

CIRCUIT CELLAR

THE MAGAZINE FOR COMPUTER APPLICATIONS

#232 November 2009

ANALOG TECHNIQUES

Low-Distortion FSK Generation Explained

Build a Precision Linear Encoded Display

Embedded Linux-Based Vehicle Tracking

Device Management via the SNMP

Repurpose a Development Platform

Parameter Passing in C Language



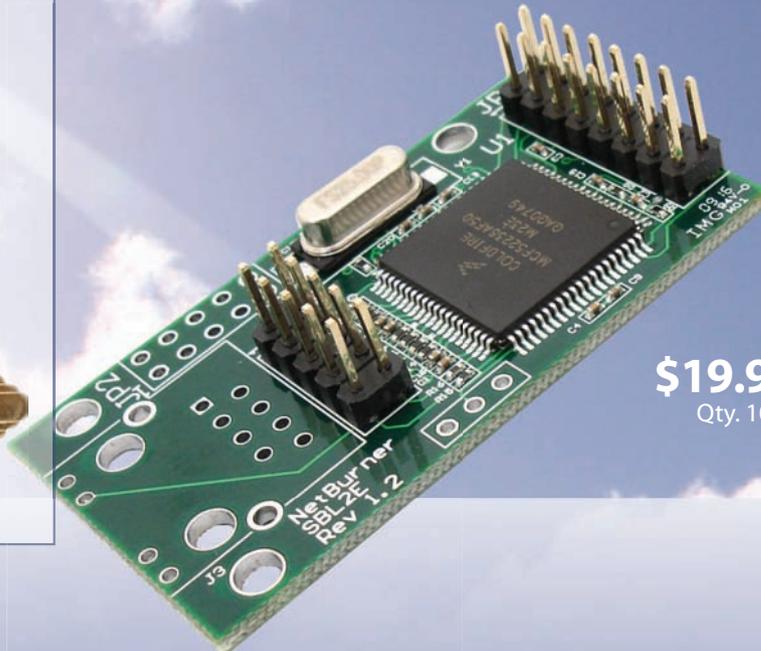
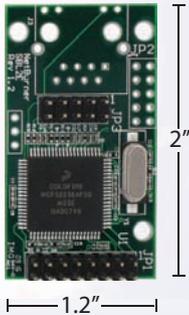
A Low Cost Solution for
Industrial Serial to Ethernet
with Digital I/O and Analog to Digital Converters

Hardware Features

- Two TTL serial ports
- 10/100 Mbps Ethernet
- Up to 10 digital I/O
- Four 12-bit A/D converters
- Operating temperature -40 to 85 C
- Dimensions: 1.2" x 2"
- 32-bit performance

Software Features

- No programming required
- TCP / UDP / Telnet / HTTP modes
- DHCP / static IP support
- Web-based configuration



\$19.95
Qty. 1000

The all-new
NetBurner SBL2e
INDUSTRIAL SERIAL TO ETHERNET SOLUTION

A NetBurner Complete Hardware & Software Solution

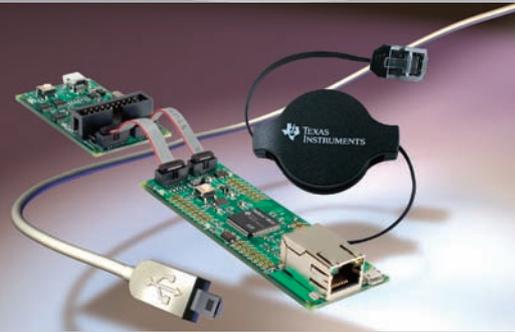


Board Part Number | SBL2e-200IR
Information and Sales | sales@netburner.com
Web | www.netburner.com
Telephone | 1-800-695-6828



Stellaris®

Connected. Versatile. Cost-effective.



Stellaris LM3S9B90 Ethernet Evaluation Kit

featuring USB, Ethernet
and CAN

\$99



Stellaris LM3S8962 Evaluation Kit

featuring integrated Ethernet
and CAN

\$89



Stellaris LM3S3748 Evaluation Kit

featuring USB Host/Device

\$109

Stellaris® Means:

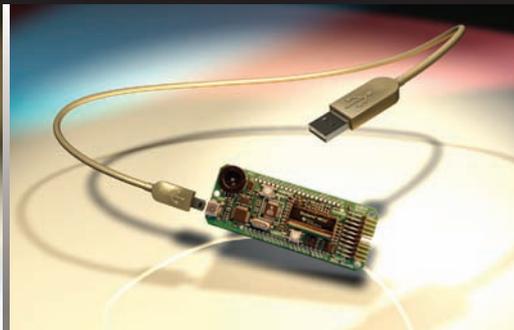
- Deterministic ARM® Cortex™-M3 cores
- Real-time multi-tasking MCUs
- Advanced interrupt-driven software with Stellarisware® software



Stellaris LM3S6965 Ethernet Evaluation Kit

demonstrating an embedded
web server out-of-the-box

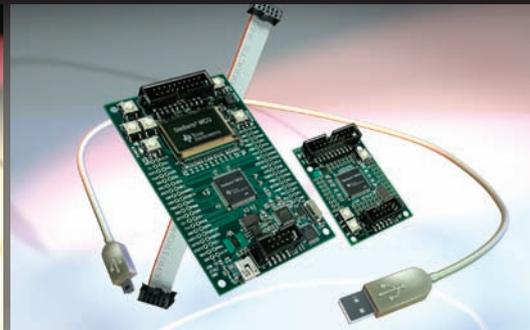
\$69



Stellaris LM3S811 Evaluation Kit

featuring basic embedded
control

\$49



Stellaris LM3S2965 CAN Evaluation Kit

demonstrating a CAN network
out-of-the-box

\$79



Awaken the **W**onderful **W**izard in **YOU!**



iMCU
**DESIGN
CONTEST**

Nov 2009 ~ Apr 2010

Your creativity and design project idea could win
you a share of **\$15,000** in Cash Prizes
and Recognition in Circuit Cellar magazine.

For details, visit www.WIZwiki.net



iMCU7100EVB Contest Special Price: 49.00 USD
Co-sponsor Official Sample Purchase: www.FutureElectronics.com

Internet Embedded MCU

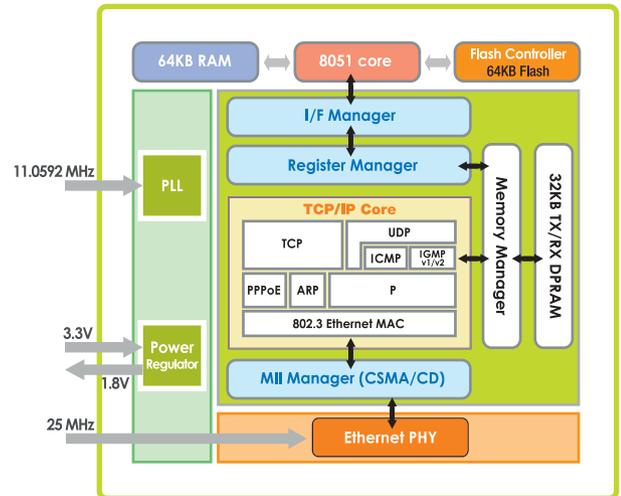
New Paradigm of MCU for Internet Devices

iMCU W7100



Easier, Faster & More Stable!

- H/W TCP/IP + MAC + 10BaseT/100Base TX Ethernet PHY
- Fully software compatible with standard 8051
- Pipelined architecture with standard 8051
- 64K Bytes e-Flash memory
- 64K Bytes SRAM memory
- 100-LQFP Lead-free package
- Single Chip Serial-to-Ethernet Gateway



iMCU7100EVB



RS232/RJ45/32GPIO/LCD

Open Source Codes

- TCP lookback
- UDP lookback
- DHCP Client
- DNS Client
- Serial-to-Ethernet Gateway
- Internet LCD display
- HTTPC
- Telnet

Available at www.WIZwiki.net

You're Covered

If you need it, we've got it. Whether you are looking for solutions to design problems, helpful programming tips, ideas for exciting new projects, or insight on the new technologies coming down the pipe, *Circuit Cellar* has you covered. That is our main objective, and it always has been. Ask any staffer who has been around since Issue 1 and you'll find that we've always considered ourselves a "one-stop shop" for everything embedded. But this month we've outdone ourselves. How? We literally have something for *everyone*. Linux lovers, C language devotees, surveillance enthusiasts, power-hungry design engineers, innovative embedded network developers, signal theory buffs, and do-it-yourselfers who love popping open MCU-based gadgets and repurposing them for exciting new applications—we've got you covered! Heck, we even have a project for those of you who have metal lathe shops next to your circuit cellars!

In "Embedded Linux-Based Location System," James Stephanick explains how to build a vehicle location device (p. 16). An ARM-based USB network storage link, a prepaid GSM cell phone, and a USB-GPS device comprise this handy location system.

Turn to page 26, where Noel Rios presents a switched-mode power supply design. With it, he can adjust voltage from 1 to 20 V and current from 0 to 1 A. He does this by entering the values on a keypad.

When precision matters, consider a linear encoder. Carl Stoesz explains how he mounted an optical linear encoder to a metal lathe that he uses to create small precision parts (p. 36). He uses the setup to send speed and displacement information to an LCD.

There's a big difference between "hacking" and "repurposing." Hacking is about deconstruction and investigation. Repurposing takes things a bit further: you take apart a design and use its features to create something new or improved. On page 44, Cyrille de Brebisson explains how to repurpose an HP financial calculator and use it for a variety of embedded design apps.

Check out George Martin's "Passing Parameters" for yet another lesson from the trenches of C coding (p. 50). He makes a convincing argument for passing parameters in the first place, and then describes how to do so and get results.

Do you remember Dennis Seguire's 2006 *Circuit Cellar* 194 article titled "Simplified FSK Signal Detection"? There he presented an efficient method for detecting FSK signals. This month he describes how to generate FSK modulation with low-distortion and low-transition phase error (p. 54).

On page 60, Tom Cantrell presents a truly exciting concept: wireless power on a microelectronic scale. That means wireless power delivery! The future is now.

And lastly, we arrive at a topic important to everyone: device management. Jeff Bachiochi explains how to monitor networked devices via the simple network management protocol (SNMP). With the SNMP, you can keep build an MCU-based managed device to monitor numerous networked modules and log critical data (p. 66).

cj@circuitcellar.com



CIRCUIT CELLAR®

THE MAGAZINE FOR COMPUTER APPLICATIONS

FOUNDER/EDITORIAL DIRECTOR

Steve Ciarcia

CHIEF FINANCIAL OFFICER

Jeannette Ciarcia

MANAGING EDITOR

C. J. Abate

MEDIA CONSULTANT

Dan Rodrigues

WEST COAST EDITOR

Tom Cantrell

CUSTOMER SERVICE

Debbie Lavoie

CONTRIBUTING EDITORS

Jeff Bachiochi
Robert Lacoste
George Martin
Ed Nisley

CONTROLLER

Jeff Yanco

ART DIRECTOR

KC Prescott

NEW PRODUCTS EDITOR

John Gorsky

GRAPHIC DESIGNERS

Grace Chen
Carey Penney

PROJECT EDITORS

Gary Bodley
Ken Davidson
David Tweed

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.rakoczypphoto.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 \$29.95, Canada/Mexico \$34.95, all other countries \$49.95. Two-year (24 issues) subscription rate USA and possessions \$49.95, 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 © 2009 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.

60 MHz - 1 GHz Starting from \$1,000



WaveAce™
60 MHz - 300 MHz

WaveJet®
100 MHz - 500 MHz

WaveSurfer®
200 MHz - 1 GHz

Experience the New LeCroy Oscilloscopes*

Debug, verify, characterize, troubleshoot, analyze — oscilloscopes do it all, but LeCroy's breakthrough oscilloscopes do more. More measurements, more math, and more time-saving tools to easily make measurements in even the most challenging situations.

LeCroy

*To learn more about LeCroy's full line of oscilloscopes, including the 30 GHz WaveMaster 8 Zi, the world's fastest real-time oscilloscope, visit www.lecroy.com or call 1-800-5-LeCroy

INSIDE ISSUE

232

BONUS CONTENT

Measuring Propagation Delay with a Universal Counter

November 2009 • Analog Techniques



Vehicle Location Made Simple, p. 16

16 Embedded Linux-Based Location System
James Stephanick

26 Energize a Circuit
An Innovative Switched-Mode Power Supply Design
Noel Rios

36 Precision Linear Encoder Display
Carl Stoesz

44 How to Repurpose a Development Platform
Cyrille de Brebisson

54 Low-Distortion FSK Generation
Dennis Seguire



Linear Encoder Display, p. 36

50 LESSONS FROM THE TRENCHES
Passing Parameters
George Martin

60 SILICON UPDATE
Power Pitcher
Wireless Power on a Microelectronic Scale
Tom Cantrell

66 FROM THE BENCH
Device Management
Monitor Networked Devices Via SNMP
Jeff Bachiochi



Wireless Power Delivery, p. 60

TASK MANAGER 4
You're Covered
C. J. Abate

NEW PRODUCT NEWS 8
edited by *John Gorsky*

TEST YOUR EQ 15

CROSSWORD 74

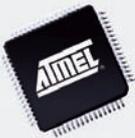
INDEX OF ADVERTISERS 79
December Preview

PRIORITY INTERRUPT 80
Keeping the Lights On
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/>

ATMEL
Everywhere You Are®

LPC314x DEVELOPMENT KIT

The LPC3141 Developer's Kit is a ready-to-use platform that will let you explore the many features of NXP's LPC314x microcontroller series in general and the LPC3141 in particular. The kit is perfect for running Linux with its large on-board RAM and flash memory, network and graphics capabilities. Using the kit as a prototyping platform makes it possible for you to focus on the value-added application. The kit includes all the software components needed to get up and running with software development on day one. It is easy to transfer the application and the OEM board from the prototyping environment into your final target application.

The following resources are available at the kit's support site: a user's manual, LPC3141 OEM board schematics, LPC31xx base board schematics, LPC31xx base board component placement, links to datasheets for key components, a multimedia presentation about how to get started with the board, and USB DFU drivers for Windows. NXP's Common Driver Library framework has been used to develop a number of sample applications for the LPC3141 OEM board.

Included in the kit are an LPC3141 OEM board, an LPC31xx base board v2.0, a USB cable, and a headset with microphone. Also included is a serial number that gives you access to the support site.

The developer's kit costs about \$260.



Embedded Artists AB
www.embeddedartists.com

COST-EFFECTIVE, TWO-CHIP PLATFORM FOR SMALL ENGINES

Faced with world-wide emissions regulations and rising gas prices, the small engine market is migrating from mechanically carbureted systems to cleaner, more efficient electronic control and electronic fuel injection. To address the global need for these eco-friendly designs, the MC233812 combined with a 16-bit MCU provides a cost-effective platform solution that's ideal for small engine control.

A reference design consisting of the MC233812, an advanced analog integrated circuit optimized to control fuel and spark in EFI and electronic carburetion (e-carb) systems, and a Freescale 512 MCU is now available. This comprehensive small engine control platform helps lower harmful emissions and reduces system complexity, bill of materials, and manufacturing cost, while accelerating time to market.

The device combines a voltage regulator, fuel injector driver, and ignition pre-driver, as well as a relay and lamp driver, a watchdog timer, and a reset generator—all optimized for small engine control. This exceptional integration eliminates up to a dozen discrete components and reduces the circuit board area for engine makers while also helping customers enhance the quality and reliability of their products.

The off-the-shelf MC233812 reference design allows developers to evaluate a two-chip solution based on the MC233812 and 512P MCU and emulate a complete small engine electronic control unit. The 512P delivers the performance benefits of a 16-bit MCU while supporting the entry-level pricing, low power, electromagnetic compatibility, and code-size efficiencies of an 8-bit MCU.

The MC233812 is a commercial-grade device available now for production at a suggested resale price of \$1.70 in 10,000-unit quantities.



Freescale Semiconductor
www.freescale.com

NEW PRODUCT NEWS

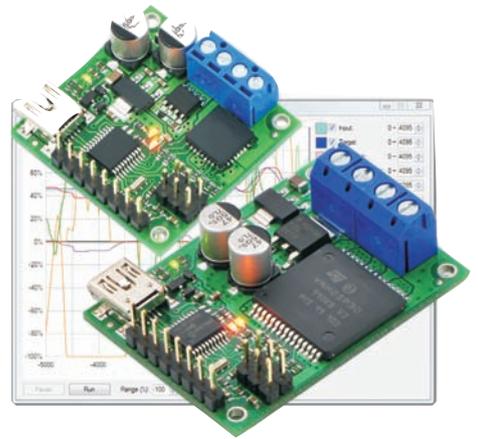
Edited by John Gorsky

USB MOTOR CONTROLLER WITH FEEDBACK

The **jrk** line of USB motor controllers includes highly configurable, versatile devices that make it easy to add open- or closed-loop control of brushed DC motors to your computer- or microcontroller-based project. The jrk supports four interface modes: USB for PC-based control, logic-level (TTL) serial for use with embedded systems, analog voltage for simple potentiometers and joysticks, and RC pulse for radio control systems. It can perform open-loop speed control, closed-loop position control with analog voltage feedback to make your own servos, and closed-loop speed control with frequency feedback from a tachometer. The jrk 21v3, the smaller of the two units currently available, has an operating range of 5 to 28 V and can deliver 3-A continuous output (5 A peak). The jrk 12v12, which is the more powerful of the two, has an operating range of 6 to 16 V and can deliver a continuous output of 12 A (30 A peak). Both devices can handle transients of up to 40 V.

A free configuration program (Windows XP and Vista compatible) is available for calibrating your system. Real-time plots of variables—such as control input, feedback, motor output, and current draw—make it easy to fine-tune settings, such as PID constants, acceleration, and current limit for your application.

The unit price is **\$49.95** for the jrk 21v3 (item #1392) and **\$99.95** for the jrk 12v12 (item #1393).



Pololu Corp.
www.pololu.com

HIGH-VOLTAGE LINE DRIVER WITH INTEGRATED DC-DC CONVERSION

The **ZIOL2401** is a high-voltage line driver with IO-Link functionality. The first member of a family of high-voltage line drivers, the chip offers two independent I/O-channels that meet the physical layer requirements of sensor and actuator systems used in factory automation applications. The ZIOL2401 combines an advanced two-channel IO-Link driver with a DC-DC converter capable of producing up to 15 V. This can power external sensors, as well as the ZIOL2401, and it saves space and cost in board designs. The IO-Link drivers are fully programmable, including slew rate and current limitation, and offer complete diagnostics as well as over temperature protection.

The maximum current driven by the output stages can be set to a value that best fits the customer's application. All configuration parameters are stored in an on-chip EEPROM and automatically loaded during power-up. The ZIOL2401 reports all status information, such as an overload condition or over-temperature. A SPI supports the access to configuration and status registers. In addition to IO-Link device and master ports, the device supports standard I/O and works with both 3.3- and 5-V systems. It also offers wake-up detection and fast direct-event signaling.

Unit prices start at **\$5** in volumes of 1,000 devices or more.

ZMDI AG
www.zmdi.com



MAGNETIC ROTARY IC

The **A55215**, a new dual-die, high-precision, analog sine and cosine output, magnetic encoder IC targets redundant motion-sensing solutions for automotive microcontroller-based systems. The IC is also ideal for safety-challenging applications such as electronic power steering systems.

The amplitude of the A5215's output signal is programmable via sensitivity and gain settings. Due to its selectable output mode signals—raw, inverted, or DC-referenced—the A55215 can fit a variety of conventional motion-sensing systems which require measurement redundancy for safety reasons.

Based on a patented stacked-die solution, the dual die A55215 provides excellent phase matching and very low sensitivity drift over the full ambient temperature range of -40° to 150°C . The A55215 is available in a small TSSOP 14 package and operates at a 5-V supply voltage.

The A55215 is priced at **\$6.40** each in 1,000-unit quantities.

austriamicrosystems
www.austriamicrosystems.com

Dual Die Motion Sensing IC for
Microcontroller-based Systems

Analog Sin/Cos output

150°C ambient
temperature

Automotive AEC-Q100 qualified

austriamicrosystems
a leap ahead in analog

8-BIT MICROS ON ULTRA-LOW-POWER TECHNOLOGY PLATFORM

The **STM8L** product family comprises 26 devices in three lines spanning a broad spectrum of performance and function. They combine high-performance 8-bit architecture with ultra-low-power innovations to save power in active modes as well as when idle.

The lines are built on an ultra-low-power technology platform using a dedicated, proprietary 130-nm process optimized for ultra-low leakage. Unique advantages include full CPU performance up to the maximum operating frequency throughout the entire supply voltage range, from 1.65 to 3.6 V. Moreover, using an on-chip regulator, power consumption is independent of supply voltage, thereby providing extra flexibility and simplifying design.

Other innovations include low-power embedded nonvolatile memory and multiple power-management modes including a 5.4- μ A Low-Power Run mode, 3.3- μ A Low-Power Wait mode, 1- μ A Active Halt with full RTC, and 350-nA Halt mode. The device can wake from Halt within 4 μ s, enabling frequent use of the lowest-power mode. Low-power peripherals include the sub-1- μ A RTC and Automatic Wake-Up. Overall, this platform enables dynamic current consumption as low as 150 μ A/MHz.

The STM8L101 line is the entry point of the STM8L ultra-low-power 8-bit portfolio. Two other product lines making up the product range are scheduled for production in early 2010. The STM8L15x line adds extra features, including up to 32 KB on-chip flash and up to 2 KB SRAM, external crystal/clock capability, enhanced reset features, and support for DMA. There are also additional peripherals, on-chip EEPROM with true Read While Write (RWW) capability. The third line, the STM8L152, features an integrated LCD controller.

Family pricing starts with the STM8L101 4-KB flash version in QFN20 3 x 3 mm low-profile package at **\$0.68** in 10,000-unit quantities.



STMicroelectronics
www.st.com

NPN

N e w n e s P r e s s

Got a PIC Question? We have the answers!



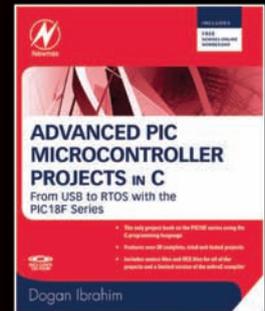
By Lucio Di Jasio
ISBN: 9780750687096
\$59.95



By Martin Bates
ISBN: 9780750689601
\$39.95



By Lucio Di Jasio
ISBN: 9780750682923
\$49.95



By Dogan Ibrahim
ISBN: 9780750686112
\$39.95



Look for the latest titles from Newnes Press to help you maintain your competitive edge at Amazon.com or your favorite online retailer.

www.newnespress.com

Register for our e-news at newnespress.com

Receive our best discounts · Hear about books before they publish
Access to free sample chapters, video tutorials and more!

PROPELLER PROTOTYPING SYSTEM

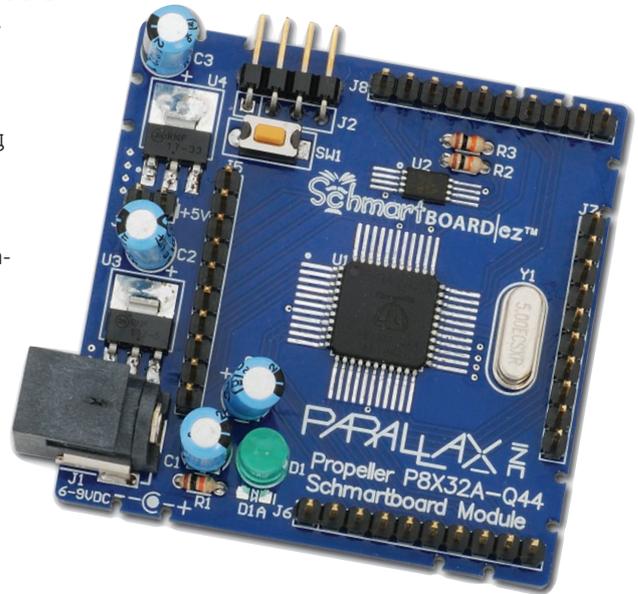
Parallax has partnered with SchmartBoard to create an easy Propeller prototyping system in the form of a kit. The **P8X32A-Q44 SchmartBoard Kit** (part number 27150) allows you to easily work with the surface mount Propeller chip.

This kit is a perfect starting point. The SchmartBoard technology makes surface-mount soldering easy. Once completed, the board will host the powerful microcontroller on this convenient development platform, allowing access to all 32 I/O pins of the multicore chip. The kit includes surface-mount and through-hole package types for some components, offering a soldering choice and challenge.

The kit features the Propeller P8X32A-Q44 microcontroller along with header sockets for easy access to the Propeller I/O pins. Also included is 512 KB of EEPROM for program and data storage, 3.3- and 5-VDC on-board regulators, surface-mount and through-hole components to populate the board, a barrel jack for a 2.1-mm center-positive 6- to 9-VDC power supply (not included), and a four-pin programming header for the Prop Plug (also not included).

The P8X32A-Q44 SchmartBoard kit costs **\$39.95**.

Parallax, Inc.
www.parallax.com



NPN

FRONT PANELS & ENCLOSURES

Customized front panels can be easily designed with our free software Front Panel Designer

- Cost-effective prototypes and production runs
- Wide range of materials or customization of provided material
- Automatic price calculation
- Fabrication in 1, 3 or 7 days

Sample price: \$43.78 plus S&H

FRONT PANEL EXPRESS www.frontpanelexpress.com
 (206) 768-0602

RF Specialists

Distributors of Low-Power RF modules, Zigbee, Bluetooth, WiFi, GPS, GSM/GPRS

FCC Part 90 Compliant

manufactured by **RADIOMETRIX**

USX2 NBFM Multi-channel UHF Transceiver with Programmable RF Power	UHX1 NBFM Multi-channel 500mW VHF Transceiver	
NEW! QPT1 VHF Narrow Band FM 2 Watt Multi-channel transmitter	NEW! QPX1 2 Watt Multi-channel VHF Transceiver	
SMX1 VHF Narrow Band FM Multi-channel Transceiver	RX1M VHF Narrow Band FM Multi-channel Radio Receiver	LMR1 VHF NBFM Low Cost Multi-channel Radio Receiver

LEMOS INTERNATIONAL www.lemosint.com
 866.345.3667 sales@lemosint.com

TOUCH AND MOTION SENSING USB STARTER KIT

The **MPLAB Starter Kit for PIC18 Microcontrollers** (part number DM180021), contains all of the components, documentation, schematics, debugging, and programming capabilities needed to evaluate the high-performance PIC18 8-bit microcontroller (MCU) family in low-power USB and touch-sensing applications. The kit includes a development board featuring a 44-pin PIC18F46J50 MCU, mTouch capacitive touch sensing, accelerometer, 64-KB flash program memory, and much more.



Hardware debugging and programming come integrated with the kit, meaning an additional debugger or programmer is not required. The kit is completely USB-powered, so a separate power supply does not need to be purchased. Pre-programmed touch-sensing code, as well as a free USB application stack and bootloader, eliminate the need for engineers to develop their own code. Additionally, the board includes demo application software that enables it to read from the memory card (MSD), or function as a USB mouse or joystick (HID), or a simple COM port (CDC). By simply scrolling through code options using capacitive-touch buttons, you can demonstrate the PIC18 MCUs with a MicroSD memory card, potentiometer, acceleration sensor, and OLED display.

The Starter Kit includes the BMA150 digital triaxial accelerometer, which enables the board to detect acceleration in three perpendicular axes for sensing the board's tilt and motion. The board ships with demo code that enables it to function as a USB mouse or joystick using the accelerometer. The accelerometer allows the cursor to move slowly when the board is tilted a small amount, and quickly when there is a steeper tilt.

The evaluation kit is available today for less than \$60.

Microchip Technology, Inc.
www.microchip.com

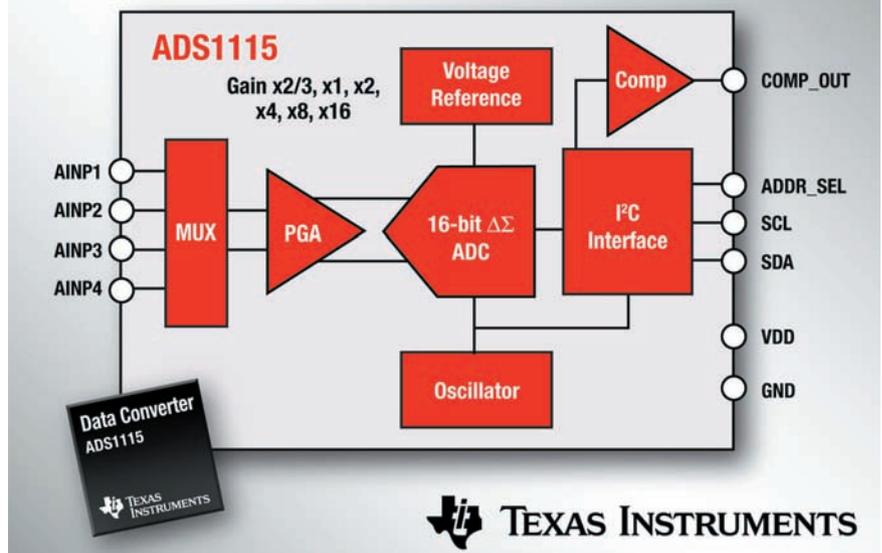
16-BIT ADC FOR INDUSTRIAL AND CONSUMER APPLICATIONS

The **ADS111x** is a new family of 16-bit analog-to-digital converters packaged in a leadless QFN measuring 2.0 mm x 1.5 mm x 0.4 mm—70% smaller than the nearest competition. In addition to significant system space savings, the ADS1115 family provides product options for scalable integration to reduce component count and simplify design. This family supports battery monitoring, portable instrumentation, industrial process control, smart transmitters, medical instrumentation, and other industrial and consumer systems.

Designed for precision, power efficiency, and ease of implementation, the devices perform conversions at programmable data rates up to 860 samples per second, consuming just 150 μ A (typical) of supply current and operating down to 2 V. The ADS1115 is the most highly integrated member of the new family, incorporating an oscillator, low-drift reference, programmable gain amplifier, comparator, and four-channel input multiplexer into a tiny package. This family also includes 12-bit versions for increased flexibility.

Pricing for the ADS1115 starts at \$2.25 in quantities of 1,000 units. The ADS1114 starts at \$2 and the ADS1113 starts at \$1.85.

Industry's smallest 16-bit ADC



TEXAS INSTRUMENTS

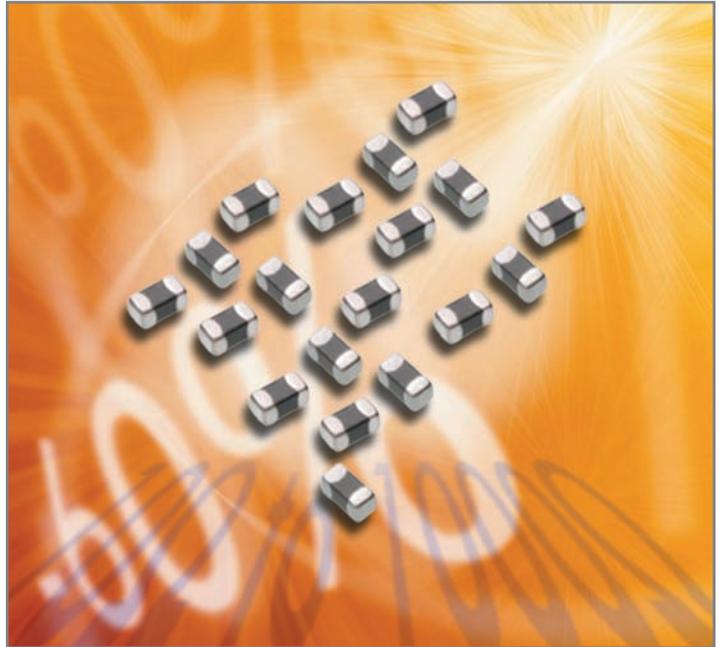
CHIP FERRITE BEADS OFFER 60% LOWER DC RESISTANCE

The BLM15AX_5N series is an advanced ferrite bead that lowers DC resistance by up to 60% compared to previous models. As part of an expanding electromagnetic interference filter (EMIFIL) line, the chip ferrite bead was developed with new ceramic materials and processing technologies that improve performance and reduce size in order to provide power and space savings in portable electronics.

Compared to its predecessors, the BLM15AX_5N chip ferrite bead has a DC resistance as low as 0.02 Ω , a reduction of 60%, while the corresponding maximum rated current has increased from 1,000 to 1,740 mA. The series's low DC resistance also helps reduce power consumption for battery-powered applications. Developed in a compact 0402-size package (1.0 mm x 0.5 mm x 0.5 mm), it is ideal for designers seeking to reduce board space without performance degradation in portable equipment. The ferrite bead's wide impedance ranges from 10 to 1,000 Ω , with an operating temperature range of -55° to 125°C.

The BLM15AX_5N series is available for \$0.016 and lead times are approximately six to eight weeks.

Murata Electronics North America
www.murata.com



INDUSTRIAL-OUTPUT SIGNAL CONDITIONERS

The MAX15500/MAX15501 industrial analog-output signal conditioners offer the industrial market a unique combination of higher functionality, greater integration, and smaller total solution size than competitive devices. They feature unprecedented integration: a programmable voltage-output amplifier and a current-output amplifier; ± 10 V and 4- to 20-mA outputs operating directly from a 24-V field supply; ± 35 -V circuit protection; extensive error-reporting capability; and daisy-chain flexibility.

The devices save board space and design time in 4- to 20-mA and bipolar/unipolar voltage-output applications. They are ideal for PLC

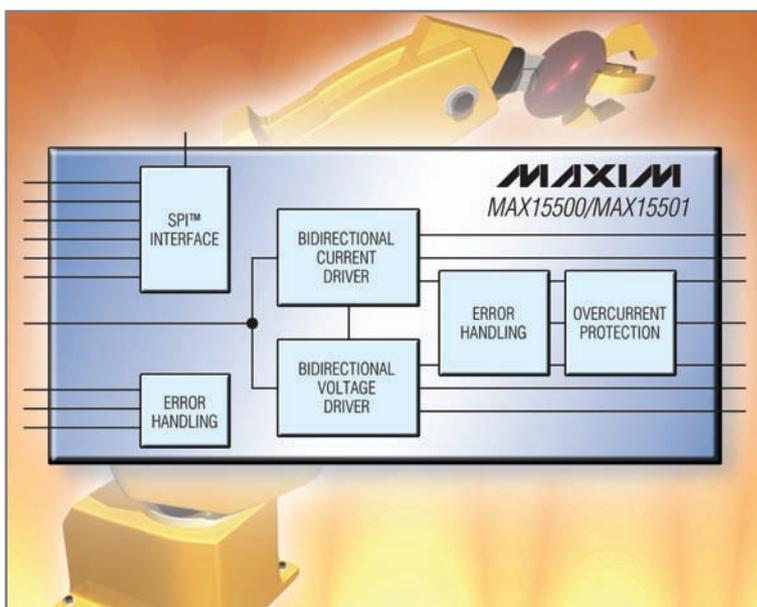
analog outputs, distributed I/O, embedded systems, and industrial control and automation. Their high integration adds flexibility for designs. Use a single device for channel-isolated applications or multiple devices for group-isolated applications.

Typical competitive signal conditioners need precision-matched resistors to track voltages well over temperature. The MAX15500/01's integrated voltage amplifier and voltage-to-current converter eliminate the need for those additional resistors. Only the current gain-setting resistor is external to the device. The devices offer extensive fault detection, including open-circuit load in current mode, short-circuit load in voltage mode, over-temperature, and supply brownout. Both intermittent and permanent load faults can be detected. If a fault occurs, the open-drain fault-interrupt pin will be pulled low. The status of the fault bits is available through the SPI interface.

With a higher ± 24 -V supply voltage, the MAX15500/01 can handle a typical load of 750 Ω in current mode. With ± 32.5 -V supplies, this increases to 1,000 Ω . Additionally, the MAX15500/01 outputs, power-supply inputs, and sense inputs are protected against overcurrent conditions, a short to ground, or supply voltages up to ± 35 V.

Prices start at \$2.96 (1,000-up, FOB USA).

Maxim Integrated Products
www.maxim-ic.com



FANLESS FLEXIBLE CANbus WINDOWS CONTROLLER

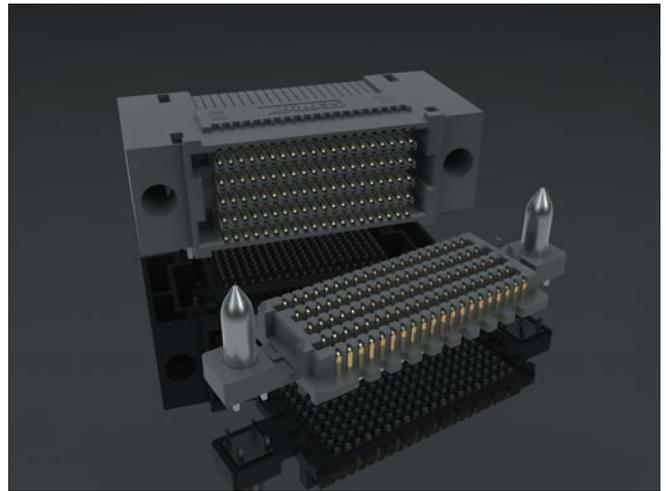
The CANatomBOX is a versatile, compact embedded controller targeted for embedded CANbus control applications. Based on Intel's 945GME 1.6-GHz Atom processor, this controller box is application-customizable even in small quantities, with a wide variety of hardware and software options. With 512-MB SD-RAM and 2-GB in CompactFlash, this system boots quickly, has passive cooling, with no moving parts at all. DIN-rail mounted units can be placed almost anywhere, due to the very small dimensions: (3.7" x 6.3" x 9.0" for standard version; 5.7" x 6.3" x 9.0" for two-PCI slot version).

Standard interface connections for CANatomBOX are provided on the front panel, and Windows Vista/XP/CE, Linux (or other choices such as QNX or VxWorks) can be preinstalled together with other application software to exactly fit customer needs. The CoDeSys SoftPLC runtime environment is also supported on this platform.

CANatomBOX features front-panel availability for two independent 10/100Mbit/s Ethernet ports, status LED's, an RS-232 serial port, two opto-isolated CAN ports, two USB2.0 ports, I/O lines, etc. Units are 10- to 28-VDC-powered, making them ideal for battery-powered operation. All inputs feature ESD protection of 2 kV.

Made in Europe by Janz AG, a leading industrial control and CANbus manufacturer, the CANatomBOX is now available exclusively in North America from Saelig Company, with units starting at \$1,399.

Saelig Co.
www.saelig.com



RIGHT-ANGLE HIGH-DENSITY INTERCONNECTS

The SEARAY high-density open-pin field-interconnect family is a new line of right-angle sockets for perpendicular and high speed micro backplane applications. The SEAF-RA series have a 0.050" x 0.050" grid array for maximum grounding and routing flexibility and are optimized to reduce skew and impedance mismatch. They feature rugged Edge Rate contacts, which are designed for superior signal integrity, reduced broadside coupling and crosstalk, and high cycles.

SEARAY right-angle sockets are currently available in four-row (up to 200 contacts) and six-row (up to 300 contacts) configurations. Eight-row (up to 400 contacts) and 10-row designs (up to 500 contacts) along with a right-angle terminal (SEAM-RA series) are in the final stages of development.

This system mates with SEARAY surface mount terminal series, which has optional guide posts for blind mating, and a choice of tin-lead or lead-free solder charged termination. Press-fit tail terminations are currently in development.

SEARAY surface-mount systems with up to 500 single-ended or 125 differential I/Os, coax cable assemblies with 180 or 240 I/Os, and high-density flex jumpers are also available.

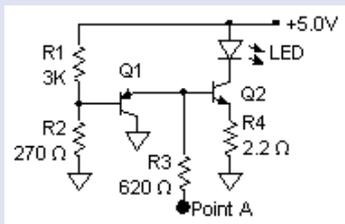
Pricing begins at \$0.05/pin in 1,000-piece quantities.

Samtec, Inc.
www.samtec.com

NPN

STATEMENT REQUIRED BY THE ACT OF AUGUST 12, 1970, TITLE 39, UNITED STATES CODE SHOWING THE OWNERSHIP, MANAGEMENT AND CIRCULATION OF CIRCUIT CELLAR, THE MAGAZINE FOR COMPUTER APPLICATIONS, published monthly at 4 Park Street, Vernon, CT 06066. Annual subscription price is \$29.95. The names and addresses of the Publisher, Editorial Director, and Managing Editor are: Publisher, Sean Donnelly, 4 Park Street, Vernon, CT 06066; Editorial Director, Steven Ciarcia, 4 Park Street, Vernon, CT 06066; Managing Editor, Carmine J. Abate, 4 Park Street, Vernon, CT 06066. The owner is Circuit Cellar, Inc., Vernon, CT 06066. The names and addresses of stockholders holding one percent or more of the total amount of stock are: Steven Ciarcia, 4 Park Street, Vernon, CT 06066. The average number of copies of each issue during the preceding twelve months is: A) Total number of copies (net press run) 19,292; B) Paid/Requested circulation (1) Mail subscriptions: 7,211; (3) Sales through dealers and carriers, street vendors and counter sales: 4,974; (4) Requested through other USPS classes: 211 C) Total paid/requested circulation: 12,396; D) Nonrequested distribution (by mail and outside the mail) (1) Outside county nonrequested (sample copies): 1,764; (4) Nonrequested copies distributed outside the mail (trade shows, other sources) 1,666; E) Total nonrequested distribution: 3,430; F) Total distribution: 15,826; G) Copies not distributed: 3,466; H) Total: 19,292; I) Percent paid/requested: 78%. Actual number of copies of the single issue published nearest to filing date (October 2009, Issue #231): A) Total number of copies (net press run) 18,010; B) Paid/Requested circulation (1) Mail subscriptions: 6,656; (3) Sales through dealers and carriers, street vendors and counter sales: 5,105; (4) Requested through other USPS classes: 187; C) Total paid/requested circulation: 11,948; D) Nonrequested distribution (by mail and outside the mail) (1) Outside county nonrequested (sample copies): 1,714; (4) Nonrequested copies distributed outside the mail (trade shows, other sources) 995; E) Total nonrequested distribution: 2,709; F) Total distribution: 14,657; G) Copies not distributed: 3,353; H) Total: 18,010; I) Percent paid/requested: 82%. I certify that the statements made by me above are correct and complete. Sean Donnelly, Publisher.

Problem 1—While reverse-engineering an LCD module with LED backlight, a colleague came across this circuit and had trouble figuring it out. Let's see if we can help him out.



Point A connects to one of the external pins of the module and appears to be used for switching the backlight on and off.

First of all, what is the purpose of R4?

Problem 2—What is the purpose of R1, R2, and Q1?

Problem 3—What can you say about the LED current with respect to the actual voltage on point A?

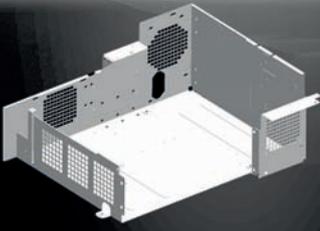
Problem 4—What can you say about the LED current with respect to ambient temperature?

Contributed by David Tweed

What's your EQ?—The answers are posted at www.circuitcellar.com/eq/
You may contact the quizmasters at eq@circuitcellar.com



Factory on Your Fingertip



From Idea to Reality

- > Design
- > Prototyping
- > Production

See Our Differences in

- > Quality
- > Service
- > Price



*Milling *Turning *Grinding *CNC
 *Wire Cutting *Laser *Plasma
 *Water Jet *Plastic Injection
 *Sheet Metal *Gear *SLA
 *FDM *SLS *LOM

CLICK TO MAKE

MachinePIER

Tel: 408-421-9840 Email: sales@machinepier.com
 Website: <http://www.machinepier.com>



PCBCORE

China PCB Supplier

(Prototype thru Production)

- ✓ 1-layer up to 30-layer
- ✓ Cost and quality
- ✓ On time delivery
- ✓ Dedicated service
- ✓ Instant Online Quote & Order

..... Day and Night

No minimum quantity - 1 piece is welcome
Check our low price and save big \$\$\$...

86(571)86795686

sales@pcbcore.com

www.pcbcore.com

Embedded Linux-Based Location System

You can build your own vehicle location device without breaking the bank. This article details how to use a few readily available parts—an ARM-based USB network storage link, a prepaid GSM cell phone, and a USB-GPS device—to build an easy-to-configure location system.

After the second time my car was stolen out of my driveway, I began looking for an inexpensive way to track my vehicle's location. The first solution I implemented was a ham radio

APRS installation. When the car was on, a ham radio and GPS unit were used to beacon the car's location every 3 minutes. Random, privately-owned receivers located seemingly everywhere picked up the beacon and forwarded the car's location to the Internet. By tapping into the APRS network or using a publicly available website, I could view the car's location (and even its path over the previous 24 hours) on Google maps. This worked well. But due to the cost of all the components—including the radio, radio installation, antenna mount, antenna, electronics to modulate the digital data, and the smart controller to turn on/off the radio—it wasn't the most affordable solution (and it requires a ham license).

When searching for commercial vehicle-tracking solutions, I typically found expensive hardware, service contracts, fairly high monthly fees, and devices that were not particularly versatile. At one point, I



Photo 1—The assembled vehicle locator includes a Linksys WLU2 USB Network storage link, a Motorola GSM phone, and a Deluo USB-GPS device.

stumbled across an inexpensive GPS GSM modem targeted at just this type of application. However, its fixed set of features and inability to enable custom processing were shortfalls.

After experimenting with embedded USB controllers for a while, I finally determined that an off-the-shelf Linux solution would be far cheaper, faster to prototype, and easier to modify than microcontroller code. Instead of developing and fabricating a custom solution for vehicle tracking, I used an affordable off-the-shelf Linksys unit I found on the Internet to construct a highly versatile, easy-to-configure, embedded Linux-based vehicle locator. In addition to a 5-V power supply, three devices connected with USB cables complete the setup: a Linksys NSLU2 USB Network storage link (i.e., a network-attached storage, or NAS, device), a Motorola V195 prepaid T-Mobile GSM phone, and a Deluo USB-GPS device (see [Photo 1](#)).

SYSTEM OPERATION

Although my design is simple, it achieves its objective. It has three primary features. One, every time my car starts, the design sends an SMS message to a default phone number with a Google Map link to the car's current location. Two, if the car is running, I can locate it by sending an SMS message (with a password) from any phone. When the unit receives the SMS message and verifies the password, it responds with a Google Map link that indicates the car's current location. Three, I can modify basic configuration parameters using a network connection and an Internet browser. Configuration parameters include the default phone number (to notify me via an SMS message when the car starts), the ability to enable or disable start-up SMS notifications, and the ability to set the authorization password for SMS location requests.

When using an Internet-connected phone—such as an Apple iPhone or Google gPhone—it is quick and easy to read the SMS message. You can then see the car's location on a Google map.

COMPONENTS & ASSEMBLY

The NSLU2 USB network storage link has 8 MB of flash memory and 32 MB of SDRAM. It uses an ARM-based Intel IXP420 processor and runs at either 133 or 266 MHz. By removing a surface-mount resistor, the older 133-MHz version can be modified to run at 266 MHz. The device contains two USB ports and one network jack. Connection points are available for adding a JTAG port, an additional USB port, and serial ports. After it is re-flashed with an embedded Linux image, the original NAS functionality is replaced and the device can become an embedded controller.

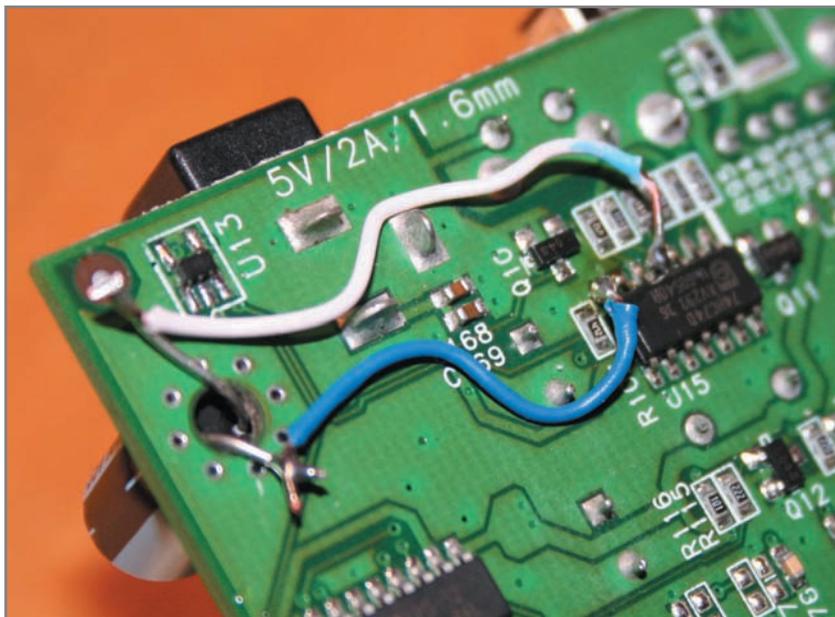


Photo 2—I installed a 100- μ F electrolytic capacitor between pin 4 of U15 and ground. The placement was not critical, so I put the capacitor on the top of the board.

The only requirements for this project's mobile phone were that it support AT-style modem commands (for sending and receiving text messages) and that it be able to connect to a host via a USB cable that also doubles as a phone charger. The Motorola V195 worked nicely. The only requirements for the GPS device were that it connect to and receive its power from the host via a USB cable. The Deluo USB-GPS device I used is based on a Motorola SiRFstarIII chipset.

Assembling this project is simple. Plug the phone's USB cable and the USB GPS device into the two USB ports of the NSLU2. After that, you're almost done. There's only one assembly-related catch. Out of the box, the NSLU2 device won't automatically power-up when power is supplied. For unattended operation, this is an absolute requirement. Luckily, the NSLU2 can be easily reworked to automatically power-up by soldering in a 100- μ F electrolytic capacitor between pin 4 of U15 (i.e., R94) and ground. Since placement is not critical, it can be installed in a convenient hole at the corner of the board. This enables it to be installed on the top of the board where there is enough case clearance (see [Photo 2](#)). If you want to make the unit "stealthy" and not beep when booting up, you can try to cut off the buzzer.

EMBEDDED LINUX & SlugOS

The secret and the magic to using Linux and the NSLU2 device as a controller comes from an open-source project: www.nslu2-linux.org. The NSLU2-Linux development group and user community "discuss, develop, and modify the firmware and hardware of the NSLU2." In addition, with a complete binary (SlugOSBE-4.8-beta-firmware.zip) from the website and re-flashing the NSLU2 firmware with "SlugOS/BE," the road to an embedded controller is very straightforward.

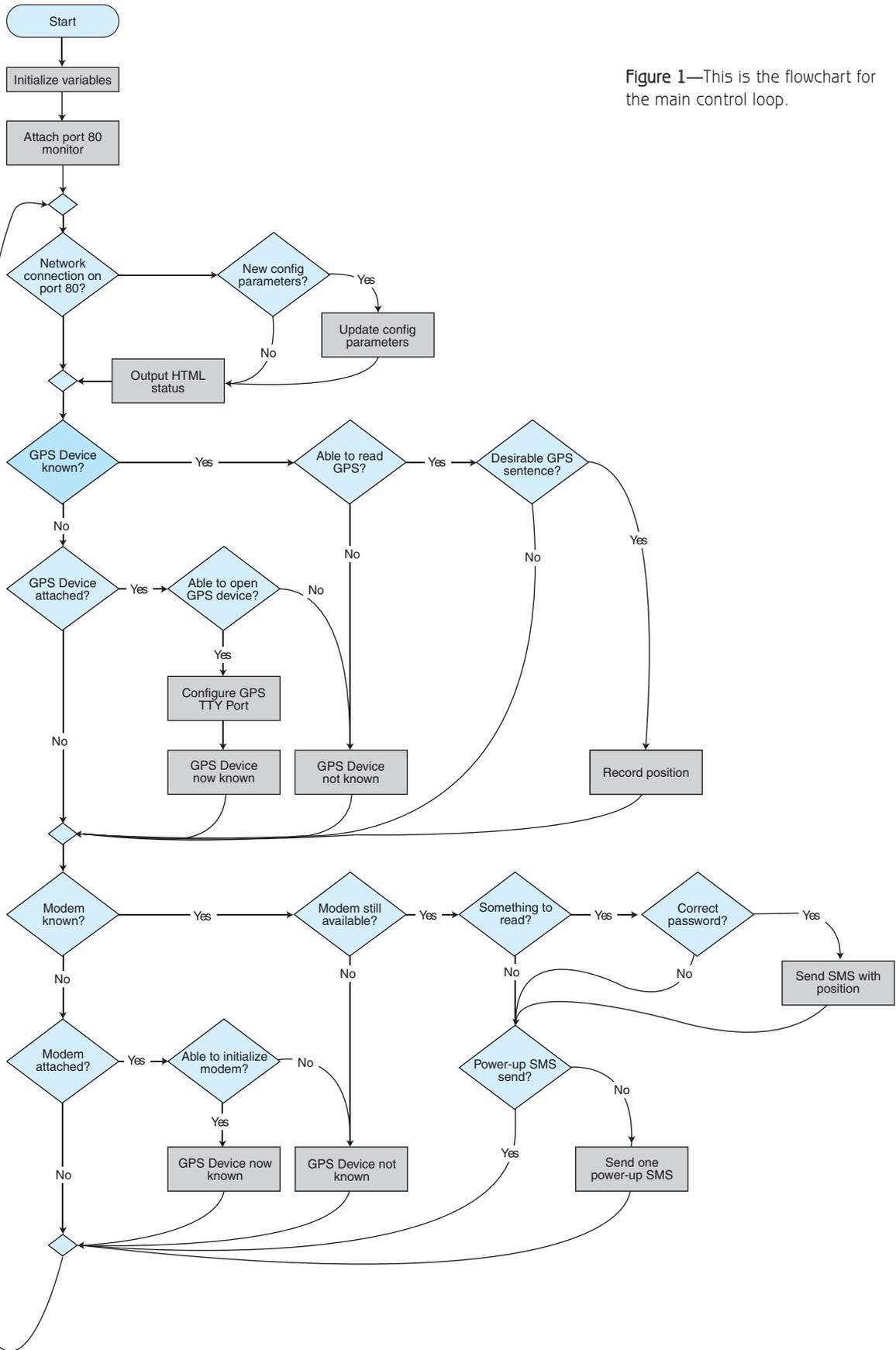


Figure 1—This is the flowchart for the main control loop.

After pulling the NSLU2 out of the box, the network parameters and administrative passwords need to be configured. Then you must put the NSLU2 into Upgrade mode by holding in the Reset button (requires a paper clip) for approximately 10 s and pressing/releasing the Power button once. In Upgrade mode, either "Sercomm's Win32 tool" (Windows) or "UpSlug2" (Linux) can be used to transfer the SlugOS image and re-flash the firmware on the NSLU2.

Although designed to work with an external disk, the pre-compiled SlugOS image is a minimal Linux configuration that can run entirely from internal flash memory without a disk or network share. The minimal configuration takes up about 75% of the 8-MB flash memory, leaving sufficient room to add an available TCL script interpreter (package "tcl"), a USB driver (package "kernel-module-usbserial"), a USB-to-serial port driver (package "kernel-module-pl2303"), and a communications modem driver (package "kernel-module-cdc-acm").

DEVELOPMENT ENVIRONMENT

Having a complete Linux desktop environment makes the NSLU2 configuration and programming task much easier. It allows easy access to main pages, editors, development environments, windowing systems, and a host of other tools. Instead of building up a machine to host the Ubuntu Linux distribution, I used Sun's xVM VirtualBox and ran Ubuntu as a virtual machine on top of Vista. (Make sure you have 3 or 4 GB of memory.) This provides the best of both worlds, having two computers operating with the same mouse and keyboard. For network connectivity, the virtual machine should be configured to use the local subnet by putting in the host network controller into Bridge mode. If the virtual machine is using NAT instead of being bridged, access to the virtual machine from the NSLU2 is difficult.

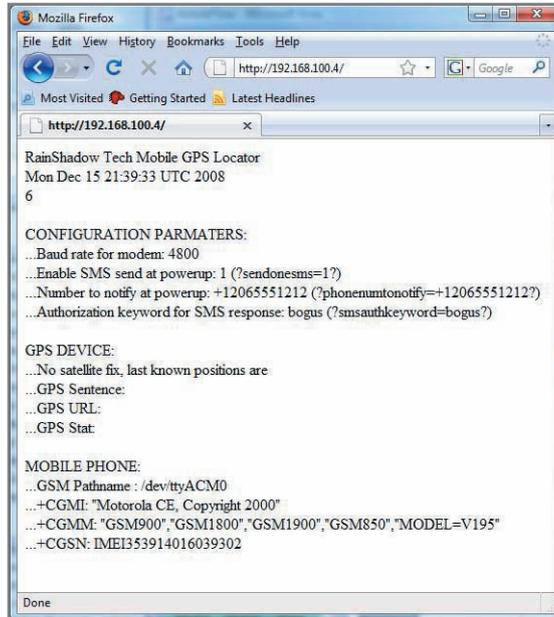


Photo 3—This is the web interface for status and configuration. I used Mozilla Firefox, but you can use the browser of your choice.

The easiest way to configure and operate the NSLU2 reconfigured with SlugOS is to use secure-shell ("ssh"). The SlugOS will allow at

least four concurrent log-ons enabling the one-device editing of multiple script files and a shell from which to execute the scripts. To simplify editing, I used the "nedit" mouse-based source editor and edited the files using the Ubuntu virtual machine. The Ubuntu virtual machine had the scripts located in a network share to which the NSLU2 had executable access. After debugging and completing the scripts, I copied them to the NSLU2 using "scp."

TCL IMPLEMENTATION

SlugOS on the NSLU2 is a complete embedded computer with a multitasking operating system, an impressive array of development tools, and significant driver support for hardware. As such, multiple programming languages are available (e.g., C, C++, Perl, and TCL).

I chose to use TCL for several reasons. One, it is an interpreter. Writing,

Touch Panel PC

10.2"

CUPC-P80 **\$699** / Qty.1

Windows XP based Touch Controller

- Touch panel & Touch controller
- AMD Geode LX800 (500MHz)
- 512M DDR RAM
- 4G FLASH HDD
- USB, ETHERNET, RS232, RS485 port
- SD card support
- Windows XP embedded, Windows 2000 / XP, Linux support

7"

CUWIN3100

- ▶ ARM9 32bit 266MHz
- ▶ WinCE 5.0 Core
- ▶ RTC
- ▶ SD CARD support
- ▶ RS232 x 2 ports
- ▶ USB port
- ▶ Speaker & Stereo Jack

From **\$299** / Qty.1

COMFILE TECHNOLOGY Toll-Free: 1.888.928.2562
<http://www.cubloc.com>

testing, and debugging is quick. Highly optimized and efficient object code is not necessary. Two, it greatly simplifies string processing with access to high-level constructs such as regular expressions. Three, there is easy access to shell commands. Four, it has constructs making it trivial to listen for TCP/IP connection and communicate with the network. The fifth reason is that I've found the TCL syntax easier to understand than Perl (especially for new programmers trying to under-

stand code written by expert Perl coders).

The one issue I stumbled across with TCL was the legacy convention of interpreting numbers with leading zeros as base-8 numbers. Because the code pulls parameters out of fixed-length strings, there is a function to strip leading zeros so as to not misinterpret the integer's value.

I completed this project using less than 400 lines of TCL code. This included simulating a web server to provide network-based status and

remote configuration.

CONTROL LOOP & CODE

TCL is not necessarily rich in data structures, but that is not necessary in this small application. I wrote this project old-school style, with a heavy dependence on a handful of global variables and only a single thread. As such, it is not difficult to understand, debug, or enhance.

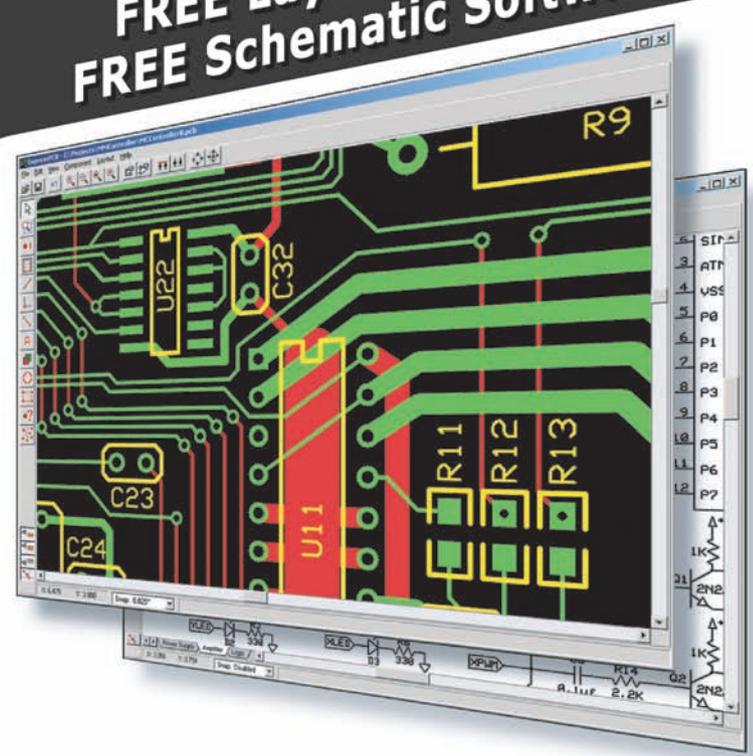
The control loop comprises three top-level functions: monitoring connections to the mini web server, monitoring the GPS device, and monitoring the mobile phone modem (see Figure 1). Monitoring port 80 of the network connection to simulate a mini web server takes only a couple lines of code. `socket -server websocketproc 80` sets up the callback routine `websocketproc` to handle any port 80 connections. Inside the main control loop, `update` relinquishes the processor to address any outstanding network connections and to call `websocketproc` when appropriate. The function `websocketproc` (contained in the source code posted on the *Circuit Cellar* FTP site) reads any data from the connection, looks for parameters to update, and then dumps HTML status information back to the port. With this technique, the device can be configured and queried for status information from any web browser (see Photo 3). To query the device status, simply use its IP address as the URL (e.g., `http://192.168.100.4`). To update a parameter, put the parameter name and new value between question marks (e.g., `http://192.168.100.4/?sendesms=1?`).

Monitoring the connection and disconnection along with the GPS device's datastream is also straightforward. This is a fixed-purpose embedded device and not a general-purpose computer, so the simplifying assumption was made that any USB device detected by the system (`/dev/ttyUSB?`) must be the GPS. Once detected, the device serial connection is configured to interact nicely with the expected data rate and buffering options.

The process of monitoring the connection and disconnection, along with incoming SMS messages, from

\$51^{For 3} PCBs

FREE Layout Software! FREE Schematic Software!



01 DOWNLOAD our free CAD software

02 DESIGN your two or four layer PC board

03 SEND us your design with just a click

04 RECEIVE top quality boards in just days

expresspcb.com

the mobile phone modem uses essentially the same techniques as reading in the GPS datastream. Any devices detected by the operating system (/dev/ttyACM?) are assumed to be phone modems. If an unexpected read error occurs when accessing the device, the device is assumed to be disconnected. The next time through, the top-level loop will try to detect and reinitialize.

SENTENCE PARSING

The Deluo GPS device conveniently adheres to the National Marine Electronics Association (NMEA) standard of position reports. Typing "nmea gps format" in a search engine produces a large quantity of reference material. All the characters in the standard are printable ASCII text (plus carriage return and line feed). The characters are sent at 4,800 bps and comprise sentences whose fields are delimited with commas. One of the most interesting associated with this project is the sentence that starts with: \$GPGGA (global-positioning system fix data). The fields of a \$GPGGA sentence provide the UTC time of the fix, the latitude, the longitude, the altitude, and the number of satellites being tracked. In one statement, the TCL code uses a regular expression to match to the \$GPGGA sentence prefix characters and extract the above data. With this accomplished, the data is massaged slightly and put into web URL form that works with Google Maps. On an Internet-connected phone (e.g., an iPhone or gPhone), it is easy to click through to the Google Maps from the received SMS.

SEND & RECEIVE COMMANDS

The modem TTY port is generally configured in a non-blocking Read mode. This enables the control loop to poll the mobile phone asking for any new data. If data is not available, it continues on to its other tasks.

When the controller wants to send "AT" messages to the phone, it temporarily puts the port into Blocking mode. In this way, the control loop stalls, waiting for the mobile phone to reply with its standard "OK" or "ERROR" response. Once the expected response is received, the

port is put back into non-blocking mode and the control loop proceeds to loop normally.

If the control loop detects an incoming message from the mobile phone, the incoming string is tested to see if it is a "+CMTI" message indicating an incoming SMS. If it is, the SMS message number is extracted from the sentence and used to query the actual SMS text by using the AT+CMGR command. After the message is read, it is immediately

deleted with the AT+CMGD command.

The previously read message is then searched for the password and originating phone number. If the password is found, the last known position (encoded in a web URL) is used as the SMS payload for the return message to the originating phone number.

LINUX DAEMON

In the same way, the device needs to automatically power-up when the

The brighter side of electronics

Noritake

The Next Generation of Character Display

All New Y Series Vacuum Fluorescent Display



Highlight Characters

Wide Temperature Range (-40 °C - +80 °C)

Display More Characters

Selectable Font Sizes

International Font Table

Serial Interface

1 (800) 779 - 5846
www.noritake-elec.com

itron®

engine is switched on. The TCL script must automatically start to execute when the Linux device finishes booting. A Linux daemon accomplishes this. I created an executable shell script that launches and forks the primary TCL script (`/bin/sh /root/main.tcl &`) and placed it in the `/etc/init.d` directory. With that complete, I created a file system link to it in the following manner:

```
In -s /init.d/executableshellscript
    /etc/rd3.d/S99executableshellscript
```

DESIGN REVIEW

The programmed vehicle locator design is working reliably in one of my vehicles. However, this off-the-shelf hardware solution has two primary shortcomings. The first is that if the vehicle is not operated frequently enough, the mobile phone battery completely discharges (because the phone doesn't turn off). Eventually, the phone shuts off and doesn't restart. Luckily, the Motorola

V195 has an impressive battery life with a quick recharge cycle. As a result, I haven't dealt with this failure mode because I drive the vehicle nearly every day. And since I expect an SMS with every vehicle trip, I can detect a dead battery right away.

The second shortcoming is a temporary failure mode. The capacitor solution to automatically power-up the Linux device may not work if the power is not removed long enough. Specifically, if the car ignition is set to "accessory" and then you turn the key to start the car, the transient power loss may turn off the Linux device without discharging the capacitor and therefore not restart the device. However, the next time the car is left off for 5 minutes, it will proceed to boot normally. Note that there are other power-on hacks available to get around this shortcoming. They completely disable the Off button should you use this controller for other purposes.

I'm considering a few new features for the design. I want it to accommodate

time zones (instead of simply reporting UTC time). I also want the design to temporarily shut off the power-up SMS capability at certain times of the day. ☒

James Stephanick (BSEE, MSEE, MBA) began designing and programming real-time embedded image-processing systems to automatically analyze cytology specimens. Later, his career path took to him to the mobile phone industry where he led the Tegic engineering team. James is now leading the engineering team at Cequent, a Seattle-based start-up company that provides enhanced caller ID solutions for North American mobile phones. You may contact him at james@rainshadowtech.com.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2009/232.

RESOURCES

Developers' Home, "Short Message Service/SMS Tutorial," www.developershome.com/sms/.

GPS Info, <http://gpsinformation.net/>.

NSLU2-Linux development group and user community, <http://www.nslu2-linux.org/>.

TCL Developer Xchange, www.tcl.tk.

SOURCES

USB-GPS Device
Deluo LLC | www.deluogps.com

Ubuntu 8.10 Desktop Edition
Canonical Ltd. | www.ubuntu.com

Linksys NSLU2 USB Network storage link
Cisco Systems | www.linksysbycisco.com

VirtualBox Open-source virtualization software
Sun Microsystems | www.sun.com

keil.com
1-800-348-8051

KEIL™
Tools by ARM

Development Solutions for ARM, 8051 & XE166 Microcontrollers

Microcontroller Development Kits	RTOS and Middleware Components
C and C++ Compilers	RTX Kernel Source Code
Royalty-Free RTX Kernel	TCPnet Networking Suite
μVision Device Database & IDE	Flash File System
μVision Debugger	USB Device Interface
Complete Device Simulation	CAN Interface

Keil PK51, PK166, & MDK-ARM support more than 1,700 microcontrollers

Keil RL-ARM and ARTX-166 highly optimised, royalty-free middleware suites

Download the μVision4 Beta Version keil.com/uv4

The Newest Semiconductors



New Products from:



Green Mode PWM Controller:
UCC28600

[www.mouser.com/
tipowermanagement/a](http://www.mouser.com/tipowermanagement/a)



STM6904 / STM6905 Multiple
Voltage Supervisor

www.mouser.com/st/a



Analog Power Supply /
Power Management Devices

www.mouser.com/freescale/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 390 leading suppliers.



a tti company

The Newest Products
For Your Newest Designs

www.mouser.com

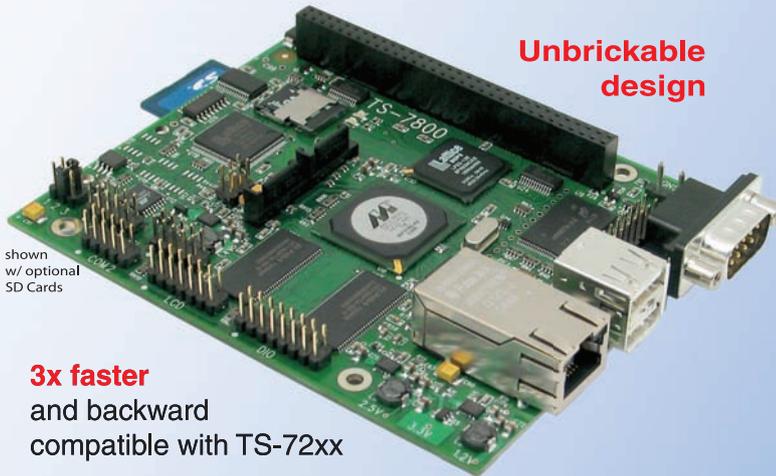
Over A Million Products Online

(800) 346-6873



Embedded Single Board Computers

High-End Performance
with Embedded Ruggedness



shown
w/ optional
SD Cards

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 customizable 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 2.6 in .67 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 Family 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
- Linux, Real Time extension, Debian
- SD card option
- VGA video
- LCD ready

- 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

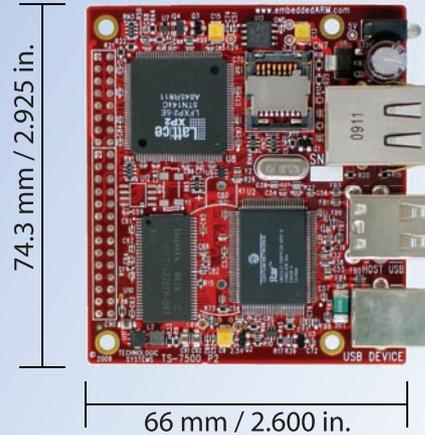
New Products

TS-7500 250 MHz ARM9

- Low power, fanless, < 2 watts
- 64MB DDR-RAM
- 4MB NOR Flash
- Micro-SD Card slot - SDHC
- USB 2.0 480Mbit/s host (2) slave (1)
- 10/100 Ethernet
- Boots Linux 2.6 in < 3 seconds
- Customizable FPGA - 5K LUT
- Power-over-Ethernet ready
- Optional battery backed RTC
- Watchdog Timer
- 8 TTL UART
- 33 DIO, SPI, I²C

NEW!

Our Smallest Computer
at Our Best Price Point



\$84
qty 100

\$99
qty 10

TS-8100 Ultra Reliable w/ 128MB ECC RAM

- POE ready
- Dual-execution unit, double-precision FPU
- Multifunctional PC/104 connector
- 12K LUT customizable FPGA
- 512MB NAND Flash
- 1 USB Host, 1 USB Device (12 Mb/s)
- Boots Linux 2.6 in < 2 seconds
- Fanless < 4W, sleep mode < 1mW
- Regulated 5-28V power input
- 2 10/100 ethernet
- 2 SDHC sockets
- 4 COM ports
- 5 10 bit ADC
- SPI & DIO
- RTC & WatchDog
- RS485/RS422
- 2 DMX Channels

NEW!

400 MHz PowerPC
with Floating Point Unit



\$229
qty 100

\$269
qty 1

shown w/ optional SD Card



We use our stuff.

Visit our TS-7800 powered website at
www.embeddedARM.com

Energize a Circuit

An Innovative Switched-Mode Power Supply Design

Energy control is an essential aspect of many electronic applications. If you build a switched-mode power supply (SMPS), you can regulate a design's current to meet your specific project goals.

Most engineers, technicians, and designers need power supplies for energizing their circuits. Do you? You can use a power supply instead of batteries. If portability is not a requirement, a power supply is a viable alternative to batteries because it costs less to use. It's nice if you can adjust the voltage, but it is even better if you can also modify the current. The switched-mode power supply (SMPS) I recently designed enables you to adjust the voltage from 0 to 20 V and the current from 0 to 1 A (see [Photo 1](#)). You can change the voltage and current by entering the value on the keypad and checking the output on the 2 × 16 LCD module.

As you can see in [Figure 1](#), the SMPS is built around a Maxim Integrated Products DS89C430 microcontroller. [Figure 2](#) shows the digital portion of the design. [Figure 3](#) depicts the analog portion. In this article, I'll describe the project from start to finish.

FLYBACK REGULATOR

A flyback regulator works by storing energy in the core material. The energy is stored while the power switch conducts. The power transformer of a flyback converter is not really a transformer but an inductor.

After the power switch stops conducting, the power transformer flies back or reverses in polarity. That is why the phasing of the power transformer of a flyback transformer has the opposite polarity when compared to a "forward" converter. The energy stored in the core material is transferred to the secondary winding by induction. The filter capacitor smoothes the regulator's output. The topology used in this design is a diagonal

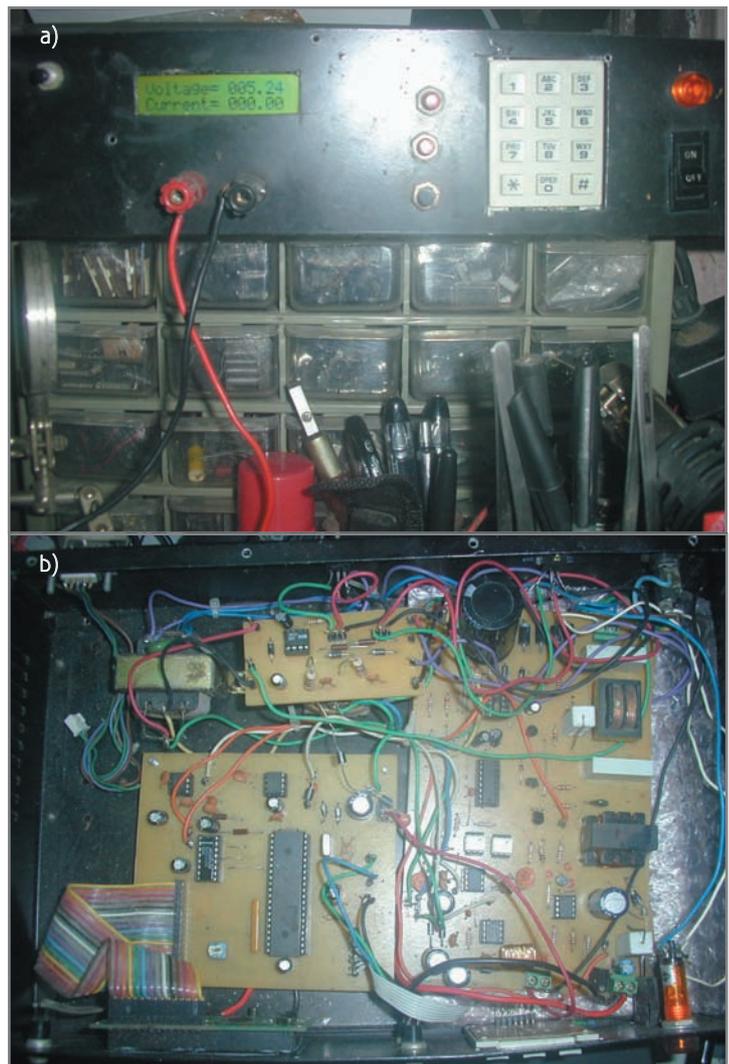


Photo 1a—As you can see, the SMPS's front panel features an LCD with backlight. **b**—Take a look inside the DS89C430-controlled SMPS. The two 60-Hz transformers are for the auxiliary supply of the analog part. The other 60-Hz transformer powers the digital part. It's clear why 60-Hz transformers are needed.

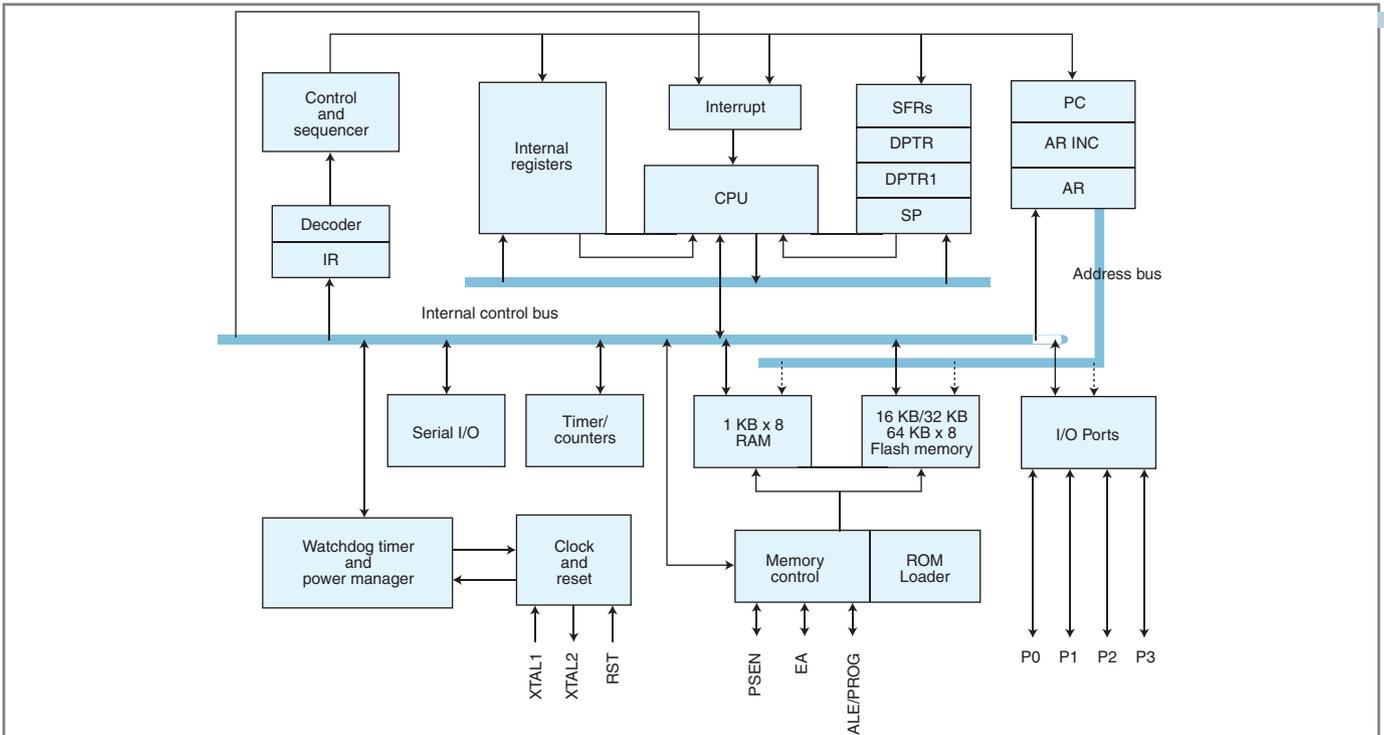


Figure 1—I used a Maxim Integrated Products D589C430 microcontroller for this SMPS project. I chose it for its speed and because it is in-system programmable.

flyback for a variable SMPS. It uses two power MOSFET switches and two fast-recovery diodes. The MOSFET, fast-recovery diodes, and the flyback transformer's primary is connected in such a way that when no energy transfer is made to the secondary, the energy stored in the primary winding of the flyback transformer is returned to the

source of the DC Bus voltage.

CIRCUITRY

The heart of the circuit is a DS89C430 microcontroller (U1), which is an 8051-compatible microcontroller. It has three timers, two serial ports, and 1-KB MOVX RAM. I

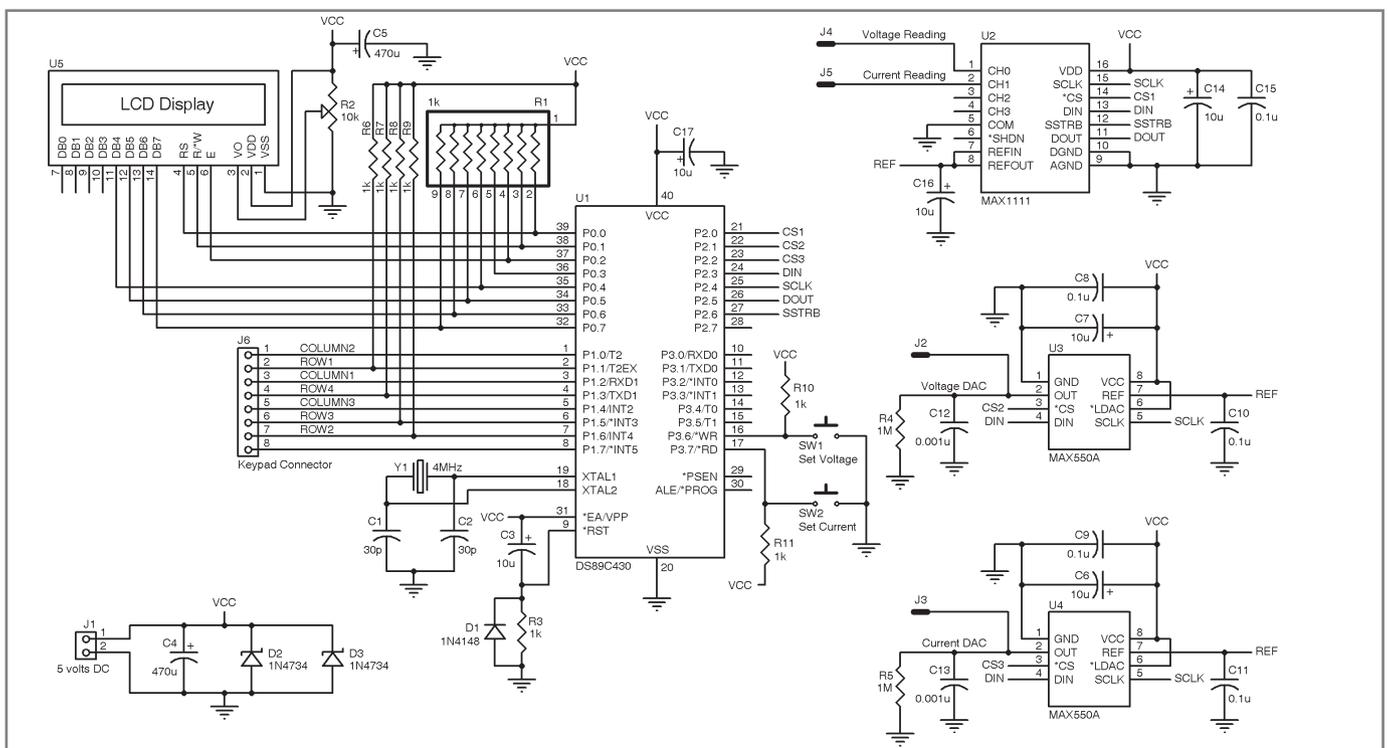


Figure 2—This is the digital portion of the SMPS. It dictates the voltage that the analog portion outputs.

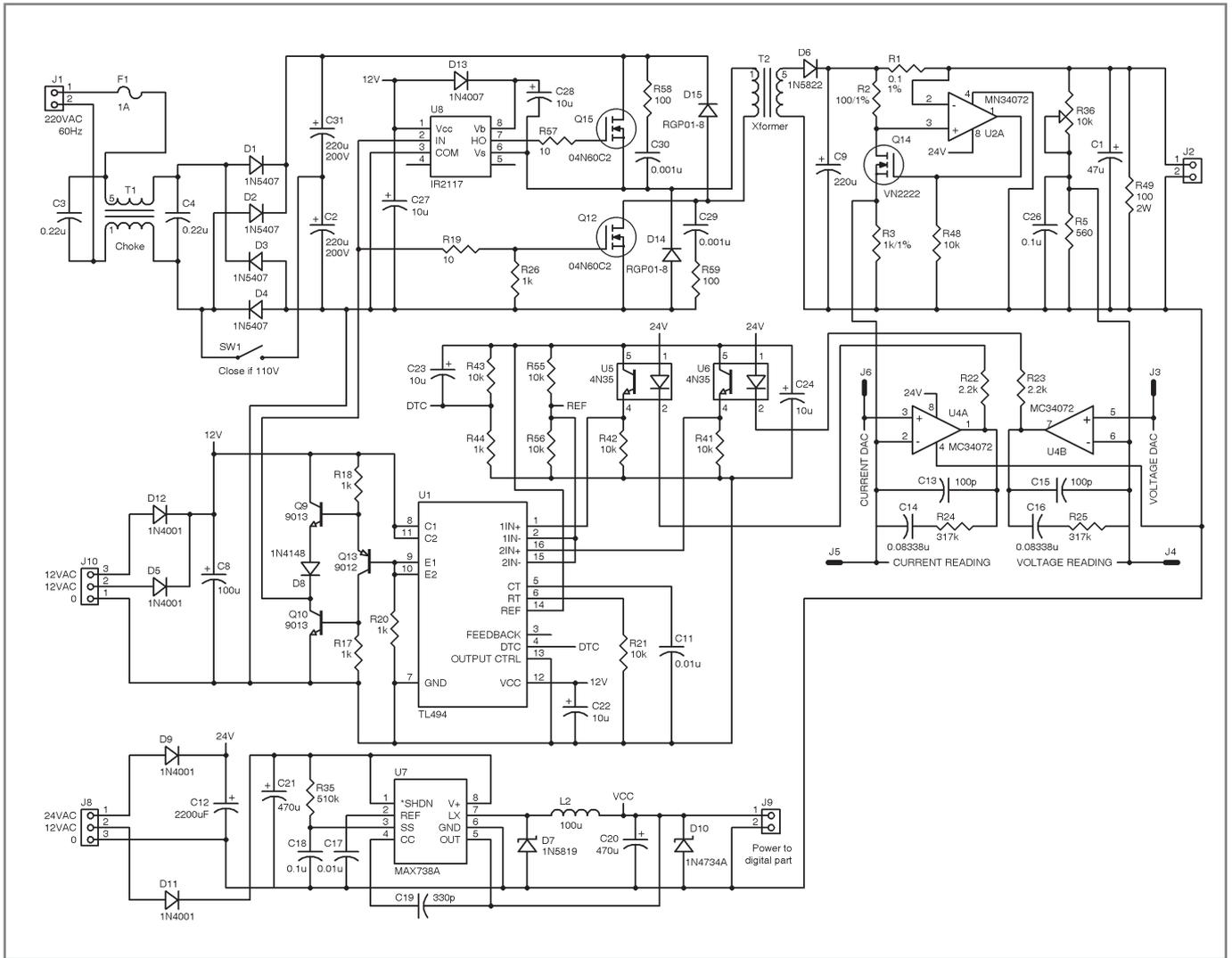


Figure 3—This is the analog portion of the SMPS.

chose this microcontroller for its speed and because it is in-system programmable. You don't need a device programmer for programming the DS89C430. All you need is to configure the microcontroller's pins and connect a MAX232 so you can connect it to a PC's serial port (see [Photo 2](#)). You can use HyperTerminal or any terminal software. After you have applied the power, you must press the Enter key so that the microcontroller can detect the data rate. However, I discovered that the maximum data rate is 19,200 bps. The DS89C430 controls both the DACs and the ADC. It also reads the keypad and processes the numbers for the proper voltage.

The MAX1111 (U2) is a serial ADC. It has four inputs that can be configured as single-ended or differential. It

is a SPI peripheral. The DS89C430 has no SPI port, so it is done by bit banging. The MAX550 (U3) is a single serial DAC. It is an 8-bit device. Its output is 0 to 2 V. The circuit uses two DACs, one for current setting and the

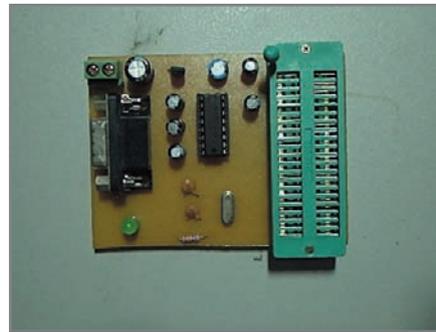


Photo 2—I used a Dallas Programmer to program the DS89C430. It is merely a socket with a MAX232. A device programmer isn't needed.

other for voltage setting. C3 (10 μ F), R3 (10 k Ω), and D1 (1N4148) form a reset circuit for the microcontroller. As the capacitor charges, a voltage is generated across the resistor and the voltage tapers off as the capacitor is fully charged. The diode discharges the voltage across the capacitor once the supply is turned off so the microcontroller will reset again if the supply is turned off and on in an instant.

C1 (30 pF) and C2 (30 pF) act as load capacitors. They are needed by the internal oscillator together with the crystal (Y1) to act as the clock for the microcontroller. C12 (0.001 μ F) and C13 (0.001 μ F) act as a filter for the DAC. C14 (10 μ F) and C15 act as supply decoupling, while C10 (0.1 μ F) and C11 (0.1 μ F) act as a filter capacitor for the entire circuit.

The Next Generation of In-Circuit Debugging

Microcontrollers

Digital Signal
Controllers

Analog

Serial
EEPROMs



The NEW MPLAB® ICD 3

The MPLAB ICD 3 In-Circuit Debugger is Microchip's most cost effective high-speed debugger for Microchip Flash PIC® Microcontrollers (MCU) and dsPIC® Digital Signal Controller devices. It debugs and programs PIC MCUs and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

- In-Circuit Debugging for PIC MCUs and dsPIC DSCs
- Full-speed, real-time emulation
- Source debugging, stopwatch, complex breakpoints and in-circuit programming
- MPLAB IDE compatible
- Firmware upgrade via MPLAB IDE
- Overvoltage and undervoltage protection
- High Speed USB 2.0 (480 Mbps)
- Target power, up to 100 MA
- Internal 1 MB memory buffer for increased download speed

MPLAB® ICD 2 RECYCLE

Return your old MPLAB ICD 2 and receive 25% off the new MPLAB ICD 3, MPLAB REAL ICE or PICKit™ 3 Debug Express. For more information on this offer, please visit:

www.microchip.com/ICD2recycle



www.microchip.com/ICD3



MP3P DIY KIT, Do it yourself

(Include Firmware Full source Code, Schematic)

• myPIC



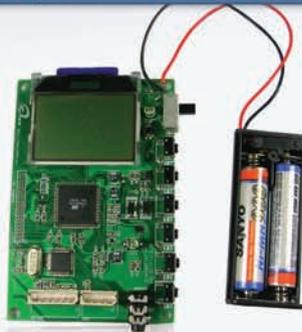
Only

\$160 **\$220**

qty 100

qty 1

• myWave (MP3 DIY KIT SD card Interface)



Only

\$150

qty 100

\$200

qty 1

• myAudio (MP3 DIY KIT IDE)

Only

\$180

qty 100

\$220

qty 1



Powerful feature

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

Specification

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

Powerful feature

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

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

Powerful feature

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

66

A TL494 (U1) voltage-mode PWM controller is the heart of this Flyback mode power supply. Inside this IC is an oscillator with a sawtooth waveform output, two error amplifiers, a comparator, commutation logic, and a reference. The PWM principle is straightforward.

99

Diodes D2 and D3 (1N4734) act as protection components and conduct when the supply exceeds 5.0 V. It is better for the Zener diode to get damaged in the case of reverse polarity or overvoltage instead of the microcontroller, DACs, or ADC. The LCD module (U5) is an HD44780-based display, which displays the voltage and current.

Capacitor C3, choke T1, and C4 form an EMI filter so noise will not propagate to the mains. Because the MOSFET switches rapidly and the bulk capacitor C2 charges quickly, it generates some noise that could affect other equipment. The choke has no value since I took it from an old power supply. D1, D2, D3, and D4 (1N54007) act as a rectifier for the mains voltage. It converts the AC voltage to DC voltage. C2 acts not only as a filter, but also as storage for the energy to be delivered by the supply.

Resistor R19 acts as a series resistance to the gate of the MOSFET Q12. The resistor prevents the MOSFET from oscillating because a voltage is impressed across the gate and source. Resistor R26 discharges the voltage across the gate of the MOSFET and source of the MOSFET. This is because the gate and the source of the MOSFET form a capacitor that holds a charge. If there is a voltage across the gate and source, this will leave the MOSFET in the "on" condition. It will be damaged if powered up, and the Texas Instruments TL494 PWM controller will also be damaged

if the gate drive circuit is defective. MOSFET Q12 and MOSFET Q15 (04N60C2) act as a power switch for the supply and conduct at the same time. It changes the steady DC supply to a square wave with a varying duty cycle. To increase the voltage output, the pulse width is lengthened. If a smaller voltage is needed, the pulse width is made shorter. Resistors R58 and R59 (100 Ω) and capacitors C29 and C30 (0.001 μF) act as a parallel snubber. It protects the MOSFET 04N60C2 by absorbing the energy as the MOSFET ceases to conduct. Two recovery diodes (D14 and D15) connected to the primary winding return the energy to the DC bus if there is no energy transfer between the primary and secondary. IC U8 (IR2117) is a high-side gate drive IC, which is the gate drive for the high-side MOSFET (see [Photo 3](#)).

Transformer T2 is a power transformer with a ferrite core. A power ferrite is needed because of the frequency the transformer operates. If you use an iron core, it heats up because of eddy currents. This is why a 60-Hz transformer has a laminated core because of the losses. By using a laminated core, the circuit path is limited to a single laminate. The power ferrite transformer converts the high-voltage DC to a smaller value by making the windings have a turns ratio according to the voltage you will need. By the way, in Flyback mode converters, the power transformer is not a transformer at all but only an inductor. As the



Pololu

Robotics & Electronics

Robot Kits
Line followers
Robot arms
Chassis

Save 10%
With coupon code
CCLLR3PI6

3pi Robot
\$99.95

High-performance, C-programmable, ATmega328P-based robot (with Arduino support!)
Items #975, #1306

Mechanical Components
Motors, servos
Wheels, ball casters



Motion Control
Motor controllers
Servo controllers

Now with USB!



Robot Controllers with Arduino support!

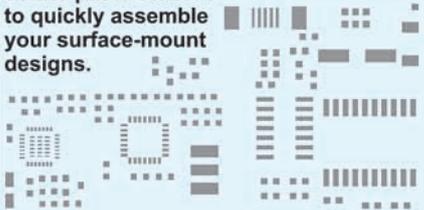


voltage regulator dual H-bridge
power LED (green) trimmer pot
red user LED programming connector
ATmega48/328P microcontroller 20 MHz clock

Solder Paste Stencils

Use our low-cost solder paste stencils to quickly assemble your surface-mount designs.

From \$25



Custom Laser Cutting

From \$25



Cut your own custom chassis, front panels, and more!

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

3095 E. Patrick Ln. #12, Las Vegas, NV 89120



Photo 3—This is the close-up view of the MOSFET's gate drive at 5 V with a 0.5-A load. The load is light, so the MOSFET's gate drive is not regular. The gate drive is just enough to keep the voltage at 5 V and the current at 500 mA.

MOSFET conducts, it stores energy in the inductor. As the MOSFET ceases to conduct, the energy is released to the secondary winding. Also note that the power transformer or inductor has windings that are out of phase. True transformers have their windings in phase and do not store any energy, unlike the power transformer of a flyback regulator. A

Schottky diode (D6) converts the square wave AC to a pulsating DC. A Schottky rectifier is needed because of its fast recovery time. As you know, an ordinary diode does not stop the flow of current in an instant as the voltage reverses in polarity.

A TL494 (U1) voltage-mode PWM controller is the heart of this Flyback mode power supply. Inside this IC is an oscillator with a sawtooth waveform output, two error amplifiers, a comparator, commutation logic, and a reference. The PWM principle is straightforward.

First, the sample voltage from the output of the power supply is fed to the noninverting input of the error amplifier and the reference of the supply is fed to the inverting input of the supply. After the two voltages are compared, an error or control voltage is generated at the output of the error amplifier. This error voltage is compared to a sawtooth voltage by a

comparator. The larger the error voltage, the wider the comparator's pulse. The comparator's output is connected to the commutation logic and then to the output transistors. In my application, I used a single-ended output by grounding the output control pin of the IC. R21 (10 kΩ) and C11 (0.001 μF) are connected to the oscillator of the PWM IC, which dictates the switching frequency of the power supply. The isolated Flyback mode power supply's switching frequency is 100 kHz.

R18 (1 kΩ), Q13 (2N3906), R17 (1 kΩ), Q9 (2N3904), Q10 (2N3904), and D1 (1N4148) form the power supply's gate drive circuit. A totem pole was used because the MOSFET must be switched on and off rapidly. If the switching of the MOSFET is slow, this will result to switching losses that will heat the MOSFET.

R20 (1 kΩ) switches on Q13 (2N3906). C23 (10 μF), R43 (10 kΩ), and R44 (1 kΩ) act as a soft start circuit. It controls the duty cycle of the PWM IC after it is powered up. The

Easy Embedded Linux

OMNI-EP



- 200 Mhz ARM9 CPU
- 10/100 Mb Ethernet
- 32 MB RAM
- 16 MB Flash
- 16 Digital I/O Lines
- 2 Ports of USB 2.0
- SPI Bus
- AC97 Amplified Audio
- Battery Backed Clock
- 2 Serial Ports
- Low Power Consumption
- RoHS Compliant

\$249 Qty 1

Our newest ARM9 Linux controller, the OmniEP doesn't cost an arm and a leg. It delivers removable storage, amplified audio, ethernet and serial RS232 communication ports in a rugged and attractive enclosure. Models without enclosure and LCD available.

The OmniEP comes preloaded with Linux to jumpstart your development process, with LCD and pushbutton drivers supplied. Large capacity USB drives can be easily mounted in the USB port.

Call **530-297-6073** Email sales@jkmicro.com
www.jkmicro.com

JK microsystems

FlashPro430 FlashPro-CC FlashPro2000 GangPro430 GangPro-CC



USB Flash Programmers for Texas Instruments' MCUs
MSP430, Chipcon CCxx, C2000 DSPs

**Reliable and the fastest programmer on the market.
 Perfect for production usage.**

- * can assign unique serial number
- * up to eight programmers can be connected to one PC and program target devices simultaneously



Elprotronic
 Incorporated

www.elprotronic.com

duty cycle slowly ramps up after the capacitor charges. R55 (10 kΩ) and R56 (10 kΩ) scale the voltage V_{REF} to 2.5 V, which is fed to the error amplifier. R39 (510 Ω), R40 (510 Ω), R37 (510 kΩ), and R38 (510 kΩ) gives gain to the error amplifier and also makes the error amplifier linear.

If there are no resistors, the error amplifier will only act like a comparator, and it will have only two values—0 or the maximum output of the error amplifier, which is 5 V or V_{REF} . R42 (10 kΩ) and R41 (10 kΩ) pull down the error amplifier's input so the error amplifier will function properly. Optocouplers U5 (4N35) and U6 (4N35) isolate the output from the input. The LED inside the optocoupler lights up if the power supply's sampled output is greater than the reference—in our case, it is the DAC.

R22 (2.2 kΩ) and R23 (2.2 kΩ) limit the current flowing through the optocouplers. A MC34072 (U4) high-performance op-amp acts as an error amplifier on the secondary side. C13 (100 pF), C14 (0.0833 μF), and R24 (317 kΩ) act as a compensation circuit for the op-amp. Compensation is needed because the power supply might oscillate at varying load conditions. The compensation circuit varies the op-amp's gain and decreases at -20 db as the frequency increases. Pole-zero compensation is used. It is applicable for discontinuous Flyback mode regulators, and it also alters the phase of the op-amp's output so that it doesn't return to 360°. Remember that negative feedback provides a 180° phase shift.

Resistors R1 (0.1 Ω), R2 (100 Ω), R3 (1 kΩ), R48 (10 kΩ), and U2 (MC34072) op-amp act as a current detector. Resistor R1 (0.1 Ω) acts as the resistive sensor for the current detector. As current passes through the resistor, a voltage drop is generated across the op-amp's inputs. The higher the current, the greater the difference at the op-amp's inputs. This makes the op-amp's output increase to make the MOSFET (Q14) conduct, which pulls down resistor R2 (100 Ω) until the voltage at the op-amp's noninverting and inverting pins are equal. Resistors R36 (10 kΩ) and R5 (560 Ω) act as a voltage divider to sample the regulator's output.

Frequency	50 kHz
B_{MAX}	0.3 tesla
A_E	0.7 cm ² (core area)
A_W	1.3 cm ² (core window area)
$N_{PRIMARY}$	268 turns
$N_{SECONDARY}$	27 turns
AWG Primary	30
AWG Secondary	26

Table 1—These are the parameters of the power transformer. The operating frequency of the power ferrite core should be equal to or greater than 50 kHz.

C1 (47 μF) acts as a filter capacitor. R49 (100 Ω) acts as a minimum load necessary for the stability of the supply. D9 and D11 (1N4001) act as a rectifier that converts AC to DC. C12 (220 μF) and C21 (470 μF) act as a filter capacitors. Resistors R35 (510 kΩ) and C18 (0.1 μF) act as a soft start for the MAX738 (U7). C17 (0.01 μF) acts as a filter for the MAX738's reference. C19 (330 pF) acts as a compensation for the MAX738. The D7 (1N5819) Schottky diode acts as a commutating diode that

conducts as the switch in the MAX738 opens up. As the 1N5829 conducts, the energy in the inductor is delivered to the load.

I used two 60-Hz transformers for the TL494's auxiliary supply. I used another 60-Hz transformer for the supply of the digital part, which uses the MAX738. I did this because I couldn't add separate windings from the power ferrite transformer because the power supply's output drops to zero. As the supply decreases to zero, the voltage on the auxiliary winding also drops to zero. In some power supplies, a series pass preregulator makes the circuit run as the power is applied. However, this is good only if the auxiliary supply takes over as the power supply runs. The series pass preregulator has a low efficiency as a linear regulator, so I decided to use a 60-Hz transformer for the auxiliary power supply. I used another 60-Hz transformer to power the digital part.

CONSTRUCTION & ASSEMBLY

This project involves exposure to high voltage, which could lead to

See It & Solve It
with
USBee Test Pods

- Logic Analyzers
- Oscilloscopes
- Signal Generators
- Protocol Analyzers
 - I2C, SPI, ASYNC, CAN
 - 1-Wire, PS/2, USB
 - I2S, SMBus, Serial
- Configurable
- Programmable
- High Speed USB
- PC-Based
- And start at \$139

USBee
USB based Electrical Engineer

www.USBee.com

electrocution or a fire if you don't use the correct fuse rating and the proper design. If you are a novice, consider switching to a linear power supply using a 60-Hz transformer.

You can use a presensitized PCB for the digital and analog parts. I made my own PCB layout, but I've since made corrections, so I won't recommend it. It is better if you make your own PCB using your own PCB layout software. The most critical part of this switch mode-isolated flyback power supply is the power transformer. Whether or not the power supply functions depends on the transformer. Table 1 includes the specs of the power transformer and ferrite E and I core.

I computed the core area product: $A_p = A_E \times A_W$. But I used a larger core, which enables me to handle more power, and there is a less chance that the ferrite core will saturate. Make sure the power ferrite core is rated for 100 kHz or else the core will heat up due to core losses. Wind the magnet

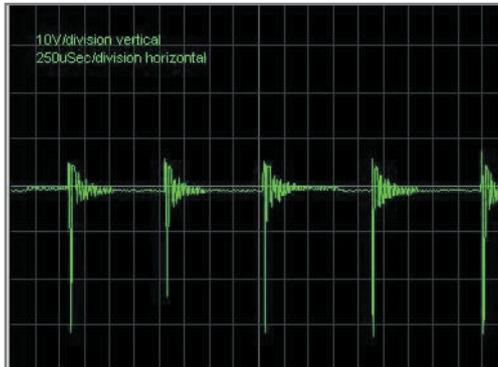


Photo 4—Take a look at the transformer's output. The Schottky diode blocks the negative voltage generated when the MOSFET conducts and when the transformer stores energy. The Schottky diode is forward-biased when the MOSFET becomes an open switch, which you can see in the waveform, and has a potential of 5 V. The filter capacitor smooths the pulsating DC, which becomes a regular DC voltage and is the power supply's output.

wire neatly and tightly so that the coupling of the primary and secondary is good. Wind the primary clockwise and the secondary counter clockwise. It's important that you wind the two in opposite directions; otherwise, your power supply won't work. (It is not

important which is clockwise and counterclockwise. Just be sure you wind in opposite directions.) The direction of the primary winding is not important because one of the primary's terminals is connected to the positive DC bus. The other is connected to the MOSFET's drain, which switches the transformer to ground. After the power switch opens, the terminal of the primary winding connected to the positive DC bus becomes negative, and the one connected to the MOSFET's drain becomes positive.

When the power switch opens on the secondary side, the polarity of the secondary winding's terminal connected to the Schottky diode becomes positive, and the other terminal becomes negative. Thus, the Schottky diode is forward-biased and delivers the energy stored in the core material. The secondary delivers the stored energy in the core material only after the MOSFET becomes an open switch because of the Schottky diode. The Schottky diode blocks the voltage

Easy Embedded Linux

\$169
Qty 1



16MB FLASH / 32MB RAM

200Mhz Arm9 CPU

16 Digital I/O

Watchdog

10/100 Ethernet

Battery backed Clock/Calendar

Audio In/Out
2 USB
2 Serial Ports

*We brought you the world's easiest to use DOS controllers and now we've done it again with Linux. The **OmniFlash** controller comes preloaded with Linux and our development kit includes all the tools you need to get your project up and running fast.*

*Out-of-the-box kernel support for USB mass storage and 802.11b wireless, along with a fully integrated Clock/Calendar puts the **OmniFlash** ahead of the competition.*

Call **530-297-6073** Email sales@jkmicro.com
On the web at www.jkmicro.com

JK microsystems

Low Cost Panel PC

PPC-E7

- Fanless ARM9 200MHz CPU
- 3 Serial Ports & SPI
- Open Frame Design
- 3 USB 2.0 Host Ports
- 10/100 BaseT Ethernet
- SSC-12S Audio Interface
- SD/MMC Flash Card Interface
- Battery Backed Real Time Clock
- Up to 64 MB Flash & 128 MB RAM
- Linux with Eclipse IDE or WinCE 6.0
- JTAG for Debugging with Real-Time Trace
- WVGA (800 x 480) Resolution with 2D Accelerated Video
- Four 12-Bit A/Ds, Two 16-Bit & One 32-Bit Timer/Counters





Setting up a Panel PC can be a puzzling experience. However, the PPC-E7 Compact Panel PC comes ready to run with the Operating System installed on Flash Disk. Apply power and watch either the Linux X Windows or the Windows CE User Interface appear on the vivid color LCD. Interact with the PPC-E7 using the responsive integrated touch-screen. Everything works out of the box, allowing you to concentrate on your application, rather than building and configuring device drivers. Just Write-It and Run-It. Starting at \$495.

For more info visit: www.emacinc.com/panel_pc/ppc_e7.htm

Since 1985
OVER
24
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

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

when the transformer is storing energy while the MOSFET conducts (see [Photo 4](#)).

Use the right gauge or a lower gauge number for the primary and secondary to minimize copper losses. Also, make the PCB trace (of the output of the transformer) to the power supply's terminal block wide enough so that there will be no voltage drop across the output. I bolted the power MOSFET to the supply's chassis so it runs cool even under full load. If you can't make the trace wide enough, apply solder to it to increase its current capacity. While you're testing the power supply, never bypass the fuse or use a breaker with a low current rating. I used a circuit breaker with a 0.5 A rating.

TROUBLESHOOTING

Don't be discouraged if the supply doesn't work the first time after construction. If it doesn't work, consider the following tips.

Test the digital and analog parts separately. Connect a 5-V supply to the digital part. Enter a voltage on the keypad. The output of the voltage DAC should be from 0 to 2 V, and the output of the current DAC is from 0 to 1 V. If the DAC has no output, check if there is a pulsating signal on the SCLK, CS2, CS3, or DIN after the voltage is entered on the keypad. Also, check for a 2.048 V at the reference pin of the DACs. This 2.048 V comes from the ADC (MAX1111). Check if the ADC and DAC have their V_{CC} and GND connected properly. Apply a voltage to the ADC's inputs from 0 to 2 V and check if the display is correct. If not, check for a pulsating signal on the SCLK, DIN, DOUT, CS1, or SSTRB.

Please note that if the LCD module is defective, or if the ADC is defective, it will hang the program because it polls the busy flag on the LCD module (and if the conversion is finished on the ADC by polling SSTRB). You should also check if the REFOUT pin is connected to the ADC's REFIN pin.

For the analog part, don't connect to the mains if you aren't sure of your design. If there is an error in the

construction, such a connection may damage the MOSFET, the 1N5407, or the power ferrite transformer.

First, apply a 12-V DC voltage to J7 and use an oscilloscope to monitor the output of the PWM controller TL494. You should see a square wave output at pins 9 and 10 on the TL494 and at the gate pin on the IRF840. If the TL494's supply drops to a low value, check if the totem pole gate drive circuit is shorted. Also, check for a sawtooth output at pin 5 on the TL494. In addition, check for a 5-V potential at pin 14 on the TL494. If you adjust the voltage from 0 to 2.5 V using a separate power supply at pins 1 and 16 on the TL494, you should see the duty cycle vary at the output of pins 9 and 10.

Check the polarity of bulk capacitor C2 or it will explode. Be sure the primary winding is connected to the MOSFET's drain. The secondary should be connected to D6 (1N5822).

After performing all of the aforementioned checks, look to see if there is 24 V powering U2 (MC34072) and U4 (MC34072) and the optocouplers U5 and U6 4N35 are good, or else this will cause the PWM controller to output a maximum duty cycle with the control loop not functioning properly. Before testing your power

supply, connect a 10-V potential at the output of the supply and adjust R38 (10-k Ω multiple-turn trimmer) so that the wiper arm's output or voltage reading pin is 10% of the output.

USING THE POWER SUPPLY

For an output of 5 V, simply key in 5.00, or press 5 and then the # key, which acts like an Enter key while the * symbol acts as a decimal point. The same goes for the current. The power supply's maximum voltage is 20 V. The maximum current is 1 A.

Please note that there will be no output if you enter a voltage on the keypad and you don't enter a current setting because the power supply will be in a constant current mode. Also, if the voltage is not reached after the load is applied, adjust the current until the voltage is correct.

POWERFUL SMPS

My SMPS can be used for powering any circuit that uses up to 20 V and a current up to 1 A. The voltage input is 220 V, which is the voltage in our country. If your mains supply is 110 V, close the switch so the voltage doubler will function. You can also use a 12-V exhaust fan to cool the SMPS in case the ambient temperature in your area is high. ☑

Noel Rios (noelrios7@yahoo.com and noelrios7@gmail.com) is an electronics design engineer who has worked with various semiconductor and electronics companies like Microcircuits, IMI, Allegro, ASTEC, Stratex, Inno vatronix, Applied Systems, and Micrologic. His interests include computers, embedded control, power conversion, GPIB control, and testing and measurement. Noel accepts C firmware development work and subcontractor jobs that involve the 8051 (and 8051 derivatives), digital circuits, and analog circuits.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2009/232.

SOURCES

DS89C430 Microcontroller

Maxim Integrated Products | www.maxim-ic.com/direct

MC34072 Op-amp

ON Semiconductor | www.onsemi.com

TL494 PWM Control circuit

Texas Instruments, Inc. | www.ti.com

Precision Linear Encoder Display

Linear encoders are essential parts in many industrial automation applications. Here you learn how to mount an optical linear encoder to a metal lathe used to create small precision parts. A few affordable parts and some C code enable you to transmit vital displacement and speed data to an LCD.

A manual 7" metal lathe is ideal for creating small precision parts. However, a manual lathe does not come with any means of measuring the shape of a part while it's being worked. Typically, the machinist must measure, cut, remeasure, and then repeat the process until the desired dimension is reached.

When using a manual lathe, you have three means of controlling the cut. The first is with the cross-slide that's mounted to a carriage (see [Photo 1a](#)). The cross-slide controls the cutting bit perpendicular to the work piece. It allows a machinist to faceoff the end of a part and move the cutting bit in and out for turning the diameter.

The second way to control the cut is with the compound rest or top slide. This sits atop the cross-slide and holds the cutting bit. The compound rest can be used to cut angles and make fine depth adjustments in turning operations. Both the cross-slide and the compound rest have vernier indicators stamped in a ring for determining their movement with a 0.001" resolution. Driven by the need for accuracy, I installed inexpensive electronic indicators that perform the same task with more convenience and have a resolution of 0.0001".

The third option is to control the carriage travel. The carriage travels axially along the part and thus controls the axial placement of the cutting tool mounted on it. The carriage travel on my lathe is less than 12". A commercially available means of accurately determining carriage travel is expensive, costing several times what I paid for the lathe in the first place!

So, driven by the need for precision on all fronts, I

began investigating various linear encoders. I was looking for something that could match the resolution of the compound and cross-slide indicators.

LINEAR TRANSDUCERS

Linear transducers come in all sorts of styles. One commonly used linear transducer is the linear variable differential transformer, or LVDT. The LVDT has three sets of coils placed end to end around the tube. The center coil is the primary, and the outer two coils are the secondaries. A ferrometallic core is attached to an object and run through the tube. An alternating current is driven through the primary causing a voltage to be produced in the secondary that is proportional to that of the primary. These devices had the precision I required; however, when I began investigating them, I found out that they were just plain too long. For example, an LVDT with an 8" stroke had a 31" body, which was longer than my lathe. They were also quite expensive, so I moved on.

I remembered scrapping an HP DeskJet printer and finding inside a strange plastic strip just behind the printer cartridges. An optical encoder strip! When I scrapped the printer, I saved both the encoder strip and the reader. The optical reader was an Agilent HEDS-9700 quadrature encoder with a TTL output. A quick Internet search gave me technical data, so I promptly hooked up the reader and encoder strip to my counter and a micrometer. I was disappointed that the resolution was only about 0.007"—good enough for a printer, but not quite enough for this project. I carefully tucked the encoder strip away for another day.

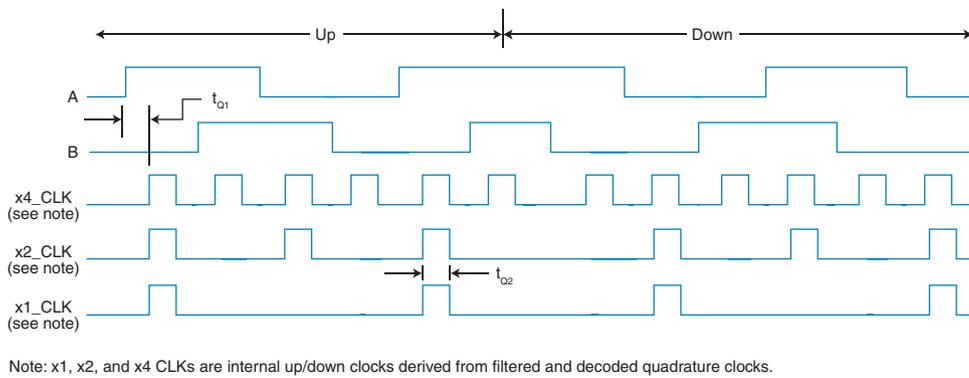


Figure 1—These are A and B quadrature signals and options of X1, X2, and X4 decoded outputs. Using X4 gives the highest resolution.

Another option would have been to use what is sometimes referred to as a “yo-yo,” or a pot string encoder. This is a string or wire wound around a spring-tensioned drum attached to a multi-turn potentiometer. As the string is pulled off the pulley and turns the potentiometer, a ratiometric voltage is produced that can be converted into displacement. Would this work? If I used, say, a 0.5- to 4.5-V range with the built-in 10-bit ADC in a Microchip Technology PIC microcontroller, the resolution would

have been only 0.0039” per bit. That wouldn’t have been good enough. I would have had to go up to a 16-bit ADC to get good enough resolution. I considered purchasing a chip, but I still wasn’t quite satisfied. I kept looking.

One day while browsing on e-Bay, I found the solution: a 12” optical linear encoder with a 1- μ m resolution manufactured by RSF Elektronik. Typically, an optical linear encoder uses a glass scale with a light source on one side and a set of four photosensors on the other. A scanning

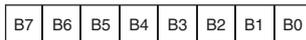
reticle with four windows sits in front of the photosensors. The reticle and photosensors are spaced to give signals that are 90° apart. The signals are then combined to produce two symmetrical sinusoidal outputs 90° out of phase. Some encoders output this signal directly. My unit converts it to a TTL quadrature signal.

The optical encoder fit perfectly on the back of the bed (see Photo 1b). The unit was ideal, but I couldn’t afford the display. At work, I’ve purchased quadrature displays for about \$500, which is way outside my budget. So, I set out to make my own with what I had on hand. I came close.

DESIGN OVERVIEW

The optical encoder outputs a TTL quadrature signal (see Figure 1). A quadrature signal has an A and a B component that are 90° out of phase. This arrangement works well to convey displacement by counting the

IR. The IR is an 8-bit register that fetches instruction bytes from the received datastream and executes them to perform such functions as setting up the operating mode for the chip (load the MDR) and data transfer among the various registers.



- B2 B1 B0 = XXX (Don't care)
- B5 B4 B3 = 000: Select none
- = 001: Select MDR0
- = 010: Select MDR1
- = 011: Select DTR
- = 100: Select CNTR
- = 101: Select OTR
- = 110: Select STR
- = 111: Select none
- B7 B6 = 00: CLR register
- = 01: RD register
- = 10: WR register
- = 11: LOAD register

The actions of the four functions — CLR, RD, WR, and LOAD — are elaborated below.

Number of bytes	OP Code	Register	Operation
1	CLR	MDR0 MDR1 DTR CNTR OTR STR	Clear MDR0 to zero Clear MDR1 to zero None Clear CNTR to zero None Clear STR to zero
2 to 5	RD	MDR0 MDR1 DTR CNTR OTR STR	Output MDR0 serially to TXD (MISO) Output MDR1 serially to TXD (MISO) None Transfer CNTR to OTR, then output OTR serially on TXD (MISO) Output OTR serially to TXD (MISO) Output STR serially to TXD (MISO)
2 to 5	WR	MDR0 MDR1 DTR CNTR None OTR	Write serial data at RXD (MOSI) into MDR0 Write serial data at RXD (MOSI) into MDR1 Write serial data at RXD (MOSI) into DTR None
1	LOAD	STR MDR0 None MDR1 None DTR CNTR OTR	None None Transfer DTR to CNTR in “parallel” Transfer CNTR to OTR in “parallel”

Figure 2—These are IR register commands and the effects of the four functions: CLR, RD, WR, and LOAD. The register to be operated on is specified in B5–B3. The action to perform is contained in B6 and B7. (Source: LSI Computer Systems)

transition edges and direction by determining which signal comes first, A or B. Each cycle of data contains four transitions, two on A and two on B.

There are three ways the transitions can be counted that determine the resolution of the measurement (see Figure 1). The first way is the 1× method, where the count is obtained only on the transition of A for each full cycle. This gives the coarsest resolution. Alternatively, the 2× way is to count the transition on A and the transition on B for each cycle. This is twice the resolution of 1×. The finest resolution is the 4× mode where each transition of A and B is counted. I wanted the finest resolution, so I was looking to implement the 4× mode.

My first approach was to use a Microchip Technology PIC18F2320 because I had several on hand. I was thinking of decoding the signal directly using the interrupt on change available for Port B. Then I thought about the carriage travel speed. In a hurry, I can move the carriage several inches per second. At 2" per second, I would need to decode a total of 101,600 pulses and then display the results in real-world units. Running at the maximum of 40 MHz, I would need to decode to display in 98 instructions. With a floating-point multiply required, that approach went right out.

My second approach was to use an LSI Computer Systems LS7084 quadrature clock converter, which I had on hand. The LS708 reads the quadrature signal and outputs a clock and a direction, thus simplifying the interpretation. All I had to do was monitor the direction and then increment or decrement a counter in interrupt. Four times a second, the PIC performed a floating-point multiply and displayed the results on an LCD. I thought everything was fine until I hooked up a bench-top counter to the LS7084 and compared the results to the PIC18F2320 output. I was missing counts, so I changed the crystal and upped the processing speed. I was still missing counts when I moved the carriage too fast, so I tried the internal phase-locked

Listing 1—Communicating with the LS7366R is simple. The `write_register8()` function is used to set up the counter. It is also used to move the count contents to buffer OTR immediately prior to a read to prevent missing counts.

```
void write_register8(int cmd, int data)
{
  // Examples:
  // write_register8(CNTR_CLR, 0); clear counter
  // write_register8(MDR0_WR, CNTR_AB | FREE_RUN | CNTR_OTR); set
  // MDR0 as a free counter, on index transfer CNTR to OTR
  // write_register8(OTR_LD, 0); move CNTR contents to OTR for OTR read
  RA5=1;
  RA5=0;
  SPI_WRITE(cmd); // set IR to write to specified register
  SPI_WRITE(data); // send data
  RA5=1;
}
```

loop (PLL) to increase the clock speed to 40 MHz. As it turned out, I had an old compiler version that did not support the PLL option. Purchasing a new compiler for this project was not an option.

For my final approach, I broke down and purchased an LSI Computer Systems LS7366R 32-bit quadrature counter with a serial interface. (I bought two for about \$30.) This put all of the processing burden on the LS7366R and left the PIC18F2320 with almost nothing to do. There is more than one way to skin a cat.

COUNTER

The LS7366R quadrature counter has loads of features for all sorts of comparator, counting, or quadrature interface operations. It can use the index signal from the encoder to output two marker flags that can be used as alarms or trigger microcontroller interrupts. These can also be used for over travel or set point travel indication. At 5-V operation in single-count mode, it has a maximum counting frequency of 40 MHz. It is configured with read and write operations over the SPI to function in the desired mode of operation. In quadrature mode, it has ×1, ×2, or ×4 mode decoding with a filter. It can be set to Modulo-N, Non-recycle, or Range-limited operation. It can be set to 8-, 16-, 24-, or 32-bit configurations. A status register stores count-related events such as carry, borrow, index latch, power loss, direction, and sign. The counting register can be zeroed or an initial value can be placed in it.

The back of the specification sheet for the serial counter contains assembly code for the serial interface to a PIC18Cxxx. I prefer to work in C language. So, using the assembly code as a guide, I wrote my own driver compatible with my CCS compiler.

The serial interface is a synchronous four-wire SPI bus controlled by the microcontroller's SCK clock. Each transmission is organized in blocks of 1 to 5 bytes of data. Communication is initiated by a high-to-low transition of the SS pin. The first byte received is always an instruction byte. The second through fifth bytes are data. The communication is terminated by a low-to-high transition of the SS pin. Data is shifted into the MOSI pin most significant byte (MSB) first on high-transition leading edges of the SCK clock. Data is shifted out of the MISO pin MSB first on low transition trailing edges of the SCK clock. Write commands cannot be combined with read commands.

The counter can be configured as a 1-, 2-, 3-, or 4-byte counter. All the registers will automatically adjust to the specified byte width. For example, if the counter is configured as a 2-byte counter, it expects 2 bytes in reading and writing operations following the instruction byte. If 3 bytes are sent to a 2-byte configured chip, then the third byte is ignored.

The LS7366R uses five registers: DTR, CNTR, STR, MDR0, and MDR1. DTR is an input data register for storing values to be written to the CNTR. It is also used for Modulo-n and Range-limit comparisons. CNTR

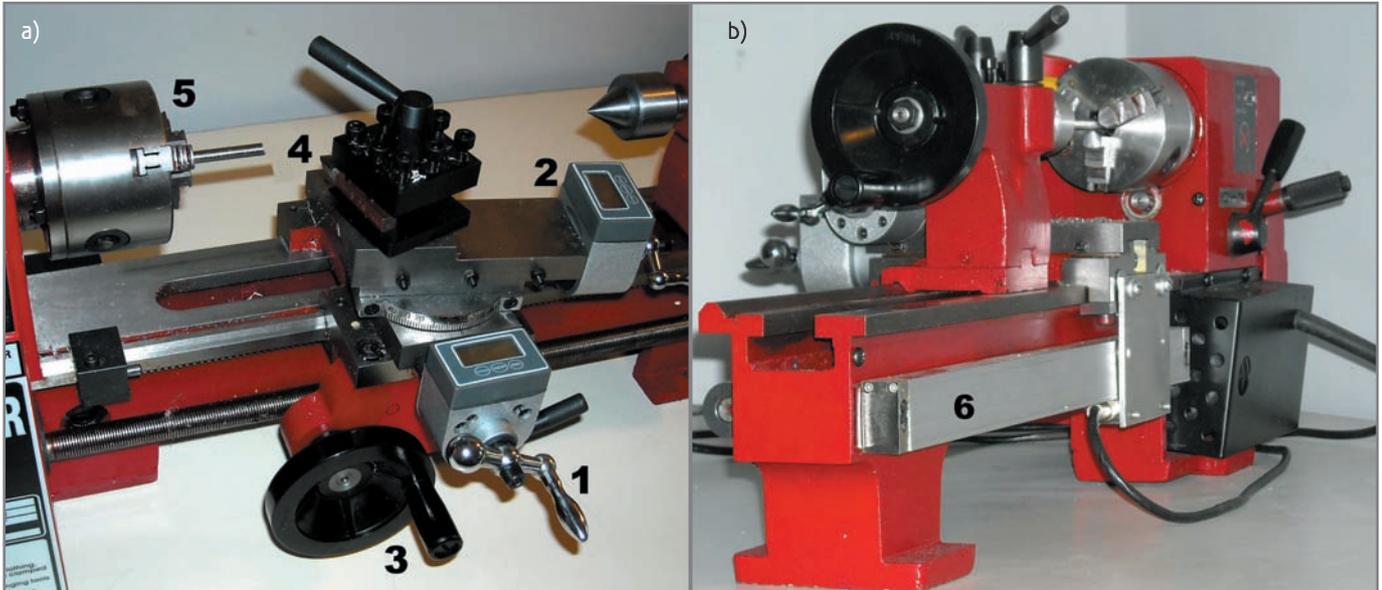


Photo 1a—This is a typical manual metal-working lathe. You can see the cross slide with indicator (1), the compound top slide with indicator (2), the carriage hand-wheel for axial travel (3), a tool post (4), and a spindle with chuck (5). **1b**—I mounted the optical linear encoder (6) to the back of the bed. It is mounted upside down to prevent debris from entering the slide.

is the counter, which tracks up and down counts from the quadrature clock signal at A and B. In non-quadrature operation, it counts each up and down pulse on input A. The CNTR can be cleared, loaded from the DTR, or transferred out to the OTR. The OTR serves as a buffer for CNTR. To prevent missed counts, the CNTR data is transferred to the OTR and then the OTR is output to the MISO pin. STR is the status register indicating carry, borrow, compare event,

encoder index, count enabled, power loss, count direction and sign. MDR0 and 1 set the operating mode for the counter.

The IR command byte controls the LS7366R (see [Figure 2](#)). The appropriate register is selected with bits 3 to 5. One of four operations on that register is selected in bits 6 and 7. The four operations are Clear (CLR), Read (RD), Write (WR), and LOAD.

All those specifications are pretty dry. There is nothing like an example

to really see how simple it is to configure and read a count (see [Listing 1](#)). Let's first clear the counter. I set up the driver defines so I didn't have to constantly transfer instructions to hex. The IR command for clearing CNTR is 0x20 or CNTR_CLR. We want to write a single byte. Using my driver interface, the program line is simply `write_register8(command, data)`, or `write_register8(CNTR_CLR, 0)`. Diving into the routine, you see what happens. First, the SS pin (RA5 in this case) is toggled high (just to make sure it actually is high) then low. Using the CCS compiler's built-in `SPI_Write()` function, the command is written followed by the data. The SS pin is then toggled high. I used multiple `write_register8()` commands to configure everything the way I wanted.

Now let's try a counter read (see [Listing 2](#)). The first thing to do is to move the counter data to the OTR register so you can read it at your leisure and not lose any counts. This is done by the line `write_register8(OTR_LD, 0)`. You next read all 32 bits with `counter = read_register32(OTR_RD)`. Again, stepping into the routine you see what happens. First, SS is toggled low and the read command is sent using

Listing 2—`Read_register32()` is used to obtain a 32-bit count value. The SS line (RA5) is toggled low, followed by setting the IR register to read. The counter value is then read in a loop and assigned to the return variable.

```
signed int32 read_register32(int reg) // Read 32-bit counter
{
    // Examples
    // write_register8(OTR_LD, 0); software request to move
    // counter to OTR for a read
    // counter = read_register32(OTR_RD); read counter from OTR
    // to avoid missing counts
    int i; int temp; int32 data=0;
    RA5=1;
    RA5=0;
    SPI_WRITE(reg); // Set IR to read register
    for(i=0; i<4; i++)
    {
        temp=SPI_READ(0x55); // Send 0x55 with read to
        // generate clock for counter

        data = data<<8;
        data= data | temp;
    }
    RA5=1;
    return data;
}
```

SPI_WRITE(OTR_RD). We then loop through four SPI_READ() cycles to read each byte. Notice that I had to send 0x55 with each read to generate a clock for the counter. That took some time to debug. In fact, it drove me nuts for a week. I'm not sure if it is a compiler thing, a PIC thing, or an LS7366R thing. It worked, so I used it. The data is ORed together and then returned. Finally, SS is toggled back high, ending the command.

I started a 32-bit write command but never finished it. It would come

in handy if you are doing any comparisons. I wasn't, so I left it for another day.

While I was setting up the driver, I played around with some of the other modes. In single-cycle counter mode, you quickly discover that it counts each transition, so be careful with that. The range limit counting allows you to count only so high. It then stops until you reverse direction.

The Modulo-n counting was something I haven't seen before in quadrature displays. The Modulo-n operation

returns the remainder of a division. For example, $38 \equiv 2 \pmod{12}$, or in English, $38 \bmod 12 = 2$. This feature can be used to count the number of events in a cycle. This is like counting in base 12. Say you want to count only from 0 to 11 in a cycle. Set the DTR register to 11. The LS7366R then returns $\text{count} \equiv x \pmod{n + 1}$. An example of this would be for counting the hours in a day and tracking AM and PM from a free-running counter.

THE PROGRAM

The program is written in C language. It displays displacement on the first line and carriage bed travel speed on the second line. I wanted to zero the display at any time and convert the readings and display metric or English units. I also wanted to have it come up in units previously used when the display was turned on.

Take a quick look at the program posted on the *Circuit Cellar* FTP site. Variables are set up, ports are configured for proper I/O, and the LCD and counter are initialized. A unit flag is read from EEPROM and the display units are set before entering the main loop. Here you look for the Zero button and the Change of Units button to be pressed. You then read the counter, convert to real-world units, and display. Nothing fancy.

There is one thing to note on the conversion of units: $1 \mu\text{m} = 0.00003937''$. This means that at $9 \mu\text{m}$ you have $0.000354''$, which will be displayed as $0.0004''$ and $10 \mu\text{m}$ will also be displayed as $0.0004''$. Thus, we have a loss of resolution in the English units. I've run into this problem before. It is even more pronounced when converting temperature units of Celsius to Fahrenheit. For example, 25.1°C converts to 77.1°F , while 25.2°C converts to 77.4°F . If you are trying to display 0.1°F , you have a real problem! For my application, I can live with missing a few ten thousandths of an inch. This also falls within the accuracy of the system.

There was one calibration that had



It takes more than a top program to win the Product of the Year Award* five times running

- ▶ For instance, reasonable prices without hidden costs.
- ▶ A fair update policy without mandatory maintenance fees or costly add-on contracts.
- ▶ A competent hotline available free of charge for every customer.
- ▶ A program development and optimization process based on an open discussion with the users.

In other words: EAGLE is one of the world's highest rated schematic capture and PCB-board layout packages because for the last twenty years we have been treating our customers the way we would like to be treated ourselves.

An example: In the course of product development we implemented a

Follow-me Router

which saves an enormous amount of time during the manual routing process. This new function is available free of charge to every customer as are all new features within a main EAGLE release. In this particular case you need to have the autorouter module of version 5.

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

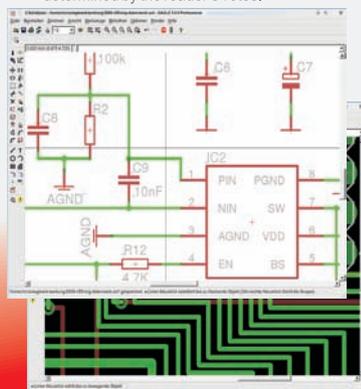
	Light	Standard	Professional
Max. number of schematic sheets	1	99	999
Max. board size	4x3.2 inch	6.4x4 inch	64x64 inch
Max. # of signal layers	2	4	16
Layout or Schematic Editor		\$249	\$498
Layout and Schematic Editor		\$498	\$996
Layout Editor and Autorouter		\$498	\$996
Layout Editor and Schematic Editor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



EAGLE
Version 5

Schematic - Layout - Autorouter
for Windows® Linux® Mac®

* Annual award of the leading German electronics magazine. The winners of eleven categories are determined by the reader's votes.



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

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

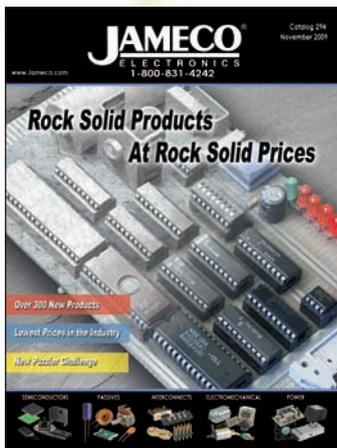
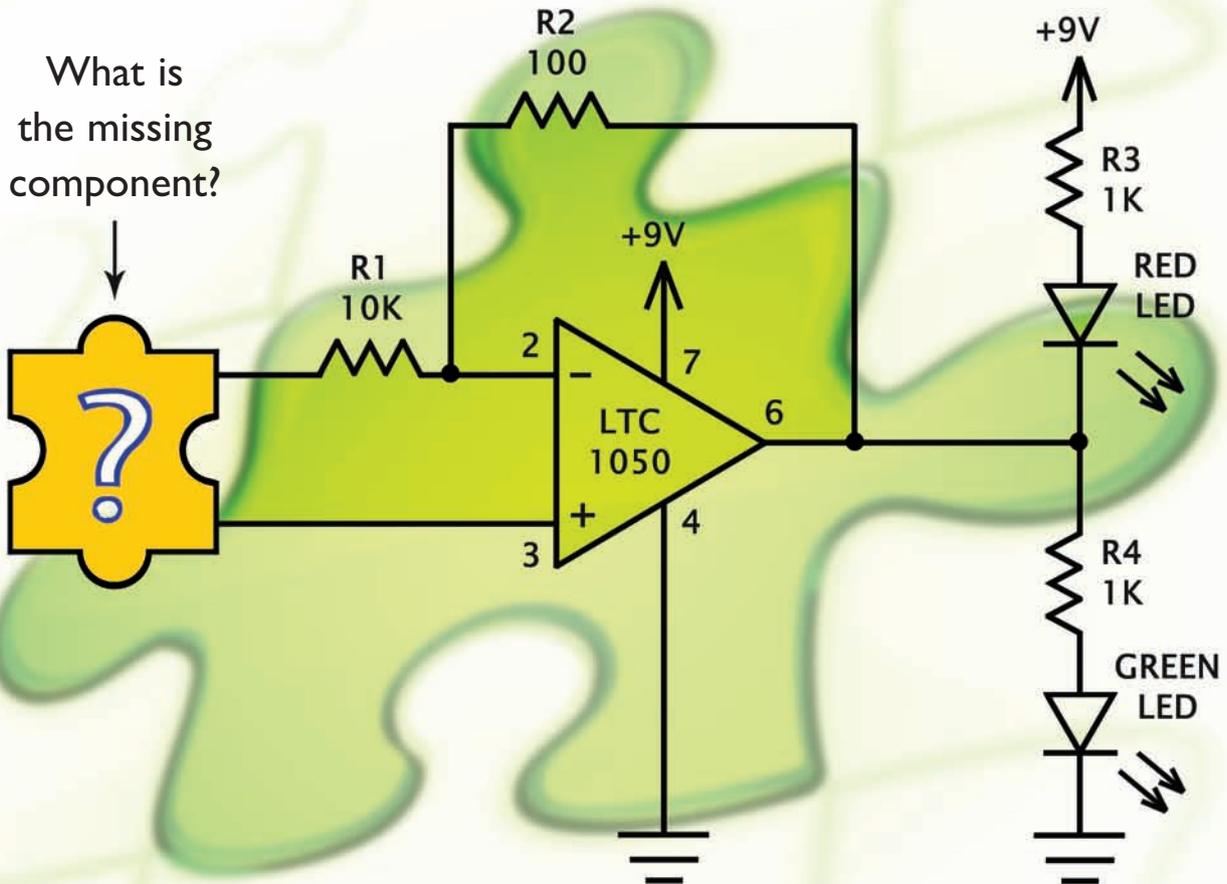
www.cadsoftusa.com

800-858-8355

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

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

Are you up for a challenge?



Industry guru Forrest M. Mims III has created a stumper. Video game designer Bob Wheels needed an inexpensive, counter-clockwise rotation detector for a radio-controlled car that could withstand the busy hands of a teenaged game player and endure lots of punishment. Can you figure out what's missing? Go to www.jameco.com/unravel to see if you are correct and while you are there, sign-up for our free full color catalog.

JAMECO[®]
ELECTRONICS
1-800-831-4242

USB Oscilloscope for \$169.50
 Logic and Spectrum Analyzers, Generator.
www.HobbyLab.us

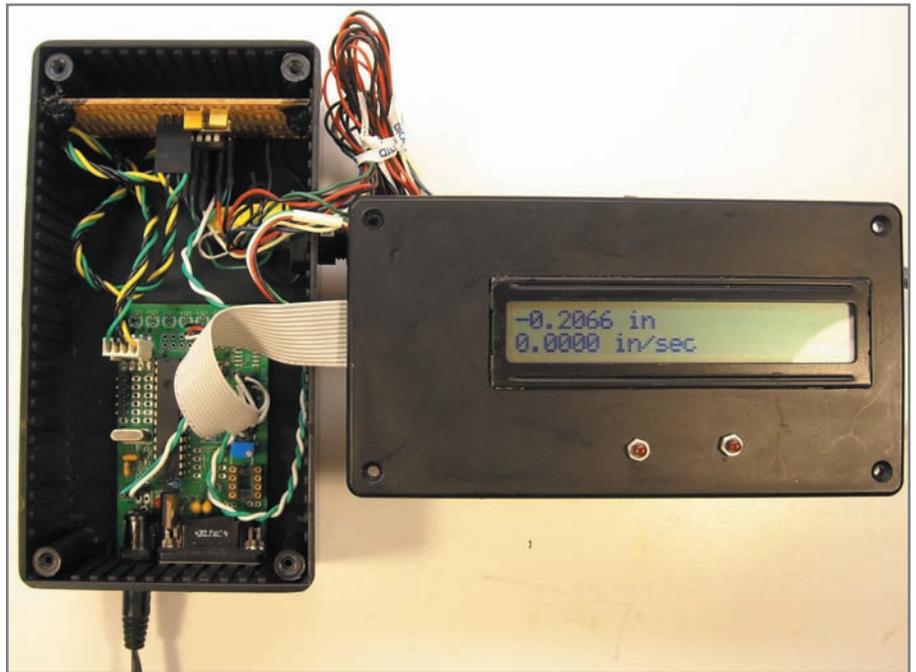


Photo 2—Take a look at the completed project box. The green PCB board contains the PIC18F2320. The board at the top contains the LS7366R. There is plenty of room for a second daughter board.

PIC-SERVO MOTION CONTROL

MOTION CONTROLLERS FOR
 BRUSH, BRUSHLESS AND
 STEPPER MOTORS.

- controller chips
- controller boards

www.picservo.com

JEFFREY KERR, LLC

Celebrating Our 10th Year Anniversary

PC Scope & Generator IN ONE!

PCSGU250
 Oscilloscope + Function Generator
 \$238.55



PGU1000
 USB PC
 Function Generator
 \$242.85

PCSU1000
 USB PC
 Oscilloscope
 \$298.99

[DesignNotes.com](http://www.DesignNotes.com)

What Your Electronic Hobby Stores Used To Be

1-800-957-6867

www.DesignNotes.com

to be performed for accurate bed travel speed display. During the development stage, I used a pin to toggle an LED on and off. I use this as a heartbeat to indicate that the PIC is in the do loop and functioning. To calibrate the bed travel speed, I simply hooked up my data acquisition card to the heartbeat and determined the time per loop. This value was then hard-coded into the software. I wasn't too concerned with bed travel speed accuracy. Bed travel speed is an indication of the feed rate when making an axial cut by hand. It is used for reference only.

PROJECT BOX

Refer to the circuit diagram (see [Figure 3](#)). The encoder inputs A and B are connected to LS7366R input pins 12 and 11, respectively. The clock connected to fCKi and fCKO is required for the filter when in quadrature mode. The serial SPI communications are on pins 4 to 6 and connect directly with the PIC18F2320 SPI pins with the exception of SS, which is connected to RA5.

The assembled project box is shown in [Photo 2](#). I used a prototyping board I had made several years ago

for a PIC16F873. All the pins mapped over to the PIC18F2320, which was quite nice. That board was one of my first and there were lots of errors. I didn't include a picture of the underside where I made all my corrections.

I used a separate prototype board for the LS7366R and connected it to the PIC with a header and cable. I then placed both boards inside a project box and tacked them in place with a spot of glue. There is no power switch. Simply plug in the wall wart to turn it on. If I use this more than anticipated, I'll end up installing a power switch.

PERFORMANCE: STOP TO STOP

How does it all perform? The only way I could think of to compare the two signals was to connect the counter to the A channel of the encoder, multiply by two, and then divide by 1,000 to get the LCD when in metric units. This was as good a method as I could think of to compare. It is not perfect because the counter cannot work with a quadrature signal. I then ran the carriage from stop to stop manually as fast as I could and then very slowly.

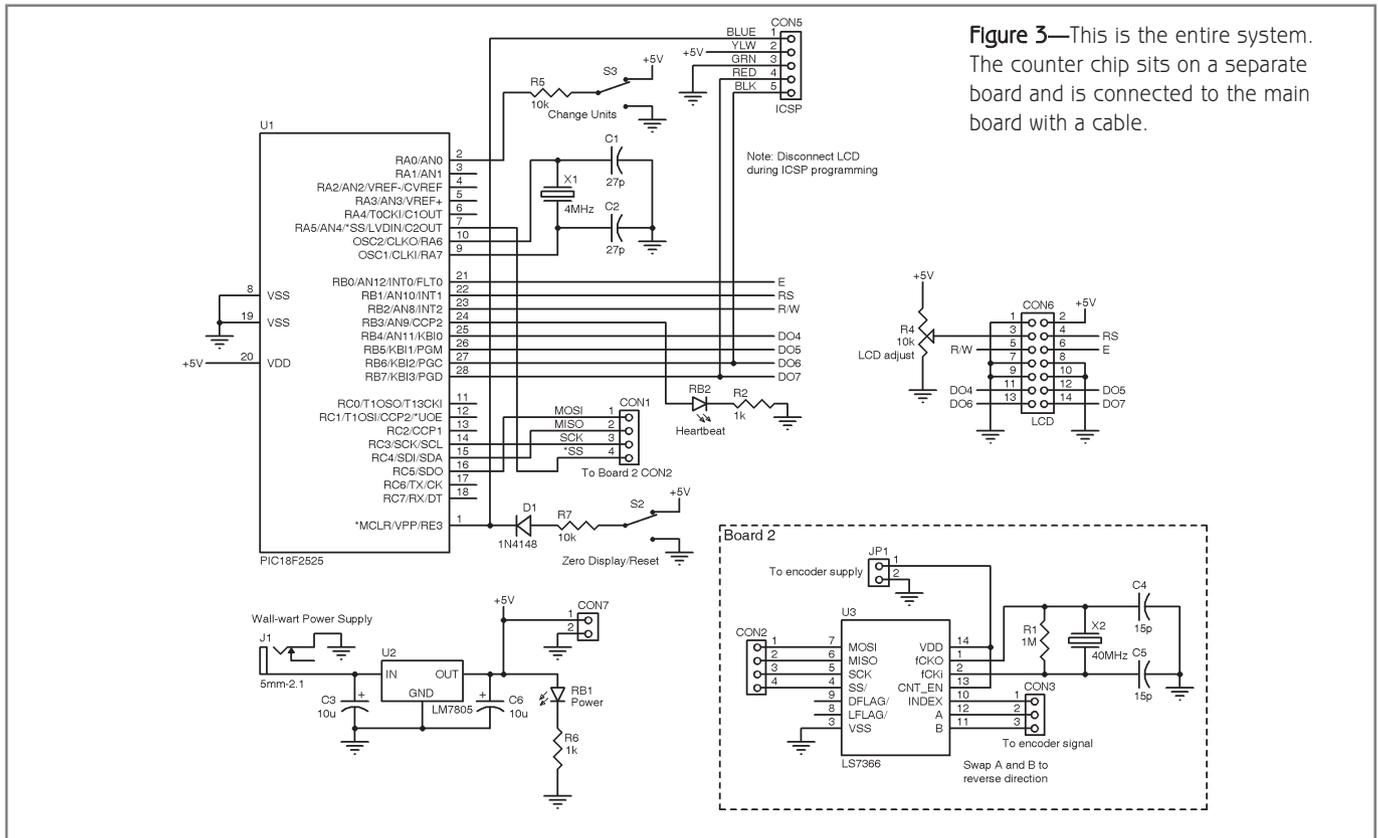


Figure 3—This is the entire system. The counter chip sits on a separate board and is connected to the main board with a cable.

I noted a typical stop-to-stop error of six to 10 counts between the bench-top counter and the display. At least two of those counts can be attributed to the lack of channel A signal when changing directions. I purchased my counter on e-Bay for \$10 so it is not likely to be a highly accurate piece of equipment. A sticker on it says the last time it was calibrated was 1987! Either way, I can certainly live with a $\pm 10 \mu\text{m}$ accuracy.

The LS7366R is a sweet little chip with more than enough muscle for this project, so much so that the PIC18F2320 is not working hard at all. I even had to insert a delay loop to slow down the update rate on the display. With extra processing available, what else can be done? At some point, I would like to add a rotary encoder to the spindle to determine the spindle RPM. I would also like to read a spindle angle with 0.5° accuracy (or better) to accurately place angular marks on a project for circumference division. I have plenty of room for a second board in my project box and more than enough processing power to accomplish this. 🛠️

Carl Stoesz (carl.stoesz@bakerhughes.com) is a mechanical engineer who works as a research and development engineer for Baker Hughes. His focus is on the integration of fiber optic sensors into oil wells for the company's Intelligent Well Systems group. Carl enjoys playing in the electrical engineering sandbox in his spare time.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2009/232.

RESOURCES

Agilent Technologies, "Small Optical Encoder Modules, Technical Data, HEDS-9700 Series," <http://content.honeywell.com/sensing/sensotec/>.

Microchip Technology, "PIC18F2220/2320/4220/4320 Data Sheet," DS39599G, 2007, www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=64.

The Sensors Reference, "Input/Output Devices: Linear Encoder Information & Linear Encoders," Machine Design, www.sensors-transducers.machinedesign.com/guiEdits/Content/bdeee4/bdeee4_3.aspx.

SOURCES

HEDS-9700 Quadrature encoder

Agilent Technologies | www.agilent.com

LS7366R Quadrature counter

LSI Computer Systems | www.lscsi.com

PIC18F2320 Microcontroller

Microchip Technology | www.microchip.com

How to Repurpose a Development Platform

With some knowhow and little bit of creativity, you can pop open an HP financial calculator and use its electronics to develop a variety of special-purpose applications. This article shows you how to use the calculator's screen, keyboard, and PCB as a development platform.

Last year, Hewlett-Packard (HP) introduced a new calculator. By itself, this could be of great interest to a *Circuit Cellar* reader; however, this is not the point of this article, especially as the HP 20b Business Consultant happens to be geared toward financial calculations. No, what is important to a *Circuit Cellar* reader is that this calculator is relatively inexpensive (around \$40 retail price), it has a screen, it has a keyboard, and it has a connector on the PCB that gives access to a serial or RS-232 port, eight GPIO lines, four ADC channels, and the JTAG programming interface of a 36-MHz Atmel AT91SAM7L128 ARM-based system on a chip (SoC) with 128 Kb of flash ROM and 6 KB of RAM (see [Photo 1](#)). In other words, HP created a calculator that is *also* a versatile development platform. Thus, you can use this calculator for small- or medium-quantity special-purpose devices. Applications could range from software-only modifications (e.g., a device designed to help implement a company's specific processes or calculations) to a software/hardware hybrid with the calculator connected to external hardware (e.g., a weather station or automation controller). The possibilities are endless!

HP also introduced a new version of the venerable HP12C based on the same Atmel microprocessor. Like



Photo 1a—This is the HP20b calculator. **b**—The HP20b, inside out! The Atmel microprocessor is hiding under the Bakelite “blob” and the JTAG connector at the bottom.

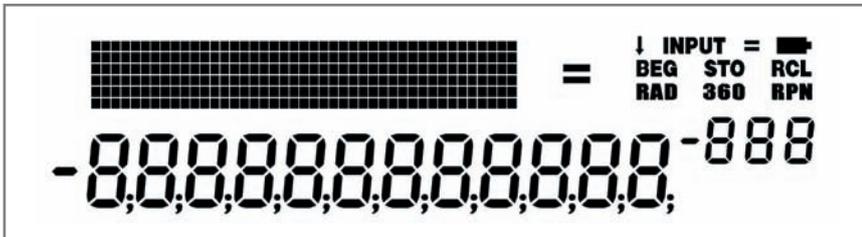


Photo 2—The HP20b screen has both a numerical display and a matrix area for text or graphic display.

the 20b, it has a serial port, but it does not have the JTAG and other exposed inputs and outputs.

PLATFORM OVERVIEW

The calculator by itself is about 15 cm × 8 cm × 1.5 cm and weighs around 110 g. Two 3-V CR2032 lithium batteries—which are expected to last for one year—power the calculator. In addition to a 37-key keyboard, the calculator has a 400-segment screen that has a 15-digit, seven-segment display area at the bottom (12 large numbers and three smaller ones), a couple of indicators on the top right, and a 6 × 43 pixel matrix that’s suitable for displaying text or small graphics (see Photo 2).

If you remove the battery door, you will see a small six-pin connector. The connector is designed primarily for software updating. (You can update any calculator with a new version of software without having to open it.) This is possible because the microcontroller is flash memory-based and not ROM-based. But you can also use the “serial port” as part of an application that needs a serial communication for debugging. HP has a cable that plugs into this port. It is available provided your project is accepted by the person at HP in charge of distributing the cables (that’s me for the moment) and that you agree not to do anything that could harm HP or the sales of the calculator. You need to use Atmel’s SAM Boot Assistant—which is referred to “SAM-BA” for short, and is also called the “AT91-ISP” for AT91 In-System Programming—software to perform software upgrades using the cables (www.atmel.com/dyn/products/tools_card.asp?tool_id=3883).

When you open the calculator, you have access to the PCB, which contains less than 40 surface-mounted resistors, capacitors, and diodes. You also can access the battery connectors and the Atmel AT91SAM7L128 CPU/SoC (in die form) that’s wire-bonded to the PCB and protected under a blob of black resin.

OPEN THE CALCULATOR

To open the calculator, you must first remove the five screws that hold the top case to the bottom case. Three of them are easily spotted behind the battery cover, two at the top and one located between the two batteries. The last two screws are located under the “no-slip” pad at the bottom of the calculator. You can use an Exacto knife or your fingernail to lift 5 mm of the material on each side and gain access to the screws without having to remove the no-slip pad. This enables you to close the calculator when you are done without leaving a visible

sign that it was opened.

The last step in opening the calculator involves separating the top and bottom cases. The two parts are held together with 10 “clips,” three on each side and two at the top and bottom. The best way to open the calculator is to start by popping the middle clip on the left side, located between the Input and the CashFlow keys. Insert a small flat screwdriver between the black and grey

plastic, push outward, and the clip should pop out. You can then use your thumb and press down and out on the grey plastic to pop the two other clips located next to the screen and Shift keys. Use the screw driver to remove the bottom two clips. After that, you should have no problem opening the calculator with your hands alone. (Decreasing the use of the screwdriver reduces the chances of damaging the calculator’s finish.)

PROBE THE PCB

The AT91SAM7L128 is a low-power ARM7 SPU/SoC that can run at up to 36 MHz. It has 128 KB of flash ROM, a 400-segment screen driver (as far as I know, this is the first commercially available ARM7 CPU with a direct LCD driver), 6 KB of RAM, and the standard slew of peripherals. Refer to Figure 1 for more details.^[1]

The AT91SAM7L128 has a lot of advantages for this application: it has a built-in LCD controller, it’s low-power; it has its own power supply (allowing it to work directly from the 3-V battery without any external power-regulation circuitry); it has a charge pump for driving the LCD; and it has quite a lot of flash ROM for application storage and future upgrades. It also has its own internal 32-kHz and 2-MHz oscillator. Thus, if your application doesn’t rely on accurate timing (i.e., less than 10%), you don’t need to use an external oscillator, which simplifies the system even more. However, you can connect an external crystal to achieve high accuracy if necessary. The AT91SAM7L128 is all about options!

You will notice other things on the PCB—mainly, unused pads. The most important of them is the 16-pad connector at the bottom of the PCB, which is the JTAG interface. JTAG is an industry-standard system for development. You can simply connect a standard 2.54-mm SMT header to the pad and start developing! In addition, the JTAG connector is placed exactly at the center of the recessed area of the HP 20b’s plastic case located below the battery door. (That area was designed as a place for your engraved name.) You can easily remove the recessed area with a cutting tool (e.g., an Exacto knife). This enables you to have a closed unit with the JTAG pin sticking out for development purposes. Basically, instead of using one of the traditional large shabby-looking (when not on an engineer’s desk) development boards, you can perform development on a small device. Once you are done for the day (or need to excuse yourself for a little while), you just unplug the JTAG and

shifter's part number (ATM3202AR). Note that you may also want to "break" two jumpers—JP6 and JP5—to disconnect the RX and TX pins of the serial connector from the main CPU. (If you don't, remember to set the appropriate GPIO pins to pure inputs with no pull-up.) If you plan to use this chip, you'll need to turn it on and off using port C's I/O 27. This is used to conserve power, of course.

J7 is located on the left of the CPU. It is an eight-pin pad that provides access to the battery voltage/power, ground, and the AT91SAM7L128's four ADCs. They can be used to connect all sorts of analog equipment to the system. They can range from temperature, light, or even force of acceleration sensors. (You could turn your calculator into a two-direction accelerometer for your car! Then you would be able to use it to map your car's power, 0- to 60-MPH time, and more.)

The last connector (10 pins) provides battery voltage, ground, and eight general-purpose I/Os, which are also multiplexed with the four pulse-width modulation generator channels of the AT91SAM7L128 and the SPI. This enables control of up to three peripherals. An IRQ signal is also available, as well as three CPU wake-up pins that can restart the CPU from Off or Backup mode.

For more in-depth information about these connectors and how to use the CPU peripheral, refer to the HP schematics that are included with the system development kit and the AT91SAM7L128 documentation.

SYSTEM DEVELOPMENT

Now that you're familiar with the platform, the question is: How do you do something with it?

The first thing to do is download the system development kit. (Refer to the link in the Resources section at the end of this article, or follow the software and drivers link from the HP 20b page on HP's website.) The kit includes the HP 20b binary (so you can reprogram your calculator with the original firmware if needed), the calculator's schematics

66 In order to program your calculator, you probably will want to use a JTAG interface. Although this is not 100% necessary, it allows you to connect the CPU directly to the development environment, which is a great plus because you can have a one-step solution for downloading your program in the calculator and step-by-step debugging. 99

ATTENTION

PRINT MAGAZINE READERS

BONUS CONTENT NOW AVAILABLE

The following Circuit Cellar bonus content is now available for you to read online or in a downloadable PDF. Just visit Circuit Cellar's home page and click on the link to All Bonus Content.

Issue #228: NimbleSig III

A New and Improved DDS RF Generator
Thomas Alldread

Sound Synthesis Made Simple (Full article plus video example)
A Multi-MIPS Music Box
Peter McCollum

Issue #229: USB I/O Expansion
Brian Millier

Issue #230: Verification and Simulation of FPGA Designs
Sharad Sinha

Issue #231: Buddy Memory Manager
Sitti Amarittapark

Arduino-Based Temperature Display
Mahesh Venkitachalam

Are you interested in writing for Circuit Cellar? Consider a submission to Circuit Cellar's bonus section in the Digital Plus venue. As you see from this statement of availability, the bonus section of Digital Plus is available to all Circuit Cellar readers. Authors are choosing to be published in our bonus section for a variety of reasons. These reasons include but are not limited to:

- Articles of various lengths can be published in the digital venue
- Follow-up articles are published in the bonus section without concern for the impact on the current issue's theme
- Articles may include audio or video enhancements
- Speed to publication. Space restrictions in the print magazine can delay publication. There are fewer restrictions on the digital side.

Whether you want to submit an article for print publication or for publication in the bonus section of Digital Plus, please write to editor@circuitcellar.com to present your ideas.

(which are critical if you plan to do any hardware modification), a readme file with important information, the source code for a sample application, and the “simulation” code that you can use to develop on the PC and the LCD artwork. The artwork is useful if you plan to draw directly on the screen because it will tell you which bit of the ARM LCD controller memory maps to which LCD pixel.

Second, in order to program your calculator, you probably will want to use a JTAG interface. Although this is not 100% necessary, it allows you to connect the CPU directly to the development environment, which is a great plus because you can have a one-step solution for downloading your program in the calculator and step-by-step debugging. If you don't have a JTAG interface, you need to compile a binary and send it to the calculator using the serial port. You can purchase a JTAG interface—from Atmel or other retailers—that is priced in the \$100 to \$200 range (look for SAM-ICE).

You will also need a development environment. HP is using the IAR Systems embedded workbench (you can download a test version at www.iar.com), but it should be possible to also use other environments or even the gnu toolkit. You can look, for example, at the HP20b repurposing project group's wiki at <http://hpwiki.fatcity.com/doku.php>. It uses the GNU toolchain.

Finally, there's the Visual C++ option. You can use Visual C++ if you want to develop your application on a PC. Although this is not a necessity, doing most of your application development and debugging on the PC can greatly simplify and speed up the process.

Let's assume that you will use HP's advised setup: JTAG, IAR, and Visual C++. Files called `HP20b_c.exe` and `at91sam7l128_flash.bin` are included in the SDK package. This is the application example (PC and target versions) created by HP in order to provide a good starting point for custom application development.

This is a simple “guess the number” application with a number between 0

and 10,000. The number keys enable you to enter a number. Input is used to find out if the number to guess is larger, smaller, or if you won. The Backspace key is used to erase the last digit entered. The On key is used to cancel an entry or a game.

The framework provided by HP includes all the drivers and initialization, as well as the power management. Within this framework, your main application function is the `KeyPress` function called each time you press a key. Key codes are given as a two-digit integer (`yx`), where `y` is the row of the key and `x` is the column. For example, the open parenthesis key keycode is 32.

The second function that you need to provide is the `init` function, which is called when the memory is erased or reset. In this framework, your application is implemented as a kind of state machine that updates itself each time a key is pressed. This leaves us with the problem of where and how to store the state.

The Atmel chip has 6 KB of RAM, which is a strange number. But there is a reason for this, and it is linked with the power-saving features of the AT91SAM7L128.

The CPU has four power modes. (Well, this is a simplification. But as far as I'm concerned for the moment, four power modes of interest.) The first one is “running” when the CPU is actively calculating. This is the mode to use when your application is “running.” The second mode is Idle. In this mode, the CPU is paused. This is the mode used when a key press has been executed, but the key has not been released yet. The CPU can't turn off because it needs to poll the keyboard regularly to detect a subsequent key press or release. In the third mode, the CPU is turned off, but the LCD is still running. This is the mode used as soon as all keys are released but the LCD is on. In the last mode, everything is off except the back-up memory. This is the mode used when you turn off the calculator.

Out of the 6 KB of RAM, 4 KB are in the CPU and 2 KB are the back-up memory. So, as soon as the CPU is

off—which is pretty much all the time because most of the time is spent when the LCD is on but a key isn't pressed—this memory is lost. These 4 KB are perfect for storing the stack and temporary variables. The other 2 KB of RAM (or back-up RAM) stay powered even when the CPU is off, and this is where you need to store your application state.

I don't know how to use the IAR compiler to indicate that a specific variable must be in the back-up RAM section, so everything is handled with a `TApplication` structure (`application.h`) that contains the application state that needs to be kept active. An `Application` variable is then created at address `0x300000` (the beginning of the back-up RAM) and is used throughout the program. Note: Do not exceed 2 KB!

HP also provides screen-drawing functions (`graphics.h`) as part of the SDK source code. The functions enable you to display text, numbers, and graphics on the screen. They also give you control of the 11 indicators so you don't have to handle each pixel “by hand.”

APPLICATION MODIFICATION

Now that you're familiar with the application architecture, you can start digging in. I suggest that you first look at the PC version of the application in the debugger. Place a breakpoint at the beginning of the `KeyPress` function in the `application.cpp` file. Then you can perform step-by-step debugging to learn how the sample application works.

Moving on, open the IAR project, compile it (F7), and program it on the calculator (Ctrl+D followed by F5 to “run”). Next come modifications to the example application. For example, transform it in an integer-only or floating-point four-operation calculator. Or just start your own project.

DEVELOPMENT TIPS

Before I finish this article, I want to describe a few tricks for developing on the calculator itself. I trust you'll find them useful.

IAR has some issues. For example, it doesn't always properly erase the

flash ROM before trying to reprogram the flash memory, so it is usually better to erase the flash manually with the cable. (Press and hold Erase, press Reset, press On, and release the Erase key.)

Sometimes, the JTAG link/connection "breaks." You need to unplug the SAM_ICE and then plug it in again.

You can't debug a CPU when it's off. In fact, the debugger closes when CPU contact is lost. This makes it hard to debug in the HP framework because it continuously turns off the CPU. So, when you are developing, I recommend commenting out the calls to `TurnCalcOff` and `PutCalcIdle` to avoid entering these modes. ☒

Author's note: If you have questions or need more information, you can contact me directly or interact with other knowledgeable HP people in any of the following public forums: www.hpmuseum.org/cgi-sys/cgiwrap/hpmuseum/forum.cgi, <http://hpwiki.fatcity.com/doku.php>, or comp.sys.hp48.

Cyrille de Brébisson (cyrille@hp.com) is a French engineer living in Boise, Idaho, where he works for Hewlett-Packard's calculator division. He earned a degree in computer science, electronics, and control from one of the French Grande Écoles. Cyrille is particularly interested in optimization.

REFERENCE

[1] Atmel Corp., "AT91 ARM Thumb-based Microcontroller," 6257AS-ATARM, 2008, http://atmel.com/dyn/resources/prod_documents/6257s.pdf.

RESOURCE

Hewlett-Packard, HP 20b Business Consultant Financial Calculator, Drivers and Software Download, <http://h20000.www2.hp.com/bizsupport/TechSupport/SoftwareIndex.jsp?lang=en&cc=us&prodNameId=3732535&prodTypeId=215348&prodSeriesId=3732534&swLang=8&taskId=135&swEnvOID=54>.

SOURCES

AT91SAM7L128 SoC
Atmel Corp. | www.atmel.com

HP 20b Business Consultant Financial Calculator
Hewlett-Packard | www.hp.com



AP CIRCUITS

PCB Fabrication Since 1984

As low as...

\$9.95

each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com







Save Up To 60% on Electronic Components



New Ethernet Mini Board

- * Ideal for Connecting Microcontrollers and Control Boards to a Network
- * Supports both 5V and 3.3V Systems
- * Suitable for both Full and Half Duplex Modes

ONLY \$19.90



Upgraded ET-AVR Stamp Module

- * Utilizes the ATmega128 Microcontroller with 128k Flash Memory
- * A/D, SPI, I2C, PWM All Supported
- * Up to 53 I/O Points
- * Ideal Embedded Controller

ONLY \$19.90



10A Dual Battery Solar Regulator

- * Automatically Switches between Batteries depending on the state of Charge
- * Microcontroller Control with Serial Mode PWM Control for High Efficiency
- * Suitable for 12V and 24V Systems

ONLY \$42.90

www.futurlec.com



Passing Parameters

How do you partition a design problem into manageable modules? With this article, you'll learn why it's smart to pass parameters in the first place, and then find out how to pass parameters and receive results with the C language.

I would like to talk about passing parameters and receiving results using the C language. I plan to cover why you would want to pass parameters in the first place and how the language performs this operation. This is a small portion of a larger topic: How do you partition a design problem into manageable modules that can be designed, coded, tested, and released into the complete system? That topic is at least a master's thesis, and probably more in the doctoral thesis arena. Let's start with the basics of passing parameters and then take a look at design partitioning.

PASSING BASICS

While searching the Internet for concisely worded material as background, I came up with several discussions on this topic. (An example is S. Joon's website noted in the Resources section at the end of this article.) Let's start with the simple rule that C passes parameters by value. The compiler makes a copy of the value to be passed and places it on the stack before the routine is called. This is certainly straightforward enough.

In [Listing 1](#), I defined two variables: INT16 A and B. Both variables are passed to the routine INT16 Avg(INT16 x, INT16 y). This routine calculates the average of the parameters passed and returns the average.

There are several points to cover in this simple example. Copies of the values of variables A

and B are passed to the Avg() routine. That routine performs an operation on them and then returns the result of the operation. That result is then saved in the variable Result. The variables A, B, and Result are defined in the routine Funct1() and are only known to the scope of the routine Funct1().

For a discussion of scope in the C language, refer to Tony Zhang's "Sams Teach Yourself C in 24 Hours," which I list in the Resources section of this article. The variables total, x, and y are only known to the Avg() routine. This design approach makes sense, considering many routines can use the Avg() routine and its mechanism remains hidden. This hiding is a great feature in several respects. On large complicated projects, you can be working on Funct1() and I can be working on Avg(). I can give you my latest version of the routine. You can use it without a complete understanding of the implementation. You don't have to understand my work, just use it. On simpler projects, this hiding can be used by one designer to divide the work. The overall problem is divided into smaller operations, and each operation is designed, coded, and tested.

VARIABLES GLOBAL

Let's stop here for a bit and talk about making all the variables global. Making them global would expose all the variables to all the routines and parameter passing would go away. This

Listing 1—A simple example of parameter passing by value

```
INT16 Avg(INT16 x, INT16 y) {
    INT16 total;

    total = (x+y)/2; // calculate the average
    return(total);
} // end of INT16 Avg(INT16 x, INT16 y)

void Funct1(void) {
    INT16 A,B,Result;

    //...do some stuff...
    Result = Avg(A,B); // get the average of A & B
    //... do some more stuff

} // end of void Funct1(void)
```

seems like a shortcut (and simpler), but it's just a good way to lose control of your project. Consider [Listing 2](#) where all variables are global. It's a lot less code, which can be a plus. But if you want to call `Avg()` from several places in your code, with different values you would have to expose the variables `A` and `B` (by declaring them as extern and transfer those values into `A` and `B` before the call. I hope you can see how cumbersome this would be for anything but a trivial program. Also, realize how fast you would lose control of your variables. Based on the design, you might sometimes have a compelling reason for making a variable global in scope. Perhaps a timer variable showing operating hours is a candidate to be global. If this timer were incremented by a timer interrupt routine and all the other uses were to read that variable for information purposes, I might consider making that timer a global variable so that any routine may use its value.

Let's say you want to pass a parameter, operate on it, and return the

changed variable. Consider this simple code fragment:

```
c = toupper(c);
```

In this example, a routine from one of the C libraries is used to convert a character that might be uppercase or lowercase to a character that is uppercase. So, we pass the variable by value and the return value gets stored in the address of the variable just passed. Caution! The C routines like `toupper()` are actually passed an `int` and also return an `int`. So be careful to understand C character representation before you start using them.

What if we want to pass a variable and have the routine modify the original instance of that variable? In [Listing 3](#), we pass two variables by value. We also pass the third by value, but in this case that value is the variable's address. Knowing the address, the called routine can modify the contents of that third variable. With one variable, we could use the return parameter of the function to change the variable. If you have two variables to change, then passing the addresses of each would be the method

to use. The notation `&Avg` is the address of variable `Avg`. The variable `Avg` is stored in a memory location `&Avg`. We are giving the address of that location to the `Operation()` routine. Now the `Operation()` routine gets an address, but an address to what? Well, that's where the `INT16 *Total` comes in. That informs the `Operation()` routine that the parameter is a pointer to an `INT16`-type variable. Said another way, it's the address of an `INT16`-type variable. While `c = toupper(c);` works with one variable, when you need to change two or more variables the passing by reference is the method to use.

Life is good. But what if we need to pass 20 variables to a routine and receive 10 results? Good point. We would not want to do all that using the stack mechanism. It would be too much of a burden on the code and CPU time. Well, let's just make them all global. Not so fast. All the problems with global variables are still relevant. Take a look at [Listing 4](#). Can you follow the code? Just start at the top. The `#define` defines the size of some character array (10 characters) that's used in the structure. Remember to leave space for the trailing null.

Next, we define a structure naming that definition as `PARAMETER_BLOCK`. Remember, it is good practice to use uppercase for constants, and the `PARAMETER_BLOCK` definition will not change as we run the code.

We then save room for one such `PARAMETER_BLOCK`, and name it `MyData`. `MyData` is where the structure is located. `MyData.ValueA` references the `ValueA` variable in the structure. `MyData.DefinitionString[2]` references the third (arrays start at 0) element in the character array.

Next, we tell the compiler that there is a function somewhere in our code named `SomeFunction()`, and that the function is passed a pointer to the start of a `PARAMETER_BLOCK` structure. In `SomeOtherFunction()`, we are working with our data and have it all loaded in the structure `MyData` before we call the routine that operates on the data.

And finally, `TheBestFunctionYet()` does the same without a lot of overhead. In `SomeOtherFunction()` I can put

Listing 2—Making all variables global

```
INT16 total;
INT16 A,B,Result;

void Avg(void) {
    total = (A + B)/2;
}

void Funct1(void) {
    //...do some stuff...
    Avg();
    ;... do some more stuff
}
```

in breakpoints to make sure I've got the correct values for the address of the structure. Once I'm sure the code is correct, I'll change to the faster version. I hope you find Listing 4 to be straightforward. It might make your head hurt, but when you use it successfully for the first time, I bet you'll have a smile from ear to ear.

Let's review. We can make variables global. Fast, but you lose control. We can pass variables by value. This makes a copy and then calls the routine. Good, but it slows down when we have lots of data to pass. We can also pass by reference. That is a good choice if you want the called routine to change the variables. And finally, we can create a structure to hold the data and pass the address of that structure. Think about passing parameters, and the more you work with the different methods, the better you'll understand the benefits of each.

Now let's take the next step toward the deep end of software design. How do you make a module?

MAKE A MODULE

Another word for a module is "file." An actual file from a project I'm working on is available on the *Circuit Cellar* FTP site. The module (file) A2D.C contains all the A2D routines for my project.

Let me explain a bit more about the problem I'm trying to solve. The input to the microprocessor is a full-wave rectified signal. I need to find the signal's peaks and valleys and add a timestamp for each. This project is in an early stage, and I can almost guarantee that the code will change as we move on.

Look at the module, the routines it contains, and the variables and how they're passed. Look how I divided the code—not to copy this code, but to understand my trade-offs and then use them as a basis for your design.

The include statements in lines 19, 21, and 22 inform the compiler about the CPU, general constants, and A/D constants. The module A2D.h will be used to give other modules needed information about this module's contents. Lines 28 and 29 define a structure that will be used to hold A/D input data and timestamps. Variables `inpt` and `tStamp` are temporary

Listing 3—Passing by reference

```
void Operation(INT16 x, INT16 y, INT16 *Total) {
    *Total = *Total + x + y;    // Keep a running total
} // end of void Operation(INT16 x, INT16 y, INT16 *Total) {

void Funct1(void) {
    INT16 A, B, Avg;          // declare these variables here

    //...do some stuff...
    Operation(A, B, &Avg);    //pass 2 variables(A&B) and address of
    Avg                       //... do some more stuff

} // end of void Funct1(void) {
```

variables for the input A/D reading and its timestamp. The routine `InitWaveIn()` does all the initialization required for my variables, but not the hardware for the A/D converter. The routine `InitA2DHDw()` initializes all the A/D converter hardware set-up. The variable `A2DCounter` is used to slow down the rate of new readings.

As I explained, this is very early code. I'm not sure of the timing of all the components. How fast do I need

data versus how long will it take to process that data? I need to run the A/D converter as fast as possible, and then I'll slow it down by using this counter. In the final version, I'll use the CPU internal registers to run the A/D converter at the rate suitable for the design. But until then, I can easily speed up or slow down the amount of data feed to my processing routines.

Lines 75 and 76 define the A/D converter interrupt routine. In that

Listing 4—Passing a pointer to a structure

```
#define      MAX_DEF_CHARS      10

struct PARAMETER_BLOCK {    // define a structure
    INT16 ValueA;
    INT16 ValueB;
    INT16 ValueC;
    INT16 ValueD;
    INT8  *PointerToValueE;
    INT8  DefinitionString[MAX_DEF_CHARS];
};

struct PARAMETER_BLOCK MyData; // declare an instance of that struct

void SomeFunction(struct PARAMETER_BLOCK *PtrToStruct);

void SomeOtherFunction(void) {
    struct PARAMETER_BLOCK *Ptr2MyData;

    // Do something
    // Fill the MyData structure with appropriate data
    Ptr2MyData = &MyData;    // Point to my data
    SomeFunction(Ptr2MyData); // Call a routine
    // Work with the results.
}

void TheBestFunctionYet(void) {

    // Do something
    // Fill the MyData structure with appropriate data
    SomeFunction(&MyData);    // An even faster call
    // Work with the results.
}
```

routine, Lines 78 and 94 are used to toggle an unused output bit on the CPU. I can then hook up a scope and measure the code's timing. Early in the A/D interrupt, I read the ADC and the CPU clock. Then I start another conversion. All this code has no conditional execution so it takes the same number of CPU clocks to save that data each time through the code. Line 84 is used to slow down the data rate. In this code, I'm using every other A/D reading. When I want to save the data Lines 86–92, I save the ADC and timestamp in an array. I do this using an index into that array. Then I increment the input index to that array. When the index is past the end of the array, I reset that index to the start of the array. This manages the rolling over of the indexes and creates a circular (never ending) buffer to the data.

Lines 98–104 define the variables I think I'll need to detect the peaks and valleys. Think about this: you will only know the peak or valley conclusively *after* you've passed it. In the case of a valley, once the input data starts increasing, you're then sure you've got a valley. So, I declared these variables to help with the process of detection.

Routine FindA2DZero() contains all the code to detect a valley and returns a Yes or No to report the results. The same with FindA2DPeak(). There are two additional routines—WaitA2DZero()

and WaitA2DPeak()—that wait for a valley or peak, respectively.

So here are all the A2D routines I need to do the job. (I hope.) Actually, I'm sure I'll need more, and the ones I defined here will change. Step back and take a look at the routines. What other variables might be needed? Well, the timestamp variable needs data from the CPU timer, and the A2D of course needs to manipulate the A2D hardware in the CPU. But other than that, this module stands alone. I know I'll need to report the actual peaks and valleys I've detected along with their timestamps. But perhaps I could put the code that uses these values in this routine to also cut down on parameter passing and minimizing the exposure of those variables.

SPEED OF EXECUTION

As I write this and explain the code, I realize that all the variables for buffering the A/D data and timestamps are global to this routine. I did that for speed of execution. This code needs to read the data, buffer it, and then process the new readings. Global variables made the most sense in this application.

I hope you take a closer look at your modules and consider parameter passing and the organization of the routines in those modules. Please drop me a line if you have any C topics you would like me to explain. ☺

George Martin (gmm50@att.net) began his career in the aerospace industry in 1969. After five years at a real job, he set out on his own and co-founded a design and manufacturing firm (www.embedded-designer.com). His designs typically include servo-motion control, graphical input and output, data acquisition, and remote control systems. George is a charter member of the Ciarcia Design Works Team. He is currently working on a mobile communications system that announces highway info. He is also a nationally ranked revolver shooter.

PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2009/232.

RESOURCES

S. Joon, "How Are Parameters Passed," 2003, www.comp.nus.edu.sg/~esim/faq/node13.html#SECTION00071000000000000000.

T. Zhang, "Sams Teach Yourself C in 24 Hours," Macmillan Computer Publishing, <http://aelinik.free.fr/c/ch14.htm>.



mach64
PROGRAMMABLE LOGIC
STARTER KIT by nurgo networks LLC

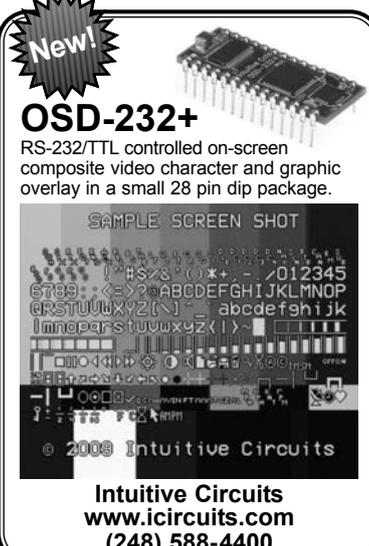
Based on the Lattice ispMach 4064.

Includes 250 page lab manual.

Learn CPLDs the fun way with the MACH64! This complete kit comes with everything you need to take you from mystery to mastery with CPLDs and programmable logic.

Learn to turn *software* into *hardware!*

www.XGAMESTATION.COM



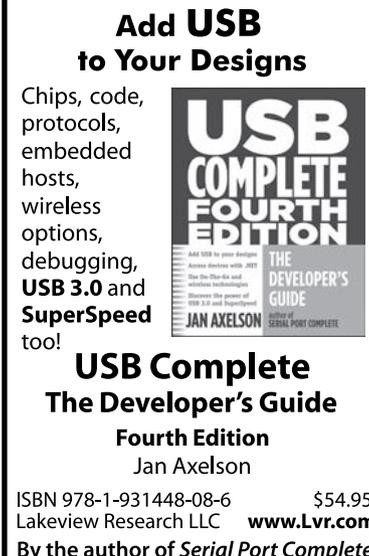
New!

OSD-232+
RS-232/TTL controlled on-screen composite video character and graphic overlay in a small 28 pin dip package.

SAMPLE SCREEN SHOT

© 2009 Intuitive Circuits

Intuitive Circuits
www.icircuits.com
(248) 588-4400



Add USB to Your Designs

Chips, code, protocols, embedded hosts, wireless options, debugging, **USB 3.0 and SuperSpeed** too!

USB Complete
The Developer's Guide
Fourth Edition
Jan Axelson

ISBN 978-1-931448-08-6 \$54.95
Lakeview Research LLC www.Lvr.com
By the author of *Serial Port Complete*

Low-Distortion FSK Generation

This article demonstrates a means to generate FSK modulation with low-distortion and low-transition phase error. You can use a PSoC's digital and analog block features and zero run-time software to achieve this goal.

Frequency Shift Keying (FSK) encodes digital data to be transmitted in an analog fashion by assigning one frequency to a logical 0 and another frequency to a logical 1. Transmitting data in this manner is a simple, robust, and standard method for communicating over telephone lines and a variety of other transmission media. A typical example is shown in [Figure 1](#) with one bit at logic 1 followed by two bits at logic 0 in a repeating sequence.

The design is demonstrated at 1,200 and 2,200 Hz, Bell 202 standard 1,200-baud modem frequencies. Common applications for these operating frequencies include telephone caller ID and Highway Addressable Receiver Transmitter (HART) modems. HART modems are used for transmission over DC power lines, often shared with 4- to 20-mA current loop data. The HART standard waveform, while not enforcing a specific requirement, has a relatively low distortion level. A simple square wave will not do.

The design is implemented in a Cypress Semiconductor CY8C29866 PSoC, although it will easily fit in several other PSoC part families. The PSoC programmable mixed-signal array has programmable digital blocks for generating timing signals, programmable impedance I/O for creating multi-step waveforms, and switched capacitor analog blocks for implementing filters. The resulting FSK generator implementation is all hardware; it takes no run-time code. The device has an efficient 8-bit CPU and lots of digital, analog, and memory resources left over to implement a measurement system that uses FSK as a means to transmit data.

DIGITAL FSK GENERATION

The basic FSK generation structure uses two counters and a pulse-width modulator (PWM), as shown in [Figure 2](#). Counter_freq1 divides the clock source to provide a fixed multiple of the output frequency at the logical 1 frequency, $160 \times 2,200$ Hz, or 352 kHz. Counter_freq0 divides the clock source to provide a fixed multiple of the output frequency at the logical 0 frequency, $160 \times 1,200$ Hz, or 192 kHz.

The PWM divides the summed output of Counter_freq1 and Counter_freq0 by 160 to provide a waveform at one or the other of the signaling frequencies. The PWM (a counter with a controllable output pulse width) has a 50% duty cycle in order to guarantee a square wave to minimize even harmonic distortion.

The remaining trick comes in picking which frequency is transmitted. The enable inputs of Counter_freq1 and Counter_freq0 are driven by the common data signal. Typically, this comes from a UART clocked at the baud rate. Counter_freq1 is enabled when the data input is high. Counter_freq0 has its enable input inverted, so that it runs when the data input is low and stops when the enable input is high.

With one of the counter enables inverted, when the output of one counter is toggling, the other is static. The outputs are added together with an exclusive-OR gate. When one input is low and static, the output follows the other toggling input. When one input is high and static, the output is inverted from the toggling input. Thus, when one of Counter_freq1 or

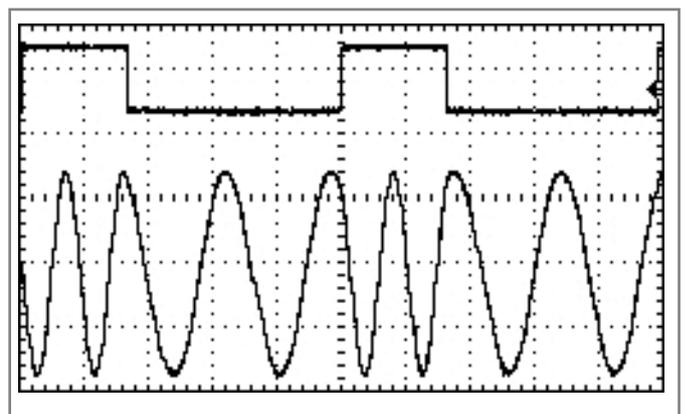


Figure 1—The FSK waveform encodes digital data as a selectable frequency. This example has 1,200 baud data, one bit high at 2.2 kHz, two bits low at 1.2 kHz.

Counter_freq0 is static, the output follows the toggling input of the other Counter. The result is a square wave output generated without software intervention.

For an initial evaluation, a PWM with 1,200 baud alternating 1s and 0s is used to provide the digital data. The input data, the two 160 times multiples of the output frequencies, and the FSK encoded signal are shown in Figure 3.

The HART modem spec allows up to 20 degrees phase error in the transition from one bit to the next. Since the output frequency is derived from a 160x clock, the phase error will be between 0 and 2.25 degrees. This clean phase transition limits the inter-symbol error and enables easier detection of the FSK data at the receiving end. What remains is turning that square wave into a respectable sine wave, or something with low enough harmonics to meet the intent of the spec.

DIGITAL HARMONIC REDUCTION

The square wave output of a PWM has harmonics (all ODD) that fall off as 1/n. Thus, the third harmonic is down 10 dB, the fifth down 14 dB, the seventh down 17 dB, the ninth down 19 dB, and so on.

The higher harmonics are easily filtered, but it is difficult to attenuate the third harmonic because it is very close to the useful band. There are several approaches: lots of filter sections, elliptical filters with strategically placed notches, or digital waveform modifications to reduce near-band harmonics before they are filtered.

Modifying the PWM to be a PWMDB (with a dead band), a three-step waveform can be generated as shown in Figure 4. The dead band is set to be one-third of each half of the waveform, or 67% duty cycle.

One of the PWMDB's outputs is inverted and the two PWMDB outputs are added together. This is not done logically, but rather arithmetically using resistors. One way is to connect a resistor (10 kΩ is sufficient) from each PWMDB output to a common point. Easier yet is using the resistive

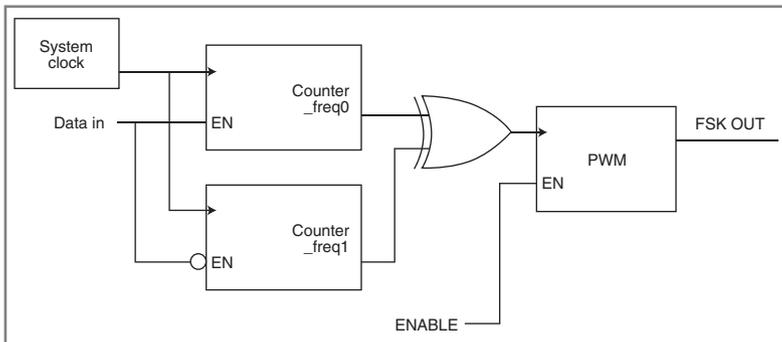


Figure 2—The data input enables one of two counters to generate a over-clocked replicas of the output frequency. The EXOR selects the valid clock as the source to the PWM which divides down to the desired output frequencies.

pull-up and pull-down outputs of the PWMDB GPIO. When both outputs are high, the strong drive of one drives the output to the positive rail (V_{DD}). When both outputs are low, the strong drive of the other output drives the summing point to the negative rail (V_{SS}). When the first PWMDB output is low (pulled down) and the second PWMDB output is high (pulled up), the combined output is resistively divided at a point at $V_{DD}/2$, resulting in the three-step waveform of Figure 4.

But why does the three-step waveform work so well? This waveform has a Fourier series:

$$f(t) = \sum_{n=1}^{\infty} \left(\frac{n\pi}{2} \right) \frac{\sin\left(\frac{n\pi d}{2}\right)}{\left(\frac{n\pi d}{2}\right)} \sin(2\pi f_{SIG} t)$$

where d is the duty cycle of the half wave pulse. Parsing the equation will help understanding: f_{SIG} is the FSK output frequency (either 1,200 or 2,200 Hz). $\sin(n\pi/2)$ yields zero when n is even, guaranteeing that the waveform has only odd harmonics. $\sin(n\pi d/2)$ is zero when $n \times d$ is an even integer, so for $d = 2/3$, $\sin(n\pi d/2) = 0$ when $n = 3, 9, 15 \dots$

The resultant spectrum is shown in Figure 5. “dBc” refers to decibels below carrier. A similar result for $d = 0.8$ results in suppression of harmonics 5, 15, 25 ... The best result is

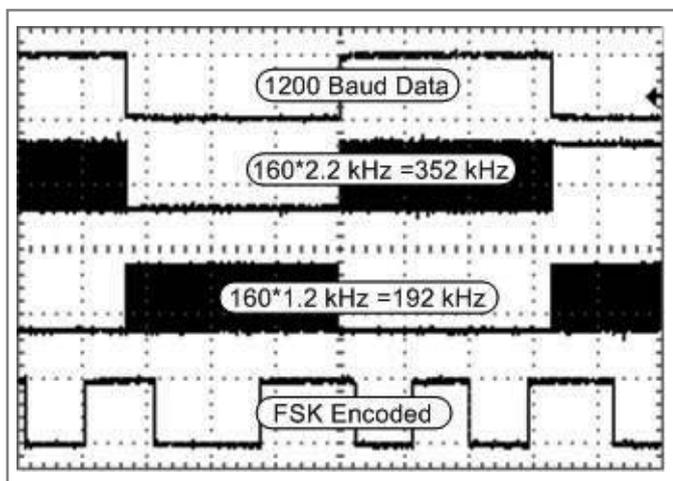


Figure 3—The outputs of the data-enabled counters of Figure 2 clearly show the over-clocked nature of the generator. The PWM divider output results in a square version of the FSK-encoded data.

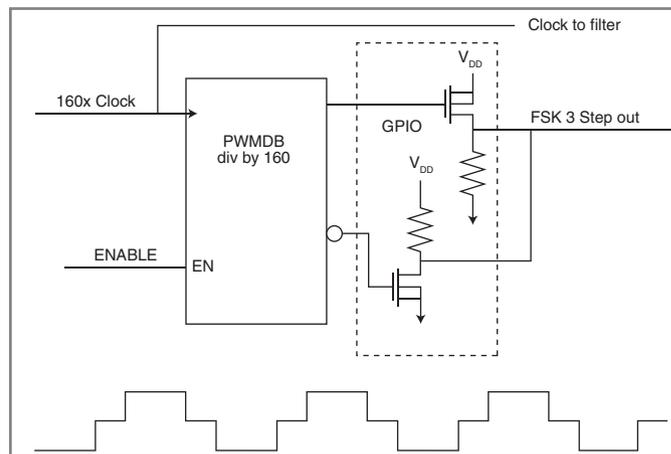


Figure 4—The PWMDB has two complementary outputs. When either side is active, the output is pulled to the respective rail. During the PWM deadband time, the outputs are resistively added to form a third level.

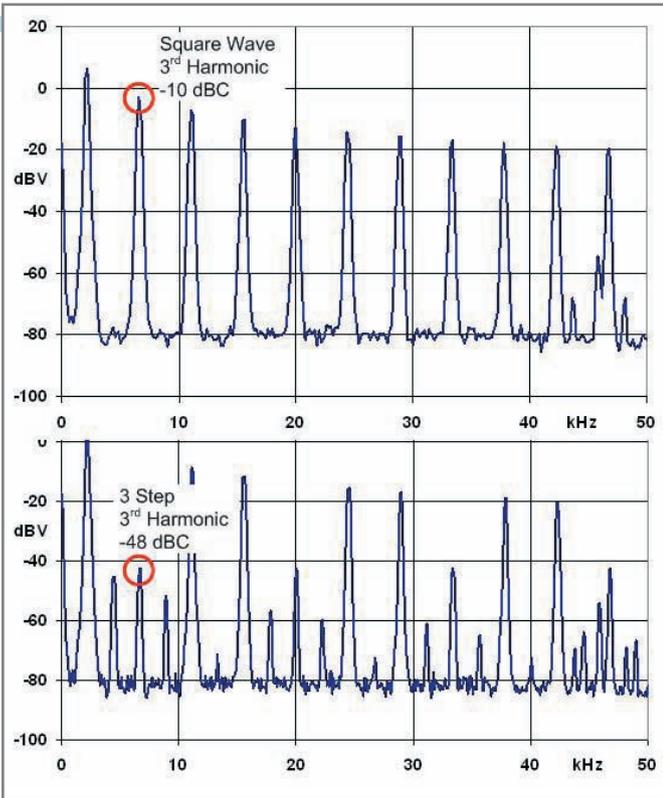


Figure 5—The spectrum of a square wave, shown in the upper plot, has components that fall off as $1/n$. A 38-dB reduction of the third harmonic can be seen with the three-step waveform.

attained when the lowest harmonics are eliminated.

FILTERING OUT HARMONICS

The third harmonic content of the waveform is less than 0.5%—a good start, but not up to the standard of the HART modem. The output of the three-step generator is filtered with a switched-capacitor band-pass filter. The switched-capacitor low-pass filter (LPF) is clocked at four times the sample rate. With the clock at 160 times the output frequency, the waveform is sampled at 40 times per sample. The three-step waveform uses the combined outputs

(XORed) of two blocks, but the clock to the filter can only come from a single block selected by the analog clock multiplexer. This signal is derived by connecting the composite clock output to a DigBuf, then using this buffer output as the clock source for the analog column. The completed digital block connection with digital signal generation and filters is shown in [Figure 6](#).

Design of the band-pass filter is enabled by using either the filter wizard in PSoC Designer or the design spreadsheet in PSoC Designer/Help/Documentation. The input to the filter is a 5.0-V_{pp} signal, which is large enough to saturate the filter output unless the filter is designed to attenuate the signal. Narrowband PSoC filters have high gain. This is a mathematical limitation based on the range of capacitor values available. While a filter bandwidth of 50 Hz would be “nice” in order to reduce harmonics, it is not achievable. At 2.2 kHz, a bandwidth of 800 Hz is achievable. This results in attenuation of the fifth harmonic by more than 20 dB (the third harmonic is already attenuated by the three-step waveform). Of course, the filter characteristic on the 1.2-kHz signal is the same as the 2.2-kHz signal because the filter performance is determined by the synchronized over-sample clock.

The filter output waveform is shown at 2.2 kHz in the time domain in [Figure 7](#) and frequency domain in [Figure 8](#). The harmonics are considerably reduced from the levels shown in [Figure 5](#). The highest harmonic is the fifth at 38.5 dB below carrier, or 1.2% distortion. All others are lower except the lines at 87.8 and 90.2 kHz; these are the sampling aliases of the filter with a level 32 dB below carrier. The sum of the low-order harmonics and the first pair of aliases results in a signal-to-noise ratio of 29 dB or 3.3% distortion.

For further reduction, the band-pass filter is followed by a 7.1-kHz LPF with a 1.0-MHz sample rate. The waveform looks even cleaner because the first filter’s sampling alias is significantly attenuated, but it is hard to evaluate distortion less than 1% with an oscilloscope.

The LPF has aliases at $1.0\text{ MHz} - 2.2\text{ kHz}$ and $1.0\text{ MHz} + 2.2\text{ kHz}$. Sampling aliases follow a $\sin(x)/x$ pattern. As the ratio of the over-sample increases, the aliases are closer to

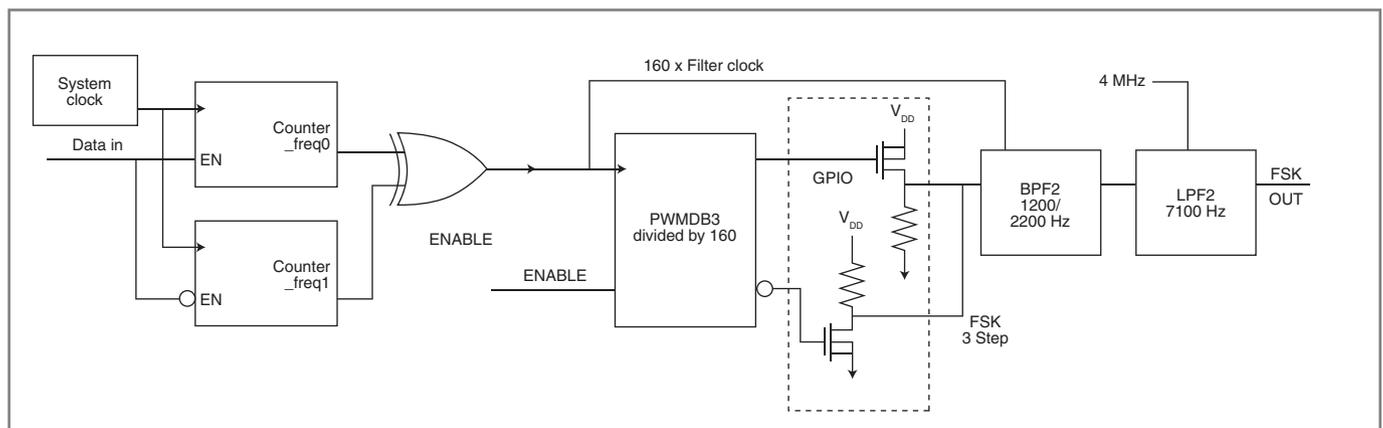


Figure 6—The complete block diagram shows the relatively simple digital structure combined with a tuned band-pass filter. The following low-pass filter removes the sampling aliases from the band-pass filter.

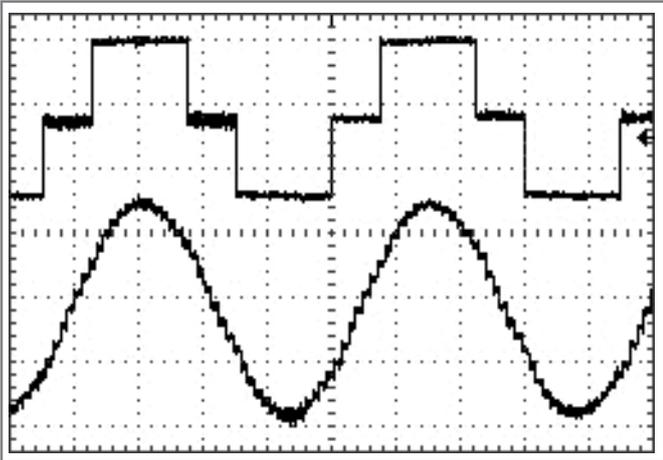


Figure 7—The three-step waveform is cleanly filtered to generate a decent sine wave, but still with significant steps at the sampling rate (40 times per cycle).

the null points of $\sin(x)/x$ and their levels drop. The LPF reduced the aliases from the band-pass filter. The spectrum scan from 0 to 100 kHz in **Figure 9** shows that the first filter's alias components are reduced to a negligible level.

The highest harmonic is the third at -44 dB. The total distortion is less than 0.8%. The highest alias is at 998 kHz at -52 dB. This is a very clean, easily generated waveform.

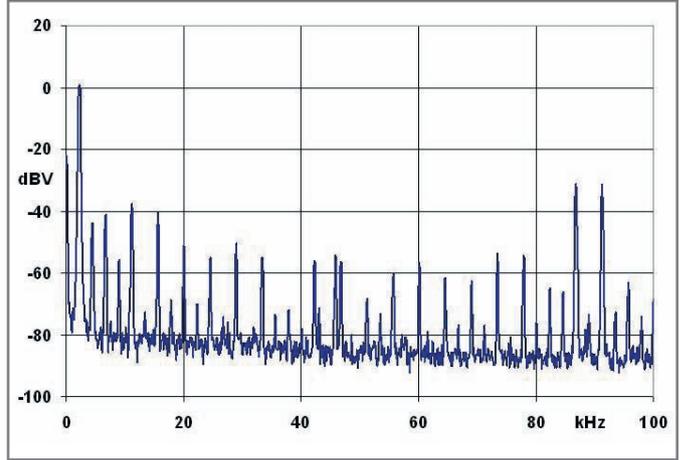


Figure 8—The band-pass filter spectrum shows that the dominant "harmonic" components are from the sample rate, not the harmonics of the intended fundamental.

The input data, three-step waveform, and the final output are shown in **Figure 10**. The steps in the output waveform are from the resolution of the digital scope, not the sampling in the FSK generator.

CONSTRUCTION OPTIONS

The FSK generator can be limited to a simple modulator driven by externally supplied data, or it can be part of a larger system built in the PSoC. A digital-only version with

THE ORIGINAL SINCE 1994

PCB-POOL®

Beta LAYOUT

Servicing your complete PCB prototype needs :

- **Low Cost - High Quality**
PCB Prototypes
- **Easy online Ordering**
- **Full DRC included**
- **Lead-times from 24 hrs**
- **Optional Chemical Tin finish**
no extra cost

Watch "ur" PCB®
Follow the production of your PCB in **REALTIME**

FREE LASER STENCIL WITH ALL PROTOTYPE PCB ORDERS

email : sales@pcb-pool.com
Toll Free USA : 1 877 390 8541
www.pcb-pool.com

Beta LAYOUT

CIRCUIT CELLAR®

back issues available as

Searchable Archives on CD-ROM

- CD-ROM #13 2008 Issues 210-221
- CD-ROM #12 2007 Issues 198-209
- CD-ROM #11 2006 Issues 186-197
- CD-ROM #10 2005 Issues 174-185

Order Online:
www.circuitcellar.com
or call 860.875.2199

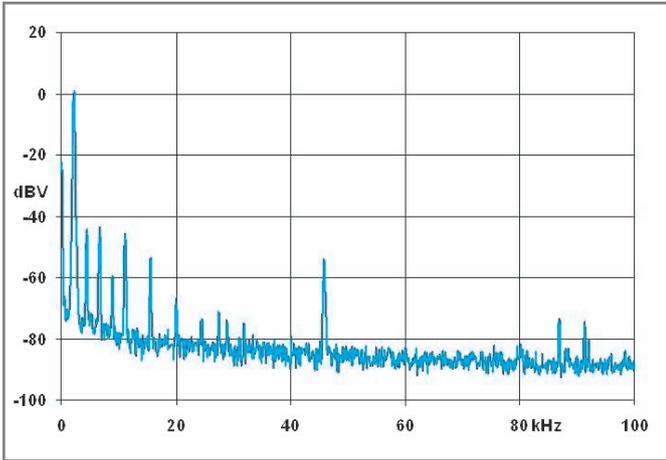


Figure 9—Adding a low-pass filter clocked at a much higher frequency results in the significant reduction of the band-pass filter’s sampling aliases.

square wave output can be constructed with three digital blocks. A three-step analog output version takes another digital block, replacing the PWM with a PWMDB. If a very clean waveform is needed, the three-step waveform and both filters are used. This takes a total of five digital blocks and four analog blocks. This fits best into a 28-pin or larger part, a Cypress Semiconductor CY8C27443 or CY8C29866. If a PGA is used for waveform input to the filter instead of Port 2.1, the design will fit into the eight-pin CY8C27143 or the 20-pin CY8C27243.

If all that is required is the FSK modulator, with modest allowed harmonics and no on-chip data generation, the design can fit easily into an eight-pin CY8C24123, which has four digital blocks and six analog blocks. The filter can be either the single band-pass, which will also help limit power requirements, or the combined band-pass and low-pass. In either case, the pin requirements are 1 for digital data input, 1 for square wave FSK data (instead of the three step waveform) to feed the filter, and 1 for FSK sine wave output.

The PSoC internal main oscillator is accurate to $\pm 2.5\%$ over the full range of voltage and temperature. While this is good enough for UART communications, the HART standard requires 1% frequency accuracy. This requirement is met using the PSoC’s 32.768-kHz external crystal oscillator with phase-locked internal main oscillator. Resource utilization boils down to the required distortion

	# Digital blocks	# Analog blocks	Current mA	% Distortion
Square wave	3	0	0.1	46.7
Square wave +1 Filter	3	2	1.3	5.1
3 Step	4	0	0.1	39.5
3 Step +1 Filter	5	2	1.3	4
3 Step + 2 Filter	5	4	2.5	0.8

Table 1—Increasing the digital waveform complexity and adding filter sections yields steadily lower distortion.

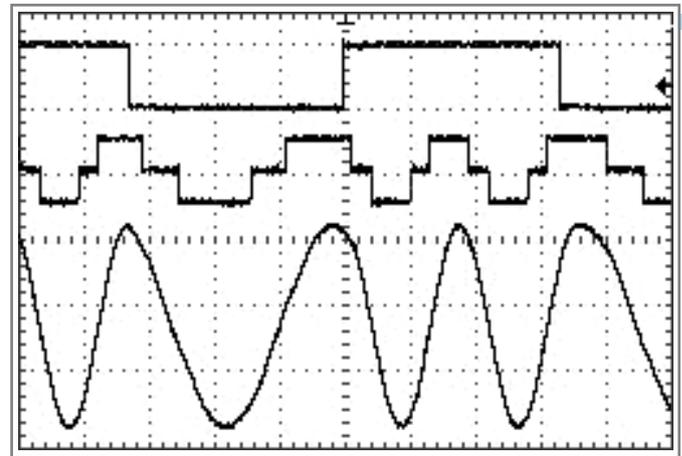


Figure 10—The final filtered output of the FSK generator driven by alternating 1 and 0 data shows a clean waveform and smooth phase transitions when the data changes. The granularity of the sine wave is a function of the scope resolution, not the generated signal.

level, with options summarized in [Table 1](#).

SOFTWARE

The entire code to operate the FSK modulator is listed below in C:

```
Counter_freq0_Start();
Counter_freq1_Start();
PWMDB8_Start();
BPF2_Start(MEDPOWER);
LPF2_Start(MEDPOWER);
```

Starting and stopping the modulator output is done by controlling the PWMDB8 Enable or Kill input. This design has a significant advantage over direct digital synthesis techniques using a DAC in that zero run-time code execution is required. When implemented as part of a complete system, this leaves the processor available to do what processors do best, calculate and control.

ASSOCIATED PROJECT

The associated project is built in a CY8C29866. Data is simulated by a PWM16. This can be replaced by a UART and clock generator, or the modulator can be fed digital data through a global input.

This signal-generation technique is useful in many applications, not just FSK transmission. The idea is to reduce the harmonic content prior to filtering and then take advantage of the PSoC’s switched-capacitor filters. 

Dennis Seguire (BSEE and MSEE, UC Berkeley) is an applications engineer at Cypress Semiconductor. He has broad experience in system and analog circuit design, and he holds eight U.S. patents. You may contact him at seg@cypress.com.

SOURCE

CY8C27xxx and CY8C29xxx PSoCs

Cypress Semiconductor Corp. | www.cypress.com



Power Pitcher

Wireless Power on a Microelectronic Scale

Nikola Tesla's century-old dream of wireless power delivery is coming true, at least for chips. This month Tom looks at a system from PowerCast that delivers power via radio waves. Combine it with low-power silicon and you can cut the wires, and ditch the wall warts, once and for all.

It's no secret that wireless gadgets (I call 'em "wadgets") are all the rage. Cell and cordless phones, Blackberries, Wi-Fi, Bluetooth, and a zillion flavors of IEEE 802.15.4 (e.g., ZigBee, 6LowPAN, and proprietary) all ride the airwaves.

More power to them. Literally. Virtually every wadget must spend at least some time connected to a wire (e.g., wall wart, car charger) to get its power fix. Sure there are some applications that are so simple and intermittent (e.g., garage door opener, keyless remote) that they just run on batteries period. But despite advances in battery technology, truly wireless wadgets are the exception rather than the rule.

Energy harvesting (EH) to the rescue. Thanks largely to the Moore-for-less march of silicon, many embedded wireless apps are able to skim the energy they need from their ambient environment. Solar (more generally, photovoltaic) conversion

is the most popular, efficient, and well-understood approach. But now EH is really getting creative with virtually any and every joule, whatever the source (e.g., vibration, thermal), fair game for the taking.

But there's a problem. Just like real crops, sometimes the harvest may not come in. For example, a solar harvester is subject to the whims of the weather (i.e., clouds). One solution is to over-engineer the design (e.g., larger panel and battery) to account, hopefully, for the worst case.

What if there were a way to have the best of both worlds (i.e., the convenience of wireless combined with the predictability and reliability of wires)? Power would be supplied on your terms—when and where you want it—and not depend on the vagaries of nature or the application.

Wireless power delivery isn't a new concept. Fully 100 years ago, Nikola Tesla used his parlor trick of illuminating a light bulb in his hand as the basis for an elaborate vision of wireless



Photo 1—A century ago, Nikola Tesla had science fiction-like visions of wireless power delivery that PowerCast delivers as science fact today. (Source: The Tesla Memorial Society of New York, www.teslasociety.com)

power transmission on a grand scale (see [Photo 1](#)).

Coming down to earth, consider your electric toothbrush. Obviously, mixing wired contacts and water isn't going to cut it. Instead, we find a transformer-like arrangement with the AC-powered primary cradle transferring energy to the secondary in the toothbrush across a hermetically sealed gap.

PowerCast is an outfit that brings Tesla's vision of a wireless power utility to life on microelectronic scale. As their company name implies, they broadcast power using radio waves. So how much power can you extract from RF? Bell Labs radio scientist H.T. Friis came up

Antenna Type	Gain	Operation
Dipole	1.5	Omnidirectional
Sleeve dipole	1.4	Omnidirectional
Short loaded dipole	1.2	Omnidirectional
Flexible dipole	1.5	Omnidirectional
Patch	3.9	Directional (from surface)
Yagi	6	Directional (from edge)

Table 1—When it comes to RF, the choice of antenna may not be everything, but it's a big deal. Directionality is a key factor, acting like a magnifying glass to focus the available energy.

the distance from the transmitter.

The black magic of RF is found in the antenna gain factors, which depends on a variety of factors including both the type of antenna and implementation issues. You don't have to be a rocket scientist to

wire) is omnidirectional, radiating in a circular donut pattern. Other antennas—such as Yagi, patch, or dish—exhibit various degrees and patterns of directionality. They offer more gain than a dipole, presuming of course that the transmitter and

	P1100	P1200	P2100
Application	Rechargeable alkaline	Rechargeable lithium	Capacitor charge and fire
Recommended operating input	0 to 20 dBm	0 to 20 dBm	-10 to 10 dBm
Voltage output	3.3 V Maximum	4.2 V Maximum	2.8 to 5.0 V Configurable
Power output	<= 100 mW continuous	<= 100 mW continuous	262.5 mW Peak

Table 2—The PowerCast harvester lineup consists of three parts targeting the most likely applications. The P1100 and P1200 feature output voltage limiting for alkaline and lithium battery charging, respectively. The P2100 is a bit different, designed to charge a supercap and then fire a configurable voltage output.

with the answer some 60 years ago. Herewith, the Friis Transmission Equation:

$$\frac{P_R}{P_T} = G_T G_R \left(\frac{\lambda}{4\pi R} \right)^2$$

On the surface, the equation is quite easy to understand. It simply states that the power received (P_R) is a fraction of the power transmitted (P_T) that's determined by three factors: the wavelength (λ), distance (R), and the effectiveness (i.e., gain) of the transmitting (G_T) and receiving (G_R) antennas.

As you can see from the equation, received power increases for a longer wavelength (i.e., lower frequency). That explains why, for example, a 900-MHz radio works better (i.e., overcomes interference and obstacles) than a 2.4-GHz radio with the same power output. And there's the familiar inverse square law in the denominator showing received power declines with the square of

imagine there's a difference between a trace on a PCB and a satellite dish. PowerCast offers a selection encompassing everything from a simple dipole to a more sophisticated Yagi and patch antenna to match the needs of different applications (see [Table 1](#)).

One key antenna factor is directionality. A simple dipole (piece of

receiver are properly aimed at each other.

Directionality (and a similar phenomenon, polarization) directly affects the application in terms of mobility. A fixed (non-mobile) installation can take advantage of directional antenna for maximum gain and efficiency. But if the receivers move around, an omnidirectional

66 The black magic of RF is found in the antenna gain factors, which depends on a variety of factors including both the type of antenna and implementation issues. You don't have to be a rocket scientist to imagine there's a difference between a trace on a PCB and a satellite dish.

99

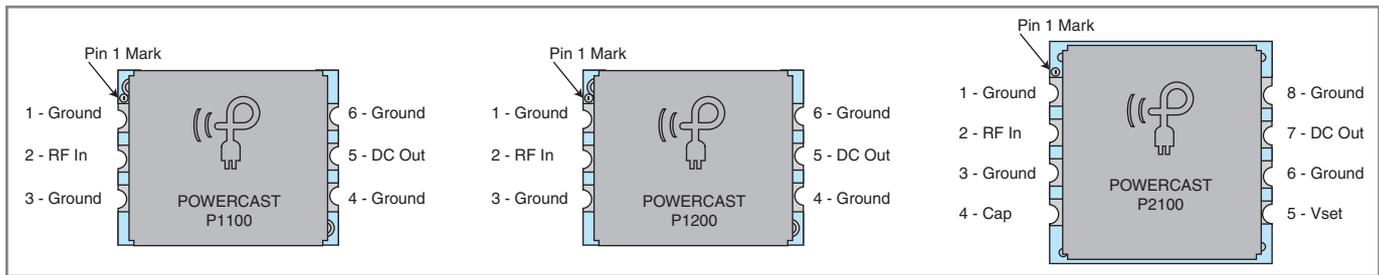


Figure 1—Designing-in the PowerCast harvesters is really simple: hang an antenna (50Ω) on one side and the load on the other.

antenna is the call.

Implementation issues include things like cable and connector losses and impedance matching. Maximum power transfer occurs when the transmitter and antenna (including intervening connectors and cable) have the same impedance. A mismatch will cause a portion of the energy to be reflected back to the transmitter rather than sent over the air. Besides loss of efficiency, in the worst case (e.g., if you forgot to attach an antenna), the amount of reflected energy might actually damage the transmitter. Thus, the PowerCast transmitters include a so-called Standing Wave Ratio (SWR—i.e., a measure of how much energy is reflected) shutdown that automatically turns off the transmitter before damage occurs. Along the same lines, there's a thermal shutdown feature.

The PowerCast lineup consists of a radio/power-generating station (i.e., transmitter) and three receivers (see Table 2). The transmitter operates in the 900-MHz band with the exact frequency selectable (by pressing a button on the unit) in 2-MHz increments between 903 and 927 MHz. The frequency-select capability allows the use of multiple transmitters at different frequencies to avoid the wave cancellation that otherwise might occur.

The receivers are tailored for slightly different applications (see Figure 1). The P1100 and P1200 offer fixed output voltage limiting that makes them a good choice for designs using rechargeable batteries (alkaline and lithium, respectively). Design-in is as simple as it can be, just connect the antenna on one side and the load on the other.

The P2100 is a bit different. It features a selectable output voltage between 2.8 and 5.0 V (set with external resistors connected to the V_{SET} pin) and is especially tailored for capacitor-based “charge and fire” designs (see Photo 2). The P2100 charges the capacitor (attached to the CAP pin) to 1.2 V and then turns on the DCOUT (stepped up to the selected output voltage) to the load. When the capacitor voltage falls to 1.05 V, the P2100 automatically turns off the DC output and begins another recharge cycle. Per Friis, Figure 2 shows how the recharge time varies depending on distance and the antenna gain.

RF power output of the transmitter is specified as 3 W, while a P1100 or P1200 can deliver up to 100 mW at the receiving end. (With a supercap to draw from, the P2100 can “fire” significantly higher peak power, up to 262.5 mW.) Sure it's lossy and nowhere near the efficiency of a wire,

but there's definitely enough power on tap to do meaningful things. The point is that PowerCast is an option when a wire isn't.

FREE AT LAST

Looking at the big picture, one can imagine a variety of wireless power application scenarios. As mentioned above, options include powering a device directly if it can get by on 100 mW, or recharging a battery or supercap for higher-power devices. The transmitter could be powered at all times or on a scheduled basis. Sampling a bunch of sensors could be as simple as turning on their power (i.e., the transmitter(s)) periodically. Alternatively, a battery-powered sensor, detecting its battery is running low, could radio a request to turn on the transmitter (i.e., via a radio receiver controlling the transmitter's power supply).

Like a cell phone network, multiple transmitters can be used to provide wide area coverage within which any number of receivers is free to move about (see Figure 3). Multiple transmitters also provide a measure of redundancy (i.e., the system can be designed to deliver an adequate level of power, even if a single transmitter should fail or otherwise be compromised—for example, in the presence of moving obstacles).

SAFE AND (IN)SANE?

Maybe you've seen the YouTube video purporting to show cell phones popping popcorn or cooking eggs? Or maybe you've seen the ads for RF-armored underwear to protect your family jewels. For those of you who believe

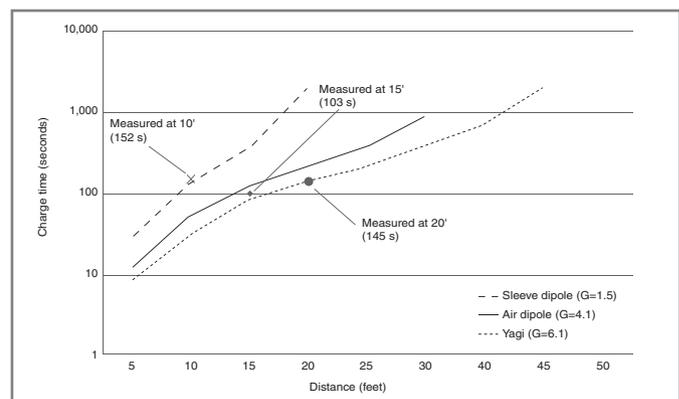


Figure 2—This graph showing the capacitor recharge time for a P2100 demonstrates the influence of key factors (i.e., antenna gain and distance) in the Friis equation.

the future of journalism is “information that wants to be free,” all I can say is you’ll get what you pay for.

The fact is, a cell phone’s transmit power is on the order of 1 W, with the power dynamically adjusted to compensate for obstacles (e.g., indoor use) and distance from the base station. So, if you’re worried about your 1-W cell phone, shouldn’t the 3-W PowerCast transmitter raise concern? But remember the inverse square law. The cell phone is an inch from your brain. I wouldn’t tape a PowerCast transmitter to my head; but otherwise, I wouldn’t get too worried.

Perusing the PowerCast website and documentation, there’s little on the subject of safety, so presumably no news is good news. The only precaution I could find is as follows: “FCC radio frequency safety guidelines require the device to be operated with the external antenna at least 20 cm (8 inches) from the user’s body. The owner of the device must ensure this condition is met even if the device is operated by a third party or the device is incorporated into another product. The owner of the device must communicate this information to any subsequent purchaser. Use and maintain the device only as specified in this manual.”^[1]

Enough of the fine print, time to get hands on. PowerCast sent me a selection of components to play with, including a transmitter, a selection of antennas, and a plug-in LED module (see Photo 3). The antenna modules each have a built-in P1100 harvester (two P1100s on the patch antenna) and a header that the LED board plugs into. It’s a simple setup, but nevertheless able to deliver useful insight into real-world application issues.

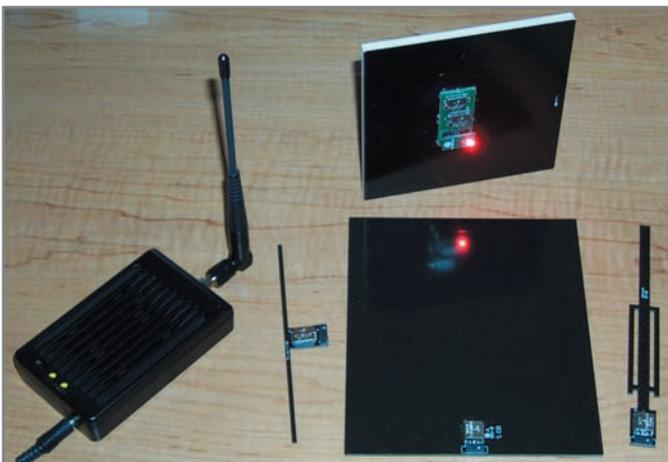


Photo 3—PowerCast gear shown here includes the transmitter, a selection of antenna modules with built-in harvesters, and a plug-in LED board.



Photo 2—PowerCast offers an evaluation board that uses a P2100 harvester to “charge and fire” the 0.22F supercap located on the lower right side of the board.

overstatement as there were clearly observable “dead zones” (i.e., positions and orientations where power delivery was somewhat attenuated). That’s not a knock on PowerCast, just RF reality.

This has implications for mobile applications—namely, you wouldn’t be able to presume continuous (i.e., uninterrupted) power in a mobile application without carefully considering limits on the orientation. Chances are, you’d need to incorporate back-up power (e.g., battery or supercap) or, alternatively, a multi-transmitter array arranged to cover all orientations. Of course, the same caveat applies in an environment with transient obstacles (e.g., people walking around).

Another interesting phenomenon I noticed was that power delivery seemed non-monotonic relative to range. For a given orientation, you’d expect the LED would brighten/dim directly with changes in distance. But I noticed as I increased the range the LED would dim and then brighten a bit before becoming yet dimmer. This is likely explained by multipath phenomenon. Since this transmitter was equipped with an omnidirectional antenna, at a given location reflections can partially reinforce



Figure 3—Like a cell phone network, multiple transmitters can help ensure full coverage, a must in situations where receivers may move around or be blocked by transient obstacles, such as people wandering around.

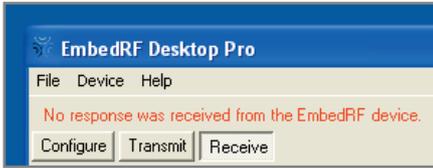


Photo 4—Needless to say, the PowerCast transmitter can overwhelm wireless radios on the same frequency. Recognizing the potential problem, the PowerCast transmitter frequency can be adjusted (via a push button) to allow it and the wireless radio to co-exist.

or attenuate energy delivery. A contributing factor may be that, according to the datasheet, the harvester conversion efficiency varies between 50% and 70% and in some cases is higher at a lower level of received energy.

What about interference? I dug through my pile of gear and found a 900-MHz embedded wireless setup, the EmbedRF modules I covered in “Here Come the Widgets” earlier this year (*Circuit Cellar* 223). The bad, though not unexpected, news is that the PowerCast transmitter completely blitzed the radio transmission (see **Photo 4**). The good news is that simply pushing the button on the PowerCast transmitter (i.e., shifting its frequency by 2 MHz) solved the problem. The PowerCast documentation notes that many embedded wireless gadgets (but apparently not the EmbedRF modules) have automatic frequency hopping built-in that might prevent problems. That got me thinking it would be nice if the PowerCast transmitter had a computer interface of some kind (e.g., UART, SPI, USB) that, with the addition of external intelligence, would allow changing frequency, monitoring the alarm conditions (SWR, thermal), and so on. According to the PowerCast website, they’re open to OEM and customization proposals, so I imagine something could be worked out if a deal made sense.

Speaking of alarm conditions, I noticed the PowerCast transmitter automatically turned off when I touched the antenna (see **Photo 5**), presumably causing an impedance mismatch and enough reflected energy to trigger the SWR shutdown feature. Ditto for removing the antenna

completely, a welcome safety feature for the absent minded.

POWER TO THE PEOPLE

Just for kicks, I fired up my cell phone, and sure enough, in close proximity it was able to light up the PowerCast receiver LED too. Will we see a time when you use your cell phone to wirelessly recharge your pacemaker? Or how about charging your laptop while you sip a latte at the local Wi-Fi hotspot?

I’ll leave it as an exercise for anybody who has more free time than

common sense to try cooking dinner with 3 W. Sounds like a good way to lose some weight.

Give PowerCast credit for thinking outside the box, and the wire. Beyond the technology itself, they were smart enough to recognize the convergence of trends (i.e., low-power silicon, embedded wireless, energy harvesting, and the green revolution) that mean the time is right for wireless power delivery.

If Tesla were still alive, he would no doubt be impressed and proud, deservedly, to say he told you so. ☞

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.

REFERENCE

[1] PowerCast Corp., “Lifetime Power Evaluation and Development Kit,” Rev. D, 2009, www.powercastco.com/PDF/UserManualWeb.pdf.

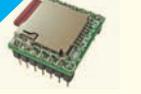
SOURCE

P1100, P1200, and P2100 Powerharvester modules
PowerCast Corp. | www.powercastco.com



Photo 5—There’s more to antenna gain than just the type of antenna. A faulty implementation (e.g., impedance mismatch) can cause a loss of efficiency or even damage the transmitter. Here the red LED on the transmitter indicates it has shut down, having detected something is amiss.



<p>Color LCD Scopes</p> <p>Best Seller</p>  <p>2-ch + trigger standalone USB bench scope. \$315 / \$499</p>	<p>2-ch 1GSa/s Scopes</p> <p>NEW! RT60L</p>  <p>2-ch 1GSa/s (25GSa/s equiv.) 50/100 MHz scope. \$595 / \$795</p>	<p>Scope + Analyzer</p> <p>NEW! RT60L</p>  <p>25MHz 2-ch /16 logic scope and logic analyzer. \$1195</p>	<p>Amazing 7 in 1 Scope! \$180</p> <p>CircuitGear CGR-101™ is a unique new, low-cost PC-based instrument which provides the features of seven devices in one USB-powered compact box: 2-ch 10-bit 20MS/sec 2MHz oscilloscope, 2-ch spectrum-analyzer, 3MHz 8-bit arbitrary-waveform/standard-function generator with 8 digital I/O lines. It also functions as a Network Analyzer, a Noise Generator and a PWM Output source. What's more – its' open-source software runs with Windows, Linux and Mac OS's!</p> <p>Only \$180!</p>  <p>"I really like this scope adapter - it's meant for teaching electronic experiments but it's ideal for engineers too."</p> <p>Alan Lowne Saelig CEO</p>	
<p>Handheld Scopes</p>  <p>20MHz / 60MHz rugged handheld USB 2-ch scope. \$593 / \$699</p>	<p>Pen Scopes</p>  <p>10/25MHz USB powered scope-in-a-probe! Up to 100MS/s. \$234 / \$369</p>	<p>XP Emb Touchpanel</p>  <p>Touch-input 10.2" LCD 12V-powered Windows PC. LX800/512MB/4GB/Ethernet/3xUSB/SD.</p>	<p>FREE COFFEE</p> <p>Call 1-888-772-3544 to get a free Starbucks Card with your >\$50 order!</p> <p>While supplies last - not available with any other offers</p> 	
<p>Low-Cost Scopes</p>  <p>2-ch 40/100/200MS/s 8-bit scope range with 5/10/25MHz. \$246 +</p>	<p>Scope/Logger</p>  <p>2-ch DSO 16-bit DSO, FFT, VM, logic analyzer, standalone + 24 I/O.</p>	<p>Mixed-Signal Scopes</p>  <p>100MHz Scope, + Spectrum/Logic Analyzer and Signal Generator. \$1259+</p>	<p>microCAM</p>  <p>Compact compressed-serial output camera module for any host system.</p>	<p>Multichannel DAQ</p>  <p>12/16 input 1ks/s 10/12-bit PC-connected voltage logger.</p>
<p>USB Bus Analyzers</p> <p>Best Value</p>  <p>Packet-Master™ - USB 1.1/2.0 analyzers and generators. \$699 +</p>	<p>16-Ch Logic Analyzer</p>  <p>Intuitive full-featured 16-ch 4MB 200MHz sampling memory. \$279</p>	<p>SPI Bus Analyzer</p>  <p>Protocol exerciser/analyzer for standard SPI and non-standard 4-wire and 3-wire serial protocol interfaces up to 50 Mbps.</p>	<p>EMC Spectrum Analyzer</p>  <p>RF & EMF Spectrum Analyzer 1Hz to 7GHz for WiFi, mikes, etc.</p>	<p>EMC Spectrum Analyzer</p>  <p>Handheld Palm PC-based 2.7GHz Spectrum Analyzer.</p>
<p>Waveform Generator</p>  <p>USB2.0 speed 16-bit digital pattern or arbitrary waveform generator.</p>	<p>I2C Xpress</p>  <p>Versatile USB 2.0 I2C protocol exerciser and analyzer.</p>	<p>Automotive Testing</p>  <p>Kits turn your PC into vehicle-electrics diagnostic tool.</p>	<p>CANminiBOX</p>  <p>Embedded controller series: 2 x CANbus, Ethernet, USB2.0, CF.</p>	<p>RF Generator</p>  <p>High-res, extremely low-noise, portable 6GHz RF generator.</p>
<p>Wireless Data Loggers</p>  <p>Log and display temp, hum, volt, event-time or pulse-counting data</p>	<p>Multiparameter Loggers</p>  <p>Mini-logger with built-in temp/hum/pressure/3-axis accel sensors.</p>	<p>USB Loggers</p>  <p>Standalone USB temp / hum / volt / current loop data logger. \$49+</p>	<p>Electronic DC Load</p>  <p>Const. current, resistance, conductance, voltage & power modes</p>	<p>60/100/120MHz AWG</p> <p>NEW! RT60L</p>  <p>60/100/120MHz USB 14-bit ARB with USB RS-232, LAN/GPIB.</p>
<p>USB to I2C</p>  <p>"Drop-in" solution connects PC to I2C/SMBUS + 32 I/O lines. \$89</p>	<p>FTDI USB ICs</p> <p>Lowest Prices</p>  <p>Popular UART and FIFO chips. Upgrade Legacy designs to USB.</p>	<p>CAN-USB</p>  <p>Intelligent CAN connection from PC's USB port. \$299</p>	<p>Serial-Ethernet Cable</p>  <p>Network serial product easily without a PC using this 28" cable. \$89</p>	<p>Lorlin Switches</p>  <p>Fantastic array of stock and custom switching devices.</p>
<p>Keyboard Simulator</p>  <p>USB board adds 55 I/O and 5 x 10-bit AD inputs, 1 x 10-bit analog O/P.</p>	<p>Instant Ethernet</p>  <p>No OS needed. TCP/IP offload, ICs improve system performance.</p>	<p>Ethernet-I/O</p>  <p>UDP/IP-controlled 24 digital I/O board 3 x 8-bit TTL ports.</p>	<p>FPGA Systems</p>  <p>Ready-to-go out-of-the-box FPGA/DSP designs for beginners and experts!</p>	<p>Wireless Solutions</p>  <p>Analog input, bluetooth wireless modules 433/868/915MHz.</p>
<p>.NET Board</p>  <p>Small (2.2" x 2.2") lowest cost .NET Micro Framework dev system.</p>	<p>Easy OLED Display</p>  <p>Compact, economical smart OLED with graphics drive from USB or RS232.</p>	<p>RF Modules</p>  <p>Simultaneously transmit composite video and stereo audio signals.</p>	<p>RS232 to 422/485</p>  <p>9p-9p or 25p-25p self-pwrd, isolated RS232-RS422/485</p>	<p>1/2/4/8/16 x RS232</p>  <p>Add 1-16 COMports via your PC's USB Port easily.</p>
<p>USB-Serial</p>  <p>A complete CP2102 USB-serial converter in a DB9 shell. \$26</p>		<p>Sound Module</p> <p>NEW!</p>  <p>14-pin module plays back pre-stored audio files from microSD card</p>		



Device Management

Monitor Networked Devices Via SNMP

The simple network management protocol (SNMP) enables you to monitor a network's activity and stock data units in a management information base (MIB). Today, many embedded microcontroller manufacturers provide SNMP support in their TCP/stack offerings. You can use this technology to build an MCU-based managed device that monitors multiple networked modules and logs critical data.

Most engineers look through a new part's datasheet before attempting to use it. This might be to simply identify the part's connections or to fully understand its working parameters. In general, a datasheet provides specific answers that cannot be gleaned from just looking at a proverbial, albeit tiny, black box. You wouldn't think of connecting a peripheral to your micro without some knowledge of how it will integrate into the design. With the pinout, you could connect all of the signals using the proper communication bus structure. You might even be able figure out if it is an input or an output device. But without knowledge of a device's inner-workings, trying to make full use of it is futile.

For years, we used serial and parallel ports on our PCs without any real knowledge of what was being plugged in. Originally, about all that was available was a printing device and driver support was unnecessary. But once manufacturers understood just how useful these ports could be, peripheral I/O devices started popping out everywhere: document scanners, barcode readers, weight scales, mice, modems, and more. One day someone asked, "Wouldn't it be nice to have some kind of universal port that you could plug anything into?" That question led to USB and FireWire technologies. USB and FireWire attempted to become everything to everyone. Without knowing what was being plugged in,

how could an application make use of a device like this? The key to this problem was to have the peripheral identify itself when plugged in. And even though this sometimes requires a special driver to be installed, adding a device is a relatively simple task (for the most part).

Like USB and FireWire, the Internet is just another of the many communication mediums available today. The Internet interconnects networks throughout the world. It uses standardized protocols to allow private/public, academic, business, and government networks to communicate with one another over various technologies. The copper, fiber optic, or wireless technologies used today offer something not available from USB or FireWire—and that is distance. For years, we've been able to pick up a telephone and connect to anyone else on the planet with telephone service. To do this, you need to "dial" a special number that the telephone network uses to route a connection to the appropriate telephone. The Internet uses an IP address to do the same thing and connect your computer with another on the Internet. While you may actually use a "domain name," this is converted into a specific IPv4 address. This dotted notation of 4 bytes (x.x.x.x) soon will be eclipsed by a 16-byte notation IPv6 (x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x). But that's another story.

You might imagine that Internet traffic

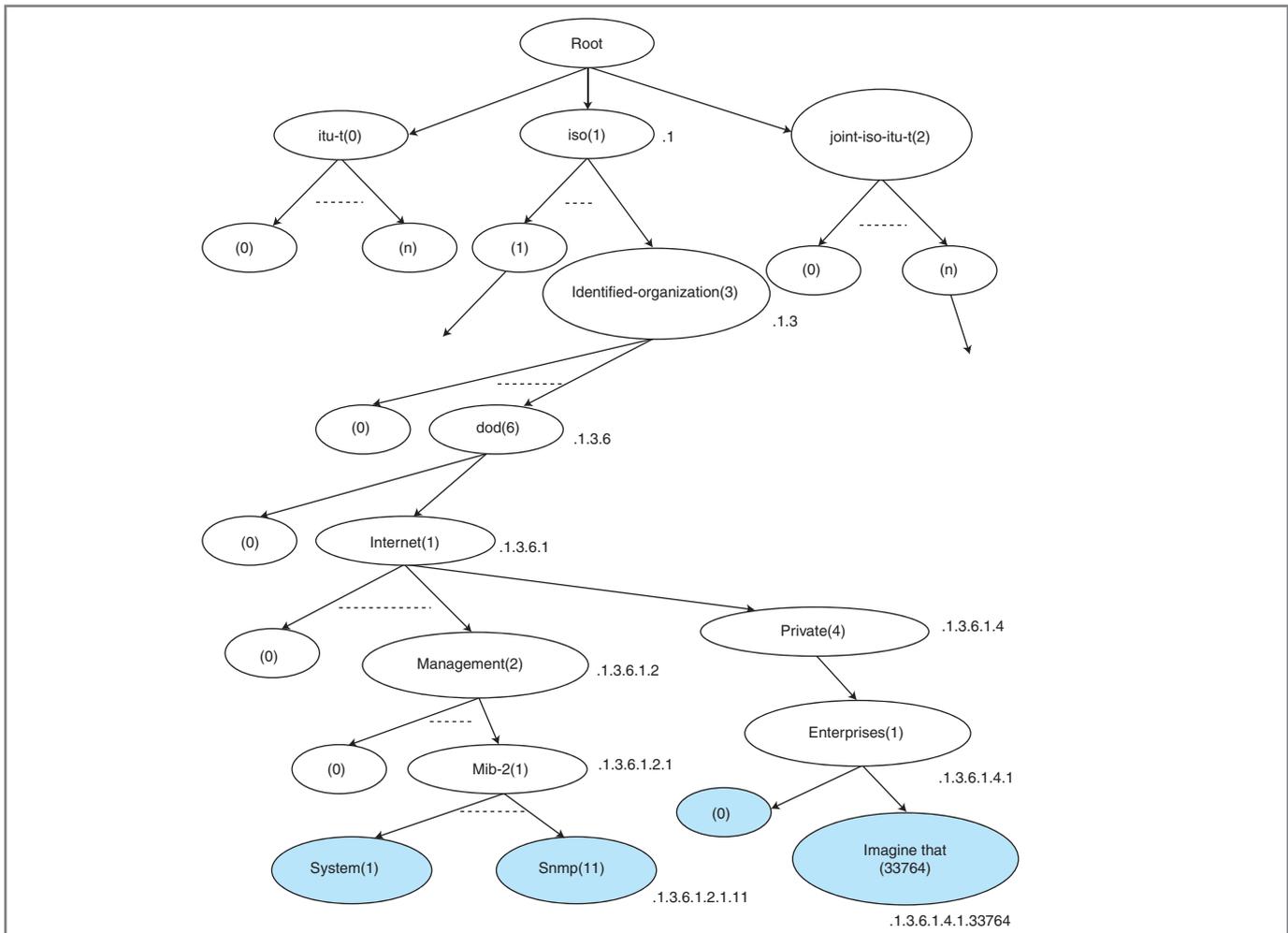


Figure 1—You can follow the root (or structure) of the management information base (MIB) down to the SNMP node using the dotted integer path (.1.3.6.1.2.1.11). Note that SNMP(11) is part of the MIB for the simple network management protocol (SNMP) found in RFC3418.

consists of packets of information carrying e-mails, photos, and media files between users—and that would be true. However, there is other traffic that doesn't have anything to do with personal communications: it carries control and status information. This information doesn't necessarily need to be presented in a viewable fashion; it can be as simple as a single prearranged value.

MANAGEMENT INFO BASE

The Open Systems Interconnection (OSI) is an attempt to standardize networking by the International Organization for Standardization (ISO) and the International Telecommunication Union (IUT). The OSI consists of a Management Information Base (MIB) that is used to identify all devices in a communications network.

The structure is made up of data objects or object identifiers (OIs) represented by a sequence of integers, which traverse a global tree identifying a particular node. The tree begins with a main root connected to a number of labeled sub-roots or nodes. Each node may, in turn, have sub-roots (or children) of its own, which are labeled. This process may continue to an arbitrary level of depth. **Figure 1** shows a partial view of the root structure. For additional information, refer to Keith McCloghrie et al's "Structure of Management Information Version 2" (SMIv2).

Most of the OIs are predefined as labeled pairings of a brief textual description and an integer in the form of [text(#)]. The integer is a shorthand representation of the node (OI). A dotted sequence of these integers identifies a specific path.

The root node itself is unlabeled, but has three children directly under it: one node is administered by the ISO, with label "iso(1)"; another is administered by the International Telegraph and Telephone Consultative Committee (CCITT), with a label "ccitt(0)"; and the third is jointly administered by the ISO and the CCITT, "joint-iso-ccitt(2)." Along the MIB tree's ISO path, you can find the Department of Defense (DOD) under the "other-organization" node. It has authority over a subtree, which includes the Internet.

In Marshall Rose and Keith McCloghrie's "Structure and Identification of Management Information for TCP/IP-Based Internets," the subtree of the Internet is defined as having four nodes: Directory, Management, Private, and Experimental. The Management sub-tree defines objects

used in the management of the Internet standard—this path is the way to the simple network management protocol (SNMP, .1.3.6.1.2.1.11)—while the Private subtree provides nodes to companies requesting a private enterprise number (PEN) from the Internet assigned numbers authority (IANA) under its enterprises node. For instance, my PEN is 33746. With this assignment, my tree path would be “.1.3.6.1.4.1.33746.” This is significant to me because I can define the sub-tree below this node. More on this shortly.

SNMP

The Hypertext Transfer Protocol (HTTP) is the application layer protocol you are probably most familiar with because you use it every day for interacting with others through graphics, audio, video, and plain text. The Simple Network Management Protocol (SNMP) is similar to HTTP, but it deals directly with network-attached devices and doesn't need the fluff required for human readability. The SNMP protocol includes an application layer (the how), a database schema (the where), and a set of data objects (the what). The SNMP exposes data objects, which describe the device configuration and even status and control values. These objects can then be queried (and even set) by managing applications.

An SNMP-managed network consists of at least one managed device, an agent, and a Network Management System (NMS). A managed device has its own pool of data related to its function, but it makes this information available to NMS using an SNMP agent. An agent is a network-management software module that resides in the managed device.

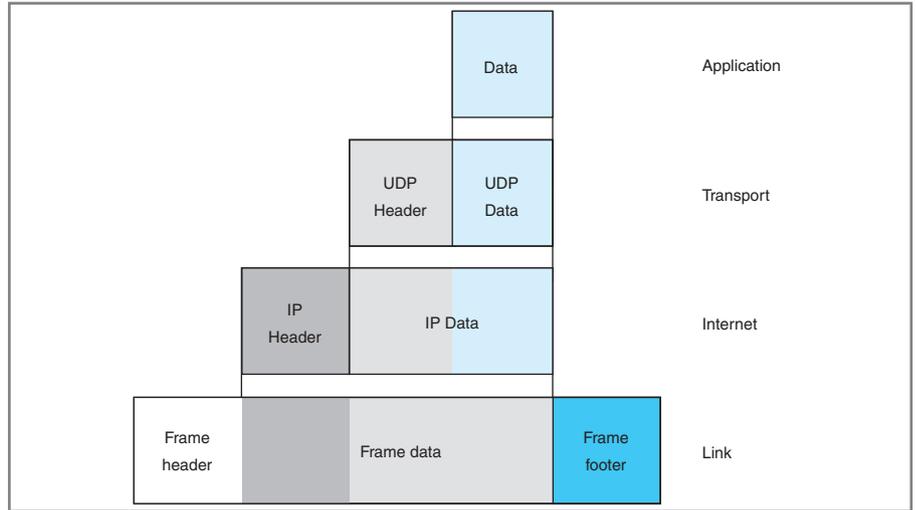


Figure 2—The SNMP message (application data) is typically wrapped in a UDP packet (transport method), which is housed within an IP packet (for use over the Internet).

The agent has local knowledge of the data and is responsible for translating it to and from SNMP message packets.

An NMS is an application that monitors and controls managed devices. These systems provide the bulk of the processing and memory resources required for network management. One or more of them may exist on any managed network.

An SNMP message (see [Figure 2](#)) is typically wrapped in a TCP or UDP packet, which is wrapped in an IP packet (see [Figure 3](#)). There are only seven message types in the present standard SNMPv2. The Get, Getnext, and Getbulk messages request data. The Set message writes data from the management application. The Response message is returned by the managed device. A Trap message is used by a managed device to signal an event or condition (there is no response). An Inform message is similar to a trap, but it results in a Response being returned. While SNMP gives you a way to pass data, how do

you know where to find this data?

DATASHEET

The key to the location of the data is the datasheet—or in this case, the MIB's OIs. The standards group has predefined data objects (OIs) for a number of standard devices, such as a modem, a cable device, an uninterruptible power supply (UPS), a printer, and so on.

Let's take a closer look at the standard MIB for a UPS. The standard MIB for a UPS uses OI .1.3.6.2.1.33, “upsMIB” (covered in RFC1628). [Table 1](#) includes the first OI in the sub-roots of this MIB. The first object in this tree is “[upsIdentManufacturer 1]” at sub-root node .1.1.1 of the “upsMIB.” Each object is described using the Abstract Syntax Notation One (ANS.1). This is a standard notation structure that ensures data will be presented in a machine-independent way. The Packed Encoding Rules (PER) that enable this data to be compressed and transmitted is complex and

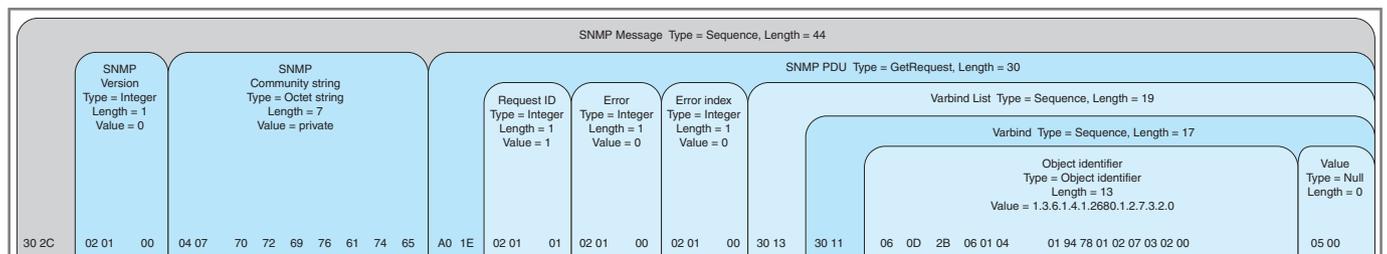


Figure 3—The SNMP data is rolled into the IP packet in [Figure 2](#).

Pick a Chip. Any Chip.

Find a Solution to your next Embedded Challenge.
Do the Research you should, but never had time for.

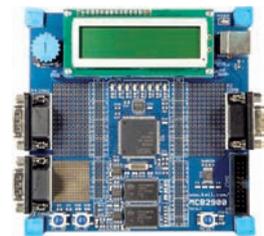
Embedded Developer's intuitive research engine helps you speed your chip evaluation time. You don't have to know the manufacturer, chip family or part number--just select the features you want and let us do the rest.

Part Number	AT91SAM7X	MCF5208	LPC2923
Manufacturer			
Core Variant	ARM7TDMI	GoldFire V2	ARM968E-S
Flash	262144	0	262144
RAM	65536	16384	16384
Max. Freq.	55	166	125
Dhrystone MIPS	50	159	156
Timer Bits	16	32	32

We help you research your best option. Nowhere else can you compare your best options side-by-side from different manufacturers. Click on the device you want, and a product page lets you select Distributor Buy/Quote options, send RFQs, download datasheets, and more. Plus--Hearst stock check gives you up-to-date inventory on every device.



Once you have the chip that meets your needs, review and compare the hardware and software development tools that support it from multiple manufacturers, and buy them on-line through our shopping cart.



Shave days off your schedule with Embedded Developer, the only site in the world where you're only clicks away from finding the chips and tools to get you up and running, quickly. Try EmbeddedDeveloper.com, or EmbeddedDeveloper.cn in Chinese.

EMBEDDEDDEVELOPER.COM
FIND. COMPARE. BUY.

EMBEDDEDDEVELOPER.CN
查询. 选型. 采购

The Sites for Engineers with a Job to Do.

upsMIB = .1.3.6.2.1.33
Groups

Sub-groups			
		Objects	Sub Objects
.1 upsObjects [upsMIB 1]	.1.1 upsIdent [upsObjects 1]	.1.1.1 upsIdentManufacturer [upsIdent 1]	
		.1.1.2 upsIdentModel [upsIdent 2]	
		.1.1.3 upsIdentUPSSoftwareVersion [upsIdent 3]	
		.1.1.4 upsIdentAgentSoftwareVersion [upsIdent 4]	
		.1.1.5 upsIdentName [upsIdent 5]	
		.1.1.6 upsIdentAttachedDevices [upsIdent 6]	
	.1.2 upsBattery [upsObjects 2]	.1.2.1 upsBatteryStatus [upsBattery 1]	
		.1.2.2 upsSecondsOnBattery [upsBattery 2]	
		.1.2.3 upsEstimatedMinutesRemaining [upsBattery 3]	
		.1.2.4 upsEstimatedChargeRemaining [upsBattery 4]	
		.1.2.5 upsBatteryVoltage [upsBattery 5]	
		.1.2.6 upsBatteryCurrent [upsBattery 6]	
		.1.2.7 upsBatteryTemperature [upsBattery 7]	
	.1.3 upsInput [upsObjects 3]	.1.3.1 upsInputLineBads [upsInput 1]	
		.1.3.2 upsInputNumLines [upsInput 2]	
		.1.3.3 upsInputTable [upsInput 3]	
		.1.3.3.1 upsInputLineIndex [upsInputTable 1]	
		.1.3.3.2 upsInputFrequency [upsInputTable 2]	
.1.3.3.3 upsInputVoltage [upsInputTable 3]			
		.1.3.3.4 upsInputCurrent [upsInputTable 4]	
		.1.3.3.5 upsInputTruePower [upsInputTable 5]	
...

Table 1—Beginning on the left is the first of three groups below the standard upsMIB node .1.3.6.2.1.33, “upsObjects.” It has nine sub-groups below it, beginning here with “upsIdent.” Note that “upsIdent” has six objects below it, starting with object “upsIdentM anufacturer.” Each object in the MIB has its own unique path sequence, which is its shorthand dotted identifier .

compact, but it doesn't lend itself to human readability. Therefore, each manufacturer prepares an MIB text file covering the objects supported by their product. The text file describes each object using ANS.1 notation. Each object consists of five fields describing the textual name, variable type, textual description, read/write access type, and status.

In Listing 1, the read-only object named `upsIdentManufacturer` is a string of up to 32 characters that contain the name of the UPS manufacturer. Other information here identifies this as the first object in the `upsIdent` group under the `upsObjects` group of the `upsMIB`. This object, and all the other objects in the `RFC1628UPS-MIB.txt` file, aren't merely for your perusal. The text file also performs as a template. At the managed device end of the network, the manufacturer uses the `MIB.txt` file so that the storage location of the data variables will be known by both the SNMP agent and the embedded application. The agent

is responsible for exchanging data packets with the NMS. Each packet contains an SNMP message holding compressed data. The agent decodes an SNMP Get message as a request for data, grabs the requested data, and encodes it into an SNMP Response packet. Or, if the decoded message is an SNMP Set message, the agents store the data at the appropriate location and return an SNMP Response message indicating the status of the request. At the managed device, the MIB allows the SNMP agent to know all about the data it will be working with. The

NMS uses this MIB again at the other end of the network.

NETWORK MANAGEMENT

The management of networks refers to the activities, methods, procedures, and tools used in the operation, administration, maintenance, and provisioning of networked systems. Keeping tabs on managed devices—such as servers, routers, and switches—gives IT personnel a way to monitor a network's health. But today, the SNMP goes beyond the task of network welfare.

The SNMP has found its way into

Listing 1—The read-only object (`upsIdentManufacturer`) is a string of up to 32 characters that will contain the name of the UPS manufacturer.

```
upsIdent OBJECT IDENTIFIER ::= { upsObjects 1 }
upsIdentManufacturer OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..31))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The name of the UPS manufacturer."
    ::= { upsIdent 1 }
```

PROFESSORS

The Circuit Cellar college program puts quality engineering information in the hands of your students every month. Signup now to get Circuit Cellar distributed to your class this semester.

Stack of books: ELECTRONIC COMMUNICATIONS, Op-Amp Design Techniques, MATHEMATICS IN ELECTRONICS, Linear IC Technology, BASIC INSTRUMENTATION, Introductory Circuit Analysis

To update your professor account or to find out more about our college program, visit www.circuitcellar.com/products/collegeprogram/

Got Serial, Need Network?

Bluetooth Qty 1 \$145

Ethernet Qty 1 \$99

Wireless Qty 1 \$199

Volume Discounts Available

gridconnect
www.gridconnect.com
+1 800 975-4743

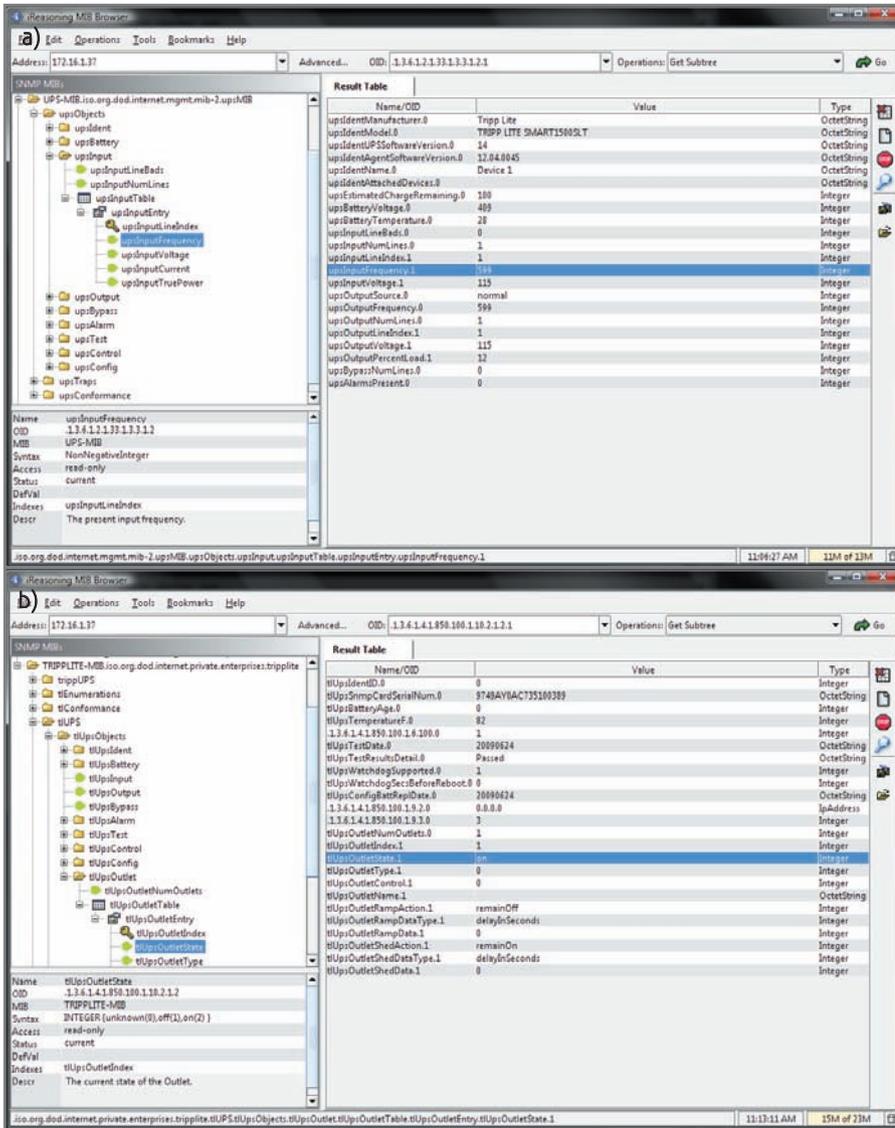


Photo 1a—MIB browsers can send SNMP messages to a managed device on the network. You'll need to know the network IP address. And it also helps to import the MIB.txt file distributed with the device. Although you can still access the device, the data will have no context. **b**—A private MIB.txt file may be used when the device's objects don't conform to one of the standard MIBs. When a private MIB has been produced for a managed device, it should use a private enterprise number allocated by the Internet Assignment Numbers Authority (IANA). This is a free service.

all industries thanks to the common thread interconnecting our world. Not only does a network standardize connectivity, but it also removes the need for local presence. The industry has seen growth in the use of SNMP monitoring and control of both commercial and residential properties. Environmental, security, automation, and data-logging applications take advantage of the SNMP.

Most network device manufacturers provide applications that give access to their products. You can find free MIB browsers that not only communicate

directly with a managed device via SNMP messages, but also have higher functions to help in the discovery of accessible SNMP MIBs (like walk). The Tripp Lite UPS on my home network has implemented some objects in the standard UPS-MIB. These can be accessed using a browser like the iReasoning MIB browser shown in **Photo 1a**. Here it's displaying an import of the standard UPS-MIB.txt file in the left panes and a subtree search of the related OIDs in the right pane. I highlighted one OID—"upsInputFrequency"—that

shows the present value (in tenths of a cycle) requested from the Tripp Lite UPS. **Photo 1b** shows a private "Tripplite-MIB." Remember our earlier discussion of a Private Enterprise Number (PEN)? Tripp Lite's PEN—"850"—gives them full control over special objects they want to define in their own Tripplite-MIB.txt file. I highlighted the "tlupsOutletState," which indicates the present state of the Tripp Lite conditioned power outputs.

If you look carefully, you can see a number of OIDs listed by their dot nodes as opposed to object names. These are objects that are not defined in the Tripplite-MIB, but exist in the device as real objects. Tripp Lite chose not to document these and others found using a (discovery) operation called "walk" in their Tripplite-MIB. Some of these are proprietary and used by their Power Alert Shutdown Agent application software.

IT'S YOUR TURN

The PEN gives every design engineer a way to create a device that can be managed over a network using the SNMP. As long as you follow the rules for declaring objects in your personal MIB—and as long as you support SNMP communication by including SNMP agent support in your product—you will have a product that can be added to the growing base of SNMP-managed devices. If your widget falls within a standard MIB, you won't even need a PEN. However, it's good to know that there's a way to define your device if it's nonstandard so that anyone can use it. And the registration process for your own private PEN is free.

You'll find that most of the popular embedded micro manufacturers already provide support for SNMP in their TCP/IP stack offerings. This should also include an MIB editor/compiler. If you want to write your own applications to make use of managed devices, you can find SNMP support for most languages running on almost any OS.

SNMP devices keep network traffic to a minimum. This is especially

useful when multiple devices must be polled in timed periodic fashion. Data-logging applications require the kind of continuous access that could be vulnerable to network bottlenecks.

In the future, I'll work on a managed device that will have multiple distributions on a network. This will require writing a management application to handle data monitoring and logging, and it will include separate alarm parameters for each device. Let me know if you're already using SNMP, or if you're just interested in reading more about this growing trend. ☺

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

SOURCES

MIB Browser | iReasoning, Inc.
www.ireasoning.com

RESOURCES

D. Bruey, "SNMP: Simple? Network Management Protocol," *RaneNote* 161, Rane Corp., 2005, www.rane.com/note161.html.

D. Harrington et al, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," RFC3411, The Internet Society, 2002, <http://tools.ietf.org/html/rfc3411>.

Internet Assignment Numbers Authority (IANA), www.iana.org.

K. McCloghrie et al, "Structure of Management Information Version 2 (SMIV2)," RFC2578, The Internet Society, 1999, <http://tools.ietf.org/html/rfc2578>.

MIB Downloads, ByteSphere Technologies, www.oidview.com/mibs/0/md-0-1.html.

R. Presuhn (ed), "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)," RFC3416, The Internet Society, 2002, <http://tools.ietf.org/html/rfc3416>.

R. Presuhn et al, "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)," RFC3418, The Internet Society, 2002, www.ietf.org/rfc/rfc3418.txt?number=3418.

M. Rose and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets," RFC1155, <http://tools.ietf.org/html/rfc1155>.

EzPCB

We add value to PCBs when others just sell it.

- One Stop Manufacturing Service
- Free Electronics Components
- Free Prototyping Assembly
- Professional Consultant

Designing Service
3D Enclosure Designing Virtual Assembly PCB Design

Design Prototype Production

Component Fpc

Quick Prototype Keypad

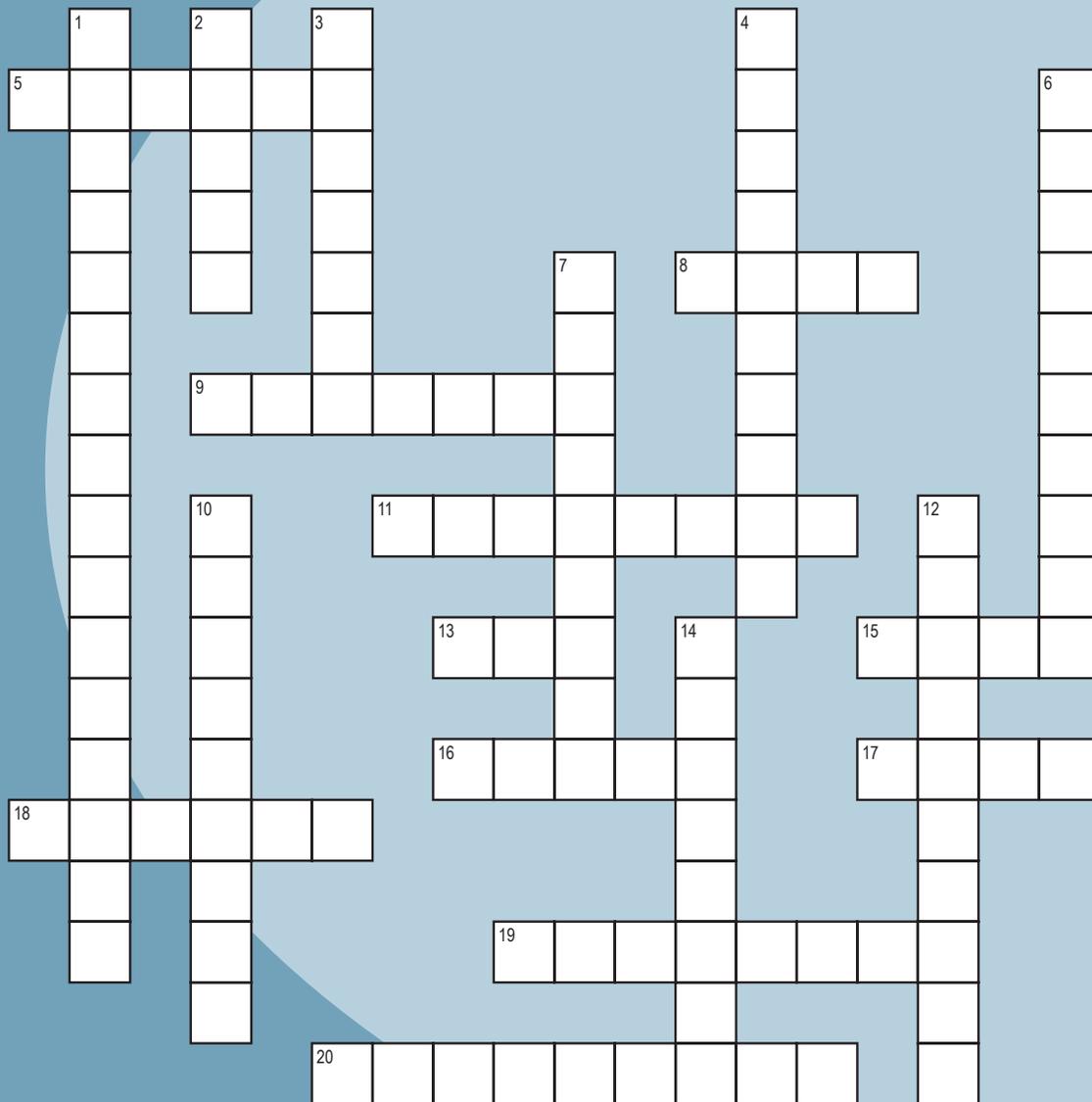
Pcb Assembly Quick Prototype

Components Fpcs Keypads

EzPCB

www.EzPCB.com Email: sales@ezpcb.com

CROSSWORD



Across

5. + Pole
8. Instruction to move ahead or back in a program
9. Current limiter
11. GUI widget for item selection
13. Address for a device on an Ether net network
15. Transmit electricity
16. Abnormal end
17. To release gas
18. An old circuit in use
19. 3,600 Joules [two words]
20. 106 cycles/second

Down

1. Cot⁻¹ [two words]
2. An ALU performs arithmetic and what?
3. Spiral
4. The "Board" type at <http://bbs.circuitcellar.com/phpbb/>
6. Time from submission to completion
7. Time from On to Off [two words]
10. Eleven-sided
12. Doesn't conduct; insulates
14. L1 is the first

The answers are available at
www.circuitcellar.com/crossword.

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.

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.

E10-Relay+ RS485
Opto232
RS232 RS485
E10-npn+

Incredibly Easy to Program!
Our software is used by many colleges for teaching PLCs!

Get Free Ladder Logic Simulator:
www.tri-plc.com/cci.htm

TRI
Triangle Research

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

DESIGN IN HALF THE TIME



CCS-C Compiler for all Microchip PIC® MCU

- ✓ Over 307 Built-in Functions for optimized code
- ✓ 133 Example Programs for quick start-up

Compilers starting at \$150

Coming Soon! New Compiler Features & Enhancements Details to come in V 4.1xx!

www.ccsinfo.com/CCNEW
sales@ccsinfo.com · 262.522.6500 x35

IC Device & Package Converters

Quick-Turn Custom Solutions

- Upgrade Your PCB
- Fix Design Problem
- Replace Obsolete Part
- All SMT Package Types

Design and assembly - Fast

Economical Replacement for QFP, PLCC, BGA, SOIC IC's

RoHS

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

Order online at:
www.melabs.com

microEngineering Labs, Inc.

Phone: (719) 520-5323

Fax: (719) 520-1867

Box 60039

Colorado Springs, CO 80960

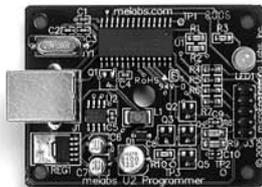
Development Tools for PIC® Microcontrollers

USB Programmer for PIC® MCUs \$89.95 (as shown)

RoHS Compliant

Programs PIC MCUs including low-voltage (3.3V) devices

Includes Software for Windows 98, Me, 2K, & XP



With Accessories for \$119.95:
Includes Programmer, Software, USB Cable, and Programming Adapter for 8 to 40-pin DIP



Serial LCDs

2-line x 16 \$39.95
4-line x 20 \$49.95

Quantity Discounts Available!

LAB-X Experimenter Boards



Pre-Assembled Boards Available for 8, 14, 18, 28, and 40-pin PIC® MCUs
2-line, 20-char LCD Module
9-pin Serial Port
Sample Programs
Full Schematic Diagram

Pricing from \$79.95 to \$349.95

PICPROTO™ Prototyping Boards



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

Pricing from \$8.95 to \$19.95

BASIC Compilers for PICmicro®



Easy-To-Use BASIC Commands
Windows 9x/Me/2K/XP Interface

PICBASIC™ Compiler \$99.95

BASIC Stamp 1 Compatible
Supports most 14-bit Core PICs
Built-In Serial Comm Commands

PICBASIC PRO™ Compiler \$249.95

32-bit signed variables and math operations
Supports Microchip PIC10, PIC12, PIC14, PIC16, PIC17, and PIC18 microcontrollers
Direct Access to Internal Registers
Supports In-Line Assembly Language
Interrupts in PICBASIC and Assembly
Built-In USB, I2C, RS-232 and More
Source Level Debugging

See our full range of products, including books, accessories, and components at:

www.melabs.com

It writes your **USB Code!**

NEW! HIDmaker FS for Full Speed FLASH PIC18F4550

Creates complete PC and Peripheral programs that talk to each other over USB. Ready to compile and run!

- Large data Reports
- 64,000 bytes/sec per Interface
- Easily creates devices with multiple Interfaces, even multiple Identities!
- Automatically does MULTITASKING
- Makes standard or special USB HID devices

NEW! "Developers Guide for USB HID Peripherals" shows you how to make devices for special requirements.



Both PC and Peripheral programs understand your data items (even odd sized ones), and give you convenient variables to handle them.

PIC18F Compilers: PICBASIC Pro, MPASM, C18, Hi-Tech C.

PIC16C Compilers: PICBASIC Pro, MPASM, Hi-Tech C, CCS C.

PC Compilers: Delphi, C++ Builder, Visual Basic 6.

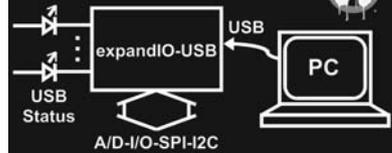
HIDmaker FS Combo: Only \$599.95

DOWNLOAD the HIDmaker FS Test Drive today!

www.TraceSystemsInc.com

301-262-0300

Revolutionary new expandIO-USB chip



- Ideal for adding USB to sensors & peripherals
 - No drivers needed for Windows, Mac, Linux
 - No microcontroller programming required
 - Also check out our USB-232 USB to UART
- www.hexwax.com - Buy from Mouser & Farnell

Amazing PIC programmer

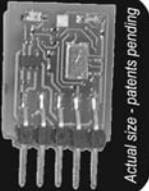
Most devices supported

ICSP, SQTP, & copy limits

\$32 at Digikey

& Mouser

www.flexipanel.com



PC/104 PERIPHERALS

SCIDYNE Offers a Full Line of Innovative Modules for PC/104 Applications

ADIO-104 *This Module Does it all!*

- 16 Analog Inputs
- 8 Analog Outputs
- 24 Digital I/O lines
- 5V Power
- Pulse Accumulator
- Open-Drain Outputs

DIO96-104

- 96 Bi-Directional Digital Channels
- Input, Output and Strobed I/O functions
- Uses Familiar 82C55A Chips

XIO-R08

Add eight High-Power relay outputs to any digital port.

Call or Visit us on the web
1-877-724-3963
www.scidyne.com

I missed issues

Fill in the Missing Pieces

with **Circuit Cellar** magazine's CD-ROM archives.

These convenient PDF archives contain complete issues, just as they appeared when they were printed. Get back on track and follow along the uninterrupted Circuit Cellar timeline.

For package deals on all of our archives, visit: www.circuitcellar.com/archives/

AVOCET SYSTEMS, INC.

www.avocetsystems.com

Microcontroller Development Tools

68000 68010 683xx
68HC11 68HC12
68HC16 Z80/EZ80
6502 65816
PIC16

Macro Assemblers
C Compilers
Software Simulators

(207) 596-7766
sales@avocetsystems.com

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 interface
- Easy to program
- In stock!

Add color graphics to any 8/16 bit embedded system. Easy, fast and flexible. Up and running in hours!

REACH TECHNOLOGY INC.
www.reachtech.com • 510-770-1417

842 Boggs Avenue • Fremont, CA 94539

Solve complex signal acquisition problems...

- positioning & control
- environmental
- acceleration
- transients
- pressure
- vibration
- sonar
- GPS
- Linux Driver
- Guaranteed in stock
- Many newly added features
- 16-bit analog inputs and outputs
- Million sample FIFO eliminates interrupts
- Wide analog input and output ranges
- -40°C to +85°C Standard
- Order 24/7, fast and easy.

www.stx104.com
Apex Embedded Systems
help@stx104.com • 608-256-0767 x24

PHYTEC

••• Inside great products. Behind great ideas.

phyCORE® System on Modules:

- shorten time-to-market
- reduce development costs and avoid substantial design issues and risks
- Windows® Embedded CE and Linux BSPs (processor-dependent)
- \$129/unit benchmark price at 1K for ARM9-based SOM
- Design Services available to assist with deployment into target applications

phyCORE-LPC3250

- ARM11: i.MX35, i.MX31
- ARM9: i.MX27, LPC3250, LPC3180
- Cortex M3: STM32F103
- ARM7: LPC2294
- XScale: PXA270
- x86: Z510, Z520, Z530 (Atom®)
- Blackfin: ADSP-BF537
- Coldfire: MCF5485
- PowerPC: MPC5554, MPC5567, MPC5200B, MPC565, MPC555

phyCORE® Rapid Development Kits include SOM, Carrier Board, LCD (kit specific), schematics, software, free BSP for applicable kits and a start-up guarantee. The Carrier Board serves as a target reference design, allowing the SOM to easily port to the user's target hardware.

www.phytec.com | 800.278.9913 | www.phycore.com

www.can232.com

Only \$108 €89

The original (Controller Area Network) CAN to RS232 serial converter.

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 independent
- 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

www.canusb.com

ezLCD - The "Smart Display" makes integrating a GUI ez!

8.0" ezLCD p/n: ezLCD+105

- *Versatile Programming LCD module
- *USB, SPI, RS232 Interfaces
- *Bright 250 Nit LED Display (800 x 600)
- *Integrated Touch Screen
- *LUA Scripting Language capable - For stand alone embedded apps
- *Memory 3.8 MB + SD to 2G
- *ezLCD's are also available in 2.7", 3.5", 5.6", 6.4", 10.4"

www.earthlcd.com
Call for Custom Display Configurations

I²C/SMBus

- Bus Monitors
- Protocol Analyzers
- Host Adapters
- Multiplexers
- Battery Applications
- Software Tools

MCC

Micro Computer Control

FC is a trademark of Philips Corporation

www.mcc-us.com

Display with Touch Screen

start at \$379 OEM

- Embedded, Standalone, C/C++ Programmable...
- Low Cost User Interface with Ethernet, USB, CF, RS232, ADC, DAC, I/Os, Relays...
- U-Drive™ 5.7" QVGA TFT, CAN, Ethernet, CF, ADC, DAC,

60+ Low Cost Controllers with TFT, ADC, DAC, UARTs, 300 I/Os, solenoid, relays, Compact-Flash, LCD, Ethernet, USB, motion control. Custom board design. Save time and money.

TERN INC. 1950 5th Street, Davis, CA 95616 USA
Tel: 530-758-0180 • Fax: 530-758-0181 www.tern.com • sales@tern.com

World Class Solutions

www.ProlificUSA.com

USB 3.0

USB 2.0

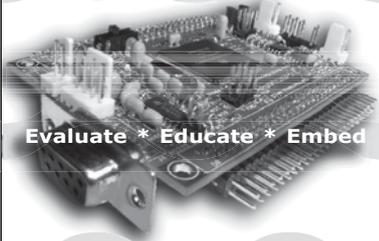
GPS

SoC

Adapt9S12XDP512
Modular Prototyping System

- * Robotics and Mechatronics
- * Electronic Fuel Injection
- * Freescale 9S12XDP512
- * RTOS-capable

Starting at \$125!



Evaluate * Educate * Embed

Program in
Assembler, BASIC, C, and Forth

www.TechnologicalArts.com

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

CROSSWARE
Tools for Embedded Development

64-BIT FLOATING POINT

A feature of the Crossware 8051 Development Suite

- C Compiler
- Assembler
- Code Wizards
- Debugging
- Simulation
- Support

Advanced software tools since 1984

www.crossware.com
info@crossware.com

techSOL
TECHNICAL SOLUTIONS INC

TPC-43A

Fanless Wall-mountable 32bit 200Mhz ARM9 CPU SDRAM/FLASH with Linux 2.6.2x with drivers & GUI



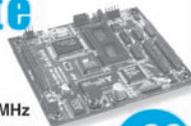
RTC w/battery
16bit Audio
10/100 Ethernet
RS232, IrDA
USB Host
USB Device
SD/MMC card
Optional: Java
- Bluetooth
- ZigBee
- WiFi

4.3" WQVGA (480x272) x 65K Color Touch Screen LCD w/backlight

From \$339

www.medallionsystem.com

Flashlite 186



- 186 processor @ 33 Mhz
- DOS w/ Flash File system
- 44 Digital I/O lines w/ CPLD
- Console / Debug Serial Port
- 7-34V DC or 5V DC power
- 2 Serial Ports
- Accepts 8MB DiskOnChip
- 2 16-bit Timers
- 512K DRAM & 512K Flash
- Watchdog Timer
- Expansion options with Peripheral Boards

\$69
QTY 1

- \$99 Development System**
- Development kit includes:
- Flashlite 186 controller
 - Borland C/C++ ver 4.52
 - FREE Email Tech Support
 - Serial Driver library
 - AC Adapter and cable

Call 530-297-6073 Email sales@jkmicro.com
On the web at www.jkmicro.com

JK microsystems

Weather Instruments
for PCs

AAG
electronica

www.aagelectronica.com

Low Cost CAD Software for Windows XP, NT and Vista

- Circuit design package with schematic entry circuit-board layout with autorouting and simulation for only \$499!
- Buy modules starting at \$119 (SuperCAD, SuperPCB, mentalSPICE & SuperSIM)
- Order and download instantly
- Full up package allows up to 16 layers plus 4 power planes
- Manufacture circuit boards at any board house

Mental Automation, Inc.

253-858-8104
www.mental.com

CUSTOM MEMBRANE KEYBOARDS / SWITCHES



- 1 TO 2 WEEKS TURNAROUND
- VERY COMPETITIVE PRICING
- Ex.: (5) 4-switch keyboards for \$395.00
- PCB backed switches
- Custom metal backplates/assemblies
- Electronic assemblies/graphic overlays
- Electronic file transfer capabilities

Picofab Inc.

4780B Blvd. Henri-Bourassa
Charlesbourg, Quebec, Canada G1H 3A7
Tel: (418) 622-5298 • Fax: (418) 622-9996
Email: sales@picofab.net

I2C SPI 1 Wire



3 Separate Buses
5V & 3V
Simple ASCII Interface
Cross Platform - All OS

www.i2cchip.com

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
78 AAG Electronica, LLC	69 Embedded Developer	22 Keil Software	77 Phytel America LLC
49 AP Circuits	20 ExpressPCB	53 Lakeview Research	78 Picofab Inc.
78 All Electronics Corp.	73 ezPCB	77 Lawicel AB	31 Pololu Corp.
77 Apex Embedded Systems	76 FlexiPanel Ltd.	5 LeCroy	77 ProlificUSA
7 Atmel	11 Front Panel Express LLC	11 Lemos International Co. Inc.	C3 Rabbit, A Digi International Brand
76 Avocet Systems, Inc.	49 Futurlec	77 MCC (Micro Computer Control)	77 Reach Technology, Inc.
33 CWAV	71 Grid Connect, Inc.	15 MachinePIER	65 Saelig Co.
40 CadSoft Computer, Inc.	42 HobbyLab, LLC	78 Mental Automation, Inc.	76 Scidyne
19 Comfile Technology, Inc.	78 I2CChip	29 Microchip Technology, Inc.	78 Technical Solutions, Inc.
78 Crossware Products, Inc.	30, 57 ICbank Inc.	75 microEngineering Labs, Inc.	24, 25 Technologic Systems
75 Custom Computer Services, Inc.	53 Intuitive Circuits LLC	23 Mouser Electronics	78 Technological Arts
42 DesignNotes	75 Ironwood Electronics	C2 NetBurner	77 Tern, Inc.
34 EMAC, Inc.	32, 34 JKmicrosystems, Inc.	21 Noritake Co., Inc.	1 Texas Instruments
77 Earth Computer Technologies	78 JKmicrosystems, Inc.	53 Nurve Networks LLC	76 Trace Systems, Inc.
32 Elprotronic	41 Jameco	15 PCBCore	75 Triangle Research Int'l, Inc.
10 Elsevier	42 Jeffrey Kerr, LLC	C4 Parallax, Inc.	2, 3 WIZnet

PREVIEW of December Issue 233

Theme: **Programmable Logic**

Retrocomputing on an FPGA: Reconstructing an '80s-Era Home Computer with Programmable Logic

Building Microprogrammed Machines with FPGAs

A Practical Parallel CRC Generation Method

WIZnet iMCU Design Contest Primer

THE DARKER SIDE Digital Modulations Demystified

ABOVE THE GROUND PLANE Memories Not Forever

FROM THE BENCH Extending and Isolating the I²C Bus

SILICON UPDATE IP Unplugged: The Internet, Silicon, and Software

ATTENTION ADVERTISERS

**January Issue, 234
Deadlines**

Space Close: Nov. 9
Material Close: Nov. 16

**Theme
Embedded Applications**

**Bonus Distribution
DesignCon**

Call Shannon Barraclough
now to reserve your space!

860.875.2199

e-mail: shannon@circuitcellar.com

PRIORITY INTERRUPT



by Steve Ciarcia, Founder and Editorial Director

Keeping the Lights On

Americans like to think that Thomas Edison came up with the light bulb in 1879 all by himself. But a patent settlement clearly gives equal credit to Joseph Swan, an English inventor. More importantly, while those early designs have evolved from glowing carbonized bamboo in a vacuum to radiant tungsten in argon gas, the concept and implementation of incandescent light bulbs have remained pretty much untouched for 130 years. Certainly no other electrified invention can claim so much history, so why dump it all now?

Barely 6% of the energy powering an incandescent bulb is light. And, while 94% radiated as heat might seem absurd, it isn't always viewed as a complete waste when there is 3' of snow outside and you're heating a house in Connecticut with \$4-per-gallon heating oil. Of course, personal rationalization aside, the vast majority of the world views incandescent bulbs as grossly inefficient. As of September 1, you can no longer purchase them in Europe, and the same situation will exist here in 2012. The primary lighting source for the immediate future is supposed to be compact fluorescent lights (with LEDs a bit more down the timeline).

Invented about the same time as the incandescent bulb, it took until the 1970s for us to have a workable CFL design, and it wasn't until the 1990s that they became affordable—today they are \$1.50. They use about one-fourth the power of an equivalent-lumen incandescent bulb, last 10 times longer, and run a whole lot cooler. What's not to love? ;-)

I am probably a very unique case, but let me just tell you some of the realities of no more incandescent bulbs from my perspective. And, before any newbie readers e-mail me to chastise me for my energy inefficiency, let me remind you that I have 10,760 W of PV panels powering my house and right now my latest electric bill (\$0) says that I still have a 700-kWhr surplus (they owe me). Based on what the electric company will give me for the surplus power when we settle up at the end, it's actually cheaper for me to continue using the remaining incandescent bulbs I have versus the cost of buying more CFLs.

OK, I'll stop ranting. I understand the bigger picture regarding energy waste, but simply substituting CFLs or LEDs for every incandescent light bulb is a myth. The naive truth is "maybe."

If you have a light next to your easy chair where you sit to read *Circuit Cellar*, it is a perfect candidate for a CFL replacement. If your Home Control System (HCS) automatically turns that light on or the wall switch controlling the light is a dimmer switch, good luck! If you have a 150-W motion-controlled flood light on your front walkway, it might seem to be the perfect candidate for a 38-W CFL flood light replacement—provided the controller uses a relay and not a triac, and that you don't mind being 10 seconds past the point of needing light before you actually see any.

I might have a unique case of triple witching (electronically speaking), but I think there are a lot of *Circuit Cellar* readers who utilize powerline or wireless communication and control technologies. You may not be running a complete HCS like I do, but the inoperability of any automated device is just as frustrating.

CFLs are wonderful, except when they don't work. For the most part, they don't like low temperatures, short-duration power cycling, or semiconductor-based power controls (like dimmers or lamp modules). X10-type automated lighting controls should be tested with the specific CFL to be used to make sure that it works and that leakage currents don't cause the CFL to flash. If you have automated lighting or high-traffic areas where the CFL constantly cycles (like a staircase), be prepared to see an 85% reduction in lifespan on most brands. If you need to keep the dimming capability and still want a CFL, be prepared to pay three to five times more for a "dimnable" CFL and wait a few seconds longer for it to light.

In my opinion, CFLs are an interim fix on the road to better lighting efficiency, but they certainly aren't the solution. In the long run, I believe LED lighting is the answer, but be careful before you jump off that cliff right now too. So far, half of the lights at my house are fluorescents in one form or another. I live with the lower lifespan in some places simply because 5 W trumps 60 and I feel like I should try. In another dozen places, I'm using LED bulbs. Their fast turn-on and long cycle life is great for staircases, bathrooms, and hallways, but they are still too expensive for general implementation (some high-brightness bulbs also radiate too much RFI). I'm currently testing a 15-W 40" fluorescent tube replacement that looks promising, but at \$70 versus \$3 I need a lot more reason to swap out the 50 or so fluorescent tubes I have.

The reality is that I have a very eclectic mix of lighting solutions that are constantly evolving. Much to the chagrin of the politicians, I have to honestly say that a few of my lighting applications are best handled with regular incandescent (or halogen) bulbs. For anyone critical enough to take me to task over it, I just say that when you're generating your own power too, we'll talk. ;-)

steve.ciarci@circuitcellar.com

Sweet!

Introducing the MiniCore™ Series of Networking Modules

Smaller than a sugar packet, the Rabbit® MiniCore series of easy-to-use, ultra-compact, and low-cost networking modules come in several pin-compatible flavors. Optimized for real-time control, communications and networking applications such as energy management and intelligent building automation, MiniCore will surely add sweetness to your design.

- **Wireless and wired interfaces**
- **Ultra-compact form factor**
- **Low-profile for design flexibility**
- **Priced for volume applications**

Wi-Fi and Ethernet Versions



MiniCore Module Development Kits

From **\$99** Limited time offer.



Buy now at: **1.888.411.7228 • rabbitwirelesskits.com**

1.888.411.7228
rabbitwirelesskits.com
2900 Spafford Street, Davis, CA 95618



NEW!

RF Transceiver

Send serial data wirelessly



The **433 MHz RF Transceiver** (#27982; \$39.99) is a very easy to use and low cost module capable of sending serial data wirelessly between microcontrollers or to a PC. The low power consumption makes this module ideal for use in battery powered applications. This module sends and receives data by AM or CPCA modulation, thus offering a higher average output power which extends its range. This module is equipped with an RSSI feature that can be utilized to improve power efficiency by waking up circuitry only when an external signal is detected.

Features:

- High speed data transfer rates: 1200-9600 bps
- SIP header allows for ease of use with breadboards
- Compatible w/many microcontrollers including the Propeller™ chip and BASIC Stamp® modules
- Power-down mode for conservative energy usage (longer battery life)
- Line of sight range up to 2500 feet (depending on operating conditions)

Application Ideas:

- Remote controlled Boe-Bot® robot
- Wireless data acquisition
- Remote signal beacon for adventure seekers
- Remote industrial monitoring
- Lighting control
- Keyless entry



\$39.99
per module

To order the **433 MHz RF Transceiver** visit www.parallax.com. Or call the Parallax Sales Department toll-free: 888-512-1024 (Monday - Friday, 7 a.m. - 5 p.m., PT).

Prices subject to change without notice. BASIC Stamp and Boe-Bot are registered trademarks of Parallax Inc. Parallax and the Parallax logo are trademarks of Parallax Inc.

PARALLAX 
www.parallax.com

Friendly microcontrollers, legendary resources.

Measuring Propagation Delay with a Universal Counter

Have you lost interest in universal counters? It's time to reexamine their use value in comparison to more expensive oscilloscopes. This article details how a counter makes a signal path delay measurement, the level of precision possible with a counter, and more. You'll also learn to identify and eliminate error in your measurement setup.

Over the past couple decades, many engineers have lost interest in using universal counters for signal-timing measurements (e.g., period, pulse width, and rise/fall time) because they prefer the more visual features of an oscilloscope. They would rather see a visual display of the signals they are trying to characterize than a digital display of a long sequence of numbers that can change to a totally different value with a slight adjustment of a single setting. Not to mention that the measurement capabilities of a scope go much further than just timing. Scopes often provide features that allow engineers to decode digital protocols and capture nonrepetitive signals, for example.

The main advantage that counters still hold over today's typical scopes is the ability to make high-accuracy and high-resolution time measurements. Typically, a quality counter offers 10 to 12 digits of resolution, while a scope typically has five to eight digits of resolution. An added bonus is that counters start at a much lower price than scopes. Thus, counters are viable solutions in applications such as timebase calibration, oscillator development/testing, and clock signal accuracy testing. A counter's high accuracy and high resolution also make it a powerful solution for high-precision digital signal propagation delay measurements. Only counters provide accurate measurement resolution down to nanosecond and even sub-nanosecond levels. This level of resolution is required for measuring the propagation delay of a cable, solder run, high-speed digital logic circuit, or any combination of the three.

How short of a delay time can counters accurately resolve? The answer depends on multiple factors, such as the frequency and accuracy of the counter's timebase,

whether statistical calculation can be employed, and more. In this article, I'll describe how a counter makes a signal path delay measurement, the level of precision possible with a counter, and how to maximize it. I also explain how to identify and eliminate error in your measurement setup.

TIME INTERVAL MEASUREMENTS

You can measure propagation delay using a counter's time interval measurement feature. A time interval measurement is an elapsed time measurement between some start event and some stop event. It is comparable to using a stopwatch to time a runner, where the start event is the signal telling the runner to go and the stop event is the runner crossing the finish line. For a counter making a time interval measurement, the start event is an electrical edge and the stop event is an electrical edge occurring later

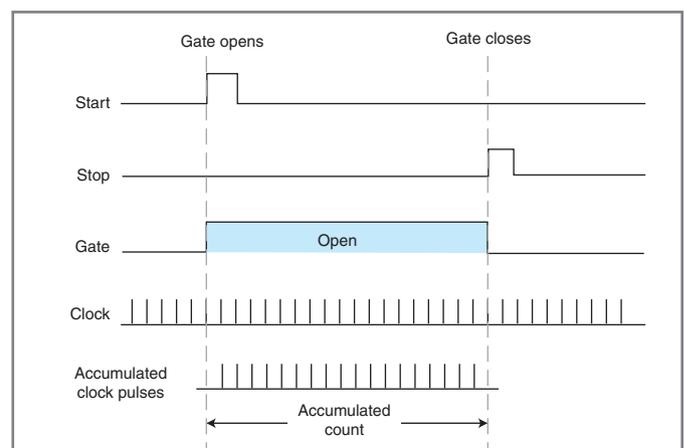


Figure 1—The time interval measurement process

in time. The time interval is measured by counting the number of clock pulses that occur between the start and stop event. Figure 1 depicts the time interval measurement process. The start event opens a “gate” that triggers the counter to start counting the timebase clock ticks. The counter continues tallying the timebase clock ticks until the stop event triggers the counter to close the gate and stop counting clock ticks.

There are two fundamental problems with this method of measuring the time interval between two events. The first is that the minimum time interval that can be measured is limited by the counter’s clock frequency. With a typical clock frequency of 10 MHz, the minimum time interval a counter can resolve is 100 ns (the period of one clock cycle). This means if you were trying to measure the signal path delay of an RG 58 A/U 50-Ω coax cable, which has a delay of approximately 0.052 ns/cm, the counter could only resolve a minimum cable length of 1,923 cm.

The second problem with this time interval measurement technique is that the counter’s clock is asynchronous to the start event. You have no way to tell if the clock count began right at the start of a clock pulse or right at the end of a clock pulse, so this can lead to a clock pulse count error of 1 to -1 in extreme cases.

To deal with these two limitations, modern counters typically employ some form of interpolation. Interpolation can come in both analog and digital forms. It can be implemented by employing a ramp voltage with a known linear slew rate. When a start event occurs, it starts the voltage ramp. Once the first clock edge occurs, an analog-to-digital converter measures the current ramp voltage. The measured ramp voltage and the known ramp rate are used to calculate the elapsed time between the start event and the first clock edge. A second interpolation circuit is used for the stop event. Combining the start and stop interpolation calculations enables a counter to capture start and stop events smaller than a single clock cycle with resolution into the sub-nanosecond realm. The ability to interpolate is typically what separates low-cost low-precision counters from high-precision universal counters. For example, Agilent’s 53131A and 53132A universal counters employ digital interpolation that enables them to resolve “single-shot” time interval measurements as low as 750 ps and 350 ps, respectively.

AVERAGING MEASUREMENTS

So far we have discussed only how low a counter can resolve a single propagation delay measurement (time-interval measurement). Single-shot measurements are done with just a single start and stop pulse versus continuous repeating start and stop events. The problem with single-shot measurements is that random error and noise in the start/stop signal—in the counter’s clock signal, in the input signal conditioning, and in the interpolation circuitry—will reduce the measurement’s useful resolution. Performing statistical averaging on a continuous start/stop signal can help reduce random error. The key here is that the error has to be random, meaning it takes on a Gaussian or normal distribution.

Averaging will not reduce systematic errors (discussed later). The more measurements you average together, the more the random error will approach zero. Of course, there is always a limit to averaging. The limit is usually the noise floor, which typically limits the benefits of averaging to more than 2 ps.

The majority of counters on the market today can store a certain number of readings in memory, and they include built-in statistical calculation capability (e.g., averaging) that can be applied to the stored readings. For instance, the 53131A and 53132A can store as few as two and up to 1,000,000 readings in memory that can be analyzed with built-in statistical tools. The only trade-off of employing averaging is a longer total measurement time.

SYSTEMATIC ERROR

There are two main types of systematic error that you have some control over when you make time interval measurements: trigger error and cabling/connector error. Let’s first focus on trigger error.

You have some control over the quality of the start and stop edges, the input settings, and the trigger-level settings. Any input start or stop trigger edges that do not have smooth voltage transitions through their rise or fall will cause timing errors, unless the irregularities are repeatable between the start and stop edges. For accurate trigger timing on digital signals, setting each input on the counter for DC coupling is important. AC coupling will have a drastic effect on the input digital signal’s slew rate. Also, the effect on the start event slew rate will differ from the effect on the stop event slew rate because the capacitance in the start input circuit will not be exactly matched with the stop input circuit capacitance. Trigger errors also surface if you set the trigger-level controls improperly on the start and stop events. You need to ensure each event triggers off the same percentage point of the slew rate.

Now let’s look at cabling and connection error. Of course, any type of cabling, adapters, and connectors needed to connect the counter to the start of the signal path and the end of the signal path you are trying to measure will add error to your delay measurement (in the form of increased time). The cables, adapters, and connectors also add loading affects to the start and stop signal, changing the slew rate and amplitude of the start signal versus the stop signal, which in turn adds to the time error of the measurement. When measuring the delay through high-speed logic, any effect the logic circuits have on the signal’s slew rate and amplitude will again cause timing error between the start and stop signal.

REDUCING SYSTEMATIC ERRORS

You can reduce systematic errors in cables by zeroing out their effects. Simply connect your start and stop cabling together along with your input signal and perform a time interval measurement to characterize the delay of your cabling. Then just subtract that measurement time from the final measurement time of the cabling plus the signal path you are characterizing.

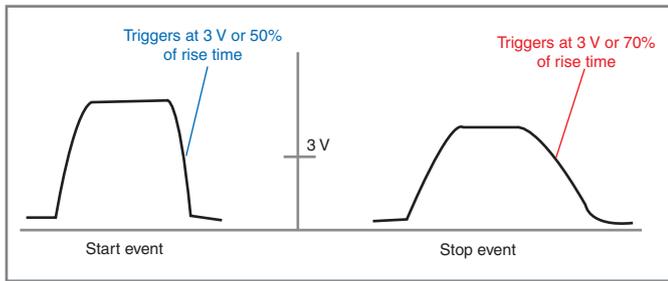


Figure 2—Measuring propagation delay through high-speed digital logic

Many counters have built-in offset capabilities that allow you to zero out the effects of cabling and connectors. When zeroing out the delay of your cabling and connectors, use as few additional connectors as possible (if any) because every connector added for the zero measurement will add timing error to the final measurement. Performing a delay measurement on your cabling to zero out its effects will also factor out any repeatable systematic trigger error. Using a single counter channel to perform your propagation delay measurement means you must use some type of T connector to route both your start and stop signals into the single channel. The delay added by the T connector cannot be zeroed out of your final measurement. The better choice is to use a two-channel counter that allows you to choose one channel as the start input and the other as the stop input. This will alleviate the need for a T connector, which will cut down on the error of your final delay measurement.

When measuring propagation delay through high-speed digital logic, you may want to reduce any error caused by slew rate and amplitude differences between the output signal (stop event) and input signal (start event). In **Figure 2** you can see that the trigger level is set for 3 V for the start and stop edges. The digital logic changed the amplitude and slew rate of the pulse, so the start and stop triggers occur at different percentage points on the edge.

To reduce this type of error, use the counter's time interval measurement capability or the built-in rise/fall time measurement capability to find the slew rate and amplitude of both the start event and stop event. Use that information to establish a trigger level for each based on a percentage of the slew rate. This step will eliminate any error caused by differences in the start signal and stop signal slew rates and amplitudes. You can also do this using a scope in conjunction with the counter for an easy visual way to match trigger levels based on a percentage of the slew rate.

ACCURACY & RESOLUTION

We discussed various sources of errors and how you can take advantage of the settings and functionality of a universal counter to overcome or reduce them in your digital propagation delay measurements. So, putting it all together, where does that leave you as far as resolution and accuracy are concerned? This is not an easy question to answer because counter specifications are often hard to read and vary from manufacturer to manufacturer.

Here is a basic framework to determine a counter's resolution and accuracy. If you're dealing with a single-shot delay measurement, your useful resolution will be limited by the manufacturer-specified time interval measurement random error, the random error of your start event, and the random error of your stop event. The total amount of useful resolution is obtained by using the root sum of squares (RSS) method on the three terms.

$$\text{Single shot resolution} = \sqrt{\text{counter random error}^2 + \text{start random error}^2 + \text{stop random error}^2}$$

For counters that employ a high-quality quartz oscillator and use some form of interpolation, the random useful resolution of the counter will typically be in the sub-nanosecond region. With a continuous start and stop signal, the averaging capabilities of the counter can be used to further improve the resolution of the counter.

$$\text{Resolution from averaging} = \frac{\text{Random timing error}}{\sqrt{N}}$$

N is the number of measurements averaged together. In theory, by continually increasing the number of readings averaged together, the random error will continue to approach zero. In reality, you will eventually hit the noise floor inside the counter. The noise floor depends on the counter itself and the environment it is in. Typically, in a quality counter, you can expect the noise floor to be less than 10 ps or somewhere between 11 and 12 digits of resolution. You can easily find the number of averages it will take to reach the noise floor by brute force—just keep trying higher averages until you no longer notice any improvement in the resolution.

Accuracy is dependent on your ability to reduce systematic errors. First, you will want to eliminate or reduce all the errors over which you have some control (discussed above)—namely, zeroing-out cable effects, setting the proper trigger level, using two channels, and setting the input for DC coupling. Then you're left with the systematic errors of the counter, which are out of your control, except for regular calibration. In quality universal counters, systematic error will typically add up to at least 1 ns or higher. But because systematic internal counter error is repeatable, you can eliminate the majority of it and most cabling effects by simply performing an offset measurement. The exception is long-term oscillator drift. Because the oscillator serves as our time reference for every measurement, we cannot perform an offset measurement to eliminate errors related to it; but fortunately, it is one of the smallest contributors to internal systematic error. For instance, with the 53131A and 53132A universal counters, timebase error can be less than 1 ps.

THE COUNTER ADVANTAGE

Counters offer higher accuracy, better resolution, and lower cost than scopes. In addition, they are viable solutions in applications that require high-precision, signal-timing measurements. This holds true when you are trying to capture the

propagation delay through a cable, solder run, high-speed digital logic circuit, or any combination of the three, down to the nanosecond range of useful resolution—and, in some cases, the sub-nanosecond range of resolution. 📏

Neil Forcier (neil_forcier@agilent.com) served in the U.S. Navy for six years as a test and measurement equipment calibration technician aboard the USS Harry S. Truman CVN-75. He earned a bachelor's degree in engineering from The Pennsylvania State University (University Park). Neil is an application engineer for Agilent Technologies, System Products Division. His areas of interest include embedded system design/testing, automated measurement technology, network architecture, and alternative energy.

SOURCE

53131A and 53132A Universal counters
Agilent Technologies | www.agilent.com