Mobile Data Storage (p. 32) • Practical, Low-Rate Battery Charging (p. 38) • The Benefits of Beamforming (p. 69)

www.circuitcellar.com

# CIRCUIT CELLAR

THE MAGAZINE FOR COMPUTER APPLICATIONS

#221 December 2008

# PROGRAMMABLE LOGIC

Use the Verilog Language for Programmable Logic Design

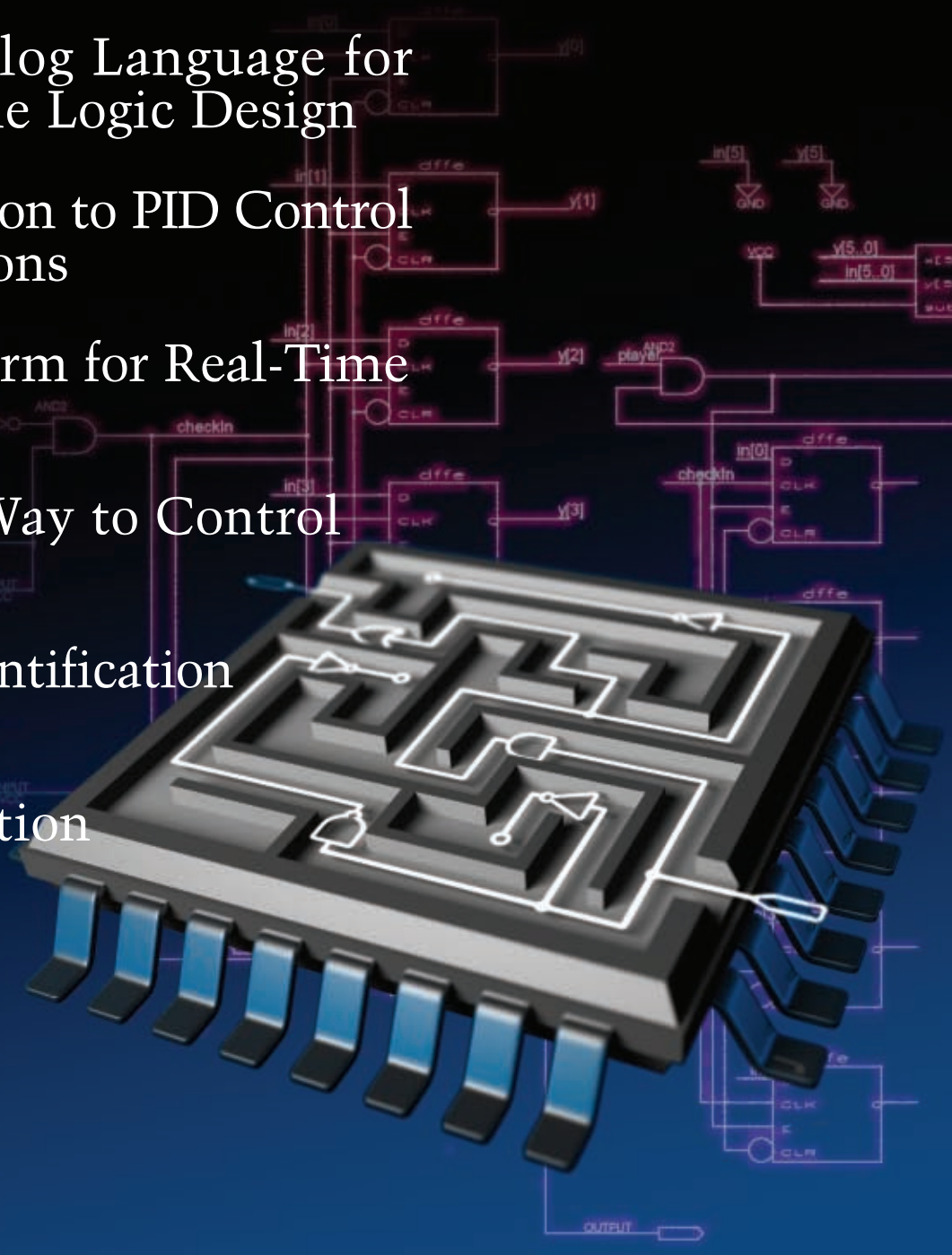
An Introduction to PID Control and Regulations

Build a Platform for Real-Time Animation

A "Handy" Way to Control Small Parts

Electronic Identification Made Simple

Next-Generation Design Tools




0 74470 75349 0
$5.95 U.S. ($6.95 Canada)

# SERIAL TO ETHERNET SOLUTIONS



Device P/N: SB70-100CR
Kit P/N: NNDK-SB70-KIT

$69
*Qty. 100*

## SB70

2-port serial-to-Ethernet server
with TTL support



Device P/N: SB72-EX-100CR
Kit P/N: NNDK-SB72EX-KIT

## SB72EX

$139
*Qty. 100*

2-port serial-to-Ethernet server
with RS-232 & RS-485/422 support



Device P/N: PK70EX-232CR
Kit P/N: NNDK-PK70EX232-KIT

$269
*Qty. 100*

## PK70EX232

4-port serial-to-Ethernet server
with RS-232 support

Simple Ethernet connectivity
for serial devices

Works out of the box -
no programming is required

Customize to suit any
application with low-cost
development kit

**Features:**

10/100 Ethernet

TCP/UDP mode

DHCP/Static IP Support

Data rates up to 921.6kbps

Web-based configuration

## Need a custom solution?

NetBurner **Serial to Ethernet Development Kits** are available to customize any aspect of operation including web pages, data filtering, or custom network applications. All kits include platform hardware, ANSI C/C++ compiler, TCP/IP stack, web server, e-mail protocols, RTOS, flash file system, Eclipse IDE, debugger, cables and power supply.

NetBurner
Networking in One Day!

**Information and Sales** | sales@netburner.com
**Web** | www.netburner.com
**Telephone** | 1-800-695-6828

freescale™
Alliance Member

# 5 Competitive Advantages

## Overseas Manufacturing

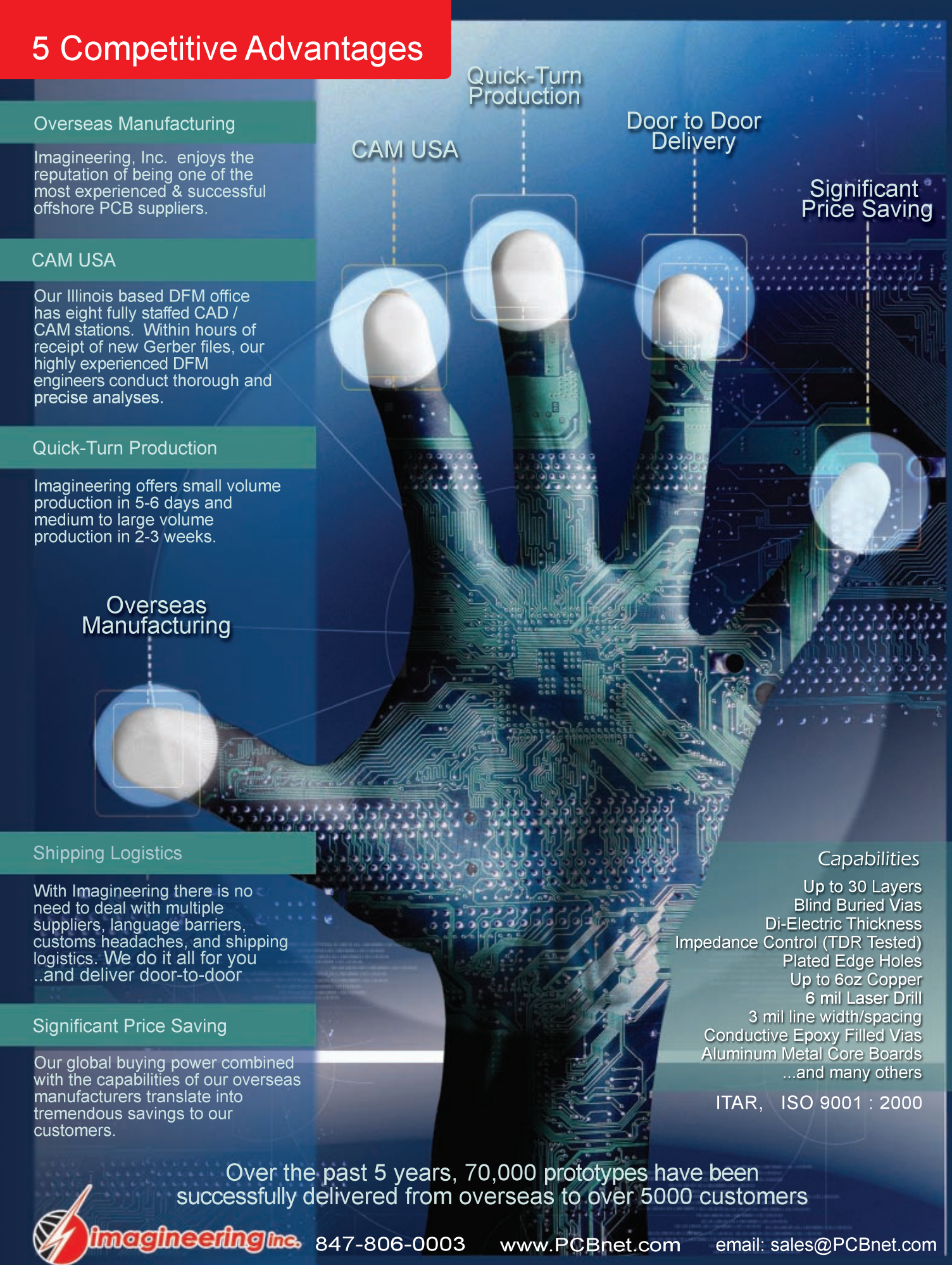
Imagineering, Inc. enjoys the reputation of being one of the most experienced & successful offshore PCB suppliers.

## CAM USA

Our Illinois based DFM office has eight fully staffed CAD / CAM stations. Within hours of receipt of new Gerber files, our highly experienced DFM engineers conduct thorough and precise analyses.

## Quick-Turn Production

Imagineering offers small volume production in 5-6 days and medium to large volume production in 2-3 weeks.




## Shipping Logistics

With Imagineering there is no need to deal with multiple suppliers, language barriers, customs headaches, and shipping logistics. We do it all for you ..and deliver door-to-door

## Significant Price Saving

Our global buying power combined with the capabilities of our overseas manufacturers translate into tremendous savings to our customers.

## *Capabilities*

Up to 30 Layers
Blind Buried Vias
Di-Electric Thickness
Impedance Control (TDR Tested)
Plated Edge Holes
Up to 6oz Copper
6 mil Laser Drill
3 mil line width/spacing
Conductive Epoxy Filled Vias
Aluminum Metal Core Boards
...and many others

ITAR, ISO 9001 : 2000

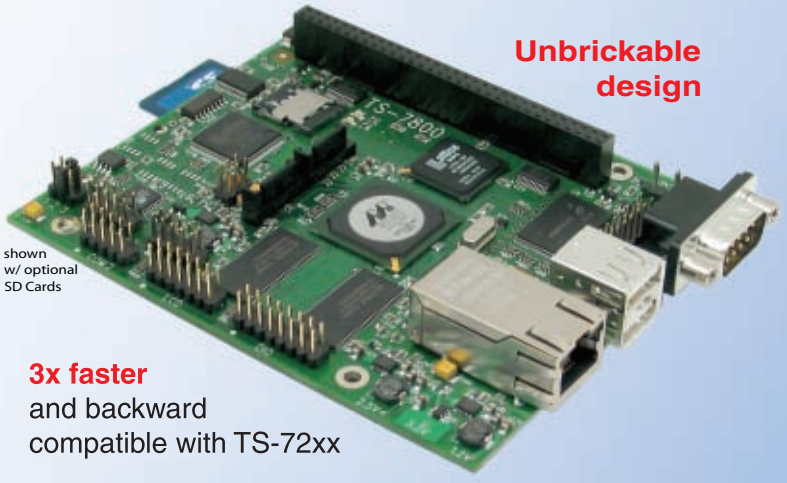
Over the past 5 years, 70,000 prototypes have been successfully delivered from overseas to over 5000 customers

imagineering inc. 847-806-0003 www.PCBnet.com email: sales@PCBnet.com

23 YEARS IN BUSINESS...AND STILL GOING STRONG

# Embedded Single Board Computers

## High-End Performance with Embedded Ruggedness




shown w/ optional SD Cards

**Unbrickable design**

**3x faster** and backward compatible with TS-72xx

### TS-7800
### 500 MHz ARM9

- Low power - 4W@5V — $229 qty 100
- 128MB DDR RAM
- 512MB high-speed (17MB/sec) onboard Flash — $269 qty 1
- 12K LUT programmable FPGA
- Internal PCI Bus, PC/104 connector
- 2 host USB 2.0 480 Mbps
- Gigabit ethernet
- 2 SD sockets
- 10 serial ports
- 110 GPIO
- 5 ADC (10-bit)
- 2 SATA ports
- Sleep mode uses 200 microamps
- Boots Linux in < 2 seconds
- Linux 2.6 and Debian by default

## Low Price, Low Power, High Reliability using Linux development tools




TS-7200 shown with optional A/D converter, Compact Flash and RS-485

- options include:
  onboard temperature sensor, A/D Converter 8 channel 12 bit, Extended Temperature, Battery Backed Real Time Clock, USB Flash, USB WiFi

### 200 MHz ARM9
Power as low as 1/4 Watt

- 8 boards, over 2000 configurations — as low as $99 qty 100
- Fanless, no heat sink
- SDRAM - up to 128MB — $129 qty 1
- Flash - up to 128MB onboard
- 10/100 Ethernet - up to 2
- DIO lines - up to 55
- 2 USB ports
- COM ports- up to 10
- Programmable FPGAs
- SD card option
- VGA video
- LCD ready
- Linux, Real Time extension, Debian

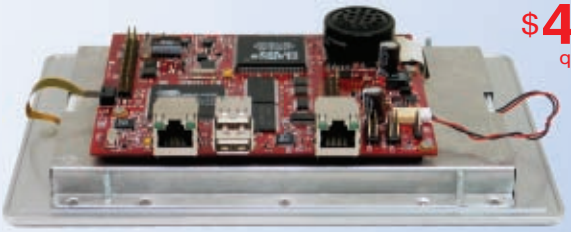
- Over 20 years in business
- Open Source Vision
- Never discontinued a product
- Engineers on Tech Support
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products stocked and available for next day shipping

**Design your solution with one of our engineers (480) 837-5200**

# Featured Products and PC/104 Peripherals

NEW!



## TS-TPC-7390

200MHz ARM9 Touch Panel Computer

$449 qty 1

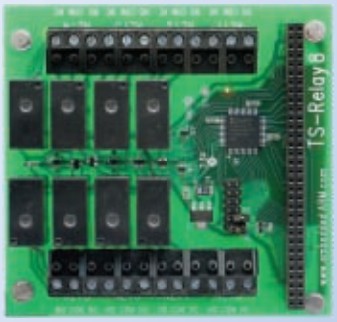
- Low Power, Industrial Quality Design
- Mountable aluminum frame
- 64MB SDRAM (128MB opt)
- 512MB Flash w/ Debian Linux
- Programmable FPGA- 5K LUT
- 7" Color TFT-LCD Touch-Screen
- 800x480 customizable video core
- Dedicated framebuffer- 8MB RAM
- Audio codec with speaker
- Boots Linux 2.6 in about 1 second
- Unbrickable, boots from SD or NAND
- Runs X Windows GUI applications
- Runs Eclipse IDE out-of-the-box

NEW!

## TS-RELAY8

Eight Software Controlled Relays

- 8 SPDT relays
- Up to 277 VAC @ 5A
- Up to 30 VDC@ 5A
- Software controlled
- 40mA draw per coil
- I/O jumpers




$89 qty 1

NEW!

## TS-ADC16

ADC, DAC and Digital I/O

- 16 16-bit ADCs
- Up to 100Ksps (10us)
- 1KB ADC RAM-FIFO
- 4 ADC voltage ranges
- Prog. pacing clock
- Externally triggered
- 4 inputs, 1 output
- 4 12-bit DACs
- 4 16 bit counters

$169 qty 1




see our website for x86 SBCs, peripherals and option details

**Technologic Systems**

We use our stuff.

Visit our TS-7800 powered website at

www.embeddedARM.com

# TASK MANAGER

## 2008 I/O

When any good engineer worth his solder finishes a long-term project, he immediately begins assessing what he accomplished in an effort to prepare for future endeavors. We like to do the same thing. So, after finishing this issue, we reviewed our 2008 editorial endeavor—publishing 12 fresh issues of *Circuit Cellar*—for the purpose of noting our accomplishments and planning what is to come.

In late 2007, we set five major goals for 2008. We achieved them all. The way I see it, we put a lot *in* and you got a lot *out*. Let's review.

**1. We planned to highlight *Circuit Cellar*'s 20th anniversary.** We did so by publishing Ed Nisley's retrospective article, "That Was Then, This Is Now" (Issue 210), and introducing a new front cover layout. We also reinstated an old fan favorite, the EQ section. Many thanks to Dave Tweed!

**2. We set out to incorporate articles from the most diverse set of engineers to date.** We achieved this by running stellar articles by talented engineers from such distant places as Dallas, Texas (Ben Smith, Issue 210), Bogota, Colombia (Diego Mendez Chaves, et al, Issue 211), Tokyo, Japan (Yoshiyasu Takefuji, Issue 213), Gorizia, Italy (Carlo Tauraso, Issue 216), North Olmsted, Ohio (James Blackwell, Issue 218), and Pune, India (Mandar Bagul, Issue 221).

**3. We aimed to publish useful articles about renewable energy resources.** This year, we ran five articles on the subject in the Intelligent Energy Solutions section of the magazine. We hope to continue publishing such articles in 2009, but we need your help. It's up to you to start planning the next batch of articles about energy-efficient applications. Send article proposals to cj@circuitcellar.com.

**4. We promised to continue running useful graphics-related articles even though we were removing the Graphics & Video issue from the editorial calendar.** We came through by publishing several graphics-related articles. James Blackwell's article titled "Content Collection and Display" (Issue 218) is an excellent example.

**5. We resolved to challenge readers to work on articles about programmable logic-related topics.** This month, we deliver an article by Kareem Matariyeh about the Verilog language and programming FPGAs and CPLDs (p. 78). More will follow in 2009.

To make this issue memorable, we packed it with articles about real-world applications and handy new design tools. Starting on page 14, John Peterson describes an expandable platform for light and animation. Design, program, and rearrange! Turn to page 24 for Tom Cantrell's useful introduction to Altium's Designer and Nanoboard. To learn about another handy tool, check out Jeff Bachiochi's article "Component Control" (p. 62). On page 32, Mandar Bagul begins an article about an affordable mobile data storage solution. On page 38, Ed Nisley covers the topics of NiMH charging and designing a three-pack charger. If you want to build an electronic identification system, check out Jan Szymanski's description of his MCU-based badge project (p. 44). Robert Lacoste demystifies the topic of proportional integral derivative (PID) control on page 54. On page 69, Martin Courtney introduces two beamforming techniques: adaptive and fixed.

We will run dozens of thought-provoking articles in 2009. You can e-mail me an article proposal or submission at any time. Our calendar is now available at www.circuitcellar.com/submissions/editorialcalendar2009.html.

cj@circuitcellar.com


# CIRCUIT CELLAR®

THE MAGAZINE FOR COMPUTER APPLICATIONS

**FOUNDER/EDITORIAL DIRECTOR**
Steve Ciarcia

**CHIEF FINANCIAL OFFICER**
Jeannette Ciarcia

**MANAGING EDITOR**
C. J. Abate

**MEDIA CONSULTANT**
Dan Rodrigues

**WEST COAST EDITOR**
Tom Cantrell

**CUSTOMER SERVICE**
Debbie Lavoie

**CONTRIBUTING EDITORS**
Jeff Bachiochi
Ingo Cyliax
Robert Lacoste
George Martin
Ed Nisley

**CONTROLLER**
Jeff Yanco

**ART DIRECTOR**
KC Prescott

**GRAPHIC DESIGNERS**
Grace Chen
Carey Penney

**NEW PRODUCTS EDITOR**
John Gorsky

**STAFF ENGINEER**
John Gorsky

**PROJECT EDITORS**
Gary Bodley
Ken Davidson
David Tweed

**ASSOCIATE EDITOR**
Jesse Smolin

## ADVERTISING

860.875.2199 • Fax: 860.871.0411 • www.circuitcellar.com/advertise

**PUBLISHER**
Sean Donnelly
Direct: 860.872.3064, Cell: 860.930.4326, E-mail: sean@circuitcellar.com

**ADVERTISING REPRESENTATIVE**
Shannon Barraclough
Direct: 860.872.3064, E-mail: shannon@circuitcellar.com

**ADVERTISING COORDINATOR**
Valerie Luster
E-mail: val.luster@circuitcellar.com

Cover photography by Chris Rakoczy—Rakoczy Photography
www.rakoczyphoto.com
**PRINTED IN THE UNITED STATES**

## CONTACTS

**SUBSCRIPTIONS**
**Information**: www.circuitcellar.com/subscribe, E-mail: subscribe@circuitcellar.com
**Subscribe:** 800.269.6301, www.circuitcellar.com/subscribe, Circuit Cellar Subscriptions, P.O. Box 5650, Hanover, NH 03755-5650
**Address Changes/Problems:** E-mail: subscribe@circuitcellar.com

**GENERAL INFORMATION**
860.875.2199, Fax: 860.871.0411, E-mail: info@circuitcellar.com
**Editorial Office:** Editor, Circuit Cellar, 4 Park St., Vernon, CT 06066, E-mail: editor@circuitcellar.com
**New Products:** New Products, Circuit Cellar, 4 Park St., Vernon, CT 06066, E-mail: newproducts@circuitcellar.com

**AUTHORIZED REPRINTS INFORMATION**
860.875.2199, E-mail: reprints@circuitcellar.com

**AUTHORS**
Authors' e-mail addresses (when available) are included at the end of each article.

CIRCUIT CELLAR®, THE MAGAZINE FOR COMPUTER APPLICATIONS (ISSN 1528-0608) is published monthly by Circuit Cellar Incorporated, 4 Park Street, Vernon, CT 06066. Periodical rates paid at Vernon, CT and additional offices. **One-year (12 issues) subscription rate USA and possessions $23.95, Canada/Mexico $34.95, all other countries $49.95. Two-year (24 issues) subscription rate USA and possessions $43.95, Canada/Mexico $59.95, all other countries $85.** All subscription orders payable in U.S. funds only via Visa, MasterCard, international postal money order, or check drawn on U.S. bank. **Direct subscription orders and subscription-related questions to Circuit Cellar Subscriptions, P.O. Box 5650, Hanover, NH 03755-5650 or call 800.269.6301.**

**Postmaster:** Send address changes to Circuit Cellar, Circulation Dept., P.O. Box 5650, Hanover, NH 03755-5650.



Circuit Cellar® makes no warranties and assumes no responsibility or liability of any kind for errors in these programs or schematics or for the consequences of any such errors. Furthermore, because of possible variation in the quality and condition of materials and workmanship of reader-assembled projects, Circuit Cellar® disclaims any responsibility for the safe and proper function of reader-assembled projects based upon or from plans, descriptions, or information published by Circuit Cellar®.

The information provided by Circuit Cellar® is for educational purposes. Circuit Cellar® makes no claims or warrants that readers have a right to build things based upon these ideas under patent or other relevant intellectual property law in their jurisdiction, or that readers have a right to construct or operate any of the devices described herein under the relevant patent or other intellectual property law of the reader's jurisdiction. The reader assumes any risk of infringement liability for constructing or operating such devices.

Entire contents copyright © 2008 by Circuit Cellar, Incorporated. All rights reserved. Circuit Cellar is a registered trademark of Circuit Cellar, Inc. Reproduction of this publication in whole or in part without written consent from Circuit Cellar Inc. is prohibited.

# BitScope USB Mixed Signal Oscilloscope

## Analog + Digital

**Digital Storage Oscilloscope**
✓ Dual Channel Digital Scope with industry standard probes or POD connected analog inputs. Fully opto-isolated.

**Mixed Signal Oscilloscope**
✓ Capture and display analog and logic signals together with sophisticated cross-triggers for precise analog/logic timing.

**Multi-Band Spectrum Analyzer**
✓ Display analog waveforms and their spectra simultaneously. Base-band or RF displays with variable bandwidth control.

**Multi-Channel Logic Analyzer**
✓ Eight logic/trigger channels with event capture to 25nS.

**DSP Waveform Generator**
✓ Optional flash programmable DSP based function generator. Operates concurrently with waveform and logic capture.

**Mixed Signal Data Recorder**
✓ Record to disk anything BitScope can capture. Supports on-screen waveform replay and export.

**User Programmable Tools and Drivers**
✓ Use supplied drivers and interfaces to build custom test and measurement and data acquisition solutions.

**Inventing the future requires a lot of test gear...**

**...or a BitScope**




## BS100U Mixed Signal Storage Scope & Analyzer

Innovations in modern electronics engineering are leading the new wave of inventions that promise clean and energy efficient technologies that will change the way we live.

It's a sophisticated world mixing digital logic, complex analog signals and high speed events. To make sense of it all you need to see exactly what's going on in real-time.

BS100U combines analog and digital capture and analysis in one cost effective test and measurement package to give you the tools you need to navigate this exciting new frontier.




Standard 1M/20pF BNC inputs

Smart POD Connector




Opto-isolated USB 2.0

12VDC with low power modes

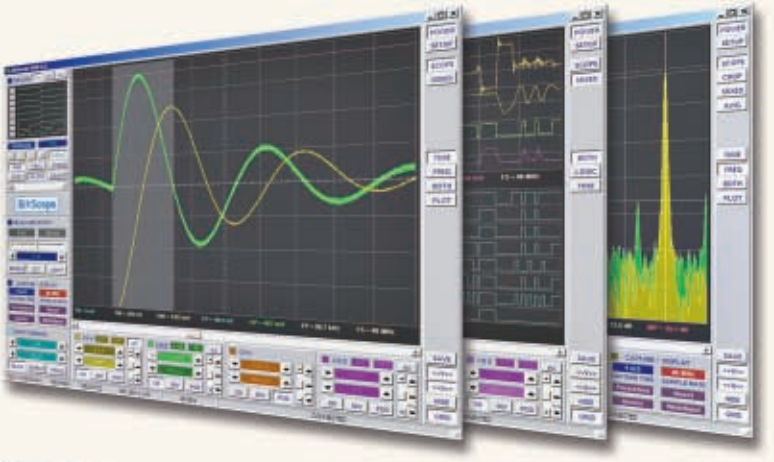
## BitScope DSO Software for Windows and Linux

BS100U includes BitScope DSO the fast and intuitive multichannel test and measurement software for your PC or notebook.

Capture deep buffer one-shots, display waveforms and spectra real-time or capture mixed signal data to disk. Comprehensive integration means you can view analog and logic signals in many different ways all at the click of a button.

The software may also be used stand-alone to share data with colleagues, students or customers.

Waveforms may be exported as portable image files or live captures replayed on another PC as if a BS100U was locally connected.





**www.bitscope.com**

# December 2008: Programmable Logic

## FEATURES

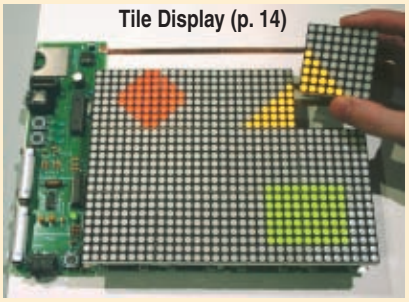

14 **A Dynamic Tile Display**
Board Layout, Real-Time Animation, and More
*John Peterson*

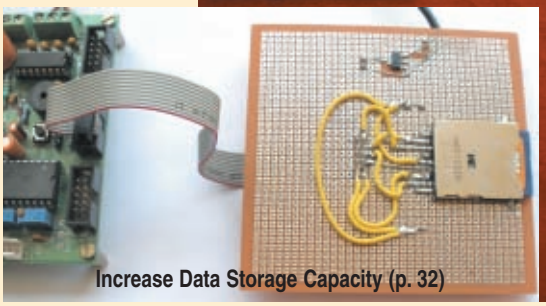
32 **Cost-Effective Mobile Data Storage**
Interface an SD Memory Card with an MCU
*Mandar Bagul*

44 **Electronic ID System**
A Display for Text, Photos, and Slide Shows
*Jan Szymanski*
Subcategory Prize Microchip 2007 Design Contest

69 **Beamforming 101**
*Martin Courtney*

78 **An Introduction to Verilog**
*Kareem Matariyeh*



Tile Display (p. 14)


Increase Data Storage Capacity (p. 32)



Electronic Identification (p. 44)

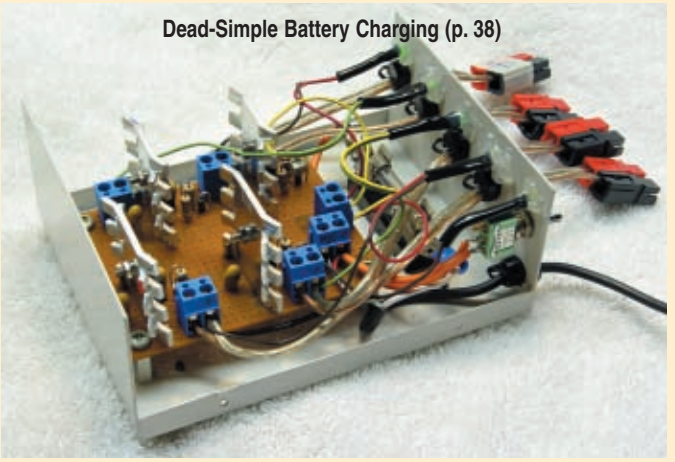
## COLUMNS


24 SILICON UPDATE
**Tool Time**
*Tom Cantrell*

38 ABOVE THE GROUND PLANE
**Real-World NiMH Charging**
*Ed Nisley*

54 THE DARKER SIDE
**PID Control Without Math**
*Robert Lacoste*

62 FROM THE BENCH
**Component Control**
A Tool for Finding and Handling Small Parts
*Jeff Bachiochi*



Dead-Simple Battery Charging (p. 38)

## DEPARTMENTS


4 **TASK MANAGER**
2008 I/O
*C. J. Abate*

8 **NEW PRODUCT NEWS**
edited by *John Gorsky*

93 **CROSSWORD**

94 **INDEX OF ADVERTISERS**
January Preview

96 **PRIORITY INTERRUPT**
My Life and Paper—Take 2
*Steve Ciarcia*

# Hammer Down Your Power Consumption with picoPower™!




# THE Performance Choice of Lowest-Power Microcontrollers

Performance and power consumption have always been key elements in the development of AVR® microcontrollers. Today's increasing use of battery and signal line powered applications makes power consumption criteria more important than ever. To meet the tough requirements of modern microcontrollers, Atmel® has combined more than ten years of low power research and development into picoPower technology.

picoPower enables tinyAVR®, megaAVR® and XMEGA™ microcontrollers to achieve the industry's lowest power consumption. Why be satisfied with microamps when you can have nanoamps? With Atmel MCUs today's embedded designers get systems using a mere 650 nA running a real-time clock (RTC) and only 100 nA in sleep mode. Combined with several other innovative techniques, picoPower microcontrollers help you reduce your applications power consumption without compromising system performance!

Visit our website to learn how picoPower can help you hammer down the power consumption of your next designs. PLUS, get a chance to apply for **a free AVR design kit!**

http://www.atmel.com/picopower/

Everywhere You Are®

© 2008 Atmel Corporation. All rights reserved. Atmel®, logo and Everywhere You Are® are registered trademarks of Atmel Corporation or its subsidiaries.
Other terms and product names may be trademarks of others.

# NEW PRODUCT NEWS

Edited by John Gorsky

Visit www.circuitcellar.com/npn for more New Product News.

## KickStart KIT FOR NXP LPC2478

The **KickStart Kit for NXP LPC2478** is a complete development kit that provides a design platform with a touchscreen display for multiinterface communication applications featuring Ethernet and USB device and host interfaces. The new development kit is ideal for industrial, medical, point-of-sale, and other connected applications. It contains all the necessary hardware and software to enable you to design, develop, integrate, and test new applications.




In addition to an LPC2478 development board and a 32-KB limited edition of Embedded Workbench for ARM, the kit also contains evaluation editions of visualSTATE, PowerPac for ARM, PowerPac USB, and PowerPac TCP/IP. A J-Link for ARM is included in the kit, along with example applications for the LPC2478 board as well as board support packages for both IAR PowerPac and Micrium RTOSes. The kit includes the LPC2478 device, a 3.5″ 320 × 200 24-bit color TFT LCD with a backlight, and a touchscreen. It also includes a wide selection of peripherals: an MP3 decoder DSP and VS1002D codec, a three-axis digital accelerometer with 11-bit accuracy, 64-MB SDRAM, a USB host and USB device connector, an IrDA transceiver, a PS/2 keyboard connector, a 100-Mbps Ethernet port, a CAN driver and connector, an RS-232 interface with ICSP control, an SD/MMC card connector, a JTAG connector, a MICTOR trace connector, a Reset button, two user buttons, a trimming potentiometer, a UEXT connector, audio input and output, and an RTC battery holder.

**IAR Systems**
**www.iar.com**

## LOW-COST LatticeECP2M DEVELOPMENT KIT

The new **LatticeECP2M PCI Express development kit** is the first in a series of development kits that will support Lattice devices and enable you to obtain working hardware in 30 min. and then a known good starting point for a design in less than 2 h. There are several demos available, including a control plane application, and a throughput demo for high-bandwidth applications. Two demos support scatter-gather DMA capabilities: a color bar demo and an image transfer demo, which also uses the Scatter-Gather DMA IP core.

Based on a new PCI Express ×1/×4 evaluation board, the kit accelerates development of PCI Express designs using the LatticeECP2M family. All of the demos use the LatticeECP2M PCI Express ×1/×4 Endpoint ispLeverCORE IP Ver. 3.3. The included low-cost LatticeECP2M PCI Express ×1/×4 evaluation board has been optimized with several features designed to make the evaluation process more productive, including support of both ×1 and ×4 link edge card fingers and LEDs for each link that are visible from outside the PC chassis to indicate whether the link is active. Also included are a 60-day evaluation license for ispLEVER software design tools (for new users), and the ability to evaluate the new LatticeECP2M PCI Express ispLeverCORE IP and project source directories, all of which can be used to create a template for design exploration.




The development kit costs **$895**.

**Lattice Semiconductor Corp.**
**www.latticesemi.com**

# NEW PRODUCT NEWS

## TRANSCODER ENABLES BIDIRECTIONAL REMOTE CONTROL

The **MT Series Transcoder IC** provides unprecedented features for bidirectional remote control and command applications. In the past, encoder and decoder ICs were commonly used to allow the status of inputs, such as switch or button closures, to be sent in one direction across a wireless link. While the MT can perform these basic unidirectional functions with ease, it also allows the bidirectional exchange of information and reception confirmation across an external wireless link, such as a Linx transceiver module. When combined with external feedback, the Transcoder can also verify that specified actions have taken place. The MT is even capable of sending variable data such as time, temperature, pressure, or text along with control information.




The transcoder provides eight lines, which can be configured as inputs or outputs and individually set for latched or momentary operation. A large 24-bit address size makes transmissions highly unique, minimizing the possibility of conflict between multiple devices. An automatic confirmation indicates that the transmission was successfully received. The MT also outputs the originating transcoder's ID for logging or identification.

In addition to these powerful features, a serial interface engine (SIE) allows the transcoder to be configured and controlled from an external microcontroller or PC. Permissions and users can be configured, modes and functions set, and external data added to the transmission stream. A targeting feature allows commands to be directed to specific devices.

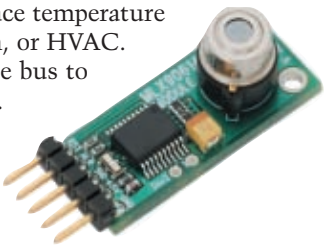
MT Transcoder ICs cost **$2.37** in 10,000-piece quantities.

**Linx Technologies, Inc.**
**www.linxtechnologies.com**

## INFRARED THERMOMETER MODULE

The **MLX90614** infrared thermometer module is an intelligent noncontact temperature sensor with a single-pin serial interface for connection to most microcontrollers. The MLX90614 sensor is designed for the noncontact temperature measurement of objects placed within the sensor's cone of detection. The sensor comprises an integrated ASIC and an infrared-sensitive thermopile detector.

The sensor communicates with an SX20AC/SS-G coprocessor over a digital SMBus, which Parallax has programmed to simplify an otherwise fairly complex communication protocol. With a temperature range of –70° to 380°C, auto-baud detection, and a programmable alarm setting, this module becomes useful in many applications such as surface temperature measurement, human/animal presence detection, or HVAC. Up to 100 modules can be connected on the same bus to make multizone temperature measurement easy.



The MLX90614 costs **$49.99**.

**Parallax, Inc.**
**www.parallax.com**


SEGGER

Software Trials Available

JTAG (J-Link™)

+++ ARM7/9 Cortex M3 +++

- USB to JTAG
- Fast 720kb/s Download Speed
- Serial Wire Debug (SWD) Support
- Multicore Debugging Support
- Auto JTAG Speed Recognition

The J-Link can be coupled with a number of available software modules to fit your application needs.

**J-Flash** is a stand-alone application used with the J-Link to program internal and external flash devices.

**J-Link RDI** permits the use of the J-Link with an RDI compliant debugger.

**J-Link GDB Server** is a remote server for GDB.

**J-Link Flash Download** is a module used to download your program into flash even if your debugger does not have a flash loader.

**J-Link Flash Breakpoint** permits you to set an unlimited number of software breakpoints while debugging in flash.

**J-Link SDK** is a standard DLL that extends the full functionality of the J-Link to your proprietary application.



We also offer a JTAG isolator which can be used to offer electrical isolation between your target hardware and the J-Link. This is essential when the development tools are not connected to the same ground as the application. It is also useful to protect the development tools from electrical spikes that often occur in some applications, such as motor control applications.

- JTAG Isolator



Special Offer

Includes the J-Link, J-Link GDB Server, and the J-Link Flash Download.
www.segger-us.com/ncu.html

J-Link Non-Commercial (NCU) Bundle

www.segger.com

# NEW PRODUCT NEWS




## NEW LOW-POWER, FLOATING-POINT PROCESSORS

Three new devices merge the inherent advantages of floating-point with a combination of connectivity peripherals, low power, and low cost that until now, has been available only with fixed-point devices. The new processors include the industry's lowest power floating-point DSPs—the **TMS320C6745** and **TMS320C6747**—as well as the **OMAP-L137** floating-point-DSP-plus-ARM applications processor.

The C6745, which runs at up to 300 MHz, includes an extensive set of serial ports for system control as well as multichannel audio serial ports (McASP) with up to 16 serializers and FIFO buffers. For additional system performance, the C6747 includes all of the features of the C6745 DSP and incorporates an additional 128 KB of on-chip RAM. The EMIFA and EMIFB on the C6747 are also upgraded to 16-bit and 32-bit, respectively.

An LCD controller is on-chip, allowing designers to quickly add QVGA or other displays. The OMAP-L137 applications processor includes the C674x floating-point DSP core plus an ARM9, providing up to 300 MHz per core.

Using half of the standby power and less than one-third of the total power of previous-generation floating-point DSPs, the C6745 and C6747 are the industry's lowest power floating-point DSPs. Both processors consume 62 mW in standby mode and 470 mW in total active power mode

You can now place an order for the C6745 DSP, C6747 DSP, and OMAP-L137 applications processor, which start at **$10.41**, **$11.72,** and **$18.93** (100-unit pricing), respectively. The C6745 DSP, C6747 DSP, and OMAP-L137 applications processor are supported by the OMAP-L137/C6747 Floating-point Starter Kit (**$395**), which offers the flexibility of Linux and DSP/BIOS kernels.

**Texas Instruments, Inc.**
**www.ti.com**



# NEW PRODUCT NEWS

## MIXED-SIGNAL MOTOR DRIVERS

The **AMIS-30511** and **AMIS-30512** bipolar stepper motor controllers eliminate the need for many external components, reducing system cost and the bill of materials (BOM). Both devices offer SPI ports and are capable of peak currents up to 800 mA.

These new mixed-signal ICs contain two embedded H-bridges that can drive two-phase stepper motors with sustained peak currents up to 400 mA and maximum currents up to 800 mA for short duration. Each also allows a host microcontroller to detect blocked rotor and end-of-run conditions without additional switches, Hall sensors, or optical encoders. In addition to stall detection, these functions allow you to program the microcontroller to calculate rotor position and dynamically adjust current or speed, as required, to prevent step loss. This capability eliminates noise and vibration at end-stops and improves motion-control accuracy and reliability without additional components.

The 30512 contains a 5-V regulator to supply both internal logic and up to 50 mA for external components, such as a microprocessor. The 30512 also includes watchdog logic circuitry. Typical uses for the 30511/512 include security cameras, building and industrial automation, motor control, textile processing, process equipment, robotics, HVAC and base station control, and status.



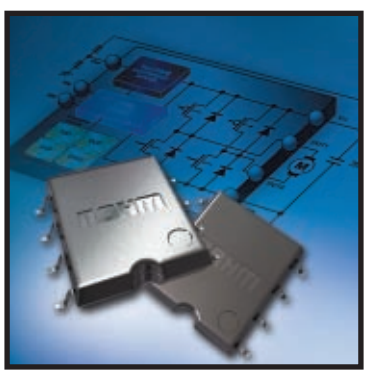

Available in a 24-pin SOIC package, the AMIS-30511 starts at **$3.81** and the AMIS-30512 is **$4.18** per unit in quantities of 1,000.

**ON Semiconductor**
**www.onsemi.com**

## LINE OF PWM OUTPUT H-BRIDGE DRIVER ICs

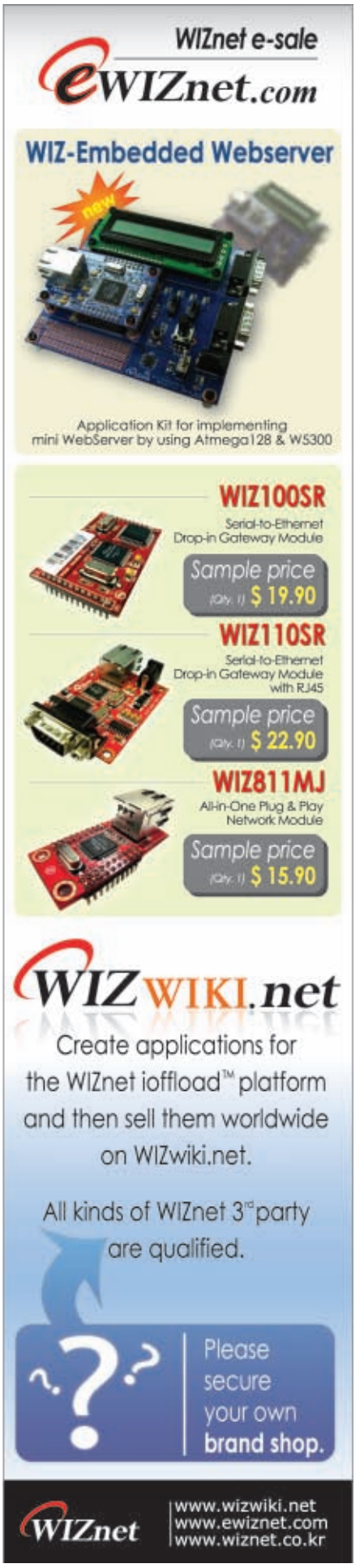
A new line of H-bridge driver ICs offers designers a simple upgrade path to PWM speed control using either analog or digital inputs. The new **BD62*xx* Series** provides both a TTL-compatible interface to an external MCU as well as an internal $V_{REF}$-PWM conversion circuit, allowing implementation of energy-saving PWM control from an analog voltage input. The product line is offered in three maximum operating voltage classes (7, 18, and 36 V) with current ratings in each class of 0.5, 1, and 2 A to cover applications ranging from portable electronics to automotive, appliance, and medical devices. Internal shut-down circuitry reduces standby current to less than 1 µA—particularly important in portable designs.

All BD62*xx* H-bridge driver ICs include a complete suite of internal protection circuits including over-voltage protection (OVP), under-voltage lock-out (UVLO), over-current protection (OCP), and high-temperature shutdown (TSD). Overlap (shoot-through) protection to prevent high currents during motor reversal or breaking is also built in. In addition, the devices are designed to withstand 4 kV ESD.



All products in the series come in low-profile surface-mount packages with a maximum height ranging from 1.15 mm (SSOP) to 2.2 mm (HRP7). All devices are specified for –40° to 85°C operation. The ICs costs from **$0.80** (one channel, 7 V) to **$2.60** (two channels, 36 V) in OEM quantities.

**ROHM Electronics**
**www.rohmelectronics.com**

# NEW PRODUCT NEWS

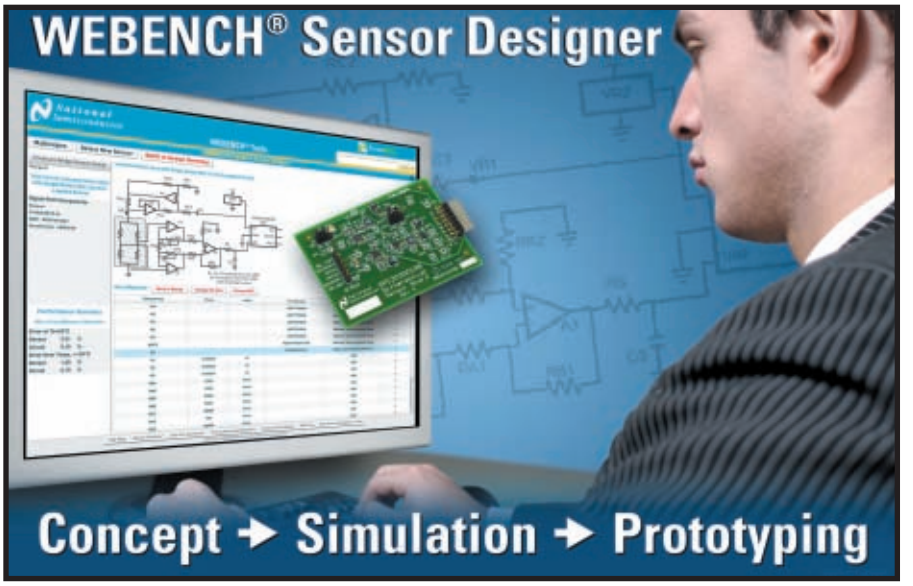
## ONLINE TOOL FOR DEVELOPING SENSOR SIGNAL PATH SOLUTIONS

The **WEBENCH Sensor Designer** tool significantly reduces design time and cost by configuring a complete sensor signal path solution with just a few keystrokes. Prior to this design tool, you would have to waste valuable development time matching sensor characteristics and performance with an op-amp's DC performance and an ADC's resolution and accuracy. Additional time would then be required for circuit evaluation and prototyping.

The WEBENCH Sensor Designer tool can combine the PowerWise LMP7731 precision amplifier with the new ADC161S626 16-bit, 250-ksps, differential input ADC. The LMP7731 features a low-input voltage noise of 2.9 nV/sqrt Hz with a 1/f noise corner of only 3 Hz, while the ADC161S626 provides signal accuracy of ±0.003% over the –40° to 85°C temperature range. Each signal path solution amplifies conditions and digitizes the sensor output to directly link to a microcontroller or DSP. The tool also allows you to order a customized evaluation kit for prototyping your simulated WEBENCH solution in the lab.

Using the "Build It!" option, you can generate solutions that you can use for evaluation and prototyping. Customized prototype kits can be shipped within 24 h to accelerate test and validation. For engineers that would like to interface directly to a PC, a separate communications evaluation board and software is available. To begin a design with the WEBENCH Sensor Designer tool, go to www.national.com/sensors.

**National Semiconductor Corp.**
**www.national.com**



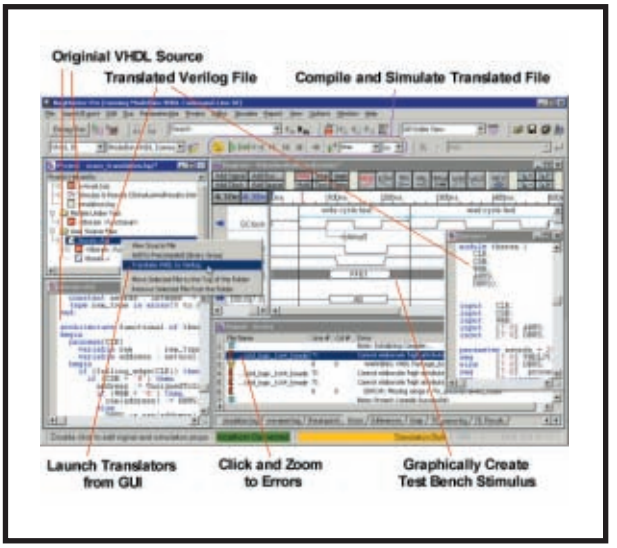

## TRANSLATORS GET VERILOG 2001 SUPPORT

A new version of **V2V tools** for translating between VHDL and Verilog source code that supports Verilog 2001 code constructs is now available. In addition, **BugHunter Pro** can now be used as a graphical debugging environment for translating and testing the new models. The tool is also able to translate code automatically in several cases that previously required manual preediting for a successful translation.

The typical translation design flow involves an iterative cycle where you must make some manual changes to either the original or the generated code. To speed up this flow, an optional graphical debugging front-end based on the BugHunter Pro HDL Debugger tool has been added. With BugHunter, you can launch the translators via point-and-click mouse operations, set options to the translators via a dialog window, and double clicking on errors and warnings generated during translation to launch a text editor and make changes at the problematic lines in the source code.

BugHunter can also launch VHDL and Verilog simulators on both the original and the translated code, dramatically speeding up the process of verifying and modifying the translated code. BugHunter also supports full single-step debugging, waveform capture, and variable inspection with the following simulators: SynaptiCAD's VeriLogger Extreme, Cadence Design Systems's Incisive, Synopsys's VCS, Mentor Graphics's ModelSim, and Aldec's ActiveHDL.

The V2V translator software is available on Windows and Linux. The software can be licensed on either a permanent or leased basis, and both floating and node-locked versions are available. Contact SynaptiCAD for more information and pricing.




**SynaptiCAD, Inc.**
**www.syncad.com**

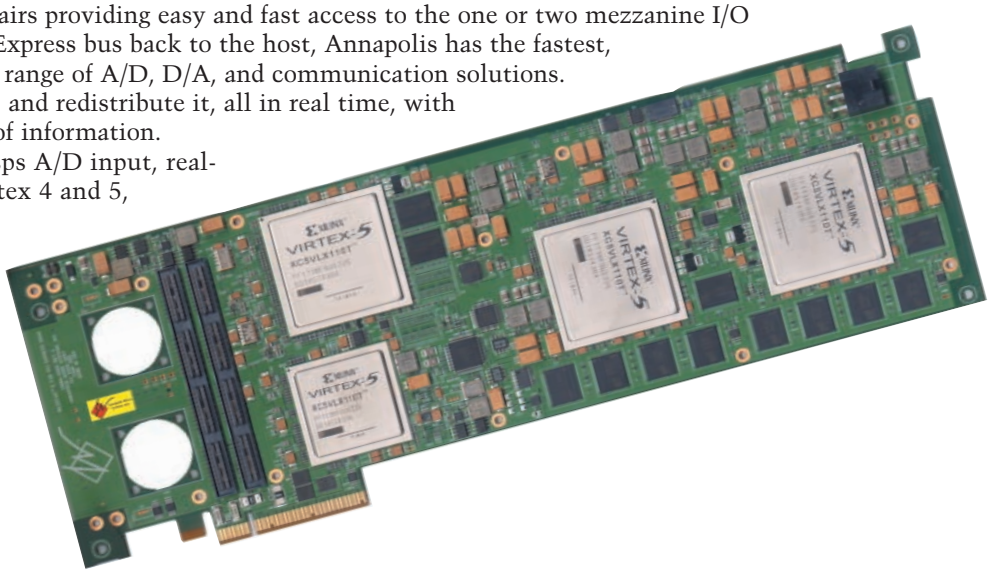
# NEW PRODUCT NEWS

## PCI EXPRESS FPGA-BASED PROCESSING BOARDS

The **WILDSTAR 5** for PCI-E FPGA-based processing boards use Xilinx's newest, biggest, and highest pin count Virtex 5 FPGAs. Now available with the XC5VLX110T, XC5VLX155T, XC5VLX220T, XC5VLX330T, and soon-to-be-delivering XC5VFX100T, XC5VFX130T, XC5VFX200T, or XC5SX240T, and with a choice of DDRII DRAM, DDRII SRAM, QDRII SRAM, SIO RLDRAM, or CIO RLDRAM. The WILDSTAR 5 provides a proven high-performance platform for the most difficult parallel processing algorithms, for military, research, and commercial applications. But it takes more than powerful FPGAs to build a successful SIGINT, DSP, or radar system. With up to 16 Rocket I/O lanes and up to 96 differential pairs providing easy and fast access to the one or two mezzanine I/O cards and DMA across the PCI Express bus back to the host, Annapolis has the fastest, widest, and most encompassing range of A/D, D/A, and communication solutions. Take in data, process it, store it, and redistribute it, all in real time, with no degradation or compression of information.

The boards offer up to 2.5-Gsps A/D input, real-time processing on Xilinx's Virtex 4 and 5, and data flowing to storage RAIDs or disks with up to 36-Gb full duplex I/O. These types of options enable you to choose the exact components needed for your particular programs.

The WILDSTAR 5 starts at **$22,529**.



**Annapolis Micro Systems, Inc.**
**www.annapmicro.com**







# A Dynamic Tile Display

## Board Layout, Real-Time Animation, and More

John designed an expandable platform for light and animation. You can use it to create modular animated signs, animated puzzles, and more. The system features a tray and tiles, which are based on 8 × 8 LED displays. Once the tiles are synchronized, you can pick them up and rearrange them.

Every time you turn on a computer screen or look at an animated sign, millions of pixels disappear into your work, your correspondence, or your entertainment. Imagine if you could capture those dots of light and scoop them up like grains of sand on a beach.

Finding some large, bright LED matrix displays inspired me to explore this concept further. I created an animated display where each tile is independent. I call it the Puzzlemation system. You can pick it up and move it around. Bring the tiles to life with animated patterns, and you have a fascinating puzzle where the pieces you're trying to put together are in motion. You can also create an animated display sign of various shapes and proportions.

Tackling this project brought up some intriguing design challenges. If each tile is an independent piece, how do you power, control, and synchronize them? How do you create and store the animation? As you'll see, this project exposed me to a broad range of technical issues, ranging from TCP/IP Ethernet communication, to real-time animation, to compact circuit board layout.

### SYSTEM OVERVIEW

The system has two main components, the tray and the tiles (see Photo 1). It is based on Microchip Technology PIC24FJ64 microcontrollers. One runs the tray and one controls each tile.

The project's display is made of a number of tiles, each about 2″ square with an 8 × 8 array of red/green LED pixels. Each tile is individually powered and animated. You can pick up the tiles to rearrange them. To set up a display, the tiles are placed in a special tray. The tray has copper strips, which carry a serial I/O signal across the bottom. Each tile rests on four spring-loaded contacts that make contact with the strips to receive data from the tray (see Figure 1).

Animations are downloaded into the tray via Ethernet and locally stored, or loaded via an SD card. The tray broadcasts animation to each of the tiles and then synchronizes them. Once the animation is synchronized and running, you can pick up the tiles and place them anywhere. The display is completely reconfigurable. Need a tall, thin display? Need a square one? Or how about a long skinny one? No problem. Rearrange the tiles as you please.

If the pieces are left in the tray, the

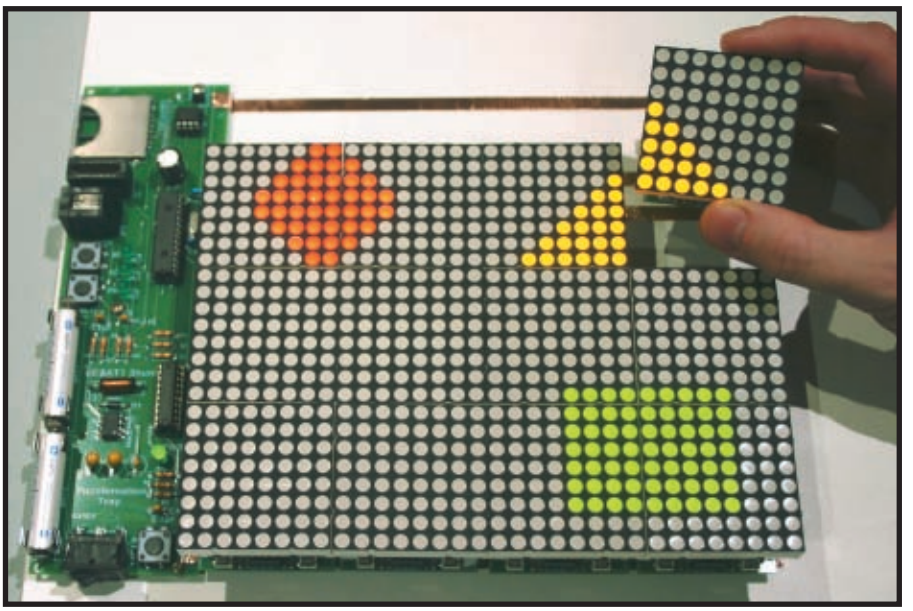


**Photo 1—**This is the Puzzlemation system. Each tile is individually powered and animated. The tray electronics on the left download data into the tiles.

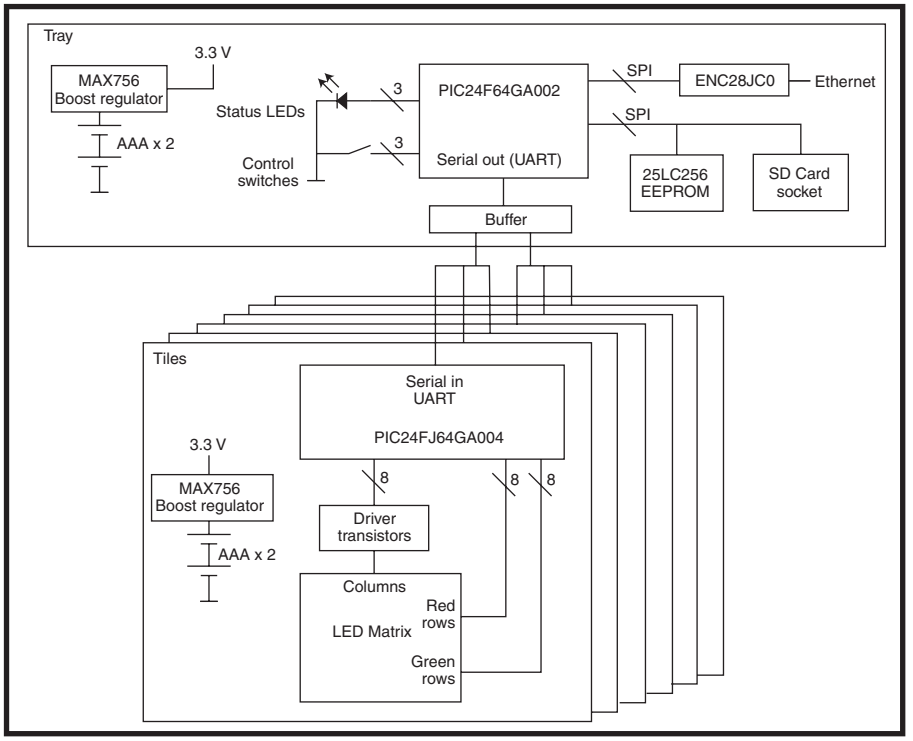



**Figure 1—**Each tile contains a microcontroller and power to drive the display. The tray receives animations from an Ethernet or SD card and broadcasts it to the tiles via serial I/O.

animation can be updated over the Ethernet connection. The tiles display the animation for several hours with their own rechargeable battery power.

The ability to rearrange the tiles also offers an interesting challenge. If you scramble the tiles before loading the animation, a unique puzzle emerges. You must then rearrange the pieces of an animated image. (This is where the name Puzzlemation comes from.) Of course, the tray can ask the pieces to show their ID numbers so you can put them back in order again.

## TILE HARDWARE

The tiles are based on 2″ square 8 × 8 LED matrix displays. As you can see in Figure 2, individual connections for the red and green LEDs are located along the rows of the matrix, for a total of 24 connections (eight columns, eight red rows, and eight green rows).

A PIC24FJ64GA004 microcontroller drives the LEDs, stores the animations, and processes the serial data (at 9,600 bps) broadcast from the tray. The microcontroller has 35 I/O pins, 24 of which are used to drive the display. Because the microcontroller can't quite sink enough current if all of the LEDs in a column are on, the column lines are connected through a ULN2803AFW Darlington transistor array to provide the necessary current drive.

Because of the matrix configuration, one column of the display is active at a time. The microcontroller scans the columns at about 1,000 Hz to provide the appearance of a steady display. The LEDs have a one-eighth duty cycle, so current-limiting resistors are not needed. The average current received by the LEDs is well below their rated maximum.

Power for each tile is supplied by two AAA batteries and fed through a Maxim Integrated Products MAX756 boost regulator to get the 3.3 V required by the microcontroller. The rechargeable NiMH AAA batteries have about 800 mAh. Each tile draws about 100 to 280 mA, depending on the number of illuminated LEDs. The boost regulator is approximately 80% efficient, giving an overall battery life of around 6 h.

Because accurate timing is required to keep the animations in sync, an external crystal oscillator is provided for each microcontroller. This is significantly

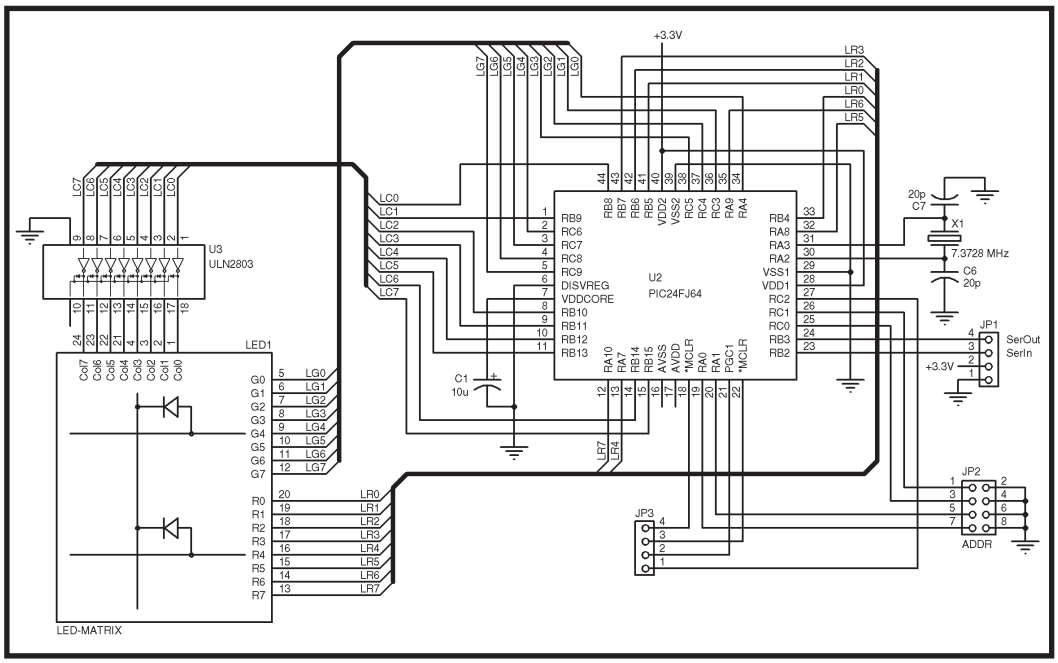



**Figure 2—**The tile is implemented on two PCBs. This one holds the LED matrix display, the PIC24FJ64GA004 microcontroller, and a ULN2803AFW transistor to sink the current for the common cathodes of the LED matrix. A crystal enables accurate timing.

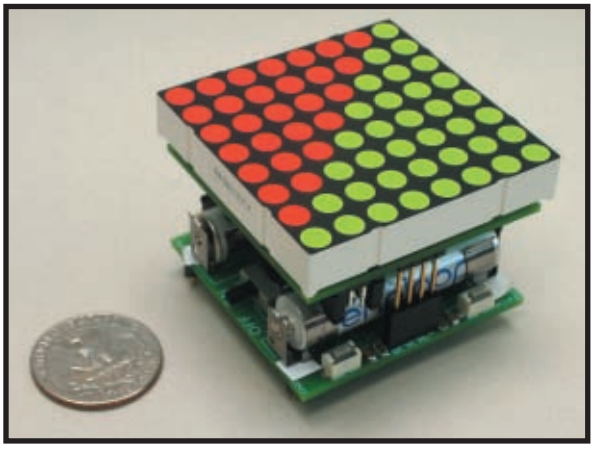


**Photo 2—**This is the tile assembly. Spring-loaded contacts on the bottom of the tile assembly (on the left side) make contact with the tray to receive serial data.

more precise than the on-chip oscillator. It uses the same 7.3728-MHz frequency as the Microchip 28-pin demonstration board.

The circuitry and power source for the tiles must fit within the 2″ × 2″ footprint of the LED matrix display in order for the tiles to fit snugly together (see Photo 2). To accomplish this, the components are split across two stacked circuit boards connected by eight pins.

The top board has the LED matrix display, microcontroller, driver transistor array, and crystal. To fit within the display's footprint, surface-mount components were used. I used Kester R276 solder paste and an old toaster oven to reflow the solder for the SMD components. This was much easier than trying to solder the tiny components by hand (see Photo 3).

A set of four "address" jumpers sets the unique ID for each tile. They allow the serial data protocol to address a particular tile. Using hardware jumpers enables the tiles to use identical firmware.

The lower board has the boost regulator and associated electronics, batteries, and on/off and reset switches. Underneath this board are four spring-loaded "pogo" connectors for making contact with the tray. Only two are currently used: serial input and a ground connection. The other two allow for serial output (e.g., for debugging or interactive games) and an external power supply. The AAA batteries just fit within the 2″ width of the tile, so low-profile battery clips were used to stay within bounds. Having the batteries right over the tray contact pins helps them make good contact and keeps the entire tile assembly stable. The eight connecting pins carry the power, serial data, reset, and in-circuit programming signals from the bottom board to the top. They're also sturdy enough to provide mechanical support for the top board (see Figure 3).

A low-profile connector is provided for hooking the tile up to the Microchip MPLAB ICD2 debugger/programmer, so the PIC24FJ64 is programmed in place. A separate adapter board was constructed to convert the Microchip ICD 2's RJ-45 connector to this one.

## TILE SOFTWARE

The firmware for the tiles is completely interrupt driven. All software functions are accessed via interrupt service routines (ISRs). Two timers control the display. The first (Timer 1 at about 1,000 Hz) scans the columns of the display in its ISR. The second (Timer 2) switches from frame to frame at a rate controlled by the animation downloaded into the tile.

Animation frames are stored in the tile as 8 bytes of red pixel values (1 byte for each row of eight pixels) followed by 8 bytes of green pixel values. Because the LED matrix has common cathode columns, pixels are displayed by columns, rather than rows. A simple routine rotates each animation frame 90° before it's displayed. Each frame also stores a 16-bit time value that controls how long the frame is displayed. Having a separate time value for each frame is useful in sign display applications, where a message is displayed for several seconds, followed by a brief transition, then another message, and so on. A frame time of zero shuts off the frame timer for a static display.

A third ISR executes when a character is received via the UART's serial input. The tiles respond to a simple command language, with the first



# Monitoring Applications Just Got Easier




## Smarter Embedded Design

The RCM4300 RabbitCore® module gives you more processing power and memory for your demanding embedded applications that require web interfaces, graphics or encryption. Data logging applications will also take advantage of the miniSD™ memory cards with up to 1GB of storage. It's easy to add the RCM4300 – or any RabbitCore module – to your pin-compatible design.



From **$80** qty. 1000

### Superior Performance and Mass Storage for:

- Web interfaces and graphics
- Data logging
- Security and encryption

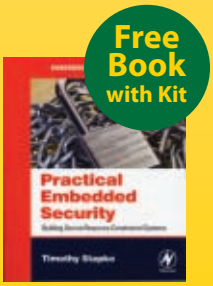
### RCM4300 RabbitCore®

- **Rabbit® 4000 @ 58.98 MHz**
- **10/100Base-T Ethernet**
- **Up to 1GB of miniSD™**
- **8 channel analog input, 12-bit**

### Get Your Development Kit Today for $259

Development kits include everything to start your development, Dynamic C® software, RCM4300 RabbitCore module, 512MB miniSD™ card, development board, programming cables and accessories.

**www.rabbitkits.com**




RABBIT™

2900 Spafford Street, Davis, CA 95618 Tel **530.757.8400**

08081

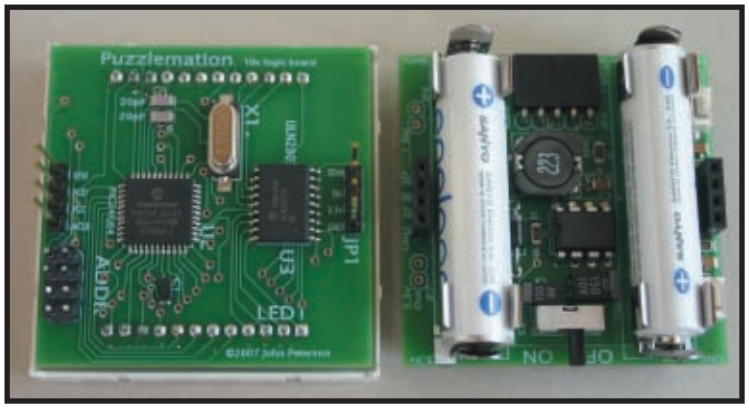
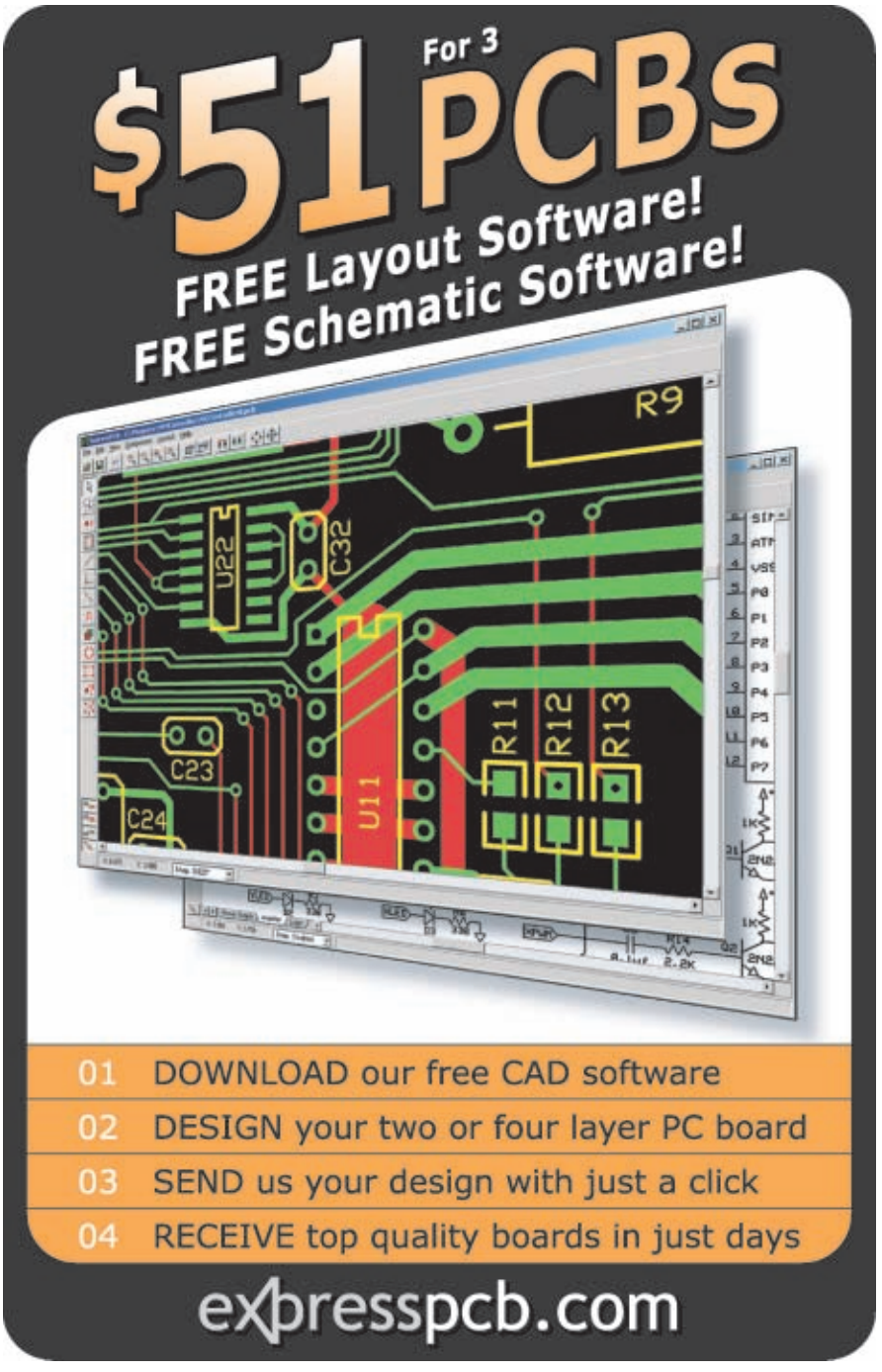



**Photo 3**—These are prototype tile boards. The top board (left) includes a microcontroller, crystal, and display driver. It connects via header pins to the power board (right) containing batteries, a Maxim Integrated Products MAX756 power converter, switches, ICD connect, and spring-loaded contacts (underneath).




character specifying the command (see Table 1).

A simple software state machine parses the commands with multiple parameters. The serial data from the tray is broadcast to all of the tiles simultaneously. Some of the commands, like those for starting the animation or displaying the tile's ID, apply to all of the tiles. Other commands specify a tile ID and apply only to a particular tile. In this case, all of the tiles must parse the command data with their state machines, but only the tile with the ID matching the one specified in the command actually stores or displays the data.

When downloading animation frames, the tile displays a spinning "pinwheel" animation while it's receiving data. If the pinwheel is red, that tile is the one currently receiving the data; otherwise, it's green. The tiles can display text using an 8 × 8 character set designed by John Reeves Hall. This is useful for sign display applications. It also provides a simple way for the tray to display diagnostic messages.

## THE TRAY

The tray takes data from the outside world, broadcasts it to the tiles, and synchronizes them. It also provides a basic user interface for controlling the tiles. It is based on a PIC24FJ64GA002 microcontroller, the 28-pin cousin of the microcontroller used in the tiles. One of the microcontroller's SPI ports connects to a Microchip Technology ENC28J60 Ethernet controller. The other is connected to either an SD card socket or Microchip 25LC256 serial EEPROM. On-board power is supplied by two AAA batteries boosted by a MAX756 voltage regulator, similar to the design of the tiles (see Figure 4). A few push buttons and LEDs provide a user interface and status feedback. The serial output of the tray's microcontroller is buffered by a 74LS244 and then connected to the copper foil strips that carry the serial data to the tiles (see Photo 4).

I used a Microchip PICtail+ Ethernet module that was designed for one of Microchip's design kits for the

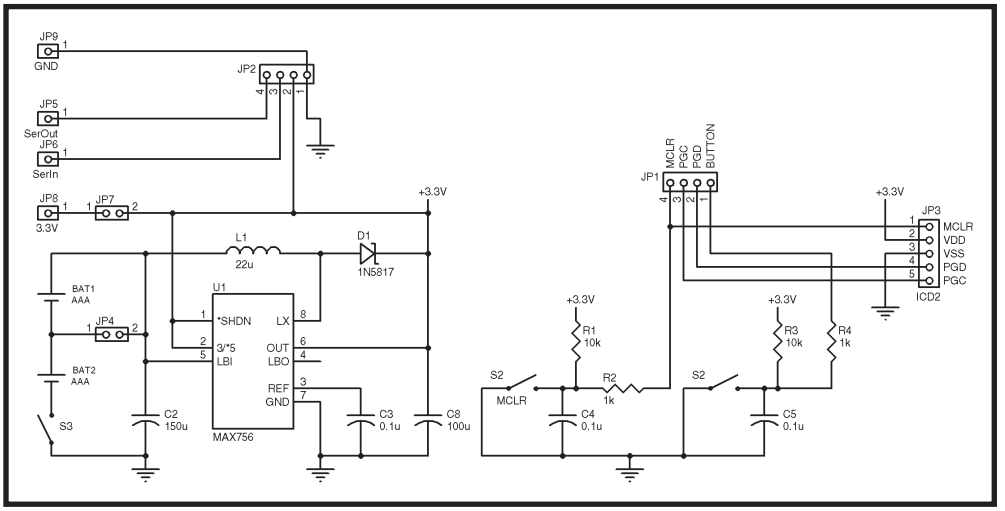



**Figure 3—**The second tile PCB holds the MAX756-based boost regulator, the batteries, control switches, and the ICD2 programming connector. Two four-pin headers connect it to the other PCB. Spring-loaded pins on the bottom connect to the serial data lines on the "tray."

ENC28J60 Ethernet interface. It requires an expensive connector and isn't mechanically stable. A better choice would have been an ENC28J60-H distributed by Olimex. Unfortunately, I couldn't locate one in stock at the time I was working out the design.

Having the Ethernet interface on a removable card has another benefit: battery life. The ENC28J60 runs rather hot, and the module consumes about 210 mA, enough to put a serious dent in the tray's battery life. With the module removed, the tray's battery life exceeds that of the tiles.

The SD card was far more useful than the 25LC256 EEPROM for storing animations. The EEPROM has limited storage (only 32 KB) and less software support than the SD card. A 128-MB SD card has more than enough space for Puzzlemation's needs and routinely sells for about $5 on the 'Net. That's less than the cost of the connector for it!

## TRAY SOFTWARE

You might think managing TCP/IP sockets and a PC-compatible file system requires a 32-bit RISC chip with a megabyte of memory running Linux. But thanks to some clever software libraries from Microchip, a $3 PIC24FJ64 handles all of this, as well as communicating with the tiles, in less than 64 KB of program memory. (Refer to the TCP/IP Sockets sidebar.)

Using the Ethernet connection, animations are downloaded over the network and stored as files on the SD card. You can start, stop, and erase animations with commands over the network. Sending simple text messages for display on the tiles is also supported. You can use a simple adapter to copy animations from your PC to the SD card. The PC card is formatted with the FAT32 file system, which all PCs (Windows, Mac, and Linux) operate with.

The TCP/IP interface is based on Microchip's TCP/IP stack library (available as C source code on Microchip's web site). This library provides a full range of TCP services, including a web server (HTTP), limited file transfer (FTP), e-mail (SMTP), and more. For Puzzlemation, only a couple of these high-level services are used. The first, DHCP, lets the TCP stack auto-configure itself by querying the network's gateway for address information. The second, ICMP, responds to echo requests. This is a useful debugging tool because you can ensure the networking software is up and running by issuing a "ping" command to the tray's IP address from a PC and see if you get an echo packet back.

To communicate with a PC host, the system sets up two servers, each listening on its own TCP port. One server listens for commands to display text messages on the tiles. Short text messages (limited in length to the number of tiles) sent to this port are displayed on the tiles. The second server listens for commands to download and manage animations. The animation port interprets the commands listed in Table 2 for downloading, displaying, and erasing animations from the SD card.

Unlike the tile software, which is completely interrupt driven, the tray's main loop calls three routines. The first two, `DisplayTextServer` and `LoadAnimServer`, process requests on the Animation and Text display ports. A third routine, `StackTask`, is defined by the TCP/IP library and takes care of all the processing required by the TCP/IP software. It's important to add calls to `StackTask` within the loops of the server routines, because that's where the TCP library does the actual data transfer.

The files on the SD card are managed with Microchip's MDD File System library, which is also available in source code form from Microchip's web site. It implements a PC-compatible FAT file system on a SPI-accessible device like an SD card. From a programming standpoint it looks a lot like the standard I/O library familiar to any C programmer.

The key to using these libraries, and getting them to fit in 64 KB of space, is the compile-time switches Microchip

| Command | Parameters | Description |
|---|---|---|
| D | <N><T><frame data> | Download N frames to tile T |
| C | <T><char> | Display character char on tile T |
| c | <T><color><char> | Display color character char on T |
| S | None | Start animation |
| R | None | Reset |
| F | None | Freeze (stop) animation |
| U | None | Unfreeze (continue) |
| T | <8 bytes of pixels> | Test (load green pixels, all tiles) |
| t | <T><8 bytes of pixels> | Load green pixels in tile T |
| V | None | Display test pattern |
| I | None | Display tile's ID number |

**Table 1—**This table shows the tile serial protocol.

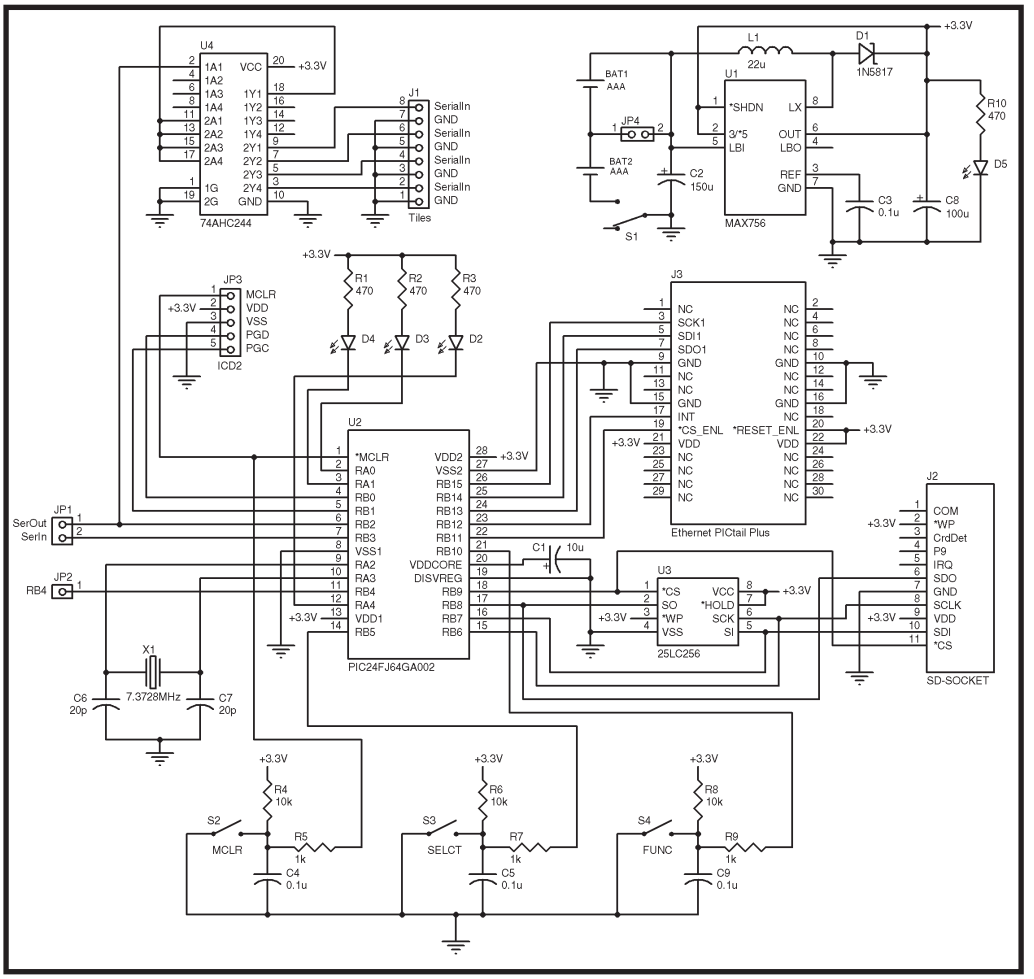



**Figure 4—**The tray interfaces the tiles to the outside world via Ethernet (implemented on a PICtail card) and provides an SD card slot for loading animations. It broadcasts animation to the tiles via a serial line that's buffered through the 74HC244. It runs on battery power using the same MAX756-based boost regulator used by the tiles.

provides to disable portions of the code that aren't used. For example, with the TCP/IP library, I enabled only the DHCP feature and ICMP services. The protocol in Table 2 was implemented with lower-level socket code. Similarly, I disabled hierarchical directory access and disk formatting in the FAT library, because I wasn't using those functions. The two libraries, along with the rest of the tray code, consume about 50 KB, or 90% of the PIC24FJ64's available code space.

There are two buttons on the tray to control the system. The first button cycles among the animation files on the SD card, displaying their names on the tiles. It also shows some diagnostics, such as displaying the individual tile ID numbers and the IP address of the tray. Pressing the other button loads that animation and starts it. Of course, it's helpful to have the tiles arranged in order when you're trying to read the filenames.

## HOST SOFTWARE

The network protocols implemented by the tray are low-level. The animation data is split up into separate blocks of frames for each tile, in a special bit-mapped format. The best way to develop content for this design is to use software tools like Adobe's Flash or Photoshop. Both of these PC-/Mac-based tools have extensive facilities for creating graphics and animation. Once the animations are created, they are easily exported to standard formats, such as a series of portable network graphics (PNG) or graphics interchange format (GIF) files.

The job of bridging the gap from high-level animation tools to low-level formats is done by a script written in Python. A nice package called the Python Imaging Library (PIL) simplifies the task of reading the standard image files, carving them up into tile-sized pieces, and reformatting them into the bitmap-based format the system uses. The specially formatted files are either sent directly to the tray over Ethernet or downloaded onto an SD card as PZM files.

## PRODUCTION VERSIONS

With a hand-made prototype up and running, it's fun to envision what a

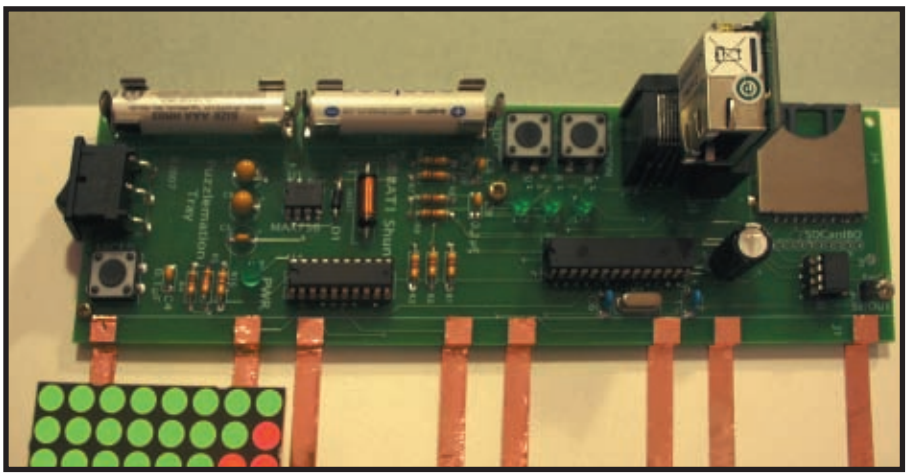


**Photo 4—**The tray takes data from the Ethernet or an SD card and broadcasts it to the tiles via the copper strips of tape. Tiles pick up the serial data via spring-loaded pins underneath them.

## TCP/IP SOCKETS

Systems for accessing TCP/IP networks, such as the Internet or a local Ethernet LAN, are usually based on a *socket* model. A "socket" represents an open TCP/IP network connection. When you write data to the socket, the computer on the end of the connection (anywhere on the planet) is able to read it, and vice versa. The underlying TCP/IP protocol takes care of the details. It chops the data into packets, routes them on their way, and even sends extra copies if some of the packets get lost.

Opening a socket requires two basic pieces of information: the *address* of the computer you're trying to connect to and the *port*, indicating the service or function on that machine you're trying to access. Internet protocol (IP) addresses are currently 32 bits and usually expressed as groups of four 8-bit decimal numbers. For example, www.circuitcellar.com is found at 206.223.65.196. A network service called domain name service (DNS) converts convenient text names into their numerical addresses. Many high-level socket libraries use names interchangeably with numerical addresses.

The "port" refers to the service you're trying to access. For example, Internet standards reserve port 80 for accessing web pages. Port 25 is used for sending e-mail. There are hundreds of different services that use different ports. Some are standardized and well established (like web and e-mail), while others, usually with port numbers above 1,024, are defined on the fly for particular applications. A computer creates a service by opening a socket on the port for that service and "listening" for connections. When a connection is made from a remote computer, it accepts the connection and begins processing the data.

Many of the protocols are surprisingly simple. For example, try typing **telnet www.google.com 80** at your favorite command line to open a connection from your keyboard to Google's web server (port 80). Then type **GET / HTTP/1.1** followed by two control-Js. (Depending on your computer's telnet implementation, your keystrokes may or may not be echoed.) You'll see the code for Google's homepage come flying back. Another useful command is **ping hostname**. It sends a special diagnostic packet to the remote host every second or so and writes a message when it gets echoed back.

Using Microchip Technology's TCP/IP library, it's simple to create a service on a PIC processor that listens for connections at a particular port. The top of Listing 1 is an excerpt from the Puzzlemation tray code that listens for text to display on the tiles. I arbitrarily chose port 1092 for it to listen for text messages and port 1093 for listening for animation data. To send text to the tiles, you can use the telnet command: **telnet nnn.nnn.nnn.nnn 1092** (where the "n"s represent Puzzlemation's IP address). Or, use a simple Python program, like the one at the bottom of Listing 1.

These days, most computers, particularly in office environments, are behind network "firewalls" that remap IP addresses and restrict traffic to a few standard ports. But, if they're connected directly to the Internet, you could run the Python program in Listing 1 on a computer in Paris and the message would appear on the Puzzlemation tiles in California—without changing a line of code. With TCP/IP sockets, a "remote" connection really does mean *remote*.

**Listing 1—**This is sample code for communicating over a TCP socket. The top part (in C) implements the server running on the Tray for displaying text messages. The bottom part (in Python) shows how text messages are sent to the tray from a remote host.

```
//
// Server for displaying text messages.
// This routine should be called from the MCU's main loop.
//
void DisplayTextServer()
{
    char msgBuffer[kMaxTiles + 1];
    int charsAvailable;
    int needDiscard = 0;
    static int serverInitialized = 0;

    if (! serverInitialized)
    {
        TextSocket = TCPListen( TRAY_TEXT_PORT ); // Port 1092
        if (TextSocket == INVALID_SOCKET)
            SendErrorMessage("No txt Sock!");
        else
            serverInitialized = 1;
    }

    if (! TCPIsConnected(TextSocket))
        return;  // No incoming connection yet...

    // How much data is waiting?
    charsAvailable = TCPIsGetReady( TextSocket );

    // Limit what we'll take to the number of tiles we have
    if (charsAvailable > kMaxTiles)
    {
        charsAvailable = kMaxTiles;
        needDiscard = 1;
    }

    // Read the data
    TCPGetArray( TextSocket, (BYTE *)msgBuffer, charsAvailable );

    msgBuffer[charsAvailable] = '\0';     // Null terminate

    // Send data to the tiles
    SendString( msgBuffer, currentTextColor );

    // Throw away what we couldn't use
    if (needDiscard)
        TCPDiscard( TextSocket );

    // Send acknowledgement back
    TCPWriteString( TextSocket, "OK\n" );
    TCPDisconnect( TextSocket );
}
#######################################################
# Python code for the host to
# send messages to Puzzlemation
#
import socket

def sendTrayMessage(msg):
    TrayHost = '192.168.0.200' # The IP address of the tray
    TrayTextPort = 1092        # Port used for sending text messages
    # Opens a socket to the text port
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((TrayHost, TrayTextPort))
    s.send(msg)
    result = s.recv( 2 ) # Check for acknowledgement from the Tray
    print result
    s.close()

sendTrayMessage( "PUZZLEMATION" )
```

| Animation port commands | | |
|---|---|---|
| **Name** | **Format** | **Description** |
| Add | A<name>,<#tiles>,<#frames>\n<data...> | Download animation |
| Start | S[<name>]\n | Start animation |
| Erase | E<name>\n | Erase animation |

**Table 2—**These are commands interpreted by the animation port. Animations are downloaded onto the SD card with this protocol.

mass-produced version would look like. A production version of the tiles could probably be built with a single circuit board, a PIC24FJ64 microcontroller, a crystal and driver on the top side underneath the display, and the power electronics on the bottom. This was undesirable in the prototype, where everything had to be accessible for testing and debugging. The batteries and contact pins could be held in custom-molded plastics. A production tray would have matching snap-in receptacles for the tiles with built-in contacts. This would make it easy to hang the display on a wall. It could also double as a charger for the tiles's batteries.

The current prototype uses 2 bits per pixel and allows four colors: dark, red, green, and yellow. By allocating more memory for the pixels, a PWM scheme could control the brightness of the LEDs, allowing for more colors. This has a serious memory trade-off, though. Adding even a few extra bits per pixel dramatically cuts the number of available animation frames.

Another interesting extension is using full-color LED matrix tiles with red, green, and blue pixels. By using a demultiplexer for the column drive lines and removing the address jumpers (the tile ID may be written into the microcontroller's EEPROM with a special command), the 44-pin PIC24FJ64GA004 would have enough I/O pins to drive a full-color display. Unfortunately, cost is a significant issue. At least in small quantities, full-color RGB versions of the LED matrices are nearly five times more expensive than the red/green display used in the prototype.

My design is a platform for creating animated LED displays. You can make everything from puzzles and games to animated signs that are easy to reconfigure for any application.

*Author's note: You can download Microchip's TCP/IP library, along with the documentation (AN833), at http://tinyurl.com/6dh7m7. Additional photos and a video of this design are available at www.puzzlemation.com.*

*John Peterson is a Silicon Valley software developer whose work appears in major applications software. He received a B.S. and an M.S. from the University of Utah, focusing on computer graphics and image processing. John holds seven U.S. patents. You can contact him at saccade@gmail.com. John's other system design projects are showcased at www.saccade.com.*

## PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/221.

## RESOURCES

J. R. Hall, "8 × 8 Bitmapped Font," http://overcode.yak.net/12.

K. Maxon, "Have You Seen My New Soldering Iron?," *Encoder*, June 2006.

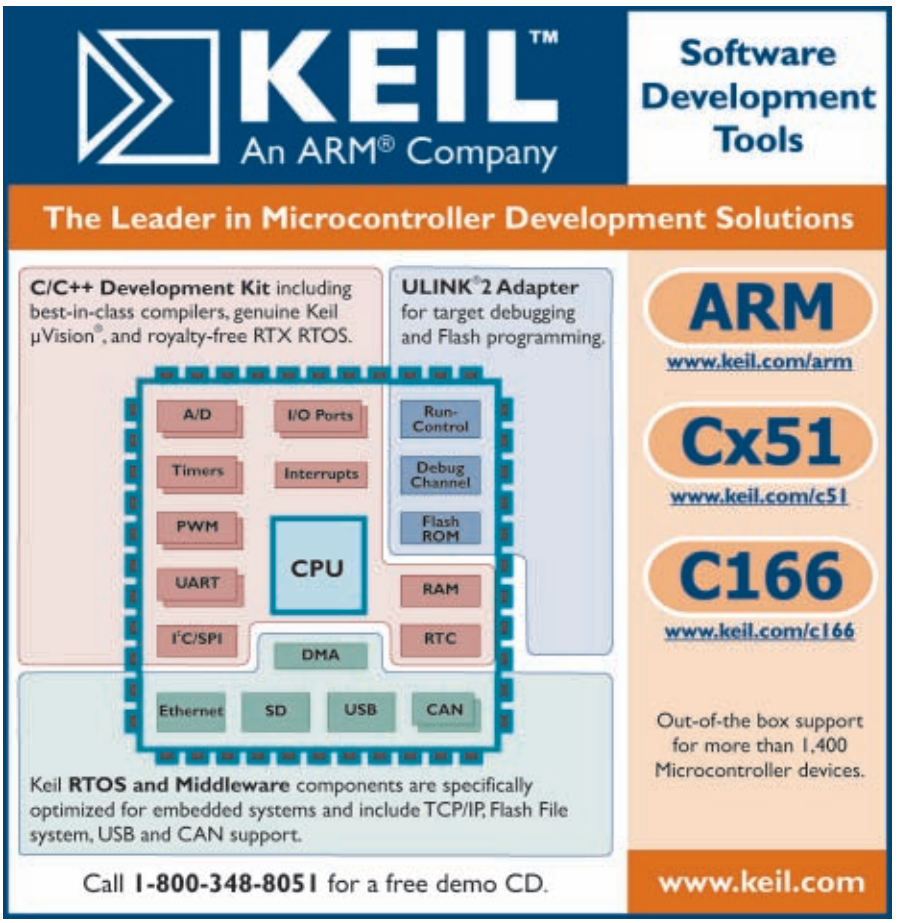
N. Rajbharti, "The Microchip TCP/IP Stack," Microchip Technology, Inc., AN833, 2008.

P. Reen and N. Mohaanswamy, "Implementing File I/O Functions Using Microchip's Memory Disk Drive File System Library," Microchip Technology, Inc., AN1045, 2008.

## SOURCES

**25LC256 Serial EEPROM, ENC28J60 Ethernet controller, MPLAB ICD2 debugger/programmer, PIC24FJ64GA002 and PIC24FJ64GA004 microcontrollers, and PICtail+ Ethernet**
Microchip Technology, Inc.
www.microchip.com

**Python Imaging Library (PIL)**
PythonWare
www.pythonware.com/products/pil/



# The Newest Semiconductors




*The ONLY New Catalog Every 90 Days*

*Experience Mouser's time-to-market advantage with no minimums and same-day shipping of the newest products from more than 366 leading suppliers.*

MOUSER ELECTRONICS®

a tti company

***The Newest Products For Your Newest Designs***

**(800) 346-6873**

**www.mouser.com**

*Over A Million Products Online*





# Tool Time

Tom has often bemoaned that the tools electronic designers rely on haven't kept up with the silicon. The answer to his complaint may be Altium's Designer, which allows you to complete an entire design, including hardware and software, from start to finish. When you're done, debug with Altium's NanoBoard.

Imagine you're embarking on a home improvement project—say, a kitchen remodel. Heading over to the local home improvement store you find all of the materials you need. There's finely milled wood, advanced composite countertops, precision-machined plumbing fixtures, and so on. Ain't technology grand?

With your cart fully loaded with material wonders of the modern age, you head over to the tools section, where you find some sharp rocks, a bin full of sticks, and woven reed rope. Gee, guess the first thing to do is build a hammer.

I'm sure you can see where this is headed. I've often lamented that the tools electronic designers rely on haven't kept up with the materials, namely silicon. The problem is that there's no Moore's Law for tools.

Yes, tools have gotten better. But they have to run as fast as they can just to keep up with the capabilities of rocket science silicon and the demands of feature-creep designs. After all, these days, even a simple "Hello World"-class application seemingly requires a 32-bit CPU with a touchscreen graphics GUI, RTOS, and megs of software.

## MISSING THE POINT

The historical reliance on "point" tools to handle various aspects of electronics design represents an obstacle to progress. Traditionally, major tasks such as chip design, system design, software programming, PCB layout, and physical packaging were handled by completely separate groups of specialists. Each group had only narrowly defined linkage and communication with the others and would otherwise work independently using their own specialized tools.

But these days, crafting a superior product requires a much more holistic design approach. Not that every engineer needs to become a jack-of-all-trades, and thus master-of-none. But the design process needs to be more team-oriented, collaborative, iterative, and flexible. Unfortunately, as each "point" tool becomes more complex and self-centered, the result is even less ability to play cooperatively with others.

Responding to the problem, a particular point tool supplier may provide a hook or path to "integration" with another. For example, Electronic Data Interchange Format (EDIF) is a well-known standard that provides a connection between various hardware design specialties (i.e., chip and system design and simulation). But there are gotchas. Few tool suppliers have the wherewithal to cover all the "point" tool bases, leaving engineers to craft their own lash-up by integrating a collection of partially integrated tool suites. At the same time, it seems that the historic vestiges of the different tools remain just underneath the surface, with only a cosmetic facade of superficial integration hiding the differences. Designers are left to deal with what feels like a mish-mash collection of parts cobbled together.

## 50,000 FEET

Originally founded in the 1980s as Protel International, the company now known as Altium was one of the first to jump on the concept of using the then-new PC as an EDA workstation for PCB design. Having succeeded with their PC-based PCB "point" tool in the mid-1990s, the company turned their attention to bigger things with the introduction of Protel, a tool that combined PCB and programmable logic design capabilities. Subsequently, like many other tool suppliers, Altium has pulled the pieces of the development process together by acquisition. An IPO in 1999 gave the company the cash to aggressively further their vision by adding logic synthesis (Metamor, Accolade Design Automation, Green Mountain Computing Systems) and embedded software (TASKING) tools to the mix.

Typically, this is about where the integration-via-acquisition strategy stalls out. Indeed, in many cases the "point" product strategy continues virtually unchanged, with "integration"

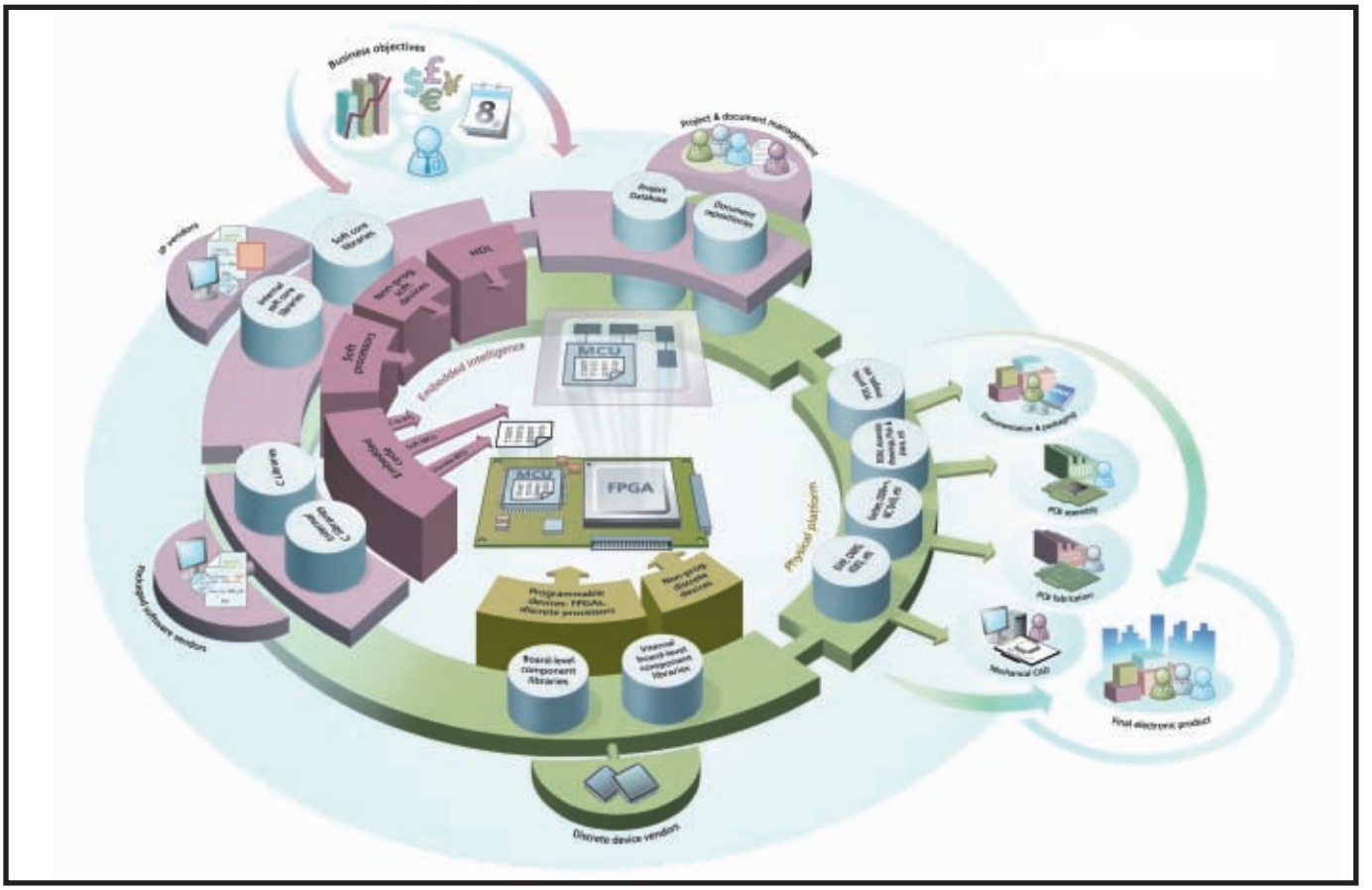



**Figure 1—**Moving beyond the "point tool" era, Altium unifies all aspects of electronic product creation into a single comprehensive solution they call Designer.

being more a financial exercise in "conglomeration" rather than a true breakthrough (i.e., the whole is little more than the sum of the parts).

So what's the Altium difference? Here's an analogy. Imagine you're handed a few pieces of a jigsaw puzzle and asked to make a picture out of them. What you would end up with depends on how many pieces you have and how well they fit together. You could probably come up with a picture of some sort, but it might not be a real work of art.

Instead, Altium starts with the big picture and gives you all the pieces of the puzzle you need to put it together (see Figure 1). With the passion of a true believer, their vision goes well beyond mere point tool "integration" to true "unification." The scope is sweeping, from the gates buried deep inside a chip up to and including the physical packaging of the entire design (see Photo 1). To Altium's way of thinking, the pieces of the puzzle (i.e., point tools) are just the means to a larger end (i.e., a successful product), not the end themselves.

Altium puts it all together in their aptly named "Designer," a single tool that can accommodate an entire design (i.e., hardware, software, PCB) from start to finish.

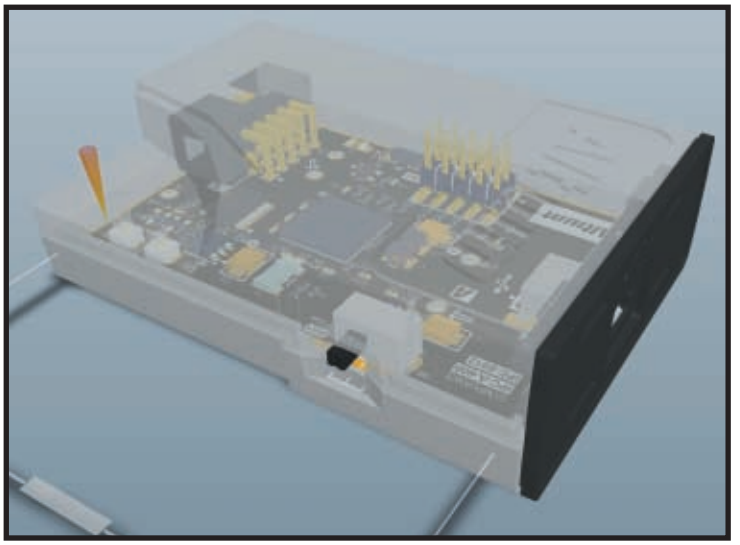


**Photo 1—**Designer takes PCB layout to the next level with solid modeling. You can change a chip on the schematic and quickly see how it affects the layout and packaging.

## NANO?

Designer's high-concept tool "unification" story is grand, but all for naught without some silicon behind it. Enter the hardware half of the story, the NanoBoard, which paired with Designer comprises what Altium calls the "Innovation Station." The "Nano" moniker is a bit ironic because it might imply something small or insignificant, but as you can see in Photo 2, the NanoBoard is anything but. "MegaBoard" is more like it.

NanoBoard is far beyond the typical "EV kit" or "demo board."

Think of it as an embedded erector set that enables you to quickly bolt together all manner of different applications.

The NanoBoard comprises three major subsystems. Along with power and USB connection to the host PC, the baseboard includes a large complement of generic I/O functions, including the usual switches, LEDs, serial, parallel, ADC, DAC, CAN, RTC, and so on. The notably nongeneric addition is the touch-screen color LCD. Altium even includes a stylus to poke it with. Although you can't see it in the photo, underneath is an audio board with stereo speakers and multicolor LEDs for the light show.

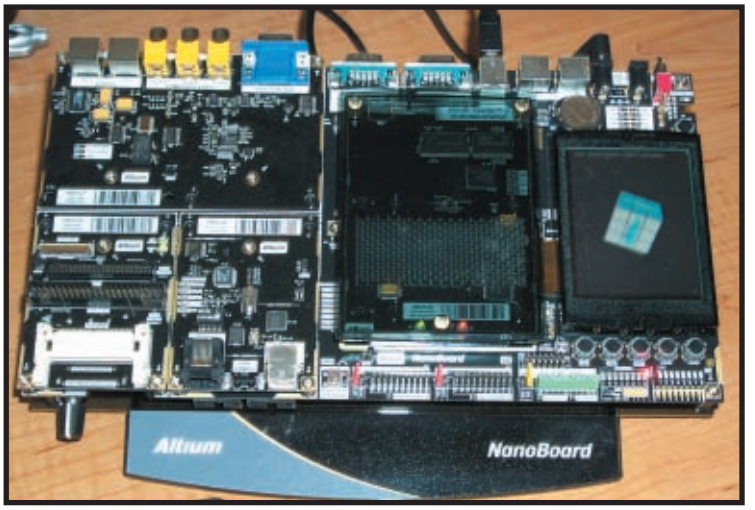



**Photo 2—**Are we having fun yet? Oh yeah. The NanoBoard is a new-age silicon sandbox for designers to play in.

Plugged into the baseboard on the left side you can see a collection of three "peripheral boards" that provide more specialized I/O capabilities. The larger board at the top (PB-01) is a complete video graphics subsystem with a 24-bit VGA and composite and S-Video inputs and outputs. On the lower left, the "mass storage" peripheral board (PB-02) provides a connection for various flavors of hard disk drives (1.8", 2.5", 3.5") as well as multimedia flash cards (e.g., CompactFlash, SD/MMC). The third board (PB-03) is for networking with 10/100 Ethernet, USB 2.0, and an IR interface (IrDA or the included TV remote). According to Altium, there are additional peripheral boards (e.g., 802.11) in the works as well.

It takes a big brain to manage all that I/O. There it is in the center of the NanoBoard (under the plastic cover) in the form of an FPGA card. Two cards, one Xilinx and one Altera, came with my kit and versions based on Actel. Lattice chips are also available. The FPGA card also has a bunch of memory, and I do mean a bunch, as in 64-MB SDRAM, 32-MB flash, and 2-MB SRAM. Finally, there are three 100-pin connectors routing signals between the FPGA pins and the I/O on the peripheral boards and baseboard.

Contemplate the NanoBoard capabilities a bit and you'll realize it can handle a wide, wide variety of applications. If there's a particular chip or I/O feature missing, you can certainly bring it into the mix via the NanoBoard "User Header" expansion connectors, which feature jumper-selectable 3.3- or 5-V power.

## FIRE EXTINGUISHER

At this point, let's take a bit of a test drive to introduce key features and get a feeling for what "unified" design is all about. Keep in mind I'm scratching only the surface of this extremely deep software. For every feature I mention, there are 10 more behind it.

A Designer "project" consolidates every file associated with a design (e.g., schematic, VHDL, "C," ASM, PCB layout, documentation, and more) in one place. In turn, every project starts with a blank "sheet," which is the highest level of the design. An object on the top sheet might be a stand-alone chip, logic in an FPGA, or yet another lower-level sheet in a hierarchical fashion. When you hit the "Build" key, Designer crunches everything (e.g., synthesizes the VHDL, compiles the "C"), drawing from extensive libraries of canned IP along the way.

Handling the gory details of FPGA implementation (e.g., place and route and timing analysis) is the single exception to Designer's "one-program-does-it-all" philosophy. Instead, at that point in the project build, Designer hands off to the vendor-specific (e.g., Xilinx, Altera) FPGA toolchain to do the heavy lifting. The transition is handled unobtrusively behind the scenes and is indeed completely transparent, at least if all goes well (i.e., no place and route or timing errors).

One of the cool things about Designer is that it enables you to design in a rather FPGA brand-agnostic fashion. In addition to vendor-specific FPGA libraries, there's also a comprehensive "generic" FPGA

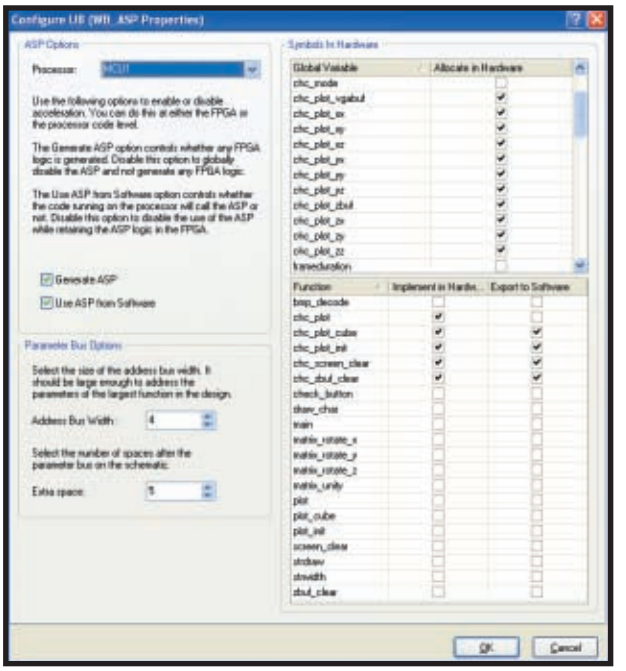



**Photo 3—**The black magic of turning "C" into stand-alone hardware is to decide what portions of your "C" program you tell the "CHC" compiler to translate to VHDL.

# Capacitive Touch Sensing In a Flexible, Single-chip Solution






Capacitive touch interfaces provide an excellent way to add low-cost, reliable and stylish buttons into your design. Microchip Technology's mTouch™ Sensing Solution includes comprehensive development kits and a free diagnostic tool to make implementation easy and fast. Our free source code can be seamlessly integrated with your existing firmware on a single PIC® microcontroller – eliminating the need for additional controllers.

## THE mTouch SENSING SOLUTION FEATURES:

- FREE license libraries and source code
- A FREE diagnostic tool
- Integration with 8- and 16-bit PIC microcontrollers
- Easy expansion, with support from 6 to 100 pins
- Low-power operation

## GET STARTED IN 3 EASY STEPS

1. Visit the mTouch Sensing Solutions design center at **www.microchip.com/mTouch**
2. Download FREE libraries and source code
3. For a limited time, save 20% off a variety of touch sensing development tools when you purchase from **www.microchipDIRECT.com** and use coupon code **mTouch4U**.



The Microchip name and logo, the Microchip logo, MPLAB and PIC are registered trademarks of Microchip Technology Incorporated in the USA and other countries. mTouch and PICDEM are trademarks of Microchip Technology Incorporated in the USA and other countries.
All other trademarks are property of their respective companies. © 2008 Microchip Technology Incorporated. All rights reserved.



MICROCHIP

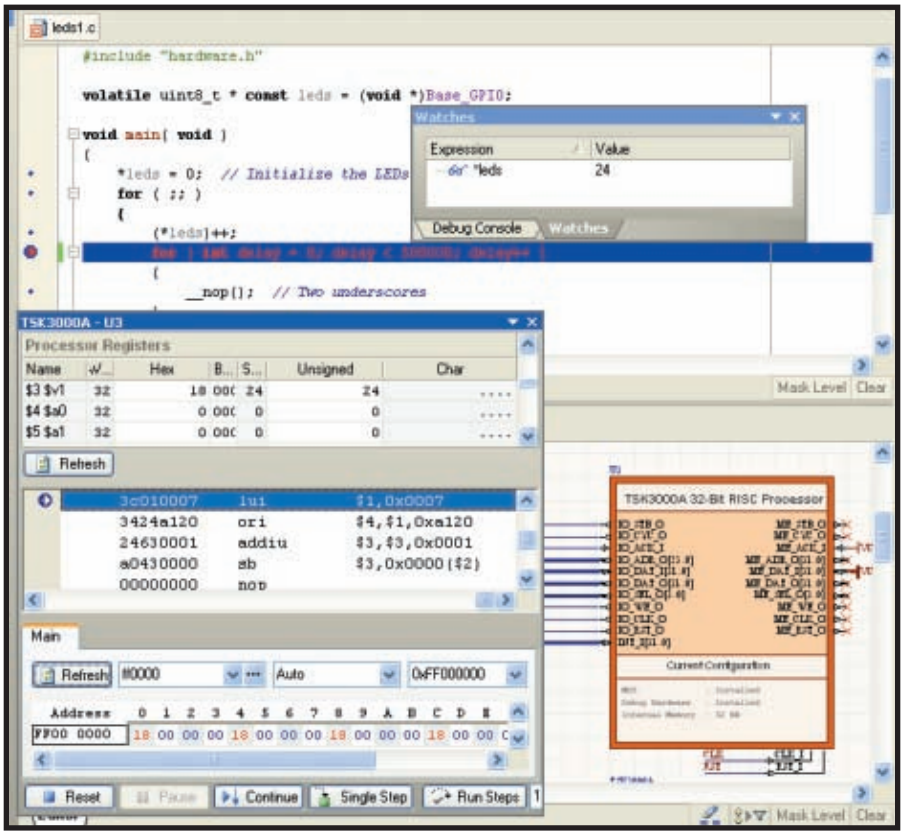



**Photo 4—**Designer comes with plenty of IP to use in your designs including the TSK3000A, a pipelined 32-bit soft-core RISC fully supported with a high-quality "C" toolchain.

library. By taking advantage of device independence, it's possible to retarget a Designer project to a different brand of FPGA late in the design cycle. The same can't be said for using vendor-specific FPGA tools, which tend to lock you into a particular brand of chip.

Altium is a big believer in soft-core processors when it comes to embedding intelligence, so Designer includes support for the usual suspects (e.g., Xilinx's MicroBlaze, Altera's Nios). Indeed, Altium claims their in-house (via the TASKING acquisition) "C" compiler generates better code than the GNU versions typically found.

Going a big step further, Altium offers their own 32-bit soft core called the TSK3000A. It's a classic *I Love the 80s*, five-stage pipelined, 32-bit fixed-length instruction, load-store RISC. Indeed, although I can't say every detail is exactly the same, it's pretty obvious that the TSK3000A harkens back to the original MIPS R3000. With big iron features (e.g., hardware divider, barrel shifter), it's not the leanest and meanest core on the block. But it doesn't matter because the most unique and compelling feature of the TSK3000A isn't its architecture, but rather that it works with different brands of FPGAs.

The fact that Designer works with a variety of processor cores and FPGAs allows you to explore various hardware options late in the cycle and could lessen the pain caused by a sudden change of plans. Imagine a scenario where you are 90% finished with a design using Xilinx-specific chips, tools, and IP (e.g., MicroBlaze). All is going well until the boss decrees: "We're switching to Altera." Not a pretty picture, eh? Designer's multivendor capabilities are like having "fire drill insurance."

## IF THE SHOE FITS

Long-time readers have probably heard my joke before. These days, every designer sits at a screen typing stuff in, be it "C" or VHDL. So, how do you tell the difference between a hardware engineer and a software programmer? Easy, just check if they're wearing shoes. Ba-dum-dum. I'll be here all week.

It may not be so funny, but it has never been more true than with Designer. Click on a sheet object and you might find a schematic symbol or some VHDL. No surprise there. But with Designer, you might also find a chunk of "C" code. This goes beyond the "C" code running on a soft-core processor. Everybody can do that. I'm talking about translating "C" to stand-alone hardware.

Here's how it works. Under the hood, Altium uses what they call a C-to-Hardware Compiler (CHC) to translate "C" to VHDL, which is then synthesized to hardware in the usual way. The beauty of the scheme is that you can initially write and debug a software version of your logic using the conventional "C" tools. Once it's verified, you can tell Designer which portions (i.e., functions and variables) of the software code to morph into hardware (see Photo 3).

The result goes into a so-called application-specific processor (ASP) that's wired up on a sheet like any other component. ASP I/O lines provide the means for variable transfer and function dispatch via automatically generated "wrapper" code that replaces the now hardened "C" code previously run by the host processor.

One of the nifty Designer demos rotates a cube image on the NanoBoard LCD using both software and hardware options. The demo starts by rotating the cube with "C" software in the conventional way (i.e., running on a TSK3000A). With all the matrix operations, it's kind of pokey. But hit a switch on the NanoBoard and operation switches to a hybrid hard and soft version comprising "C" on the TSK3000A accelerated with "CHC"-synthesized hardware. To make a long story short, when you hit the "go fast" switch, the demo runs 10× faster.

To be sure, there is a bit of fine print. The documentation alludes to the need for you to give "CHC" something to work with (i.e., code amenable to hardware optimization). Needless to say, that's a weighty subject that's on everyone's mind. To make the long story long again, you

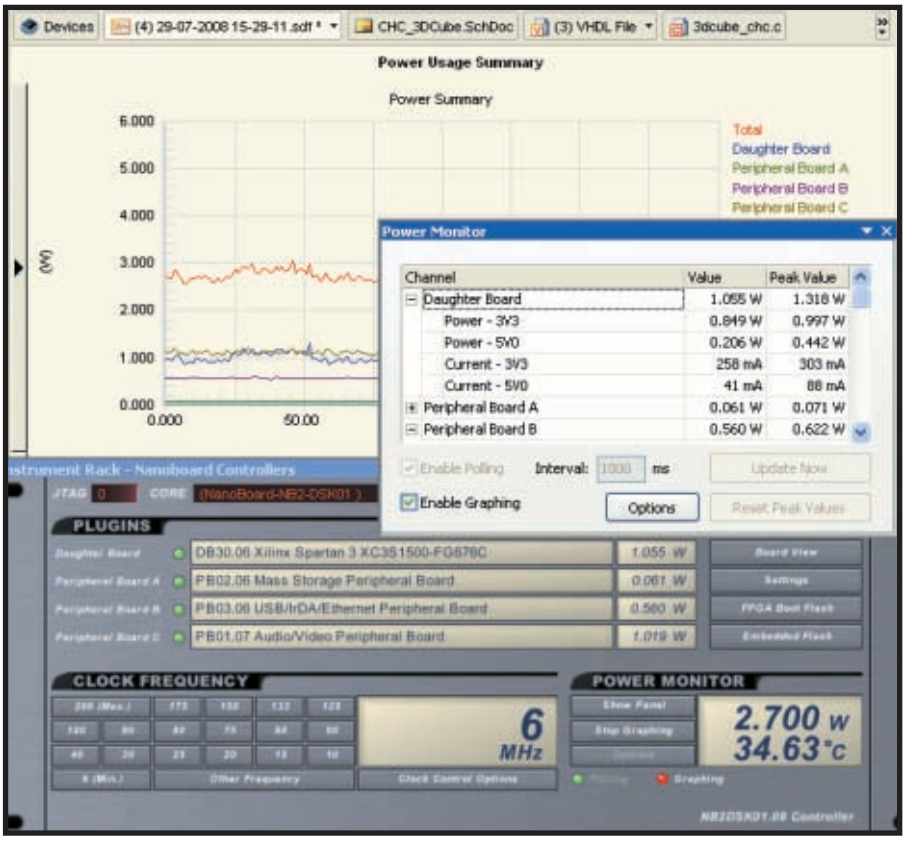



**Photo 5—**Here's an example of the power, literally, of unified design. With everything under Designer's control, it's easy to get a comprehensive view of what's going on. Try that with your lash-up of traditional point tools.

can read the 100-plus page "CHC" user manual, starting with the chapter on "Parallelism."

So don't expect "CHC" to somehow "automagically" extract parallelism and optimize hardware/software partitioning. Nevertheless, it's still pretty darn cool. As with Designer's ability to port a project to different brand FPGAs, "CHC" enables exploration of hardware and software trade-offs late into a design. Moving even a small piece of critical path software into "CHC" hardware might save the day when the deadline is looming.

## THE FUN HAS JUST BEGUN

Once your design is complete and downloaded to the NanoBoard, the fun is just beginning. It's debug time!

As you can see in Photo 4, software checkout is supported with the typical array of debug tools (e.g., breakpoints, single-step, watches, and more). But it's hardware debug where the Innovation Station really stands out. By virtue of "unifying" all the tools (i.e., Designer) and hardware (i.e.,




NanoBoard and everything on it) what otherwise might be a cumbersome lash-up of various downloaders and debuggers (e.g., MCU and FPGA) is reduced to a single USB connection to your PC. The USB umbilical cord talks to a JTAG bus connecting the various chips and peripherals on the NanoBoard, or for that matter, even across a chain of multiple NanoBoards. The result is complete visibility into all aspects of your design (see Photo 5).

Along with this traditional hardware JTAG, Designer also includes a so-called "soft JTAG" chain. Notably, it allows you to see what's going on inside the FPGA, eliminating the need for a bunch of traditional test equipment, gear that's of little use these days with everything buried deeply inside silicon.

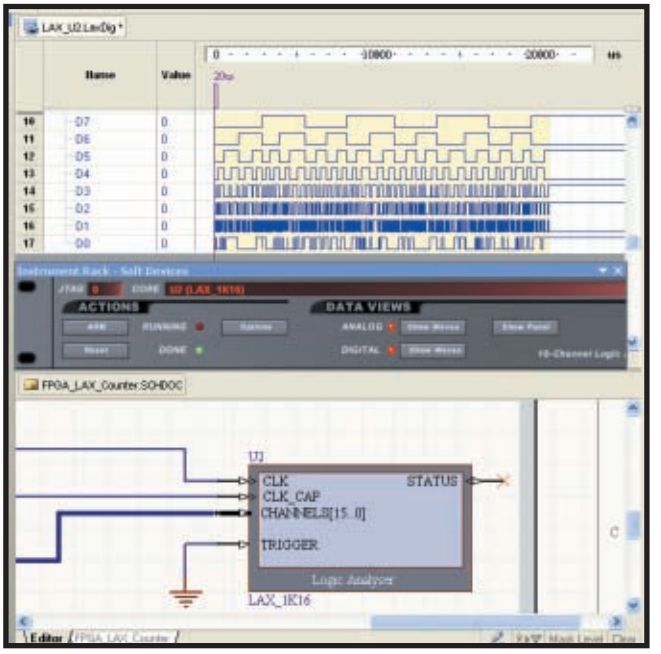



**Photo 6—**Moving all the intelligence inside chips means you've got to move all your bench-top test equipment inside too.

Designer comes with a library of predefined instruments that includes lab-bench favorites such as a frequency counter, a clock generator, and even a logic analyzer (see Photo 6). Just drop them onto the sheet and wire them in just as you would a real instrument. Even for a signal accessible externally (i.e., routed to a pin), using a soft-JTAG instrument is much easier than trying to hang a bunch of probes on a chip with a zillion tiny pins.

By now you may be getting the message that the folks at Altium seem to think of everything. It should come as no surprise to discover that Designer also gives you the ability to create your own custom-designed instruments. Just drop a custom instrument onto the sheet and craft its appearance using a selection of predefined widgets such as virtual LEDs (discrete, seven-segment, bar graph), a gauge, or a graph (see Photo 7). If you want to get really fancy, you can craft a script with some processing capability that fires periodically or in response to a particular stimulus.

## REALITY CHECK

Needless to say, I'm really impressed with this Altium stuff, including both the vision and execution. As far as I'm concerned, these folks really "get it" when it comes to embedded design, today and tomorrow. But lest infatuation cloud reason, I want to make a point of taking off the rose-colored glasses. Note that many of the concerns I raise are generic across the industry and not particular to Altium.

First is the not-so-small matter of getting everything up and running, and keeping it that way. Remember that in addition to Windows and the Designer software, you need to install the tool chains for every brand of FPGA you use. I suspect it will be a challenge to keep the entire platform in sync as each piece of software migrates through updates and revisions, with the whole XP/Vista migration complicating it further. Witness the fact that even during the short course of this evaluation I had to upgrade Designer to keep up with a new release of the Xilinx tools.

If you enjoy lots of long and leisurely coffee breaks, go ahead and try to run all this stuff on some hand-me-down PC. Otherwise, you'll want the biggest and baddest box on the block, the more hot-rodded the better. A big share of the finger-pointing is aimed at the FPGA toolchains, where place and route is seemingly a black hole for PC gigahertz and gigabytes. For instance, building the aforementioned 3-D cube demo took nearly 15 min. on my admittedly blue-collar, 1-GHz PC, but almost 90% of that was spent in the Xilinx toolchain, and that was for a design that fills just one-third of the FPGA.

It's hard to get the big "unified" picture on a small screen, so consider investing in multiple displays. I've seen Altium demo setups with two and even three screens for the full "Master of the Universe" feel. The advanced display features (e.g., 3-D solid modeling) are well-served by a fast

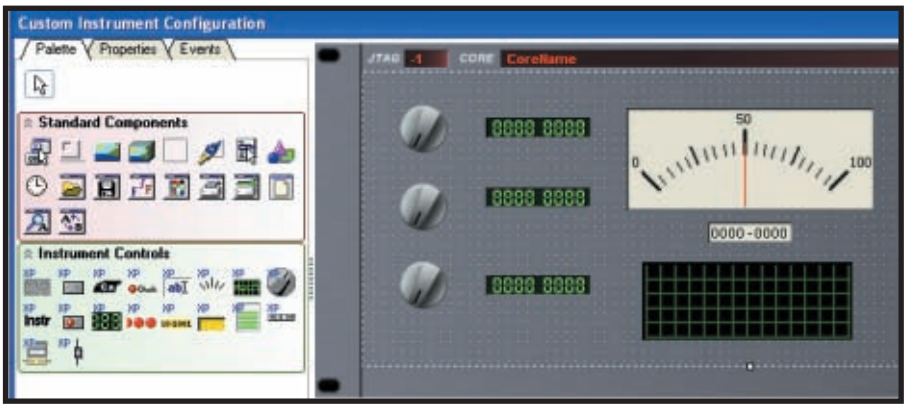



**Photo 7—**Debugging a design isn't always fun and can be a time-consuming chore. The ability to craft a custom test instrument with Designer helps zero in on bugs fast.

graphics card (e.g., DirectX accelerator) that can push pixels pronto.

And then there's the not-so-small matter of the price—$4,300 for the NanoBoard and a one-year license for a "core" version of Designer (i.e., everything except PCB layout). Once I was past the sticker shock, I realized that's actually a decent deal considering all that's included. The NanoBoard itself is by far the classiest piece of development hardware I've come across, fully capable of handling significant "real" designs. And as for Designer, you're getting all you can eat tool-wise with beefy entrées including schematic capture, VHDL, CHC logic synthesis, and multi-target "C" toolchain. And then there are the vast libraries of IP you can use for your own design soft-core processors from 8 to 32 bits, I/O functions of all sorts, the peripheral board reference designs, predefined instruments, and more.

If I have a problem, it's as much with the "licensing" model that seemingly permeates the EDA software tool market. The upfront price is one thing, but over time you can easily spend that much or more for licensing. Again, this issue is not unique to Altium, and their licensing model actually seems pretty straightforward compared to some I've seen that have a lot of fine print "nickel and dime" and "let's make a deal" aspects.

## WORLD BEYOND 'WARE

It's amazing, and frankly refreshing, to realize you can complete an entire design (i.e., hardware, software, PCB) launching a single program. Altium's vision gets to the point of modern design, which is to get beyond the point tools.

But even the most compelling vision is hostage to real-world challenges. Yes, issues such as price (as much the "licensing" model as the upfront cost), pokey place and route and the soft-core gap (i.e., price/power/performance compared to hard-core) remain a challenge. But the bottom line is that Altium has definitely got it together with a first-rate package that worked better than I expected at every turn. Indeed, this may be the first time I've ever finished a column feeling like the tools were keeping pace, or dare I say even catching up, with the silicon.

It's time to move beyond 'ware (i.e., hardware and software point tools) and get on to the bigger challenge of designing and innovating our way to a brighter future. Altium's doing their part with Designer and the Innovation Station. Now it's time for you to do yours.

*Tom Cantrell has been working on chip, board, and systems design and marketing for several years. You may reach him by e-mail at tom.cantrell@circuitcellar.com.*

## SOURCE

**Designer product design tool, NanoBoard development platform, and TSK3000A RISC processor**
Altium
www.altium.com


**No other PCB-design tool gives you more value per dollar**

Boards designed under EAGLE are developed in one-man businesses or in large industrial companies. Most of the top companies are our customers. The crucial reason for selecting EAGLE is not usually the low price, but rather the high-end functionality along with the ease of use. And EAGLE users appreciate the outstanding level of support, which at CadSoft is always free of charge, and is available without restriction to every customer.

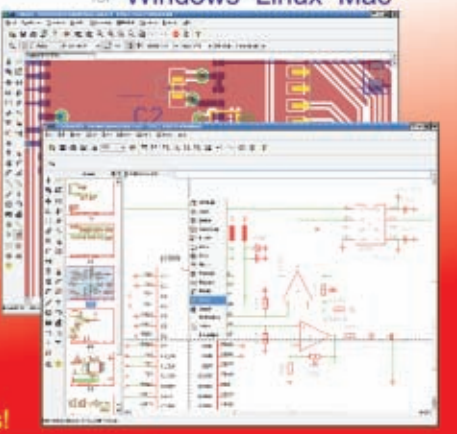
These are the real cost killers!

Version 5 is even easier to use, especially for beginners, due to an enhanced user interface.



EAGLE
Version 5
Schematic Capture • Board Layout
Autorouter
for Windows® Linux® Mac®

Version 5 Highlights
- Stand-alone schematic editor available.
- Automatic signal/contact cross references using frame coordinates.
- Right-mouse click for more consistent Windows UI.
- User-definable attributes for parts.
- Schematic sheet management.
- Hiding approved DRC and ERC errors.
- PRINT preview and text-searchable PDF output.
- Improved search engine for help.
- And much, much more.



Pick the level that is right for you — pay only the difference for upgrades!

| | Light | Standard | Professional |
|---|---|---|---|
| Max. number of schematic sheets | 1 | 99 | 999 |
| Max. board size | 4x3.2 inch | 6.4x4 inch | 64x64 inch |
| Max. # of signal layers | 2 | 4 | 16 |
| Layout or Schematic Editor | | $249 | $498 |
| Layout and Schematic Editor | | $498 | $996 |
| Layout Editor and Autorouter | | $498 | $996 |
| Layout Editor and Schematic Editor and Autorouter | $49 | $747 | $1494 |

Standard and Light Editions have full functionality except for the limitations mentioned in the table.

You can use EAGLE Light for evaluation and non-commercial applications without charge. Download it from our web site.

www.cadsoftusa.com
800-858-8355

CadSoft Computer, Inc., 19620 Pines Blvd., Suite 217, Pembroke Pines, FL 33029
Hotline (954) 237 0932, Fax (954) 237 0968, E-Mail: info@cadsoftusa.com

Windows / Linux / Mac are registered trademarks of Microsoft Corp. / Linus Torvalds / Apple Computer, Inc.

# Cost-Effective Mobile Data Storage

## Interface an SD Memory Card with an MCU

When it comes to designing an embedded application, memory matters. In this article, Mandar explains how to expand a microcontroller's nonvolatile data storage capacity with an SD card. It's an intelligent way to handle the excessive memory requirements of typical embedded applications.

In the last couple of years, the Secure Digital (SD) memory card has become a widely-accepted memory solution for a variety of portable devices and equipment, including cameras, PDAs, mobile phones, GPS navigation modules, gaming consoles, MP3 players, and more. Another useful technology is the high-performance 8-bit RISC Atmel AVR microcontroller, which is suitable for a variety of applications. The ATmega32 has 32 KB of flash memory that can be used to store code (e.g., Hex files). But what if your embedded system is memory hungry for applications, such as a data logger for a weather station or an industrial automation application to log alarms and trends? Both systems require more nonvolatile memory storage than the ATmega32 has to offer (flash memory/EEPROM).

In this article, I will describe how to interface an SD card to expand an AVR microcontroller's nonvolatile data storage capacity, which should fulfill the excessive memory requirements of a typical embedded application. In this project, an ATmega32L is interfaced with 256- and 512-MB SD cards from Kingston Technology and a 1-GB SD card from Transcend Information and SanDisk (see Photo 1).




**Photo 1**—This is an SD card compared with a 5-rupee coin.

## DESIGN CONSIDERATIONS

The first design considerations for this type of application are the supply voltage, the current requirement, and I/O lines and form factor. SD cards typically operate at 3.3 V, with a modest current requirement of 75 mA (maximum). Next, consider that the number of microcontroller I/O lines used to connect external storage must be kept to a minimum to interface. SD cards support the SPI protocol, so just four I/O lines are required to interface. The form factor of an SD card is 24 mm × 32 mm × 2.1 mm. (Refer to the file labeled pinout details.doc posted on the *Circuit Cellar* FTP site.) The previous design considerations are the main advantages of interfacing an SD card in an embedded system.

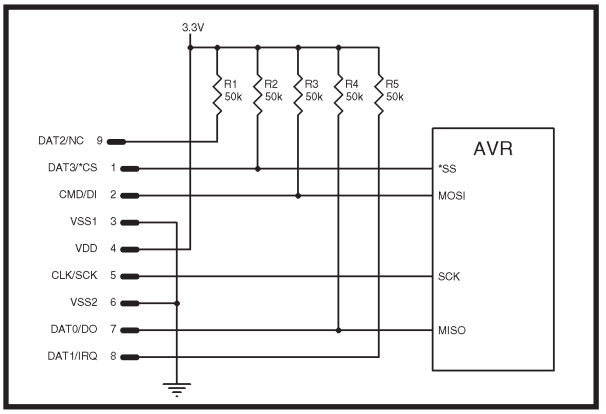



**Figure 1**—This is a schematic of the SD card and AVR microcontroller.

Compared with other available flash memories, an SD card is the better option. For instance, the 512-KB AT29LV512 flash chip requires 15 address lines, eight data lines, and three control lines, when interfaced with a microcontroller.

SPI and I²C-based external memory devices require four or two interface lines, but the maximum memory capacity of these devices can go up to 64 MB (i.e, the AT45DB642D, which costs $6 at Mouser Electronics). Cost is an important factor. A 1-GB Transcend SD card costs $6.15 (Source: Amazon.com).

The on-chip integrated SPI available in AVR microcontrollers is an advantage of interfacing and accessing the SD memory card (see Figure 1). Compared

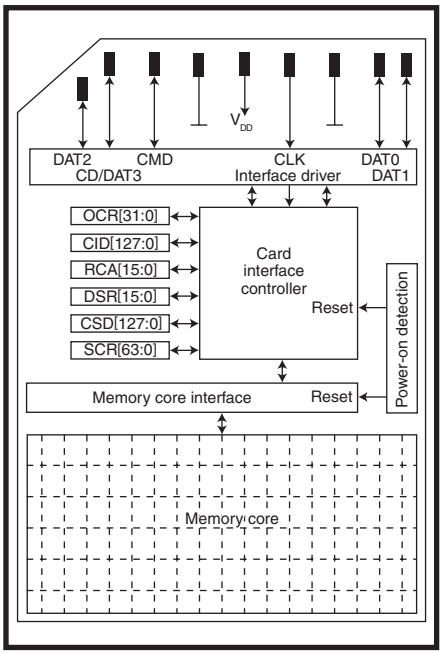



**Figure 2—**This is the internal architecture of an SD card. (Source: Transcend, www.transcendusa.com)

with other popular detachable memory cards (i.e., CompactFlash), the SD card supersedes it in terms of form factor, cost, and ease of interfacing (number of connections).

## HISTORY

The SD card is specifically designed to meet the security, capacity, performance, and environmental requirements inherent in newly emerging audio and video consumer electronic devices. In August of 1999, Matsushita Electric Industrial, SanDisk, and Toshiba announced an agreement to jointly develop, specify, and widely promote a next-generation secure memory card called the SD memory card. At the 2000 Consumer Electronics Show, the three companies announced that a new industry-wide association would be created to set industry standards for the proprietary SD memory card and promote its wide acceptance in digital applications.

The new organization, named the SD Card Association (SDA), is based in California and its executive membership includes about 30 world-leading high-technology companies and major content companies. Sampling of the SD card began in the first quarter of 2000 and production shipments commenced in the second quarter of 2000.

## HARDWARE

The SD card has a controller in it (see Figure 2). The flash memory control enables you to perform basic functions including erase, read, write, and error control inside the memory card. Data is transferred between an SD card and a host-interfaced microcontroller in units of 512 bytes per block.

The first piece of hardware in the project is the SD memory card header. To enable you to easily connect to the SD card, I used a 3M card connector SD (Mouser Electronics, SD-RSMT-2-MQ).

The second piece of hardware is the Transcend 1-GB TS1GSDC SD memory card. It is fully compatible with SD card specification V1.1. To test the setup, 256- and 512-MB SD cards (from Kingston) and a 1-GB SD card (from SanDisk) were also procured and tested.

The third piece is the ATmega32L microcontroller. The ATmega32L-8PU is the best choice for this setup because the operating voltage is from 2.7 to 5.5 V and it operates at 8 MHz. The 1-KB internal SRAM is advantageous because you can declare a buffer size of 512 bytes to exchange data with the SD card. A SPI is an integral part of the microcontroller. The full SPI details are covered in the ATmega32's datasheet. The microcontroller and SD card are powered up at 3.3 V.

The fourth piece is the serial interface. The setup has a MAX232 interfaced to a microcontroller, by which the microcontroller can transmit the block (512 bytes) read from the SD card and the response (R1) to HyperTerminal (see Photo 2).

## FIRMWARE DESIGN DETAILS

The SD card has a dual personality. The first one is SD Protocol (1 bit or 4 bits) Native mode. The second, which is supported, is the SPI mode of the SD card protocol. It is distinct from the 1- and 4-bit protocols in that the protocol operates over a generic and well-known bus interface, SPI (see Figure 3).

A SPI is a synchronous serial protocol that is extremely popular for interfacing peripheral devices with microcontrollers, where a microcontroller is configured as a master and an SD card as a slave. The initialization is covered in the firmware routine. The microcontroller development board communicates with a PC via an RS-232 interface. The status and read data is transmitted to HyperTerminal.

## SD COMMAND FORMAT

As you can see in Figure 3, the command that is transmitted by the microcontroller to the SD card is in a 6-byte packet. The first byte is the command. The next 4 bytes are the argument. The sixth byte is the CRC

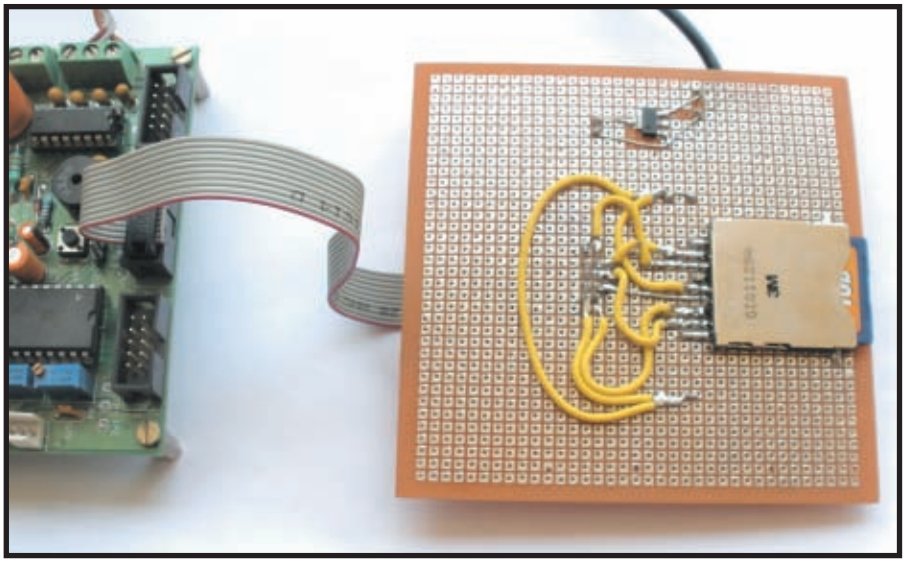


**Photo 2—**The SD card is interfaced with an AVR microcontroller development board.

HOLTEK
www.holtek.com

# Integrated Communication Solutions

Holtek's HT95R2X/3X series of I/O Phone MCUs, with their fully integrated DTMF generator and receiver, offer users a wide range of excellent price/performance ratio device solutions for an extensive array of communication products. Expanded flexibility with the inclusion of both SPI and $I^2C$ interfaces allow for easy and convenient communication to other external hardware. With their diverse array of other peripheral functions, the device range can expand the application possibilities of multi-function telephones, cordless telephones, household security products as well as a host of other related application areas.






10 - pin MCU | USB MCU | Dual Slope MCU | EEPROM | Power Management | TinyPower™ MCU | Flash MCU | UART MCU | Audio MCU | **Phone MCU**

- Operating Voltage Range: 2.2V~5.5V
- SPI/$I^2C$ Interfaces (HT95R35)
- Industrial Specifications: -40°C~+85°C
- Internal DTMF generator and receiver
- Excellent noise immunity and ESD protection
- PFD generator

| Part No. | ROM | RAM | I/O | Timer | RTC | DTMF Generator | DTMF Receiver | Package |
|---|---|---|---|---|---|---|---|---|
| HT95R22 | 4K x16 | 0.5K x8 | 26 | 16-bitx2 | V | V | - | 28SOP |
| HT95R23 | 4K x16 | 1K x8 | 36 | 16-bitx2 | V | V | - | 48SSOP |
| HT95R24 | 8K x16 | 2K x8 | 36 | 16-bitx2 | V | V | - | 48SSOP |
| HT95R25 | 16K x16 | 2K x8 | 52 | 16-bitx2 | V | V | - | 64LQFP |

| Part No. | ROM | RAM | I/O | Timer | RTC | DTMF Generator | DTMF Receiver | Package |
|---|---|---|---|---|---|---|---|---|
| HT95R33 | 4K x16 | 1K x8 | 28 | 16-bitx2 | V | V | V | 48SSOP |
| HT95R34 | 8K x16 | 2K x8 | 28 | 16-bitx2 | V | V | V | 48SSOP |
| HT95R35 | 16K x16 | 2K x8 | 44 | 16-bitx3 | V | V | V | 64LQFP |

HOLTEK SEMICONDUCTOR INC.


***Holtek Semiconductor(USA), Inc.***
*46729 Fremont Boulevard, Fremont, CA 94538*
*Tel: (510) 252-9880 Fax: (510) 252-9885*
*E-mail: sales@holtek.com*
*http://www.holtek.com/usa_only.html*

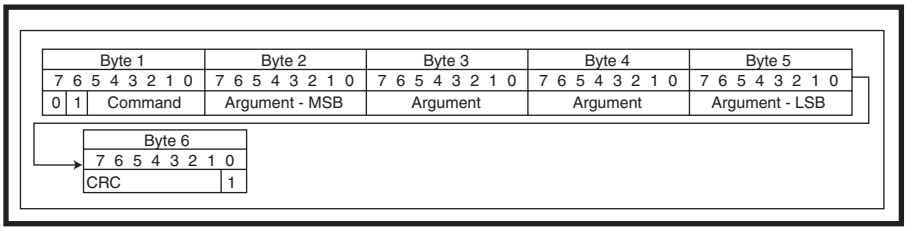



**Figure 3—**SPI-mode SD commands are issued to the card in a 6-byte format.

value. To construct the first byte, you have to OR the SD card command with 0x40.

Note that for this application and the sake of simplicity in explanation, I've ignored CRC checksum implementation. Refer to Table 1 for important SD commands.

## COMMAND RESPONSE

The three command response formats: R1, R2, and R3 depend on each SD command. A byte of response R1 is returned for most of the commands.

The card sends the R1 response token after every command with the exception of SEND_STATUS commands. It is 1 byte long and the most significant bit (MSB) is always set to zero. The other bits are error indications. An error is signaled by a 1. The structure of the R1 format can be seen in Figure 4.

The idle state flag means the card is in the idle state and running the initializing process. Erase reset means an erase sequence was cleared before executing because an out-of-erase sequence command was received. In illegal command, an illegal command code was detected. In communication CRC error, the CRC checks of the last command failed. In erase sequence error, an error in the sequence of erase commands occurred. In address error, a misaligned address that did not match the block length is used in the command. In parameter error, the command's argument (e.g., address, block length) was outside the allowed range for this card.

| Command | Mnemonic | Argument | Reply | Description |
|---|---|---|---|---|
| 0 (0x00) | Go_IDLE_STATE | None | R1 | Resets the SD card |
| 9 (0x09) | SEND_CSD | None | R1 | Sends card-specific data |
| 10 (0x0a) | SEND_CID | None | R1 | Sends card identification |
| 17 (0x11) | READ_SINGLE_BLOCK | Address | R1 | Reads a block at byte address |
| 24 (0x18) | WRITE_BLOCK | Address | R1 | Reads a block at byte address |
| 55 (0x37) | APP_CMD | None | R1 | Prefix for application command |
| 59 (0x3b) | CRC_ON_OFF | Only bit 0 | R1 | Argument sets CRC on (1) or off (0) |
| 41 (0x29) | SEND_OP_COND | None | R1 | Starts card initialization |

**Table 1—**These are selected SD memory card commands.

## INITIALIZE THE CARD IN SPI MODE

During powerup, the SD card defaults to the proprietary SD bus protocol. To switch the card to SPI mode, the following procedure must be followed: After power on, wait a few milliseconds for the power supply to stabilize. Then, set the MOSI and Slave Select (SS) lines to logic high and apply more than 74 pulses to the SCLK line and the card will be able to accept a native command. Set the SS line to logic low. This enables/selects the SD card. Now send CMD0. Reset the SD card (i.e., 0x40 0x00 0x00 0x00 0x00 0x95). The first byte is the command and the last byte is CRC.

This switches the card to SPI mode, but it is still in the IDLE state. Send 0xFF and check the response. If you get 0x01, go to the next step or repeat sending 0xFF for eight to 10 times. If you don't see a 0x01 (R1) response in eight to 10 retries, something is wrong with the setup or the SD card. So let's talk about interfacing with an SD card in SPI mode!

Now that the SD card is in SPI mode, initialize the card by sending command CMD 1 (starts card initialization for low-capacity cards like 256- or 512-MB SD cards or the 1-GB SD card from SanDisk). Set the SS to logic low. This enables/selects the SD card. Send the following 6 bytes: 0x41 0x00 0x00 0x00 0x00 0xFF. This is the format of transmitting the CMD1 command to the SD card. This leads to the SD card's initialization. Send 0xFF and check the response. If the return value is "0xFF," ignore it and repeat sending byte "0xFF" eight to 10 times. If you see 0xFF as the response, something is wrong with the setup or the SD card. If you get 0x00 as the response, continue with the next step. If you get 0x01, the card is still in the idle state. Set the SS to logic high, send 0xFF to the SCLK, and ignore the return value. This is the end of command clocking. At this point, the card is initialized and ready to go for single-block write and single-block read operations.

## INITIALIZATION PROCEDURE

While interfacing an SD card with an AVR microcontroller, SD cards of different makes and capacities were procured: a Transcend (1 GB), a SanDisk (1 GB), a Kingston (512 MB), and a Kingston (256 MB). To support backward compatibility with the MultiMediaCard (MMC), lower capacity 256-MB and 512-MB SD cards from Kingston support the command CMD1 for card initialization by default.

A 1-GB card from SanDisk would not accept the CMD1 command because the card fully complies with the SD card standard "Physical Layer Specification" (Version 2.00). According to these specifications, CMD1 is reserved. It should not be used for SD card initialization.

There is an initialization sequence for a SanDisk 1-GB SD card. A different set of commands needs to be transmitted after CMD0 (i.e., the SD card is in "idle state"). The R1 response is 0x01. Now the initialization sequence is different. Transmit

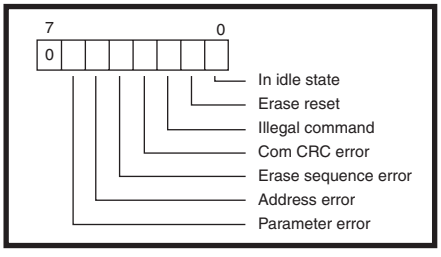



**Figure 4**—This is the R1 response format.

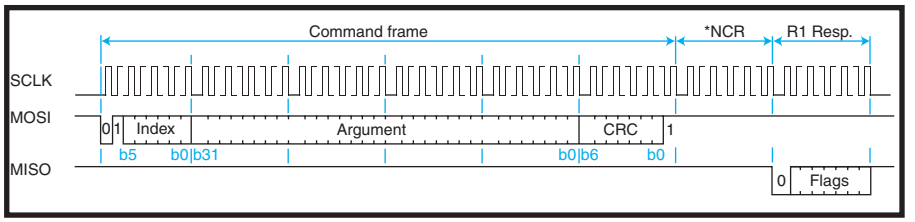



**Figure 5—**This is a timing diagram of SPI lines that connect the SD card with a microcontroller. The command response time (NCR) is the time delay between the transmission of the command packet on the MOSI line and the R1 response received by the microcontroller on the MISO line.

CMD55 is the prefix command for application-specific command ACMDxx. The command format is (0x40 | 0x37), 0x00, 0x00, 0x00, 0x00, 0xFF. Check for the R1 response by transmitting 0xFF eight times (the maximum Ncr response time). The R1 response for CMD55 can be ignored.

Now transmit the application-specific command to start card initialization (ACMD41). The command format is (0x40 | 0x29), 0x00, 0x00, 0x00, 0x00, 0xFF. Check for the R1 response by transmitting 0xFF eight times. The R1 response for ACMD41 should be 0x00. This indicates that the card is initialized and ready to perform a read/write operation.

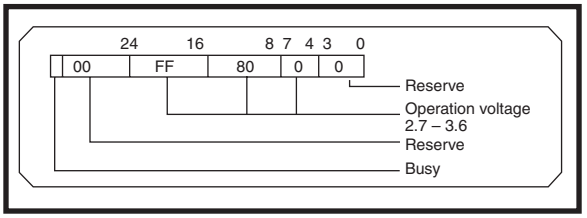



**Figure 6—**This is an R3 response interpretation for the OCR command.

## COMMAND RESPONSE TIME

Some SD cards took a longer time to respond for a command than a specific NCR time. The command response time can vary from 0 to 8 bytes after NCR. So the best option is to transmit 0xFF, after 6-byte command packets and wait for maximum NCR time by transmitting 0xFF eight times (see Figure 5).

## SD CARD REGISTERS

Six registers are defined within the card interface: OCR, CID, CSD, RCA, DSR, and SCR. They can be accessed only by corresponding commands. If the application demands, you can read these registers at this stage. For instance, to read the operating conditions register (OCR), the 32-bit operation conditions register stores the VDD voltage profile of the card. The SD card is capable of executing the voltage recognition procedure with any standard SD card host using operating voltages from 2 to 3.6 V.

Accessing the data in the memory array, however, requires 2.7 to 3.6 V. The OCR shows the voltage range in which the card data can be accessed. Send CMD58. Read OCR. Send byte "0xFF" five times after transmitting CMD58 to read the R3 response. You get an R1 response for every CMD58 command transmitted to the SD card (see Figure 6).

The other two important SD card registers are card identification (CID) register and card specific data (CSD) register. Please refer to the secure digital card product manual from SanDisk for more information.

One or more data blocks will be sent in a write operation from the microcontroller to the SD card and vice-versa in a read operation. A data block is transferred because a data packet consists of Data Token, Data Block, and CRC (see Figure 7).

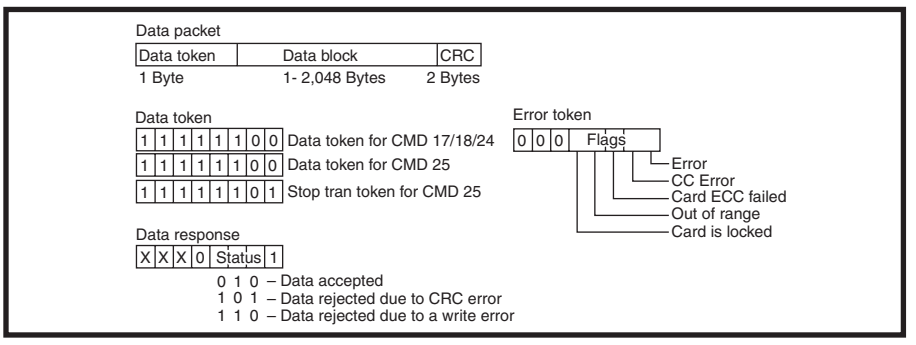



**Figure 7—**This is the structure data packet and data response.

## SINGLE BLOCK WRITE

To write a single block (512 bytes) of data to the SD card, the microcontroller transmits the command CMD24 (0x40 | 0x18) to the SD card. Once the command is accepted by the SD card, it sends an R1 response as 0x00 to the microcontroller (see Figure 8). A microcontroller sends a Data Token 0xFE, followed by a Data Packet of 512 bytes. A card responds with Data Response 0xE5 (i.e., data accepted).

## SINGLE BLOCK READ

To read a single block (512 bytes) of data from the SD card, the microcontroller transmits the command CMD17 (0x40 | 0x11) to the SD card. Once the command is accepted by the SD card, it sends an R1 response as 0x00 to the microcontroller. The command is transmitted along with an argument that specifies the location to start to read in bytes. A read operation is initiated and the read data block will be sent to the microcontroller (see Figure 9). After a valid data token (0xFE) is detected, the microcontroller receives the data block followed by 2 bytes of CRC.

At this point, you have completed a single block write task and a single block read. If you are interested in a multiple-block write and read, refer to the Resources section.

## COMMERCIAL APPLICATION

Markets such as industrial automation, medical, automotive, data collection, and portable computing devices demand more memory in less space. In those markets, the need for reliable and high-performance data storage in a small form factor is increasingly critical.

In industrial automation, a large

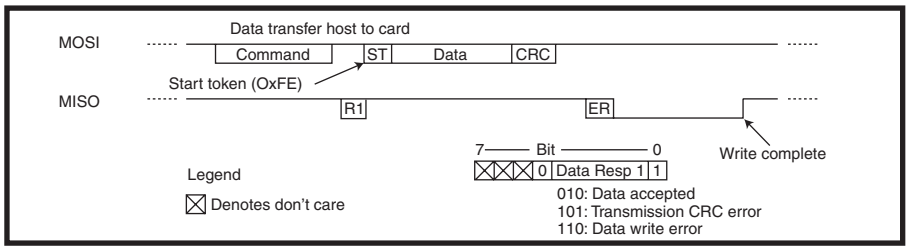



**Figure 8—**A write operation is a form of data transfer from a microcontroller to an SD card.

amount of data (e.g., alarms, trends, and different parameters) is logged for analysis for preventive maintenance or to optimize process operation. One option is a network of field devices. Or if the field device is stand-alone or even if the application is demanding for redundancy in case of network failure, then the field device will have an interface for detachable nonvolatile memory storage (SD card). Consider a particular example: a human machine interface (HMI) works as a stand-alone device and is interfaced with a logic controller. It has to log alarms, trends, and different parameters at a regular interval onto the SD card. The logged data will be used for predictive analysis and preventative maintenance or process optimization.

## FINDINGS

An SD memory card enables you to implement robust, reliable, high-performance, and cost-effective memory storage for today's most demanding industrial applications and portable computing devices. SD cards offer a cost-effective way to store a large amount of data in a removable memory storage device. Its interfacing flexibility and small form factor make it ideal for use with microcontrollers in embedded applications. Multivendor product support and a wide range of applications incorporating SD cards is another plus point.

Combined with the low-cost, low-power ATmega32 and its advanced features (including an on-chip SPI), you can quickly implement an affordable data-logging solution.

*Author's note: I want to thank Yashomani Kolhatkar (Ph.D.) for proofreading this article. He provided a lot of valuable advice.*

*Mandar Bagul (mandar.bagul01@gmail.com) is a lead engineer in the Department of Industrial Electronics at General Electric's India Innovation Center in Hyderabad, India. During his six years at GE, he has worked extensively in the field of embedded product design and development, which includes microcontrollers and programmable logic controllers. Mandar earned a Bachelor's degree in electronics engineering at Shivaji University in Maharashtra, India.*

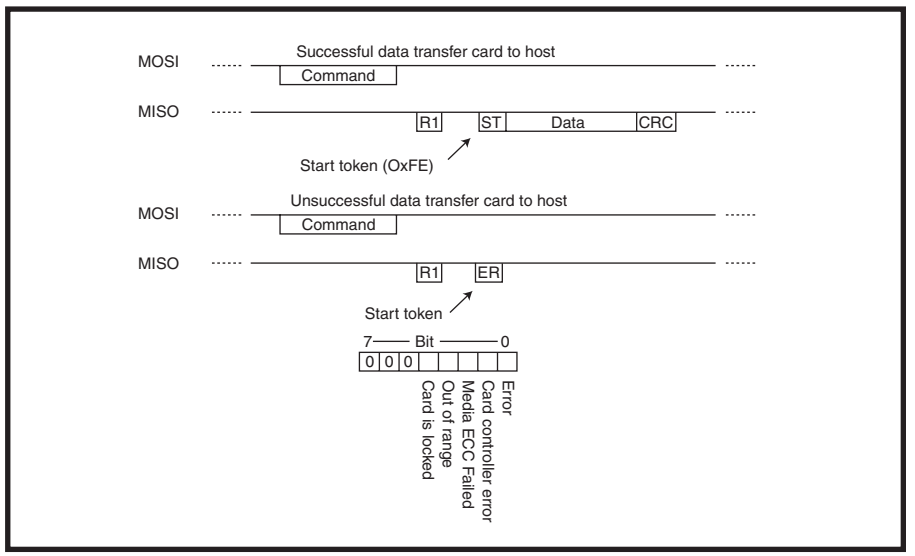



**Figure 9—**A read operation is a form of data transfer from an SD card to a microcontroller.

## PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/221.

## RESOURCES

Atmel Corp., "ATmega32 Datasheet: 8-bit Microcontroller with 32K Bytes In-System Programmable Flash," 2503N, 2008.

AVR freaks, www.avrfreaks.net.

SanDisk Corp., "Secure Digital Card Product Manual," version 1.9, 80-13-00169, 2003.

Santa Clara University, Engineering department, Embedded Systems Resources, www.mil.ufl.edu/~chrisarnold/components/microcontrollerBoard/AVR/avrlib/docs/html/index.html.

SD Card Association, www.sdcard.org.

P. Shinde and T. Love, "Interfacing SD/MMC Cards With TMS320F28xxx DSCs," Texas Instruments, Inc., Application report SPRAAO7, 2007, http://focus.tij.co.jp/jp/lit/an/spraao7/spraao7.pdf.

## SOURCES

**AT29LV512 Flash memory, AT45DB642D DataFlash, ATmega16L-8PU microcontroller, ATmega32L microcontroller, and AVR Studio**
Atmel Corp.
www.atmel.com

**256- and 512-MB SD Cards**
Kingston Technology Corp.
www.kingston.com

**Card connector SD**
3M
www.3m.com

**TS1GSDC SD Memory card**
Transcend Information, Inc.
www.transcendusa.com

**WinAVR**
WinAVR
http://winavr.sourceforge.net





# Real-World NiMH Charging

During bike trips, Ed uses Icom handie-talkies for communication. To keep the radios' batteries working, he uses a three-pack charger that he designed at his workbench. The charger has three 250-mA channels and one 100-mA channel so he can charge different packs.

An early version of the GPS radio interface described in my October column accompanied us on our summer bicycling vacation ("HT Audio, GPS for APRS, and What Works," *Circuit Cellar* 219, 2008). I hadn't finished the replacement battery-pack enclosure, so the circuitry traveled between foam-rubber slabs strapped to the back of the radio, with everything zipped into a pannier on our daughter's bike. Somewhat surprisingly, the lash-up worked fine, although there's a bit of RFI that a full metal jacket should cure.

The radios can use any external voltage from 4.5 to 16 V, but require only 9.6 V to transmit at their rated 5 W. Icom provided 4.8- and 9.6-V NiCd battery packs with ratings in the 650- to 800-mA•h range; the lower-voltage pack produced about 1 W of RF. Despite an absurd level of pampering, all my original Icom packs have long since failed, so our radios have sprouted pigtails terminated with Anderson Power Products Powerpole connectors for external battery packs.

I made six- and eight-cell NiMH packs from ordinary AA cell holders with Powerpole-terminated pigtails. NiMH cells have improved over the years: the current crop sports a 2,600-mA•h rating. The radios draw about 40 mA while idle and I find that the packs last about a week in ordinary use. The GPS interface adds another 70 mA, so its battery packs last a few days.

But on our bicycling vacations we use the radios all day, nearly every day, and that presents a problem: I needed a three-pack charger with a dead-simple interface. Given the vagaries of travel, anything requiring fussy handling just wasn't going to be useful.

In this column, I'll describe the trivially simple charger I lashed together, then explore the history of my battery packs. It turns out that batteries don't always behave the way you expect, even when you expect odd behavior.

## CHARGING COMPLEXITY

As you saw in my April 2007 column, safely charging NiMH cells presents some challenges and, if your application requires high-speed charging, you must deploy rather complex circuitry to eliminate high-temperature failure modes ("Battery Capacity: Charge," *Circuit Cellar* 201). Some straightforward arithmetic reveals the magnitude of the problem.

Most "fast chargers" require an hour or so to bring AA cells up to full charge, but my Energizer CH15MNCP4 charger does the job in 15 minutes for the quartet of 2.2-A•h NiMH AA cells included in its package. The array of components on the circuit board shown in Photo 1 attest to the fact that this is not a dumb charger; a second board holds a high-current DC/DC power converter.

Because NiMH cells have a charging efficiency of about 2/3, injecting a full charge requires 150% of the nominal capacity. Charging a cell in 15 minutes means you must deliver:

$$13.2\ \text{A} = \frac{1.5 \times 2.2\ \text{A} \cdot \text{h}}{0.25\ \text{h}} \qquad [1]$$

The data plate embossed on the back of the charger states that it delivers 7.5 A to each AA cell; it can also charge AA cells at 3.0 A. Working backwards, that gives a charging time of:

$$26\ \text{min.} = \frac{60\ \text{min.}}{1\ \text{h}} \times \frac{1.5 \times 2.2\ \text{A} \cdot \text{h}}{7.5\ \text{A}} \qquad [2]$$

If you ignore the charging efficiency, that's 17.6 minutes. If you also figure that most cells aren't completely discharged, the usual recharge time should be under 15 minutes.

Just for fun, I discharged one of the 2.2 cells at 2.2 A•h down to 0.5 V, well below the usual 0.9-V limit, and then recharged it. The discharge required 54 minutes and delivered 2.0 A•h, which is quite good performance at that current. The recharge took 17 minutes until the green "charged" indicator blinked on and 29 minutes until the



**Photo 1—**The amount of circuitry required for a safe 15-minute AA charger precludes a homebrew solution for larger packs. A second circuit board behind this one converts 16-V, 4-A input power from the wall wart into the 7.5-A current required for high-speed charging.

fan stopped.

The instructions tell you to wait 10 minutes after the green indicator comes on for a full charge, so the actual times are pretty close to my estimates. They are, however, somewhat longer than the advertised "15 minutes," so perhaps it's no surprise that the specs for the same charger now indicate that it ships with four 1.85-A•h AA cells. That lower rating means they recharge in exactly 15 minutes at their nominal capacity and 22 minutes if you allow for charging efficiency.

However, that data applies to a single cell: charging four AA cells simultaneously requires four times the power. The charger's data plate claims an output of 5.6 V at 7.5 A, which is 42 W, and the unit's wall wart delivers 40 W (10 V at 4 A). So, in round numbers, the charger dumps 10 W into each AA cell.

Remember that NiMH cells mark their end-of-charge state by heating up, to the extent that you can actually build a good charger by noting the rapid temperature rise. Missing that signal, however, will allow a rapid charger to overheat the cells or, worse, burst them. That Energizer charger has a fan for a good reason: the cells get rather warm!

The eight-cell packs for our radios use 2.6-A•h cells, so a true "15-minute" charger would require:

$$15.6\ \text{A} = \frac{1.5 \times 2.6\ \text{A} \cdot \text{h}}{0.25\ \text{h}} \qquad [3]$$

The Energizer charger evidently uses power FET switches to connect the cells in series to its power supply and switch them out as they're charged. I'm certain the charger adjusts its nominal 5.6-V supply to match the number of cells in series.

In round numbers, therefore, an eight-cell charger would require a 12-V source that could deliver nearly 16 A: nearly 200 W. Charging three packs at once becomes a 600-W proposition, which seems aggressive even to a guy who built a kilowatt resistance soldering unit.



**Photo 2—**The charger has three 250-mA channels and one 100-mA channel, suited for the different battery packs on our bikes. Those salvaged heatsinks barely keep up with the thermal load.

A 1-h charger draws a quarter of that power and takes four times longer, but that's still a 150-W supply for three packs. I decided that a slow three-pack charger made a whole lot more sense: it can work while I sleep without fuss, bother, fans, or risk of damage.

## DEAD-SIMPLE CHARGING

The dumbest of dumb NiMH chargers consists of a current source set at 10% of the battery's nominal capacity: 260 mA for a 2,600-mA•h battery. After 15 h, more or less, that current will bring a completely dead cell back to life. While further charging isn't recommended, contemporary NiMH AA cells include sufficient catalyst to recombine all the oxygen liberated by low-rate overcharging without venting. The cells become warm, but will not deteriorate due to overheating.

Photo 2 shows what I came up with in the week before we left: three 250-mA and one 100-mA current sources in a box, with Powerpole connectors on short pigtail leads. The thin black wire supplies 14-V power from a surplus laptop charger.

A slightly less dumb charger would incorporate a 15-h (more or less) timer to cut off the current after the maximum allowed charging time. An even brighter charger could monitor the pack temperature to detect the reliable end-of-charge heating. A truly smart charger would monitor the charging voltage to detect the end-of-charge plateau. All of those frills aren't really

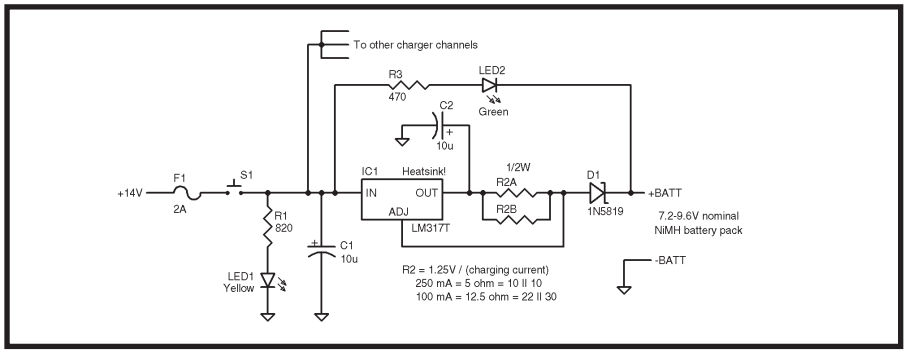



**Figure 1—**Each charger channel uses a simple LM317 current source that dissipates nearly 2 W, making a good heatsink mandatory. A surplus laptop charger provides the 14-VDC supply voltage.

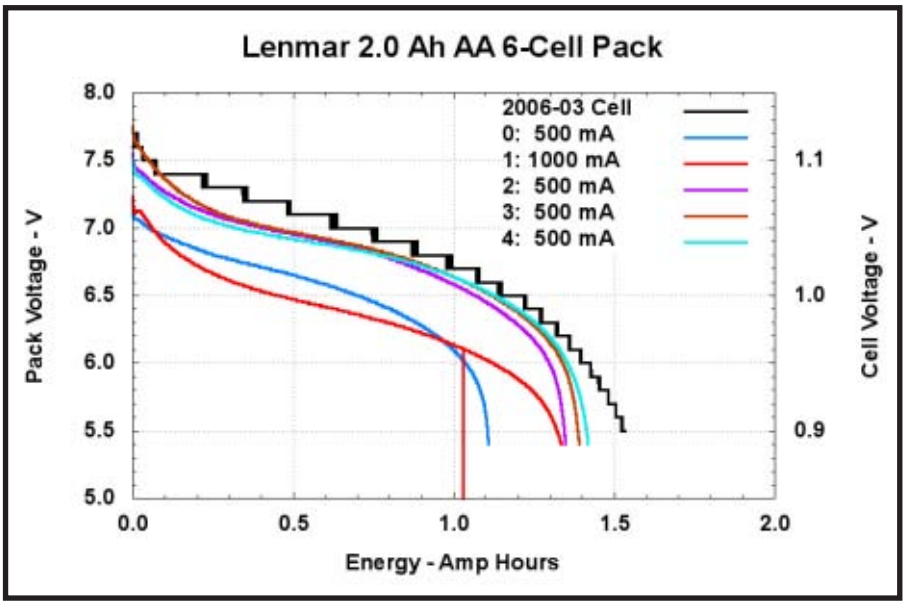



**Figure 2—**The black trace shows how a single cell performed in early 2006. A six-cell battery made from the same group of cells shows the capacity has declined about 10%. The blue trace shows the charge remaining after storage. A data dropout caused the spike in the 1-A trace.

necessary if your plans include daily charging and constant moving.

Figure 1 shows the trivial schematic for a single charger channel. The power source, common to all the channels, is a surplus laptop charger that delivers a regulated 14 V at up to 3.5 A. The 2-A fuse will blow at about twice the maximum total charging current.

A venerable LM317 three-terminal voltage regulator acts as the current source. The regulator adjusts its output voltage to maintain 1.25 V between its Out and Adj terminals, so the parallel combination of R2A and R2B determines the output current:

$$I_{OUT} = \frac{1.25\ V}{R2A \parallel R2B} \quad [4]$$

I picked 250 mA because it requires a 5-Ω resistor that's merely a pair of 10-Ω resistors. The 100-mA channel, chosen to charge the slip-on batteries I rebuilt in the original Icom packs, calls for 22 Ω in parallel with 30 Ω.

The main disadvantage of this circuit comes from its profligate power use: each LM317 regulator must dissipate the product of its load current and the voltage difference across it. The lowest voltage across a six-cell battery can be under 6 V at the start of charging, R2 always drops exactly 1.25 V, and D1 accounts for about 0.5 V: call the total 8 V. The regulator will thus see about 6 V, making its power dissipation:

$$1.5\ W = 6\ V \times 0.25\ A \quad [5]$$

The full-charge voltage for a six-cell pack is about 9 V, so the regulator dissipation eventually drops to:

$$0.8\ W = 3.25\ V \times 0.25\ A \quad [6]$$

I selected a 14-V laptop power supply from my parts heap to ensure the regulator had (just barely) enough voltage headroom for a fully charged eight-cell pack at 12 V.

A rackmount server power supply in my heap disgorged the four TO-220 heatsinks and mounting hardware appearing in Photo 2. They were designed for use in a forced-air environment: the LM317s run *much* hotter than I prefer, despite the rows of vent slots in the aluminum box I used. A thermocouple poked into a blob of heatsink grease on one of the regulators showed that it hits 80°C at the beginning of the charge cycle.

LM317 regulators shut down when their junction temperature reaches 125°C. The TO-220 package has a 4°C/W junction-to-case thermal resistance, so the internal temperature at 1.5 W falls well under that limit. It's too hot for long-term reliability, but at least the regulator won't incinerate itself.

The charging process requires essentially no thought at all. At the end of the day's ride, I unplug three battery packs from our bikes and plug them into the charger's three 250-mA connectors. Because even a full day's use doesn't completely discharge the pack, it's generally ready for use in much less than 15 h. Simply touching each pack tells me when to unplug it as it reaches full charge, but no harm ensues if I do nothing until the next morning.

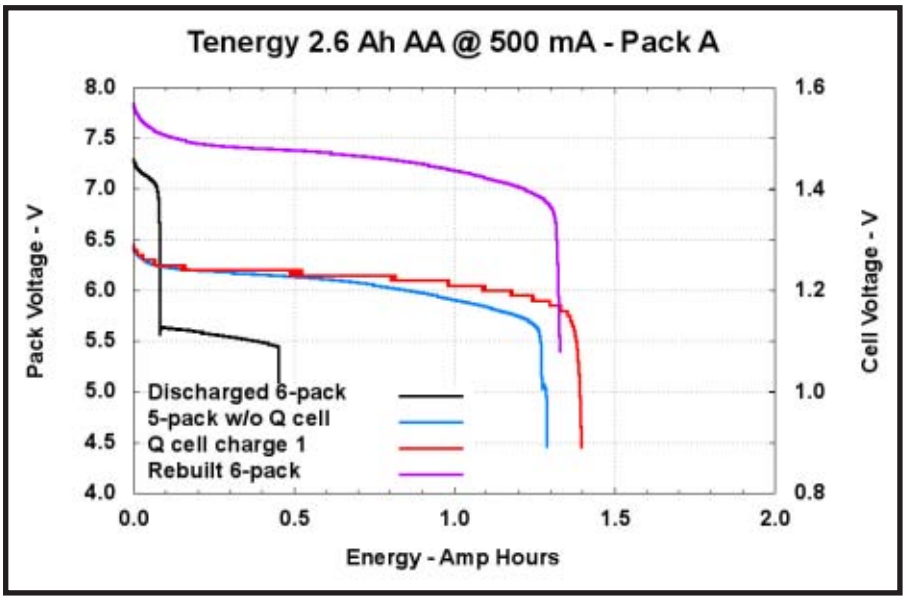



**Figure 3—**The black trace shows that one cell had about 400 mA•h less charge than the others in this six-cell pack. Separately recharging that cell and the rest of the pack restored them to good health, although they're not up to their original spec.




# Experience a Faster Way to Search for Components!

Jameco Electronics' new catalog and enhanced Jameco.com website are two tools that are designed to work together to give electronic professionals faster access to the hottest components in the industry.

Color coded references throughout the catalog assist you in analyzing a wide range of brand choices (from franchise sourced to house brands to factory overruns) offering you more pricing options than you'll see anywhere else. Web codes allow you to quickly jump from catalog to website to view additional specifications, application notes, photos and more products. You'll find exactly what you're looking for without having to wade through hundreds of thousands of products on a complicated website or wielding a twenty pound catalog.

With a flip of the page or a click of the mouse, you have all the tools you need at your fingertips. Reach for Jameco first and order your catalog today.




**Order your web-enabled catalog today!**

JAMECO® ELECTRONICS

***Call 1-800-831-4242 or Visit www.Jameco.com/Speed for the ride of your life!***

TOTAL PHASE

**Beagle USB 480 Protocol Analyzer**




High-speed USB monitor with real-time capture and filtering

**Aardvark I2C/SPI Host Adapter**




I2C, SPI, & GPIO interface to USB emulates master and slave devices

**Beagle I2C/SPI/MDIO Protocol Analyzer**




Non-intrusive I2C, SPI, and MDIO bus monitor with 20 ns resolution

**All products include**: Windows and Linux support, free software, free API with language bindings for C, C++, .NET, Python, VB6, and more.

**Special Offer for Circuit Cellar Readers**

Visit the link below for a special offer on Total Phase tools

www.totalphase.com/offer/CC238953

www.totalphase.com
(408) 850-6500 sales@totalphase.com

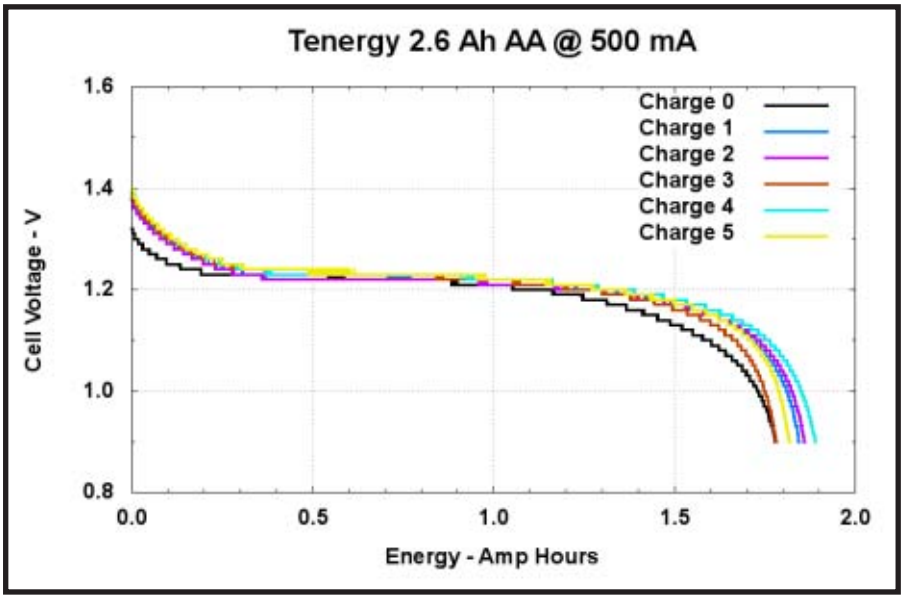



**Figure 4**—A new 2.6-A•h NiMH cell shows little evidence of improving performance during its first several cycles. It provides much less than its rated capacity at a C/5 discharge. I used a C/4 charger for the initial charge and a 4C charger for the others.

A friend points out that building a battery charger with a touch-dependent charge termination requires a certain, um, personality. All I know is: it works for me!

## PACK PERFORMANCE

I've been using AA NiMH battery packs for about three years now and, having both method and motivation for tracking their performance, can relate some interesting real-world data.

Figure 2 shows the results for the oldest cells in my collection: 2.0-A•h cells made by Lenmar Enterprises. As nearly as I can tell, it's reasonable to assume that all non-tier-1 NiMH cells come from anonymous Chinese factories, which makes the brand name largely irrelevant.

In March 2006, one of those cells provided just over 1.5 A•h at 500 mA, its C/4 rate, as shown by the black trace (use the right-side Y scale). The stairstep effect reveals the 10-mV voltage resolution of my West Mountain Radio CBA II battery tester.

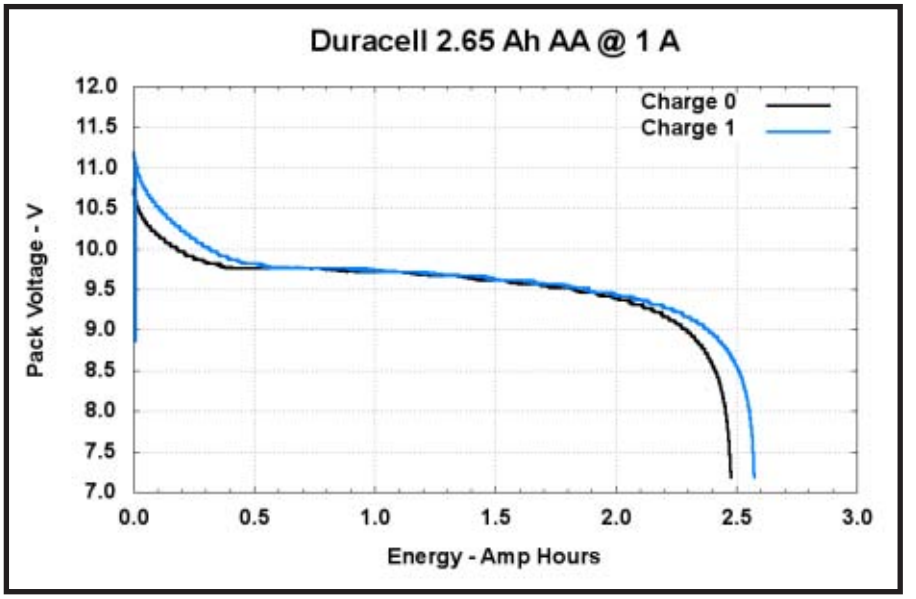



**Figure 5**—An eight-cell battery made from Duracell AA NiMH cells delivers nearly its rated 2.65-A•h capacity, even discharging at 1 A.

A cell's capacity strongly depends on the discharge current, with lower currents extracting more energy from the cell. Different manufacturers may use C/10 and C/20 rates, but typical uses require far more current than that; after all, NiMH cell advertising touts their "high drain" abilities.

The remaining traces show the capacity of a six-pack made from those Lenmar cells, but tested in recent months. These traces use the left-side Y scale, where the CBA II's resolution isn't nearly so obvious.

The blue trace corresponds to the first discharge of the pack after I pulled it from my battery FIFO. I rotate the packs so that that they're neither freshly charged nor totally dead. NiMH cells self-discharge up to about 1% per day, so I estimate this pack has been on the shelf about three weeks.

The red trace shows roughly the same capacity for a 1-A discharge as it does at 500 mA, which is a good sign. However, the other traces (made at 500 mA) show that the cells have lost 10% of their capacity during the last two or three years.

Figure 3 shows the results of a year-old six-pack made from Tenergy 2.6 A•h at 500 mA. I ran the test that produced the black trace shortly after removing the pack from the bike; you can see that one cell was almost totally discharged, even though the remainder of the pack still had nearly 20% of the rated capacity. By definition, the weakest cell goes flat first.

The blue trace shows the other five cells can provide about 1.3 A•h after recharging. The final discharge slope has a step at 5.0 V, revealing the next-weakest cell in the pack. However, they're reasonably well matched.

Recharging the weakest cell, which I labeled Q, produced the red trace (use the right-hand Y scale). Obviously, it was suffering from under-charging, the typical cell failure mode, rather than reaching its end-of-life point.

Putting the pack back together again and charging it for a good long while produces the purple trace. The pack now has about 1.3 A•h of capacity and the final dropoff shows that all the cells go down together.

Adding the GPS interface to our radios increases the current drain, so I bought more AA cells to make up the eight-cell, 9.6-V packs required for 5-W transmitter power and the TinyTrak3+ voltage regulator. Surprisingly enough, the new cells sport the same Tenergy brand and the same 2.6-A•h rating as before. I have no way to tell if they also have the same internal construction.

The traces in Figure 4 document the first six charge-discharge cycles applied to one cell. Common knowledge has it that NiMH capacity increases during the first few cycles, but that's certainly not obvious from the data.

The first discharge, shown by the black trace, had the lowest capacity, but the other five traces don't show any particular progression. In fact, under these conditions, the cell delivers only about 70% of its rated capacity. While that's better than the year-old Tenergy cells, it's not really up to spec.

Figure 5 shows that this need not be the case. Earlier this year I made up an eight-cell battery from Duracell 2.65-A•h cells; the additional 0.05 A•h has no significance outside marketing staff meetings. Although I cannot vouch for how long this pack sat in my rotation, the initial discharge and subsequent cycle deliver nearly the entire rated capacity. Notice that this is at 1.0 A, rather than the 500 mA used in the other tests.

The bottom line, I think, is that there's a difference between Tier-1 batteries and the rest, but that there's an even larger difference due to the care and feeding the cells receive during their lifetime.

## CONTACT RELEASE

When we're not on vacation, we spend about an hour a day riding on errands, to stores, and making other trips around the neighborhood. I change the battery packs on Friday afternoons so that we're all charged up for our longer weekend rides, and carry a spare pack. Under those conditions, all of the packs continue to provide good service, but I plan a major weak-cell purge when I get all three GPS units running.

Even such a simple project requires analog design, power management, and EMI control. When you're handed a new project, remember that it's never as simple as it seems and you can't work on any of the pieces in isolation!

*Ed Nisley is an EE and author in Poughkeepsie, NY. Contact him at ed.nisley@ieee.org with "Circuit Cellar" in the subject to avoid spam filters.*

## PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/221.

## RESOURCES

E. Nisley, "Battery Capacity" *Circuit Cellar* 199 & 201, 2007.

———, "Battery Power, Feeding the Z3801A," *Circuit Cellar* 155, 2003.

PowerStream Technologies, "NiMH Battery Charging Basics," www.powerstream.com/NiMH.htm.

## SOURCES

**Powerpole connectors**
Anderson Power Products
www.andersonpower.com/products/singlepole-connectors.html

**TinyTrak3+ Tracker and GPS receivers**
Byonics
www.byonics.com

**Energizer CH15MNCP4 15-minute charger with four NiMH AA batteries**
Eveready Battery, Inc.
www.energizerbatteries.com/productdetail.asp?device=DIGCAM&prodcode=15505

**2.0-A•h Cells**
Lenmar Enterprises, Inc.
www.lenmar.com
(Distributor: www.batteries.com)

**2.6-A•h Cells**
Tenergy Corp.
www.all-battery.com

**Computerized battery analyzer**
West Mountain Radio
http://westmountainradio.com

BEST USE OF PERIPHERAL PIN SELECT

Microchip 16-Bit Embedded Control Design Contest

# Electronic ID System

## A Display for Text, Photos, and Slide Shows

Jan's electronic ID system is an MCU-based electronic badge that displays color slide shows. Many electronic badges display scrolling text using a matrix of discrete LEDs. This low-power system features a graphical OLED display and can be easily upgraded to handle animation, video, or RF communication.

Most of the time, we want to protect and hide our personal identification information. But, every so often, it helps to be easily identified. For instance, it makes sense to wear a badge at exhibitions, trade shows, and seminars. If you've been to such an event recently, you've probably noticed that electronic badges are increasing in popularity. The electronic badges currently on the market use a matrix of discrete LEDs to display scrolling text (e.g., names and advertisements). Few, if any, of them feature an OLED display. That's why I decided to build my own.

My Mini-Badge design is an electronic badge built around a Microchip Technology PIC24FJ64GA004 microcontroller (see Photo 1). The handy badge displays scrolling text, photos, and color slide shows. In this article, I'll explain how I built the low-power system.

### BADGE BASICS

The Mini-Badge is designed to display a color picture slide show, but you can develop its features depending on your needs. For instance, you can add an RFID for remote identification.

The requirements for portable devices are usually demanding, technically challenging, and often contradictory. When I started this project, I set out to design an aesthetically pleasing, small (lightweight), low-power (long battery life) design that included decent display quality. Because I planned to incorporate a small battery, I needed an easy-to-use user interface that could display custom information. I achieved my goals.

The hardware and software are relatively simple. However, completing

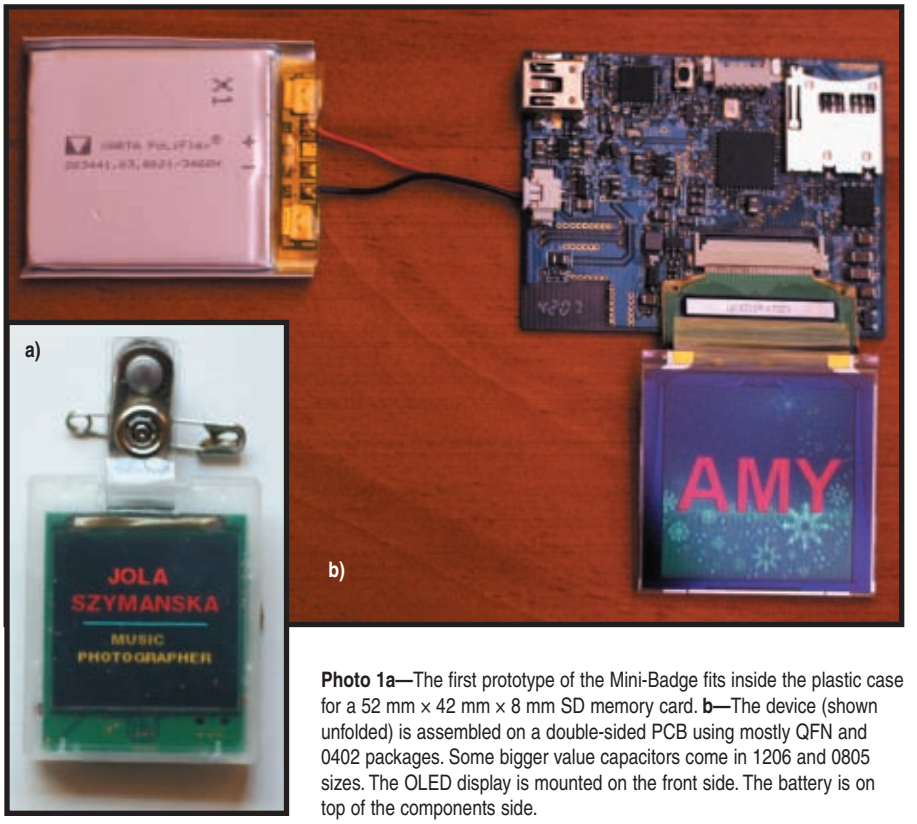



**Photo 1a—**The first prototype of the Mini-Badge fits inside the plastic case for a 52 mm × 42 mm × 8 mm SD memory card. **b—**The device (shown unfolded) is assembled on a double-sided PCB using mostly QFN and 0402 packages. Some bigger value capacitors come in 1206 and 0805 sizes. The OLED display is mounted on the front side. The battery is on top of the components side.

such a project requires sound engineering skills. Small surface-mount devices like 0402 resistors, fine-pitch connectors, and ICs in QFN packages are used. I completed the prototype assembly by hand with a good soldering iron and a magnifying glass. The system's layout is shown in Figure 1 and Figure 2.

## OLED DISPLAY

An organic LED is a relatively new display technology, where thin organic layers serve as the source of light. Its advantages over other technologies include attractive colors and brighter pictures, a lower power consumption than LCDs or TFTs with a backlight, a thin form factor (no backlight is needed), and a fast response time, which can be used for video, a wide viewing angle and high contrast, and a wide temperature range (–30° to 70°C). The technology has some disadvantages such as a limited lifetime, a low yield (especially for bigger sizes), and an uncertain future.

The display in this project is a 1.5" Univision Technology UG-2828GFEFF01 OLED, which has 128 × 128 RGB pixels, 262,144 colors, and a COF package with a built-in controller and SRAM with 132 × 132 × 18 bits (SSD1339 from Solomon Systech). With a built-in controller and display memory, there is no need for an external controller and SRAM. The readily available SD1339 is a good compromise between the physical size, current consumption, unit price, and interface requirements. In addition, you can programmatically select the picture quality with the following color modes: 262,143 colors with 3 bytes per pixel, 65,536 colors with 2 bytes per pixel, and 256 colors with 1 byte per pixel.

As for the hardware interface to the microcontroller, you have several options: parallel interfaces with 8-, 9-, and 18-bit with a Motorola 6800-series-style bus or an Intel 8080-series-style bus and a serial interface with a SPI bus.

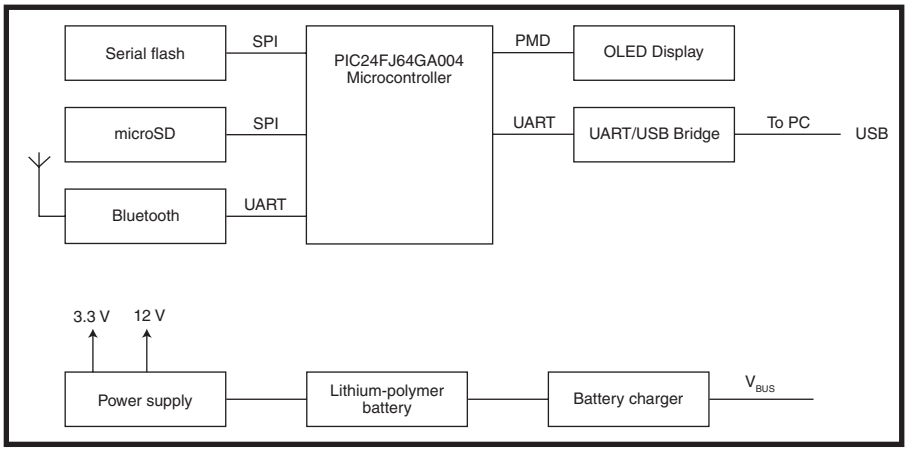



**Figure 1—**The Mini-Badge includes a PIC24FJ64GA004 microcontroller, the OLED display, serial flash memory, and 3.3-V and 9-V power supplies. The additional modules are TransFlash (an alternative mass storage), a USB-to-serial bridge for PC connectivity, a battery charger, power On/Off, switchover, and an optional Bluetooth module.

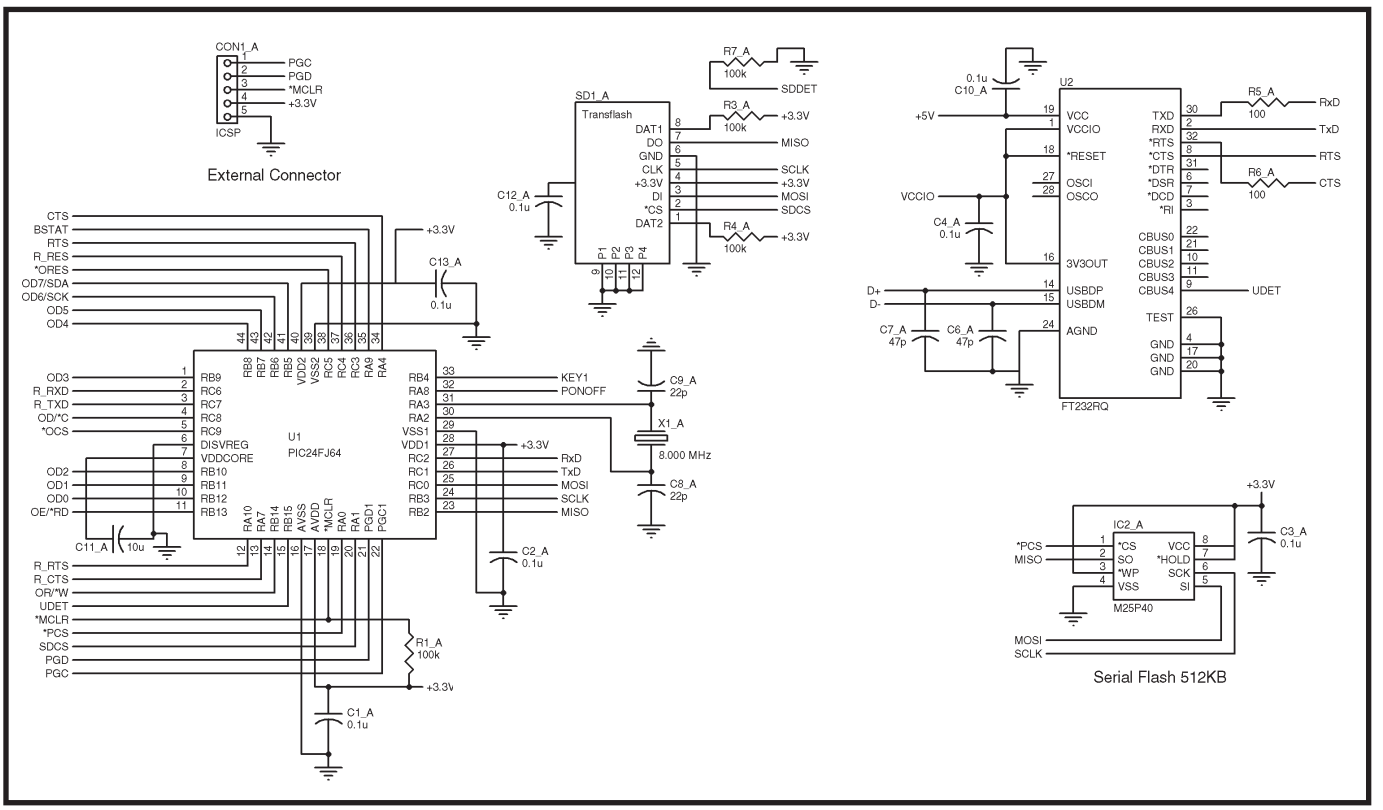



**Figure 2—**A Microchip Technology PIC24F64GA004 is at the heart of the system. Data is stored on an STMicroelectronics M25P40 serial flash memory chip (or the microSD card, TransFlash). Both use a SPI. A Future Technology Devices International FT232RQ serial-to-USB bridge provides communication with a PC (virtual serial port).

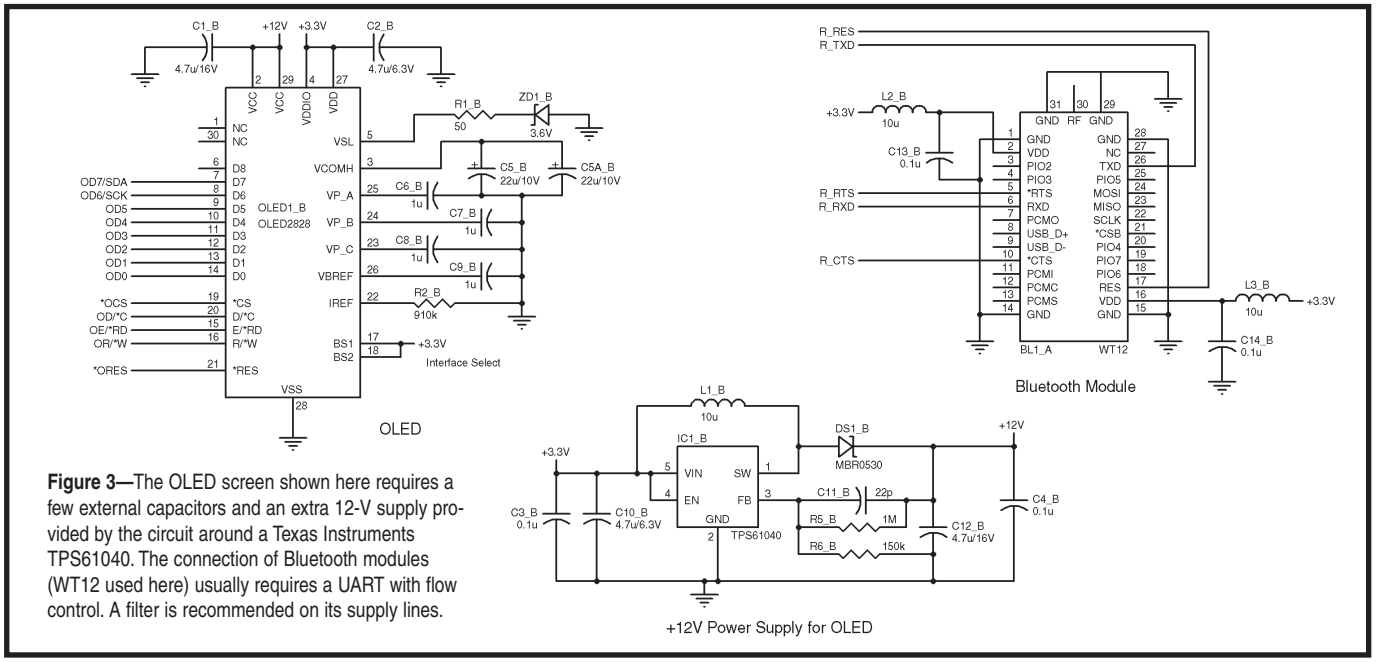



**Figure 3—**The OLED screen shown here requires a few external capacitors and an extra 12-V supply provided by the circuit around a Texas Instruments TPS61040. The connection of Bluetooth modules (WT12 used here) usually requires a UART with flow control. A filter is recommended on its supply lines.

For an 8-bit parallel display interface, after checking the timing specification, the minimum time to write a single byte to the display memory is PWCSL = 60 ns.

With 128 × 128 pixels and 3 bytes per pixel, the total time to fill the display is 128 × 128 × 3 × 60 ns = 49,152 × 60 ns = 2,949,120, or approximately 3 ms. This is a best-case scenario and a rough estimate to show that the display is capable of displaying 30- or 50-fps video (as 1 s/3 ms = 333 fps).

For a SPI, if you want to display 30 fps, you need a SPI clock of 30 × 128 × 128 × 3 × 8 (for 8 bits per byte), or approximately 11.8 MHz. That is, a SPI clock speed that cannot be achieved on any of the current PIC microcontrollers. Also, when you check the SSD1339 datasheet for the SPI minimum $t_{CYCLE}$, it is 250 ns, which corresponds to a 4-MHz maximum SPI clock.

The 8-bit parallel 8080-style interface was a good compromise between speed and complexity. SPI is too slow. In addition, an 18-bit interface is too complex for PCB layout and it takes up too many microcontroller pins

## "PIC"-ING THE BADGE

The system has a serial flash memory that requires a SPI with a clock speed of up to 40 MHz. The microSD memory card SanDisk (TransFlash) also requires a SPI with a clock of up to 25 MHz or more. The Microchip PIC24F Family Reference Manual states that the PIC24*xxx* series with an $F_{CY}$ of 16 MHz has a maximum SPI clock equal to 8 MHz. The PIC33*xxx* series with an $F_{CY}$ of 40 MHz has a maximum SPI clock equal to 10 MHz.

This is not good enough to play video using serial memories like serial flash or TransFlash and a 262,143-color picture quality. However, for the lower quality, it will work with 65,536 and 256 colors because it needs less data leading to a lower SPI clock.

The PIC's (8080-like, 8-bit mode) parallel master port gives it an advantage over GPIO. The USB interface or the UART together with an external UART-USB bridge can be used to interface to a PC. The second UART is handy for a Bluetooth interface (see Figure 3).

The PIC24FJ64GA004 microcontroller is a compromise between the speed of execution, power consumption, available peripherals, and more. It has more than enough program flash memory and data RAM, so it leaves room for expansion.

The GPIO, PMP, UARTs, SPIs, and timer peripherals are used. There are some unused peripherals, which are disabled to save power. It also comes in a space-saving QFN package.

One of the nice features is a peripheral pin select that enables you to assign pin numbers to some peripheral functions. It can ease the PCB layout process and make it possible to implement functional changes without changing the hardware.

## MASS STORAGE

The STMicroelectronics M25P40 serial flash and the microSD memory card (TransFlash) are used to store images and configurations. The popular SD memory card is physically too big. Both devices (the serial flash memory chip and microSD memory card) use similar hardware SPIs. The file system is FAT16, the configuration file is in XML, and the pictures are BMPs. The current firmware recognizes only pictures with 128 × 128 pixels saved as 24-bit BMPs, giving 49,206 bytes per file without any compression. Figure 4 is a brief description of the bitmap format as it is used for the project.

The basic version of the Mini-Badge described here has no FAT table and no directory. Instead, the files are stored and accessed from a fixed location within the serial flash memory chip. Because the sector size for the M25P40 is 64 KB and the file size is 49,206 bytes, there is one file per sector, which makes addressing easy. It might look like a waste of memory, but this arrangement provides software

RENESAS SuperH

# Scalability reaches 1Mbyte with 100MHz single cycle embedded Flash.

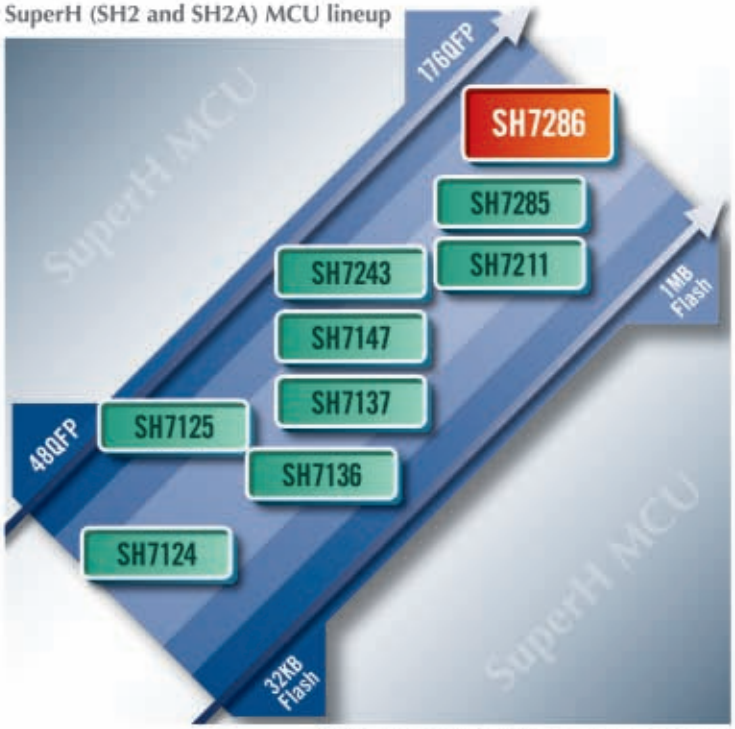
Extensive internal peripherals including USB 2.0 Full Speed, multi-function timer unit, and 12-bit A/D converter

## Renesas Technology

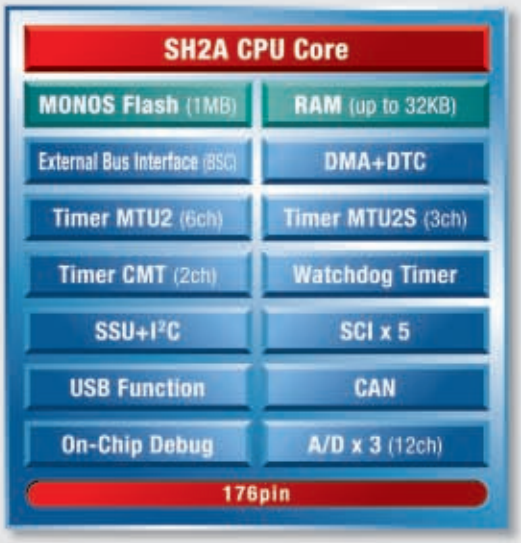
### No. 1* supplier of microcontrollers in the world

proudly presents the addition of the SH7286 to the SuperH Family of devices. The SH-2A core is geared towards systems that demand real-time, high-precision control and require a combination of a high-performance CPU and high-speed flash. This family is the ideal scalable solution for a wide range of 32-bit high-performance applications. The scalable family means that no matter what your application, there is probably a SuperH Family device to suit your requirements. And as your application grows, there will still be a SuperH device that fulfils your needs. With superscalar performance, high-speed access to on-chip flash memory, and up to 200MHz (360DMIPS) CPU operation for SH-2A devices, the SuperH Family allows you to incorporate all of your brilliant ideas into super high-performance applications.

SuperH (SH2 and SH2A) MCU lineup




**HOT Products** SH7286




### Top Reasons to Select SH7286

- Up to 2 instructions per clock cycle
- Single cycle flash access up to 100MHz
- 6 cycle interrupt response time
- 2 x 16-bit Multi-Function Timers for advanced motor control
- Automatic output disable for fast motor shutdown
- 12-bit A/D with 3 Sample/Hold units and 1us conversion time
- 8ch of DMA with 4 external inputs
- CAN controller with 16 message buffers
- External Bus Interface (BSC) for SRAM, SDRAM, and Burst ROM interface

*Source: Gartner "Semiconductor Applications Worldwide Annual Market Share: Database" Hiroyuki Shimizu, 27 March 2008, GJ08218 ＊This is 2007 ranking

#1MCU REACH FURTHER

## Get Started Today -

Go online and register to access special promotions on starter kits.*

**www.america.renesas.com/ReachSH/d**

*Only available to customers in North and South America




Starter Kit for SH7286: R0K572867S000BE; includes USB debugger

Renesas Technology Corp.

Everywhere you imagine. RENESAS

compatibility between different hardware versions—the one using serial flash with a limited file system and the one using TransFlash with a full FAT16 file system implementation.

The XML file is located at address 0 and the seven images are located at the addresses n × 0x10000 (n = 1..7). There are eight pages (0, 1, 2, 3, 4, 5, 6, 7) for the M25P40 serial flash memory, and n is a place holder for the page number of serial flash. The XML file is at address 0x00000 (the hexadecimal notation for 0). So, image 1 is at address 0x10000 (the hexadecimal notation of 65,536), image 7 is at address 0x70000 (the hexadecimal notation of 458,752), and so on. The images are in BMP

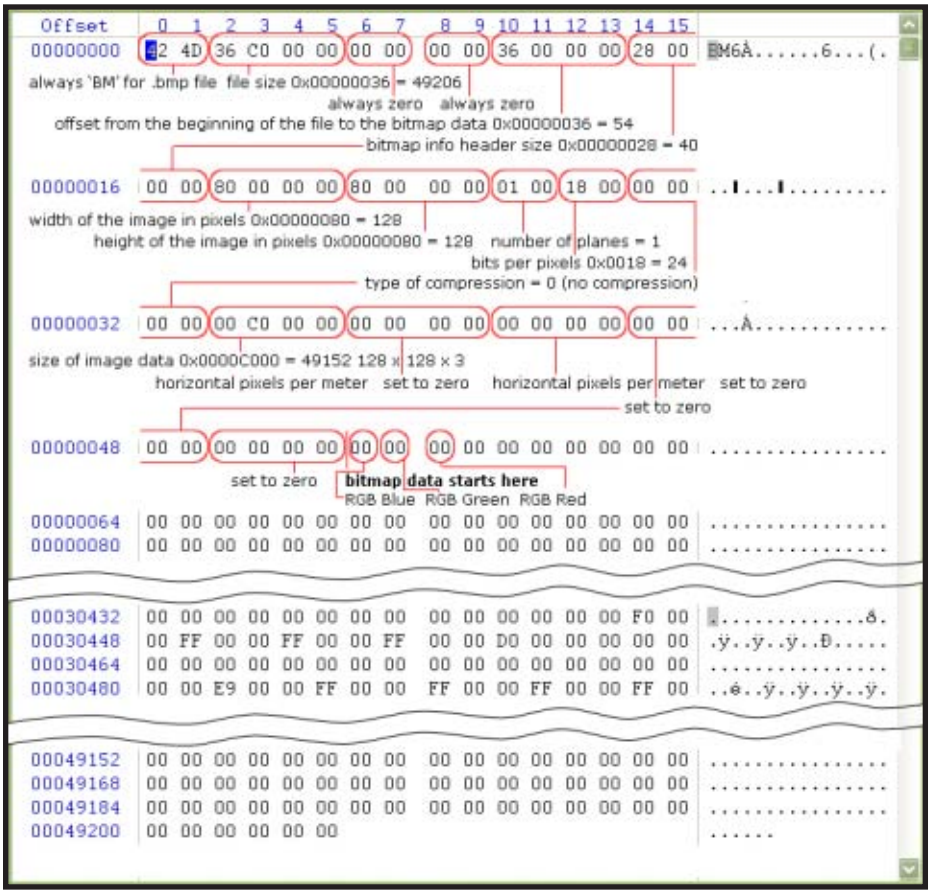



**Figure 4—**The is a BMP file format example, as seen in the hex editor. Every pixel is represented by 3 bytes of RGB (24 bits per pixel). Only 6 bits of each byte are used by the OLED in 18-bit mode. (See the double bit shift operation in the code to ignore 2 LSB.) The linear mapping of BMP data into display RAM makes for easy the loading of BMP files into the OLED display.



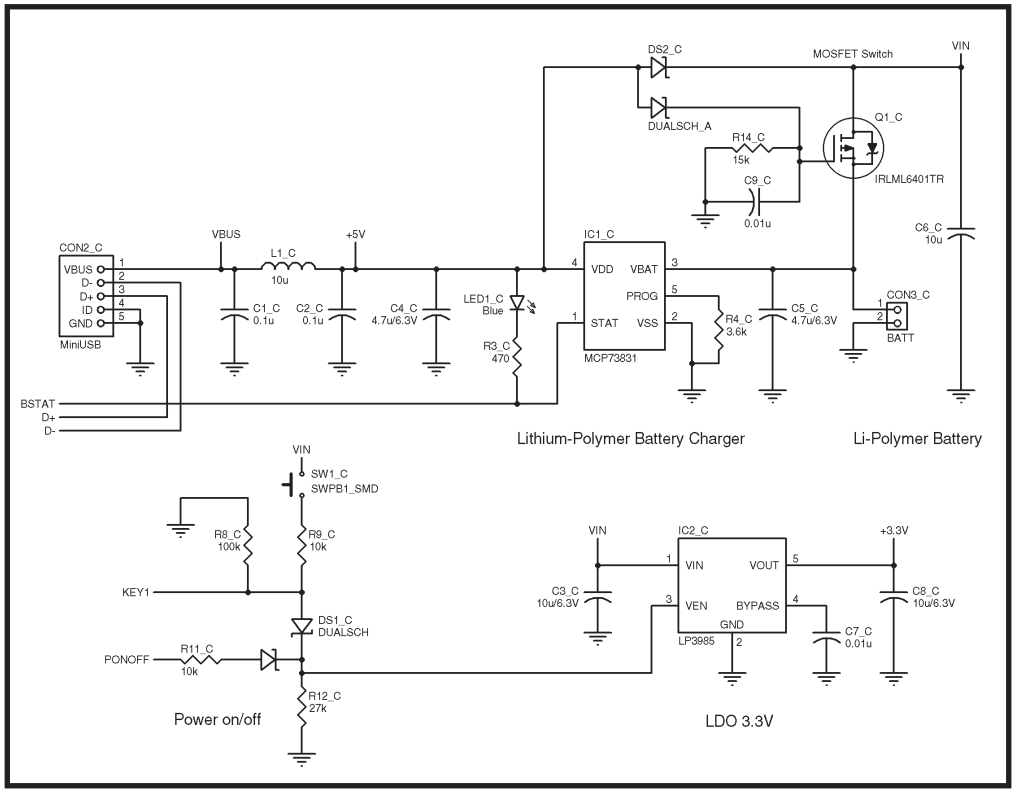




**Figure 5**—The main ultra-low-dropout regulator provides a 3.3-V supply. A lithium polymer battery is charged through USB using a Microchip Technology MCP73831 charging controller. The MOSFET diodes combination function is to switch between different power sources. (When USB is available, the battery is charged and the circuit is supplied from USB.) There is a push button with logic to power on and off.

format. So, for a 128 × 128 bitmap with a 24-bit quality, the size of a file is 49,206 bytes. The XML and BMP files are stored within the serial flash memory (and microSD memory card) without any modification or compression.

The XML and BMP files can be copied from a PC into the microSD card; while for the serial flash, they are downloaded through the USB link. A virtual serial port on your PC is created when the Future Technology Devices International FTDI232RQ USB UART IC is connected and enumerated. (The drivers for it are built-in for Windows XP.)

## PC INTERFACE

The FT232RQ is used as a USB-to-UART bridge. One side of it is connected to the microcontroller's UART TxD and RxD lines. Two




general-purpose port lines are used as RTS and CTS for flow control. The FT232RQ acts as a virtual serial port, so you can use HyperTerminal to test the communication. The hardware flow control is implemented using RTS and CTS signals to cater to a proper timing.

The 460,800 data rate works well even with a data rate difference due to the limitation of possible microcontroller settings. To check the data integrity, a simple protocol is implemented, as described in the art7_SerialProtocol.doc file posted on the *Circuit Cellar* FTP site. To download the files, I use terminal software (HyperTerminal) on a PC. Refer to the System Tests section for more details.

**Listing 1**—Use the following code to sense the USB connection through Interrupt on Pin Change. First Interrupt on change is initialized and enabled. After detecting the USB connection/disconnection through the Interrupt, the UART module connected to the external USB-UART bridge is turned on or off accordingly to save power.

```
        // RP15/RB15/CN11 is USB SLEEP# (to detect USB)
        _TRISB15 = 1;  //
        _RB15 = 0;
        CNEN1bits.CN11IE = 1;         // INT on change on pin
        IFS1bits.CNIF = 0; // clear the flag
        IEC1bits.CNIE = 1; // enable INT on change
// To process the corresponding INT use the code
void _ISR _CNInterrupt(void)
{
        unsigned int temp;
        // read port
        temp = PORTB;
        _CNIF = 0; // clear the flag
        set_timer(OS_USB_ONOFF, SEC_1, check_usb);
}
```

## POWER SUPPLY

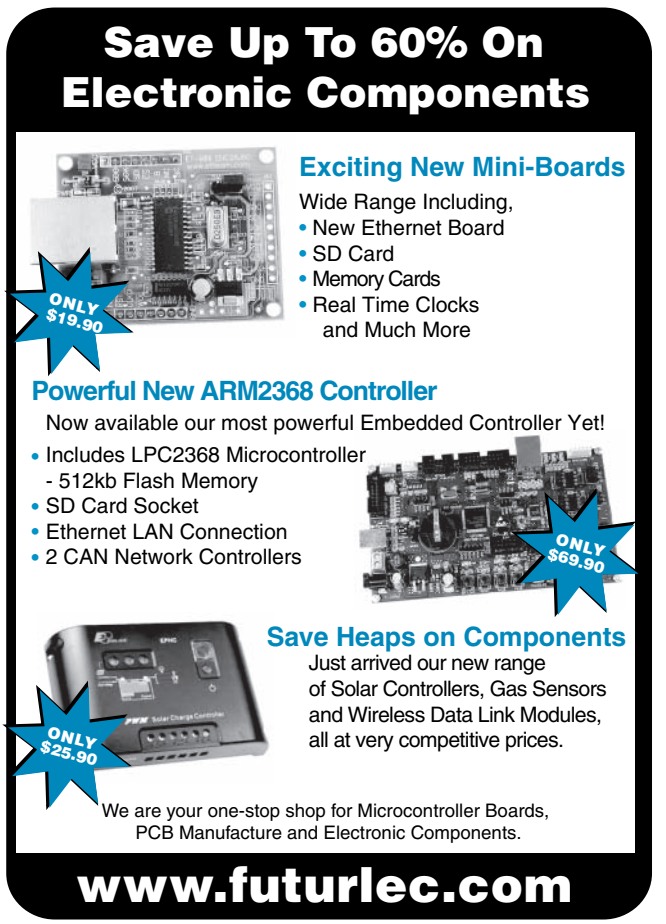
The power source is a Varta PLF263441B lithium polymer battery (see Figure 5). The battery is rechargeable and is charged through the USB. The charging circuit is based on a Microchip Technology MCP73831 charging IC. Only the current programming resistor needs to be set depending on your battery capacity. Check the device specification for the formula and the selection criteria.

The device has a standard Mini-USB type-B connector for easy connection to a PC to charge the battery and download pictures. There is a small push button on the back of the device for turning the power on and off.

The ultra-low-dropout linear regulator provides the 3.3-V supply for most of the circuitry. I used a little switcher, but it proved that the efficiency for ultra-low current is comparable or better when using a linear regulator (instead of a switching one). With the more complex circuit for a switching regulator and more noise, it has the advantage over the linear regulator for







**Listing 2**—Initialization is achieved through a table of parameters. The benefit of this approach is that any changes can be done by simply modifying only the table.

```
const unsigned char oled_init_tab [36][2]= {
{0, 0xa0},          // command, 0xa0 Set Re-map / Color Depth
                    // 1010 0000
{1, 0xb2},          // data, 0xbc 262K_Color 8 Bit
//{1, 0x70},        // data, 0x70 65K_Color 8 Bit

{0, 0xa1},          // command, 0xa1 Display_Start_Line
{1, 0x00},          // data, 0x00

{0, 0xa2},          // command, 0xa2 Display_Offset
{1, 0x80},          // data, 0x80

{0, 0xa6},          // command, 0xa6 Display_Mode

{0, 0xad},          // command, 0xad Master_Conf
{1, 0x8e},          // data, 0x8e

{0, 0xb0},          // command, 0xb0 Power_Saving
{1, 0x05},          // data, 0x05

{0, 0xb1},          // command, 0xb1 Reset_Pre_charge
{1, 0x11},          // data, 0x11

{0, 0xb3},          // command, 0xb3 Osc_Freq
{1, 0xf0},          // data, 0xf0

{0, 0xbb},          // command, 0xbb Precharge_Color
{1, 0x1c},          // data, 0x1c
{1, 0x1c},          // data, 0x1c
{1, 0x1c},          // data, 0x1c

{0, 0xbe},          // command, 0xbe VCOMH
{1, 0x1f},          // data, 0x1f

{0, 0xc1},          // command, 0xc1 Contrast_Current
{1, 0x80},          // was 0xaa data, colour A
{1, 0x80},          // was 0xb4 data, colour B
{1, 0x80},          // was 0xc8 data, colour C

{0, 0xc7},          // command, 0xc7 Master_Contrast
{1, 0x0c},          // was 0x0f data, 0x0f

{0, 0xca},          // command, 0xca Mux_Ratio
{1, 0x7f},          // data, 127+1

{0, 0xaf},          // command, 0xaf Display_ON

{0, 0x15},          // set column range
{1, 0x00},          // 0 - 127
{1, 127},

{0, 0x75},          // set row range
{1, 0x00},          // 0 - 127
{1, 127}
};
/*************************************************************/
/*  Name           : oled_init                               */
/*  Parameters     : none                                    */
/*  Returns        : none                                    */
/*  Function       : initializes OLED after power on         */
/*-------------------------------------------------------------*/
void oled_init(void)
{
    unsigned char i;

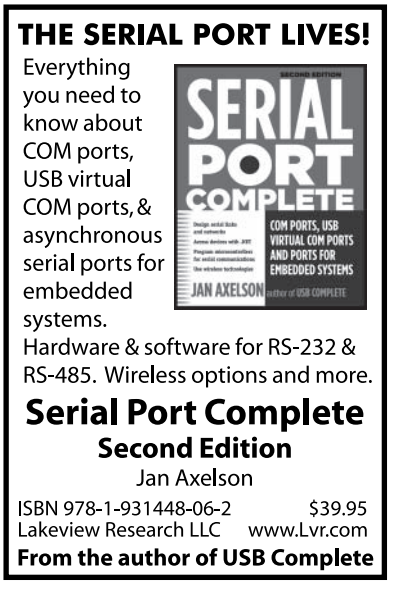
    oled_reset();

    for(i = 0; i < 36; i++)
    {
        if(oled_init_tab[i][0] == 0)
            oled_writeCmd(oled_init_tab[i][1]);
        else
            oled_writeData(oled_init_tab[i][1]);
    }
    oled_cls();
}
/*-------------------------------------------------------------*/
/* End of Function: oled_init                                  */
/*************************************************************/
```







**Listing 3—**The response to the view file command (`$PBVF,010000*13`) has the following format. It can be used to debug your design because it examines the contents of serial flash memory. The data is displayed in ASCII and hex formats.

```
$PBVF,View File Command

00000000     <  ?  x  m  l     v  e  r  s  i  o  n  =  "  1  <?xml version="1
00000010     .  0  "  ?  >  0D 0A <  P  r  e  s  e  n  t  a  .0"?>..<Presenta
00000020     t  i  o  n  >  0D 0A    <  S  c  r  e  e  n  1  tion>.. <Screen1
00000030     >  0D 0A       <  C  l  s  >  t  r  u  e  <  /  >..  <Cls>true</
00000040     C  l  s  >  0D 0A       <  D  e  l  a  y  >  5  Cls>..  <Delay>5
00000050     0  0  <  /  D  e  l  a  y  >  0D 0A       <  T  00</Delay>..  <T
00000060     y  p  e  >  i  m  a  g  e  <  /  T  y  p  e  >  ype>image</Type>
00000070     0D 0A       <  S  c  r  o  l  l  >  l  e  f  t  ..  <Scroll>left
00000080     <  /  S  c  r  o  l  l  >  0D 0A       <  F  i  </Scroll>..  <Fi
00000090     l  e  >  0  x  1  0  0  0  0  <  /  F  i  l  e  le>0x10000</File
000000A0     >        0D 0A       <  D  e  l  a  y  >  3  0  >  ..  <Delay>30
000000B0     0  0  <  /  D  e  l  a  y  >  0D 0A    <  /  S  00</Delay>.. </S
000000C0     c  r  e  e  n  1  >  0D 0A    <  S  c  r  e  e  creen1>.. <Scree
000000D0     n  2  >  0D 0A       <  C  l  s  >  t  r  u  e  n2>..  <Cls>true
000000E0     <  /  C  l  s  >  0D 0A       <  D  e  l  a  y  </Cls>..  <Delay
000000F0     >  5  0  0  <  /  D  e  l  a  y  >  0D 0A       >500</Delay>..
end of view
```

higher-current applications. The power switchover between USB and the battery is achieved through the MOSFET and diodes circuit.

Part of the power supply includes a 12-V low-current supply circuit built around a Texas Instruments TPS61040 switch boost converter. The voltage should actually be between 8 and 12 V. I use 9 V, which can be set by resistors R5_B and R6_B (see Figure 5).

To save power, the circuits sense USB connection through one of the microcontroller's lines capable of interrupt on pin change—specifically RP15/RB15/CN11. After detecting USB connection/disconnection, the firmware turns on or off the UART module connected to an external USB-UART bridge accordingly for power saving. To initialize the feature refer to Listing 1.

The OLED display doesn't like static pictures, but rather fast-changing ones. There are periods when the screen is blank when you perform a slide show on the device. During that time, the microcontroller goes into low-power mode.

## SOFTWARE

The embedded software was developed using Microchip's MPLAB IDE and C30 compiler. The ICD2 was used to program and debug the code. Because I had to use a small connector for in-circuit programming, I had to build an adapter between it and a standard ICD2 connector.

The images are downloaded into the serial flash memory or copied into the microSD memory card together with the script file. I use the XML document to define the slide show. The use of XML is due to its simplicity, popularity, and compatibility with different platforms.

The embedded software works as a simple non-preemptive scheduler. The sources contain a number of modules corresponding to the functionality. For example, there are sflash.c and sflash.h source files for the serial flash module. Refer to the files posted on the *Circuit Cellar* FTP site for details. I will briefly explain some software functionality of major modules.

The badge.c file posted on the *Circuit Cellar* FTP site is the main one for the project and contains the power on initialization and the scheduler for system tasks. OLED software contains the driver and library for the OLED display. I use a table of parameters to initialize the display using the manufacturer's recommendation and supplied examples. When checking the initialization code in Listing 2, you will see that there is no waiting or delays between writings typical for a classic character-based LCD.

After that, write to the display memory through the parallel master port. When checking the OLED controller's specifications, notice that there are higher-level graphical functions available, so you might consider using them in your project.

For serial communication, I use the buffered, interrupt-driven code, which is always a good way for fast, responsive systems. I have to use the flow control RTS/CTS mainly because the fast transmission from the PC has to be held while the serial flash memory is reprogrammed or erased (slower process). Although STMicroelectronics provides a library for its range of serial flash memory chips (available at www.st.com), I used my own code, which might be simpler, but it only covers one device (M25P40).

## SYSTEM TESTS

As with any hardware prototype, make sure that the assembly process is neat and clean. Perform your usual inspections by looking at solder quality, unwanted solder bridges, and more.

When building a prototype, the idea is to follow, if possible, a specific sequence. First, assemble the power supply circuitry and check the output voltages and the idle current consumption. The operation of the On/Off button is a combination of hardware and software. If there is no software (blank device) or when changing the software, hold the button pressed. A better way is to solder the temporary link between the pins on the push On/Off button (SW1_C) to turn the power on permanently while you are programming and testing. Disconnect it when the test is finished.

You need a simple adapter to connect

your ISP programmer to CON1_A. Assemble the microcontroller circuitry, open the MPLAB IDE application and the project work space, connect your ISP programmer, and program the microcontroller. Now you can assemble the rest of the circuit either gradually or in one go. Do not connect the OLED display yet. Because it has a zero insertion force (ZIF) socket, you can connect or disconnect it at any time. (Make sure the power is disconnected when you do it.)

Connect the USB cable to a PC. The enumeration process should happen and the FTDI232R should be recognized automatically. On Windows XP, the driver is built-in, so you don't need to install it.

Next, open HyperTerminal (or another terminal application). Then, select the newly created virtual serial port number with the 460800, 8, N, 1, hardware flow control settings. Use menu Transfer>SendTextFile. Then, browse to your downloaded test_files directory and select test_0.txt. You should see a response like this:

```
$PBUB,EWWWWw$BPUB*0B
```

In this step, program the serial flash memory with a configuration file. Repeat the aforementioned procedure, but select PBVF_0.txt this time. You should see a response like the one in Listing 3.

Next, press any key to examine the contents of serial flash memory. Then press the ESC key to receive the response. In this step, program the first picture into serial flash memory. In a similar way, you can download other test files, if needed. Disconnect the power and connect the screen (OLED display). When you reconnect the power, the Mini-Badge system should be operating.

## SMART BADGE

Investigating new technologies and ideas is always a great experience. The purpose of the project was to create a new device by taking advantage of a few electronic technologies. Microchip microcontrollers are useful for intelligent devices. One of the unique features introduced in 16-bit PIC devices is peripheral pin select, which enables the programmable assignment of a peripheral function to a pin number. They have other supporting devices like an integrated intelligent battery charger and the ultra-low dropout regulator. The OLED displays have an impressive picture quality combined with a small size and low-power consumption. The lithium polymer batteries come in small sizes and big capacities. The availability of components in small packages enables you to create small, lightweight devices.

An electronic badge like mine is more than a nice gadget. You can easily make it an intelligent device. It already has a hardware provision for a Bluetooth module, so adding wireless connectivity would be easy. That would make it useful for use in remote identification systems. If you interface some sensors to it, it could become a node in a wireless sensor network. With your mobile phone in one pocket and a PDA in the other one, do you need another device? The answer will be yes if it does what other devices can't do.

*Jan Szymanski (janek@bigpond.net.au) is an electronics engineer who specializes in embedded systems design. He runs Cherry Microsystems, a consulting company in Australia. In his spare time, Jan enjoys music and sports.*

## PROJECT FILES

To download code and additional files, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/221.

## RESOURCES

Microchip Technology, Inc., "MCP73831: Miniature Single Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controller," DS21984A, 2005.

———, "PIC24F Family Reference Manual," DS39699B, 2007.

———, "TC2014/2015/2185: 50 mA, 100 mA, 150 mA CMOS LDOs with Shutdown and Reference Bypass," DS21662E, 2006.

National Semiconductor Corp., "LP3985: Micropower, 150mA Low-Noise Ultra Low-Dropout CMOS Voltage Regulator," DS101364, 2005.

Solomon Systech, "SSD1339: Advance Information: 132RGB x 132 with 2 Smart Icon Lines Dot Matrix OLED/PLED Segment/Common Driver with Controller," 2006.

SparkFun Electronics, "UG2828GFEFF01 OLED Display Module Datasheet, Controller Specifications, and Example Code," Univision Technology, Inc., www.sparkfun.com/commerce/product_info.php?products_id=712.

Texas Instruments, Inc., "TPS61040: Low-Power DC/DC Boost Converter in SOT-23 and SON Packages," SLVS413D, 2007.

Varta Microbattery, "PLF 263441 B PoLiFlex Battery datasheet," 2005, www.elektronik.ropla.eu/pdf/stock/vmb/plf263441.pdf.

## SOURCES

**WT12 Bluetooth module**
Bluegiga Technologies
www.bluegiga.com

**FTDI232RQ USB UART IC**
Future Technology Devices International
www.ftdichip.com

**MCP73831 Charge management controller, MPLAB C30 C compiler, and PIC24FJ64GA004 microcontroller, and TC2015 LDO**
Microchip Technology, Inc.
www.microchip.com

**SSD1339 OLED Driver**
Solomon Systech
www.solomon-systech.com

**M25P40 Serial flash memory chip**
STMicroelectronics
www.st.com

**TPS61040 Switch boost converter**
Texas Instruments, Inc.
www.ti.com

**UG-2828GFEFF01 OLED**
Univision Technology, Inc.
www.univision.com.tw

**PLF263441B Lithium polymer battery**
Varta Microbattery
www.varta.com





by Robert Lacoste

# PID Control Without Math

If you need to design an optimized control loop for a hardware control problem, consider trying a proportional integral derivative (PID) controller. In this article, Robert explains that PID regulations are simple to code, wire, and tune to find the results you're looking for.

Welcome back to the Darker Side. The vast majority of real-world systems are based around feedback loops, which are used to manipulate the inputs to a system to obtain a desired effect on its outputs in a controlled way. Your DVD player uses a feedback loop to drive its spinning motor for a precise rotation speed. Your mobile phone has a feedback loop to adjust its transmit power to the required level. And, of course, your car has plenty of feedback loops (not only in the cruise control module). Systems incorporating feedback work even if the relationship between the desired value (e.g., the spinning speed of a motor) and the controlled one (in that case, the current applied to the motor) is not straightforward. Delay effects, inertia, and nonlinearities make life more interesting, and so do external conditions. For example, a cruise control device needs to apply more torque on the motor if you are climbing a hill. It should keep overshoots as small as possible to avoid getting a fine after you get to the top of a hill.

On the theoretical side, these problems have been well-known for years and are covered by the Control System theory. The theory explains how to design an optimized control loop for a given problem, at least if the problem is well formalized. The theory—which started with the works of James Clerk Maxwell in 1868, followed by a lot of inspired mathematicians, such as Alexander Lyapunov and Harry Nyquist, to name a few—is not simple. The first pages of control system books usually start with mathematical notions like pole placement, Z-transforms, and sampling theorems, which nonspecialists may find difficult to deal with, even if good books have adopted a more engineer-oriented approach (e.g., Tim Wescott's *Applied Control Theory for Embedded Systems*, Newnes, 2006). Should you give up? No. Fortunately, some classical control algorithms are applicable to a lot of problems. More importantly, for design guys like us, they are easy to implement in firmware or hardware. The most usual is the ubiquitous proportional integral derivative (PID) control, which has one interesting characteristic: it is

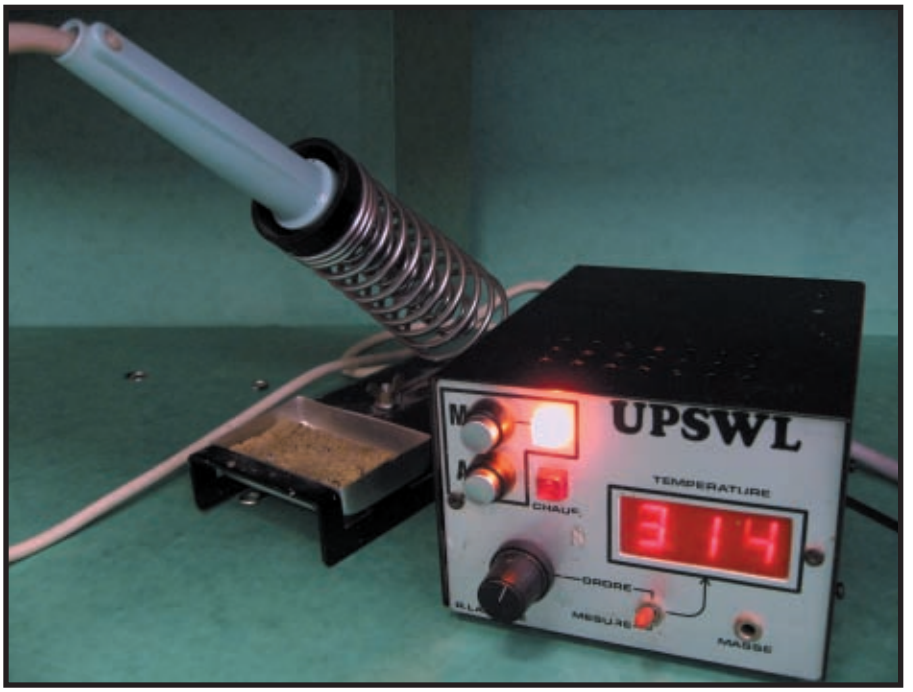



**Photo 1—**This is the power supply and controller I built in 1987 and its Weller VR20 soldering iron. Its first basic "threshold comparator" design was working, but it had a ±3°C oscillation around the preset temperature. A simple control improvement reduced the oscillation to ±1°C.

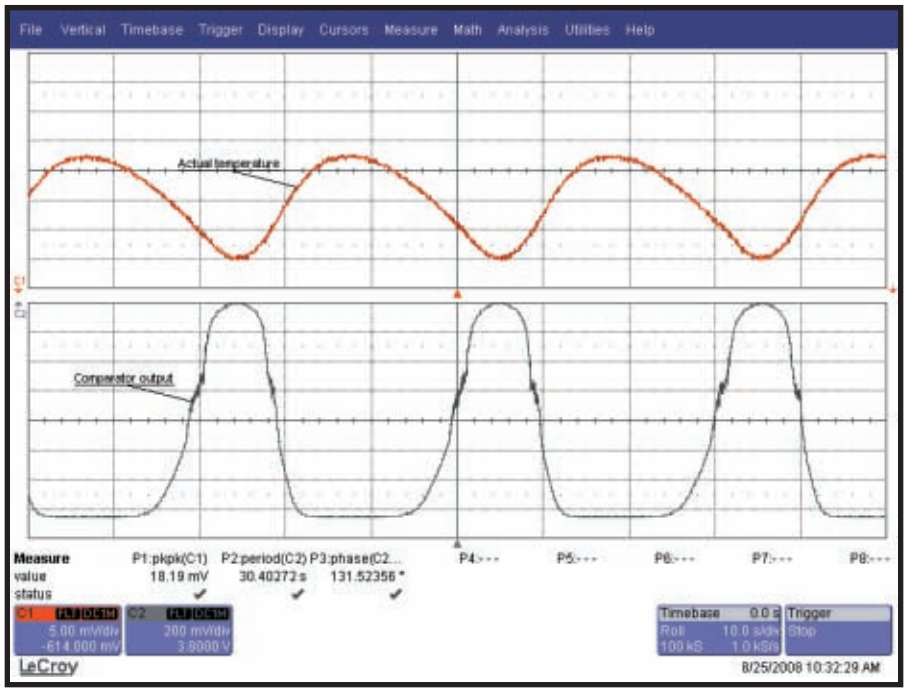



**Photo 2—**This plot was taken on my new LeCroy WaveRunner 6050A oscilloscope connected to the iron controller in its initial version. The top channel is the sensor measurement. The bottom is the command sent to the output TRIAC. In this first design, I used a simple comparator. Both curves are exactly in phase; but as a consequence, the oscillation is high due to thermal inertia: 19 $mV_{PP}$ corresponding to 6°C. Heating cycles are roughly 30 s each.

tuned by only three or four parameters. As you will see, the parameters can be determined by empirical methods, without even knowing the exact behavior of the controlled process. Don't get me wrong. I am not saying that a PID will solve any control problem, especially if the system has multiple inputs and outputs. But you may want to give it a try before digging into more complex solutions. In any case, it's a must-have for every engineer. So, let's go with PID!

## A BASIC CASE

The most basic example of a feedback loop is a temperature controller used to drive a heater or cooler to get a precise temperature (or temperature profile) on a temperature sensor on a device. Regular readers may remember my article about a home-made reflow oven controller based on a firmware PID loop ("Easy Reflow: Build an SMT Reflow Oven Controller," *Circuit Cellar* 168, 2004). This month, I will take an even simpler example. Twenty years ago, I bought a Weller VR20 soldering iron, which had a built-in resistive temperature sensor. At that time, I couldn't afford the corresponding power supply, so of course I built my own (see Photo 1). It was easy. I used a 24-V transformer, a potentiometer to tune the temperature, a digital voltmeter

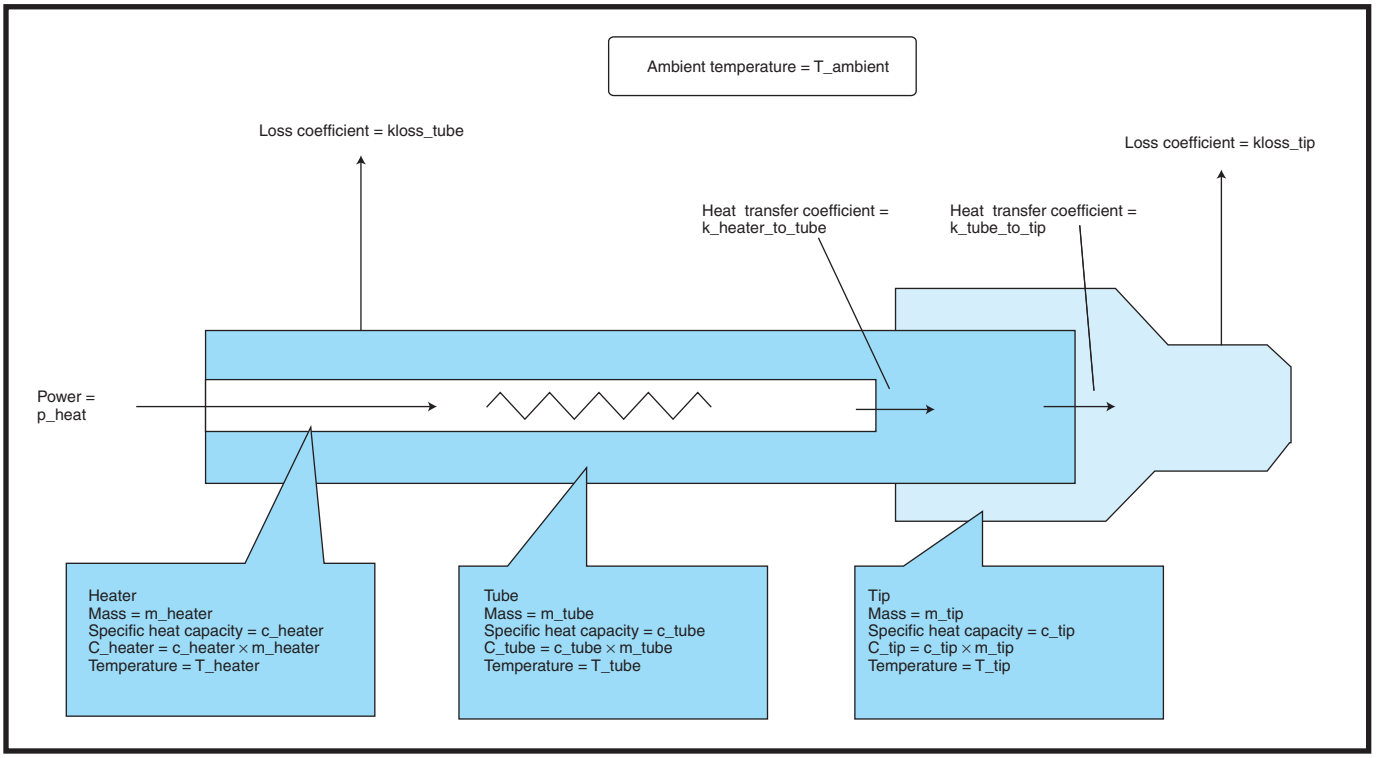



**Figure 1—**For this article, I developed Scilab code based on this thermal latency model. The heater is heating a tube, which heats the iron tip. I assumed the respective temperatures of the heater, tube, and tip to be homogeneous. The heat power transferred from heater to tube is supposed to be a constant factor multiplied by the temperature difference between both elements. The same goes for transfer from tube to tip or to ambient air. This simplistic model enabled me to get reasonably realistic simulation results.

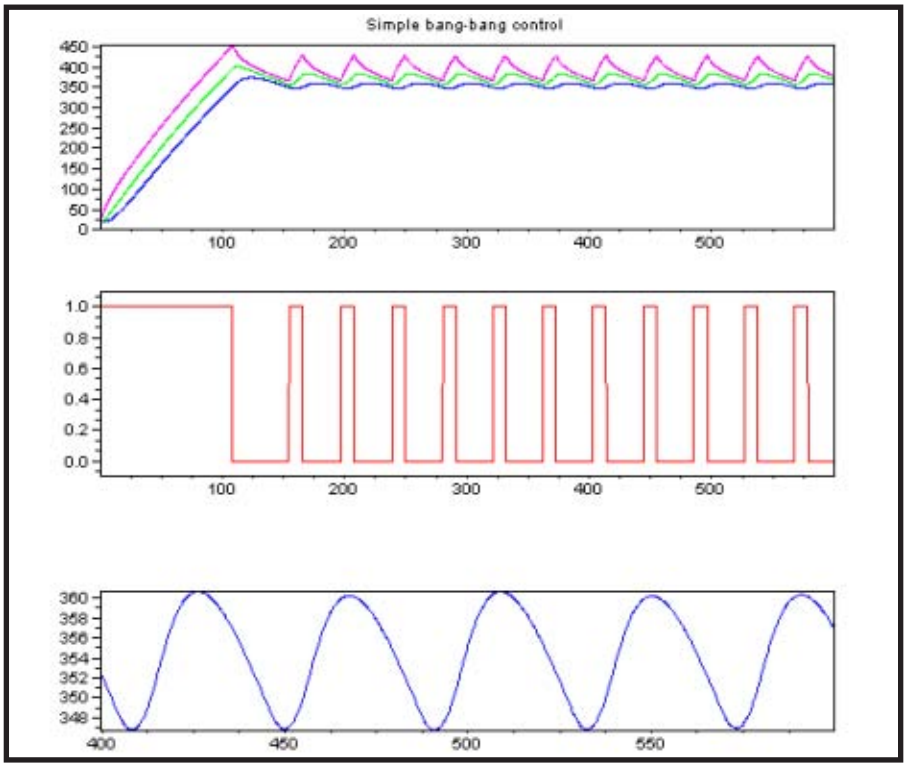



**Figure 2—**This is the simulation result for a simple "bang bang comparator" system like the one used in the initial version of my iron controller. The top curves are respective heater, tube, and iron temperatures. The middle plot is the power applied on the heater. The bottom curve is a zoom on the tip temperature, showing a 12°C swing around the 350°C target with the parameters I have used.

(Remember the old CA3161/CA3162 three-digit voltmeter chipset?), and a crude "control system." I used an LM311 comparator to drive an output TRIAC on and off, whether the measured temperature was above or below the threshold. For control system specialists, this is the most basic form of a "bang-bang control" algorithm—and it is so for obvious reasons. (Just imagine yourself driving your car with only two settings: throttle fully open or brakes fully engaged.) It was working, but the temperature regulation was oscillating around 3°C above and below the preset value. That corresponded to an 18-mV oscillation on the sensor measurement (see Photo 2).

Why? Just because the heater is not in direct contact with the sensor and because the assembly does not weigh 0 g. The heat takes some time to go from the heater to the tip, so if you wait for the sensor to reach the target temperature before switching the power off, it is already too late. The heat will continue to flow from the heater to the tip and you will get a significant overshoot, which ultimately gives oscillation around the threshold. It is easy to model this behavior (see Figure 1). I have even coded a Scilab simulation of this system (see Figure 2). Close to the behavior of my old iron controller, isn't it?

Just as a reminder: Scilab is an open-source Matlab-like tool with great simulation toolboxes. The Scilab simulation sources I coded for this article are posted on the *Circuit Cellar* FTP site. Don't hesitate to read them because I've commented them heavily.

As I will show you at the end of this article, I improved this iron controller a couple of years after its assembly and got a drastically improved regulation. On the hardware side, I do not recommend that you duplicate this design because it is based on obsolete technology. But it is a perfect example to introduce PID controls.

## PROPORTIONAL?

How do you improve the thermal regulation? Using a full on/off drive on the heater is simple, but it is not the best solution. Why should you use 80 W of heating power if you are close to the target? It should be reduced more and more as you approach. The simplest

**a)** P control :

```
Error = actual - target
Command = Kp . Error
Command = Limit(Command,Commandmin,Commandmax)
Pheater = Pmax . Command
```

**b)** PD control (theoretical) :

```
Error = actual - target
Command = Kp . Error + Kd . d(Error)/dt
```

**c)** PD control (implementation) :

```
Previous error = Error
Error = actual - target
Command = Kp . (actual - target) + Kd/timestep . (Error - Previous error)
```

**d)** PID control (theoretical) :

```
Error = actual - target
Command = Kp . Error + Kd . d(Error)/dt + Ki . Integral (Error.dt)
```

**e)** PID control (implementation) :

```
Previous error = Error
Error = actual - target
Integral = Integral + error
Command = Kp . (actual - target) + Kd/timestep . (Error - Previous error)
          + Ki . Timestep . Integral
```

**Figure 3—**A P control simply calculates the error between the target and measured values, and uses it to drive the output. This is better than a simple threshold comparison but may induce overshoots. A PD control damps the system with the addition of a derivative term, which is usually calculated as the difference between successive errors. Lastly, the full PID control also adds an integral term, which is the sum of the error over time, in order to avoid any systematic error.

ICbank
Gateway to Electronics

# MP3P DIY KIT, Do it yourself

## (Include Firmware Full source Code, Schematic)

## · myPIC




Only
**$160** qty 100 **$220** qty 1

### Powerful feature

- MP3 Encoding, Real time decoding (320Kbps)
- Free charge MPLAB C-Compiler student-edition apply
- Spectrum Analyzer
- Application: Focusing for evaluation based on PIC
- Offer full source code, schematic

| Specification |
|---|
| Microchip dsPIC33FJ256GP710 / 16-bit, 40MIPs DSC |
| VLSI Solution VS1033 MP3 CODEC |
| NXP UDA1330 Stereo Audio DAC |
| Texas Instrument TPA6110A2 Headphone Amp(150mW) |
| 320x240 TFT LCD |
| Touch screen |
| SD/SDHC/MMC Card |
| External extension port (UART, SPI, I2C, I2S) |

## · myWave (MP3 DIY KIT SD card Interface)



Only
**$150** qty 100
**$200** qty 1

### Powerful feature

- Play, MP3 Information, Reward, forward, Vol+/-
- Focusing for MP3 Player
- SD Card interface
- Power: battery
- offer full source code, schematic

| Item | Specification |
|---|---|
| MCU | Atmel ATmega128L |
| MP3 Decoder | VS1002 / VS1003(WMA) |
| IDE Interface | Standard IDE type HDD(2.5", 3.5") |
| Power | 12V, 1.5A |
| LCD | 128 x 64 Graphic LCD |
| Etc | Firmware download/update with AVR ISP connector |

## · myAudio (MP3 DIY KIT IDE)

Only
**$180** qty 100
**$220** qty 1



### Powerful feature

- Play, MP3 Information, Reward, forward, Vol+/-
- Focusing for full MP3 Player (Without case)
- IDE Interface
- Power: Adapter
- Offer full source code, schematic

ICbank
Gateway to Electronics

**Youngsan Cho / Senior Account Manager / Sales & Marketing ICbank Inc.**

Tel : 82-707-019-3924, Fax : 82-2-6340-1888, Cell : 82-17-268-2078, Email : youngsan@icbank.com
18F Ace High -End 2, 222-14 Guro Digital Valley, Guro-dong, Seoul, 152-050, Korea Website : https://www.icbank.com/

way to do so would be to calculate the heating power as proportional to the distance to the target. This is where the "P" of PID comes from. Rather than just comparing the measured value to a threshold and switching the output on and off, a proportional controller manages it more smoothly. The algorithm calculates the error, which is the target value minus the measurement, multiplies it by a given gain (usually denoted Kp), and uses the result of this calculation to drive the output after limiting it to reasonable values. Refer to the pseudocode algorithm in Figure 3a.

You may criticize this approach as a linear power supply stage would be needed to implement such a proportional control, with its power inefficiency and added complexity, and you would be right. But for a thermal controller, nothing forbids you from using the "Pheater" value to directly drive a high-speed PWM output rather than a DC power generator. The heating power will be proportional to the mean PWM value, which is the desired command. What improvement could you get with such a proportional control system? I did the simulation in Figure 2. The results show that the temperature oscillation is reduced from 12° down to 7°C, at least under the hypothesis of the simulation (see Figure 4).

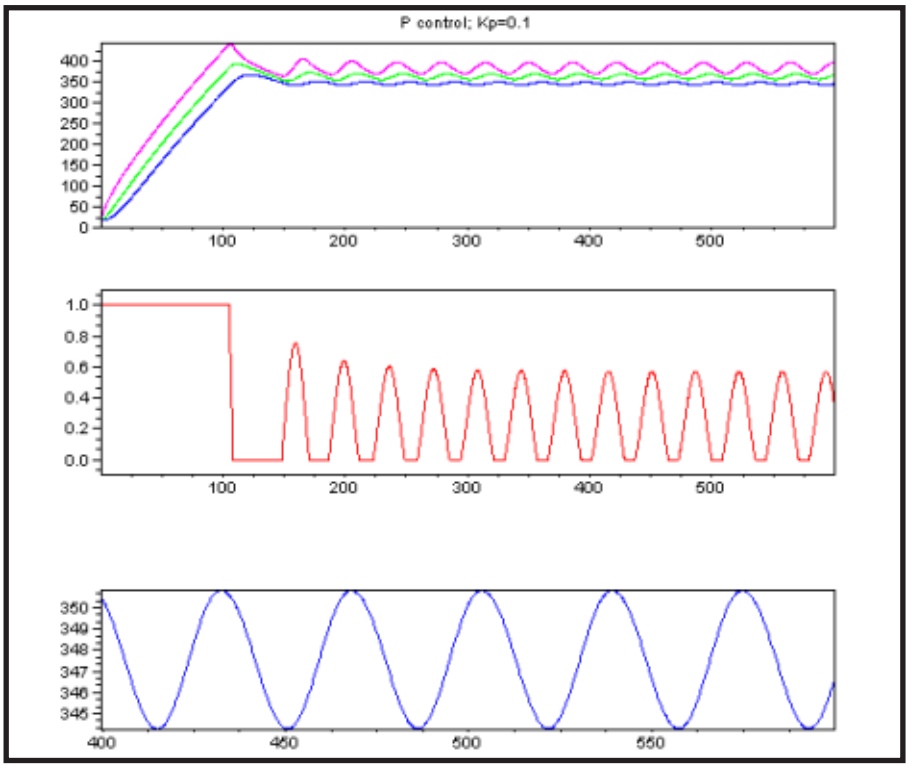



**Figure 4—**A proportional control enables you to generate an analog output value. It's low if the current measurement is close to the preset target but higher if the target is far (medium curve). The overall temperature oscillation is reduced from 12° down to 7°C, as compared to the "bang-bang" control in Figure 2.

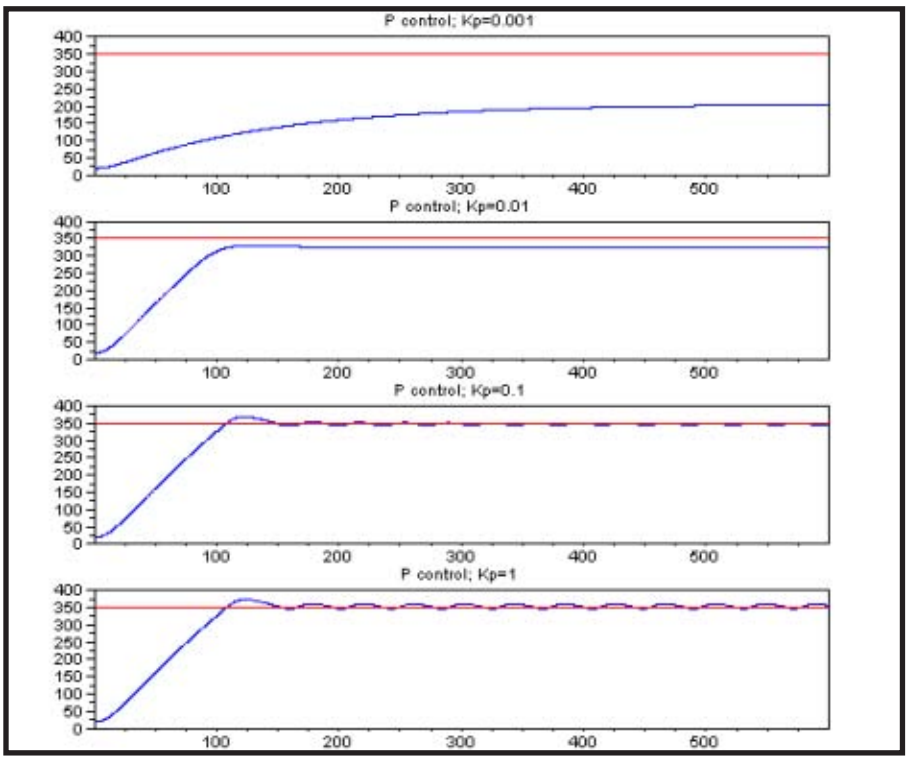



**Figure 5—**This simulation shows you the behavior of the simulated system with a proportional control and different values of the Kp gain. With small Kp values (top plot), the actual temperature takes a long time to reach the target, and may not reach it. The loop is "soft." If you increase the Kp gain, the regulation becomes quicker and quicker up to a point where overshoot starts to appear. The optimum value is often just after the appearance of oscillations (Kp = 0.1 in this instance).

The optimal value of Kp must be determined for each application because it is dependent on the system parameters and your preferences. Figure 5 illustrates the system's behavior with different Kp values. My experience tells me that it is to start with low Kp values and increase it up to a point where oscillations and ringing starts to be a little too high.

## DERIVATIVE HELPS!

A proportional controller uses only the current measurement to determine the output value. It doesn't have memory or forecasting to improve the regulation. When you press the brake as you park your car in your garage, you don't apply only a pressure proportional to the distance between your car and the back wall. You also use other information, such as the speed at which your car is approaching the wall, and this makes sense. As you may remember from your youth, speed is the derivative of

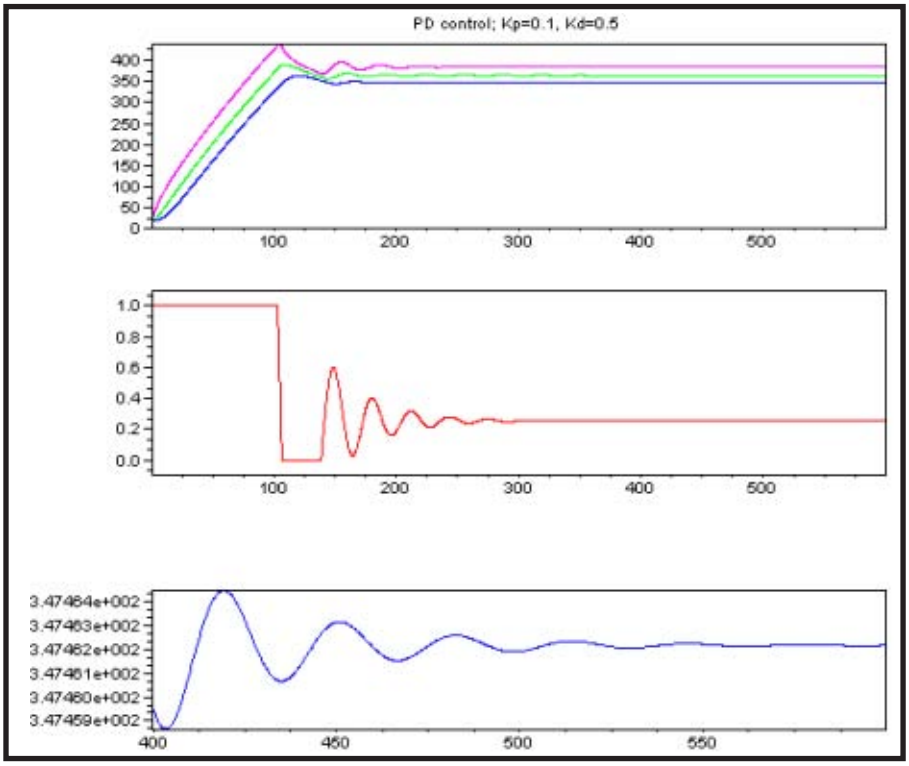



**Figure 6—**The addition of a derivative parameter gives a PD control loop. Thanks to the derivative, the oscillations are damped out and the system reaches a steady state, at least in this simulation. However, note that the stabilized value is not exactly the preset 350°C. It's a little lower (347.4°C) due to heat dissipation in the ambient air.

the distance over time. A proportional-derivative control (PD) adds the derivative of the error to the calculation with another gain noted as Kd (see Figure 3b).

For discrete time systems (e.g., microcontroller-based), the derivative of the error can be approximated as the current error minus the previous one, divided by the duration of the calculation time step. The result is the algorithm in Figure 3c.

How does it work? If the error is increasing quickly, then the d(Error)/dt term is high. So, for the same absolute error, the command applied on the output is higher. This enables you to return to the target quicker. On the contrary, if the error is quickly reducing, then the d(Error)/dt term is negative. This reduces the power applied on the output, which reduces risks of overshoots. For simple systems, like the one I simulated, a PD algorithm provides an impressive improvement, even if the situation is not so easy in real life (see Figure 6).

How do you tune the Kd coefficient? As illustrated in Figure 7, the system is more and more damped when Kd is higher. If Kd is null, then we are back to a simple proportional control. An increased Kd reduces oscillations and overshoots but also increases the time needed to reach a new setpoint. If you increase Kd even more, the system starts to be soft and often too soft. This is why I previously said that the Kp proportional parameter should usually be set a little higher than the value which gave the first oscillations because Kd will help reduce them. So, in a nutshell, with a PD regulation, you can first set Kd and Kp to 0, increase Kp until small oscillations appear on the output, and then increase Kd until there is enough damping but not more. The resulting parameter set is often close to optimal for common systems.

## INTEGRAL, FOR FULL PID

Take another look at the PD control simulation in Figure 6 and you will see that the system reaches a steady state, which is not exactly the target value. On this simulation, the long-term sensor value is 347°C, 3° lower than the preset 350°C target. This is because the system is dissipative, meaning that some energy flows

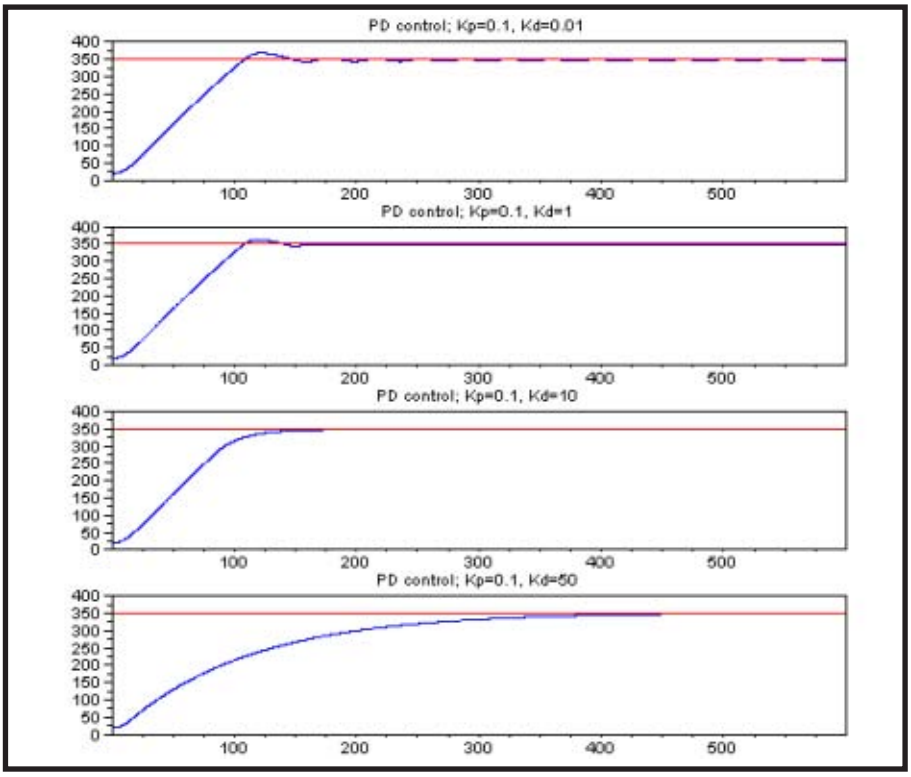



**Figure 7—**This is what usually happens when the derivative gain Kd is increased. The oscillations are damped more and more. At a given point, the system is no longer oscillating but starts to be long to reach its target. The optimal setting depends on the system constraints.

from the iron to the ambient air. At equilibrium, the heat loss to the ambient air is exactly equal to the 3°C error multiplied by Kd. The system is stable but will never reach its target setpoint: the error stays constant so its derivative is null and Kd is useless.

"PID" appears in this article's title. I already covered "PD" feedback loops, so we need to add the "I" in order to avoid such long-term errors. Not only do you need to take into account the error and its derivative over time, but also its integral over time. If there is a constant error, this integral will be higher and higher over time. If you add it to the command through another Ki gain, then the equilibrium state will be forced to be exactly at the setpoint value (see Figure 3d). The result, for time-sampled systems, is the algorithm in Figure 3e.

This works and it is the final form of the PID control algorithm (see Figure 8). However, please take care. The integral term must be manipulated with caution. Contrary to the Kd and Kp gains (at least with reasonable values), an improper Ki gain can make the system unstable. Moreover, the effect of Ki is usually opposed to Kd: a higher Ki gives higher oscillations and longer stabilization time. Practically speaking, it is best to always start

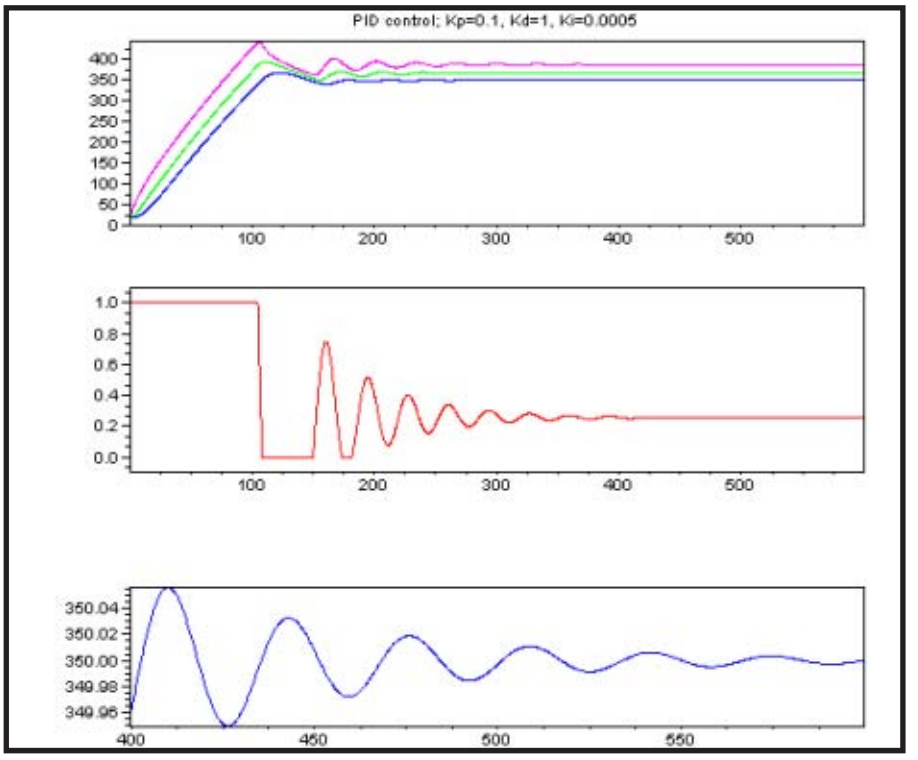



**Figure 8—**The addition of an integral term enables you to ensure that the steady point is equal to the preset 350°C target. However, note that the oscillations and stabilization time are a little higher than with a simpler PD control loop as on Figure 6.

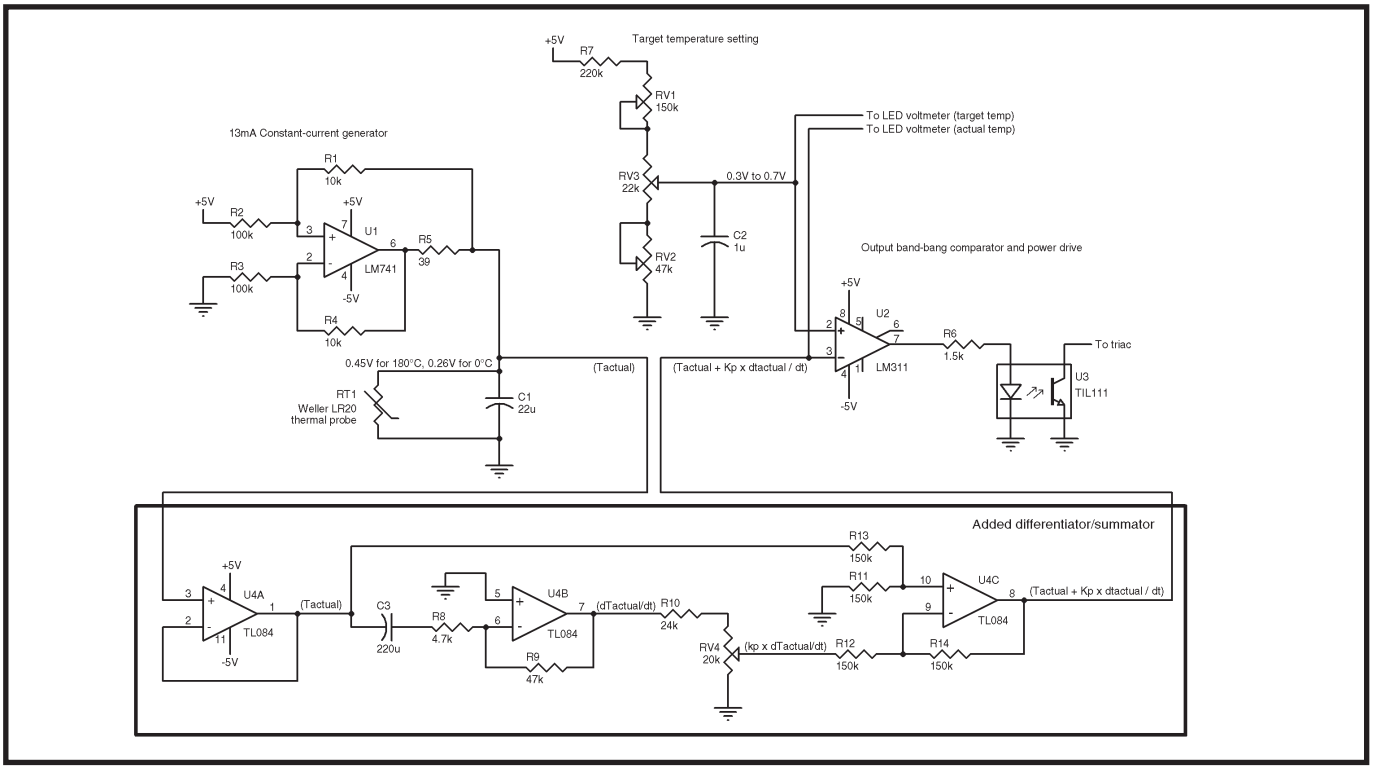



**Figure 9—**At the top is the original schematic of the temperature control section of my iron controller. A 13-mA constant current generator was driving the resistor temperature sensor, providing a voltage roughly proportional to the temperature. The voltage was simply compared to a preset threshold through an LM311 comparator, then driving a TRIAC through an optoisolator. The modification involved the addition of a quad TL084 op-amp to buffer (U4.A), derive over time (U4B), and sum (U4.C) the signal with its derivative. The potentiometer RV4 sets the Kd gain.

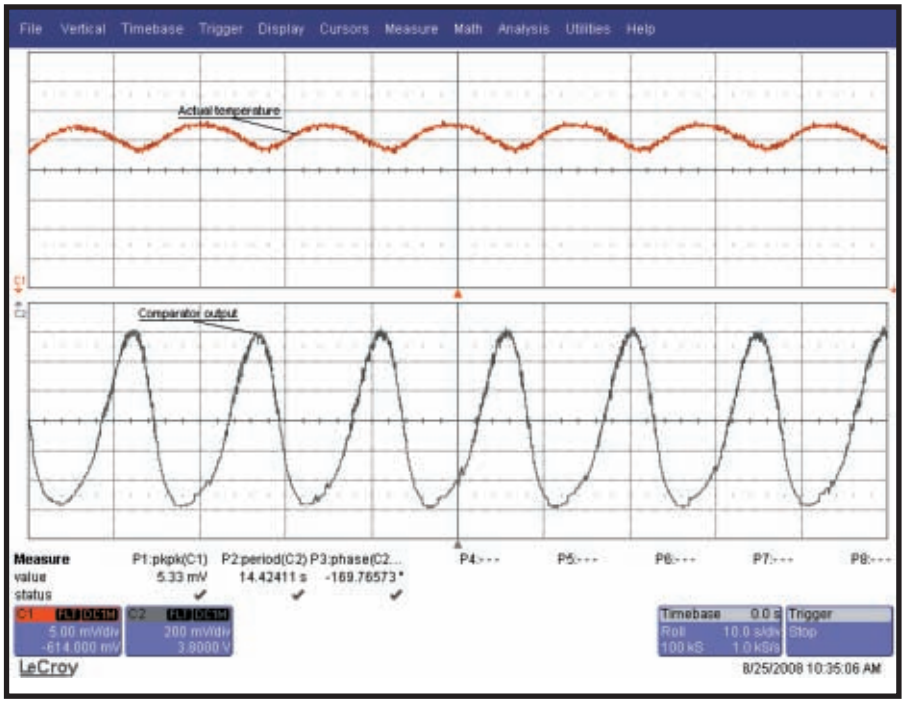



**Photo 3—**This is how the actual iron temperature regulation behaves with the addition of the derivative term. The oscillation of the sensor output is reduced to a little more than 5 $mV_{PP}$, four times less than with the simple comparator, providing a stability of 1°C.

with Ki = 0. If it is mandatory, increase Ki a little after determining the optimal Kp and Kd parameters, just to a value providing a good, long-term convergence. Then, you will probably need to retune Kd and Kp to readapt the short-term behavior, and then Ki again, and more. Another good way to reduce the risk of instability is to limit the maximum and minimum value of the integral term to a given range with a new parameter MaxIntegral. This is another value to determine by experimentation; but globally, you end up with only four numbers to optimize, which is far easier than going through the full control system theory.

## HARDWARE IMPROVEMENTS

It is time to go back to my dear 1987 soldering iron regulator. What did I do in 1989 to reduce the temperature oscillations? I simply added a differential term. I didn't even change the output stage, which is still a TRIAC driven by a comparator in full on/off mode, but I no longer compared the measured value to the preset value. I compared the measured value plus Kd times d(measured value)/dt to the preset value. Think about it twice. This is exactly the same as the PD algorithm as long as the preset value is constant. A PD control loop can be 100% analog (see Figure 9). What were the actual improvements? Compare the oscillogram in Photo 3 with the initial one (see Photo 2). The addition of a Kd parameter reduced the temperature oscillation from 6° down to 1.3°C. That's not bad with just a couple of op-amps more. Photo 3 shows the derivative term in action. The output is no longer fully in phase with the sensor. It starts to increase as soon as the sensor temperature starts to reach its maximum, even if the actual temperature is still above target. This is anticipation.

## WRAPPING UP

To be honest, I no longer use the iron controller on a daily basis because I have a newer one. However, I still use it from time to time, even if it is not lead-free compatible. Anyway, I hope I have demonstrated that PID regulations are simple to code, or even to wire with a couple of op-amps. Moreover, they are easy to tune at least for simple systems. Using only the proportional term may already give good results. The derivative term could be added to provide damping (or shaping) of the response. The addition of an integral term ensures that there will be no systematic errors. If the average error is not zero, the integral term will increase over time, reducing the error. However, the integral term is a little more difficult to manage than the Kp and Kd terms because it can make the system unstable. Handle it with care, or add limits on the integral term allowed values. Don't forget that these coefficients can be negative too.

Lastly, note that playing with the Kp, Ki, and Kd coefficients is easy, and looking at their effects on the real-life controlled system is fun. Consider experimenting with a small controlled system rather than exercise yourself directly on your nearby nuclear plant. Anyway, I hope that PID is no longer on the darker side for you!

*Robert Lacoste lives near Paris, France. He has 18 years of experience working on embedded systems, analog designs, and wireless telecommunications. He has won prizes in more than 15 international design contests. In 2003, Robert started a consulting company, ALCIOM, to share his passion for innovative mixed-signal designs. You can reach him at rlacoste@alciom.com. Don't forget to write "Darker Side" in the subject line to bypass his spam filters.*

## PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/221.

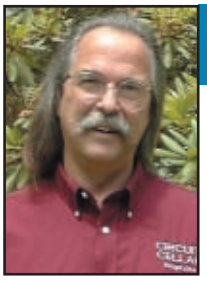
## RESOURCES

T. Wescott, *Applied Control Theory for Embedded Systems*, Newnes, Burlington, MA, 2006.

Wikipedia, "Control Theory," http://en.wikipedia.org/wiki/Control_theory.

## SOURCE

**WaveRunner 6050A Oscilloscope**
LeCroy Corp.
www.lecroy.com





by Jeff Bachiochi

# Component Control

## A Tool for Finding and Handling Small Parts

It's time to take control of your components. You can identify tiny surface-mount parts with a handheld LCR meter. The lightweight tool automatically determines the type of component, inductor, capacitor, or resistor (LCR), and selects the proper range and signal frequency for the most accurate measurement. The component type, measurement results, and test conditions used to determine the results are displayed on a dot-matrix display.

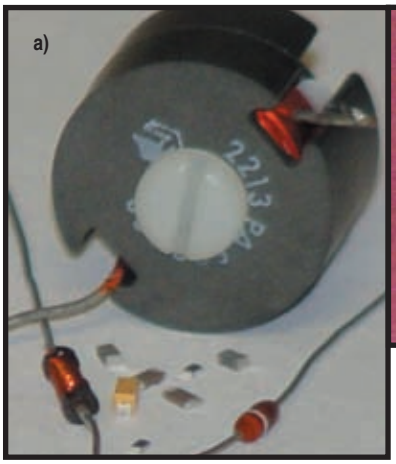
If your work area is like mine, you probably have a small box that collects all the odd components that end up on your bench after a project (see Photo 1a). I am a little more careful when it comes to surface-mount parts. They are usually purchased as taped strips (reels) of parts and are better protected from getting jumbled together with other parts (if left taped). However, one of the big problems with SMT stuff is that it is usually too small to be labeled. Thus, you can easily lose track of parts and their values. I can't tell you how many times I've picked up an SMT part only to have it shoot out from the tweezers never to be seen again!

Frankly, I'm surprised I haven't seen a good tool like the Advance Devices Smart Tweezers handheld LCR meter sooner (see Photo 1b). The small, lightweight format fits comfortably, enabling operation with just one hand. Although it is designed to easily grab SMT devices with its tweezer-type probes, it can perform its magic on through-hole parts as well.

You can use the Smart Tweezers (ST) to identify and evaluate loose components and those already soldered on a PCB (unpowered). Of course, you may need to have some knowledge of the circuitry to determine what other devices might be in parallel with and affecting the device under test (DUT).

### WHAT DOES IT DO?

The Smart Tweezers meter automatically determines the type of component, resistor, capacitor, or inductor (RCL), and selects the proper range and signal frequency for the highest accuracy measurement. Using a small 32 × 96 dot graphics display, the Smart Tweezers meter clearly displays the component type, measurement results, and test conditions used to determine the results. The display is divided into three areas. The central



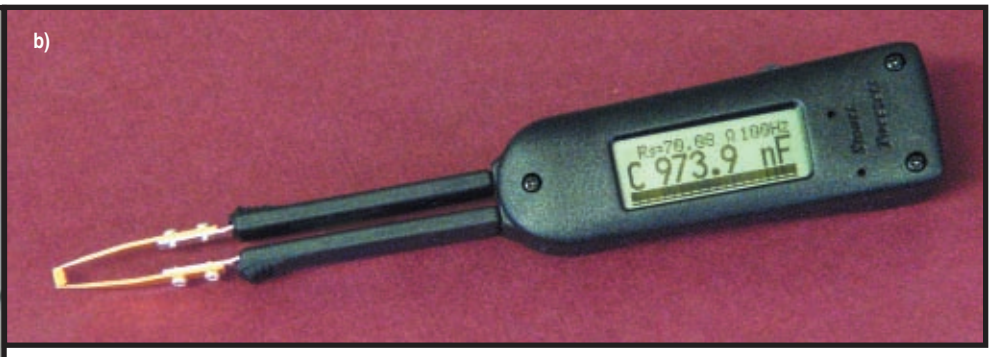




**Photo 1a—**Here's a few of the hundreds of unmarked (or unreadable) parts in my "junk" box. Every one of these parts is identifiable using the Advance Devices Smart Tweezers LCR handheld meter. **b—**The Smart Tweezers meter can identify a component type and value by picking it up. The auto function will identify capacitors, resistors, and inductors.

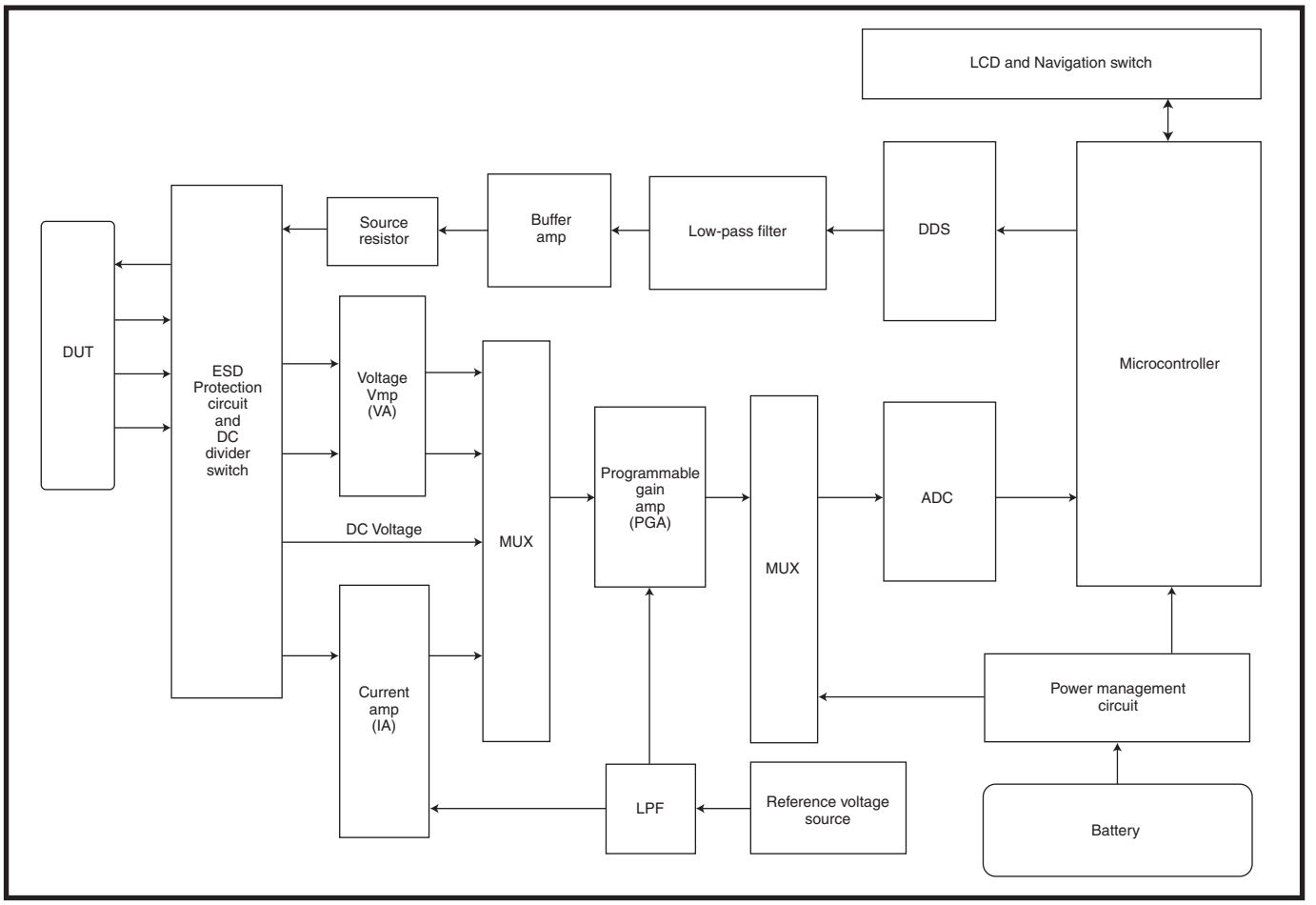



**Figure 1—**A Texas Instruments MSP430F135 microcontroller orchestrates the user I/O, D/A, A/D, and power management circuitry to identify common devices, including resistors, capacitors, inductors, and diodes.

(and largest) area contains the component type and determined value. The top area shows additional parameters and testing conditions. The bottom area displays a horizontal bar graph indicating the value's percentage of the test range.

If you've had the pleasure of assembling any circuit using surface mount technology (SMT) components, you probably have at least one pair of tweezers for picking up and placing surface-mount devices (SMDs). The Smart Tweezers meter has conductive pincer-like contacts that can measure and identify an SMD if picked up the correct way (see Photo 1b). The small and lightweight enclosure fits well in your hand, kind of like writing with a pen or pencil. At first, I thought the small LCD might be custom glass with specific glyphs. But after playing with it a while, I realized it was a simple dot-matrix display. The design can be used with either of your hands. When configured for lefties, all of the LCD text and graphics are rotated 180°. I'm guessing this is done rather simply by manipulating the LCD data and where the data is stored (mirroring each most significant for least significant bit/register).

Automatic mode determines the device type (LCR) and measures, calculates, and displays the appropriate value. Manual modes are available to force the tests to use particular parameters. All modes are accessed via a jog dial—actually three switches (push left, push right, and push down).

## WHAT IS GOING ON?

Figure 1 shows the system's major building blocks. The microcontroller of choice is a 16-bit Texas Instruments

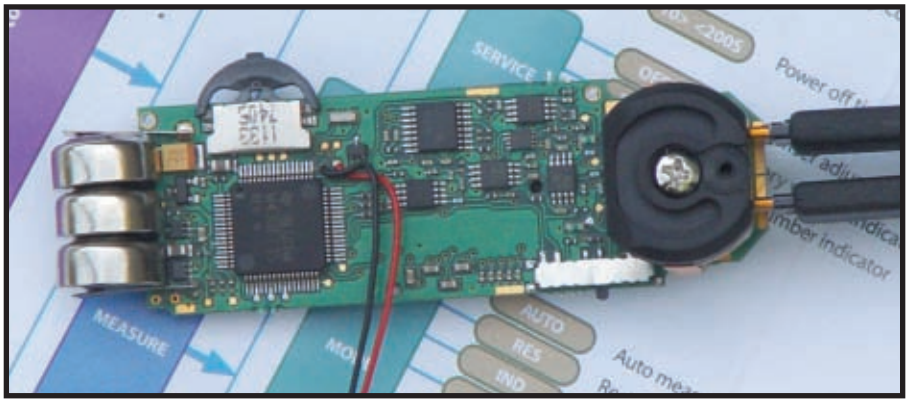



**Photo 2—**The internal PCB has the MSP430F135 microcontroller and all the other SMT parts (except the LCD) mounted on one side. The three button cell batteries can provide over 80 h of use.

MSP430F135 (see Photo 2). Besides handling the user input/output through a jog-dial/LCD, the microcontroller calculates the impedance of a DUT by measuring the voltage across it and the current through it, as shown in the front-end circuitry (see Figure 2). The MSP430F135 controls both the amplitude and frequency of the voltage source ($V_S$) and is typically approximately 1 V (sine wave). $V_S$ is applied to the DUT through the source resistance $R_S$. $DUT_V$ is the voltage measured by the first amplifier (AU) directly across the DUT and is measured by one A/D channel. Current through the DUT ($DUT_I$) is converted to a proportional voltage by a second amplifier (AI) and the feedback resistor $R_I$ ($V = DUT_I \times R_I$) so it can be read by a second A/D channel.

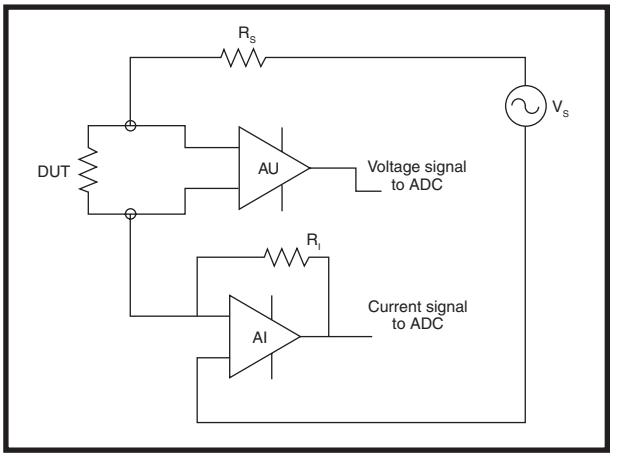



**Figure 2—**The analog front end uses a DAC to provide AC excitation to the DUT. The resultant voltage across and current through the device is measured by A/D circuitry to provide data back to the microcontroller.

The DUT's impedance is a combination of resistance and capacitive or inductive reactance. The proportions of each of these are based on the relationships between applied voltage and the measured voltage and current. With an ideal resistive DUT, the impedance, and thus resistance, is the same because the $DUT_I$ and $DUT_V$ are in phase. With an ideal capacitor DUT, the impedance is the capacitive reactance because the $DUT_I$ will lead the $DUT_V$ by 90°. The capacitance must be calculated based on:

$$C = \frac{-1}{2 \pi f X_C} \quad [1]$$

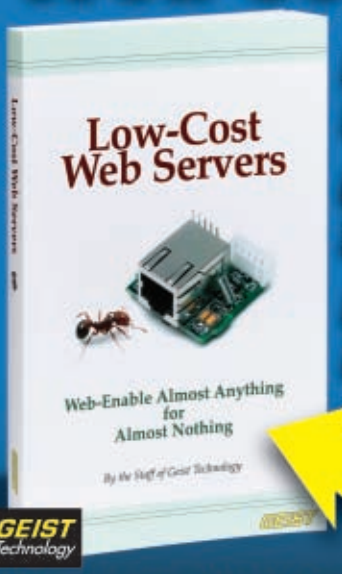
With an ideal inductor DUT, the impedance is the inductive reactance because the $DUT_I$ will lag the $DUT_V$ by 90°. The inductance must be calculated based on:

$$L = \frac{-1}{2 \pi f X_L} \quad [2]$$

Manufacturing and materials will play a role in a device's quality. Capacitors, inductors, and even resistors may be less than ideal components. We all consider a resistor to be pretty much an ideal device. However, many resistors are made by depositing a material on a form and etching or cutting a spiral pattern in the coating, creating a lengthy strip to achieve the required value, but also looking a bit like an inductor. This is perhaps more clearly seen in wire-wound resistors. While this is usually


Cellular and GPS capable
Data Mover


• Flash file System
• 4 Chan 12-Bit A/D
• 1MB SRAM
• 512KB FLASH
• 4 Isolated Inputs
• 4 Hi Current Outputs
• 2 External RS-232
• 2 Internal RS-232
• Bat Backed Clk/Cal
• Cell Modem Option
• Internal GPS Option
• Metal Case Option
• 1ma Standby Option
It's easy and cost-effective to do mobile or solar-powered data collection and asset monitoring with the JK*micro's* **Data Mover**.
With the ability to integrate a Cellular Modem, GPS and DOS-based embedded controller in a single rugged enclosure, you can capture and transmit your data quickly, easily and at low cost. Inexpensive development kits including Borland C/C++ and PowerBasic are available now. Call or email us for more details.
Call **530-297-6073** Email **sales@jkmicro.com**
On the web at **www.jkmicro.com**
JK microsystems

We Build Tiny Web Servers


Lowest-Cost Designs
Complete Software
Fast Delivery
FREE BOOK!
Request your free book at (512) 331-8676
or FreeBook@GeistTek.com

minimal, it must be considered in circuits where the frequencies used may cause this inductance to affect circuit operation.

All current-carrying conductors are inductors to some extent. That is, the magnetic field created by a change in current in them is not instantaneous and therefore opposes the change. This opposition can be enhanced by providing a medium (core) that collects and hangs onto the magnetic energy, as well as concentrating (wrapping) the conductors around the core. As shown in the previous equations, there is a direct relationship between the inductance (opposition to change) and frequency (rate of change). Therefore, testing a component at a number of frequencies is required to obtain measurement results that are within analytical limits. It's likely an inductor will be somewhat less than ideal. Series resistance will be part of the DUT's impedance. This can be determined by the phase relationship between the $DUT_V$ and the $DUT_I$. With the ideal resistor (phase shift = 0°) and the ideal inductor (phase shift = 90°), any inductor that is not ideal will have a resistive/inductive reactance relationship of phase difference equals arctan ($X_L/R$). The relationship ($X_L/R$) is also known as the quality factor (Q). The closer the inductor is to ideal, the higher the Q.

A capacitor, on the other hand, opposes a change in voltage due to its physical makeup. The leads of a capacitor are connected to large plates held just a fraction of an inch apart by thin insulating material. Ideally, there is infinite resistance between the plates. When a voltage is applied across the plates, maximum current flows because the potential difference between the source and the plates is greatest. Electrons are in fact not flowing through the device but being stored on (or released from) the plates. The electrons create an electric charge (voltage) across the plates. As the potential between the source voltage and the plate voltage decreases, so does the current flow. Capacitance goes up as the plate size increases or the spacing decreases. Although the dielectric insulating material also affects capacitance, it has a greater effect on losses that are affected by things like temperature and frequency. As shown in the previous equations, there is a direct relationship between the capacitance (opposition to change) and frequency (rate of change). Therefore, testing a component at a number of frequencies is required to obtain measurement results that are within analytical limits. It's likely a capacitor will be somewhat less than ideal. There will be some parallel resistance that will be part of the DUT's impedance. This can be determined by the phase relationship between the $DUT_V$ and the $DUT_I$. With the ideal resistor (phase shift = 0°) and the ideal capacitor (phase shift = 90°), any nonideal capacitor will have a resistive/capacitive reactance relationship of phase difference equals arctan ($X_C/R$). The relationship ($R/X_C$) is also known as the dissipation factor (D). The closer the



capacitor is to ideal, the lower D will be.

## FINGERTIP CONTROL

The jog dial provides up, down, and select functions. An initial press (in any direction) initializes the Smart Tweezers and configures it to the last selected functions before it turns itself off. You can immediately begin taking measurements of components. A second push brings up a menu of functions you may choose from. The menu road map is shown in Figure 3. Choosing AUTOSET resets all parameters to the default setting and returns you to Operational mode. There are two other selections you can choose instead of EXIT, SYSTEM, and MEASURE. If you choose SYSTEM, you can select one of the four system settings: SOUND, DISPLAY, TIMEOUT, and SERVICE. If you choose MEASURE, you can select one of the seven modes of operation: AUTO, RES, IND, CAP, DIODE, VOLTAGE, and TRACE.

The system settings enable you to customize the operation of the Smart Tweezers. The SOUND setting enables/disables a beep whenever a measurement is taken. The DISPLAY setting lets you set the display's contrast and determine whether the display is formatted for right- or left-hand use. Finally, the TIMEOUT setting lets you set the timeout (the delay after any measurement or selection before shutting off) from 10 to 200 s.

While the Smart Tweezers can measure most devices using the AUTO mode (RES/IND/CAP), you can use the measurement settings to change or force a mode of operation. For instance, if you have a wire-wound resistor, you can select inductor (IND) to determine the resistor's inductive reactance due to its physical makeup. If the DUT is a diode, you must force a switch to DIODE mode. A pictograph of a diode will be displayed with its orientation matching the polarity of the connection. The VOLTAGE selection allows for measurements of up to ±8 V. A tiny slide switch on the side of the case must be moved to enable this function. (The LCD prompts you when to slide it.) Note that the auto-off function is disabled while this switch is used. TRACE mode is similar to VOLTAGE. Instead of a text display of the voltage value, a graphic display over time is used, like an oscilloscope. You can dynamically change the time base using the left/right jog dial switches. Because of the delicate and sensitive front end, you should not measure live circuits with more than 1.6 V without selecting VOLTAGE or TRACE modes and moving the slide switch as prompted.

## PRACTICAL LIMITS

There are a few things that might seem obvious but must be taken into consideration for the most accurate measurements. When measuring small resistances, the physical connection resistance between the (gold plated) tips and the DUT is approximately 25 mΩ (this can vary with contact pressure). You must subtract this from the measured value, if appropriate, when it is a large percentage of the DUT. For the most accurate measurements of small capacitors (pF), you should force the test frequency into its highest range (10 kHz).







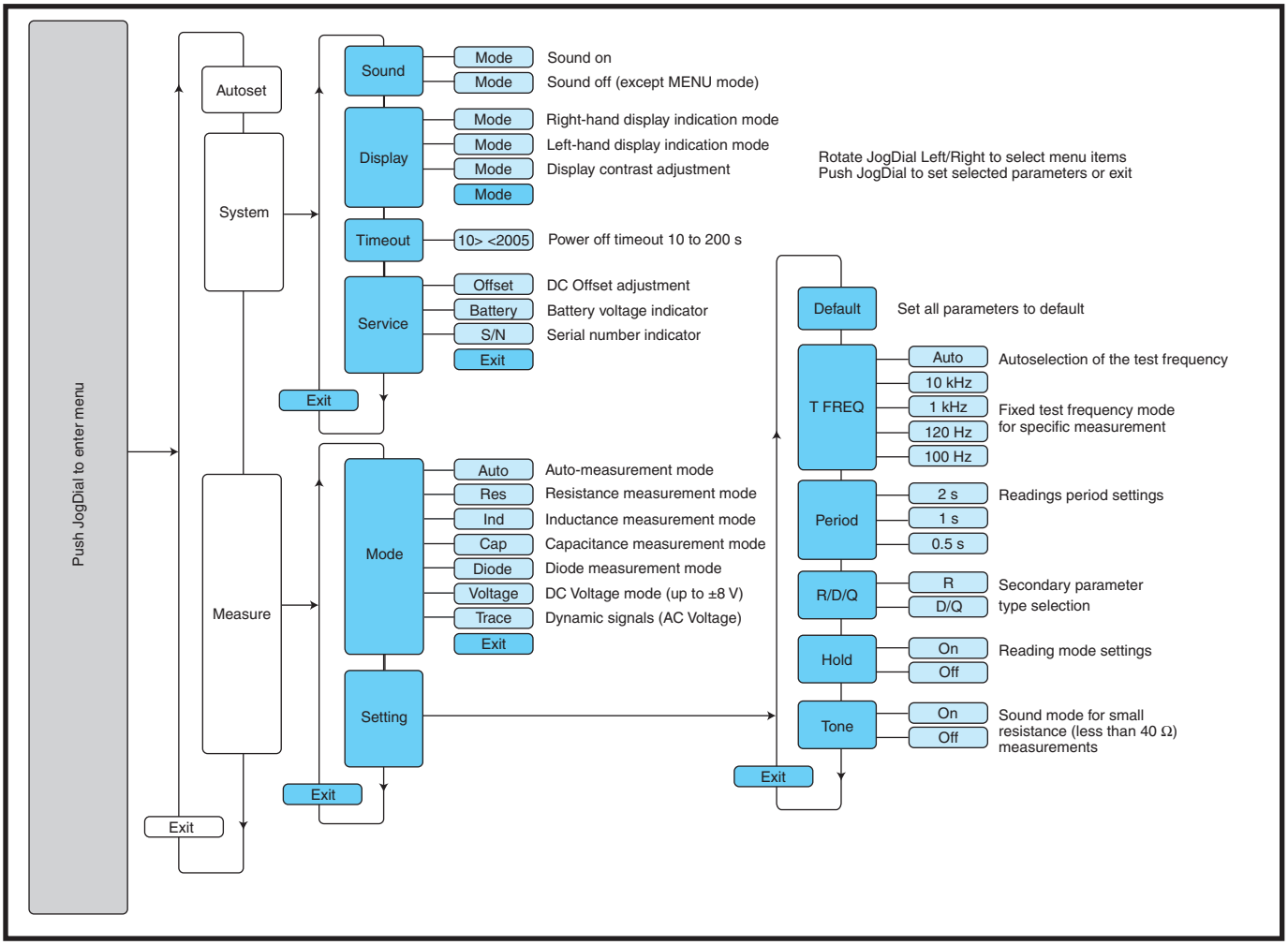



**Figure 3—**The dot-matrix LCD provides resultant data as well as function menus for changing default parameters. You can scroll through the menu selections using the jog dial's left/right switches. By pressing it, you can choose the highlighted selection. Use this chart to follow the menu hierarchy.

Measuring physically small size capacitors will inherently add a small offset of up to a couple of puffs (picofarads) due to the close proximity of the tips. Measuring inductive extremes is most accurate by forcing the test frequency as well (highest frequency for low values and lowest frequency for high values).

## USE IT

After each project, I spend a little (sometimes too little) time cleaning up my workspace. It's not unusual for there to be a bunch of leftover components on my bench top. If I'm bad, I sweep them into a box of miscellaneous parts. If I'm worse, I toss them out! Many parts are unique enough to be easily separated and put back into their proper places. Components like connectors and ICs (generally anything that is large enough) can be handled through divide-and-conquer techniques. When it comes to smaller discrete or SMT parts, I need to get out the hand magnifier to read values. While the colored stripes on 5% resistors are noticeable, I often can't tell which end is which with the resistors that are 5%. Even though I know 0603 SMT resistors have digit values on them, I can't make them out, even with a magnifier. Other small surface-mount capacitors and inductors have room only for a logo. I no longer need to strain my eyes and guess at which parts I'm trying to store away because I can use Smart Tweezers to pick up, identify, and place the part in one easy movement.

I've always used a small pair of tweezers to help position SMT components on PCBs that I hand assemble. I'm pretty good at soldering tiny devices. One of the most frustrating things is to find that I've either used the wrong part in the right place or the right part in the wrong place. Being able to verify a part's type and value prior to hand soldering is helpful in the long run. Likewise, being able to identify components and values that have been placed and soldered can save time. You don't have to remove and replace a part because you don't know if it's the right part. This goes a long way in reducing stress, especially when a circuit is not working correctly. Of course, you need to know how the component is connected in the circuit to determine if its measured value is being affected by the surrounding components. Note that you should always measure components with the circuit in an unpowered state, unless you are specifically using the Smart Tweezers in the voltage mode.

Have you ever had a circuit you were trying to understand? Maybe you

needed to determine the cut-off frequency of an analog filter? The ST is a great tool to help you identify the components and values used in a circuit so you can make the calculation necessary to determine operating parameters. While the Smart Tweezers is made for measuring those parts that fit between its fingertips, you can put clip leads on its tips and measure physically larger parts as well.

Are you a coil winder? Say you need that power inductor for some switching circuit and want to wrap your own coil. After winding what you've calculated to be the proper turn ratio, you will wonder if it will be close enough to do the job. The Smart Tweezers can give you peace of mind by measuring the inductance (and the ESR as well). You can also experiment with how the inductance is affected by the core material you are using.

## PORTABILITY

Thanks to the extremely low power requirements of Texas Instruments's MSP430F135, the Smart Tweezers device operates for over 80 h on three alkaline (357 A) button cells. For me, shelf life will most likely be the limiting factor on operational life. The auto-off feature will keep your batteries from running down should you forget to turn it off. Even with intermittent use in a production environment, your batteries should last weeks. If you don't want to be bothered with keeping a supply of batteries on hand, there is a rechargeable option.

You can upgrade your standard Smart Tweezers device any time because the inductive charger is field-upgradeable. Simply replace three alkaline cells in the Smart Tweezers with the 3.6-V NiMH battery and solder in the inductive receiver. The inductive transmitter is fully assembled and can run off either an AC wall wart or via USB (power only).

Inductive charging is becoming a hot item for cell phones and other devices. I think inductive charging deserves a column of its own, so let me know if you'd like to see more on that technology, or any other topic for that mater.

The Smart Tweezers device is just the kind of tool I would have liked to design. It combines a good mix of analog and digital circuitry into a compact, easy-to-use unit. The low-power design is kind on current consumption, which helps stretch battery life. You will find this tool invaluable when verifying, analyzing, and identifying components both in and out of circuit.

*Jeff Bachiochi (pronounced BAH-key-AH-key) has been writing for* Circuit Cellar *since 1988. His background includes product design and manufacturing. You can reach him at jeff.bachiochi@imaginethatnow.com or at www.imaginethatnow.com.*

## SOURCES

**Smart Tweezers Handheld LCR meter**
Advance Devices Corp.
www.advancedevices.com

**MSP430F135 Microcontroller**
Texas Instruments, Inc.
www.ti.com






# Beamforming 101

Beamforming enables you to find problems in your system before they occur. As Martin explains, you can perform your own investigation with a computer, a dual-channel A/D card, a pair of microphones, and some software.

Tensions are high and a hostile submarine lurks in the waters just off the coast. A patient exhibits troubling symptoms—do layers of tissue mask a tumor? A driver checks his mirrors and begins to change lanes but never sees the truck hiding in his blind spot.

It's better to go looking for trouble than to have it find you. But how do you do that? Whether it's sub-hunting sonar, medical ultrasound, or collision-avoidance radar, how do you uncloak danger?

One important arrow in your quiver is beamforming. Think of it as a way to focus and measure the energy falling on an array of sensors as you "look" in different directions. You can do your own hands-on investigation with only a computer, a dual-channel A/D card, a pair of microphones, and the software developed in this article.

Beamforming techniques come in two flavors: adaptive and fixed. Adaptive methods are haute cuisine. Fixed methods are burgers with fries. Each has its place, but blue-collar fare will be served up here.

## TIME DOMAIN BEAMFORMING

To understand how beamforming works, picture a line of identically spaced microphones. Next, imagine a wavefront consisting of noise plus the signal you want to detect, striking this uniform line array. Finally, make two common simplifications, valid for many applications. One, assume that the wavefront originated in the far field so that it can be approximated as a flat surface propagating toward the array and any curvature it possessed near the source can be disregarded. Second, ignore the three-dimensional quality of wave propagation and restrict your attention to the two-dimensional plane. With this in mind, Figure 1 shows what it would look like if you wanted to focus the energy arriving perpendicular (broadside) to the array.

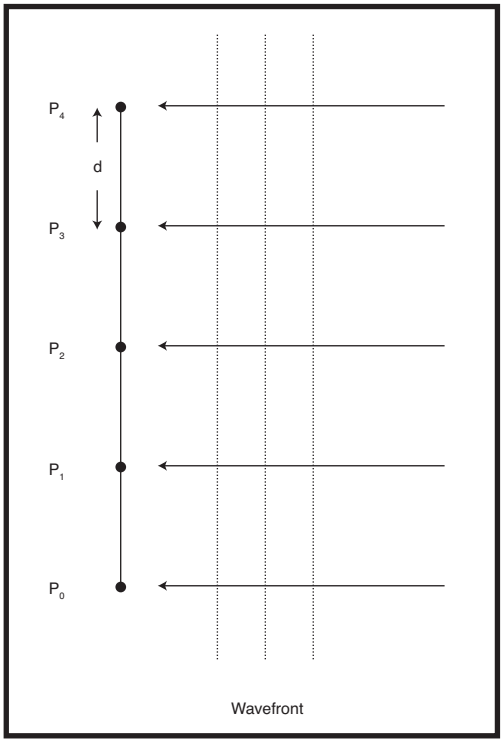



**Figure 1**—When a wavefront arrives from broadside, it strikes every sensor of a linear array at the same time.

Each phone is excited by the same wavefront at the same time. To get the total energy falling on the array at any instant, sum together the data from each phone. Refer to Figure 2 if you want to focus the energy from some other direction.

To sum the energy across a given wavefront, you need to insert delays because the wavefront has to travel a different distance to strike each phone. The extra distance the wavefront travels from one phone to the next is $d \sin \theta$, where d is the uniform spacing between the phones and $\theta$, the look direction, is by convention measured clockwise from broadside to the array. If the wavefront travels through the medium (e.g., air or water) at speed c, then

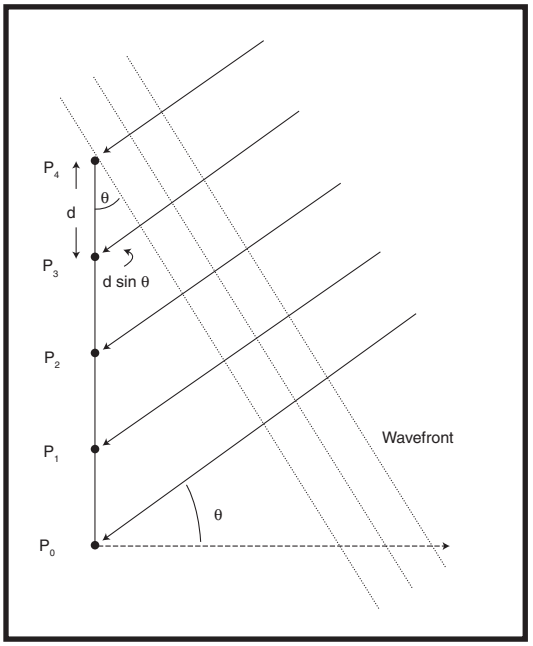



**Figure 2—**When a wavefront arrives from any direction other than broadside, it strikes each sensor at a different time. This time lag must be accounted for when processing the data. One way to proceed is to introduce compensating delays. For a uniform line array, the time lag is the same for any adjacent pair of sensors.

the difference between the time it strikes one phone and the next is (d sin θ)/c. Generalizing, from Figure 2 the wavefront strikes phone p at (pd sin θ)/c units of time before it strikes the reference phone. So, for phone p and time t, if you call the phone's response $y_p(t)$, then the time-shifted summed output of the line array is given by:

$$z(t) = \sum_{p=0}^{P-1} y_p\left(t - \frac{(pd \sin\theta)}{c}\right) \quad [1]$$

where P is the number of phones in the line array, numbered from 0 to P – 1.

Great, you're done. Just sweep around the array and find the direction that gives the maximum response, right? Actually, in practice, this time domain, delay, and sum beamformer has drawbacks because the data acquired is sampled, not continuous.

For an ADC clocking in data at some constant interval Δt, t can assume only values that are integer multiples of Δ. Of course, that also means y is known only at integer multiples of the sample interval as well. Equation 1 says you want the value read from each phone (the y values) at time:

$$t = n\Delta t = k\Delta t - \frac{(pd \sin\theta)}{c} \quad [2]$$

With Δt, d, and c fixed, and n and k restricted to be integers, this limits the available choices for the look direction, θ. What should you do?

## PHASE SHIFT BEAMFORMING

If you're looking for periodic signals, like the vibration produced by rotating machinery, you can use the Fourier transform and work with Equation 1 in the frequency domain instead of the time domain. (If you're not familiar with the Fourier transform, you can treat it as a black box whose output provides the magnitude and phase of the frequencies inherent in a time domain input.)

To begin, let the frequency-dependent function $Y_p(\omega)$ be the Fourier transform of the time-dependent function $y_p(t)$. Then, according to the shifting property, the Fourier transform of:

$$y_p\left(t - \frac{(pd \sin\theta)}{c}\right) \quad [3]$$

is as follows:

$$Y_p(\omega)e^{-j\omega\frac{(pd \sin\theta)}{c}} \quad [4]$$

Equation 1 transforms as:

$$Z(\omega) = \sum_{p=0}^{P-1} Y_p(\omega)e^{-j\omega\frac{(pd \sin\theta)}{c}} \quad [5]$$

The variables in Equation 5 are continuous. However, because your data is sampled, in practice you will apply the discrete Fourier transform with N bins, and for a representative bin n the discrete form of Equation 5 is:

$$Z_n = \sum_{p=0}^{P-1} Y_{pn}e^{-j\omega_n\frac{(pd \sin\theta)}{c}} = \sum_{p=0}^{P-1} Y_{pn}e^{-j\left(\frac{n2\pi f_s}{N}\right)\left(\frac{pd \sin\theta}{c}\right)} \quad [6]$$

where $f_S$ is the sampling rate. By doing this you've turned the time delay beamformer into a phase shift

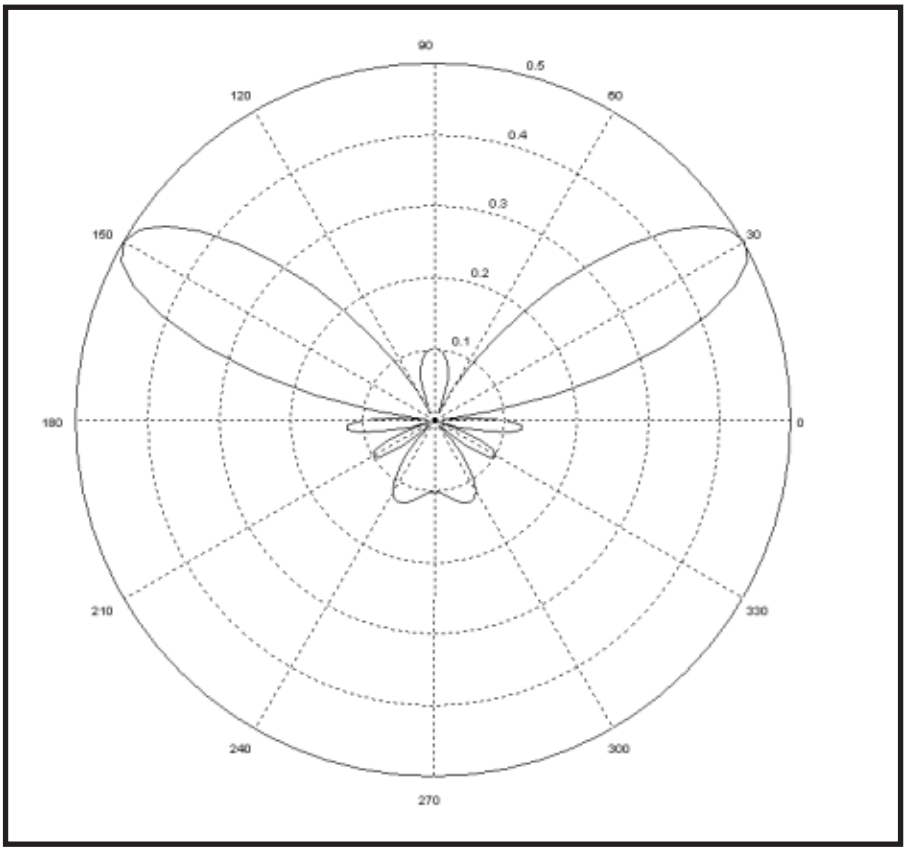



**Figure 3—**Plotting the peak power versus the look direction for a narrowband source results in a pattern of lobes. This figure was generated by steering the beamformer about a point and calculating the power spectrum for each look direction. The maximum value in each spectrum was then plotted.

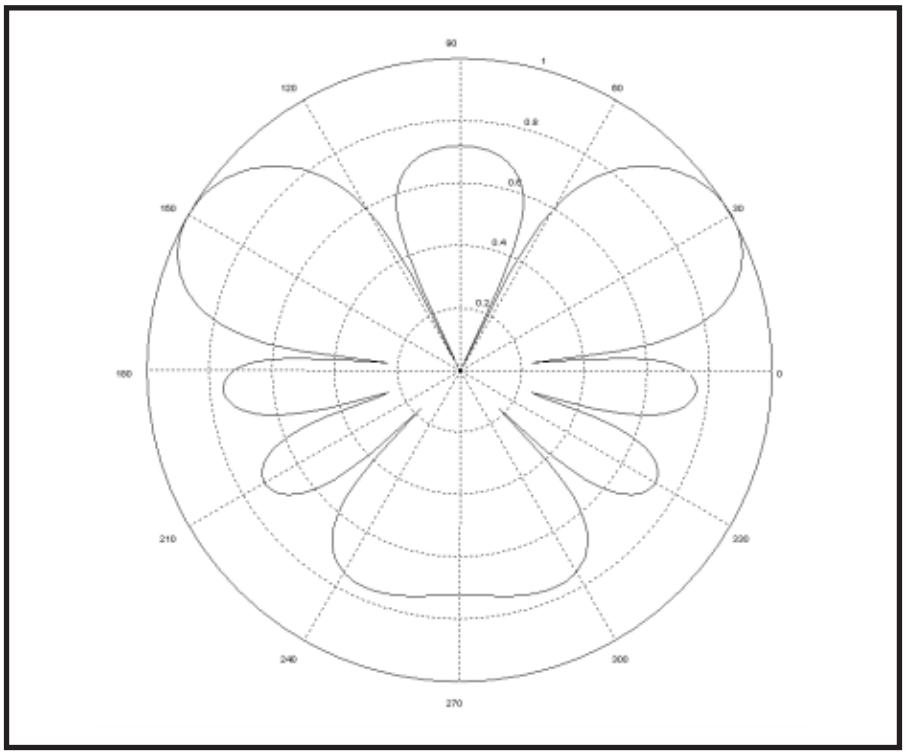


**Figure 4**—A beam pattern shows what the beamformer will produce when it is pointed in a fixed direction and a source at a given frequency moves around the array. Ideally, you would have pencil-thin lobes, and they would be present only in the look direction.

narrowband beamformer and you can point the steering vector wherever you want. The Narrowband_Beamformer.sce file posted on the *Circuit Cellar* FTP site shows how to apply Equation 6 using Scilab, a freeware alternative to Matlab.

The five-channel, 5-kHz sampled

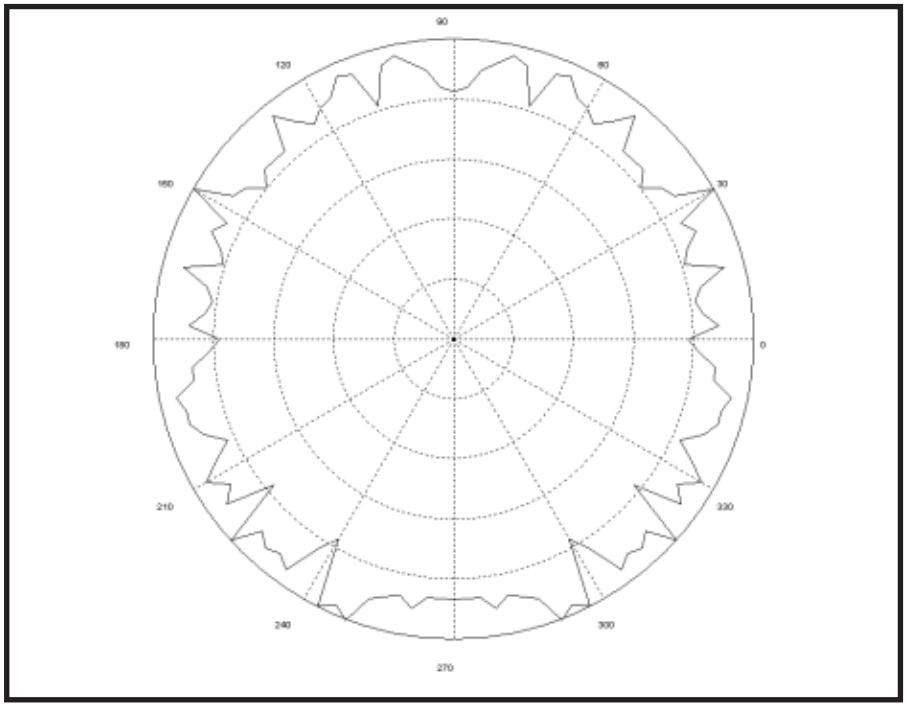
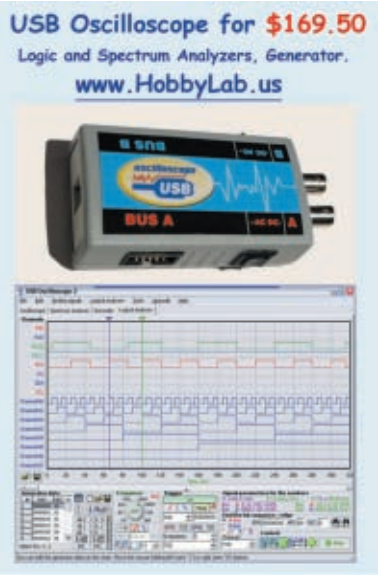


**Figure 5**—While the phase shift beamformer is useful for direction finding on a narrowband source, it is not effective when used on broadband energy. In the latter case, a plot of the power spectrum gives no hint where the source is located.










ICbank
Gateway to Electronics

# RS485 (3Port) to Ethernet

## Strong protocol converter board



Only
$170



## Powerful feature

- Protocol converter RS485 between Eternet
- Offer TCP/IP Communication to RS485 I/F board

| Specification | |
|---|---|
| **Network** | : TCP, UDP, DHCP, ICMP, IPv4, ARP, IGMP, PPPoE, Ethernet, Auto MIDI/MDIX, 10/100 Base-TX Auto negotiation (Full/half Duplex) |
| **Serial** | : RS485 3 Ports, 1200~115200 bps, Terminal block I/F Type |
| **Control program** | : IP Address & port setting, serial condition configuration, Data transmit Monitoring |
| **Accessory** | : Power adapter 9V 1500mA, LAN cable |
| **Etc** | : - DIP Switch(45 Baud Rate setting) - LED: Power, Network, 485 Port transmission signal |


ICbank
Gateway to Electronics

**ICbank Inc.**

**Youngsan Cho / Senior Account Manager / Sales & Marketing ICbank Inc.**

Tel : 82-707-019-3924. Fax : 82-2-6340-1888. Cell : 82-17-268-2078. Email : youngsan@icbank.com
18F Ace High -End 2, 222-14 Guro Digital Valley, Guro-dong, Seoul, 152-050, Korea

**Website : http://www.icbank.com/**

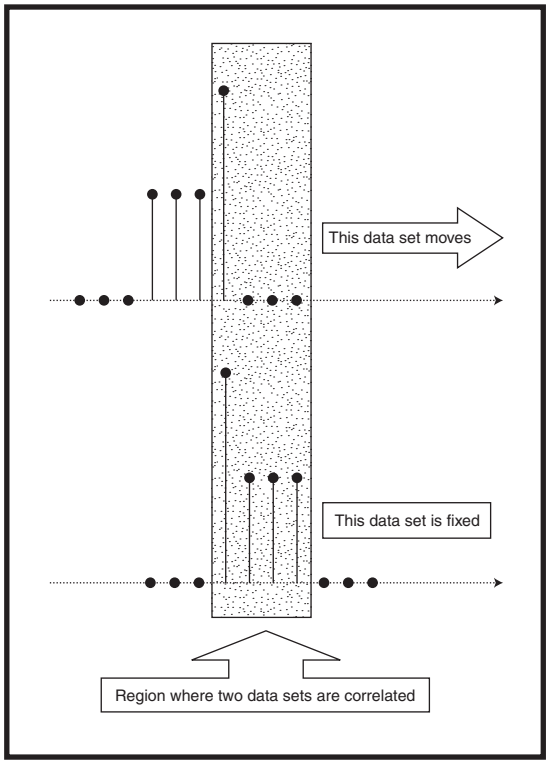



**Figure 6**—Here you see the correlation of two data sets. In the shaded region (the interrogation area), the vertically aligned points are multiplied and the products are summed to produce one output point. The upper-data set steps by one and the process repeats until all points have moved through the interrogation area.

wavefile (Narrowband.wav, posted on the *Circuit Cellar* FTP site) contains noise plus synthesized data for a 1,250-Hz narrowband signal traveling through a medium at 1,500 m/s (a typical sound propagation speed in ocean water), falling on a five-element linear array with uniform spacing of 0.6 m between phones, at an angle of arrival of 30°. (The phone spacing for this example was chosen deliberately. Ideally, you'd like the spacing to be half a wavelength or less.) When that data is processed using the code in the Narrowband_Beamformer.sce file posted on the *Circuit Cellar* FTP site and the maximum magnitude of the beamformer output (by frequency) is plotted for look directions around a circle in 3° increments beginning at 0°, the result is shown in Figure 3. The absolute maximum occurs when the array is steered to point at the source—just what you want! As a bonus, the use of an array improves the signal-to-noise ratio because the noise component tends to average out as more phones are added.

But there's more to the story. Note the shape of the beamformer output. It's telling you that if you point a little bit away from the source, your beamforming will leave the waves slightly out of phase, but you'll still pick up a good deal of power. But as you point farther and farther away from the source, the waves become increasingly out of phase and add destructively, causing the power received to drop rapidly. Keep going and the waves come back into phase a bit, adding more constructively before going back out of phase and adding destructively again. This process repeats as you look around a circle centered on the array.

The same idea holds if you fix the look direction and move the source around the array instead. This is the

AP CIRCUITS
PCB Fabrication Since 1984

As low as...
$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

*Unmasked boards ship next day!*

www.apcircuits.com

VISA MasterCard PayPal IPC MEMBER ASSOCIATION CONNECTING ELECTRONICS INDUSTRIES BBB

WIRELESS RS-232

WCSC (Willies Computer Software Co)



* Extremely easy to install and use
* Distances up to 100 meters (329 feet), 500 meters with extended antenna (1640 feet)
* Bluetooth Serial Profile
* One to One virtual cable or selectively connect one to many.
* Communicates with Bluetooth enabled PDAs, Laptops, Smartphones & Computer
* No special software required!

Other Products

- Professional serial communication libraries & development tools.
- PCI, PCMCIA, Universal PCI, & ISA multiport RS232, Rs422, & Rs485 cards
- Bluetooth/USB Dongles
- USB/Serial Adapters

http://CircuitCellar.wcscnet.com
sales@wcscnet.com 281.360.4232

$$h(-3) = 2 \times 2 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 4$$
$$h(-2) = 1 \times 2 + 2 \times 1 + 0 \times 1 + 0 \times 1 = 4$$
$$h(-1) = 1 \times 2 + 1 \times 1 + 2 \times 1 + 0 \times 1 = 5$$
$$h(0) = 1 \times 2 + 1 \times 1 + 1 \times 1 + 2 \times 1 = 6$$
$$h(1) = 0 \times 2 + 1 \times 1 + 1 \times 1 + 1 \times 1 = 3$$
$$h(2) = 0 \times 2 + 0 \times 1 + 1 \times 1 + 1 \times 1 = 2$$
$$h(3) = 0 \times 2 + 0 \times 1 + 0 \times 1 + 1 \times 1 = 1$$

**Figure 7—**Here is what you get when you apply Equation 10.

basis for plots called the beam pattern or radiation pattern of an array. The variables in these plots are the number of elements in the array, the location of the elements, the frequency of the wave, the angle of arrival of the wavefront, and the look direction. For a simple uniform line array, the beam pattern b(θ) can be expressed as:

$$20 \log \left\{ \frac{\left[\sin(nx)\right]}{\left[n \sin(x)\right]} \right\} \quad [7]$$
$$\text{where } x = \left[\frac{\pi d}{\lambda}\right]\left[\sin(\theta) - \sin(\theta_s)\right]$$

$\theta_s$ is the steering angle and $\lambda = c/f$ is the wavelength. Using Equation 7, the beam pattern for the array in the Narrowband_Beamformer.sce file on the *Circuit Cellar* FTP site for a wave at 1,250 Hz is plotted in Figure 4.

The large teardrop-shaped curve in the look direction is called the mainlobe. The other curves are called sidelobes. When plotting beam patterns for a line array, you need to deal with factors of 0/0. L'Hospital's rule (remember your calculus!) comes to your rescue here. Using it you can show that this ratio equates to 1 for the case at hand.

Why should you care about the beam pattern? Because narrowband tones may fall on your array from more than one direction. It can happen if the wavefront is reflected and takes multiple paths to your array. It can happen if there is more than one source present. In any event, these undesired interferers can feed into the sidelobes and adversely affect the output from the beamformer. Depending on your application, you may be able to ignore the problem caused by sidelobes. But if you can't, there are methods for minimizing them called array shading that apply




$$\frac{h(-3)}{1} = 4$$
$$\frac{h(-2)}{2} = 2$$
$$\frac{h(-1)}{3} = 1.67$$
$$\frac{h(0)}{4} = 1.5$$
$$\frac{h(1)}{3} = 1$$
$$\frac{h(2)}{2} = 1$$
$$\frac{h(3)}{1} = 1$$

**Figure 8—**Divide each sum by the size of the interrogation area.

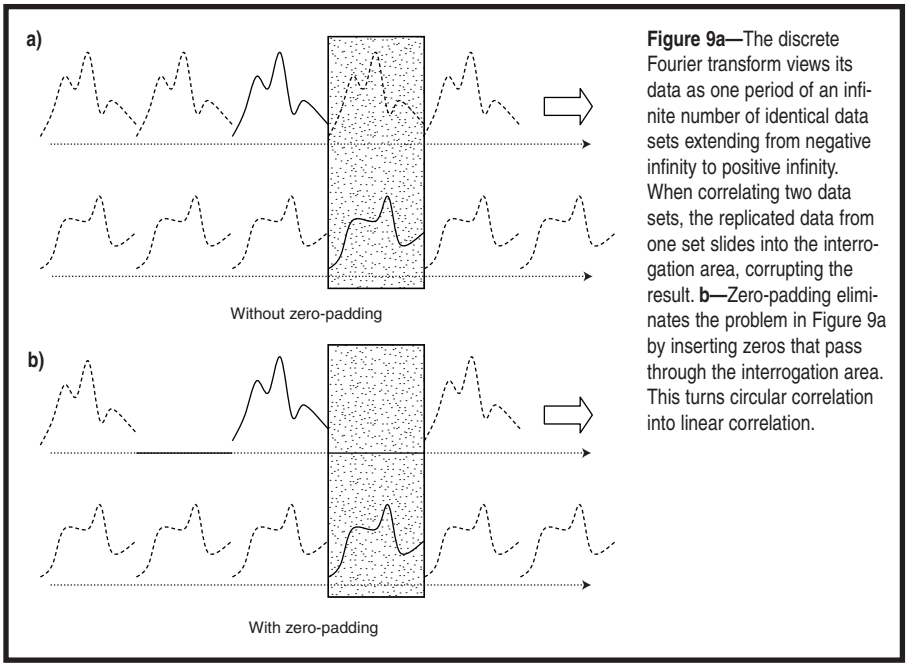



**Figure 9a—**The discrete Fourier transform views its data as one period of an infinite number of identical data sets extending from negative infinity to positive infinity. When correlating two data sets, the replicated data from one set slides into the interrogation area, corrupting the result. **b—**Zero-padding eliminates the problem in Figure 9a by inserting zeros that pass through the interrogation area. This turns circular correlation into linear correlation.

weights to the data on a phone by phone basis.

Let me make a final point concerning Figure 3 and Figure 4 regarding the symmetry they possess. From the standpoint of a line array for a given angle, the difference between the times a wavefront strikes any particular pair of phones is the same regardless of the side of the array from which the wavefront arrives. For this reason, depending on your application, you might want to consider using a different phone configuration or using two nonparallel line arrays.

What if the wavefront consists of broadband data and you can't resolve narrowband tones (e.g., the data in the BroadbandStochastic.wav file on the *Circuit Cellar* FTP site consists of noise)? The parameters are the same as those used with the Narrowband.wav file on the *Circuit Cellar* FTP site, except that the sample rate is 44.1 kHz and the phone separation is about 20.41 m. Making those adjustments to the code in the Narrowband_Beamformer.sce file on the *Circuit Cellar* FTP site and running it against this new file produces the output in Figure 5.

Although this wavefront is also falling on the array at an angle of 30° that's not what the phase shift beamformer is telling you. What do you do? The solution is to turn to the cross-correlation beamformer.

## CROSS-CORRELATING

Beamforming by cross-correlating works by comparing the wavefront received at pairs of phones. The data from one phone is fixed in time while you slide the data from the second phone past it. For discrete data, the sliding is really stepping, where each step represents one tick of the sample clock of your ADC (see Figure 6). At each step, calculate the similarity of the two data sets. The goal is to find the number of steps you have to shift the second data set to get the best agreement with the first data set. Once you know the shift as a lead or a lag (often nominally called a lag in either case), it's easy to determine the angle of arrival of the wavefront.

The first question is how do you measure similarity. That's where cross-correlation comes into the picture. For two functions f(t) and g(t), their cross-correlation is denoted by:

$$f \star g \qquad [8]$$

which is defined as:

$$h(\tau) = f \star g = \int_{-\infty}^{+\infty} f(t)g(t+\tau)dt \qquad [9]$$

which for a finite set of discretely

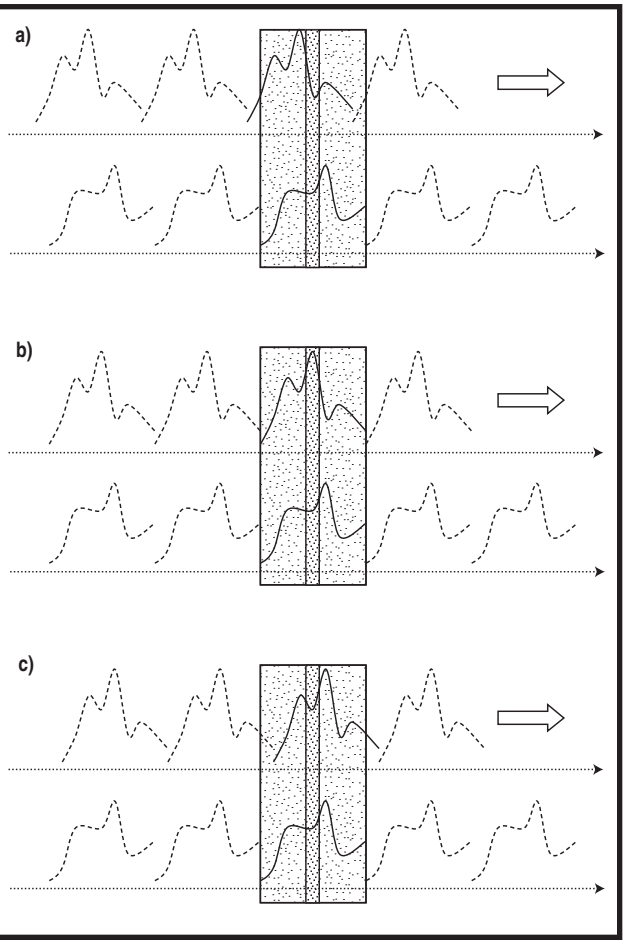



**Figure 10—**If the maximum lag (indicated by the width of the region between the two vertical lines in the center of the interrogation area) is relatively small, the effects of circular correlation will be relatively small (over the time of interest), as well. The impact will be further minimized if a window, such as a Hamming window (not shown), is applied to the data before the Fourier transform is performed, because it will taper the ends of the data toward zero. Figures **10a**, **10b**, and **10c** show the progression of the first data set and how little circular correlation comes into play for a small maximum lag.

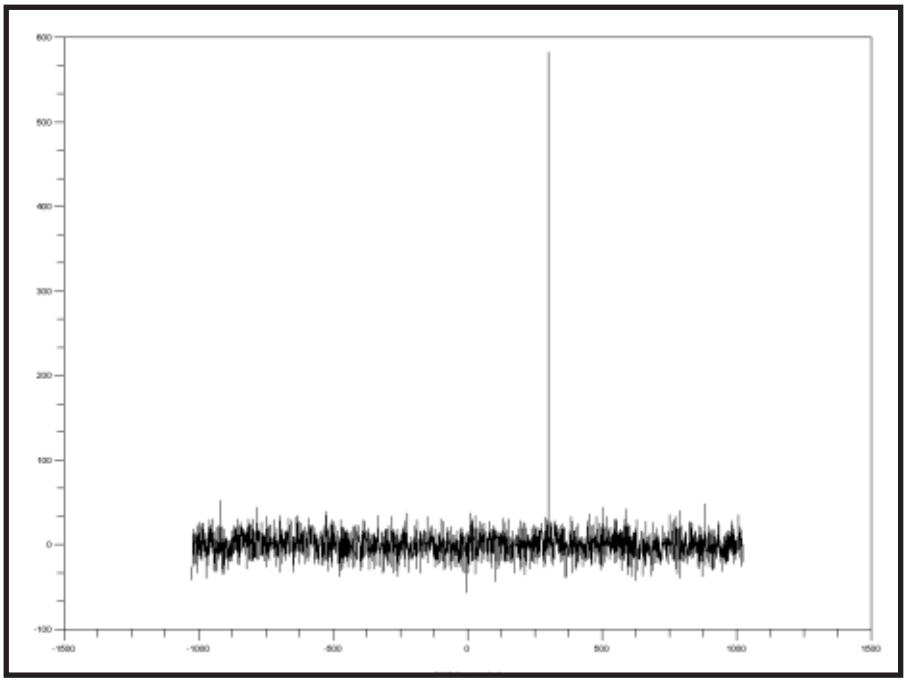


**Figure 11—**The cross-correlation beamformer is effective for direction finding on a broadband source. The strong peak indicates the number of sample delays needed to best match the data received at one sensor with the data received at the other sensor. The sample delay count can be readily converted into a bearing.

sampled data becomes:

$$h(m) = f \star g = \sum_{n=-N}^{N} f(n)g(n+m) \qquad [10]$$

Be careful about blindly applying Equation 10. Suppose you collected two sequences 2, 1, 1, 1 and 1, 1, 1, 2. Applying Equation 10 gives Figure 7. You can see that because the data sets are finite. The result at each stage is influenced by the number of nonzero products in the sum—more so, in fact, than by how well the peaks align. To adjust the result, you can divide each sum by the size of the interrogation area (see Figure 8). As expected, the maximum occurs when the peaks are aligned.

There is one drawback to Equation 10. It is computationally expensive for all but the smallest data sets. Once again, the solution is to use the discrete Fourier transform and work in the frequency domain. This is called fast correlation. Instead of generating a set of values where each value is formed by summing products (as in Figure 7), fast correlation requires only the discrete Fourier transform, a single pointwise multiply of the two transformed data sets, and an inverse discrete Fourier transform. It only *sounds* longer than the direct method!

In symbols, if x and y are the data sets to be correlated, and if F and $F^{-1}$ are the Fourier transform and inverse Fourier transform operators respectively, then:

$$x \star y = F^{-1}\{[F(x)][F(y)]^*\} \qquad [11]$$

where * means the complex conjugate.

There is a twist. The discrete Fourier transform assumes your data is periodic. That means that when you perform fast correlation, the result is the same as what you get by performing direct correlation on the two periodic sequences in Figure 9a. The periodic extension of the actual data bleeds into the sum of products and you've got a mess. This is called circular correlation (because you can equivalently think of the actual samples of the first data set as wrapping around in a circle as they march past the actual samples in the second data set). If you want to match the results of direct correlation (also called linear correlation), you need to zero-pad your input data (see Figure 9b).

But zero-padding is more work to code (in real-time C/C++ code, anyway). In general, if one data set is N samples long and the other set is M samples long, to correlate in the frequency domain you must tack zeros to the end of both sets so that each set has length N + M – 1. Then, following the inverse transform, you need to select the proper subset. More importantly, padding makes for larger, slower-to-execute Fourier transforms. So skip it! You can if the worst lag possible (which occurs when the wavefront arrives from endfire) is small relative to the number of sample counts by which the phones are separated. That's

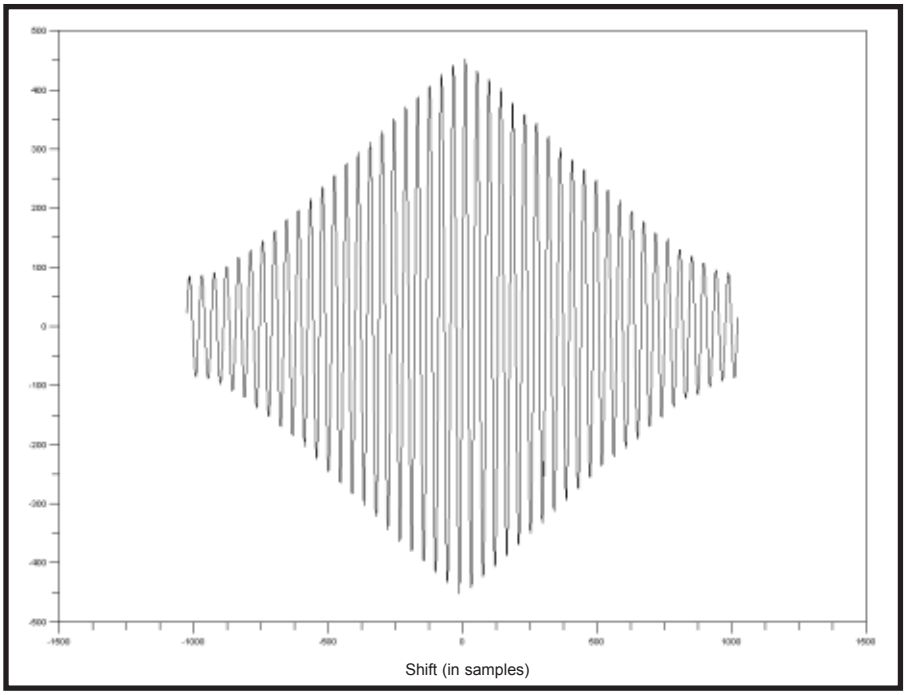



**Figure 12—**The basic cross-correlation beamformer can have trouble when both broadband and narrowband data are present. The dominant periodic peaks from the narrowband tones mask the relationship of the broadband data received at the two sensors.

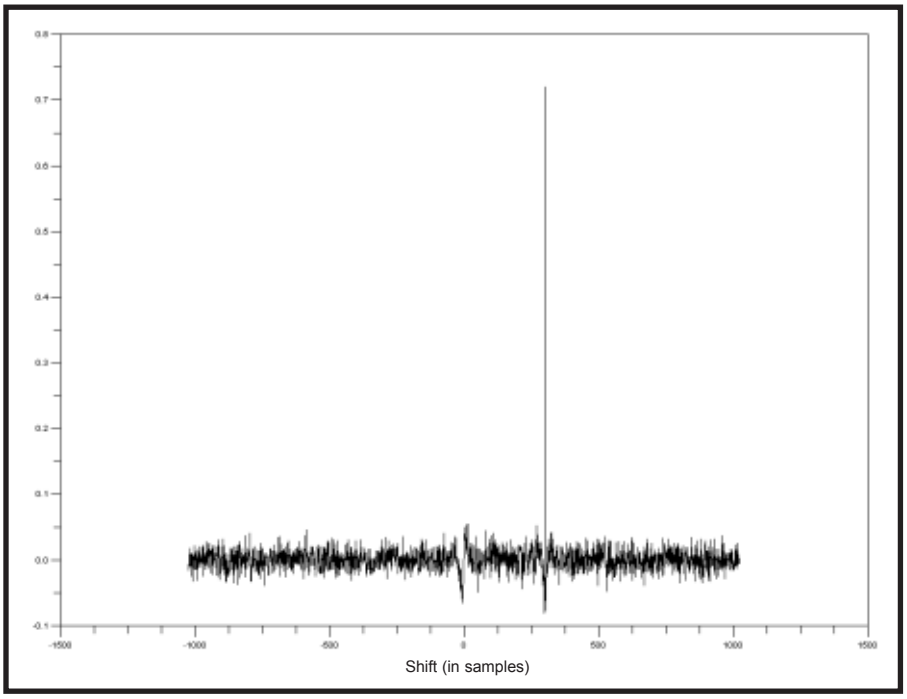



**Figure 13**—When the basic cross-correlation beamformer is modified by using the phase transform (PHAT), the relationship of the broadband data received by the two sensors is enhanced. The PHAT is just one of many modifications occasionally made to the cross-correlation beamformer.

because over the interval of interest, circularity will come into play for only a few samples (see Figure 10). Add to that the fact that windowing typically precedes the application of the Fourier transform, tapering the ends of the sequences toward zero, and you can see that the effects of circularity are further minimized.

Once you've determined the lag, all that's left is to turn it into a bearing. That's easy. As you can see in Figure 2, the time difference of arrival could be expressed as:

$$\Delta t = \frac{(d \sin \theta)}{c} \quad [12]$$

But note that:

$$\Delta t = \frac{L}{f_s} \quad [13]$$

where $f_s$ is the sample frequency and L is the number of lags required to align the data received at the relative phone with the data received at the reference phone. (A delay is a positive shift and an advance is a negative shift.) So, the bearing is given by:

$$\theta = \sin^{-1}\left(\left(\frac{L}{f_s}\right)\left(\frac{c}{d}\right)\right) \quad [14]$$

The Broadband_Beamformer.sce file on the *Circuit Cellar* FTP site pulls these ideas together with Equation 11 to calculate a broadband bearing.

When this code is run against the BroadbandStochastic.wav file on the *Circuit Cellar* FTP site, the result is shown in Figure 11. The peak occurs at a shift of 300 sample counts, which translates into a bearing close to the actual value of 30°.

Typically, when beamforming via cross-correlation, you can use multiple phone pairs—for N phones you can form N(N – 1)/2 phone pairs—and add the results from each. This gives a better estimate of direction of arrival. It is also especially valuable if the phones are not collinear because it eliminates bearing ambiguity. The true bearings will add constructively while the false bearings will add randomly and contribute to the noise.

If there is a complaint to be had about the cross-correlation beamformer, it is that it will be fooled by a noticeable narrowband tone within a stochastic broadband signal. But there's a solution for that too!

A slight modification to the cross-correlation beamformer makes a big difference in the result. After multiplying the Fourier transform of the two input sequences, the phase transform, or PHAT, divides the complex value in each frequency bin by its magnitude. (This is equivalent to applying a filter to the original time series data because multiplication in the frequency domain corresponds to convolution in the time domain.) With the magnitudes equalized, the correlation then relies on only the phase of the input sequences. Interfering narrowband tones are neutralized. To see what a dramatic difference this can make, refer to Figure 12 (cross-correlation without the PHAT) and Figure 13 (cross-correlation with the PHAT). The input file (BroadbandStochasticPlusTone.wav on the *Circuit Cellar* FTP site) is the same as BroadbandStochastic.wav, except for an added narrowband tone at a bearing of 64°.

Despite their low-brow status, fixed beamformers make their way into commercial products, where they prove their worth. Here's hoping they can point you in the right direction too, so the next time you go looking for trouble, you find it!

*Formally trained in mathematics, Martin Courtney has been writing software for more years than he cares to admit. Although his preference is signal processing and algorithm development, he has written code for real-time embedded systems and graphically intensive PC-based applications as well. You can reach Martin at mcourtn1@san.rr.com.*

## PROJECT FILES

To download additional files, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/221.

## RESOURCES

W. S. Burdic, *Underwater Acoustic System Analysis*, Peninsula Publishing, Los Altos, CA, 2002.

R. V. Churchill, *Operational Mathematics*, McGraw-Hill, New York, NY, 1972.

Scilab, www.scilab.org.

# An Introduction to Verilog

If you are new to programming FPGAs and CPLDs or looking for a new design language, Kareem has the solution for you. In this article, he introduces you to Verilog. Although the hardware description language has been used in the ASIC industry for years, it has all the tools to help you implement complex designs, such as a creating a VGA interface or writing to an Ethernet controller.

Programmable logic has been around for well over two decades. Today, due to larger and cheaper devices on the market, FPGAs and CPLDs are finding their way into a wide array of projects, and there is a plethora of languages to choose from. VHDL is the popular choice outside of the U.S. It is preferred if you need a strong typed language. However, the focus of this article will be on another popular language called Verilog, which is a hardware description language that is similar to the C language.

Typically, Verilog is used in the ASIC design industry. Companies such as Sun Microsystems, Advanced Micro Devices, and NVIDIA use Verilog to verify and test new processor architectures before committing to physical silicon and post-fab verification. However, Verilog can be used in other ways, including implementing complex designs such as a VGA interface. Another complex design such as an Ethernet controller can also be written in Verilog and implemented in a programmable device.

This article is mostly tailored to engineers who need to learn Verilog and do not know or know little about the language. Those who know VHDL will benefit from reading this article as well and should be able to pick up Verilog fairly quickly after reviewing the example listings and referring to the Resources at the end of the article. This article does not go over hardware, but I have included some links that will help you learn more about how the hardware interacts with this language at the end.

## THE VERILOG LANGUAGE

First, it is best to know what variable types are available in Verilog. The basic types available are: binary, integer, and real. Other types are available but they are not used as often as these three. Keep everything in the binary number system as much as possible because type casting can cause post-implementation issues, but not all writers are the same. Binary and integer types have the ability to use other values such as "z" (high impedance) and "x" (don't care). Both are nice to have around when you want a shared bus between designs or a bus to the outside world. Binary types can be assigned by giving an integer value. However, there are times when you want to assign or look at a specific bit. Some of the listings use this notation. In case you are curious, it looks like this: `X'wY`, where `X` is the word size, `w` is the number base—`b` for binary, `h` for hex—and `Y` is the value. Any value without this is considered an integer by default. Keeping everything in binary, however, can become a pain in the neck especially

| Type | Length | Notes |
|---|---|---|
| Binary | 1-bit to n-bit | No type declaration needed just variable name (unsigned) |
| Wire | 1-bit to n-bit | A connection that does not store its value (typically used in combinational logic and module connections, some derivatives have tristate logic) (unsigned). |
| Reg | 1-bit to n-bit | Similar to wire but needs to be used in sequential designs. It retains its value until updated (unsigned). |
| Integer | 32-bit | Thirty-two-bit integer (a binary word can be signed) |
| Real | 64-bit | Double-precision word. When used, its value is truncated to a whole number (signed). |

**Table 1**—These are basic variable types frequently used in Verilog.

| Operator | Symbol | HW Equivalent | Notes |
|---|---|---|---|
| Addition | A + B | Binary adder | (Binary and integer result) |
| Subtraction | A – B | Binary subtraction | (Binary and integer result) |
| Shift left | M << N | Logical binary shifter | Shift M left N times and appends zeros at LSB. N is treated as an integer. |
| Shift right | M >> N | Logical binary shifter | Shift M right N times and append zeros at MSB. (Does not sign extend.) |
| | | | |
| Relational | A > B  A < B  A >= B  A <= B | Logical comparator | If an input value is "X" or "Z," the result is automatically false. This is for all relational operators. |
| | | | |
| Equality | A == B  A !=B | Logical comparator | If an input value is "X" or "Z," the result is "X." |
| | | | |
| Concatenate | {M, N, …} | Group words | Groups words into a larger word with M being the most significant (binary result) |

**Table 2—**These are commonly used operators in Verilog.

when dealing with numbers larger than 8 bits. Table 1 shows some of the variable types that are available in Verilog. Integer is probably the most useful one to have around because it's 32 bits long and helps you keep track of numbers easily. Note that integer is a signed type but can also be set with all "z" or "x." Real is not used that much, when it is used the number is truncated to an integer. It is best to keep this in mind when using the real type, granted it is the least popular compared to binary and integer. When any design is initialized in a simulator, the initial values of a binary and integer are all "x." Real, on the other hand, is 0.0 because it cannot use "x." There are other types that are used when interconnecting within and outside of a design. They are included in the table, but won't be introduced until later.

Some, but not all, operators from C are in Verilog. Some of the operators available in Verilog are in Table 2. It isn't a complete list, but it contains most of the more commonly used operators. Like C, Verilog can understand operations and perform implicit casting (i.e., adding an integer with a 4-bit word and storing it into a binary register or even a real); typically this is frowned on mostly due to the fact that implicit casting in Verilog can open a new can of worms and cause issues when running the code in hardware. As long as casting does not give any erroneous results during an operation, there should be no show-stoppers in a design. Signed operation happens only if integers and real types are used in arithmetic (add, subtract, multiply) operations.

## VERILOG MODULES

In Verilog, designs are called modules. A module defines its ports and contains the implementation code. If you think of the design as a black box, Verilog code typically looks like a black box with the top missing. Languages like Verilog and VHDL encourage black box usage because it can make code more readable, make debugging easier, and encourage code reuse. In Verilog, multiple code implementations cannot have the same module name. This is in stark contrast to VHDL, where architectures can share the same entity name. The only way to get around this in Verilog is to copy a module and rename it.

In Listing 1, a fairly standard shift register inserts a binary value at the end of a byte every clock cycle. If you're experienced with VHDL, you can see that there aren't any library declarations. This is mainly due to the fact that Verilog originated from




an interpretive foundation. However, there are include directives that can be used to add external modules and features. Obviously, the first lines after the `module` statement are defining the modules' port directions and type with the reserved words `input` and `output`. There is another declaration called `inout`, which is bidirectional but not in the listing. A module's input and output ports can use integer and real, but binary is recommended if it is a top-level module.

The `reg` statement essentially acts like a storage unit. Because it has the same name as the output port it acts like one item. Using `reg` this way is helpful because its storage ability allows the output to remain constant while system inputs change between clock cycles. There is another kind of statement called `wire`. It is used to tie more than one module together or drive combinational designs. It will appear in later listings.

The next line of code is the `always` statement or block. You want to have a `begin` and `end` statement for it. If you know VHDL, this is the same as the `process` statement and works in the same fashion. If you are completely new to programmable logic in general, it works like this: "For every action X that happens on signals indicated in the sensitivity list, follow these instructions." In some modules, there is usually a

**Listing 1—**This is an example of a shift register modeled in Verilog.

```
module shiftr (clk, inbit, outbyte); // module name and its connectors

// direction definitions
// if there is no type to the dedition its automatically binary
input clk;
input inbit;

output [7:0] outbyte;

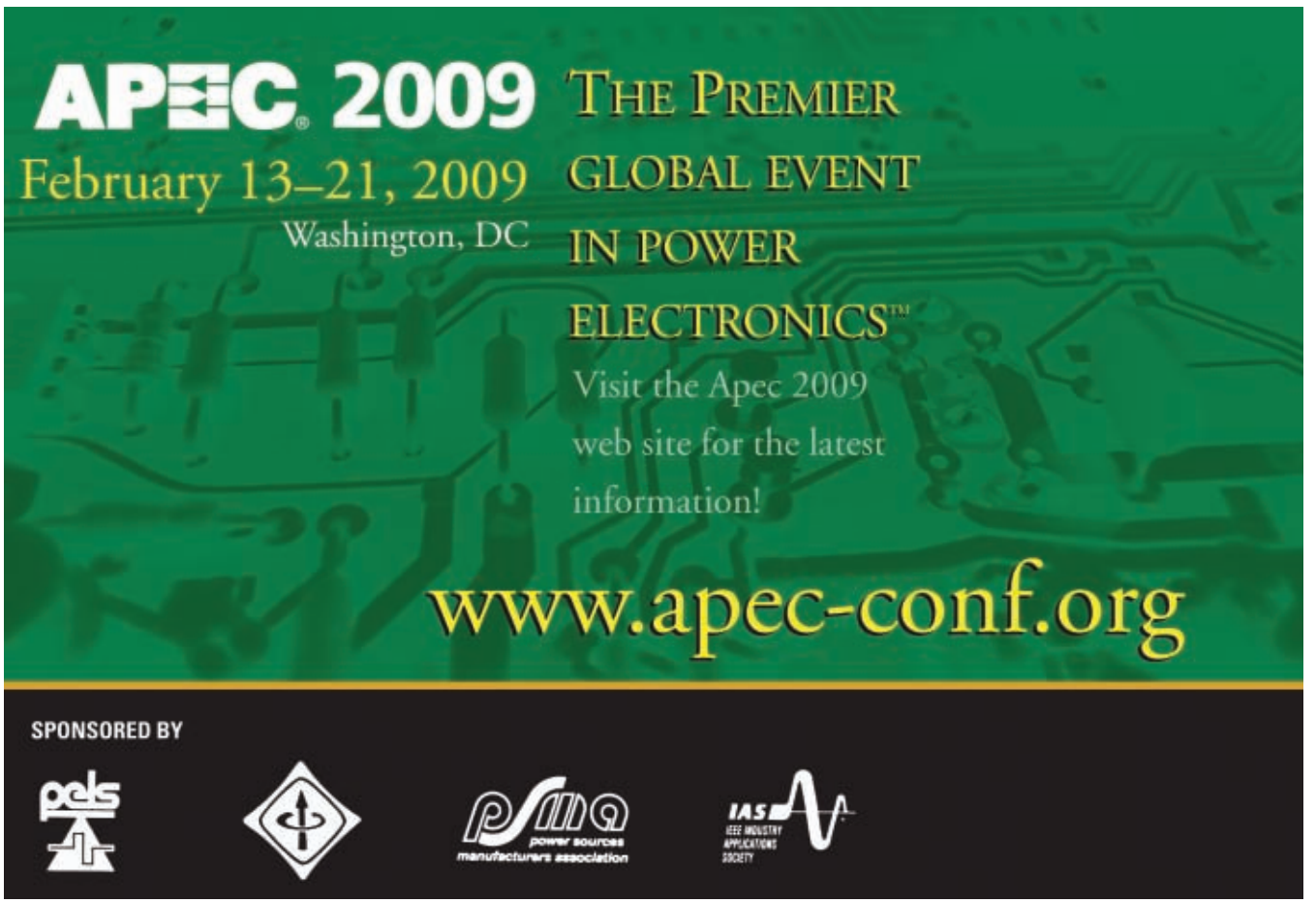
// bucket to hold the output
reg [7:0] outbyte;

// implementation
always @ (posedge clk)
begin // block of code that triggers when the clk has a positive edge

              outbyte <= outbyte << 1;
              outbyte[0] <= inbit;

end

endmodule  // this is the end of the module
```



**Listing 2**—This is an example of using functions to perform simple math.

```
module listing2(itema, itemb, resultc, opcode);
input [31:0] itema;
input [31:0] itemb;
input [1:0] opcode;

output [7:0] resultc;// function

reg [31:0] resultc;

// function
function integer myalu;
input integer a; // arguments
input integer b;
input [1:0] opc;
begin // function block

    case(opc) // case statement
         // (if X'bY did not exist, we would have to use 0, 1, 2, and 3)
        2'b00: myalu = a + b;      // binary add
        2'b01: myalu = a - b;      // binary subtract
        2'b10: myalu = a << b;     // shift a left b times
        2'b11: myalu = a & b;      // logical and
        default: myalu = 0;
    endcase
end

endfunction

assign resultc = myalu(itema, itemb, opcode);

endmodule
```

**Listing 3**—This is an example of using tasks to keep tank contents equal.

```
module listing3(tank1, tank2, pump1, pump2);

input [7:0] tank1;
input [7:0] tank2;

output pump1;
output pump2;

`include "tankcheck.v"

always begin
            check(tank1, tank2, pump1, pump2);
end

endmodule

// tankcheck.v contents
task check;
input [7:0] vol1;
input [7:0] vol2;
output switch1;
output switch2;

if(vol1 == vol2) begin
            switch1 = 0;
            switch2 = 0;
end
else if(vol1 > vol2) begin
            switch1 = 0;
            switch2 = 1;
end
else  begin // vol2 > vol1
            switch2 = 0;
            switch1 = 1;
end

endtask
```



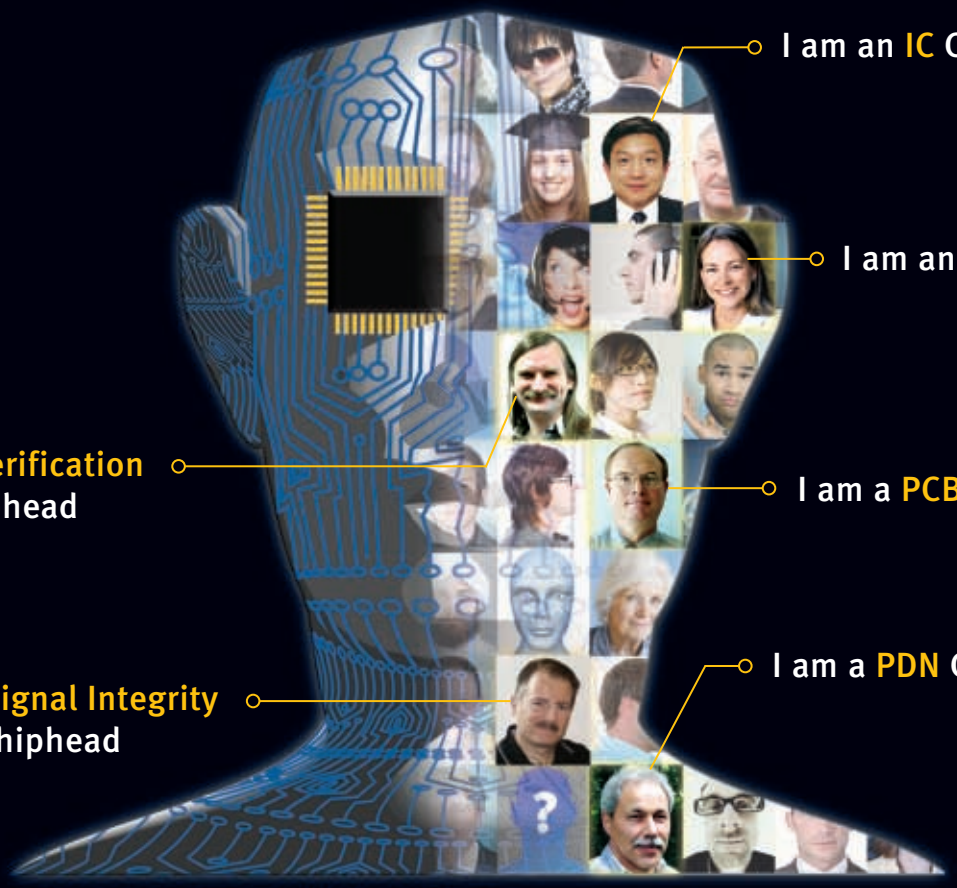

# DesignCon 2009

The BRIGHTEST Minds in Electronic Design

CONFERENCE February 2–5, 2009
EXHIBITION February 3–4, 2009

Santa Clara Convention Center
Santa Clara, California

# Are You a Chiphead?



I am an IC Chiphead

I am an IP Chiphead

I am a Verification Chiphead

I am a PCB Chiphead

I am a Signal Integrity Chiphead

I am a PDN Chiphead

**What kind of Chiphead are you?**
**Go to designcon.com**



PRESENTED BY

iec.org

OFFICIAL HOST SPONSOR

Agilent Technologies

SPECIAL FEATURE

IP SUMMIT at DesignCon

**Listing 4—**This is a shift register and parity generator tied in different modules.

```
module listing4 (clk,
                      inbit, out9word);

// input words
input clk;
input inbit;

// output words
output [8:0] out9word;

// wires
wire [7:0] shiftByte;
wire paritybit;


// instances of modules
// format: module_name   unique_inst_name   port_map
shiftr u0 (clk, inbit, shiftByte);
epg u1 (shiftByte, paritybit);

assign out9word = {paritybit, shiftByte[7:0]};

endmodule

// file epg.v
module epg(inbyte,
                  outparity);

input [7:0] inbyte;
output outparity;

wire outparity;

assign outparity = (inbyte[0] ^ inbyte[1] ^
                                    inbyte[2] ^ inbyte[3] ^
                                    inbyte[4] ^ inbyte[5] ^
                                    inbyte[6] ^ inbyte[7]);
endmodule


// file shiftr.v
module shiftr (clk, inbit, outbyte);

// port defs
input clk;
input inbit;

output [7:0] outbyte;

// bucket to hold output
reg [7:0] outbyte;

// implementation
always @ (posedge clk)begin
          outbyte <= outbyte << 1;
          outbyte[0] <= inbit;
end

endmodule
```

`begin` and an `end` statement. This is the equivalent of curly braces seen in C/C++. It's best to use these with decision structures (i.e., `always`, `if`, and `case`) as much as possible.

Finally, the last statement is a logical left shift operation. Verilog bitwise operators in some instances need the keyword `assign` for the operation to happen. The compiler will tell you if an `assign` statement is missing. From there, the code does its insertion operation and then waits for the next positive edge of the clock. This was a pretty straightforward example; unfortunately, it doesn't do much. The best way to get around that is to add more features using functions, tying-in more modules, or using parameters to increase flexibility.

## TASKS & FUNCTIONS

Tasks and functions make module implementation clearer. Both are best used when redundant code or complex actions need to be split up from the main source. There are some differences between tasks and functions.

A task can call other tasks and functions, while a function can call only other functions. A task does not return a value; it modifies a variable that is passed to it as an output. Passing items to a task is also optional. Functions, on the other hand, must return one and only one value and must have at least one value passed to them to be valid. Tasks are well-suited for test benches because they can hold delay and control statements. Functions, however, have to be able to run within one time unit to work. This means functions should not be used for test benches or simulations that require delays or use sequential designs. Experimenting is a good thing because these constructs are helpful.

There is one cardinal rule to follow when using a function or task. They have to be defined within the module, unlike VHDL where functions are defined in a package to get maximum flexibility. Tasks and functions can be defined in a separate file and then attached to a module with an `include` statement. This enables you to reuse code in a project or across multiple projects. Both tasks and functions can use types other than binary for their input and output ports, giving you even more flexibility.

Listing 2 contains a function that essentially acts like a basic ALU. Depending on what is passed to the

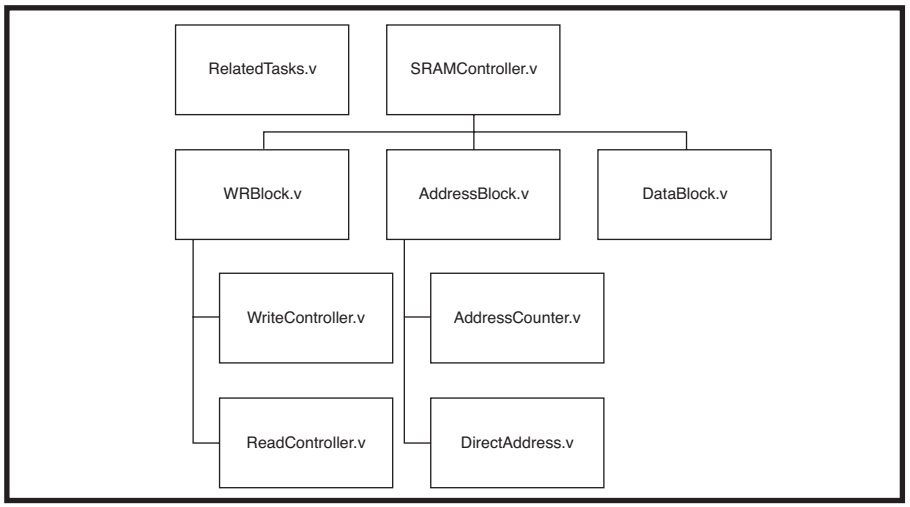



**Figure 1—**This is a top-level design view of a Verilog project.

function, the function will process the information and return the calculated integer value. Tasks work in the same way, but the structure is a little different when dealing with inputs and outputs. As I said before, one of the major differences between a task and a function is that the former can have multiple outputs, rather than just one. This gives you the ability to make a task more complicated internally, if need be.

Listing 3 is an example of a task in action with more than one output. Note how it is implemented the same way as a function. It has to be defined and called within the module in order to work. But rather than define the task explicitly within the module, the task is defined in a separate file and an include directive is added in the module code just to show how functions and tasks can be defined outside of a module and available for other modules to use.

## BUILDING WITH MODULES

If too much is added to a module, it can become so large that debugging and editing become a chore. Doing this also minimizes code reuse to the point where new counters and state machines are being recreated when just using small modules/functions from a previous project is more than adequate. A good way to get around these issues is by making multiple modules in the same file or across multiple files and creating an instantiation of that module within an upper-level module to use its abilities. Multiple modules are good to have for a pipelined system. This enables you to use the same kind of module over multiple areas of a system. Older modules can also be used this way so less time is used on constant recreation.

That is the idea of code reuse in a nutshell. Now I will discuss an example of code reuse and multiple modules. The shift register from Listing 1 is having its data go into an even parity generator and the result from both modules is output through the top-level module in Listing 4. All of this is done across multiple files in one listing for easier reading. In all modular designs, there is always a module called a top-level entity, where all of the inputs and outputs of a system connect to the physical world. It is also where lower-level entities are

**Listing 5**—In this instance, parameters are overloading two modeled memories.

```
module dummy(in1, out1);

// local parameters
parameter x = 1;
parameter bus1w = 10;
parameter bus2w = 8;
parameter intbus = 1 << (bus1w + bus2w);

input [bus1w-1:0] in1;
output [bus2w-1:0] out1;

// rest of dummy implementation here

endmodule

module listing5 (inbusa, inbusb, outbusa, outbusb);

// local parameters
parameter a = 16;
parameter b = 10;

input [a-1:0] inbusa;
input [b-1:0] inbusb;

output [b-1:0] outbusa;
output [a-1:0] outbusb;

wire [b-1:0] outbusa;
wire [a-1:0] outbusb;

// defparam method
defparam I0.x = 2;
defparam I0.bus1w = a;
defparam I0.bus2w = b;

dummy I0 (inbusa, outbusa);

// newer method
dummy  #(.x(3),
                    .bus1w(b),
                    .bus2w(a)) I1(inbusb, outbusb);

endmodule
```

**Listing 6**—Test benches in Verilog are modules with no ports and have an instance of the top level-module within.

```
module listing6tb ();

reg clktb;
reg resettb;
wire [7:0] counttb;

// create an instance of the module
listing6 mycounter(.clk (clktb),
                   .reset (resettb),
                   .count (counttb));

// setup the initial state of the system
initial
begin
$display ("Time\tClk\tReset\tCount");

$monitor("%d\t%b\t%b\t%b", $time, clktb, resettb, counttb);

clktb = 0;
resettb = 0;
counttb = 0;

#20 // delay for 20 units

// test reset condition
#20 resettb = 1;
#20 resettb = 0;

// stop testing at 200 time units
#200 $finish;
end

// clock generation
always begin
  #10 clktb = !clktb; // every 10 time units invert clocks value
end

endmodule

// counter module
module listing6 (clk, reset, count);

// inputs
//input clk;
input reset;
input clk;

// outputs
output [7:0] count;

// buffers
reg [7:0] count;

// implementation
always @(posedge clk, posedge reset)
begin

            if(reset) begin
                    count <= 8'b0; // reset counter
            end
            else begin
                    count <= count + 1; // inc counter
            end
end

endmodule
```

spawned. Subordinates can spawn entities below themselves as well (see Figure 1). Think of it as a large black box with smaller black boxes connected with wires and those small black boxes have either stuff or even smaller black boxes. Pretty neat, but it can get annoying. Imagine a situation where a memory controller for 10-bit addressing is created and then the address length needs to be extended to 16 bits. That can be a lot of files to go through to change 10 to 16. However, with parameters all that needs to be changed is one value in one file and it's all done.

## PARAMETERS

Parameters are great to have around in Verilog and can make code reuse even more attractive. Parameters allow words to take the place of a numerical value like `#define` in C, but with some extra features such as overriding. Parameters can be put in length descriptors, making it easy to change the size of an output, input, or variable. For example, if a VGA generator had a color depth of 8 bits but needed to be changed to 32-bit color depth, then instead of changing the locations where the value occurs, only the value of the parameter would be changed and when the module was recompiled it would be able to display 32-bit color. The same can be done for memory controllers and other modules that have ports, wires, or registers with 1 bit or more in size. Parameters can also be overridden. This is performed just before or when a module is instantiated. This is helpful if the module needs to be the same all the time across separate projects that are using the same source, but needs to be a little different for another project. Parameters can also be used in functions and tasks as long as the parameter is in the same file the implementation code is in. Parameters with functions and tasks give Verilog the flexibility of a VHDL package, granted it really isn't a package, because the implementation is located in a module and not in a separate construct.

There are many ways to override

parameters. One way is by using the `defparam` keyword, which explicitly changes the value of the parameter in the instantiated module before it is invoked. Another way is by overriding the parameter when the module is being invoked. Listing 5 shows how both are done with dummy modules that already have defined parameters. The `defparam` method is from an older version of the language, so depending on the version of Verilog being used, make sure to pick the right method.

## TEST BENCHES

Any system, regardless of whether it's a simple multiplexer or a large complex memory controller, should be run through a test bench for testing and simulation before moving into a hardware medium for physical verification. A test bench is really just a normal Verilog module but with unsynthesizable code such as delays, system directives, and loops. Some test bench code can be put directly into a normal module like delays. When the compiler synthesizes, it will just ignore the delays and continue onward to fitting and timing analysis. Still, it is best to try and keep all of this kind of code in the test bench module and not in the unit under test.

Designing a test bench is like making a new top-level module, this time with no ports defined, and with an instance of the main project module inside. From there, test code can be added, comprising keywords like `initial`, which sets the initial state of the test bench when started. System commands like `$monitor`, `$display`, and `$finish`, which can be used to display values of signals when they change, display signal values when called, and then stop the simulation after so much time has passed respectively are also good to have in a test bench. Another directive that is useful is timescale. Timescale defines how big a time unit is within a test bench. There is no default value for a time unit and it needs to be defined in the test bench for a simulation to occur.

Listing 6 is an example of a test bench for an 8-bit up-down counter with async reset. It runs through a simulation for 200 time units (with a time unit equal to 1 ns). The `$monitor` command prints any changes that happen to the input and output signals. This is a common test bench used in Verilog. It's usually best to pick up a book on test benches to learn how to write more elaborate and thorough test benches. In industry, verification and testing of designs alone account for a huge amount of work. Most people on a design team will do verification work to make sure the architecture designed operates as intended and does not lock into any invalid states. For small or hobbyist projects, a simple simulation should suffice, unless the design runs into errors that lock up the system. If that happens, run the code through a test bench to find where the issue is located in code.

## MANAGING LARGE PROJECTS

Regardless of what is done in a programmable logic project, it is going to get huge. Although design abstraction at such a high level makes code more understandable to a human, it still needs to be verbose enough for the compiler to synthesize. VHDL tends to excel at large designs due to having actual packages, configuration statements, and generics. Verilog has some of these features. Initially, the language did not because it was made for modeling lower-level constructs. But, as time progressed, newer versions were released that made higher level abstraction available. When managing large projects in Verilog, keeping track of all the functions/tasks, modules, and parameters used gets hard. IDEs like Xilinx ISE and Altera Quartus II help manage project files and module levels. They should be used when writing Verilog projects because they help keep a project on track during development.

Other good ideas include keeping most functions and tasks in a separate file, and trying to keep parameters in a separate file as well. This way, when system-wide changes are needed or modules need more abilities, all that has to be added to the code are include statements. This is a bit crude when compared to VHDL and packages, but it gets the job done. Don't worry if the function file has 100 functions. The compiler will synthesize only the number of function instantiations used in the actual implementation code.

Take care of the order of compilation. Keeping track of this will ensure simulation and test results stay the same and do not change radically. It is best to ensure the type of top-level module and make sure all other modules are in their correct levels in the file hierarchy. If these guidelines are followed, there shouldn't be any issues when writing a Verilog project.

Designing with a programmable logic controller is probably the most fun and rewarding experience an engineer can have. If you have any questions, or if you want to start practicing the language, refer to the Resources section of this article.

*Kareem Matariyeh (asm2750@gmail.com) graduated with a B.S. in computer engineering from the University of Nevada, Las Vegas, where he focused primarily on programmable logic, verification and testing, and embedded systems. His final design project involved implementing an HTDV pattern generator using a CPLD and HDMI transmitter. Kareem's interests include RC aircraft and PC modding.*

## RESOURCES

ASIC World, A great source of Verilog tutorials, www.asic-world.com.

Information about large projects, www.opencores.com.

W. K. Lam, *Hardware Design Verification: Simulation and Formal Method-Based Approaches*, Prentice Hall, Upper Saddle River, NJ, 2005.

A programmable logic community web site, www.fpga4fun.com.

# Saelig

## UNIQUE PRODUCTS & SUPPORT

www.saelig.com

Testgear

### USB Bus Analyzers




USB1.1/2.0/WiMedia analyzers and now - **EX260** can test USB3.0 5Gbps too! Emulate host/device, verify compliance.

**TR110 / EX200 / EX260**
**$999 / $2999 / CALL**

### 2-ch 1GSa/s Scope




2-ch 1GSa/s (25GSa/s equiv.) 50/100 MHz scope. Competes head-to-head with high-end makes. 1-touch auto-setup - easy to use! 1 Mpts memory depth - great for circuit debug. **DS1000E** **$595**

### Scope + Analyzer




25MHz 2-ch /16 logic scope and logic analyzer. 2000 wfm/s refresh rate. See waveforms and logic data simultaneously. Trigger on data or signal edge.

**DS1022CD** FreeCase! **$699**

### 60/100/120MHz AWG




60/100/120MHz USB 14-bit ARB w. USB RS-232, LAN/GPIB. 10 built-in wfms; 6 digit 200MHz frequency counter; opt. 16-ch gen. **DG3061A** **$1895**

**DG3101A/DG3121A** **$2795/$3995**

### 1/2GHz RF Generators



High accuracy/stability, wide range, low phase noise/leakage, serial control. Keypad/rotary setting. TGR1040: 10MHz-1GHz, FM int/ext TGR2050: 150kHz-2GHz AM/FM/PM

**$2498/4558**

### EMC Spectrum Analyzer



Handheld Palm PC-based 2.7GHz Spectrum Analyzer. Multiple sweep settings – store wfms, set-ups, etc. Use WiFi PC for email, reports, calculations.

**PSA2701T** **$1990**

### Color LCD Scope




**Owon -** 25MHz 2-ch + trigger standalone USB bench scope with 7.8" color LCD. Battery powered optional.

**PDS5022S** (25MHz) **$325**
**PDS6062T** (60MHz) **$599**

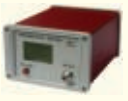
### Low-Cost Scope



2-ch 40/100/200MS/s 8-bit scope range with 5/10/25MHz bandwidth and USB2.0 for fast screen updates. Inc. scope/FFT/logging software.

**PS2203/4/5** **from $297**

### RF Generator



High-res, extermely low-noise, portable RF generator. Outstanding signal purity (phase noise -125dBc/Hz at 1GHz, harmonics <-40dBc). Rapid 2ms power/frequency switching/setting. 9k-3.3GHz.

**APSIN3000** **$4999**

### Logic Analyzer



ZeroPlus - Intuitive full-featured 16-ch 4MB 200MHz sampling memory. Sophistcated triggering. I²C, SPI, and UART monitoring. Includes clipleads, USB cable and software.

**LAP-16128U** **$299**

### 20MHz Scope




**PS5204 -** 2-ch 250MHz Scope with 128 MS/s and real-time sampling of 1GS/s. Adv triggering. 250MHz spec. analyzer and 125MS/s waveform gen.

**PS5203** 32MS buffer **$2237**
**PS5204** 128MS buffer **$3360**

### GPS Logger



**TrackStick II -** Pocket-sized logger tracks position data for vehicles, personnel, etc. Records date, time, location, route and speed. Shows data on Google maps! **$175**

**Super TrackStick II** **$239**

### 6 in 1 Scope



6-in-1 USB scope adapter! 200kHz 2-ch 10-bit scope, 2-ch spectrum analyzer, 2-ch chart recorder, 16-ch 8MHz logic analyzer, 5-ch signal gen, 8-ch pattern gen. SPI/I2C/1-w/UART decoding.

**PoScope with probes** **$199**

### Automotive Testing



**Auto Diagnostics -** Award-winning kits turn yours into vehicle-electrics diagnostic tool. **PS3423 kit $2293**

3-ch logger for 16-bit current/voltage/temp msmt. **KLARI-MOD MC $1900**

### Mixed -Signal PC Oscilloscope



100 MHz Scope and Logic Analyzer lets you do complex triggering to find hard-to-get-at glitches, spikes, etc. Huge 4 or 8 MS buffer for deep data drilling and zooming. Optional built-in swept signal generator. 2 Analog channels x 10, 12, 14 bit with more than 60 dB dynamic range. 8 digital inputs for mixed signal display/trigger.

**CS328A-4** (4MS Buffer) **$1259**
**CS328A-8** (8MS Buffer) **$1474**
**CS700A** (signal generator) **$299**

### Handheld Scope



20MHz or 60MHz rugged handheld USB scope with 3.8" color LCD. Built-in meter - great for your tool kit.

**HDS1022MN** (20MHz) **$593**
**HDS2062M** (60MHz) **$699**

### Pen Scope



**PS2105** - 25MHz USB powered scope-in-a-probe! Up to 100MS/s, 24kS buffer C/VB/Delphi/LabView/VEE drivers.

**PS2105** (25MHz, 100MS/s) **$369**
**PS2104** (10MHz, 50MS/s) **$234**

### RF Testing



Economical, portable RF test enclosures, for initial compliance testing. **from $1295**

Shielding Tents with external frame, instant up/down for emissions-free room. **from $3500**

### Multiparameter Loggers



Mini-logger with built-in temp/humidity/pressure/3-axis accel sensors. Customize msmt/record parameters; send data to PC after or during logging. Synchronize multiple loggers, merge data to single record

**MSR145S** **from $468**

### SPI Bus Analyzer



**SPI Bus Analyzer -** SPI protocol exerciser and analyzer for standard SPI and non-standard 4-wire and 3-wire serial protocol interfaces up to 50 Mbps. PC control via free GUI 8PI Control Panel Software.

**SPI Xpress** **$783**

### Internet Radio



**WIZ-iRadio** – play internet-streamed audio files. Ready-made answer for creating an Internet Radio - doesn't need a PC. Backlit 128 x 64 LCD. IR remote control. **$60**

### Wireless LAN



Tiny 802.11b/g wireless LAN module. Ethernet-Radio bridge with RS232 serial interface, webserver. LAN datastreaming to 33Mbps. add wireless without a redesign!

**WIZ-600Wi** **$62**

### PSoC Starter



Get going quickly with PSoC visual design environment, inc. "Beginner's Guide to PSoC Express", CD media (devtools/compiler/projects/video-tutorial), PSoC demo board.

**PSoC Starter** **$199**

Misc

### Serial-Ethernet Cable



Network your serial product easily without a PC using this 28" cable. Provides an instant IP address. RS-232 port compatible with most 10/100mpbs Ethernet hubs.

**eCOV-110-P** **$89**

### Instant Ethernet



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

**W5100** (integrated PHY) **$5.88**
**WIZ110SR** **$35**

### Ethernet-IO



**Ether-IO -** UDP/IP-controlled 24 digital I/O board 3 x 8-bit TTL ports each independently programmable. Connects to any TCP/IP Ethernet network.

**Ether-IO 24** **$119**

### .NET Board



Smallest (2.2" x2.2"!) lowest cost Microsoft's .NET Micro Framework development system. USB host & FAT file. Run C#; Visual Studio for programming. SD/USB memory, other I/F. **USBizi** **$99**

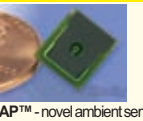
### Wireless Analog



4-ch analog input wireless module 433/868/915MHz. Low power serial interface. Easy configure. 24 bit ID to avoids cross- talk. 2 digital output.

**EmbedRF Module** **$59**
**EmbedRF DevKit** **$299**

### Temp/RH Sensors



**UPSICAP™** - novel ambient sensors and modules accurately measure temp/RH. Amazing range/response: 0% to 100% RH, -90 to +85degC; 0.25s! Interchangeable without recalibration. **from $3 / $40**

### USB to I2C



**Devasys -** Provides a simple "drop-in" solution for connecting your PC to 90kHz I²C + 20 I/O lines. Free software. Use multiple boards for more I²C/IO.

**USB I2C/IO** **$89**

### Easy OLED Display



**4D Systems -** Compact, economical smart OLED with graphics for 'stand-alone' functionality to your project, drive from USB or RS232.

**uOLED-96-G1** **$89**
**uOLED-128-G1** **$87**

### RF Modules



**RF TX/RX Modules -** simultaneously transmit composite video and stereo audio signals. 2.4 GHz ISM band. 4-ch RF, 5V operation

**AWM630TX** **$16**
**AWM634RX** **$27**

### TinyOne Zigbee™



Complete RF solution for fast & reliable data transmission. Miniaturized OEM RF module ensures fast and easy integration into wireless applications.

**TinyOne Zigbee** **$58**
**B915FH-tiny/40-SMD-WA** **$63**

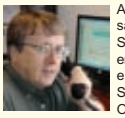
### 1/2/4/8/16 x RS232



Add COMports via your PC's USB Port. **USB-COM-S** **$20**

**USB-2COM** **$48**
**USB-4COM** **$105**
**USB-8COM** **$195**
**USB-16COM-RM** (rackmount) **$459**

### Saelig



Alan Lowne says: "I started Saelig as an engineer - for engineers! Saelig - an Olde English word meaning 'happy, prosperous, blessed' - which is what I want for my customers, my staff, and myself! See my blog for behind-the-scenes company details, business hints, kudos, etc. "

Saelig.com
unique electronics
888-7SAELIG info@saelig.com

*We search the world for unique, time-saving products - see our website for 100s more: volt/temp data loggers, wireless boards, LCD display kits, Ethernet/IO, USB/RS232/485, USB-OTG, instant Ethernet-serial, CAN/LINbus, USB cables/extenders, line testers, logic analyzers, color sensors, motion controllers, eng. software, wireless boards, SMD adapters, I2C adapters, FPGA kits, GPS loggers, automotive testing, security dongles, video motion detectors, crystals/oscillators, custom switches, barcode scanners, robots, DSP filters, PLCs, Remote MP3 players, etc. Don't forget to ask for your FREE Starbucks card with your order! Check* ***www.saelig.com*** *often for special offers, bargains, business hints, blog, etc.*

# IDEA BOX

## THE DIRECTORY OF PRODUCTS AND SERVICES

**AD FORMAT**: Advertisers must furnish digital submission sheet and digital files that meet the specifications on the digital submission sheet. **ALL TEXT AND OTHER ELEMENTS MUST FIT WITHIN A 2″ x 3″ FORMAT**. Call for current rate and deadline information. E-mail adcopy@circuitcellar.com with your file and digital submission or send it to IDEA BOX, Circuit Cellar, 4 Park Street, Vernon, CT 06066. For more information call Shannon Barraclough at (860) 875-2199.

The Vendor Directory at www.circuitcellar.com/vendor/
is your guide to a variety of engineering products and services.


**USB**

**Add USB to your next project—it's easier than you might think!**

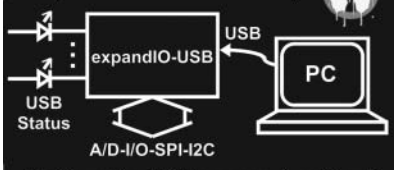
- USB-FIFO up to 8 mbps
- USB-UART up to 3 mbps
- USB/Microcontroller boards pre-programmed with firmware
- **2.4GHz ZigBee™ & 802.15.4**
- **RFID Reader/Writer**

Absolutely NO driver software development required!

**www.dlpdesign.com**

DLP Design

**Revolutionary new expandIO-USB *chip***




- Ideal for adding USB to sensors & peripherals
- No drivers needed for Windows, Mac, Linux
- No microcontroller programming required
- Also check out our USB-232 USB to UART

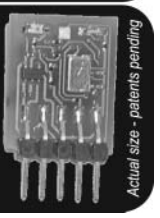
*www.hexwax.com - Buy from Mouser & Farnell*

**Amazing** PIC programmer

Most devices supported
ICSP, SQTP, & copy limits

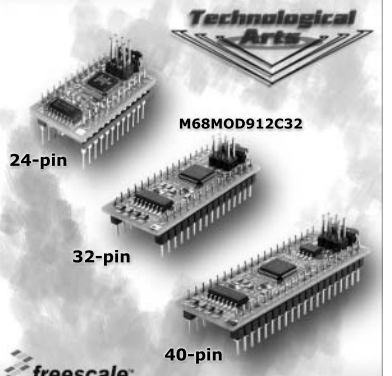
**$32** at Digikey & Mouser

*www.flexipanel.com*



*Actual size - patents pending*

**Freescale 9S12C microcontroller DIP modules**

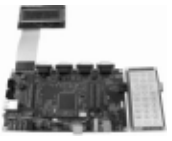



freescale Alliance Member

**www.nanocore12.com**

Toll-free: 1-877-963-8996

**NXP LPC2468**
**Industrial Reference Design Kit**



The LPC2468 IRD is targeted at RTOS based embedded systems. The kit features an NXP LPC2468 core board, a generic base board, a 27 key membrane keypad and a 4 x 20 LCD. Kit includes Micrium µC/OS-II Real Time Operating System (RTOS), IAR EWARM toolset and JLink debugger, 10/100Base Ethernet, USB Host/Dev, CAN, RS-232, & I2C. Flexible Core and Base PCBs allow you to change processor type/family and use different displays (VFD vs. LCD) and keypads. A JTAG interface and Micrium's system monitor feature support full Software development and debug. In-System-Programming (ISP) is available using FDI's USB-ICP-LPC2K USB based programmer.

P/N IRD-LPC2468-DEV, priced at $495.00, is available directly from Digi-Key www.digikey.com and Mouser www.mouser.com.

FDI also provides low cost demo kits for NXP ARM7 LPC2xxx, LPC9xxx and ICP/ISP FLASH programmers.

FDI
www.teamfdi.com

**Future Designs, Inc.**
**2702 Triana Blvd**
**Huntsville, AL 35805**
**(256) 883-1240**
**Fax (256) 883-1241**

**ALL ELECTRONICS CORPORATION**

Electronic and Electro-mechanical Devices, Parts and Supplies. Wall Transformers, Alarms, Fuses, Relays, Opto Electronics, Knobs, Video Accessories, Sirens, Solder Accessories, Motors, Heat Sinks, Terminal Strips, L.E.D.S., Displays, Fans, Solar Cells, Buzzers, Batteries, Magnets, Cameras, Panel Meters, Switches, Speakers, Peltier Devices, and much more....

**www.allelectronics.com**

***Free 96 page catalog***

***1-800-826-5432***

**SpectraPLUS 5.0**
***Audio Spectrum Analysis***

**Features**
Sound Card based I/O
FFT sizes to 1048576pts, 1/96 Octave
Up to 24 bit, 200kHz sampling rates
3-D Surface and Spectrogram
Digital Filtering, Signal Generation
THD, IMD, SNR, Transfer Functions
DDE, Macros, Data Logging,
Vibration Analysis, Acoustic Tools

**FREE 30 day trial!**
***www.spectraplus.com***

**PHS**
Pioneer Hill Software
360 697-3472 voice
pioneer@telebyte.com

I²C/SMBus
- Bus Monitors
- Protocol Analyzers
- Host Adapters
- Multiplexers
- Battery Applications
- Software Tools

MCC
Micro Computer Control

I²C is a trademark of Philips Corporation

www.mcc-us.com

tincan TOOLS

Linux in a 40-pin DIP
- 200 Mhz ARM9 CPU Board
- 0.1" centers - easy to prototype
- use your existing prototype tools
- full Linux with 2.6 kernel
- CD includes source and schematic

HAMMER

Perfect for robotics, web servers, remote monitoring & communications

www.TinCanTools.com

NEW from Microchip
Enhanced Mid-Range PIC® MCU C Compiler Support

Fundamental Compiler Enhancements
- Code Size Reduction
- New Powerful String Functions
- Full use of New Opcodes and Address modes

PCM Compiler $150
Customers on Active Maintenance Download Version 4.100

Recognized Microchip Third-Party Tool Provider

CCS

262-522-6500 x35
Sales@ccsinfo.com
www.ccsinfo.com/EN16

Display with Touch Screen

- Embedded, Standalone, C/C++ Programmable…
- Low Cost User Interface with Ethernet, USB, CF, RS232, ADC, DAC, I/Os, Relays…
- U-Drive™ 5.7" QVGA TFT, CAN, Ethernet, CF, ADC, DAC, ….

start at $379 OEM

60+ Low Cost Controllers with TFT, ADC, DAC, UARTs, 300 I/Os, solenoid, relays, Compact-Flash, LCD, Ethernet, USB, motion control. Custom board design. Save time and money.

TERN INC.
1950 5th Street, Davis, CA 95616 USA
Tel: 530-758-0180 • Fax: 530-758-0181
www.tern.com • sales@tern.com

PCB Fab & Assembly Since 1997

Experts in Quick Total Turnkey Assembly at Low Prices!
- PCB Design, Fab & Assembly
- Cable Assembly
- Box Integration
- Functional Testing
- BGA
- Product Development
- ISO Certified

CapTron Corporation
301-869-6100
sales@captroncorp.com

IMAGEcraft C

C development tools for:
- AVR & (X)MegaAVR
- HCS12, HC08
- TI MSP430
- Propeller
- ARM

hardware kits:

NEW: the eBox- AVR!
Perfect for students and hobbyists

Elmicro kits - HC08 & HCS12 / 12 ref. design & starter kits

Ethernut - Complete AVR TCP/IP solution

starting at $249! ($99 for students)

Version 7 Tools
Full-featured 45-day demos on our website!

VISA / MC / AMEX
Discover / Check

Since 1994
www.imagecraft.com
info@imagecraft.com
(650) 493-9326 • FAX (650) 493-9329

Can a computer travel back in time?
Yes it can... Travel back with Retro

Return to the fundamentals of computers and computer programming with the Retro computer system. The 8-bit microcomputer with a built-in BASIC language interpreter and minimal operating system has been reborn. Take a journey back with Retro and bring it into your future.

SOME OF RETRO'S MANY FEATURES
- Built-in BASIC language interpreter
- Over 55 statements, functions, and operators
- Floating point math • Numeric&string variables/arrays
- 40 x 30 Text • 320 x 240 Pixels • 64 Colors
- Address/data bus expansion port for external devices
- Serial, VGA, keyboard, and joystick ports
- Single channel sound with a line level output jack
- 32K of RAM • Built-in EEPROM for program storage
- Socket provided for an addtional storage EEPROM

MULTILABS
Web - www.multilabs.net
Email - support@multilabs.net
Phone - (775) 852-7430

Instant LCD

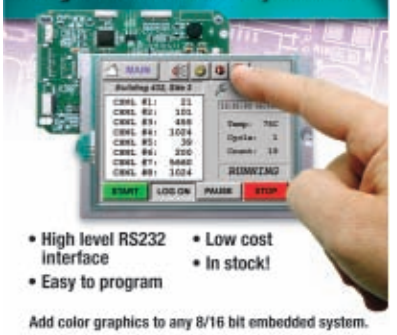


ezLCD-002 - $189

- Versatile Programmable Module
- Serial/I²C/USB/Parallel Interface
- AVR/BASIC Stamp/VB Compatible
- Onboard Flash Bitmap Memory
- Downloadable TTF Fonts
- 2.7" or 5.6" TOUCH Color TFT LCD
- 240x160 or 320x240 resolutions
- Transflective w/ LED Frontlight (2.7")
- Transmissive w/ 350 nit backlight (5.6")
- 512 colors or 65,535 colors

EARTHLCD.com We Make LCDs Work.™

Add a color touch interface to your embedded product!



- High level RS232 interface
- Low cost
- In stock!
- Easy to program

Add color graphics to any 8/16 bit embedded system. Easy, fast and flexible. Up and running in hours!

REACH TECHNOLOGY INC.

www.reachtech.com • 510-770-1417

842 Boggs Avenue • Fremont, CA 94539

Solve complex signal acquisition problems...

- positioning & control
- environmental
- acceleration
- transients
- pressure
- vibration
- sonar
- GPS
- Linux Driver
- Guaranteed in stock
- Many newly added features
- 16-bit analog inputs and outputs
- Million sample FIFO eliminates interrupts
- Wide analog input and output ranges
- -40°C to +85°C Standard
- Order 24/7, fast and easy.



www.stx104.com

Apex Embedded Systems

help@stx104.com • 608-256-0767 x24

Going Wireless is Easy!

ZigBee™ PRO Module



NEW

ETRX2

Worlds first ZigBee™ PRO compliant module is now finished and ready for use in full healing mesh network

LEMOS INTERNATIONAL 1-866-345-3667 sales@lemosint.com

www.lemosint.com

Free Training

for Rabbit Developers

Learn More at Rabbit-U.com

08078

www.can232.com

Only $108 €89



CAN232 Features:
Free sample programs
8-15VDC supply via CAN
Timestamp in mS
Small size 2.7" by 1.2"
100% Bandwidth up to 125Kbit
Both 11 & 29 bit ID support
32 Message Receive FIFO
Works up to 1Mbit CAN
Simple ASCII protocol
Supports RTR Frames
Max 230Kbaud RS232
Firmware upgradable
No drivers needed
OS independant
CE Approved

CANUSB Features:
Free ActiveX component
PC, MAC & Linux support
Both 11 & 29 bit ID support
Simple CAN logger included
Free Threaded Windows DLL
Firmware upgradable via USB
Sample programs in C, C++, VB, Delphi, C#, PureBasic etc.
No need for external power
Works up to 1Mbit CAN
Supports RTR Frames
USB 2.0 Full Speed
Free USB drivers
CE Approved



Only $154 €129

The original (Controller Area Network) CAN to RS232 serial converter.

www.canusb.com

EASY TO USE READY TO GO

NEW! ANTERO ARM 7 SBC

LEARN MORE AT VESTAtech.com

Solutions for Embedded Systems Development

PHYTEC

phyCORE® SBCs Accelerate 32-bit Designs



phyCORE-LPC3250

CDTS Single Board Computers:
- shorten time-to-market
- reduce development costs
- forgo substantial design issues and risks
- Windows® Embedded CE and Linux Board Support Packages (processor-dependent)
- $170/unit benchmark price at 1K for ARM9-based SBC

ARM: LM31 (ARM11), LM27 (ARM9), LPC3250 (ARM9), LPC3180 (ARM9); LPC2294 (ARM7)

XScale: PXA320, PXA270, PXA255

PowerPC: MPC5554, MPC5200B, MPC565, MPC555

ColdFire: MCF5485 Blackfin: ADSP-BF537

- Rapid Development Kits start at $399
- include SBC, Carrier Board, software, docu as well as kit-specific cables, adapters and LCD
- SBC module easily ports from Carrier Board to user target hardware
- Carrier Board serves as target reference design



www.phytec.com ■ 800.278.9913 ■ www.phycore.com

NEW

HI-TECH C® PRO for the Silicon Labs

8051 MCU Family

C Compiler

FREEWARE

New PRO compiler also operates in Lite mode with no time or size limits:

silabs.htsoft.com/portal/CC-20

HI-TECH SOFTWARE

CIRCUIT CELLAR®

ANNOUNCES A

Mega-

Sale!

Just in time

for the holidays:

Our entire stock of

BACK-ISSUES

is on sale!

$5 EACH

& FREE SHIPPING*

To see the entire list of what's available, visit

http://www.circuitcellar.com/archives/backissuesprint.html

*U.S. orders only, but similar savings apply to foreign orders. Only applies to instock issues. See website for details.



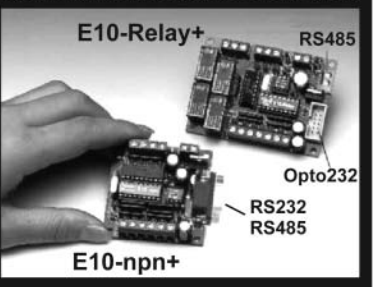
Weather Instruments for PCs

AAG electrónica

www.aagelectronica.com

The $69 PLC

Work as Stand-Alone Ladder Logic PLC.
Or as Smart Remote I/Os of PC/ PLCs.
RS485 allows 256 units to be networked.




Incredibly Easy to Program!
Our software is used by many colleges for teaching PLCs!
Get Free Ladder Logic Simulator:
www.tri-plc.com/cci.htm
TRI Triangle Research
Tel: 1-877-874-7527 - PLC specialist since 1993

Embedded Ethernet

ONLY $98




- 10Base-T Ethernet
- 186 Processor @ 40 MHz
- DOS w/ Flash File System
- 16 Digital I/O
- 5V DC
- Console / Debug Port
- Hardware Clock/Calendar
- Socket for DiskOnChip
- 512K DRAM & 512K Flash
- (2) Serial Ports
- (2) 16-bit Timers
- Watchdog Timer
- 3.75" x 2.50"

$129 Development System

Development Kit Includes:
- picoFlash Controller
- Borland C/C++ 4.52
- TCP/IP, PPP & Web Server
- Serial Driver Library
- AC Adapter and Cables

Call 530-297-6073 Email sales@jkmicro.com
On the web at www.jkmicro.com
JK microsystems

CROSSWARE®
Tools for Embedded Development

ARM®
8051
COLDFIRE®

Three quality tool suites
- ❑ C/C++*
- ❑ Code Wizards
- ❑ Debugging
- ❑ Simulation
- ❑ Support

* Embedded C++ for ARM and ColdFire only

Advanced software tools since 1984

www.crossware.com
360-812-2397

When contacting our advertisers, tell them where you found their ad...




...tell them, "Circuit Cellar sent me."

USB Data Acquisition

ADU208 -USB Relay I/O Interface




FEATURES
-8, 5-AMP relay outputs
-8, ISOLATED digital inputs
-Port Powered, Aux 5VDC output $189.00 QTY 1

Other Models:
ADU200- 4 Channel Version with RS232 $139.00
ADR218- Solid-State Version 8 Channel $225.00
ADU100- 3 CH, 16-Bit ISOLATED Analog Inputs, PGA, 4 digital I/O, RS232 and 5 AMP Relay Output $199.00

ONTRAK CONTROL SYSTEMS INC.
PH: (705) 671-2652 Fax: (705) 671-6127

www.ontrak.net

# PRINTED CIRCUIT BOARDS

**QUALITY PRODUCT**
**FAST DELIVERY**
**COMPETITIVE PRICING**

PROTOTYPE through PRODUCTION

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

**SPECIAL OFFER:**
**10 pcs** (3days) 1 or 2 layers **$249**
**10 pcs** (5days) 4 layers **$695**
(up to 30sq. in.ea.) includes tooling, artwork, L.P.I. mask & legend

## PULSAR, INC

9901 W. Pacific Ave. Franklin Park, IL 60131 • Phone 847.233.0012
Fax 847.233.0013 • www.pulsar-inc.com • sales@pulsar-inc.com






IRCUIT
LLAR
FOR COMPUTER APPLICATIONS

Make sure you're signed up to receive Circuit Cellar's monthly electronic newsletter. *News Notes* will keep you up to date on Circuit Cellar happenings. Stay in the loop!

*Register now. It's fast. It's free.*
**www.circuitcellar.com/newsletter/**



**Full Speed**
# It writes your USB Code!

**NEW! HIDmaker FS for Full Speed FLASH PIC18F4550**

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.

HIDmaker FS Combo: Only $599.95

**DOWNLOAD the HIDmaker FS Test Drive today!**
www.TraceSystemsInc.com
301-262-0300

Trace SYSTEMS, Inc.



# PROFESSORS

The Circuit Cellar college program puts quality engineering information in the hands of your students every month. Sign up now to get Circuit Cellar distributed to your class this semester.




To update your professor account or to find out more about our college program, visit www.circuitcellar.com/productsicollegeprogram



Order online at: www.melabs.com

## microEngineering Labs, Inc.

Development Tools for PIC® Microcontrollers

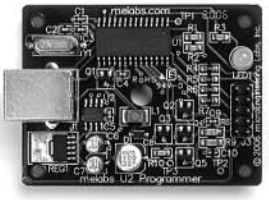
Phone: (719) 520-5323
Fax: (719) 520-1867
Box 60039
Colorado Springs, CO 80960

**USB Programmer for PIC® MCUs** $89.95 (as shown)

*RoHS Compliant*

*Programs PIC MCUs including low-voltage (3.3V) devices*

*Includes Software for Windows 98, Me, 2K, & XP*




*With Accessories for $119.95:*
*Includes Programmer, Software, USB Cable, and Programming Adapter for 8 to 40-pin DIP.*




**Serial LCDs**
2-line x 16 $39.95
4-line x 20 $49.95
*Quantity Discounts Available!*

**LAB-X Experimenter Boards**



*Pre-Assembled Boards Available for 8, 14, 18, 28, and 40-pin PIC® MCUs*
*2-line, 20-char LCD Module*
*9-pin Serial Port*
*Sample Programs*
*Full Schematic Diagram*

**Pricing from $79.95 to $349.95**

**PICPROTO™ Prototyping Boards**



*Double-Sided with Plate-Thru Holes*
*Circuitry for Power Supply and Clock*
*Large Prototype Area*
*Boards Available for Most PIC® MCUs*
*Documentation and Schematic*

**Pricing from $8.95 to $19.95**

**BASIC Compilers for PICmicro®**



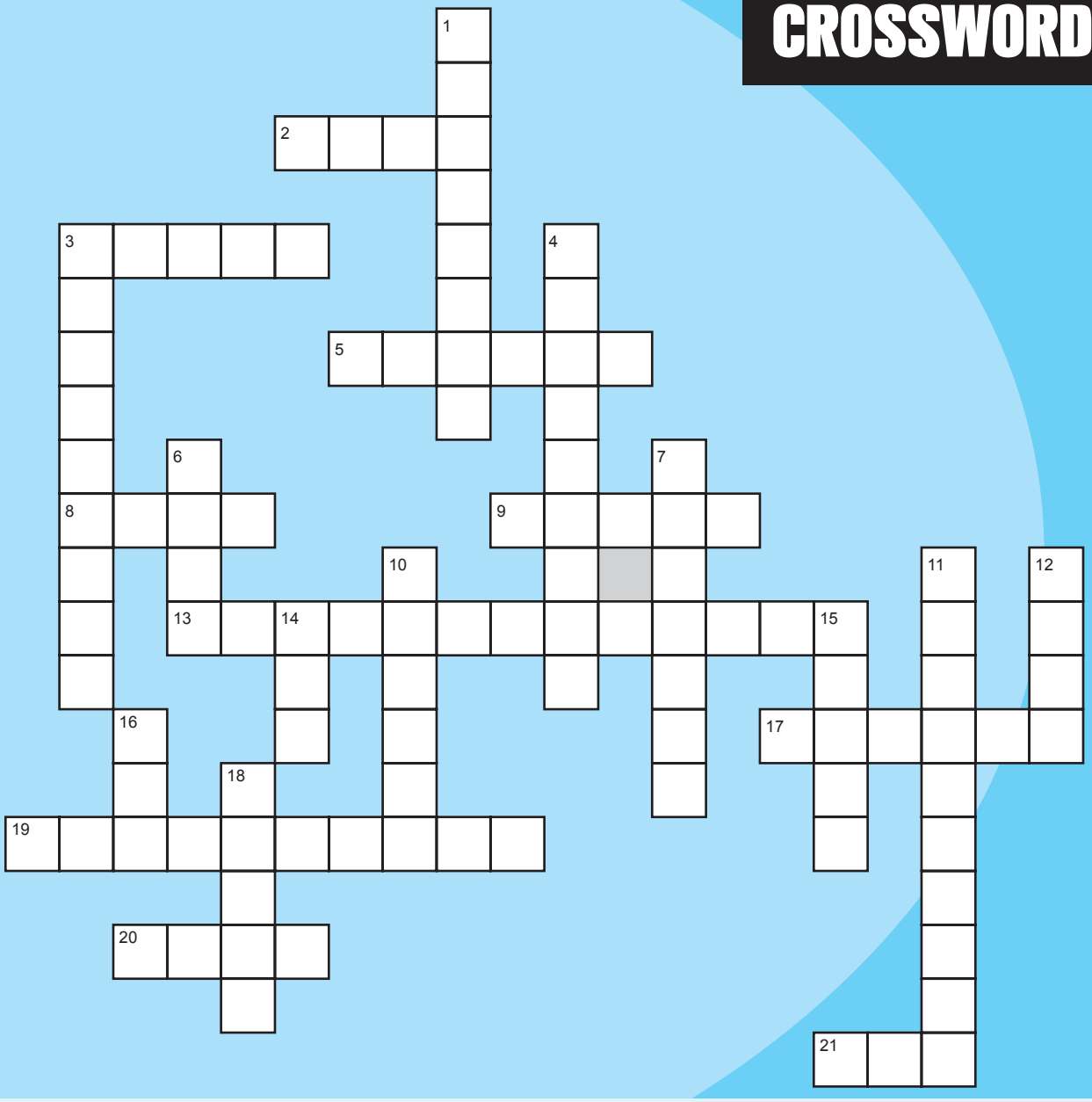
*Easy-To-Use BASIC Commands*
*Windows 9x/Me/2K/XP Interface*

**PICBASIC™ Compiler $99.95**
*BASIC Stamp 1 Compatible*
*Supports most 14-bit Core PICs*
*Built-In Serial Comm Commands*

**PICBASIC PRO™ Compiler $249.95**
*32-bit signed variables and math operations*
*Supports Microchip PIC10, PIC12, PIC14, PIC16, PIC17, and PIC18 microcontrollers*
*Direct Access to Internal Registers*
*Supports In-Line Assembly Language*
*Interrupts in PICBASIC and Assembly*
*Built-In USB, I2C, RS-232 and More*
*Source Level Debugging*

*See our full range of products, including books, accessories, and components at:*
**www.melabs.com**

# CROSSWORD



## Across

2. Place on the back of a computer for a PCB
3. Algorithmic language
5. 42 gallons
8. Bell Labs Intelligent Terminal (or "Bacon, Lettuce, and Interactive Tomato")
9. Invented the relay in 1835
13. AVR, ARM, PIC
17. Your screen name
19. ZH
20. A GUI theme
21. Singleton

## Down

1. Cat
3. A blog maintained by an anonymous author
4. Its reciprocal is the period T
6. 0.16 inches
7. Bridge + router
10. One-millionth of a meter
11. The layer of software that links different software systems together
12. Nonuple
14. Fossil fuel gas substitute. Combustion ≠ greenhouse gas.
15. #
16. Katal
18. Circular option button

**The answers are available at www.circuitcellar.com/crossword.**

# INDEX OF ADVERTISERS

**The Index of Advertisers with links to their web sites is located at www.circuitcellar.com under the current issue.**


| Page | | Page | | Page | | Page | |
|---|---|---|---|---|---|---|---|
| 91 | AAG Electronica, LLC | 74 | Embedded World Expo | 51 | Lakeview Research | 17, 50 | Rabbit, A Digi International Brand |
| 73 | AP Circuits | 18 | ExpressPCB | 90 | Lawicel AB | 90 | Rabbit, A Digi International Brand |
| 80 | APEC | 66 | ezPCB | 13, 90 | Lemos International | 90 | Reach Technology, Inc. |
| 88 | All Electronics Corp. | 88 | FDI-Future Designs, Inc. | 95 | Luminary Micro | 47, 71 | Renesas Technology |
| 90 | Apex Embedded Systems | 88 | FlexiPanel Ltd. | 89 | MCC (Micro Computer Control) | 87 | Saelig Co. |
| 7 | Atmel | 50 | Futurlec | 27 | Microchip | 9 | SEGGER Microcontroller Systems LLC |
| 5 | Bitscope Designs | 64 | Geist Technology | 92 | microEngineering Labs, Inc. | 49 | Solder By Numbers (Schmartboard) |
| 65 | CWAV | 68 | Grid Connect, Inc. | 23 | Mouser Electronics | C3 | Tech Tools |
| 31 | CadSoft Computer, Inc. | 90 | HI-TECH Software LLC | 89 | Multilabs | 2, 3 | Technologic Systems |
| 89 | CapTron Corp. | 57, 72 | ICbank Inc. | C2 | NetBurner | 88 | Technological Arts |
| 10 | Comfile Technology, Inc. | 71 | HobbyLab LLC | 51 | Nurve Networks LLC | 89 | Tern, Inc. |
| 91 | Crossware Products, Inc. | 34 | Holtek Semiconductor, Inc. | 91 | Ontrak Control Systems | 89 | Tin Can Tools, LLC |
| 29, 89 | Custom Computer Services, Inc. | 89 | IMAGEcraft | 16 | PCB-Pool | 42 | Total Phase, Inc. |
| 88 | DLP Design | 1 | Imagineering, Inc. | C4 | Parallax, Inc. | 92 | Trace Systems, Inc. |
| 82 | DesignCon | 64, 66 | JKmicrosystems, Inc. | 90 | Phytec America LLC | 91 | Triangle Research Int'l, Inc. |
| 71 | DesignNotes | 91 | JKmicrosystems, Inc. | 88 | Pioneer Hill Software | 90 | Vesta Technology |
| 79 | Digilent, Inc. | 41 | Jameco | 81 | Pololu Corp. | 73 | WCSC (Willies Computer Software Co.) |
| 68 | EMAC, Inc. | 51 | Jeffrey Kerr, LLC | 13 | PROPOX Sp. z o.o. | 11 | WIZnet |
| 90 | Earth Computer Technologies | 22 | Keil Software | 92 | Pulsar, Inc. | | |


## Preview of January Issue 222
## Theme: Embedded Applications

**Digital Radio Reception:** Build a Longwave AM Radio Receiver

**Remote Audio Control:** Build a Virtual Instrument Interface

**Change a Processor:** Put the Flexibility of Soft-Core Processors to Work

**Microcontroller Clock-Locking:** A Frequency Reference Synchronized to a GPS Clock

**Design with FPGAs:** A Step-By-Step Guide for Embedded Development

**Power Switching in the Real World (Part 1):** Basic Applications and Safety Precautions

---

**LESSONS FROM THE TRENCHES More on Pointers?:** Pointers in C Language

**FROM THE BENCH Access SD Memory Cards (Part 1):** Solid-State Storage Media

**SILICON UPDATE Hot Chips 20:** A Review of New Technologies


## ATTENTION ADVERTISERS

**February Issue 223 Deadlines**

Space Close: Dec. 12
Material Close: Dec. 19

**Theme:**

**Wireless Communications**

Call Shannon Barraclough
now to reserve your space!
**860.875.2199**
e-mail: shannon@circuitcellar.com

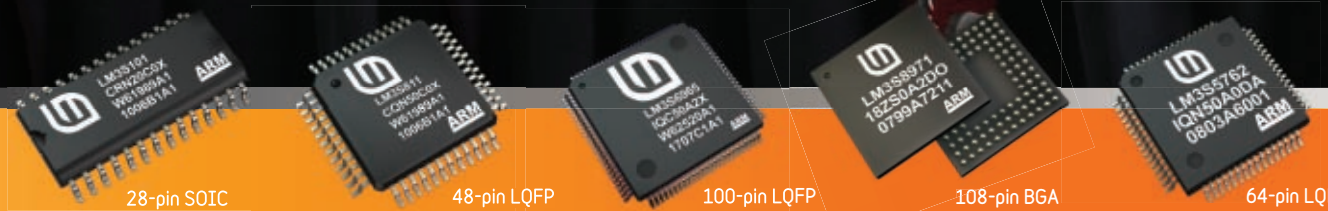
LUMINARY MICRO®

# Redefining Microcontrollers: Connected. Versatile. Cost-effective.



We understand that it is all about the software, so we've changed the rules.

See how our connected, versatile and cost-effective Stellaris devices will help you change the rules in your market.



28-pin SOIC | 48-pin LQFP | 100-pin LQFP | 108-pin BGA | 64-pin LQFP

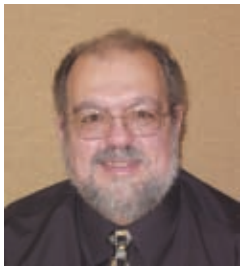
Learn more about how we have changed the rules today:

www.luminarymicro.com/newrules

STELLARIS® microcontrollers

Cortex™ ARM

LUMINARY MICRO®



# PRIORITY INTERRUPT

by Steve Ciarcia, Founder and Editorial Director

## My Life and Paper—Take 2

I remember talking about the concept of the paperless office a few years ago ("Filers Versus Pilers," *Circuit Cellar* 185, 2005). You know, now that we all have computers, paper in our lives would become unnecessary and obsolete. Everything will be digital and we'll only require a few key-placed, multifunction, high-resolution displays to satisfy all our text and graphics interactions without crushing trees and filling landfills. However, if I remember correctly, I suggested at the time that, contrary to expectations, evidence suggested that the advent of cheap printers had actually increased the use of paper. Surely we've progressed since then.

It isn't like this has been a nagging topic on the back burner for years that I suddenly had to write about. It's just that I had a few curious observations this week that prompted me to think about it again.

Last Thursday was the trigger. I was about to sit at the breakfast table when I realized I was moving about 30 to 40 lb. of newspapers and magazines out of the way so I could fit the dishes. I'm typically an obsessive neatnik, but I have this thing about periodicals. Magazines and newspapers get piled on the table all week before they go to the recycling bin. The trouble is that it seems I need two bins these days. Wanting only the Sunday newspaper resulted in getting it all week for the price of just Sunday's paper. Add to that two more daily newspapers, a few each of news, science, automobile, and cooking magazines, along with a plethora of trade journals that refuse to take me off their list (for the life of me I can't even guess how I ended up receiving metal plating, erosion control, and food engineering journals, among others), and the pile is over 2′ high!

After breakfast I stopped at the doctor's office for a flu shot. Before seeing the doctor, I was given a clipboard with about 20 different printed forms that I had to fill out or sign for their records. There was the form that confirmed that I knew what a flu shot entailed, the one about egg allergies, the one about conflicts with current medications, the one about who was going to pay for it (that I wasn't indigent), the one to list all of the medical events in my life, the one authorizing them to contact my insurance company, and on, and on. There was even a form to sign that authorized distribution of the forms.

"Aren't you guys computerized yet? Where's the portable tablet with all this stuff on one electronic form?"

The receptionist's answer was both interesting and horrifying. "Certainly we're computerized. We spend all day typing your answers into our database so we have a printed record to go in your record folder here at the office. Then, we put all the hard copies in our filing cabinets in case they are ever needed. And finally, at some point, we either scan all the documents or physically ship them to a large medical records company for archive storage."

Say what? Hand reentry, physical storage, archiving tons of paper? What gives?

Welcome to computers, medicine, and the law. Doctors may have efficient in-house databases, but it seems that laws in many states don't like electronic copies when it comes to medical liabilities. In many places, and I guess here in Connecticut, for some specified period of time there still has to be a complete physical paper trail when it comes to medical stuff. I grinned and went in to get my flu shot (and signed another piece of paper that I actually received it).

Unfortunately, my next stop was the Department of Motor Vehicles—a true bastion of bureaucratic competence and computerized efficiency! Seriously, the Connecticut DMV has been unusually efficient these days when it comes to the few things I do, but transactions are always a snake pit of uncertainty. We all have horror stories about missing records, transposed numbers on keyed data entries, and obtuse rules requiring the presentation of all the same physical documents again and again that are clearly electronically archived (usually on the terminal in front of the examiner you're talking to) from the last time you did a related transaction. I confirmed that the world is still firmly paper-based as I left the DMV with a dozen more sheets of paper that I must securely store forever or my cars and I may get erased.

So, is there less paper in my life than the last time I talked about this? Yes and no. I looked around the Circuit Cellar and there are indeed fewer piles of paper that I generate. Certainly the greatest factor for me was simply adding a second LCD on my computer. It turned out that a considerable amount of what I used to print was simply so I could refer to it while viewing something else on my computer screen (like a schematic and a spec sheet). Switching back and forth between windows just doesn't hack it. Now I just put the spec sheet on one display and the schematic on the other—no paper.

Of course, even though I've reduced that volume of paper in my life, keeping everything electronic has introduced an entirely new problem. It used to be that I could just thumb through the stack of papers or manuals and quickly find things. Now I need a Google map or personal butler to find whichever hard drive, CD ROM, DVD, flash drive, or digital storage thingy contains the actual file I want. Worse yet, once I find the media, can I still read it? Electronics is nice, but something tells me that paper is still king!

steve.ciarcia@circuitcellar.com



# PC Based Logic Analyzers

## NEW Model DV3400

- Faster Sampling (200/400 Msps)
- More Channels (36)
- More Memory (4x)
- Advanced Match circuits (8)
- ±6V Adjustable Threshold (x2)
- Cascadable Sequencers (4)
- Enhanced Compression

from $499.00

DigiView




## Professional Hardware Capture + Software Analysis

- Automatic Real-time Hardware Compression eliminates the need to reduce resolution. Our newest version makes "dead time" insignificant.
- Edge and Pattern Triggers on all models. Advanced Model also includes Range (>, <, =, !=, >=, <=) and Stable Matches with Duration, and 4 flexible cascadable sequencers with pass-counters.
- Pattern Searches with Match & Duration.
- Specialized Sequential Searches for Serial and State Mode Signals.
- Display I²C, Synchronous (SPI), Asynchronous (RS-232), State, Boolean, Bus and Analog Data.
- Single or Dual Waveform Views.
- Resolution Zoom in Wave Views.
- Time based Link Groups for all views.
- Specialized Exports from Data Tables and List Views.
- Multi-Signal Data Tables.
- Drag & Snap Markers.
- "Click to Center" function.
- "Snap Previous/Next" function.
- Print or Save Images with comments.
- USB 2.0 (480 Mbps).
- USB 1.1 compatible (12 Mbps).

Very flexible, "easy to configure" Advanced triggering.

## PICmicro® MCU Programmer

Multi-Function, In-Circuit & Gang Operation

only $199.00




QuickWriter™

See our Memory Emulators from $179.00

TechTools

www.tech-tools.com

(972) 272-9392 • sales@tech-tools.com

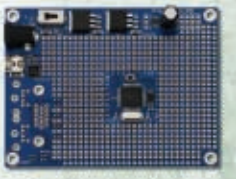
Copyright © 2007 TechTools • DigiView, FlexROM, EconoROM and QuickWriter are trademarks of TechTools • PICmicro is a registered trademark of Microchip Technology Inc.

# WISH LIST

Eager to get programming or to supplement your electronics projects? Parallax has what you need!




**StampWorks Experiment Kit** (#27297; $289.99) 35 great BASIC Stamp projects on the Professional Development Board.



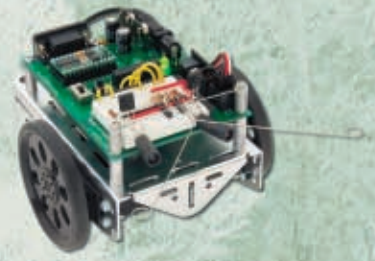
**Propeller Proto Board USB** (#32812; $39.99) Perfect for dedicated projects! Requires only a USB A to Mini-B cable for the PC programming connection.

**Sensor Sampler Kit** (#28028; $179.99) A collection of our nine best selling sensors.






**Propeller Professional Development Board** (#32111; $169.99) This feature-rich board sports built-in peripheral connectors to easily tap into the Propeller chip's multi-processing power and flexibility.



**Boe-Bot Robot Kit** (Serial/USB #28132; $159.99) Our best-selling robot is an excellent introduction to electronics, programming, and sensor-based navigation. Versatile and expandable with our selection of Robotics Accessory Kits.

**HYDRA Game Development Kit** (#32360; $199.99) Develop games, graphics, and media applications with this Propeller-powered console.




**BASIC Stamp Discovery Kit** (Serial/USB #27207; $159.99) Our most complete starter kit for microcontroller programming and experimentation.



**BASIC Stamp 2 module** (#BS2-IC; $49.00) A single board computer that runs Parallax PBASIC language interpreter in its microcontroller. Ideal for beginners yet quite powerful, with a vast resource base and sample code.




Order 24-hours a day at www.parallax.com or call our Sales Department toll-free at 888-512-1024 (Monday - Friday, 7:00 a.m. - 5:00 p.m., PT).

BASIC Stamp and Boe-Bot are registered trademarks of Parallax Inc. Propeller, Parallax, and the Parallax logo are trademarks of Parallax Inc.

PARALLAX INC

www.parallax.com