TIPS: Math Routines Using dsPIC Features
LOCATION: Australia
PAGE: 16

INSIGHT: Risk Analysis for Embedded Design
LOCATION: Canada
PAGE: 34

EE ESSENTIALS: IIR Filters Explained
LOCATION: France
PAGE: 58

CIRCUIT CELLAR

THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

OCTOBER 2012
ISSUE 267

SIGNAL PROCESSING

DIY, MCU-Based Camera Controller

Avoid Concurrency Problems When Designing

MOSFETs at Different Temperatures

A MEMS Solution for Gyro Stability

FAT Cache Technique to Optimize SD Card Access

PLUS
Embedded Inquiries
// STT-RAM-Based Memory Hierarchy
// The Future of System Design
// Next-Generation NVM Technology
// And More



$9.00US $10.00CAN
10>
0 74470 75349 0
www.circuitcellar.com

Now with 32MB Flash and 64MB RAM!



NANO54415

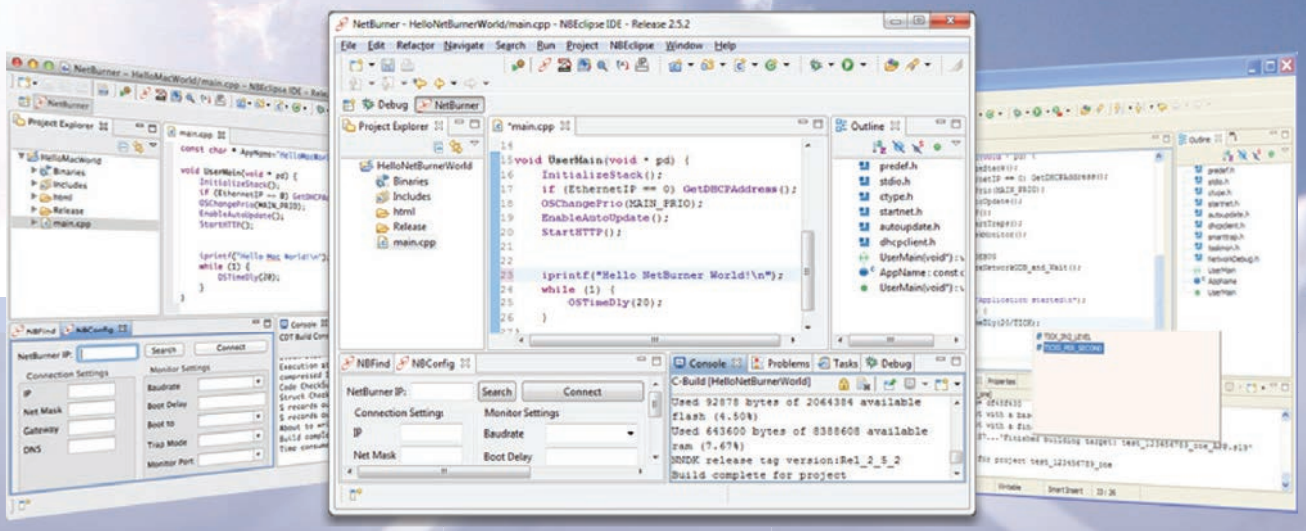
$69.00 Qty. 100



MOD54415

$89.00 Qty. 100

MOD54415 Core Module	NANO54415 Core Module
32-bit 250 MHz processor	32-bit 250 MHz processor
64MB DDR2 RAM	64MB DDR2 RAM
32MB flash	**8MB flash**
10/100 Mbps Ethernet	10/100 Mbps Ethernet
44 general purpose I/O	**30 general purpose I/O**
Eight UARTs	Eight UARTs
Five I2C	**Four I2C**
Two CAN	Two CAN
3 SPI	3 SPI
1-Wire®	1-Wire®
5 pulse width modulators (PWM)	**8 pulse width modulators (PWM)**
SSI	SSI
MicroSD flash card	MicroSD flash card ready
8 analog to digital converters (ADC)	6 analog to digital converters (ADC)
Two digital to analog converters (DAC)	Two digital to analog converters (DAC)



Quickly create and deploy applications from your Mac or Windows PC

Low cost NetBurner development kits are available to customize any aspect of operation including web pages, data filtering, or custom network applications. The kit includes platform hardware, ANSI C/C++ compiler, TCP/IP stack, web server, e-mail protocols, RTOS, flash file system, Eclipse IDE, debugger, and cables. The kit enables you to communicate with peripherals that use SD/MMC Flash Card (including SDHC), SPI, I²C, or the general purpose digital I/O interface. The NetBurner security suite option includes SSH v1, v2 and SSL support.

Development Kit for MOD54415
Part No. NNDK-MOD54415-KIT
$99.00 for a limited time

Development Kit for NANO54415
Part No. NNDK-NANO54415-KIT
$99.00

NetBurner
Networking in One Day!

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

freescale™
Alliance Member

Tickets & Registration: www.electronica.de/en/2012

Messe München International

the world needs innovative electronics. they are on display here.



electronica 2012

inside tomorrow

25th International Trade Fair
for Electronic Components,
Systems and Applications
Messe München
November 13–16, 2012
www.electronica.de

TASK MANAGER

Continuity of Content

I thank all the *Cellar Cellar* members who have contributed to CircuitCellar.com and followed the editorial department via Twitter (@editor_cc). The increase in site traffic and growing numbers of Twitter followers prove that you're finding our online and social offerings both interesting and handy.

Since the website's redesign in February 2012, we've been publishing projects, insightful articles, workspace write-ups, interviews, and event reviews. As we move forward, we'll create more continuity between the site, the print magazine, and the digital edition. The result will be easier access to project information, products, and additional content (e.g., project files, code, and videos). For instance, consider the content we've provided relating to Richard Lord's Photo-Pal camera controller for high-speed photography. We published Richard's article "MCU-Based 'Photo-Pal' Camera Controller" on CircuitCellar.com in late May 2012. The short online article introduced his project and featured some amazing photos he took of high-speed occurrences. This month, the print magazine includes the first article in Richard's multipart series about the project (p. 20). He describes the system's hardware and provides detailed schematics.

We also feature articles on topics ranging from FAT cache to IIR digital filters. Let's review.

On page 16, Stuart Oliver details how to use math routines that include the dsPIC hardware features, such as the accumulators and barrel shifter. He uses the math for implementing Assembler routines.

Turn to page 30 to learn how Kerry Imming uses FAT cache for SD card access. You can implement his cache technique in a variety of other applications.

Before you start a new project, familiarize yourself George Novacek's tips on managing project risk (p. 34). He explains how to define, evaluate, and handle risk. Better yet, why not just reduce risk by avoiding as many problems as possible?

Bob Japenga addresses this issue as well (p. 38). In the third part of his series on concurrency in embedded systems, he details how to avoid concurrency-related problems, which can be difficult because the more concurrency you add to a project, the more complicated it becomes.

Ed Nisley presented a MOSFET tester in his August 2012 article, "MOSFET Channel Resistance." In this issue, Ed covers temperature measurement, the control circuitry, the firmware's proportional integral control loop, and more (p. 42).

Check out Robert Lacoste's article on page 58 for an introduction to IIR digital filters. You'll learn about the differences between IIR filters, FIR filters, and analog filters.

Working with an unstable mechanical gyro? As Jeff Bachiochi explains, a MEMS system is the solution (p. 68).

Lastly, check out the interview with Helen Li on page 54. You'll find her impressive research exciting and inspirational.

cj@circuitcellar.com



CIRCUIT CELLAR®
THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

EDITORIAL CALENDAR

Issue	Theme
258 January	Embedded Applications
259 February	Wireless Communications
260 March	Robotics
261 April	Embedded Programming
262 May	Measurement & Sensors
263 June	Communications
264 July	Internet & Connectivity
265 August	Embedded Development
266 September	Data Acquisition
267 October	Signal Processing
268 November	Analog Techniques
269 December	Programmable Logic

Analog Techniques: Projects and components dealing with analog signal acquisition and generation (e.g., EMI/RF reduction, high-speed signal integrity, signal conditioning, A/D and D/A converters, and analog programmable logic)

Communications: Projects that deal with computer networking, human-to-human interaction, human-to-computer interaction, and electronic information sharing (e.g., speech recognition, data transmission, Ethernet, USB, I²C, and SPI)

Data Acquisition: Projects, technologies, and algorithms for real-world data gathering and monitoring (e.g., peripheral interfaces, sensors, sensor networks, signal conditioning, A/D and D/A converters, data analysis, and post-processing)

Embedded Applications: Projects that feature embedded controllers and MCU-based system design (e.g., automotive applications, test equipment, simulators, consumer electronics, real-time control, and low-power techniques)

Embedded Development: Tools and techniques used to develop new hardware or software (e.g., prototyping and simulation, emulators, development tools, programming languages, HDL, RTOSes, debugging tools, and useful tips and tricks)

Embedded Programming: The software used in embedded applications (e.g., programming languages, RTOSes, file systems, protocols, embedded Linux, and algorithms)

Internet & Connectivity: Applications that deal with connectivity and Internet-enabled systems (e.g., networking chips, protocol stacks, device servers, and physical layer interfaces)

Measurement & Sensors: Projects and technologies that deal with sensors, interfaces, and actuators (e.g., one-wire sensors, MEMS sensors, and sensor interface techniques)

Programmable Logic: Projects that utilize FPGAs, PLDs, and other programmable logic chips (e.g., dynamic reconfiguration, memory, and HDLs)

Robotics: Projects about robot systems, devices capable of repeating motion sequences, and MCU-based motor control designs (e.g., mobile robots, motor drives, proximity sensing, power control, navigation, and accelerometers)

Signal Processing: Projects and technology related to the real-time processing of signals (e.g., DSP chips, signal conditioning, ADCs/DACs, filters, and comparisons of RISC, DSP, VLIW, etc.)

Wireless Communications: Technology and methods for going wireless (e.g., radio modems, Wi-Fi/IEEE 802.11x, Bluetooth, ZigBee/IEEE 802.15.4, cellular, infrared/IrDA, and MCU-based wireless security applications)

UPCOMING IN CIRCUIT CELLAR

FEATURES

Digital Camera Controller (Part 2), by Richard Lord

Debugging USB Firmware, by Jan Axelson

GPS-Based Timeserver, by Scott Weber

DIY Automated Component Dispenser, by Thiadmer Riemersma

COLUMNS

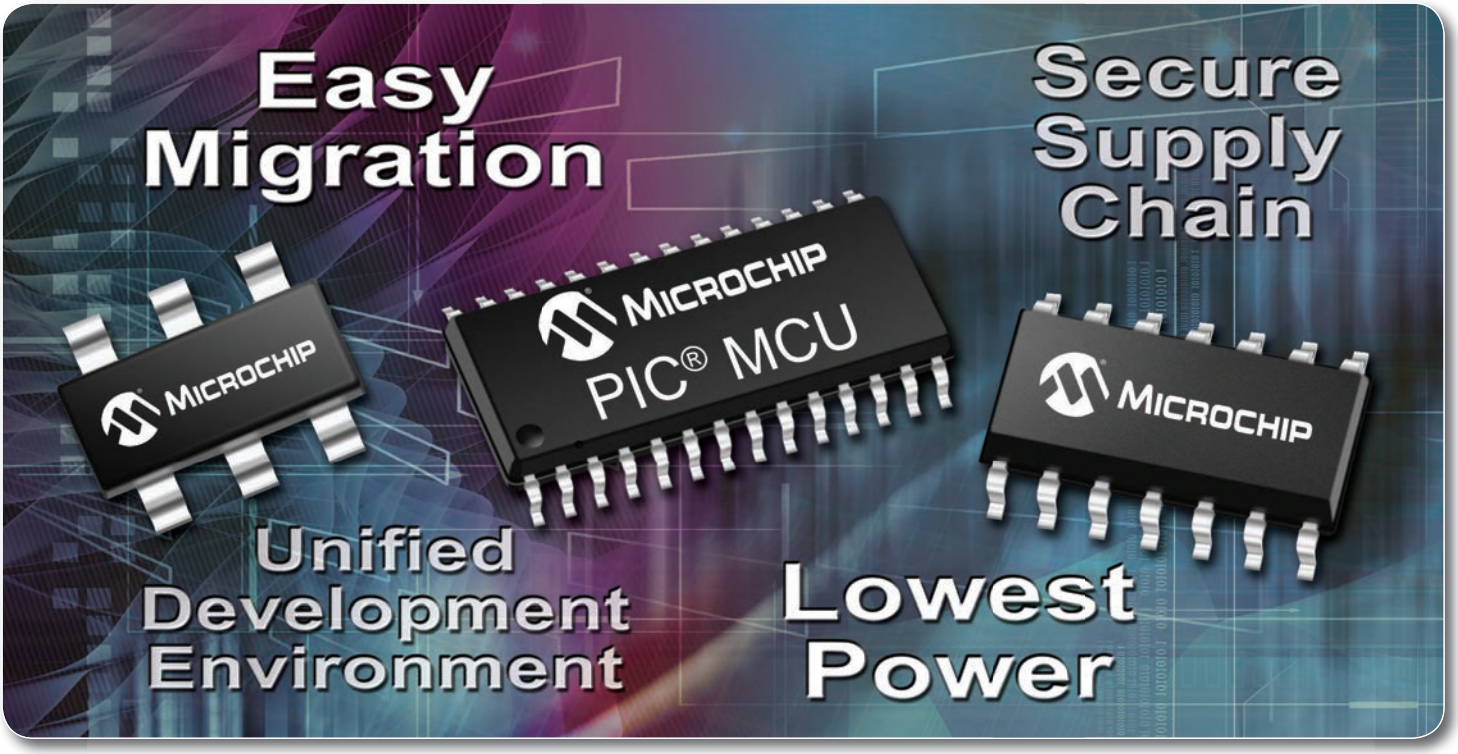
Mechanical Gyroscope Replacement (Part 2), by Jeff Bachiochi

Tachometer Design, by George Martin

Product Reliability, by George Novacek

True Random Number Generation, by Patrick Schaumont

The Most Popular 8-bit Microcontrollers



See what makes Microchip the most popular choice for embedded designers:

- Broad portfolio of more than 325 8-bit PIC® microcontrollers
- Easy migration with pin and code compatibility
- Industry's lowest active and sleep power consumption
- Integrated peripherals for USB, CAN and Ethernet with free software
- Interface to the world with LCD drivers and capacitive touch
- Continuous innovation with 70 new 8-bit MCUs in the last two years
- MPLAB® IDE is free, and supports ALL of Microchip's microcontrollers
- Low-cost development tools help speed up prototyping efforts
- Comprehensive technical documentation, app notes and code examples
- World-class 24/7 technical support and training

GET STARTED IN 3 EASY STEPS

1. Purchase the XLP 8-bit Development Board
2. Download free MPLAB® IDE
3. Order samples and start designing!
   www.microchip.com/8bit



XLP 8-bit Development Board
(DM240313)

MICROCHIP

Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

The Microchip name and logo, the Microchip logo, dsPIC, MPLAB and PIC are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
©2012 Microchip Technology Inc. All rights reserved. 3/12

INSIDE ISSUE 267

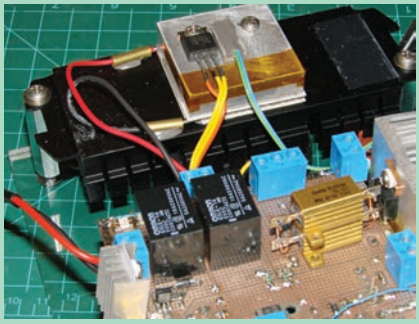
October 2012 • Signal Processing

16 **Math Routines Using dsPIC Features**
Stuart Oliver

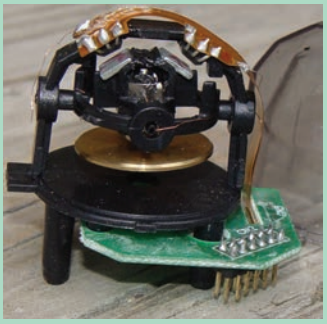
20 **Digital Camera Controller (Part 1)**
Hardware & Construction
Richard Lord

30 **FAT Cache**
Using File Allocation Table (FAT) Cache for Fast and Efficient SD Card Access
Kerry Imming

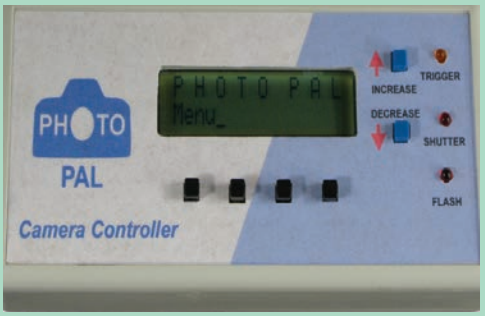
MOSFET Tester & Temperature Control p. 42



MEMS Solution Stabilizes a Gyro p. 68



DIY MCU-Based Controller for High-Speed Flash Photography p. 20





34 **THE CONSUMMATE ENGINEER**
Managing Project Risk
George Novacek

38 **EMBEDDED IN THIN SLICES**
Concurrency in Embedded Systems (Part 3)
Avoiding Concurrency Problems
Bob Japenga

42 **ABOVE THE GROUND PLANE**
MOSFET Tester
Peltier Temperature Control and Results
Ed Nisley

58 **THE DARKER SIDE**
Introduction to IIR Digital Filters
Robert Lacoste

68 **FROM THE BENCH**
Mechanical Gyroscope Replacement (Part 1)
A Microelectromechanical Systems (MEMS) Solution
Jeff Bachiochi

TASK MANAGER 2
Continuity of Content
C. J. Abate

NEW PRODUCT NEWS 10

MEMBER PROFILE 14

TEST YOUR EQ SOLUTIONS 15

QUESTIONS & ANSWERS 54
Embedded Inquiries
An Interview with Hai (Helen) Li
Nan Price

CROSSWORD 76

PRIORITY INTERRUPT 80
Go Team!
Steve Ciarcia


mouser.com
Distributing semiconductors and electronic components for design engineers.
Authorized Distributor
Texas Instruments
MAXIM
INNOVATION DELIVERED
AVAGO TECHNOLOGIES
CREE
FAIRCHILD SEMICONDUCTOR
infineon
NXP
VISHAY
ST
life.augmented
CYPRESS
PERFORM
freescale
Atmel
SILICON LABS
MICROCHIP
ON Semiconductor
ON
PLX TECHNOLOGY
THE I/O INTERCONNECT SOLUTION
LATTICE SEMICONDUCTOR
mouser.com
GO Faster.
The Newest Products for Your Newest Designs®
MOUSER ELECTRONICS
a tti company
Mouser and Mouser Electronics are registered trademarks of Mouser Electronics, Inc. Other products, logos, and company names mentioned herein, may be trademarks of their respective owners.

THE TEAM

FOUNDER/EDITORIAL DIRECTOR:	Steve Ciarcia	PROJECT EDITORS:	Ken Davidson, David Tweed
EDITOR-IN-CHIEF:	C. J. Abate	PUBLISHER:	Hugo Van haecke
ASSOCIATE EDITOR:	Nan Price	ASSOCIATE PUBLISHER:	Shannon Barraclough
CONTRIBUTING EDITORS:	Jeff Bachiochi, Bob Japenga,	ART DIRECTOR:	KC Prescott
	Robert Lacoste, George Martin,	CONTROLLER:	Jeff Yanco
	Ed Nisley, George Novacek,	CUSTOMER SERVICE:	Debbie Lavoie
	Patrick Schaumont	ADVERTISING COORDINATOR:	Kim Hopkins

THE NETWORK

audioXPRESS

elektor

CIRCUIT CELLAR
THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

techthe.future
Tech the Future explores the solutions for a sustainable future provided by technology, creativity and science.

VOICE COIL

OUR INTERNATIONAL TEAMS

United Kingdom
Wisse Hettinga
+31 (0)46 4389428
w.hettinga@elektor.com

USA
Hugo Van haecke
+1 860 875 2199
h.vanhaecke@elektor.com

Germany
Ferdinand te Walvaart
+49 (0)241 88 909-0
f.tewalvaart@elektor.de

France
Denis Meyer
+31 (0)46 4389435
d.meyer@elektor.fr

Netherlands
Harry Baggen
+31 (0)46 4389429
h.baggen@elektor.nl

Spain
Eduardo Corral
+34 91 101 93 85
e.corral@elektor.es

Italy
Maurizio del Corso
+39 2.66504755
m.delcorso@inware.it

Sweden
Wisse Hettinga
+31 (0)46 4389428
w.hettinga@elektor.com

Brazil
João Martins
+351214131600
joao.martins@editorialbolina.com

Portugal
João Martins
+351214131600
joao.martins@editorialbolina.com

India
Sunil D. Malekar
+91 9833168815
ts@elektor.in

Russia
Nataliya Melnikova
8 10 7 (965) 395 33 36
nataliya-m-larionova@yandex.ru

Turkey
Zeynep Köksal
+90 532 277 48 26
zkoksal@beti.com.tr

South Africa
Johan Dijk
+27 78 2330 694 / +31 6 109 31 926
J.Dijk@elektor.com

China
Cees Baay
+86 (0)21 6445 2811
CeesBaay@gmail.com

Issue 267 October 2012 ISSN 1528-0608

CIRCUIT CELLAR® (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 $50, Canada $65, Foreign/ROW $75. All subscription orders payable in U.S. funds only via Visa, MasterCard, international postal money order, or check drawn on U.S. bank.

Cover photography by Chris Rakoczy—www.rakoczyphoto.com

Subscriptions
Circuit Cellar, P.O. Box 462256, Escondido, CA 92046
E-mail: circuitcellar@pcspublink.com
Phone: 800.269.6301, Internet: www.circuitcellar.com
Address Changes/Problems: circuitcellar@pcspublink.com

Postmaster: Send address changes to Circuit Cellar, P.O. Box 462256, Escondido, CA 92046.

US Advertising
Strategic Media Marketing, Inc.
2 Main Street, Gloucester, MA 01930 USA
Phone: 978.281.7708, Fax: 978.281.7706, E-mail: peter@smmarketing.us
Internet: www.circuitcellar.com
Advertising rates and terms available on request.

New Products: New Products, Circuit Cellar, 4 Park Street, Vernon, CT 06066, E-mail: newproducts@circuitcellar.com

MEMBERSHIP COUNTER

We now have  members in  countries.

Not a member yet?

Sign up at www.circuitcellar.com

SUPPORTING COMPANIES

2012 International CES 51
All Electronics Corp. 78
AP Circuits 35
ARM. 64
ARM TechCon 2012. 47
Beta Layout, Ltd. 57
BusBoard Prototype Systems. 78
Circuit Cellar 25th Anniversary USB 41, 75
Cleverscope. 57
Comfile Technology 37
Custom Computer Services 79
Electronica 2012 1
Elektor 52, 53
Elektor 73
Elprotronic, Inc. 74
Elsevier, Inc. 15
EMAC, Inc. 35
ExpressPCB. 67
Flexipanel, Ltd. 79
Front Panel Express 65
FTDI Chip. C3
Grid Connect, Inc. 65
Holtek Semiconductor, Inc. 27
Humandata, Ltd. 40
Imagineering, Inc. C4
Ironwood Electronics 78
Jameco Electronics 23
Jeffrey Kerr, LLC. 61
JK microsystems, Inc. 13, 77
Labcenter Electronics 45
Lakeview Reasearch 61
LinkSprite 79
MaxBotix, Inc. 79
MCC, Micro Computer Control 77
Microchip Technology, Inc. 3
Microengineering Labs, Inc. 78
Mosaic Industries, Inc. 78
Mouser Electronics, Inc. 5
NetBurner C2
Newark element14 29
PCBMAIN Technology Co., Ltd. 77
Pico Technology, Ltd. 33
Pololu Corp. 19
Reach Technology, Inc. 77
Saelig Co., Inc. 62
SiliconRay 13
SoC Conference 55
Technologic Systems 8, 9
Tern, Inc. 77
Triangle Research International, Inc. 79
UltraTek 61

Not a supporting company yet?

Contact Peter Wostrel (peter@smmarketing.us, Phone 978.281.7708, Fax 978.281.7706) to reserve your own space for the next issue of our member's magazine.

Head Office
Circuit Cellar, Inc.
4 Park Street, Vernon, CT 06066, Phone: 860.875.2199

Copyright Notice
Entire contents copyright © 2012 by Circuit Cellar, Inc. 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.

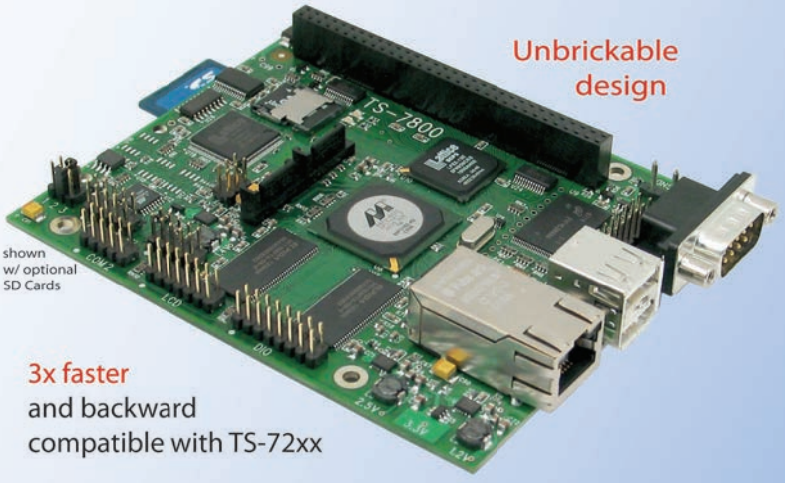
Disclaimer
Circuit Cellar® makes no warranties and assumes no responsibility or liability of any kind for errors in these programs or schematics or for the consequences of any such errors. Furthermore, because of possible variation in the quality and condition of materials and workmanship of reader-assembled 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.

© Circuit Cellar 2012 Printed in the United States

Embedded Systems

High-End Performance with Embedded Ruggedness



Unbrickable design

3x faster and backward compatible with TS-72xx

TS-7800
500MHz ARM9

- Low power - 4W@5V
- 128MB DDR RAM
- 512MB high-speed (17MB/sec) onboard Flash
- 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 0.7 seconds
- Linux 2.6 and Debian by default

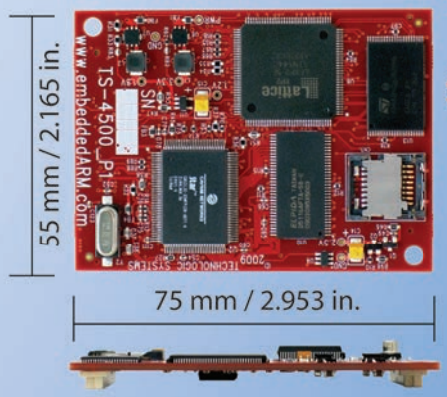
$229 qty 100

$269 qty 1

TS-SOCKET Macrocontrollers
Jump Start Your Embedded System Design

TS-SOCKET Macrocontrollers are CPU core modules that securely connect to a baseboard using the TS-SOCKET connector standard. COTS baseboards are available or design a baseboard for a custom solution with drastically reduced design time and complexity. Start your embedded system around a TS-SOCKET Macrocontroller to reduce your overall project risk and accelerate time to market. Current TS-SOCKET products include:

- TS-4200: Atmel ARM9 with super low power
- TS-4300: 600MHz ARM9 and 25K LUT FPGA
- TS-4500: Cavium ARM9 at very low cost
- TS-4700: 800MHz Marvell ARM with video
- TS-4800: 800MHz Freescale iMX515 with video
- Several COTS baseboards for evaluation & development



- Dual 100-pin connectors
- Secure connection w/ mounting holes
- Common pin-out interface
- Low profile w/ 6mm spacing

series starts at

$92 qty 100

$139 qty 1

- Over 25 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

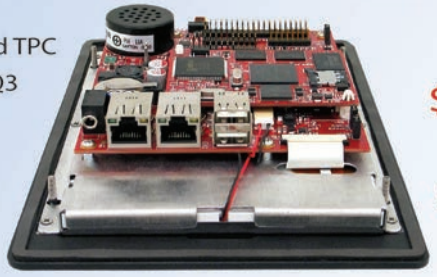
Touch Panel Computers
800MHz with Video Acceleration

- Resistive touchscreen, LED backlit display
- Gasketed construction
- Tough powder coated finish
- Fanless operation from -20°C to +70°C
- 800MHz ARM CPU
- 256MB RAM, 256MB SLC XNAND Drive
- MicroSD slot
- 5K LUT programmable FPGA
- Dual Ethernet, USB ports
- CAN, RS-232 ports, RS-485
- Mono speaker on PCB, stereo audio jack
- SPI, DIO



Fully enclosed TPC
available Q3

NEW!



series
starts at
$415
qty 100

$479
qty 1



picture of TS-8820-BOX

NEW!

series
starts at
$199
qty 100

$229
qty 1

Technologic Systems now offers three powerful computers targeting industrial process control. Implement an intelligent automation system at low cost with a minimal number of components.

Industrial Controllers
Powerful, Rugged, Affordable

- 250MHz (ARM9) or 800MHz (ARM9 or Cortex-A8) CPU
- Fast startup (under 3 seconds)
- Fanless operation from -20°C to +70°C
- User-programmable opencore FPGA
- Program in Ladder Logic or C
- Debian Linux
- Modbus support
- PoE capable 10/100 Ethernet, USB 2.0 Host Ports
- Industrial screw-down connectors
- Opto-Isolated DIO, Digital Counters, Quadrature
- Up to 46 DIO with PWM
- Opto-Isolation available for RS-232, RS-485 and CAN
- DIN mount option

Technologic Systems

We use our stuff.

Visit our TS-7800 powered website at

www.embeddedARM.com

cPCI BLADE FEATURES EIGHT-CORE INTEL XEON PROCESSOR

The **cPCI-6930** is a high-performance, next-generation, 6U dual-slot CompactPCI (cPCI) processor blade with flexible expansion capabilities. The blade features a 1.8-GHz, eight-core Intel Xeon processor E5-2448L, an Intel C604 chipset, and DDR3-1333 ECC registered memory up to 48-GB capacity. On-card connectivity includes four front-panel gigabit Ethernet (GbE) egress ports, a 4-GB soldered SATA NAND flash, an optional bootable CompactFlash socket, and an optional SAS/SATA direct connector for an onboard 2.5" drive to provide flexible storage capabilities. Rear I/O connectivity includes up to three SATA ports and up to eight SAS ports supporting hardware RAID via two mini-SAS connectors. The cPCI-6930 is well suited for server-grade applications, providing telecom and datacom system integrators, robust computer makers, and security sectors with a flexible, cost-effective solution for mission-critical applications with a reliable, smooth path for scalability and expansion.



The cPCI-6930 uses Intel Hyper-Threading Technology to efficiently utilize processor resources. This enables multiple threads to run on each core to increase processor throughput, which improves overall performance on threaded software. Intel Turbo Boost Technology enables the cPCI-6930 to run faster than the base operating frequency if it is operating below power, current, and temperature specification limits, maximizing performance and energy efficiency.

An Intel 82580EB GbE controller enables high-speed data transfer via four front-panel GbE ports on the cPCI-6930. Two additional GbE ports are routed to the J3 connector to support PICMG 2.16. Additional onboard I/O includes one VGA port, two USB 3.0 ports, three USB 2.0 ports, and one RS-232 serial port. The cPCI-6930 also provides a PMC/XMC expansion site supporting 64-bit, 133-MHz PCI modules or PCI Express x8 XMC modules.

Contact ADLINK for pricing.

ADLINK Technology, Inc.
www.adlink.com

RFID LABEL WITH MICROPROCESSOR

The latest version of ZeitControl's **BasicCard ZC7.5** is now available with a 13.56-MHz RFID prototyping board. The BasicCard is a BASIC-programmable smart card that is well suited for most smart card applications (e.g., identification card, medical card, Internet security, driver's license network access, software key, access control, and gift and loyalty programs).

The BasicCard development environment enables the creation of highly secure applications within a few hours. For security purposes, BasicCard supports Data Encryption Standard (DES), TripleDES, and Advanced Encryption Standard (AES). The latest version features 4,096-bit RSA or 512-bit elliptic curve cryptography (ECC).

BasicCard ZC7.5 is available in a variety of shapes and easily attaches to most surfaces. It is compatible with ISO14443A, comes with 16-KB E²PROM, and can work as a near-field communication (NFC) label.

Contact ZeitControl for pricing.

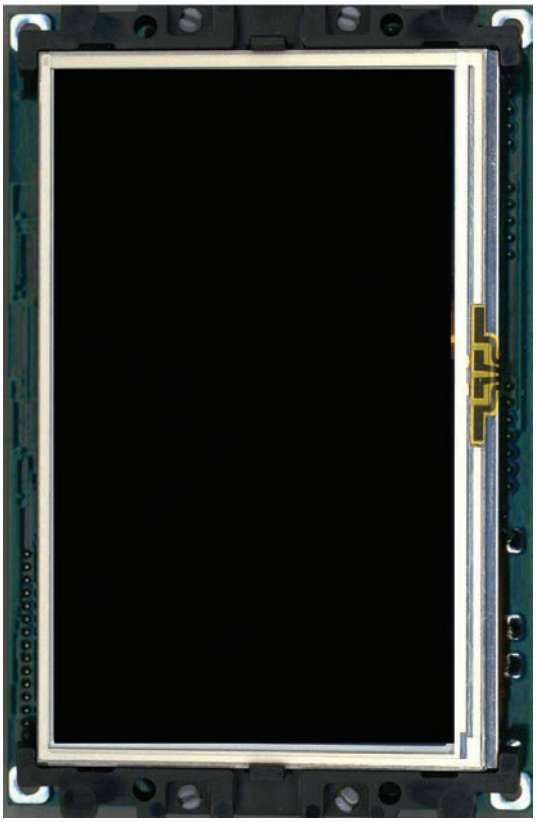


ZeitControl cardsystems
www.zeitcontrol.de

ULTRA-COMPACT PANEL PC WITH RESISTIVE-TOUCH SCREEN

The **PPC-E4+** is an ultra-compact panel PC with a 4.3" WQVGA (480 × 272) TFT color LCD and a four-wire resistive-touch screen. The PPC-E4+'s dimensions are 3" × 4.8", which is approximately the same size as a touch cell phone.

The PPC-E4+ comes with either Windows CE 6.0 or EMAC's embedded Linux distribution installed and fully configured on the on-board flash memory. Once power is applied, the PPC-E4+ user interface appears on a vivid color LCD screen. The PPC-E4+ compact panel PC utilizes a system on module (SoM) for the processing core, enabling users to easily upgrade to a faster SoM if more processing power, memory capacity, or storage are required. The panel PC uses an embedded ARM 9-based SoM that features a 400-MHz fanless low-power processor with video and touch. The PPC-E4+ supports up to 256 MB of SDRAM, up to 1 GB of flash memory, and up to 8 MB of serial data flash memory. Typical power consumption is less than 5 W. To further lower power consumption, the device's LED backlight turns off when not in use.



It has three RS-232 serial ports and one RS-232/422/485 port. The product also includes: one SPI port, one I^2C port, one USB 2.0 (high-speed) host port, one USB 2.0 (high-speed) OTG port, an audio beeper, and a battery-backed real-time clock. Additional flash memory storage is provided by two microSD flash card sockets. You can connect the PPC-E4+ to a network using the 10/100 Base-T Ethernet controller and its on-board RJ-45 connector. Optional power over Ethernet (PoE) and audio with line in/out is also available.

EMAC provides a free Eclipse IDE for Linux development. All the compiling, linking, downloading, and debugging built into software development can be easily accomplished from one high-level interface. When developing for Microsoft Windows CE 6.0 applications, you can use Microsoft Visual Studio 2005/2008. Both Visual Studio and Eclipse provide everything needed for developing PPC-E4+ applications. EMAC provides an SDK for the PPC-E4+, which contains source examples and drivers.

The PPC-E4+ costs **$375**.

EMAC, Inc.
www.emacinc.com

FULLY INTEGRATED CONTROL SYSTEM USES APPLE'S iOS 5 DEVICES

Compass Control is a fully integrated major control system designed to use Apple's iOS 5 devices (e.g., iPad, iPhone, and iPod touch) as its foundation. Compass Control is flexible and can be used for small or large systems and in residential or commercial applications.



Compass Control is capable of controlling devices via RS-232, IR, or TCP/IP. The control system provides installers with bidirectional IR, two-way RS-232 with feedback, and two-way TCP/IP control for unlimited flexibility and maximum compatibility with new or existing devices. Relays and sensors can also be used for control.

Compass Navigator software must be used to program the Compass Control system. The software programs Compass Control devices through an intuitive, step-by-step process utilizing simple drag-and-drop gestures. Compass Navigator provides installers with three separate database managers to make programming faster than traditional control systems. It also enables installers to edit command code sets.

Compass Control's base system includes a KD-MC2500 master controller, which has up to eight IR or RS-232 ports and powerful TCP/IP/LAN networking control support. The KD-MC2500 enables IR learning and features two sets of switchable contacts for relay control of external equipment.

Contact Key Digital for pricing.

Key Digital Systems
www.keydigital.com

INSTRUMENTATION AMPLIFIER WITH ZERO-DRIFT ACCURACY

The **AD8237** micropower, zero-drift precision instrumentation amplifier provides a power-efficient solution for precision signal and sensor conditioning. The amplifier features a low-input offset drift (less than 0.3 µV/ºC) and an input signal range that extends 300 mV beyond the supply rails. Combined with a maximum gain error and drift of 0.005% and 0.5 PPM/°C, respectively, and 114-dB common-mode rejection ratio (CMRR), the device offers precise measurement, even at low gain.



With a low-quiescent supply current of just over 100 µA, the amplifier is well suited for signal conditioning bridge, temperature, and similar sensors used in low-power designs. Target applications include portable and battery-powered healthcare and consumer and precision instrumentation equipment (e.g., infusion pumps, mobile blood pressure monitors, 4-to-20-mA sensors, gaming peripherals, and consumer weigh scales).

The AD8237 operates from a single supply voltage of 1.8 to 5.5 V and ships in a compact MSOP-8 package. It functions at a –40°-to-125° temperature range. The gain (1 to 1,000) is set by the ratio of two external resistors, which, combined with gain-error performance, provides superior gain accuracy over temperature. The amplifier operates well in a system with a variety of SAR ACDs (e.g., the AD7988-1 ultra-low-power 16-bit SAR converter).

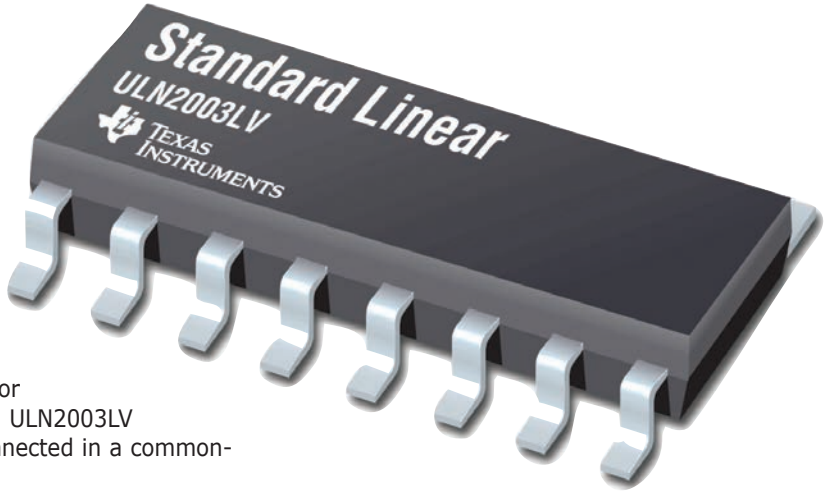
The AD837 costs **$0.93** in 1,000-unit quantities.

Analog Devices, Inc.
www.analog.com

LOW-POWER 3.3- AND 5-V RELAY DRIVER

The **ULN2003LV** is a low-voltage, low-power upgrade to TI's ULN2003 family of seven-channel Darlington transistor arrays. The ULN2003LV is a fully integrated, seven-channel relay driver capable of supporting low-voltage relays from 1.8 to 5 V. It replaces a handful of discrete components with a single device to drive lower voltages for data relays and rails within telecommunications equipment. The ULN2003LV integrated circuit (IC) can be used to reduce board space by up to 80% while lowering power consumption and cost.

The ULN2003LV sink driver features seven low-output impedance drivers to support low-voltage relay and inductive coil applications. The low-impedance drivers minimize on-chip power dissipation, making it up to five times lower for typical 3-V relays.



The ULN2003LV driver is pin-to-pin compatible with the ULN2003 family of devices in similar packages. It supports a 3.3-to-5-V CMOS logic input interface, making it compatible with a range of microcontrollers and other logic interfaces.

The ULN2003LV's additional features include: an improved input interface that minimizes the input DC current drawn from the external drivers, an input RC snubber that improves performance in noisy operating conditions, channel inputs that feature an internal input pull-down resistor enabling input logic to be tri-stated, and support for other logic input levels (e.g., TTL and 1.8 V). Each ULN2003LV output features an internal freewheeling diode connected in a common-cathode configuration at the COM pin.

Under typical conditions and with all seven channels connected in parallel, the ULN2003LV can increase current sink capability and support up to 1 A of load current. The device can also be used in a variety of other sink driver applications.

The ULN2003LV is available in 4-mm × 10-mm, 16-pin SOIC and 4.5-mm × 5.1-mm, 16-pin SOIC and 16-pin TSSOP packages. The drivers cost **$0.15** in 100-unit quantities.

Texas Instruments, Inc.
www.ti.com

HIGH-PRECISION OVEN-CONTROLLED CRYSTAL OSCILLATOR

The **GPS1300-10-1000** is a 10-MHz, high-precision GPS disciplined oven-controlled crystal oscillator (OCXO) that measures only 10.5 mm × 32.5 mm × 40 mm. The oscillator is well suited for base stations, wireless networks, system synchronization systems, instrumentation, critical timing and calibration applications, and precision portable reference units.

The precision OCXO uses control algorithms to regulate to the onboard satellite Caesium standard. If the GPS signal is temporarily lost, the module provides holdover through the internal OCXO, which retains its previously acquired control settings until the lock is restored.

The OCXO's features include: 1-PPS low-voltage complementary metal oxide semiconductor (LVCMOS), long-term stability 1 × 10^{-12} PPM, short-term stability 1 × 10^{-11} PPM, excellent phase noise, low aging, and NMEA datastream.

Contact Lemos for pricing.



Lemos International Co., Inc.
www.lemosint.com

NPN

SiliconRay
Online Electronics Store
• Boards • Kits • Modules • Components • Tools • Instruments
FOR
• Companies • Professionals • Students • DIYers • Amateurs

Mini multimeter DT-830B
Free, Portable, Multifunction
$0 Add to cart

RE4015 Aluminum Enclosure
Extremely Well Built, Ideal For Class A
$220 Add to cart

USB AMR JTAG Simulation J-Link v8
"Must have" tool for ARM developers
$25 Add to cart

SA-22 HiFi Tube Amplifier
High-End Diy kit For Serious Amateurs
$282 Add to cart

Simplify your electronics projects by visiting
http://www.siliconray.com

What's your flavor Linux or DOS?

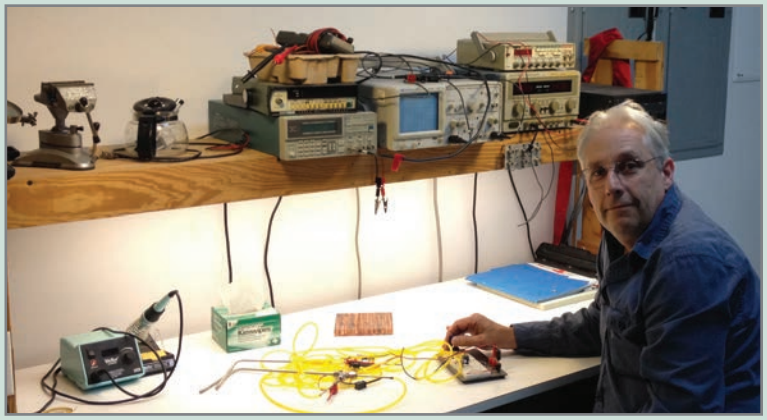
OmniFlash
Linux Kernel 2.4 Installed
16MB FLASH / 32MB RAM
200 Mhz Arm9 CPU
16 Digital I/O Lines
10/100 Base –T Ethernet
2 USB and 2 Serial Ports
Hardware Clock Calendar
Watchdog and Audio In/Out
5V DC Power
$169.00
$129.00 Qty 100

Flashlite 186
Preloaded with DOS & Flash File System
33MHz 186 Compatible Processor
512K Flash, 512K DRAM
32 Pin DIP Socket
44 Digital I/O Lines
2 Serial Ports
Console / Debug Port
Watchdog & (2) 16 bit Timers
7-34V DC or 5V Power
$83.00
$66.00 Qty 100

Made in the USA - Distributed Worldwide

JK microsystems, Inc.
530-297-6073 www.jkmicro.com sales@jkmicro.com

MEMBER PROFILE: Carl Stoesz



Member Name: Carl Stoesz

Location: Blacksburg, VA

Education: BS, University of Houston

Occupation: Mechanical Engineer

Member Status: Carl has been a subscriber for five years.

Technical Interests: Carl's interests include fiber optics, wind energy, and energy scavenging. He describes energy scavenging as projects that utilize energy from the environment to create electricity that keeps a device going.

Most Recent Embedded Tech-Related Purchase: Carl recently acquired a Microchip Technology MPLAB ICD 2 real-time debugger and programmer, which he says is great for the older, smaller PICs. "I eventually need to upgrade to the ICD 3," he said.

Current and Recent Projects: Carl said he is not working on anything at the moment. He is in limbo waiting for something to "strike my fancy."

Thoughts on the Future of Embedded Technology: "This technology will continue to grow by leaps and bounds. I realized this when I picked up my first musical get well card," Carl said. "It won't be long until the term 'embedded technology' means biologically embedded," he added.



Edited by David Tweed

Test Your EQ

ANSWERS for Issue 266

Answer 1—An infinite impulse response (IIR) filter incorporates feedback in its datapath, which means that any particular input sample can affect the output for an indefinite (infinite) time into the future. In contrast, a finite impulse response (FIR) filter uses only feedforward in its datapath, which means any given input sample can only affect the output for a time corresponding to the number of storage (delay) stages in the filter.

Answer 2—While it's technically true that the impulse response of an IIR filter implemented, say, with fixed-point arithmetic is effectively finite, this has no real bearing on its classification in terms of either its design or application. It's still an IIR filter for all practical purposes.

Answer 3—The accumulator in this filter eventually settles at the value `0xFFFE8100`. If you simply take the upper 16 bits of this, then the output value appears to be `0xFFFE`. But if you properly round the accumulator by adding `0x00008000` before dropping the LSBs, then the output value is the correct value of `0xFFFF`.

Answer 4—Changing the `$offset` value to 0xFF effectively adds a bias to each input sample, which averages out to `0x00007F00` in the accumulator. The effect of this is to add the required rounding offset to the accumulator so truncating the LSBs to create the 16-bit output value comes up with the correct answer.

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



by Stuart Oliver (Australia)

Math Routines Using dsPIC Features

Small embedded systems frequently require real-number math routines. Good performance and accuracy can be achieved at a low cost using small math routines and the dsPIC hardware features. This article describes some routines that use the accumulators and barrel shifter.

Using math in small systems can be a little awkward. You may only need to calculate and display a few numbers, but accomplishing that is not simple. Using a high-level language and compiler is one method, but the available packages are often intended for a different use, leading to increased resource requirements that are at odds with the project scope. The language used may be optimal for a different purpose, or the routines may be so incorporated into the integrated development environment (IDE) that they are not easy to use outside of that environment.

Standard packages are constructed to process (and store) as efficiently as possible, so they offer the classical integer/float datatypes and the associated methods and tradeoffs. These methods are sometimes used outside their intended purposes. In fact, single-precision floating point (i.e., using 8 bits for the exponent and the remaining 23 or 24 bits for the significand) is only accurate to about seven decimal digits (i.e., it starts to fall over at the parts per million level), which can be inconvenient. Using double precision is usually overkill and can impact performance.

As "lightweight" math would be useful (that's a good excuse!), I decided to look at writing some suitable Assembler routines for my own purposes. The scope was to enable a few variables to be used for a moderate amount of calculation, maintaining accuracy in a small code size.

USING A 64-BIT DATA FIELD

For tasks that only require small amounts of math, the emphasis on efficiency can be relaxed. For a few variables and modest speed requirements, using four words for each variable won't be a problem. By centering the radix—the "decimal point"—the 2^{64} item space then enables a variable to have a scope of 2^{32} integers, with 2^{32} fractional quanta apiece.

In essence, this uses a very large fixed-point technique to emulate real numbers. It can also be thought of as using a composite data type that encompasses integer and fraction fields at the expense of some efficiency. Of course, if strict conformance to IEEE standards using the ordained protocols is legally required, it is probably best to keep to the standard commercial methods and packages available. On the other hand, with a small-scale project, an expedient method and set of routines can be sufficient.

If you are using Assembler, some small routines are all that is necessary to implement basic number handling and math operations. The required routines are: the conversion from ASCII text and decimals to binary, the math operations (e.g., add, subtract, multiply, and divide), and the conversion back to decimal and ASCII text form. The summarized routines shown in the listings provide a small but convenient math capability.

Note: The method and routines in the listings use the idea of a composite data type and make profligate use of (i.e., shamelessly waste) memory. The dsPIC elegant signal processing facilities are used for almost superficial purposes. People with acute mathematical sensitivities may need to seek a quiet, restful place somewhere.

My routines use four words to hold a numerical value. The first two words are used as a binary integer. The last two words hold an appended binary fraction, as necessary. Routine binary math is used to process the combined entity, usually without having to use separate handling. Descriptions of the methods and routines follow in their natural order of use.

Listing 1—The two operands' least-significant words, `vxaccA4` and `vxaccB4`, are copied to the W4 and W5 working registers and added.

```
mov inpAdeclo, W4          ;
mov inpBdeclo, W5
add W4, W5, W0
mov W0, outptdeclo

mov inpAdechi, W4          ;
mov inpBdechi, W5
addc W4, W5, W0
mov W0, outptdechi

.
.
```

Listing 2—The subtraction is very similar to the addition, as shown in Listing 1, but instead uses `sub` and `subb`.

```
mov inpAdeclo, W4          ;
mov inpBdeclo, W5
sub W4, W5, W0
mov W0, outptdeclo

mov inpAdechi, W4          ;
mov inpBdechi, W5
subb W4, W5, W0
mov W0, outptdechi

.
.
```

Listing 3—The hardware multiplier is used for the 64-bit multiplication by processing the 16-bit word combinations. Here is an example of one combination.

```
mov inpBdechi, W0;
mov inpAdechi, W2;
; unsigned multiply ; output is in W3,W2
mul.uu W0, W2, W2
mov W3, ACCBH
mov W2, ACCBL
add A

.
.
```

INPUT CONVERSION

The input is a sequence of ASCII numeric characters, which are first converted to BCD. The integer binary is generated by scanning through the input integer digits and summing the relevant multiples of powers of 10. (This is something of a standard function, so I won't cover it further. To learn more, search online for "asc2bin" or "dec2bin.") Similarly, the fraction binary is generated by scanning through the input decimal fraction digits and summing the corresponding binary fractions.

Carefully considered mapping is used to enable the fraction conversion. Projecting the decimal fraction system onto the 2^{32}-item maximum leads to a useful range of 1 billion quanta, which maps well to nine decimal places. So, to construct the binary fraction, the decimal fraction is deconstructed and the base binary number's corresponding parts are summed (e.g., 0.1 would translate to a 10^{th} of 2^{32}). An accumulator is used to sum the amounts as they are generated.

The binary equivalents are usually not exact and can introduce quantization and truncation errors. Each part's last bit has inexactitude, which is also added. For nine or so fraction digits, this could amount to (pessimistically) eight or nine least-significant bits in the summed binary fraction. To reduce the error, the base binary figure used is upscaled before the fractional parts are tabled. After the summed result is available, it is divided down by the prior factor to bring it back to size and a nominal truncation correction is applied. As a consequence, the final error is divided and very reduced.

To implement this, the accumulators' 40-bit mode is used, which provides convenient headroom of 7 bits. Accordingly, a factor of 128 is used for upscaling (i.e., it uses a 2^{39} base number). The barrel shifter is used to divide the result by 128 in one cycle by shifting the accumulator contents seven places right. The possible error is then reduced to a fraction of the least-significant bit.

MATH OPERATIONS

Now that the pain is over, standard machine instructions can be used to easily add, subtract, multiply, and divide the binary numbers. Starting with the least-significant words, the two operands are added. The instruction `add` is used for the least-significant word pair. Thereafter `addc` is used, which obligingly includes the carry bit, which is set up by the previous addition. Other commands can be used instead (e.g., to pipeline, etc.), but keeping it as simple as possible is a good idea at this stage.

In Listing 1, the two operands' least-significant words, `vxaccA4` and `vxaccB4`, are copied to the W4 and W5 working registers. The addition takes place and the result is moved to `outptdeclo`, which is the answer's least-significant word. The process is then appropriately repeated for the other words, using `addc`. The subtraction is similar to the addition but uses `sub` and `subb`, respectively (see Listing 2).

MULTIPLICATION & DIVISION

Do you remember learning long multiplication? Binary multiplication is formally a summation process. For each bit that is true in the multiplier, the number to multiply is multiplied by the appropriate power of two and added to the running total. If this was used as a method, it would involve a large number of cycles and be slow. In practice, the dsPIC hardware multiplier uses an accelerated method to provide a worthwhile speed improvement.

There are four words in each number to be multiplied. Multiplying (A + B + C + D) times (a + b + c + d) provides 16 combinations to sum for the result with the smallest product, Dd, being insignificant and subsequently ignored. The multiplication consists of the iterations of sequentially loading the input registers for the hardware multiplier; the multiply itself; and using the accumulators to sum the results, using a barrel shifter, as necessary, to shift the accumulating result. The results of the largest words' multiplications will be overrange, if not zero, so these are calculated to catch overflow errors. Overall, using unoptimized instructions provides a nominal 64-bit multiplication in a small number of instructions.

Listing 3 shows one of the combinations. W0 and W2 are set up to hold the two words to multiply, the multiplication is carried out, and the two-word result is returned in W2 and W3. The result is transferred to accumulator B. The two accumulators, A

and B, are added with accumulator A holding the running total. For more efficiency, there are other commands that pipeline the transfers and so forth.

The division process is basically the inverse of multiplication. It is useful to think of the required answer as being a sequence of bits. Each bit specifies a multiple of the divisor, which has been summed to make up the number being divided. So, for each bit considered in the answer field, the divisor is multiplied by the appropriate power of two and compared to the number being divided. If it is equal or greater, the bit is set and the multiplied divisor is subtracted from the number being divided. The process is repeated for the next bit, and so on, until the smallest limit of precision is reached or there is no remainder. Unfortunately, division cannot use the assumptions made in multiplication, and subtraction using the accumulators is not as straightforward as addition, so the full process is carried out.

The routines for division set up two large shift registers to hold the operands and the previous process is followed. Auxiliary routines carry out the multiplication by powers of two (by left shifting), comparison, and subtraction. The code is quite large in this case, but the fundamental process is shown in Listing 4. The first line calls the comparison of the shift registers, which sets the carry bit if the comparison result was true. If it was set, the subtraction takes place (followed by setting the answer bit, not shown). Otherwise, the process bypasses it and moves to the next answer bit.

Listing 4—For division, each result bit requires a comparison and possible subtraction.

```
.

call divcompvxa
btss SR, #C
goto divbitz
call vxasub
.
.
```

Listing 5—This is an example of one of the steps needed to convert back to decimal and ASCII.

```
SFTAC B, #1                ; halve it
btsc fracdechi, #15
add A
.
.
```

CONVERSION BACK TO DECIMAL & ASCII

This is basically the inverse of the previously described conversion. The integer binary is used to add and sum the appropriate decimal quantities (search online for "bin2dec" and similar). The fraction binary is used to add and sum the appropriate binary fractions of the large decimal number that corresponds to the range of expected decimal point digits, and again the accumulators' extended range is used to assist in dividing out errors. The fractions are generated by simply halving the previous number at each iteration, by using the barrel-shift instruction to shift the accumulator content one place right. This time, the number used is 400 billion, which is divided by four to provide a field corresponding to 100 billion. As there are 32 additions involved, the total error may be 32 parts in 400 billion, a comfortable fit with the intended scope of eight or nine decimal digits in the fraction. Lastly, the decimal number is deconstructed into BCDs and finally ASCII, incorporating rounding before the final display.

The example in Listing 5 is one of the steps. The first line shows accumulator B's current contents being divided by two by shifting one place. The second line shows bit 15 in one of the fraction words being tested. If the bit is set, the next line adds the accumulators summing to accumulator A; otherwise, the next line is bypassed.

Note: It is typical to include a capability for processing negative numbers by using the two's complement method, which uses the most-significant bit in a binary number as a sign bit. This efficiently yields much of the sign logic needed and is useful for rapid processing (as in compiled code). In this scope, the rapidity is not as important. Although the sign bit is still used in storage, for math operations, a sign bit is included in the variables' information flags instead and appropriately used. This simplifies the use of the accumulators and barrel shifter.

Listing 6—Here are some specific results that can be compared with other systems.

```
3.141592654              pi (input as the ascii and then output)
9.869604404              pi squared
0.142857143              1/7, a test of ppm accuracy and more
1.333820449              Infamous Pentium division :  4195835 / 3145727

31415.9265               10000 multiplied by pi
31415.9275               0.0010 added to 10000 multiplied by pi
```

And a table of calculated sin and cos values, x = input.

```
x = 0.000000 sin = 0.0000 cos = 1.0000
x = 0.100000 sin = 0.0998 cos = 0.9950
x = 0.200000 sin = 0.1987 cos = 0.9801
x = 0.300000 sin = 0.2955 cos = 0.9553
x = 0.400000 sin = 0.3894 cos = 0.9211
x = 0.500000 sin = 0.4794 cos = 0.8776
x = 0.600000 sin = 0.5646 cos = 0.8253
x = 0.700000 sin = 0.6442 cos = 0.7648
x = 0.800000 sin = 0.7174 cos = 0.6967
x = 0.900000 sin = 0.7833 cos = 0.6216
x = 1.000000 sin = 0.8415 cos = 0.5403
x = 1.100000 sin = 0.8912 cos = 0.4536
x = 1.200000 sin = 0.9320 cos = 0.3624
x = 1.300000 sin = 0.9635 cos = 0.2675
x = 1.400000 sin = 0.9854 cos = 0.1700
x = 1.500000 sin = 0.9974 cos = 0.0708
x = 1.570000 sin = 0.9998 cos = 0.0008
```

TESTING & RESULTS

As something of a test, lists of numerical examples (chosen to represent the possible word combinations in operands) are used as inputs. The results are gratifying. Multiplication showed a small error in the last word, which was tracked down to a typographic Assembly code error and fixed. Answers accurate to eight significant places are produced, with accuracy to the ninth place commonplace, and the code size remains very small.

In addition, first-quarter sin and cos derivations were implemented using the arithmetic to calculate the Taylor series approximations. The sine and cosine results are accurate to the limits of the series used. Listing 6 includes some token, but specific, results you can compare to other systems

FUTURE IMPROVEMENTS

Using 64-bit centered-radix math routines enables a convenient small math routine that encompasses integer and fractional qualities. There are other advantages for interpreter applications, and the resulting code size is quite small. Of course, homebrew routines are not meant to compare with Microchip's formidable math routines, but it is satisfying to see the results. If I was to do it again, I would probably use the two's complement process, if possible. It would be good to try the full table of fraction equivalents and see if the last figure's accuracy improved at all. But it would use more program space. There are also extreme-throughput DSP-based commands that concurrently operate and transfer data and results, which will wait for a future chapter. In the meantime, there are the prodigious capabilities of the dsPIC architecture and Microchip products (e.g., dsPIC30F/33F digital signal controllers and the dsPIC33FJ128GP*xx* microcontroller family) to explore further.

If you wish to use it, a small (28-Kb program size) dsPIC-based interpreter using these methods and routines is available on *Circuit Cellar*'s FTP site. It is free for use for green initiatives, easing unemployment, or in countries that are within sight of the Pacific Ocean.

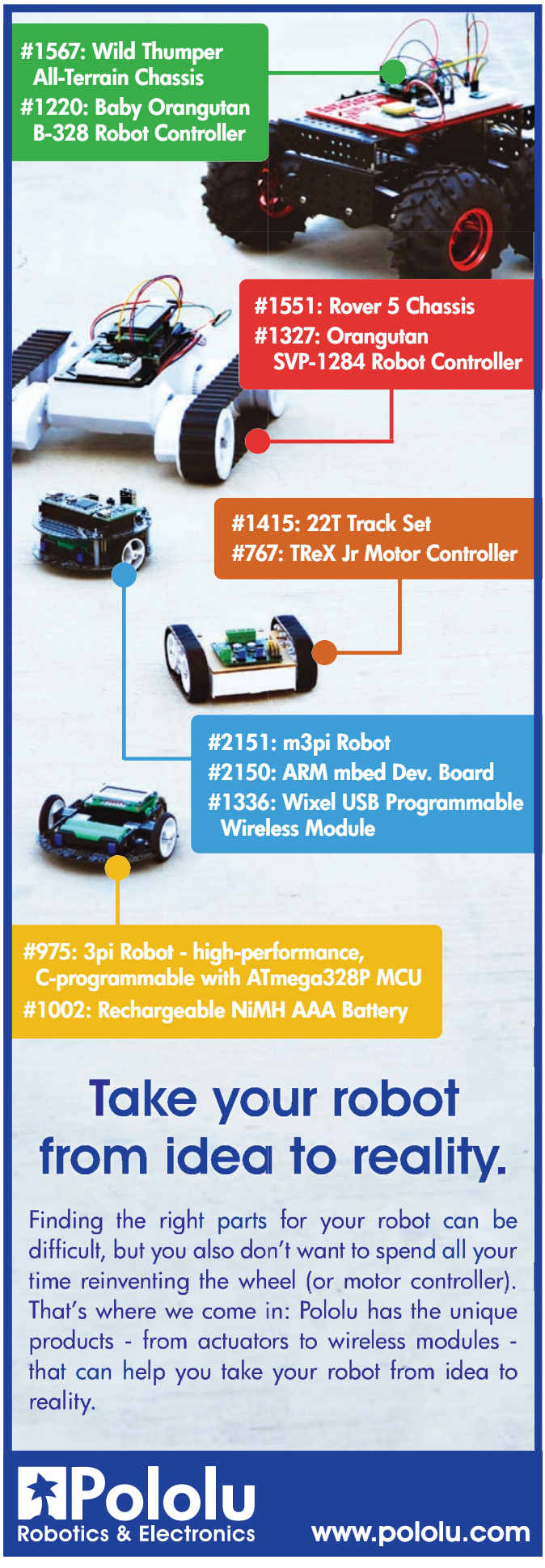
Stuart Oliver (themaracot@gmail.com) started programming at the age of 12 using punched cards and an ancient educational Fortran system. He began his career combining science and computing. After completing a PhD and spending some years working in chemistry, Stuart changed direction and began IT and programming, gaining languages and DBA-level accreditation in Oracle RDBMS systems. Over the last few years, Stuart has taken up Assembler, using dsPICs as a hobby. He tries to catch as much of the Sydney sunlight as possible.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/267.

SOURCES

dsPIC30F/33F Digital signal controller and dsPic33FJ128GP*xx* microcontroller
Microchip Technology, Inc. | www.microchip.com



by Richard Lord (USA)

Digital Camera Controller (Part 1)

Hardware & Construction

The Photo-Pal is a microcontroller-based electronic flash-trigger camera controller that features four modes of operation: triggered shutter, triggered flash, multiple flash, and time lapse. Now you too can take sound-triggered photos.

Back in the 1930s, Harold Edgerton, an MIT graduate student, invented something that changed the world. Edgerton's invention was the electronic flash, which is today incorporated into many cameras and safety warning devices. More than 80 years after its conception, this invention is known throughout the planet and annual sales of products incorporating Edgerton's ideas still exceed the combined sales of cell phones, smartphones, tablets, and personal computers.

Edgerton, affectionately known as "Doc Flash," wasn't content to have developed something that replaced flashbulbs. He realized the extremely brief flashes from his xenon tubes could be used to take clear, sharp, still photographs of fast-moving objects in a way that had never before been seen. Over the next decade, he managed to capture hundreds of iconic images that still amaze us today, such as bullets ripping through apples and playing cards, balloons bursting, and droplets of splashing milk.

We are still fascinated with the ability to freeze motion. An online search of "high-speed flash" turns up dozens of modern variations of the images Edgerton produced. These pictures use the sound made by a gun or the passage of a milk droplet through a photodetector to start a short delay and trigger the flash at just the right moment to capture the desired image.

Over the years, there have been many articles in various hobbyist magazines showing how to build simple gadgets that could trigger an electronic flash from a microphone, but most of these have had imprecise means of setting the delay. None of the existing sound-trigger products offered the flexibility I needed. Being an inveterate gadget freak who never passes up the opportunity to put a microcontroller to work, it was inevitable that I would build my own flash-trigger device. What I came up with has features that make it useful for other photographic tasks as well, so I named it "Photo-Pal" instead of "the flash-trigger gadget" (see Photo 1).



Photo 1—The Photo-Pal is a Microchip Technology PIC16F873A microcontroller-based camera controller.

PHOTO-PAL

The Photo-Pal has four modes of operation. Each mode's setup connections are shown in Figure 1.

In Triggered Shutter (TS) mode, a sound at the microphone input (or a switch closure) can be used to trigger a camera's remote shutter input after an adjustable delay (see Figure 1a). You can use this mode to capture pictures of birds at the feeder, raccoons in your garbage can, or people passing through your photocell-equipped door. In addition to the delay parameter, you can set the length of time the Photo-Pal must wait before it can take another picture as well as the maximum number of pictures that can be taken.

Triggered Flash (TF) mode uses the Photo-Pal's trigger input to activate an external electronic flash unit instead of the camera shutter (see Figure 1b). TF mode is used to capture breaking light bulbs and bouncing liquid droplets. In this mode, the camera shutter is activated by an arming switch input, which then enables the trigger input. Once triggered, the Photo-Pal waits for the programmed delay time and triggers the electronic

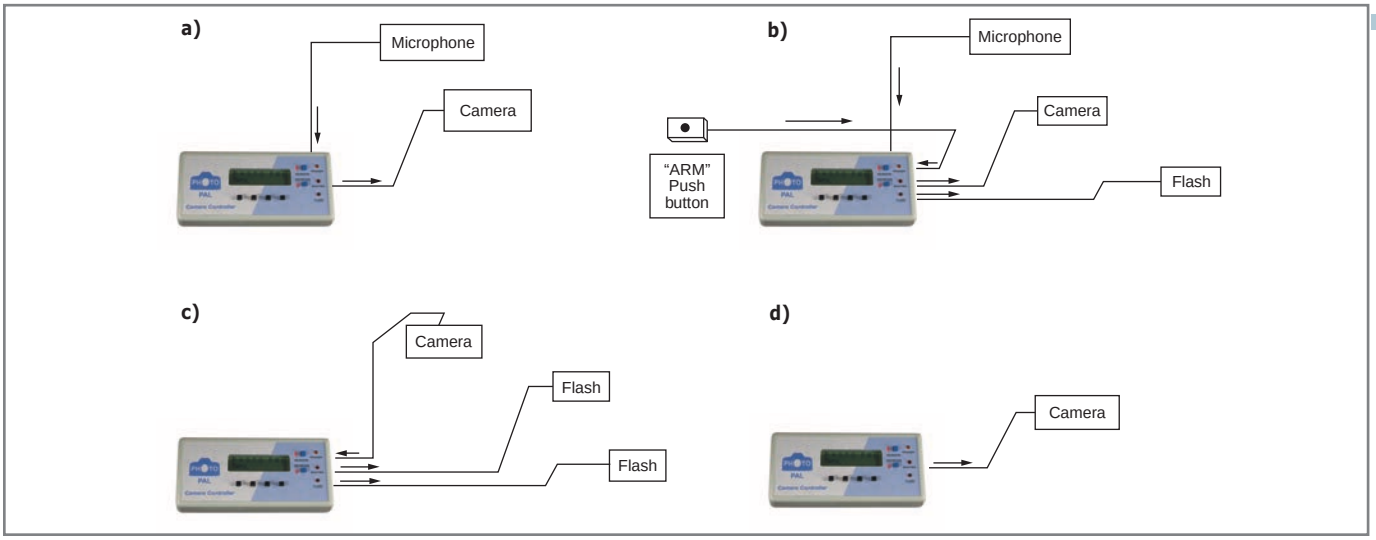


Figure 1—The Photo-Pal has four modes of operation: triggered shutter **(a)**, triggered flash **(b)**, multiple flash **(c)**, and time lapse **(d)**.

flash. After the flash has fired, the camera shutter is deactivated. In this mode, an optional setting enables the delay to be more precisely set to a value between 100 and 19,999 µs instead of the usual 0-to-59,999-ms setting.

Multiple Flash (MF) mode creates a burst of flashes when triggered by a sound, the camera's flash-trigger output, or a switch closure (see Figure 1c). It takes advantage of the flash unit's ability to be retriggered when the output power is reduced. In MF mode, you set the initial delay, the interval between flashes, and the total number of flashes. Photo-Pal doesn't control the camera shutter in this mode, so the shutter output jack is available to trigger additional flash units to enable a larger or faster burst of flashes. The Photo-Pal can utilize each output jack's tip and ring connection to sequentially trigger up to four flash units.

In Time Lapse (TL) mode, the Photo-Pal activates the camera shutter periodically after waiting a programmed interval of time (see Figure 1d). You can specify the interval anywhere from 1 s to 99 h, 59 min., and 59 s. You can also specify how many pictures will be taken.

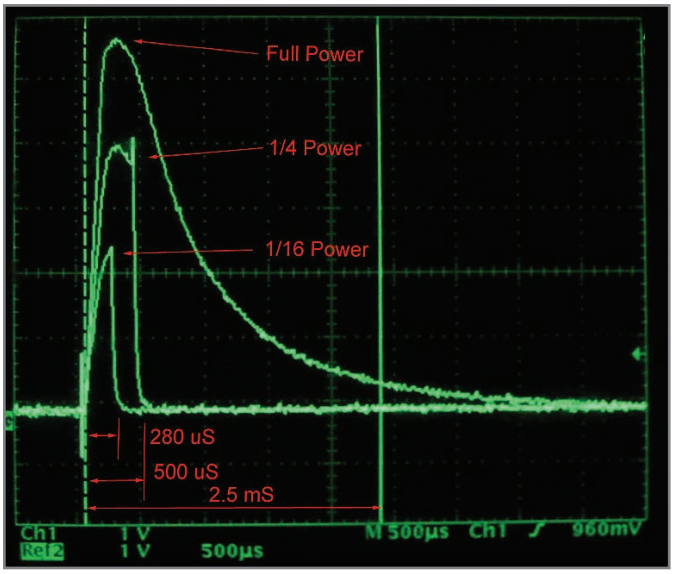


Photo 2—This graph shows a comparison of light output versus time on my Canon 430EX-II Speedlite.

FREEZING THE ACTION

Electronic flash units work by storing a high voltage (typically around 300 to 600 V) in an electrolytic capacitor. A xenon-filled flash tube is connected across this capacitor. A trigger circuit ionizes the xenon gas, causing it to suddenly conduct and discharge the capacitor. The current flowing through the xenon causes it to produce a brilliant white light.

Many modern photographic flash units communicate with the camera through its "hot shoe" and can adjust the flash's output energy to create the proper exposure. They work by switching off the flash tube current once the camera detects there is enough light for the picture. Some flash units can also be manually set to deliver a fraction of the full power. My Canon 430EX-II Speedlite can be adjusted down to 1/64 of full output power. This early turn off leaves energy in the capacitor and shortens the length of time needed to recharge it between flashes. If there is sufficient charge left in the capacitor, the flash can be retriggered to produce multiple flashes.

Thanks to this early turn off, lowering the power setting also shortens a picture's exposure time. At full power, my Speedlite's flash output lasts for about 2.5 ms or a 1/400 of a second. At 1/64 power, the flash lasts for about 1/8,000 of a second or about 125 µs (see Photo 2). Something traveling at 100 ft/s will have traveled 3" during the full power flash, but about 0.15" during the low-powered flash. You can see that a quick-acting flash is important when trying to get clear, non-blurry photographs of exploding light bulbs and bursting balloons.

TAKING THE PICTURE

Look at how I used the Photo-Pal to take a high-speed photograph of an incandescent light bulb shattering as it was struck by a hammer. The microphone is placed somewhere near the target light bulb and connected to the Photo-Pal's trigger input. The gain is adjusted until the yellow Trigger LED only momentarily flashes when it detects a moderately loud sound. A remote push button connected to the Photo-Pal's Arm input is a



Photo 3—Using a bolt through a hammer's handle, I set up a hammer so it would swing in a fixed arc and strike the center of a light bulb. The hand-held flash, which is on the leftside of the photo, was set at 1/64 power.

convenience, but you can use the Photo-Pal's Arm button if you don't have a remote button.

The electronic flash unit is positioned so it provides good illumination of the target object and is connected to the Photo-Pal's flash output. Next, the camera is placed on a tripod and manually focused on the target. The camera's remote shutter input is plugged into the Photo-Pal's shutter output and the camera's shutter is set to "bulb" so the shutter stays open as long as it is activated by the Photo-Pal.

You can use a camera that doesn't have an external shutter input if the shutter can be manually set to take exposures lasting several seconds. You have to work fast to get the picture while the shutter is open, but it can be done. It's just a lot easier if you can let the Photo-Pal's shutter output help you.

Most modern cameras have double-acting Shutter buttons. When you press the button halfway down, the camera sets the exposure and activates the autofocus and image stabilization. Pressing the Shutter button the rest of the way takes the picture. The camera's remote shutter input provides both half-press and full-press switch connections. Turning off autofocus and image stabilization helps, but there is still an unpredictable delay if you only use the full-press contact to trigger the shutter. To provide a predictable response, the Photo-Pal asserts the half-press output, waits for a programmable shutter pre-assert delay time, and then asserts the full-press output.

While I wouldn't normally keep you in the dark about how the next part works, that is what it takes to capture a photo of an exploding light bulb. The picture has to be taken with the lights off so the only illumination comes from the flash. Since I was working in the dark, I rigged a hammer to pivot (using a bolt through the handle) so it would swing in a fixed arc and strike the center of the light bulb (see Photo 3). Tiny shards

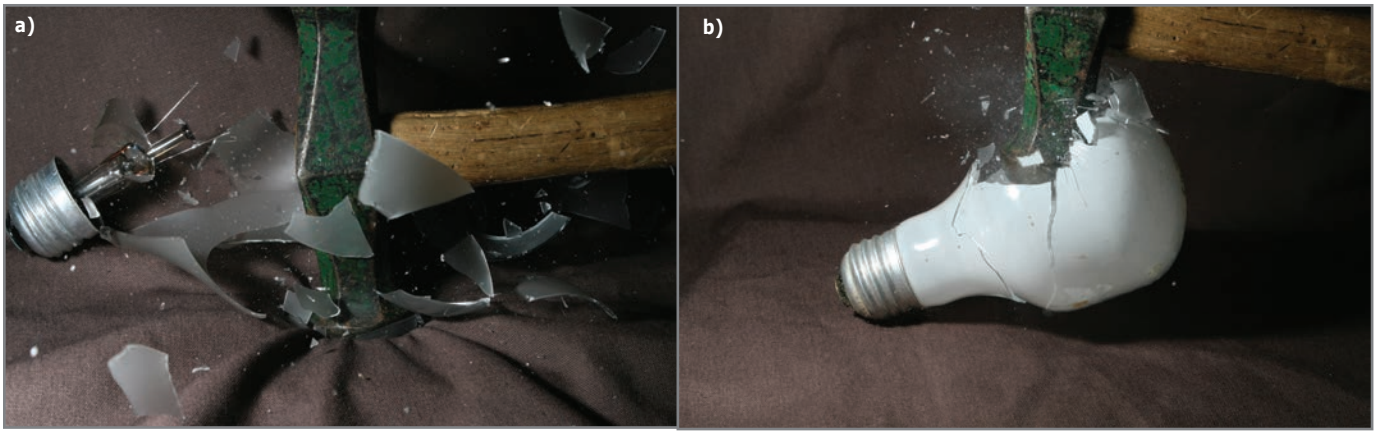
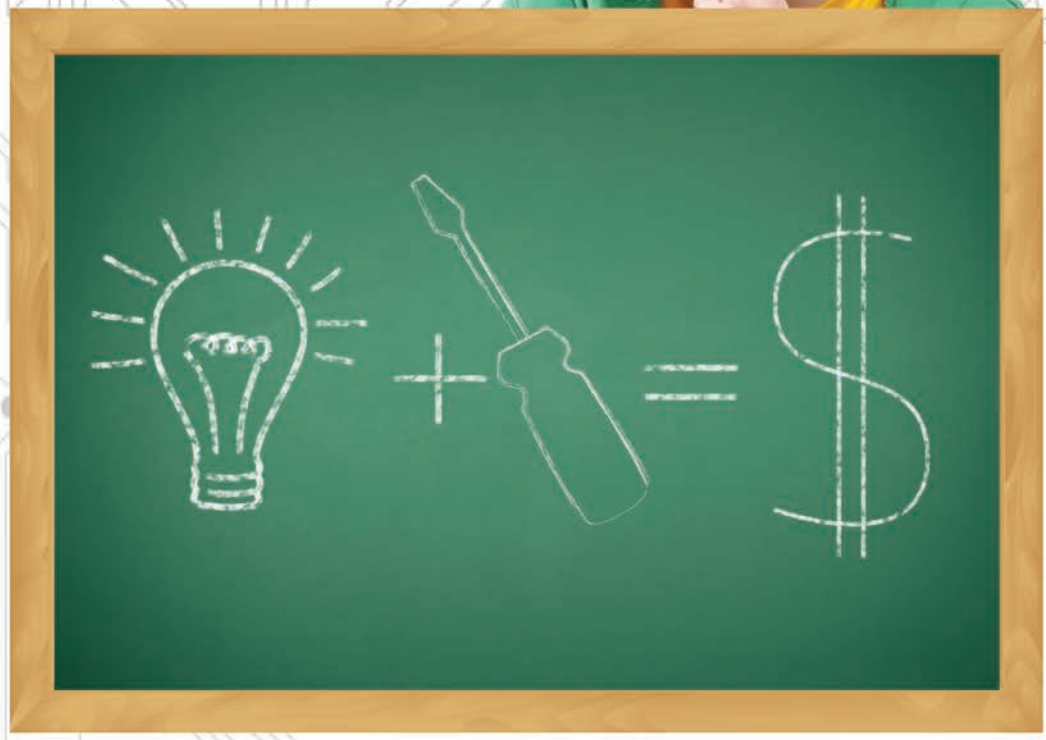


Photo 4a—Using the Photo-Pal, I set the delay at 100-ms, which was too late to capture the bulb explosion. **b—**This time, I set the delay to 10 ms.

Imagine, Design, EARN!

club JAMECO®





You design an electronics project. We'll turn your bright ideas into cash!

HAVE A GREAT PROJECT IDEA?

Design an electronics project at ClubJameco.com, identify the components, write step-by-step instructions, and that's it. We'll do everything else!

Start Earning Now!
www.ClubJameco.com

Copyright © 2012, Jameco Electronics. All Rights Reserved. 1355 Shoreway Rd., Belmont, CA 94002 USA Info@ClubJameco.com

Parameter	Min	Max	Unit	Setup from	Used by	Definition
Delay	0	59,999	ms	TS, TF, MF	TS, TF, MF	Delay between trigger event and response
FlshHld	0	59,999	ms	SP	TF, MF	Duration of flash output
Max Pix	1	199	#	TS, TL	TS, TL	Maximum number of pictures to take
FlDelay	100	19,999	µs	TF if dly=µs	TF	Alternate microsecond delay: trigger to flash
FlWidth	20	19,999	µs	TF if dur=µs	TF	Alternate microsecond duration of flash output
Intervl	0	59,999	ms	MF	MF	Time between subsequent flash outputs
# Flashes	1	99	#	MF	MF	Number of flashes in multi-flash burst
# Units	1	4	#	MF	MF	Number of multi-flash outputs
ShtrPre	0	19,999	ms	SP	TS, TF, TL	Time between shutter half press and full press
ShtrHld	0	19,999	ms	SP	TS,TL	Duration of full-press shutter output
ShtrRel	0	59,999	ms	TS, SP	TS	Hold-off time after shutter release
Trig =	*	*	*	SP	TS, TF, MF	Trigger src = NO, NC Switch, audio in, mic + DC
Fl dly= dur=	**	**	**	SP	TF	Selects alternate microsecond delay and/or flash duration
Time=h:m:s	00:00:01	99:59:59	h:m:s	TL	TL	Time-lapse hours: minutes: seconds
Timer Cal	0	255		CALIB	TL	Millisecond adjustment for time-lapse calibration
Low Bat	0	1020	.01V	CALIB	boot-up	Threshold for "low battery" detection

*Four choices		** Four choices		TS = Triggered Shutter mode
Normally open switch		ms : ms		TF = Triggered Flash mode
Normally closed switch		ms : µs		MF = Multiple Flash mode
Audio in		µs : ms		TL = Time Lapse mode
Mic + DC bias		µs : µs		SP = System Parameters mode
				CALIB = Calibrate mode (accessed on power-up)

Table 1—Here is a list of the Photo-Pal's parameters, including their value ranges.

of glass will fly for several feet, so safety glasses and gloves should be worn to prevent injury. Once the lights are turned off, press the Arm button, which opens the camera's shutter and enables the trigger input. The Photo-Pal is triggered by the sound of the bursting bulb. After the programmed delay, the Photo-Pal will trigger the flash, capture the image, and deactivate the shutter. Then you can turn the lights back on and survey the damage.

In the age of energy awareness and compact fluorescent lamps (CFLs), it's hard to come up with a steady supply of burned-out incandescent bulbs. Since it's against my Yankee upbringing to smash perfectly good bulbs, it's important to test things out before smashing. You need to take a few nondestructive shots to ensure the focus is sharp and the aperture and ISO speed are set to get a proper flash exposure. This is done by shouting at the microphone to trigger the flash and reviewing the photo on the camera's view screen (one of the miracles of digital photography).

For my first bulb smash picture using Photo-Pal, I set the delay to 100 ms (see Photo 4a). While there were still shards flying through the air, this was a bit too late to capture the actual bulb explosion. However, the photo was sharp and well lit, so this was promising. For my next attempt, I set the delay to 10 ms (see Photo 4b). This picture was much more satisfying, showing the bulb just starting to explode. Obviously, it would be interesting to see what happens at 20 and 30 ms, but I had run out of nonfunctioning light bulbs.

The trigger source can be a switch closure or a photodetector's output instead of a microphone sound. The switch can be normally open and trigger the unit when it is closed, or it can be normally closed and trigger the unit when it opens. A photointerrupter works much better than sound when trying to capture a liquid droplet splashing off a surface. The Photo-Pal's long, precisely programmable delay can take into account the length of time needed for the droplet to fall from the point where it passes the photodetector to the moment it strikes the surface where it will be photographed.

MENU NAVIGATION

There are four software-defined push button keys beneath the Photo-Pal's display. Their functions change as you move through the display screens. The bottom line of the 2 × 16 character display is used to label the action performed by each key for the screen displayed. Unlabeled keys perform no action for that screen. At power-up, the screen first shows `[PHOTO-PAL]` on the top line and the software version on the bottom line. After 1 s, the bottom line displays `[MENU ]`, which shows that pressing the leftmost key brings you to the first of the menu screens. Pressing the left key causes the top line to read `[Trigger Shutter]`, while the display's bottom line shows `[NEXT SETUP RUN]`, which indicates that the first key will take you to the next menu item. The second key doesn't do anything

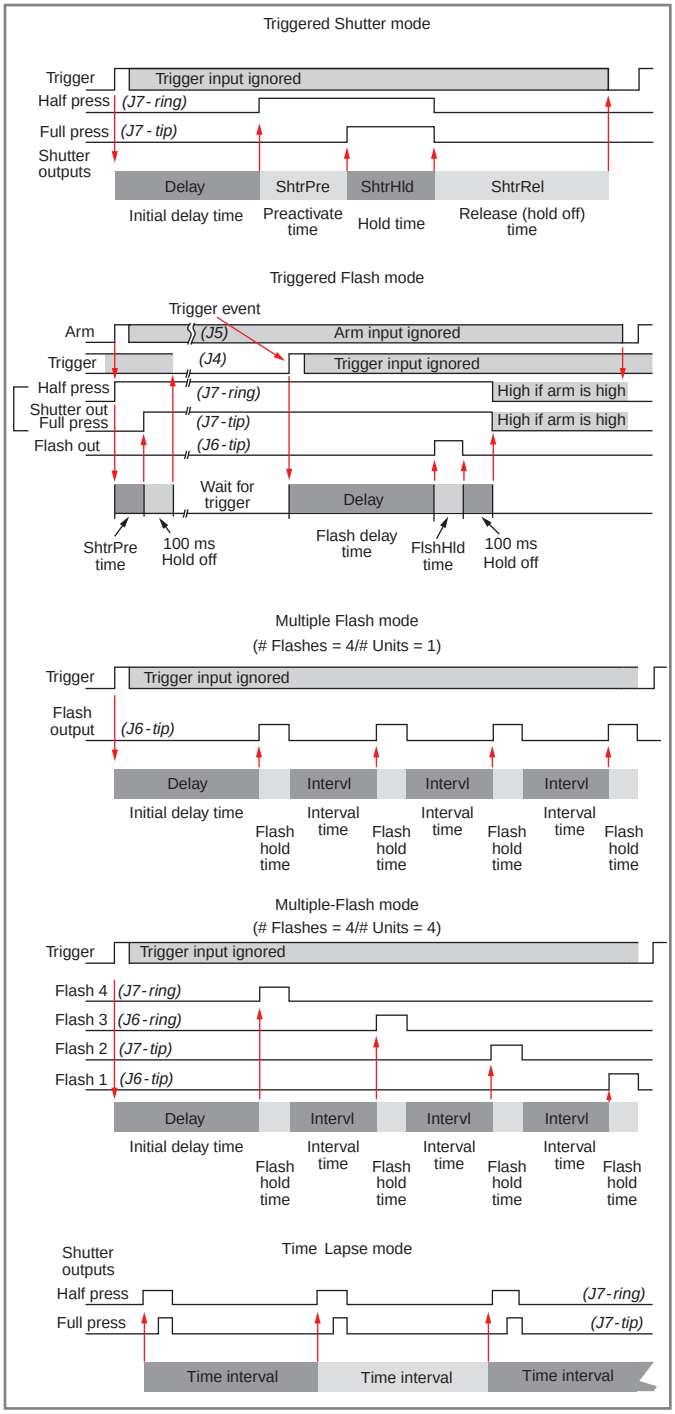


Figure 2—These timing diagrams show how the parameters apply for each mode of operation.

because it is not defined for this screen. The third key brings you to a setup screen where you can change the parameter values associated with the current mode. The fourth key starts the program running in the current mode.

Repeatedly pressing Next causes the screens to cycle through the four operating modes (TS, TF, MF, and TL) and a fifth screen, System Params, where global parameters can be altered. As you cycle through these main screens, the labels on the display's bottom line stay the same, except the Run key disappears for the System Params screen. Pressing the Run key for any of the other main menu screens starts the operation in the displayed mode. The bottom line changes to show the operation's current status on the left side of the line. The right side of the bottom line displays End, which indicates that the operation can be stopped by pressing the rightmost key.

Pressing the Setup key displays the name and value of a parameter associated with the current mode. For TS, TF, and MF, the screen displays the "Delay" parameter and its current value. For each parameter entry screen, the bottom line displays `[BACK <-  -> ENT]`, which shows that the leftmost key will take you back to the previous screen without altering anything. The rightmost key enters the current value of the parameter, stores it in EEPROM, and displays the next parameter, if any, or the main screen for that mode if the last parameter has been entered. Table 1 is a list of parameters and their value ranges. It also shows which operating modes apply for each parameter and which setup menus provide access to them. The timing diagrams in Figure 2 show how the parameters apply for each mode of operation.

The value entry screen's middle two keys (labeled "<-" and "->") move the data entry cursor left and right. At the cursor location, you can use the up and down arrow keys to the right of the display to increase or decrease the displayed parameter value. The value is increased or decreased once each time the key is pressed until it reaches its maximum or minimum value. If you hold either key down, the value will automatically increment or decrement repeatedly—after a 1-s pause—at a rate of 10 steps per second until you release the key. Incrementing by one may be fine for small numbers, but you would have to hold the Increase button down for more than 90 min. to get from 0 to 59,999. That's where the two cursor keys come into play. If you move the cursor to the left one position, the value increases and decreases by 10 for each step. Move it left again, and the hundreds digit now increments and decrements. Move it left again, and the value changes by 1,000 for each step. The value change occurs for the digit where the cursor is blinking. The cursor movement is always constrained to the number of significant digits valid for that parameter.

HARDWARE & CONSTRUCTION

The Photo-Pal's hardware design is fairly simple. At its heart is a Microchip Technology PIC16F873A microcontroller, which drives a 2 × 16 LCD via a 4-bit data interface using the upper 4 bits of the PIC16F873A's Port B (RB4–RB7), as shown in Figure 3. These same four pins are also used to read the state of the front-panel key switches by using the Port B soft pull-up feature when RB4–RB7 are defined as inputs in the Data Direction register. To read push buttons SW1–4, pin RC4 (U2–15) is brought low. If any of the four push buttons are pressed, the associated Port-B pin is brought low by the current flowing through the switch and isolation diode. To read push buttons SW5–6, pin RC4 is brought high and pin RC3 (U2–14) is brought low. When both RC3 and RC4 are high, the diodes isolate the switches so they don't interfere with the LCD operation.

The six push buttons and three status LEDs are mounted on a separate front-panel circuit board that connects to the main board through P2/J2. When a trigger input is detected, pin RA4

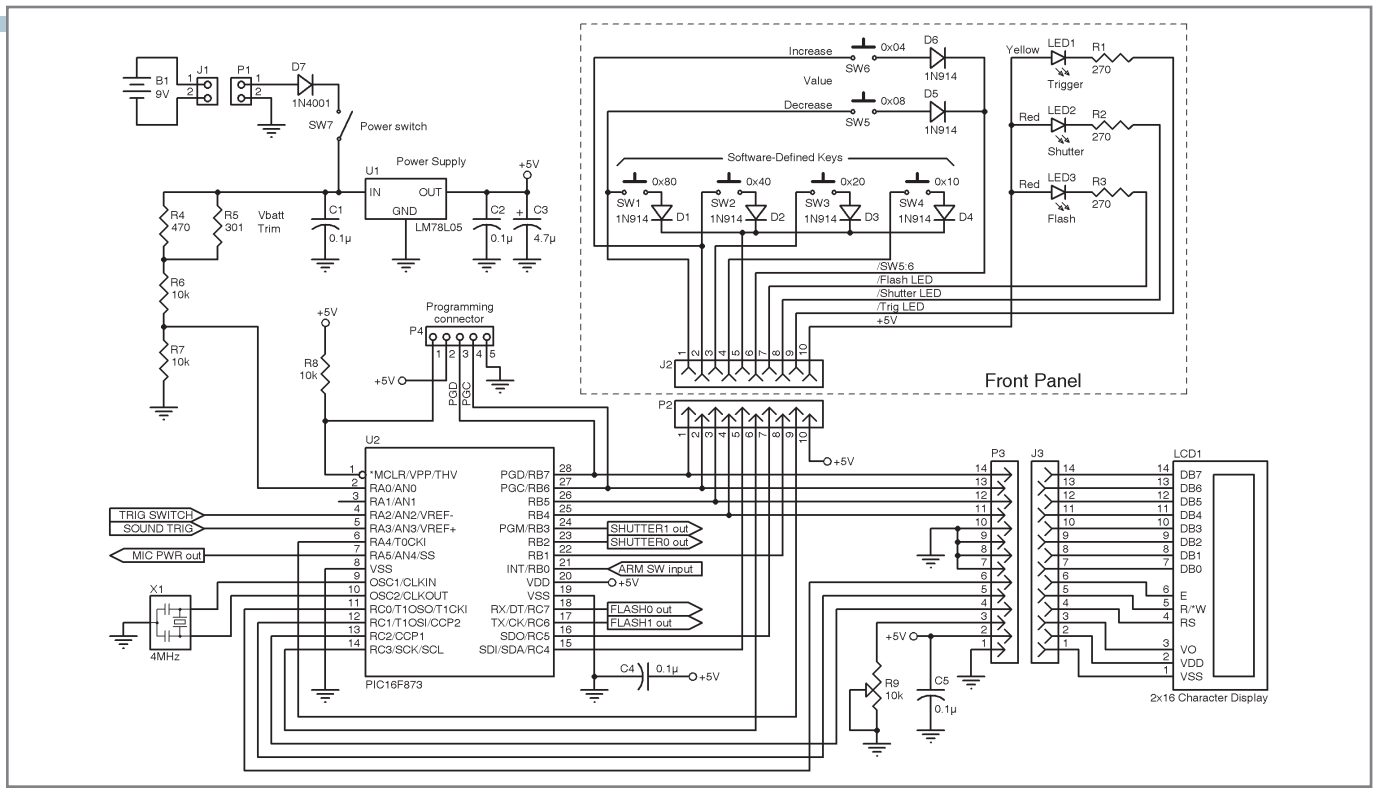


Figure 3—A Microchip Technology PIC16F873A microcontroller is at the Photo-Pal's core. The microcontroller drives a 2 × 16 LCD via a 4-bit data interface using the upper 4 bits of its Port B (RB4–RB7).

(U2–6) is driven low to illuminate yellow LED1. When the shutter output is activated, pin RB1 (U2–22) is driven low to illuminate LED2. During flash output, LED3 is illuminated by bringing pin RC5 (U2–16) low.

A 9-V battery plugged into P1 powers the Photo-Pal. Diode (D7) protects against accidental polarity reversal. With SW7 closed, power is supplied to the Texas Instruments LM78L05 regulator that supplies 5 V to the board. The input to the regulator also passes through a voltage divider (R4–R7) that enables the PIC16F873A to read the battery voltage using its

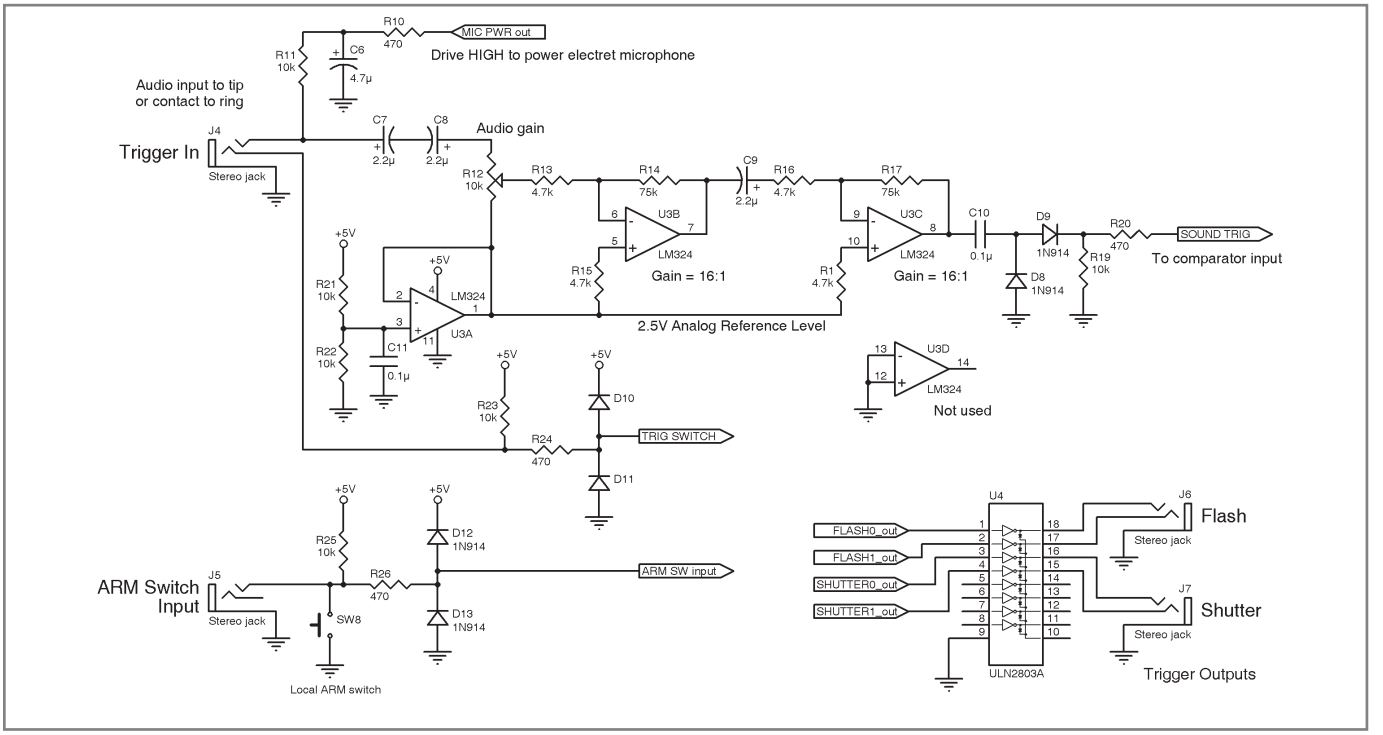
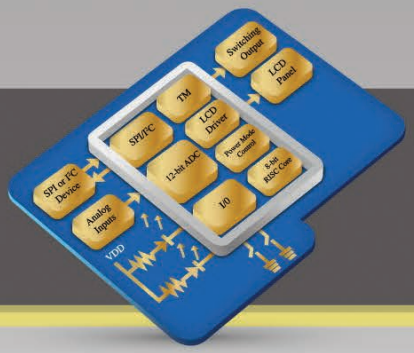


Figure 4—This is the audio amplifier and I/O section of my PIC16F873A-based Photo-Pal.

HOLTEK
www.holtek.com

All In One

Ultra Low Power Consumption - Highly Functionally Integrated Flash MCU Solutions



The Holtek Flash MCU range of devices encompass an extensive range of peripheral functions making them suitable for use in a diverse application area such as health care products, instrumentation products, household appliances, industrial control, consumer products, automotive peripheral products to name but a few. With one range types also including Holtek's unique TinyPower™ technology, the ability is provided to meet the demands of today's environmentally conscious products.

HT68Fxx	Internal Multi-function Timer Modules. Internal oscillators with four frequency selections. Multi-function I/O pins with re-mapping function. 4 LCD SCOM outputs for direct driving of LCD panels. Multiple Communication Interfaces.
HT66Fxx	Includes all the HT68Fxx functions with an additional 12-bit ADC and internal reference voltage source.
HT67Fxx	Includes all the HT66Fxx functions but also uses Holtek's TinyPower ultra low power technology and also includes an internal R-type/C-type LCD driver. Also includes dual SPI or I^2C interfaces which can be used simultaneously as well as internal Data EEPROM.

STD Flash MCU HT66/67/68Fxx Series

- Wide Operating Voltage Range: 2.2V~5.5V
- Multiple Low-power Operating Modes
- Fast Wake-up Function
- Industrial Specifications: -40°C ~ +85°C
- Multiple External Communication Interfaces
- Internal Accurate High/Low Speed Oscillators
- High Noise Immunity and Excellent ESD protection



HOLTEK SEMICONDUCTOR INC.

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

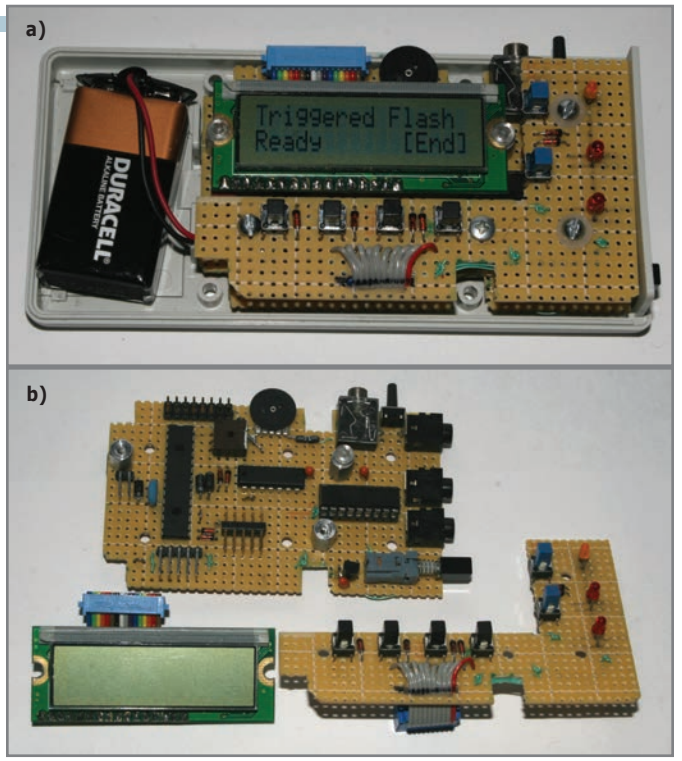


Photo 5a—Here is the assembled Photo-Pal with the top of the case removed. **b—**These are the individual boards and display.

10-bit ADC. Trim resistor R5 is used so the ADC would read fullscale (1023) if the input were 10.23 V, so the ADC reads directly in tens of millivolts.

Figure 4 shows the audio amplifier and I/O section. For normal sound-triggered operation, an electret microphone is connected between the tip and sleeve of a 1/8" stereo phone plug inserted into J4. The electret element's required DC bias is supplied through R11 when the PIC16F873A's RA5 pin (U2–7) is driven high. The RA5 output is filtered by R10 and C6 to eliminate noise. When an electret microphone is not being used, RA5 is driven low and R11 appears as a 10-kΩ input load to ground.

The audio signal from J4 passes through a nonpolarized input capacitor (back-to-back electrolytics C7 and C8) into the gain adjustment potentiometer, R12. The gain control output drives a two-stage audio amplifier formed by the B and C sections of the LM324 quad op-amp, U3. The op-amps are biased to the middle of their operating range by the 2.5-V reference formed by divider R21–R22 and op-amp U3's A section.

The amplifier output passes through a detector formed by C10, D8, and D9 and into the PIC16F873's comparator input on the AN3/RA3 pin (U2–5). When an audio peak is detected and AN3 is driven above the internal reference voltage, the comparator produces an interrupt, which provides the trigger input for the software.

The Photo-Pal can also be triggered by a switch contact input to J4's ring terminal. This input is pulled high by R23 and supplied to the second comparator of the PIC16F873 via the AN2/RA2 pin (U2–4). R24 and diodes D10–D11 protect this input from overvoltage. Software determines whether the trigger is caused by contact closure or opening.

An additional switch input at J5 is used to "arm" the trigger detection mechanism for the TF mode. SW8 provides a local push button for this input, which goes to the RB0 interrupt input pin of the PIC16F873 (U2 –21). This input can also act as a trigger for the TS and MF modes.

Outputs to activate the flash and the two-stage camera shutter are supplied by the collectors of a Texas Instruments ULN2803A Darlington transistor array (U4). Each output can sink up to 500 mA and switch up to 50 V. The flash output tip and ring are driven by the PIC16F873's RC6 and RC7 pins (U2–17, 18) while the shutter outputs are driven by the RB2 and RB3 pins (U2–23, 24). U4 is socketed, so it can be easily replaced if it becomes damaged.

The Photo-Pal's main circuitry was built on a prototyping board shaped to fit into the bottom of a plastic box. A second board was shaped to fit around the display to mount the front-panel switches and LEDs. Surface-mount resistors and capacitors were used on the boards' underside wherever practical. Components were connected using wire-wrap soldered directly to the components' pins. Photo 5a shows the assembled Photo-Pal with the top of the case removed. Photo 5b shows the individual boards and display.

PHOTOGRAPHIC POSSIBILITIES

The Photo-Pal camera controller has been a satisfying project, both to build and to use. I'm just beginning to explore the device's photographic possibilities. I am now investigating the possible use of high-power white LEDs in place of my Canon Speedlite to create even shorter flash durations. The microsecond resolution of the Photo-Pal's delay timing and flash output duration will help me explore this. In building this camera controller, I was able to develop ideas for a generic front-panel configuration and parameter-entry software that will be directly applicable to future projects. In Part 2 of this article series, I wil explore those ideas in more detail.

Richard Lord (rhlord@comcast.net) holds a BS in Electrical Engineering and an MS in Biomedical Engineering. During his career, he has designed digital electronics for an aerospace company and several telecommunication test equipment manufacturers. Working as a consultant in the 1980s, Richard designed several medical pulmonary test instruments and the electronics for an autonomous underwater robot. His interests include digital electronics, photography, jazz, and river conservation.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/267.

RESOURCE

The Edgerton Digital Collections project, http://edgerton-digital-collections.org.

SOURCES

PIC16F873A Microcontroller
Microchip Technology, Inc. | www.microchip.com

LM78L05 Regulators and ULN2803A Darlington transistor array
Texas Instruments, Inc. | www.ti.com



COMPLETE ENGINEERING SOLUTIONS

Start here.

At Newark element14, all your engineering needs come together in one source—vast product range from world-class brands, fast online search, seamless purchasing tools, resources and services, one-on-one support, and a community of experts. Here, you'll find simpler, smarter and faster ways to do business.

"The navigation and ordering process are easy to work. Thanks."

– Richard, Newark element14 customer

Newark element14

HOW MAY WE HELP YOU TODAY?

COMMUNITY: element14.com
WEBSITE: newark.com
PHONE: 1.800.463.9275
LEARN MORE: newark.com/together

by Kerry Imming (USA)

FAT Cache

Using File Allocation Table (FAT) Cache for Fast and Efficient SD Card Access

The FAT file system provides a convenient standard for data storage on SD flash memory cards. But that convenience comes with a performance penalty, since file system information must be accessed in addition to the file data. This article describes an algorithm that maximizes file data bandwidth by prefetching and compressing the file system information, virtually eliminating the file system overhead.

While today's multimegahertz processors may eliminate the need for code optimization, using a small microcontroller costs less and consumes less power. The microcontroller's limited code space and small RAM size challenge you to create the most efficient code possible to achieve maximum performance.

The current application is an audio card that plays back three tracks of 4-bit adaptive differential pulse-code modulation (ADPCM) audio through an audio DAC accessed via a SPI bus. Audio data files are stored on a microSD flash card. The sound files are named with a two-hex-digit index. The audio card is part of a set of controllers I developed and named "PinKit" since they were designed to control a custom pinball machine. The cache technique described in this article is applicable to other situations where processor throughput and/or I/O bandwidth is at a premium when accessing SD flash memory.

SD CARD FILE SYSTEM

Two options were considered for data storage on the SD flash memory card. Since a fixed number of read-only audio files are stored on the card, an easy implementation would divide the card into 1,000 fixed-size blocks. The number of files could be increased by using two block sizes (e.g., 64 KB and 4 MB). Another way to think of this is to make the file allocation table (FAT) cluster size larger than any files you want to store. The file length could be stored as the first block's value. The block's starting sector could be calculated from the sound file number. The downside to this option is that special software is required to write the data onto the SD card.

The second option is to stick with a standard PC file system. This has the advantage that anyone with a PC and an SD card reader can format and load audio data onto the card. This option also provides automatic handling for bad data sectors on the SD card, but this is a minor consideration since bad data would likely be unnoticed in the audio playback. If the standard file system's overhead can be tolerated, option two's common availability makes it the preferred choice.

The file information caching system I describe provides the best of both of these options. Files are written onto the SD card in a FAT16 file system. The file information from the FAT is then preloaded in memory as the starting sector and length. The preloaded data is all the information needed to read directly from the file data.

PC FILE SYSTEM

PCs still support the FAT file system, but as of Windows XP Service Pack 2, the default for flash memory format on the PC is exFAT, a Microsoft proprietary file system that is incompatible with FAT16. There are also some limitations about what can be formatted as FAT16, since it cannot address drives larger than 2 GB. The HP USB disk storage format tool enables you to format larger flash drives as a 2-GB FAT16 drive.

To open a file, you must search the directory entries. A sector read is required for every 16 entries searched. Keeping the most frequently used files at the top of the directory structure improves performance.

Reading from a file requires access to the FAT to find the next cluster. This is converted into a sector address that is used to access the media. Subsequent reads continue looking at consecutive sectors until the cluster boundary is

reached. Maximizing the number of sectors per cluster improves performance.

I observed the following FAT characteristic behaviors on Windows XP. There are other valid ways to handle files, but this behavior is consistent. Files are allocated starting at the first cluster and chained through sequentially increasing clusters. Overwriting a file marks the original file as deleted (using a special character in the filename) in the directory and creates a new directory entry at the end. The clusters for the new file are allocated at the end of the previous last file (i.e., clusters are not immediately reused when they're deleted). This is good, as it helps prevent file fragmentation.

I strongly recommend that you format the SD card before loading data onto it and before overwriting any files. Deleting files is not enough, even if you're overwriting them with the same length files. It's acceptable to add new files.

FAT16

I don't want to provide a tutorial on the FAT file system, but an understanding of some of FAT's key features can help you understand the cache structure. The 16 in FAT16 refers to the data size of the cluster pointer, which is 16 bits. A cluster is a contiguous block of sectors, typically eight for a 4-KB cluster size, but can be as large as 64 or 32 KB per cluster. In most cases, a sector is 512 bytes. When disks increased in size beyond 2 GB, the 16-bit cluster pointer was no longer sufficient, resulting in FAT32, which is now more common.

Directory entries are 32 bytes each, so one 512-byte sector contains 16 directory entries. FAT entries are 2 bytes each, so one sector contains 256 entries.

MDD FILE SYSTEM LIBRARY

The Microchip Technology Memory Disk Drive (MDD) file system library is a great resource and provides functions to initialize (mount) the file system, open, read, write, and close files. Many additional functions are available, but they can be disabled to save code space.

To get maximum performance out of the library, you must understand some of its key aspects. The library uses two main 512-byte data buffers. All SD card operations are done in 512-byte (sector-size) blocks, which is the most efficient access length for the microSD card. One buffer caches the most recently accessed FAT sector. The second buffer caches the most recently used data sector. One data buffer is allocated for each open file, but the FAT buffer is shared by all open files.

Data is read by two main functions `FSfOpen` and `FSfRead`. The `FSfOpen` function searches the directory for the specified filename, initializes the file structure, loads the FAT sector for the file if it's not already in memory, and loads the first sector into the data buffer. `FSfRead` reads data into the target data buffer. If the sector containing the data is currently in memory, data is copied from that data buffer into the target data buffer. If not, the next sector is first read from the SD card. If a cluster boundary is reached, the FAT is read to find the next cluster. If the correct FAT is not in the memory, a new FAT sector is loaded from the SD card.

For most applications, this is more than adequate. Even when simultaneously reading from multiple files this works well, as long as the reads are done in large blocks, ideally 512 bytes, and the accessed files have their file information in the same FAT sector.

One FAT sector can address up to 256 clusters with each cluster containing up to 32 KB of data. Once the combined file sizes exceed 8 MB (i.e., 32 × 256 KB), different files will likely need different FAT sectors to index into the file data.

The brute-force solution is to keep multiple FAT sectors in memory, but that quickly uses up valuable microcontroller RAM. One FAT buffer and one data buffer (1 KB for both) are needed for each open file. A better option is to compress contiguous chains of clusters into large blocks by tracking the start and length.

SD CARD PERFORMANCE

SD cards can operate in a wide (4-bit) data bus mode or in a 1-bit SPI mode. My PinKit audio board uses the 1-bit mode because the hardware SPI controller makes it more efficient than running the 4-bit data bus in software. The maximum data frequency is 400 kHz.

Like other engineering choices, the FAT cache design has some tradeoffs. SD card capacity has increased much faster than the bandwidth to access the data. For this application, wasting capacity to optimize data bandwidth is a good compromise. FAT16 saves data memory space with the reduced size for the cluster pointers at the expense of restricting the microSD capacity to 2 GB.

REAL-TIME EMBEDDED PERFORMANCE CHALLENGES

It's important to fully understand the library functions to calculate the resources used for performance-critical operations. A read operation that, in theory, uses 512 bytes of SD card bandwidth may need to read two FAT sectors before reading the data. In highly fragmented files, this could potentially be on every cluster boundary, but careful usage can avoid fragmentation.

With today's high-performance microcontrollers, it's easy to disregard this type of performance tuning as obsolete. But, what was historically needed to minimize code and data size, or to optimize real-time performance to achieve function, will be revisited in the future to minimize power.

FAT CACHE DETAILS

The cache structure includes two data structures, `FILEcache` and `FATcacheRLE`. `FILEcache` contains the file data (e.g., size, `current_sector`, and pointers to the `FATcacheRLE` structure).

The `FATcacheRLE` structure is an array of two entry elements, a sector number, and a length (in sectors). If the cluster chain is noncontiguous (i.e., fragmented files), additional entries are created for each segment. The last segment is indicated by negating the length in the final entry. This saves the storage needed for a dedicated end-of-chain field.

Entries in `FATcacheRLE` are compressed in a version of run-length encoding. Since an optimally stored file is stored in consecutive clusters, the entire file (or at least large blocks) can be accessed by knowing the starting sector number and the number of sectors in the block. The number of entries in the cache is configured by a constant that can be changed at compile time.

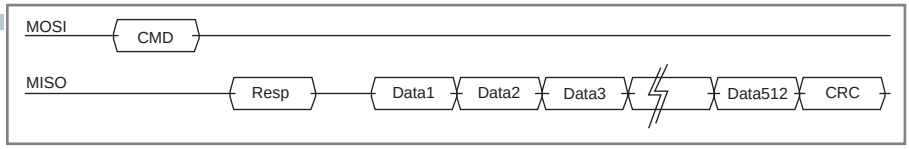


Figure 1—In the command, response, data sequence, the microprocessor master transmits the 6-byte command to the SD card slave and the SD card responds with a 1-byte response followed by 512 bytes of data and a 1-byte CRC.

FAT CACHE FUNCTIONS

This section describes the functions added to extend the MDD file system with the new, optimized reads. The files with file system information that has been prefetched into the FAT cache do not count against the permitted number of open files. All the normal MDD file system functions can be used in parallel to these new functions. In other words, you can freely mix opening and reading other files with reads from cached files.

The `FSfprefetch(FSFILE*stream, int filenum)` function preloads the cluster chain from the FAT into the compressed data structure. `*stream` is an `FSFILE` pointer to a previously opened file. `filenum` is the index number where the file information will be stored. This index will be used as `filenum` for future read operations. In the intended usage, files that will be read multiple times during runtime operation would all be opened and prefetched at startup. Thus, the file system overhead's microSD bandwidth is only consumed at startup.

The `FSfread512(void *ptr, int filenum)` function reads 5l2 bytes for the file prefetched as `filenum` from the SD card directly into a user buffer space pointed to by `ptr`. This direct read is important in this case, as it also avoids the overhead of copying the data out of the library data buffer into the user data buffer while preserving that buffered data for the open file to which it belongs.

The `FSfrewind512(int filenum)` function resets cached file `filenum` to the start of the file. The `FSf close512(int filenum)` function closes cached file `filenum` by removing its entries from the FAT cache. Closing a prefetched file is optional since prefetching a new file will replace "`filenum`" with the new entry.

The `FSerror()` function returns an integer representing the status of the last `FSfread512` operation. Zero = good, nonzero = end of file (EOF) or an error occurred. You can differentiate the nonzero cases, but this application treats all nonzero return codes as EOF. Note that `FSerror` is an existing function, but is not normally included in the base file operations.

IMPROVED PERFORMANCE

At a 400-kHz SPI data rate, each bit time is 2.5 µs (i.e., 20 µs per byte). A command to the SD card is 6 bytes long. The command response is 1 byte followed by the data, 512 bytes in this application. Data is followed by a single byte of CRC, which is not checked. Figure 1 shows the command sequence. A total of 520 byte times are required on the SPI bus to perform a 512-byte read, or 10.4 ms (520 bytes × 20 µs/byte) per transfer. 512 bytes per 10.4 ms is a data rate of 49.2 KBps. This is a theoretical maximum bandwidth, since it doesn't account for the overhead of data handling in the microprocessor.

The previous calculations assume all the SD card accesses are data, but that is not the case when using a file system. Accesses must also be made to the directory and the FAT.

Figure 2a shows the case of two files accessed with the directory and FAT entries needed contained in single sectors. A file "A" read needs directory and FAT access prior to the data read. Subsequently, a file "B" read only needs directory access since it can use the cached FAT sector. After this, reads of "A" and "B" have no file system overhead.

Figures 2a, 2b, and 2c show the number of sector reads required for the standard file system compared to those with the FAT cache. The scenarios are shown after startup, so prefetching of the file system information into the FAT cache has already completed.

Figure 2b shows where files "A" and "B" are in disk clusters that are not in the same FAT sector. In this case, whenever a 32-KB cluster boundary is crossed, an extra access must be made to the FAT to find the next cluster. This is a realistic worst-case scenario and results in the loss of half of the data bandwidth.

With the new FAT caching functions described here, the entire FAT cluster chain is prefetched at power on. All overhead of directory and FAT accesses during runtime are eliminated. This

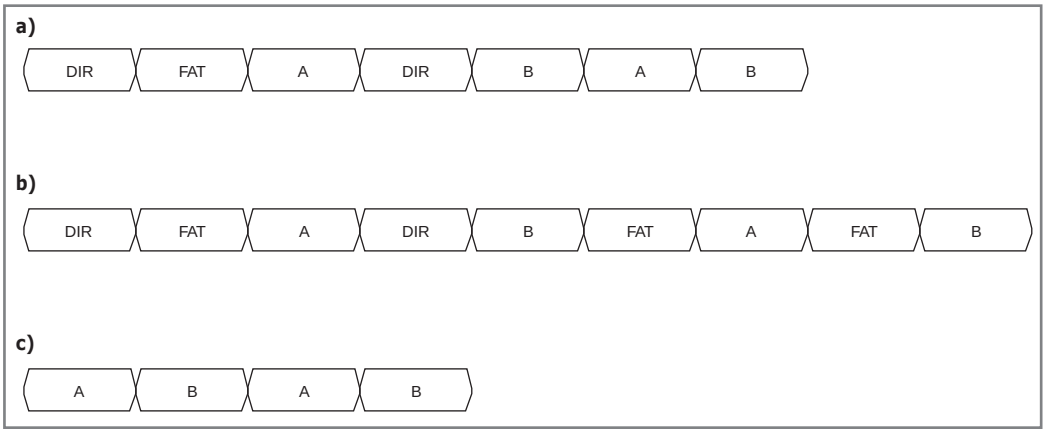


Figure 2a—This is the typical sequence when reading two files, A and B. A directory sector is searched followed by a cluster lookup in the FAT prior to reading the sector containing the requested data. An additional directory lookup is needed to read a second file. The cached FAT sector is used to look up the cluster prior to reading file B. **b—**This is a similar example showing the cluster data for the two files in different FAT sectors. In this case, a FAT read is required at each cluster boundary. **c—**This shows the access for the FAT cache described in this article. All file cluster information is cached in a compressed data structure in microprocessor memory. File system access overhead is not incurred for reading from the FAT-cached files.

case is shown in Figure 2c.

There is another minor performance improvement in the `FSfread512` function. Since it is optimized for streaming, the data is not stored in the MDD sector data buffer. The SD card data is transferred directly into the user's data buffer. This eliminates the 512 pass loop to copy the data. Since the loop is about 10 instructions long, this saves about 0.3 ms (i.e., (512 × 10)/16 MIPs). While this is only a secondary advancement, it still amounts to an improvement of about 3%.

POSSIBILITIES & SUCCESS

The cache could be extended to keep file names in the `FILEcache`. This would increase flexibility at the expense of having to search the `FILEcache`. In this case, the `FSfopen` function would need to be enhanced to first search the cache. A seek instruction could be added that chains through the `FAT cache` entries and sets the position for the next read operation.

A more general change that caches the cluster chain during a file open operation could be made to the library. The library memory buffer for the FAT sector could be replaced with a compressed chain for the open file.

The Microchip MDD file system library is a well-designed set of routines that provides a robust function set. The cache extension described here is an optimization based on a specific usage model: multiple files, file reuse, interleaved access, and streaming access.

Adding the FAT caching functions provides access to 64 files without any overhead for switching between them. The FAT entries are all preloaded at start-up. Keeping one file open using the standard routines and accessing the other two (or more) using the cache structure is almost as efficient access as using all cached files. Using the FAT16 file system and a PC to load data onto the SD card, following strict rules to prevent deleted file entries and fragmentation, and preloading files using the FAT cache structures described in this article provided the bandwidth needed to simultaneously stream three audio files.

Kerry Imming (kcimming@pobox.com) is an electrical engineer working on VLSI chip design. He has an MSEE from the University of Minnesota and a BSEE from Iowa State University. Kerry has worked on designs spanning from supercomputers to midrange computers to the PlayStation 3 processor. He also has a broad range of programming experience in both computers and microcontrollers.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/267.

RESOURCES:

J. Bachiochi, "Access SD Memory Cards (Part 1): Solid-State Storage Media in Embedded Apps," *Circuit Cellar* 222, 2009.

CNET, "HP USB Disk Storage Format Tool," 2010, http://download.cnet.com/HP-USB-Disk-Storage-Format-Tool/3000-2094_4-10974082.html.

Microchip Technology, Inc., "Microchip Application Libraries," www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2680&dDocName=en547784.

P. Reen and N. Mohaanswamy, "File I/O Functions Using Microchip's Memory Disk Drive File System Library," Application Note AN1045, Microchip Technology, Inc., 2010, www.microchip.com/msd.

SD Association, "SD Specifications Part 1: Physical Layer Simplified Specification Version 3.01," 2010, www.sdcard.org/downloads/pls/simplified_specs/Part_1_Physical_Layer_Simplified_Specification_Ver_3.01_Final_100518.pdf.

SOURCE

Memory Disk Drive (MDD) file system library
Microchip Technology, Inc. | www.microchip.com



by George Novacek (Canada)



Managing Project Risk

Risk is an integral, unavoidable part of project development. This article describes the process of defining and evaluating risk factors that could negatively impact project cost and delivery.

Life is full of risk. When embarking on a new project—even if you think it will be a cinch—don't forget Murphy's Law: "Anything that can go wrong, will go wrong." Every executive worth his salt knows things can go wrong during a project's lifespan. Sometimes design features don't work as expected, components are discontinued, a key employee quits, or the union goes on strike. There are zillions of reasons why you've missed the deadline, but the customer doesn't care. He wants his project delivered on time and on budget.

To mitigate risk, it must first be defined and evaluated. Only then can effective steps be taken to bring it under control. Risk analysis is the first step, but unfortunately, there is no cookbook on which to rely. Everything depends on the type of product, the potential risk, and a multitude of factors—including the weather or sun spots—all of which are completely out of your control.

To conclude this project management-based series of articles, I will discuss a method to keep you aware of potential project risks and help you avoid or mitigate them in time. I've used this approach to report to my superiors and customers about project statuses and to make executive decisions for the company.

"To mitigate risk, it must first be defined and evaluated. Only then can effective steps be taken to bring it under control. Risk analysis is the first step, but unfortunately, there is no cookbook on which to rely. Everything depends on the type of product, the potential risk, and a multitude of factors—including the weather or sun spots—all of which are completely out of your control."

EVALUATING RISK

What if you were to evaluate the risk of developing an embedded controller including hardware and software? You can express the possibility of your failure to finish the project on time as:

$$P_F = \frac{M_{HW} + M_{SW} + C_{HW} + C_{SW} + D_F}{X}$$

Denominator X is the same as the number of the numerators. Here X = 5. If you were only evaluating software, for example, hardware factors would be left out, and X = 3.

M_{HW} reflects the hardware maturity factor. Do you have experience with the technology to be used? Are the circuits similar to those used in prior projects? Do you expect potential challenges you haven't experienced before (e.g., thermal issues)?

M_{SW} is the software maturity factor. How experienced is your software development team? Do you have the software development process, verification, and validation under control? Must new algorithms be developed? Has the memory been correctly sized? Does a new microcontroller need to be used?

C_{HW} is the hardware complexity factor. Is this

a) M_{HW}	The design has been successfully used several times in the past. Only a minor circuit modification is required.	0.2
C_{HW}	An off-the-shelf purchased switching module is used. Some overvoltage protection circuits are added.	0.1
D_F	The design is independent of any existing systems, facilities, or subcontractors. Two module sources exist.	0.05
P_F	**Possibility of failure – no real risk to worry about**	**0.117**

b) P_{EF}	No consequences are envisaged if some fallback is needed (e.g., purchasing the switcher module from the second source).	0.1
S_{UF}	Supportability will be unaffected if the fallback above is exercised.	0.1
C_{OF}	There is a minor risk that the second source product might cost more.	0.05
S_{CF}	There shouldn't be a slip in the schedule due to second source usage.	0
C_F	**Consequence of failure**	**0.0625**
R_F	**The risk factor indicates overall low risk**	**0.172**

Table 1—These are examples of hardware risk factor calculations.

a) M_{SW}	Software maturity factor. Good experienced team, reuse of some previously developed code, but some unknowns exist.	0.4
C_{SW}	The complexity factor is low and the software is fairly straightforward.	0.2
D_F	There is no dependency on outside contractors, facilities, or systems.	0
P_F	**Possibility of failure**	**0.2**

b) P_{EF}	Medium-level consequences are envisaged if fallback is needed, mainly due to the extent of documentation update.	0.55
S_{UF}	The new version number needs to be communicated by a circular.	0.2
C_{OF}	Cost factor is an issue, as this is a fixed-price contract.	0.6
S_{CF}	Extensive software planning has been done to ensure working software is delivered on schedule. Some impact is expected if fallback is needed.	0.25
C_F	**Consequence of failure: medium**	**0.4**

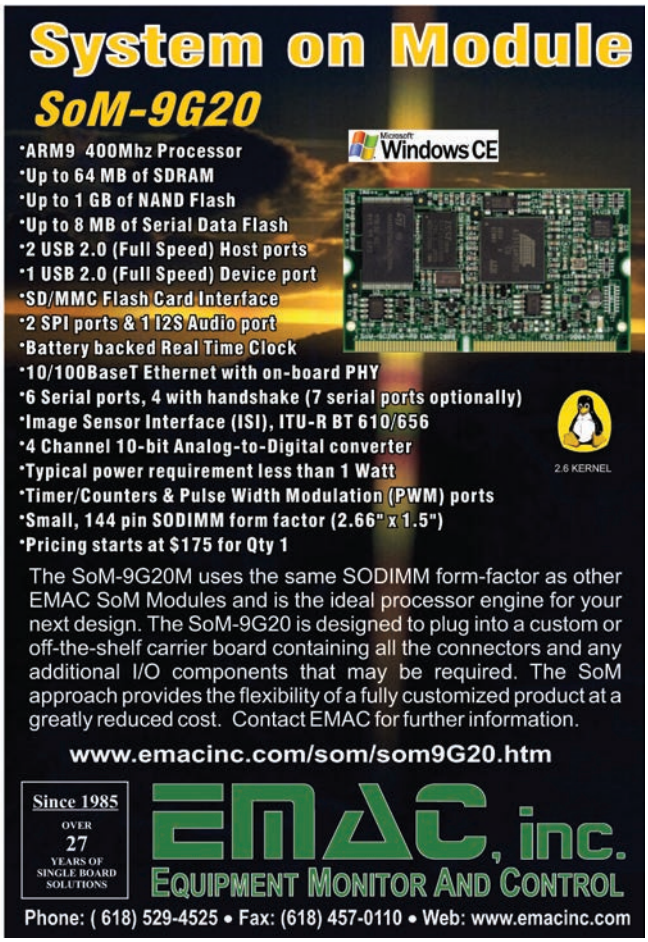
Table 2—Here are examples of software risk factor calculations.

essentially a standard design or is it going to be hardware worth bragging about? Similar questions are raised about the software complexity factor C_{SW}. Is it a simple or a sophisticated system?

The dependency factor D_F reflects whether the project is independent of any other system or can be affected by factors outside your control (e.g., interfaces).

Numbers between 0 and 1 are assigned to those factors. The higher the number, the greater the risk. It's reminiscent





Item	Header	Risk Description	Probability	Detectability	Severity	Rating
1	Engineering	Pressure sensor will drift with time and temperature	4	3	4	11
2	Program	Tracking to schedule	2	1	3	6
3	Test	Survival of higher EM and HIRF levels	2	1	1	4
4	Human resources	Union or engineering team issues	1	1	1	3
5	Engineering	Impact if DO254 compliance is required	3	3	5	11
6	Finance	Exceeding estimated cost	3	1	1	5
7	Manufacturing	Process for BGA installation	2	5	3	10
8	Software department	Certification issues	3	3	3	9
9	Program	Reliability of customer-selected connectors	5	3	5	13
10	Human resources	Project staffing	1	1	1	3
Overall company risk: medium						**7.6**

Table 3—This is an example of a risk management status report prepared for top management.

of being asked by a doctor to rate your pain using a number between zero and 10. It is very subjective, depending on your judgment and experience. But you will quickly learn not to be overly optimistic or pessimistic.

Next, the consequence of your failure needs to be determined using the following equation:

$$C_F = \frac{P_{EF} + S_{UF} + C_{OF} + S_{CF}}{4}$$

The performance factor P_{EF} provides an idea of the degree of consequences, if you were to fall back on another option or a recovery. Supportability factor S_{UF} describes the effects of that fallback on the supportability of the project. C_{OF} is a cost factor. If you are paid for all your costs plus a little profit, the cost effect of a fallback is zero. If you are designing for a fixed cost, a fallback means money out of your own pocket and it could be serious.

The schedule factor S_{CF} may not be a big deal if the recovery is minor and you have built slack into the schedule. But a major problem could result in a significant delay with major repercussions. Based on the above equation, the overall risk factor can be calculated as:

$$R_F = P_F + C_F - (P_F \times C_F)$$

Typically, R_F less than 0.3 is considered low risk. R_F from 0.3 up to 0.7 is moderate risk. R_F greater than 0.7 is considered high risk.

Risk factors can be assigned to specific development phases and reduced as the work progresses. You should always work on the high-risk items first and plan on reducing their risk factors by a certain percentage at some milestones. For example, once you have complete software specification, you could reduce the risk factor by 20% and keep reducing it when certain milestones are met until a 0% risk factor is reached when you are ready to ship.

RISK FACTOR CALCULATIONS

Engineering project risk analyses is not the same as product risk based on probability calculations. Table 1 shows risk calculations for a DC power supply. No software is involved, thus M_{SW} and C_{SW} are not included and X = 3.

Having obtained the same price from the second source, you may reduce C_{OF} to 0. Software risk estimation is very similar (see Table 2).

Having defined the risks, you need to develop fallback actions for each potential failure. You need back-up plans, which may be as simple as substituting some components with better or more powerful ones, or as frightening as a fundamental redesign.

Using the same principles, you can evaluate and manage risk for anything, beginning with a circuit design, computer data security, a home fire, or a company's performance. The evaluation criteria will be tailored accordingly. Table 3 is an example of the weekly company status report presented to me by management. Pictures are worth a thousand words, and the situation can be grasped in seconds. Using colors, the PowerPoint presentation immediately points to issues the management must concentrate on first.

To conclude, the most important engineering and management challenge is to always understand the current situation. You don't want any surprises. Knowledge is power. If you understand the challenges awaiting you, you can face them on your terms, in a timely manner, and be successful.

George Novacek (gnovacek@nexicom.net) is a professional engineer with a degree in Cybernetics and Closed-Loop Control. Now retired, he was most recently president of a multinational manufacturer for embedded control systems for aerospace applications. George wrote 26 feature articles for Circuit Cellar *between 1999 and 2004.*

NEED-TO-KNOW INFO

Knowledge is power. In the computer applications industry, informed engineers and programmers don't just survive, they *thrive* and *excel*. For more need-to-know information about some of the topics covered in this article, the *Circuit Cellar* editorial staff recommends the following content:

—

Is the Door Closed?
Why Every Safety-Critical Decision Matters
by George Novacek
***Circuit Cellar* 242, 2010**

Part of developing a "robust" system involves ensuring your design is safe and secure. Whether you're building a small embedded app or a complicated door-opening control system for a plane's landing gear bay, you must protect both your design and the end users by choosing the proper parts and design techniques. Topics: Safety Critical, Design, Risk, Security

—

Go to *Circuit Cellar*'s webshop to find this article and more: www.cc-webshop.com



▶ CUWIN6300

▶ CUWIN5500

▶ CUWIN5200

Windows CE 6.0 Touch Controller

CUWIN

The CUWIN is a series of Windows CE touch controllers that are more cost-effective than a PC, but with more features than an HMI touch screen.
Create sophisticated applications with C++ or any .Net language.

533MHz ARM CPU
128MB SDRAM & Flash
SD Card Support
Ethernet, RS-232/485
USB, Audio Out
Windows Embedded CE 6.0



C-Programmable Modular Industrial Controller

MOACON

The MOACON is a modular, C programmable, ARM-based automation controller designed for industrial environments.

Choose from a diverse, feature-rich selection of modules including:
Digital I/O
Analog I/O
RS-232/485
Motor Control
Ethernet
High-speed Counter & PWM

BASIC with LADDER LOGIC CONTROLLER



New

Integrated CUBLOC Controller and I/O Board

CB210

The CB210 is an inexpensive, integrated CUBLOC controller and I/O board programmable in both BASIC and Ladder Logic.

I/O Ports x20	10-bit A/D x6
PWM x3	RS-232 x1
80KB Program Memory	3K Data Memory

The CUPC is a series of industrial touch panels with all the features of a modern PC for the most feature-rich user experience.

ATOM N270 1.6GHz CPU
2GB RAM
320GB HDD
SD Card Support
Color Display (1024 x 768)
RS-232, Ethernet
USB, Audio Out
Windows XPe/XP Pro

CUPC

Industrial Touch Panel PC with ATOM Processor



▶ CUPC-P200

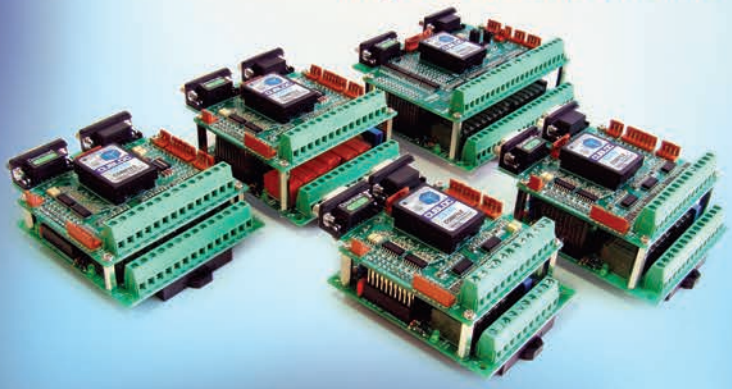
▶ CUPC-P220

The CUSB is a series of compact, CUBLOC-integrated, industrial I/O boards programmable in both BASIC and Ladder Logic.

CUBLOC CB280 Core Module
80KB Program Memory
3KB Data Memory
DC 24V Inputs x9~16
A/D Inputs x2~6
Relay Outputs x6~16
PWM x2~6
High-speed Counters x0~2
RS-232/485 x1~2

CUSB

Integrated Industrial Controllers



COMFILE TECHNOLOGY

1175 Chess Dr., Suite F, FOSTER CITY, CA 94404
call : 1-888-9CUBLOC (toll free)
1-888-928-2562
email : sales@comfiletech.com

www.comfiletech.com

by Bob Japenga (USA)



Concurrency in Embedded Systems (Part 3)

Avoiding Concurrency Problems

The first two parts of this article series introduced concurrency in embedded systems and discussed two common concurrency design problems. The third part of this series examines generic ways to avoid some of these problems.

The introduction to this article series discussed some common pitfalls when using concurrency. The last article examined case studies of specific pitfalls. This month, I want to step back and look at concurrency from a higher level, and provide some generic ways to help design robust systems with concurrency.

CONCURRENCY WITHOUT PREEMPTION

Concurrency is often used to remove the complexity from a design with many unique and independent functions. For example, consider a simple device with a control function, a data storage function, a networking function, and a user interface (UI) with a keyboard and a touch screen. Conceptually, it may be easier to break these four functions into six separate concurrent threads. The UI would have three threads: one for the keyboard, one for the touch screen, and one to manage the user inputs to create the display. Additional "threads" may exist in the interrupt routines. Once broken into these concurrent threads, the designer can think through each function separately and seamlessly intersperse necessary waits. Breaking these into separate concurrent threads provides other significant advantages, namely there is less coupling between functions, it is easier to delegate the tasks to separate people, and it is easier to debug.

If the threads are designed so no thread runs for more than 10 to 20 ms and there are no high-speed and/or hard real-time requirements, preemption (one thread interrupting another) may be unnecessary, except in the interrupt service routines (which are inherently preemptive). Ideally, you may be able to eliminate custom interrupt routines, which can make everything easier. This can also simplify resource sharing by eliminating the need to coordinate reading and writing to these shared resources. It prevents the possibility of deadlock and priority inversion. It eliminates the need for semaphores, mutexs, and other locking mechanisms, which can cause more problems than they solve. There is no need to worry about file sharing. Without preemption, you also don't need to worry about atomic operations or which library functions are thread safe.

Identifying what libraries are thread safe is no trivial task. One real-time operating system (RTOS) manufacturer that touts its reliability has almost 20 pages of documentation about which functions are completely thread safe and which are partially thread safe. It includes special cases of partially thread-safe functions and functions that are thread safe except for one shared variable. The POSIX standard lists about 100 standard functions that are not thread safe.

CONCURRENCY WITH LIMITED PREEMPTION

Late in a major new product's development cycle, we began noticing some intermittent and difficult to duplicate errors. After significant pain and effort, we found a particular thread-safe library function was not thread safe. Our solution proved quite simple. We set all threads' priorities the same except for one that absolutely had to be preemptive. We ensured this thread did not use our "not so safe"

thread-safe library. The fix was quick and painless.

The other obvious way to use limited preemption is to include preemption in your interrupt routines. Then, you only need to worry about the concurrency pitfalls in routines with less resource sharing.

CONCURRENCY ON ACID

ACID—which stands for atomicity, consistency, isolation, and durability—is a database paradigm developed in the late 1970s as a means of evaluating the reliability of distributed databases. ACID is applicable to designing embedded systems with concurrency. Real-time multiuser databases have a lot of the same problems we have when designing embedded systems. Imagine if you and your spouse went to different ATMs to withdraw funds from your checking account. You would have concurrency, and you would have problems if the concurrency was not properly handled at the design stage. Ensuring that all your embedded design aspects have this paradigm's properties will help ascertain safe and reliable operation. The following sections describe each of ACID's elements.

ATOMICITY

My last article provided a detailed discussion on atomicity. Let's step back and look at it from 5,000'. A good way to describe atomicity from a higher level is to say: An atomic operation is one that can assure us that in all cases (including power failures, system resets, and all errors), the operation will complete as desired or have no effect on the system.

In my last article, I did not address the issue of an operation being atomic across power outages or system crashes, which are much overlooked concurrency problems. Of course, these are only problematic if you are operating on a nonvolatile resource. I am always amazed how often we don't take this into account in our designs. Imagine the simple case of writing an important parameter in its primary location and its backup. You carefully lock out other tasks from touching the parameter (i.e., change the primary, change the backup, and then unlock the access). You think you have created an atomic operation and prevented normal concurrency from affecting your design. But have you covered the case where the system powers down or crashes between writes? You now have inconsistent data on your system. When you power up, which one is correct, the primary or the backup? It is not enough to make the operation atomic under normal operation. You must make it atomic across system crashes or power outages.

One solution is to provide some sort of nonvolatile "dirty" indicator to each element. The steps are to set the first dirty bit, write the primary, set the second dirty bit, write the backup, and clear both dirty bits. When the system comes up, if only the first dirty bit is set, use the backup. If both are set (or neither is set), use the primary.

An approach I find useful in my designs is to ask myself every time I modify something that is nonvolatile: What happens to the system if this operation does not fully complete? Have I provided a mechanism to handle an incomplete operation? Finally, since I don't really trust my own ability to do this perfectly, all our systems must be tested under tens of thousands of power cycles (e.g., simulating crashes or asynchronous power outages) in an attempt to verify we have done this correctly.

CONSISTENCY

An embedded system that exhibits consistency cannot end up in an inconsistent state. I addressed consistency in my last article when I discussed writing multiple-byte data structures in an environment without atomicity. But consistency goes beyond temporary data. Many of the systems we design as embedded engineers must run 24/7 and endure hundreds of power failures in their lifetime (and unfortunately a few crashes). As previously discussed, power outages and crashes are a much-overlooked kind of concurrency. Ensuring the system does not find itself in an inconsistent state is no small task. I have often been involved in designs where these inconsistent states are discovered in testing (and usually late in testing).

Here are some things that have helped me avoid inconsistent states: Identify all the states that could be considered inconsistent. Repeatedly ask myself questions (e.g., Have I handled this case?). Put the checks in startup for inconsistent states. Even if I think there isn't a chance of it getting into that state, why not put the code in at startup to correct the case? Check for inconsistent states during normal operation. Concurrent operations could also create such a state. Is there code in it that can handle these inconsistent states? The definition's key is "end up in inconsistent states." Your system may get into this state temporarily. Ensure at startup and during operation you have provided a mechanism to restore it to a consistent state.

ISOLATION

Sometimes I think the ACID acronym creators needed an "I" for their acronym when they chose the word "isolation." I prefer to think of this principle as making your embedded design order of operation independent insofar as the operations are order independent. With concurrency, cases where operations are interleaved can often be created. And variable timing can create race conditions. Race conditions in software have killed people (e.g., the famous Therac-25 failures).[1]

How can you prevent race conditions and create order independent operations? In M. Barr's, "Firmware-Specific Bug #1: Race Condition" (*EmbeddedGurus*, 2010), a respondent to the blog wrote, "Without sharing resources, race conditions won't happen." That is just not true. Race conditions can happen in single-threaded software systems that have hardware with a mind of its own. As I stated in the first article in this series, concurrency can happen with the hardware being the concurrent thread. So, eliminating shared resources can reduce, but not eliminate, the probability of a race condition. The following are some of my basic guidelines to prevent concurrency-based order-of-operation-type issues.

Identification: Start by identifying operations that must be ordered. Look at your design and ask: Does this depend on timing from an external resource (hardware or software)? This is nontrivial. I didn't see most race conditions I created until it was too late. Once identified, start asking yourself: What if this device responds much more slowly? What if this gets interrupted? What resource (hardware or software) under me can change during this operation? Can I design this in an order-independent manner?

HUMANDATA

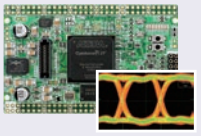
FPGA /CPLD Boards
from JAPAN

SAVING COST=TIME with readily available FPGA boards

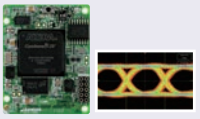
- Basic and simple features, single power supply operation
- Same board size and connector layout - ACM/XCM
- All stocked items are ready to be shipped immediately
- Over 100 varieties of FPGA/CPLD boards are available
- Free download technical documents before purchasing

ALTERA FPGA Board

CycloneIV GX F484 FPGA board
ACM-024 series
Cyclone IV GX | DDR2 | SIF40
EP4CGX50CF23C8N
EP4CGX75CF23C8N
EP4CGX110CF23C8N
EP4CGX150CF23C7N
Credit card size (86 x 54 mm)
RoHS compliant

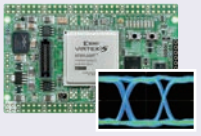


Cyclone IV GX F484 FPGA board
ACM-108 series
Cyclone IV GX | DDR2
EP4CGX50CF23C8N
EP4CGX110CF23C8N
EP4CGX150CF23C8N
Compact size (43 x 54 mm)
RoHS compliant

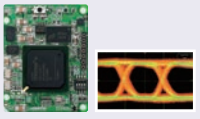


XILINX FPGA Board

Virtex-5 LXT FFG665 FPGA board
XCM-017 series
Virtex-5 | SDRAM | RocketIO | SIF40
XC5VLX30T-1FFG665C
XC5VLX50T-1FFG665C
Credit card size (86 x 54 mm)
RoHS compliant



Spartan-6 LXT FPGA board
XCM-111 series
Spartan-6 | DDR2 | RocketIO
XC6SLX45T-2FGG484C
XC6SLX75T-2FGG484C
XC6SLX100T-2FGG484C
XC6SLX150T-2FGG484C
Compact size (43 x 54 mm)
RoHS compliant



PLCC68 series

XP68-03 Spartan-6 PLCC68 FPGA Module
Spartan-6 | PLCC 68
XC6SLX45-2CSG324C
16Mbit Configuration Device
Two User LEDs
One User Switch(Slide)
RoHS compliant



AP68-04 Cyclone III PLCC68 FPGA Module
Cyclone III | PLCC 68
EP3C25U256C8N
16Mbit Configuration Device
Two User LEDs
One User Switch(Slide)
RoHS compliant



www.hdl.co.jp/CC/

HuMANDATA LTD.
E-mail: s2@hdl.co.jp
Fax: 81-72-620-2003

Preallocation: If possible, allocate or reserve all the resources necessary for an operation before you begin.

Identify the ordering: Use timestamps or sequence numbers in operations where appropriate. One of my maxims is: "Use of sequence numbers or timestamps covers a multitude of evils." This is especially true in networking. You can be surprised at the order your data comes back to you with load-balancing servers and powerful multithreading in the devices you are talking to.

Protect against preemption: Any code that must require specific ordering (again avoidance is the best medicine) that deals with external shared resources should be protected to guarantee this specific ordering.

Test, test, and test some more: Purposely shake up the testing order (i.e., stress test the identified sections). For example, if you can significantly add or reduce delays and response times of the shared hardware or software, you may uncover race conditions you have missed. I highly recommend testing on as many different hardware platforms as possible. Another one of my maxims is: "For every order of magnitude increase in production, you discover another problem in your perfect system."

DURABILITY

Durability means once an action is taken, it is taken. Sounds like a Yogism. When using an operating system or a file system, remember there is concurrency you don't even know about. We had an issue with a JFFS2 system on Linux where we would write a variable to a flash file system file prior to rebooting. We did this to let the system know the system's state from the last boot when it powered up. We wrote the variable, flushed the cache, synched the file system (`fsync`), and the variable was occasionally still not there. Once the action was taken, it was not really taken. We then discovered we needed to `fsync` the directory file as well. This closed a few more holes. Finally, one more note in the API said, "If the underlying hard disk has write caching enabled, then the data may not really be on permanent storage when `fsync()` returns." Oh, great! So much for durability. Our solution was to dismount the file system and remount it read only. Finally, the underlying file system cooperated. The moral of the story: Not all actions taken are taken!

KNOWLEDGE = POWER

Most systems have concurrency in them. You must understand it in order to plan for it. The more concurrency you build into the system, the more complexity you bring to the table. Rigorous and thoughtful designs are hard to achieve but worth the effort. Next time, I will examine building concurrency into an embedded Linux system.

Bob Japenga has been designing embedded systems since 1973. In 1988, along with his best friend, he started MicroTools, which specializes in creating a variety of real-time embedded systems. With a combined embedded systems experience base of more than 200 years, they love to tackle impossible problems together. Bob has been awarded 11 patents in many areas of embedded systems and motion control. You can reach him at rjapenga@microtoolsinc.com.

REFERENCE

[1] Wikipedia, Therac-25, http://en.wikipedia.org/wiki/Therac-25.

RESOURCES

M. Barr, "Firmware-Specific Bug #1: Race Condition," *EmbeddedGurus*, 2010, http://embeddedgurus.com/barr-code/2010/02/firmware-specific-bug-1-race-condition.

B. Japenga, "Concurrency in Embedded Systems (Part 1): An Introduction to Concurrency and Common Pitfalls," *Circuit Cellar* 263, 2012.

_____, "Concurrency in Embedded Systems (Part 2): Atomicity and TOCTTOU," *Circuit Cellar* 265, 2012.

ONE POWERFUL TOOL FOR YOU

The Entire *Circuit Cellar* Magazine Archive on a Limited-Edition 25th Anniversary USB drive!

CIRCUIT CELLAR 25 YEARS OF EMBEDDED INSIGHT



Here's what's included:

- Special anniversary USB drive or our traditional CC GOLD USB drive. *Your choice!*
- PDFs of all *Circuit Cellar* magazine issues in print through date of purchase
- Article code
- Design Challenge projects*
- Two years of *Elektor* magazine issues in PDF format (2010–2011)
- Two years of *audioXpress* magazine issues in PDF format (2010–2011)
- A USB memory upgrade from 16 GB to a whopping 32 GB!
- Free gift! *Circuit Cellar* 25th anniversary hat

Order today at ***www.cc-webshop.com***

* Design Challenge add-ons:
Atmel AVR Design Contest 2006, WIZnet iEthernet Design Contest 2007, Microchip Technology 16-Bit Embedded Control Contest 2007, Texas Instruments DesignStellaris Contest 2010, WIZnet iMCU Design Contest 2010

by Ed Nisley (USA)



MOSFET Tester

Peltier Temperature Control and Results

Because semiconductor properties vary with temperature, a MOSFET tester must control the transistor's temperature. Ed's Arduino-based tester maintains a constant temperature using a linearized thermistor and a Peltier thermoelectric module. This column describes the temperature controller and presents some MOSFET characteristics over a 0 ºC to 50 ºC range.

The MOSFET tester presented in my August column can apply 2 A at 1.6 V, so the MOSFET power dissipation can reach 3 W during parts of the measurement sequence and unpredictably change the transistor's temperature. Photo 1 shows how I avoided that problem: the transistor rests on an aluminum block atop a Peltier thermoelectric module that maintains a preset temperature. The firmware also reports the actual temperature as part of each test data point.

In this column, I'll describe the temperature measurement and control circuitry, explain the firmware's proportional-integral control loop, then present some MOSFET characteristics at different temperatures.

THERMISTOR LINEARIZATION

The first step in temperature control is, of course, temperature measurement. I used Semitec DCA8025 103CT-4 thermistors from my parts heap that present a simple voltage to an Arduino analog input pins. The thermistor datasheet provides the equation for resistance as a function of temperature:

$$R = R_0 e^{(B/T - B/T_0)}$$

The datasheet also specifies the constants: R_0 = 10 kΩ at T_0 = 298 K = 25 ºC and B = 3270 K. Note that the equation requires temperature in kelvin, which are roughly:

$$K = ^\circ C + 273$$

These thermistors have a negative temperature coefficient: the resistance decreases as the temperature increases. You're probably more familiar with the negative exponential found in RC and L/R time constants, but in this case the exponent varies with the reciprocal of the temperature.

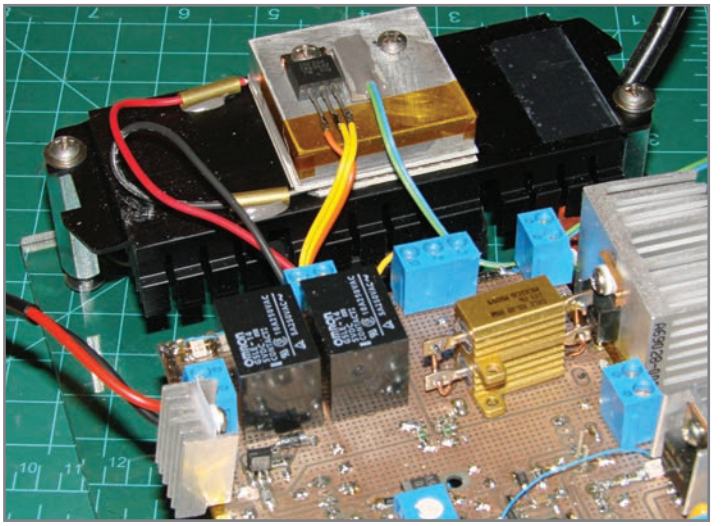


Photo 1—A fan under the black CPU heatsink keeps it near ambient temperature, so that the Peltier module under the aluminum block can control the MOSFET temperature. The gray epoxy block holds a linearized thermistor circuit connected to the Arduino microcontroller under the PCB. Two black SPDT relays operate in unison to switch the current direction through the Peltier module to allow both heating and cooling, with the current controlled by the MOSFET behind the small heatsink on the left edge of the PCB.

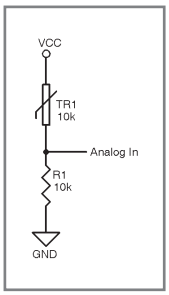


Figure 1—The canonical Arduino thermistor circuit uses a single resistor to produce a voltage output. Setting the resistor value equal to the thermistor's resistance at a particular temperature linearizes the output in a narrow range around that temperature.

Homework: plot R/R_0 against 1/T on semi-log axes, draw the obvious straight line, and compare its slope to B. Ponder the effect of errors in B to errors in the calculated temperature.

Thermistors intended for precision applications covering wide temperature ranges generally use the third-order Steinhart-Hart approximation relating temperature to resistance. If your project needs that level of precision, however, an Arduino's analog input probably won't look like the right hammer for the rest of the job.

The schematic in Figure 1 shows the simplest possible thermistor circuit: R1 is a series resistor equal to the R0 resistance of TR1, the thermistor. The voltage at their junction therefore equals $V_{CC}/2$ at 25 ºC, although you can choose a different R1 to put a different temperature at the midpoint. Putting the thermistor on the V_{CC} side of the circuit means that the measured voltage increases as the temperature rises and the resistance decreases, which seems easier to understand.

Of course, the circuit's output voltage isn't directly proportional to the thermistor resistance. The Arduino's ADC produces a binary number representing the ratio of each channel's analog input voltage to the supply voltage V_{CC} (actually, the ADC's reference voltage, but the Arduino sets that to V_{CC}), which varies with the reciprocal of the thermistor resistance:

$$\frac{V}{V_{CC}} = \frac{R1}{R1 + TR1}$$

You could plug the thermistor's exponential equation in place of TR1, then solve for the temperature as a function of voltage ratio, but there's an easier way: fit a straight line to the temperature-vs-ADC-value curve in the region you'll be using and do the conversion in a single step with a linear equation. Obviously, the accuracy of that approach won't be as good as using the datasheet equation and a circuit model, but it may be close enough for many applications.

Figure 2 shows that the error falls within ±2 ºC in the range of –20 ºC through +60 ºC. Note that the error isn't symmetrical around the thermistor's T_0 = 25 ºC point where its resistance equals R_0 = 10 kΩ.

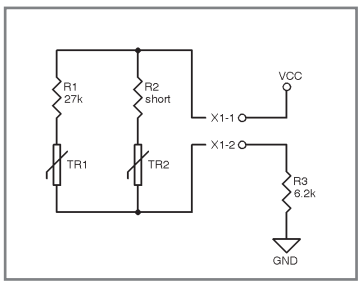


Figure 3—Adding a second thermistor and blending resistor improves the linearity of the voltage-vs-temperature curve. I found that R2 = 0 produces the best results with identical thermistors.

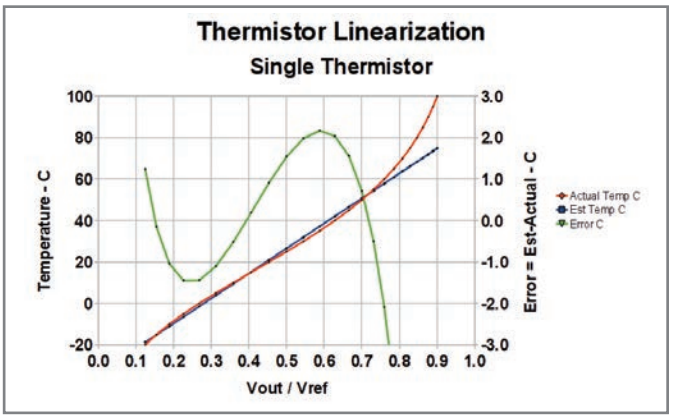


Figure 2—A single 10 kΩ resistor puts the error between a linear approximation and the actual temperature within ±2 ºC over a useful range.

I found an old U.S. patent awarded to Yellow Springs Instruments (from the days when they made discrete thermistors) that taught the technique of using multiple thermistors with different temperature characteristics to improve linearity over a wide temperature range. Because my parts heap has very few thermistors with known characteristics, I built the circuit in Figure 3 using two *identical* thermistors and a blending resistor. Surprisingly, it produced the result in Figure 4: the error remains within about ±1 ºC between –10 ºC and +60 ºC. Spreadsheets in the downloadable file for this column show the details behind those graphs.

The linearized dual-thermistor circuit was sufficiently accurate for my purposes, so I assembled a pair of thermistors and a resistor, cast them into a thin epoxy brick, and glued it to the aluminum slab as shown in Photo 1. If you need better accuracy, you must also have sufficiently good instrumentation to verify your approximations and equations against actual temperature.

Now that the firmware can measure the MOSFET's temperature, it should be able to heat or cool the transistor to a preset temperature.

PELTIER POWER

Because the circuit I'll eventually build with the MOSFETs must operate in both winter and summer conditions, the tester must cool and heat the transistors. The schematic in Figure 5

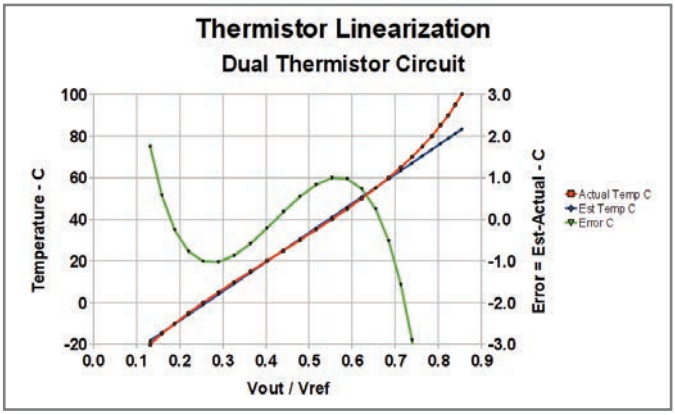


Figure 4—Adding a second thermistor and blending resistor reduces the error to ±1 ºC over the same temperature range as Figure 2.

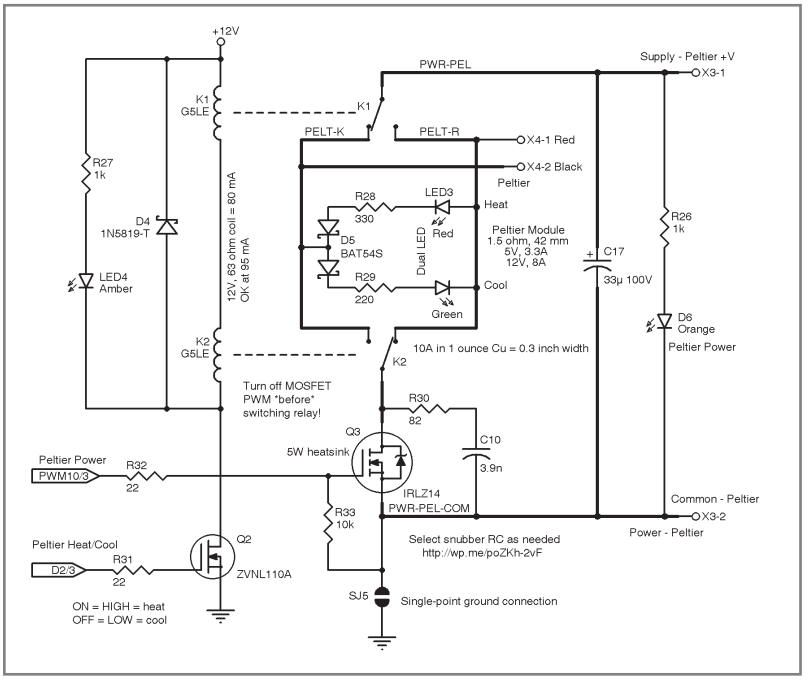


Figure 5—A pair of SPDT relays route high current through the Peltier module to allow both heating and cooling, while the MOSFET switch applies PWM switching to control the amount of power. The resonant RC snubber formed by R30 and C10 reduces ringing in the LC tank circuit formed by the module, wiring, and stray capacitance.

shows the circuitry to accomplish that task: a pair of SPDT relays direct the current through the Peltier module and Q3 is a logic-level IRLZ14 MOSFET switch. Although a full H-bridge would be quieter than relays, the module's 1.5 Ω resistance could draw more current than the solid-state bridges I had on hand.

The contacts have a 10 A carry current rating that allow the Peltier module to run from a 15 V supply, but their 8 A DC switching current rating requires cold switching. Therefore, the firmware turns Q3 off before switching the relays, with a delay to ensure the relay contacts settle before turning Q3 on again. Also, remember that a relay's AC current rating will be much higher than its DC rating, because AC current extinguishes the contact arc at each zero crossing, while DC current sustains the arc until the contacts fully open; always use the right spec!

For the first tests I used a 5 V supply for the Peltier module, which produced a 3 A drain current well within the MOSFET and relay ratings. The Peltier's heating and cooling capacity depends directly on its current, but, somewhat to my surprise, it could reach 0 ºC and 50 ºC from the 15 ºC ambient conditions on my workbench. In warmer weather, the module may require an 8 V supply rated for 6 A to reach the lowest temperatures. Even though that's less current than I expected when I laid out the circuitry, I still soldered copper wires along the traces to improve their current-carrying capacity.

I harvested the black aluminum heatsink in Photo 1 from an obsolete PC. A small fan on its finned lower surface can hold it within 5 ºC of ambient temperature with the Peltier module running at 15 W. The tradeoff between more Peltier power, additional waste heat, total cooling capacity, and heatsink size requires more analysis, but this simple setup worked well enough.

The Arduino output pin directly drives Q3's gate with a PWM signal, taking advantage of the IRLZ14 logic-level gate drive spec. As you saw in the August column, I set the PWM outputs to 32 kHz to simplify filtering the gate voltage and drain current control voltages, so the gate drive PWM for the Peltier module also runs at 32 kHz.

Circuit Cellar's technical reviewer suggested that much lower PWM frequencies, on the order of 1 Hz, will produce much better Peltier module efficiency. The Arduino PWM hardware can't generate that low a frequency and I'm not ready to integrate a software PWM routine in the tester firmware, but it seems like a great topic for a future column!

Photo 2 shows the IRLZ14 drain voltage during one PWM cycle. I didn't expect much stray inductance in the Peltier module wiring, so the ringing when the MOSFET turned off took me by surprise. The initial spike topped out at the IRLZ14's 60 V drain breakdown specification, where the MOSFET body diode began conducting the 3 A current, and

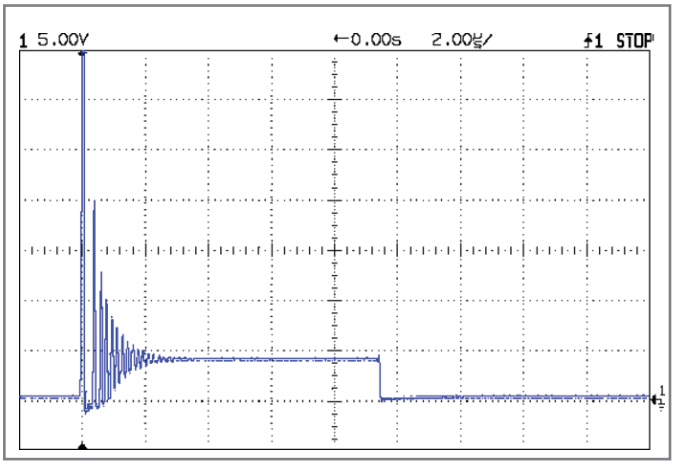


Photo 2—The drain of the Peltier drive transistor breaks down at about 60 V when the body diode begins conducting the 3 A drain current. The ringing continues until the circuit resistance dissipates the power.

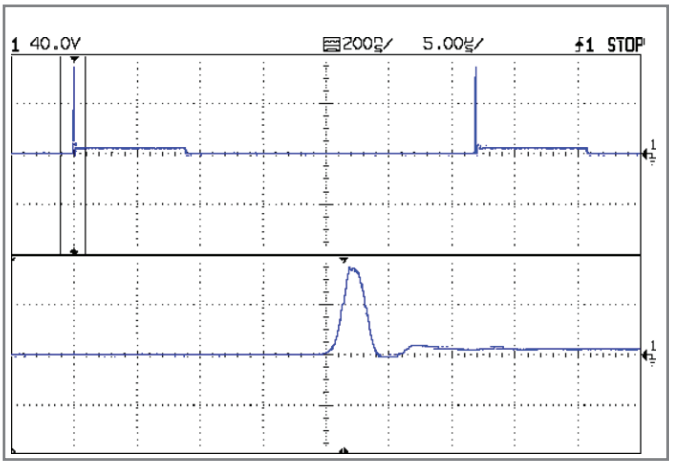


Photo 3—Adding a resonant-tuned RC snubber to the drain circuit eliminates the ringing by dissipating the power within one cycle. The lower trace shows a magnified view of the MOSFET's on-to-off transition.

ROUTE FASTER !



WITH PROTEUS PCB DESIGN

Our completely new manual router makes placing tracks quick and intuitive. During track placement the route will follow the mouse wherever possible and will intelligently move around obstacles while obeying the design rules.

All versions of Proteus also include an integrated world class shape based auto-router as standard.

PROTEUS DESIGN SUITE

Features:

- Hardware Accelerated Performance.
- Unique Thru-View™ Board Transparency.
- Over 35k Schematic & PCB library parts.
- Integrated Shape Based Auto-router.
- Flexible Design Rule Management.
- Polygonal and Split Power Plane Support.
- Board Autoplacement & Gateswap Optimiser.
- Direct CADCAM, ODB++, IDF & PDF Output.
- Integrated 3D Viewer with 3DS and DXF export.
- Mixed Mode SPICE Simulation Engine.
- Co-Simulation of PIC, AVR, 8051 and ARM7.
- Direct Technical Support at no additional cost.

Visit our website and use Promotional Code CLR2011JGB for an extra 10% Discount. Prices from just $249!

labcenter Electronics

www.labcenter.com

Labcenter Electronics Ltd. 411 Queen St. Suite 201, Newmarket, Ontario, Canada
Toll Free 866.499.8184, www.labcenter. com or Email: info@labcenter-electronics.com

the ringing continues until the Peltier module's resistance dissipates the stored energy.

Photo 3 shows the effect of adding the resonant-tuned RC snubber formed by R30 and C10: a single peak followed by one cycle of ringing. The initial peak still reaches the MOSFET's body diode breakdown voltage, but the relatively high snubber resistance quickly dissipates the remaining energy. I discussed the method for determining resonant snubber RC values in my August 2009 column and the references include a link to a Maxim app note explaining the subject in more detail.

With the Peltier module in place, the firmware can measure the MOSFET temperature, set the relays to heat or cool, and apply power. All that's left is to make the test temperature match the desired value: it's time for some servo control.

TEMPERATURE CONTROL LOOP

The MOSFET tester program has three nested loops:

- Temperature
- Gate Voltage
- Maximum Drain Current

Each pass though the Arduino `loop()` function requires a 100 ms delay to stabilize the test conditions before each measurement, which means the `SetPeltier()` function in Listing 1 runs every 100 ms. The aluminum block between the test MOSFET and the Peltier module provides enough thermal mass to reduce abrupt temperature changes, but that mass also prohibits rapid control updates.

The `SetPeltier()` function compares the current temperature with the desired setpoint to determine both the direction of current into the Peltier module and the PWM duty cycle. I used only the proportional and integral terms of the usual proportional-integral-differential (PID) control loop, because some experimentation

Listing 1—The main loop calls this routine every 100 ms to set the Peltier module current based on the error between the measured temperature and the desired setpoint. The code adds a slight delay after switching the relays to ensure that the contacts have settled in place.

```
int SetPeltier(float TNow, float TSet) {

float TErr, TErrMag;
int TSign;
float PelDrive;

int EnableHeat,OldEnableHeat;
static float Integral;
int TZone;
int PWM, PWMSigned;

    TErr = TNow - TSet;              // what is the temperature error
    TErrMag = abs(TErr);             //  ... magnitude
    TSign = sign(TErr);              //  ... direction

    if (TErrMag >= T_LIMIT)          // beyond outer limit
      TZone = 3;
    else if (TErrMag >= T_DEADBAND)  // beyond deadband
      TZone = 2;
    else if (TErrMag >= T_DEADBAND/2) // within deadband
      TZone = 1;
    else                             // pretty close to spot on
      TZone = 0;

    switch (TZone) {
      case 3:                // beyond outer limit
        PelDrive = TSign;    //  drive hard: -1 heat +1 cool
        Integral = 0.0;      //  no integration this far out
        break;
      case 2:                // beyond deadband
      case 1:                // within deadband
      case 0:                // inner deadband
        Integral += TErr;    // integrate the offset
        PelDrive = T_PGAIN*TErr + T_IGAIN*Integral;    // use PI control
       break;
      default:               // huh? should not happen...
        PelDrive = 0.0;
        break;
    }

    EnableHeat = (PelDrive > 0.0) ? LOW : HIGH;  // need cooling or heating?
    OldEnableHeat = digitalRead(PIN_ENABLE_HEAT); // where is the relay now?

    if (OldEnableHeat != EnableHeat) {        // change from heating to cooling?
      analogWrite(PIN_SET_IPELTIER,0);// disable PWM to flip relay
      digitalWrite(PIN_ENABLE_HEAT,EnableHeat);
      delay(15);                              // relay operation + bounce
    }

    PWM = constrain(((abs(PelDrive) * AO_PEL_SCALE) +
    AO_PEL_OFFSET),0.0,255.0);
    analogWrite(PIN_SET_IPELTIER,PWM);
    return (TZone <= 1);
}
```

IMMERSE YOURSELF



Santa Clara Convention Center Oct 30-Nov 1, 2012

Welcome to the premier annual industry event for engineers and developers of next-generation smart, connected devices. Explore chip and system design. Connect with software development pros. Ponder the next phase of innovation in IT for the enterprise. It's all here at ARM® TechCon™, where the future starts today.

Register online and **Save 15%** with promo code ARM15.
Register here: www.ARMTechCon.com

ARM® 2012 TechCon™
Join the community defining the future

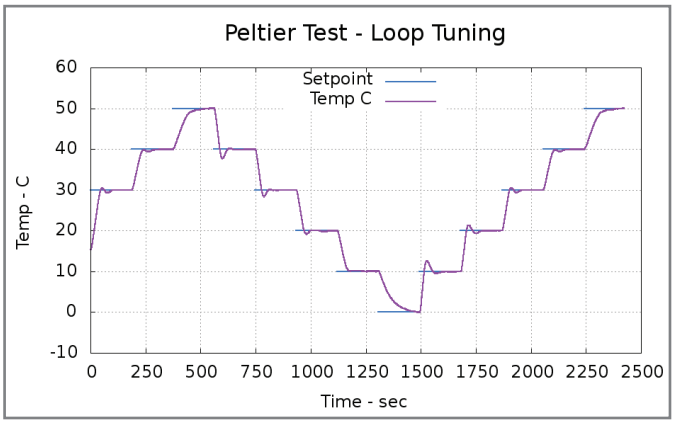


Photo 4—The Peltier module can change the temperature less than 0.3 ºC/s and obviously strains to reach 0 ºC and 50 ºC with a 5 V supply.

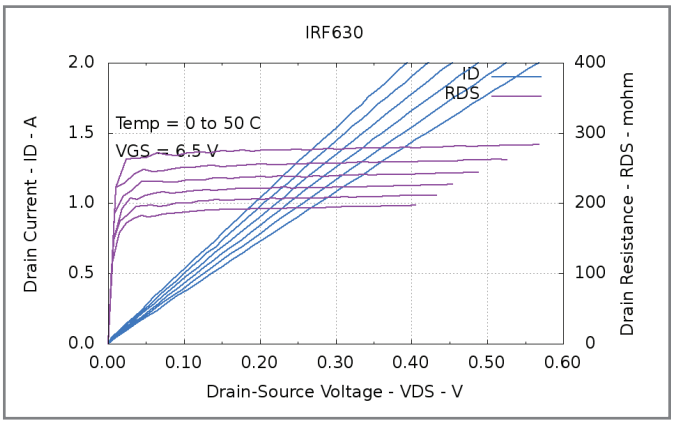


Photo 5—For a fixed VGS = 6.5 V, the R_{DS} of an IRF630 varies by about 40% between 0 ºC and 50 ºC. Lower resistances correspond to lower temperatures.

showed that temperature changes were so slow that computing and applying a differential term wouldn't improve the results.

The graph in Photo 4 shows the output of a test routine that stepped the temperature by 10 ºC every 180 s. The control loop applies full power (PWM = 1.0) to the Peltier module when the temperature error exceeds `T_LIMIT` = 3 ºC, but the initial slope of each temperature step is less than 0.3 ºC/s. That means the temperature changes by less than 0.03 ºC between each `SetPeltier()` call, a difference well down in the measurement noise.

Homework: Knowing the Arduino's ADC voltage resolution, use the circuit model and thermistor data to find the actual temperature resolution.

Only when the temperature error drops below `T_LIMIT` does the loop begins applying proportional-integral control. The proportional term uses only the temperature error: larger errors produce a higher duty cycles. However, proportional control can result in a fixed offset: the `T_PGAIN` factor may not reduce the error to zero under all conditions.

Therefore, the loop also applies integral control that depends on the accumulated error. The loop adds the temperature error to the `Integral` variable on each iteration, then adjusts the duty cycle using `T_IGAIN` factor. As a result, the integral term will eventually drive any constant error to zero. You can see that in action after each of the overshoots following some temperature steps.

Although the overshoots suggest that more loop tuning is in order, the actual overshoot magnitude is quite small. The main loop begins taking measurements when the temperature is within 1.5 ºC of the setpoint and pauses if the temperature exceeds that limit until the control loop reduces the error. Obviously, tighter temperature specifications would require closer control, but this suffices for my purposes.

The Peltier module obviously strains to reach both 0 ºC and 50 ºC with its 5 V supply. Speeding those transitions or widening the temperature limits would require more current and, quite likely, a larger heatsink and fan. As Captain Kirk always said, "We need more power, Scotty!"

However, given the range of voltages, currents, and temperatures that will confront my MOSFETs in actual use, the tester can produce meaningful results.

MOSFET CHARACTERISTICS VS. TEMPERATURE

The curves in Photo 5 show how the drain resistance of an IRF630 MOSFET varies with temperature at a fixed gate voltage; this is the same transistor you saw in my August column. The datasheet specifies R_{DS} < 400 mΩ at V_{GS} = 10 V, I_D = 4.5 A, and T_C = 25 ºC, so the 200 mΩ resistance shown here for V_{GS} = 6.5 V at 0 ºC may come as a surprise.

The resistance increases from 200 mΩ to 280 mΩ, about 40%, over the 50 ºC range. As you should expect, the change isn't linear, although it's around 0.8 %/ºC in this range.

The much higher drain resistance of the W7NB80 MOSFET in Photo 6 seems more suitable as a current-sensing resistor than the IRF630, at least for the relatively small currents that I expect in my circuit. The drain resistance varies about 0.9 %/ºC, but it looks close enough to being linear that a simple approximation should work.

At a given temperature and gate voltage the drain resistance of both transistors remains essentially constant through the entire current range I'm interested in. That's not surprising, because it's exactly what the MOSFET equations predict, but it's always comforting to see such things work out as expected.

Photos 5 and 6 show only the MOSFET's linear region, but the measured data also includes the active region with V_{GS} just above the cutoff region, where the transistor acts as a high-impedance constant-current source. Photo 6a from my August

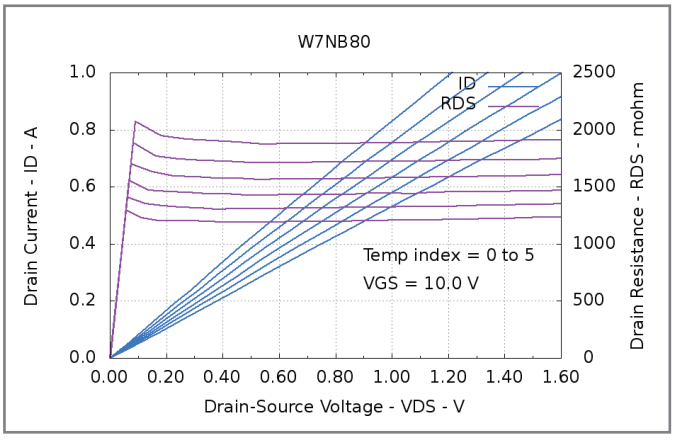


Photo 6—A W7NB80 MOSFET has a relatively high R_{DS} that also shows the expected variation with temperature.

CONNECT WITH

Circuit Cellar

For people who are passionate about hardware and software design, embedded development, and computer applications. Awesome projects, tutorials, industry news, design challenges, and more!



Connect.
Follow us on Twitter.
Like us on Facebook.

www.circuitcellar.com

@CircuitCellar
@editor_cc

circuitcellarmagazine

column shows the strongly nonlinear relation between V_{GS} and R_{DS} in that region. I suspect a reasonably simple approximation could predict the gate voltage required for a specific drain current, given the temperature and drain voltage, which would allow using the MOSFET as a variable resistor.

Of course, massaging the data to build that model is just a simple matter of software!

CONTACT RELEASE

The downloadable file for this column includes the test results for several transistors in my collection. The data won't apply to your transistors, but should give you an idea of how the tester firmware operates. The file also contains the source code and EAGLE design files.

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

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/267.

RESOURCES

Gnuplot, www.gnuplot.info.

R. W. Harruff, "Extended Range Thermistor Temperature Sensing," U.S. Patent 3316765, www.google.com/patents/US3316765.

Maxim Integrated Products, "Application Note 3835: CCFL Push-Pull Snubber Circuit," 2006, www.maxim-ic.com/app-notes/index.mvp/id/3835.

E. Nisley, "MOSFET Channel Resistance: Theory and Practice," *Circuit Cellar* 263, 2012.

_____, "MOSFET Channel Resistance: Tester Hardware," *Circuit Cellar* 265, 2012.

Yellow Springs Instruments, www.ysi.com.

SOURCES

Arduino microcontrollers
Arduino | http://arduino.cc

CadSoft Eagle PCB Software
CadSoft | www.cadsoftusa.com

AN3835 CCFL Push-pull snubber circuit
Maxim Integrated Products | www.maxim-ic.com

103CT-4 Thermistors
Semitec Corp. | www.semitec-usa.com/product/index.html

IRLZ14 MOSFET
Vishay Intertechnology, Inc.| www.vishay.com/docs/91325/sihlz14.pdf

Get PUBLISHED. Get NOTICED. Get PAID.

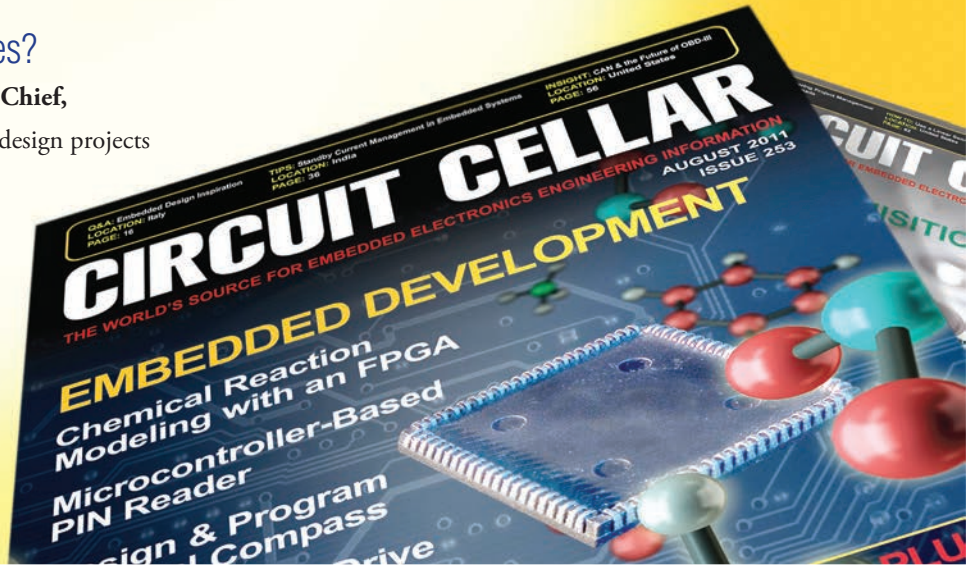
Circuit Cellar feature articles are contributed by professional engineers, academics, and students from around the globe. Each month, the editorial staff reviews dozens of article proposals and submissions. Only the best make it into the pages of this internationally respected magazine.

Do you have what it takes?

Contact **C. J. Abate, Editor-in-Chief,** today to discuss the embedded design projects and programming applications you've been working on and your article could be featured in an upcoming issue of *Circuit Cellar* magazine.

editor@circuitcellar.com

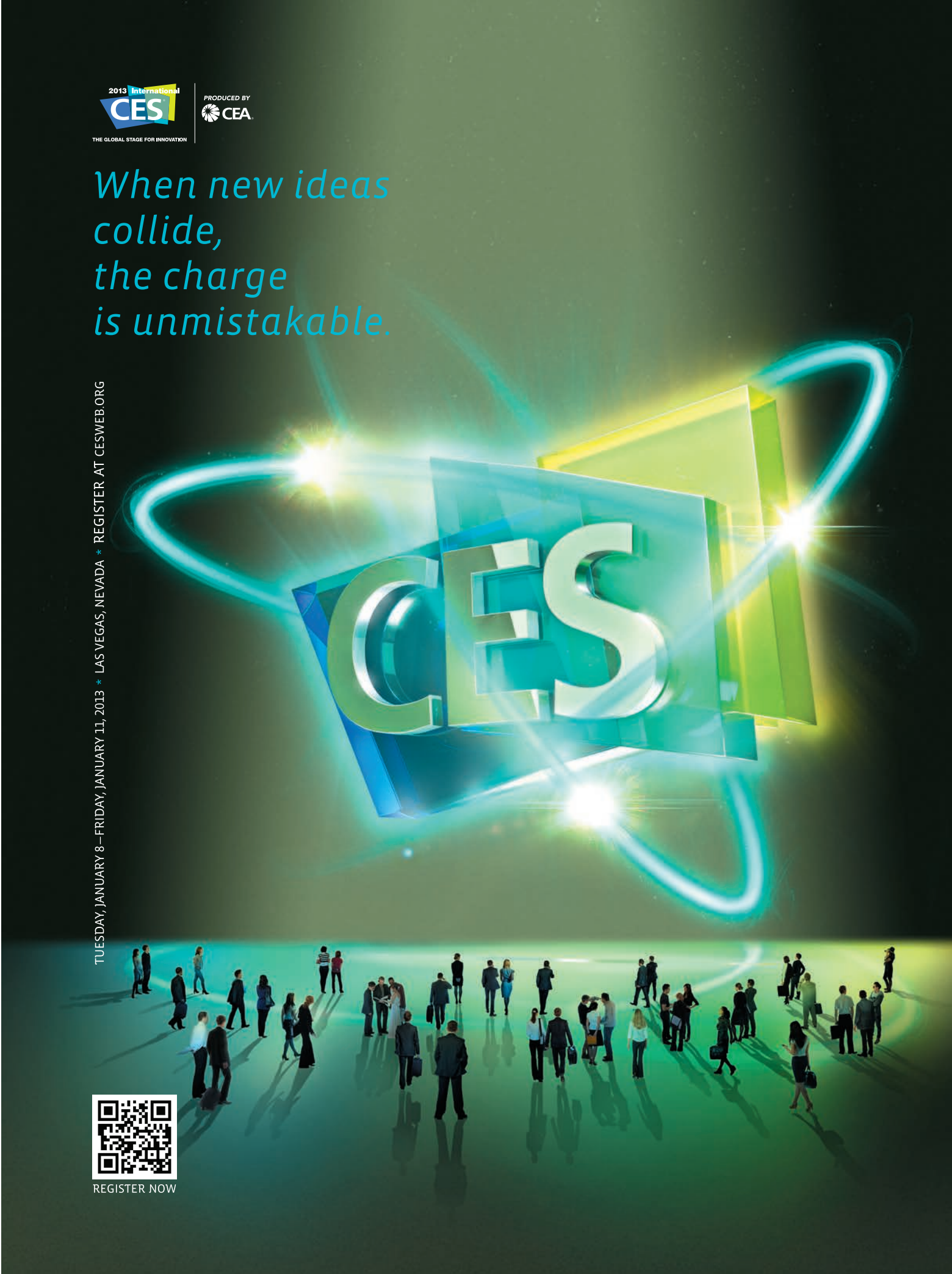
CIRCUIT CELLAR



2013 International CES
THE GLOBAL STAGE FOR INNOVATION

PRODUCED BY CEA

When new ideas collide, the charge is unmistakable.



TUESDAY, JANUARY 8—FRIDAY, JANUARY 11, 2013 * LAS VEGAS, NEVADA * REGISTER AT CESWEB.ORG

REGISTER NOW

BOARDS, BOOKS, DVDs AND MORE AT WWW.ELEKTOR.COM/SHOP

Elektor Shop

The world of electronics at your fingertips!

Books



NEW!

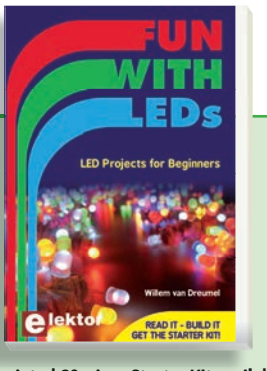
This book and more are available at www.elektor.com/books

LabWorX 2: Straight from the Lab to your Brain

Mastering Surface Mount Technology

Mastering Surface Mount Technology takes you on a crash course in techniques, tips and know-how to successfully introduce Surface Mount Technology in your workflow. Even if you are on a budget you too can jumpstart your designs with advanced fine pitch parts. Besides explaining methodology and equipment, attention is given to parts technology and soldering technique. Several projects introduce you step by step to handling surface mounted parts and the required technique to successfully build SMT assemblies. Many practical tips and tricks are disclosed that bring surface mounted technology into everyone's reach without breaking the bank.

282 pages • ISBN 978-1-907920-12-7 • $47.60

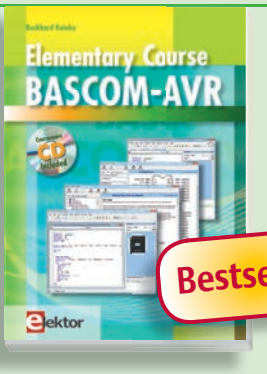


Associated 60-piece Starter Kit available

Fun with LEDs

This booklet presents more than twenty exciting projects covering LEDs, aimed at young & old. From an Air Writer, a Party Light, Running Lights, a LED Fader right up to a Christmas Tree. Use this book to replicate various projects and then put them into practice. To give you a head start each project is supported by a brief explanation, schematics and photos. A couple of projects employ the popular Arduino microcontroller board that's graced by a galaxy of open source applications. The optional 60-piece Starter Kit available with this book is a great way to get circuits built up and tested on a breadboard, i.e. without soldering.

96 pages • ISBN 978-1-907920-05-9 • $38.00



Bestseller!

Free Software CD-ROM included

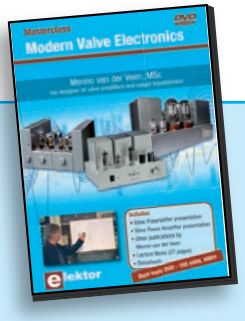
Elementary Course BASCOM-AVR

The Atmel AVR family of microcontrollers are extremely versatile and widely used. Elektor magazine already produced a wealth of special applications and circuit boards based on ATmega and ATtiny controllers. The majority of these projects perform a particular function. In this book however the programming of these controllers is the foremost concern. Using lots of practical examples we show how, using BASCOM, you can quickly get your own design ideas up and running in silicon.

224 pages • ISBN 978-1-907920-11-0 • $56.40

Prices and item descriptions subject to change. E. & O.E

CD/DVD-ROMs



Presented by Menno van der Veen

DVD Masterclass Modern Valve Electronics

This filmed seminar (165 mins. video and more) starts with a short discussion of the classic approach using valve load line graphs, followed by current sources and current foldback techniques. Next, the negative effect of cathode electrolytics is covered as well as reducing supply voltage interference. With the help of state of the art measurement techniques the (in)correctness of feedback is proven, while also clarifying what's happening deep within the core of the output transformer.

ISBN 978-1-907920-10-3 • $40.20



Circuits, ideas, tips and tricks from Elektor

CD 1001 Circuits

This CD-ROM contains more than 1000 circuits, ideas, tips and tricks from the Summer Circuits issues 2001-2010 of Elektor, supplemented with various other small projects, including all circuit diagrams, descriptions, component lists and full-sized layouts. The articles are grouped alphabetically in nine different sections: audio & video, computer & microcontroller, hobby & model-ling, home & garden, high frequency, power supply, robotics, test & measurement and of course a section miscellaneous for everything that didn't fit in one of the other sections. Texts and component lists may be searched with the search function of Adobe Reader.

ISBN 978-1-907920-06-6 • $55.70



Embedded Linux Made Easy

Today Linux can be found running on all sorts of devices, even coffee machines. Many electronics enthusiasts will be keen to use Linux as the basis of a new microcontroller project, but the apparent complexity of the operating system and the high price of development boards has been a hurdle. Here Elektor solves both these problems, with a beginners' course accompanied by a compact and inexpensive populated and tested circuit board. This board includes everything necessary for a modern embedded project: a USB interface, an SD card connection and various other expansion options. It is also easy to hook the board up to an Ethernet network.

Populated and tested Elektor Linux Board

Art.# 120026-91 • $93.30

Elektor is more than just your favorite electronics magazine. It's your one-stop shop for Elektor Books, CDs, DVDs, Kits & Modules and much more!

www.elektor.com/shop

Elektor USA
4 Park Street
Vernon, CT 06066
USA
Phone: 860-875-2199
Fax: 860-871-0411
E-mail: order@elektor.com



TAPIR Sniffs it Out

Attention boy scouts, professionals and grandfathers! This ultrasensitive wide-band E-smog detector offers you two extra senses to track down noise that's normally inaudible. TAPIR - short for Totally Archaic but Practical Interceptor of Radiation - also makes a nice project to build: the kit comprises everything you need - even the enclosure, ingeniously consisting of the PCB proper. Using the TAPIR is dead easy. Connect the headphones and an antenna and switch it on. Move it around any electrical device and you'll hear different noises with each device, depending on the type and frequency of the emitted field.

Kit of parts

Art.# 120354-71 • $21.50



AndroPod

With their high-resolution touchscreens, ample computing power, WLAN support and telephone functions, Android smartphones and tablets are ideal for use as control centres in your own projects. However, up to now it has been rather difficult to connect them to external circuitry. Our AndroPod interface board, which adds a serial TTL port and an RS485 port to the picture, changes this situation.

Andropod module with RS485 Extension

Art.# 110405-91 • $74.00

Kits & Modules

Embedded Inquiries

An Interview with Hai (Helen) Li



Computers have always fascinated Helen Li. As an undergraduate student at Tsinghua University in Beijing, China, she knew she wanted to learn more about computer principles. The more she learned, the more she wanted to explore. In August, Helen and I talked about her passion for learning, teaching, and researching computer engineering—from the real-time embedded system design and very large-scale integration (VLSI) classes she teaches at the Polytechnic Institute of New York University to the numerous grants she has received for research on spin-transfer torque random access memory (STT-RAM)-based memory hierarchy and memristor-based computing architecture.—Nan Price, Associate Editor

NAN: Where are you located?

HELEN: I currently live in Pittsburgh, PA. I grew up in China and, in 2000, I came to the U.S. to study for my PhD. After graduating from Purdue University in 2004, I took a job and stayed in the states.

NAN: When did you first become interested in computer engineering?

HELEN: In 1997, when I was an undergraduate student at Tsinghua University in Beijing, China, the public computer lab was open to all the students. Since then, I started to learn more and more about computers and their working principles, and I became interested in computer engineering.

NAN: What were some of your most notable experiences working for Intel, Qualcomm, and Seagate?

HELEN: The industrial working experience is very valuable to my whole career life. At Seagate, I led a design team on a test chip for emerging memory technologies. Communication and understanding between device engineers and design communities is extremely important. The joined effects from all the related disciplines (not just one particular area anymore) became necessary. The concept of cross layers (including device/circuit/architecture/system) cooptimization, and design continues in my research career.

NAN: In 2009, you transitioned from an engineering career to a career teaching electrical and computer engineering at the Polytechnic Institute of New York University (NYU). What prompted this change?

HELEN: After five years of working at various industrial companies on wireless communication, computer systems, and storage, I realized I am more interested in independent research and teaching. After careful consideration, I decided to return to an academic career and later joined the NYU faculty.

NAN: How long have you been teaching at the Polytechnic Institute of NYU? What courses do you currently teach and what do you enjoy most about teaching?

HELEN: I have been teaching at NYU-Poly since September 2009. My classes cover a wide range of computer engineering, from basic circuit design ("Introduction to VLSI"), to advanced computer architecture ("VLSI System and Architecture Design"), to system-level applications ("Real-Time Embedded System Design").

Though I have been teaching at NYU-Poly, I will be taking a one-year leave of absence from fall 2012 to summer 2013. During this time, I will continue my research on very large-scale integration (VLSI) and computer engineering at University of Pittsburgh.

I enjoy the interaction and discussions with students. They are very smart and creative. Those discussions always inspire new ideas. I learn so much from students.

NAN: You've received several grants from institutions including the National Science Foundation and the Air Force Research Lab to fund your embedded systems endeavors. Tell us a little about some of these research projects.

HELEN: The objective of the research

10th International System-on-Chip (SoC) Conference, Exhibit & Workshops

October 24 & 25, 2012 – Hilton Irvine/Orange County Airport

www.SoCconference.com

Early Bird Registration Is Now Open!

Don't Miss Out!

Platinum Sponsors

Microsemi

intel

UCIrvine | Extension

- Table-Top Exhibit (Free Passes)
- Memory Trends & Technologies
- Analog & Mixed-Signal Designs
- Green Chips
- Sub 20nm Design Challenges
- 3-D ICs
- Heterogeneous Multi-Core Designs
- Multicore Software Development
- SoC Design & Verification
- Many Door Prizes
- SOI vs. CMOS
- FPGAs –Trends & Designs
- Multicore SoC Platforms
- Memristors
- Network-on-Chips (NoCs)
- Informative Panels
- Mixed-Signal SoCs
- IEEE Student Design Contest
- Networking Opportunities
- And Much More. . .

Keynote Speakers



Altera
Dr. Mike Peng Li,
Fellow,
Altera Corporation.



HP
Dr. Janice Nickel,
Cognitive Systems Laboratory
at Hewlett Packard
Laboratories.



Microsemi
Jim Aralis,
Chief Technology Officer
(CTO),
and Vice President of R&D.



JPL
Dr. Reza Ghaffarian,
Jet Propulsion Laboratory,
California Institute of
Technology.



UCI
Dr. Payam Heydari,
University of California
Irvine.

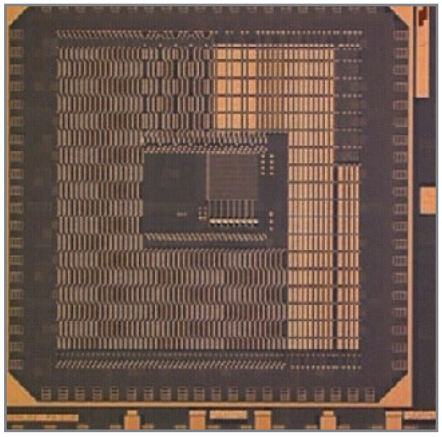
Promote Your Company, Technology, Products & Services at The Most Informative, Targeted & Affordably Priced Chip Design Conference, Exhibit & Workshops of the Year!

Selected Participating Companies & Universities

Western Digital · Altera · HP · Microsemi · Intel · JPL Jet Propulsion Laboratory California Institute of Technology · ST · IEEE Advancing Technology for Humanity · XFAB · Synopsys · XFAB · Target The ASIP Company · ARM · Xilinx · S3 Group · Freescale semiconductor · Breker The SoC Verification Company · Toshiba · Calypto Design • Optimize • Verify · AMD · WiGig · The University of Tokyo · EAG Evans Analytical Group · Qualcomm · UCI University of California, Irvine · SOI industry consortium · CEA leti

For More Information or Questions, Please Contact the SoC Conference Organizing Committee at:
SoC@SoCconference.com or (949) 851-1714

www.SavantCompany.com & www.SoCconference.com



Helen and her students are working on developing a 16-Kb STT-RAM test chip.

for "CAREER: STT-RAM-based Memory Hierarchy and Management in Embedded Systems" is to develop an innovative spin-transfer torque random access memory (STT-RAM)-based memory hierarchy to meet the demands of modern embedded systems for low-power, fast-speed, and high-density on-chip data storage.

This research provides a comprehensive design package for efficiently integrating STT-RAM into modern embedded system designs and offers unparalleled performance and power advantages. System architects and circuit designers can be well bridged and educated by the research innovations. The developed techniques can be directly transferred to industry applications under close collaborations with several industry partners, and directly impact future embedded systems. The activities in the collaboration also include tutorials at the major conferences on the technical aspects of the projects and new course development.

The main goal of the research for "CSR: Small Collaborative Research: Cross-Layer Design Techniques for Robustness of the Next-Generation Nonvolatile Memories" is to develop design technologies that can alleviate the general robustness issues of next-generation nonvolatile memories (NVMs) while maintaining and even improving generic memory specifications (e.g., density, power, and performance). Comprehensive solutions are integrated from architecture, circuit, and device layers for the improvement on the density, cost, and reliability of emerging nonvolatile memories.

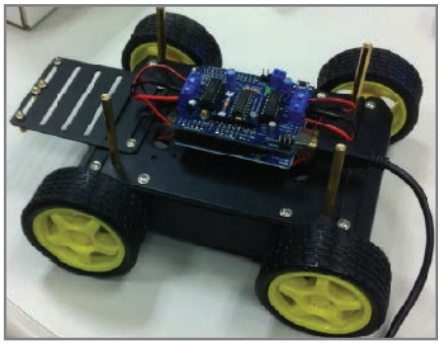
The broader impact of the research lies in revealing the importance of applying cross-layer design techniques to resolve the robustness issues of the next-generation NVMs and the attentions to the robust design context.

The research for "Memristor-Based Computing Architecture: Design Methodologies and Circuit Techniques" was inspired by memristors, which have recently attracted increased attention since the first real device was discovered by Hewlett-Packard Laboratories (HP Labs) in 2008. Their distinctive memristive characteristic creates great potentials in future computing system design. Our objective is to investigate process-variation aware memristor modeling, design methodology for memristor-based computing architecture, and exploitation of circuit techniques to improve reliability and density.

The scope of this effort is to build an integrated design environment for memristor-based computing architecture, which will provide memristor modeling and design flow to circuit and architecture designers. We will also develop and implement circuit techniques to achieve a more reliable and efficient system.

"Memristors have recently attracted increased attention since the first real device was discovered by Hewlett-Packard Laboratories (HP Labs) in 2008. Their distinctive memristive characteristic creates great potentials in future computing system design.

NAN: As part of a team, you have applied for and received many patents from "Three-Dimensionally Stacked Nonvolatile Memory Unit," to "Defective Bit Scheme for Multilayer Integrated Memory Device" to "Bit Set Modes for a Resistive Sense Memory Cell Array" to "Write Current Compensation Using Word Line Boosting Circuitry." How do you work together to come up with patent ideas?



An electric car model controlled by programmable emerging memories is in the developmental stages.

HELEN: My colleagues and I communicate and brainstorm to come up with creative ideas. The discussions happened everywhere: meetings, coffee breaks, lunches, and dinners. Our success came from at least two factors. First, our different backgrounds and complementary expertise help a lot from technical perspective. Second, we have a great passion for believing what we are doing and working hard to enable it.

NAN: You have contributed to many books and journal publications and, along with Yiren Chen, coauthored a book about flash memory, *Nonvolatile Memory Design: Magnetic, Resistive, and Phase Changing*, CRC Press, 2011. Tell us about the book and what readers can expect to gain from it.

HELEN: The book introduces three promising candidates of next-generation nonvolatile memory technologies: phase-change memory (PCM), magnetic random access memory (MRAM), and resistive random access memory (RRAM). One important mutant of RRAM—memristor—is also introduced as a special "analog" memory device.

The book illustrates the fundamental storage mechanism of those nonvolatile technologies and shows the exciting difference from CMOS flash memory techniques. The practical illustrative examples of nonvolatile memory design are shown insightfully, aligned with the

> "The explosive increase in the functional variety of modern systems motivated many circuit and architecture-level innovations. However, since processing elements's functions are determined at system architect and design stages, any changes beyond the existing multiprocessor system-on-chip's capability require architecture change, circuit redesign, or even new chip fabrication. I believe neuromorphic computing, which is inspired by the working mechanism of human brains, will solve the contradiction between the computing systems and the ever-increasing variety of applications."

unique characteristics of the discussed memory technologies. The key design methodologies, as well as the various functions and capabilities of three nonvolatile memory technologies, are deeply discussed, based on the latest technical advances.

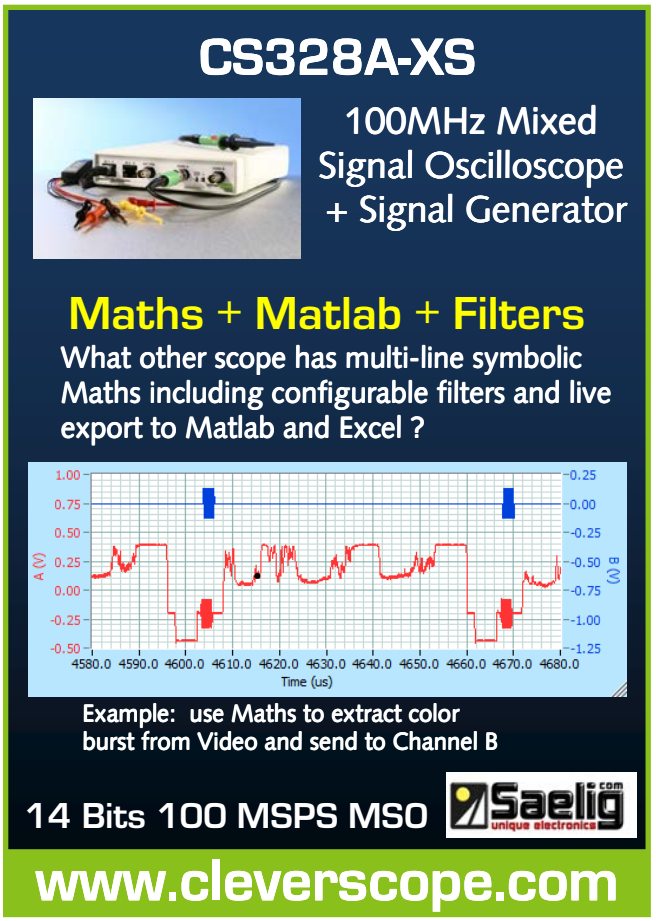
NAN: What types of projects are you and your students currently working on?

HELEN: Our major efforts are on device modeling, circuit design techniques, and novel architectures for computer systems and embedded systems. We primarily focus on the potentials of emerging devices and leveraging their advantages. Two of our latest projects are a 16-Kb STT-RAM test chip and an electric car model controlled by programmable emerging memories.

NAN: What are your thoughts on the future of computing?

HELEN: The explosive increase in the functional variety of modern systems motivated many circuit and architecture-level innovations. However, since processing elements' functions are determined at system architect and design stages, any changes beyond the existing multiprocessor system-on-chip's (MPSoC) capability require architecture change, circuit redesign, or even new chip fabrication. I believe neuromorphic computing, which is inspired by the working mechanism of human brains, will solve the contradiction between the computing systems and the ever-increasing variety of applications. Besides the existing efforts on neural-network algorithm enhancement and the system implementations built on conventional computing units, emerging technologies—such as memristors—will inspire new design techniques and novel architecture and lead to a revolutionary direction.





by Robert Lacoste (France)



Introduction to IIR Filters

Infinite impulse response (IIR) filters are feedback systems that use their outputs as inputs. It's easy to design a simple IIR filter and program it in your application if you know what tools to use. This article uses some simple math to provide the basics on IIR filters and explains the difference and similarities between IIR filters, finite impulse response (FIR) filters, and analog filters.

Hello, all. Welcome back to The Darker Side. I started this column in August 2007 ("Let's Play with EMI!," *Circuit Cellar*, 205). At that time, I thought I would have interesting electronics engineering topics to share with you for a year, maybe two. I am delighted to still be in front of my keyboard five years later. But, it's thanks to you. Over these five years, the great feedback and e-mail discussions with you, *Circuit Cellar* readers, have motivated me to continue! Moreover, my article topics are often inspired by readers' comments.

This month's article is a good example. My second Darker Side article was about finite impulse response (FIR) digital filters ("No Fear with FIR: Put a FIR Filter to Work," *Circuit Cellar* 207, 2007). I received five or six e-mails after it was published. Two of you asked me to write something about the other kind of digital filters—the so-called infinite impulse response (IIR) filters. At the time, I was reluctant. I found it difficult to present IIR filters in a short article, probably because I was not familiar enough with them. Time has passed, and now I will try to fulfill your requests.

One of the difficult things about IIR filters is that they are usually presented through exotic mathematical concepts (e.g., poles and zeros placement, Z-transforms, etc.) While these methods are invaluable, I want to explain the basics of IIR filters using simple words, math, and electronic concepts. Relax and enjoy. I think I found a solution. In this article, I will use the much-loved "soldering iron language" to show that IIR filters are similar to classic analog filters.

FIR & IIR

Let's start with the basics. A digital filter is a piece of software, firmware, or logic circuit that

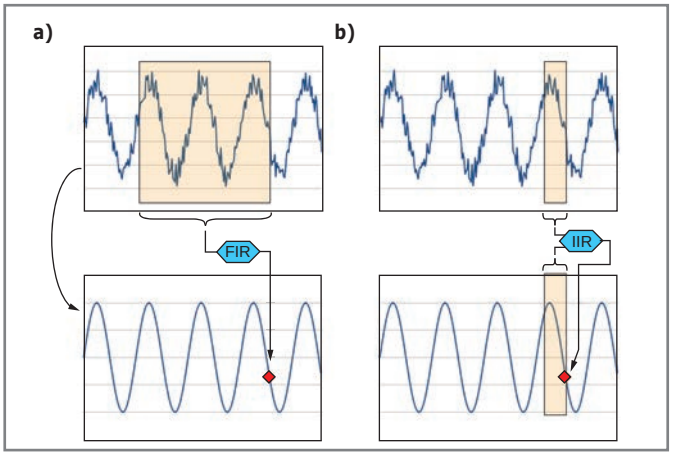


Figure 1—A FIR filter calculates the output signal's next value based on the input signal's fixed-duration segment. In contrast, an IIR filter uses the input signal's previous values and previously calculated output values. This introduces a kind of feedback and enables the use of smaller data sets for the same performance.

```
For each input sample do
    Average = 0
    for i = 0 to 9 do average = average+inputsample[N-i]
    outputsample = Average/10
```

Figure 2—Produce this algorithm by building a moving average filter by calculating the average of the last 10 input values.

```
outputsample=first input value
    For each input sample do
    outputsample = (1-x) * previous outputsample + x * new inputsample
```

Figure 3—Produce this algorithm by building a moving average filter by calculating the last 10 input values' average.

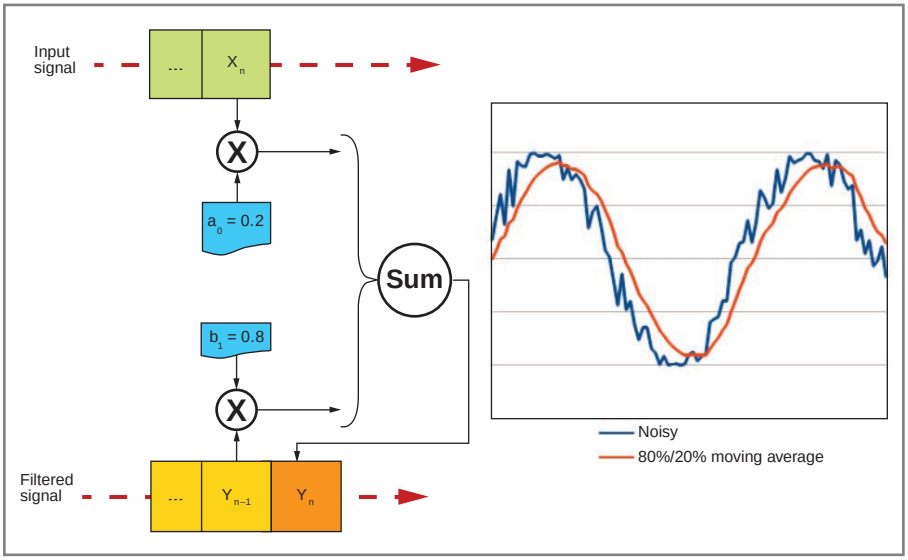


Figure 4—A moving average filter is a basic example of an IIR filter. Here, the next output value is calculated as a weighted average of the input signal (20%) and previous output (80%). This implements a simple low-pass filter.

takes a digital data flow as an input and provides a filtered version of this signal on its output. As we live in a digital world, this data flow is a discrete-time signal (i.e., its successive values represent the signal value at discrete time steps). There's nothing new here. However, as an efficient time machine has yet to be invented, at a given time step, a real-time digital filter must calculate the next output value based only on older values. There are two ways to do this. The simplest one uses the FIR filter presented in my 207 article. As indicated by the article title, this type of FIR filter has a finite "memory" over time. In practical words, a FIR filter takes the N previous input samples and uses them to calculate the next output value (see Figure 1a). For example, you can build a moving average filter by calculating the average of the last 10 input values. Its algorithm would look like what is shown in Figure 2.

That's a kind of FIR filter, as the new output value is calculated as a linear combination of a fixed number of previous input values, here 10. An IIR filter uses a slightly more complex, but more powerful approach. Rather than just using the previous input values, an IIR filter calculates the next output value based on the previous input values and the previously calculated output values (see Figure 1b). So, an IIR filter is a feedback system. Its output is used as an input. As a consequence, an input change may influence the output for a long, theoretically infinite, time. Let's go back to the example of a moving average filter. Instead of calculating the last 10 input values' average, which is a FIR, you could calculate a moving average by adjusting an already calculated average based on the next input value. This will give you a basic IIR. The corresponding algorithm is shown in Figure 3.

Figure 4 shows this type of filter on a noisy sine signal with x = 0.2. Simply increase the x factor for a lower filtering factor, or decrease it to have a stronger filter. In this basic example, the output signal is calculated with only the last input and output values, multiplying each of them by a given constant and summing the results to get the new output value. But think about it, you could build more complex filters by adding more parameters. There is nothing to prevent you from using the same method but with a larger number of previous input and output values, not just the last one of each. This provides an IIR filter's generic form (see Figure 5). The principle is still the same simple algorithm. Store the N previous input samples and the previously calculated output values in a circular buffer, multiply each buffer entry with a corresponding numerical constant at each time step, and sum them all to get the new output value.

If you think about it, a FIR is a specific type of IIR, with all feedback coefficients set to zero. An IIR must be more powerful than a FIR, but what are IIR filters' advantages compared to FIR filters? Thanks to the feedback mechanism, the same filtering performance could be achieved with much fewer calculations. Look again at the moving

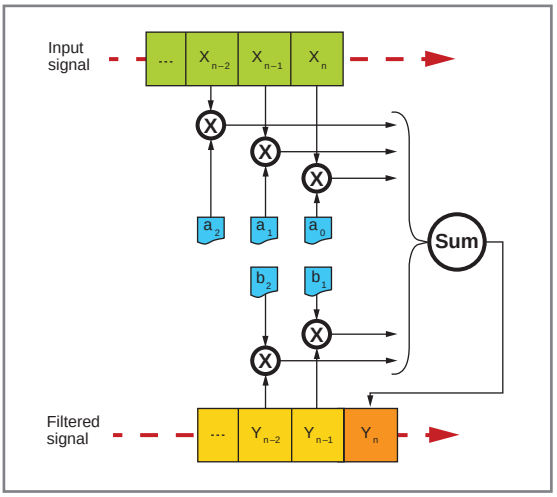


Figure 5—Here is the generic form of an IIR filter. The next output value is a linear combination of the previous N input and output values. The number N is the filter's rank, which is then defined by 2N-1 coefficients.

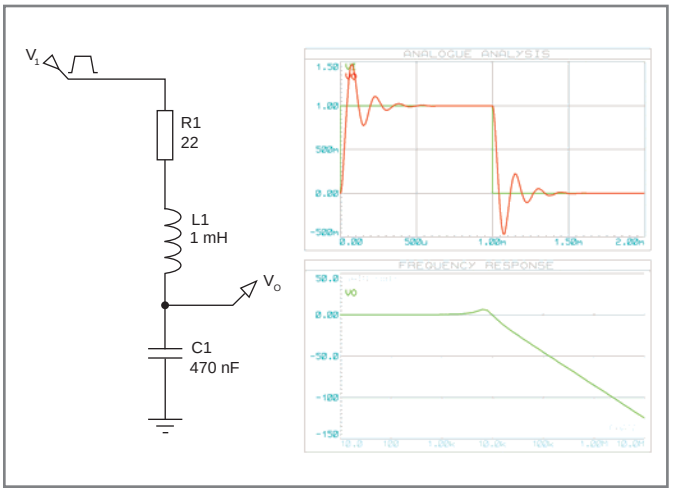


Figure 6—Labcenter's Proteus or any other SPICE-based simulator can be used to produce a simple RLC low-pass filter and its time and frequency domain simulation. Here the cutoff frequency is a little lower than 10 kHz.

average example. The FIR variant required 10 multiplications per sample, where the IIR version needs only two. So, an IIR filter costs far less than a FIR in terms of instructions per second or logic gates. This must be a disadvantage, right? You bet. First, a FIR filter can easily implement virtually any frequency response you wish, where designing an IIR for something other than classical filters (e.g., low pass, bandpass, high pass, or notch) is a real challenge. That's a limitation, even if this satisfies 99% of the applications. Second, it's difficult to find the IIRs coefficients' values. This is tricky, especially as an IIR filter may easily diverge or provide random results with improper coefficients. For example, you can check what would provide the "moving average" algorithm if the sum of both coefficients was accidentally a little higher than one. The output value will be infinite very soon. Last, as an IIR filter uses feedback, any error (e.g., truncation errors), may accumulate over time. This makes the designer's life difficult.

THE ANALOG WAY

As promised, I want to show you that an IIR filter is quite close to a classical analog filter. I will use the example of a basic low-pass RLC filter (see Figure 6). For this example, I selected a 1-MHz inductance and a 470-nF capacitor, which implies a resonant frequency of:

$$\frac{1}{\left(2\pi\sqrt{(LC)}\right)} = 7.34\ \text{kHz}$$

A small serial 22-Ω resistor attenuates this resonance a little, moving the 3-dB cutoff frequency up to around 11 kHz. I used Labcenter's Proteus design software to create the simulation shown in Figure 6. The transient analysis shows a classic low-pass characteristic with a large overshoot, which could be easily reduced with a larger resistor. But that's not my topic here, this is just an example. The corresponding simulation file is available on *Circuit Cellar*'s FTP site.

Now it's time for some calculations. Assume you only have a spreadsheet and you are asked to calculate the output of this filter with a defined input signal, like in the transient simulation just executed. What would you do? First, draw a circuit diagram like the one shown in Figure 7, reopen your textbook, take a deep breath, and I'll walk you through these simple calculations step by step. If you are really reluctant to use formulas, you could skip to the "real-world method" section, but you may miss something.

Let's start with the basic equations. First, Kirchhoff's voltage law says, the input voltage is the sum of the voltages across three parts:

$$V_I = V_R + V_L + V_O \quad [1]$$

Regarding the resistor, Ohm's law says, the voltage across the resistor is proportional to the current:

$$V_R = Ri \quad [2]$$

Then, even if you may have forgotten, the two equations governing the inductor and capacitor are:

$$V_L = L\frac{di}{dt} \quad [3]$$

and

$$i = C\frac{dV_O}{dt} \quad [4]$$

In these equations, the notation dx/dt means a derivative over time of the variable x, which means the slope of the curve x(t). The result of merging Equation 1, Equation 2, and Equation 3 is:

$$V_I = Ri + L\frac{di}{dt} + V_O \quad [5]$$

Let's replace the current i by its value provided in Equation 4. The result is:

$$V_I = RC\frac{dV_O}{dt} + LC\frac{d^2V_O}{dt^2} + V_O \quad [6]$$

This is called a second-order partial derivative equation, and it is the fundamental equation governing this RLC circuit. The input voltage V_I is linked to the output voltage V_O but also to its first and second derivative over time. You will find plenty of textbooks explaining how to mathematically calculate its solutions. Depending on the parameter's values (i.e., depending on the parts values), you will find either oscillating solutions or damped responses, as in any linear feedback system.

However, remember you have only a spreadsheet. Rather than trying to find an exact solution, take another approach, and try to approximate it with a discrete time-step method.

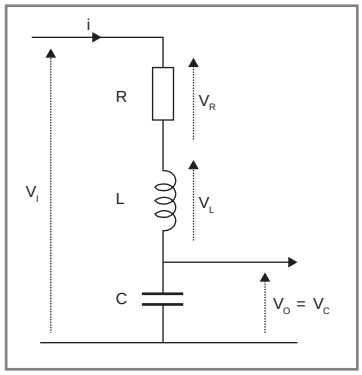
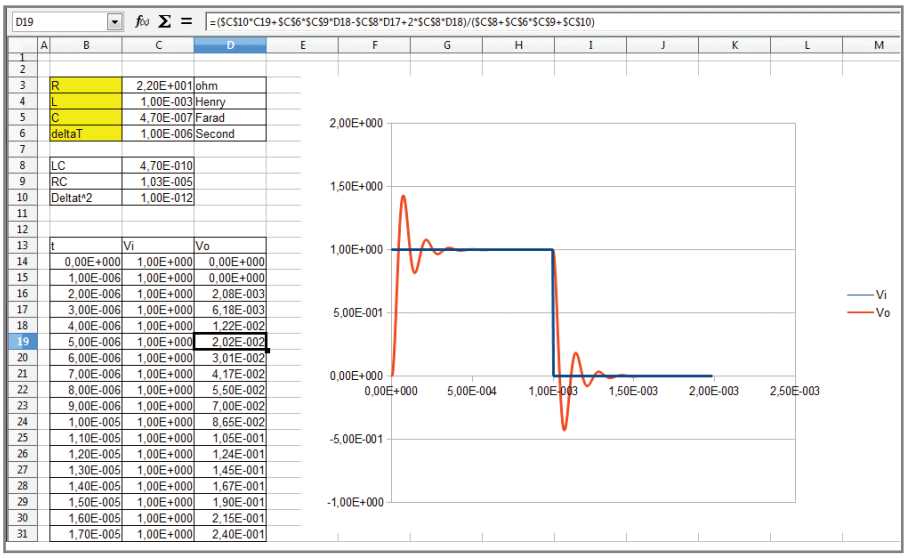


Figure 7—A model of the RLC filter enables you to calculate the output voltage based on the input signal.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2													
3		R	2,20E+001	ohm									
4		L	1,00E-003	Henry									
5		C	4,70E-007	Farad									
6		deltaT	1,00E-006	Second									
7													
8		LC	4,70E-010										
9		RC	1,03E-005										
10		Deltat^2	1,00E-012										
11													
12													
13		t	Vi	Vo									
14		0,00E+000	1,00E+000	0,00E+000									
15		1,00E-006	1,00E+000	0,00E+000									
16		2,00E-006	1,00E+000	2,08E-003									
17		3,00E-006	1,00E+000	6,18E-003									
18		4,00E-006	1,00E+000	1,22E-002									
19		5,00E-006	1,00E+000	2,02E-002									
20		6,00E-006	1,00E+000	3,01E-002									
21		7,00E-006	1,00E+000	4,17E-002									
22		8,00E-006	1,00E+000	5,50E-002									
23		9,00E-006	1,00E+000	7,00E-002									
24		1,00E-005	1,00E+000	8,65E-002									
25		1,10E-005	1,00E+000	1,05E-001									
26		1,20E-005	1,00E+000	1,24E-001									
27		1,30E-005	1,00E+000	1,45E-001									
28		1,40E-005	1,00E+000	1,67E-001									
29		1,50E-005	1,00E+000	1,90E-001									
30		1,60E-005	1,00E+000	2,15E-001									
31		1,70E-005	1,00E+000	2,40E-001									

Photo 1—Here is a spreadsheet version of the RLC filter transformed into an IIR filter. Look at the formula in the header bar. It is an exact transcription of the IIR calculation formula. The time-domain signal is, as expected, very close to the SPICE simulation.

The problem in this equation is the derivatives. But, if the time step is small compared to the signal variations' speed, the derivative could be approximated over time of a function by the difference of two successive samples divided by the time step:

$$\frac{dx}{dt} \approx \frac{x(n) - x(n-1)}{\Delta t} \quad [7]$$

Similarly (trust me, or calculate it yourself), the second derivative can be approximated by:

$$\frac{d^2x}{dt^2} \approx \frac{x(n) - 2x(n-1) + x(n-2)}{\Delta t^2} \quad [8]$$

Using these approximations, you could rewrite Equation 6 without any derivatives. It's a little boring, but I got :

$$V_I(n) = \frac{RC}{\Delta t}\left[V_O(n) - V_O(n-1)\right] + \frac{LC}{\Delta t_2}\left[V_O(n) - 2V_O(n-1) + V_O(n-2) + V_O(n)\right] \quad [9]$$

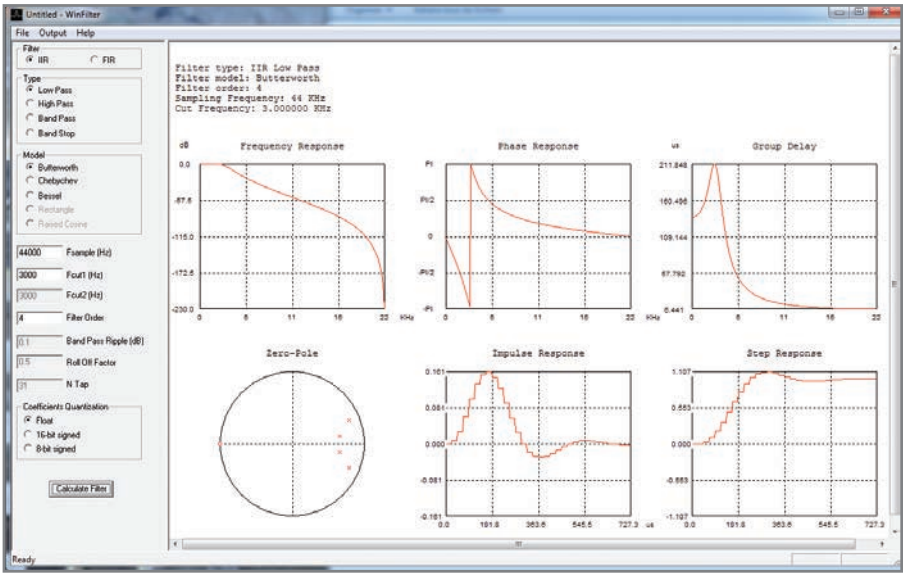
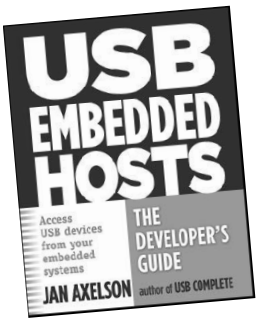


Photo 2—WinFilter enables you to calculate and simulate all kinds of IIR filters by entering their key characteristics (left). The plots shows the resulting frequency and time behavior.

Access USB Devices from your Embedded Systems



USB Embedded Hosts
The Developer's Guide
Jan Axelson
$29.95 LVR.COM
From the author of *USB Complete*

PIC-SERVO
MOTION CONTROL
MOTION CONTROLLERS FOR BRUSH, BRUSHLESS AND STEPPER MOTORS.
• controller chips
• controller boards
www.picservo.com
JEFFREY KERR, LLC

Still Using "Grandma" USB?
Try New Flipper® Connector
"Works Either Way & Costs Less..."*
Grandma USB uses too much plastics
Flipper contacts "float" in the middle
Product of USA
(Now coming to USA... Millions sold/already in use in Asia)
Try Flipper Sample/Demo Pack...only $25
(Regular $40, including 1x 4GB Flash drive, 2x Adapters, 10x Flipper male A connectors & FREE Shipping within the Continental USA.)
www.flipperUSB.com
info@flipperUSB.com
(310) 408-9711
Reps Wanted
Patented in US, Canada, Japan, Taiwan & China
* Costs less than regular USB from major distributors

BEST SCOPE SELECTION
& lowest prices!

WORLD'S SMALLEST
World's smallest MSO! This DIP-sized 200kHz 2-ch scope includes a spectrum analyzer and Arbitrary Waveform Gen. Measures only 1 x 1.6 inches in size!
XPROTOLAB $49

iPHONE SCOPE
5MHz mixed signal scope adapter for the iPhone, iPad and iPod Touch! The *FREE* iMSO-104 app is available for download from the Apple App Store.
IMSO-104 $297.99

25MHZ SCOPE
Remarkable low cost 25MHz, 2-ch plus trigger USB benchscope with 8-inch full color LCD display. Spectrum analysis and features autoscale functions.
PDS5022T $269

60MHZ SCOPE
60MHz 2-ch scope with 500MSa/s rate and huge 10MSa memory! 8" color TFT-LCD and *FREE* carry case!
SDS6062 $359

100MHZ SCOPE
High-end 100MHz 2-ch 1GSa/s benchscope with 1MSa memory and USB port + FREE scope carry case. New super low price!
DS1102E $399

100MHZ MSO
2-ch 100MSa/s scope + 8-ch logic analyzer. USB 2.0 and 4M samples storage per channel with advanced triggering & math functions.
CS328A $1359

HANDHELD 20MHZ
Fast & accurate handheld 20MHz 1-ch oscilloscope.
- 100 M/S sample rate
- 3.5 in. color TFT-LCD
- 6 hour battery life
FREE rugged, impact-resistant case!
HDS1021M $269.95

WWW.SAELIG.COM

Saelig.com
unique electronics

Listing 1—For the low-pass example filter, I used WinFilter-generated C code with 16-bit integer coefficients.

```
    /*****************************************************************
    WinFilter version 0.8
    http://www.winfilter.20m.com
    akundert@hotmail.com

    Filter type: Low Pass
    Filter model: Butterworth
    Filter order: 4
    Sampling Frequency: 44 KHz
    Cut Frequency: 3.000000 KHz
    Coefficents Quantization: 16-bit

    Z domain Zeros
    z = -1.000000 + j 0.000000
    z = -1.000000 + j 0.000000
    z = -1.000000 + j 0.000000
    z = -1.000000 + j 0.000000

    Z domain Poles
    z = 0.657133 + j -0.115245
    z = 0.657133 + j 0.115245
    z = 0.785067 + j -0.331076
    z = 0.785067 + j 0.331076
    *****************************************************************/
    #define NCoef 4
    #define DCgain 512

    __int16 iir(__int16 NewSample) {
        __int16 ACoef[NCoef+1] = {
             5315,
            21261,
            31892,
            21261,
             5315
        };

        __int16 BCoef[NCoef+1] = {
             8192,
            -23629,
            26498,
            -13541,
             2647
        };

        static __int32 y[NCoef+1]; //output samples
        //Warning!!!!!! This variable should be signed (input sample width +
Coefs width + 4 )-bit width to avoid saturation.

        static __int16 x[NCoef+1]; //input samples
        int n;

        //shift the old samples
        for(n=NCoef; n>0; n--) {
           x[n] = x[n-1];
           y[n] = y[n-1];
        }

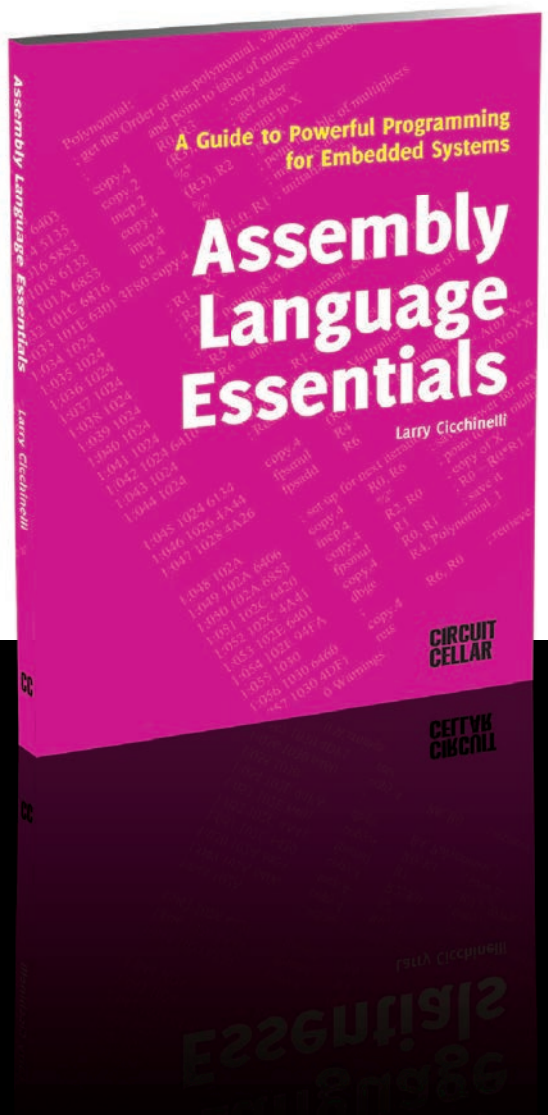
        //Calculate the new output
        x[0] = NewSample;
        y[0] = ACoef[0] * x[0];
        for(n=1; n<=NCoef; n++)
            y[0] += ACoef[n] * x[n] - BCoef[n] * y[n];

        y[0] /= BCoef[0];

        return y[0] / DCgain;
    }
```

Assembly Language Essentials

Circuit Cellar's first book, ***Assembly Language Essentials***, is a matter-of-fact guide to Assembly that will introduce you to the most fundamental programming language of a processor.



Author Larry Cicchinelli provides readers with:

- An introduction to Assembly language & its functionality
- Essential terminology pertaining to higher-level programming languages & computer architecture
- Important algorithms that may be built into high-level languages — multiplication, division, and polynomial evaluation
- Overview of Interrupt Service Routines
- Free, downloadable Assembler program …

and more!

On Sale NOW!

$47.50

WWW.CIRCUITCELLAR.COM • WWW.CC-WEBSHOP.COM

CIRCUIT CELLAR
THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

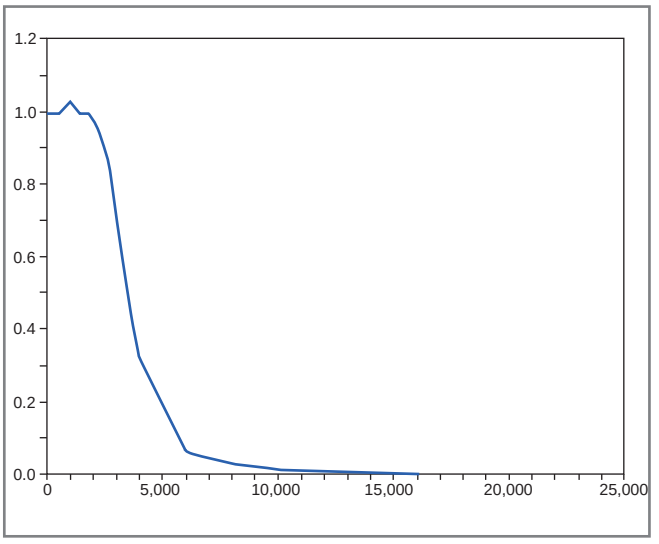


Figure 8—The WinFilter-based filter was simulated with SciLab. This graph shows the output amplitude with varying signal frequency, which is close to what is expected.

Just shuffle around these terms and we will find how to calculate the next output value $V_o(n)$ based on the other parameters :

$$V_O(n) = \frac{\Delta t^2}{a} x V_I(n) + \frac{RC\Delta t + 2LC}{a} x V_O(n-1) - \frac{LC}{a} x V_O(n-2) \quad [10]$$

with

$$a = LC + RC\Delta t + \Delta t 2^2$$

which is a constant.

Here we are! Take another look at this formula. The next output sample is calculated as the last input sample $V_I(N)$ multiplied by a given constant, plus the previous output value $V_O(n-1) \times$ another constant, plus the previous output value $V_O(n-2)$ also multiplied by a constant. Does it look familiar? Of course, this is exactly an IIR filter algorithm! I used The Document Foundation's LibreOffice Calc to code this calculation on a spreadsheet (see Photo 1). Compare it with the plot in Figure 6, which should convince you that the calculation was correct. In fact, all circuit simulators (e.g., SPICE) use these types of discrete-time methods to internally do their calculations, so it is no surprise.

REAL-WORLD METHOD

I have demonstrated how to construct an IIR filter from a standard analog filter. However, I'm sure you would not want to do this kind of calculation for a 12th-order bandpass filter. It would work, but it would require some long calculations.

Fortunately, there are solutions for the lazy engineer in the form of ready-to-run IIR filter synthesis software. I know at least three free options. Refer to the Resources section.

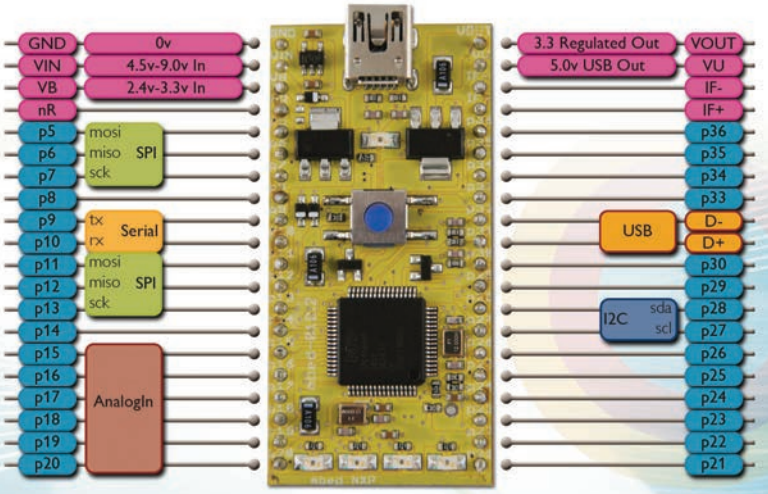
The first solution is an online web-based IIR filter calculator developed by Tony Fisher from the University of York, England. You don't need to install anything. You simply enter the type of filter you want (e.g., low pass, high pass, bandpass, or notch); select your preferred response (e.g., Bessel, Butterworth, or Chebyshev); enter the filter order, sampling frequency, and corner frequencies; and click on a browser button.

The second option is WinFilter software, named and developed by Adrian Kundert. Its last version is dated from 2004, but it still worked flawlessly on my Windows 7 PC and provided a great graphical interface. If you prefer open-source, generic-purpose software, you can also use SciLab Group's SciLab, which I use quite often in projects featured in this column. SciLab is like a free version of MathWorks's MATLAB, provided with a large library of signal-processing functions, including IIR synthesis routines. In SciLab Group's "Signal Processing with SciLab," there is a 35-page chapter on IIR filters. If you are not constrained by free solutions, you can also find some great, inexpensive commercial software (e.g., Iowegian International's ScopeIIR software or Fiwiz from Berkeley University's International Computer Science Institute).

I'll use WinFilter as an example. Assume you are working on a 44-ksps digitized sound signal and you need to

mbed

mbed NXP LPC11U24 Microcontroller



Rapid prototyping for USB Devices, Battery Powered designs and 32-bit ARM® Cortex™-M0 applications

http://mbed.org

build a solid 3,000-Hz Butterworth low-pass filter to separate human voice from high-frequency noise or to reduce the sample rate. Imagine your system requires an attenuation of at least 20 dB at 6 kHz (i.e., one octave above the cutoff frequency). A first-order filter provides an attenuation of roughly 6 dB per octave, so you will need at least a fourth-order low-pass filter (e.g., 4 × 6 dB = 24 dB). So, you have your specifications. Now just download, install, and launch WinFilter, then enter the data. Click on Calculate Filter, and the magic happens. Within some hundreds of milliseconds, the tool will provide a simulated IIR filter that will fulfill your requirements and show you the simulated filter amplitude, phase response, impulse response, and group delay (see Photo 2). This is an invaluable tool.

But the magic doesn't stop there. Open the Output menu and you will find options to generate the corresponding C code for you, or even a VHDL version if you are FPGA addict. With WinFilter, there is the option to generate floating-point or integer code. The 16-bit integer version of the code generated for our 3,000-Hz low-pass filter is shown in Listing 1. This one will be easy to implement in any microcontroller using 32-bit arithmetic. However, before using such a filter in, say, a life-critical application, you'd better check its actual response. Using either an integer implementation or a fixed-resolution floating-point version will inevitably imply truncation errors. And, if the filter is too critical (i.e., mathematically too close to the stability limit), unpleasant things may happen, as explained. This is difficult to anticipate without a closer look at the theory, but as a basic rule, the larger the pole number and the sharper the filter, the higher the risk. I must be honest. This is where IIR filters start to get complicated and nasty. A filter may work well with floating-point numbers and diverge when you port it to integers. This could be due to overflow errors but also due to truncation. Recursive algorithms such as IIR can sometimes be annoying.

To check the filter's behavior, I

"If a filter is too critical (i.e., mathematically too close to the stability limit), unpleasant things may happen, as explained. This is difficult to anticipate without a closer look at the theory, but as a basic rule, the larger the pole number and the sharper the filter, the higher the risk. I must be honest. This is where IIR filters start to get complicated and nasty. A filter may work well with floating-point numbers and diverge when you port it to integers. This could be due to overflow errors but also due to truncation. Recursive algorithms like infinite impulse response can sometimes be annoying."

Custom Front Panels & Enclosures
FREE Software
Front Panel Designer
Sample price $57.32 + S&H
Designed by you using our FREE software, Front Panel Designer
- Cost effective prototypes and production runs
- Powder-coated finish and panel thickness up to 10mm now available
- Choose from aluminum, acrylic or customer provided material
- 1, 3 and 5-day lead times available
FRONT PANEL EXPRESS
FrontPanelExpress.com
1(800)FPE-9060

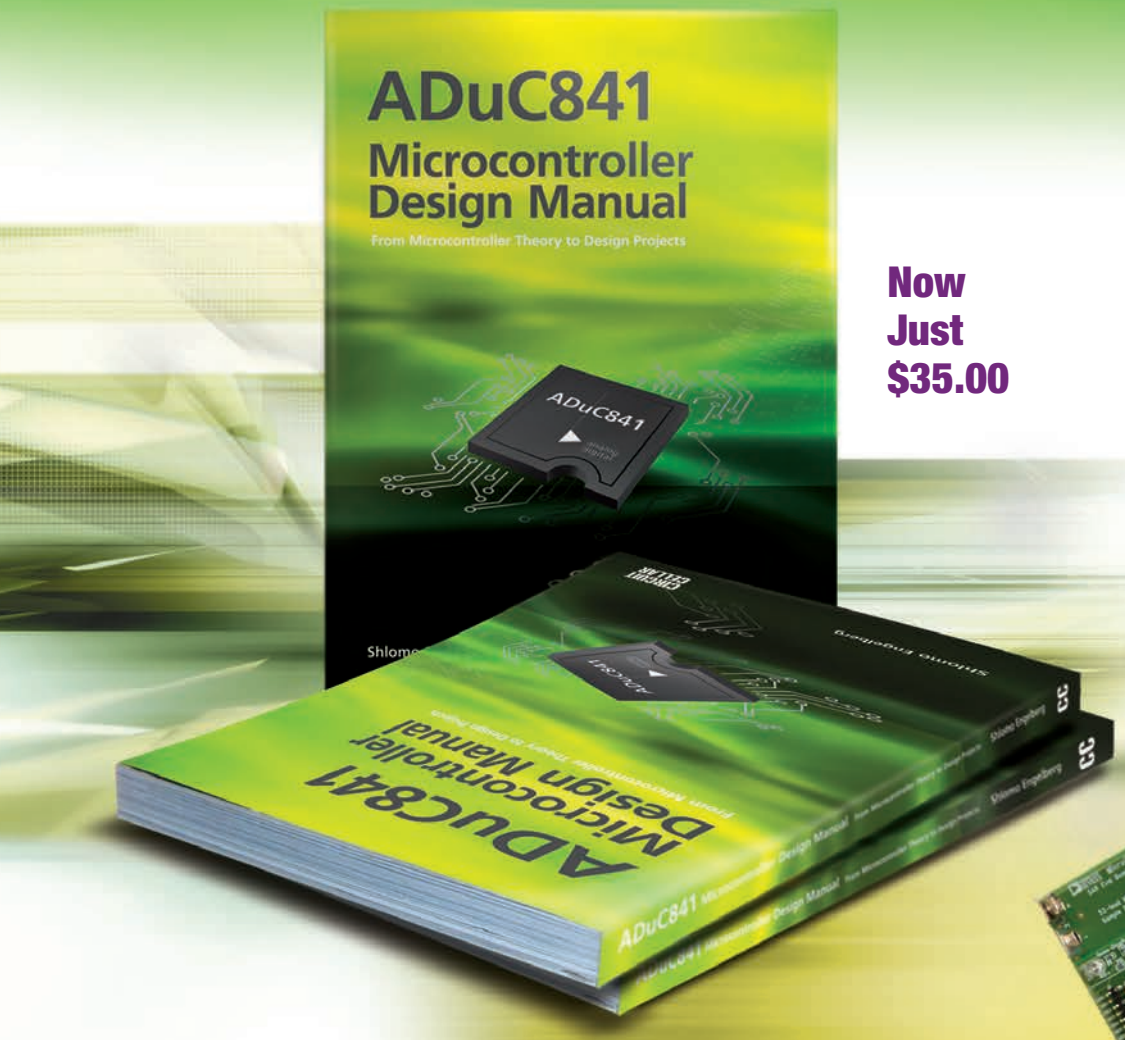
Gigabit Technology
- ARM-based
- System on a Chip
- Gigabit Ethernet
- Small, Cheap, Fast
- Both QFP and BGA Packages
- Standard Development Tools
- 10 Year Life Guarantee
- Royalty Free RTOS, TCP/IP V4/6
$10.00 each (Qty 100)
The gridARM™ System on a Chip (SOC) is a high performance, low cost, low power, highly integrated single chip with 10 / 100 / 1000 Mbps Ethernet, USB, CAN, Serial, SRAM Memory, SPI, I2C, RTC and internal peripherals designed to provide a complete solution for embedded applications.
THE NETWORKING EXPERTS
gridconnect
800.975.4743 USA • 1 630.245.1445
gridconnect.com/gridarm.html
Leaders in the embedded and networking marketplace providing network hardware, high quality software and services

CIRCUIT CELLAR

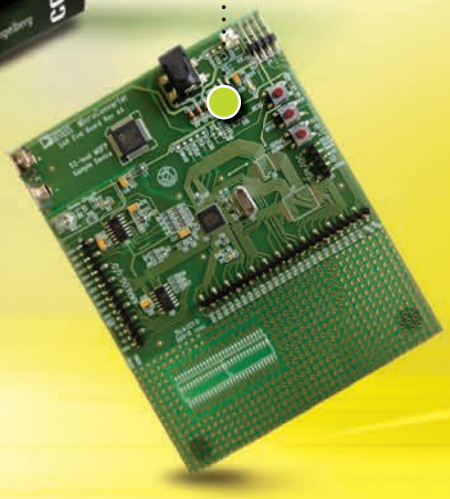
ADuC841 Microcontroller Design Manual:

From Microcontroller Theory to Design Projects

If you've ever wanted to design and program with the ADuC841 microcontroller, or other microcontrollers in the 8051 family, this is the book for you. With introductory and advanced labs, you'll soon master the many ways to use a microcontroller. Perfect for academics!



Now Just $35.00



Buy it today!

www.cc-webshop.com

translated it into a SciLab code for you. The results are available on *Circuit Cellar*'s FTP site. The code simply generates a sine signal at increasing frequencies, passes it through the IIR filter, and plots the output amplitude based on frequency. The result, which is close to the expectation, is shown in Figure 8.

WRAPPING UP

You can't really understand a feedback system's behavior—particularly IIR filters—without an actual mathematical analysis of the processing loop. I haven't even explained the meaning of a pole and a zero. For complex filters, you will need to understand stability concerns and mathematical tools (e.g., Laplace transforms) to optimize them or even to design a solid system. However, as I've tried to show you in this article, designing a simple IIR filter and programming it in your application can be simple, thanks to the available tools. You should try. It will take one more technique out of your darker side!

Robert Lacoste lives near Paris, France. He has 23 years of experience working on embedded systems, analog designs, and wireless telecommunications. He has won prizes in more than 15 international design contests. In 2003, Robert started a consulting company, ALCIOM, to share his passion for innovative mixed-signal designs. You can reach him at rlacoste@alciom.com. Don't forget to write "Darker Side" in the subject line to bypass his spam filters.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/267.

RESOURCES

Adrian Kundert, WinFilter, www.winfilter.20m.com.

The Document Foundation, Libre Office Calc, www.documentfoundation.org.

Tony Fisher, "Interactive Digital Filter Design," www-users.cs.york.ac.uk/~fisher/mkfilter.

The Scilab consortium, www.scilab.org.

SciLab Group, "Signal Processing with SciLab," http://wiki.scilab.org/Tutorials%20archives?action=AttachFile&do=view&target=signal.pdf.

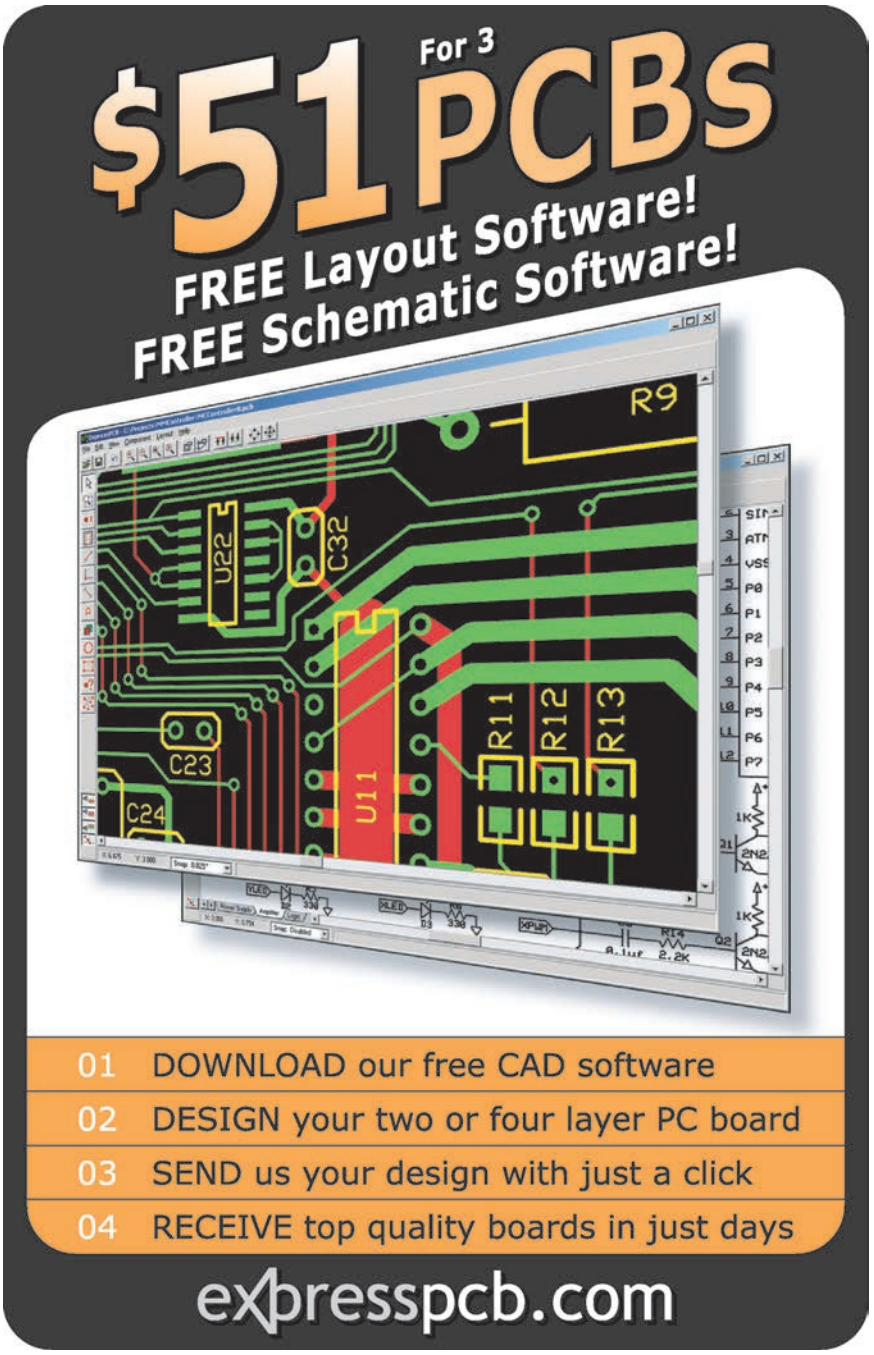
SOURCES

Fiwiz Digital filter design software
International Computer Science Institute (ICSI) | www1.icsi.berkeley.edu/~storn/fiwiz.html

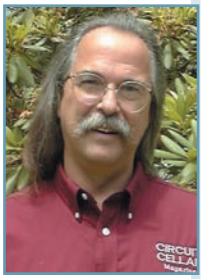
ScopeIIR: IIR Filter design software for Windows
Iowegian International Corp. | www.iowegian.com

Proteus PCB design software
Labcenter Electronics, Ltd. | www.labcenter.com

MATLAB computing language
MathWorks | www.mathworks.com



by Jeff Bachiochi (USA)



Mechanical Gyroscope Replacement (Part 1)

A Microelectromechanical Systems (MEMS) Solution

A gyro uses internal circuitry to drive its masses into resonance, sample capacitive-coupled displacement sensors, and filter out unwanted artifacts. This article describes how a MEMS solution provides stability to a mechanical gyro. One sensor helps make up for the other's weaknesses when using a gyro, an accelerometer, and a magnetometer together.

In 2009, I purchased a quadcopter named XUFO. From the start, I had problems with batteries overheating while recharging and instabilities in the mechanical gyro, which is supposed to maintain a stable platform, even in breezy conditions. I determined the platform became unstable in a tethered state if its tilt exceeded 30°. When this happened, the spinning rotor hopped around like a top that had come in contact with some object. Photo 1 shows the gyro's makeup with its cover removed. Gimbals, which provide roll-and-pitch output, are used to suspend the tiny motor and rotor.

Hall-effect devices are used to measure the gimbals' positions, referenced to the gyro's frame. One Hall device's output measures any roll-axis deviation. A second Hall device's output measures any pitch-axis deviation.

Roll is the rotation around a nose-to-tail centerline, wingtip up or down. Pitch is a rotation around the wingtip-to-wingtip centerline, nose up or down. The output is a measurement of the deviation in degrees (±) from zero, normally parallel to the earth's surface. The remaining dimension, yaw, is the rotation about a third of an axis perpendicular to the roll and pitch, also known as the gravity vector. Yaw is like a compass bearing, or direction, also in degrees.

The quadcopter's roll and pitch control is an adjustment to the RPMs (speed) of each rotor, based on gyro output. It's easy to imagine how, if all the rotors were running at equal output, the quadcopter could rise from the ground, assuming the rotors could produce enough lift. But where is the front? There are actually two basic configurations (see Photo 2). The first uses forward rotors

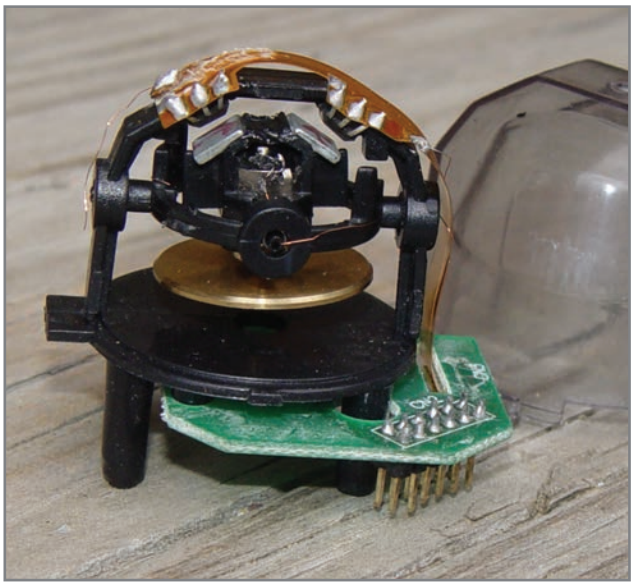


Photo 1—My XUFO quadcopter uses a mechanical gyro that is an inexpensive yet elegant attempt to counteract wind gusts. With its protective shield removed, you can see the motorized spinning rotor that sustains equilibrium as its frame moves. It has an unfortunate erratic behavior (resulting in major instability) when subjected to high accelerations or a tilt greater than 30°.

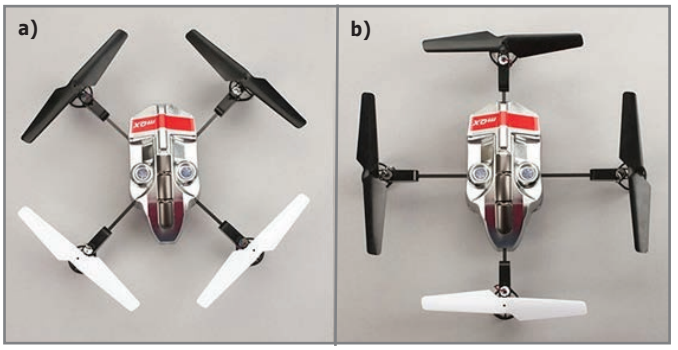


Photo 2—A quadcopter can use two types of orientations. **a—**Rotors are defined in adjacent front/back and left/right pairs in a square formation. **b—**Rotors are defined in opposite front/back and left/right pairs in a diamond formation. (Photo courtesy of Red Rocket Hobbies)

and aft rotors. In this mode, the relationship between forward and aft rotor speeds provides pitch, while the relationship between the left and right rotors controls the roll. With this implementation, each rotor will affect two dimensions. If the forward left rotor's speed is altered, the roll and pitch are both affected.

The second configuration uses a nose rotor and opposing tail rotor and a leftwing rotor and opposing rightwing rotor. With this orientation, opposing nose and tail rotors affect the pitch while opposing leftwing and rightwing rotors affect the roll. My quadcopter uses this second configuration. Blinking LEDs on each of the four rotors help the operator. The nose rotor is identified by a red LED, the others are blue LEDs.

STATIC MEASUREMENTS

To examine the quadcopter, I tethered it down and put an oscilloscope on the gyro outputs. Under level conditions, the Hall-effect devices output 0.5 V_{CC}. When tilted, the corresponding output rose or lowered about 1 V with a maximum 30° tilt. I found an online forum for XUFO that included a schematic of the mechanical gyro. Something about the schematic didn't look right. It showed some level shifters and an EEPROM, which were not on my gyro. Since the schematic was dated 2006, I thought this might be an earlier version. While the actual gyro connections matched my gyro, the schematic's circuitry might actually be on the main board.

After reading some messages on an online forum, I found an indication of copycat copters that had flooded the market to take unauthorized advantage of the original's popularity. The key points explained how tell the forgeries. I had purchased a rip-off (see Photo 3). I was on my own to find a solution.

If a breeze caused the craft to roll or pitch, the mechanical gyro's output voltage would change. The change would look like an error to the system. The system should counteract this error by performing a roll or pitch maneuver to oppose the error, attempting to level out the craft. The system will only cease adjustments if the error is reduced to zero (level). So this mechanical gyro is a relative position sensor. I want to replace this unacceptable mechanical gyro with some solid-state sensor.

MEMS GYRO

Many of the microelectromechanical systems (MEMS) gyros available today use a tuning-fork style arrangement, where opposing masses are driven into resonance. Rotation about an axis perpendicular to the resonance axis will create a displacement from the plane of resonance. The opposing masses will deviate in opposite directions (due to the Coriolis effect). The deviation result is measured as the rate of rotation. Thus the output is not position, but a change in position. If changes are integrated in position over time, relative position can be calculated.

What should you look for when choosing a gyro? Looking at gyro specifications may make this question easier to answer. First, start with the power requirements. Many MEMS devices use a lower operating voltage than the 5 V we are accustomed to seeing. There are level-translation issues that must be overcome when interfacing to circuitry that uses a different operating voltage. To simplify this problem, you might consider lowering your system operating voltage so all devices are using the same voltage. Operating current is the next factor. For designs where a low-power (i.e, sleep) mode is possible, many gyros have one or more reduced function modes to achieve a lower (i.e., from mA to µA) current draw.

A gyro uses internal circuitry to drive its masses into resonance, sample the capacitive-coupled displacement sensors, and filter out unwanted artifacts. The operational specifications are derived from this result. The gyro's fullscale range is the number of degrees per second the maximum output represents. An analog gyro might have a maximum output swing of ±1 V about some reference. A digital gyro might use a two's-complement output of x bits (e.g., signed 15 bit). In either case, the output would represent a rotation in either direction (e.g., ±360°/s). At rest, the gyro should ideally output a zero-rate level (ZRL) of 0°/s, but there will be some offset (e.g., ±10°/s) that may be further affected by device soldering. Since this is



Photo 3—My XUFO box is decorated with loosely translated English (and often humorous) descriptions.

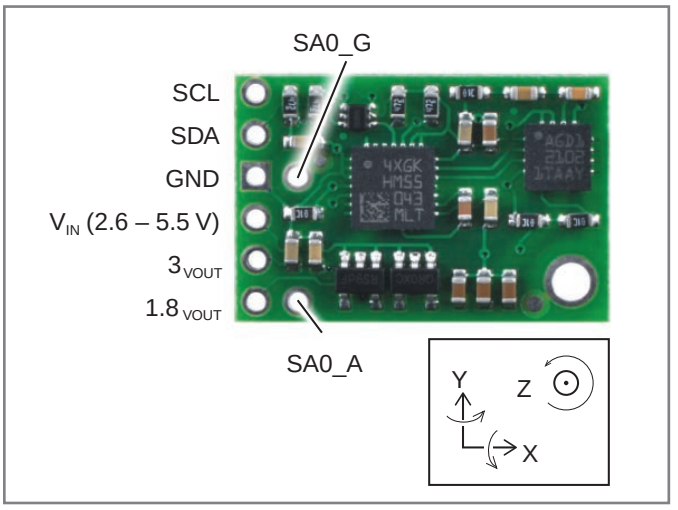


Photo 4—The Pololu MinIMU-9 module's sensors (L3G4200D and LSM303DLM) are oriented with their x, y, and z axes aligned with one another. This makes it easier to coordinate all three sensors' specific axis rotation, gravity, acceleration, and magnetic flux measurements.

critical (as will be apparent later) there is also a spec on the temperature's effect on the zero-rate level (ZRL).

The sensitivity is the least amount of change that can be represented. For an analog gyro, this would be V/1°/s (e.g., FullScale V/FullScale°/s = 1/360 = 2.7 mV/dps). The digital gyro's sensitivity is provided as °/s/1 bit (e.g., FSdps/FSbits = 360/32767 = 0.0109°/s/bit).

Finally, the gyro has a system bandwidth provided in hertz. This is the maximum rate at which you can ask for samples (e.g., 100 Hz = 1 sample every 1/100 = 10 ms). Based on the sample rate, there is also a noise-rate density factor given in dps/sqrt (Hz) that indicates the kind of long-term error expected of the device over time. While the device's random deviation equals zero, cumulatively this could become a significant unknown. Hence, MEMS gyros often have "long-term inaccuracy."

Previously, I mentioned the importance of ZRL. If this offset is left unchecked, cumulative offset errors would quickly create a "long-term inaccuracy" that surpasses the rate-noise density (RND). A solution to both might be to toss out any relative positioning if you could determine when the device was motionless. The ZRL offset could be minimized by performing some calibration routine. However, the gyro's calculated position will still become less certain over time (albeit much more slowly). For systems with a GPS, the gyro is used as a secondary position sensor during those times when the signal is lost. Without GPS, this is an ongoing problem. The MEMS gyro is best suited for those times when it is not sitting still.

> "A gyro uses internal circuitry to drive its masses into resonance, sample the capacitive-coupled displacement sensors, and filter out unwanted artifacts. The operational specifications are derived from this result."

MEMS ACCELEROMETER

Possibly the first MEMS device developed was described in a patent filed in the late 1960s as an electromechanical resonator, which used a cantilevered substrate. The MEMS sensor has drastically reduced the cost of sensors since modified semiconductor device fabrication technologies can be used to produce them. Today's most widely used MEMS devices include the pressure sensor and the accelerometer. The accelerometer has a mechanical design similar to the gyro previously described. In this instance, an internal mass is not driven, yet its movement perpendicular to its axis of support is measured much like the gyro. The accelerometer's suspended mass is affected by any change in velocity (acceleration). Gravity has an acceleration constant of 32 ft/s/s, considered 1 G. An accelerometer will measure gravity's effect. It doesn't know the difference between gravity and any other 1-G acceleration. This has both good and bad consequences.

Since accelerometers are manufactured to cover a large spectrum of accelerations, select the best-fit device based on your particular application's specifications. Like the previously described gyro, accelerometers operate at a lower voltage (typically 2.5 V and 3.3 V). While current requirements are often lower for accelerometers, microamp-range power-down modes can be similar.

The typical fullscale range is a few G, however accelerometers used for air bag deployment and sports equipment can measure hundreds of G. Many manufacturers provide a choice of ranges within a device. Be aware that the specifications will change with range selection. Accelerometers are available with either analog or digital outputs. An accelerometer's sensitivity is the outputs change per unit of acceleration. For ±3-G analog outputs devices, typical sensitivity is 300 mV/G. For a digital device, this value is a function of the device's resolution. Even though a 16-bit register is used, not all bits may be significant. A 12-bit resolution (signed 11 bit) is common. This would be ±2,048 bits/FS or 683 bits/G. Sensitivity due to temperature is typically less than 0.10%.

While the accelerometer's 0-G level may have offset issues due to assembly and physical orientation, it does not have the same error issues associated with a gyro. A calibration routine can be used to easily cancel these bias levels.

An accelerometer's system bandwidth or frequency response is determined by each axis's physical makeup and internal circuitry. In a triple-axis device, two axes are parallel to the fabrication layer while the third axis is perpendicular. It is more difficult to create this third axis, and often it will have different specifications than those parallel to the substrate. Be aware that given specifications may be axis specific. Noise performance is the amount of noise produced by the innards that can affect the sensor's output. This value is typically low, less than 1 mG/sqrt (Hz).

An accelerometer at rest can indicate orientation with respect to the earth. Once it accelerates, you may see 1 G on multiple axes. In this situation, position cannot be determined, as there are multiple outputs suggesting a gravity vector. The accelerometer can be most effective while no axis is changing (i.e., a constant velocity, possibly at rest).

CIRCUIT CELLAR

Microprocessor Design Using Verilog HDL

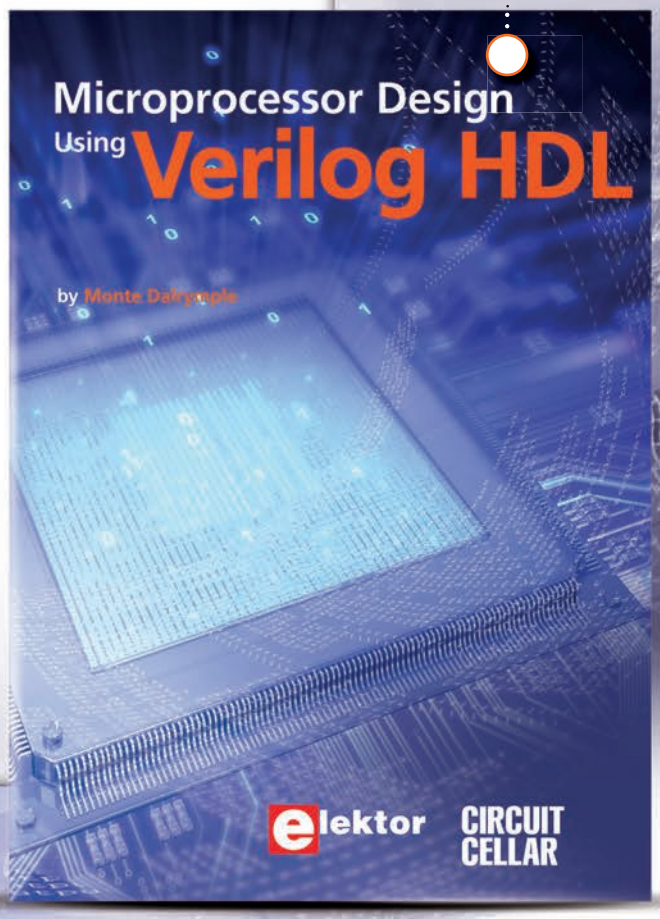
With the right tools, such as this **new book, designing a microprocessor can be easy.** Okay, maybe not easy, but certainly less complicated. Monte Dalrymple has taken his years of experience designing embedded architecture and microprocessors and compiled his knowledge into one **comprehensive guide to processor design in the real world.**

Yours for just **$45.00**

Monte demonstrates how Verilog hardware description language (HDL) enables you to **depict, simulate,** and **synthesize an electronic design** so you can **reduce your workload and increase productivity.**

Microprocessor Design Using Verilog HDL will provide you with information about:

- Verilog HDL Review
- Verilog Coding Style
- Design Work
- Microarchitecture
- Writing in Verilog
- Debugging, Verification, and Testing
- Post Simulation and more!



www.cc-webshop.com

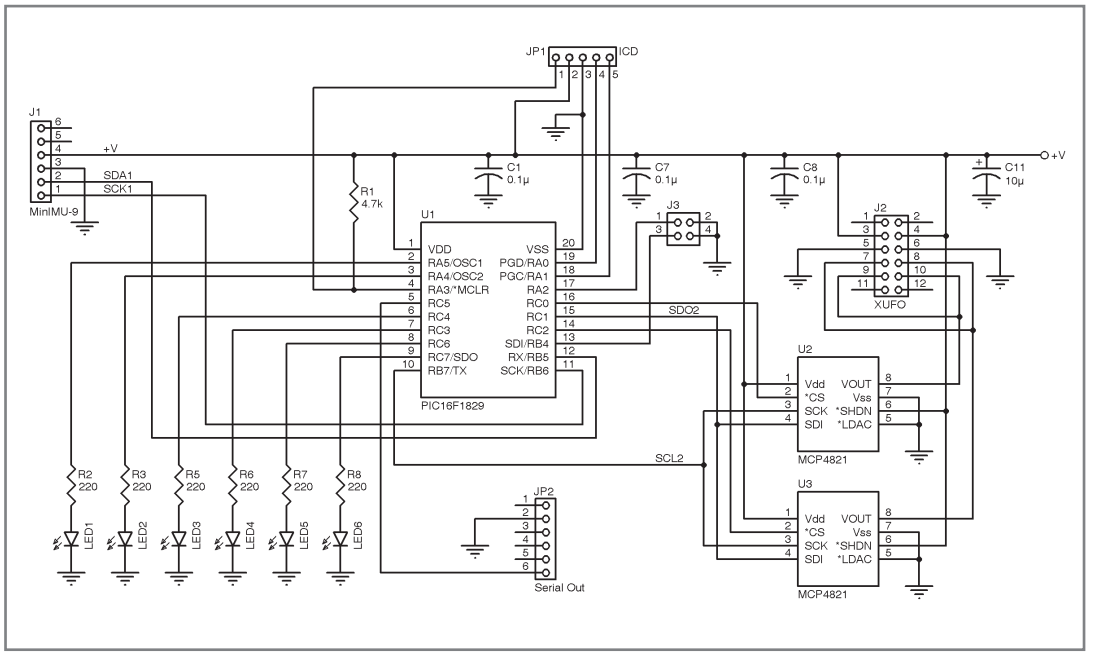


Figure 1—The prototype schematic uses an array of MEMS sensors and provides voltage output signals that mimic the mechanical gyro interface.

FYI

A third sensor that is popular these days is the magnetometer, which measures the strength of a magnetic field. While this might come from a manmade source (e.g., a coil or motor), in terms of navigation, it is the earth's magnetic field that is of interest. One widely used unit of magnetic strength is the gauss (g). The strength of the earth's magnetic field is in the range of 20 to 80 mg. Natural and manmade magnets can be many times stronger than this, so keeping them at a distance (or at least being aware of them) is important. Even metal and ferrous materials can affect the magnetic lines of flux used to align a magnetometer with the north/south orientation of the earth's field. As long as these items are fixed in relation to the magnetometer, they can be compensated for if the device is calibrated.

Voltage and current specifications are similar to the gyro and accelerometer. A magnetometer with a fullscale range of ±2 gauss using a signed 11-bit magnitude would have a sensitivity of 2 g/2,048 bits or a resolution of 1 mg/bit. Since the perpendicular axis is constructed a bit differently than the two parallel axes, it will often have slightly different characteristics.

In my article, "Location Notification: A Look at Anisotropic Magnetoresistance Sensors," (*Circuit Cellar* 227, 2009), I used a Honeywell HMC5843 digital compass IC (and an Analog Devices ADXL303 accelerometer) to create a digital compass. In that article, I discussed how the magnetic lines of force travel parallel to the earth's surface at the equator, but gradually end up going directly into (out of) the earth (at the magnetic poles). The compass can use its multi-axis sensors to measure the direction of these lines of force. To achieve any accuracy, the compass must be held parallel to the earth for both calibration and any following measurements. In this project, the accelerometer was used to measure gravity and indicate when the compass was level. A level compass (user feedback from the accelerometer) simplifies the calculations to determine direction. An unlevel compass would require calculating compensation for the degree of tilt error.

AXIS ALIGNMENT

When used together, a gyro, an accelerometer, and a magnetometer become a team in which one sensor helps make up for the others' weaknesses. Theoretically, this can happen as long as they are in any fixed position (relative to one another). But it simplifies calculations (and understanding) if they are all positioned so their axes are aligned with one another.

Note that, even though I have not mentioned it specifically, my discussion so far assumes devices that can measure in all three axes. Any of these devices are available with less than three axes of sensitivity. Not all applications require three axes. However, when multiple devices are used together, it is important that they are positioned on your PCB (or in the system using multiple PCBs) with each device's corresponding axis aligned. Once aligned, their x-axis outputs (for example) should represent a polarity and magnitude that is relative to that axis's sensitivity, in this case, rotation around the axis or acceleration and magnetic flux through it.

I used Pololu's MinIMU-9 inertial measurement unit for prototyping (see Photo 4). Using the MinIMU-9 might be overkill for my specific task, but the simple 5-V interface and not having to solder some fairly tiny SMT parts makes it a great fit for initial experimentation.

PCB REPLACEMENT

My XUFO's gyro module is a piggyback PCB that mates the mechanical gyro to the processor PCB. The PCB has no other active components. My replacement PCB must provide a mating for the two row by five column, 1-mm vertical interfacing connector along with the appropriate mounting hole spacings. I started by measuring the existing processor PCB and creating a prototyping PCB of the same size with matching mounting holes. These holes provide maximum flexibility to the

Elektor Print

Classical reading: on paper



+

Elektor Digital

Contemporary & innovative reading: on PC, notebook or tablet

=

Elektor PLUS

Comprehensive reading: at home and on the road

Read Elektor with the premium PLUS subscription!

Subscribe now or upgrade: www.elektor.com/usa

mounting possibilities. The prototyping board's edge has an area free of pads, which provided an interference-free place to mount the smaller 1-mm connector independent of the grid of 0.1" holes (see Photo 1).

The MinIMU-9 needed to be mounted at a 45° angle to the PCB for its axis to be aligned with those of the mechanical gyro I was attempting to replace. I used a Dremel tool to hand drill the holes for this 1 × 5, 0.1" connector (the MinIMU-9) and the two row by five column, 1-mm interface connector.

The rest of the circuitry consists of a processor, which uses an I^2C interface to interrogate all the sensors on the MinIMU-9 module. This processor also communicates with two DACs and uses a SPI to provide voltage outputs to the interface connector. Finally, a bunch of LEDs and UART output to provide some indication of operation (and also help with the code debugging). The parts in the schematic in Figure 1 were sprinkled around a PCB to avoid all the mounting holes.

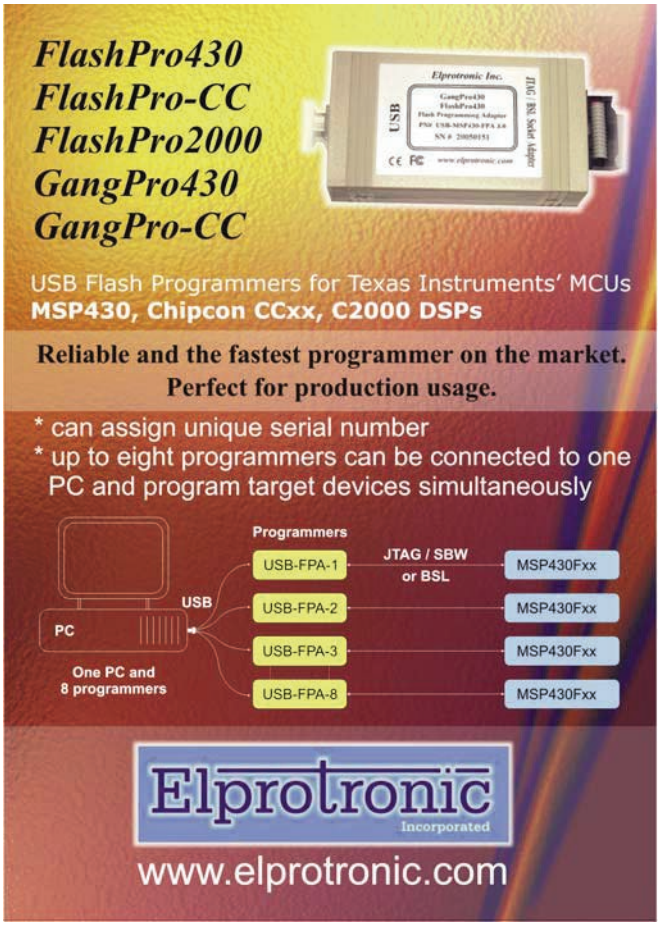
My microcontroller choice was based on having two synchronous serial ports. While it's theoretically possible to use the same port and redefine it each time you want to change communication mode (e.g., SPI versus I^2C), having two ports eliminates this overhead and enables communication to simultaneously exist over both. The MinIMU-9 uses the two-wire I^2C to communicate with all sensors (e.g., a clock and a data line). The Polulu LSM303DLM compass (magnetometer and accelerometer) uses a 7-bit device address of 0x18 for the accelerometer, as A0 is pulled down on the MinIMU-9. The magnetometer has a 7-bit address of 0x1E and uses registers 0x00–0x3A, while the accelerometer uses registers 0x00–0x3F. The L3G4200D uses a 7-bit device address of 0x69, as its A0 is pulled up on the MinIMU-9. Registers 0x00–0x38 provide access to the (angular rate) gyroscope.

The DACs are SPI devices that require individual chip selects in addition to the SPI bus's clock and data (output) lines. Although defined by SPI, data input is not used here as no data is available from the DAC. This 8-pin DAC (a Microchip Technology MCP4*xxx* digital potentiometer) is very flexible. It is available with an internal 2.048-V internal reference or an external reference (which can be connected to V_{CC}). It is also available in 8-, 10-, or 12-bit resolutions. All pin-compatible devices come packaged in both through hole and SMT, which makes size and weight reductions a simple feat.

STAY TUNED

Next time, I'll examine the registers and engage the sensors to try and figure out how best to succeed in replacing the mechanical gyro. This, although rather limited in its useable range, is still a mechanical marvel to behold. I'm hoping integrating the MEMS gyro, accelerometer, and magnetometer into this pin-compatible interface will improve my quadcopter's stability if it's tilt exceeds 30° by a wind gust.

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.*



RESOURCES

J. Bachiochi, "Location Notification: A Look at Anisotropic Magnetoresistance Sensors," *Circuit Cellar* 227, 2009.

J. Esfandyari, R. De Nuccio, and G. Xu, *Solid State Technology*, "Introduction to MEMS Gyroscopes," 2012, www.electroiq.com/articles/stm/2010/11/introduction-to-mems-gyroscopes.html.

M. Krass, Scribd Inc., "Gyro Programming for FRC Robots," 2006, www.scribd.com/doc/39013314/GyroProgrammingforFRCRobots.

W. Stockwell, "Random Angle Walk," Crossbow Technology, Inc., www.xbow.com/pdf/AngleRandomWalkAppNote.pdf.

SOURCES

ADXL330 Accelerometer
Analog Devices, Inc. | www.analog.com

HMC5843 Compass IC
Honeywell | www.honeywell.com

MCP4*xxx* Digital potentiometer
Microchip Technology, Inc. | www.microchip.com

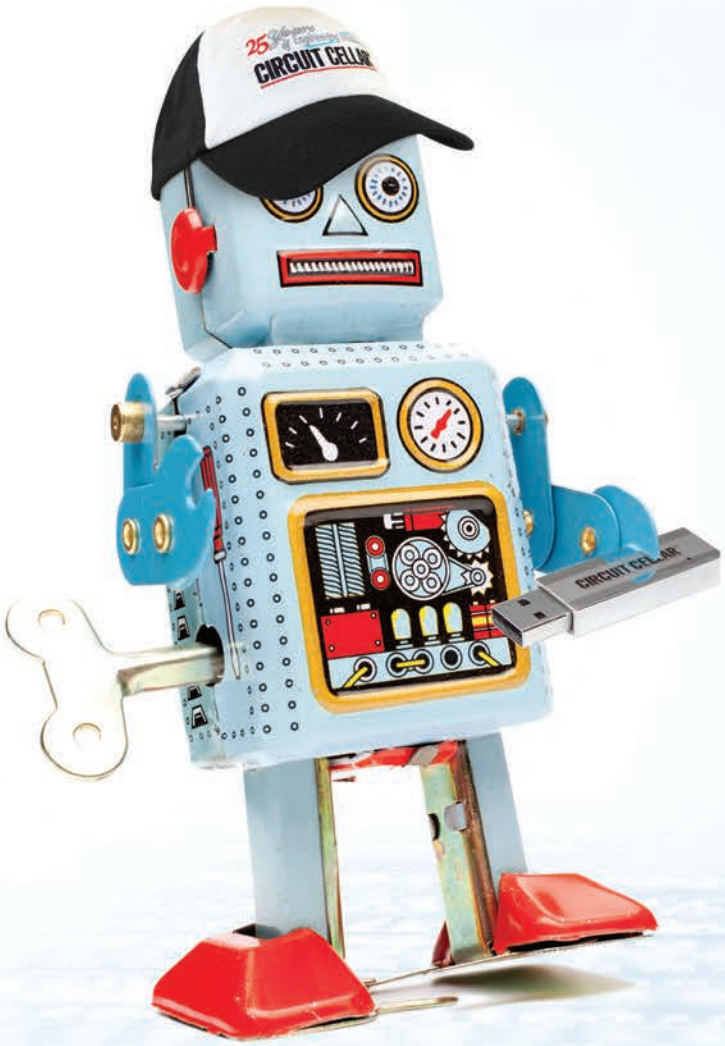
MinIMU-9 Inertial measurement unit, LSM303DLM compass, and L3G4200D carrier/breakout board
Pololu Corp. | www.pololu.com

Bringing You 25 Years

of Engineering Excellence and Cutting-Edge Technology

The Entire *Circuit Cellar* Magazine Archive on a Limited-Edition 25th Anniversary USB drive!

CIRCUIT CELLAR 25 YEARS OF EMBEDDED INSIGHT



Here's what's included:

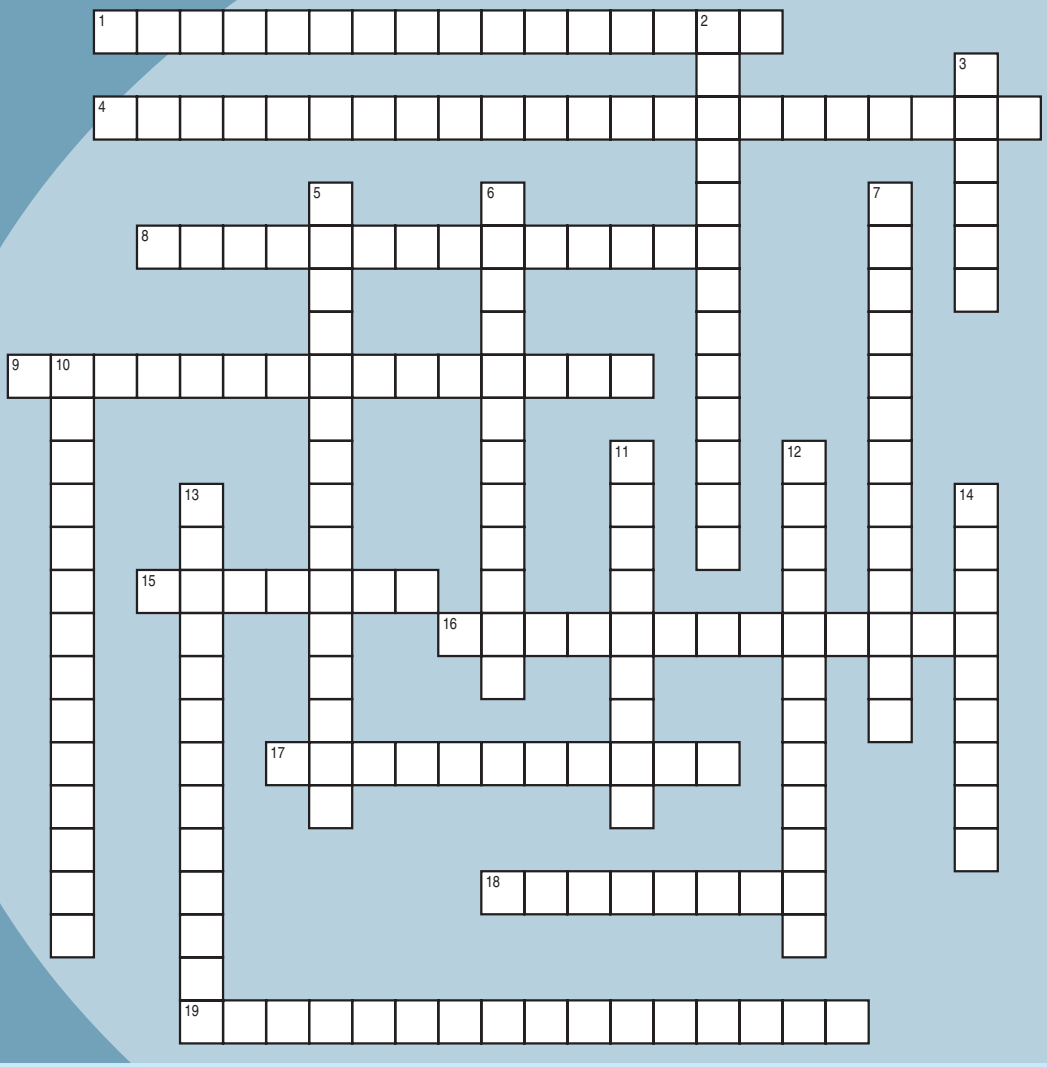
- Special anniversary USB drive or our traditional CC GOLD USB drive. ***Your choice!***
- PDFs of all *Circuit Cellar* magazine issues in print through date of purchase
- Article code
- Design Challenge projects*
- Two years of *Elektor* magazine issues in PDF format (2010–2011)
- Two years of *audioXpress* magazine issues in PDF format (2010–2011)
- A USB memory upgrade from 16 GB to a whopping 32 GB!
- Free gift! *Circuit Cellar* 25th anniversary hat

For a limited time, receive the blue tin robot featured here!

Order today at www.cc-webshop.com

* Design Challenge add-ons:
ATMEL AVR Design Contest 2006, WIZnet iEthernet Design Contest 2007, Microchip Technology 16-Bit Embedded Control Contest 2007, Texas Instruments DesignStellaris Contest 2010, WIZnet iMCU Design Contest 2010

CROSSWORD



Down

2. The design in Mark Pedley's article, "eCompass: Build and Calibrate a Tilt-Compensating Electronic Compass" (*Circuit Cellar* 265, 2012), was built using one of these
3. New Zealand-based *Circuit Cellar* contributor and recent interviewee who is fascinated with advanced robot technologies
5. This control system generates an output frequency, which can be either higher or lower than the input, based on a reference input clock [three words]
6. This month's interviewee, Helen Li (p. 54), believes this type of computing will solve the contradiction between the limited functions of computing systems and the ever-increasing variety of applications
7. This type of cell consists of a thin plastic film sandwiched between two metal stators
10. Its output voltage is greater than its input voltage [two words]
11. American engineer (1890–1954) who invented the regenerative circuit, the super-regenerative circuit, the superheterodyne receiver, and modern frequency modulation (FM) radio transmission
12. Used to measure tilt
13. American engineer (1860–1898) who invented an induction meter to measure alternating current
14. The functional equivalent of a synapse

Across

1. Can be produced using two sensors spaced at odd half-slot multiples around a single track [two words]
4. This type of system's size ranges from 20 µm to 1 mm
8. A low-to-high transition means "0" and a high-to-low transition means "1" [two words]
9. Because these devices have only one track per bit of resolution, they can require large diameters, which give them a nonvolatile and unique output for each position [two words]
15. Italian physicist Alessandro Volta (1745–1827) is credited with inventing the first one of these in the 1800s
16. A piece of software, firmware, or logic circuit that takes a digital data flow as an input and provides a filtered version of this signal on its output [two words]
17. Topic of columnist Bob Japenga's ongoing article series, which began in *Circuit Cellar* 263, 2012
18. 1,152,921,504,606,846,976 bytes
19. Amount of water vapor in the atmosphere expressed as a percentage of the total amount the air can hold at the current temperature [two words]

The answers are posted at www.circuitcellar.com/crossword and will be available in the next issue.

IDEA BOX

THE DIRECTORY OF PRODUCTS AND SERVICES

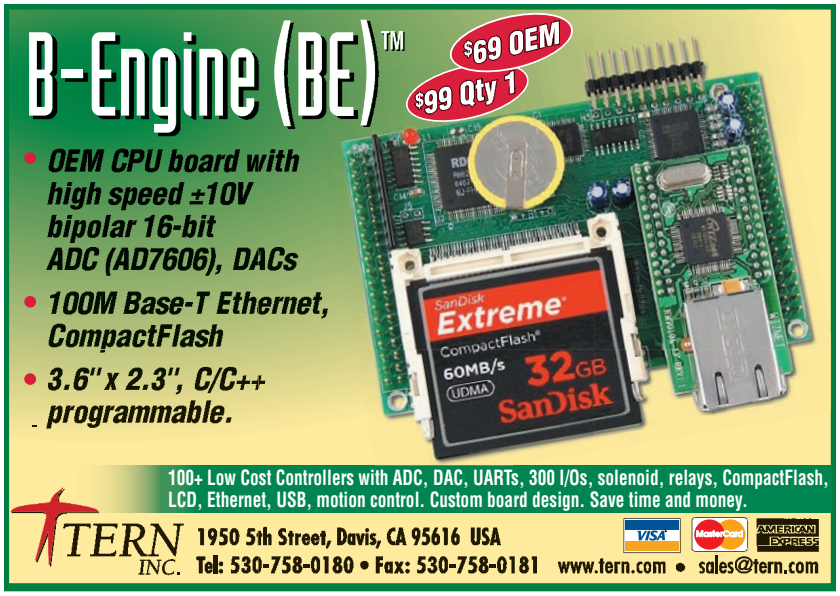
AD FORMAT: Advertisers must furnish digital files that meet our specifications (www.circuitcellar.com/advertise). ALL TEXT AND OTHER ELEMENTS MUST FIT WITHIN A 2" x 3" FORMAT. E-mail adcopy@circuitcellar.com with your file or send it to IDEA BOX, Circuit Cellar, 4 Park Street, Vernon, CT 06066. For current rates, deadlines, and more information contact Peter Wostrel at 978.281.7708 or peter@smmarketing.us.

The Vendor Directory at www.circuitcellar.com/vendor is your guide to a variety of engineering products and services.











BPS BusBoard Prototype Systems
Prototyping PC Boards
Many Patterns
BusBoard
StripBoard
PadBoard
ProtoBoard
ProtoBoard-2H
PowerBoard
SMTBoard
SMTpads
PC BreadBoards
See our site
www.BusBoard.us
Available From
JAMECO ELECTRONICS
MOUSER ELECTRONICS
a tti company
amazon.com
amazon.co.uk

For Any PCB - GHz Bandwidth BGA/QFN Sockets
Quick-Turn Custom Sockets
• Use on any existing PCBs
• No mounting holes
• No soldering
• Patented placement/ epoxy system
• 0.32mm larger than IC (per side)
• Bandwidth to 40 GHz
• 0.3mm pitch & up
RoHS
Ironwood ELECTRONICS
1-800-404-0204
www.ironwoodelectronics.com

NOW AVAILABLE!
CC Electronic Toolbox App
Databases, schematics, calculators, & more!
Circuit Cellar
Tech Writing
Resistors
Capacitors
Transistors
FETs
Triacs
Thyristors
Diodes
ICs
Resistor
Capacitor
For iPad, iPhone, & iPod Touch
Available on the App Store

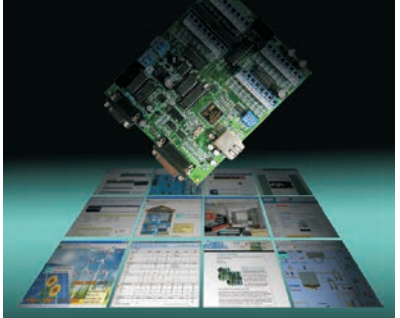
PDQ Board™ - A Fast I/O-Rich Single Board Computer
$159/100s
• Low cost 2.5"x4" C-programmable computer
• 16-bit HCS12 processor clocked at 40 MHz
• 8 PWM, 8 counter/timer, and 8 digital I/O
• 16 10-bit A/D inputs
• Dual RS232/485 ports, SPI and I²C ports
• 512K on-chip Flash, 512K RAM with Flash backup
• Plug-in I/O expansion, including Ethernet, Wi-Fi, GPS, 24-bit data acquisition, UART, USB, Compact Flash card, relays, and more ...
Mosaic Industries Inc.
tel: 510-790-1255 fax: 510-790-0925
www.mosaic-industries.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

microEngineering Labs, Inc.
www.melabs.com 888-316-1753
Programmers for Microchip PIC® Microcontrollers
melabs U2 Programmer
microEngineering Labs, Inc.
www.melabs.com
Starting at $79.95
PC-Tethered USB Model (shown):
•Standalone software
•Command-line operation
•Hide GUI for automated use
•Override configuration with drop-downs
Stand-Alone Field Programmer:
•Power from target device or adapter
•Program file stored on SD-CARD
•Programming options stored in file
•Single-button operation
Program in-circuit or use adapters for unmounted chips.
Zero-Insertion-Force Adapters available for DIP, SOIC, SSOP, TQFP, and more.
PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.

LISTEN
TO YOUR MACHINES

Ethernet PLCs for OEMs



FMD88-10
and FMD1616-10

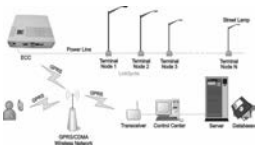
Integrated Features :
- ETHERNET / Modbus TCP/IP
- 16 or 32 digital I/Os
- 10 analog I/Os
- RS232 and RS485
- LCD Display Port
- I/O Expansion Port
- Ladder + BASIC Programming

$229 and $295
before OEM Qty Discount

tel : 1 877 TRI-PLCS
web : www.tri-plc.com/cci.htm

TRIANGLE RESEARCH INTERNATIONAL

Street Lighting System, Power-Line Communication, Long Range RFID Reader, Arduino



- Power-Line Communication Street Lighting System
- WiFi Street Lighting System
- Power-Line Communication Modules: Spyder and McLaren
- Arduino with WiFi: Diamondback, Redback
- Arduino Shields: Cupperhead and Juniper WiFi, PLC, GPRS, MIDI, Touch, gameduino
- Long Range UHF Gen2 RFID
- JPEG Serial Cameras
- 4TV
- Robotics

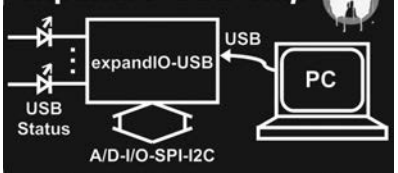




www.linkspritedirect.com



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

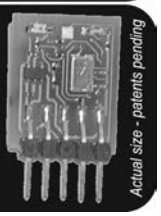
www.hexwax.com - Buy from Mouser & Farnell

Amazing PIC programmer

Most devices supported ICSP, SQTP, & copy limits

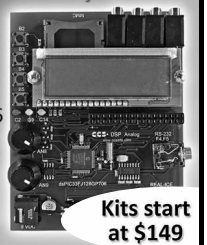
$32 at Digikey & Mouser

www.flexipanel.com



DSP Analog Development Kit for Microchip dsPIC® with CCS C Compiler

- DSP audio conditioning prototype board features dsPIC33FJ128GP706



- Contains optimizing C compiler with examples for FIR and FFT filters

Kits start at $149

- Targets dsPIC® DSP accumulator registers from C source code

DSP Analog Development Kit with Programmer & Tutorial

Use discount code DSPAK when ordering

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

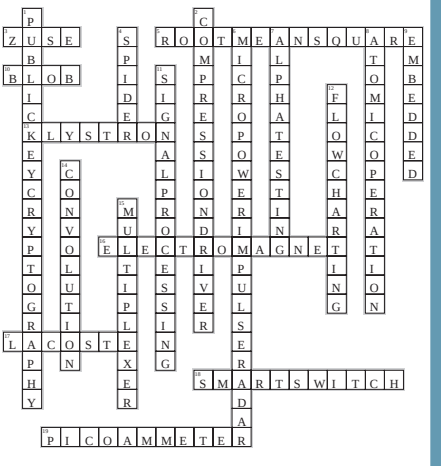
CROSSWORD ANSWERS from Issue 266

Across

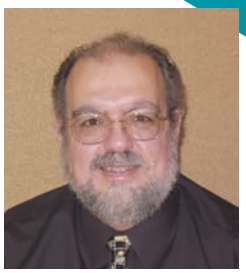
3. ZUSE—German engineer and inventor (1910–1995) who is credited with creating the Z3, a program-controlled Turing-complete computer
5. ROOTMEANSQUARE—Alternating voltage/current with the exact same energy content as the same value of direct current; a.k.a., quadric mean [three words]
10. BLOB—Stores binary data; synonym: drop
13. KLYSTRON—A specialized linear-beam vacuum tube
16. ELECTROMAGNET—English physicist and inventor William Sturgeon (1783–1850) is credited with using electric current to develop the first one of these objects in 1825
17. LACOSTE—*Circuit Cellar* columnist who frequently writes about frequency
18. SMARTSWITCH—An energy-saving device that was the topic of Fergus Dixon's article (*Circuit Cellar*, 263 2012) [two words]
19. PICOAMMETER—Measures low current

Down

1. PUBLICKEYCRYPTOGRAPHY—Decodes using two pieces of information, one public and one private [three words]
2. COMPRESSIONDRIVER—A loudspeaker that uses a consolidating technique to achieve high efficiencies [two words]
4. SPIDER—The flexible collar that helps keep a voice coil magnetically centered
6. MICROPOWERIMPULSERADAR—A pocket-sized radar that runs off AA batteries and is often used as a basic motion sensor for security applications [three words]
7. ALPHATESTING—Check performed by an independent team on a system installed at a place other than the targeted customer's site [two words]
8. ATOMICOPERATION—An action that is non-interruptible by any other one and never presents partial results to an outside observer [two words]
9. EMBEDDED—As *Circuit Cellar* prepares to celebrate its 25th anniversary, a past, present, and future theme of the magazine centers on this type of technology
11. SIGNALPROCESSING—Involves measuring physical quantities with time and spatial variances [two words]
12. FLOWCHARTING—Jeff Bachiochi describes how to use this technique to write code in issue 266
14. CONVOLUTION—Mark Csele's article, "DSP-Based Color Organ" (*Circuit Cellar*, 249 2012), used this technique to create high-performance filters
15. MULTIPLEXER—A device that combines input signals, shares a single transmission channel, and enables data compression



PRIORITY INTERRUPT



by Steve Ciarcia, Founder and Editorial Director

Go Team!

Several weeks ago, I was following the successful landing of the Curiosity rover on Mars, which got me reminiscing about the importance of teamwork on large engineering projects. Obviously, a large project requires a significant number of people due to the sheer amount of work. But, more importantly, a project's various tasks require a balanced mix of skills for successful and timely completion.

Naturally, you want engineers working in areas where they have the skills and confidence to succeed. That's when they'll do their best work. At a basic level, all engineers share a distinctive trait: the ability to make something you want from technology and materials. This is the best definition of "engineer" I have heard. But, as I said last month, different engineers have different interests, skills, and experience. Some engineers are good at understanding the subtleties of how a large systems' components interact, while others are good at low-level details (e.g., analog circuit design, mechanical design, or software programming). Diversity of skills among team members is important and can greatly strengthen a team.

At some point, we all look to ascend the corporate ladder and, for most companies, that involves engineers taking on management responsibilities. Actively encouraging engineers to work in areas outside their comfort zones encourages greater diversity in problem-solving approaches. Further, inspiring engineers to seek responsibility and expand their comfort zones can make them better engineers for the long term. While this is mainly true about engineers who are employees, it also applies to any contractor or consultant involved in a long-term company relationship.

Some engineers can jump into just about any area and do well. However, it is rarely in the interest of the project or good team dynamics to follow that impulse. Those engineers need to enable the specialists to do the work they're best at and only jump into situations where they can do the most good. In other words, when team management is your primary task, engineer or not, you need to take on a mentoring role, often teaching rather than doing.

Communication among team members is also key. There must be enough of it, but not too much. I have seen teams schedule so many meetings there isn't any time left for individuals to make progress on their assigned tasks. Meetings need to be short, to the point, and involve only those people who have a vested interest in the information being exchanged. This can range from two engineers conversing in a hallway to a large project-wide meeting that keeps everyone in sync on a project's overall goals and status. But beware, it can be difficult to keep big meetings from getting diverted into the minutiae of a particular problem. This needs to be avoided at all costs.

Even when the schedule is tight, overstaffing usually has a negative benefit. The math suggests that project completion time should go down by the inverse of the number of team members. However, this ignores the overhead of more communication among team members, which goes up by the square of the number of participants. If there are too many team members, they may start getting in each other's way and have less sense of ownership in what they're doing. Basically, there are some tasks that take a fixed amount of time. As the saying goes, nine women can't make a baby in one month!

Motivating the team is another key factor that should be a priority shared by the team and technical management. No matter how large or small a team member's assigned tasks, if he feels he has the responsibility and the recognition for getting that task done, he'll be more engaged and motivated to do well. Moving team members from task to task destroys any sense of ownership. Granted, every project has the occasional fire that needs to be extinguished—all hands on deck—but, if a project is constantly in that state, then it's pretty much doomed to fail.

Regardless of whether you are engineering a Mars rover at NASA or creating the next great social media "widget" at a venture-capital funded start-up, the dynamics of successful project management have an established methodology. Design engineers are creative and it is important to give them the flexibility to unleash their creativity—but keep it within bounds. Most projects are time and cost sensitive. Ratifying your step up the corporate ladder only comes by ensuring the project is completed within budget and in a timely fashion.

steve.ciarcia@circuitcellar.com



www.ftdichip.com

FTDI Chip

ENHANCED USB PERFORMANCE

Streamlined USB Bridge Solutions

X-CHIP

EXtensive Interfaces
UART, FIFO, SPI, I²C, FT1248

EXtended Features
Battery charger detection
Low active power (8 mA, typical)
Internal MTP memory
Expandable clocking; clock generation and system clock out

EXceptional Drivers
Windows, MacOS, Android, and Linux




PC Boards, Components, Assembly
because no one should ever have to wait again...
... imagineering has Revolutionized the Online Turnkey Quoting Process
Join the Quoting Revolution Happening Now @
PCBnet.com