pin drop at 15 feet! Full 2 watts output drives any speaker for a great sound. Makes a great bug ear microphone to listen to the “wildlife” both in the field and in the city! Runs on 6-15 VDC.

**LED Blinky**
Our #1 Mini-Kit for 35 years! Alternately flashes two jumbo red LED’s. Great for signs, name badges, model railroad, and more. Used throughout the world as a fun learning kit for students young and old! Great solder practice kit. Runs on 3-15 VDC.

**Universal Timer**
Build a timer delay something on for a preset time, prevent clock pulses or provide an audio tone, all using the versatile 555 timer. Comes with circuit theory and a lots of application ideas and schematics to help you learn the 555 timer. Runs on 5-15 VDC.

**Voice Activated Switch**
Voice activated (VAD) provides a switch output when it hears a sound. Great for a hands free PTT switch! Or to make an LED blink! Directly switches relays or low voltage loads up to 100mA. Runs on 6-12 VDC.

**RF Preamplifier**
The famous RF preamp that’s been written up in the radio & electronics magazines! This looking preamp covers 100 KHz to 1000 MHz! Unconditionally stable gain is greater than 40dB! 50-75 ohm input. Runs on 12-15 VDC.

**Mad Blaster Warble Alarm**
If you need to simply get attention, the “Mad Blaster” is the answer, producing a LOUD ear shattering raucous electronic alarm and home alarms as well. Drives any speaker. Runs on 9-12VDC.

**Water Sensor Alarm**
This little 58 kit can really “bail you out”! Simply mount the alarm where you want to detect water, and connects the one end to the remote first! Runs on a standard 9V battery.

**Power Saver Timer**
Add time delay power-off to any of your projects “just in case”. Momentary push button selects desired time delay (1H, 2H, 4H, 8H, 24H) and LED flashes faster as time approaches. On-board 10A relay. Runs on 120/240VAC.

**Steam Engine & Whistle**
Simulates the sound of a vintage steam engine locomotive and whistle! Features variable engine speed and volume. Whistle Blows at a touch of a button! Great for model train setups. Includes the speaker. Runs on a standard 9V battery.

**SMT-Multi-Color-Blinky**
The ultimate blinky kit! The 8-pin microcontroller drives a very special RGB LED in 16 million color combinations! Uses PWM methods to generate any color with the micro, with switchable pulse selection. SMT construction with extra parts when you lose them! Runs on 9V battery.

---

**Electronic Siren**
Exactly duplicates the upward and downward wall of a police siren. Switch closure produces upward, releasing it makes it return downward. Produces a loud SWISH Note! and will LED’s show open! Horn speakers sound the best! Runs on 6-12 VDC.

**Walking Electronic Bug**
Built around a pair of subminiature cell phone motors, this bug wanders around looking things to “sting” into! Sensors below his LED eyes sense proximity and make him turn away! Steer him with flashlights too! Runs on two N” batteries.

**SMT High Intensity Blinky**
Cools sound the best! Runs on 6-12 VDC.

---

**It’s Not Possible to give you full specs on these products in a 1” space!**

**A lot of our kits are also available “factory assembled and tested” if you don’t want to build it!**

**We have over 350 products currently available, and all those don’t fit here!**
Learn and build!

AM/FM IC Radio & Training Course
- Learn all about AM/FM radio design and theory!
- Build a high-performance radio with an IC!
- Step-by-step assembly and test procedures
- Great for student projects!

This Superhet 108 AM/FM Radio Kit is a "superheterodyne" receiver of the standard AM (modulation) and FM (frequency modulation) broadcast frequencies. The unique design of the Superhet 108 includes an integrated circuit (IC), 9 transistors, 4 diodes, and other related resistors, capacitors, coils, and misc. parts. Because good soldering practices are important, a soldering guide is also provided as well as a component color code and pictorial identification chart.

The actual assembly is broken down into 9 sections. The theory of operation for each section, or stage, should be read before the assembly is started. This will provide the student with an understanding of what that stage has been designed to accomplish, and how it actually works. After each assembly you will be instructed to make certain tests and measurements to prove that each section is functioning properly. If a test fails to produce the proper results a troubleshooting guide is provided to help you correct the problem.

After all of the stages have been built and tested a final alignment procedure is provided to peak the performance of the receiver and maximize the Superhet 108’s reception capabilities. Runs on a standard 9VDC battery, not included. (This is today’s version of the “5 tube superhet radio I built in Mr. Baehre’s 9th grade electronics shop class!)

AM/FM Integrated Circuit Radio & Training Course $34.95

SMT Soldering Course & Lab
- Learn all about those tiny little parts!
- Develop and practice your SMT soldering skills!
- Includes SMT training sessions
- End up with a neat "Decision Maker"

Surface Mount Technology (SMT) has become a way of life for electronics. Virtually all new high tech products utilize SMT design and components. Many of our latest Ramsey kits include various SMT components. There are two problems with SMT: First, the parts are extremely small, at times the size of a pin head! Therefore, you need to have the proper tools, not only to hold them and solder them, but to see them! (Check out the tools section at ramseykits.com for help!) Secondly, unlike "through hole" parts, SMT components are soldered directly to the surface of the PC board (hence the name SMT). This is an entirely new method of soldering, and it takes practice to get it right. Trust me. I’ve created a number of solder bridges myself when I first started soldering SMT components!

Taking this course and practicing on the neat little kit will prepare you for the future in electronics. Includes the PC board, all the SMT and other components required, and a comprehensive training course covering all aspects of SMT soldering, desoldering, removing, and component placement. The finished project is a neat little "Decision Maker" that includes 6 flashing SMT LEDs, complete with a professional overlay panel. The finished Decision Maker runs on a standard 9VDC battery, not included.

SM200K Practical SMT Soldering Course & Lab $20.95

Electronic Learning Labs

PL130 130 In One Learning Lab Kit $42.95
PL200 200 In One Fun Learning Lab $49.95
PL300 300 In One Advanced Learning Lab Kit $69.95
PL500 500 In One Super Learning Lab Kit $169.95

Non-Conductive Tweezers
- Great for SMT and small parts
- ESD safe!

We’ve discussed tweezers and magnifiers while working with SMT components. Working with some highly sensitive components our engineers asked for some ESD safe tweezers. We looked, we found, and they are a life saver! This set of 4 non-conductive tweezers are perfect for any static sensitive devices you may be working on. And...they are priced right!

VTTW5ET2 Non-Conductive Tweezers Set $3.95

Get The Catalog!

Get the latest 2007 Ramsey Hobby Catalog! 96 value packed pages of the newest goodies around! Order yours on-line today or give us a call! Or download the PDF at www.ramseykits.com.
BY L. PAUL VERHAGE

SOME SENSORS FOR YOUR BALLOONSA TS AND THE GREAT PLAINS SUPER LAUNCH

NOW THAT YOU HAVE A BalloonSat flight computer (see the November ‘06 issue of *Nuts & Volts*), what are you going to measure with it? Well, this month we’ll look at two sensors: a miniature near space weather station and a temperature sensor array. With these sensor arrays, your BalloonSat can chart environmental conditions from the ground to near space and measure the temperatures of your experiments. Afterwards, I’ll give you a short report on the Great Plains Super Launch. So, fill your balloon with helium and let’s get started.

NEAR SPACE WEATHER STATION

The miniature weather station measures air temperature, pressure, and relative humidity along with the internal temperature of the BalloonSat (Figure 1). I’ve flown this sensor array on several BalloonSats and I’m very happy with the way it functions. To fly as a BalloonSat experiment, the weather station must be small and lightweight (because BalloonSats are typically limited to one pound of weight). I like the weather station for its compact size and also for its affordability (around $50, as I recall). Now, the only thing that this weather station can’t measure is the wind speed and direction. However, since balloons are forced to move with the wind, the GPS receiver on board the near spacecraft that is carrying the BalloonSat will fill in the wind data. Aside from near space use, this miniature weather station could be part of a traditional home weather station or part of a Mars or Titan model rover.

Most of these items are available from electronics companies like Jameco and they don’t need further explanation. But we will take a closer look at the sensors at the heart of the miniature weather station.

The LM335 is a voltage controlled zener diode in a TO-92 form factor. So, it looks like a small transistor (e.g., the 2N3904). The LM335 drops a voltage that’s proportional to its temperature. Specifically, it drops one volt per one hundred kelvins and (ideally) zero volts at zero kelvins. The kelvin is equivalent to a degree Celsius, but the Kelvin temperature scale begins at absolute zero – the coldest temperature possible. You can download the datasheet for the LM335 at the National site ([www.national.com/pf/LM/LM335.html](http://www.national.com/pf/LM/LM335.html)).

The HIH4000 is a three pin SIP relative humidity sensor that is manufactured by Honeywell. The sensor produces a voltage that is linear with respect to relative humidity. At 0% RH, the sensor produces an output of 0.8 volts and it maxes out at 100% relative humidity with a voltage of 3.86 volts. That makes its voltage output

**NEAR SPACE WEATHER STATION PARTS LIST**

- Two LM335 temperature sensors
- SM5812 absolute pressure sensor (0–15 PSI)
- HIH-4000 (or the equivalent older HIH-3610) relative humidity sensor
- Two 1K 1/4W resistors
- 0.1 µF capacitor
- Weather station printed circuit boards (see the diagram in this month’s column)
- 24 AWG stranded wire
- Headers (0.1” spacing)
- Receptacles (0.1” spacing)
- Thin heat shrink

FIGURE 1
0.0306 volts per percent of relative humidity. Exercise caution when using this sensor and do not touch its face; moisture and skin oil can damage the sensor. You can find the datasheet for this sensor at Phil Anderson’s website (www.phanderson.com/hih4000.pdf). I found a distributor for this sensor at Newark (www.newark.com).

The SM5812 is a silicon micro-electric mechanical system (MEMS) based pressure sensor. Inside the sensor is a vacuum sealed “cup” with a thin silicon lid. As the air pressure on the sealed cup changes, the stress of the silicon lid also changes. Stress in the silicon lid changes its resistance and therefore, the voltage dropped across it. The circuitry in the SM5812 calibrates and amplifies the output voltage of the sensor. The sensor produces 0.5 volts at zero PSI of pressure and 4.5 volts at 15 PSI.

Between the two extremes, the output voltage is linear with respect to pressure. You can find more information on this sensor at the Silicon Microstructures website (www.si-micro.com) under the SM58 product series. The SM5812 is available from Servoflo Corporation (www.servflo.com). Since Servoflo is not ready to take online orders, call them at (866) 830-9572 to order a SM5812.

ASSEMBLING THE MINIATURE NEAR SPACE WEATHER STATION

You’ll notice that the second temperature sensor (for the interior of the BalloonSat) is mounted on a separate printed circuit board (PCB). Since the rest of the weather station is mounted outside of the BalloonSat, the BalloonSat airframe needs an opening to pass the cable between the two PCBs. To keep this opening in the BalloonSat airframe small, I designed the external portion of the weather station to be connected to the internal portion with a header and receptacle.

There’s nothing difficult about assembling the miniature near space weather station. However, I would still recommend placing lowest lying components first, but there are no tight fitting components that require you do it this way. The additional holes in the PCB’s ground plane are strain relief for the wires in the weather station cables. So, pass wires through those holes before you solder them to the PCB. I would recommend making the power wire red and the ground wire either green or black.

Cut the five wires that you soldered to the external PCB to a length of six inches. The five wires terminate with a 1x5 header to form a unified cable that connects the external weather station to the internal temperature sensor through its receptacle. I prefer to make my cable by tinning the short pins of the 1x5 male header and the 1/4 inch long bare ends of the wires. Then slide thin heat shrink over the ends of the wires and press a tinned wire to a tinned header pin. When the pin and wire are heated, the solder in each fuses together. Remove the heat and slide the heat shrink over the soldered wire and header. Now you only have to repeat this four more times.

The internal temperature sensor board has its own cable that plugs the weather station into the BalloonSat’s ADC channels. I make this cable the same way as the last one (except there are six wires in this cable). If you
have a three row header, use it for the cable. If you don’t — like me — then use two 2x3 headers to terminate the I/O cable. Again, it would be a good idea to use color coded wires for this cable (Figure 4).

The weather station’s external section needs to be protected from direct exposure to sunlight. That’s because direct sunlight affects the outputs from the relative humidity and temperature sensors. For my weather station, I bolted this section to the inside of a short length of cardboard tube (Figure 5). In a second weather station I made, the housing is made from Correplast plastic, hot glue, and white tape.

Whatever the housing, I recommend coloring it white to reflect as much sunlight as possible; you don’t want a dark housing changing what the LM335 measures as the air temperature. Be sure to leave the housing open so that air can flow across the LM335 temperature sensor.

**OUTPUT FROM THE NEAR SPACE WEATHER STATION**

Assuming the weather station is connected to the BalloonSat Flight Computer’s eight-bit ADC, here are the Excel formulas needed to calculate temperature, pressure, and relative humidity.

**Temperature Equation**

\[ \text{Temperature} = (((((X1/255)*500)-273)+40)*1.8)-40 \]

where X1 is the cell address of the ADC’s temperature value loaded in the spreadsheet.

You’re probably asking what is going on with this formula. The formula converts the eight-bit ADC reading into a percentage of the ADC range (0 to 255) then into a temperature in units of kelvins by multiplying by 500. Next, 273 is subtracted to convert the value from the Kelvin scale to the Celsius scale. Now the Celsius and Fahrenheit temperature scales have the same temperature at -40°. So -40°F and -40°C are the same temperature. If you were to chart the Celsius and Fahrenheit temperature scales relative to the Fahrenheit scale and shift the zero degree point back 40°, there would be no Y-intercept in the two lines; they would only vary by their slope. The formula takes advantage of this relationship and adds 40° to the Celsius reading to remove the Y-intercept and allow us to multiply (or divide if you want to go the other direction) the Celsius temperature by the difference in the slope between the Fahrenheit and Celsius scales (there are 1.8 Fahrenheit degrees in a Celsius degree). Then, 40° is subtracted from the result to shift the result properly back to the Fahrenheit scale. I like this method of converting between Fahrenheit and Celsius because the concept is so simple.

**Pressure Equation**

\[ \text{Pressure} = ((X1-25.6)/204.8)*1013 \]

where X1 is the cell address of the ADC’s pressure value loaded in the spreadsheet.

Since the pressure sensor bottoms out with 0 PSI at 0.5 volts, the formula subtracts 0.5 volts from the ADC reading (25.6) then divides the result by the maximum swing in ADC readings (204.8) to calculate the air pressure relative to mean sea level pressure. The ratio is multiplied by 1013 — sea level atmospheric pressure in millibars — to calculate the pressure in units of millibars. Alternatively, the 1013 can be replaced by 14.7 to calculate the pressure in units of pounds per square inch.

**Relative Humidity Equation**

\[ \text{Relative Humidity} = ((X1-41)/155)*100 \]

where X1 is the cell address of the ADC’s relative humidity value loaded in the spreadsheet.

Since the relative humidity sensor produces 0.8 volts at 0% relative humidity, 41 must be subtracted from the ADC value. The highest relative humidity reading is 3.86 volts, or 196. So, the result from the initial subtraction is divided by 155 (196 minus the initial
value of 41) and then multiplied by 100% to calculate the relative humidity.

Once you’ve finished editing your data in Excel or other spreadsheet, you need to include a column for altitude. You can look at the TNC log from the mission to get this information or calculate it by determining the climb rate of the near spacecraft and multiplying by mission elapsed time. When you’ve added the altitude to the spreadsheet, you can generate charts like the ones in Figures 6 through 9 (these charts come from a mission launched in late 2006).

The three I like best are Figures 7, 8, and 9 (the pressure seems boring to me). So let’s take a closer look at each of them. Notice that the air temperature decreased with increasing altitude until the BalloonSat reached an altitude of 50,000 feet. From there, the air temperature rises with increasing altitude.

The air temperature cools in the troposphere and rises in the stratosphere. The transition between the two is called the tropopause. The troposphere cools with increasing altitude primarily because the balloon is moving away from its major source of heat — the earth’s surface. The stratosphere, on the other hand, warms with increasing altitude because the balloon is moving closer to its primary source of heat — the sun.

Ozone in the stratosphere blocks some ultraviolet radiation and converts it into thermal energy. So it is the presence of ozone that we’re detecting above 50,000 feet when we measure an increasing temperature.

In summer time, I typically see the tropopause at 50,000 feet and with a temperature of -60 degrees Fahrenheit. In the winter, I typically see it lower to 40,000 feet and drop down to -90°. The latitude does affect the altitude and temperature of the tropopause.

The relative humidity chart in Figure 7 shows us that the air gets drier with increasing altitude. But notice that the relative humidity spiked three times at 8,000, 12,000, and around 35,000 feet. I don’t recall the weather conditions on this flight, but I suspect there were clouds at these altitudes.

The temperature chart in Figure 9 shows that the interior of the BalloonSat stays significantly warmer than the outside air. The airframe of this particular BalloonSat was 1/2 inch thick Styrofoam. At 80 minutes mission elapsed time, the balloon burst. The movement of air over the BalloonSat as it fell chilled the external temperature sensor and cooled the interior of the BalloonSat.

### Temperature Sensor Array

The last sensor I want to discuss this month is an array of temperature sensors (Figure 10). Many years ago, I became curious about how the color of my near spacecraft would affect its internal temperature. To find out, I created four identical foam blocks and covered them in materials with different colors (black, white, silver, and light blue). To reduce variations in the experiment, I flew all four cubes on the same mission. That meant I needed four temperature sensors for the mission. Instead of building a temperature sensor for each cube, I designed the following circuit board.

**ASSEMBLING THE TEMPERATURE SENSOR ARRAY**

There’s no preferred order when soldering the array together. Do note

---

**Temperature Sensor Array Parts List**

- Four 1K resistors
- Four LM335 precision temperature sensors
- Thin gauge wire
- A 3 x 4 male header (or equivalent)
- Thin heat shrink
- Printed circuit board (see the copper pattern on the Nuts & Volts website at [www.nutsvolts.com](http://www.nutsvolts.com))
though that each LM335 is soldered to the ends of three wires. The cable allows each LM335 to be routed to experiments that are remote from the PCB. Here’s how I solder a LM335 to the end of a cable:

1) Cut the leads of a LM335 to half their length and tin.
2) Strip 1/4 inch of insulation from the three wires in the cable and tin.
3) Slide 1/2 inch long heat shrink tubing over the wires in the cable.
4) Press the LM335 lead against the bare end of a wire and heat with a soldering iron.
5) After the solder in the LM335 lead and wire fuse together, remove the heat and let cool.
6) Slide heat shrink over the exposed solder joint and shrink.

This process beats twisting and soldering the wires together and it’s strong enough for the LM335s. One thing however, be sure you have the LM335s in the correct orientation before soldering its leads to the wires in the cable. It won’t work if it’s backwards (like most things in life).

To keep the cable neat and tidy, twist the wires in each cable together. Notice that only two wires are really needed to solder the LM335 to the PCB. I recommend using all three wires to make the cable stronger.

After assembling a cable, wrap its temperature sensor with a small sticky label that has the sensor’s channel printed on it. Then shrink clear heat shrink tubing over the label to protect it. You’ll appreciate seeing these labels on your sensors when you’re putting your experiment together. By the way, be sure to write down the channel of each temperature sensor and its experiment during assembly. You’ll need that information when you put the spreadsheet together.

The last cable is the interface to the ADC of the flight computer. Terminate it just like the other sensors in this article with a 3 x 4 male header and heat shrink. If you don’t have access to three row male headers, then use two 2 x 3 headers to terminate it.

Since the temperature sensor array uses the same temperature sensor as the miniature near space weather station, use the same Excel formula.

The chart in Figure 12 is an example of the data that I’ve collected from the temperature array sensor. The temperature probes were routed to four cubes: one was a plain blue Styrofoam; the second, a Styrofoam cube covered in black construction paper; the third, a cube covered in multilayer insulation (MLI — see note
at end of this month’s column); and the last, a cube covered in MLI and a black jacket of construction paper. As you can see, they cooled close to the same rate initially. But as they neared the stratosphere after 35 minutes of flight, their temperatures deviated widely from one another. At balloon burst (83 minutes into the mission), they chilled to nearly the same temperature after 10 minutes.

THE GREAT PLAINS SUPER LAUNCH OF 2007

The largest amateur radio, high altitude balloon (ARHAB) event is the Great Plains Super Launch (GPSL). Eleven weather balloons were launched on July 7th in Grand Island, NE, the location of GPSL 2007. The Central Nebraska Near Space Project (CNNSP) hosted the two-day conference this year in their home town. Friday was spent in discussions, with several near space groups giving presentations on the art and science of their near space programs. There were discussions on air density and its effect on a balloon’s coefficient of drag, flight computers, balloon release mechanisms, and meteorology.

Along with the presentations there was lots of socializing, especially over lunch and dinner. The new attendees were presented with a ton of information. One of this year’s sponsors was Nuts & Volts Magazine (thanks guys). But the best was Saturday morning. That’s when we filled and launched our near spacecraft.

DePauw and Taylor Universities were there along with amateur groups from Kansas, Nebraska, Idaho, Colorado, Oklahoma, and Maine. The balloon filling took place early in the morning so the balloons could be launched before the surface winds picked up.

For additional insurance though, we filled the balloons behind a row of organic wind blockers (otherwise known as tall trees in Nebraska). I really have to hand it to Roger and CNNSP for the great job they did finding a conference center and launch site. Figure 13 shows a photograph from GPSL 2007.

Every near spacecraft was recovered after they landed except for one. We lost contact with its tracking capsule shortly after lift off. I had to leave for Idaho just after lunch on Saturday, but most of the groups stayed around for an informal dinner on Saturday evening. I hear this dinner was very successful and will be included in GPSLs in the future.

There’s one more super launch planned for this year and several for 2008. The next GPSL will be held by Near Space Ventures in Kansas City, MO. You can find the date and other information at the Super Launch website at http://superlaunch.org. I hope to see you next year at one of the super launches.

Onwards and Upwards
Your Near Space Guide

BRIEF NOTE ABOUT MULTILAYER INSULATION (MLI)

One of the best insulators is the Dewar flask (which is the same thing as a thermos bottle). Usually though, spacecraft cannot carry a glass thermos to insulate themselves. So they’re covered in a fabric version of this called MLI. I make my MLI with three alternating layers of space blanket (aluminized Mylar) and scrim (plastic wedding veil material). In a good hard vacuum, this stuff will work great. But in near space with its near vacuum, it’s not nearly as effective. But it still beats launching a glass thermos bottle on a balloon.
Intelligent Display Engine

IntelliLCD

- 7" Wide TFT Color Touch Screen
- Supports TrueType Windows Fonts
- RS232 Interface
- 800 x 480 resolutions
- 260k Colors
- SD Card Support
- BMP File Support
- LED Backlight
- USB PC sync, keyboard, mouse
- Display Commands: Line, Circle, Box, Print...
- Control Commands: Button, Slider, CheckBox...

$399 qty1

COMFILE TECHNOLOGY

www.comfiletech.com
Toll-Free: 1.888.928.2562

CHINA PCB SUPPLIER

DIRECT SALE FROM CHINA

PROTOTYPE TO PRODUCTION

instant online quote
shopping cart ordering system
China competitive prices
free Electrically test

web: http://www.pcbcart.com
email: sales@pcbcart.com
Tel: +86-571-87013819
Fax:+86-571-87038705
Add: No.2 Huixing Road, Hangzhou, China

www.pcbcart.com

Product Development
Industrial Design
Electrical & Mechanical Engineering
Prototyping and Manufacturing

1,600+ Products
Free Forums and Resources
www.DesignNotes.com

Project Cases
As Low As $4.75

Connectors & Adapters
As Low As $0.28

100's of Kit Projects
PIC Programmer
& Experiment Board
Only $37.75

Test Equipment
USB PC 2 CH
60MHz Oscilloscope
$329.99
+ free gift

DesignNotes.com
Affordable Prototyping Products

Toll Free
1-800-957-6867
15 Day Money Back Guarantee
DesignNotes.com, Inc.
2014 Ford Road, Unit B • Bristol, PA 19007
We’re semi nuts

We’ve got semis on the brain. Jameco offers more major brands of semiconductors than anyone — almost twice as many as these catalog distributors.* It’s another Jameco advantage.

<table>
<thead>
<tr>
<th>Mouser®</th>
<th>Newark®</th>
<th>Allied®</th>
<th>Jameco®</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atmel Semiconductor</td>
<td>Altera</td>
<td>Analog Devices</td>
<td>Altera</td>
</tr>
<tr>
<td>Avago Technologies</td>
<td>Analog Devices</td>
<td>Avango Technologies</td>
<td>Analog Devices</td>
</tr>
<tr>
<td>Cypress Semiconductor</td>
<td>Avago Technologies</td>
<td>Cypress Semiconductor</td>
<td>Avango Technologies</td>
</tr>
<tr>
<td>Diodes, Inc.</td>
<td>Cypress Semiconductor</td>
<td>Fairchild Semiconductor</td>
<td>Fairchild Semiconductor</td>
</tr>
<tr>
<td>Freescale Semiconductor</td>
<td>Freescale Semiconductor</td>
<td>Integrated Devices</td>
<td>Freescale Semiconductor</td>
</tr>
<tr>
<td>Intersil</td>
<td>Intel Corporation</td>
<td>Intersil</td>
<td>Intersil</td>
</tr>
<tr>
<td>Lattice Semiconductor</td>
<td>Lattice Semiconductor</td>
<td>Maxim</td>
<td>Lattice Semiconductor</td>
</tr>
<tr>
<td>Lite-On Semiconductor</td>
<td>NXP (formerly Philips)</td>
<td>National Semiconductor</td>
<td>Maxim</td>
</tr>
<tr>
<td>NEC Corporation</td>
<td>Maxim</td>
<td>National Semiconductor</td>
<td>Integrated Devices</td>
</tr>
<tr>
<td>Sharp Microelectronics</td>
<td>National Semiconductor</td>
<td>NXP (formerly Philips)</td>
<td>Maxim</td>
</tr>
<tr>
<td>ST Microelectronics</td>
<td>ST Microelectronics</td>
<td>NXP (formerly Philips)</td>
<td>Maxim</td>
</tr>
<tr>
<td>Texas Instruments</td>
<td>Texas Instruments</td>
<td>ST Microelectronics</td>
<td>Texas Instruments</td>
</tr>
</tbody>
</table>

We’re semi nuts
We’ve got semis on the brain. Jameco offers more major brands of semiconductors than anyone — almost twice as many as these catalog distributors.* It’s another Jameco advantage.

<table>
<thead>
<tr>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freescale Semiconductor</td>
<td>Integrated Devices</td>
<td>Intel Corporation</td>
<td>Intersil</td>
<td>Lattice Semiconductor</td>
</tr>
<tr>
<td>Intersil</td>
<td>Lattice Semiconductor</td>
<td>Maxim</td>
<td>Maxim</td>
<td>Maxim</td>
</tr>
<tr>
<td>Lattice Semiconductor</td>
<td>Maxim</td>
<td>National Semiconductor</td>
<td>National Semiconductor</td>
<td>National Semiconductor</td>
</tr>
<tr>
<td>Lite-On Semiconductor</td>
<td>Maxim</td>
<td>National Semiconductor</td>
<td>National Semiconductor</td>
<td>National Semiconductor</td>
</tr>
<tr>
<td>Maxim</td>
<td>Maxim</td>
<td>National Semiconductor</td>
<td>National Semiconductor</td>
<td>National Semiconductor</td>
</tr>
</tbody>
</table>

**Free shipping on these and 79 other brands. Call for details.**

**OTHER JAMECO ADVANTAGES:**
- More major passive, interconnect and electro-mechanical brands than other distributors.
- 99% of catalog products ship the same day.
- Lowest prices guaranteed, or we pay 10%.
- Major brand names and generic equivalents for even greater cost savings.

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

©Jameco Electronics. *According to their web sites on July 23, 2007. Trademarks are the property of their respective owners.
Class A/B Amp

Q: Can you tell me how the AC and DC analysis are done on the class A/B transistor amplifier (Figure 1) and answer these questions?

1) Why is the signal input between the diodes?

2) Does the signal voltage affect the DC bias?

3) How can I use loop equations on this circuit configuration?

— Sal Bivona

A: This is a non-linear circuit, so you can’t do an accurate analysis without knowing the resistance values and voltages. However, you can linearize the circuit by assuming a constant 0.6 volt for the diode drop and Vbe. Then, Ve1 = Vin and the solution is trivial except that at some point, as Vin increases, the current through D1 will become zero and the output will be constant (the AC output will be clipped).

The signal input is between the diodes because it is a symmetrical circuit and you want a symmetrical output signal. The presence of input signal will increase the current output of the circuit but since DC bias is defined as the condition in the absence of signal, the question is a non-sequitur.

Since the circuit is symmetrical, analysis of the NPN circuit will also apply to the PNP circuit. The only question of interest is: Where will the output be clipped? Let us find that point.

- Since Id1 = 0, then
  \[ I_b1 = (Vcc - Ve1 + .6)/R1 \]
- Also, \( Ve1 = (Ib1 + Iq1)(R3+R5) \)
- And, \( Iq1 = \beta * I_b1 \)
- This leads to: \( Ve1 = [(Vcc - Ve1 + .6)/R1] \cdot (\beta + 1)^* (R3 + R5) \)
- To simplify, let \( K = (\beta + 1)^* (R3 + R5)/R1 \)
- Then: \( Ve1 = (Vcc+6)^*K - Ve1*K \)
- Therefore the clipping point is: \( Ve1 = (Vcc+6)^*K/(K+1) \)

60 Hz Pickup Circuit

Q: I have a need for a circuit which will pick up the 60 Hz power line signal as radiated and generate a zero-cross signal.

The circuit is battery powered so it is not connected to the AC mains. The circuit’s main source is a solar-panel array. I would expect an AM-ferrite type antenna and op-amp or FET device could be used.

— Paul KJ4UO

A: You don’t say how far from the power line you are. At any rate, the AM radio antenna that is designed for MHz will not be efficient at 60 Hz. While it is true that the US Navy transmits 10 kHz signals long distances, it requires a massive antenna. The 60 Hz transmission lines are designed not to transmit, so any sig-
nal will be down in the noise of lighting and automotive ignition. A better solution is to generate an accurate 60 Hz from a crystal as shown in Figure 2.

The crystal can be pulled a little so if the frequency is off, you can increase C1 and C2 to lower the frequency or decrease them to increase the frequency. The crystal tolerance is typically 50 ppm. I divided the crystal frequency down to 120 Hz then divided by two because the 4060 does not have a Q11 output. The added advantage is that the output is a symmetrical square wave.

The procedure for finding the divider outputs to decode the frequency is this: Divide the input frequency by the output frequency to find the division factor (N). This must be a whole number. Find the highest power of 2 that can be subtracted from N. Do the subtraction; the result being a new value for N. Repeat until the result is zero. The powers of 2 that you use are the Q outputs needed to decode.

**LINEAR RAMP**

**Q** I enjoyed T.J. Byers' column for years and am delighted that you are carrying on that quality with your column.

I have a question that could have applications in many circuits and so may be of interest to a variety of readers.

I am controlling a motor's speed based on a reference voltage: where five volts equals no speed, eight volts equals full speed forward, and two volts equals full speed in reverse. I need a simple circuit to provide a linear acceleration/deceleration ramp where an instant voltage change at the input to the circuit (from say five volts to eight volts) would produce that same change at the output of the circuit, in the range of five to 10 seconds.

It is important to me that this ramping up (or down) be linear. Simply charging a capacitor through a resistor won't do. I would prefer to use simple gates or op-amps and/or discrete components, as I don't get along well with microprocessors or any of their relatives.

Any help or suggestions would be most appreciated.

— Clark W. Kuhl

**A** The circuit in Figure 3 provides the linear ramp that you want. The B section of the op-amp provides a five volt reference so the motor can be reversed. You could use a five volt power supply instead. A five volt change in the output of the A op-amp, caused by a three volt input, takes 5.5 seconds. You can increase the time by increasing R4 or using a larger capacitor (C1).

This circuit cannot maintain zero speed for very long because it is not possible to match the reference voltage perfectly. The speed will drift higher over time in one direction or the other. Some kind of tachometer feedback would be needed to maintain zero speed.

**FLY ZAPPER**

**Q** I am looking for a small size fly zapper, but I can't find one. It is either a big commercial unit, or it uses some scented refills.

How hard would it be to make...
one? Say, you start with a cheap carriage lantern from a home improvement store, remove the glass and the guts, install a UV LED, make two cylinders out of wire mesh (one inside the other), and then hook them up to a high voltage DC-DC inverter. Would it work?

Once the fly or mosquito gets between “the fences,” it would get zapped by a spark and fall down. This is called a non-clogging zapper, since the fly should not get baked to the grid.

There may need to be some calculations done on what the voltage should be at different spacing and grid sizes of the fence (bigger flies, small mosquitoes, or both) and UV LED would need to be covered somehow so it cannot be looked directly into.

— Dusan

Your idea is good. Get some hardware cloth from the local store and form it into two cylinders, one 3/4 inch smaller in diameter than the other. To maintain the spacing, use 3/8 inch wood dowels impregnated with wax. Thirty thousand volts are needed to arc 3/8 inch at sea level. Adjust the voltage (see Figure 4) so the arcing is only occasional. Connect the outer screen to a spike in the ground to insure that curious fingers don’t get zapped.

In the circuit in Figure 4, R3 adjusts the duty cycle of the 555 to change the voltage output. The nominal frequency is 20 kHz, but varies with duty cycle. I chose a 500 volt, 14 amp transistor which may be overkill, but I don’t want the transistor to smoke when you are making adjustments. I chose the value of R1 by guesstimating; the transistor drain has to go to 300 volts in order to get 30,000 volts output. If you don’t get enough output with R3 at maximum, make R1 larger. R1 and C1 just provide an upper limit to the drain voltage. R1 will be dissipating nine watts so a cement filled, 10 watt resistor would be appropriate. All other resistors are 1/4 watt. I expect that Q1 will operate without a heatsink because the normal current should only be a couple of amps.

The voltage drop for blue LEDs is about 3.5 volts so you can have two in series and two in parallel as in Figure 5.

A

I have several Greenlee punches. I don’t remember what I paid for them but I assure you that I did not pay hundreds of dollars! When I make square holes, I use your method: Drill a hole in the center, cut the outline with a coping saw or hacksaw and file to fit. I have flat and square files for that purpose. I found a URL that shows how to make rectangular holes using several methods, including using a chisel or punch: [www.makezine.com/pub/a/extras/15.html](http://www.makezine.com/pub/a/extras/15.html?page=last&amp;x-maxdepth=0).

— C. P. Furney, Jr.

A

I have a very simple question. I build a lot of simple nifty electronic gadgets, many from Nuts & Volts and usually put them in plastic project boxes. I have never figured out how to cut a clean rectangular or square hole in one of these boxes. Usually, I need round holes which are easily cut with a hardware store type step drill, but square or rectangular holes are needed for switches, transistor sockets, and meters to name a few.

I lay out the hole, then drill four small round holes at the corners, and then try to saw or cut out the hole with my Dremel tool. It usually ends up a ragged mess. Stores want $500 for a punch. Isn’t there a better way?

— C. P. Furney, Jr.
PRINTED CIRCUIT BOARDS

QUALITY PRODUCT
FAST DELIVERY
COMPETITIVE PRICING

- Aluminum Backed PCB
- Single & Double sided
- SMD/B Richards
- LPI mask
- Through hole or SMT
- Nickel & Gold Plating
- Routing or scoring
- Electrical Testing
- Artwork or CAD data
- Fast quotes
- Flex Circuits

We will beat any competitor’s prices!!!

www.pulsar-inc.com - sales@pulsar-inc.com

10 pcs (3 days)
$249.98
1 or 2 layers $249
10 pcs (5 days)
4 layers $695
(up to 30 sq. in. ea.)
- includes tooling, artwork, LPI mask & legend

10 pcs
1 or 2 layers
$695
4 layers
$895

Lakeview Research LLC
www.lvr.com

ActiveWire® USB
Simple USB Interface!!

$59
plus shipping

ActiveWire, Inc.
www.activewire.com
ph +1.650.465.1000 fax +1.209.301.5606

USB Oscilloscope for $169.50
- 6-8 digital + 2 analog channels
- Oscilloscope - generator - Logic Analyzer
- with UART, SPI, I2C and 1-Wire interfaces

Great for PIC and AVR projects!

www.HobbyLab.us

Add USB to Your Projects
USB Mass Storage: Designing and Programming Devices and Embedded Hosts
ISBN 1-931448-04-3
287 pages $29.95

USB Complete: Everything You Need to Develop Custom USB Peripherals
ISBN 1-931448-02-7
572 pages $49.95

Lakeview Research LLC
www.lvr.com

USB Oscilloscope

- 6-8 digital + 2 analog channels
- Oscilloscope - generator - Logic Analyzer
- with UART, SPI, I2C and 1-Wire interfaces

Great for PIC and AVR projects!

www.HobbyLab.us

Add USB to Your Projects
USB Mass Storage: Designing and Programming Devices and Embedded Hosts
ISBN 1-931448-04-3
287 pages $29.95

USB Complete: Everything You Need to Develop Custom USB Peripherals
ISBN 1-931448-02-7
572 pages $49.95

www.HobbyLab.us

Add USB to Your Projects
USB Mass Storage: Designing and Programming Devices and Embedded Hosts
ISBN 1-931448-04-3
287 pages $29.95

USB Complete: Everything You Need to Develop Custom USB Peripherals
ISBN 1-931448-02-7
572 pages $49.95

www.HobbyLab.us

Add USB to Your Projects
USB Mass Storage: Designing and Programming Devices and Embedded Hosts
ISBN 1-931448-04-3
287 pages $29.95

USB Complete: Everything You Need to Develop Custom USB Peripherals
ISBN 1-931448-02-7
572 pages $49.95

Lakeview Research LLC
www.lvr.com
LOW COST DDS FUNCTION GENERATOR

B&K Precision Corporation Model 4045 Programmable Synthesized Function Generator can generate sine, square, triangle, and arbitrary waveforms with crystal controlled sampling rates of up to 20 Hz, 12 bit vertical resolution, and up to 1,000 points. A menu-driven front panel operation with an easy-to-read graphic LCD display makes the model 4045 easy to operate.

Parameter changes and data entry can be made using the rotary knob or via the built-in RS-232 interface. Waveform editing can be done from scratch or by modifying standard waveforms. A built-in PC software program allows the user to easily create, edit, and download complex waveforms. Waveforms can be stored in the 4045’s Flash memory.

Programming can be remotely controlled by the built-in RS-232 interface. All parameters, modes, and functions are programmable. Priced at $995, it is a perfect instrument for R&D, manufacturing test systems, teaching arbitrary waveform fundamentals, and service and repair.

All waveforms are internally generated with amplitudes to 10vp-p into 50 ohms. An offset generator allows generation of signals with large offsets. A full range of triggering capabilities is available including internal-external trigger source, gated, and burst modes of operation. Some of the unique features of the B&K Precision Model 4045 include:

• 20 MHz sweep frequency range (sine and square only)
• Sine, square, triangle and arbitrary waveforms
• Modulation in both AM and FM
• Lin or Log sweep function
• Bright, informative LCD
• RS-232 interface

Measuring a compact 3.5 x 8.3 x 8.4, the model 4045 weighs only 5.5 lbs. and is available for immediate delivery. It comes with the B&K Precision warranty, AC line cord, and a user manual (on CD).

For the name and location of an authorized distributor near you, contact B&K Precision Corporation.

For more information, contact:
B&K Precision Corporation
22820 Savi Ranch Pkwy.
Yorba Linda, CA 92887
Tel: 714-921-9095
Fax: 714-921-6422
Web: www.bkprecision.com

SOLDERING IRON MAINTENANCE KITS

A new line of soldering iron maintenance kits that were developed to assure top performance and long life for heavy-duty industrial soldering irons is being introduced by American Beauty Soldering Tools.

American Beauty Soldering Iron Maintenance Kits include everything necessary to remove flux residue, contamination, and oxidation that build up between the inside of an iron’s heating element and the surface of a heavy-duty tip. Requiring very little time, regular maintenance promotes optimum heat transfer and prolongs the life of any industrial soldering iron, claims the firm.

Packaged in handy storage cases, the maintenance kits each contain a special spiral brush, steel wire brush, and flexible non-woven abrasive pads that wrap around a tip’s shank. The spiral brushes have 1/4” hex shanks to fit most drills. Five individual kits for servicing 40 to 550 watt soldering irons and a complete kit for all sizes are offered.

The maintenance kits are priced from $19.99 (list), depending upon size. Literature and pricing are available upon request.

For more information, contact:
American Beauty Soldering Tools
1177 West Maple Rd.
Clawson, MI 48017
Tel: 800-550-2510
Fax: 248-280-2878
Email: es@americanbeautytools.com
Web: www.americanbeautytools.com

MODEL 8063 ETHERNET-TO-DIGITAL INTERFACE BOARD

ICS Electronics has announced a new Ethernet-to-Digital interface for controlling digital devices over a company network or over the Internet.
Called the Model 8063, this new interface box provides 48 digital I/O lines that the user can control from any computer with an NIC interface or from a TCP/IP network. Typical applications for the 8063 are interfacing digital devices, controlling relay matrices, or acting as an Ethernet interface for devices or instruments with digital interfaces.

The Model 8063 is an Ethernet-to-parallel interface that provides 48 parallel I/O lines that can be configured as inputs or outputs in eight-bit bytes. Data transfer can be done by a combination of three methods, depending upon the needs of the devices connected to the 8063. First is by transferring data directly to or from a specific byte, second by strings of data characters to or from multiple bytes, or third, by setting or reading individual bits in a byte.

Handshake lines are provided for synchronizing the data transfers or for latching data into external devices. The 8063 can also monitor 15 input lines and generate VXI-11 Service Request messages when an enabled line changes state. The 8063 is an IEEE-488.2 compatible interface that responds to the 488.2 common commands and uses SCPI commands to configure its digital interface. The user can customize the 8063’s IDN message to integrate the 8063 into a system. All settings are saved in Flash memory.

The Model 8063 is a VXI-11.3 compliant interface. VXI-11 is a communication standard developed by the VISA consortium in 1995 in conjunction with the VISA Specification. The VXI-11.3 sub-standard covers TCP/IP-to-Instrument servers like the 8063 and is used for LXI devices. Communication with the 8063 is via VXI-11 RPC protocol over a TCP/IP network.

The 8063 can be controlled several ways: the model 8063’s VXI-11 Service can be accessed by LabVIEW, VEE, Visual Basic, and C language application programs that make VISA calls by selecting the 8063 as the TCP/IP resource. Both Agilent and National Instruments provide VXI-11.3 compliant VISA libraries. Linux, Unix, and other programmers who do not want to use a VISA library can access the 8063’s VXI-11 Service by RPC calls from the application program. The VXI-11 Standard includes the necessary RPCGen header files for adding RPC calls to any program. ICS provides a VXI-11 keyboard program which lets users with a WIN32 computer interactively control the 8063 and other VXI-11 compatible instruments without having to write a program.

ICS’s 8063 Ethernet-to-parallel interface has several unique features which include: the 8063 is 100% VXI-11.3 compliant (which is an open...
communication standard); the 8063 supports reverse channel Service Request messages to alert the client application when an event occurs; and it supports multiple clients as part of its standard firmware.

The 8063 is RoHS compliant and is housed in ICS’s small Minibox case that can be rack mounted in a 1U high space. The 8063 is physically interchangeable with ICS’s 4863 GPIB and 2363 serial-to-parallel interfaces. All three interfaces support the same SCPI command set so switching interfaces has minimal program impact.

Pricing for the Model 8063 is $650 each in quantities of one to four units, FOB Pleasanton, CA. Delivery is two to four weeks ARO.

For more information, contact:
ICS Electronics
7034 Commerce Cir.
Pleasanton, CA 94588
Tel: 925-416-1000
Web: www.icselect.com

---

**$51 PCBs**

**FREE Layout Software! FREE Schematic Software!**

For 3

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

---

New Product in the **NUTS & VOLTS** Store
Recently named as one of the Best Inventions of 2006 by **Time Magazine**

**H-racer & Solar Hydrogen Station**

With the H-racer, you can witness the power of new energy technology in the palm of your hand. Horizon has created a unique, patented, miniature fuel-cell car and hydrogen refueling station. The Refueling Station creates hydrogen by electrolysis using water, and once the car’s hydrogen tank is full, the car can run on its own hydrogen fuel-cell system with the simple flip of a switch.

For more information see page 100

---

Do you know how many watts (YOUR MONEY) are going down the drain from "THE PHANTOM DRAW?"

Use the Kill-A-Watt Usage Meter to determine your monthly cost from the phantom draw.

Just plug it in and see how much you’re spending on "NOT IN USE APPLIANCES".

To order call 1-800-775-5834 or order online www.nutsandvolts.com - $29.95 plus tax.
Renesas Technology

No. 1* supplier of microcontrollers in the world

introduces the R8C/Tiny Series of microcontrollers. Its powerful 16-bit CPU core running at 20MHz provides the performance never imagined in 8-bit MCUs. R8C/Tiny MCUs high level of integration and peripheral set will enhance your application’s functionality while reducing the overall system cost.

R8C/Tiny Product Roadmap

**Get Started Today -**

Go online and register to be eligible for a FREE Starter Kit

**www.america.renesas.com/ReachR8C/e**
I considered just slowing the fan to some speed that produced less noise. The problem with just running the fan slower is that the cooling may be inadequate. A rise in ambient temperature or higher power dissipation in the circuit would cause everything to overheat. So, I decided to adjust the fan speed with temperature, running at minimum speed when the circuit is cool and increasing fan speed as the temperature increased. I measured the temperature of the hot component with a thermistor and adjusted the fan speed based on that temperature.

A schematic of the circuit is shown in Figure 1. The circuit uses a single quad op-amp, U1, an LM324. Q1 is a PNP power transistor that supplies current to the motor. T1 is a thermistor that provides temperature sensing, and is mounted to the component that needs to be cooled.

T1 is an NTC (negative temperature coefficient) thermistor, which means that the resistance goes down as temperature goes up. The resistance of T1 is about 10K at room temperature, falling to about 2.8K at 70°C. Since most commercial electronics operate to about that temperature, the fan needs to keep the temperature below that point.

Small DC fans are typically one of the less reliable components in an electronic system, so the circuit includes a fault output to indicate when the temperature rises beyond acceptable limits (in this case, a bit less than 70°C). This could happen if the fan fails or if the ventilation holes become plugged with dust or covered with a piece of paper. The fault output can notify a processor to shut off power or take some action to reduce power dissipation in the circuit.

Q1 operates in linear mode, supplying current to the motor. With a larger, higher current fan, this arrangement could potentially cause significant heating from Q1 itself. The fan used here draws less than 100 mA at full supply voltage, so that is not an issue. Note that the feedback network (R2/R3) connects to the non-inverting input of U1A (pin 3). Ordinarily, feedback to an op-amp goes to the inverting input (pin 2, in this case), but in this circuit, Q1 provides an additional stage of inversion. This effectively swaps the functions of the inverting and noninverting input pins.

Q1 is turned on by current flowing to ground through the base. Note that there is no direct relationship between the voltage at the base of Q1 and the output voltage. Q1, being a bipolar transistor, is a current amplifier, so U1A will sink sufficient current through the base to produce the appropriate voltage at the output.

The fan I used will operate from 5V to 12V. At 5V, the noise is almost...
inaudible, so the circuit always supplies 5V to the fan, through U1B. As the temperature rises so that the output needs to rise above 5V, U1A begins to sink current, raising the fan supply voltage. Operating the fan in this way provides two advantages: First, there is always some airflow through the box. Second, it simplifies the circuit since there is no need to add circuitry to turn off the output when it would be below the minimum fan operating voltage.

U1C provides fault detection. When the temperature rises above approximately 65°C, U1-8 goes low. U1D is just an inverter to drive Q2, providing an open-collector output. This allows multiple fan circuits to share a common fault signal. U1C includes a small amount of hysterisis through R9 to insure that the fault output is a well-defined transition.

Note: R13 is not installed for a fan that can operate at 5V. For a fan with a minimum operating voltage higher than 5V, R13 is needed to provide that voltage to the fan. See the sidebar “Working the Equations” for details.

Calculating Circuit Values

In an op-amp that is linear (the output is not saturated at either supply rail), the inputs (U1-3 and U1-2, in this case) will be at the same voltage. The voltage at U1-2 is controlled by the voltage divider comprised of R3 and R2, and varies with fan voltage.

The equation for U1-2 can be written as:

\[
\frac{(V_o - 5V) \times R_2 + 5V R_2}{R_2 + R_3}
\]

The equation for U1-3 can be written as:

\[
12V \times R_1 = \frac{(V_o - 5V) \times R_2 + 5V R_1 + 7K}{R_2 + 10K}
\]

Since Vo is 5V, the right side of the equation reduces to 5V and R1 is 5K.

At the other end of the range, the equations are:

\[
12V \times 5K = \frac{(12 - 5V) \times R_2 + 5V}{R_2 + 10K}
\]

which makes R2 equal to 3.12K. Since I didn’t need precise temperature control, I used the nearest 5% resistors; a more precise circuit would use 1% or better parts. Note that R1 reduces to a simple equation only because the reference for pin 3 is equal to the minimum fan voltage. A different reference value (such as a precision 2.5V reference) would result in solving both equations simultaneously (remember high school algebra!) for the values of R1 and R2. Also, many 12V fans will not operate as low as 5V, and a different minimum fan voltage would also change the result.

For more precise control, you would use the following 1% values:
In this circuit, the 5V and 12V supply voltages are used as references, so the temperature control will vary with supply voltage. More precise control would require more precise voltage references.

U1B sets the minimum fan voltage at 5V. If you use a fan with a higher minimum voltage, you would need to apply that voltage to pin 6 through a voltage divider. For example, a fan that needs at least 7V to operate would make the equations look like this:

\[
\begin{align*}
7V: & \quad \frac{12V \times R1}{R1 + 7K} = \frac{(7 - 5V) \times R2}{R2 + 10K} + 5V \\
12V: & \quad \frac{12V \times R1}{R1 + 4K} = \frac{(12 - 5V) \times R2}{R2 + 10K} + 5V
\end{align*}
\]

If you solve these (see “Working the Equations”), you get \( R1 = 6.25K \) and \( R2 = 4.95K \). These values would produce 7V to the fan at 40°C, and 12V to the fan at 60°C. In some cases, when you solve equations like this, you get a negative resistance value. In those cases, you have to change a resistor or reference voltage in the circuit to get the result you want. The exception is if you get a quadratic equation (like you will if you work through this example) with one positive and one negative root. Then the positive root is the resistance you want.

To complete the 7V conversion, U1B would be set to produce a 7V output by making R13 equal to 2.5K.

Most thermistors have very low mass, so if you change component values, you want to be careful with R1. If the value of R1 is too small, the thermistor will dissipate enough power to heat itself and affect the sensed temperature. The maximum power dissipated by the thermistor in this circuit is about eight milliwatts.

**Thermistor Attachment**

The thermistor can be used to measure air temperature inside the enclosure, but you typically want to measure the temperature of an IC, heatsink, or other component. In this case, I used a small bead-type thermistor, glued into the barrel of a ring terminal (Figure 2). The ring terminal was then attached to the hot component with a machine screw and nut. This would work with a component such as a hard drive, or a part with a heatsink. If the part is, say, a TO-220 transistor, you could attach the thermistor with the same screw that attaches the transistor to its heatsink.

If you are measuring the temperature of an IC, you would want to glue the thermistor to the IC package or to a heatsink that is then glued to the package. Thermally conductive epoxies for mounting thermistors are available from companies such as Omega (www.omega.com). Various manufacturers, including RTI Electronics (www.rti.electronics) and Vishay (www.vishay.com) make thermistors already mounted to metal tabs with mounting holes.

**Testing**

Testing is fairly simple: Connect +12V and +5V, and (optionally) connect a fan. The output voltage should be about 5V. Use a heat gun or hair dryer to heat the thermistor; the fan speed and voltage should increase as the temperature goes up. To test the fault output, connect a 10K pullup from the fault output to +5V; the fault output should go low around 70°C.

**Temperature Control**

The control circuit does not attempt to maintain a constant temperature. Instead, the temperature is allowed to float. For a given ambient temperature and circuit power dissipation, the temperature will tend to stabilize at some value. However, given the significant thermal mass in this enclosure (and associated delay between changing fan speed and resulting temperature change), maintaining a constant temperature would be difficult, anyway.

Although maintaining a fixed temperature would seem more “normal” for a control system, in this case, it would result in higher fan noise. At higher ambient temperatures, or under increased load, a constant-temperature circuit would run the fan...
faster to maintain the temperature. By allowing the temperature to float up with ambient temperature and circuit power dissipation, the average fan speed will be lower (although the temperature will usually be higher than a fixed setpoint would be).

So, what happens if the fan runs at full speed all the time and the temperature is still above the upper limit? That means the fan is too small to start with, or that the ventilation is either inadequate or blocked. In that case, it doesn’t matter if the circuit is a constant-temperature controller or not — no type of fan control can provide adequate cooling.

The resistance of an NTC thermistor does not have a straight-line relationship with temperature; instead the curve is approximately logarithmic (Figure 3). Since the fan supply voltage varies with thermistor resistance, the fan supply voltage will increase faster around 5V, and slower as it approaches 12V.

### Other Types of Fans

Another approach to this project would be to control fan speed directly (instead of controlling fan voltage) using a fan with a tach output. These fans produce a pulse once or more per revolution. Of course, this approach requires some method of translating the tach signal to a control voltage — typically with a microcontroller or fan controller IC.

#### PWM (pulse width modulation)

### Working the Equations

The fan used in this circuit will operate from 5V to 12V. Some 12V fans will not operate down to 5V; they may only operate from 7V to 12V. For those who want to follow the math, here is how the op-amp equations are solved for 7V to 12V operation:

For 7V:

\[
\frac{12V \times R1}{R1 + 7K} = \frac{(7 - 5V) \times R2}{R2 + 10K} + \frac{5V}{R2 + 10K} = \frac{7R2 + 50K}{R2 + 10K}
\]

Expand and then collect terms:

\[
12R1R2 + 120KR1 = 7R1R2 + 49KR2 + 50KR1 + 350K
\]

\[
5R1R2 + 70KR1 = 49KR2 + 350K
\]

For 12V:

\[
\frac{12R1}{R1 + 4K} = \frac{7R2}{R2 + 10K} + \frac{5V}{R2 + 10K} = \frac{12R2 + 50K}{R2 + 10K}
\]

Expand and collect terms:

\[
12R1R2 + 120KR1 = 12R1R2 + 48KR2 + 50KR1 + 200K
\]

\[
70KR1 = 48KR2 + 200K; R1 = \frac{48KR2 + 200K}{70K}
\]

Substitute second value for R1 into first equation:

\[
\frac{145KR2 + 200K}{70K} = \frac{49KR2 + 350K}{5R2 + 70K}
\]

Expand and consolidate:

\[
240KR2^2 + 4360KR2 + 14000K = 3430KR2 + 24500K
\]

\[
240KR2^2 + 930KR2 - 10500K = 0
\]

Solving this with the quadratic equation results in:

\[
R2 = 4.95K \quad \text{(the other root is negative). Inserting that back into either equation for R1 yields R1 = 6.25K.}
\]

Substituting a different thermistor or different temperatures just means you have to substitute the resistance values for the thermistor/temperature you want to use. For example, if your thermistor has a resistance of 8K at the lowest fan temperature, you would substitute 8K for 7K in the equation for 5V operation (or 7V operation, in the above example).

Calculation of R13:

The original circuit did not use R13 because the fan operates at 5V. However, for a fan requiring a minimum of 7V, U1B needs to produce a 7V output instead of a 5V output. Since the reference to U1B (pin 6) is 5V, we need U1-5 to be at 5V when the fan voltage is at 7V:

\[
5V = \frac{7V \times R13}{R13 + R7} = \frac{7V \times R13}{R13 + 1K}
\]

\[
5R13 + 5K = 7R13
\]

\[
5K = 2R13; R13 = \frac{5K}{2} = 2.5K
\]

The same thing could be accomplished by leaving out R13 and connecting a voltage divider from 12V to ground, using values that produce 7V at the junction of the two resistors. U1-6 would connect to the 7V junction point.

The thermistor used in this project was a surplus unit from an unknown manufacturer. A commercial part would have different resistance/temperature values. For example, the Vishay 2381 645 90169/NTCALUGE2C90169 is cemented to a mounting lug and has a resistance of 5.3K at 40°C and 2.49K at 60°C. The corresponding circuit values would be:

R1: 3.9K
R2: 3.9K
R3: 7.5K
R10: 4.5K
could also be used to control the fan, and would eliminate any problems with power dissipation in Q1. However, this requires a fan that is capable of PWM operation — many low-voltage DC fans use brushless DC motors with internal controller ICs that don’t respond well to PWM control.

**Minimum Current**

Although the fan used here is a low-current device, if you were operating from batteries, you might want to turn the fan completely off if the temperature is low enough. The simplest way to do this is to add a microcontroller (covered next). You could also add circuitry to inhibit the fan output completely below 5V.

**Adding a Microcontroller**

By now, some readers are asking why not just use a microcontroller for this circuit? There are a number of controllers with on-chip ADCs to read the thermistor, and adjusting the temperature setpoints is a simple matter of changing the firmware.

A microcontroller-based design would work, but many of the parts in this circuit would still be needed: the thermistor voltage divider, the output transistor, and an op-amp to drive it. In addition, the microcontroller would require a DAC unless one was available on-chip. A microcontroller solution would potentially increase EMI (not a consideration in my one-of-a-kind project, but certainly a consideration on a production design).

A microcontroller is suitable for this design if any of the following are true:

- A microcontroller is already used elsewhere, so added cost is minimal.
- PWM motor control is needed.
- The temperature-fan speed curve needs to be “linearized” to compensate for the logarithmic thermistor curve.
- Setpoint adjustment needs to be customized to the application, or environment.
- Constant temperature control is needed.
- You want to turn the fan completely off below a particular temperature for lower current drain (as in a battery application).

**Better Precision**

The largest sources of error in the circuit are:

- 5% resistors
- 12V reference for thermistor circuit
- Thermistor tolerance

Better precision would be desired, but 1% values would produce better accuracy. The 12V and 5V references could be replaced with precision 2.5V or 5V references. However, doing so means that the equations must be reworked with the thermistor you plan to use. It is possible to pick values that cannot be “mapped” to the output voltages you want. This will be obvious when you do the math (see “Working the Equations”) because you will end up with a negative value for one or more resistors.

Finally, the thermistor tolerance affects precision. Thermistors typically have a 5% tolerance, but 1% or better parts are available. You could also characterize the thermistor you plan to use, but if you need that level of precision, you probably need to do some sophisticated airflow analysis.

**Conclusion**

This basic circuit allows you to control the speed of a fan to suit your noise tolerance. Using the example equations, you can change the temperature ranges and thermistor to adapt it to your needs and specific circumstances, and keep your components cool!
Measuring In-Flight Acceleration

A three-axis accelerometer was the first type of sensor evaluated using the prototype flight recorder. The reasons for this were that in addition to providing details about the rocket’s basic performance, integrating the signal from the vertical axis could produce a complete velocity profile of the flight and a second integration should yield peak altitude. I was also interested to see if an indication of aerodynamic stability could be obtained by measuring horizontal acceleration.

The device chosen was a MMA7260Q three-axis accelerometer from Freescale Semiconductors. This micromachined IC has integral signal conditioning with low-pass filters. Sensitivity is selectable in four steps (±1.5g/2g/4g/6g), depending on the logic levels placed on two control pins. Current consumption is only about 0.5 mA over the supply range of 2.2-3.6V and the analog output swings to within 0.25V of either rail.

By operating both the accelerometer and flight recorder from a single three volt supply, a good match was obtained between the sensor’s analog output and the microcontroller’s ADC input range.

Last month, we introduced you to the basics of model rocketry and outlined the hardware design of a compatible flight recorder.

This article will describe the embedded software required to record and replay three channels of acceleration, and includes a plot of actual flight data in Microsoft Excel. This is followed by examples of some additional sensors that may be used with this versatile system, including calibrated ‘smart’ versions that have integral signal conditioning and serial digital interfaces.
Signal conditioning circuitry is therefore limited to the simple RC low-pass filters that are included as standard on the ADC inputs of the sensor port.

Figure 2 shows the connections required between the breakout board and the sensor port. Although it is possible to operate with different sensitivity on each axis, by changing the state of the control pin ranges between successive ADC readings, the associated settling time would adversely affect the sampling rate. The range of all three channels was therefore fixed at the maximum specified range of ±6g. (However, the ADC accepts signal levels that exceed this range.)

The accelerometer is only available in a difficult-to-hand-solder surface mount package, but this limitation was overcome by using a small breakout board from Spark Fun Electronics (www.sparkfun.com) that has the MMA7260 connections brought out to 0.1 inch pads. Silicone glue was used to mount the breakout board onto the flight recorder that seemed to provide reasonable rigidity while offering the accelerometer some protection against excessive g-forces.

If we now turn our attention to the Y and Z traces, we see that they understandably show lower levels of acceleration. They do, however, indicate that the rocket performed a definite ‘wiggle’ after leaving the launch rail that decayed in amplitude over the next three seconds. Following touchdown, the horizontal traces seemed to show the nose cone rolling after touchdown. It is possible that the initial wiggle and ground roll were both caused by the effects (upon the fins and chute) of a moderate side-wind at the launch site.

**PICAXE Firmware**

Figure 4 lists the PICAXE firmware that was used to record three channels of acceleration during flight and export the data to Microsoft Excel. The function of the trigger, record, and playback routines were explained in last month’s article. Note that the setfreq m8 command doubles the default PICAXE clock rate to 8 MHz, which means that data is sent at 9.6 kbaud (not 4.8 kbaud as listed). The SelmaDAQ macro is used in Excel to accept data and place it in the spreadsheet for subsequent processing and display.

Martin Hebel, of SelmaWare Solutions and a professor in the Electronic Systems Technologies program at Southern Illinois University Carbondale, wrote SelmaDAQ. This application only uses a fraction of the capabilities of this extremely flexible software tool, which may be downloaded from www.selmaware.com

**Additional Sensors**

As horizontal accelerometer channels normally provide less angle below the chute, and then began a relatively sedate descent. Touchdown at 18 seconds appears to have been surprisingly gentle. The nose cone eventually came to rest on its side, with the vertical accelerometer measuring near zero g.
information about a rocket’s performance than the vertical channel, a typical payload would only include a single-axis accelerometer. The remaining two ADC channels could therefore be used to measure signals from the following analog sensors.

Of the many sensors used in amateur rocketry, probably the simplest type is a sun detector. In clear weather and low sun elevation conditions, this can be used to measure a rocket’s rate of roll by detecting the sun through a narrow vertical slot in the nose cone. I have found that yellow LEDs make quite good sun detectors. The voltage generated across the LED is approximately one volt when viewing general sky, but can exceed 1.5 volts when exposed to direct sunlight. By connecting the cathode of the LED to the sensor port ground and the anode to an ADC input (via an op-amp voltage follower), the PICAXE is able to record the level peaks in light as the rocket rotates. Roll rate is an important parameter, because fin-induced rotation is often used to improve a rocket’s directional stability, but too high a rate can tangle the lines of the recovery chute during deployment.

A possible application for the remaining ADC channel could be

```
setfreq m8
symbol memmin = 0
symbol memmax = 8191
symbol memaddress = W6
symbol rate = 8
symbol launch = pin6
symbol cable = pin7
symbol power = 2
symbol redled = 5
symbol download = 7
high power
pause 5000
playback:
high redled
pause 1000
serout download, T4800, ("CLEARDATA",CR)
pause 1000
serout download, T4800, ("3-axis acceleration",CR)
i2cslave %10100000, i2cfast, i2cword
for memaddress = memmin to memmax step 3
  read12c memaddress, (b0,b1,b2)
  serout download, T4800, ("DATA","",#b0,"",#b1,"",#b2,CR)
next memaddress
if cable = 0 then trigger
  low power
trigger:
  if launch = 1 then record
  high redled
  pause 500
  low redled
  pause 500
goto trigger
record:
for memaddress = memmin to memmax step 3
  readadc 0, b0
  readadc 1, b1
  readadc 2, b2
  write12c memaddress, (b0,b1,b2)
  pause rate
next memaddress
low power
```

*FIGURE 4. Three-axis accelerometer programmer for PICAXE-18X.*
sampling the amplified output from a magnetoresistive sensor that is used to measure the vertical component of the Earth’s magnetic field in order to detect when the rocket tips over at apogee. Honeywell and Phillips produce devices that have the required 0.5 gauss sensitivity for this task (but the output signal would have to be amplified before feeding the ADC).

By comparing the recorded magnetic field strength and vertical acceleration data, it should be possible to measure any delay between apogee and nose cone ejection. This information will help determine the engine delay charge required to deploy near the rocket’s minimum speed, in order to reduce the risk of damaging the chute.

Having employed all three of the flight recorder’s ADC channels in the above examples, we still have the option of interfacing sensors via the sensor port’s serial I²C bus. A good example of an I²C compatible smart sensor is the SCP1000 barometric pressure IC from VTI Technologies. Pressure and temperature output data are calibrated and compensated internally. This device operates from a 2.4V-3.3V supply and has the potential to provide a more accurate altitude profile than could be derived by integrating vertical acceleration data.

The SHT11 humidity and temperature IC from Sensirion AG is another smart sensor that has a two-wire serial bus. However, it is not I²C compatible and would have to be interfaced by ‘bit banging’ sensor port I/O lines. This device operates from a 2.4V-5.5V supply and the output data is calibrated and compensated internally. It could be used to study microclimates by measuring the relative humidity and temperature of a column of air (e.g., to obtain a profile through a layer of mist). However, because the sensor’s update rate is in the order of seconds, measurements are best taken during a slow descent under a generously-sized parachute.

Conclusions

With the exception of the sun and magnetoresistive sensors, all the above examples could be connected directly to the sensor port without additional signal conditioning or power supply circuitry.

The flight recorder is a compact and low power instrumentation tool that can be rapidly configured to accommodate a range of applications. In addition, the use of SelmaDAQ makes spreadsheet data import, processing, and display an almost seamless task.

Small, Sleek, Smart. Seamlessly alternates between multiple interfaces:
* Serial
* Radio Control (RC)
* Analog Voltage
  * **Switch back & forth instantly!**

Five RC or analog input channels:
* 2 bidirectional motors
* Auxiliary channel (unidirectional motor)
* Flip channel for invertible robots
* Override channel switches between autonomous and human control

* Dimensions: 1.86" x 1.86" x 0.8"  
* 13 A continuous, 30 A peak  
* 6-16 volts  
* Precision motor control with ±1 μs pulse width measurement and 127 motor speeds/direction  
* Highly configurable  
* Intelligent learning  
* Advanced safety features

1-877-7-POLOLU  www.pololu.com

TRex Dual-Motor Controller

Pololu Robotics & Electronics  
6000 S. Eastern Ave. 12D, Las Vegas, NV 89119

This book is available in our store on page 99.  
Visit www.nutsvolts.com or call 800-783-4624.  
$24.95
4 Channel Guitar Amplifier Kit
KC-5448 $57.50 + post & packing
The input sensitivity of each of the four channels is adjustable from a few millivolts to over 1 volt, so you can plug in a range of input signals from a microphone to a line level signal from a CD player etc. A headphone amplifier circuit is also included for monitoring purposes. A three stage EQ is also integrated, making this a very versatile mixer.

Water Level Indicator Kit
KC-5449 $20.50 + post & packing
This simple circuit illuminates a string of LEDs to quickly indicate the water level in a rainwater tank. The more LEDs that illuminate, the higher the water level is inside the tank. The input signal is provided by ten sensors located in the water tank and connected to the indicator unit via-light duty figure-8 cable. Kit supplied with PCB with overlay, machined case with screenprinted lid and all electronic components.

IR Remote Control Extender MKII
KC-5432 $14.50 + post & packing
Operate your DVD player or digital decoder using its remote control from another room. It picks up the signal from the remote control and sends it via a 2-wire cable to an infrared LED located close to the device. This improved model features fast data transfer, capable of transmitting Foxtel digital remote control signals using the Pace 400 series decoder. Kit supplied with case, screen printed front panel, PCB with overlay and all electronic components.

Fuel Cut Defeat Kit
KC-5439 $11.75 + post & packing
This simple kit enables you to defeat the factory fuel cut-out signal from your car’s ECU and allows your turbo charger to go beyond the typical 15-17psi factory boost limit. Note: Care should be taken to ensure that the boost level and fuel mixture don’t reach unsafe levels. Kit supplied with PCB, and all electronic components.

Variable Boost Kit for Turbochargers
KC-5438 $11.75 + post & packing
It’s a very simple circuit with only a few components to modify the factory boost levels. It works by intercepting the boost signal from the car’s engine management computer and modifying the duty cycle of the solenoid signal. Kit supplied in short format with PCB and overlay, and all specified electronic components.

How to order: Phone: Call Australian Eastern Standard Time Mon-Fri on 1800 784 0263 Email: techstore@jaycar.com Post: PO Box 6424, Silverwater NSW 1811, Australia Expect 10-14 Days For Air parcel delivery
When I first started this series, I knew I wanted to build a wireless weather station. In no way could I have imagined how well the overall system would function, nor could I have envisioned its versatility.

In this article, it all comes together.

If you have already built a PC or microcontroller-based weather station connected to a 1Wire network, you may be asking yourself why you should go to all the effort and expense of converting it to wireless.

The wired 1Wire network has five main disadvantages:

1) Your system is tethered and you have to run a cable to the location where you want to place the display. This can be problematic and time consuming.

2) You are limited to the length of the cable and the number of items you can reliably place on the cable.

3) If you want to add additional sensors to various locations, this can prove almost impossible if you don’t have access to attic or basement areas.

4) With a wired system, you generally only have a single display system.

5) If lightning strikes your weather pole, you run the chance of blowing up your computer or display system.

With a wireless system, all these problems will be solved. The only cable that needs to be run is for a power source. This is simply two wires and in many cases can be a very short run. You can even power your weather pole with a set of solar cells and a rechargeable battery.

The heart of our wireless system is a Zigbee mesh network. You have probably read about mesh networks, but this is your chance to actually build one and put it to work. Once the network is built, you will be able to add sensors or displays to any location in your home.

In addition, we are going to add extra features like the ability to create as many display systems as we like without having to change the network in any way. For instance, I can add a PC interface to the system by simply adding a Zigbee unit to an RS-232 card and I can start collecting data. At the same time, I can build a small LCD display for my desktop. I can add a large LED sign display to the front entrance of my lab.
area or a voice that announces various weather or alarm data.

I tried several Zigbee modules and selected the Maxstream (www.maxstream.net) XBee modules shown in Figure 2. I recommend the whip antenna module as you get a bit more range over the small chip-based antenna module. The XBee whip antenna module will cost you about $19 each and you will need at least three to build your weather station network. Because each module’s firmware must be loaded with special mesh network firmware, you will need to get a starter kit (Maxstream #XB24-DKS) which consists of the following:

- RS-232 development board
- USB development board
- Two XBee OEM RF whip antenna modules
- Various cables and connectors
- AC adapter
- Configuration software

With the purchase of this starter kit, you will need at least one additional RF module; more if you want to add additional sensor networks and display systems.

Before I continue, it’s important that I point out that Maxstream has just released a set of Series 2 modules and development kits. The Series 2 modules are not currently compatible with the existing XBee modules and will not work with the instructions presented here. I can assure you, however, that the current mesh network firmware for the existing XBee modules works just fine. My wireless weather station has been up and running for months now and has handled all sorts of power outages and other interruptions without a single glitch.

**Just What Do You Gain by Using a Mesh Network in Your Home Weather Station?**

I had three Zigbee/sensor modules located in my work area but wanted to collect data from a nearby barn. The barn is out of range of my current network. By placing a module in a small shed located between the two buildings as shown in Figure 3, I was able to add the barn to my network. This also has the benefit of allowing me to monitor the pool temperature.

Our weather station mesh network is set up in such a way that all the modules broadcast their data. This allows any module placed in the network to be able to collect or transmit telemetry data.

**Updating the Module’s Firmware**

When setting up your weather station mesh network, you will need to have one coordinator. All the other modules are set up as routers. The coordinator needs to have special mesh network coordinator firmware loaded. All the other modules need to have special mesh network router firmware loaded. To load the proper firmware into the modules, you need to follow these steps:
STEP 1: Install the X-CTU software that came with the development kit.

STEP 2: Load up the software and select the com port you have connected to the development board. Note that when using the RS-232 development board, you will use one of your existing com ports as shown in Figure 4. The USB development board will create a new com port and will be labeled as shown in Figure 5. The actual com port assigned to the USB interface will vary depending on the current com ports already installed on your system.

STEP 3: Select the modem configuration tab and hit the Download New Versions button as shown in Figure 6. This will download the latest mesh network firmware files from the Maxstream website. In order to do this, you will need to be connected to the Internet during this step.

STEP 4: Once you have updated the X-CTU firmware files, you will need to update the firmware on each module. I also suggest you use a small piece of tape and mark each module so you can keep track of the node identifiers you assign each one. Start by programming one of the modules as a coordinator. To do this, insert the module into the connected development board and hit the Read button on the configuration tab. This will load the current settings for the installed module. Note that the modules are shipped with the baud rate set to 9600. In the Functions Set field, select the XBEE ZIGBEE COORDINATOR AT (BETA) setting, and then hit the Write button as shown in Figure 7. This will cause the X-CTU software to upload the new firmware into the module.

STEP 5: Once the firmware update is complete, hit the Read button. Make sure you label the module so you can keep track.

STEP 6: Now you need to load the router firmware into all the remaining modules. As before, insert the module into the development board and hit the Read button. I found that after changing the module, the X-CTU software would no longer respond. If this happens, exit the program and restart it. For the routers in the Functions Set field, select the XBEE ZIGBEE ROUTER AT (BETA) setting, and then hit the Write button as shown in Figure 8.

STEP 7: Once the firmware upload is complete, hit the Read button. Make the following changes manually or use the included file called XBMeshrouter.pro. The fields that need to be changed include the following:

- **PAN ID** = 234
- **Destination Address Low** = FFFF
- **Node Identifier** = R1
- **Packetization Timeout** = 25

Note that each router will need to have a different Node Identifier. I used R1-Rn in my network. It doesn’t really matter what you use, as long as they are different. Write the changes and repeat for each router module.

You now have all your modules configured for your mesh network and are ready to build your wireless weather station.

**Constructing the Network**

For the basic network, there
are three components that I will call satellites:

- PC Interface Satellite
- Indoor Weather Satellite
- Outdoor Weather Satellite

**PC Interface Satellite**

Since the PC interface satellite is nothing more than a USB or RS-232 development board connected to the PC, we already have one of our satellites. The USB development board shown in Figure 9a is a great board for the PC interface satellite. Since it’s powered directly by the USB port, it’s simple to connect. The only problem with using this board is that when the PC is powered down, we lose the ability for this satellite to act as a router. If this is a problem, you can always plug it into a powered USB hub. If you decide to use the RS-232 development board shown in Figure 9b, you will need to connect the included AC adapter.

I recommend that you mount the development boards between two pieces of plastic as shown in Figures 9a and 9b. This will protect your board from shorts and will allow you to stand the board on end. Figure 10 shows my PC interface satellite sitting on one of my shelves.

**Indoor Weather Satellite**

You will need a couple of sensors indoors as part of your weather station. As a minimum, I recommend a barometric pressure gauge and a temperature sensor. I also added a humidity gauge to mine. Since we need a coordinator module on our network, the indoor satellite is the perfect candidate. Its job is to set up the network routing for all the other modules (routers) on the network.

The whole unit shown in Figure 11 is controlled by a DiosPro 28. The DiosPro can talk directly to one or more 1Wire networks, as well as to various other sensors. Since I am using a Hobby Boards pressure gauge, a 14 VAC adapter is needed to power this satellite. As an option, you may also use the SparkFun pressure sensor that was featured in the July issue of this series.

The components needed for this satellite are:

- DiosPro 28 chip
- Dios Carrier 1 kit (Note: This is built with the headers facing up.)
- One amp regulator kit
- XBee module
- XBee interface board
- Hobby Boards or SparkFun pressure sensor
- Humidity gauge (optional)
- AC adapter
- PlexiGlass and standoffs
- Double sided foam tape
- SchmartBoard jumpers

Notice that I listed an XBee interface board. This is a board that you will need to build. It will provide the 5 to 3.3 volt converter needed to connect the microcontroller. On the Kronos Robotics website, there are four application notes showing how to build various interface boards:

- [www.kronosrobotics.com/Projects/MaxStreamInterface1.shtml](http://www.kronosrobotics.com/Projects/MaxStreamInterface1.shtml)
- [www.kronosrobotics.com/Projects/MaxStreamInterface2.shtml](http://www.kronosrobotics.com/Projects/MaxStreamInterface2.shtml)
- [www.kronosrobotics.com/Projects/MaxStreamInterface3.shtml](http://www.kronosrobotics.com/Projects/MaxStreamInterface3.shtml)
- [www.kronosrobotics.com/Projects/MaxStreamInterface4.shtml](http://www.kronosrobotics.com/Projects/MaxStreamInterface4.shtml)

In my opinion, the type 4 interface is the smallest and easiest to use, but types 2 and 3 will work just as well. Next month in Part 2, we will go step-by-step in the overall assembly.
and testing process of this satellite.

Outdoor Weather Satellite

The outdoor weather satellite collects all of the data from the instruments on our weather pole and transmits the data to our mesh network. All the components that make up this satellite must fit in a weather-tight box like the one shown in Figure 12. If you followed along with this series, you saw how I built a weather station using various 1-Wire components. We will utilize this interface as well with our outdoor weather satellite. The main advantage that we gain is that once the DiosPro is programmed, you need only supply 7-12V to the pole and the rest is done for you. You may also use a battery and a solar cell to remove even these requirements.

You will need the following components for the outdoor weather satellite:

- DiosPro 28 chip
- Dios Carrier 1 kit (Note: This is built with the headers facing up.)
- Xbee module
- XBee interface board
- SchmartBoard jumpers
- SchmartBoard .1” prototype board
- 5V regulator chip
- Two 100 µF capacitors
- .1 µF capacitor
- 1K resistor

Since space is at a premium with the outdoor satellite, you need to build the Type 4 XBee interface board. You can find a complete application note explaining the process at www.kronosrobotics.com/Projects/MaxStreamInterface4.shtml.

I also used a SmartBoard prototype board to build a one amp regulator, as well as the 1-Wire bus interface. There is plenty of room on this board for any other circuitry you might want to add to this satellite. For instance, you may want to use the built-in 10-bit A-to-D (analog-to-digital) ports on the DiosPro to create some sort of moisture meter.

In Part 3, I will take you step-by-step in building this satellite.

Satellite Display Units

One of the advantages of building a wireless weather station is that you can tap into any of the data being transmitted. I have created a very simple protocol and various DiosPro and PC routines to parse the network data for just about any display system you might want to include.

The LED display shown in Figures 13 and 14 is a Dios LCD board with a Xbee module mounted on the bottom, using one of the various interfaces. This particular display uses a single button to toggle through the many different display items. It even counts lightning strikes collected by the outdoor satellite and keeps track of rain fall totals.

The display in Figure 15 is the same LCD carrier board, but it has a graphic LCD attached. In this particular display, I collect the data given by the indoor satellite and display a 12 hour forecast.

Figure 16 shows a BetaBrite LED sign connected to a Dios Carrier board that scrolls real-time weather data across the screen.

The beauty of these displays is...
that they are not tethered to the rest of the system and can be placed anywhere in the house. You can even build a portable display that runs off of a battery.

For the PC interface, I have created several Zeus routines for parsing the network data using the USB or RS-232 development boards from Maxstream. Figure 17 shows one of the very simple display applications I use for displaying various pieces of data. You can also use some of my gauge routines from some of the previous articles to plot the data. This particular display will be used next month in Part 2 as we test our indoor satellite.

**Final Thoughts**

This portion of the Control Your World series is probably one of the most complicated projects to-date. However, I plan on taking you step-by-step through the various sections as we proceed. As we move forward, you will be able to take the knowledge gained from these and the previous weather articles to further enhance your weather station and home automation system. If you plan on building this system, I urge you to purchase the Maxstream starter kit and an extra XBee module or two. You should go ahead and set them up for a mesh network. Since you get two development boards, you can open two instances of the X-CTU software and run some simple tests.

**What’s Next**

Next month, we will build the indoor weather satellite. Using one of the Maxstream development boards, we will test and display the results from this system. I will also break down the protocol that I use for transmitting data over the network. By understanding this protocol, you should be able to build your own display systems using any language or controller.

Be sure to check for updates at www.kronosrobotics.com/Projects/wirelessweather.shtml.
If you had a separate accurate voltage reference available, you could determine if your DMM is “telling the truth.” This is but one example of why you might want an accurate voltage reference. Other possible reasons include using one to calibrate your DMM, or to construct a precision voltage-to-frequency converter or current source.

The applications for a precision reference fall into two general categories: instrument (DMM, DVM) calibration/accuracy verification, or as a circuit component, such as the reference for an A/D converter.

The construction part of this article is geared towards voltmeter calibration and/or accuracy verification. However, the following discussion of various reference voltage parameters applies equally well to using a voltage reference as a circuit element.

I’ll quickly go through several voltage reference options, starting with inexpensive and fairly crude devices and progress up to a precision lab-quality circuit.

### Zener Diodes

On the low end of the reference accuracy continuum, we have the lowly zener diode. It is readily available and 5% accurate devices are inexpensive. A 5.1V, 1N751A zener can be purchased for about a dime and is a good starting point for our discussion.

Zeners provide their stated breakdown voltage only with a particular test current, I_t, flowing through them. This means that the series-limiting resistor shown in Figure 1 must be sized to guarantee that this current will, indeed, flow.

For instance, with a 9V supply and a 1N751A which has a I_t of 20 mA, the resistor value is (9-5.1)/20 mA = 195 ohms. Note that the value of R1 will be different if anything other than a high impedance is placed across the zener.

I don’t want to dwell too much on the lowly zener but there is one point worth emphasizing. You may be asking, what is the advantage of using a zener if an accurate supply voltage and a precision resistor are needed to get the precise desired voltage.

So, just how accurate is your DMM?

---

**NOTE:**

PCB patterns and artwork are available on the Nuts & Volts website at [www.nutsvolts.com](http://www.nutsvolts.com).
current flowing through the device? Well, fortunately, things are not as critical as they might first appear. The reverse impedance of the zener is quite low, which means that the current can vary quite a bit and the voltage will still remain relatively constant. As part of the research for this article, I tested five 1N751A zeners from Fairchild Semiconductor. I used a 200 ohm, 5% series resistor whose measured value was 197 ohms. When I varied the supply from 8V to 10V, the current through the least accurate zener of the group varied from approximately 14.3 mA to 24.3 mA, but the zener voltage stayed within 2.3% of 5.1V. The best of the five devices was within 0.2% of 5.1V! Not bad for such a simple and inexpensive component. Your mileage may vary.

Three-terminal Regulators

Next up, we have the common 7805 three-terminal regulator. The MC7805 from Fairchild (and others) costs about $.50, gives 4% accuracy under light load, is short-circuit proof, and can more easily accommodate wide swings in load current than the simple zener regulator. Higher accuracy devices are available such as the 2% KA7805AE for $.80. Figure 2 shows a typical application circuit.

For 1% accuracy, you might want to consider the LM4040 DIZ-5.0 from National Semiconductor. This device provides 5.00V within 2.5 mV (.05% accuracy) and has a low temperature coefficient of 5 ppm/°C, an aging rate of 15 ppm/1,000 hours, and a supply current of 3 mA. It will cost you a little over $9, however.

At this point, I need to explain a little about aging rate — also known as long-term stability — specifies how the reference voltage will change over an extended period of time. In other words, even though the temperature, applied voltage, and load current may be constant, all voltage references will slowly drift over time. A 15 ppm/1,000/hour aging rate means that after 1,000 hours (about 42 days), the voltage can be expected to be 15/1,000,000 or .0015% higher or lower than its original value.

Going Up the Accuracy Food Chain

Next up on the accuracy scale, you could try a LM4040AI-5.0; this device is accurate to 0.1% and has a moderate temperature coefficient of 100 ppm/°C. Now we are getting somewhere! But, as usual, you get what you pay for. This increased performance will approximately double the cost to a little over $2.

Next, we can consider the Analog Devices AD586L shown in Figure 4. This device provides 5.00V within 2.5 mV (.05% accuracy) and has a low temperature coefficient of 5 ppm/°C, an aging rate of 15 ppm/1,000 hours, and a supply current of 3 mA. It will cost you a little over $9, however.

Now for the pièce de résistance! Intersil has recently come out with a series of voltage references that significantly improve upon the historical...
price/performance ratio. The series includes devices with 1.024V to 5V reference voltages with high accuracy, low temperature coefficient, low aging, and very low current consumption. The X60008C is used in the construction part of this article and costs about $7.50, provides 5.000V accurate to within 500 µV (0.01%), has a low temperature coefficient of 5 ppm/°C, low aging rate of 10 ppm/1,000 hours, and incredibly low supply current requirement of 800 nA. Figure 5 shows a schematic using this device.

Circuit Description

The 5V precision reference is powered by a 9V battery. The circuit draws about 4 mA with a fresh battery, which results in a battery life in excess of 48 hours of continuous operation. Increased battery run-time can be achieved by applying power, letting the unit stabilize, making your measurement, and then turning the power off. D4, a TL431 “variable zener” whose breakdown voltage is set by R2 and R4, produces 6.5V to supply power to the X60008.

The .01% accuracy of the X60008 is specified with a supply voltage of exactly 6.5V; the 5.000V reference output can deviate as much as 100 µV per volt away from 6.5V. For example, if the supply is 7.5V, the 5.000V reference could be off as much as 600 µV (500 µV from the initial .01% tolerance plus another 100 µV from the supply being 1V from the ideal 6.5V); 1% resistors and a 0.5% TL431 are used to keep the 6.5V supply between 6.3V and 6.7V.

Diodes D1 and D2 prevent damage to the circuitry if the battery is installed backwards. A low current LED, D3, is used as a visual power-on reminder. I designed the circuit to function correctly with a battery voltage as low as 8.0V. However, I was pleasantly surprised when I measured the first couple of prototypes and found that the 5V reference voltage was within .01% with a battery voltage all the way down to 6.7V! Hopefully, you will get equally good results.

Besides supply voltage and component tolerances, there are two more potential error sources that need to be mentioned. Mechanical stress on the die inside the X60008 package can cause a voltage shift in the reference voltage. For instance, pressing down on the board and causing it to flex can cause stress. Looking at the photo in Figure 7, you will see two slits in the printed circuit board (PCB) on either side of the X60008. The purpose of these slits is to de-couple the X60008 from the rest of the board and make it more difficult for mechanical stresses to be transmitted to the reference.

Another potential error source is temperature stress. When soldering the leads of the X60008 to the circuit board, work quickly! It is stressful to the IC die to have one lead at an elevated temperature and other leads at a lower temperature. Leaving the soldering iron on a pad too long can result in the reference voltage permanently shifting to an out-of-spec value. If you have an adjustable temperature soldering iron, don’t turn it up beyond 600°F. Use of a flux pen can be very helpful.

The X60008 package has eight leads, but only four are used. I purposely left off the four PCB pads associated with the unused IC leads; this minimizes the number of leads that need to be soldered, which lowers the amount of
thermal stress that the die is exposed to. If you aren’t confident in your soldering ability, you may want to have someone help you who is more skilled.

Another alternative would be to purchase the PCB with the X60008 soldered onto the board as listed in the Parts List. I have successfully hand soldered four prototypes which resulted in all devices easily meeting their .01% specification. With a reasonable amount of skill, you should achieve similar results.

Assembly

To help keep the cost down, I decided on designing a single-layer PCB. Since the actual voltage reference IC is surface mount, this pretty much dictated that everything be soldered on the component side of the board. This is a little unconventional but no big deal, and as can be seen in Figure 6, results in a compact assembly. The PCB sits on top of the plastic 9V battery holder and is supported by small bosses on the holder that act as PCB standoffs. After component leads are inserted through the board and soldered, cut off the excess lead length flush with the bottom of the board.

Normal anti-static precautions need to be taken during assembly (anti-static work surface and wrist strap). Use a small tip on your iron and solder the X60008. As mentioned above, don’t apply more heat or time than is necessary to achieve a reliable connection. Place and solder the remaining components, saving the battery holder for last. Orient the LED properly by noting that the cathode lead is shorter than the anode and is inserted into the square pad on the PCB. To provide access for your soldering iron tip, space D3, D4, C2, and C3 up off the board a bit. Take a couple of cut-off resistor leads and form them into a “U” shape and insert them into the + and -5V reference output pads.

After all the components are in place and their leads cut off flush with the bottom of the board, insert the two leads of the battery holder through the holes in the board. Next, use a 2-56 x 1/4” screw and nut to hold the PCB firmly against the battery holder. Finally, solder the battery holder wires to the PCB.

Test

Slide the power switch to the OFF position. Insert a fresh 9V alkaline battery into the holder and slide the switch to the ON position. The LED should light up; if not, look for proper polarity of all the diodes including the LED. Using your DMM, measure from the – output terminal to the cathode of D2. This voltage should be between 6.25V and 6.7V. Finally, put your meter across the + and – output terminals and you should see 5.000V displayed.

For highest accuracy, there are three things to keep in mind. First, the voltage reference should be used at temperatures as close to 25°C (77°F) as possible. A few degrees either way won’t make much difference, but you wouldn’t want to put the unit out in the sun and expect to get .01% accuracy. Secondly, allow the circuit to stabilize for 30 minutes with power applied before use. Lastly, the X60008 can source and sink up to 10 mA, but for highest accuracy it should only be connected to a high impedance load such as a voltmeter.

If higher current is needed, a buffer amplifier should be constructed; check the Intersil website (www.intersil.com) for sample circuits. Figure 7 shows a completed prototype with the reference voltage displayed on a recently calibrated .003% accurate, 6.5 digit, Hewlett Packard 34401A DMM.

Parting Thoughts

The previous brief survey of voltage references mentions only a small fraction of the devices that are available. For additional products, check the web pages of Analog Devices, Intersil, Linear Technology, Maxim Integrated Products, National Semiconductor, On Semiconductor, Texas Instruments, and Zetex. These manufacturers also have application notes available that delve into the details of voltage references and will help insure that you have the knowledge to apply them properly.

### PARTS LIST

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Supplier/Part No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>9V alkaline battery</td>
<td>Duracell/MN1604</td>
</tr>
<tr>
<td>B1</td>
<td>Battery holder</td>
<td>Mouser/12BH611</td>
</tr>
<tr>
<td>C1</td>
<td>4.7 µF 16V electrolytic capacitor</td>
<td>Mouser/140-XRL16V4.7-RC</td>
</tr>
<tr>
<td>C2</td>
<td>0.01 µF ceramic capacitor</td>
<td>Mouser/581-SR215C103KAR</td>
</tr>
<tr>
<td>C3</td>
<td>1,000 pF ceramic capacitor</td>
<td>Mouser/80-C315C102K5R5TA</td>
</tr>
<tr>
<td>D1</td>
<td>1N5819 Shottky diode</td>
<td>Mouser/625-1N5819-E3</td>
</tr>
<tr>
<td>D2</td>
<td>1N914 signal diode</td>
<td>Mouser/78-1N914</td>
</tr>
<tr>
<td>D3</td>
<td>LM431CCZX variable zener</td>
<td>Mouser/512-LM431CCZX</td>
</tr>
<tr>
<td>D4</td>
<td>HLMP-K150 low current LED</td>
<td>Mouser/638-HLMPK150</td>
</tr>
<tr>
<td>R1</td>
<td>560 ohm 5% 1/4W resistor</td>
<td>Mouser/291-560-RC</td>
</tr>
<tr>
<td>R2</td>
<td>40.2K 1% 1/4W resistor</td>
<td>Mouser/271-40.2K-RC</td>
</tr>
<tr>
<td>R3</td>
<td>5.1K 5% 1/4W resistor</td>
<td>Mouser/291-5.1K-RC</td>
</tr>
<tr>
<td>R4</td>
<td>24.9K 1% 1/4W resistor</td>
<td>Mouser/271-24.9K-RC</td>
</tr>
<tr>
<td>S1</td>
<td>Slide switch</td>
<td>Mouser/688-SSSS22700</td>
</tr>
<tr>
<td>U1</td>
<td>X60008CIS8-50</td>
<td>Digi-Key/X60008CIS8-50</td>
</tr>
<tr>
<td>1 ea 2-56 x 3/8” screw</td>
<td>Digi-Key/H130-ND</td>
<td></td>
</tr>
<tr>
<td>1 ea 2-56 nut</td>
<td>Digi-Key/H212-ND</td>
<td></td>
</tr>
<tr>
<td>Printed circuit board</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An etched, drilled, and screened printed circuit board with the X60008-50 soldered on the board is available from Doug Malone, PO Box 1542, Battle Ground, WA 98604 for $18 plus $2.50 shipping & handling. WA residents please add 8% sales tax.

---

**Parts List**

- **9V alkaline battery**: Digi-Key/H130-ND
- **Battery holder**: Digi-Key/H212-ND
- **4.7 µF 16V electrolytic capacitor**: Digi-Key/H212-ND
- **0.01 µF ceramic capacitor**: Digi-Key/H212-ND
- **1N5819 Shottky diode**: Digi-Key/H212-ND
- **1N914 signal diode**: Digi-Key/H212-ND
- **LM431CCZX variable zener**: Digi-Key/H212-ND
- **HLMP-K150 low current LED**: Digi-Key/H212-ND
- **560 ohm 5% 1/4W resistor**: Digi-Key/H212-ND
- **40.2K 1% 1/4W resistor**: Digi-Key/H212-ND
- **5.1K 5% 1/4W resistor**: Digi-Key/H212-ND
- **24.9K 1% 1/4W resistor**: Digi-Key/H212-ND
- **Slide switch**: Digi-Key/H212-ND
- **X60008CIS8-50**: Digi-Key/H212-ND
- **2-56 x 3/8” screw**: Digi-Key/H212-ND
- **2-56 nut**: Digi-Key/H212-ND
- **Printed circuit board**: Digi-Key/H212-ND

**Supplier/Part No.**

- **Duracell/MN1604**
- **Mouser/12BH611**
- **Mouser/140-XRL16V4.7-RC**
- **Mouser/581-SR215C103KAR**
- **Mouser/80-C315C102K5R5TA**
- **Mouser/625-1N5819-E3**
- **Mouser/78-1N914**
- **Mouser/512-LM431CCZX**
- **Mouser/638-HLMPK150**
- **Mouser/291-560-RC**
- **Mouser/271-40.2K-RC**
- **Mouser/291-5.1K-RC**
- **Mouser/271-24.9K-RC**
- **Mouser/688-SSSS22700**
- **Digi-Key/X60008CIS8-50**
- **Digi-Key/H130-ND**
- **Digi-Key/H212-ND**

---

**Additional Products**

- **Slide switch**: Digi-Key/H212-ND
- **Printed circuit board**: Digi-Key/H212-ND

---

**Notes**

- **Parts List**
- **Supplier/Part No.**
- **Additions**: Digi-Key/H212-ND
- **Removals**: Digi-Key/H212-ND

---

**Additions**

- **Parts List**
- **Supplier/Part No.**
- **Additions**: Digi-Key/H212-ND
- **Removals**: Digi-Key/H212-ND

---

**Removals**

- **Parts List**
- **Supplier/Part No.**
- **Additions**: Digi-Key/H212-ND
- **Removals**: Digi-Key/H212-ND

---

**September 2007**
Build Your Own: THEREMIN

Theremax
The Ultimate Theremin

BAIA
Electronic Music Kits and More

360 697-3472 voice
360 697-7717 fax
pioneer@telebyte.com

Features
FFT to 1048576 pts
24 bit, 192kHz
1/96 Octave
Dual Channel
3-D Surface Plot
Spectrogram Plot
Digital Filtering
Signal Generation
RT-60, Leq, Delay
THD, IMD, SNR
DDE, Logging
Vibration Analysis

SpectraPLUS 5.0
FFT Spectral Analysis System
Download 30 day FREE trial!

www.spectraplus.com

Starting at $295

Pioneer Hill Software
24460 Mason Rd
Poulsbo WA 98370
pioneer@telebyte.com

Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
With our popular Servo Erector Set you can
easily build and control the robot of your dreams.
Our interchangeable aluminum brackets make
the ultimate precision mechanical assemblies.
Our Visual Sequencer for SSC-32 provides
powerful PC, Atom, or BS2 based control.
Visit our huge website to see our complete line of
aluminum and Lexan based robot kits,
electronics, and mechanical components.

www.lynxmotion.com
Toll Free: 866-512-1024

56 NUTS & VOLTS September 2007
Testing USB RF

PDS5022S with 7.8" color LCD.

25MHz or 60MHz 2-ch + trigger

unique ID and Internal EEPROM,
new "security-dongle" feature. Each

FT232RL

Devasys - Provides a simple "drop-in" solution for connecting your PC to 500kHz I2C + 20 I/O lines. Free software. Use multiple boards for more PC/IO.

I2C

USB to I2C

Blue 320 x 240 LCD

Ethernet to I2C

I2C for PCs

I/O

Stepper Motor

Touchpanel Controller

USB to 24 x I/O

FATfile Storage

Easy LCD

I2C

Stepper Motor - Advanced stepper motor drive and control system for unipolar stepper motors up to 15V/0.3A - ideal for robotics/industrial control - turns you into an instant motor control expert from $84

Easy USB

Pocket-sized logger

RS-232 port compatible with most

UM232R        $359

time, accurate, inexpensive!

USB in one IC

FT232RL Module

USBwiz

USBwiz - Easily add USB peripherals to your system. Includes FATX system for USB thumb and external USB hard drive. No knowledge needed. Plug-n-play IC 109 Demo board $59

USB

microUSB - Tiny ready-made USB interface to 1 Mbps (0.7" x 0.5"

www.saelig.com

NEW

TinyOne Zigbee

TinyOne Zigbee - World's best capacitive touch IC for switching & control. Patented Adj. Key Suppression. Spread-spectrum. In use worldwide by the world's largest consumer appliance mfr's from $1

RF

Emgis Pi-Bule - 100m-Bit Bluetooth module (UART master/slave bridge).


LPRs easyRADION™ makes wireless data transmission simple for US and Europe and adds wireless capability to your project today!

肽Q bI9,AN     from $38 (100)

Ethernet

Serial-Ethernet Cable

Ethernet to Serial

Ether-IO


WiZnet TCP/IP IC

10/100 Ethernet controller on a chip. Fully hardwired TCP/IP core. No OS required. TCP offload improves system performance.

M320422422485 device server for

RIAAG a         from $17

URSA

FS serial data in - FATfile data out. USB powered radio transceiver boards add wireless RS232 to your project! Get remote control for your product from far away!

TinyOne Zigbee

f$5

USB-powered radio transceiver boards add wireless RS232 to your project! Get remote control for your product from far away!

TinyOne Zigbee

f$5

USB-Serial Adapter

USB-Wiz

Easy USB

microUSB - Tiny ready-made USB interface to 1 Mbps (0.7" x 0.5"
based on SiLabs CP2101. GNAR/TX/SPDX suspend, Reset, System OFF, power save mode. Plug-n-play IC 109 Demo board $59

microUSB - Tiny ready-made USB interface to 1 Mbps (0.7" x 0.5"
based on SiLabs CP2101. GNAR/TX/SPDX suspend, Reset, System OFF, power save mode. Plug-n-play IC 109 Demo board $59

www.saelig.com

info@saelig.com

Pen Scope

Logic Analyzer

ZeroPulse - Multifunction full-featured 16-channel 4MB memory. Sophisticated triggering, IC, SPI, and UART monitoring. Includes USB power, USB cable and software.

AP-16128U $299

NEW

SuperTrackStick II

TrackStick II - Product-sized stick-

FTDI - Based serial port laptop adapters adapt older RS232 products to USB! Included drivers provide virtual COM port.

USB-Wiz

Easy USB

microUSB - Tiny ready-made USB interface to 1 Mbps (0.7" x 0.5"
based on SiLabs CP2101. GNAR/TX/SPDX suspend, Reset, System OFF, power save mode. Plug-n-play IC 109 Demo board $59

microUSB - Tiny ready-made USB interface to 1 Mbps (0.7" x 0.5"
based on SiLabs CP2101. GNAR/TX/SPDX suspend, Reset, System OFF, power save mode. Plug-n-play IC 109 Demo board $59

www.saelig.com

info@saelig.com

new products every day!

www.saelig.com

new products every day!

www.saelig.com

new products every day!

www.saelig.com

new products every day!
began my career as a hardware engineer many years ago, but I’ve spent the last 12 years writing Java software for large commercial applications; far removed from the underlying hardware on which the applications run. Only recently (after reading all the back issues of Nuts & Volts and MAKE magazines available in my local library) have I had the urge to get back to the basics and write some software that runs directly on top of the hardware. The hardware, in this case, being small microprocessors or microcontrollers. I could tell from reading the various articles the authors were having fun experimenting with the small computers and controlling just about anything they could think of. It had been a long time since I had designed hardware or programmed such small computers. You could say I have come full circle.
The event that finally pushed me over the edge and down this slippery slope was I saw an ad for a microcontroller development kit from Texas Instruments (TI) called the eZ430-F2013 that could be had for $20 plus shipping. At first I didn’t believe it was possible. I thought the ad was a misprint but it wasn’t. That did it. I had to have one. It turns out this development kit is an incredible deal, perfect for experimentation with what turns out to be the “world’s lowest power” microcontroller family. A $20 development kit does truly bring microcontroller programming and experimentation to the masses. See the sidebar for more information about the eZ430-F2013 development kit.

Then the only question was, “What will I do with it?” It turns out the answer to this question came about as fast as the development kit came in the mail. For years, I have wanted to design a state-of-the-art color organ similar to the ones I built in my younger days, but this time do it to the extent possible in the digital domain (instead of the analog).

For those not familiar with what a color organ is, it is a device that splits music up into numerous frequency bands and modulates colored lights according to the musical content. Typical color organs are three or four channels with one color of light associated with each frequency band. With a color organ, you can see the music, as well as hear it.

Further fueling the desire for a state-of-the-art color organ was the availability of inexpensive super bright LEDs. The color organs I built in the past used incandescent lights which introduced a time lag into the musical response as the light’s filaments had to heat up and cool down. If LEDs were used instead, not only would this time lag be eliminated, but due to the life expectancy of today’s LEDs, you would probably never have to replace one.

So, I had my development kit in hand which included a bunch of documentation, a single target board, an emulator, C compiler, assembler, linker, and debugger and I had my idea of building a digital color organ. Could my grandiose idea be implemented on such a small microcontroller? My initial answer to this question was a definite maybe. Take a look at Tables 1 and 2 and let’s compare the capabilities of the MSP430-F2012 microcontroller to the requirements of my application and see.

After contrasting the controller’s capabilities with my application’s requirements, it was immediately apparent that if all of the required functionality were going to fit in the Flash memory, the overhead imposed by use of a high level language (HLL) could not be tolerated. This was true both in terms of the space required by a runtime library

<table>
<thead>
<tr>
<th>TABLE 1. MSP430-F2012 CAPABILITIES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attribute</strong></td>
</tr>
<tr>
<td>Clock speed</td>
</tr>
<tr>
<td>Flash memory</td>
</tr>
<tr>
<td>RAM</td>
</tr>
<tr>
<td>ADC</td>
</tr>
<tr>
<td>PWM</td>
</tr>
<tr>
<td>Timers</td>
</tr>
<tr>
<td>Power</td>
</tr>
<tr>
<td>I/O capabilities</td>
</tr>
<tr>
<td>Hardware Multiplication</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE 2. APPLICATION REQUIREMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attribute</strong></td>
</tr>
<tr>
<td>At least four channels of frequency selective lighting control</td>
</tr>
<tr>
<td>Noise gate</td>
</tr>
<tr>
<td>AGC</td>
</tr>
<tr>
<td>Audio inputs</td>
</tr>
<tr>
<td>Output lighting</td>
</tr>
<tr>
<td>Internal timing control</td>
</tr>
</tbody>
</table>
and/or for the performance hit of an interpreted language like some Basics. If this color organ was going to work, I would need to code in assembly language and do so as efficiently as possible.

The first hurdle I had to face was that digital filters of the IIR (infinite impulse response) variety are typically implemented using difference equations of the form:

\[ y[n] = A \cdot y[n-1] + B \cdot y[n-2] \]

where \( y \) is a three element array of output history and \( A \) and \( B \) are floating point filter coefficients that are determined from the type and response of the filter being implemented. To use such filters, I needed to be able to multiply floating point numbers on a microcontroller that doesn’t even know how to multiply integers. Floating point arithmetic is something one takes for granted when using a HLL, but as I mentioned this was not an option here.

**Horner’s Method to the Rescue**

In my investigation of how floating point arithmetic might be done, I stumbled across a TI application note (see Resources) that described Horner’s method for floating point multiplication and division. Horner’s method provides reasonably accurate results while only requiring shift/rotate and add instructions; something the MSP430 controller family has and does very well in a single clock cycle.

The remainder of this article describes this method along with a technique called Canonical Sign Digit or CSD that can be used to optimize the Horner results. Finally, I will present a program I wrote called Horner.java, that generates Horner equations and can even generate the MSP430 code for performing floating point multiplications and divisions. Note: Even though the focus here is on the MSP430 family of microcontrollers, the techniques presented are applicable to any processor.

I won’t delve into the theory behind Horner’s method; instead, I will show you how the theory is applied. (The Resources sidebar has some pointers to arithmetic theory for those interested in reading further.) Suffice it to say, Horner’s method allows multiplication and division of signed and/or unsigned floating point numbers using a rather elegant approach which I will illustrate. Horner’s method is not without its drawbacks, however, which include:

- When multiplying and dividing, the multiplier or divisor must be known in advance and cannot be changed dynamically. This is not an issue with most DSP applications because most coefficients are known at design time and don’t change at runtime. (This is true in general, but probably not true of everyone’s applications.)
- A division remainder is not available using Horner’s method like it is in most other division techniques. (Why this is so will be shown later.)
- The results of multiplication and division are rarely 100% accurate and the errors vary depending upon the types of numbers being processed. (I will illustrate this with examples shortly.)

For my color organ application, none of these drawbacks turned out to be significant. The process of using Horner’s method can be broken down into the following series of steps. Given a known multiplier or divisor:

1) If dividing, invert the divisor so that it becomes a multiplier.

2) Convert the multiplier to its binary representation of the required bit length; usually 10, 12, or 16 bits.

3) Optionally apply CSD to the binary representation to optimize it. (CSD will be described later.)

4) Generate the Horner equations for the multiplication from the binary representation.

5) Generate the computer code which implements the Horner equations.

The example that follows will show how this is done. It is actually kind of fun to generate the equations and the code by hand the first couple of times to aid in really understanding the process. Horner.java came about because I had to generate a lot of multipliers and the fun quickly turned to tedium. Let’s generate the equations for the multi-

“... a color organ is a device that splits music up into numerous frequency bands and modulates colored lights according to the musical content ... With a color organ, you can see the music, as well as hear it.”
plier 0.123456. We will do this using 12 bits and no CSD. The 12 bit binary representation of 0.123456 is 0.0001111111001. Starting from the rightmost bit which is the Least Significant Bit (LSB) and moving towards the Most Significant Bit (MSB), find the first one bit in the binary representation. This occurs at the 2^-12 bit LSB position. From there, count the number of places to the next one bit which, in this case, is three. The number of places between the one bits is called the distance and a distance of three results in the first Horner equation:

\[ T_1 = X \times 2^{-3} + X \]

Here, X represents the number the multiplier will be multiplied by. T1 is just a temporary accumulator for the calculation. The final X is added in because there are more one bits remaining in the representation. The complete set of Horner equations for our multiplier is then:

\[
\begin{align*}
T_1 &= X \times 2^{-3} + X \\
T_2 &= T_1 \times 2^{-1} + X \\
T_3 &= T_2 \times 2^{-1} + X \\
T_4 &= T_3 \times 2^{-1} + X \\
T_5 &= T_4 \times 2^{-1} + X \\
T_6 &= T_5 \times 2^{-1} + X \\
\text{Result} &= T_6 \times 2^{-4}
\end{align*}
\]

The 2^-4 in the final equation is the bit weight of the leftmost one in the binary representation.

Let’s check our results. If we were to set X to 1,000, the result we would expect on a calculator, for example, would be 123.456. Using 1,000 in the Horner equations yields 123 for an error of 0.456, which is less than one least significant bit. Not bad accuracy for many applications.

Division, as mentioned, is accomplished by multiplication using the inverse or reciprocal of the divisor. That is X / Y is the same as X * 1/Y. So, if you were trying to divide 100/10 with Horner, you would actually be multiplying 100 * 0.1. Now you see why the remainder from the division isn’t available; there really is no division.

Separate sets of Horner equations are generated for a multiplier or divisor that has both a decimal and a fractional part. The decimal portion of the number is processed from the MSB to the LSB, just opposite the direction the fractional part of the number is processed. Another difference is that the distances between ones in the decimal portion of the number are positive values which result in positive powers of two in the Horner equations. Consider a multiplier of 654.321.

Decimal representation:
001010001110

The equations for the decimal portion of the multiplier are:

\[
\begin{align*}
T_1 &= X \times 2^{-1} + X \\
T_2 &= T_1 \times 2^{-1} + X \\
T_3 &= T_2 \times 2^{-1} + X \\
T_4 &= T_3 \times 2^{-1} + X \\
T_5 &= T_4 \times 2^{-1} + X \\
T_6 &= T_5 \times 2^{-1} + X \\
\text{Decimal result} &= T_6 \times 2^{-4}
\end{align*}
\]

Notice the positive powers of two in the above equations.

The equations for the fractional portion of the multiplier are:

\[
\begin{align*}
T_1 &= X \times 2^{-4} + X \\
T_2 &= T_1 \times 2^{-3} + X \\
T_3 &= T_2 \times 2^{-2} + X \\
\text{Fraction result} &= T_3 \times 2^{-2}
\end{align*}
\]

Total Result = Decimal result + Fraction result

---

**Listing 1**

```
mov  #29,x  ; Set the multiplicand
mov  x,acc  ; Do the decimal portion of the multiply
rla  acc
rla  acc
add  x,acc  ; acc = x * 2^2 + x
rla  acc
rla  acc
rla  acc
add  x,acc  ; acc = acc * 2^4 + x
rla  acc
add  x,acc  ; acc = acc * 2^1 + x
rla  acc
add  x,acc  ; acc = acc * 2^1 + x
rla  acc
add  x,acc  ; acc = acc * 2^1
mov  acc,out ; Save the decimal result in out
mov  x,acc  ; Reload x for the fractional part
rra  acc
rra  acc
rra  acc
add  x,acc  ; acc = acc * 2^-4 + x
rra  acc
rra  acc
rra  acc
add  x,acc  ; acc = acc * 2^-3 + x
rra  acc
rra  acc
add  x,acc  ; acc = acc * 2^-2 + x
rra  acc
rra  acc
add  acc,out ; out = dec result + frac result
```

---

Using a calculator to again check our results when X is equal to 29, the product should be 18975.309. Running 29 through the Horner equations yields 18966 for the decimal result and 9 for the fractional result; their sum being 18975. Here, the error is again less than the least significant bit.

Generating code to implement these equations is easy. A rotate right instruction (“rra” for the MSP430 family) is used for every negative power of two and a rotate left instruction (“rla”) is used for each positive power of two. For every +X operation, an “add” instruction is used. For every -X operation (discussed in the context of CSD), a “sub” instruction is used. Assuming the symbols “x,” “out,” and “acc” are register aliases, the code is as shown in Listing 1.

This may seem like a lot of instructions, but remember, each register operation executes in one clock cycle on the MSP430 family of controllers. The above multiplication executes in about 32 clock cycles. If the processor is running at 16 MHz, the total execution time is a respectable two microseconds.

Of course, a C programmer would be able to perform the multiplication in a single line of code such as:

```c
double result = 29 * 654.321;
```

and get exact results. Such is the plight of the assembly language programmer.

**Canonical Sign Digit**

Canonical Sign Digit or CSD can be used to reduce the number of arithmetic operations (rotates and adds) required to implement the Horner equations. CSD is an optimization resulting in fewer instruction executions required to achieve the same result. CSD is a processing step that is inserted in the process flow where previously described. If a multiplier or divisor consists of decimal and fractional parts, the CSD process is applied across the complete binary representation. CSD processing proceeds from the LSB bit to the MSB and works by grouping adjacent one bits in the binary representation and replacing them with a simpler term.

To do this, it is required that our binary representation consisting of ones and zeros be replaced with a trinary representation consisting of three symbols (0, 1, and -1). The -1 is a special symbol that indicates some number of consecutive ones (a run of ones) have been replaced. After the application of CSD, there won’t be any consecutive ones left in the binary representation of a number. It is necessary to restart CSD processing after each run replacement as a new run of ones may be created by the insertion of the replacement.

CSD processing is performed in a loop as follows:

---

**The TI eZ430-F2013 Development Kit**

Yes it’s true. You can get a complete TI MSP430 microcontroller development kit for $20 plus shipping. It’s called the eZ430-F2013 and is in the form of a USB stick (envision a jump drive). Amazingly, both the debugger interface and a target board fit inside the stick. This development system works with all MSP430F20xx devices. Even someone as frugal as myself could not pass up a deal like this. This offer puts microcontroller development within everyone’s reach.

The following information was extracted from the TI website.

**Description**

The eZ430-F2013 is a complete MSP430 development tool including all the hardware and software to evaluate the MSP430F2013 and develop a complete project in a convenient USB stick form factor. The eZ430-F2013 uses the IAR Embedded Workbench Integrated Development Environment (IDE) to provide full emulation with the option of designing with a stand-alone system or detaching the removable target board to integrate into an existing design. The USB port provides enough power to operate the ultra low power MSP430 so no external power supply is required.

**Features**

- eZ430-F2013 development tool including a USB debugging interface and detachable MSP430F2013 target board
- LED indicator
- Removable USB stick enclosure
- Debugging interface supports development with all MSP430F20xx devices
- Integrated IAR Kickstart user interface which includes an assembler, linker, simulator, source-level debugger, and limited C compiler
- Full documentation on CD-ROM

**What’s Included**

- CD-ROM including software and documentation
- IAR Embedded Workbench (Kickstart Version) IDE
- eZ430-F2013 Development Tool

In addition to the development system, a set of three MSP430-F2012 target boards is available for $10. TI’s part number for these boards is eZ430-T2012. These boards are about the size of a quarter cut in half. The F2012 processor itself is about the size of the fingernail on your little finger. I used one of these target boards for my color organ project.

The included C compiler is limited to the production of 4K of code which means it cannot be used on the largest of the MSP430F20xx devices. TI makes available a free C compiler on its website that can generate up to 8K of code. Since my first MSP430 project was written entirely in assembly language, I didn’t try out this alternative C compiler.
While there are consecutive ones in the representation
Starting at the LSB of the binary representation search leftward for a run of ones.
Create and insert a replacement string for the run of ones.

An example will illustrate the point. Say we have a multiplier M. Its binary and CSD representations are shown next:

**Binary representation:** 0.0011100000

**CSD representation:** 0.0100-100000

As you can see, the -1 was inserted at the start of the run of ones. Zeros replace the ones in the run and finally a new one bit replaces the zero that broke the run of ones. You can probably surmise by looking at the trinary representation that the CSD conversion will result in one fewer Horner

---

**Running Horner.java**

To run Horner.java, you will need to have a Java Development Kit (JDK) installed on your computer. The program was written for use with Java 5 which is available for free from Sun Microsystems at [http://java.sun.com/javase/downloads/index.jdk5.jsp](http://java.sun.com/javase/downloads/index.jdk5.jsp). The newer version 6 of Java should also work. An executable version of the Horner.java program, as well as the source code, are available at the Nuts & Volts website ([www.nutsvolts.com](http://www.nutsvolts.com)) in this article's archive file: Horner.jar.

The code can be run in a couple of ways. First, the source file Horner.java can be extracted from the jar file, compiled, and then run using a command shell (cmd.exe) with the following procedure:

```
jar xf Horner.jar  This command extracts the source file from the jar
javac Horner.java  This command compiles the source file
java Horner <command line arguments> This is how you run the program
```

Alternately, the code can be run directly from the jar file without having to extract anything. This is done as follows:

```
java -jar Horner.jar <command line arguments>
```

If neither of these methods work successfully, make sure the bin directory of the JDK installation is on the Path in your command shell. Alternately, you can change directories to the bin directory and run the commands from there.

Horner.java has quite a few command line argument possibilities. Help is available by executing the program (using either of the above methods) without specifying any arguments. You should see the following:

```
No command line arguments specified
```

Horner.java V1.0 copyright 2007 by: Craig A. Lindley
Generates Horner equations and MSP430 code for multiplication using only shifts and adds

Program usage is as follows:

```
java Horner [nocsd] [csd] [bin] [16bit] [12bit] [10bit] [code] [equ] [both] [help | ?] multiplier
```

Where:

- nocsd - use binary not csd numbers in the calculations
- csd - use csd optimizations in the calculations
- bin - show binary representation of multiplier
- 16bit - use 16 bit binary precision
- 12bit - use 12 bit binary precision
- 10bit - use 10 bit binary precision
- code - generate the MSP430 code for the multiplication
- equ - generate Horner equations for the multiplication
- both - generate the MSP430 code and the equations for the multiplication
- help or ? - displays this message
- multiplier - the floating point number being multiplied. The multiplier must be the final argument.

Program defaults: 16bit csd both
Please try again

To see the Horner equations and the code generated for a multiplier, specify the multiplier as the final command line arguments as in the following:

```
java -jar Horner.jar 0.12345
```

which will produce the following results:

**Multiplier:** 0.12345 **Bits of precision:** 16

**Horner equations:**

T1 = X * 2^-2 - X  
T2 = T1 * 2^-2 + X  
T3 = T2 * 2^-2 - X  
T4 = T3 * 2^-6 + X  
Fraction result = T4 * 2^-3

**MSP430 code:**

```
Code conventions:
Register “in” has the X value to be multiplied
Register “acc” is the accumulator used for temporary values
Register “out” holds final result

mov  in,acc
rra  acc
rra  acc
sub  in,acc
rra acc
add in,acc
rra acc
sub in,acc
rra acc
add in,acc
rra acc
mov  acc,out
```

You can play around with the program using the various command line arguments to see what their effects are.
equation with its rotate and add. Therein lies the optimization.

Generating the Horner equations for a CSD optimized multiplier is the same as was described previously, except instead of adding +X to each equation, -X is added for each -1 in the representation.

Here are the CSD optimized Horner equations for our previous example of 654.321:

**Binary representation:**
001010001110.010100100010

**CSD representation:**
0010100100-10.010100100010

**Horner equations:**
\[ T_1 = X \cdot 2^2 + X \]
\[ T_2 = T_1 \cdot 2^3 + X \]
\[ T_3 = T_2 \cdot 2^3 - X \]

Decimal result = \( T_3 \cdot 2^1 \)

\[ T_1 = X \cdot 2^{-4} + X \]
\[ T_2 = T_1 \cdot 2^{-3} + X \]
\[ T_3 = T_2 \cdot 2^{-2} + X \]

Fraction result = \( T_3 \cdot 2^{-2} \)

Total Result = Decimal result + Fraction result

You can see there is one less equation as a result of the CSD optimization.

**Horner.java**

As mentioned, Horner.java is a program I wrote to quickly generate Horner equations and executable code for performing floating point multiplication and/or division on the MSP430 family of microcontrollers. Check out the sidebar that shows how to run this program and what program options are available. The source code for Horner.java can be extracted from the included jar file listed in the sidebar. You may like to look at the code to see how signed multiplications are performed.

**Conclusions**

Horner’s method can be used to provide floating point support on microprocessors or microcontrollers that don’t have support built in. It is ideal for applications like mine, that aren’t written using a high level language. Horner’s method was just what I needed for my color organ application. After developing the digital filters using Horner’s method, I can say that the code for the color organ project can fit in the 2K bytes of Flash memory provided by the F2012 microcontroller (though just barely).

It may seem like a lot of work to use the techniques presented here, but often optimizing performance and/or memory usage of an application usually is work. NV
Metal Fabrication

1-2 Day Lead Times Available

Both of these parts were ordered on Tuesday and shipped FedEx on Wednesday.

Materials from 0.001” to 6.000” in thickness.
Parts from practically any material.

* Quick
* Affordable
* Precise
* No Minimums

Complete Fabrication Center
Integrated Ideas & Technologies, Inc.
Precision Laser, Waterjet, Plasma, Machining, Forming, and Welding Capabilities

3896 N. Schreiber Way • Coeur d’Alene, ID • 83815-8362 • USA
Ph (208) 665-2166 • Fax (208) 665-5906 • www.iitmelfab.com

---

CIRCUIT BOARDS

Manufactured Domestically Since 1984

As low as...

$9.95 each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards for even less!

e-pcb.com

---

Pick Your Voltage
(not your nose)

AnyVolt Micro — Steps voltage down or up. Great for solar powered projects. $20
Negatron — Takes positive voltage and generates an adjustable negative voltage. $20
DE-SWADJ — Steps voltage down and has adjustable output. $15

www.dimensionengineering.com

---

Quick
Affordable
Precise
No Minimums

---

---

---

---

---

September 2007 NUTS&VOLTS 65
go to www.nutsvolts.com click Electro-Net
FLAT-SCREEN TV STEREO SPEAKERS

Full-range 15W, 4 ohm stereo speakers for Phillips flat-screen television. Very good sound reproduction produced by two 2” 8W woofers and one 1” 15W tweeter. An oval 4.7” x 1.9” resonant diaphragm helps accentuate the bass response. Unbreakable plastic enclosures are 15” x 3” x 2.55” deep. Set of two speakers, left and right.

CAT# SK-314

$8.00 per pair

10 for $75.25 each

20W HALOGEN LAMP, 6V OPERATION

6 Volt, 20 Watt MR-16 halogen lamp. 1.97” diameter reflector with cover glass. 12” wire leads with 2-pin female connector and mating male header. 0.1” centers. 38 Degree, wide angle.

CAT# HLP-206

$2.00 each

10 for $1.80 each

100 for $1.65 each

S-VHS TAPE, TDK SUPER PRO-ST-160

Extra high output, very low noise, ultra-refined Super Avilyn formulation delivers outstanding picture definition and audio reproduction. 327 meters. EP=480 min., SP=160 min.

CAT# S-VHS

$3.00 each

10 for $2.75 each

MINIATURE 12VDC PUSH-TYPE SOLENOID

Ledex™, 0.5” dia. x 1” long tubular solenoid with 3/8” diameter threaded bushing for mounting; 1/16” diameter x 1/2” long plunger. 40 Ohm coil. Operates on 12 Vdc. 1” wire leads.

CAT# SOL-102

$3.25 each

2 SELF-STRIPPING BANANA PLUGS

Sturdy, gold-plated banana plugs designed for speaker wire connections. Requires no stripping and no special tools - just cut and connect to 16-12 Gauge wire. Sold in pairs. Retail card packaging.

CAT# BP-92

$3.75 per pair

2-CHANNEL LINE-OUT CONVERTER

Linear Research #LR9-801. Two-channel speaker-level to RCA adapter with adjustable gain controls for each channel. Allows you to add an aftermarket amplifier to auto radios that don’t have line-out capability.

CAT# LOC-1

$4.50 each

SOLAR CELL

Output: approximately 3 Volts @ 40 mA.

100 for $3.25 each

38mm square x 2.5mm thick epoxy-encapsulated silicon photovoltaic cell. Solid, almost-unbreakable module with solderable foil strips on backside. Ideal for solar-powered battery chargers and other projects.

CAT# SPL-61

$3.75 each

EL BACKLIGHT, 3” X 5”

Salmon color in off-state. Glows light-green when energized by 120 Vac or Inverter. For backlighting control panels, LCDs, special-effects lighting, models etc. Solderable pins extend 0.16” beyond plastic laminate exterior.

CAT# EL-13

$9.75 each

10 for $92.25 each

12VDC 300MA SWITCHING POWER SUPPLY

Friwo # SPA4UL. Input: 100-240V 50-60Hz.

Output: 12Vdc 300mA.

6’ cord with 2.1mm coax power plug, center +.

CAT# PS-1236

$5.50 each

S-VHS TAPE, TDK SUPER PRO-ST-160

Extra high output, very low noise, ultra-refined Super Avilyn formulation delivers outstanding picture definition and audio reproduction. 327 meters. EP=480 min., SP=160 min.

CAT# S-VHS

$3.00 each

10 for $2.75 each

Video AV Cable

3.5mm (4-conductor) plug on one end, three RCA jacks (red, white, yellow) on the other end. 3 ft cable. Allows iPods and many camcorders to connect directly to television or video monitor via standard A/V stereo cable.

CAT# CB-218

$3.00 each

20 for $1.50 each

100 for $3.25 each

BANANA PLUGS

Sturdy, gold-plated banana plugs designed for speaker wire connections. Requires no stripping and no special tools - just cut and connect to 16-12 Gauge wire. Sold in pairs. Retail card packaging.

CAT# BP-92

$3.75 per pair

2 SELF-STRIPPING BANANA PLUGS

Sturdy, gold-plated banana plugs designed for speaker wire connections. Requires no stripping and no special tools - just cut and connect to 16-12 Gauge wire. Sold in pairs. Retail card packaging.

CAT# BP-92

$3.75 per pair

2-CHANNEL LINE-OUT CONVERTER

Linear Research #LR9-801. Two-channel speaker-level to RCA adapter with adjustable gain controls for each channel. Allows you to add an aftermarket amplifier to auto radios that don’t have line-out capability.

CAT# LOC-1

$4.50 each

SOLAR CELL

Output: approximately 3 Volts @ 40 mA.

100 for $3.25 each

38mm square x 2.5mm thick epoxy-encapsulated silicon photovoltaic cell. Solid, almost-unbreakable module with solderable foil strips on backside. Ideal for solar-powered battery chargers and other projects.

CAT# SPL-61

$3.75 each

EL BACKLIGHT, 3” X 5”

Salmon color in off-state. Glows light-green when energized by 120 Vac or Inverter. For backlighting control panels, LCDs, special-effects lighting, models etc. Solderable pins extend 0.16” beyond plastic laminate exterior.

CAT# EL-13

$9.75 each

10 for $92.25 each

12VDC 300MA SWITCHING POWER SUPPLY

Friwo # SPA4UL. Input: 100-240V 50-60Hz.

Output: 12Vdc 300mA.

6’ cord with 2.1mm coax power plug, center +.

CAT# PS-1236

$5.50 each

S-VHS TAPE, TDK SUPER PRO-ST-160

Extra high output, very low noise, ultra-refined Super Avilyn formulation delivers outstanding picture definition and audio reproduction. 327 meters. EP=480 min., SP=160 min.

CAT# S-VHS

$3.00 each

10 for $2.75 each
The chip employs a single memory controller from IBM. The Cell is aided by a dual-channel off-chip I/O interface from Rambus.

According to Andi Smithers, a Sony games developer, while the Cell typically has a PowerPC core and eight SPE cores, the Cell chip in the PS3 has a PowerPC and only seven—not eight—SPEs.

The operating system (OS) always takes one core for itself, leaving six (actually 5.5 cores are left, because one core has to be in a unique mode and the OS also needs to use part of it), according to Smithers.

**How Fast?**

How fast does the Cell chip cruise? According to Peter Hofstee, a Cell chip architect at the Sony–Toshiba-IBM (STI) design center, IBM, the Cell/B.E. is about one-third the speed that IBM’s Deep Blue supercomputer was.

The Roadrunner supercomputer will use Cell chips to deliver a Petaflop, or 1,000 Teraflops, according to Hofstee. “At about 300 chips per wafer (assuming 100% yield), one hundred wafers would provide all the processors required for the Roadrunner supercomputer [currently under construction],” says Hofstee.

“The frequency of the Cell/B.E. is 3.2 GHz. Marketplace PCs have comparable frequencies. Unlike PC processors, the Cell/B.E. isn’t marketed at different speeds. But, in the lab, we run Cell/B.E.-based systems at more than 6 GHz,” commented Hofstee.

Element Interconnect Bus technology—which runs at half the chip’s system clock rate—is used to optimize the speed between the cores.

The current 90 nm processor rides on a 300 mm [millimeter for this reference, not nanometer] wafer (the 65 nm model is in production as of March this year). Piling the wafers on top of each other expands their processing power in a modular architecture. The Cell is arguably the most broadly publicized commercial “system-on-a-chip” to date.

**Big Picture, Little Cell**

STI is constructing a new version of the Cell chip using fabrication technologies at the 65 nm level. STI fabricated the first commercial Cell at 90 nm. Commercial Cell chips sold to date use this technology. STI builds the Cell using Silicon On Insulator (SOI) transistors and low-K dielectric insulation.

Cell supports the high performance needs of future consumer electronics products, as well as blade servers and supercomputers. Consumer electronics and network products that will use broadband—products like games, movies, music, digital broadcasting—will benefit from the parallel processing speed and low power consumption of the Cell (the Cell’s DMA memory speed on the PS3 is approximately 25 GBps, according to IBM).

According to Hofstee, IBM based “QS20” blade server on the Cell; Mercury Computer based the PCI-express “Cell Accelerator Board” on it, as well. Creators IBM, Sony, and Toshiba will benefit from the technology by developing their own Cell-based products. Raytheon is also working with the Cell/B.E., according to Hofstee.

“IBM has a roadmap that includes at least two additional generations of Cell/B.E. based blades. Also, Los Alamos National Laboratory and IBM have announced their intent to build a supercomputer based on
Cell that should achieve 1 PetaFlop of sustained performance (about 1.4 Petaflop peak performance),” says Hofstee. (This is the Roadrunner supercomputer project.)

How Cell Technology is Catapulting Higher Education

“Because Sony Computer Entertainment, Inc. (SCEI), allows people to install the open source Linux operating system on the Playstation 3, many students already have easy access to a programmable Cell/B.E. based platform,” says Hofstee.

This Linux environment is the same on all Cell/B.E. based blades and systems up to and including the Roadrunner, according to Hofstee.

College and university students from 25 countries recently competed for monetary awards for the “most innovative” application programming based on the Cell chip, according to IBM media information.

The 2007 competitions (the inaugurals) were held from February through July, with the winners being announced around September 5th. In addition to the educational opportunities and experiences for students globally, the contest provides IBM with an on-going source of new application ideas for its new chip.

With the aid of IBM, MIT has recently finished teaching its first course based on the Cell chip. During the four-week course, students designed projects to run on the PS3 using “open standards” software, according to IBM.

The course applied the Cell inside the PS3 to an introduction to parallel programming (programming for multicore chips, of which the Cell is an example). Students formed groups to develop their applications, using the Cell SDK (development kit) from IBM’s developerWorks.

In the meantime, the Budapest Polytechnic Institution is starting a Cell Broadband Engine course this fall that will count towards a Bachelor of Science in Computer Science and Engineering, according to IBM media relations.

Intelligence Everywhere

Think about a future with intelligent clothing, glasses, and watches that work together via PANs (Personal Area Networks), to perform all the functions you do with a cell phone and other devices today.

People can embed intelligent buildings with Cell chips to amp up processing for environmental controls, communications between the home and homeowner, and appliances through sensors, monitors, and body-worn devices.

The Cell chip can enable intelligence everywhere, inside and out. Everything built today as separately functioning devices and applications can share intelligence to improve the performance of each.

The Cell has applications in HDTV sets and Internet access.

<table>
<thead>
<tr>
<th>CELL-BASED RESEARCH</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Cell/B.E. is central to several research efforts:</strong></td>
</tr>
<tr>
<td>1) The University of Washington held a workshop this August that was dedicated to the use of Cell/B.E. for medical imaging applications, according to Peter Hofstee, STG, Technology Development DE, Architect, STI Design Center, IBM. “Also, of course, we continue to work on image rendering,” says Hofstee.</td>
</tr>
<tr>
<td>2) In August, IBM “showed off” its Cell-based Interactive Ray Tracer (“iRT”) at the SIGGRAPH (Special Interest Group on Graphics and Interactive Techniques) conference, according to Hofstee. The iRT is a “scalable, ray-tracing engine” with interactive frame rates up to HDTV resolutions thanks to clustering multiple Cell processors, according to IBM.</td>
</tr>
<tr>
<td>3) “The folding@home project at Stanford is getting most of its performance on protein folding from contributing PLAYSTATION 3 systems [to its clusters; these of course house a version of the Cell chip],” says Hofstee.</td>
</tr>
<tr>
<td>4) “While perhaps not research,” comments Hofstee, “an interesting commercial application of the Cell/B.E. is a tool for making the masks to build chips (developed by Mentor Graphics and Mercury Computer). This tool will help us build next-generation processors and other chips.”</td>
</tr>
<tr>
<td>5) Many Universities use Cell/B.E.-based blade centers for application development, as well as research. The Boston University, for example, will use it to research “fragment-based drug design,” according to Hofstee. “Others will use it for research related to game development, virtual world technologies, or new image compression schemes, just to name a few,” he adds.</td>
</tr>
<tr>
<td>6) A broader focus of research is to make parallel programming for the Cell — as well as hybrid and multicore systems — easier for programmers, according to IBM’s Hofstee.</td>
</tr>
</tbody>
</table>
devices (both of which create a need for high bandwidth chips). Cell technology will make it possible to integrate several IP-addressable devices into one.

Companies like IBM and Mercury Computers are already producing Cell-based blade servers for High-Performance Computing (supercomputing) with lower power consumption.

The Cell has other advantages. “With nine processor cores with media capabilities in each, and the ability for software to control movement of data and code in detail, there are substantial performance benefits,” says Hofstee.

“Cell has unique capabilities in the areas of security,” he adds, “with the ability to configure each of the eight synergistic processor elements as a secure processor on the fly.” Cell also offers hardware support for real-time processing, “even when there are multiple tasks running in parallel (such as web browsing and streaming content to a TV),” Hofstee explains.

**Cell Sells Grid**

As technologies like the Cell chip proliferate and join one another through the Internet, there comes an economic incentive for people to move toward a global computer grid. The global computer grid or earth grid has been under discussion by scientists and the military for some time now (www.osf.org).

The grid will consist in grid databases and other valuable and incumbent grid technologies (sensors, mass massaging of the world's data collected by these sensors, and the list goes on).

Suddenly, the grid will insinuate itself into our lives. It won't work across traditional computers only, but cell phones can also share processing power and information, as well as can any other device or appliance.

“IBM made Cell/B.E. available to selected universities and research institutions via remote access early on, and a Cell/B.E.-based blade system was just put into service on the Chinese computing grid and is accessible by a number of universities there. Stanford's ‘folding@home’ is a type of loosely coupled grid computer,” says Hofstee.

Other Cell/B.E. applications (initially) include medicine, industry, and defense, adds Hofstee.

**Conclusion**

Applications and experimentation with the Cell/B.E. technology abound. Its long-term benefits depend on the imaginations, aspirations, and perspiration of those developing to it. **NV**
A Logic Analyzer Tutorial

PART 1

A logic analyzer — like so many electronic test and measurement tools — provides a solution to a particular class of problems. These include digital hardware debugging, design verification, and embedded software debugging. A logic analyzer is an indispensable tool if you design and troubleshoot digital circuits.

by Vaughn D. Martin

Logic analyzers simultaneously measure numerous digital signals with challenging trigger requirements. If you are using new devices, you’ll soon discover that debugging microprocessor-based designs requires more inputs than oscilloscopes can offer. Logic analyzers — with their multiple inputs — solve these problems. These instruments have steadily increased, both in their acquisition rates and channel counts, keeping pace with advancing digital technology.

There are similarities and differences between oscilloscopes and logic analyzers. To better understand how the two instruments address their respective applications, let’s compare their individual capabilities.

After triggering on a complicated sequence of digital events, a logic analyzer can copy large amounts of digital data from the SUT (system under test). This feature is particularly useful in capturing and analyzing complex digital signals.

What a Logic Analyzer Does

A logic analyzer solely measures digital, not analog signals! It can capture many digital signals simultaneously and display their often complex timing relationships to one another. However, some logic analyzers slightly transgress into the scope’s domain by detecting glitches and setup and hold timing violations. But mainly, logic analyzers debug elusive, intermittent signals, with some advanced ones even correlating source code with specific hardware problems.

HISTORY OF LOGIC ANALYZERS

The first logic analyzer appeared in 1967 as HP engineer Gary Gordon’s personal “bench project.” Just six years earlier, Gary became a company hero as an intern in the oscilloscope lab by solving their digital sampling scopes drifting problem when the time base changed.

Gordon’s involvement with digital oscilloscopes before inventing the logic analyzer is not surprising since logical analyzers evolved from DSOs. Logic analyzers evolved almost simultaneously with the first commercially available microprocessors. The first logic analyzers emphasized operations closely akin to oscilloscopes for hardware debugging and test, later becoming more concerned with monitoring microprocessor signal activity and software debugging.
More on the Digital Oscilloscope

The DSO is for general-purpose signal viewing. Its sample rate (up to 20 Gs/sec) and bandwidth enable it to capture many data points over a short span. This allows measurements of signal transitions (edges), transient events, and small time increments. While DSOs can view the same digital signals as a logic analyzer, most DSO users concentrate on analog measurements such as rise- and fall-times, peak amplitudes and the elapsed time between edges. Figure 1 illustrates the oscilloscope’s strengths. The waveform (though taken from a digital circuit) reveals the analog characteristics of the signal, all of which can have an effect on the signal’s ability to perform its function. Ringing, overshoot, roll off in the rising edge, and other aberrations appearing periodically exist here.

With a modern oscilloscope’s built-in tools such as cursors and automated measurements, it’s easy to verify signal integrity problems impacting your design. Purely analog signals — such as the output of a microphone or digital-to-analog converter — can only be monitored with an instrument that records analog details (see Figure 2).

Logic Analyzer Details

The most obvious difference between these two instruments is the number of channels (inputs). Typical digital oscilloscopes have up to four signal inputs. Logic analyzers have between 34 and 136 channels. Each accepts one digital signal. Some complex system designs require thousands of input channels. Appropriately scaled logic analyzers are available for those tasks, as well.

A logic analyzer detects logic threshold levels (see Figure 3). When the input is above the threshold voltage, the level is said to be “high” or “1,” conversely, the level below the threshold voltage is a “low” or “0.” When a logic analyzer samples an input, it stores a 1 or a 0, depending on the level of the signal relative to the voltage threshold.

A logic analyzer’s waveform timing display is similar to that of a timing diagram found in a data sheet or produced by a simulator. All of the signals are time-correlated and effectively show progressive system “snapshots” through time.

A logic analyzer’s digital design verification and debugging features — such as sophisticated triggering — allow you to specify the conditions under which the logic analyzer acquires data. High-density probes and adapters simplify connecting to the SUT. Analysis capabilities translate captured data into processor instructions and correlate it to source code.

Logic Analyzer Architecture and Operation

There are four steps to using a logic analyzer:
• Probe (connect to the SUT)
• Setup (clock mode and triggering)
• Acquire
• Analyze and display

Each block in the simple logic analyzer block diagram shown in Figure 4 symbolizes several hardware and/or software elements. The block numbers correspond to the four steps just listed. The acquisition probes connect to the SUT. The probe’s internal comparator is where input voltage comparison occurs against the threshold voltage and the signal’s logic state (1 or 0) is determined. You set the threshold value, ranging from TTL levels to CMOS, ECL, or your own user-definable ones.

Probe impedance (capacitance, resistance, and inductance) becomes part of the overall load on the SUT. All probes exhibit loading characteristics. The logic analyzer probe should introduce minimal loading on the SUT, and provide an accurate signal to the logic analyzer.

Probe capacitance tends to “roll off” the edges of signal transitions (see Figure 5). This roll-off slows down the edge transition by an amount of time represented as “t” in the figure. Remember, slower edges cross the logic threshold of the circuit later, introducing timing errors in the SUT. This problem becomes more severe as clock rates increase.

In high speed systems, excessive probe capacitance can potentially prevent the SUT from working. It is always critical to choose a probe with the lowest possible total capacitance. It’s also important to note that probe clips and lead sets increase capacitive loading on the SUT. Use a properly compensated adapter whenever possible. The impedance of the logic analyzer’s probe affects signal rise times and timing relationships.

Logic analyzers capture data from multi-pin devices and buses. The term “capture rate” refers to how often the logic analyzer samples the inputs. It is the same function as the time base in an oscilloscope. Logic analyzer literature interchangeably uses the terms “sample,” “acquire,” and “capture.”

Timing acquisition captures signal timing information. In this mode, an internal clock samples data. The faster it...
samples data, the higher the resolution of the resulting measurement. There is no fixed timing relationship between the target device and the data the logic analyzer acquires. Use this acquisition mode when you are concerned with the timing relationship between SUT signals.

The acquisition mode acquires the “state” of the SUT. A signal from the SUT defines the sample point (when and how often data is required). The clock signal you use in the acquisition mode may be:

- The system clock
- A control signal on the bus
- A signal that causes the SUT to change states

You sample data on the active edge, which represents the SUT when the logic signals are stable. The logic analyzer samples only when the selected signals are valid. What transpires between clock events is irrelevant.

In all instances, the “event” appears when signals change from one cycle to the next. You can use many conditions to trigger your logic analyzer, such as a specific binary value on a bus or counter output. Other triggering choices include:

- Words: Specific logic patterns defined in binary, hexadecimal, etc.
- Ranges: Events that occur between a low and high value.
- Counter: The number of events you program that you are tracking by a counter.
- Signal: An external signal such as a system reset.
- Glitches: Pulses that occur between acquisitions.
- Timer: The elapsed time between two events or the duration of a single event, tracked by a timer.
- Analog: Use an oscilloscope to trigger on an analog characteristic and to cross-trigger the logic analyzer.

With all these trigger conditions available, it is possible to track down system errors using a broad search for state failures, before you refine your search with increasingly explicit triggering conditions.

I hope this introduction has taught you a few things about this powerful benchtop tool — look forward to more in Part 2! NV
Robot fish swim like the real thing, robot snakes slither on the ground, crab-like robots scale mountain faces, and insect-like robots can stay aloft for several minutes at a time. Just as the environment selects for the life form with the optimal form of locomotion, the environment in which a robot must operate dictates the optimal form of locomotion.

Continue reading to learn how to build a simple robot designed to travel over steel or iron surfaces—a magnetic inchworm—with a few servos, a pair of neodymium magnets, and an R/C system.

### Inchworm Locomotion

Assuming a two-segment design, inchworm locomotion involves cyclically advancing the front segment of the robot while the rear remains fixed, and then fixing the front segment while pulling the rear segment forward. Perhaps the best known robot that employs inchworm locomotion is HeartLander, from Carnegie Mellon University.

As shown in Figure 2, the robot is only about two inches long, and consists of front and rear segments, each connected to a vacuum line. The robot is designed to walk along the surface of a beating heart. It accomplishes this feat with suction variably applied to the area below each segment. Guide wires between the segments enable the human operator to steer the robot to the appropriate area of the heart.

### Magnetic Inchworm

The magnetic inchworm project described here borrows from the HeartLander robot in the manner in which the robot is steered and in how the segment spacing is cycled to move the robot forward. Figure 3 shows the completed magnetic inchworm in a clockwise turn. Note the front segment is angled down and clockwise, in preparation for the pull on the rear segment. Because stainless steel rods connect front and rear segments, it’s important to have ample play in the front and rear segment connection points, as illustrated in the figure.

### Operation

Operation of the magnetic inchworm involves the use of a remote control unit to actuate the single...
You’ll require the following materials:

- One 2.5 x 5.5 x 1 inch plastic project box, cut in half about 1-3/4 inches from one end.
- Four GWS Servo Nano Pro/Std or equivalent servos. The Nano Pro is a 13g device that measures 27 x 12 x 22 mm. About $11 each from Tower Hobbies.
- About one foot of bare solid wire to mount the magnets to the servo horns.
- A four channel R/C transmitter/receiver system.
- Two 4-40 six-inch threaded stainless steel push rods with clevis, trimmed to 6”.
- Two closed solder lugs, one soldered to the unthreaded end of each push rod.
- Two 0.25” x 0.2”L Neodymium magnets; $1 for the pair, less shipping, from All Electronics.
- Two 1” 4-40 bolts and four nuts to mount the push rods to the front segment.
- Glue gun to mount the servos.
- Epoxy to secure the mounted magnets to the servo horns.

First, securely mount each magnet to a servo horn using the bare wire. I used a circular horn and drilled a pair of extra holes in the horn to accommodate two loops of wire to mount the magnet. Cover the magnets and the bare wire with epoxy, as shown in Figures 5 and 6. Each magnet face should be flush with the outer edge of its servo horn.

Next, take the plastic project box and (using a hacksaw) cut the box to create the front and rear segments. The dimensions of each segment depend on the servos you have available for the project. Mount the servo horns with magnets on two servos and fit a servo in each segment, noting where the magnet and horn must clear the case. Mark the case and then (using a drill) create an oval for the magnet and servo horns. The face of the magnet should just touch the surface below the segment, as in Figure 6. Note that the magnets rotate along the long axis of the robot.

Next, mount each of the pushrods on a 4-40 bolt; attach the bolts to either side of the front segment, using a drill to create holes for the bolts. Use a glue gun to mount the servos carrying the magnets. With a pair of diagonal pliers, cut notches in the case for each of the two remaining servos, as in Figure 6. Secure the servos with hot glue and attach the horns and pushrods. Place a 1” cube of foam over each servo and attach the case cover on the front and rear segments.

Plug the servos into the appropriate channels on your receiver and connect power to the receiver. I found a three foot, 22 gauge power cord provided ample flexibility. Finally, calibrate your transmitter and move the servo horns so that you can easily control the magnet servos and the position of the rear segment servo arms.
The magnetic inchworm, as described above, leaves significant room for innovation. Consider a few of the following modifications/additions:

- Substitute electromagnets for the servo-magnet assemblies. By varying current to the electromagnets, you’ll be able to finely tune the attraction of each segment of the inchworm to the underlying steel/iron surface. You’ll probably discover that climbing a steep incline requires more and longer duration electromagnet current than traversing a horizontal surface.

- Substitute other effectors for the magnets, such as soft rubber pads for hardwood floors and claws for carpet.

- Add more segments to the basic design, with a goal of enabling the worm to navigate and climb vertical structures.

- Use a microcontroller instead of an R/C unit. Either a Stamp or PIC would work nicely. Consider using an accelerometer to determine the angle of the surface and adjust the sequencing accordingly. For example, at a steeper grade, the robot is more likely to slip unless you increase the amount of time the magnets face the supporting iron or steel surface.

- Try flexible pushrods with less play in the connection points. The robot will handle better on flat surfaces, but may be hard to control on an incline or decline.

- Consider adding a wireless camera to the front segment for remote monitoring.

I like hot glue because it’s a great rapid prototyping tool. Once I’m happy with a design, removing servos and other components is relatively easy — they can normally be pried loose without damage to the components or the supporting chassis. However, if your goal is to build a magnetic wall climber, then cut back on all unnecessary weight. Use a dab of epoxy instead of hot glue to affix each servo, and don’t bother attaching the top cover on each segment.

FIGURE 6. Rear segment of inchworm showing three servos and magnet rotated toward surface cutout.

REFERENCES
[1] HeartLander photo courtesy of Nicholas Patronik and Marco Zenati, MD, Carnegie Mellon University. HeartLander is funded by NIH and NASA, with past funding from NSF and The Pittsburgh Foundation.

RESOURCES
- Tower Hobbies (www2.towerhobbies.com) — Servos, pushrods with clevis.
- All Electronics (www.allelectronics.com) — 0.25”d x 0.2”l Neodymium magnets.
October 25 - 26, 2007
San Jose McEnery Convention Ctr
San Jose, CA

Register by Sept. 14 and SAVE $200.

The RoboDevelopment Conference & Exposition is the industry’s first event focused on the technical design and development of commercial robotic products.

This unique and unprecedented event will draw over 1,000 robotics industry personnel from across the globe and will feature:

- Comprehensive Conference Program covering the most critical industry topics including Design, Development and Standards, Tools and Platforms, and Enabling Technology
- Over 40 leading industry speakers
- Keynote presentations by Tandy Trower, General Manager, Microsoft Robotics Group; Paolo Pirjanian, President and CEO, Evolution Robotics; Lloyd Spencer, CEO, Coroware; and Dan Kara, President, Robotics Trends
- A packed Expo Floor offering hands on access to the latest design and development solutions for producing robots and robotic technology
- Great networking opportunities, special events and much more

REGISTER TODAY AT
www.robodevelopment.com
or call 800-305-0634
Seetron Serial LCDs

Interface a sharp LCD display to your BASIC Stamp® or other micro-controller project with ease. No-solder wiring harnesses and easy mounting kits available too. See www.seetron.com today.

- 3.2 x 1.4 in. supertwist LCD
- 2400/9600 baud serial
- Low (=2mA) current draw
- Great with BASIC Stamps®

- 3.2 x 2 in. backlit LCD
- 1200-9600 baud serial
- Advanced protocol, 4 switch inputs
- EEPROM for configuration settings
- Favorite for OEM applications

- 3.2 x 1.4 in. graphics LCD
- 2400/9600 baud serial
- Font and 15 screens in EEPROM
- Easily draw points, lines, screens

- 3 x 2 in. supertwist LCD
- 1200-9600 baud serial
- ESD-protected, 4x4 keypad input
- Store up to 95 screens in EEPROM

Scott Edwards Electronics, Inc.
1939 S. Frontage Rd. #F, Sierra Vista, AZ 85635
phone 520-459-4802 • fax 520-459-0623
www.seetron.com • sales@seetron.com
Dan’s boss, Mark, allowed me to visit the shop last year and I was like a kid in a candy store — I love special FX and prop technology and was in absolute awe of everything in their shop; I think I had to run to the parking meter three or four times because I just didn’t want to leave! Most of the props I got to see (and play with!) were articulated: a few with cables that were attached to a control device called a “waldo” (Author’s note: WALDO is a trade-mark of The Creature Shop, and yet the term is used somewhat generically in the business due to its roots in a Robert Heinlein short story).

The waldo allows extraordinarily-skilled puppeteers to bring life to a prop while a scene is being shot. One of the cable-controlled props that I was allowed to play with was the cat from *Date Movie* which, I must say, was much more fun than having a kitten chase a flashlight beam. Other props use servos to control the motion, many being driven by standard R/C transmitters — the problem with using an R/C transmitter is that the movement values can’t be recorded.

Even if you don’t want or need a device like this, I think that you’ll find a lot of the code useful for other projects. In short, what this device will do is take the input from a standard analog joystick and send position values to a couple servos. By pressing a button with a record jumper enabled, the device will store the joystick movements into a 24LC512 (64K) EEPROM; press the button again and the record cycle stops; remove the jumper and the cycle that was just recorded can be played back.

I was really pressed for time this week, so I decided to use the SX28 proto board again. It turned out to be a really good choice because there were features of the proto board that are perfect for this application. Specifically, there are two major sub-systems for the waldo: 1) analog-to-digital conversion of joystick data and, 2) storage of servo values in an EEPROM.

It turns out that I’m using eight pin DIP devices: the ADC0832 for analog conversion and the 24LC512 for the memory — and the SX28 proto board includes two DIP8 socket locations in the prototyping area. In Figure 1, the red boxes outline the DIP8 locations; note, too, that inside the DIP8 pads are pads for SOIC8 devices.

Another — far more subtle — feature of the SX28 proto board is voltage level selection and control for the servo headers (blue box in Figure 1). Note the purple box; if we solder a male, three-pin header to this area, we can select the voltage applied to the center terminal of the servo headers, either 5V (from the regulator) or the

---

**WHERE’S WALDO?**

It seems like the animation controller from May was a hit. I got a lot of very positive email and many readers have been creating derivative applications for controlling servos. A few weeks ago, I got a call from one reader — my friend Dan — who works in one of Hollywood’s leading prop shops (if you saw the movie *300*, then you’ve seen some of their amazing creature work). Dan suggested that I build a recording servo controller because — in Hollywood prop shops, anyway — the live performance is what really counts and the ability to play back a great performance is a convenience. This kind of device would also be cool for an animated Halloween or Christmas prop, something I have a lot of interest in.
Vin from the power connector (be careful with using Vin; anything above 7.2 volts will probably destroy your servos). The center pin of the voltage select header is routed through the power switch to the servo headers so that we can power up the SX without applying power to the servos. When we want the servos to run, we move the power switch to the far-right position. These features shouldn’t be too surprising as they make using the SX28 proto board on the Parallax Boe-Bot chassis very easy.

WALDO SUBSYSTEMS

The first major subsystem is the analog-to-digital conversion of the joystick input. Part of the parameters for this project was to use an off-the-shelf, PC-compatible analog joystick. The trouble with analog PC joysticks is that we only get two connections per pot; we don’t get the ends plus the wiper – we get one end (common, on pin 1) and the wipers from each axis. What we’re forced to do, then, is add a second resistor to create a divider. We’ll tap off that divider and run it into an input channel of the ADC0832. Figure 2 shows the joystick interface for the circuit. As you can see, the pot outputs from the joystick (pins 3 and 6) are connected to 10K pull-downs and then on to the ADC0832. The application of 5V to pin 1 of the joystick connector routes this voltage to the pots. While this arrangement works, we don’t get a centered output value when the pot is in its center position; I don’t like “fixing” circuits in code, but this is a case where there is just no choice.

Let’s have a look at the ADC code. The ADC0832 has two inputs and it can be configured on-the-fly to provide singled-ended or differential output. Configuration of the ADC is accomplished by clocking four (mux) bits into the device. As you can see in the function code, \( \backslash 4 \) is used to limit the number of bits sent to the ADC. For convenience, the waldo program has constants defined for single-ended channels 0 and 1 that include the ADC start bit. After shifting the mux bits out to the ADC, an eight-bit value can be shifted in.

```stamps
FUNC ADC0832
tmpB1 = __PARAM1
Clk = 0
CS = 0
SHIFTOUT Dio, Clk, MSBFIRST, tmpB1\4, 4
SHIFTIN Dio, Clk, MSBORST, tmpB1, 4
CS = 1
RETURN tmpB1
ENDFUNC
```

Let me point out a couple things having to do with the use of `SHIFTOUT` and `SHIFTIN`. First, both of these instructions simply invert the clock pin (twice) for each bit. What this means is that we need to preset the clock level before using `SHIFTOUT` or `SHIFTIN`; neglecting to do this is a common mistake for those porting BASIC Stamp programs to SX/B. Later versions of SX/B include an optional clock speed multiplier.

Without the multiplier, the data rate is about 83K bits per second, but many devices will operate significantly faster than that so we can take advantage by using the multiplier. Another point of consideration for this particular project is that a “virtual” servo controller is running in the Interrupt Service Routine so all foreground operations are...
slowed — the multiplier restores the access speed to the ADC. The other major subsystem is the I²C memory. I selected the 24LC512 because it holds 64K bytes of data and is really simple to use. Figure 3 shows the connections to the 24LC512.

When you look through the complete listing, you’ll see a lot of code having to do with the EEPROM. The reason is that there are four instructions in SX/B (I²CSTART, I²CSTOP, I²CSEND, and I²CRECV) that get encapsulated in subroutines or functions to save code space.

I’ve also created a subroutine that lets us write a byte or word to the EEPROM, and two functions for reading: one for bytes, the other for words. All of this code is highly portable and you can use it in many applications. The only critical note is that the SCL pin is aligned with the SDA pin, i.e., the SCL pin always follows the SDA pin on the same port (RA, RB, RC, RD, or RE). For example, if SDA is RA.2, then we must connect SCL to RA.3.

Since writing to the EEPROM is a critical task for this program, let’s have a look at the PUT_EE function. If you compare the code to the 24LC512 data sheet, you should see that it’s an easy match — my point is that once you’ve got the I²C routines set up, access to any I²C device is very straightforward.

```
SUB PUT_EE
  IF __PARAMCNT = 3 THEN
    tmpW1 = __WPARAM12
    tmpB1 = __PARAM3
    i2cMulti = 0
  ELSE
    tmpW1 = __WPARAM12
    tmpB1 = __PARAM3
    tmpB2 = __PARAM4
    i2cMulti = 1
  ENDIF
  I²C_START
  I²C_OUT SlaveWr
  I²C_OUT tmpW1_MSB
  I²C_OUT tmpW1_LSB
  I²C_OUT tmpB1
  IF i2cMulti = 1 THEN
    I²C_OUT tmpB2
  ENDIF
  I²C_STOP
  '{IFDEF WRITEWAIT}
  DO
    I²C_START
    I²C_OUT SlaveWr
    LOOP UNTIL ackNak = Ack
  '{ENDIF}
ENDSUB
```

This subroutine expects a word address and then one or two bytes after that. Since we want to use the same subroutine to write bytes and words, we’re forced into accepting the address as two bytes. If we’re using a word variable for the EE address, the compiler will sort that out for us. When using a constant value, however, we have to be careful. Let’s say, for example, that we want to write a value to EEPROM address $000F. Here’s how we have to do that when using constants:

```
PUT_EE $0F, $00, value
```

As you can see, we’re using two bytes for the address and the bytes are aligned Little-Endian. If we did this:

```
PUT_EE $000F, value
```

the compiler wouldn’t understand that the address is a word since the address value is less than 256 — so a byte is assumed and used. If the value to be written is also a byte, the compiler will complain that we don’t have enough parameters for the subroutine. By forcing a two-byte address, the subroutine can determine whether we want to write a byte (__PARAMCNT is three) or a word (__PARAMCNT is four). When we are writing two bytes, a flag is set that gets used later in the routine.

The subroutine might look a little complicated but it is, in fact, very straightforward. We start with the I²C start sequence, write the device address (a constant in this program, but could be a variable if you want to expand to multiple EEPROMs), the address to write to, and then the byte(s) to write. As you can see, we use the flag to control writing the second byte. Finally, the I²C stop sequence is generated to tell the EEPROM to save its buffer contents.

EEPROMs are not particularly fast and this device can take up to five milliseconds to store the values we just sent to it. We don’t care about this delay because we’ll only access the EEPROM every 20 milliseconds, but in other applications, we can have the subroutine wait for the write cycle to complete before returning to the caller. We do this by generating another start, and then writing the slave address. While the EEPROM is busy with its write cycle, the acknowledge bit will be set to NAK (1). A conditional-compilation constant allows us to enable or disable the write-wait option.

**PUTTING IT TOGETHER**

Like the animation controller project in May, this program uses a virtual servo controller that runs in the ISR; since we’ve been through that in detail, we won’t hash through it again. The only thing that’s been added to the ISR is an LED control option. The reason is that the program has four modes:

1) Idle (servos follow joystick)

2) Recording (servos follow joystick and positions are saved to EEPROM)
3) Playback (servo positions are played back from the EEPROM)

4) Paused (servos hold position until pause button is pressed again)

I started with a three-leaded, bi-color LED — one of those LEDs that has red and green elements with a common leg. The problem was that the body was clear and when both LEDs were on, it didn’t really look yellow as I had hoped. Well, in my supplies I found a two-leaded, bi-color LED that had a milky, translucent body which I thought might work better. It did, but it takes a little more code to create the yellow effect.

This is accomplished by reversing current flow through the LED very quickly. Since the interrupt runs 100,000 times per second, we can do it there. Here’s how:

```
Check_Yellow_LED:
  IF runMode = M_PAUSE THEN
    ledPort = ledPort ^ %1100
  ENDIF
```

What this does is invert the state of the LEDs each pass through the ISR when the program is paused. Figure 4 shows how the LED is connected. There is only one resistor because current can only flow through one LED at a time.

Okay, we can read the joystick and save values, so let’s get to the meat of the program. At the top, we’re going to wait for a new servo frame (the ISR sets a flag bit that we’ll wait on). When we get the new frame, we read the joystick axis values:

```
Main:
  WAIT_SYNC
  joyX = ADC0832 Ch0
  joyY = ADC0832 Ch1
```

Since the output from the ADC doesn’t match what we need to drive the servos, we’ll have to apply a little math to adjust things.

Let me take a bit of a detour here and encourage you to develop your programs in sections and with separate test programs. I did this with all of the systems of the waldo program and have included my test programs for you to work with should you choose. (Files are available at www.nutsvolts.com.) When testing the joystick, I got values between 25 and 255, and what we need for the servo is 100 to 200. The challenge doesn’t end there, however, as the center value for the joysticks was 50.

So, we have two sections: one side goes 25 to 50, the other side of the stick goes 50 to 255. To adjust the first section for servo pulse values, the math is pretty simple:

```
position = position * 2 + 50
```

By applying this formula, we’ll move 25-to-50 to 100-to-150. For the other side, we get lucky again and the formula works out to:

```
position = position / 4 + 137
```

I say that we got lucky because the multiplier and divider in the formulas above are powers of two; this lets us use shift operators instead of multiplication and division operators (both generate a fair bit of code).

Here’s the adjustment section:

```
Check_Buttons:
  IF BtnX = Pressed THEN
    INC btnTmr(0)
  ELSE
    btnTmr(0) = 0
  ENDIF

  IF BtnY = Pressed THEN
    INC btnTmr(1)
  ELSE
    btnTmr(1) = 0
  ENDIF
```

Each button has its own debounce counter. When either counter reaches five (which means the button is pressed for five times), it’s okay, the result will still be correct.

Once the program was running, I found that the servos moved opposite to the actual joystick movement so both axis values had to be inverted (100 becomes 200, and vice versa). This is easy math, too — just subtract the axis value from 300. Note that the compiler will complain about a truncated literal. This happens because we’re using a value greater than 255 in the equation and the output is a byte; it’s okay, the result will still be correct.

The joystick has two buttons; one for each axis. We can scan and debounce the buttons like this:

```
Check_Buttons:
  IF BtnX = Pressed THEN
    INC btnTmr(0)
    btnTmr (0) = 0
  ELSE
    btnTmr(0) = 0
  ENDIF

  IF BtnY = Pressed THEN
    INC btnTmr(1)
    btnTmr (1) = 0
  ELSE
    btnTmr(1) = 0
  ENDIF
```

The STAMP APPLICATIONS
ton was held down for five consecutive cycles), it is considered valid.

How a button press affects the program is based on the current operational mode. The next step in the program is to jump to the handler for the current mode:

```
Mode_Handler:
  IF runMode = M_IDLE THEN Check_Start
  IF runMode = M_REC THEN Recording
  IF runMode = M_PLAY THEN Playing
  IF runMode = M_PAUSE THEN Play_Paused
```

The first mode, M_IDLE, is where we'll sit and wait for a start button press, all the while the servos will follow any motion of the joystick. When the start (x axis) button is pressed, we'll look at the Play/Record jumper and take things from there.

```
IF btnTmr(0) = BtnOK THEN
  IF PlayRec = RecordNow THEN
    numRecs = 0
    runMode = M_REC
  ELSE
    numRecs = GET_EE2 0, 0
    IF numRecs > REC_LAST THEN
      GOTO Empty_EE
    ELSE
      runMode = M_PLAY
      GOTO PlayPaused
    ENDIF
  ENDIF
  eePntr = 2
  btnTmr(0) = 0
ENDIF
GOTO Main
```

When the Play/Record jumper is out, the number of records is read from the EEPROM at address $0000. Blank EEPROMs usually have all locations set to $FF, so if we see a bad value in the numRecs variable, the program flashes the red LED and then jumps back to the top in idle mode — what this means is that we need to record some movements first.

Install the Play/Record jumper and press the start/stop button on the joystick — press it quickly, though. The [movement] records counter will be cleared and the mode set to M_REC which directs the program to this section that follows:

```
Recording:
  ledPort = LED_RED
  pos0 = joyX
  pos1 = joyY
  IF btnTmr(0) = BtnOK THEN
    GOTO Stop_Recording
  ELSE
    PUT_EE eePntr, joySticks
    INC numRecs
    IF eePntr < REC_LAST THEN
      eePntr = eePntr + 2
    ELSE
      GOTO Stop_Recording
    ENDIF
  ENDIF
  GOTO Main

Stop_Recording:
  PUT_EE 0, 0, numRecs
  runMode = M_IDLE
  btnTmr(0) = 0
  GOTO Main
```

We start by refreshing the red LED to indicate record mode and then update the background servo controller with the current joystick values. If the start/stop button has not been pressed, then the axis values are written to the memory, the records count is updated, and then we check to see if there's any room left in the EEPROM. If the start/stop button has been pressed or we have run out of memory, the recording process is stopped. Here we write the number of records to address $0000 of the EEPROM and reset the mode to idle.

Remove the Play/Record jumper and press the start/stop button again — you should see the moves you just recorded played back. I'm easily entertained, but when this worked the first time my face lit up with a very big smile. Here's the playback code:

```
Playing:
  ledPort = LED_GRN
  pos0 = GET_EE2 eePntr
  pos1 = __PARAM2
  DEC numRecs
  IF btnTmr(0) = BtnOK THEN
    runMode = M_IDLE
    btnTmr(0) = 0
    GOTO Main
  ENDIF
  IF btnTmr(1) = BtnOK THEN
    runMode = M_PAUSE
    btnTmr(1) = 0
    GOTO Main
  ENDIF
  IF numRecs > 0 THEN
    eePntr = eePntr + 2
  ELSE
    IF Repeat = Yes THEN
      numRecs = GET_EE2 0, 0
      eePntr = 2
    ELSE
      runMode = M_IDLE
      GOTO Main
    ENDIF
  ENDIF
```

```
Parts List

- SX28 proto board
- ADC0832, dual-channel ADC
- 24LC512 EEPROM
- Eight-pin DIP socket
- DB-15F socket (solder cup type)
- 220 ohm resistor
- 1K resistor
- 4.7K resistor
- 10K resistor
- 0.1 µF capacitor
- bi-color LED (two lead)
- 0.025 male post headers (0.1" centers)
- post-header jumpers
```

84 NUTSVOLT September 2007
In playback mode, the green LED is lit and we pull a new set of servo values from the EEPROM. Note that the GET_EE2 function returns a word, but we’re placing the result in a byte (pos0). The compiler is smart enough to put the LSB of the returned value into pos0. We can get the MSB of the return manually by copying __PARAM2. When a function returns a word, the LSB is in __PARAM1; the MSB is in __PARAM2.

After retrieving a movement record, we decrement the records count and check for button presses. We can stop the playback cycle or, by pressing the y axis button, put it into a pause mode. Assuming no button presses, we check to see if the playback cycle is complete. If not, then the EE pointer is advanced to the next set of position values. When we do get to the end, the program checks the loop-back jumper. When this is installed, the records count is reloaded, the EE pointer sent back to the beginning, and the playback continues. With no loop-back jumper, the mode is returned to idle and playback stops.

The pause mode doesn’t do anything except monitor button presses. We can either resume the cycle by pressing the y axis button again, or stop it by pressing the x axis button.

```
Play_Paused:
IF btnTmr(1) = BtnOK THEN
   runMode = M_PLAY
   btnTmr(1) = 0
   GOTO Main
ENDIF
IF btnTmr(0) = BtnOK THEN
   runMode = M_IDLE
   btnTmr(0) = 0
ENDIF
GOTO Main
```

And there we have it — a two-channel servo controller that plays live, records, and plays back. Figure 6 shows my final board. The six-pin header on the right edge is where I connected my DB-15 joystick adapter (using 0.025" post-header sockets). Note that the SX28 proto board has busses for Vdd and Vss so connecting the pull-ups and pull-downs for the joystick connection is very easy at this point.

Connections beneath the board are made with wire-wrapping wire, except for the power connections to the joysticks. Here’s a bit of a tip for your toolbox: Get a roll of blue painter’s tape. It’s great for holding wires and components while soldering, and won’t leave a sticky mess when you pull it off — I always have it on hand when I’m building circuit boards.

The board does work as well as I’d hoped, but one thing I will do is order an LTC1298 and replace the ADC0832 with it. I didn’t happen to have an LTC1298 in my supply, so I went with what I had. It’s a little pricey (about $11), but I think the additional resolution will be worth having, and the pin-out exactly matches the ADC0832. Next up for me is to attach a neat little pan-tilt servo head that I bought from Lynxmotion and make it dance!

Have fun, and until next time, Happy Stamping! NV
No Compromise Oscilloscope

Other oscilloscopes in this price range force you to compromise on one of the key specifications: sampling rate, memory depth or bandwidth.

The PicoScope 5000 series is a no compromise PC oscilloscope at a price every engineer can afford.

1G5/s sampling rate
250MHz bandwidth
128M sample buffer memory
125MS/s 12 bit AWG built in

PicoScope 5203
32M buffer $2270
PicoScope 5204
128M buffer $3410

www.picotech.com/pco394
1-800-591-2796
To accomplish the tasks of replying to a PING and sending UDP datagrams to the EDTP Internet Test Panel, we will need to code up an ARP reply routine and a simple UDP echo module. The new ARP routine will replace the manual stuffing of the Ethernet MINI's IP and MAC addresses into the PC's ARP cache I outlined in the last Design Cycle column. The assembly of the UDP echo source code module will do more than just allow us to send UDP datagrams to the EDTP Internet Test Panel Visual Basic application. UDP is the basis of DHCP message communication and having the basic UDP coding techniques under our belts will make our coding lives easier when we are ready to code up the DHCP modules. ARP is currently the most important piece of coding we have to do. So, let's take a swing at getting our Ethernet MINI hardware to respond to an ARP request.

**CODING THE ARP FIRMWARE**

ARP is network shorthand for Address Resolution Protocol. In our situation, the Ethernet MINI is normally the recipient of an ARP request. An ARP request is generated by any host on a network that is searching for another particular host on the network. In our case, the requesting or searching host is a PC attached to the Ethernet MINI's network. The problem for the searching host is that it only has the desired destination host's IP address. The host that is on the search mission also needs to have the desired destination host's MAC (hardware) address as well, in order to communicate directly to it.

The reasoning behind this is that many differing hosts on different (but connected) networks may all have the same IP address. However, the MAC address of every host on any network must be unique to that particular host. The uniqueness of the MAC addresses is guaranteed if the network host's hardware address was issued by the IEEE. By issuing an ARP request embedded with the desired destination host's IP address, the searching host can obtain the desired destination host's unique MAC address it needs to make the network connection with the desired remote network host that has the IP address specified in the ARP request. Now that you know why we need to code an ARP module, let's do just that.

The ARP components take up the same packet space as the IP header (and then some). Thus, an ARP packet is its own thing, as an ARP message is not encapsulated within an IP packet. Despite this seeming lack of respect for IP, ARP must still use elements of IP to do its job. Consider the following code snippet:

```
;********************************************************
;*    ARP Layout
;********************************************************
arp_hwtype CON $0E
arp_prtype CON $10
arp_hwlen CON $12
arp_prlen CON $13
arp_op CON $14   ;arp request or response
arp_shaddr CON $16   ;arp source mac address
arp_sipaddr CON $1C   ;arp source ip address
arp_thaddr CON $20   ;arp target mac address
arp_tipaddr CON $26   ;arp target ip address
```

Take a look at the ported PICBASIC PRO source that I've provided. Notice that the ARP_hwtype and ipVersLen fields are located at the same location within an Ethernet packet. Note also that the ARP framework is four bytes larger than the area occupied by the IP header. We will use these differences and similarities in our Ethernet MINI firmware to fish out an incoming ARP message and act upon it. The stream we will fish in for ARPs runs through the get_frame subroutine. Here's how we bait the hook.
;process an incoming ARP request packet
if packet[enetpacketType0] == $08 &&
packet[enetpacketType1] == $06 then
  if packet[arp_hwtype+1] == $01 &&
  packet[arp_prtype] == $08 &&
  packet[arp_prtype+1] == $00 &&
  packet[arp_hwlen] == $06 &&
  packet[arp_prlen] == $04 &&
  packet[arp_tipaddr]   == ipaddrc[0] &&
    select case packet[arp_op+1]
    case $01
      gosub arp_reply
    case $02
      for i8 = 0 to 5
        remotemacaddrc[i8] = packet[arp_shaddr+i8]
      next i8
      gosub set_arpflag
    end select
    endif
  endif
endif

The ARP Ethernet packet type of $0806 tugs on our fishing line as the bait is being taken. An incoming IP packet would have $0800 in the Ethernet packet type field. The hwtype or hardware address type field's $01 denotes a 10 MB Ethernet. Recall that ARP does indeed use IP components. So, the protocol type field is filled with $0800, which denotes the IP protocol.

A MAC address is made up of six bytes. This is conveyed in the arp_hwlens field with a value of $06. The arp_prlns array entry tells us that the IP address consists of four octets. Octet is Internet document speak for byte. It is possible to only filter on the Ethernet packet type and the IP address to determine that we have hooked an incoming ARP message. However, to be safe, I decided to verify every field I thought that I should to make sure the incoming message was indeed an ARP and that the captured ARP message was directed to the hardware running this ported PICBASIC PRO firmware.

The arp_op array entry is the operation that is being requested. A $01 in the operation field represents an incoming ARP request that we must reply to. If our Ethernet MINI generated an ARP request, we would look for the operation field to contain $02 and the returned ARP message to contain the MAC address of the replying host.

You can immediately see what happens when an incoming ARP message contains the in-question MAC address. We simply stuff the contents of the arp_shaddr (source hardware address) array fields into the remotemacaddrc array fields for later use. If you're wondering about the “c” in remotemacaddrc, that’s a holdover from the port from C. I use “c” to denote byte variables and “i” to identify integer variables when there are a bunch of them all mixed up under my care.

We really need to respond to an incoming ARP request before we can do anything useful with the Ethernet MINI. So, branch to the arp_reply subroutine in the Ethernet MINI driver PICBASIC PRO source code. The arp_reply subroutine is simply building an ARP reply packet in the PIC18F67J60’s internal transmit buffer memory. Recall that we have already hacked out transmit and receive buffer areas within the PIC18F67J60’s 8K of packet buffer SRAM.

Once the transmit buffer write pointers (EWRPTH/EWRPTL) are initialized and the mandatory control byte is placed into the PIC18F67J60’s transmit buffer, we use the source MAC address we gleaned from the incoming ARP message to use as a return hardware address. The addition of our MAC address as the sender (source) in the initial address fields (DLC Header) of the packet will not fulfill our reply obligation. The MAC address the requesting host is looking for is contained within the body of the ARP message. The best way to illustrate the inner workings of the PICBASIC PRO ARP code is with an illustration.

Figure 1 is a screen capture taken from a Network General Sniffer Portable Ethernet sniff session. The sniff session participants are a laptop and an Ethernet MINI. Note the PC’s MAC address (Source = Station 001B3807CFC7) is contained within the DLC Header area and the body of the ARP packet. The reason for this is that networks aren’t always simple, router-less two-node dances. The Internet is built around routers and routers need addresses to be able to forward packets around the Internet. Thus, the DLC Header addresses are used by the routers (and host nodes for that matter) for addressing information as a router isn’t programmed to dig deep into packets to retrieve address information.

We’ve already talked about most everything contained
in the ARP/RARP frame fields of Figure 1 with the exception of the MAC and IP fields. Figure 1 is a sniff representation of an ARP request that was sent by the laptop on the two-node Ethernet MINI network. Since the PC knows the Ethernet MINI’s IP address but does not know the Ethernet MINI’s MAC address, the ARP request is transmitted as a broadcast message with the target MAC address left “blank.” The Ethernet MINI’s IP address is 192.168.0.150 and the Target protocol address field in Figure 1 was populated by the PC before sending the ARP request. The laptop is looking for the Ethernet MINI to respond to this ARP request and fill in the “blank” MAC address, which is, of course, the Ethernet MINI’s MAC address.

The Ethernet MINI fulfilled its ARP obligation in Figure 2. Although the Ethernet MINI has inserted its MAC address into the DLC Header reply, the requesting application will actually retrieve the Ethernet MINI’s MAC address from the sender’s hardware address fields within the ARP frame. Thus, our ported Ethernet MINI ARP PICBASIC PRO source code built an ARP reply frame and turned around the source and destination addresses to realize and send along the ARP reply packet shown in Figure 2. When you study the PICBASIC PRO arp_reply subroutine, you will find that there is nothing in the coding that is complex at all. In fact, I didn’t have to do much of anything to port many of the original C source lines to the PICBASIC PRO language.

At this point, the laptop has retrieved the Ethernet MINI’s MAC address information and placed it into its ARP cache. You can use the arp /a command on the laptop to see the Ethernet MINI’s IP and MAC cache entries. In that, the Network General Sniffer caught the ARP request and reply sequence, and I could see the Ethernet MINI’s address information in my laptop’s ARP cache. We can therefore be fairly sure that the Ethernet MINI’s ARP PICBASIC PRO code worked as designed.

PINGING WITH PICBASIC PRO

The first major step in our PICBASIC PRO code port has been taken. We are now able to identify our PICBASIC PRO-laden Ethernet MINI module to other hosts on an Ethernet network via an ARP reply.

A PING is really an ICMP operation. So, we must port the original C ICMP code to PICBASIC PRO if we want to PING our Ethernet MINI that is running the PICBASIC PRO driver code. Our PING coding takes place in the icmp subroutine. However, the icmp subroutine does not get called unless the correct conditions are met in the code snippet that follows:

```plaintext
; process an IP packet
if packet[enetpacketType0] == $08 &&
packet[enetpacketType1] == $00 &&
packet[ip_destaddr]   == ipaddrc[0] &&
select case packet[ip_proto]
case PROT_ICMP
  gosub icmp
  endif
  case PROT_UDP
  if packet[UDP_srcport] == DHCP_SERVER_PORT then
    gosub dhcp_state_engine
  else
    gosub udp
  endif
  case PROT_TCP
  gosub tcp
  endif
```

Unlike the ARP frame, this is a true IP datagram as denoted by the $0800 Ethernet type value. Since this is an IP packet, there are a number of things that can be wrapped up inside of it. According to the code I just offered that processes an IP packet, we could have an ICMP, a UDP, or a TCP operation encapsulated within the incoming IP datagram. Another look at the IP packet parsing code tells us that all we need to do is make sure it is indeed an incoming IP datagram and that the incoming IP datagram is actually addressed to our Ethernet MINI. Then — depending on the encapsulated protocol type — we branch off into analyzing and acting upon the payload of the incoming IP datagram we have captured with the Ethernet MINI. Since we’re on the subject of PINGing, we’ll turn our attention to the PROT_ICMP path, which invokes the ported PICBASIC PRO icmp subroutine.

A PING response is basically nothing more than an echo of the received data payload. However, we must readdress the PING reply packet and calculate some checksums before sending our PING reply.
converted PICBASIC PRO subroutine called setipaddrs is responsible for the address reconfiguration.

```c
;********************************************************************************
;*   This subroutine builds the IP header.
;********************************************************************************

;******************************************************************************
;move IP source address to destination address
packet[ip_destaddr]=packet[ip_srcaddr]
packet[ip_destaddr+1]=packet[ip_srcaddr+1]
packet[ip_destaddr+2]=packet[ip_srcaddr+2]
packet[ip_destaddr+3]=packet[ip_srcaddr+3]

;make ethernet module IP address source address
packet[ip_srcaddr]=ipaddrc[0]
packet[ip_srcaddr+1]=ipaddrc[1]

;move hardware source address to destination address
packet[enetpacketDest5]=packet[enetpacketSrc0]
packet[enetpacketDest6]=packet[enetpacketSrc1]
packet[enetpacketDest7]=packet[enetpacketSrc2]
packet[enetpacketDest8]=packet[enetpacketSrc3]
packet[enetpacketDest9]=packet[enetpacketSrc4]
packet[enetpacketDest10]=packet[enetpacketSrc5]

;make ethernet module mac address the source address
packet[enetpacketSrc0]=macaddrc[0]
packet[enetpacketSrc1]=macaddrc[1]
```

Porting the initial portion of the setipaddrs subroutine was a piece of cake as all I had to do was remove the semicolons from the ends of the ported statements. In actuality, I could have left the semicolons in place as they are comment delimiters in PICBASIC PRO. The downside to this laziness is that sometimes an active statement following the semicolon that needs to be ported gets the commented color scheme, and can be accidentally erased or ignored. During porting, I had the “ignore” scenario occur just before I was about to execute the “erase” scenario. So, I’ve eliminated the end-of-C-statement semicolons in our C-to-PICBASIC PRO port. As you can see in the setipaddrs code snippet, the setipaddrs subroutine simply swaps the source and destination IP and MAC addresses in preparation for the transmission of the PING reply packet.

Just when you think things are in hand, the bottom of your basket falls through. In the C source we’re porting, the 16-bit IP header and ICMP header checksums are calculated using 32-bit variables. Unfortunately, PICBASIC PRO knows what a 32-bit variable is, but it has no orders to do anything about them. The PICBASIC PRO DIV32 mnemonic and some tricky Darrel Taylor PICBASIC PRO/assembler code are all that we have to work with when it comes to using ported unsigned long C variables with PICBASIC PRO. Unfortunately, I was unable to think (or should that be trick) my way through the 16-bit checksum problem using Darrel’s readily-available algorithms. That was a loss as Darrel’s DIV32 code is compact and efficient. So, I had to resort to Plan B from outer space. Here’s the first of the alien invasion code:

```
chksum16 var word
acc3 var word
acc2 var word
acc1 var word
acc0 var word
arg3 var byte
arg2 var byte
arg1 var byte
arg0 var byte
```

Let’s translate. Note that there are four acc word variables and four arg byte variables. The acc variables are accumulator variables while the arg variables I’ve declared are arguments or mathematical operands. When all is said and done, the chksum16 variable will contain our magic checksum, which was calculated with the help of the accumulators and arguments.

The IP checksum is defined as the 16-bit one’s complement sum of all 16-bit words in the header. PICBASIC PRO can natively manipulate 16-bit variables. Our problem lies in the fact that when accumulating 16-bit values, an overflow into the 17th bit can possibly occur. PICBASIC PRO will throw the 17th bit in the bit bucket. Our checksum calculation uses all of the bits from bit 17 up to bit 31. Our job is to arrange all of the IP header bytes into words, add them all together with the carries out into bit 17 and beyond, and invert the sum by performing a one’s complement against it. Simple, huh? Yep ... and here’s how we’ll do it.

In your mind, group the arg variables into a 32-bit variable with arg3 representing the most significant byte of the 32-bit long variable and arg0 acting as the least significant byte of the 32 bit variable. For instance, if arg3=$03, arg2=$02, arg1=$01, and arg0=$00, that would equate logically to a 32-bit value of $3210. Get the idea? The same logic applies to the accumulators with a twist.

Each accumulator is a word variable and is 16 bits in length. PICBASIC PRO doesn’t natively allow the programmer to interrogate carry situations. So, if we declared each accumulator variable as a byte, the addition of $FF and $01 in an accumulator variable would render $100, which is out of the bounds of an eight-bit variable. The carry out of the addition of $FF + $01 would be lost to us as the accumulator variable would simply roll over to $00.

The accumulators are all declared as words to trap any byte addition overflows. Consider this. We load acc0 with $FF. We then load arg0 with $01. The values are then added together and stored in acc0 like this: acc0 = acc0 + arg0. Since acc0 is a word variable, the value stored in acc0 following the addition will be $0100. Here’s where our logic becomes magic.

We have arranged the accumulators as four logical words with acc0 being the least significant word and acc3 as the most significant word. That’s logically 64 bits. The magic comes in as we treat each accumulator as a byte not as a word. The most significant byte of each accumulator...
variable is simply a bit bucket for the carry bits. The value in the most significant byte of acc0 is added to the least significant byte of acc1 and the most significant byte of acc0 is cleared to $00. Do you get it?

The most significant byte of each of the accumulator variables is added to the least significant byte of the next accumulator in the 64-bit accumulator chain we logically created. After each carry-over addition, the most significant byte of the accumulator that overflows is cleared to zero. Thus, we logically have only 32 bits in our accumulator chain (the accumulator’s least significant bytes), with each accumulator having its own carry bit bucket (the most significant byte of each accumulator) from which we can dip. The bit buckets are serviced by the ripple_crc subroutine, which I present to you here:

```plaintext
ripple_crc:
if acc0 > $FF then
    acc1 = acc1 + 1
    acc0 = acc0 & $00FF
endif
if acc1 > $FF then
    acc2 = acc2 + 1
    acc1 = acc1 & $00FF
endif
if acc2 > $FF then
    acc3 = acc3 + 1
    acc2 = acc2 & $00FF
endif
if acc3 > $FF then
    hserout[13,10, "CHECKSUM OVERFLOW ERROR",13,10]
endif
return
```

Since the IP checksum specifies that we add 16-bit numbers, we logically group the arg byte variables into pairs, which we logically add to the accumulators as 16-bit values. You can see this happening in the remainder of the setipaddrs subroutine PICBASIC PRO code. Here’s the ported PICBASIC PRO code that calculates the IP header checksum:

```plaintext
;calculate the IP header checksum
packet[ip_hdr_cksum]=00
packet[ip_hdr_cksum+1]=00
hdrlen = (packet[ip_vers_len] & $0F) * 4
;addr = &packet[ip_vers_len]
i16 = 0
gosub clrcrc
while(hdrlen > 1)
    arg1=packet[ip_vers_len+i16]
    i16 = i16 + 1
    arg0=packet[ip_vers_len+i16]
    i16 = i16 + 1
    acc0 = acc0 + arg0
    gosub ripple_crc
    acc1 = acc1 + arg1
    gosub ripple_crc
    hdrlen = hdrlen - 2
wend
if(hdrlen > 0) then
    arg1=packet[ip_vers_len+i16]
    arg0=$00
    cc0 = acc0 + arg0
    gosub ripple_crc
    acc1 = acc1 + arg1
    gosub ripple_crc
endif
acc0 = acc0 + acc2
gosub ripple_crc
acc1 = acc1 + acc3
gosub ripple_crc
chksum16= ~(((acc1 << 8) + acc0);
packet[ip_hdr_cksum] = (chksum16 & $FF00) >> 8
packet[ip_hdr_cksum+1] = chksum16 & $00FF
return
```

The theory behind our accumulator-based checksum calculator seemed to work well. However, the real proof would come when a PING request jumped from my laptop to the Ethernet MINI.

I am vindicated in Figure 3. Our ported PICBASIC PRO checksum code in the setipaddrs subroutine calculated the IP header checksum value in the PING response packet you see sniffed in Figure 3. Our PICBASIC PRO checksum code was again put to the test in Figure 4. We correctly calculated the ICMP checksum, which is defined as the 16-bit one’s complement of the one’s complement sum of the ICMP message starting with the ICMP Type.

A PORTED UDP APPLICATION

The original C source we’re using as a base for our PICBASIC PRO port contains a simple UDP application that echoes characters sent from a Visual Basic program running on a PC. I’ve included the Ethernet MINI UDP application code in this month’s Design Cycle download package. I don’t think you’ll have any problems following the
PICBASIC PRO UDP code flow as it looks just like everything we’re mulled over up to now. Now that we have ported the code that drives the PIC18F67J60 Ethernet hardware, most of what we are doing consists of parsing fields of incoming packets, calculating checksums, and correctly placing IP and MAC addresses.

When you go over the UDP code, note that UDP adds yet another concept we must consider in our code. In addition to an IP and MAC address, UDP utilizes source and destination port addresses. The UDP application I’ve supplied in the download package (www.nutsvolts.com) looks for a message sent to well-known port 7, which is the echo port. “Well-known” means that this is a standardized port. You can use port 7 for other things but you may run into a problem if the other guy or gal you want to communicate with has used port 7 for its original intended purpose.

I’ve also included the time-tested EDTP Internet Test Panel application, which runs on a PC. The EDTP Internet Test Panel application you see in Figure 5 is easy to use. All you have to do is enter the Ethernet MINI’s IP address in the Target IP Address window and start typing. Ignore the Target Port and LCD Data Entry windows as they were part of a past EDTP application. If things work as designed, what you type into the Original Data window will be echoed by the Ethernet MINI back to the Echoed Data window in the EDTP Internet Test Panel application frame.

STACKING UP THE PROTOCOLS

We’ve worked our way through the coding of the PIC18F67J60 physical layers. With the completion of this month’s discussion, we’ve worked our way through the PICBASIC PRO 32-bit limitation and established a foothold on the coding of the basic Internet protocols. With the successful porting of the ARP, ICMP, and UDP protocols behind us, we are climbing towards the most famous protocol: TCP. Next time, we’ll talk more about UDP and port the Ethernet MINI DHCP and TCP C source to PICBASIC PRO. I’m having a blast coding with PICBASIC PRO.

SOURCES

EDTP Electronics, Inc. (www.edtp.com): Ethernet MINI

microEngineering Labs, Inc. (www.melabs.com): PICBASIC PRO

THE DESIGN CYCLE

FIGURE 4. Our 16-bit ported PICBASIC PRO checksum routines have passed the test as we have now calculated the IP header and ICMP checksums. Note that the Network General Sniffer points out that this is an Echo Reply and references the byte field in the hex dump at the bottom of the shot. All hail the Network General Ethernet Sniffer!

FIGURE 5. This application has been around for a couple of years and has seen widespread use. It’s simply a tool that was created to test the functionality of the UDP firmware written at EDTP Electronics. All you have to do is dial in a Target IP Address and start typing.

The Standard for checking Capacitors in-circuit
Good enough to be the choice of Panasonic, Pioneer, NBC, ABC, Ford, JVC, NASA and thousands of independent service technicians.

Inexpensive enough to pay for itself in just one day’s repairs. At $209, it’s affordable.

And with a 60 day trial period, satisfaction guaranteed or money-back policy, the only thing you can lose is all the time you’re currently spending on trying to repair all those dogs you’ve given up on.

CapAnalyzer 88A
Available at your distributor, or call 561-487-6103
Electronic Design Specialists
www.eds-inc.com

Locate shorted or leaky components or conditions to the exact spot in-circuit
Still cutting up the pcb, and unsoldering every part trying to guess at where the short is? $209

Your DVM shows the same shorted reading all along the pcb trace. LeakSeeker 82B has the resolution to find the defective component. Touch pads along the trace, and LeakSeeker beeps highest in pitch at the defect’s pad. Now you can locate a shorted part only a quarter of an inch away from a good part. Short can be from 0 to 150 ohms

LeakSeeker 82B
www.eds-inc.com

□ EDTP Electronics, Inc. (www.edtp.com): Ethernet MINI
□ microEngineering Labs, Inc. (www.melabs.com): PICBASIC PRO
Distributed exclusively by MCM Electronics

Tenma sets the standard for high quality, value priced products including:

- Oscilloscopes
- Pattern generators
- Function generators
- DMMs
- Spectrum analyzers
- Frequency counters
- Thermometers
- Replacement test leads
- Soldering and desoldering stations
- Solder
- Desoldering braid
- Irons and accessories

For specifications visit www.mcmelectronics.com

Source Code: NVM61
**NEW! HIDmaker FS for Full Speed FLASH PIC18F4550**

*DOWNLOAD the HIDmaker FS Test Drive today!*

- Creates complete PC and Peripheral programs that talk to each other over USB. Ready to compile and run!
  - Large data Reports
  - 64,000 bytes/sec per Interface
  - Easily creates devices with multiple Interfaces, even multiple Identities!
  - Automatically does MULTITASKING
  - Makes standard or special USB HID devices

*NEW! “Developers Guide for USB HID Peripherals” shows you how to make devices for special requirements.*

- Both PC and Peripheral programs understand your data items (even odd sized ones), and give you convenient variables to handle them.

**PIC18F Compilers:** PICBASIC Pro, MPASM, C18, Hi-Tech C.

**PIC16C Compilers:** PICBASIC Pro, MPASM, Hi-Tech C, CCS C.

**PC Compilers:** Delphi, C++ Builder, Visual Basic 6.

**Front Panels?**

Download the free Front Panel Designer to design your front panels in minutes.

Order your front panels online and receive them just in time.

**Front Panel Designer website:** www.frontpanelexpress.com

**NUTS AND VOLTS**

*It's here! The complete 2006, 05, and 04 years of Nuts & Volts on CD. Stored as PDFs, you can print, search, and easily store your digital magazines.*

**To order, go to:**

www.nutsvolts.com or call 1.800.769.4624

**Nuts & Volts 2005** Buy 2 or more CDs for only $21.95 each. Pack 25 for only $51.95.

**HIDmaker FS Combo:** Only $599.95

**Download the HIDmaker FS Test Drive today:**

www.TraceSystemsInc.com 301-262-0300

September 2007 NUTSVOLTS 95
Looking for robots? www.servomagazine.com

CLASSIFIED ADVERTISING
$50.00 Per Inch — No extra charge for color. Classified ad must be paid in full prior to the closing date or the ad will be placed in the following issue, at our discretion. Minimum charge is one inch with half-inch increments. No proofs will be sent. Ads to be typeset by Nuts & Volts must be received by the closing date. Supplied ads must be received by the artwork due date.

Call 951-371-8497 or email classad@nutsvolts.com for closing dates, available sizes, and special prepay discount offers.

MILITARY SURPLUS

Electronic Military Surplus

Fair Radio Sales
Website: fairradio.com
E-mail: fairradio@fairradio.com
Phone: 315-722-6252
Fax: 315-722-6252
Lima, OH 45804
Visa, Mastercard, Discover

Regular Retail Equipment: Tubes, Antenna

MICROCONTROLLERS

MicroStamp11 World’s Smallest 68HC11 Microcontroller Module!

- Tiny (1 by 1.4 in.), lightweight (0.5 oz.)
- on-board 5V reg., crystal, & reset chip
- choice of 8K or 32K EEPROM
- or 32K RAM + 32K EEPROM (64K version)
- SCI, SPI, Output Compare and Input Capture channels, timer, pulse accumulator
- all 14 I/O lines and 2 interrupt lines brought out to versatile 20-pin connector
- program in BASIC, assembler, or C
- easy code-loading with Docking Module

Starter Packages:
- 8K EEPROM (#MS11SP8K).......$49
- 32K EEPROM (#MS11SP32K)...$77
- 32K EE/32K RAM (MS11SP64K)$90

* includes MicroStamp11, manual, PC software (assemblers, SBASIC compiler, MicroLoad utility, and sample programs), serial cable, Docking Module, & accessories.

www.technologicalarts.com

AUDI/OVIDEO

www.MAT-COM.com

1.2 GHz 4 100Mbit Wireless ASSMBLY AV
- Super high power
- Standard transmit range
- 1200 ft.
- 2 dc power supplies

www.MAT-COM.com

4 Channel Color Wireless Camera Kit - Max Color CMOS Camera - 1 lux - 390 TVL - 2.4 GHz
- Good for 60 ft. in open
- Power supplies for both cameras & receiver included
- Output power 10 watts

www.LITEXPO.INC.com

Smart Media Player DAV
- 32GB capacity
- USB 2.0 (Hi-Speed)
- RJ45 Ethernet port
- Front panel controls
- 32GB TFcard slot
- 3.5mm headphone jack
- Optical remote control

www.BATTERYs.com

5V/3.3V from 1 or 2 AAAA

BATTERY DESULFATORS

www.wizbangplus.com

SOLAR PRODUCTS

www.HittConsulting.com

READER FEEDBACK

Fonte mentioned in the article that a more sophisticated filtering circuit would be better. I agree but am wondering if he has tried anything that he is happier with.

Thanks for any help you can provide. The ARB looks very interesting and I want to build it as soon as possible. Keep up the good work!

Jim Rybak
Grand Junction, CO

Response: The important characteristics of the op-amp in question are: 1) operate from a single supply; and 2) be fast enough (gain-bandwidth product >1 MHz). The LMC6042 has a GBW of only 0.1 MHz and is too slow. If used, there can be additional problems at the high frequency end. The LMC6442 has a GBW of only 9.5 kHz and cannot be used. (I used the LMC6082 because I had it on hand.)

The LM6142 is a good substitute. It operates from a single supply and has a GBW of 17 MHz. It is available from Jameco and other suppliers in the eight-pin DIP package (check the website at www.jameco.com). Note: The National Semiconductor website (and most other IC manufacturers) will often provide samples for free and/or provide access to small quantities for evaluation. The important requirements are that you have to provide a business address and a proper engineering job description. (A school address will probably work, too.)

In regards to the filtering question, this application seems to be an ideal project to incorporate a tunable switched-capacitor filter. There is already a variable high-speed clock available that is 25 times faster than the output. The July 2007 issue of N&V (Making Waves) discusses SC filters.

Hope this helps. It’s always good to get feedback from readers.
— Gerard Fonte

September 2007
Check out our new webstore for many more great titles!!

The Nuts & Volts Hobbyist STORE

CD-ROMS

GREAT ARTICLES – GREAT PROJECTS!

Catch up on 2002-2006 back issues
These CD-Roms are PC and MAC compatible. They require Adobe Acrobat Reader Ver. 6 or above. Don't worry. We have included Acrobat Reader Ver. 7 on each of the discs.

$24.95

Buy 2 or more $21.95 each

ELECTRONICS

+Practical Electronics for Inventors
by Paul Scherz
This intuitive, applications-driven guide to electronics for hobbyists, engineers, and students doesn't overload readers with technical detail. Instead, it tells you — and shows you — what basic and advanced electronics parts and components do, and how they work. Chock-full of illustrations, Practical Electronics for Inventors offers over 750 hand-drawn images that provide clear, detailed instructions that can help turn theoretical ideas into real-life inventions and gadgets. $39.95

+Troubleshooting and Repairing Major Appliances: 2nd Edition
by Eric Kleinert
Now covering both gas and electric appliances, the updated second edition of Troubleshooting and Repairing Major Appliances offers you a complete guide to the latest tools, techniques, and parts for troubleshooting and repairing any appliance. Packed with over 200 illustrations, the book includes step-by-step procedures for testing and replacing parts, instructions for reading wiring diagrams, charts with troubleshooting solutions, advice on using tools and test meters, safety techniques, and more. $59.95

+Electromechanical Devices & Components Illustrated Sourcebook
by Brian Elliott
Master the key elements of electromechanical components. Essential to the design, repair, or operation of your electromechanical devices, this quick-find reference provides illustrations, descriptions, and relevant calculations for 2,000 time-tested electromechanical components. $89.95

Troubleshooting Electronic Equipment
by R. S. Khandpur
From cell phones to medical instruments to digital and microprocessor-based equipment, this hands-on, heavily illustrated guide clearly explains how to troubleshoot, maintain, and repair all types of electrical equipment. The author covers all the essentials such as necessary tools, soldering techniques, testing, fundamental procedures, and mechanical and electrical components. $49.95

Encyclopedia of Electronic Circuits, Volume 7
by Rudolf F. Graf / William Sheets
An extensive library of 1,000 circuits from the bestselling, seven-volume Encyclopedia of Electronic Circuits. A virtual treasurehouse. An invaluable reference tool for every hobbyist, technician, student, and design professional. $39.95

Build Your Own Electronics Workshop
By Thomas Petruzzellis
YOUR DREAM ELECTRONICS LAB IS WAITING INSIDE!
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 — popular electronics author Tom Petruzzellis’ Build Your Own Electronics Workshop has it all! And the best part is, this book will save you money, big time! $28.95

Electronic Circuits — Fundamentals & Applications: Third Edition
by Mike Tooley
In Electronic Circuits, Mike Tooley provides all the essential information required to get to grips with the fundamentals of electronics, detailing the underpinning knowledge necessary to appreciate the operation of a wide range of electronic circuits, including amplifiers, logic circuits, power supplies, and oscillators. The third edition now offers an even more extensive range of topics, with extended coverage of practical areas such as circuit construction and fault finding, and new topics including circuit simulation, electronic CAD, and a brand new chapter devoted to the PIC microcontroller. $34.95
THE EVIL GENIUS SERIES

$24.95 each
Subscribers Save 10%
on two or more!

GREEN POWER ENERGY

Alternative Energy Demystified
by Stan Gibilisco
Publish Date: October 23, 2006
The fast and easy way
to get up-to-speed on
alternative energy.
Because of current
events, geopolitics, and
natural disasters, the
cost of fuel is front
and center in our
lives. This book pro-
vides a concise look
at all forms of energy,
including fossil fuels, electric, solar,
biodiesel, nuclear, hydroelectric, wind, and
renewable fuel cells. You will get explana-
tions, definitions, and analysis of each alter-
native energy source from a technological
point-of-view. $19.95

Designing & Building Fuel Cells
by Colleen Spiegel
Designing & Building Fuel
Cells equips you with a
hands-on guide for the
design, modeling, and
construction of fuel cells
that perform as well or
better than some of the
best fuel cells on the
market today. Filled with
over 120 illustrations
and schematics of fuel
cells and components,
this “one-stop” guide covers fuel cell appli-
cations, fuels and the hydrogen economy,
fuel cell chemistry, thermodynamics and
electrochemistry, fuel cell modeling, materi-
als, and system design, fuel types, delivery,
processing, and much more. $89.95*

The Complete Idiot’s Guide to
Solar Power for Your Home
by Dan Ramsey / David Hughes
Publish Date: May 2007
The perfect source
for solar power —
fully illustrated. This
book helps readers
understand the
classics of solar
power and other
renewable energy
sources, explore
whether solar
power makes sense
for them, what their
options are, and what’s involved with
installing various on- and off-grid systems.
$18.95

Pocket Ref
by Thomas J. Glover
This great little book
is a concise, all-pur-
pose reference fea-
turing hundreds of
tables, maps, formu-
las, constants, and
conversions. AND it
still fits in your shirt
pocket! Goes where
you go! $12.95

Pocket Idiot’s Guide
to the iPhone
by Brown, Damon
Publish Date: August 7, 2007
See what this new
technology can do for
you. The Pocket Idiot’s
Guide to the iPhone is a
quick reference guide
that includes trou-
bleshooting info and
helpful tips for the non-
geeky iPhone owner. This
could be helpful for those
on the fence about pur-
chasing, as well. $9.95

New To Our Store
Gadgeteer's Goldmine
by Gordon McComb
Gordon McComb’s
Gadgeteer’s Goldmine
is one of the most
exciting, well-rounded
collections of
electronic projects
available anywhere,
featuring experiments
in everything from
magnetic levitation
and lasers to high-tech
surveillance and digital
communications. $29.95

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

September 2007 NUTSVOLTS 99
Attention Subscribers ask about your discount on prices marked with an *

The Nuts & Volts Hobbyist STORE

MICROCONTROLERS

Programming the PIC Microcontroller with MBASIC
by Jack Smith
No microcontroller is of any use without software to make it perform useful functions. This comprehensive reference focuses on designing with Microchip’s mid-range PIC line using MBASIC, a powerful but easy to learn programming language. It illustrates MBASIC’s abilities through a series of design examples, beginning with simple PIC-based projects and proceeding through more advanced designs. $59.95

Programming and Customizing the PICAXE Microcontroller
by David Lincoln
Here’s everything you need to harness the power of PICAXE, the inexpensive yet versatile chip that’s taken the electronics community by storm. This beginner-friendly guide from IT pro and PICAXE expert David Lincoln shows you just what Revolution Education’s PICAXE can do — and helps you make it do it! Packed with ready-to-build projects for all the flavors of PICAXE, $39.95

PIC Microcontroller Project Book
by John Iovine
The PIC microcontroller is enormously popular both in the US and abroad. The electronics hobbyist market has become more sophisticated. This new edition is fully updated and revised to include detailed directions on using both versions of the microcontroller, with no-nonsense recommendations on which is better served in different situations. $29.95

BACK ROOM

BOOK BLOWOUT SPECIAL
Physical Computing
Sale Price $9.65, Reg. $29.99
The computer revolution has made it easy for people with little to no technical training to use a computer for everyday tasks. But what about more imaginative purposes such as starting your car, opening a door, or tracking the contents of your refrigerator? Physical Computing will not only change the way you use your computer, it will change the way you think about your computer, how you view its capabilities, how you interact with it, and how you put it to work for you. It’s time to bridge the gap between the physical and the broader world of computing. Only a Few Copies Left

Physical Computing

TO OPEN THE BOOK OF KNOWLEDGE GO TO www.nutsvolts.com AND CLICK ON STORE. PERUSE YOUR WAY THROUGH OUR GREAT SELECTION OF OUTSTANDING TITLES.

SPECIAL
PICmicro Microcontroller Pocket Reference
by Myke Predko
Designed to complement Programming & Customizing the PICMICRO, this book contains a minimum of verbiage and serves as an immediate device, code, and circuit lookup for experienced PICMICRO applications designers. Sale Price $23.95, Reg. $29.95 Only a Few Copies Left

BACK ROOM

PROJECTS

H-racer-and-Hydrogen-station
With this kit, you now can see and feel the future of energy generation in your own hands! Recently named as one of the Best Inventions of 2006 by Time Magazine, the H-racer is now the best selling fuel cell product in the world. For more info, go to our online store at www.nutsvolts.com, $118.00*

Magic Box Kit
As seen in our April issue. We received such great reviews on this magic box kit that we decided to offer it in our store. This unique DIY construction project blends electronics technology with carefully planned handcraftsmanship. Its delightful innovation will surely amuse you. More importantly though, it bewilders, baffles, and mystifies those observing this subtle magic trick. Subscriber’s Price $39.95 Non-Subscriber’s Price $45.95 Both include an article reprint.

To open the book of knowledge go to www.nutsvolts.com and click on store. Peruse your way through our great selection of outstanding titles.
QUESTIONS

I am looking for a specific recommendation of hardware and software within the PIC, BASIC Stamp, etc., single board microprocessor products that will allow me to trigger (i.e., send an ASCII string) and then take the continuous output from an electronic gauge via RS-232 and display it on an LCD screen. I get overwhelmed with all the choices of available products. I would presume to program in some form of Basic or C (preferably Basic).

Bill Ryder
Victoria, MN

I'm looking for a schematic for a low power wireless switch. Something simple, using transistors, with a range of 25-50 feet. Both the transmitter and receiver should be battery powered and act as a simple switch.

Bill Sutkowski
Fairfield, CA

Our club was vandalized several times in the last month. I need to put a video camera at the entry way and turn on a video recorder using a motion detector to record anyone who enters. A commercial motion detector security camera with a 12 minute digital video recorder made to look like a clock is available, but costs $180.

Chet Dera
via email

I would like to use an old 12 inch B&W monitor (RCA connection a Dell Optiplex computer) to show text parameters using this as an EMC CNC controller only.

Jesse Brennan
Austin, TX

I'm looking for plans to replace an old Dranetz 626 meter (records AC power spikes, voltage drops, and their pulse widths). I'm hoping someone may have something that can provide the same functionality.

Jim M.
via email

I would like to see a design for a variable frequency drive to operate single phase 110/220V induction motors in the 1/2 to 2 HP range. I think this would be of interest.
to many readers having stationary machine tools if it could be built at a reasonable price.

#09076  Morty Goldsmith
Quebec, Canada

I need a four-way directional switch which includes the soft rubber button cover. I’m wanting something similar to the D-pad of a game controller. Where can I look for something like this?

#09077  Joe Kissell
via email

Is there a way to figure out the pinout of a late model auto radio, out of the vehicle, using test equipment typically purchased at a yard sale? No plug on the back. I want to bench test the radio to see if it works.

>>> ANSWERS

#09078  Clarence Wilken
Freeport, IL

[7072 - July 2007]

I would like to read a barcode attached to the bottom of a Model Railroad Car (HO Scale). This is to sense location on the track. The barcode would be read as the car passes over the reader. Multiple readers would allow a computer to track train progress on a large layout (14’ x 204’).

The sensor would be similar to a wand but with a .75” read range.

As you already know, the most simple barcode reader is the LED reflective wand. This requires the operator’s hand movement to provide the scanning action, and also may require contact with the barcode label.

It’s possible to modify a wand to project the beam and avoid contact, using a relay lens. Experimental and inexpensive lenses can be found in surplus catalogs or here (www.surplushed.com/pages/category/lenses_1.html).

A problem with this method is that the rolling stock must be moving through the beam for it to register. Also, the multiple visible red lights might distract from the aesthetics of the model railroad layout.

Another barcode reader technique uses a scanning laser (handheld pistol, supermarket checkout counter scanner), but this is a complex precision mechanism with moving parts, and likely emits noise, too.

Have you considered RFID instead of barcodes? There are now hobby level RFID products from www.parallax.com. The Reader Module might work under the existing track, or could be fashioned into an adjacent model building or similar. No contact is required, and the tags can be read when the rolling stock is not moving.

Peter J Stonard
Campbell, CA

[7074 - July 2007]

I recently bought six solar cells and started playing with them. I noticed that even though these solar cells are from the same company and the same make, they produce different outputs. I was wondering if this was normal? Also, when I connected two cells together in parallel, if I did not put a diode in series with the cells for each cell, I would get a lower value than the two cells divided by two ([(Cell1 + Cell2)/2]. I was not expecting this result.

Do you always need to use a diode in parallel with each cell when trying to build up the amperage? I guess it goes without saying that you only need one diode in series with two cells in series for greater voltages.

Background: If you test the cells by themselves (open circuit), four of these solar cells are producing 15 to 17 VDC with two only producing eight volts. The short circuit current is .100 amps to .140 amps.

Also, could you explain how to choose a solar cell for a project? I understand if I am running 4X 12V fans at 100 mA, I would add up the total amps: 400 mA for the project. But what I am seeing from my meter is my voltage drop is a lot lower than what I would expect for a parallel circuit. Do I need to add a capacitor or something?

Depending upon where you got your PV cells, they may be factory seconds that were not performing to the tight specs of premium cells. So, it’s normal for cells to have wide performance specs, and factory constructed arrays are created from selected and matched cells.

PV cells should not be directly connected in parallel. A PV cell will shunt current if it is not illuminated enough to overcome the voltage from other cells in the network. The correct arrangement is to place isolating diodes in series with cells that are connected in parallel.

To get higher voltage, cells may be placed in series, and a single diode placed in that series string before the array is placed in parallel with a similar array or multiple arrays (i.e., one diode per series string). These are called “blocking diodes.”

Additional diodes should be placed across each cell in a series string. These are called “by-pass diodes.” Dark cells will rob voltage from the string (and may be damaged) unless a diode is installed. An illuminated cell generates about 500 mV; useful panels are constructed of multiple cells in series (i.e., 36 cells produce about 18V when illuminated — enough to charge a 12V battery).

For an in-depth discussion, read this technical paper: www.windsun .com/PDF/ieee6.pdf

The PV cells are not perfect and have internal resistance that causes the terminal voltage to drop under load (as you recorded). A fix for this is to oversize the panel and provide a regulator to feed steady voltage to your load (the fans). A large capacitor might help, but a better solution is a storage battery, which would be charged by the panels during full sunlight and discharged by the fans during shade or at night. To protect the fans, use a linear regulator (LM7812 or similar) to derive 12V from the varying PV panel voltage, and to protect the battery from over-discharge, consider a low voltage cut-out circuit.

Peter J. Stonard
Campbell, CA

[7076 - July 2007]

I can purchase a DC motor speed controller for a 1.5 HP motor for my
Bridgeport mill, or an inverter duty AC motor and controller, too. What are the advantages to both of these setups and why were they made in the first place?

DC motors seem to be losing favor in the face of 3ph + VFD conversions which are generally cheaper and easier to install. Bridgeport type OEM motors have a somewhat uncommon motor mount and adapters must be made to accommodate replacements; especially DC where the motor is more likely to have an atypical form factor.

A Google on Bridgeport motor mount adapters brings up several articles on this, as well as an adapter for a base mount (non-face mount) motor. DC motors have some theoretical advantages if the controller is designed to exploit them. See the discussion at www.truetex.com/dcdrv.htm. DC motors do not have a fixed speed; speed is entirely controlled by the applied voltage and hence can operate over very wide ranges from 10 or so RPM to 5-8K RPM. DC controllers that chop rectified 60 Hz will necessarily give very poor results at low speed because the pulses are so coarse and marked cogging will occur below the low to mid 100s RPM motor speed.

By increasing the pulse rate and using feedback from current draw, the motor can be smoothed out significantly and operate with good torque at lower speeds. PWM drivers operating above 5 kHz will have imperceptible cogging at low speeds.

Since Bridgeport spindles are designed for the 2,500-4,500 speed range from 1,200-1,800 RPM AC motors, higher motor speed than 2,000 or so is not needed. Since power output of any motor is speed-dependent, a 1 HP DC motor at 4,000 RPM will put out 0.5 HP at 2,000 RPM.

A DC motor should be TEFC and industrially rated with awareness that operated at substantial speed reductions and high current, the fan may not be running fast enough to cool the motor.

In regards to 3ph AC/VFD, if your OEM motor is 3ph and 220-440V rated, it will work great as-is with a 220V single phase VFD. Inverter rated motors are specially insulated to withstand high voltage pulses that 440V VFDs generate, and this is not a problem with 220V VFD.

The same speed limit exists: Although any VFD worth its cost (generally much cheaper than an equivalent DC drive new) can easily drive a motor up to 100 Hz, it is not a good idea to operate the machine tool more than 25% above its rated max spindle speed. VFDs operate at 10-17 kHz switching speed, so cogging is never a problem, but as you lower the speed, the motor effective power output goes down proportionately. Practically, this means not going below 20 Hz.

Steven Hodges
Birmingham, AL

---

**Announcing the 2nd Annual Schmartie Awards**

$1000+ Grand Prize

SchmartBoard is looking for the next SchmartModule design. SchmartModules are functional circuit blocks such as RS232 or Power that physically connect to other SchmartBoards’ to make prototyping easier. The winner will get $1000 plus 10% continued commission and the notoriety of having their name on every one of the winning SchmartModules sold worldwide.

- **2nd Prize** - A Link Instruments DSO8502 500Mhz Digital Oscilloscope
- **3rd Prize** - A Weller WD1002 Soldering Station
- **Honorable Mentions** - Three people will win a Parallax Boe-Bot

**All Valid Entries** - Will receive a free SchmartBoard Prototyping Boards and a SchmartBoard t-shirt

**Co-Sponsored By:** Servo and Nuts and Volts Magazines, Intellect Lab, RB Technology, Jameco Electronics, Parallax, Cooper Tools, Fry’s Electronics, Link Instruments, Circuit Specialists and Mouser Electronics

Visit [www.schmartboard.com](http://www.schmartboard.com) for details
Ramsey Electronics is a leading distributor of sophisticated devices for both the hobbyist and the electronic experimenter. Located just south of Rochester, NY, they serve this market both by catalog as well as online sales. The business was started in the early ‘70s by John Ramsey while he was attending college. As a youngster, he was always interested in electronics, and built a number of Heath kits and Knight-Kits. Taking note of the interest in this type of products, he began selling digital clock kits and LED “blinky” kits. Both technologies were new to the hobby. He knew it was necessary to provide more than just a bag of parts and assembly instructions, so each of his kits included the complete theory of operation, as well as circuit applications. Customers took note that when they built a Ramsey Kit they not only ended up with a working product, but also learned the entire circuit theory and all of its applications. When Heath and Knight-Kit closed their doors in the mid ‘80s, Ramsey kit sales skyrocketed. Hobbyists everywhere looked forward to the next edition of the “Ramsey Catalog” to see what new products were available. The catalog – originally eight pages in black and white – changed to spot color in the late ‘80s, and since 1994, it has been published in full four-color offset.

In 2000, Mr. Ramsey retired from the business for some long deserved relaxation. Today, the company is privately owned and operated by the key management team that helped make it successful over the past several decades. Mike Leo is the current Vice-President of Operations. When interviewed for this column, he replied to the following questions:

**Marvin:** Mike, what is your previous history and experience in the electronics field? How far back does that go?

**Mike:** I go back to ninth grade when I built my first five-tube superhet AM radio kit! From there, it was the Knight-Kit Star Roamer SWL receiver and even the Heathkit color TV. I will never forget the absolute thrill I had when I first turned those kits on and they worked. It’s the same thrill we always try to insure for all our customers with our hobby kits. If we have done that, we have succeeded!

**MM:** How large are the Ramsey facilities? How many employees do you have?

**ML:** We have an 11,000 square foot facility that handles all of our operations from design engineering to shipping. From the original digital clock and LED Blinky Kits of the ‘70s, our product line has expanded to the point where we now carry over 200 manufactured kits and products. We currently have 20 plus employees, some of whom have been with us for more than two decades.

**MM:** What would you say is your most popular product today?

**ML:** Actually, the most popular hobby product we’ve ever released is also one of our more recent. It’s our UP24 pressure sensor, and we’re really proud of it. It can measure elevations to accuracies of a third of an inch! It’s proven to be extremely popular not only for the hobbyist but for land surveyors and other professionals. It’s also one of the most complex kits we have brought to the hobbyist. When they get done putting one of these together, they’ll know all there is to know about SMT (surface mount technology) and a lot more about the latest techniques.

**MM:** Finally, what can you tell us about the business today?

**ML:** In regard to hobby kits, a lot has gone on over the past few years. We have seen through-hole components become less and less available as SMT components replace them. That is a major change for the hobbyist. You simply can’t use the Weller soldering gun I used back in the ninth grade. We saw this coming early on and started provided SMT training kits and soldering guides right away. Today, that soldering gun is replaced with a miniature soldering station, tweezers, and a magnifying glass.

Just like the early digital clock and LED Blinky, it’s just new technology, and we will continue to provide whatever is necessary for the hobbyist to learn and understand it. We are known for unique products and unsurpassed customer support. It is our goal to continue that legacy regardless of technology changes.
Our Premium All in One Repairing System

• All in One system. Combines the function of a Hot Air Gun, a Soldering Iron and a Desoldering Gun.
• Microprocessor controlled ESD safe unit. All digital display of hot air temperature, soldering iron temperature, desoldering gun temperature and air pressure with touch type panel controls.
• The desoldering tool comes with zero crossing circuit preventing electrical surges and is equipped with air cylinder type strong suction vacuum pump.
• The 24V soldering iron is compatible with the compound tip design by connecting the ceramic heater, sensor, control unit and tip as one. Designed for efficiency. Replacement of tips with easy slip in/out method.
• Compatible with various type of air nozzles.
• Full compliment of nozzles & tips are available.
• Use with lead-free or standard solder.

Soldering Equipment & Supplies  Soldering Stations

Programmable DC Power Supplies

The CSI 3600 Series Programmable DC Power Supplies are equipped with a back-lit LCD display, number keypad and a rotary code switch for ease of use & quick programming. Voltage, Current & Power can all be displayed on the LCD or computer screen (with optional RS-232 interface module). It can be operated at constant current mode, constant voltage mode & constant power mode. It also can be set with maximum limits for current & power output. Ideal instruments for scientific research, educational labs or any application requiring a sophisticated DC-power source.

<table>
<thead>
<tr>
<th>Model</th>
<th>CSI3644A</th>
<th>CSI3645A</th>
<th>CSI3646A</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC Voltage</td>
<td>0-18V</td>
<td>0-36V</td>
<td>0-72V</td>
</tr>
<tr>
<td>DC Current</td>
<td>5A</td>
<td>3A</td>
<td>1.5A</td>
</tr>
<tr>
<td>Power (max)</td>
<td>90W</td>
<td>108W</td>
<td>108W</td>
</tr>
</tbody>
</table>

Details at Web Site  Test Equipment  Power Supplies

Protek 20MHz 2CH Analog Oscilloscope w/Component Tester

• 2 channels 2 traces
• 20MHz Bandwidth
• Z axis (intensity modulation)
• TV video sync filter
• Component test function
• Vertical Deflection Operating Mode: CH A, CH B, DUAL, ADD
• Sensitivity: 5mV/20mV
• Bandwidth: DC-20MHz / AC: 10Hz-20MHz
• Horizontal Deflection Sweep Mode: AUTO, NORM
• Sweep Time: 0.2us-5s/div

Details at Web Site  Oscilloscopes/Oustanding Prices

ESD Safe CPU Controlled SMD Hot Air Rework Station

The heater and air control system are built-in and adjusted by the simple touch of the front keypad for precise settings. Temperature range is from 100°C to 480°C / 212°F to 896°F, and the entire unit will enter a temperature drop state after 15 minutes of non-use for safety and to eliminate excessive wear.

• CPU Controlled
• Built-in Vacuum System
• Temperature Range: 100°C to 480°C / 212°F to 896°F
• 15-Minute Stand-By temperature "sleep" mode
• Power: 110/120 VAC, 320 W maximum

Details at Web Site  Oscilloscopes/Oustanding Prices

Dual Output DC Bench Power Supplies

High stability digital read-out bench power supplies featuring constant voltage and current outputs. Short-circuit and current limiting protection is provided. SMT PC boards and a built-in cooling fan help ensure reliable performance and long life.

• Source Effect: 5x10^-8 A/V
• Load Effect: 5x10^-8 A/V
• Ripple Coefficient: <250uV
• Stepped Current: 30mA +/- 1mA

“All 3 Models have a 1/5SDC Fixed Output on the rear panel”

CS13003X-5: 0-30Vdc/0-5amp 1-4: $105.95 5+: $99.50
CS15003X: 0-50Vdc/0-5amp 1-4: $114.95 5+: $109.00
CS13005X: 0-30Vdc/0-5amp 1-4: $119.00 5+: $114.00

Details at Web Site  Test Equipment  Power Supplies

Triple Output Bench Power Supplies

with Large LCD Displays

• Output: 0-30VDC x 2 @ 3 or 5 Amps & 1ea. fixed output @ 3VDC@3A
• Source Effect: 5x10^-8 A/V
• Load Effect: 5x10^-8 A/V
• Ripple Coefficient: <250uV
• Stepped Current: 30mA +/- 1mA
• Input Voltage: 110VAC

CS13003X: 0-30VDC @3A $180.00 5+: $183.00
CS15005X: 0-30VDC @5A $239.00 5+: $229.00

Details at Web Site  Test Equipment  Power Supplies
The CX-102 series of digital panel meters is a sophisticated PC based scope adaptor providing performance Digital Storage Oscilloscope. This is a simplified PC based scope adaptor providing performance comparable to mid-high level stand alone products costing much more! Comes with two probes.

**Digital Storage Oscilloscope Module**

Convert any PC with USB interface to a high performance Digital Storage Oscilloscope. This is a sophisticated PC based scope adaptor providing performance comparable to mid-high level stand alone products costing much more! Comes with two probes.

**Price Breakthrough!**

PC based Digital Storage Oscilloscope, 200MHz 5GS/s USB interface.

<table>
<thead>
<tr>
<th>Model</th>
<th>Sampling Rate</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDS-2064</td>
<td>60MHz, 2Ch DSO</td>
<td>$399.00</td>
</tr>
<tr>
<td>GDS-2102</td>
<td>100MHz, 2Ch DSO</td>
<td>$799.00</td>
</tr>
<tr>
<td>GDS-2202</td>
<td>200MHz, 4Ch DSO</td>
<td>$1470.00</td>
</tr>
<tr>
<td>GDS-2204</td>
<td>200MHz, 4Ch DSO</td>
<td>$1800.00</td>
</tr>
</tbody>
</table>

Details at Web Site | Test Equipment | Oscilloscope/Outstanding Price

**Jumbo LCD 9V Independent Ground Panel Meter**

The PM-1028A is an enhanced version of our best selling PM-128A. The E-version can be set to work with either a SYD or SYVC power source, will perform with either a common ground or an isolated ground, and is supplied with easy to use jumper points so the end user can easily set the measurement range required.

**Price Breakthrough!**

Jumbo LCD 9V Independent Ground Panel Meter (enhanced version)

**Details at Web Site** |**Panel Meters**|**Digital Panel Meters**

**Soldering Equipment & Supplies**

- SMD Hot Tweezers
- SMD Pliers
- 3-1/2 Digit LED Panel Meter
- 3-1/2 Digit LED Panel Meter/3V Common Ground

**Additional Panel Motors on Web Site**

**Focused Infrared Soldering System**

The CSI-IR1 is a modular and flexible infrared technology soldering system for effective high yield rework of even the most advanced packages, as well as being suitable for lead or lead-free soldering. It brings sophisticated targeted network soldering systems to the simplicity of a handtool and can be used solely for reworking BGA’s, micro-BGA, QFPs, PLCCs, SOICs and SMDs.

**Price Breakthrough!**

Focused Infrared Soldering System

**Details at Web Site** | **Industrial Continuity Tester**

Instek Test Equipment at website!
Why the Propeller Works

by Chip Gracey, creator of the Propeller chip and President of Parallax Inc.

I am the person who designed, debugged, tuned, and tested the Propeller chip. This project took eighteen years of my time, plus two years of a layout engineer's time. An enormous amount of attention went into every aspect of the Propeller's implementation and testing, and I allowed no compromises.

The Propeller was an entirely "full-custom" effort. Every polygon of the Propeller's mask artwork was made here at Parallax. We designed our own logic, RAM, ROM, PLLs, bandgap references, oscillators, and even 15+ hard-wired I/O pads. All these structures were first fabricated on test chips and then thoroughly examined, often resulting in design changes. This yielded an ideal set of known-good blocks, which could be confidently applied to the overall design. Then, the whole chip was fabricated and subsequently tested at many levels. This allowed us to fix any problems resulting from integration and to fine-tune the clocking systems that are key to the Propeller's low-power consumption. The final chip, which is the only version we've ever sold, is the third iteration of this whole-chip process and has no known problems.

The Propeller was given the kind of thorough design treatment that almost no other chips receive today. It used to be that every chip was full-custom, and all of its transistors and wiring were designed by hand, for the point of application. As semiconductor technology shrunk, though, the prevailing design methodology shifted away from the kind of specialization toward generalization and abstraction, so that designs of greater complexity could be practically realized. The modern design methodology centers around hardware description languages, IP block reuse, and the automated placement and interconnection of potentially billions of gates. The end silicon result is necessarily an incomprehensible rest nest of wiring, standard cells, and IP blocks, usually more of which were designed by the engineers applying them. This methodology is certainly a boon for very complex designs, but it has become the standard approach for designing almost any chip containing logic today. For chips, the old method means small dies, high speed, low power, and excellent performance, whereas the modern method tends to generate bigger dies, lower speed, more heat, and sometimes bugs from IP which you have no control over. I'm sure you get the idea.

Through an exceptional effort, we made the Propeller as electrically robust and efficient as we could. Yet, the core quality of the Propeller really resides in its architecture. The architecture is what took the first six years of development to iron out, and the architecture is what makes people. All the effort that went into the silicon implementation was to ensure that this core quality was ideally housed.

The story of the Propeller's architecture would be a book in itself but get an idea of the pitch, you can visit the Propeller discussion forum (http://forums.parallax.com) and witness the excitement of people doing things they never thought possible. They are finding the Propeller to be a great vehicle for invention and discovery, as well as the means to realize complex embedded systems that are not possible with any other chip. A few forum members have even said that the Propeller has drawn them back into software and electronics after long absences.

The Propeller is tough, reliable, and low-power. It's no illusion, and no accident.

We pride on a very long sales life for the Propeller and we have no intention of diluting the concepts with many slight variants, for which you'd inevitably be getting end-of-life notices after a few years. This is good news for customers, because they are the ones who are going to be making investments in programming that will, in turn, dwarf the energy that we spent developing the Propeller. We made a platform that is, hopefully, deserving of their coming efforts.

Sincerely,

Chip Gracey
President
Parallax, Inc.

Learn more at www.parallax.com/propeller