An Intro to USB MCUs (p. 24) • Menu Code for a Remote Control (p. 42) • A Serial Terminal Quick Fix (p. 58)

www.circuitcellar.com

CIRCUIT CELLAR

THE MAGAZINE FOR COMPUTER APPLICATIONS

#219 October 2008

SIGNAL PROCESSING

Build a Variable-Frequency Signal Generator

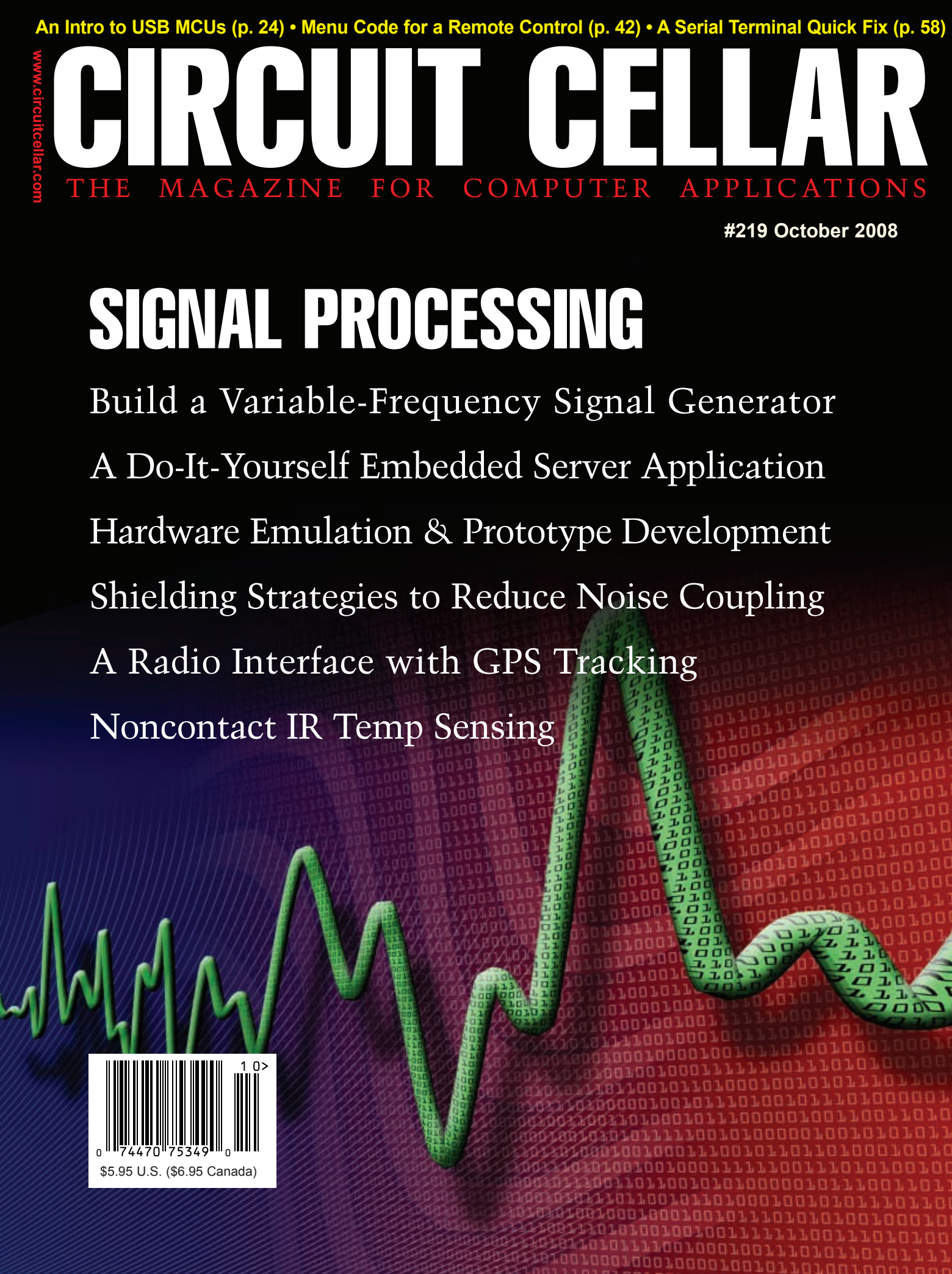
A Do-It-Yourself Embedded Server Application

Hardware Emulation & Prototype Development

Shielding Strategies to Reduce Noise Coupling

A Radio Interface with GPS Tracking

Noncontact IR Temp Sensing



0 74470 75349 0

$5.95 U.S. ($6.95 Canada)

NetBurner Customer Feedback: Why are you using NetBurner?

The excellent out-of-the-box experience

I would like to comment on the quality of a product that I purchased - the MOD5270. I've purchased products in the past from other companies that claim "plug and play" functionality. Usually this means plug it in and try to figure out why it doesn't work. With the MOD5270 I literally plugged in the power, connected it to my router, and ran the factory demo.

I am also impressed with the number of example applications that are included and the fact that all the development tools are included. It was nice to be able to create a new project, import one of the examples, and run it within 5 minutes.

Keith Gilman, Sr. Design Engineer

The complete hardware and software solution

We did a previous project using different companies for the RTOS, network stack, and compiler. Yeah, three vendors and all the associated finger pointing. We had to buy the flash file system as an extra/add-on. We put an 8M flash device on our processor bus, and a guy had to write a driver for the device (data register, command register, poll the ready bit, blah, blah), it was a 3+ month job. I thought I was in for the same thing. Thanks to you, we will drop the soldered-on Flash and put on the SD connector which will be ready to roll with your file system that runs 'right out of the box'. I was sweating about all the Flash work, and now its gone.

John Ramsy, Project Engineer

The active forum filled with NB Gurus

NetBurner offers a highly versatile and affordable solution set for embedded design, remote control, and product development. From the very active user forum (with Netburner gurus in attendance), to the great IDE and tool chain. NetBurner is the first and only solution for TCP/IP-Internet-Web product automation in my book.

Chris Ruff, Software Engineer

The quality of the technical support

I've been really impressed with the quality of the tech support. The guys answering the questions are right on and super helpful (probably has to do with your policy of having the engineers handle the tickets).

Nicolae P. Costescu, Project Engineer

The full-featured tool set, examples, and documentation

The device and the NetBurner tool-set far exceeded our expectations. I spent a week familiarizing myself with the NetBurner tools, and then I wrote a first-cut of the original application that I had envisioned for the SB72EX. As it turned out, two of the example applications formed the core of the networking capabilities of my application. The bulk of my time was spent implementing a simple command interface to cause the device to interrupt its default IP-to-serial modes to receive and store a stream of data and then FTP that data to a remote server.

The NNDK interface to the IP/socket facilities of the SB72EX proved to be the simplest part of the application - the interface is straightforward and the documentation is excellent (the large collection of application examples didn't hurt). Midway through my efforts I had a couple of questions that I elected to send to the NetBurner support group - the response was almost immediate, the person(s) responding were obviously knowledgeable and experienced - how refreshing!

The SB72EX device and (perhaps more importantly) the NetBurner tool-set have proven to exceed our immediate needs and have started the gears turning as to how we might use them to address other requirements.

Fred Craft, Software Engineer

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

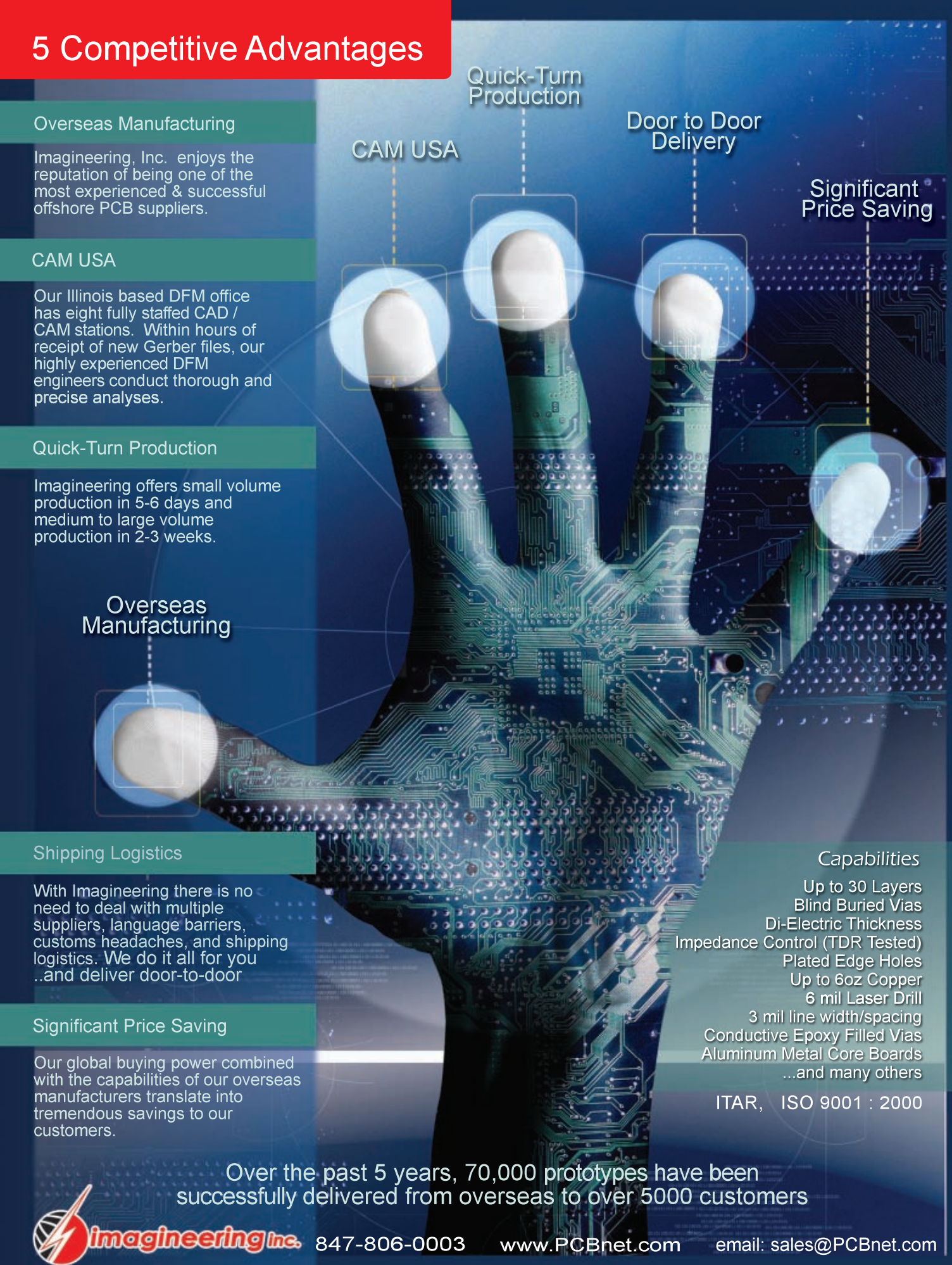
Imagineering, Inc. enjoys the reputation of being one of the most experienced & successful offshore PCB suppliers.

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

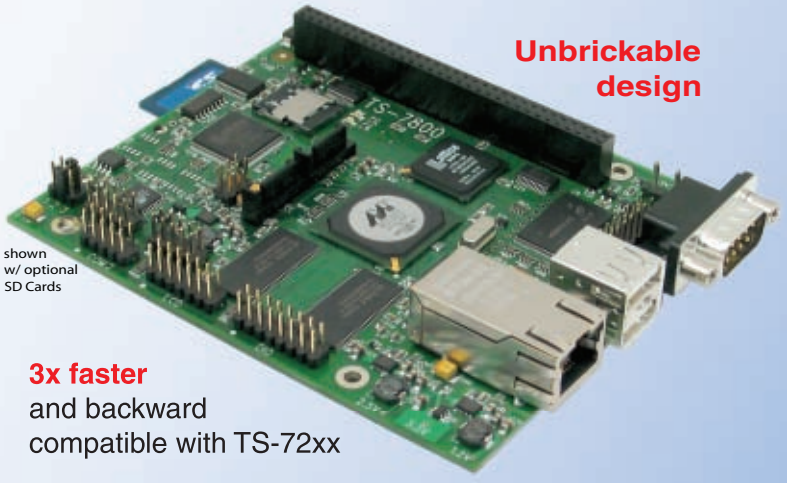
Over the past 5 years, 70,000 prototypes have been successfully delivered from overseas to over 5000 customers

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

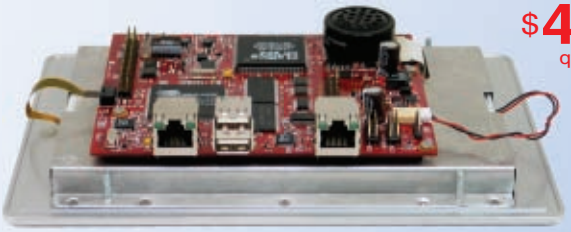
- Over 20 years in business
- Open Source Vision
- Never discontinued a product
- Engineers on Tech Support
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products stocked and available for next day shipping

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

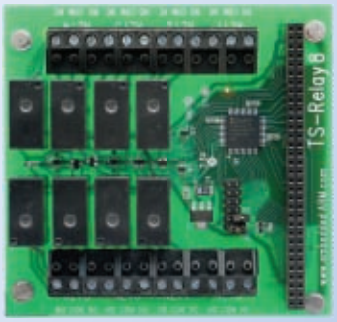
- Low Power, Industrial Quality Design
- Mountable aluminum frame
- 64MB SDRAM (128MB opt)
- 512MB Flash w/ Debian Linux
- Programmable FPGA- 5K LUT
- 7" Color TFT-LCD Touch-Screen
- 800x480 customizable video core
- Dedicated framebuffer- 8MB RAM
- Audio codec with speaker
- Boots Linux 2.6 in about 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

What Makes an Engineer Tick?

This month, you will read two articles by designers who have published with us in the past. During the last four years, both Neal Martini ("DIY Signal Generation," p. 12) and Peter Montgomery ("Dynamic Animatronic Remote (Part 2)," p. 42) have presented articles about interesting projects. This isn't anything new. As I recently told a first-time author who was preparing an article for an upcoming issue, we have dozens of repeat writers—designers who found presenting their ideas to the *Circuit Cellar* community to be a rewarding, career-enhancing experience. Many of you long-time readers could probably list the names of several talented designers who have published more than one helpful article. But have you ever wondered what makes those designers tick? Sure, you've read the short bios at the ends of their articles, but have you ever wanted to know a bit more about how they became engineers, how they developed and coded their designs, or what designs they're planning for the future?

If so, consider subscribing to *Circuit Cellar News Notes*, which is our free monthly interactive electronic newsletter. In addition to in-depth interviews with working engineers, the newsletter features a variety of useful content ranging from reader polls and product sample announcements to information about embedded community events. Go to www.circuitcellar.com/newsletter to sign up and read the latest installment.

After reading this issue, you may want to learn more about one of the engineers. If so, feel free to e-mail me your questions. If there's demand for a particular interview, we'll make it happen. It doesn't get any more interactive than that!

This month, Neal Martini describes how to create a signal source for RF applications. The design generates frequencies from 0 to 3 GHz with a frequency accuracy of less than 200 Hz.

If you're interested in USB connectivity, turn to Jeff Bachiochi's special feature about Microchip Technology's line of USB PIC microcontrollers (p. 24). With Microchip's 8-, 16-, and 32-bit USB MCUs, embedded host and On-The-Go (OTG) solutions are at your fingertips.

Starting on page 42, Peter Montgomery presents the final article in his series about his animatronic remote. He covers the software and menuing code associated with the project.

In a detailed article titled "Automated Data Mining," Matt Pennell and Aaron Thomas describe how they designed an amazing embedded server application to help them with their travel plans (p. 50). The WIZnet W5100-based system uses an online travel search engine to look for flights.

Authors Phil Laird and Jeff Shewan recently built a robust hardware platform to run rule sets for old pinball machines (p. 62). The custom design emulates the hardware for nearly 500 pinball machines.

As usual, our columnists touch on a wide variety of topics. Ed Nisley covers the topics of combined audio-digital circuitry and GPS technology (p. 34). In his second article of the issue, Jeff Bachiochi explains how he tackled an RS-232-port-related problem with an innovative serial terminal solution (p. 58). Robert Lacoste describes the process of reproducing old cable shielding experiments and then explains the results (p. 70). And finally, beginning on page 78, Tom Cantrell introduces you to infrared noncontact temperature sensing. Perhaps his article will lead you to implement an infrared noncontact temperature-sensing IC in your next design.

cj@circuitcellar.com

CIRCUIT CELLAR®

THE MAGAZINE FOR COMPUTER APPLICATIONS

FOUNDER/EDITORIAL DIRECTOR
Steve Ciarcia

MANAGING EDITOR
C. J. Abate

WEST COAST EDITOR
Tom Cantrell

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

NEW PRODUCTS EDITOR
John Gorsky

PROJECT EDITORS
Gary Bodley
Ken Davidson
David Tweed

ASSOCIATE EDITOR
Jesse Smolin

CHIEF FINANCIAL OFFICER
Jeannette Ciarcia

MEDIA CONSULTANT
Dan Rodrigues

CUSTOMER SERVICE
Debbie Lavoie

CONTROLLER
Jeff Yanco

ART DIRECTOR
KC Prescott

GRAPHIC DESIGNERS
Grace Chen
Carey Penney

STAFF ENGINEER
John Gorsky

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.





Tiny Tool. Big Productivity.

Introducing Logic, the new Logic Analyzer with SPI, Serial, and I2C. Now shipping for $149.

Beautiful Utility

Encased in anodized aluminum, Logic delivers 8 channels, a sample rate of up to 24MHz, and a sample depth in the hundreds-of-millions.

Only the Best

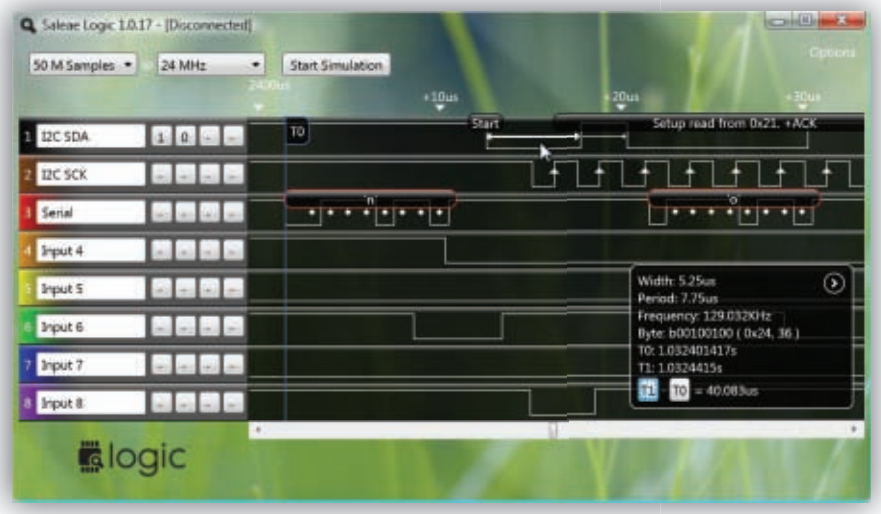
Logic comes standard with professional-grade E-Z-Hook XKM Micro-Hook probes and an ultra-flexible wire harness.

Intuitive & Powerful

While just below the surface Logic provides some serious problem-solving horsepower, you won't have to struggle to get it up and running.

Viewport Bliss

Click and drag, or even flick the viewport to scroll. Seamlessly zoom in and out with the scroll wheel. Comprehensive meta data is displayed wherever you point, and timing markers conveniently snap to waveform edges.



Protocol Analyzers

Logic ships with support for Serial, SPI, and I2C protocols. Setup is quick and intuitive with Logic's graphical analyzer setup dialogs. An SDK is also available to make your own custom protocol analyzer.

Data Export

Logic provides several ways to export data and take screenshots. It also lets you save and load complete sessions. These sessions are compressed and can even be emailed to colleagues.

logic

Decide if Logic is worthy of your workbench on **saleae.com**.

Saleae

October 2008: Signal Processing

FEATURES

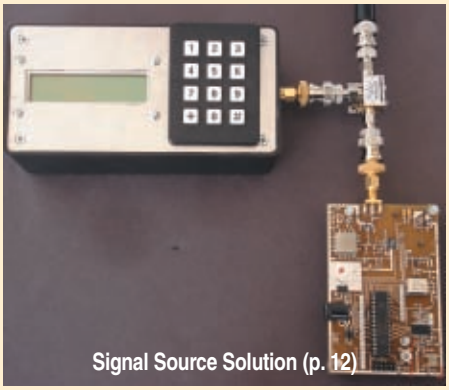
12 **DIY Signal Generation**
Create a Signal Source for a Variety of RF Applications
Neal Martini

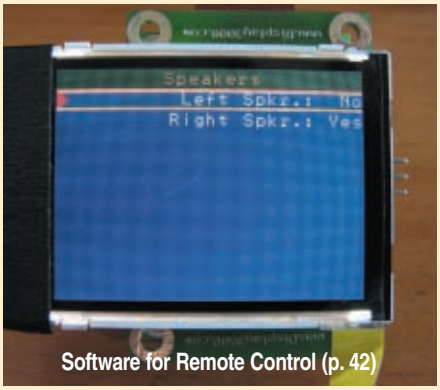
24 **SPECIAL FEATURE**
Get Started with PIC USB Connectivity
Jeff Bachiochi

42 **Dynamic Animatronic Remote (Part 2)**
Software and Menuing Code
Peter Montgomery

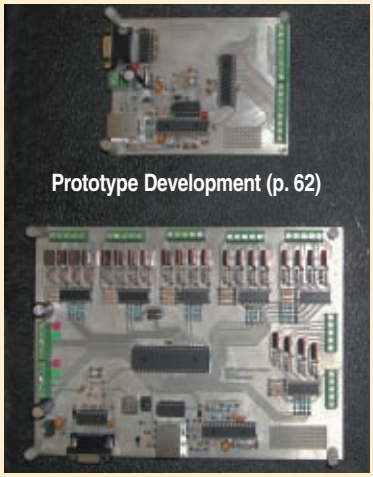
50 **Automated Data Mining**
Build an Embedded Server Application
Matt Pennell and Aaron Thomas
Honorable Mention WIZnet iEthernet Design Contest 2007

62 **Universal Machine Control**
How Hardware Emulation Leads to Prototype Development
Phil Laird and Jeff Shewan


Signal Source Solution (p. 12)



Software for Remote Control (p. 42)


Prototype Development (p. 62)

COLUMNS

34 **ABOVE THE GROUND PLANE**
HT Audio, GPS for APRS, and What Works
Ed Nisley

58 **FROM THE BENCH**
Serial Terminal Solution
Jeff Bachiochi

70 **THE DARKER SIDE**
Cable Shielding Experiments
Robert Lacoste

78 **SILICON UPDATE**
Hot Enough for You?
Seeing Red (Infrared, That Is)
Tom Cantrell


A Rebuilt Radio Interface with GPS Tracking (p. 34)

DEPARTMENTS

4 **TASK MANAGER**
What Makes an Engineer Tick?
C. J. Abate

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

93 **CROSSWORD**

94 **INDEX OF ADVERTISERS**
November Preview

96 **PRIORITY INTERRUPT**
Horses and Hard Drives
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.

DUAL-CHANNEL 40-MHz DIGITAL STORAGE OSCILLOSCOPE



The new **Model 2532** dual-channel digital storage oscilloscope (DSO) delivers an unmatched combination of performance and value. This versatile bench-top unit is a 40-MHz digital oscilloscope with analog style knobs, controls, and a 500-Msps sample rate. The AUTO button makes this oscilloscope easy to use. Advanced triggering, automatic measurements, and FFT functions provide you with many tools to debug circuits.

The Model 2532 comes with PC software that can be used to easily capture, save, and analyze waveforms and measurement results. It is an ideal oscilloscope for use in education, training, maintenance, manufacturing, and quality control.

The Model 2532's features include a color LCD, a 4,000-point record length for each channel, and one-touch automatic setup for ease of use. It also offers a USB device port on the back, eleven automatic measurements, FFT standard plus four additional math functions, and extensive triggering capabilities including pulse-width and line-selectable video trigger.

Measuring 11.4″ wide × 11.8″ deep × 5.9″ high, the Model 2532 weighs 10 lb. and comes complete with an operating manual, a 10:1 probe set (two pieces), a power cord, a USB interface cable, and a PC software installation disk. The unit costs **$699** and is available for immediate delivery.

B&K Precision Corp.
www.bkprecision.com

PROGRAMMABLE SMART VIBRATION SWITCH



The **Series 686B USB Programmable Smart Vibration Switch** is designed for 24/7 continuous monitoring and protection of cooling towers, fin fans, pumps, HVAC systems, and other critical machinery.

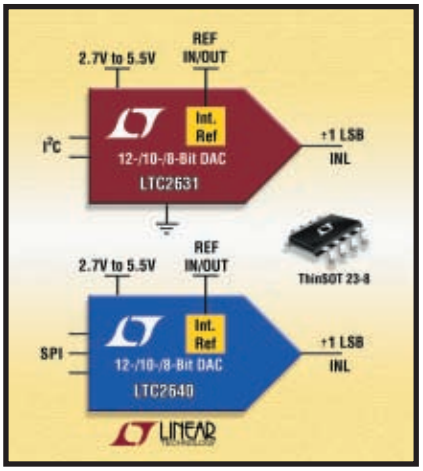
The 686B offers a smart alternative to unreliable mechanical switch technology. The electronic switch has much better accuracy and reliability than traditional mechanical switches and is easily installed in place of existing mechanical switches because it requires only two wires and can replace legacy switches without the need for additional cable runs. The Series 686B also offers a remote reset capability and USB programmable delays to avoid false trips. This universally powered unit switch is hermetically sealed for use in the harshest of environments and mounts with a single stud like a sensor. It is available for use in hazardous areas.

The Series 686B switches cost under **$700** each and are available from stock.

IMI Sensors
www.pcb.com

I²C DAC FAMILY WITH AN INTERNAL REFERENCE IN A TSOT-23 PACKAGE

The **LTC2631** is a family of pin- and software-compatible 12-bit, 10-bit, and 8-bit DACs in a tiny TSOT-23 package, making it the smallest I²C DAC family with an internal reference on the market today. DACs in such small packages are ideal for fine-tuning bias voltages in applications such as optical networking and RFID systems. The LTC2631 makes the integrated 10 ppm/°C reference output available to drive the reference inputs for all other data converters on the board, while maintaining good stability over temperature.



The LTC2631 releases alongside a SPI-equivalent LTC2640 12-/10-/8-bit family, each offering the choice of a 2.5- or 4.096-V full-scale output range. Alternatively, an external reference can be applied if more accuracy or a nonstandard voltage range is required. The LTC2631/LTC2640 DAC outputs can power up to zero-scale or mid-scale with the internal or external reference, allowing more flexibility for designs that cannot be forced to ground when power is first applied.

The LTC2631 and LTC2640 DACs both offer excellent 12-bit DC performance of ±1 LSB (max) integral nonlinearity error, enabling operation in open-loop as well as closed-loop systems.

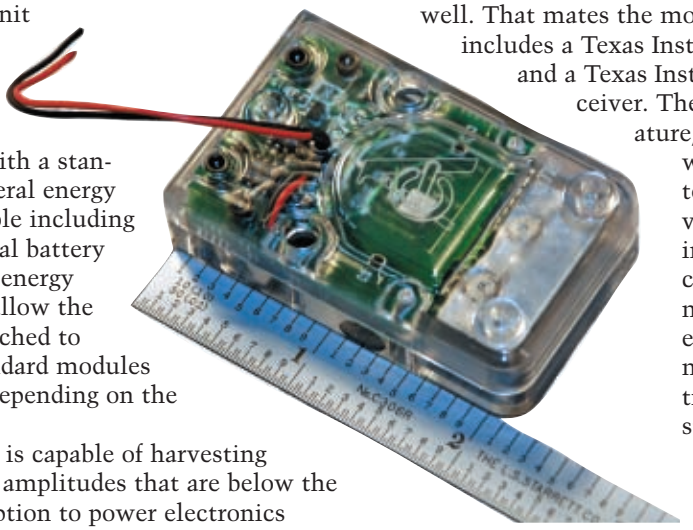
The controllers begin at **$1.03** each in 1,000-piece quantities.

Linear Technology Corp.
www.linear.com

NEW PRODUCT NEWS

ENERGY HARVESTING MODULE

The **Joule-Thief Energy Harvesting Module** is a compact, fully self-contained vibration energy harvester. The module contains an energy harvesting beam based on proprietary ruggedized laminated piezo (RLP) technology, as well as proprietary energy collection and storage circuitry. The module is the highest output vibration energy-harvesting device per unit weight and volume of any commercially available energy harvester.



The module is designed to provide you with a standard 3.6-VDC output. Several energy storage options are available including capacitors as well as several battery technologies. The various energy storage options available allow the storage capacity to be matched to your application. The standard modules store from 1.4 to 430 mJ depending on the storage device used.

The Joule-Thief module is capable of harvesting usable energy at vibration amplitudes that are below the threshold of human perception to power electronics devices such as wireless sensors. The modules are designed to harvest energy from typical AC-powered devices. Thus, they are available in standard frequencies of 60 and 120 Hz with custom frequencies available upon request.

A wireless sensor demonstration kit is available as well. That mates the module to a sensor board, which includes a Texas Instruments MSP430 microcontroller and a Texas Instruments CC2500 wireless transceiver. The sensor board includes a temperature, pressure, and light sensor, as well as a switch and potentiometer to allow experimentation with vibration-powered wireless sensing, switching, and proportional control. All of the sensors, the microcontroller, and the transceiver are completely powered by the module. The sensor data is then transmitted to any PC. Demonstration kits are available for **$699**.

AdaptivEnergy
www.RLPenergy.com





WIZWIKI.net

Create applications for the WIZnet ioffload™ platform and then sell them worldwide on WIZwiki.net.

All kinds of WIZnet 3rd party are qualified.

- VAR (Value Added Reseller)
- SP (Solution Provider)
- DH (Design House)
- Prosumer (Producer + Consumer)



Please secure your own brand shop.

[Inquiry]
skype ID : wizwiki



WIZnet
www.wizwiki.net
www.ewiznet.com
www.wiznet.co.kr

NEW PRODUCT NEWS

SURFACE-MOUNT LED WITH THROUGH-HOLE T-5-MM LED FEATURES

The **OctoLEDs (SML-H1505*xxx*-TR)** line of surface-mount devices represents the first commercial technology to combine the convenience of SMT with the features and capabilities of through-hole, leaded T-5-mm epoxy lens LEDs. The new OctoLEDs deliver brightness levels of up to 7,000 mcd (dependent on color)—a market first from any conventional SMT LED driven by the common I_F of 20 mA.

The LEDs can be used anywhere the convenience of surface mounting is required, with the high brightness, color, and viewing angle choices of standard T-5-mm LEDs. Such applications include front panel, PCB multi-LED, and industrial control or fault indicators. OctoLEDs have an off-the-board height of 11 mm and they occupy a circuit board footprint which is 6 mm × 6 mm. They are available on standard 24-mm tape for automatic insertion.

The OctoLEDs are available in the full range of colors including red, yellow, green, blue, or white. Bicolor as well as RGB versions are also offered. The LEDs are available with a variety of lens finishes, opening the door to different viewing angle options. (The angle will vary depending on the type of lens.)



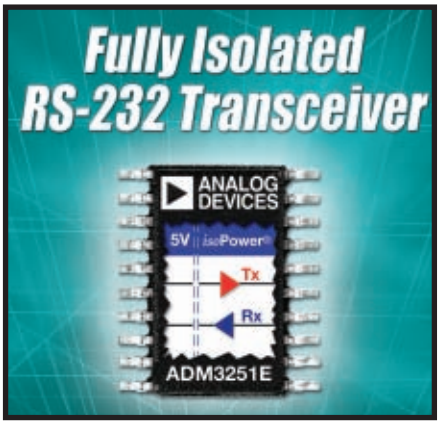
The cost of the unit depends on the color specified and quantity ordered. It is approximately **$0.30** per unit in production quantities.

Lumex, Inc.
www.lumex.com

FULLY ISOLATED RS-232 TRANSCEIVER FOR HARSH ENVIRONMENTS

Featuring a small form factor SMT package, the **ADM3251E** transceiver integrates a line driver, line receiver, oscillator, rectifier, regulator, voltage doubler, voltage inverter, and transformers into a single-chip solution that isolates both the data and power lines on chip. The highly integrated device provides a robust serial communications interface with isolated power for use in harsh HVAC, industrial, and medical applications.

The ADM3251E transceiver incorporates iCoupler and isoPower isolation technologies. It is the first surface-mount RS-232 transceiver to feature full isolation of both the data lines and power. With a 2.5-kV isolation rating, the new RS-232 transceiver complies with industry-standard isolation regulations, including UL1577 and DIN VDE 0884-10, ensuring it will meet the robustness levels required in noisy operating environments such as isolated RS-232 diagnostic data ports in HVAC building control systems, industrial automation and control applications, point-of-sale equipment, programmable logic controllers, telecom, and medical equipment. The SMT package reduces board space by 45% compared with traditional ICs and helps accelerate the manufacturing process and lower system development costs.



The transceiver costs **$2.99** per unit in 1,000-unit quantities.

Analog Devices, Inc.
www.analog.com

NEW PRODUCT NEWS

ICC FOR PROPELLER

The **ICC for Propeller** is an ANSI C development tool for the Parallax Propeller chip. The IDE features project-based design and supports C86 dialect source and C-based Propeller objects. The Propellent library is directly supported by the IDE for ease in build-to-run development cycles.

ICC for Propeller comes in both a standard version and a non-commercial version. Both versions feature an ANSI C compiler for Propeller LMM code generation with an editor, project manager, and code browser. The non-commercial version supports programs as large as 16 KB while the standard version supports programs as large as hub RAM.

The Propeller chip makes it easy to rapidly develop embedded applications. Its eight processors (cogs) can operate simultaneously, either independently or cooperatively, sharing common resources through a central hub. The developer has full control over how and when each cog is employed. There is no compiler-driven or operating system-driven splitting of tasks among multiple cogs. A shared system clock keeps each cog on the same time reference, allowing for true deterministic timing and synchronization.



The standard version costs **$249**. The non-commercial version is only **$99**.

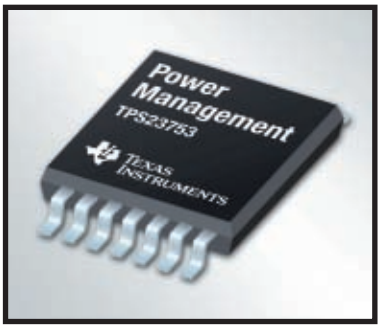
ImageCraft, Inc.
www.imagecraft.com

POWER-OVER-ETHERNET CONTROLLER TO OPTIMIZE SUB-13-W DESIGNS

The **TPS23753** is a 13-W PoE-powered device controller with an on-board DC/DC converter specifically designed to meet the needs of IEEE 802.3af-compliant applications. The highly integrated device eases system design and size for PoE applications, including IP phones, IP security cameras, wireless access points, security card readers, and RFID scanners, to name a few. The TPS23753 is optimized for isolated DC/DC switching topologies, including flyback topology.

The TPS23753 will operate on a wide range of local power supplies in cases where PoE power is not available. The device operates from 12 to 57 V, including operation down to 9.5 V to support droop on any 12-V local power supply. The DC/DC converter allows for operation in this range with an 80% duty cycle and current slope compensation, reducing output rectifier stress for 12-V adapters.

To ease design with local power supplies, the TPS23753 features adapter priority programming that enables you to select which power source to use when both PoE and local power are available. Native support for this prioritization eases design and reduces system size by eliminating external components.



The suggested resale price is **$1.45** in quantities of 1,000. Evaluation modules with the TPS23753 are also available.

Texas Instruments, Inc.
www.ti.com



DIY Signal Generation

Create a Signal Source for a Variety of RF Applications

Neal describes how he built an inexpensive and accurate variable-frequency signal generator on a two-sided PCB. The system delivers frequencies from 0 to 3 GHz with a frequency accuracy of less than 200 Hz.

If you pursue designs that take you into the world of radio frequency (RF) technology, you may need a stable, high-frequency signal source. If you are interested in a quality source that won't break the bank, read on.

Last fall, I began working on a project involving high-frequency narrow-band frequency shift keying. The project required several stable, high-frequency signal sources located at multiple physical locations simultaneously. Rather than build several narrowband sources or purchase a number of expensive wideband sources, I decided to build a variable-frequency stable source that was general-purpose, accurate, and affordable.

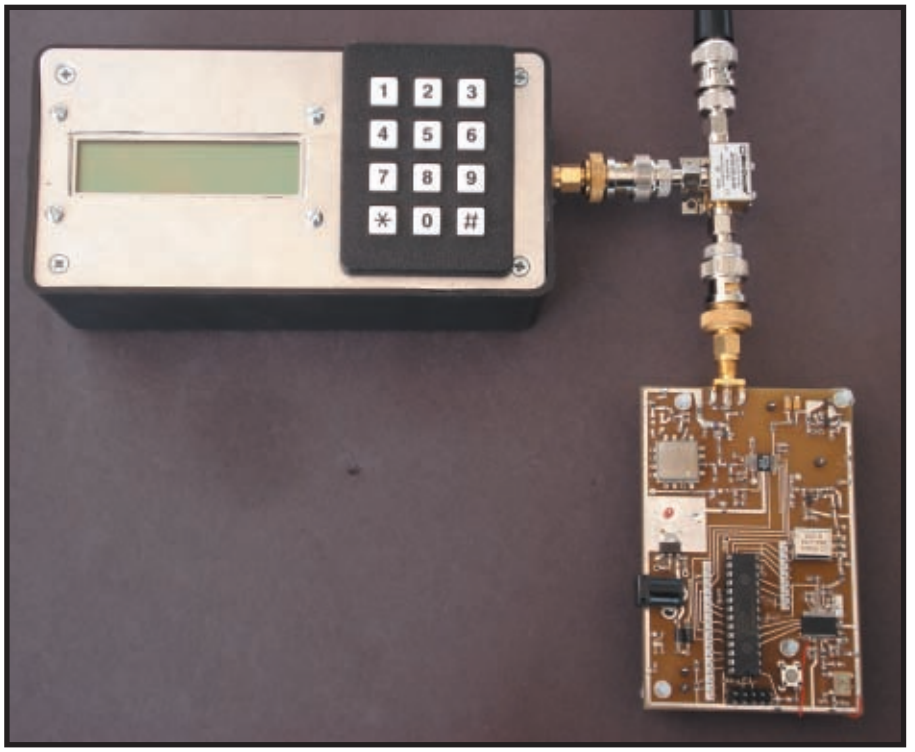


Photo 1—Two RF generators and an off-the-shelf passive RF mixer are used to generate lower frequencies. If, for example, one generator is set to 2 GHz and the other generator is set to 2.3 GHz, the difference signal output of the mixer will be a 300-MHz signal.

The goal for the RF signal generator was for it to deliver an operator-selected frequency from 0 to 3 GHz with a frequency accuracy of less than 200 Hz (see Photo 1). The targeted output power level was 10 dBm, with a phase noise specification of –80 dBc at a 10-kHz offset. My other goal was to keep the harmonic levels greater than 40 dB below the fundamental. To control costs, I decided to implement the design on a single two-sided PCB, including the power supply.

Many of these design goals were met or exceeded. Some are still "works in progress." I think you will agree that this design will serve as an excellent signal source for many RF applications.

ARCHITECTURAL TRADE-OFFS

A phase-locked loop (PLL) is an obvious choice for controlling frequency in signal generator designs. But a PLL alone does not work in this application.

Look at the dotted-line box in Figure 1. These are the primary elements of a typical PLL. It contains a voltage-controlled oscillator (VCO), a PLL controller, and a loop filter. Typically, a fixed-frequency clock provides a stable reference for the PLL control chip. The PLL output frequency is controlled by setting the values of the "R" and "N" dividers. Once the

dividers are set up, the control chip compares the reference clock (F_{REF}) divided by R to the VCO output (F_{VCO}) divided by N. The phase detector in the controller compares the two divided-down signals and generates an error signal proportional to the difference in frequencies. The error signal is filtered by the loop filter and drives the VCO until the PLL locks:

$$\frac{F_{VCO}}{N} = \frac{F_{REF}}{R} \quad [1]$$

The equation for the VCO output frequency is:

$$F_{VCO} = F_{REF} \times \left(\frac{N}{R}\right) \quad [2]$$

I considered only an integer PLL for this design. With an integer PLL, R and N can have only integer values. A fractional PLL allows fractional values for N, but it has a noisy output spectrum and is not yet ready for prime time at these frequencies. Because R and N are integers, the size of the PLL's frequency steps is determined by the rate that the phase comparison is done in the PLL controller. This rate is called the phase detect frequency (PDF):

$$PDF = \frac{F_{VCO}}{N} = \text{Frequency step size} \quad [3]$$

N has to be large to take small frequency steps, but there are two problems with this. First, the selected PLL control chip determines the maximum size of N. For the chip I selected, the maximum N is 524,288. With a 3-GHz VCO, N = 524,288 results in a minimum frequency step of 5,722 Hz. Even if the control chip had a larger N value, there is a more compelling limitation. It turns out that the noise from sources in the PLL (reference, phase detector, and loop filter) are increased by a factor of 20 log N decibels. This says you want to keep N as small as possible. Another factor that says you want N to be small and PDF large is that the PLL's closed-loop bandwidth can be increased as the PDF increases. Increasing bandwidth makes for a responsive PLL.

I had a problem! I wanted a small PDF (large N) for small frequency steps and a large PDF (small N) for low-noise signals and good bandwidth. Faced with this problem, I went to the Internet to find how others solved the problem. (Refer to the References section at the end of this article.) There, I found an excellent have-my-cake-and-eat-it-too solution. Using direct digital synthesis (DDS) technology to precisely vary the F_{REF} term in Equation 2 was the answer.

ARCHITECTURE

DDS technology gives you the ability to generate sampled sine waves whose frequencies can be digitally controlled in small, precise steps. In fact, if DDS was available up to 3 GHz, it could be a good stand-alone solution in this application, but the technology isn't there yet.

The best way to explain how DDS is applied here is with an example. Assume that the reference clock F_{REF} into the PLL is 10 MHz. Also assume you want to use a PDF of 1 MHz, which enables you to have relatively small N values. Setting the PDF to 1 MHz establishes the PLL step size at 1 MHz. With PLL steps of 1 MHz, you need a way to get the generator to produce the frequencies between PLL steps. Refer to Equation 2. This can be accomplished by varying the F_{REF} term. So, if you

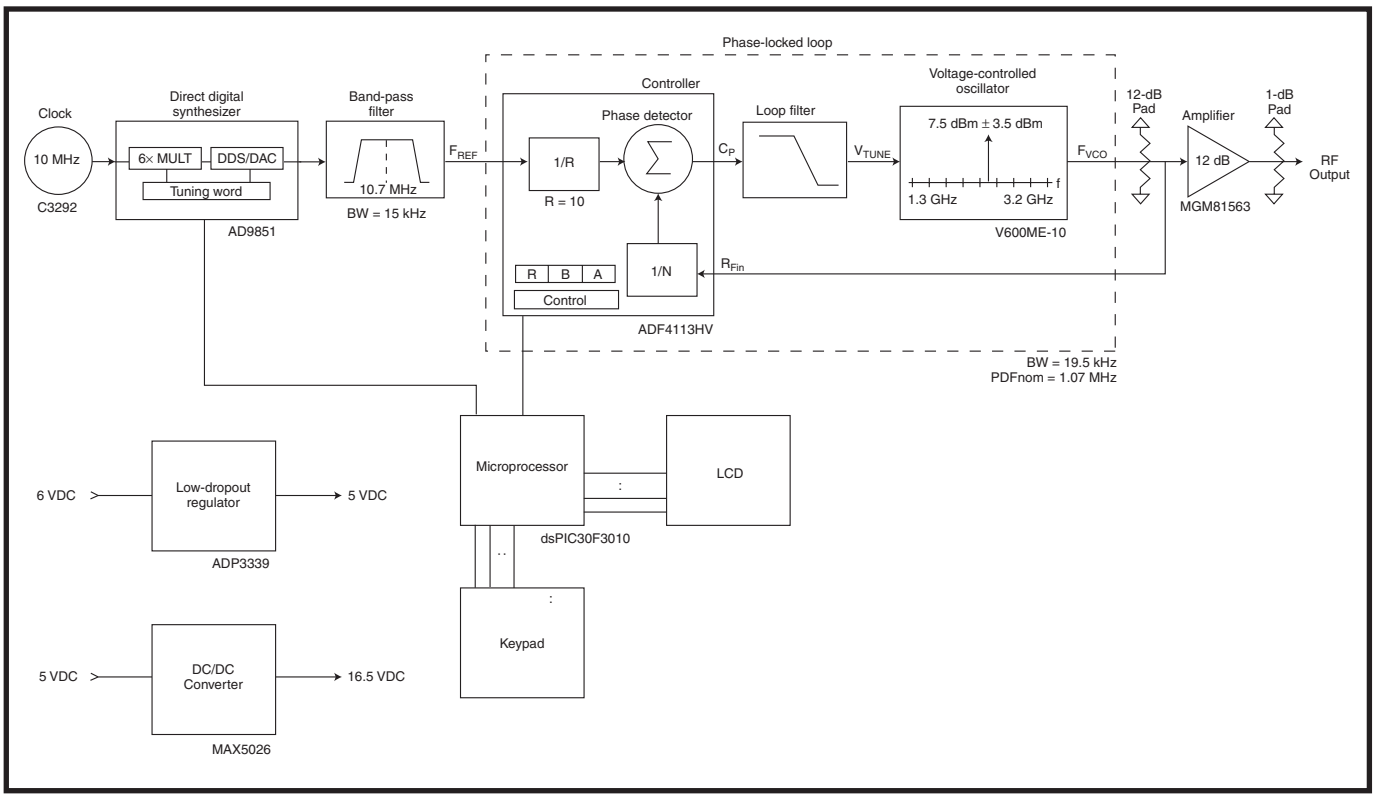


Figure 1—A DDS generator is used to supply the reference clock for the PLL. By microstepping the DDS, the RF generator can output frequencies with less than 4-Hz accuracy.

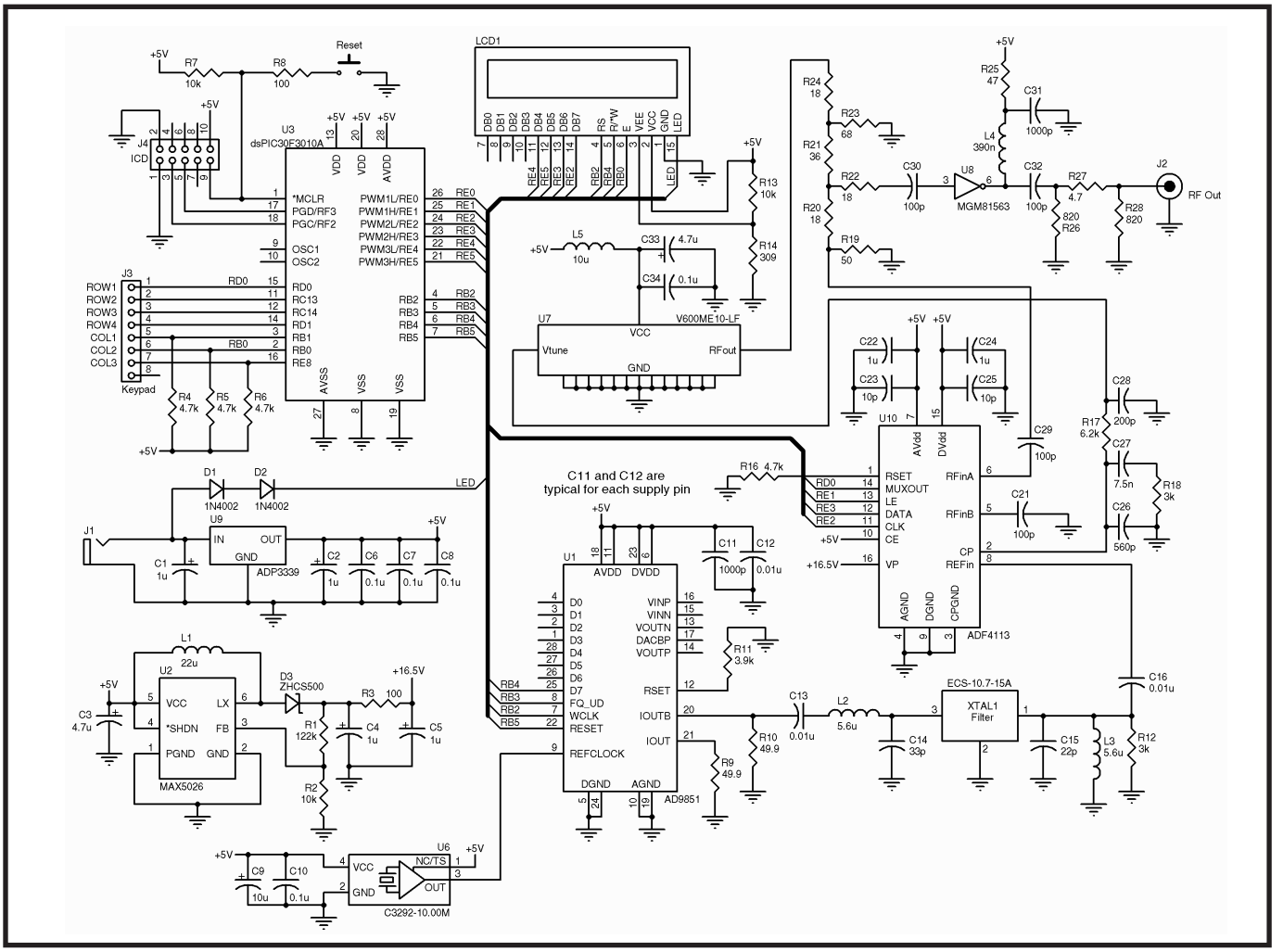


Figure 2—This RF generator schematic shows the simplicity of the design. The main PLL components are an Analog Devices ADF4113HV and a Z-Communications V600ME10-LF. The heart of the DDS is an Analog Devices AD9851. Also note the simplicity of the power supplies.

vary the F_{REF} frequency in small steps, you vary the VCO output frequency in small steps as well. In my example, let's say you have R and N set up to generate a 3-GHz signal. Therefore, with a PDF of 1 MHz, R = 10, and N = 3,000. Keeping R and N fixed and increasing the F_{REF} by, say, one to 10,000,001 Hz, you get:

$$F_{VCO} = 10{,}000{,}001 \times \frac{3{,}000}{10} = 3{,}000{,}000{,}300 \text{ Hz} \quad [4]$$

This is a frequency step size of 300 Hz at 3 GHz. As it turns out, you can get much smaller F_{REF} steps using DDS (0.014 Hz), which yields much smaller VCO output steps (4-Hz steps at 3 GHz). Consequently, I have what I want: small, precise step sizes using DDS and excellent output noise performance using a large PLL PDF.

Now let's consider the RF generator's flow diagram shown in Figure 1, starting with the PLL enclosed by the dashed-line rectangle. To simplify the design, I cover as much of the desired output spectrum with a single VCO. Here I employ a wideband VCO that produces a sinusoid with a frequency of 1.3 to 3.2 GHz. The output of the VCO is fed back to the PLL controller. This feedback signal is divided by N and then compared to the input reference clock divided by R, as described earlier. The controller's error signal output is then filtered by the loop filter to drive the VCO. When the loop locks:

$$\frac{F_{REF}}{R} = \frac{F_{VCO}}{N} \quad [5]$$

The VCO also feeds an RF buffer amplifier, which delivers the desired output levels.

The F_{REF} input to the PLL is provided by the DDS generator. The DDS section consists of a fixed-frequency input clock, a DDS generator chip, and an output filter. A 10-MHz clock is input to the DDS controller and is multiplied



Photo 2—The RF generator fits on a single two-sided PCB, including the power supply. The bottom side is mostly ground plane, and most of the components are SMDs. The LCD and keypad plug directly into the PCB as well.

by the DDS control chip to develop a 60-MHz internal system clock. The DDS controller generates a sampled sine wave whose frequency is a function of the value in the controller's tuning word. The sampled sine wave is internally converted to analog via a 10-bit high-speed DAC. The nominal frequency generated by the DDS is 10.7 MHz. (I'll explain the reason for this frequency later.) But the exact frequency varies from the nominal 10.7 MHz by an amount required to microstep the output of the VCO.

The raw output of the DDS controller is not a clean sine wave. Because the signal is inherently a sampled sine wave, standard Nyquist criteria apply. Consequently, there are a variety of alias components that need to be filtered out. That is the function of the 10.7-MHz filter located at the output of the DDS controller. It should be noted that as long as the DDS signal falls within the bandwidth of the filter, it will pass intact. Therefore, the allowable output frequency range for the DDS controller is 10.7 MHz ±7.5 kHz. As I will explain, this is adequate for varying the F_{REF} to the PLL enough to cover the frequency gaps between the available steps in the PLL.

The rest of the flow is fairly straightforward. The microprocessor controls all of the registers in the chips, the keypad is used to input frequency settings, and the LCD is used to display the current state of the generator. A 5-VDC regulator and a 16.5-VDC regulator are included to generate the needed DC supply voltages.

HARDWARE

A schematic of the RF signal generator is shown in Figure 2. The PCB is shown in Photo 2.

You want a VCO that covers a wide frequency range. A single VCO that covers 0 to 3 GHz is not available. So, I selected a Z-Communications V600ME10-LF, which operates from 1.3 to 3.2 GHz. It was chosen for its wide frequency coverage and because it operates off a 5-V power supply. It's specified output level is 7.5 dBm ±3.5 dBm, which is sufficient to deliver the power levels I want. Its phase noise specification is –89 dBc at 10 kHz offset, which meets my stated goals. The V_{TUNE} input that controls the VCO's

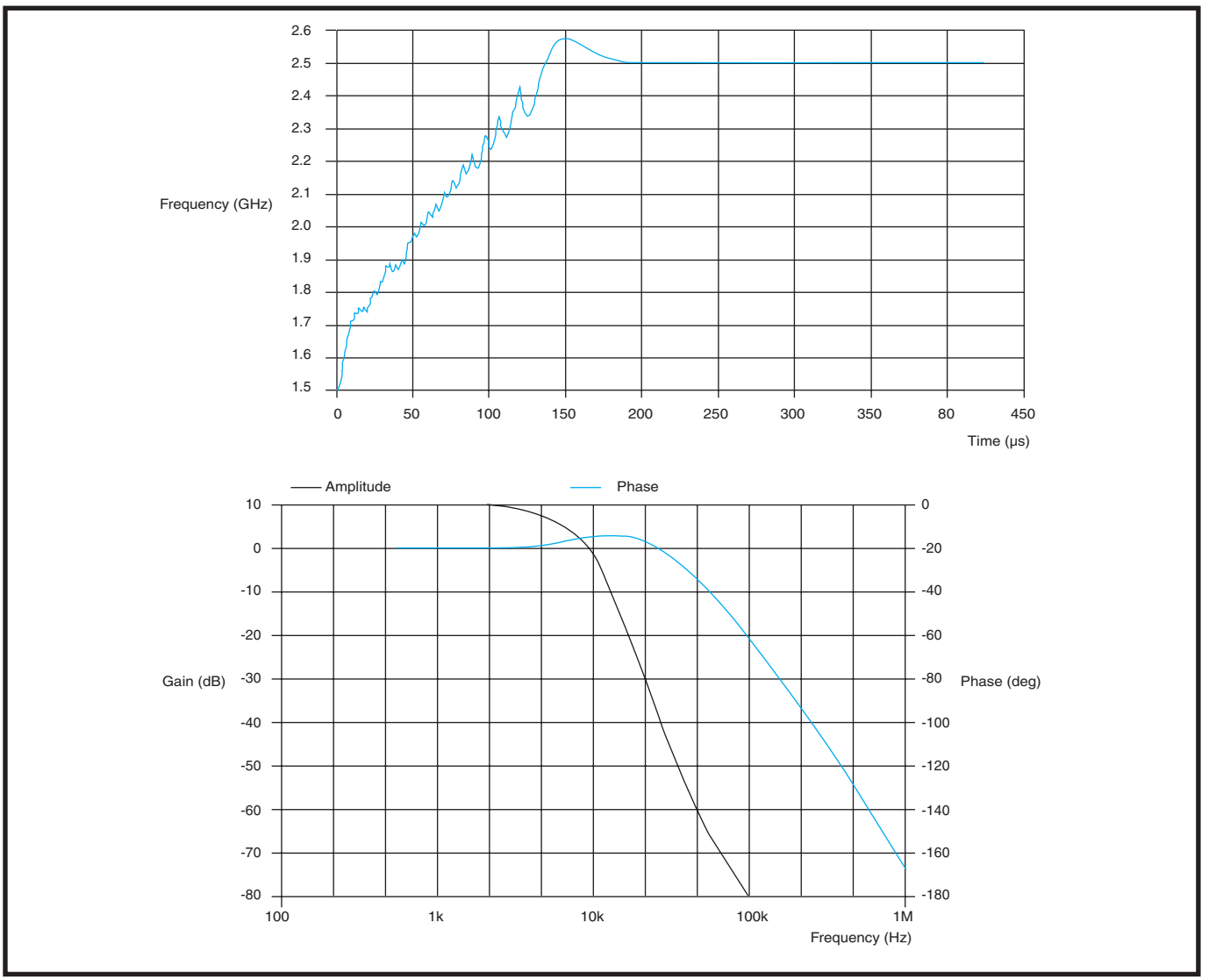


Figure 3—The ADIsimPLL tool from Analog Devices does a thorough job designing the PLL loop filter. The PLL step response and closed-loop transfer plots shown here are a few examples of output generated by the tool.

frequency has a range of 0 to 20 V.

I used an Analog Devices ADF4113HV PLL control chip. I picked it because it operates from 200 MHz to 4 GHz. More importantly, however, this relatively new component from Analog has an internal high-voltage charge pump. This eliminates the need for an external op-amp at the output of the control chip, which is normally included to convert the low-voltage charge pump signal to a level high enough to drive the V_{TUNE} control line of the VCO.

There are five components (R17, R18, C26, C27, and C28) that make up the PLL loop filter. The filter's main purpose is to take the current pulse error signal generated by the PLL control chip and generate a clean proportional DC voltage to the VCO's V_{TUNE} control line. This filter affects loop stability, loop BW, and the overall PLL output noise. There are textbooks that describe how to design these filters. Fortunately, there are online resources available to help with the heavy lifting. I used the ADIsimPLL tool from Analog to design the filter for this application. Because I used an Analog PLL control chip, I just needed to input the specifications of the Z-Communications VCO and the phase detector frequency I wanted. The ADIsimPLL did the rest. It output a schematic and open- and closed-loop transfer plots, step response plots, and a variety of other performance data. Some examples of the output of the tool are shown in Figure 3. The resulting PLL designed using this tool has a bandwidth of 19.5 kHz and an output phase noise of –81 dBc at a 10-kHz offset (assuming an ideal reference clock input).

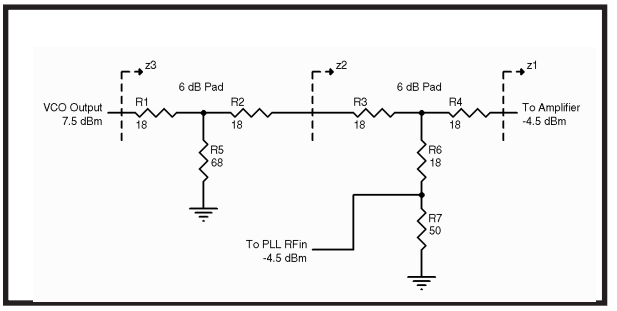


Figure 4—Two 6-dB pads in series accomplish the attenuation to provide the right signal level to the PLL control chip. The attenuation also minimizes the effects of source/load mismatches at the output of the VCO.

On the output side of the VCO, a network of resistors is used to feed back the output signal to the control chip. The network interfaces the VCO to an output RF amplifier. When working with RF frequencies, make sure that the loads seen by devices like the VCO are 50 Ω. This minimizes the effects of source/load mismatches like power level variations and frequency shifting. Attenuating the output is a technique used to minimize mismatch effects. In addition, you need to feed back the VCO's output to the ADF4113HV. However, the feedback signal must be in the range of 0 to –10 dBm. Attenuation of 12 dB solves both problems.

The 12-dB attenuation is accomplished by using resistor pads. Figure 4 shows how the resistor network is constructed. The network consists of two 6-dB T pads in series. Each pad has an input and output impedance of 50 Ω. Consequently, the z1, z2, and z3 impedances in Figure 4 are all 50 Ω (this assumes that the amplifier input is 50 Ω). Assuming the VCO output is at 7.5 dBm (0.54 V_{RMS}), and doing the voltage divider math, you will see that the signal level at the 50-Ω resistor that feeds back to the PLL RF_{IN} is 0.133 V_{RMS}. This corresponds to a –4.5-dBm signal level required by the ADF4113HV PLL chip. The same is true for the signal level into the RF buffer amplifier.

Assuming I have the nominal REF_{IN} clock input to the PLL of 10.7 MHz and that I want to have a nominal PDF of 1.07 MHz, I so far have a stable, low-noise PLL with a frequency step size of 1.07 MHz. Next, I need to microstep the REF_{IN} clock so I can have the PLL generate frequencies between the 1.07-MHz steps. If you examine Equation 2, REF_{IN} must be able to vary over a range of ±3,578 Hz to microstep within the 1.07-MHz PLL step. The finer the REF_{IN} steps, the better the output resolution. Now let's examine how this is done with the DDS portion of the schematic.

The heart of the DDS circuit is the AD9851 DDS synthesizer chip. It takes a 10-MHz CLK IN, multiplies the frequency by six, digitally synthesizes a sine wave, and D/A converts it to generate a sinusoid with precise frequency control. The equation controlling this device is:

AD9851 output frequency = (Tuning Word) (6) (CLK IN)/2^{32}

The Tuning Word is 32 bits long, so high resolution is possible. With a 10-MHz CLK IN, the AD9851 frequency step size is 0.014 Hz. A 0.014-Hz step size in the DDS corresponds to an RF generator output step size of less than 4 Hz over the full operating range of 1.3 to 3.2 GHz. That's excellent frequency resolution!

The AD9851 is a digitally sampled waveform generator, so it has several components in its output spectrum that you need to clean up. That is why the AD9851 is followed by a narrowband crystal filter. The filter's bandwidth is wide enough to pass the signal over the ±3,578-Hz variations required for microstepping, but it is narrow enough to eliminate the undesired components of the AD9851 sampled

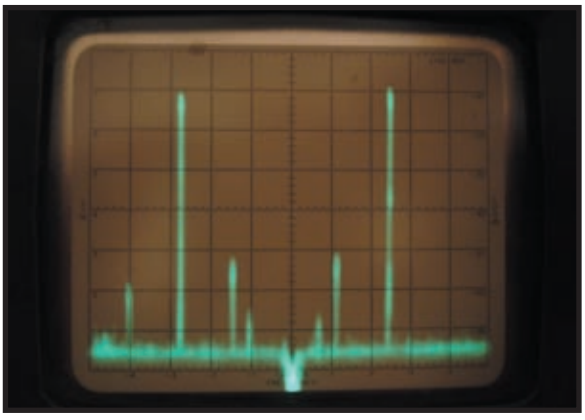


Photo 3—To view the fundamental and harmonics of a 3-GHz signal, an older-model spectrum analyzer was used. The right-most line is the 3-GHz fundamental. Moving left, the next two lines are the 6- and 9-GHz, second and third harmonics, respectively. The other lines are due to the analyzer's harmonic mixing and can be ignored.

New OLED Switch & Display

NKK® SWITCHES



INFINITE CLARITY

DO YOU SEE WITH INFINITE CLARITY?

With NKK's new line of OLED products, now you can! From the market leader for miniature and illuminated switches comes the newest and most cutting-edge technology offered in broadline OLED programmable switches and displays. Suitable for a range of industries, NKK's OLED devices feature:

- 180° Viewing angle
- 65,536 colors
- Real-time sequencing
- Less power consumption
- Higher contrast
- Brighter
- Vivid clarity



OLED SmartSwitch™ Development Kit

SMARTSWITCH™

ACTIVATIONS MADE SIMPLE

http://cc.nkksmartswitch.com
1.877.2BUYNKK (228.9655)

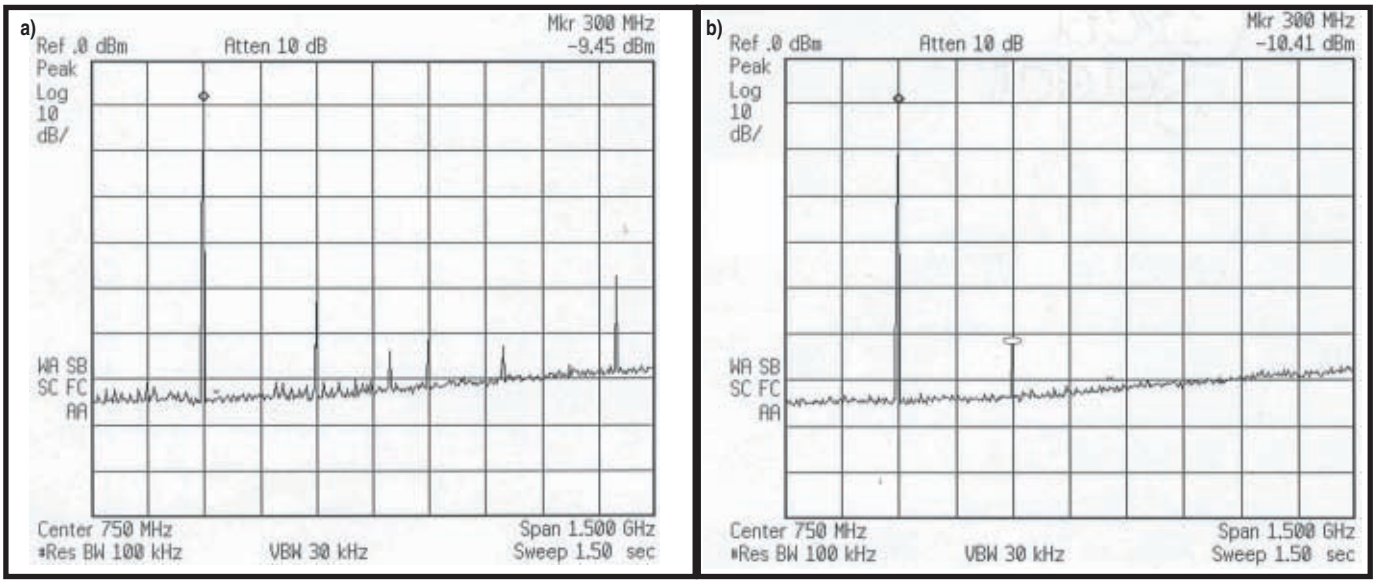


Photo 4a—This is the output obtained using two RF generators and a mixer. **b**—This is the output of an expensive generator set at 300 MHz.

waveform. An inexpensive 10.7-MHz crystal filter was selected for this function, and it is the reason why the nominal reference frequency to the PLL was set at 10.7 MHz.

The input impedance of the crystal filter is nominally 3,000 Ω/2 pF, and it is driven from a 50-Ω source, so an impedance-matching network is required. Crystals are sensitive to having their input and output impedances matched to deliver their specified filtering. An LC network is used to accomplish this matching. The L and C values were determined using an online tool. (Refer to the References section at the end of this article.) The tool called for L = 5.7 µH and C = 38 pF. Taking into account stray capacitance, the 2 pF of internal capacitance at the crystal input, and the standard values available, the values of 33 pf and 5.6 µH were selected. How do these small components match a 50-Ω source to a 3,000-Ω load? An intuitive explanation is that the shunt capacitor is there to lower the crystal's 3,000-Ω input impedance to 50 Ω, and the inductor is included to cancel out the reactive component introduced by the capacitor. Mathematically, if you look at the impedance of the shunt capacitor (38 pF) and the 3,000-Ω crystal input in parallel, you get:

$$Z = \frac{-(391j)(3{,}000)}{-391j + 3{,}000} = \frac{\frac{1{,}173{,}000}{-90^\circ}}{\frac{3{,}025}{-7.4^\circ}} = \frac{388}{-82.6^\circ} \quad [6]$$

$$Z = 50 - 385j$$

So, if you get rid of the –385j reactive term, you will have the 50-Ω impedance you want. To do this, because the inductor's impedance adds in series with the above Z, set jwL = 385j, yielding L = 5.7 µH. This leaves you with the desired 50-Ω impedance. Incidentally, the beauty of impedance matching with an LC is that you also get some additional filtering at 10.7 MHz. There is also an LC at the output of the crystal, but it is not there for

KEIL™
An ARM® Company
Software Development Tools
The Leader in Microcontroller Development Solutions
C/C++ Development Kit including best-in-class compilers, genuine Keil µVision®, and royalty-free RTX RTOS.
ULINK®2 Adapter for target debugging and Flash programming.
A/D
I/O Ports
Run-Control
Timers
Interrupts
Debug Channel
PWM
CPU
Flash ROM
UART
RAM
I²C/SPI
RTC
DMA
Ethernet
SD
USB
CAN
Keil RTOS and Middleware components are specifically optimized for embedded systems and include TCP/IP, Flash File system, USB and CAN support.
ARM
www.keil.com/arm
Cx51
www.keil.com/c51
C166
www.keil.com/c166
Out-of-the box support for more than 1,400 Microcontroller devices.
Call 1-800-348-8051 for a free demo CD.
www.keil.com

impedance matching because the crystal is driving a 3-kΩ load. It is simply an LC tuned to 10.7 MHz (it takes into account the 10-pF input impedance of the ADF4113HV) to provide additional filtering.

I chose a Microchip Technology dsPIC30F3010 microprocessor because it is fast, it has an internal clock, it has enough memory to allow for C programming (including `stdio` functions), and it has enough I/O pins to satisfy the design. Plus, I had a few left over from another project. I wanted to make sure I had plenty of bandwidth to expand the functionality of the RF generator in the future. The microprocessor controls all the registers in the ADF4113HV and the AD9851, and it performs all of the calculations using high-speed integer arithmetic. The interfaces to the keypad and LCD are also handled by the microprocessor.

I used an ADP3339 low-dropout regulator to generate the 5-V supply. It is a good solution because it can provide all of the necessary power without any heatsinking. The ADF4113HV also requires an available low-current 16.5-VDC regulated supply. I generated the supply with a MAX5026 DC/DC converter. It takes the 5 V available from the main supply and generates 16.5 V of clean output. The 16.5-V output is determined by the values of R1 and R2. The 16.5-V level covers the tuning range of the VCO while not exceeding the maximum level allowed by the ADF4113HV high-voltage charge pump supply.

	m x F_{RF}				
n x F_{LO}	2.3 GHz	4.6 GHz	6.9 GHz	9.2 GHz	11.5 GHz
2 GHz	0.3	1.4	—	—	—
4 GHz	—	0.6	—	—	—
6 GHz	—	1.4	0.9	—	—
8 GHz	—	—	1.1	1.2	—
10 GHz	—	—	—	0.8	1.5

Table 1—This table shows the frequency locations of the n x F_{LO} ± m x F_{RF}. The dashes are "don't care" values because they fall above the frequency range of the spectrum analyzer used. F_{LO} = 2 GHz, F_{RF} = 2.3 GHz, and F_{IF} = 300 MHz.

RF LAYOUT

There are a few layout considerations to think about when building RF circuits. Pay close attention to supply bypassing. Suppliers of RF components will typically suggest capacitor values, SMD component sizes, and, in some cases, even tell you how close the bypass capacitor needs to be to the pin you are bypassing. It is also good practice to return all ground connects to ground with separate vias. In other words, if you have several pins on an IC that need to be grounded, connect them to the ground plane with their own via to avoid any subtle interactions.

In this two-sided design, the bottom of the PCB is almost completely a ground plane. It is not a single plane. Instead, it is three separate planes connected to the power supply ground at a common point. This "star" configuration of the ground plane helps keep the different sections of the circuit from interacting via unwanted ground loops.

Finally, there are those who will disagree with me on this, but if you keep your high-frequency traces short (less than 0.1× the wavelength), you don't have to be too concerned about having PCB trace widths designed to yield a 50-Ω loading. You are better off if you keep things short and match the trace width to the SMD parts you are interconnecting.

FIRMWARE

I have been using C to program my past few projects. It was a little painful coming up to speed, but now I am a real convert. Using C enabled me to do 64-bit integer arithmetic with ease and gave me access to some built-in functions that were extremely useful. The `sprintf` built-in standard I/O function, for example, lets you easily create a formatted ASCII array, making formatted LCD output quite simple. For example:

```
sprintf(Array,"Frequency = %u Hz", Fvco);
```

The `sprintf` statement is all that is needed to have the LCD read "Frequency



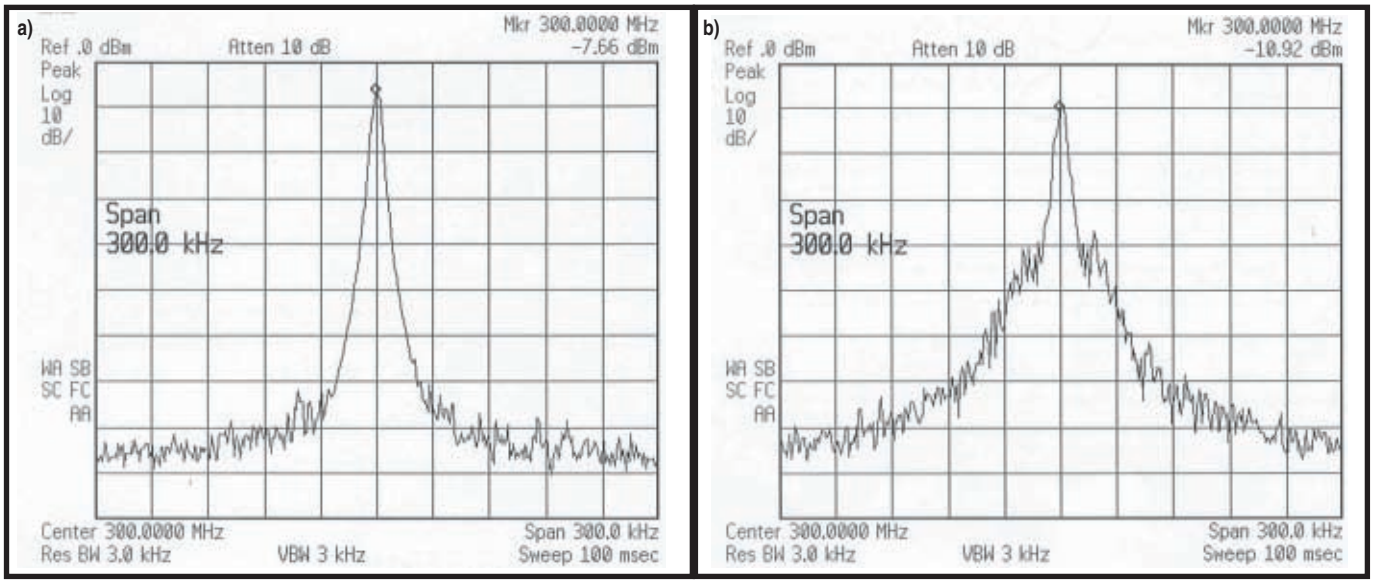


Photo 5—To look at phase noise, you need to look closer in frequency at the fundamental. **a—**This is the fundamental of an expensive generator. **b—**This plot is my RF generator. Things are comparable at 100-kHz offset, but closer in, the expensive unit has less noise.

= 300 Hz," assuming F_{VCO} is 300 at the time the command is executed.

Likewise, the standard I/O function `sscanf` lets you go the other way by taking formatted ASCII input and converting it to numerical values. This is extremely useful when you are inputting keypad values that you want to convert to numbers.

If you are tired of rewriting LCD and keypad interface programs because you are using different I/O pins on the microprocessor, look at the C code in this project. You don't have to rewrite the two programs for future projects. The two programs are written in such a way that you have to change only the pin assignments in the header files and you're set to go. For experienced C gurus, don't be too critical of my C skills. As I said, I am coming up to speed.

HOW IS IT WORKING SO FAR?

Take a look at how the RF generator performs. For operating at 3 GHz, the only instrument I have access to that shows the frequency spectrum up to 9 GHz (to see harmonics) is an old Hewlett-Packard 141T/8555A spectrum analyzer. Be careful when looking at the output of the analyzer because it uses harmonic mixing and displays the resulting spectrum in a bit of a confusing manner.

Photo 3 shows the analyzer's output. Starting on the right, the 3-GHz fundamental is displayed. Moving left, the next line is the second harmonic at 6 GHz. The next left line is the third harmonic at 9 GHz. The other lines on the display are just the same three components that were shifted into the display spectrum by harmonics of the internal local oscillator. Consequently, they can be ignored. The analyzer was used to check the output levels because the frequency of the RF generator was varied.

The results show the output level is consistently between 8 and 10 dBm as the frequency is varied from 1.3 to 3.2 GHz. Also, the second and third harmonics are typically at –42 dBc and –54 dBc, respectively.

I tested for frequency accuracy by connecting the output of the RF generator to a 3-GHz universal counter that is sensitive enough to detect signals as low as –15 dBm. I was pleased with the results. To have the exact output frequency, I had to add the capability (in the firmware) of inputting (via the keypad) the actual frequency of the clock into the AD9851. A 1-Hz error assumed for this clock results in an output error of 280 Hz at 3 GHz and 140 Hz at 1.5 GHz. In retrospect, I should have used a higher-quality temperature-compensated part for the clock into the AD9851. The part I used required me to measure the clock more often than I wanted for an exact frequency output. With that said, once the actual AD9851 input clock was measured and input, the frequency was accurate to less than 4 Hz over the full range of frequency. I found that impressive!

FREQUENCIES BELOW 1.3 GHz

At this point, you might be saying: "Wait a minute! I thought you said the generator was capable of generating signals from 0 to 3 GHz. So far, I see only 1.3 to 3.2 GHz."

Look at Photo 1 to see how the lower frequencies are obtained. Two RF generators (one in an enclosure and one with only the PCB) are plugged into an off-the-shelf passive RF mixer (Mini-Circuits ZX05-43LH). One generator provides the RF input to the mixer and the other serves as the local oscillator. The third port is the intermediate frequency (IF) output, which contains, among other things, the difference frequency that is the desired signal. For example, if one generator is set at 2 GHz and the other is set at 2.3 GHz, the generated difference frequency will be 300 MHz.

Because I was now much lower in frequency, I was able to use a modern spectrum analyzer to look at the resulting spectrum. The output spectrum of the RF generator is shown in Photo 4. For comparison, you can see the output of an expensive signal generator, also generating a 300-MHz signal.



COMPETITION ECLIPSED

Imagine an industry first MCU capable of operating down to 0.9 V. Imagine solutions that are more power efficient than you thought possible. Imagine smaller form factors and lower system costs. Imagine best-in-class tools that enable immediate system design. Thanks to the remarkable engineering gains of our new line of MCU's, you can stop imagining and start creating your next great product.

WWW.SILABS.COM/POINT9

LOWEST VOLTAGE / GREATEST POWER EFFICIENCY
HIGHEST FUNCTIONAL DENSITY / BEST-IN-CLASS TOOLS



INTRODUCING THE C8051F9xx: THE FIRST LINE OF MCUs
DESIGNED TO OPERATE FROM A SINGLE-CELL BATTERY (0.9–3.6 V)

SILICON LABS

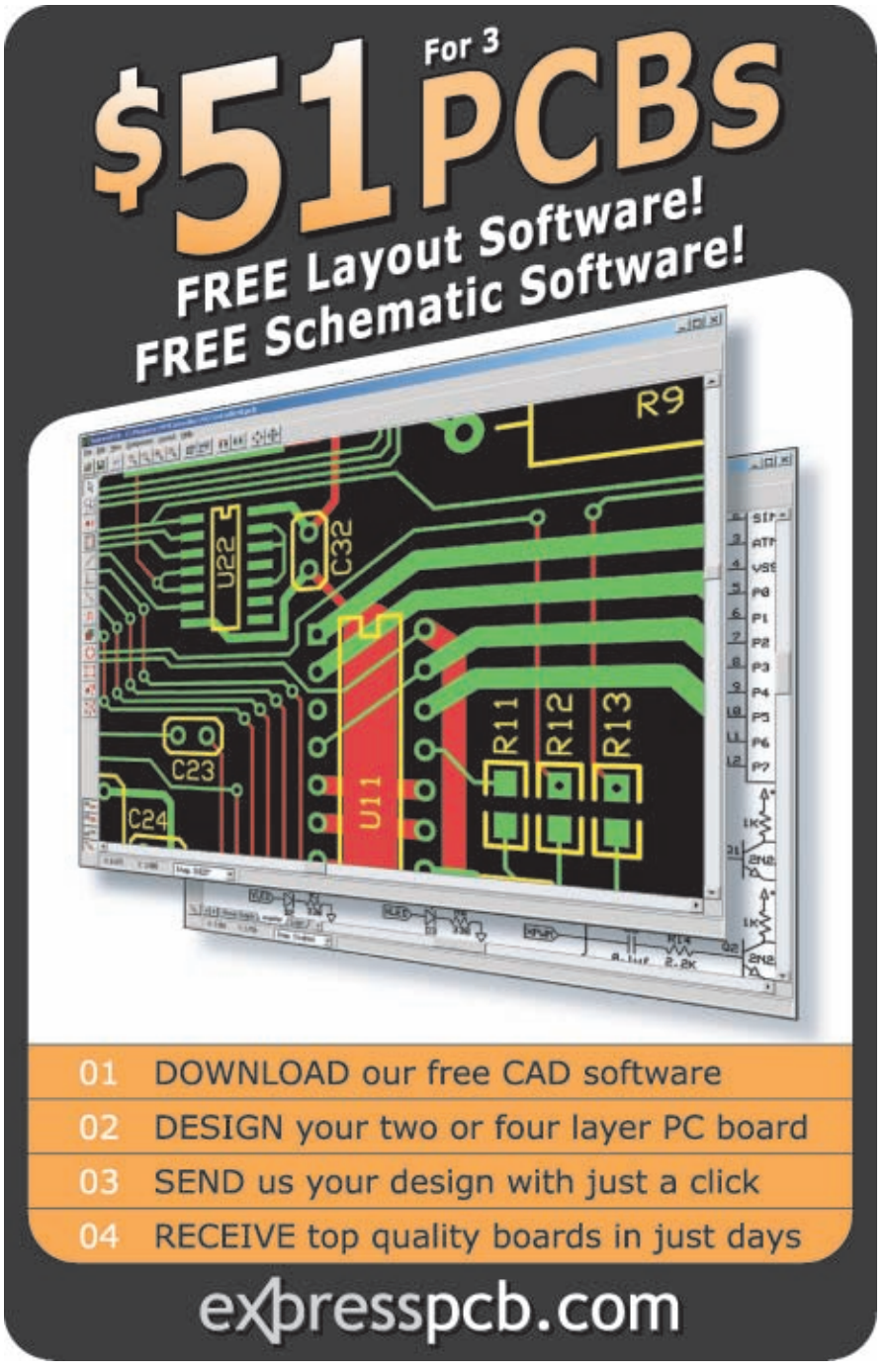
Photo 4 shows that the harmonic levels of the RF generator are not down as far as the expensive generator, but they are still relatively low. The second harmonic for the RF generator is down 45 dB. To put this in perspective, say you have a 10-dBm (0.71 V_{RMS}) fundamental. The unwanted second harmonic would be –35 dBm (0.004 V_{RMS}). That is a small distortion.

There are other frequency components in the RF generator's output spectrum. Unlike the previous example where the spectrum analyzer caused the extra spectral components, the components are real and result from the mixing process. Besides the normally expected sum and difference frequencies generated when mixing, output components are generated by the harmonics of the local oscillator and harmonics of the RF input signal. The location of the components is calculated by the equation $n \times F_{LO} \pm m \times F_{RF}$. Using this equation, Table 1 shows the location of these components for the 300-MHz signal that's generating. As you can see in Photo 4, the components are located exactly where the table predicts they should be. Again, the components are still down at least 40 dB from the fundamental, which is just fine in most cases. If you generate this 300-MHz signal by mixing a 2.7-GHz signal with a 3-GHz signal, none of the components of $n \times F_{LO} \pm m \times F_{RF}$ would be seen in the displayed spectrum because they would fall outside the frequency range displayed by the spectrum analyzer.

Note that there are also attenuated versions of the RF and local oscillator signals that leak through any mixing process. They are usually attenuated by greater than 30 dB and can be filtered out by low-pass filtering if your application requires it. They do not appear in Photo 4 because they fall outside the frequency range displayed by the spectrum analyzer.

Frequency accuracy and signal levels are important, but the quality of the output signal in terms of noise is more important. Photo 5 shows a closer frequency look at the fundamental for an expensive signal generator and my RF generator side by side.

At a 100-kHz offset from the fundamental center frequency, the noise level is comparable. But below 100 kHz, the expensive generator is less noisy. This really bothered me when I first observed it until I figured out why this was happening. It turns out that the source of this added noise in the RF generator is the clock source I used for the AD9851. The part was specified to have phase jitter less than 1 picosecond. I thought this was excellent at the time; but after precise measurements, I found that the clock's phase noise was –80 dBc at a 10-kHz offset. If you put that into the ADIsymPLL tool, the results shown in Photo 5 are exactly what is predicted. Using a part with a phase noise of –110 dBc would reduce the nose levels of the generated signal for offsets below 100 kHz. A part like this is not a simple drop-in replacement; however, because this kind of



clock chip comes with a 0.8-V_{pp} output, an additional chip is needed to convert to the CMOS level required by the AD9851. This is the work in progress that I mentioned at the beginning of the article.

WHAT'S NEXT

Besides adding a better AD9851 clock source, I intend to add FM modulation code to the generator's feature set. With a 30-MIPs processor, a 19.5-kHz PLL bandwidth, and a 3.3-MHz update rate for the DDS chip, there is plenty of horsepower to FM-modulate the signal.

Currently, I use off-the-shelf Mini-Circuits attenuators to vary the output signal level a fixed amount. They connect directly to the RF generator's output. But for more flexibility, I am also in the process of designing a separate PCB to accomplish output level control and AM modulation. In addition to having its own microprocessor and supply, it will use a variable-gain RF amplifier and RF log detector in a closed-loop system.

This project was fun. It caused me to learn new things and ultimately yielded a high-precision RF signal source. That's what keeps me coming back for more!

Neal Martini (nealmartini@cableone.net) holds an M.S.E.E. from the University of Missouri, Rolla. He is retired after 24 years of working for Hewlett-Packard in the LaserJet and InkJet printing businesses. In addition to being involved with a variety of boards, Neal works independently in product development in several application areas. In his spare time, he enjoys spending time with family, woodworking, racquetball, golf, and playing the piano.

PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/219.

RESOURCES

"L/C Impedance Matching Network Design Tool," www.hoflink.com/~mkozma/match19c.html.

P. Nelson, "PLL Combines With Digital Synthesis," *EE TIMES*, September 2003.

S. Sprowls, "Scotty's Modularized Spectrum Analyzer Web Site," www.scottyspectrumanalyzer.com.

SOURCES

AD9851 DDS Synthesizer, ADF4113HV charge pump, ADIsim-PLL software, and ADP3339 low-dropout regulator
Analog Devices, Inc.
www.analog.com

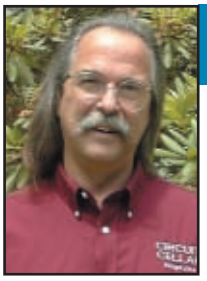
141T/8555A Spectrum analyzer
Hewlett-Packard
www.hp.com

dsPIC30F3010 DSC
Microchip Technology, Inc.
www.microchip.com

ZX05-43LH Frequency mixer
Mini-Circuits
www.minicircuits.com

V600ME10-LF VCO
Z-Communications, Inc.
www.zcomm.com





by Jeff Bachiochi

Get Started with PIC USB Connectivity

You can add USB to your next project with Microchip Technology's new array of parts. The 8-, 16-, and 32-bit microcontrollers will complement your designs with USB peripheral support, embedded hosting, and dual-role and OTG capabilities.

Many of us are just coming to grips with the idea of supporting a USB interface as the communications medium of choice for our product designs. PC manufactures have eliminated the DE9 (standard for years on every PC) and replaced it with USB (and FireWire) connectors. We've been forced to learn more about the protocols than we ever wanted to know. Unless my product incorporates PC-type architecture, I can't take advantage of all the peripherals out there. So, I can forget about adding a storage device, keyboard, or other USB peripheral to my project. Microcontrollers just don't have the room to handle a USB host's responsibilities.

USB On-The-Go (OTG) technology was announced on December 18, 2001, by the USB Implementers Forum (USB-IF). Let's see, that's about seven years ago. You'd think by now that we would have access to all of those USB products that have shipped to date—at least that was the promise made back then. About all that comes to mind is some high-priced digital camera that supports downloading directly into a printer. That's hardly what I would call utopian connectivity.

Well, stand back, because the tools to make this a reality are finally here. Microchip Technology has released 16- and 32-bit microcontrollers that support embedded hosting, dual-role, and OTG capabilities, complementing the original 8-bit microcontrollers with USB peripheral support (see Photo 1). Let's look a little closer at the supported USB capabilities.

USB (AS WE KNEW IT)

We know the PC as a USB host device consisting of complex software applications with an operating system and processor. The host must manage and control all USB functions. It initiates all data packet activity on the bus, managing data flow and doing error checking. The host must detect peripheral insertion (enumerate the device and match it with a particular class driver) and extraction (close and clean up after itself). It also provides power to the peripheral. (For more information about USB and its inner workings, refer to my columns listed at the end of this article.)

A USB connection is made via a cable with differentiated plugs (see Photo 2). The host is mated using a type-A plug/receptacle while the peripheral is connected with a type-B



Photo 1—If you need a full-speed USB 2.0 device, an embedded host, or dual-role and OTG solutions, you're in the right place. These should work for you.

plug/receptacle. Some peripherals have a micro type-B plug/receptacle to save space.

A USB peripheral responds only to a host and does not have the capability to initiate data transfers. It must be concerned with power consumption and check for errors in the data from the host.

EMBEDDED HOST

An embedded host is defined as a product with limited host capabilities. That is, while it must provide management and control of the bus, it can support only the peripherals that it was designed for. This can be as specific as a peripheral or class of peripherals. An embedded host will have a full-size type-A receptacle. The embedded host may optionally support SRP. (I'll present more on the power conservation session request protocol a bit later.)



Photo 2—USB type-A connectors are used upstream for connection to host, hub, and embedded host devices. USB type-B connectors are used downstream for connection from hubs, dual-role embedded host (as peripheral), and peripheral devices. USB type-AB connectors are the only connectors used for OTG devices.

ON THE GO

OTG technology has its own limitations and extensions. An OTG device has a dual-role capability: it can act as an embedded host or a peripheral. It is designated by the use of one micro type-AB plug/receptacle. The receptacle can accept either a micro type-A plug or a type-B plug. OTG devices must support both HNP (Am I the host or peripheral negotiation protocol?) and SRP.

DUAL ROLE

This term can be confusing depending on how it is used. An OTG device can be considered a dual-role device. A dual-role device needs both a standard A connector and one of the B-type connectors. However, there are applications that might require an embedded host device to serve as a host to one peripheral and as a peripheral to another host.

NEW PROTOCOLS

The ability to host a connection brings with it a new complexity for the interface circuitry as the device's role changes from that of a peripheral, and that includes new protocols (see Figure 1). The HNP protocol enables the OTG devices to negotiate who will be the boss when the two are connected. Let's assume you have an OTG camera and an OTG PDA. An OTG cable (micro-type-A to micro-type-B) connects the two. Each device can be a host to, say, a printer. When connected together, the cable determines which device will begin as the host.

This OTG cable has a fifth connection

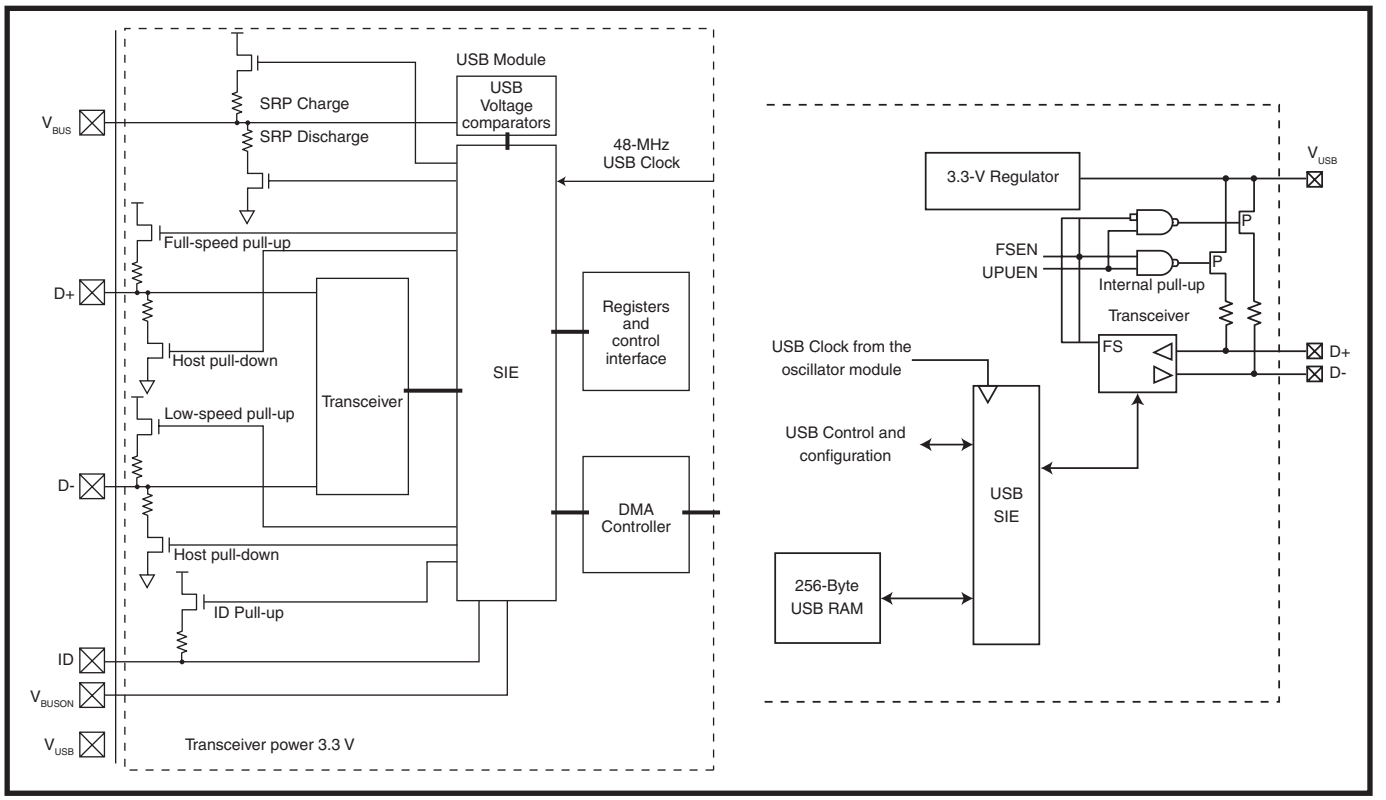


Figure 1—Peripheral devices need simple USB interfacing (right). Embedded host and OTG devices require a more sophisticated arrangement to tackle both up- and downstream connections (left).

with pin 5 connected to ground on the type-B side. Ra_PLUG_ID is 10 Ω max and Rb_PLUG_ID is 100 kΩ min. The type-A side becomes the host and must supply power, at least 8 mA, to the type-B peripheral. The host will identify its peripheral as an OTG device and can allow the peripheral to ask for a swap in functionality. (You may remember that a peripheral can indicate its operating speed as full by attaching a pull-up on the D+ line. New interface circuitry enables a device to enable/disable functions so it can act as a host or a peripheral.) Once the device at the type-B end of the cable takes over, it can relinquish control back to the type-A side, if necessary, by the same procedure. With this approach, you never have to worry about which end of the cable to plug into which OTG device. It's all automatic.

One of the main concerns with embedded hosts and OTG devices is power consumption because most are battery powered. The second new protocol SRP addresses this concern. OTG hosts are allowed to turn off V_{BUS} in the USB connection when the bus is idle. Even with no V_{BUS} on the USB connection, the peripheral device can ask for service when necessary. This is done by first toggling D+ with its pull-up. If the host monitors the data lines, it can respond by turning on V_{BUS} and initiating a session. In this case, the peripheral will see V_{BUS} and abort its SRP. If the host doesn't respond to the data toggling, the peripheral toggles V_{BUS} (8 mA of current max). If the host monitors V_{BUS}, it can respond by turning on V_{BUS} and initiating a session. OTG devices are the only ones required to use SRP, although it can be (optionally) supported by any device. Detailed signal information for HNP and SRP can be found at www.usb.org/developers/docs/USB_OTG_1-3.pdf.

TPL

Although an OTG device must support some host qualities, it is limited to a specific set of support peripherals designated by the designer. OTG devices require a targeted peripheral list (TPL) to decide whether a connected peripheral is supported or not.

The TPL requires a manufacturer, model number, and description of those peripherals it can handle. If the peripheral is not in the TPL, the OTG device is required to indicate the problem to the user in some fashion, preferably in an error message (e.g., an LED or a beep). Class support is not part of OTG. Although there is no reason it couldn't be used, that would void OTG certification. (This should be revisited by the USB-IF.)

USB-IF Class specifications	Proprietary protocols (potential)
HID	Bluetooth HCI
Hub	MS RNDIS
Printer	USB-to-Serial
Digital image capture	USB-to-Ethernet
Audio	MP3 Players
Mass storage (MSD)	Still digital cameras
Communications (CDC)	Webcams
USB Chip/smart card	Video capture devices
IrDA	Host-to-host cables
USB Monitor control	
Video	

Table 1—The left column lists functions covered by standard class support. The right column lists functions that may require special support beyond the standard class.

CLASS SUPPORT

The USB-IF standard classes supported for an embedded host are listed in Table 1. With memory at a premium for embedded devices, supporting every class would be virtually impossible. Designing for a class minimum makes sense. This at least enables your product to continue functioning with any manufacturer's widget of that class.

A BIRD IN THE HAND

There is something about having a product in your hands to assure you it's real. But that's just the start. Microchip Technology's parts get you thinking about how completely they support USB. The 8-bit PIC18F14K50 (20-pin), PIC18F4550 (40-pin), and PIC18F66J50 (64-pin) function as USB peripherals. The 16-bit PIC24FJ256GB110 (100-pin) supports USB embedded hosting, peripheral, and OTG functions. The 32-bit PIC32MX460F512L (100-pin) also supports USB embedded hosting, peripheral, and OTG functions.

While these parts just hint at all of the USB variants available, they demonstrate the extensive products supported that use a single (free) integrated development environment. Common software libraries provide an easy migration path across all devices. Each of the parts has a development, demo, or starter kit available for under $60. Let's take a quick look at these offerings.

COMMON FEATURES

The chips I'm covering feature on-chip transceivers and are USB 2.0-compliant. They support low-speed (LS, 1.5 Mbps) and full-speed (FS, 12 Mbps) connections and all four types of transfer: control, interrupt, isochronous, and bulk. Table 2 compares these side by side.

The 8-bit chips can source/sink 25 mA through their I/O pins, while the 16- and 32-bit chips can handle 18 mA. All of the chips support dedicated edge-triggered interrupts as well as interrupt-on-change (with internal pull-ups) on some or all of their I/O pins.

In terms of on-chip peripherals, each chip includes one or more capture/compare/PWM modules, some with enhanced PWM features. The enhanced PWM features include dual- (half-bridge) and quad-PWM (full-bridge) outputs with selectable polarity, dead time, and auto shutdown/restart. The master synchronous serial port (MSSP) module supports four modes of SPI and both master and slave I^2C modes. An enhanced USART supports RS-232/485 as well as LIN 1.2/2.0 and IrDA interfaces in some cases. The 10-bit ADC module has automatic acquisition and conversion during sleep capability. Analog comparators offer rail-to-rail operation and independent input

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

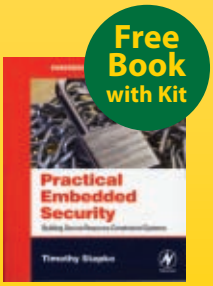
RCM4300 RabbitCore®

- **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

multiplexing.

Each microcontroller's external crystal/clock input can run the oscillator up to 48 MHz (32 MHz on the PIC24FJ256GB110). A 4× phase-locked loop (PLL) can boost input when necessary for the USB. The system clock can come from the USB block, one of two alternate internal RC oscillators, or the optional low-frequency RTC clock. (Timer1 uses an external 32.768-kHz crystal.) There is even a failsafe clock monitor that allows for shutdown in the event of an oscillator failure. There are also an RTC oscillator and a watchdog timer that operate independently of the CPU. The watchdog timer runs from the 31-kHz internal RC oscillator source.

Each chip includes an internal regulator that can operate the core logic. Power is managed through three or more states: Run, Idle, and Sleep. Brown-out reset (BOR) and power-on reset (POR) functions are implemented as well as optional two-speed start-up that allows execution from the internal oscillator. The oscillator start-up timer (OST) waits for the mandatory 1,024-cycle delay of the external oscillator.

PIC18F14K50

The 8-bit PIC18F14K50 is the smallest member of the group with a 20-pin package (PDIP, SSOP, or SOIC). It is geared toward low-cost, tiny devices that need USB connectivity. It has some special features to support such devices, including interrupt on change for D± (detect connection). The slew rate on each output port can be reduced as an aid to EMI reduction. A 1.024-V fixed internal reference for the ADC is available.

This part has a 3.6-V internal voltage regulator derived from a 5.5-V input. The LF version operates from a 3.6-V maximum input voltage and does not have the regulator. Both versions can run down to 1.8 V (without the internal regulator).

The low-pin-count development kit (DV164126) was created with the USB novice in mind. It includes a self-directed course and lab material designed to ease the learning curve associated with adding USB connectivity to embedded systems. The kit contains two development boards, one fully populated and one unpopulated spare development board, one PIC18F14K50 debug header, a CD containing the user guide, course materials, and product documentation. In addition, a PICkit 2 (and USB cable) is included for debugging and programming. The PIC18F14K50 costs $2.50 per unit and the demo board costs $60.

PIC18F66J50

The 8-bit PIC18F66J50 is for cost-sensitive applications that need more

Feature	PIC18F14K50	PIC18F66J50	PIC18F4550	PIC24FJ256GB110	PIC32MX460F512L
USB Modes	Device	Device	Device	Device, OTG, embedded host	Device, OTG, embedded host
USB Endpoints	16	32	32	32	32
USB Dual-port RAM	256 bytes	3.9 KB	1 KB	DMA	DMA
Power management states (maximum speed)	Run (12 mA), Idle (5 mA), Sleep (0.7 µA)	Run (15 mA), Idle (6.5 mA), Sleep (1.4 µA)	Run (40 mA), Idle (18 mA), Sleep (0.1 µA)	Run (24 mA), Idle (5.6 mA), Doze, Sleep (1 µA)	Run (75 mA), Idle (30 mA), Sleep (28 µA)
RTC Current	1.3 µA	24 µA	4.5 µA	7 µA	10 µA
WDT Current	3.6 µA	7 µA	3.8 µA	3 µA	10 µA
I/O Pin current	25 mA	25 mA	25 mA	18 mA	18 mA
I/O Interrupts	Three edge-triggered, seven change-of-state	Four edge-triggered, four change-of-state	Three edge-triggered, four change-of-state	Five edge-triggered, all change-of-state	Five edge-triggered, all change-of-state
CCP with PWM	One enhanced	Two standard, three enhanced	One standard, one enhanced	Nine standard	Five standard
MSSP	One	Two	One	Three	Two
Enhanced USART	One RS-232/485 LIN 2.0	Two RS-232/485 LIN 1.2	One RS-232/485 LIN 2.0	Four RS-232/485 LIN, IrDA	Four RS-232/485 LIN 1.2, IrDA
ADC	10 bits, nine channels	10 bits, eight channels, 100 ksps	10 bits, 13 channels, 100 ksps	10 bits, 16 channels, 500 ksps	10 bits, 16 channels, 500 ksps
Analog comparator	Two	Two	Two	Three	Two
Parallel port	—	PMP	SPP	PMP/PSP	PMP/PSP
Maximum CPU clock	48 MHz, 12 MIPS	48 MHz, 12 MIPS	48 MHz, 12 MIPS	32 MHz, 16 MIPS	80 MHz, 80 MIPS
Internal RC oscillators	31 kHz, 16 MHz	31 kHz, 8 MHz	31 kHz, 8 MHz	31 kHz, 8 MHz	32 kHz, 8 MHz
Hardware multiplier	8 × 8, one cycle	8 × 8, one cycle	8 × 8, one cycle	17 × 17, one cycle	32-bit Multiply/divide unit
Interrupt priority	Two-level	Two-level	Two-level	Multilevel	Multilevel
Core voltage	1.8 to 3.6 V, 3.2-V Regulator	2 to 3.6 V, 2.5-V Regulator	3 to 5.5 V, 3.3-V Regulator	2.3 to 3.6 V	2.3 to 3.6 V
Packaging	20-pin PDIP, 20-pin SSOP, 20-pin SIOC	64-pin TQFP	40-pin PDIP, 44-pin TQFP, 44-pin QFN	100-pin TQFP	100-pin TQFP
Other	USB Interrupt on change, 1.024-V internal reference, I/O slew rate control	—	HLVD	CTMU, PPS, five timers, CRC, 32-/16-bit divide	256-Byte prefetch, four DMA channels

Table 2—These five chips support a wide range of USB-enabled applications, from the low-pin-count PIC18F14K50 to the high-performance PIC32MX460F512L. This table shows some of the key differences among them.



Educators and Students:

Register Today

to Receive Exciting Microcontroller Resources!

RENESAS UNIVERSITY PROGRAM

Renesas — the #1 supplier of microcontrollers in the world — is launching Renesas University, an exciting educational program that gives educators a way to teach microcontroller (MCU) technology using a modern architecture and professional-grade tools. It also offers many valuable resources that help students learn about MCUs and how they can be applied in significant embedded system designs.

#1 MCU REACH FURTHER

Renesas is a worldwide leader in:

- Microcontrollers
- Embedded flash microcontrollers
- MCUs in car navigation systems
- Power amplifiers for GSM phones
- LCD controllers for color mobile displays

The Renesas University program nurtures an online community where educators and students come together to share ideas, address technical issues and discuss microcontroller topics. It is characterized by:

Publish ▸ Renesas actively encourages academics and students to publish microcontroller-related papers. We provide assistance in publishing course material and microcontroller related books.

Toolchain ▸ The Renesas integrated development environment with toolchain is the commercial version of our development tools – with full C compiler, assembler, linker, and debugger. It is not a typical capability-reduced "educational" version. The only limitation is a 64KB code size after 60 days of use.

Modern ▸ Renesas microcontrollers utilize a modern architecture designed specifically for C and other high-level languages. Our devices handle the most demanding applications of today and tomorrow.

Complete Development Kits



Renesas Starter Kits provide a USB-powered, MCU-based system board with in-circuit debugger/flash memory programmer. They include a CD containing our integrated development environment with toolchain, plus documentation, example firmware and interesting projects.

- **Free for Educators:** Register at the Renesas University website to receive ten free Starter Kits per semester. In return, we request the submission of material that enriches Renesas University; i.e., code, student projects, technical papers, embedded control designs, etc.
- **Low cost for Students:** If actively enrolled in an educational institution, a Starter Kit can be purchased at a very low cost after registering at the Renesas University website.

Visit us at ESC Boston!
Booth #1008
Embedded Systems CONFERENCE BOSTON

For more information on Renesas University and how to enroll, please visit
www.renesasuniversity.com or email: University@rta.renesas.com



© 2008 Renesas Technology America, Inc. Renesas Technology America, Inc. is a wholly owned subsidiary of Renesas Technology Corp.

Everywhere you imagine. RENESAS



Photo 3—The PIC24F starter kit contains an on-board debugger, USB connectors (host and peripheral), a tricolor LED, a capacitive touch pad, and an OLED display. It comes preprogrammed with a menu-driven demo supporting data logging, a thumb drive, and graphics.

I/O support. It comes in a 64-pin TQFP package and includes an 8-bit parallel master port/enhanced parallel slave port for communicating with external devices like QVGA and memory.

This part has a 2.5-V internal voltage regulator derived from a maximum 3.6-V input (it has 5.5-V tolerant digital I/Os) and can run down to 2 V (without the internal regulator).

The PIC18F87J50 FS USB plug-in module (PIM) is a full-speed USB demonstration board (MA180021). While the PIM is made for use with the HPC explorer board, it can be used on its own. The PIM comes with preinstalled application software that demonstrates how to use a PIC18F87J50 as a HID peripheral mouse. You can use your ICD2 or PICkit 2 to debug and program the microcontroller on this board. The PIM ships with a CD-ROM containing example USB projects that can be used directly with the board. The PIC18F66J50 costs $3.50 per unit and the demo board costs $40.

PIC18F4550

The 8-bit PIC18F4550 is the oldest member of this group. A streaming parallel port (SPP) can be used in conjunction with the USB port for rapid data movement to and from an external device.

The PIC18F4550 operates from a power supply of 3 to 5.5 V. There is an –LF version that operates down to 2 V. In addition to BOR and POR, this chip includes high/low voltage detect (HLVD) functions.

The PICDEM FS_USB demonstration board (DV163025) isn't new. In fact, I've had mine for years. While the board comes preloaded with code to access a temperature sensor, a potentiometer, LEDs, and switches via a PC application, it was the bootloader that I found most useful. The onboard serial port is also helpful in making the jump from a serial RS-232 connection to USB. You can use your ICD2 to debug and program the microcontroller on this board. However, the PIC18F4550 can be reprogrammed in circuit without an external programmer when using a bootloader. The PIC18F4550 costs $4.50 per unit and the demo board costs $60.

PIC24FJ256GB110

The 16-bit PIC24FJ256GB110 is the least expensive member of the group that offers OTG capability. It has several useful peripheral functions that don't appear on the other chips. A CTMU supports capacitive touch applications. With peripheral pin select, you can remap the I/Os of many peripherals in real time. It also has a hardware RTCC and a programmable cyclic redundancy check (CRC) generator. External devices like QVGA and memory can be controlled through the 8-bit parallel master/slave port (PMP/PSP).

The PIC24F starter kit (DM240011) contains everything needed to begin exploring the high performance and versatility of the PIC24F microcontroller family (see Photo 3). This inexpensive kit includes an integrated in-circuit debugger and programmer, a USB device and host connectors, a tricolor LED, a capacitive touch pad, and an OLED display. Menu-driven demonstration software supports data logging, a thumb drive, and graphics applications to test the PIC24F microcontroller.

The PCB includes integrated hardware debug and programmer circuitry to develop, program, and test applications on the board's PIC24F microcontroller. A USB connection to a host computer supplies communications and power to the board. No additional external power supply is needed. The kit comes with preinstalled application software that demonstrates the interactive, menu-driven display using a parallel master port (PMP), capacitive touch sensing with the charge time measurement unit (CTMU), time and data display using the real-time clock and calendar (RTCC), RGB LED control with three PWMs and peripheral pin select (PPS), a USB flash drive interface (USB embedded host), real-time data graphing using the ADC, display multitasking, and real-time data capture using multitasking (USB embedded host).

The kit contains a PIC24F board with an integrated debugger, a USB cable, and the MPLAB starter kit CD containing an MPLAB IDE, an MPLAB C compiler for PIC24 MCUs, application notes, and sample programs. The PIC24FJ256GB110 costs $6 per unit and the demo board costs $60.

PIC32MX460F512L

The high-performance 32-bit PIC32MX460F512L is for applications



Photo 4—The PIC32 USB starter kit contains an on-board debugger, USB connectors (host and OTG), an expansion header (for PIC32 I/O), and a web address for downloading all documentation.

a)	Technical briefs
TB054	An Introduction to USB Descriptors (with a gameport-to-USB gamepad translator example)
TB055	PS/2-to-USB Mouse Translator
TB056	Demonstrating the *set_report* request (with PS/2-to-USB keyboard translator example)
TB057	USB Combination Devices (demonstrated by a combination mouse and gamepad device)
TB058	Demonstrating the Soft Detach Function (with a PS/2-to-USB translator example)

b)	Application notes (general)
AN258	Low-cost USB Microcontroller Programmer (The building of the PICkit 1 Flash Starter kit)
AN950	Power Management for PIC18 USB Microcontrollers with nanoWatt Technology
AN956	Migrating Applications to USB from RS-232 UART with Minimal Impact on PC Software
AN1045	Implementing File I/O Functions Using Microchip's Memory Disk Drive File System Library
AN1148	Cyclic Redundancy Code (CRC)
AN1149	Design a Load-Sharing System Power Path Management with Microchip's Stand-Alone Li-ion Battery Charger

c)	Application notes (PIC18)
AN1189	Implementing a Mass Storage Device Using the Microchip USB Device Firmware Framework

d)	Application notes (PIC24)
AN1003	USB Mass Storage Using a PIC MCU
AN1140	USB Embedded Host Stack
AN1141	USB Embedded Host Stack Programmer's Guide
AN1142	USB Mass Storage Class on an Embedded Host
AN1144	USB Human Interface Device Class on an Embedded Host
AN1145	Using a USB Flash Drive on an Embedded Host
AN1189	Implementing a Mass Storage Device Using the Microchip USB Device Firmware Framework

e)	Application notes (PIC32)
AN1140	USB Embedded Host Stack
AN1141	USB Embedded Host Stack Programmer's Guide
AN1142	USB Mass Storage Class on an Embedded Host
AN1145	Using a USB Flash Drive on an Embedded Host
AN1163	USB HID Class on an Embedded Device
AN1164	USB CDC Class on an Embedded Device
AN1166	USB Generic Function on an Embedded Device
AN1169	USB Mass Storage Class on an Embedded Device
AN1176	USB Device Stack for PIC32 Programmer's Guide

Table 3—Here's a list of the USB technical briefs and application notes presently on the Microchip web site.

that need a lot of local computing power in addition to the USB OTG capability and other peripheral functions. It also has a hardware RTCC. External devices like QVGA and memory can be controlled through the 16-bit parallel master/slave port (PMP/PSP).

The PIC32MX460F512L has a 256-byte prefetch module that speeds the execution of the execution unit (EU) while delivering the fastest context switch and interrupt response possible. There are four channels of DMA for remarkable throughput without EU intervention. Atomic bit manipulation provides separate registers for each I/O register to clear, set, and invert any bits of that I/O register.

The PIC32 USB starter board (DM320003) provides the easiest and lowest-cost method to experience the USB OTG family of PIC32 microcontrollers (see Photo 4). You can develop USB embedded host, device, dual-role, or OTG applications by combining this board with Microchip's free USB software. The USB starter board has an expansion connector and can be used with the PIC32 I/O expansion board and PICtail Plus daughter cards. The starter board comes with a standard type-A to micro-type-B cable for the built-in debugger, a standard type-A to micro type-B cable for USB application development, and a quick-start card directing users to web-based instructions for software download and installation. No software is provided in the box.

The free 32-bit software resources available include USB host and device stacks, a graphics and audio library, a TCP/IP stack with SSL, 16- and 32-bit file system support, and CAN software. The PIC32MX460F512L costs

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/CC312409

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







$7.50 and the demo board costs $50.

SUPPORT

Obviously, just offering the OTG hardware in a microcontroller to handle a USB host is not a total solution. Originally intended as a replacement for serial and parallel connections on a PC to connect mice, keyboards, and printers, USB has grown to be the standard interface between a PC and its many and varied peripherals. With the addition of an embedded host and OTG, the interface has expanded to any product that would benefit from its features, like audio devices (digital file sharing, audio and MP3 connectivity), handheld scanners and meters (download data), and industrial control and medical monitoring (upgrade software, download diagnostics for analysis). To help you integrate USB into your product, Microchip offers a three-step USB solution. First, it offers a wide variety of 8-, 16-, and 32-bit USB MCUs for basic, low-cost applications to complex and highly integrated systems. Second, Microchip offers free license software libraries (including source code). Finally, they have a variety of low-cost and full-featured development tools and a free IDE, so you can quickly integrate USB functions into your existing application using a single microcontroller, reducing total system cost, development risk, and time to market.



Photo 5—The PIC18F USB starter kit contains two (one populated) low-pin-count development boards, a PIC18F14K50 debug header, and a CD containing the user guide, course materials, and product documentation.

Table 3 lists USB-related technical briefs and application notes presently found on the Microchip web site. Microchip's "MCHPFSUSB Framework v2.2" and "USB Device and Embedded Host Stack for PIC32 v1.03"—which cover PIC18/24 and PIC32 devices—are two files that contain a variety of USB-related firmware projects, drivers, and other resources. Microchip also has Web-Seminars that you can view on the site. Archived USB-related broadcasts include "Emulating RS-232 Over USB Using the PIC18F4550" and "Tutorial for the MPLAB Starter Kit for PIC24F."

USB-IF

To use the USB logo, a product must pass compliance testing as defined by the USB-IF. For instance, OTG-compliance testing includes the physical inspection of the microcontroller-AB receptacle and user interface display, signal quality as a host and as a device, OTG electrical test (OET), OTG protocol test (OPT), interoperability with all devices listed on the targeted peripheral list, and submission of a completed USB OTG compliance checklist. The compliance tests ensure that your product plays well with others. Visit www.usb.org for more information about compliance testing.

HORIZONTAL MARKET STRATEGIES

Up to this point, "plays well with others" has meant the PC. Although there will still be many products that must connect to a PC, OTG devices may never need to connect to a PC. They may never have to deal with a standard or custom Microsoft driver. For instance, the camera/printer OTG pairing makes a happy couple and requires no PC intervention.

The temporary and permanent

storage and exchange of audio, video, and data files is thriving thanks to solid-state storage media. By far, the most popular function for USB is the moving of data. USB has made the thumb drive common place. I see them around necks, on key chains, and hanging up in groups waiting their turn in a rotational game of daily system backups. A product that supports MSD may never need to connect to anything other than a thumb drive.

One of the features that flash memory microcontrollers have brought to products is the ability to update their own operating systems. Back in *Circuit Cellar* 200, I used a USB MSD device in a logger for my electric bicycle project ("Embedded USB Breakthrough," 2007). When a thumb drive was plugged in, the files were searched for a particular file name. If it found a file with that name, it used the bootloader to read in the file (a new operating system) and reboot. This, however, is cause for the USB-IF to sit up and take notice because this may violate compliance. The USB-IF states that if any changes are made to the product, creating "significant differences" between the production product and the product samples tested, retesting is required. They state that as long as the PCB hasn't changed and the USB function code remains unchanged, retesting is not required.

I'm predicting that the sensor market will take up USB with open arms. Up to this point, USB peripheral developers have been wearing blinders. Products were developed only if the PC could make use of it. With full support for embedded hosting and OTG, I think we will see smart sensors being developed for embedded hosts. Smart sensors provide their hosts with the information necessary to make use of the sensor without any manual configuration.

Microchip has readied itself for the USB onslaught. It can provide you with the parts, tools, and information to make it happen (see Photo 5).

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 www.imaginethatnow.com.*

RESOURCES

J. Bachiochi, "Accessing the USB Framework: A Quick Transition from UART to USB," *Circuit Cellar* 195, 2006.

———, "Create a USB Hybrid Hub," *Circuit Cellar* 170, 2004.

———, "Embedded USB Breakthrough," *Circuit Cellar* 200, 2007.

———, "USB DMX," *Circuit Cellar* 172, 2004.

———, "USB in Embedded Design," Parts 1 and 2, *Circuit Cellar* 165 and 166, 2004.

Microchip Technology, Inc., "PIC18F87J50 Family Data Sheet: 64/80-Pin High-Performance, 1-Mbit Flash USB Microcontrollers with nanoWatt Technology," DS39775B, 2007.

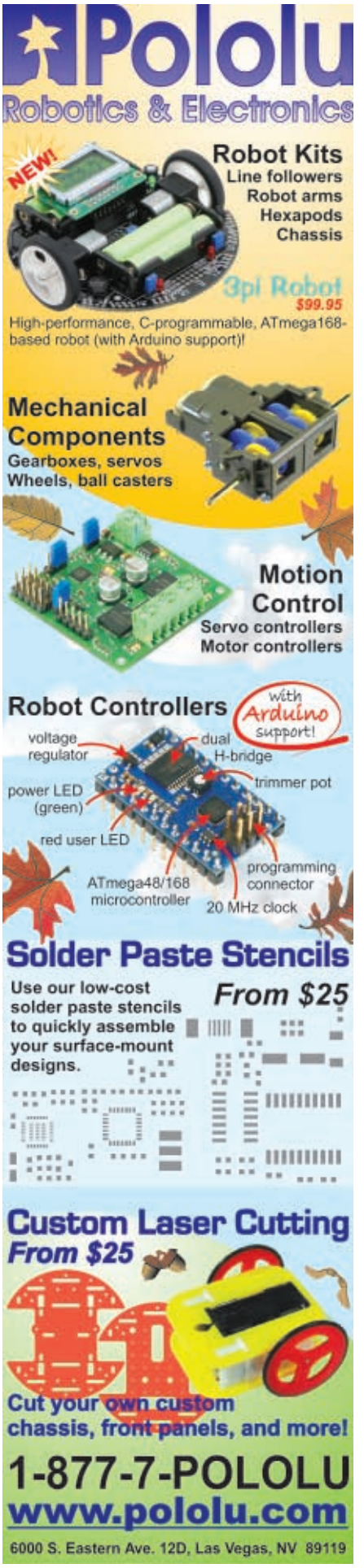
———, "PIC32MX3XX/4XX Family Data Sheet: 64/100-Pin General Purpose and USB 32-Bit Flash Microcontrollers," DS61143D, 2008.

K. Otten, "Tutorial for MPLAB Starter Kit for PIC24F," Microchip Technology, Inc., 2008, http://techtrain.microchip.com/webseminars/documents/PIC24FStrKit_031708.pdf.

R. Rojvanit, "Emulating RS-232 over USB with PIC18F4550," Microchip Technology, 2004, http://techtrain.microchip.com/webseminars/documents/EmulatingRS-232overUSB_121004.pdf.

SOURCE

PIC18F14K50, PIC18F4550, PIC18F66J50, PIC18F87J50, PIC24FJ256GB110, and PIC32MX460F512L Microcontrollers, DM164127 development kit, DM240011 starter kit, DM320003 starter board, DV163025 demonstration board, and PICkit 2 development programmer/debugger
Microchip Technology, Inc.
www.microchip.com





by Ed Nisley

HT Audio, GPS for APRS, and What Works

Several years ago, Ed built a system that enabled him to communicate with a fellow rider via an electret microphone connected to an amateur radio during bicycle rides. This month, he explains how he rebuilt the interface and combined it with a GPS tracking system.

Seven years ago, I wrote about electret microphones and the circuitry that interfaced one with the amateur radio mounted on my recumbent bicycle. That circuitry let my wife and me talk during family rides and provide safety communications for public-service events, but the years have taken their toll on the hardware.

I decided to rebuild the radio interface and combine it with a GPS tracker, so that event organizers can monitor our exact location along with our status reports. Photo 1 shows the result: an Icom amateur radio (under the Byonics GPS2 receiver puck), my audio interface board, and a Byonics TinyTrak3+ GPS tracker. As you'll see, the competing demands of an electret microphone, a data modem, and a handheld radio pose some interesting challenges.

Engineers know that a good design requires more than just a schematic and a circuit board layout. I've discovered just how efficiently a bicycle can destroy mechanical and electronic devices, so I'll also share some observations on component reliability. If you think a connector is just a connector, read on!

RADIO INTERFACES

Bob Bruninga developed the Automatic Packet Reporting System (APRS) in the early 1990s to provide a digital link between amateur radio stations. It's become a global information system that can exchange tactical information, report telemetry values, and control devices. A GPS receiver can feed its position into the APRS network for immediate tracking.

APRS stations normally transmit and receive digital data packets on 144.39 MHz, but most of our Mid-Hudson Valley public service events use voice simplex frequencies near 146.50 MHz or a voice repeater for wide-area events. I didn't want to haul two radios, antennas, and batteries around on my bike, which meant I had to combine analog voice and digital data packets for transmission on a single radio, then separate them at the receiver.

Fortunately, I didn't have to invent the entire system. Several companies supply trackers that accept standard NMEA serial data from a GPS receiver and encode APRS data packets for transmission through a voice radio. I used the TinyTrack3+ interface shown in Photo 1 along with the GPS2 puck-style GPS receiver. The PIC microcontroller on the TT3+ board stores APRS routing paths, transmitter timing values, and other fixed information, so that mobile operation requires just a source of DC power.

The TT3+ can monitor the voice radio's push-to-talk (PTT) signal and send its data immediately after a voice transmission. It also monitors the receiver audio and won't send data when it hears other activity. As a result, its packets rarely collide with transmitted or received audio.

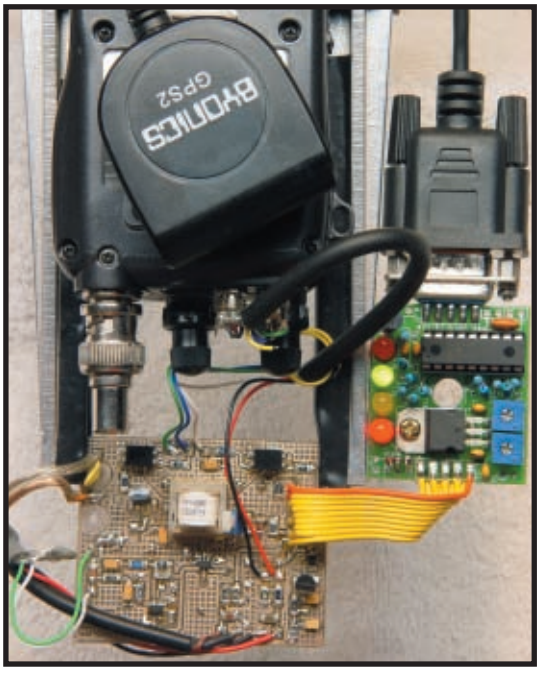


Photo 1—The audio interface board just below the radio combines analog voice with GPS modem data and attenuates the radio's speaker-level audio to a comfortable earphone level. The board will eventually fit in a case that replaces the radio's 650-mAh NiCd battery with a cable to an external battery pack constructed from 2,600-mAh AA NiMH cells.

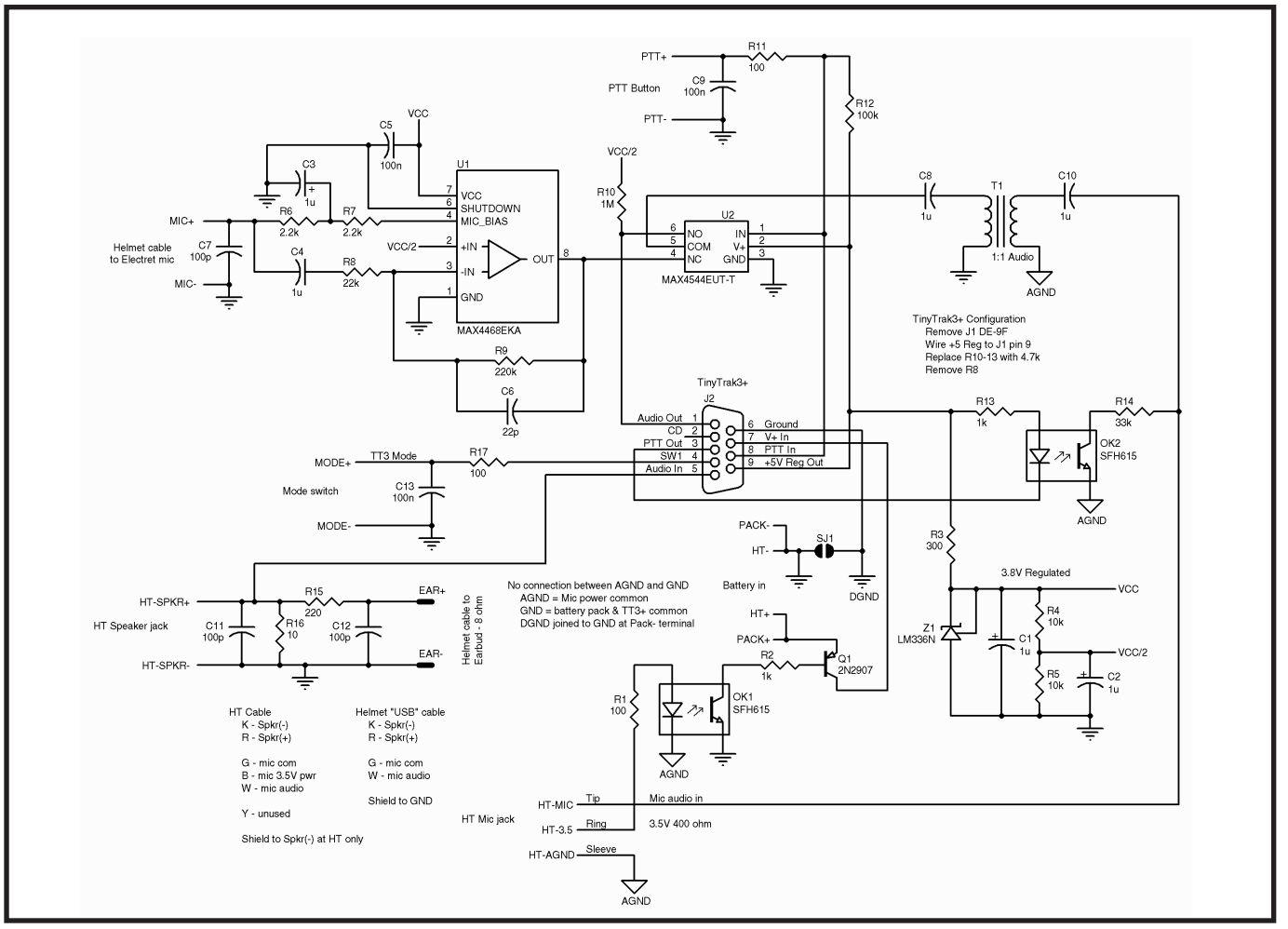


Figure 1—The audio transformer and optocouplers break a galvanic loop that would otherwise link the two radio jacks through the board. The Maxim Integrated Products MAX4544 analog switch isolates the microphone and the Byonics TinyTrak3+ audio from each other, while the LM336 isolates the MAX4468's supply from the TT3+ microcontroller's digital noise. Optocouplers OK1 and Q1 activate the TinyTrak3+ when the radio power is on.

At the event control station, a packet radio modem extracts the digital data from the radio receiver's audio output. A laptop running XASTIR, a free software program that integrates GPS position data with the APRS network, then displays our location on a map of the area.

On solo rides, when I'm not interested in talking or breathing too hard to chat, I can switch the radio to 144.39 MHz and transmit GPS tracking data directly into the existing APRS network covering the Mid-Hudson Valley.

Our Icom IC-Z1A and IC-W32A radios have the standard Icom "external interface" consisting of a 3.5-mm external speaker jack and a 2.5-mm microphone jack. Although the dimensions don't quite work out, they were formerly known as 0.125" and 0.1" subminiature phone plugs and jacks.

According to the radio interface schematics, the sleeve terminals of both jacks connect to the radio's common voltage, which is also the battery's negative terminal. The 3.5-mm ring terminal is a bidirectional serial data interface for radio control, although that seems to be unused on the Z1A. The 2.5-mm ring terminal supplies 3.5 V through a 400-Ω resistor for electret microphone bias. The tip terminals connect to the external speaker and microphone, respectively.

AUDIO ROUTING

The schematic in Figure 1 has three main sections: the electret microphone amplifier, the TinyTrak3+ interface, and power control. I put SMD capacitors on all of the inputs to keep RF out of the



Photo 2—This washable push button switch has worked reliably for seven years in an environment where open-frame switches fail within months. Hot-melt glue secures it to the handlebar and seals the cable connection.

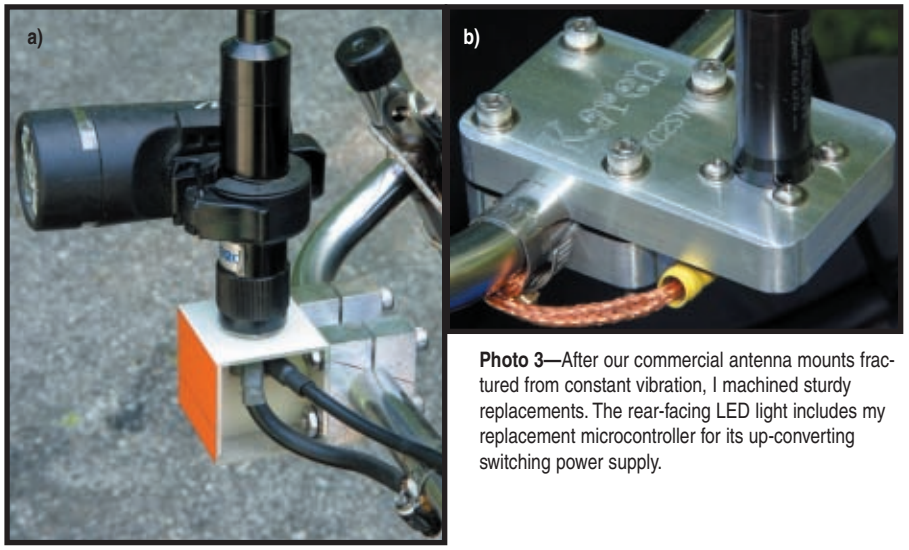


Photo 3—After our commercial antenna mounts fractured from constant vibration, I machined sturdy replacements. The rear-facing LED light includes my replacement microcontroller for its up-converting switching power supply.

circuitry, which works just as well and is far simpler than the ferrite toroids in my previous amplifier.

I used a Maxim Integrated Products MAX4468 electret microphone amplifier that's designed specifically for this task. It includes a shutdown function to save power; but at its highest, the amplifier draws less than 1% of the power required by the GPS receiver, so I didn't bother using it.

My original design amplified the voice signal by a factor of two, which made us rather soft-spoken. A few weeks of experimentation showed that a gain of 10 was about right. Your radios will certainly have different requirements, but this is the wrong place for a mechanical trimpot.

The MAX4468 has a decompensated amplifier that must be used with a gain of five or more. If your radio requires a gain less than five, use the MAX4467 instead: it's compensated for unity-gain stability.

The TT3+ includes a 5-V linear regulator, but its output carries all of the usual digital noise from the PIC microcontroller. I used an LM336, normally a voltage reference, as a simple shunt regulator to get rid of most of that hash. A resistive divider produces the $V_{CC}/2$ virtual ground required for single-supply op-amps.

The PIC microcontroller generates 1.2- and 2.2-kHz audio signals through a 4-bit discrete-resistor DAC, with only a dozen or so steps in each cycle. It has no reconstruction or antialiasing filters, because the radio's 3-kHz voice cutoff acts as a steep low-pass filter that eliminates most of the harmonics. The output is capacitively coupled from a trimpot level control.

My first-pass circuit combined the voice and data signals with an op-amp adder. That was a serious blunder, because the DAC's inactive output coupled a noisy background jangle behind the voice transmissions. I should have known better, right?

The second version of the circuit, shown in Figure 1, uses a MAX4544 single-pole/double-throw (SPDT) analog switch that connects one of the two inputs to the radio at a time. The voice signal now has no trace of machine chatter and the digital signal doesn't carry any noise from the microphone.

Speaking of microphone noise, a simple 20-mm ball carved from acoustic foam eliminates nearly all of the wind noise you'd expect from a microphone in a 25-MPH breeze. It's surprisingly effective and utterly trivial, with only one drawback: rain floods the microphone capsule and kills the audio. I still haven't figured out a waterproof and audio-transparent fix for that problem!

POWER CONTROL

Although the radio's microphone and speaker jacks share a common sleeve connection, I decided to isolate the microphone jack's common from the circuit board common. This prevents power supply current from flowing through the microphone cable, where it would add directly to the low-level audio signal. This may not be necessary in all applications, but it's a good trick to keep in mind when you're dealing with audio and digital circuitry.

A small 1:1 transformer passes the audio signal while isolating the two grounds. C8 blocks the $V_{CC}/2$ virtual ground bias on the input side.

Connecting the radio's microphone input line to the sleeve contact through a 33-kΩ resistor activates the transmitter, requiring AC coupling for the audio signal through C10 on the output side of the transformer. The radio has a 2-kΩ input impedance and the lowest audio frequency is about 300 Hz, so the blocking cap must be larger than:

$$C = \frac{1}{2\pi \cdot 300 \cdot 2 \times 10^3} = 270\ \text{nF} \qquad [1]$$

C8 and C10 are effectively in series, so I used tiny 1-µF caps to get 500 nF. The output is down about 1 dB at 100 Hz.

Optocoupler OK2 isolates the board's PTT signal from the radio's microphone common. The TT3+ includes diode steering logic that combines its internal PTT with the voice PTT push button on the handlebars, activating OK2 for both voice and data transmissions.

The GPS2 receiver requires about 70 mA, roughly twice the Icom's idle current. I didn't want a separate power switch for the GPS hardware, because I knew I'd never remember to turn it off, so I controlled the GPS power with the radio's microphone bias supply. OK1, another SFH615 optocoupler, has a minimum current transfer ratio of 65%, which, combined with the 2N2907 transistor's minimum hFE of 100, ensures that 8 mA



Photo 4—These PS/2 keyboard connectors allow the helmet to part easily from the bike. The joints have been reliable but the cable attached to the helmet fails every year or two.



Photo 5—The 3.5- and 2.5-mm plugs repeatedly ripped the jacks off the radio's circuit board, so the blob of epoxy putty fixes them firmly in place and eliminates that problem. The joints inside the connector shells have become intermittent, despite being potted in hot-melt glue.

from the microphone supply saturates the transistor when the radio is turned on.

I ran wires from the TT3+ power terminals directly to the 2N2907 and the board's battery power entry, thus keeping its relatively high currents out of the board's ground planes. I didn't do that in the first version and the difference is clearly audible: all this finicky analog layout stuff really does matter.

I've taken the circuit for some test rides and it works fine, although reliable reception poses more of a problem for digital signals. I must now do some CNC machining for a case that will replace the radio's battery pack and hold the two boards. We'll see how everything behaves during the radio club's fall public service event schedule.

Speaking of changes, the circuit board isn't the only thing that's changed since that early design. Here's a quick rundown of some other components that required attention.

THINGS THAT WORK AND FAIL

Let's start with the one component that has been absolutely reliable: the push-to-talk switch shown in Photo 2. I soldered a washable through-hole push button switch to a cable with a molded 3.5-mm plug on the other end, secured it to the handlebars with a blob of hot-melt glue that also sealed the cable end, and that was that.

The key attributes turned out to be washable and sealed, as similar switches failed within months. Mary preferred taller buttons that were, alas, available only on open-frame switches. After a year of intermittent chatting as the switches failed, she grew accustomed to that flat button.

On the RF side of the radio, I bought a pair of antenna mounts designed to clamp on truck mirror support bars, figuring that they'd last forever on our bikes' seat frame tubes. Six years later, metal fatigue fractured both mounts

WIRELESS RS-232

WCSC (Willies Computer Software Co)



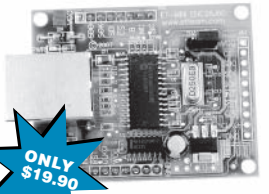
* Extremely easy to install and use
* Distances up to 100 meters (329 feet), 500 meters with extended antenna (1640 feet)
* Bluetooth Serial Profile
* One to One virtual cable or selectively connect one to many.
* Communicates with Bluetooth enabled PDAs, Laptops, Smartphones & Computer
* No special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

Save Up To 60% On Electronic Components



ONLY $19.90

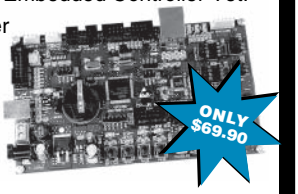
Exciting New Mini-Boards

Wide Range Including,
- New Ethernet Board
- SD Card
- Memory Cards
- Real Time Clocks and Much More

Powerful New ARM2368 Controller

Now available our most powerful Embedded Controller Yet!

- Includes LPC2368 Microcontroller - 512kb Flash Memory
- SD Card Socket
- Ethernet LAN Connection
- 2 CAN Network Controllers



ONLY $69.90



ONLY $25.90

Save Heaps on Components

Just arrived our new range of Solar Controllers, Gas Sensors and Wireless Data Link Modules, all at very competitive prices.

We are your one-stop shop for Microcontroller Boards, PCB Manufacture and Electronic Components.

www.futurlec.com

during a 350-mile bike tour.

Photo 3a shows an antenna mount I made from a box-section extrusion, Delrin plate, and CNC-machined circumferential clamps. Photo 3b shows a simpler mount machined from an aluminum plate. Both are absolutely rigid, which means the antenna mast or the coils and capacitors along it will fail next.

My December 2007 column explored the power supply inside the white LED blinky light clamped to the antenna in Photo 3a ("LED Power," *Circuit Cellar* 209). It's worked fine ever since and, in fact, NiMH AA cells last much longer than in the original lights. I'm tempted to replace the microcontrollers in our other lights, but I'm even more reluctant to mess with things that work.

My original helmet-to-amplifier connection used delicate earbud wire and PS/2-style mini-DIN keyboard connectors. That wiring disintegrated within one riding season, so I soldered a female DIN connector to a stout cable that replaced the helmet wiring.

That lasted another season before the cable fatigued at the helmet and the connector pins disintegrated. After trying an increasingly bizarre series of unsuccessful strain-relief ideas, I gave up and used molded keyboard cables on both ends, joined by a female-to-female adapter, as shown in Photo 4.

The heatshrink tubing that stabilizes the joints isn't entirely watertight, so rain affects the microphone signals. I could use watertight connectors, but that joint must slip apart to allow the helmet and bike to go their separate ways when I drop the bike. Yes, I've tested that feature.

Photo 5 shows the old microphone amplifier and radio connections. As I'd hoped, the surface-mount components in the microphone amplifier proved more durable than through-hole resistors, but the radio connections failed when vibration broke the internal jacks from their circuit-board moorings.

I finally stabilized the plugs with a blob of epoxy putty molded to the top and rear of the radio, a fix that promptly moved the failure point to the cables between the jacks and the radio. I had potted the connector shells with hot-melt glue and strain relief bushings, but that obviously wasn't enough.

Despite not being designed for a high-vibration environment, the radio itself held up remarkably well. We've been using long-obsolete Icom IC-Z1A radios and I'm setting up the GPS interface with the similar and slightly less obsolete IC-W32A shown in Photo 1. The current crop of water-resistant radios looks attractive, but these oldies continue to work.

The radio rides inside a foam-padded seat-wedge pack, strapped in a water bottle holder attached to the lower seat rail with two circumferential clamps. The tight packing and cables made battery changes quite difficult, so I converted an alkaline AA-cell holder into a remote battery connection. The black-and-blue Anderson Powerpole Products connectors, shown in Photo 5, mate reliably with an external pack that I can replace without disturbing the radio and its cables.

All this may seem like a nasty case of forcing cheap consumer hardware into the wrong application and then griping about the failure rate, but the truth is a bit more complex. I could use rugged components, but their cost and bulk seem unwarranted for a lightweight installation.

Worse, there's no single weakest link that would benefit from an upgrade. Apart from that PTT button, I've seen failures in essentially every component. Add up, if you dare, the cost of building everything with ruggedized and weatherproof components, then factor in a new radio. Multiply that total by three to outfit my family's bikes and I think you'll agree that desirable though an ideal system may be from a design standpoint, the bottom line simply isn't justifiable.

Having learned a lot more about vibration and strain relief, I hope that the current iteration will survive with fewer repairs and less down time, despite the additional complexity.

CONTACT RELEASE

For reasons best known to the manufacturers, no two radios have the same jack configuration. You should assume that you'll need a different radio cable as well as a different amplifier gain.

By the time this reaches print, you should be able to track the KE4ZNU-9's bicycle-mobile progress around the Poughkeepsie, NY, area through any of the APRS mapping web sites.

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/2008/219.

RESOURCES

B. Bruninga "Automatic Packet Reporting System," www.aprs.org.

D. DePriest, "GPS Data Format," www.gpsinformation.org/dale/nmea.htm.

E. Nisley, "Building Boxes," *Circuit Cellar* 173, 2004.

———, "Capacitors at RF," *Circuit Cellar* 175, 2005.

———, "LED Power," *Circuit Cellar* 209, 2007.

SOURCES

Powerpole connectors
Anderson Power Products, Inc.
www.andersonpower.com/products/single pole-connectors.html

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

Amateur radio products: IC-W32A dual-band transceiver and IC-Z1A handheld transceiver
Icom America, Inc.
http://icomamerica.com/en/amateur/default.aspx

X Amateur tracking and information system
XASTIR
www.xastir.org

RENESAS M16C Platform

M16C/64 & 65 - high performance additions to the M16C Platform

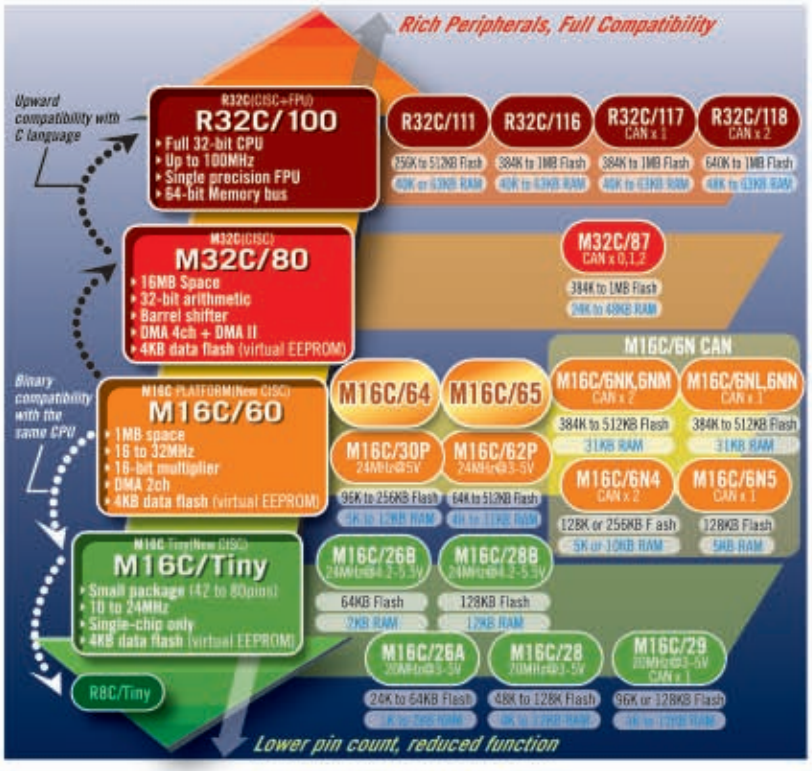
Up to 768K Bytes Flash, up to 47K Bytes RAM in 80- to 128-pin package

Renesas Technology

No. 1* supplier of microcontrollers in the world

presents the M16C/64 and 65 additions to the M16C Family, the industry's only fully code compatible platform that addresses the entire 8-bit through 32-bit price/performance application space. The M16C Family of devices protects your code investment with seamless code and pin compatibility across the platform. The M16C Platform features optimized peripheral mapping and on-chip memory that permits applications ranging from 24K Bytes to 1M Bytes, with between 42 and 144 pins, in an efficient 16-bit CISC architecture. Reduce your overall development time and cost with a cross-platform code base and development tools from Renesas Technology.

M16C Platform Product Lineup



HOT Products — **M16C/65 Group**

Setting a new standard for mid-range devices

M16C/65 Block Diagram (*M16C/65 100pin example)



Top Reasons to Select M16C/65

- **Broad Platform compatible to M16C/62P**
  - 128KB - 768KB Flash; 12KB - 47KB RAM; 80 - 128 pins
- **System cost down, through integration**
  - 40MHz On-Chip Oscillator
  - 125KHz On-Chip Oscillator for Watchdog
  - Power-On Reset circuit (POR)
  - Low-Voltage Detection (LVD)
  - Real-Time Clock (RTC)
- **Powerful 32MHz CISC operation**
  - High-speed hardware multiplier
  - 4 Direct-Memory-Access control channels (DMA)
- **Extensive communication interfaces**
  - 6ch scalable async/sync serial I/O channels
  - 2ch synchronized channels additional
  - 1ch I2C master interface
- **Other features**
  - 10-bit A/D with 1.72us conversion
  - 2ch 8-bit DAC output
  - 11ch 16-bit timer
  - Three-phase motor control unit

*Source: Market research firm iSuppli Corp. (El Segundo, Calif.)

#1MCU REACH FURTHER

Get Started Today -

Go online and register to be eligible for a FREE M16C Starter Kit*

www.america.renesas.com/ReachM16C/d

*Only available to customers in North and South America



DEVCON Developers Conference

Oct. 13-15 San Diego

www.RenesasDevCon.com

Renesas Technology Corp.

Everywhere you imagine. RENESAS

Saelig

UNIQUE PRODUCTS & SUPPORT

www.saelig.com

Testgear

Bargain MSO



25MHz 100 MS/s 2/16ch scope/logic-analyzer 5ns-5s/div with complex triggering. See wfm + data lines. 7.8" TFT Color LCD. USB-connected. 4MS buffer.

MSO5022 SALE! $799 ~~$899~~

100/200/300MHz DSO

NEW! RIGOL



100/200/300MHz 2 ch, 2000 wfm/s refresh rate. FreeCase!

DS1102C	(400MS/s)	**$999**
DS1102A	(2GS/s)	**$1295**
DS1202A	(2GS/s)	**$1595**
DS1302A	(2GS/s)	**$1995**

Scope + Analyzer

NEW! RIGOL



100MHz 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.

DS1102CD FreeCase! **$1499**

15/20MHz Wfm Gen

NEW! RIGOL



15/20MHz USB 14-bit Function/Arbitrary wfm Generators. 10 built-in wfms; FM, AM, PM, FSK, SWEEP and BURST; 6 digit 200MHz frequency counter.

DG1011 / DG1021 $650 / $795

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

Best Seller



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

50MHz USB Scope



PS3206 - Is an impressive 2-ch 200MS/s 8-bit PC scope adapter - 10GS/s for repetitive signals! No power supply needed, simple USB 2.0 connection

PS3206 Bundle $1574

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

World's Fastest



Winner!

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**

60/100/120MHz AWG

NEW! RIGOL



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

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**

Electronic DC Load



For testing dc power/battery sources Const. current, resistance, conductance, voltage and power modes 0 to 80 V and 0 to 80 A 300W Transient generator. Variable drop-out.

LD300 $1532

DMM

NEW! RIGOL



6 1/2 digit 50KS/s, 2MS mem. Opt. 16-ch mux. Automeasure DC/AC, 2/4-wire R, C, Continuity, Diode, Frequency, Period, Ratio, Temp, Sensor, High/Low Limit Math.

DM3061 $795

USB Bus Analyzers

NEW!



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

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

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

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

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

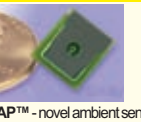
Wireless Analog



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

I2C for PCs



PCI93LV: industry-standard I2C card for PCs. **WINI2C/PCI** software gives windows-interface to develop and debug I2C bus systems. **UCA93LV** is USB version. Transparently monitor at 400kHz! **$499**

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**

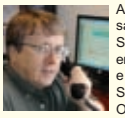
1/2/4/8/16 x RS232



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.*



Let your geek shine.

Meet Pete Lewis, lead vocalist for the band Storytyme. Pete recently created the RS1000, a new personal monitor system for performing musicians. It was SparkFun's tutorials, products and PCB service that enabled him to take his idea to market in less than a year.

The tools are out there. Find the resources you need to let your geek shine too.

sparkfun ELECTRONICS

Sharing Ingenuity

WWW.SPARKFUN.COM

©2008 SparkFun Electronics, Inc. All rights reserved.

Hear music from Storytyme at **www.storytymeband.com**, or check out Pete's RS1000 at **www.rockonaudio.com**.

Dynamic Animatronic Remote (Part 2)

Software and Menuing Code

Peter continues describing how he built an easy-to-use remote for his animatronic systems. This month, he describes the software and menuing code.

Each year, I put on an elaborate Halloween show at my house. The computer-controlled performance features custom animatronics systems run by multiple programs on my laptop. In previous years, the configuration process was a nightmare because my laptop was stationed in my house while the distributed computer hardware and animatronics were outside. To solve that problem, I designed a remote that can poll the various programs and create menus on the fly. As a result, I now can remotely control the laptop programs when I'm outside monitoring the animatronics show.

Last month, I covered the hardware and foundation software. In this article, I'll explain the menuing code and



Photo 1—PC programs create the menus on the remote's 2″ screen. The menus are well organized in the LCD's text format. **a**—This is the top-level menu on the remote. It's the only hard-coded menu in the system. All of the other menus are generated at runtime by querying the PC to discover which programs are running and which parameters they publish. **b**—Here I'm selecting the "Sounds" option. **c**—On this screen, I'm selecting the "Speakers" option. **d**—I can easily control the left and right speakers.

Listing 1—These three structures provide the core of the menu system. The MENU_GROUP structure holds an array of MENU_ENTRY structures. If a MENU_ENTRY is a modifiable parameter, it holds a PARAM_ENTRY structure.

```
typedef struct PARAM_ENTRY {
  unsigned char ControlNum; // Number of control given by PC
   unsigned char ParamType; // What type of parameter is it?
   int Min, Max;            // Min and max of range entries
   int CurrVal;             // current value of this parameter
   unsigned char Increment; // Amount to change the control each time
                            // the scroll wheel moves a unit
   char* ListString;        // If this is a list control, then this
                            // is the string to display
   char NumBytesListStringMalloc; // Store the number of bytes of memo
                                  // ry we malloc'ed for the list string
   struct PARAM_ENTRY* ParamMalloc; // If NULL, this entry is static
                            // memory. If not, then it's malloc'ed memory
   InternalFuncHandler InternalHandler; // Call this function for handling
                            // internal work (not sending messages) in
                            // response to a button push
} Param_Entry;

typedef struct MENU_ENTRY {
   char* MenuText;          // Menu text to display on the LCD
   unsigned char MenuType;  // What type of menu entry?
   struct MENU_ENTRY* MenuMalloc; // If NULL, then this entry is static
memory.  If not, then it's malloc'ed memory
/* NOTE! A menu item can only be either a param or a menu item, but
not both. Because of this, one of the members below must be NULL if
the other has a value. */
   struct MENU_GROUP* SubMenu;
   struct PARAM_ENTRY* Param;
   char PcNode;             // Halloween network node address for the
                            // PC program this menu belongs to.
} Menu_Entry;               // Single menu entry

typedef struct MENU_GROUP {
   struct MENU_GROUP* GroupMalloc; // If NULL, then this entry is
                                   // static memory.
                            // If not, then it's malloc'ed memory
   unsigned char Capacity;         // Number of possible menu entries
   unsigned char NumEntries;       // Number of menu entries used
   struct MENU_ENTRY** MenuItems;  // Array of menu entry pointers
   struct MENU_ENTRY* ItemsMalloc; // If NULL, then this entry is
                                   // static memory. If not, then it's
                                   // malloc'ed memory
   char ActiveMenuIndex;    // Which line is highlighted
   unsigned char TopVis;    // Top entry visible on screen. Needed when
                            // there are more entries than vertical
                            // screen space
   unsigned char ControlNum; // If this is a control in a PC program,
                            // it will have a number. Otherwise it's zero.
   unsigned int Extra;      // This can be used for any extra info
                            // needed for this group
} Menu_Group;
```

Listing 2—Menu entries can act as many different things. These constants are used to enable the menu code to determine how to handle a menu entry. The value is stored in the menu type member of the MENU_ENTRY structure.

```
#define MTYPE_FIXED 1 // Used for fixed headers. Simply displays text.
#define MTYPE_MENU  2 // Normal menu that leads to another menu or parameter
#define MTYPE_DISPLAY4 /* Menu that simply displays a named value, such
as "Ver. 1.12" Uses C string constants (%d, %s, etc) for value to display */
#define MTYPE_PARAM 5 // Displays a parameters that can be changed. Uses C
                      // string constants (%d, %s, etc) for value to display.
#define MTYPE_GROUP 6 // Menu that points to a sub-menu from the PC Program.
```

describe how I got the serious software running (see Photo 1).

THE MENU CODE

The basic premise of the menu code is that every menu that is displayed consists of menu entries stored in a menu group. The menu groups are stored on the menu stack. The stack is implemented as a simple array of MENU_GROUP structures, with a variable used as an index that tells you which entry is at the top of the stack. To show a menu, the code takes all the menu entries from the highest group and displays them on an LCD. If a menu entry leads to a submenu, that menu entry has a pointer to a menu group for that submenu. When you click on a submenu, the menu code copies that group onto the stack and displays the menu entries it contains. If there are more submenus in that group, you can continue to move down the menu hierarchy. When you move back up, the menu just pops the topmost group off the stack to reveal the previous group. The menu entries in this group are then displayed. Because each group contains information, such as which row is highlighted and which rows are visible, the previous menus are displayed exactly as you left them before descending. This provides a consistent user experience.

I'll spare you the details of the earlier versions of the menu code. Let's focus on the final version. For projects like this, it all starts with defining the structures that will be used to implement the system. I ended up with three key structures: one for menu entries, one for parameter entries, and one for menu groups (see Listing 1).

At start-up, the remote creates a root menu group and pushes it onto the menu stack. The group contains a few menu entries that are displayed as the initial and topmost menu. The root group is never popped off the stack because you can't ascend any higher than the root. Note that the MENU_GROUP structure contains an array of MENU_ENTRY pointers called menu items. This is in conjunction with the NumEntries member of the structure, which enables the menu code to figure out how many menu entries should be displayed. The group structure also has members that indicate which row of the menu is visible at the top of the LCD. This enables automatic scrolling through menus that have more entries than the LCD has rows to display them on.

Each MENU_ENTRY structure starts with the text that will be displayed on the LCD. The next member is the menu type. Menu types are defined in Listing 2.

Constants are used to figure out how

to display the text for the menu entry. For example, an MTYPE_FIXED menu is displayed on the top row of the LCD with the text centered. All of the other menu entries display the text as right-justified. The MTYPE_MENU is a normal menu entry that displays the text as it appears in the entry. MTYPE_PARAM entries are parameters. The text stored in the MENU_ENTRY structure is actually a string with "sprintf( )" formatting sequences. This enables the menu code to add the parameter's value at runtime using "sprintf( )" to format the values as ASCII text. MTYPE_GROUP menu entries get a > appended to the end. This informs you that you can click the Menu button to descend down a level. The text on a group menu is displayed as an MTYPE_FIXED entry when that group is displayed. This provides a reference at the top of the LCD that tells you what the menu entries represent. A sample menu running on the LCD is shown in Photo 1a.

Note that the MENU_ENTRY structure has two structure pointers in it. One is to a MENU_GROUP called SubMenu, and the other is to a param_entry structure called Param. Only one of the pointers can have a value at a given time. If the SubMenu pointer isn't NULL, the menu entry leads to another group of menu entries. If the Param pointer isn't NULL, the menu entry is actually a parameter and enables you to change a value on a PC program.

The Param entry structure was designed as a substructure to conserve memory. Every menu item needs a MENU_ENTRY structure to represent it, but only parameters need the extra information to let the menu entry act as a parameter. Why put all of the members in a single structure when most of them will go unused? By having the substructure, you can conserve memory and locate all of the important parameter information in a single, dedicated structure.

All of the menu code for the remote makes extensive use of malloc( ). This is a touchy point for some. Many designers feel that malloc should never be used in embedded projects due to the possibility of a malloc call failing at runtime, leaving the program without the memory it needs. I went back and forth on whether I should use malloc, but ultimately decided to use it. The standard non-malloc solution is to

Listing 3—This simplified function shows the conventional approach to memory allocation for a menu entry that has text assigned to it. Note there are two calls to malloc().

```
struct MENU_ENTRY* MENU_CreateNewEntry(char* EntryText{

      struct MENU_ENTRY* NewMenuItem;
      NewMenuItem = malloc(sizeof(struct MENU_ENTRY));

      NewMenuItem->MenuText = malloc(strlen(EntryText));
}
```


Data Acquisition & Control Computer

iPac 9302



- ARM9 200 Mhz Processor
- Floating Point Math Engine
- 2 USB 2.0 Host Ports
- SD/MMC Flash Disk Slot
- 48 Digital GPIO Lines
- 10/100 Base-T Ethernet port
- 5 channels of 12 bit A/D & 2 PWMs
- 1 RS232 & 1 RS232/422/485 Serial Port
- Battery Backed Real Time clock/calendar
- OpenEmbedded Based Linux Distribution
- Eclipse Linux Development Environment

.net Micro Framework

Microsoft® Windows®CE

2.6 Kernel

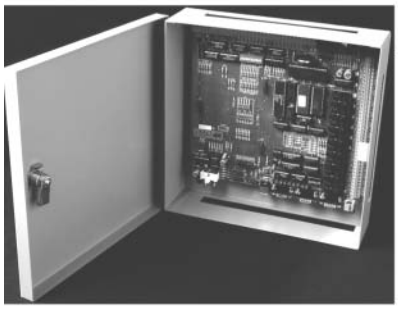
The iPac has enough I/O for demanding applications & with a size of 3.5" x 3.8" it can fit almost anywhere. Prices start at $150.00. For more information please visit our website: www.emacinc.com/sbc_microcontrollers/ipac_9302.htm

Since 1985 OVER 23 YEARS OF SINGLE BOARD SOLUTIONS

EMAC, inc.
EQUIPMENT MONITOR AND CONTROL

Phone: (618) 529-4525 • Fax: (618) 457-0110 • Web: www.emacinc.com

HomeVision-Pro



CONTROLS:
- Lighting
- HVAC
- Security
- A/V
- Much more...

HomeVision-Pro is the newest, most powerful automation controller in the award-winning HomeVision® product line. It's configured from a PC, but can then run standalone. Features:

- Two-way X-10 & IR
- Video screens on a TV
- 8 relays
- 16 digital & 4 analog I/O
- 64 digital temp sensors
- 1 USB & 3 serial ports
- Powerful scheduling (>8000 lines, >2000 objects)
- Clock
- PC web server
- Expandable
- Software included
- 3-year warranty

Custom Solutions, Inc.

800-398-8882, Web: www.csi3.com, E-mail: csi@csi3.com

Listing 4—This simplified function shows a more memory-efficient approach to memory allocation for a menu entry that has text assigned to it. Note there is just a single call to `malloc()`.

```
struct MENU_ENTRY* MENU_CreateNewEntry(char* EntryText){

  struct MENU_ENTRY* NewMenuItem;
  NewMenuItem = malloc(sizeof(struct MENU_ENTRY) + strlen(EntryText) + 1);

  NewMenuItem->MenuText = NewMenuItem + sizeof(struct MENU_ENTRY);
}
```

Listing 5—This Delphi code shows the basic usage of the Pub/Sub library. The library is responsible for updating all GUI elements that have been published when messages are received from the Halloween remote.

```
// Create Pub/Sub object
PS := T_PMPubSub.Create(MainFRM);
if (assigned(PS)) then
  begin
  // Create menus for Halloween Remote
  CreateRemoteMenus();
  end;
```

Listing 6—This Delphi code shows how simple it is for a program to publish GUI elements, which are displayed as menus and submenus, on the Halloween remote.

```
procedure TMainFRM.CreateRemoteMenus;
begin
// This start group is for the entire program
PS.StartGroup(_PSG_GROUP_SIMPLE, MainFRM.Caption);

PS.PublishScrollbar(MasterVolumeSB, 'Master Volume');

{   This group is type "_PSG_GROUP_SELECTOR" because the 1st entry ("WAV
    Files") acts as a channel selector.  When it changes, it brings up
    new params in the PC program which the Remote has to then re-query
    to get the new values and display them. }
PS.StartGroup(_PSG_GROUP_SELECTOR, 'Sounds');
PS.PublishRadioGroup(RadioGroupForRemoteRDGRP, 'WAV Files');
PS.PublishScrollbar(VolAttenSB, 'Volume Attenuation');
PS.PublishButton(PlayBTN, 'Play');
PS.StartGroup(_PSG_GROUP_SIMPLE, 'Speakers');
PS.PublishCheckbox(SpeakerLeftCB, 'Left Spkr.');
PS.PublishCheckbox(SpeakerRightCB, 'Right Spkr.');
PS.EndGroup(); // End of "Speakers" group
PS.EndGroup(); // End of "Sounds" group

PS.PublishCheckbox(CheckBoxForRemoteCB, 'Silence');
PS.EndGroup(); // End group for entire program
PS.SortEntries;
end;
```

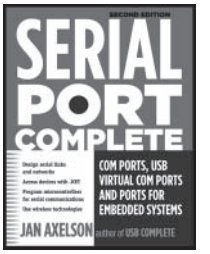
preallocate fixed arrays of memory for things like the `MENU_ENTRY`, `PARAM_ENTRY`, and `MENU_GROUP` structures. This sounds good until you need one more entry of a particular memory type and your program doesn't have it. Furthermore, there is no way to know in advance what type of structure memory I need the most of. Should I allocate 10 entries for each type? Twenty? If so, I'll probably end up wasting most of the memory entries while still running the risk of not having enough of one of them. And what about dynamic strings? The values displayed on the LCD for a parameter change as you move the scroll wheel. Do I create some big pool of memory to hold and build strings? I'd end up writing my own `malloc`-like library before long, simply by trying to avoid the use of `malloc`.

Using `malloc` enables me to avoid all of these problems. I never have an excess amount of one type of memory sitting around because each type is allocated on an "as needed" basis. In addition, I have many strings that





THE SERIAL PORT LIVES!



Everything you need to know about COM ports, USB virtual COM ports, & asynchronous serial ports for embedded systems.
Hardware & software for RS-232 & RS-485. Wireless options and more.

Serial Port Complete
Second Edition
Jan Axelson
ISBN 978-1-931448-06-2 $39.95
Lakeview Research LLC www.Lvr.com
From the author of USB Complete

don't change. If I had a parameter for sound volume, I might have a formatting string that looks like this:

"Volume: %03d"

This is a sprintf()-style string where the %03d lets sprintf insert the volume value and format it as a three-digit decimal string. I would then display the resulting string on the LCD, which might look like this:

"Volume:-20"

The resulting string is generated on the fly, sent to the LCD, and then "free'ed," but the formatting string is a constant once it has been created. I used a small trick to store these. When making the malloc call, you have to tell malloc how many bytes are needed. Let's say I need to malloc a menu entry structure. The function (simplified for clarity, it normally does things like check to make sure the malloc call succeeded) might look like Listing 3.

I made two separate calls to malloc: one for the structure itself and the other for the memory to store the string that was passed in the call. However, my function looks like Listing 4.

Notice that I added extra memory to the malloc call to accommodate the text used for the menu entry. The pointer from malloc is used as the pointer to the menu entry structure. However, I also assign the pointer to MenuText by calculating the first byte after the structure from the block of "malloc'ed" memory. This provides a few benefits such as smaller and fast code (one less call to malloc), but the biggest benefit is limiting memory fragmentation. If I allocate a lot of small blocks and then free some of them, I could end up in a situation where I have the physical memory, but not in large enough contiguous blocks. Instead, I allocate a smaller number of bigger memory blocks. While fragmentation is still possible, it's less likely with the larger allocations.

THE PC LIBRARY

Of course, because this is a networked device, the code and hardware for the remote are only half of the equation. I also had to write a library on the PC side that supported the remote. As I mentioned earlier, I have a large, prewritten library of code on the PC side that handles my Halloween network. I wrote it in Delphi, which is CodeGear's object-oriented Pascal. One of my goals on the PC side was to make adding support for the remote as simple as possible. I have a number of programs that run my Halloween show, and the thought of doing extensive modifications to all of them was not pleasant.

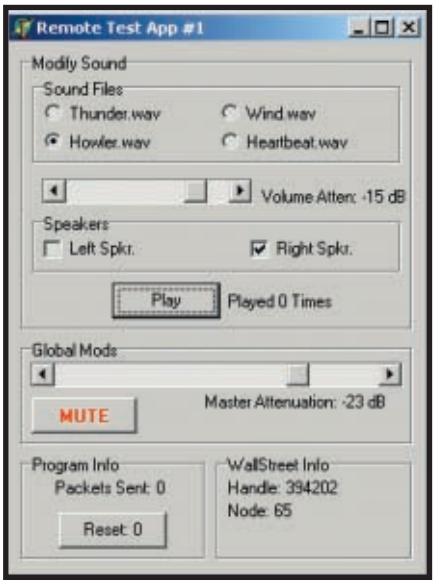


Photo 2—GUI elements on the PC are converted to text menus on the remote. Changes on the remote will update the GUI on the running program, as if the user modified it with the mouse.

I ended up creating a Publish/Subscribe object. To use it, a program simply creates the object using code like Listing 5. The call to CreateRemoteMenus( ) is not part of the Pub/Sub object. It's a local function that is specific to the program that wants to publish controls. The name is not specific, just what it has to do. An example of this function might look like Listing 6. This creates the hierarchical menu shown in Photos 1b, 1c, and 1d. The remote test application screenshot is shown in Photo 2. The StartGroup and EndGroup procedures tell the Pub/Sub object which entries make up a group. Notice that nested StartGroup/EndGroup pairs are allowed. This is what lets me create hierarchical menus with submenus, sub-submenus, and more. Next, you can see the various Publish. . . procedures. If there is a GUI element in a program that I want the remote to control, I simply make a Publish...call to the appropriate procedure, passing the GUI element and the name I want to appear in the remote as the parameters.

Internally, all of these calls build a list of controls that the remote can display or modify. After the last EndGroup, the SortEntries procedure is called. The sorting is done to create a list that makes sense to the remote. In order to keep memory usage on the remote down, I need all of the controls that make up a menu group to increase in value without any gaps or jumps. Without this, I would have to add an additional member to the MENU_ENTRY structure (and code to support it) that would indicate which control comes next in a menu group. Unsorted, the hierarchy from Listing 6 would look like Listing 7.

The number after the text is the control number assigned as controls were added. The > after some controls indicates that it leads to a submenu. The problem here is that the controls that belong to the group called Test Program are 1, 2, and 9. Clearly, there is a gap. After sorting, the hierarchy looks like Listing 8.

Instead of forcing you to add controls in the order necessary for the remote, the library lets you add them in the order you want them displayed. The SortEntries procedure then makes them work for the remote.

After that, the program doesn't have to do anything special. Each PC program already has code in place that handles the events generated by doing things like clicking on GUI elements with the mouse or typing on the keyboard. Once a GUI element has been published, the remote can send a message to the PC over the Halloween network and it will be received and processed by the Pub/Sub object. The Pub/Sub object will automatically and invisibly generate the correct events for the correct GUI element and the program

USB Connectivity for Embedded Designs



If you need Full-Speed USB 2.0 device, embedded host, dual role and On-The-Go solutions, Microchip Technology has them available today. We offer 8-, 16- and 32-bit MCUs with USB connectivity, providing easy migration with a single development environment. This maximizes pin compatibility and seamless code migration from 20 to 100 pins, enabling you to scale your USB design with ease.

Download FREE USB software including source code:

- Host Stack
- OTG Stack
- Device Stack
- Class Drivers (HID, Mass Storage and CDC Drivers)
- Thumb Drive Support (Mass Storage Driver, SCSI Interface, 16-bit and 32-bit File Management, Application Software)

USB Starter Kits accelerate development of USB designs using 8-, 16- or 32-bit MCUs starting at only $59.98



Core	Flash Program Memory	Pins	USB Type
8-bit	Up to 128 Kbytes	20 - 80	Device
16-bit	Up to 256 Kbytes	64 - 100	Device, Embedded Host, Dual Role, OTG
32-bit	Up to 512 Kbytes	64 - 100	Device, Embedded Host, Dual Role, OTG

The Microchip name and logo, the Microchip logo and PIC are registered trademarks of Microchip Technology Incorporated in the USA and in other countries. © 2008, Microchip Technology Incorporated. All rights reserved.

will then handle the event as if it came from the mouse or keyboard. If a program already works with the Halloween network, adding Pub/Sub capability is as simple as creating the Pub/Sub object, publishing some controls, and then sorting them. The rest happens without any additional code. I can add remote capability to my Halloween programs in a matter of minutes, which is exactly the way it is supposed to be.

IT'S SHOWTIME!

The finished remote uses an OKW case with a 0.125″ aluminum panel that fits in the top. I used Front Panel Express to create an exceptional panel to hold the LCD and main board. It was a wise $75 spent. I used a rugged connector from Neutrik to provide a cable hook-up to the remote. I originally planned to use industrialized Ethernet connectors, but they were too big to fit on my remote case. The cable is CAT-5, and the other end has a standard RJ-45 jack that plugs into my Halloween network. The Halloween network provides full-duplex RS-485 as well as 12 VDC, so plugging it in provides both data and power.

I'm really looking forward to this Halloween. I can't wait to modify

Listing 7—Without sorting, it's easy for controls to become nonsequential. Note that the Test Program menu would have controls 1, 2, and 9.

```
Test Program 0
          Master Volume 1
Sounds > 2
          WAV Files 3
          Volume Attenuation 4
          Play 5
          Speakers > 6
                  Left Spkr. 7
                  Right Spkr. 8
Silence 9
```

Listing 8—By sorting, the controls are added to menu groups in a way that makes the source code logical and easy to follow, while ensuring all controls in a group are sequential. The `Test Program` menu now has controls 0, 1, and 2.

```
Test Program 0
          Master Volume 1
          Silence 2
Sounds > 3
          WAV Files 4
          Volume Attenuation 5
          Play 6
          Speakers > 7
                  Left Spkr. 8
                  Right Spkr. 9
```



and control the show from outside while keeping the laptop inside. I won't be calling my friend in the yard on his cell phone, and I won't be running inside and outside like a madman.

This remote project turned out to be far more involved than I thought. The menu code went through at least three major iterations, each time gaining features and becoming more flexible. Interestingly, the code became smaller and more streamlined as I refined the data structures and came up with more elegant ways to achieve the same goals. The serial backpack I created for the Display3000 LCD was an entire project unto itself, but it made the project simpler overall. Furthermore, I have my eye on a bigger LCD from another vendor. After I buy that, I will create a new backpack for it and then just swap it in. If all goes well, I shouldn't even have to change any of the code running on the Halloween remote's main board.

My goal was to create an easy-to-use remote for my Halloween show that was not hard-coded to the programs I am running today. I wanted a future-proofed design—something that could literally handle programs I haven't even thought of yet. By creating a system that integrates into my existing Halloween network, I leveraged a lot of previous work. By using a system that polls the PC and talks to the programs that are running on it, the remote can generate menus on the fly, no matter what programs are running. The user interface includes three buttons and a scroll wheel, so learning how to use it takes about 30 seconds of playing around. It was a lot of work, but I know I'll be smiling on Halloween night!

Author's note: For videos and additional photos of the show in action, go to www.socalhalloween.com.

Peter Montgomery (servoanimation@socalhalloween.com) spent 12 years working as a visual effects supervisor on films such as Mortal Kombat *and* Ace Ventura: When Nature Calls *before becoming a director. He has directed dozens of commercials and made the transition to episodic television with The Disney Channel's* Lizzie McGuire. *Self-taught in both programming and digital hardware design, Peter uses his super powers only for good, not evil.*

PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/219.

SOURCES

Delphi IDE
CodeGear
www.codegear.com

D012 Display module
Display3000
www.display3000.com/html/english.html

Custom panel
Front Panel Express
www.frontpanelexpress.com

Z8F4821 Microcontroller
Zilog, Inc.
www.zilog.com





2007 WIZnet iEthernet DESIGN CONTEST
HONORABLE MENTION

Automated Data Mining

Build an Embedded Server Application

This well-designed embedded server application enables you to find airfare deals on the 'Net. The WIZnet W5100-based system uses Kayak, an online travel search engine, to search for flights. The data is then graphed to show you the best time of year to travel.

The bone-chilling wind and weekly ice storms that come with winter in central Illinois often leave us wondering where we can go to get away from the cold. Far away. Destinations on our short list this year include Jeju (South Korea), Buenos Aires, Acapulco, and Honolulu. As engineers, we didn't want to make a random decision with no logic behind it. We wanted to have a process. We wanted to travel to one of these far-off destinations, but how would we decide where to go and when?

As cost-conscious engineers, we wanted to locate the best airfare prices and travel dates. A few searches on one of the many available online travel search engines revealed that airfare can fluctuate significantly depending on the time of year. Airfare even fluctuates with the days of the week. We knew the only way for us to make a proper decision would be to collect about a year's worth of airfare data for our desired destination and then inspect the data for the cheapest time of year to travel. But manually executing a year's worth of airfare searches to a desired destination would have taken a long time. In fact, we would have spent the entire winter sitting in front of our computers, sipping hot chocolate, and running searches. But we had basketball to play and beaches to lounge on. Thus, we did what any engineers repulsed by the thought of endless repetitive work would do: we automated the entire process with a novel design. We call our design the Travel WIZard. It is an embedded server application featuring a WIZnet W5100 hard-wired TCP/IP Ethernet controller and a Microchip Technology Explorer 16

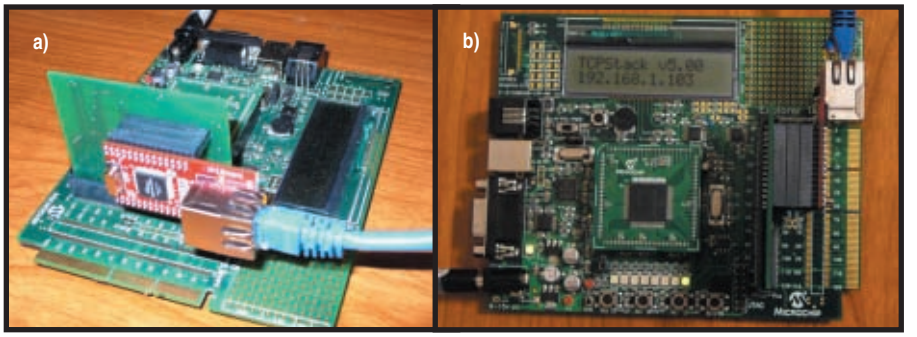


Photo 1a—The WIZnet WIZ810MJ network module plugs into a pair of female pin headers soldered to a 120-pin card edge connector board. The connector board plugs into the Explorer 16 development board. **b—**This is the top of the module.

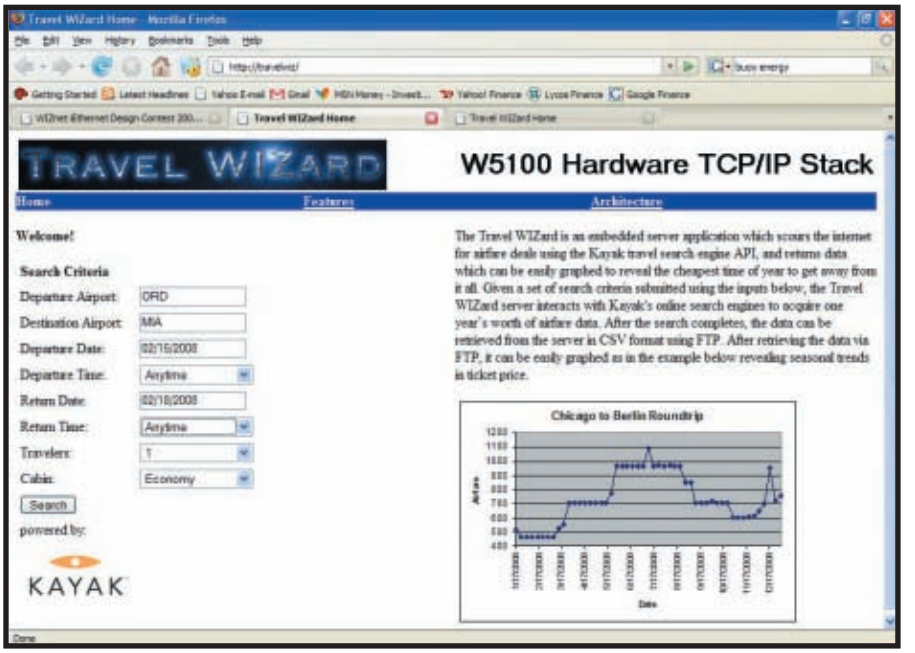


Photo 2—The Travel WIZard's server interacts with Kayak's search engine to gather a year of useful airfare information.

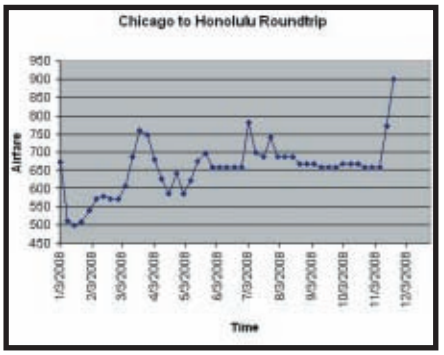


Photo 3—Data retrieved from the Travel WIZard is in comma-separated values (CVS) format. Flight search results retrieved via FTP are graphed to reveal seasonal trends in ticket prices.

development board (see Photo 1).

To use the Travel WIZard, simply enter your search criteria (e.g., departure airport, destination airport, and trip length) on the Travel WIZard web site (embedded in EEPROM on the Explorer 16). When you click the button to start your search, the embedded server interacts with the Kayak flight search engine (www.kayak.com) to acquire one year's worth of airfare data (see Photo 2). After the data is acquired, you can retrieve the results in a comma-separated value format via FTP. The resulting data can be easily graphed to reveal seasonal trends in ticket prices. Photo 3 is a screenshot of the results from a Chicago-to-Honolulu flight search.

HARDWARE

We used a Microchip Explorer 16 development board populated with a PIC24FJ128GA010 microcontroller as the basis for the project (see Figure 1). The Explorer 16 brings out many of the microcontroller's pins to a 120-pin card edge connector, including two SPI bus interfaces and the PIC24's parallel bus interface, the parallel master port (PMP). We acquired a WIZnet WIZ810MJ network module to handle our Ethernet interface. The WIZ810MJ is an excellent choice for quick Ethernet prototyping because it comes as a plug-in module with pin headers and it includes a MagJack and supporting circuitry for the W5100. It also brings out the W5100's SPI and parallel bus interfaces to the pin headers, making communicating with the device a snap.

The only problem was figuring out how to physically connect the WIZ810MJ to the Explorer 16 board (see Figure 2). Additional schematics, figures, and files are available at www.circuitcellar.com/wiznet/winners/001130.html.

After considering some kind of breadboard-fly-wire-solder combination, we decided there was no better time than the present to design our first PCB. Although it was intimidating at first, the PCB layout process turned out to be an enjoyable experience. We used the freeware version of CadSoft Computer's Eagle layout editor to design the board. The freeware version has some limitations with respect to board size, number of layers, and circuit complexity, but it was more than adequate for our purposes. Advanced Circuits fabricated the board. It took about two weeks to lay out the board, and then an additional week to receive prototype boards. The completed assembly is shown in Photo 1. The WIZ810MJ plugs into two pin headers, which are soldered to the adapter board. The adapter board plugs into the 120-pin card edge connector on the Explorer 16.

We included both W5100 bus interface options in our layout to maximize flexibility. We use the PIC24FJ128GA010's general-purpose I/O pin RF1 to select between the SPI and parallel bus interfaces via the WIZ810MJ's SPI enable control line. At build time, a preprocessor definition selects between the SPI bus driver and the parallel bus driver, so only the necessary code is included in the build.

The PIC24FJ128GA010 has a PMP peripheral specifically designed to interface with parallel bus-enabled devices such as the W5100. The PMP is convenient because it includes special hardware to handle chip selects, read/write lines, and wait states. However, its many pins are multiplexed with various other peripherals of the PIC24FJ128GA010, including a second SPI module, which is used to interface with the EEPROM that stores our embedded web page. We didn't want to lose the ability to serve a web page from our board just because we switched to the parallel bus, so each time our W5100 parallel bus driver software functions are called, the PMP is enabled, data is transferred, and then the PMP is disabled. When the PMP is disabled, the associated pins revert to the function they had been previously assigned. In this case, that means the SPI module pins go back to SPI functionality. Enabling the PMP only when we need to access the W5100 incurs a slight run-time penalty because we have to execute more than the necessary number of instructions every time we access

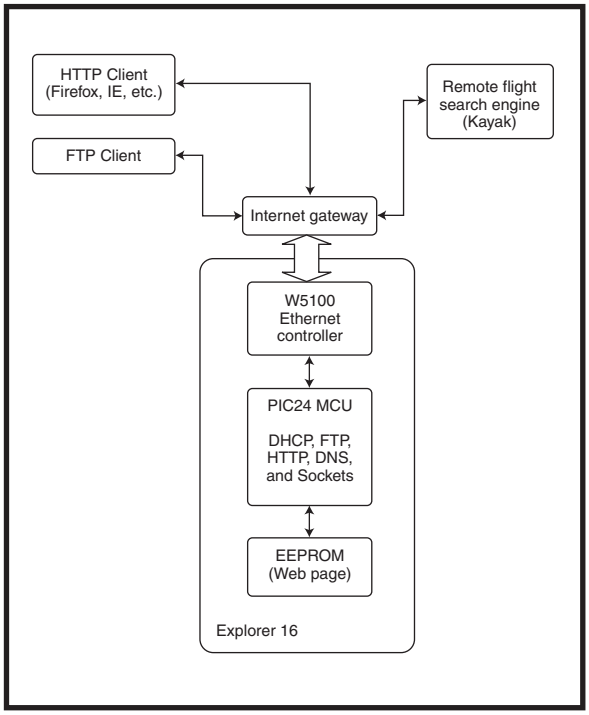


Figure 1—This block diagram shows the Microchip Explorer 16 development board populated with a PIC24 microcontroller at the heart of the system. You can connect to the system via HTTP or FTP to begin a search session or retrieve search results. Once search criteria have been entered, the Explorer 16 automatically interacts with the Kayak flight search engine to retrieve airfare search results.

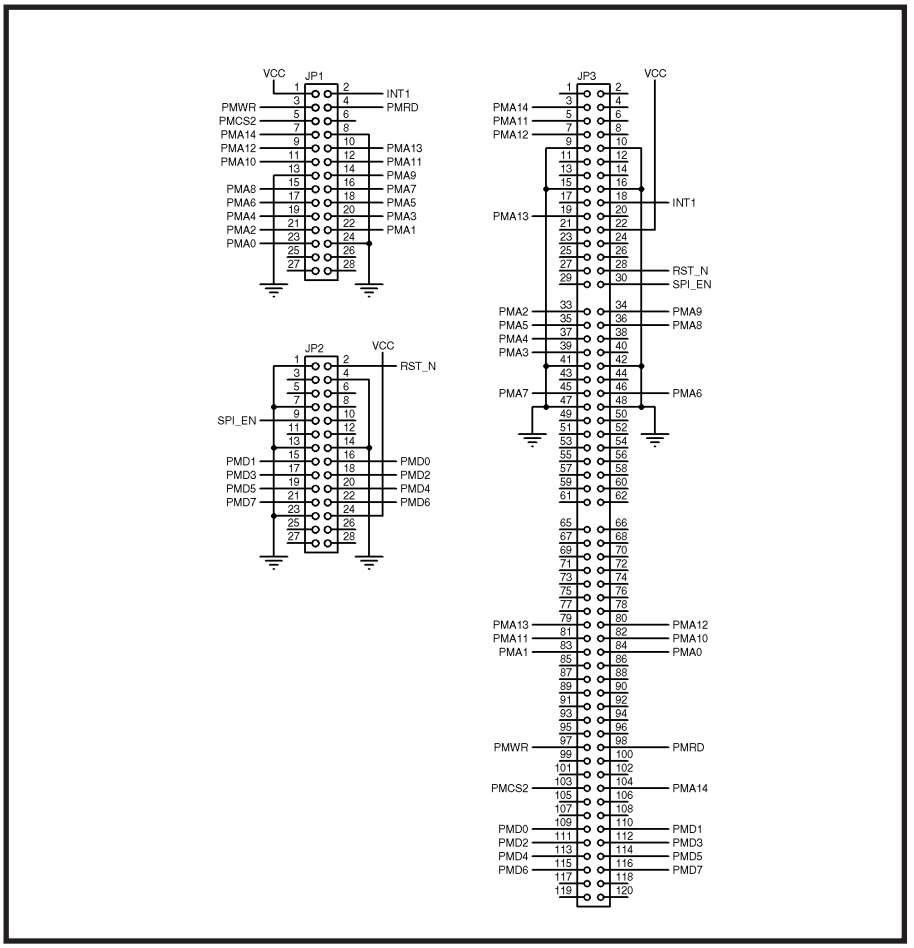


Figure 2—The two jumpers are located on the WIZ810MJ development board. On the left, you can see the connections from the WIZ810MJ 28-pin connectors to PIC24 processor pins. On the right, you can see the Explorer 16 board's 120-pin card edge connector. These pins go directly to the PIC24 processor.

the device. However, it enables us to maximize the PIC24FJ128GA010's available pin functions.

TCP/IP STACK

Microchip provides a free TCP/IP software stack designed to interface with its ENC28J60 Ethernet controller. The stack is a modular implementation of the five-layer TCP/IP model. ENC28J60 driver routines are at the lowest level of the stack. On top of the hardware drivers are the network layer ARP and IP protocols, followed by the transport layer TCP and UDP protocols, and finally the application layer protocols such as HTTP, FTP, and dynamic host configuration protocol (DHCP). In a layered software environment, the rule is that each layer can depend on only the functions in the same layer or lower. That means functions in the network layer should not be calling functions from the higher transport layer. Layering software also means that data and implementation details can be encapsulated within a layer. The advantage is that implementation details may change in a given layer; but as long as the functionality stays the same, the upper layers will never be the wiser. Understanding the layered structure of the stack is important because it is the mechanism that enabled us to replace the software stack with the W5100's stack, which implements everything from the transport layer on down in hardware.

Let's clarify all of this abstraction with a specific example. Listing 1 is a code snippet from Microchip's HTTP server routine. The logic flow is not too informative when taken out of context in this snippet, but it is useful to notice that the routine is relying on transport layer functions—such as `TCPDisconnect`, `TCPGetArray`, and `TCPDiscard`—to do its work. Listing 2 reveals the internal details of the software stack `TCPDiscard` function, which is used to clear a socket's receive RAM. Compare that to Listing 3, which shows the same `TCPDiscard` function. This time it is implemented for the hardware stack. Notice that the function prototype is the same, as is the overall functionality (the socket's receive RAM is cleared). Under the hood, however, the implementation details have changed drastically. Nevertheless, from the perspective of the HTTP server routine, nothing has changed. It can call the same function and obtain the same result.

For another example, refer to Listing 3 for the hardware stack that we relied on for our W5100 driver routines `W5100WriteReg` and `W5100ReadWord`, which we used to access registers on the Ethernet controller. Doing so enabled us to swap out a SPI bus driver with a parallel bus driver without having to make a single change to any layer-accessing W5100 registers. This incredible technique enabled us to switch from using the SPI bus interface to using the parallel bus interface in a single evening. It was amazing considering it took several weeks to get the entire application functioning properly with the SPI bus.

The Microchip stack implements numerous application-layer protocols. Because of the modular implementation, we were able to rip out the guts of the lower layers of the stack and replace them with W5100 socket driver routines. We were careful to keep the functionality and external interface of the transport and network layers exactly the same. No modifications were necessary to the application layer software. This powerful technique enabled us to reuse all of Microchip's application protocols. At one time or another, we ran HTTP (server and client), DHCP, TELNET, FTP (server and client), DNS, and NBNS application protocols using the W5100. As we tested each protocol one by one, we had to learn enough about it to be able to

Listing 1—Microchip Technology's application layer implementation of HTTP relies on transport layer functions to handle the details of TCP. The implementation details of TCP are hidden from the application layer.

```
static void HTTPProcess(HTTP_HANDLE h)
{
…

if(w == 0xFFFFu)
{
       if(TCPGetRxFIFOFree(ph->socket) == 0)
       {
          // Request is too big, we can't support it.
          TCPDisconnect(ph->socket);
       }
       break;
}

lbContinue = TRUE;
if(w > sizeof(httpData)-1)
       w = sizeof(httpData)-1;

// Populate httpData array with w bytes
TCPGetArray(ph->socket, httpData, w);
httpData[w] = '\0';

// Discard remaining data from this socket
TCPDiscard(ph->socket);

…
}
```

Listing 2—This code snippet reveals the implementation details of the `TCPDiscard` function from Microchip's software stack.

```
void TCPDiscard(TCP_SOCKET hTCP)
{
          TCB_STUB *ps;

          if(TCPIsGetReady(hTCP))
          {
                  ps = &TCBStubs[hTCP];
                  LoadTCB(hTCP);

                  // Delete all data in the RX buffer
                  ps->rxTail = ps->rxHead;

                  // Send a Window update message
                  SendTCP(hTCP, ACK);

                  SaveTCB(hTCP);
          }
}
```

know whether or not it was working correctly. Doing so enabled us to acquire a mountain of knowledge on the inner-workings of the Internet during our testing.

FLIGHT SEARCH PACKAGE

After mastering the low-level details of Internet communication, it was time to move on to the task at hand: automating airfare data mining. We implemented a custom TCP client software package for retrieving airfare information from a remote server. The client software uses the transport layer driver interface to communicate over Ethernet. The package is a group of related files that enable the flight search TCP client to be easily integrated into the top-level application. The centerpiece of the package is the flight search task, which actually implements the API for communicating with the remote flight search server.

The flight search task waits for a software flag to be set, indicating that search parameters have been entered and a search may begin. The flag is set when a button is clicked on our board's web page. We added code to the Microchip HTTP server to kick off a flight search session. When the flag is set, the next time the flight search task runs, it connects to the remote flight search server Kayak.com using DNS to resolve Kayak's IP address. After the IP address of Kayak's server was resolved, we configured a W5100 socket to be used in TCP client mode. The TCP client mode socket establishes the connection with Kayak, and then uses that connection to acquire airfare data.

The first step toward interacting with Kayak is to acquire a search session ID, which is needed to submit search queries. The Kayak server uses an HTTP interface so all communication is done using TCP. Our client software calls the transport layer TCP functions to transmit to and receive data from Kayak.

Kayak returns all data in XML format. For us, that meant writing a couple of simple parser functions in our client to retrieve data from within Kayak's XML tags. We learned that the equivalent of a function call is made over the Internet by using a web address as the function name followed by CGI-formatted function arguments. Furthermore, the data transmitted over the Internet is simply ASCII text. For example, to acquire a session ID from Kayak, we make the "function call" by writing "api.kayak.com/k/ident/apisession?token= cRGMxxxxxxxxhT7kSEZE2Dw" to the W5100's transmit RAM and flushing our TCP socket. The "token" argument is actually a developer key, which must be acquired from Kayak in order to use the API. Kayak responds to the function call with an XML document. One of the elements within the XML document is the session ID. We parse through the socket's receive RAM to

locate the session ID. We then store it safely away into processor RAM and clear the receive RAM buffer.

The session ID, after being acquired once, is used with all ensuing flight search queries. The session ID expires after 30 min. of inactivity (no search queries). To manage the load on its servers, Kayak restricts the frequency of search queries. Developers are limited to about 41 queries per hour. Beyond that, Kayak responds with a message indicating that you have run out of search quota and you need to wait (a message we received frequently during development).

We submit a search query using data supplied from the user entered into our web page via HTTP. We then wait 90 s before polling for results. After retrieving the results from Kayak's XML document, we wait another 90 s, update our search dates to a week later, and run another query. The process continues until 52 searches—covering one year from the initial search date—have been performed. At our current quota limit of 41 queries per hour, it takes about 156 minutes to complete a search run.

We save the top result (the cheapest result from each flight search query). The result is parsed out of Kayak's XML data in the W5100 socket's receive RAM. A snippet of an XML reply from Kayak is shown in Listing 4. This is the actual data that the W5100 buffers into its receive RAM. From this, we parse out the airline and the price (in bold) and store them into an array in microcontroller RAM along with our calculated date.

During a flight search run, all four sockets on the W5100 are active simultaneously. We have one socket in TCP Server mode listening to TCP port 80 for HTTP connections. Another TCP server socket is listening on TCP port 21 for FTP connections. A third socket is configured for UDP to listen for NetBIOS name service requests. The fourth socket does double duty. It manages our DHCP lease, and it is used in TCP Client mode to connect to Kayak and retrieve flight search results. We can use a single socket to manage DHCP and connect to Kayak because, in both cases, the socket does not need to stay connected all of the time.

Listing 3—This code snippet illustrates how implementation details of the `TCPDiscard` function were changed to support the W5100's hardware stack. The interface and functionality remain the same, so no changes are required to higher level software. With this new implementation, the function relies on lower-level driver routines to communicate with the W5100.

```
void TCPDiscard(TCP_SOCKET hTCP)
{
      WORD ReceiveDataLength, ReadPointer;

      if((ReceiveDataLength = TCPIsGetReady(hTCP)))
      {
        /* Retrieve the current read pointer */
        ReadPointer = W5100ReadWord(Sn_RX_RD0(hTCP));

        /* Increment by the size of available data */
        ReadPointer += ReceiveDataLength;
        W5100WriteReg(SnRXRD0(hTCP),((ReadPointer&0xFF00)>>8));
        W5100WriteReg((SnRXRD0(hTCP)+1),(ReadPointer & 0x00FF));

        /* Delete all data in the RX buffer */
        W5100WriteReg(Sn_CR(hTCP),W5100_SOCKET_RECV);

      }
}
```

Listing 4—The Kayak flight search API returns results in XML format. Key off of the tag names to parse out results of interest such as airfare price and the airline offering the trip.

```
<?xml version="1.0"?>
<searchresult>
  <searchinstance></searchinstance>
  <searchid>s_c7Lu3a6QgVzrppeNtQ</searchid>
  <count>15</count>
  <morepending>true</morepending>
  <trips>
          <trip>
          <price url="/book/flight?code=2-
zEGArPtH5_L$r14jZSJ1.s_c7Lu3a6QgVzrppeNtQ.19-
ZGnLUYLfGuVsthbFkgeW.F.AA.20880.0&amp;_sid_=19-ZGnLUYLfGuVsthbFkgeW"
currency="USD">209</price>
            <legs>
                <leg>
                <airline>AA</airline>
                <airline_display>American Airlines</airline_display>
                <orig>ORD</orig>
                <dest>MIA</dest>
                <depart>2008/01/17 20:10</depart>
                <arrive>2008/01/18 00:05</arrive>
```

FINDINGS & IMPROVEMENTS

We learned a lot about the Internet as we worked on this project. Sure, we were already familiar with terms such as "MAC address," "DHCP", "gateway server," and "client application," and we knew enough to get by. But we didn't know too much about how the Internet really functions under the hood. This project taught us a lot about the underlying details of Internet communication.

If you're planning to put your current embedded project online, we recommend obtaining a network protocol analyzer. It will do wonders for your debugging capabilities. We used an excellent open-source application called Wireshark (formerly Ethereal). Wireshark captures Ethernet packets in real time from your PC's wireless or LAN cards. Packet contents are

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

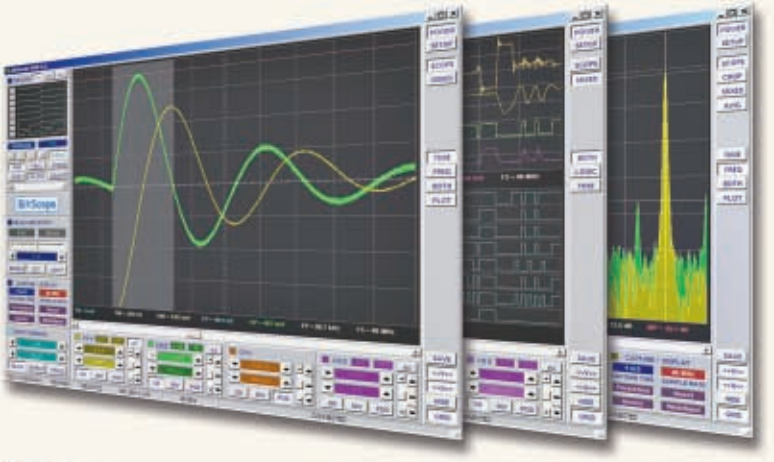
BitScope DSO Software for Windows and Linux

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

displayed in a user-friendly, easy-to-read format. Depending on the packet's protocol, the contents are broken out byte-by-byte according to the different fields contained in the packet. The program also has numerous filtering options to limit the number of captured packets to only what you are interested in. Wireshark was essential to the development of our project.

PLANNING AHEAD

We have used the Travel WIZard to acquire airfare data to destinations all over the world. It is interesting to observe seasonal trends in ticket prices to different locations. However, the application has plenty of room for improvement. The once-per-week sample rate is rather sparse, allowing opportunities for price variation that we could miss if we happen to sample on the wrong day. We plan to implement a more dense mechanism for storing results so we can pack a greater number of results into the same amount of RAM. (Currently, we store in ASCII format.)

We also want to be able to view the search progress from the embedded web page. Google offers a graphing API (Google Chart), in much the same way that Kayak offers an API, which can be used to graph data. Using the Google Chart API, we could graph the flight search results as they come in, right on the embedded web page.

Many web sites offer easy-to-use APIs just like Kayak. Using the W5100 and the Internet application layer protocols, the number of applications is limitless. We are anxious to add more automated packages to the Travel WIZard to make it an even more productive machine. Doing so should enable us to spend more time on the beach while the system does the dirty work for us.

Editor's note: To learn more about this project, you can review the schematics, photos, diagrams, and additional files posted at www.circuitcellar.com/wiznet/winners/001130.html.

Matt Pennell (mgpennell@gmail.com) is an embedded software engineer for Caterpillar's Electronics & Connected Worksite Division. He specializes in hardware drivers, TPU microcode, and eTPU microcode for engine control units. Matt earned a B.S.E.E. at the University of Illinois at Urbana-Champaign. His technical interests involve designing hardware and software for real-time embedded systems. Outside of work, Matt enjoys weight lifting and sports.

Aaron Thomas (thomaal@gmail.com) is an engineer for Caterpillar's Electronics & Connected Worksite Division. He specializes in FPGA development for electronic control modules. Aaron earned a B.S. in computer engineering from the University of Toledo in Ohio. His technical interests include computer design, programming, and automation using embedded systems. In his spare time, Aaron enjoys traveling, sports, and video games.



PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/219.

SOURCES

EAGLE Light Edition
CadSoft Computer, Inc.
www.cadsoftusa.com

Wireshark Network analyzer
Gerald Combs
www.wireshark.org

ENC28J60 Ethernet controller, Explorer 16 development board, and PIC24FJ128GA010 microcontroller
Microchip Technology, Inc.
www.microchip.com

WIZ810MJ Network module and W5100 Ethernet controller
WIZnet, Inc.
www.wiznet.co.kr/en

LUMINARY MICRO®

Connected. Versatile. Cost-effective.

Stellaris means:

- The most advanced integrated MCUs with high-performance Ethernet, USB, CAN and serial communications
- Complete open-tooled, ready-to-implement reference design kits for connected applications
- Cost-effective modules for rapid time to market
- Advanced interrupt-driven software with Stellaris Driver Library support
- Entry into "$1 to 1GHz" ARM architecture compatibility





RDK-IDM
Intelligent Display Module

Featuring color touch QVGA display with Ethernet and Power-over-Ethernet or direct power input

$ 219.00

RDK-S2E
Serial-to-Ethernet Module

Featuring two serial UARTs to 10/100 Ethernet, with software

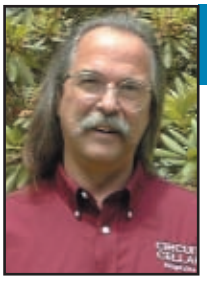
$ 139.00

www.luminarymicro.com

LUMINARY MICRO®

STELLARIS® microcontrollers

FAIRCHILD SEMICONDUCTOR the power franchise Cortex ARM



Serial Terminal Solution

Jeff used to keep an old laptop in his lab just because it had a real RS-232 port. But he found himself in the lurch recently when the laptop broke in the middle of a design project that required an RS-232 port for operational verification. Fortunately, Jeff had another work in progress that involved the hardware he needed to fix the problem. In this article, he describes his serial terminal solution.

Serial terminals continue to be one of the most useful gadgets in our arsenal of peripheral equipment. Whether they are used for user input, an output display, or a combination of both, serial ports continue their reign even after a methodical burial by USB. But, with the elimination of RS-232 from every new laptop and desktop computer, it's becoming harder and harder to find ways of interfacing to equipment that requires a serial terminal for some function.

USB dongles that present themselves as serial ports are finicky. They are touted as foolproof; but as anyone who's used one knows, they can be rather frustrating. When they don't work, you can't tell whether it's a driver issue or a faulty

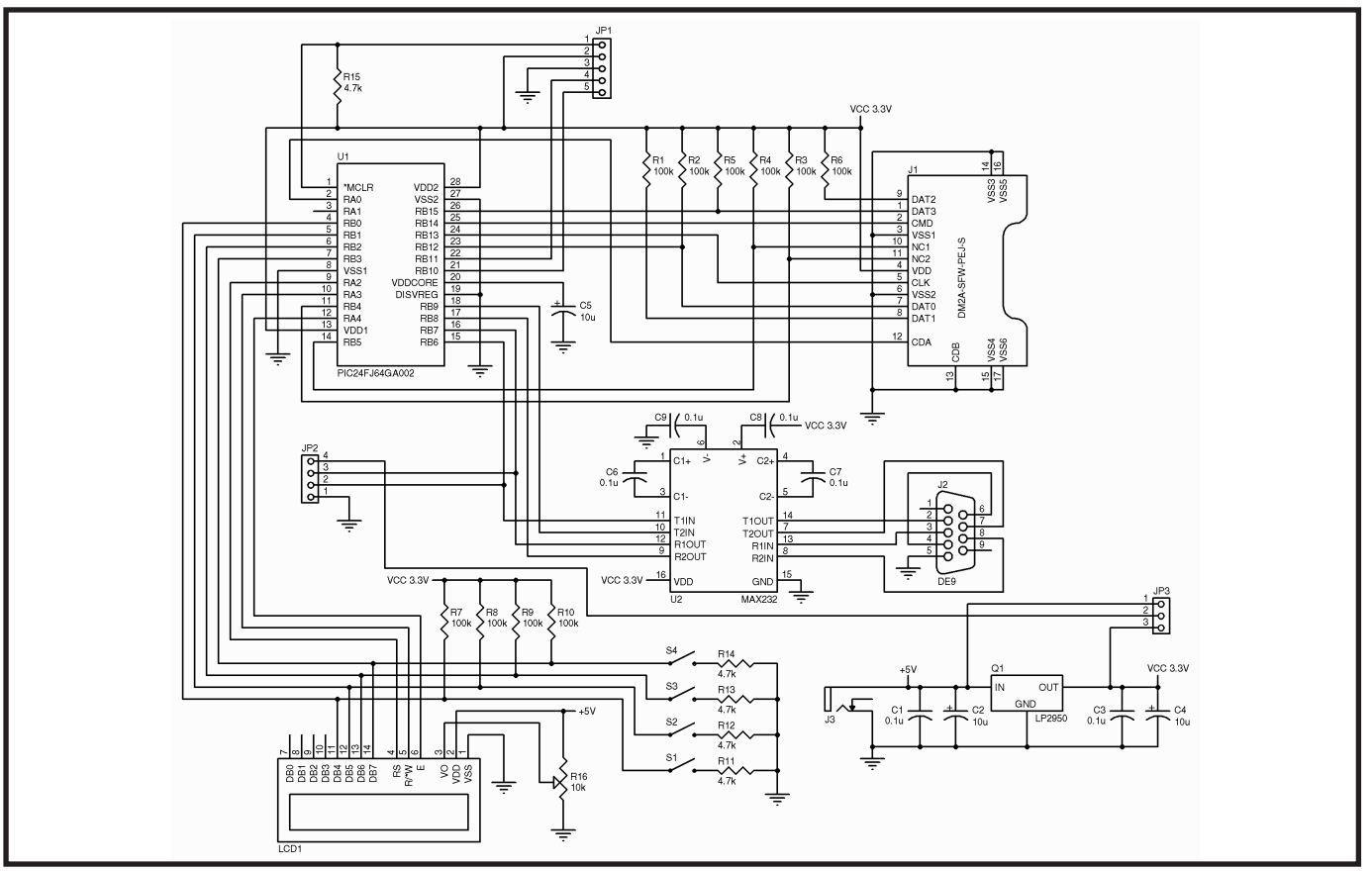


Figure 1—I borrowed this schematic from an upcoming column because it has all of the circuitry necessary for this application. The PIC24FJ64GA002 runs on 3 V, but has 5-V-tolerant inputs and can drive higher voltages using open collector drivers for 5-V outputs. The LCD, push buttons, and RS-232 converter are the only peripheral parts used. Note that S4 should not be used because it could interfere with reading the LCD's busy bit on DB7.

dongle. I used to keep an old laptop around because it had a real RS-232 port on it. During a recent design project, a company required an RS-232 port as part of the design for operational verification. (Imagine RS-232 as a design requirement!) With the design prototyped, I had just begun to exercise some code when my laptop flew south for the summer and every desktop I owned no longer had any RS-232 hardware.

As luck would have it, a project I was developing for an upcoming column had just the hardware necessary to give me what I needed to keep the present project on track. This was a good excuse to check out the prototype hardware that was patiently waiting its turn for the limelight. This proposal seemed a little like watching a "big-screen epic" on one of those handheld LCD televisions. Using a 2 × 20 display as a serial terminal has serious issues. Let's see if this application has any merit.

STOLEN PROTOTYPE

The schematic diagram for my future project is shown in Figure 1. J1 will not be used in this circuit's incarnation. Although the microcontroller runs on 3.3 V, it has 5-V tolerant inputs and can directly interface high-voltage circuitry via configuring outputs as open drain. The Microchip Technology PIC24FJ64GA*xxx* microcontroller has an interesting feature that is especially valuable in low-pin-count microcontrollers, peripheral pin select (PPS). It allows up to 26 I/O pins to have their default functions reconfigured. Potential candidates for the PPS function are listed in Table 1. Locking registers prevent the configuration from being altered (unless the correct procedure is followed). While this allows changes on the fly, it can prevent changes by errant code.

Input selections	Output selections
	NULL
External interrupt 1	
External interrupt 2	
Timer2 external clock	
Timer3 external clock	
Timer4 external clock	
Timer5 external clock	
Input capture 1	Output compare 1
Input capture 2	Output compare 2
Input capture 3	Output compare 3
Input capture 4	Output compare 4
Input capture 5	Output compare 5
Output compare fault A	Comparator 1 output
Output compare fault B	Comparator 2 output
UART1 Receive	UART1 Transmit
UART1 Clear to send	UART1 Request to send
UART2 Receive	UART2 Transmit
UART2 Clear to send	UART2 Request to send
SPI1 Data input	SPI1 Data output
SPI1 Clock input	SPI1 Clock output
SPI1 Slave select input	SPI1 Slave select output
SPI2 Data input	SPI2 Data output
SPI2 Clock input	SPI2 Clock output
SPI2 Slave select input	SPI2 Slave select output

Table 1—Input and output functions of these peripherals in the PIC24FJ64GA*xxx* can be configured to any of the remappable port pins.

The circuit uses the legendary MAX232-type RS-232 level shifter to interface the serial port interface to the microcontroller. The microcontroller multiplexes an LCD and a few push buttons as user I/O. Using the 4-bit LCD mode reduces interfacing to just seven lines. Three of the four data lines have push buttons that can bring the data lines low when not being driven by the microcontroller. Because the microcontroller needs to read only the most significant data line (DB7) to monitor the LCD's busy bit, the push buttons (on the other three data lines) will not interfere with any data.

UART

The real challenge is not in the circuitry, but how the available circuit is used to make the application useful. So, let's start with the UART, which is responsible for clocking data out at a predetermined fixed rate and format. At idle, the transmitter remains in the Mark, or the TTL logic level 1 state. (Space is known as a TTL logic level 0 state.) The clocking rate, or data rate, is the duration of a particular state (Mark or Space), usually defined as states or bits per second.

This is a one-wire transmission, so the receiver must be able to stay in sync with the transmitter. This is possible if the timing logic of the transmitter and the receiver are set to the same data rate and the asynchronous transmission always begins with a start bit (a Space), telling the receiver to begin its reception of bit states. The bits directly following the start bit are sampled at some point during the expected duration of the bit to determine the state (Mark or Space) of that bit. Data bits are sent least significant bit first.

One of the things that made the serial port popular was its ability to be configured to accept data in several ways. First, the serial data could be streamed at different rates. While initial speeds were less, 300 bps became the standard as acoustically coupled modems were used in the early 1960s (see Photo 1). Baud is the data rate in number of pulses or bits per second. Early TeleTYpewriter (TTY) devices used reduced character sets requiring 5 bits to send a data character (actual data).



Photo 1—The first data modems used acoustic tones for a whopping 300 bps. If you still have one of these on the shelf, raise your hand. (Source: http://en.wikipedia.org/wiki/Image:Coupleur-accoustique-IMG_0298.JPG)

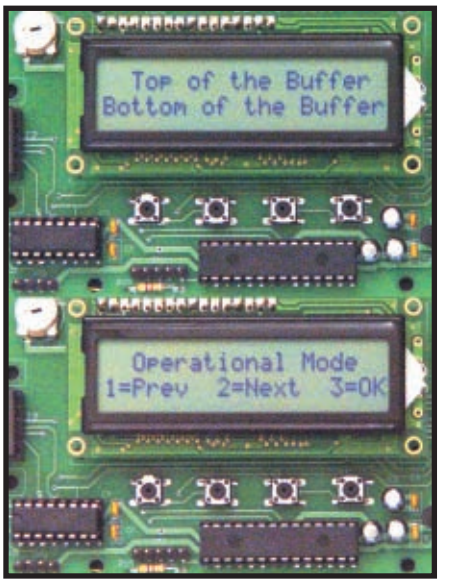


Photo 2—The photo of the top display shows what is displayed for an empty UART buffer. The lower photo shows the start of the properties mode that allows changes to the UART (data rate, number of bits, parity, number of stops, and flow control) on the fly.

Today, 8-bit data is more or less the standard using a 7-bit ASCII set or 8-bit binary data.

Early clocking mechanisms were often unreliable, so a simple parity bit could be appended to the data to help assure correct reception or at least point out an error. This parity bit (a one or zero) was inserted so that the total number of ones (including parity) was an odd or even number. Thus, the parity if used was defined as O (odd) or E (even).

Finally, to give the receiving device time to digest the newly received data and get ready to sync on the next start bit, a number of stop bits (as Mark) are added to each byte of the transmitted data. Stop bits are no more than idle time between the last data bit of one character and the start bit of the next character.

When vast amounts of data are passed, the receiver may need extra time to store data (or spend time executing other timely events). A method of instructing the transmitter to go away was needed. In situations where ASCII data is being transmitted, two special characters can be used to indicate Busy/Ready. To request that the transmitter suspend transmission, the receiver sends an X-OFF, character 0x1. To request transmission resumption, the receiver can send an X-ON, character 0x11. When the data being transmitted is binary the X_ON/OFF characters may be part of the transmitted data and therefore cannot be used. In this case, only hardware flow control can be used. Additional I/O lines clear to send from the receiver (CTS) talks to ready to send at the transmitter (RTS) to accomplish the flow control request.

Function	Setting
Data rate	300
	1,200
	2,400
	9,600
	19,200
	38,400
	250,000
Data bits and parity	Eight bits no parity
	Eight bits even parity
	Eight bits odd parity
	Nine bits no parity
Stop bits	1
	2
Flow control	None
	Software
	Hardware

Table 2—The three buttons and 2 × 20 LCD available to this application can be reconfigured.

ON-THE-FLY FORMATTING

Sometimes you don't know the format or data rate of a serial stream of data. Being able to reconfigure a receiver's UART at any time may come in handy. You can use hardware jumpers to accomplish the configuration task. Because I have buttons and an LCD, this can be done through the program. For most functions, I designate button 1 as Up, button 2 as Down, and button 3 as Select. Where applicable, the mode is presented on the upper line of the LCD, while the keys are labeled using the lower line of the display (see Photo 2). You can choose it using Select, or change modes using Up or Down to display other modes.

The application has two modes of operation: Operational mode and Properties mode. The main loop of the program handles Operational mode (see Figure 2). This is where any data in the UART buffer can be called to the LCD. In this mode, Up (increment) or Down (decrement) changes the `DisplayString` variable, which is used to select a string from the UART buffer. Select is used to change modes.

When Properties mode is selected, you are presented with a number of choices for setting a new data rate, the number of data bits (including any Parity), the number of stop bits, and the flow control. Use Up or Down to display choices. Use Select to make them permanent. Table 2 shows the possible choices made available using this UART.

Changes are made to the UART when you exit Properties mode. This has the added benefit of also clearing the data/display ring buffer. I will discuss the buffer in more detail shortly. The UART can be set to just about any standard (or oddball) data rate depending on the value placed in the appropriate baud rate generator (BRG) register. Based on an 8-MHz oscillator, the available limits of the BRG are approximately 4 to 1 million bps.

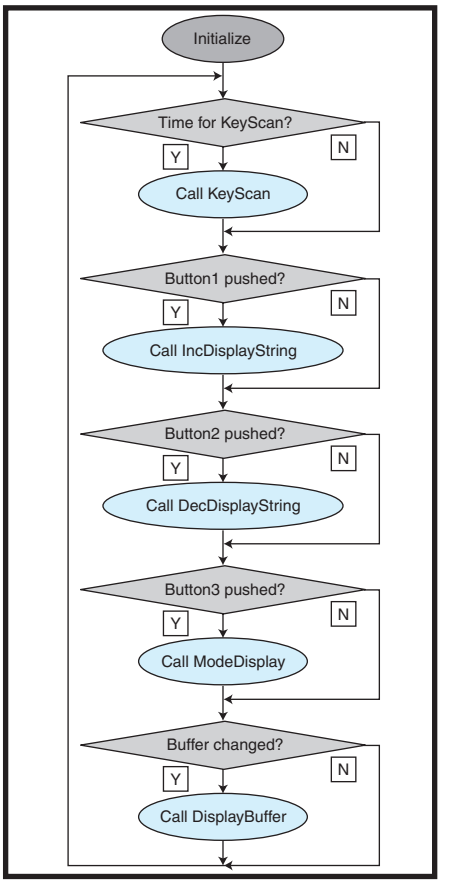


Figure 2—This block diagram shows the basic loop for the application. Enabled interrupts include Timer1 overflow (100 µs), Timer2 overflow (10 ms), and UART RX. When a character is received, it is added to the UART buffer. The UART input buffer gets a carriage return added every 20 characters (the width of the display) if none of the data contains one.

ONE-WAY RING BUFFER

An area in RAM is set aside for data reception. A RAM ring (circular or endless) buffer gives the received data a place to reside while waiting to be operated on by the system. Two buffer pointers (initialized to the buffer's bottom) are used to keep track of the data in the buffer. The head pointer moves toward the top of the buffer as data is added. A second pointer or tail moves toward the head pointer as the data is read out of the buffer by the system. When either pointer reaches the top of the buffer, it is redirected to the bottom of the buffer. (This action is where the buffer gets its name—ring.) As long as the data buffer is large enough and the data continues to be read out of the buffer by the system at an average rate equal to the input rate, the head pointer

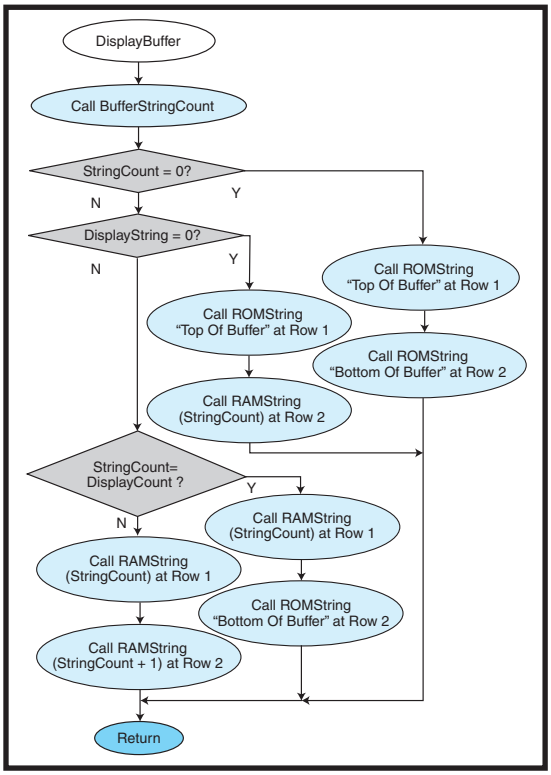


Figure 3—This block diagram shows what data is selected to display on the two LCD lines. Two ROM strings are defined to indicate the buffer's top and bottom. RAM strings come from the UART input buffer and are selected using the variable `DisplayString`.

will never reach the tail pointer. This would mean the buffer is completely filled and is about to overflow into unread data.

For this application, data will never be removed from the buffer. Instead, old data will be overwritten by new data. In essence, you want to hang on to the most recent data. Naturally, the amount of data you can hold depends on the size of the data buffer, so the larger the buffer the better. If new data can continuously overwrite previous data in the buffer, you need a way to determine where the old data ends and the new data begins. This is accomplished by allowing the head pointer to push the tail pointer ahead of it.

From the user's point of view, using a two-line LCD presents a number of issues. Because the window into the buffer is rather limited, you need some way of knowing when you've bumped into the top or bottom of the buffer. Let's assume the buffer is empty (no data has been received). StringCount holds the number of lines in the buffer (zero in this case). A line could be a single-character carriage return (0x0D) or a group of characters up to and including a carriage return. The `BufferStringCount` routine calculates this by counting the number of carriage returns between the head and tail pointers.

Data is sent to the LCD whenever a character is received or Up or Down has been pressed (while in Operational mode). Four routines are used to display buffer data (see Figure 3). The routines are based on the StringCount and the DisplayCount. When StringCount = 0, the string "Top of Buffer" is displayed on the first line of the LCD and "Bottom of Buffer" is displayed on the second line of the LCD.

If the buffer is not empty when DisplayCount (a value between 0 and StringCount that is controlled by the Up or Down buttons) = 0, the string "Top of Buffer" is displayed on the first line of the LCD and the first string in the buffer is displayed on the second line of the LCD. If DisplayCount = StringCount, the last string is displayed on the first line of the LCD and "Bottom of Buffer" is displayed on the second line of the LCD. Finally, if none of these conditions are met, string number DisplayCount is displayed on the first line of the LCD and string number DisplayCount + 1 is displayed on the second line of the LCD.

The LCD limits the view to only 20 characters of any received string, so how can you view longer strings? To simplify the operations, I discounted horizontal scrolling. Instead, I opted to reduce the size of incoming strings to 20 characters (the width of the LCD). Thus, as characters are received and placed in the buffer, a counter (CharCount) keeps track of the number of received characters. If CharCount reaches 20 and the character is not a carriage return, then an extra 0x0D character is added to the buffer and CharCount is reset to zero. Checking for whole words is not performed in this application. Although it might be easy to keep track of the last <space> character and replace it with a 0x0D if the CharCount reaches 20 and the character isn't a carriage return or a <space>.

Not all of the characters in the ASCII character set are printable. For instance, characters 0x00 to 0x1F are control characters (e.g., X-ON and X-OFF) used for other purposes. When received, they are tossed into the bit bucket (eliminated) except for 0x0D, which normally indicates the end of a string. Note that most LCDs have a <space> (blank) located at these positions in their character tables. If they are not filtered out, they will print as blanks. While they aren't informative, you might decide to include them for display purposes. You could even program the locations in the LCD character table to display special characters that you provide.

TAKE MY PORT

Using circuitry from a future From The Bench project, I was able to keep a new contracted product design on track. I needed to monitor RS-232 output, but using this circuit's 2 × 20 character LCD seemed like an exercise in futility. With a little creativity and the simple user interface, I now have a useful tool that allows you to scroll through its serial port's receive buffer. So, go ahead, take away my serial port. See if I care!

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.*

PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/219.

SOURCES

PIC24FJ64GA002 Microcontroller
Microchip Technology, Inc.
www.microchip.com

NHD-0220GZ-FSW-GBW-L LCD Character module
Newhaven Display International
www.newhavendisplay.com

Universal Machine Control

How Hardware Emulation Leads to Prototype Development

Hardware emulation is an effective phase of the prototyping process. The purpose of this project was to build a robust hardware platform to run the rule sets for old pinball machines. Using the PinMAME program, the design emulates the hardware for almost 500 pinball machines and runs the rule sets.

The electronic gaming industry has never been more popular than it is today. Modern consoles are making the gaming experience more realistic. This begs the question: Why did we design and build a universal pinball machine controller?

The simple answer is that while the conversion of discrete electronics into embedded systems may have a wide range of applicability, there will always be a market for vintage systems. Beautiful casings and hardware from 1970s electronics now languish in limbo because many components are unavailable. There is a large commercial opportunity in aesthetic gaming.

You can trace back the origins of pinball to the latter part of the nineteenth century. In fact, in 1871, a U.S. patent was awarded to British inventor Montague Redgrave for an improvement to an existing table game called Bagatelle, which used a "plunger" to fire a small metal ball on a playing surface. Although there were no electronics on his table, it is widely accepted as the birth of the pinball table. This primitive form of pinball doesn't resemble the form that you have come to know. It wasn't until 1947 that an innovator in the industry, Gottlieb, added player-controlled flippers, which introduced a skill factor to the game. The introduction of solid-state electronics in the early 1970s brought about the golden age of pinball. The microprocessor enabled games to be built with complex rule sets with onboard sound and speech. Pinball machines laid the groundwork for the video-game era that was quickly taking over the electronic gaming industry.

Like every industry, the gaming

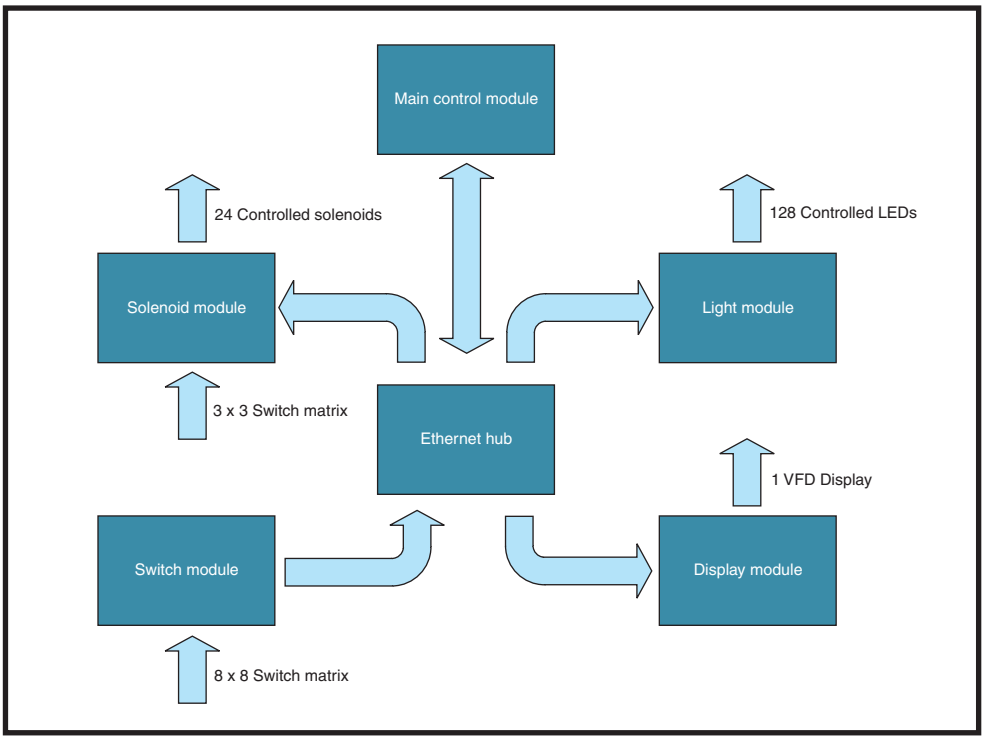


Figure 1—This is a modular approach to the functionality of the pinball table. The control module acts as a client and all other modules are servers on the game network.

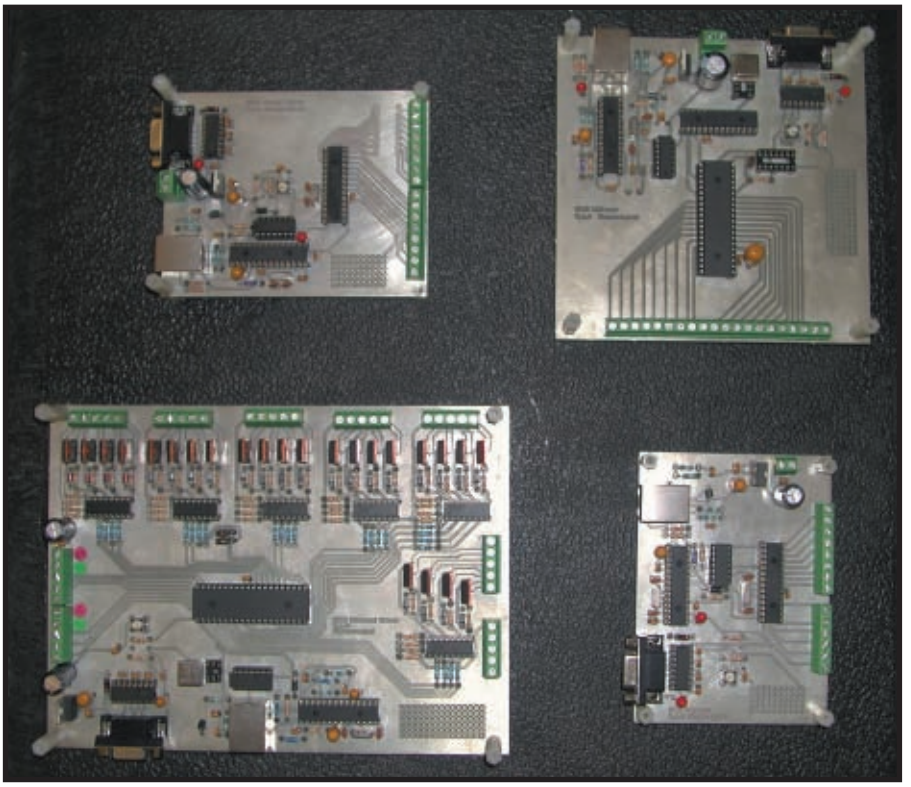


Photo 1—Each of the four server modules was fitted with screw terminals, making it easy to interface to existing hardware. Clockwise from the top left are the switch module, the LED driver, the VFD driver, and the solenoid driver.

industry is subject to the fundamental rules of economics. Early arcade games such as Pac-Man and Asteroids required relatively little maintenance, took up less space, and simply made more money than pinball machines. Since the introduction of the first arcade games, pinball has been on a steady decline, leaving many tables rusting and out of repair. Today, it is hard to find replacement electronic boards for many old tables because manufacturers have stopped producing them.

The dwindling availability of parts, a broken pinball machine, a penchant for gaming, and an emulation program called pinball multiple arcade machine emulator (PinMAME) were the inspiration for our universal pinball machine controller (UPMC). We developed the prototype for our final project in the Computer and Electronics Engineering Technology course at Camosun College in Victoria, British Columbia.

THE PROJECT

For our prototype, we used a Gottlieb System 80 pinball machine called Panthera. The machine was manufactured in 1980 and was not functioning when we began.

When we started this project, our goal was to build a robust hardware platform that used PinMAME to run the rule sets for old pinball machines. We wanted the design to be open-source so anyone could add functionality after we finished our original prototype. Any pinball enthusiast will note the wide variety of playfield parts found on machines that are not covered by this project.

Because PinMAME already knows how to play pinball, half of the challenge was simply to input information into the emulator and direct its outputs. The other half of the challenge was to build a flexible hardware platform that could adapt to many pinball machines and their various playfield parts. For our prototype, we decided to tackle four components found on any pinball table: switches, displays, lamps, and solenoids.

UNIVERSAL SYSTEM DESIGN

One of the problems we encountered when trying to design a universal system was how to separate the functionality of the pinball table into manageable and flexible parts. Each pinball table is unique and contains different numbers of switches, displays, lamps, and solenoids. For this reason, we chose a modular system that uses the TCP/IP protocol to communicate. Figure 1 shows the modular system, which is the essence of our UPMC.

We used TCP/IP as the system's backbone for a few reasons. The protocol had to run fast enough that latency was not an issue. Most modern protocols run at a speed that is sufficient for our application, but the client/server relationship of TCP/IP lent itself well to the UPMC. We had no latency issues while running our system using TCP/IP at 10 Mbps. Most importantly, the brain of the system is a PC, so we had to choose a protocol that would interface easily to a computer. Using TCP/IP enabled us to simply connect our modules together using a hub and standard Ethernet cables.

In the future, we hope to take advantage of the ability of our boards to connect to the Internet to implement a few features that we were not able to cover during our original design. Adding an external EEPROM to each board would give us the ability to store a web page on each board (see Photo 1). The web page would act as a visual aid, and we could use it to map the I/O and ports on the chip to simplify the boards' installation and configuration. We also hope to implement an online scoreboard where any pinball machine running the UPMC system could update the online scoreboard automatically.

Although the decision to split functionality into modules multiplied the amount of hardware used, it simplified the design. A universal system should be designed in a generic way. Therefore, a modular design seemed most logical.

Each module has a defined and limited functionality, which can be combined with other modules to fulfill the requirements of a specific pinball machine. The modular system enabled (and required) us to complete blocks of our prototype and helped

facilitate our development process.

PinMAME

PinMAME is a derivative work of the multiple arcade machine emulator (MAME), which was originally designed to emulate the hardware of arcade systems in software. The emulator acts as a software layer between the original game and the platform it's running on. It does so by interacting with ROMs that contain all of the individual game programs, graphics, and sounds.

Currently, PinMAME can emulate the physical hardware of more than 480 pinball machines. In our project, PinMAME interfaces the ROMs to our custom hardware. Although you can download PinMAME and MAME and the full source is available, they are not considered open-source software because commercial use and redistribution is prohibited due to the fact that many game ROMs are still covered by copyright.

HARDWARE EMULATION

Interfacing to the PinMAME hardware emulation and getting our system to run without latency was the single largest challenge for this project. From the start, we decided that there was an acceptable amount of latency that would not be noticed in a relatively slow electromechanical system. For most events, a latency of 100 ms would not be noticed by the player with the exception of the main playfield bumpers and flippers, which were handled directly by the solenoid module that we will discuss later. The original hardware was running at a frequency of 2 MHz, giving us a big advantage in terms of processing power. With each module running at 40 MHz and the backbone of the system running at 10 Mbps, we found that most transactions averaged less than 10 ms in an isolated situation. As long as our game network did not get flooded with excessive communications, a latency of less than 100 ms was easily attained.

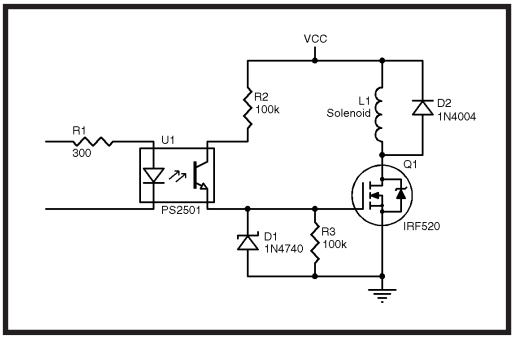


Figure 2—This is the solenoid driving circuit. L1 is the actual solenoid. D2 is mounted across the coil on the playfield. The first version of our solenoid module contains 24 of these circuits.

Therefore, the challenge became figuring out how to gather an update from the emulation and send it out to our modules in an efficient manor. This was done by gathering the information from its peripheral interfacing chips, the MOS Technologies RAM input output timer (RIOT) 6532 chips (specific to the Gottlieb System 80), and sending out an update to our modules.

To get the UPMC to work on a specific table, we added functionality to the PinMAME source code. Specifically, the files that contain the machine code for its peripheral interfacing

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

Got Serial, Need Network?


Volume Discounts Available
gridconnect™
www.gridconnect.com
+1 800 975-4743

Listing 1—Writing information to the MAX6954 is easy. The MAX6954 must receive a digit select command (0x68) preceding the data for eight lights. `LEDdata [ ]` holds all of the information received in our asynchronous packet from the control module.

```
//write data to P0 and P1 digit 0a
   LEDDRIVER_CS_IO =  0;     //set Chip Select lo to select the MAX6954
   SSPBUF=0x68;              //write 0x68 which is the command to write the same data to
                             //plane 0 and plane 1
   while(!PIR1bits.SSPIF);   //wait until the SPI module has finished writing
   PIR1bits.SSPIF  =  0;     //reset the SPI interrupt flag for the next transmission

   SSPBUF=ledData[1];        //write desired data
   while(!PIR1bits.SSPIF);   //wait until SPI module has finished writing
   LEDDRIVER_CS_IO = 1;      //set chip select high to deselect MAX6954
   PIR1bits.SSPIF = 0;       //reset the SPI interrupt flag for the next transmission
```

chips had to be altered.

On the Gottlieb System 80, three RIOT 6532 chips were used to interface to its peripheral devices. It used one chip for the displays, one for the switches, and one for solenoids and lights. The PinMAME source code supports this series of chips. The files pertaining to this chip were found in /src/machine/riot6532.c.

The RIOT 6532 uses two bidirectional 8-bit ports to address and control the playfield. The emulation of the RIOT 6532 does exactly what the original chip did in hardware. The raw data found on the outputs and inputs of these ports are where we can interact with the actual game. By providing information from our physical playfield to the inputs of its switch RIOT and decoding the outputs of its display, lamp, and solenoid RIOTs, our hardware platform can interact with the game ROM.

MAIN CONTROL MODULE

The main control module is a PC motherboard with an OS dedicated to running PinMAME. For our project, we used an old PC with a sound card and a 10BaseT network card. An Intel Pentium III 800 with 128 MB of RAM was sufficient to run the hardware emulation without a visual front end.

The main control module acts as the client. All of the other modules act as servers on the game network. We used sockets to get information in and out of PinMAME and into our TCP/IP network. Sockets enable the control module to prioritize communications and ensure that only one server is talking to the client at a



given time. This is required because each server module has unique communication and timing demands on the network.

The switch module is the only module that inputs information into the emulator. This is done by frequent inquiries by the control unit (the client) to the switch module about the status of the switches. The maximum time between requests has to be 10 ms because this is the time required for a valid switch press. For each request, the switch module returns a packet containing updated switch information as an array of 8 bytes. PinMAME emulates an active high switch matrix. Each byte received from the switch module contains the active level of eight switches. As PinMAME cycles through each column in its switch matrix, the appropriate byte from our playfield is assigned to the inputs of the RIOT.

The solenoid, LED, and display modules only receive information from the main control module and are never required to send input to PinMAME. The display modules receive asynchronous packets and are addressed by the control module only when the score or the game status has changed. The light driver module receives a synchronous update from the main control module every 100 ms, so the playfield lights don't "lag" behind game play. The solenoid module receives an asynchronous packet from the control module when program-controlled solenoids are activated.

The control module also handles the in-game sounds. PinMAME already has the ability to play the sounds stored in the game's sound ROM, so all we had to do was enable the emulator to run and enable the PC to output the sounds via the sound card.

HARDWARE MODULES

We designed each of the four hardware modules with the Microchip Technology PIC18F*xxxx* series of chips. With an external 10-MHz oscillator, and by enabling the internal phase-locked loop, a PIC18F2620 can operate at 40 MHz. This series of chips also has a built-in SPI module that can be set up in code. The speed of the SPI bus is set by the PIC18F2620. It is 10 MHz ($f_{OSC}/4$). This was needed for communication with the Ethernet interface and the chip that we chose for the light module.

For development, a Maxim Integrated Products MAX232 chip connects the PIC18F2620 to the PC via an RS-232 serial port. The MAX232 uses a 5-VDC supply to internally generate a ±10-VDC supply for RS-232 communication. Each board contains a nine-pin female serial port connector.

Each module has the ability to communicate on a network via a Microchip Technology ENC28J60 stand-alone Ethernet controller and a Microchip TCP/IP stack version 4.16. The ENC28J60 communicates with the PIC18F2620 using an industry-standard SPI. Each module, which is assigned a unique IP and MAC stored in the PIC18F2620 EEPROM, acts as a server on the TCP/IP network.

SOLENOID DRIVER MODULE

The solenoid driver module is the only module in the UPMC that





requires high voltages and currents. For this reason, we used optical isolation to separate our high-voltage circuitry from our control circuitry. The high-voltage output circuitry was designed to withstand continuous currents of up to 4 A and uses MOSFETS to switch the high-powered coils.

The solenoid-driver module accepts input from the playfield and the main-control module and powers all of the playfield solenoids. The playfield solenoids were broken into two categories: software-controlled and direct-controlled.

The direct-controlled solenoids are the flippers, bumpers, and pop-bumpers. They fire immediately when the ball comes in contact with the playfield part closing a switch or the switch is closed manually in the case of the flippers. To minimize latency, these switch hits are sensed directly by the solenoid board via a three-by-three active-low switch matrix. This was required to limit latency caused by a transaction with the main control module. It was also required because most pinball tables operate in a similar fashion, which means the hardware emulation would not support it.

The program-controlled solenoids are all of the other solenoids on the board, including the ball return and the drop-bank resets. The timing is controlled by the main control module and ultimately by the game rules.

Both program- and direct-controlled solenoids are driven with a common circuit (see Figure 2). We wanted this driving circuitry to be adaptable to various voltages used on different pinball machines. The solenoids on our table required 24 VDC, but some tables require higher voltages of up to 50 VDC. When the input matrix senses a hit or a packet is sent by the main control board, the chip drives the input of a PS2501 optical isolation chip for 100 ms.

The PS2501 provides optical isolation of up to 5,000 V_{RMS} to protect the PIC18F2620 from the high-power driving circuitry. The PS2501 is a transistor output optical isolation chip with a high collector-to-emitter voltage of 80 VDC. This collector-to-emitter voltage enabled us to use a supply voltage to drive the solenoids up to 80 VDC. The output of the PS2501 is connected as shown in Figure 2. D1 is a 10-VDC Zener diode, 1N4740, used to protect the gate of the power MOSFET. When the circuit is activated, the gate of the International Rectifier IRL520 MOSFET is pulsed with 10 VDC. This turns it on and enables current to flow from drain to source through the solenoid. Resistor R3 is a bleed resistor used to discharge the energy stored by the gate capacitance of the MOSFET after it has fired.

The first version of our solenoid-driver module can power 24 playfield solenoids. The circuitry on this board could be

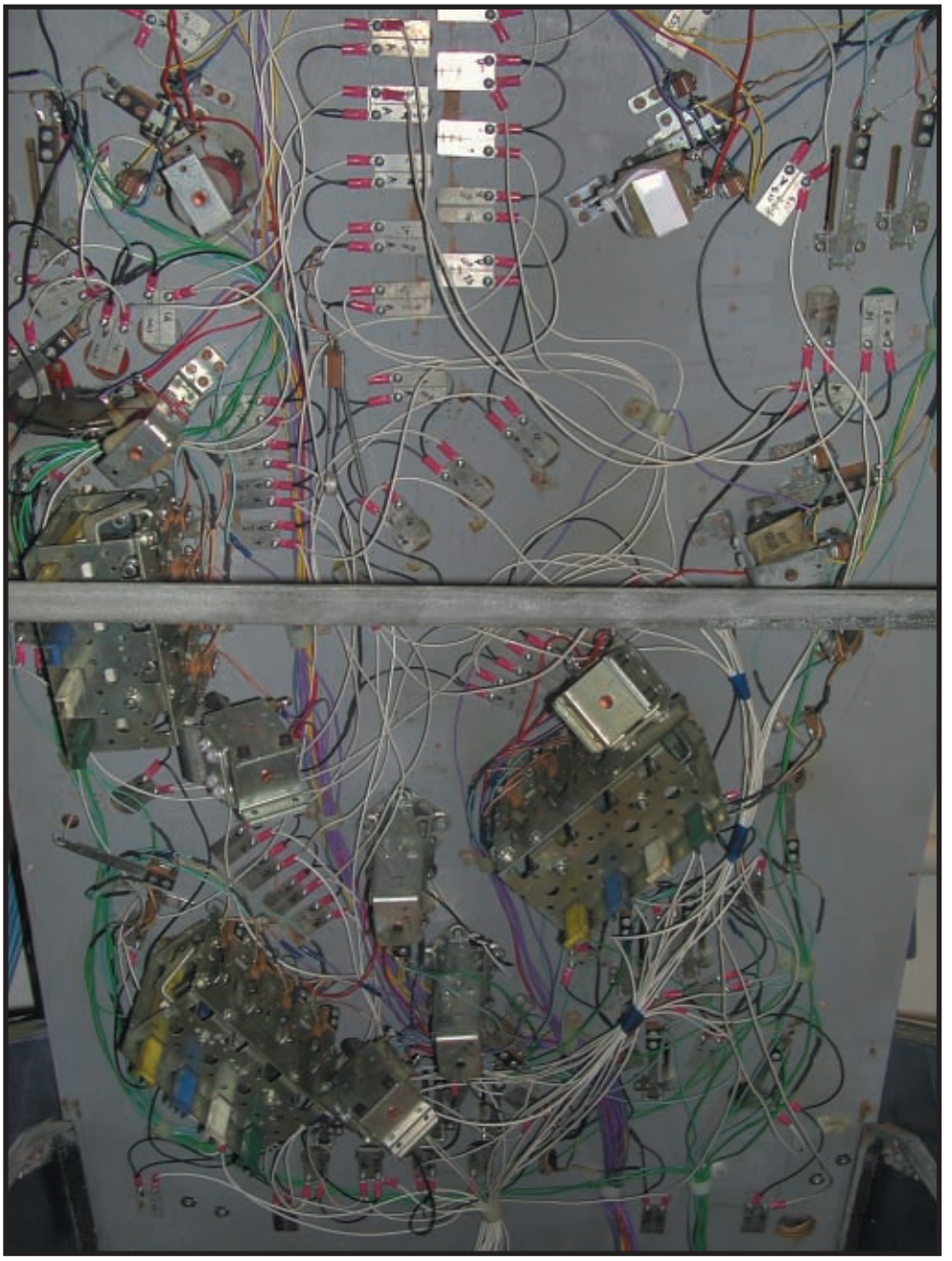


Photo 2—We rewired the entire table with custom LED boards (down the middle). Using ring terminals makes it easy to replace lights when they burn out.

Listing 2—Check out the switch debouncing routine. Using code looping to debounce switches allowed us to debounce all 64 switches simultaneously.

```
LATA=0xFE;                                  //set output lo

Nop();
                                            //make sure port has had a chance to change
if(PORTBbits.RB0 == 0 && sReady[0]&0x01)
                                            //check if input is lo and switch is hitable


{

  sCount[0]=sCount[0]+1;
                                            //increase debounce count
  if(sCount[0]>=DEBOUNCEDELAY)
                                            //if count greater than debounce delay, valid hit
  {

     sNum[0]|=0x01;
                                            //activate valid switch hit so detected switch hit is
                                            //sent out over network
     sReady[0]&=0xFE;
                                            //say switch is no longer hitable
  }

}

else if(PORTBbits.RB0)

{

     sCount[0]=0x00;
                                            //reset debounce count
     sReady[0]|=0x01;
                                            //say switch is hitable
}
```

used to drive other devices, such as magnets and servos found on many pinball machines.

LIGHT DRIVER MODULE

When designing the light module, we replaced all of the incandescent bulbs with LEDs. We did this for two reasons. One, LEDs are more efficient and will last longer than traditional incandescent bulbs. And two, it enabled us to use a sophisticated IC, the Maxim MAX6954 to control them (see Listing 1).

The MAX6954 operates off of a logic supply of 5 V and supplies up to 40 mA per LED. Compared to the original incandescent bulb in the machine, which required 6.3 V at 150 mA, the difference in power consumption is considerable. One drawback to using LEDs is their dispersion pattern, which tends to illuminate directionally as opposed to the halo created by an incandescent bulb. We chose LEDs with a 140° viewing angle to mimic the effect of an incandescent bulb. Some manufactures have created LEDs that behave like incandescent bulbs, which would be suitable for this application. For our prototype, trying these LEDs did not fit into our schedule or budget.

The MAX6954 is an LED driver chip that can control up to 128 discrete LEDs. The MAX6954 uses 19 I/O pins and a technique called Charlieplexing to drive the LEDs. Charlieplexing was developed by Charlie Allen, a Maxim employee, as a way of driving large numbers of LEDs with a reduced number of I/O pins relative to traditional multiplexing. This is done by using pins (eight in our case) as both inputs and outputs. Each of the eight bidirectional pins sinks the current for 16 LEDs at any given point in time. The MAX6954 uses an external resistor to set the current, which flows through each discrete LED (see Photo 2).

The MAX6954 communicates with the PIC18F2620 via a SPI, which is already used on the board for the ENC28J60 chip, requiring only one extra I/O pin. A typical update from the main control module to the light board is a TCP/IP packet containing 16 bytes of data. Each bit in this packet corresponds to a playfield light and is either a one or zero corresponding to an on or off. The bytes are then written over the SPI bus to the MAX6954 one byte at a time. The MAX6954 expects 2 bytes of data in each communication transaction. The first represents a group of eight LEDs and the second represents which LEDs in the group are on. The PIC18F2620 has to write 32 bytes of information to the MAX6954 to update all 128 of the playfield lights. Operating at 10-MHz latency is not an issue because the write time takes less than 5 µs. The light driver module updates the playfield lights every 100 ms when it receives an update from the main control module.

SWITCH MATRIX MODULE

The switch-matrix module is the simplest of the four modules. It

implements an active-low switch matrix to scan and debounce switch hits. We used a PIC18F2620 with two eight-pin ports to give us 64 switches. For the Gottlieb System 80, a 5 × 5 switch matrix would have been enough because it is common on pinball tables to tie switches on the table with the same point value together. The switch-matrix module holds an array containing the current status of the switches. When the main-control module inquires for the status of the switches, the switch module returns a packet containing the 8 bytes of data required for 64 switches. Each bit in the packet corresponds to a switch on the playfield. It is up to the main control module to decide what to do with the switch information (see Listing 2).

For systems with mechanical switches, debouncing routines are required to distinguish between noise and valid switch presses. In order to be able to debounce all of the switches simultaneously, delay routines are not used. Instead, when a switch is sensed, it is assigned a count value, which is decremented each time we loop through the scanning routine. If at any time during the count the switch returns to an idle state, the count resets. When the switch has been active for the entire count, the switch matrix status array is updated. A switch can not be "hit" again until it has returned to an idle state. For the switch matrix module, debounce timing is set by the count value. The count value is defined as a macro, making it easy to change the debounce timing.

An interesting characteristic of the pinball table switch matrix is the nonmomentary switches found on the drop targets. Once hit, the switches are always down, which can cause ghosting in a switch matrix. To solve this problem, blocking diodes were required on all non-momentary switches.

DISPLAY MODULE

The display module is the most difficult module to design for a universal system because there are so many different displays on pinball machines. Thus, our display module works only on machines using VFDs. The displays on the Gottlieb System 80 are eight-segment six-digit VFDs. The VFD modules contain onboard driving chips, which require logic levels to control. So, our display module can interface to any eight-segment six-digit VFD with onboard driving circuitry. Most VFDs, including the ones on the System 80, require a DC voltage as well as an AC voltage to excite the filaments. Our system needed 60 VDC and 5 VAC for the score displays and 42 VDC and 3 VAC for the game-status display. These voltages were provided directly from the power supply.

Communication between the main control module and the display module is simple. PinMAME emulates the hardware for each machine, so it can provide the appropriate digit data in the form of 6 bytes. To avoid sending excessive packets on our network, we wait for a change in the score before a display module is addressed. Even though there are four six-digit score displays, PinMAME treats them as two 12-digit displays. Therefore, we must distinguish between which display in the group of two is being talked to. PinMAME uses chip select pins to determine which display is being addressed. When the first group is selected, the first 6 bytes of data are sent to the first display board and the second 6 are sent to the second display board. Each byte corresponds to a value on a corresponding display digit.

The other two score displays are treated in a similar way. Our display module then parses the digit information to provide the VFD modules with the appropriate segment and digit data. Our PIC18F2620 uses six pins on PORTA for digit data and all eight pins on PORTB for segment data. The VFD's digits are multiplexed using positive logic, and 1-ms digit-select pulses are required to light the displays.

THE FUTURE

We already have several ideas about improving the capabilities of our UPMC because our first prototype has some limitations. The initial design is flexible enough to be tested on many machines from the same era but lacks the functionality to be truly universal. Multiple modules could be easily used on tables requiring larger numbers of controlled playfield parts. New modules could be added to control the functionality of playfield parts found on newer machines. As long as new functionality is built with the same design principles in mind, a universal platform could be a reality.

PinMAME is updated on a regular basis. Each update adds more pinball tables to the list of machines it can emulate. Each table, or series of tables, requires us to alter the PinMAME source to interface to our platform. Once the raw data is extracted from the emulation, the UPMC platform can take care of the rest.

Phil Laird (lucaboq@hotmail.com) is a recent Computer Engineering Technology graduate. He is interested in computer gaming, embedded systems, and software engineering.

Jeff Shewan (jeff.shewan@gmail.com) is a recent Electronics Engineering Technology graduate. He is interested in computer gaming and the audio engineering side of electronics.

PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/219.

RESOURCES

PinMAME emulation home page, http://pinmame.retrogames.com.

S. Young, "The Pinball Resource," www.pbresource.com.

SOURCES

IRL520 MOSFET
International Rectifier
www.irf.com

ENC28J60 Ethernet controller and PIC18F2620 microcontroller
Microchip Technology, Inc.
www.microchip.com



Cable Shielding Experiments

Robert recently came across an old book about electronic design. After reading about the topic of noise coupling with various cables and wiring setups, he reproduced the cable shielding experiments described in the book. This month, he shares the results.

Welcome back to the Darker Side. I always do my best to find a topic that coincides with the monthly *Circuit Cellar* theme, but this time I failed. I will not talk about signal processing, and I apologize. But I have an excuse. I just rediscovered a book in my technical library titled *Noise Reduction Techniques in Electronic Systems* by Henry W. Ott (AT&T Bell Labs). Although the book is 40 years old, I was amazed at its timeliness. It is the kind of book that any electronic design engineer must forever keep close by because the content is too fundamental and important to be outdated as technology changes. Its first edition was published in 1975, and then updated in 1988. In addition to chapters on grounding, filtering, shielding, ESD, thermal noise, PCB layout rules, and more, Ott devoted a chapter to cabling issues. What grabbed my attention was that he presented not only theories but also actual experiments on noise coupling with different kinds of cables and wiring setups. I thought this was definitively an interesting "dark side" subject, so I decided to try to reproduce the spirit of these experiments. This month, I'll share my results.

THEORY REFRESH

Imagine that you have a wire conducting a low-level signal—say, from a sensor—and another nearby wire conducting a noisy signal—say, a digital signal. You know from experience that some noise will probably appear on your sensor signal. But how? As long as the signal and noise frequencies stay low, two main coupling mechanisms could be involved: capacitive coupling and inductive coupling. Capacitive coupling is the interaction of electric fields between the two wires. Inductive coupling is the interaction of magnetic fields between the two circuits. At higher frequencies, or at a longer distance, a combination of both effects can lead to electromagnetic coupling. But I won't cover this. I will focus on short-distance effects.

Let's start with capacitive coupling. When two wires are close to each other, there is a parasitic capacitance between both wires, which can be depicted on an equivalent electric circuit (see Figure 1). The parasitic capacitance depends on the length, distance, and geometry of the wires. When associated with the loading impedance of your signal wire, the capacitance implements a high-pass RC filter between the noise source and your signal receptor because the capacitance is in series between the noise source and the receiver circuit. Look at the formula in Figure 1's caption. First, the noise voltage added to your precious signal will be proportional to the noise source voltage, so the more noise you have, the more noise you get. But it also will be proportional to the noise frequency, so capacitive coupling is mainly a problem with high-frequency noise. In addition, it is inversely proportional to the signal load impedance, meaning that capacitance coupling noise will be a major problem with high-impedance signals. Last but not least, it is proportional to the value of the paracitic capacitor: the higher the capacitance between the two wires, the more noise you get. The good news is that this capacitive coupling is quickly reduced when the distance between the two wires is increased. The other good news is that using shielded cables theoretically removes any capacitive coupling as long as the shield is grounded at least at one point, except for the portion of the cable that will inevitably extend beyond the shield. So, capacitive coupling is mainly a problem when two unshielded cables are close to each other, such as between wires of a ribbon cable.

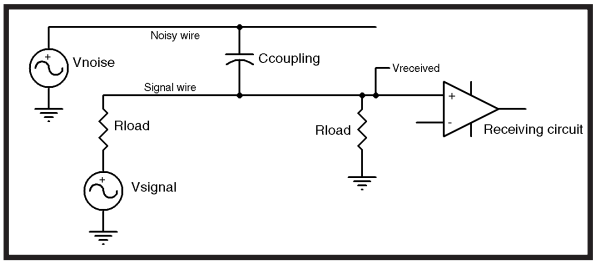


Figure 1—This is the equivalent circuit of a capacitive coupling between two wires. The noise voltage added on $V_{RECEIVE}$ is calculated as $2\pi F_{NOISE} V_{NOISE} R_{LOAD} C_{COUPLING}/2$. Capacitively coupled noise is then a problem mainly with high-frequency noise, high impedances, and closely spaced wires.

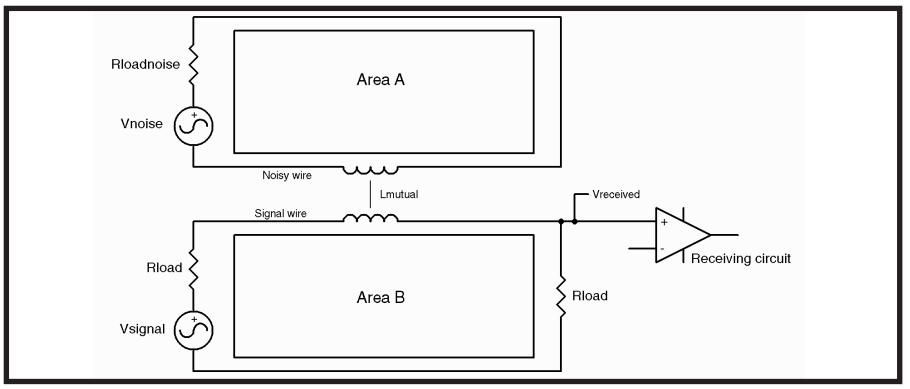


Figure 2—For inductive coupling, the noise is coupled on the signal wire as though it is a transformer. The coupling factor, or mutual inductance, is mainly dependent on the areas A and B of the two coupled circuits, their distance, and their mutual orientation.

Inductive coupling is a bit more nasty and difficult to imagine. When a current flows through a circuit, in this case the noise generator circuit, there is a corresponding magnetic flux generated proportional to the current going through the noise generator circuit. The magnetic flux will generate a current in any other nearby circuit, proportional to the magnetic flux level, proportional to the noise frequency, and proportional to the mutual inductance between both circuits (see Figure 2). The mutual inductance is roughly proportional to areas A and B of the two circuits, which are the areas between each signal and its corresponding return path (see Figure 2). That's why the current return path should always be as close as possible to the signal path—to reduce the inductive loop area.

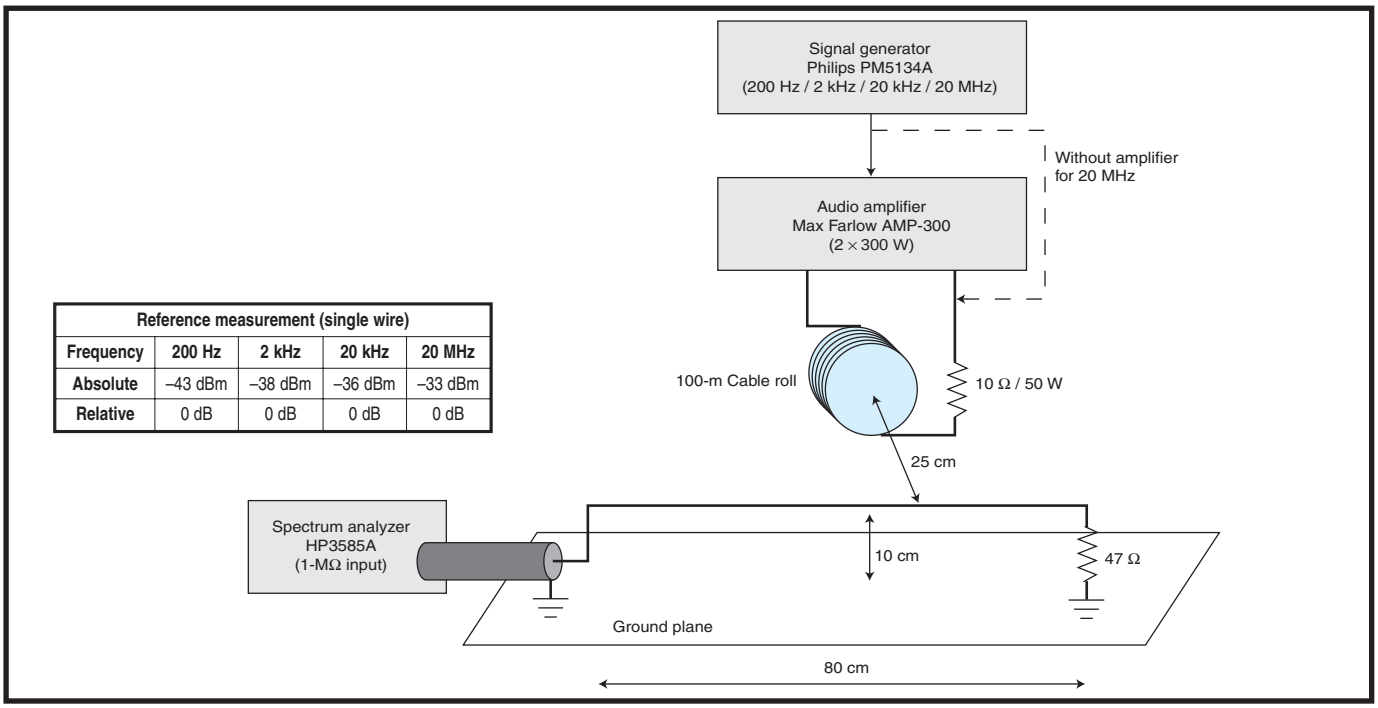
Now the bad news. The coupling acts like a transformer with windings in series with both circuits, so the noise generated on the signal circuit is a voltage added in parallel to the wire. This means it is independent of the signal load impedance. In other words, when there is an inductive coupling, the generated noise voltage level is roughly the same for low-impedance or high-impedance signal circuits. (The contrary is true with capacitive coupling.)

There's even a way to distinguish between capacitive and inductive coupling. If you change the load impedance and the noise level stays roughly the same, you have inductive coupling. The other bad news is that a shield may not reduce, even theoretically, an inductively coupled noise especially if it is grounded only at one point. I will cover this topic in more detail later in this article. However, there is an efficient solution to inductive coupling: twisted wires. A twisted wire pair enables you to keep the net area between a conductor and its current return path at zero, because the polarities are out of phase at each half-twist, at least if the currents are perfectly balanced.

EXPERIMENTAL SETUP

I've covered the theory. Now I'll focus on the process of testing inductive coupling. I will also describe good shielding and wiring strategies to reduce noise coupling.

It took me just 30 minutes to build the small test bench illustrated in Figure 3. First, I needed to generate a strong magnetic field. I used an old Philips/Fluke PM5134A lab signal generator and tuned it to frequencies from 200 Hz to 20 kHz. To amplify the signal, I hooked a 300-W audio amplifier and connected the speaker output to a coil



Reference measurement (single wire)				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–43 dBm	–38 dBm	–36 dBm	–33 dBm
Relative	0 dB	0 dB	0 dB	0 dB

Figure 3—The reference configuration uses a single wire exposed in a strong magnetic field with a return path through a ground-plane situated 10 cm below the wire. This configuration is our 0-dB reference by definition.

made with a 100-m roll of cabling wire. I added a 10-Ω series power resistor to reduce the risk of blowing out the amplifier. I also used the setup with a 20-MHz frequency—this time with a direct drive of the coil from the signal generator because an audio amplifier doesn't have such a bandpass.

I then used a 1.20 m × 50 cm metallic grid to build a reasonably good ground plane and placed a wire 10 cm above the ground plane and 25 cm away from the noise-generating coil. I connected a 47-Ω resistor between the wire end and the ground plane, emulating a 50-Ω null signal source, and used a spectrum analyzer to measure the level of noise received by the wire through the other end. Photo 1 shows the actual installation.

Next, I switched on the power and measured the noise level at different noise frequencies. The measured noise power levels ranged from –43 dBm at 200 Hz and –33 dBm at 20 MHz (see Figure 3). Just remember that 0 dBm = 1 mW, –10 dBm = 100 µW, –20 dBm = 10 µW, and so on. The formula is:

$$P\,(\text{in W}) = 1\ \text{mW} \times 10^{\text{dBm}/10} \quad [1]$$

So the received levels were ranging from 0.05 µW to 0.5 µW. We'll use these values as a reference to compare other wiring schemes. These powers are "0 dB." Let's name this reference "configuration A."

COMPARATIVE RESULTS

Imagine that this noise is far greater than what you can tolerate on your



Photo 1—This is the actual test bench. The frequency generator is on the top and set to 1.9987 kHz. It is driving the 300-W power amplifier on the right (the one with blue LEDs). The amplifier output then excites the big coil simply made with a 100-m white wire roll. This coil generates a strong magnetic field, which is received by a simple wire. This victim wire is positioned 10 cm above a ground plane and connected to ground through a 47-Ω resistor on the right and to a spectrum analyzer on the left. The boxes are empty.



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/

CIRCUIT CELLAR®

back issues available as

Searchable Archives on CD-ROM

CD-ROM #12 2007 Issues 198-209
CD-ROM #11 2006 Issues 186-197
CD-ROM #10 2005 Issues 174-185
CD-ROM #9 2004 Issues 162-173

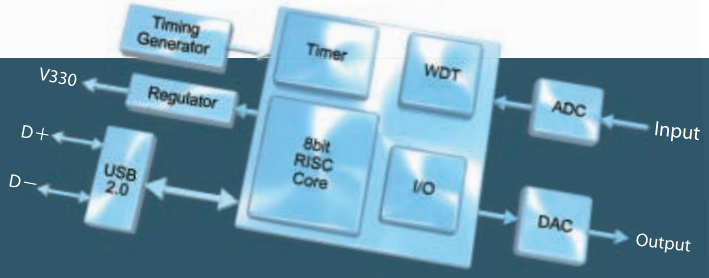
Order Online:
www.circuitcellar.com
or call 860.875.2199



HOLTEK
www.holtek.com

Easy USB MCU Integration

Easy resource integration with these USB microcontrollers



Holtek has released a series of embedded USB interface 8-bit microcontrollers complete with fully integrated SPI interface, A/D converter, D/A converter etc. functions. With a lowest operating voltage of 2.2V these devices are suitable for use in computer peripherals, digital photo-frames, toys, USB telephones, USB speakers etc. Their extensive internal functions and their comprehensive device range add up to enhance the competitiveness of your products.

- Wide operating voltage: 2.2V ~ 5.5V
- Program Memory Capacity: 2K ~ 8K
- USB Endpoints: 2 ~ 6
- Supplied in both OTP and Mask versions

Part No.	VDD	Program	Data	Endpoint	I/O	A/D	D/A	SPI	VDD I/O	RTC
Low Speed										
HT82M99E	3.3 ~5.5V	2K x 15	96 x 8	2	12					
HT82M9AE	3.3 ~5.5V	4K x 15	224 x 8	3	16					
HT82M9BE	3.3 ~5.5V	8K x 16	224 x 8	4	20					
HT82K94E	2.2 ~5.5V	6K x 16	224 x 8	4	40					
HT82K95E	3.3 ~5.5V	4K x 15	160 x 8	3	32					
HT82K96E	4.4 ~5.5V	4K x 15	160 x 8	3	32	V				
HT82J97E	4.0 ~5.5V	2K x 14	96 x 8	2	20	V				
Full Speed										
HT82A520R	2.2 ~5.5V	4K x 15	160 x 8	4	24			V	V	
HT82A523R	3.3 ~5.5V	4K x 15	192 x 8	4	40			V	V	
HT82A620R	2.2 ~5.5V	4K x 15	160 x 8	4	24	V		V	V	
HT82A623R	2.2 ~5.5V	4K x 15	160 x 8	4	32	V		V	V	V
HT82A821R	3.3 ~5.5V	2K x 15	192 x 8	3	8		V			
HT82A822R	3.3 ~5.5V	4K x 15	704 x 8	3	24		V			
HT82A834R	3.3 ~5.5V	4K x 15	192 x 8	5	44	V	V	V		
HT82A836R	3.3 ~5.5V	8K x 16	352 x 8	5	24	V	V	V		

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

Accelerate your Design Time with EmbeddedDeveloper.com



Speed up your design time during the critical evaluation phase of your project by avoiding locating and comparing devices from different manufacturers. We make it easy: just one visit to **EmbeddedDeveloper.com** will shave hours or days off your schedule!

Embedded Developer's simple navigation and intuitive search engines help you quickly locate, compare, and evaluate thousands of different microcontrollers from different manufacturers--without knowing part numbers.

Compare all devices by their performance and features in seconds, download a data sheet, find and buy development tools, or link to a distributor for an RFQ or chip sample--without ever leaving the site!

Embedded Developer is the only site in the world where you're only a few clicks away from buying the right device or tool for your next job. So fasten your seat belts and log-in today--we guarantee this site will put you in pole position in the time-to-market race!

HEARST *business media*
ELECTRONICS GROUP

EMBEDDEDDEVELOPER.COM
FIND. COMPARE. BUY.

ConVergence PROMOTIONS

a)

Single shielded, open loop				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–45 dBm	–40 dBm	–38 dBm	–34 dBm
Relative	–2 dB	–2 dB	–2 dB	–1 dB

b)

Single shielded, end grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–58 dBm	–53 dBm	–54 dBm	–64 dBm
Relative	–15 dB	–15 dB	–18 dB	–31 dB

c)

Single shielded, both grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–58 dBm	–53 dBm	–54 dBm	–64 dBm
Relative	–15 dB	–15 dB	–18 dB	–31 dB

d)

Single shielded, out grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–113 dBm	–106 dBm	–101 dBm	–62 dBm
Relative	–70 dB	–68 dB	–65 dB	–29 dB

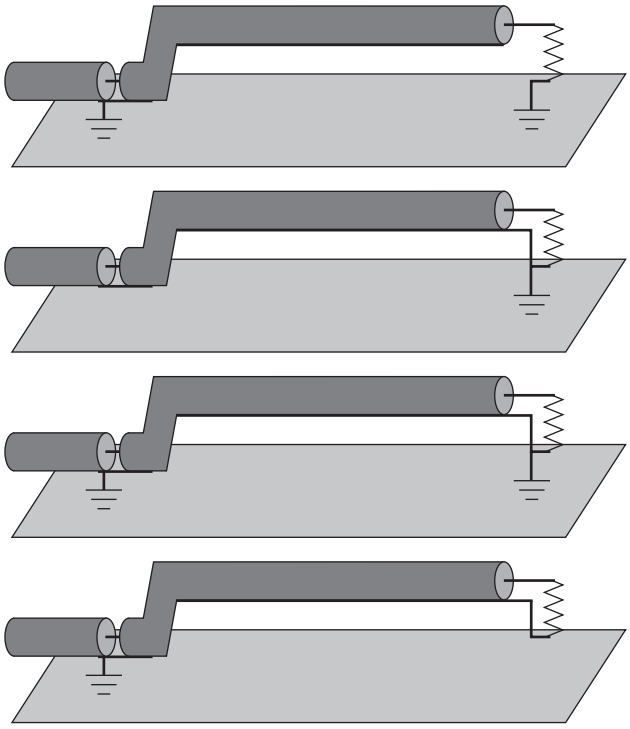


Figure 4—Using a shielded wire doesn't improve the situation if the sensor emulation is still grounded **(a)** because the return path is still through the ground plane. However, an impressive improvement is achieved with a floating sensor **(d)**. The two other configurations provide intermediate performances.

signal. Your first idea would probably be to use shielded cable, and that's what I've done first too. The results are illustrated in Figure 4. If you replace the unshielded cable with a shielded one and connect only the shield to ground at one point, then the noise is reduced only by 2 dB (see Figure 4a)! If you think twice, that's normal because we are in Inductive Coupling mode and the shielded cable doesn't reduce the surface of the magnetic flow coupling area. Current can't flow through the shield. The return path is still 100% through the ground plane, which means it is 10 cm away from the cable. But we can do better. Moving the ground point at the end of the cable close to the signal source reduces the noise by 15 dB, which is a significant improvement (see Figure 4b). Grounding the cable at both ends doesn't change the result (see Figure 4c); however, a huge improvement is possible with the same setup. If you connect the sensor ground to the cable shield (not to the ground plane), the noise is reduced by an impressive 65 to 70 dB, at least up to 20 kHz (see Figure 4d)! This is because this configuration forces all return current to flow through the shield close to the signal wire. However, at 20 MHz, the improvement is less impressive: it's "only" 29 dB.

But were these improvements due to the shield or the fact that the return path was far closer to the wire than in configuration A? To answer that question, I made a second test by replacing the shielded cable with a simple two-wire configuration (see Figure 5). Stangely, the unshielded configuration (Figure 5a) seems a little better than its shielded equivalent (see

a)

Dual wire, both grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–68 dBm	–63 dBm	–60 dBm	–38 dBm
Relative	–25 dB	–25 dB	–24 dB	–5 dB

b)

Dual wire, end grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–91 dBm	–85 dBm	–82 dBm	–40 dBm
Relative	–48 dB	–47 dB	–46 dB	–7 dB

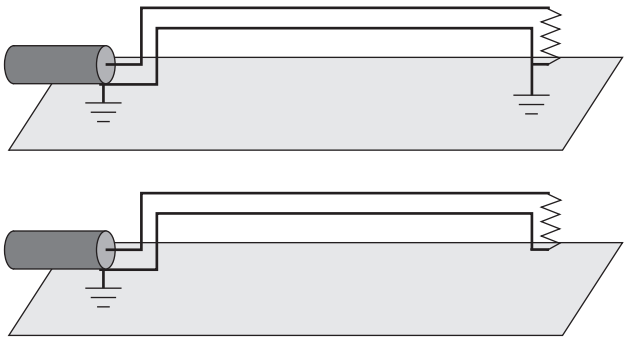


Figure 5—In comparison with Figure 4, a simple two-wire configuration already provides a reasonably good result in low frequencies, but not in the higher frequency range.

a)

Dual shielded, out grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–155 dBm	–111 dBm	–106 dBm	–75 dBm
Relative	–72 dB	–73 dB	–70 dB	–42 dB

b)

Dual shielded, both grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–105 dBm	–104 dBm	–102 dBm	–70 dBm
Relative	–62 dB	–66 dB	–66 dB	–37 dB

c)

Dual shielded, end grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–111 dBm	–109 dBm	–104 dBm	–58 dBm
Relative	–68 dB	–71 dB	–68 dB	–25 dB

d)

Dual shielded, both + signal grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–89 dBm	–84 dBm	–82 dBm	–58 dBm
Relative	–46 dB	–46 dB	–46 dB	–25 dB

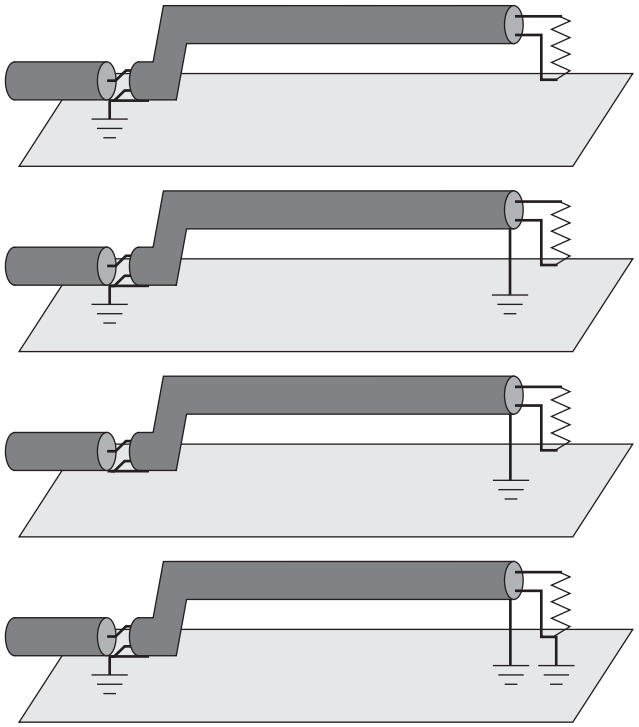


Figure 6—A two-wire plus shield cable is a clear improvement over a single-wire shielded cable because the current return path is decorrelated from the shield. The configuration in **Figure 6a** is better than the others.

Figure 4c) up to 20 kHz, but it is far worse at 20 MHz. Similarly, Figure 5b is significantly worse than Figure 4d. In a nutshell, routing two wires close to each other drastically improves the situation (as compared to Figure 3), but a shielded cable nearly always performs better, especially at high frequencies. Not a surprise.

Can we do even better? You bet. One of the solutions is to use a dual-wire shielded cable. The shield can be dedicated to a shield function, and the two inner wires can be used respectively for signal and return current. This is a better configuration to avoid inductive coupling because the two wires are even closer and protected against electric fields by the shield. Figure 6 shows the corresponding configurations and measurements. Once again, the best results are achieved when the shield is grounded only at the receiving side (see Figure 6a). If you compare it with our previous winner (Figure 4d), you'll notice that the performances are better and especially improved at high frequencies: 42-dB attenuation at 20 MHz. The other configurations (Figures 6b–d) aren't bad, but they aren't as good as the configuration in Figure 6a.

Lastly, I have tested an unshielded twisted pair (see Figure 7). Once again,

a)

Twisted, out grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–120 dBm	–109 dBm	–93 dBm	–40 dBm
Relative	–77 dB	–71 dB	–57 dB	–7 dB

b)

Twisted, end grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–115 dBm	–97 dBm	–84 dBm	–31 dBm
Relative	–72 dB	–59 dB	–48 dB	2 dB

c)

Twisted, both grounded				
Frequency	200 Hz	2 kHz	20 kHz	20 MHz
Absolute	–68 dBm	–61 dBm	–59 dBm	–42 dBm
Relative	–25 dB	–23 dB	–23 dB	–9 dB

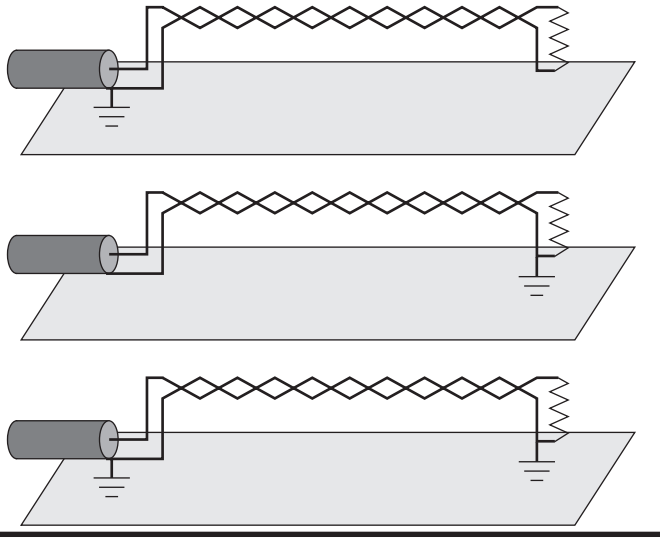


Figure 7—A simple twisted cable is a good solution at low frequencies. Unfortunately, the lack of a shield is a problem at higher frequencies. That's why shielded twisted cables are a good mix.

the best configuration is when the return wire is grounded only on the receiver side, but the measurement results are really impressive (see Figure 7a)! At 200 Hz, the noise attenuation is even better than with a two-wire untwisted shielded cable (–77 dB against –72 dB). The attenuation is similar at 2 kHz, but the shielded cable is by far the winner at higher frequencies. In particular, the unshielded twisted pair doesn't seem to provide any significant attenuation at 20 MHz (–7 dB), where the two-wire shielded cables provide –42 dB.

Unfortunately, I didn't have a shielded twisted cable on hand that day, but I can assume that such a cable would add the benefits of a two-wire shielded cable (see Figure 6) and the benefits of an unshielded twisted pair (see Figure 7). This cable would then be as good as a twisted pair at low frequencies, and as good as a shielded cable at higher frequencies, a definitive winner. That's why shielded twisted pairs are everywhere in networks and industrial applications.

RULES TO KEEP IN MIND

After performing these experiments, I concluded that the theory was correct. The first thing to keep in mind is that the current return path of any signal should always be kept as close as possible to the signal to reduce inductive coupling effects. This can be done by either routing the active wire as close as possible to a ground plane or disconnecting (if possible) the signal reference from the ground and using a two-wire cable ideally shielded and twisted (see Figure 5b).

I also concluded that it is always better to ground a cable at one point. If you are not convinced, examine Figures 4c–d for examples. This is because any "ground loop" will generate currents in the loop when exposed to a magnetic field so noise current would flow in the ground. The current will then generate noise in the signal either through a non-null ground impedance effect or capacitive coupling. However, keep in mind that this recommendation applies only for low-frequency signals, not for radio waves where standing wave effects start to appear. At radio frequencies, it is usually better to connect shields to ground at as many points as possible to form a meshed ground structure. But for audio signals, the "one point to ground only" rule seems applicable.

But which side of the cable should be grounded? Figures 4b–d (or Figures 6a–c) show the clear answer. With an ungrounded source and a grounded receiver (the spectrum analyzer in this case), it is always better to ground the shield at the receiver end, not at the source. Even though I have not tested it, the situation would be reciprocal with a grounded source and an ungrounded receiver. The shield should then be grounded at the source side. Figure 8 shows these configurations. If you check again, you'll see that this configuration once again reduces the ground loop size.

All of these tests were done with a victim cable exposed under a fixed high magnetic flux for immunity concerns. Keep in mind that the same analysis is applicable if you need to reduce magnetic flux emitted by a noisy wire (e.g., to comply with spurious emission restrictions). If a given configuration is immune to external fields, it will not generate any field if the internal wire is excited. This is the beauty of reciprocity.

WRAPPING UP

I had fun performing the aforementioned experiments. Although some of the results surprised me, they always ended up being compatible with the underlying theory—and that's good news for both the theory and my sanity. They don't exactly match the results published in the original works of Henry Ott, but they are qualitatively close. Once again, this demonstrates that you always need to think twice before putting an old book in your basement, or worse. I hope you are now tempted to do your own shielding experiments. I sure was after reading Henry's book. Cable shielding is no longer on the darker side for me, and I hope it isn't for you.

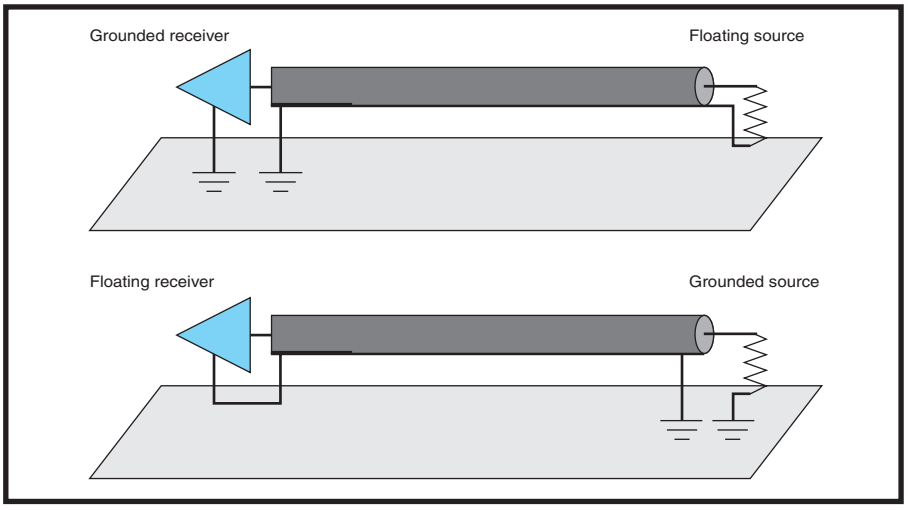


Figure 8—This is the best way to ground a shield, depending on which side of the connection is grounded. If both ends are grounded, the performances will be worse and you'll need to test to see if it is better to ground the shield on one side or both. This may depend on the environment.

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.

RESOURCE

H. W. Ott, *Noise Reduction Techniques in Electronic Systems*, Second edition, John Wiley & Sons Inc., Hoboken, NJ, 1988.



Hot Enough for You?

Seeing Red (Infrared, That Is)

As Tom explains, having all the MIPS and megahertz in the world means nothing unless your design has some meaningful real-world data to process. To gather such meaningful data, you need innovative sensors. This month, Tom introduces a handy option you should consider for your next temperature-sensing design—an infrared noncontact IC thermometer.

This month, let's take a look at yet another new MCU—not!

Don't get me wrong. As far as I'm concerned, MCUs are the shining stars of the silicon revolution, right up there with fancy microprocessors and memory chips. MCUs may not get the headlines, but they sure get the sockets.

Yes, it's grand to have more MIPS and megahertz on tap than ever before, but it's all for naught unless there's some meaningful real-world data to crunch on. And when we're talking about "meaningful real-world data," we're talking about sensors.

Last June, while I was waiting for the Sensors Expo and some neat gadgets to write about, I decided to play with the one sitting on my desk. I thought, why wait? The silicon climate is always changing and sensor season started a bit early this year.

ASK AND YE SHALL RECEIVE

Measuring temperature is a fundamental task at the heart of myriad applications. That's reflected in the wide variety of temperature sensors on the market. Thermocouples, resistance temperature detectors (RTDs), thermistors, and IC temperature sensors are just some of the arrows in a designer's quiver. These are all fine devices that continue to serve designers in good stead.

But, pardon the pun, to one degree or another, they all come with certain compromises depending on the task at hand. It's up to designers to juggle all the trade-offs: cost, power, temperature range, accuracy, response time, and so on. A temperature sensor is like a balloon. Squeeze the specs here and out they pop there. Notably, a trade-off between range and accuracy is a common theme.

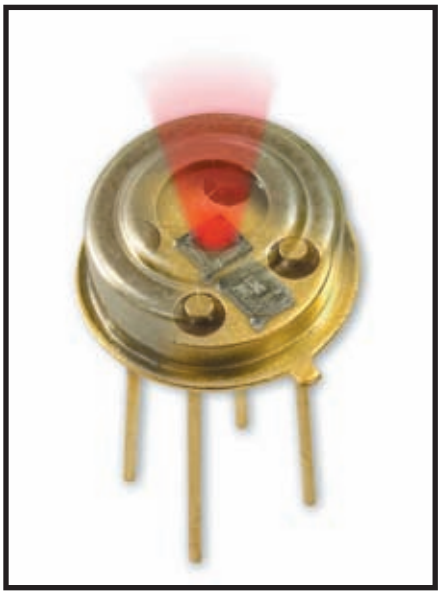


Photo 1—The Melexis Microelectronic Systems MLX90614 brings noncontact IR temperature sensing to an application near you.

The devices most designers are familiar with are "contact" temperature sensors that require physical mounting to the object of interest. In turn, that can limit the allowed temperature range. Any contact sensor can tolerate temperate applications (e.g., –40° to 85°C), but the list narrows quickly beyond that. You simply can't use a plastic IC sensor in a blast furnace.

When it comes to accuracy, in many applications, a couple-of-few degrees between friends is fine; but in others, it isn't. Medical applications come to mind. You wouldn't want the doctor to think your temperature is normal when in fact you have a fever. Once again, the list of temperature sensor options shrinks if you up the ante with demands for better (i.e., fractional-degree) accuracy.

Similarly, response times measured in seconds are fine for human-interest applications like your home thermostat. But it's not hard to imagine real-time scientific and industrial processes that need to track and respond to temperature changes more quickly.

Another headache taken for granted is the wires that connect the sensor to the data-acquisition gear. The problem isn't just stringing the wire, which can itself be a real challenge in harsh environments. There's also the issue of signal conditioning—nontrivial for something like a thermocouple that puts out just a few measly microvolts per degree Celsius.

In a way, it's funny that designers take all of the compromises and trade-offs in stride. Just haul out the same old bag of temperature sensor tricks and make do the best you can. But we don't cut other high-tech gadgets any slack. They're always expected to do Moore for less. Why do temperature sensors get a free pass?

I say it's time for designers to stand up and demand more. By golly, I want a temperature sensor that has more range and more accuracy than I'll ever need. Oh yeah, and while you're at it, make sure it's low-power and easy to wire up. Better yet, just make it noncontact and digital to get rid of the wires and signal-conditioning hassles altogether.

My wish has been granted. Say hello to the Melexis Microelectronic Systems MLX90614 (see Photo 1).

TEMPERATURE IS THE MESSAGE

Of course, one way to make a wireless thermometer is to mate a

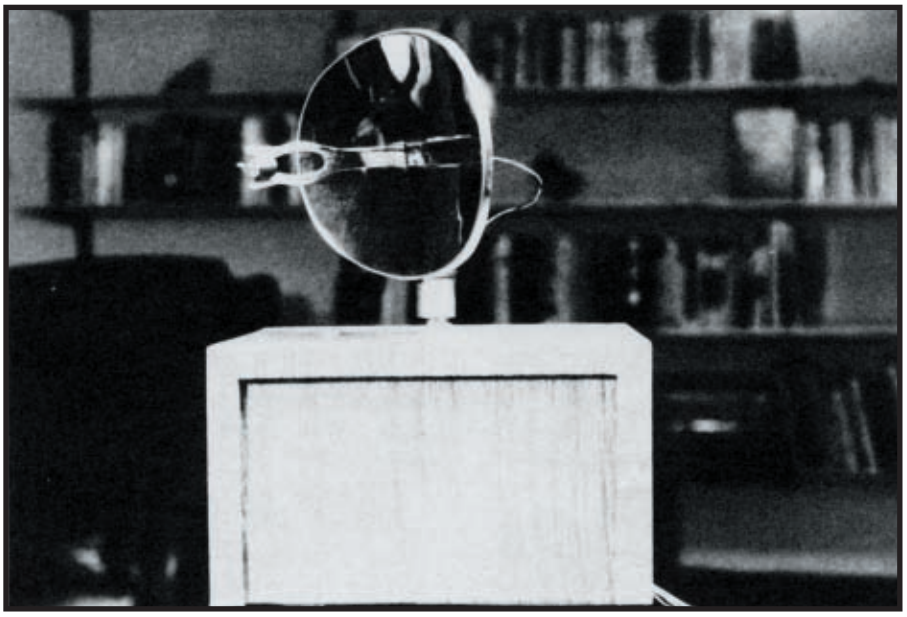


Photo 2—Long-time *Circuit Cellar* readers may remember when Steve Ciarcia took on the challenge of IR temperature sensing way back in 1978. His lash-up used an IR photodetector with a blue-collar optical assist (i.e., focus) from a solar cigarette lighter.



traditional (i.e., wired) temperature sensor with one of the ever-proliferating low-cost ISM-band radios (e.g., IEEE 802.15.4). Instead, the MLX90614 determines an object's temperature indirectly by measuring its IR emissions. In other words, temperature is both the medium and the message.

The idea of using an IR detector to measure temperature isn't new at all. Old-timers may remember Steve Ciarcia's design from the 1970s. His whimsical lash-up combined an IR photodiode with a solar-powered cigarette lighter (i.e., a parabolic mirror) acting as an IR magnifier (see Photo 2).[1] The Melexis parts don't do anything new; they just do it a heck of a lot better.

Ironically, under the hood of this "noncontact" temperature sensor is, you guessed it, a "contact" temperature sensor. A tiny thermocouple arrangement (the "thermopile") is heated by IR radiation that passes through the circular window on top of the package. To minimize sunlight interference, the window is treated

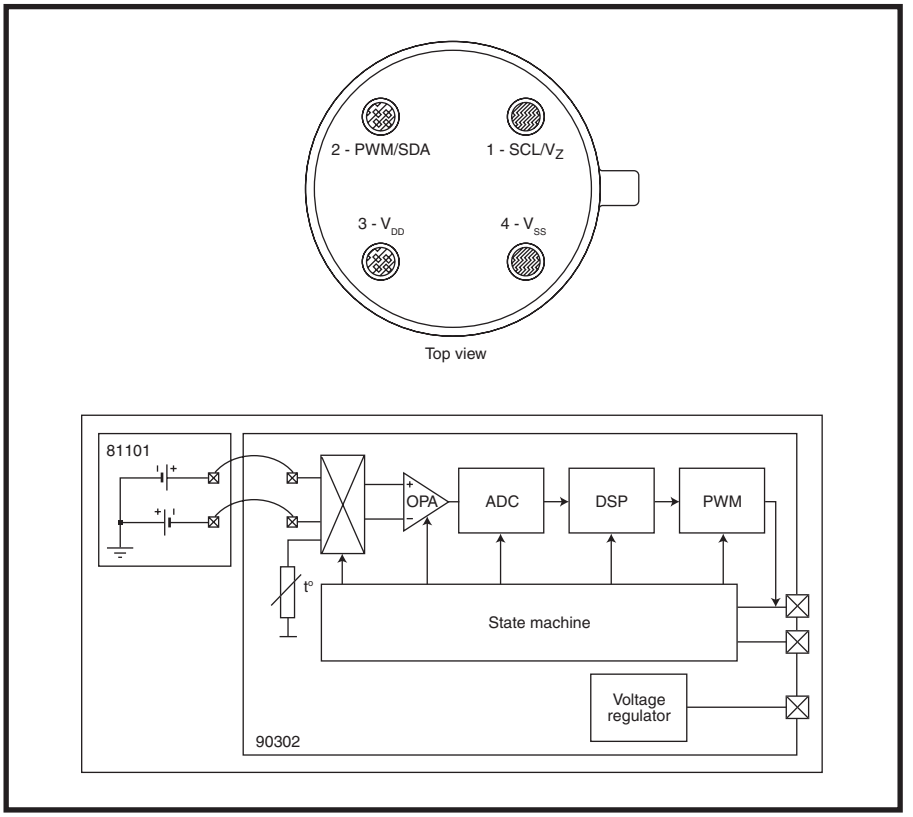


Figure 1—Under the hood of the MLX90614 are two chips. One is the IR sensor itself (the "thermopile"). The other is a signal conditioner integrating the signal chain (amp, ADC, filter DSP), digital interface (i.e., SMBus, PWM), and EEPROM for configuration and calibration settings.

SAVANT
COMPANY INC

EARLY BIRD REGISTRATION OPEN! REGISTER TODAY & SAVE!

Platinum Sponsors

tensilica

UK TRADE & INVESTMENT

novelics

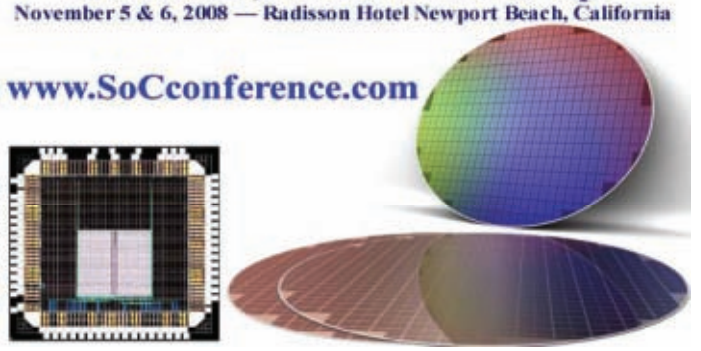
Gold Sponsor

Mentor Graphics

6th International System-on-Chip (SoC) Conference, Exhibit & Workshops

November 5 & 6, 2008 — Radisson Hotel Newport Beach, California

www.SoCconference.com



Questions, Please Contact: SoC@SavantCompany.com
or (949) 851–1714 — www.SavantCompany.com

The Most Informative, Targeted & Affordably Priced SoC, ASIC, CSSP, FPGA, and Foundry Technology Conference, Exhibit & Workshops of the Year
Don't Miss Out!

www.SoCconference.com

AKING NEWS... BREAKING NEWS... BREAKING



Jameco Releases First Web-Enabled Catalog

Order Your Web-Enabled Catalog Today!



Buying electronic components can be difficult with hundreds of thousands of products to choose from. Unfortunately, the choice has been between wielding a twenty pound catalog or sifting through a complicated web site. Jameco solves the problem of finding the products you need with the industry's first and only web-enabled catalog.

The Jameco Catalog... It's packed with features to help you quickly narrow your search. The catalog presents our most popular products in a way that makes it easy to compare everything from franchise sourced brands to house brands and factory overruns.

The Jameco Website...
Once you've found the section you like in the catalog, just use the product category web code found on each page to continue your search at Jameco.com where you'll find more information, more choices, the latest pricing and availability with none of the complications that other websites have.

Jameco has everything you would expect from a components distributor, plus a few surprises.

Reach for Jameco first.

JAMECO® ELECTRONICS

For More Information
Call 1-800-731-4242 or Visit www.Jameco.com/News

to block visible wavelengths, acting as silicon sunglasses if you will. To accurately measure the temperature of the target object, the module's own temperature (i.e., the thermocouple "cold junction") needs to be known as well, so there's also an onboard temperature sensor.

Mated with the IR sensor is a signal conditioner that handles the nontrivial challenge of turning the low-level sensor output into a meaningful temperature measurement (see Figure 1). The signal chain comprises a low-noise amplifier feeding a 17-bit ADC and a mini-me DSP that massages the digital data using filter coefficients and calibration constants stored in on-chip EEPROM.

Melexis offers a family of parts with a variety of options, including single- or dual-zone (i.e., one or two IR detectors), 35° to 90° field of view, and 3- or 5-V operation (see Figure 2). Note that there are a couple of functional differences associated with the latter option. The 3-V part has a power-down mode that cuts power consumption by a factor of 1,000 (i.e., from a few milliwatts active to a few microwatts in power-down), which is helpful for battery-powered applications. Meanwhile, the 5-V part has a regulator feature that enables running off a higher voltage (e.g., 12 V).

Standard (–40° to 85°C) and extended temperature range (–40° to 125°C) versions are available. Note that this refers to the operating temperature of the module itself, not its temperature-sensing range. Indeed, the MLX90614 is factory-calibrated to measure temperatures all the way from –70° to 380°C. Accuracy is decent (i.e., within a few degrees) even at the extremes, but it is notably excellent (within ±0.5°C) for temperature ranges of human interest (e.g., module temperature of 0° to 50°C and object temperature of 0° to 60°C). There's even a version of the part screened for "medical" applications that require 0.1°C precision (see Figure 3).

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.

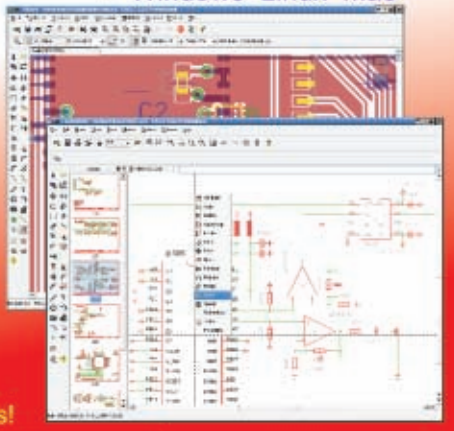
These are the real cost killers!

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

SPLIT PERSONALITY

I'm always amazed at how clever chip designers are at cramming a bunch of functionality behind a few pins, just four of them in this case. With two pins devoted to power and ground, that leaves just two to play with, but the MLX90614 still manages to accommodate three distinct modes of operation.

The most powerful mode uses the two pins (SCL and SDA) for a system management bus (SMBus), an I²C-like clock serial interface that offers a host MCU complete access to all of the MLX90614's features and controls. The serial bus slave address is programmable (i.e., stored in onboard EEPROM), so you can daisy chain multiple units, keeping in mind bus-loading considerations (longer chain equals more power and less speed).

There are also two "stand-alone" modes of operation. The simplest turns the module into a thermal relay that compares the object's temperature with a programmable threshold and outputs the result (i.e., 1 or 0) on the SDA pin. Separate minimum and

maximum threshold registers allow dialing in some hysteresis to prevent output chattering. The output can be configured as push-pull or open-collector. The latter option allows the alarm outputs from multiple modules to be wire-ORed together.

The second stand-alone mode uses the SDA pin as a continuous PWM output of the temperature reading. This means the sensor can provide a measure of control itself, possibly even eliminating the need for an MCU.

I've got mixed feelings about the PWM mode. It's got some nice features, such as programmable frequency and repeat factor plus the ability to "zoom in" on a temperature subrange of interest. There's also an option to output two different readings (e.g., object and ambient temperature, or two object temperatures for a dual-zone part) interleaved on the single PWM pin. The PWM format is a bit odd because the duty cycle doesn't vary between 0% and 100%, as you might expect (see Figure 4).

These days, most MCUs come with ADCs. So, you might be tempted to hang an RC filter on the PWM output and convert the 1-and-0 PWM pulse train to an analog voltage for sampling with the MCU's ADC. That's certainly an option, but not without caveats.[2] If you decide to go that route, make sure the ADC is referenced to the MLX90614's V_{DD} pin, lest power supply noise and drift compromise measurement accuracy and repeatability.

It is possible to dynamically change between the different modes (i.e., Host, Alarm, and PWM) by in-system programming the configuration EEPROM. For instance, a module might be configured to automatically power-up in PWM mode, but a host MCU can take charge at any time by toggling the SCL pin to switch back to Serial Bus mode.

From the host MCU's perspective, the module appears as two banks of 32 16-bit registers, one bank for EEPROM, and one for RAM (see Table 1). Many of these are for housekeeping (e.g., ID number) or reserved, while others store settings such as factory calibration factors or the programmable temperature thresholds and ranges.

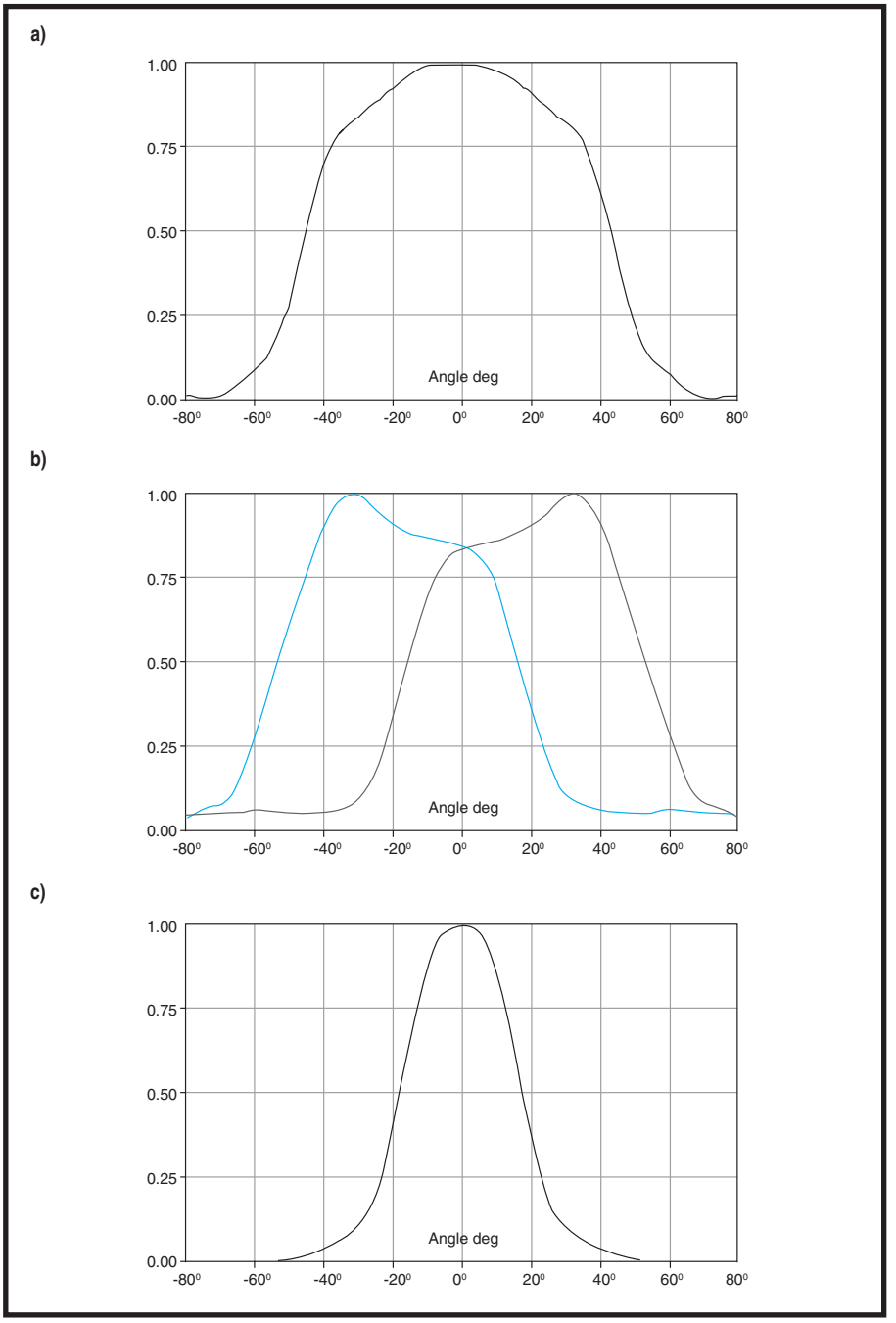


Figure 2—What you see is what you get. The wide 90° field of view single- **(a)** and 70° dual-zone parts provide large area coverage **(b)**. The 35° option narrows the focus **(c)**.

I noticed there are a few cases where a single register contains both bits you might want to program (e.g., configuration) and ones you shouldn't (i.e., factory calibration). Just be careful slinging those ones and zeros around.

PAINT IT BLACK

I'd rather walk the Planck than launch into a physics tutorial on heat transfer and the like, a duty for which I have neither the time nor the expertise. Sure this is one smart gadget; but nevertheless, determining the actual temperature of a particular object isn't a simple point- and-shoot affair. Yes, what you see is what you get, but only if you know what you're seeing.

There are two major considerations that affect what number the thermometer spits out. One is so obvious that it is easily overlooked, and that's the "field of view" (FOV).

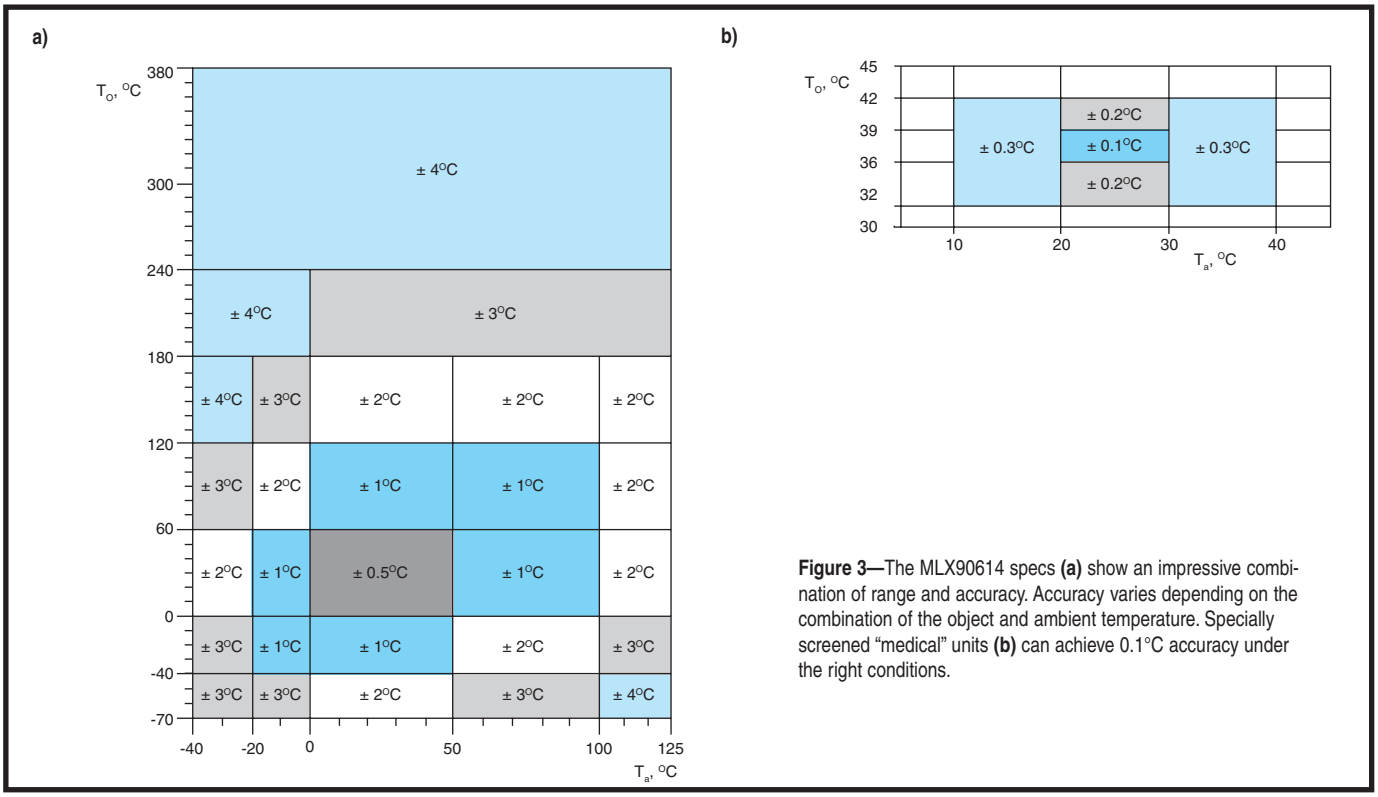


Figure 3—The MLX90614 specs **(a)** show an impressive combination of range and accuracy. Accuracy varies depending on the combination of the object and ambient temperature. Specially screened "medical" units **(b)** can achieve 0.1°C accuracy under the right conditions.

Consider that a 90° FOV spec means the area covered from a distance of 5′ is a 10′ diameter circle. It's clear that absolute temperature accuracy requires optimizing the FOV (i.e., the object fills the frame) and avoiding clutter (i.e., thermal interference from objects of noninterest creeping into the picture).

Like me, a lot of you probably have kids. Have you been through the parenthood ritual of the school science project? You know, the one which is invariably due tomorrow? Herewith, my harried dad science project bailout. Take two clear plastic bottles, paint one black and wrap the other in tin foil. Fill them with water, stick them in a sunny window for an hour, and measure the water temperature.

That brings us to the second absolute accuracy consideration, namely the "emissivity" of the object being measured. It's all due to the fact that materials vary to the degree they absorb, versus reflect, thermal energy. The difference is significant because emissivity affects not only the temperature calculations, but also varies a lot in the real world. Two objects that are actually at the same temperature might measure quite differently from an IR perspective if, for example, one is plastic (e.g., emissivity = 0.9) and the other aluminum (e.g., emissivity = 0.2).

Recalling that school science project, there's an easy way to guesstimate the emissivity coefficient for a particular object, namely just paint it black (turn it into a "blackbody" in physics-speak). Then all you have to do is compare the IR readings you get from two of the objects that are at the same temperature, one painted black to minimize IR reflection.

Just keep in mind that while IR is like visible light in many ways, there are some differences. For instance, there are some materials that are opaque to one but not the other and vice versa. Consider that while glass easily passes visible light, it actually blocks quite a bit of IR. So, if you point the thermometer at a window, you'll mainly be sensing the temperature of the glass whether it's light or dark outside.

There's one final set of dials to tweak. You can reprogram the coefficients for the digital FIR and IIR filters to trade-off speed (i.e., settling time) versus noise. With minimal filtering, settling time is fast at less than 50 ms. Even with maximum filtering (i.e., lowest noise), a

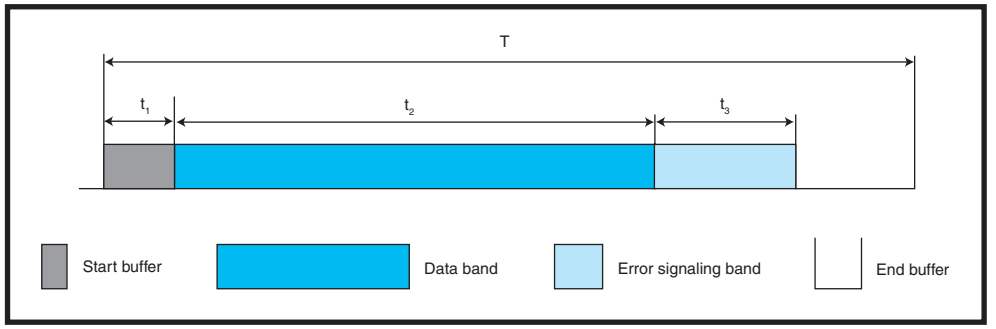


Figure 4—The MLX90614 PWM format is different than most. The temperature measurement occupies only 50% of the cycle. The first 12.5% of the cycle is always high, acting as a "start bit." The last 32.5% of the cycle is normally low, but will be high if the chip detects an internal error. Thus, the temperature measurement itself is scaled between a 12.5% and 62.5% duty cycle.

temperature reading takes less than 1 s.

EASY RIDER

Enough of this book learning. It's time to wire something up. I got my hands-on fix courtesy of Parallax and their MLX90614 application module (see Photo 3). It combines the Melexis part with a dedicated coprocessor (i.e., MCU) that provides a high-level command set and UART host connection. Pairing the module with BASIC Stamp, it takes only three jumper wires and a few lines of code to take a temperature reading.

Address	EEPROM Function	RAM Function
0	Object temperature threshold/range max	Reserved
1	Object temperature threshold/range min	Reserved
2	PWM Options configuration	Reserved
3	Ambient temperature range min/max	Reserved
4	Emissivity correction coefficient	Reserved
5	Configuration and factory calibration	Reserved
6	Reserved	Ambient temperature
7	Reserved	Object 1 temperature
8	Reserved	Object 2 temperature (dual zone)
14	SMBus Slave address	Reserved
28	ID Number	Reserved
29	ID Number	Reserved
30	ID Number	Reserved
31	ID Number	Reserved

Table 1—The register map of the MLX90614 includes EEPROM to store configuration and factory calibration settings and RAM for the IR and ambient temperature readings.

The timing was right because I had a perfect application in mind. I have a Harley-Davidson motorcycle with an air-cooled engine, something that hasn't changed in 100 years. The other thing that hasn't changed is the fact that the engine can get hot in low-airflow situations (e.g., traffic jams), especially the rear cylinder.[3]

"How hot?" you ask. That's a good question—a question historically answered with another question like "Are my pants on fire?" I've installed an oil temperature gauge that uses a dipstick in the oil tank, but its reading is rather sluggish and not necessarily a true indicator. Indeed, contemplating failure scenarios (e.g., low oil level or failing oil pump), it's apparent that the oil temperature gauge could be misleading. Low oil temperature might simply mean the bike isn't warmed up yet. But it might also mean the oil isn't circulating—or worse, that the level is so low the dipstick is measuring air temperature!

I want to know the actual temperature of the engine (i.e., the rear cylinder head, not the oil). While there are plenty of contact temperature sensor options, I'm not inspired by the concept of bolting/taping/gluing some kludge to my engine. Voila, a perfect application for a noncontact temperature sensor.

I hacked together a simple temperature-logging program, which is pretty self-explanatory (see Listing 1). As written, it takes up to 1,800 readings (max_count) at 1-s (PAUSE 1000) intervals, which makes for 30 min. of capacity. If you want to expand the program or data capacity, just make sure everything will still fit. Both the program and data are stored in the Stamp's EEPROM and no doubt bad things will happen if you log data on top of the program! Also, this program isn't power-optimized. A more battery-friendly version can take advantage of features like the Stamp SLEEP and NAP commands and the module "Z" (continuous read at low power) command.

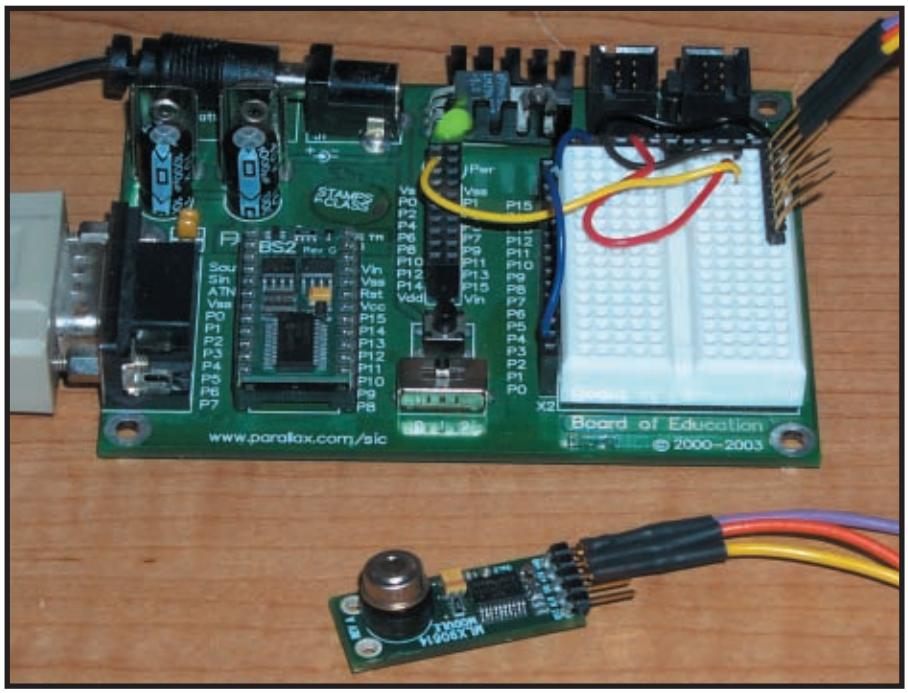


Photo 3—A Parallax BASIC Stamp and MLX90614 application module make it super quick and easy to get started with IR temperature sensing.

Heading out on the highway (see Photo 4), even this quick and dirty experiment yielded some insight (see Photo 5). I was surprised to see how long it takes the engine to warm up. Ignoring the idling temperature spikes near the end of the run, it almost looked as though temperature was leveling off, but it would clearly take a longer ride to verify that.

You could experiment with the emissivity register setting or otherwise calibrate the setup to get an accurate absolute temperature. However, this is a good time to note that many applications don't really need absolute measurements, but rather just relative consistency (e.g., linearity) and repeatability.

And it's obvious that engine temperature indeed shoots up quickly when idling, but fortunately just as quickly moderates once you get going again.

HEAT WAVE

The MLX90614 is a novel solution



Photo 4—Things get hot on the highway to hell, especially the rear cylinder of this air-cooled twin. Here we see both the traditional oil temperature gauge and the MLX90614 lash-up.

to the age-old application challenge of measuring temperature. It surely deserves a place in your quiver of temperature-sensing solutions.

But as is often the case, a technical advance can offer more than just a new way of implementing an old application. The more interesting outcome is when designers discover that a new technology provides a new way of doing something new. Witness the way MEMS accelerometers—which got their start as automotive "airbag chips"—have transformed video game systems like the Nintendo Wii. I'm excited by the prospects for IR temperature sensing to similarly spark novel new applications.

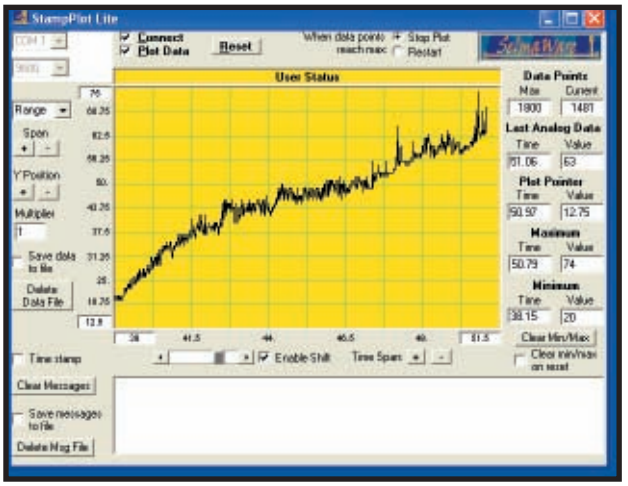


Photo 5—Interesting results came from my initial experiment, a roughly 30-min. ride. The engine was more cold-blooded than I had imagined and took quite a while to warm up. The first 20 min. or so were mostly highway cruising and then you can clearly see the temperature spikes as I hit stoplights near the end of the ride.

Once again, it's time for designers to think outside the box. When it comes to the race to invent the next big thing, the heat is always on.

Listing 1—This simple data logger program shows how easy it is to use the MLX90614. Just issue the temperature read command using a SEROUT statement and read back the high- and low-byte of the temperature data along with a checksum using a SERIN statement. The funny calculation that follows uses the BASIC Stamp integer math to convert Kelvin to Centigrade.

```
' {$STAMP BS2}
lowtemp VAR Byte
hightemp VAR Byte
pecbyte VAR Byte
temp VAR Word
i VAR Word
samples CON 1799

INPUT 1

IF (IN1=1) THEN dump 'otherwise log data

'begin by clearing sample buffer
FOR i = 0 TO samples
     WRITE i,255 'clear sample buffer
NEXT
'now read temp, convert K to C and store in sample buffer
FOR i = 0 TO samples
     PAUSE 1000
     SEROUT 0,396,[0,"!TEMR",$5A,$07]
     SERIN 0,396,[lowtemp, hightemp, pecbyte]
     temp = ((((hightemp * 256)+lowtemp)*2)-27300)/100
     WRITE i,temp
     'DEBUG DEC temp,CR
NEXT
END 'if sample buffer full

dump:
'output temp samples via UART
FOR i = 0 TO samples
     READ i,temp
     IF (temp=255) THEN quit 'at end of sample data
     DEBUG DEC temp,CR
NEXT
quit:
END
```

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.

REFERENCES

[1] S. Ciarcia, "I've Got You In My Scanner: A Computer Controlled Stepper Motor Light Scanner," *Byte*, Vol. 3, Issue 11, 1978.

[2] Melexis Microelectronic Systems, "Application Note: Analog Voltage Output with MLX90614," 39011 90614 01, 2006, www.melexis.com/Assets/Analog_voltage_output_with_MLX90614_5176.aspx.

[3] J. Minton, "Hot New Harleys: Richening Your Engine's Air/Fuel Mixtures," *American Rider Magazine*, www.americanrider.com/output.cfm?ID=1279963.

SOURCES

MLX90614 Infrared thermometer
Melexis Microelectronic Systems
www.melexis.com

MLX90614 Application module
Parallax, Inc.
www.parallax.com

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.

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!

Trace SYSTEMS, Inc.

www.TraceSystemsInc.com
301-262-0300

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

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

NXP ARM7 LPC2K TFBGA Programming Adapters



SAB-TFBGAxxx Low Cost Socket Adapter Boards allow the user to program and test ARM7 LPC2K BGA devices outside of their application circuit. The SAB-TFBGAxxx Family includes support for most NXP LPC2xxx devices in the following packages: TFBGA100, TFBGA180 and TFBGA208. SAB-TFBGAxxx is used along with our USB-ICP-LPC2K, In-Circuit Programmer - $69.00.

Stand-alone mode allows the SAB to be used for verification of basic functionality of the user software. The board includes 12MHz oscillator circuit, POR circuit, and a single LED connected to a user I/O signal. The kit includes Power Supply and Programming Cable for only $399.00.

The USB-ICP-LPC2K & SAB-TFBGAxxx are available directly from Digi-Key www.digikey.com and Mouser www.mouser.com.



NXP ARM7 LPC2K TFBGA Programming Adapters

USB-ICP-LPC2K with SAB-TFBGA180 Adapter

XA Development Kits, I2C, SPI & MDIO tools in stock, visit www.teamfdi.com for details.

FDI
www.teamfdi.com

Future Designs, Inc.
2702 Triana Blvd
Huntsville, AL 35805
(256) 883-1240
Fax (256) 883-1241



CIRCUIT CELLAR

Designer's Notification Network

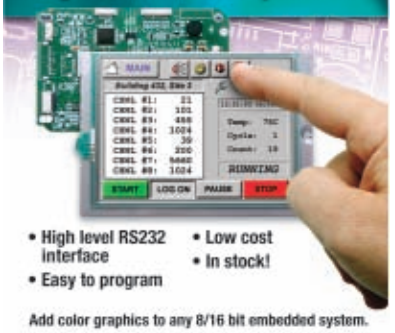
Circuit Cellar design contest entrants have received thousands of valuable development tools and product samples. Because of their contest participation, these engineers receive advance e-mail notice from Circuit Cellar as soon as new samples become available. Now you too can benefit from this early notification.

Welcome to the Designer's Notification Network. Print subscribers are invited to join the Network for advance notice about our new sample distribution programs.



Find out more at www.circuitcellar.com/network.

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

iMAGECraftC

C Compilers for:
MSP430
CPU12
ARM
AVR
etc.

high powered development, garage powered prices!

starting at $249!

• First commercial whole program compression optimizer
• Multiple versions
• Global optimizer
• Simple to use IDE
• Multiple platform support

www.imagecraft.com
info@imagecraft.com
(650) 493-9326 • FAX (650) 493-9329

MCS - 51
EXCELLENT noise immunity!



V-E54 v1.0 (USD 59)
Embedded board MPC82E54A, the most efficient MCU from Megawin, 16K Flash speedx12. 8ch. 10bit A/D, 2ch. 12bit D/A

PRO-MISP (USD 99)
Megawin 8051 ISP Programmer, High Speed download via USB Port, 89E58AE, 82E54AE2, developer board included, 3 Pins to target board (+5V, Data, GND)



www.ispmicro.com
sales@ispmicro.com

Free Training
for Rabbit Developers

Learn More at
Rabbit-U.com
08078

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

PHYTEC

phyCORE® SBCs Accelerate 32-bit Designs



phyCORE-LPC3250

COTS 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: LMX31 (ARM11), LMX27 (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

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

Going Wireless is Easy!
Embedded Bluetooth Module



NEW

Parani-ESD 1000

- Class 1 OEM style
- Bluetooth (v2.0 + EDR)
- Up to 100 meter transmit distance

LEMOS INTERNATIONAL
1-866-345-3667
sales@lemosint.com

www.lemosint.com

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.™

iMAGECraft C

NEW: the eBox- AVR!

Compact Complete AVR Kit
- 16K ImageCraft ICCAVR compiler
- Mega16 kit with LED, keypad, LCD and more
- 10 example software projects
- Battery INCLUDED

Perfect for learning embedded programming and the Atmel AVR.

only $199!

www.imagecraft.com
info@imagecraft.com
(650) 493-9326 • FAX (650) 493-9329

EMULATE A USB KEYBOARD+MOUSE WITH RS-232 INPUT

MODEL USB-KM232
-Send keystrokes or mouse actions
-Control individual key up/down and combinations
-RS232 input, USB output
-Powered by the USB port

888-690-9080
HAGSTROMELECTRONICS.COM
11 Fiddlers Green, Lansing NY 14882

CAN controllers

- Embedded, Standalone, C/C++ Programmable
- CAN to Ethernet, USB, CF, TFT, RS232/485, ADC, DAC, I/Os, Relays,...
- CAN-Engine™ Controller Area Network (CAN) with variable baud rate, low level register access, Ethernet, CompactFlash, ADCs...

$39 OEM
$99 Qty 1

60+ Low Cost Controllers with TFT, ADC, DAC, UARTs, 300 I/Os, solenoid, relays, CompactFlash, 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

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

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

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

iMAGECraft C

The Best Selling C Compiler for AVR

- High quality code generation
- First commercial whole program compression optimizer

high powered development, garage powered prices!

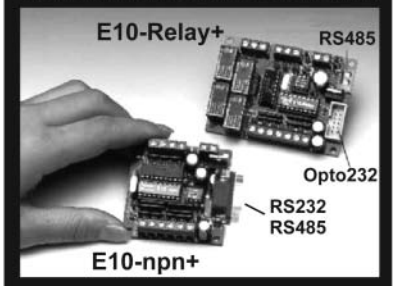
starting at $249!

- Application builder
- Multiple versions
- Global optimizer
- Simple to use IDE
- Multiple platform support
- Flexible licensing schemes

www.imagecraft.com
info@imagecraft.com
(650) 493-9326 • FAX (650) 493-9329

The $69 PLC

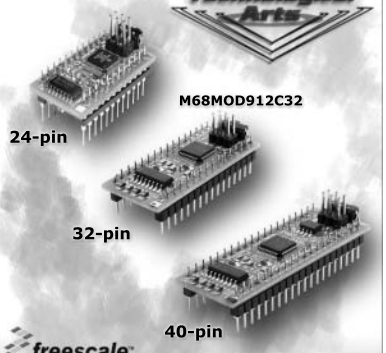
Work as Stand-Alone Ladder Logic PLC.
Or as Smart Remote I/Os of PC/ PLCs.
RS485 allows 256 units to be networked.



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
Triangle Research
Tel: 1-877-874-7527 - PLC specialist since 1993

Freescale 9S12C microcontroller DIP modules

Technological Arts



freescale Alliance Member
www.nanocore12.com
Toll-free: 1-877-963-8996

TStik!tm

Rugged TINI Java™ Module with 10/100 BaseT



TStik is a ruggedized TINI400 chipset in the familiar SIMM72 form factor. Upgrade most DSTINI1 (TINI390) systems or use our new TILT socket boards (TILT Pro is shown above).
TStik is about $100, and sockets start at under $60.

SYSTRONIX®
full details at www.TStik.com

NEW from ccs Microchip Third Party Tool Provider

Audio Experimenter's Board



- On board dsPIC33FJ128GP706
- Audio codec; analog/digital signal input/output
- SD card memory expansion
- Character LCD display

DSP Analog

Development Kits · USB $619 · DSP $469
with full IDE PIC®MCU Compiler

USB Master



- USB Host Compatability
- Examples for:
  - Printers
  - USB Flash Memory
  - Joy Sticks
  - Keyboards...
  - and more

www.ccsinfo.com/UD
sales@ccsinfo.com
262.522.6500 x35

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"

NEW! picoFlash

$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



Weather Instruments for PCs

AAG electrónica

www.aagelectronica.com

USB Data Acquisition

ADU208 -USB Relay I/O Interface

Complete SDK Online at: www.ontrak.net



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

Incredibly Low-Cost USB Chips at HexWax

- ***USB-232*** USB to serial UART, smarter and cheaper than the FT232R
- ***USB-SPI*** USB to synchronous serial (I2C, too!)
- ***USB-DAQ*** Data logger, plugs in like a flash drive
- ***USB-Config*** Configuration & diagnostic module
- ***USB-TakeOff*** Power take-off & battery charger
- ***WattLogic*** Domestic power consumption logger
- ***AnniLogic*** Anniversary reminder
- ***expandIO-USB*** I/O expanders
- ***TEAleaf-USB*** Security dongle

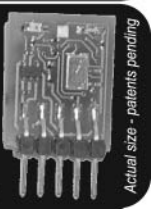
No drivers needed!

Visit www.hexwax.com today!

Amazing PIC programmer

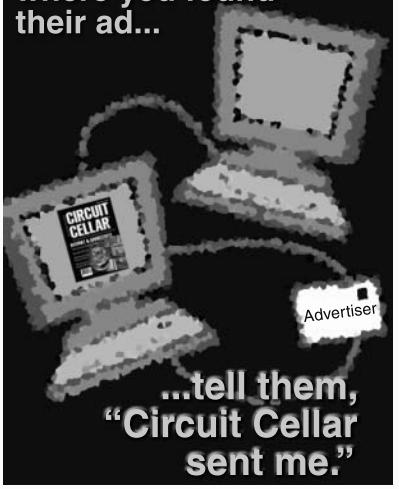
Most devices supported
ICSP, SQTP, & copy limits

$32 at Digikey & Mouser
www.flexipanel.com



Actual size - patents pending

When contacting our advertisers, tell them where you found their ad...



...tell them, "Circuit Cellar sent me."



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!

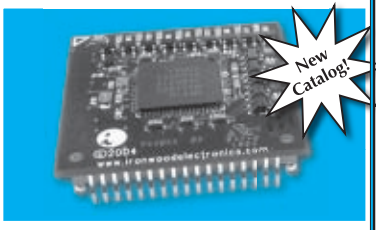
Register now. It's fast. It's free.
www.circuitcellar.com/newsletter/

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

IC DEVICE AND PACKAGE CONVERTERS



*** Upgrade Your PCB**
*** Fix Design Problem**
*** Replace Obsolete Part**

Our innovative engineers can QUICKLY solve your problem. We provide solutions for all IC package types including BGA, QFP, PLCC, SOIC, DIP, PGA, and QFN.

Request our Free Catalog today!

Ironwood Electronics
1-800-404-0204
www.ironwoodelectronics.com

Temperature Monitoring Over the Internet

-View temperatures using a web browser

-Connect up to four temperature sensors

-Send email alerts or alarms

-Two built-in relays



CONTROL by WEB
www.ControlByWeb.com

(435) 750-5999

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

Order online at:
www.melabs.com

microEngineering Labs, Inc.
Development Tools for PIC® Microcontrollers

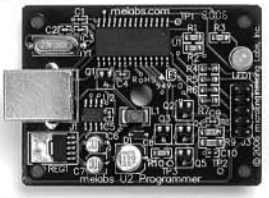
Phone: (719) 520-5323
Fax: (719) 520-1867
Box 60039
Colorado Springs, CO 80960

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.

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

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



Serial LCDs
2-line x 16 $39.95
4-line x 20 $49.95
Quantity Discounts Available!

PICPROTO™ Prototyping Boards

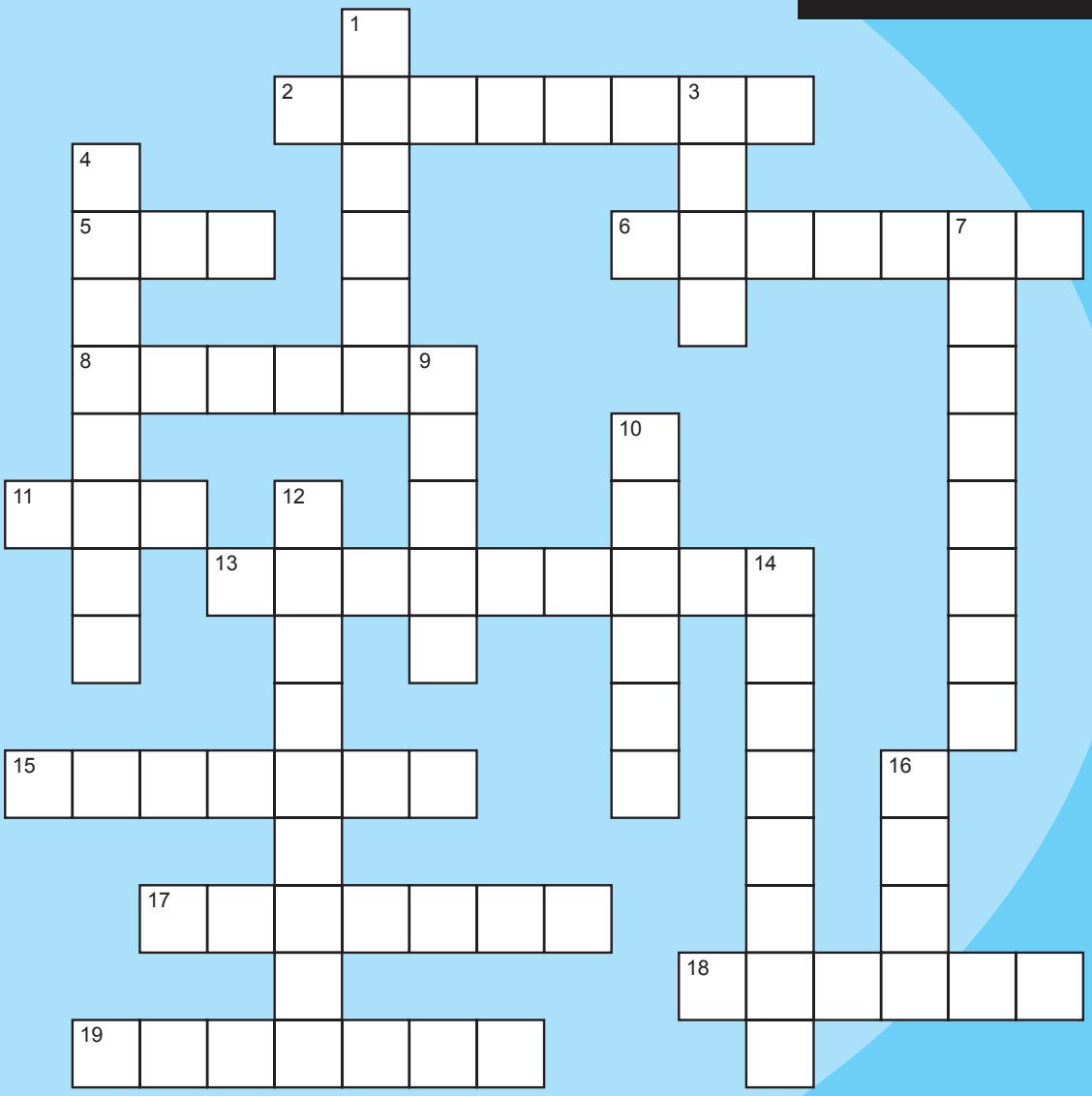


Double-Sided with Plate-Thru Holes
Circuitry for Power Supply and Clock
Large Prototype Area
Boards Available for Most PIC® MCUs
Documentation and Schematic

Pricing from $8.95 to $19.95

See our full range of products, including books, accessories, and components at:
www.melabs.com

CROSSWORD



Across

2. The unique name of a computer system on a network
5. The "9000" in *Space Odyssey*
6. Series of monomers in a chain
8. Text message: "probably"
11. The "K" in CMYK
13. A set of acceptable e-mail addresses
15. Staging area
17. A program that searches the Internet for new web pages
18. Time between wave repetitions
19. A unit of electric charge; French physicist

Down

1. A computer that runs Windows and includes an Intel chip
3. A basic particle with a mass about 200 times the size of an electron that is common in radiation near the Earth's surface
4. An old-timer interested in embedded design
7. Dot dot dot
9. 10^{-24}
10. A particle detector used to measure radiation
12. A small version of a picture on a computer
14. Wolfram
16. An extremely small web server

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	37	Futurlec	57	Luminary Micro	80	SoC Conference
66	AP Circuits	64	Grid Connect, Inc.	89	MCC (Micro Computer Control)	40	Saelig Co.
92	All Electronics Corp.	90	Hagstrom Electronics	47	Microchip	5	Saleae LLC
90	Apex Embedded Systems	90	HI-TECH Software LLC	92	microEngineering Labs, Inc.	11	SEGGER Microcontroller Systems LLC
7	Atmel	32	HobbyLab LLC	95	Mouser Electronics	21	Silicon Laboratories, Inc.
55	Bitscope Designs	73	Holtek Semiconductor, Inc.	89	Multilabs	48	Solder By Numbers (Schmartboard)
65	CWAV	89	ISP Micro	17	NKK Switches	41	Spark Fun Electronics
82	CadSoft Computer, Inc.	32, 89	IMAGEcraft	C2	NetBurner	91	Systronix
56	Comfile Technology, Inc.	90	IMAGEcraft	45	Nurve Networks LLC	C3	Tech Tools
91	Crossware Products, Inc.	1	Imagineering, Inc.	91	Ontrak Control Systems	2, 3	Technologic Systems
91	Custom Computer Services, Inc.	92	Ironwood Electronics	19	PCB-Pool	91	Technological Arts
44	Custom Solutions, Inc.	64, 66	JKmicrosystems, Inc.	C4	Parallax, Inc.	90	Tern, Inc.
87	DLP Design	91	JKmicrosystems, Inc.	89	Phytec America LLC	89	Tin Can Tools, LLC
32	DesignNotes	81	Jameco	92	Pioneer Hill Software	31	Total Phase, Inc.
44	EMAC, Inc.	45	Jeffrey Kerr LLC	33	Pololu Corp.	87	Trace Systems, Inc.
90	Earth Computer Technologies	18	Keil Software	87	Pulsar, Inc.	91	Triangle Research Int'l, Inc.
74	Embedded Developer	23	LPKF Laser & Electronics	79	R4	37	WCSC (Willies Computer Software Co.)
22	ExpressPCB	45	Lakeview Research	27, 49	Rabbit, A Digi International Brand	10	WIZnet
9	ezPCB	90	Lawicel AB	89	Rabbit, A Digi International Brand	92	Xytronix Research & Design, Inc.
87	FDI-Future Designs, Inc.	9, 89	Lemos International	89	Reach Technology, Inc.		
91	FlexiPanel Ltd.	49	Linx Technologies	29, 39	Renesas Technology		

Preview of November Issue 220
Theme: Analog Techniques

Radiation Detection: A System that Measures Digital and Analog Pulses

Embedded Cause and Effect: A Lighting System that Responds to Audio Stimuli

Time Server Design: Synchronization with the U.S. WWVB Timecode Signal

Low-Cost Serial-to-USB Migration

Digital Stompboxing: An Easy-to-Use Digital Signal Processing Platform

Weather-Monitoring Circuitry: A Design for Indoor-Outdoor Use

---

LESSONS FROM THE TRENCHES **What Can C Structures Do for You?**

FROM THE BENCH **Drive by Wire:** A Look at Noncontact Sensing

SILICON UPDATE **It's the Sensors, Stupid:** A Review of New Sensor Technology

ATTENTION ADVERTISERS

December Issue 221 Deadlines

Space Close: Oct. 10
Material Close: Oct. 20

Theme:
Programmable Logic

Call Shannon Barraclough
now to reserve your space!
860.875.2199
e-mail: shannon@circuitcellar.com

The Newest Semiconductors

Texas Instruments Authorized Distributor · ST · Atmel · NXP founded by Philips · Microchip · Fairchild Semiconductor · freescale semiconductor · Cypress Perform · Silicon Labs · Infineon · Vishay · Actel · ON Semiconductor · Sharp Microelectronics of the Americas · FTDI Chip · NEC Electronics America · Cirrus Logic · Lattice Semiconductor Corporation · zilog · Catalyst Semiconductor, Inc. · Diodes Incorporated · PLX Technology · CEL California Eastern Laboratories · Toshiba Leading Innovation · ROHM · Supertex · Pericom Enabling Serial Connectivity · SMSC Success by Design · Central Semiconductor · Power Integrations · Epson · Rectron · Advanced Linear Devices, Inc. · Ramtron · Zetex Semiconductors · Analogic Tech · JRC NJR Corporation · IXYS · DigitalView · SII · ISSI · Enpirion · power electronics in motion eupec · WJ is Now Part of TriQuint Semiconductor · Taiwan Semiconductor

New Products from:



Texas Instruments Authorized Distributor

TMS320C672x Digital Signal Processor
www.mouser.com/tidsp/a



Cirrus Logic

CS48540 32-bit DSP
www.mouser.com/cirruslogic/a



Microchip

DSPIC33F Digital Signal Processors
www.mouser.com/microchip/a

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

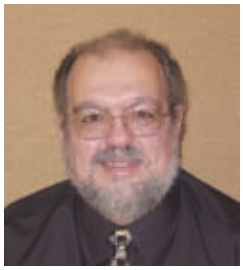
The Newest Products For Your Newest Designs



www.mouser.com

Over A Million Products Online

(800) 346-6873



PRIORITY INTERRUPT

by Steve Ciarcia, Founder and Editorial Director

Horses and Hard Drives

I doubt that there are many other people in the technical landscape with enough sense of humor to equate the evolution of horses and cars to hard drives and solid state drives (SSDs), but here I am. Horses, for the most part, were reliable, low-cost, had predictable operating parameters, and were cheap to operate (grass). Cars, on the other hand, were noisy and smelly (neglecting that horses had their own unique fragrance), had suspect reliability, had an extremely high purchase cost, had unpredictable operation (bad roads), and fuel was hard to find (no gas stations). We should all still be riding horses, right?

That would certainly be the case if this was strictly a price/performance comparison of two nonevolving technologies at a single point in history. But cars continued evolving while horses did not. Flash memory is a similar story of evolution. Embedded developers, iPod users, cell phone crazies, digital photographers, etc., all rely on the significant advantages of nonvolatile flash memory as a core design element in the functions of these products. Of course, incorporating flash successfully had its learning curves and some experiences are hard to forget.

Back in the early 1990s, I was called in as a consultant to help a company having problems getting one of its new communication products to stop failing. The device was unique for the time and offered significant advantages over the competition. The bad news was that after about two weeks of operation, it lost its marbles (i.e., forgot all its setup information and user inputs). I suppose I could have milked the job for a few days playing with a scope probe, but after looking at the schematic, I chuckled and asked the software guys how often they were writing the setup information to the flash memory.

You know software guys. When doing a task one time works, doing it a whole lot more often is better. ;-) It turns out that they were refreshing the configuration and user settings to the on-board flash chip every time they dumped the communication traffic RAM buffer to the hard drive—about once a minute. Worse yet, they were writing it to the same flash memory address every time. I guess no one on the hardware team told the software guys that any single address on the flash memory was only good for about 10,000 write cycles. It should have only lasted one week. They were lucky it lasted for two.

I can't speak for all embedded engineers, but it is experiences like these that add a certain personal skepticism about incorporating flash memory in everything. Certainly, technology has changed a lot through the years and flash write cycles are now in the hundreds of thousands (I believe I even heard 1 million quoted by one vendor) and flash makes absolute sense in digital cameras, cell phones, and A/V gadgets. The likelihood of writing to any single address a couple hundred thousand times in an 8-GB MP3 player or camera memory is so remote that all skepticism is removed. Flash cards in these devices just aren't worked all that hard.

Flash-based SSDs (versus spindle hard drives—HDDs) is another story and still a work in progress. Like the horse-versus-car scenario, there's no doubt in my mind that in the long run SSDs will win, and at least for laptops and subnotebooks, the transition will be sooner than later. SSDs survive hostile environments better than mechanical systems—HDDs don't like extreme high or low temperatures. (Ever try starting your laptop after it's been sitting in the trunk overnight during the winter?) Higher random access data transfer speeds over equivalent-size hard drives—still a lot of testing going on but electrons beat mechanical rotation. High reliability—beyond environmental attributes, SSD manufacturers quote significant life spans in normal use. More energy efficient than HDDs—this appears to depend on brand names and the SSD generation technology.

Still, for a graybeard who knows that flash chips have write cycle limitations, stuffing a $1,000 SSD in my next notebook so that all forms of Microsoft bloatware can disk cache the heck out of it until it turns into silicon dust is a hard sell. But, old dogs can learn new tricks, and disk endurance is one of them. HDDs are good, but SSDs can ultimately be better.

Writing to flash in an SSD is not the same as writing to a raw flash chip. Specifically because they incorporate life-cycle limited technology, SSDs have controllers that count the number of times that memory blocks are written to and dynamically reallocate physical blocks transparently to spread data across the whole memory space. Basically, you'd have to write a file the size of the entire SSD for whatever thousands of write cycles to hit the endurance life of any one cell. The controller provides many layers of virtualization between the software and silicon.

Don't get me wrong. HDDs still have a long life, and it may be years before the curves cross. The lowest cost/MB, high memory density, large data caches, and decent energy efficiency continue to extend their design life. Still, the handwriting is on the wall, and I can see the future. At least for one old dog, my greatest fear in using SSDs has been addressed.

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™

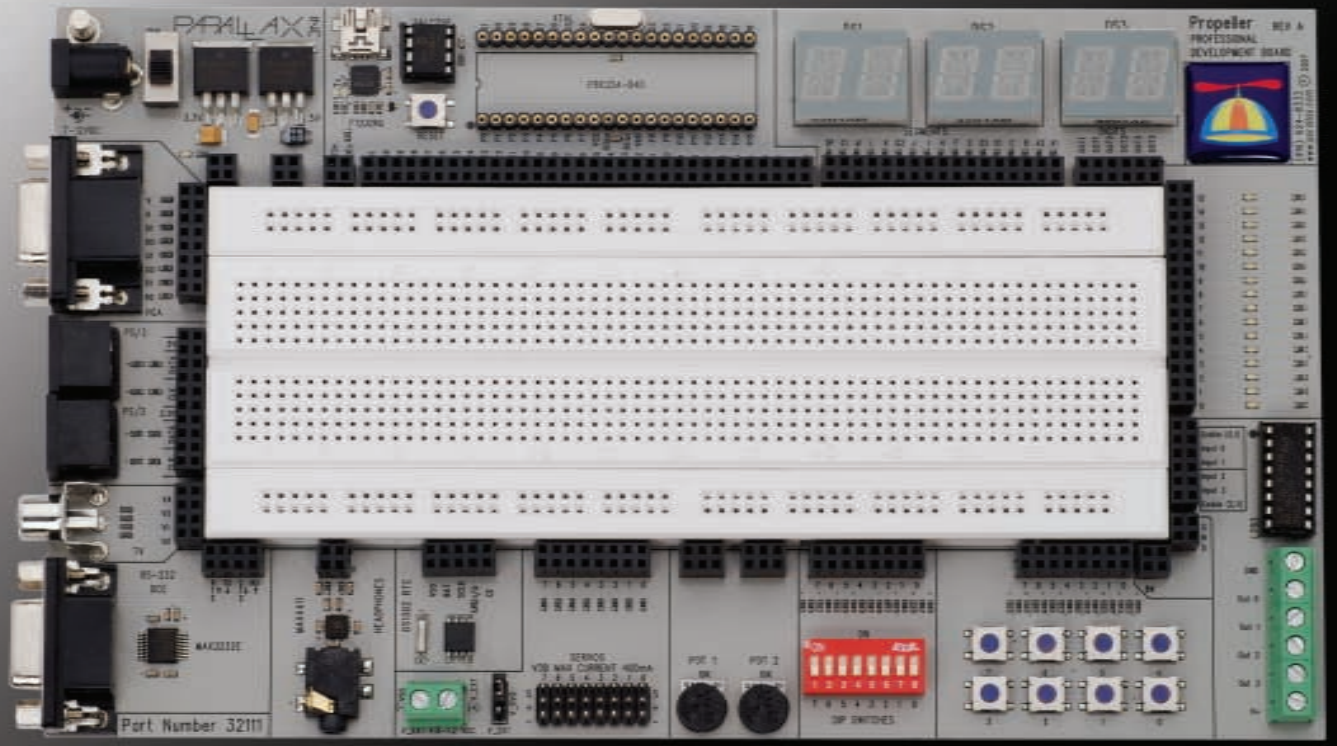
See our Memory Emulators from $179.00

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.



new! Propeller Professional Development Board (PPDB)

The Propeller Professional Development Board (PPDB) is a high-quality, fully-integrated development platform for the Propeller Chip. A wide variety of typical I/O (LEDs, buttons, etc.) devices and circuitry are built into the PPDB, providing the developer with an ideal platform for rapid Propeller Chip development. A 40-pin socket allows easy development and replacement using a through-hole version of the Propeller Chip (sold separately).

Features of the PPDB Include:

- 40-pin DIP socket for Propeller chip (#P8X32A-D40 sold separately)
- 32 K EEPROM (24LC256 included) in socket
- Socketed Crystal (5 MHz included)
- 6-digit, 16-segment LED Display (Blue, RHDP)
- 16 blue LEDs
- L293D quad push-pull driver
- 8 push-buttons
- 8 DIP switches
- 2 potentiometers (10K)
- 8 servo headers with selectable internal/ external voltage source
- DS1302 real-time clock with backup battery input (battery not included)
- Stereo headphone amplifier with 1/8" (3.5 mm) stereo phone jack
- RS-232 line driver (MAX3232E)
- RCA jack for TV/Composite video output
- 2 PS/2 connectors for mouse/keyboard interfacing
- VGA connector
- 2.1 mm barrel power jack and power switch
- 5V and 3.3V power supply connections
- Ground Terminal
- On-board USB Interface (mini-B, 5 pin)

Note: The USB cable (mini-B) and power supply are sold separately. We recommend the 12 V/1 A power supply (#750-00007) for the PPDB.

PARALLAX INC
www.parallax.com

Order the new **Propeller Professional Development Board** (#32111; $169.99) online at www.parallax.com or call our Sales Department toll-free 888-512-1024 (M-F, 7am-5pm, PDT).

Propeller, Parallax and the Parallax logo are trademarks of Parallax Inc.