CIRCUIT CELLAR

THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

APRIL 2011
ISSUE 249

EMBEDDED PROGRAMMING

Design Microprogrammed Systems

Create Reliable Software

Reprogrammable UAV Autopilot

PLUS

Audio-Visual Design

DSP, Convolution, & Rapid Digital Filter Implementation

// Build a Color Organ
// Convolution Defined
// The Effective Use of Filters
// Constant Volume Control Algorithm

0 4>
7 25274 75349 9
$7.50 U.S. ($8.50 Canada)
www.circuitcellar.com

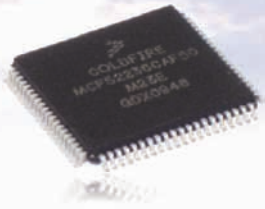
Low-cost Industrial
Serial to **Ethernet** Solutions

NetBurner
Networking in One Day!

- Instantly network-enable any serial device
- No programming is required for serial to Ethernet application
- Customize to suit any application with a development kit

SBL2e Chip

2-port serial-to-Ethernet server with eight A/D converter inputs and optional SPI, I2C peripheral device support



SBL2e 200

2-port serial-to-Ethernet server with four A/D converter inputs, optional I2C peripheral support and 10-pin header

SBL2e 100

2-port serial-to-Ethernet server with four A/D converter inputs, optional I2C peripheral support and RJ45 connector

SBL2e

Serial-to-Ethernet server with RS-232 support

SBL2e XA

External server with up to four A/D converter inputs, up to eight digital I/O and up to two UARTS (one RS-232)

Hardware Features

Up to three serial ports, 10/100 Mbps Ethernet, up to 10 digital I/O, 12-bit A/D converters, operating temperature -40 to 85 C, 32-bit performance

Software Features

TCP/UDP/Telnet/HTTP modes, DHCP/Static IP support, web-based or AT command configuration

Low Prices

SBL2e Chip ..... $10.50 (Qty. 10k, Device P/N: SBL2eCHIP-236IR)
SBL2e 200 ..... $19.95 (Qty. 1000, Device P/N: SBL2e-200IR)
SBL2e 100 ..... $21.95 (Qty. 1000, Device P/N: SBL2e-100IR)
SBL2e X ..... $79.00 (Qty. 1000, Device P/N: SBL2eX-100IR)
SBL2e XA ..... $79.00 (Qty. 1000, Device P/N: SBL2eXA-100IR)

Need a custom solution?

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

Information and Sales | sales@netburner.com

Web | www.netburner.com **Telephone** | 1-800-695-6828

5 Competitive Advantages

Overseas Manufacturing

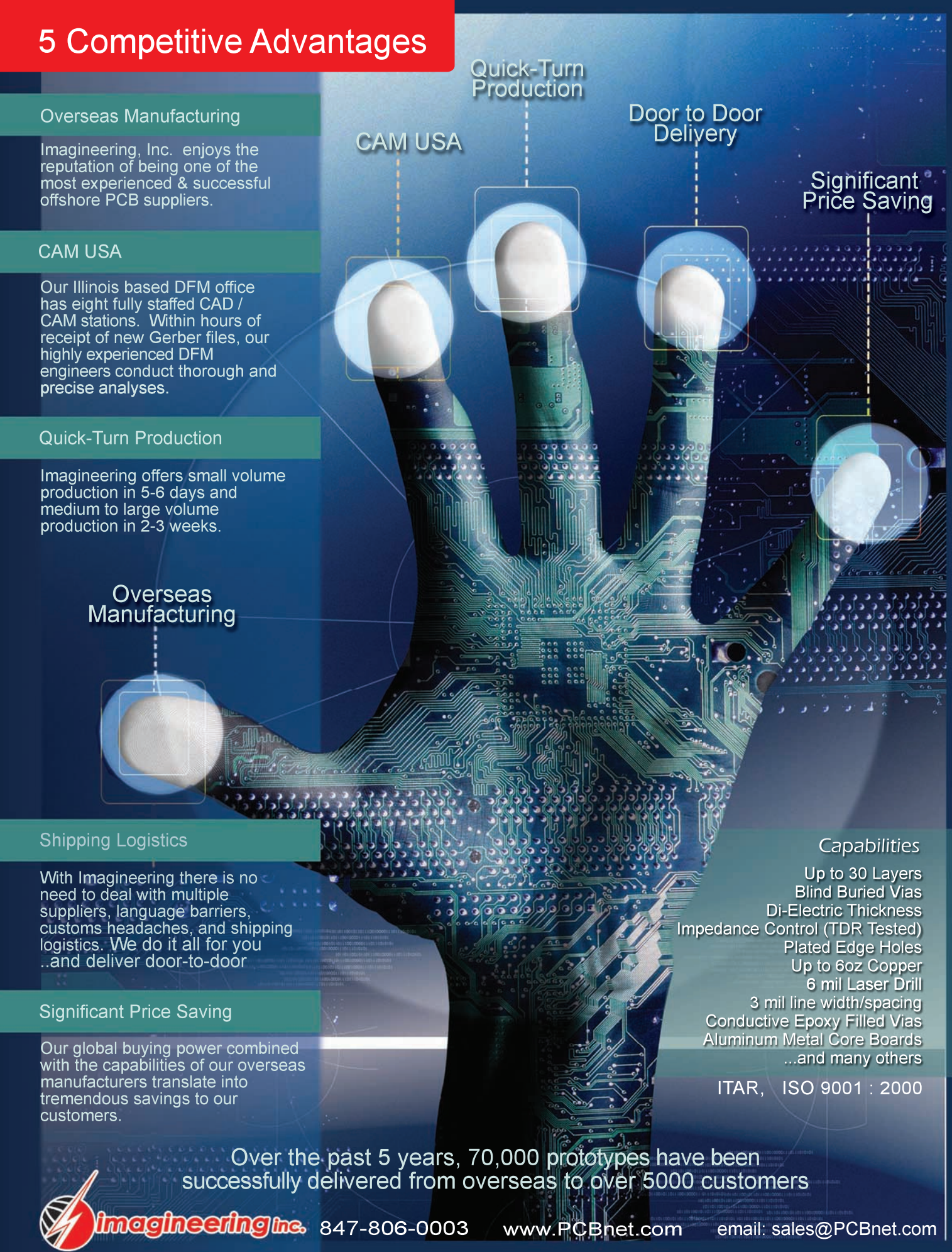
Imagineering, Inc. enjoys the reputation of being one of the most experienced & successful offshore PCB suppliers.

CAM USA

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

Quick-Turn Production

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



Shipping Logistics

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

Significant Price Saving

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

Capabilities

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

ITAR, ISO 9001 : 2000

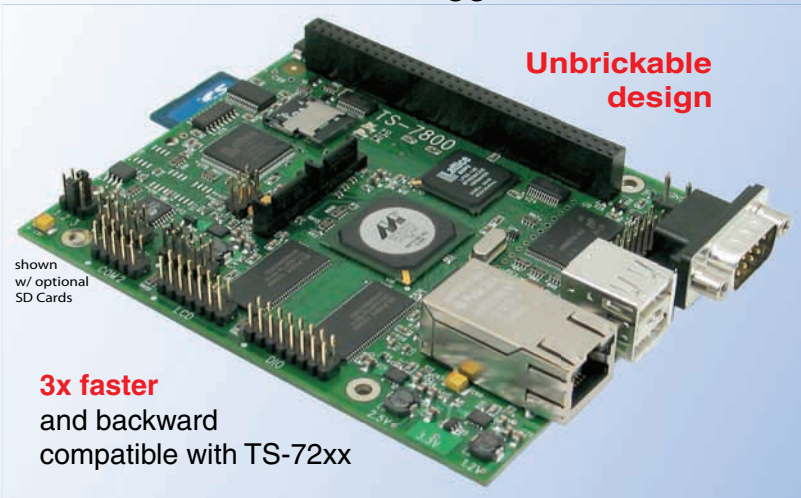
Over the past 5 years, 70,000 prototypes have been successfully delivered from overseas to over 5000 customers

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

23 YEARS IN BUSINESS...AND STILL GOING STRONG

Embedded Systems

High-End Performance with Embedded Ruggedness



shown w/ optional SD Cards

Unbrickable design

3x faster and backward compatible with TS-72xx

TS-7800
500 MHz ARM9

- Low power - 4W@5V — $229 qty 100
- 128MB DDR RAM
- 512MB high-speed (17MB/sec) onboard Flash — $269 qty 1
- 12K LUT 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

TS-TPC-7390
7" Color Touch Panel Computer

- Low Power, Industrial Quality Design
- Mountable aluminum frame
- 200 MHz ARM9
- 64MB SDRAM (128MB opt)
- 512MB Flash w/ Debian Linux
- 800 x 480 video core
- Dedicated framebuffer- 8MB RAM
- Audio codec with speaker
- Boots Linux 2.6 in less than 2 seconds
- Unbrickable, boots from SD or NAND
- Runs X Windows GUI applications
- Runs Eclipse IDE out-of-the-box

$449 qty 1

More Touch Panel Computers on our website

- 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

TS-WIFIBOX-2

A Complete Solution for 802.11g WiFi Applications

NEW!

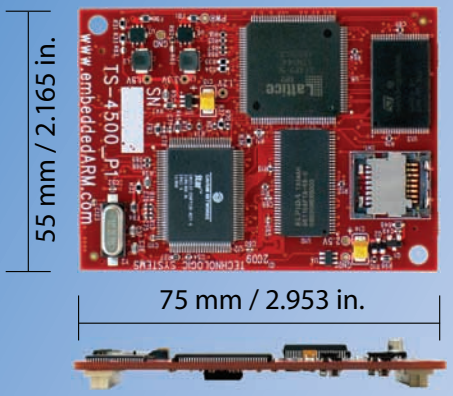
$185 qty 1

- Low power (3.2 watts), fanless
- Power via 5-12 VDC, USB, PoE (opt.)
- 64MB DDR-RAM
- 256MB ultra-reliable XNAND drive
- Micro-SD Card slot
- RS-232, RS-485, CAN, RTC
- Header with SPI and 11 DIO
- 480Mbit/s USB, Ethernet, PoE option
- Boots Linux 2.6.24 in < 3 seconds
- Un-brickable, boots from SD or flash
- Customizable FPGA - 5K LUT
- Optional DIN mountable enclosure

Ideal for gateway or firewall, protocol converter, web server, WiFi audio, and unattended remote applications

NEW!

series starts at $92 qty 100



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 projects include:

- TS-4200: Atmel ARM9 with super low power
- TS-4300: Cavium ARM11 with dual 600 MHz and FPU
- TS-4500: Cavium ARM9 at very low cost
- TS-4700: Marvell PXA168 with video and 1.2 GHz CPU
- TS-4800: Freescale iMX515 with video and 800 MHz CPU
- Several COTS baseboards for evaluation & development

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

Technologic Systems

We use our stuff.

Visit our TS-7800 powered website at

www.embeddedARM.com

TASK MANAGER

Keeping It Professional

What do you think of the current state of free content overkill? I think I know.

Like the late-'90s analyst trying to find truth in a new dot-com's financial statements, today's professional engineer finds it difficult to target reliable engineering information when he's faced with hundreds of press releases, free trade magazines, and anonymous online project postings. Type an engineering term or part number into your favorite search engine and you'll find thousands of pages of information. But what's reliable? What's been tested? On whom can you rely? Do you trust the company plugging its own products? Do you value the opinion of an industry analyst communicating via one-minute videos? What kinds of credentials do all those wiki contributors have? Or maybe you simply turn to your peers who post comments on blogs and in forums? But how many peer reviews do you need to read? Ten? Fifty? Who's monitoring that forum anyhow?

We know how you feel. It's exciting that there's a wide range of useful information out there ready to be soaked up. But finding what you need can be daunting. The amount of data filling your inboxes, appearing in your newsfeeds, and downloading to your smartphones is pretty ridiculous. There comes a point when you need to cut the clutter and focus on what matters.

That's where *Circuit Cellar* comes in. We're more than a magazine. We're dedicated to delivering professional information in whatever manner you require during this ever-changing era of "anything goes" content swapping, DIY videos, and data saturation. Sure, we like the idea of peer-to-peer content sharing—and we plan to stimulate more of that among our readership in the coming months—but in the end, we'll remain a publisher of the embedded projects, engineering ideas, design tips, and programming methods you demand.

That's great, you say. But how will *Circuit Cellar* stand out as the number of social media sites, smartphone apps, and amateur project pages increases? The answer is simple. We'll adjust to *your* needs rather than blindly follow social trends. As your needs evolve, we'll adapt to deliver the information you need in the ways you request it. For instance, during the past few months, we've found many of our readers and advertisers on a few key social media sites. Thus, we've begun to engage them in those venues. And in the coming months, you'll see more of *Circuit Cellar* on Twitter, Facebook, and the like. With that said, we won't waste time and resources in areas that don't interest our readers.

Now that your minds are at ease, sit back, relax, and let our authors inspire you to start your next project. This issue's main topic, embedded programming, is truly what separates novice "designers" from the pros.

C. J. Abate, cj@circuitcellar.com

CIRCUIT CELLAR®

THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

FOUNDER/EDITORIAL DIRECTOR
Steve Ciarcia

EDITOR-IN-CHIEF
C. J. Abate

ASSOCIATE EDITOR
Nan Price

CONTRIBUTING EDITORS
Jeff Bachiochi
Robert Lacoste
George Martin
Ed Nisley

NEW PRODUCTS EDITOR
John Gorsky

PROJECT EDITORS
Ken Davidson
David Tweed

STAFF ENGINEER
John Gorsky

PUBLISHER
Hugo Van haecke

ASSOCIATE PUBLISHER
Shannon Barraclough

CUSTOMER SERVICE
Debbie Lavoie

CONTROLLER
Jeff Yanco

ADMINISTRATIVE COORDINATOR
Valerie Luster

ART DIRECTOR
KC Prescott

MARKETING ASSISTANT
Daniel Netkin-Colllins

GRAPHIC DESIGNERS
Grace Chen
Carey Penney

Circuit Cellar's mission is to collect, select, and disseminate need-to-know information around the world in the fields of embedded hardware, embedded software, and computer applications. Circuit Cellar uses an assortment of print and electronic content-delivery platforms to reach a diverse international readership of professionals, academics, and electronics specialists who work with embedded, MCU-related technologies on a regular basis. Our aim is to help each reader become a well-rounded, multidisciplinary practitioner who can confidently bring innovative, cutting-edge engineering ideas to bear on any number of relevant tasks, problems, and technologies.

ADVERTISING
800.454.3741 • 978.281.7708 • www.circuitcellar.com/advertise

ADVERTISING REPRESENTATIVE
Peter Wostrel
Strategic Media Marketing, Inc.
1187 Washington St., Gloucester, MA 01930 USA
800.454.3741 • 978.281.7708
peter@smmarketing.us • www.smmarketing.us
Fax: 978.281.7706

ADVERTISING COORDINATOR
Kim Hopkins
E-mail: khopkins@circuitcellar.com

Cover photography by Chris Rakoczy—Rakoczy Photography
www.rakoczyphoto.com

PRINTED IN THE UNITED STATES

CONTACTS

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

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

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

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

CIRCUIT CELLAR®, THE MAGAZINE FOR COMPUTER APPLICATIONS (ISSN 1528-0608) is published monthly by Circuit Cellar Incorporated, 4 Park Street, Vernon, CT 06066. Periodical rates paid at Vernon, CT and additional offices. **One-year (12 issues) subscription rate USA and possessions $45, Canada/Mexico $60, all other countries $63. Two-year (24 issues) subscription rate USA and possessions $80, Canada/Mexico $110, all other countries $116.** All subscription orders payable in U.S. funds only via Visa, MasterCard, international postal money order, or check drawn on U.S. bank. **Direct subscription orders and subscription-related questions to Circuit Cellar Subscriptions, P.O. Box 5650, Hanover, NH 03755-5650 or call 800.269.6301.**

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

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

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

Entire contents copyright © 2011 by Circuit Cellar, Incorporated. All rights reserved. Circuit Cellar is a registered trademark of Circuit Cellar, Inc. Reproduction of this publication in whole or in part without written consent from Circuit Cellar Inc. is prohibited.

Scan Here

mouser.com
Semiconductors and electronic components for design engineers.

It's **hip** to be square.

Advanced **semiconductor** solutions to fuel your creativity.

Come to Mouser first for what's next from these industry leaders.

mouser.com The Newest Products For Your Newest Designs™

MOUSER ELECTRONICS®

a tti company

A BERKSHIRE HATHAWAY COMPANY

Mouser and Mouser Electronics are registered trademarks of Mouser Electronics, Inc. The Newest Products for Your Newest Designs is a registered trademark of Mouser Electronics, Inc

INSIDE ISSUE 249

April 2011 • Embedded Programming

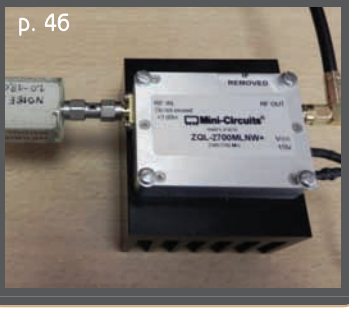
UAV Control System

p. 24

Digital Signal Processing

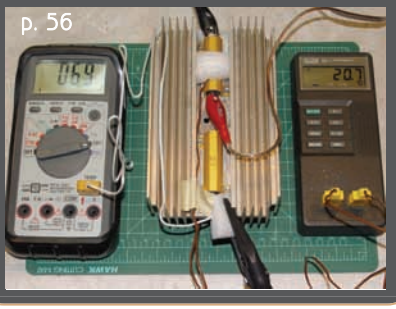
p. 38

Noise Analysis



p. 46

Heat Measurements



p. 56

Making Your Devices Smart & Simple!

Internet Connectivity Wizard

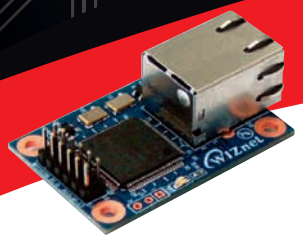
WIZnet's uniquely patented Hardware TCP/IP technology has proven effective in millions of network-enabled devices around the world, allowing users to easily add internet to the devices with greater stability to ensure faster product releases out to the market.

Hardware TCP/IP Offload Technology

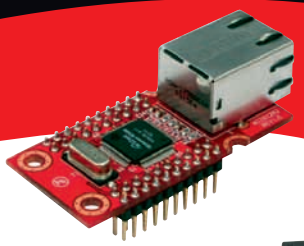
- Offloading Protocol Processing from a System MCU
- Stable Hardwired TCP/IP Logic
- Silicon-proven IP for SoC and ASSP
- High Performance & Low Cost Platform
- No Booting Time
- Simple System Composition
- Best-fits OS-less Device

Applications

- Smart Grid and Smart Meter
- Medical Device Management
- Industrial Control
- Robot Control
- Security System



Serial-to-Ethernet Module



Drop-in Network Module

Wireless LAN Module

External Device Server

Hardware TCP/IP Chip

Get more information at www.wiznettechnology.com

3003 North First Street, San Jose, CA 408-232-5415
sales@wiznettechnology.com

WIZnet TECHNOLOGY

32-BIT PIC PORTFOLIO EXPANDS WITH ADDITIONAL MEMORY OPTIONS

A new, six-member family of 32-bit **PIC32MX5/6/7** microcontrollers that provides the same integrated Ethernet, CAN, USB, and serial connectivity peripherals with new, more cost-effective memory options is now available from Microchip. Additionally, design enhancements have been made that provide lower power consumption of 0.5-mA/MHz active current, higher flash memory endurance of 20,000 read/write cycles, and better EEPROM emulation capability. By maintaining common pinouts, the PIC32 portfolio provides designers with a seamless migration path to achieve the correct balance of memory and cost for their high-performance applications.

Embedded designers are constantly looking for ways to lower their costs without sacrificing performance or functionality. Microchip's newest 80-MHz PIC32 microcontroller family meets these needs by maintaining best-in-class performance of 1.56 DMIPS/MHz, and integrating Ethernet, CAN, USB, and multiple serial communication channels, while offering more cost-effective memory options. Specifically, the family offers 32 KB of RAM and up to 140 KB of flash.

All six members of this new PIC32MX5/6/7 family are available today for volume production. Additionally, all six are available in 100-pin TQFP 12 mm × 12 mm, TQFP 14 mm × 14 mm, and BGA packages, as well as 64-pin TQFP and QFN packages. Pricing for 10,000 units ranges from **$3.71** each for the PIC32MX534F064H and up to **$4.93** each for the PIC32MX764F128L.

Microchip Technology, Inc.
www.microchip.com

FREESCALE EXPANDS TOWER SYSTEM

The **Freescale Tower System** development platform continues to expand with new, feature-rich development boards. Recently, Freescale announced four new controller modules, two interchangeable peripheral modules, and five complete Tower System kits for the reconfigurable development platform.

The new controller modules feature the Kinetis family of MCUs based on the ARM Cortex-M4 core (TWR-K40X256 and TWR-K60N512), as well as 8-bit S08 (TWR-MC9S08JE) and 32-bit ColdFire Flexis MCUs (TWR-MCF51JE). Two new peripheral modules add high-precision analog functionality and low-power Wi-Fi for sensor and embedded applications that require battery-powered wireless connectivity.

The newest Tower System Wi-Fi module (TWR-WIFI-G1011MI) provides easy-to-use Wi-Fi connectivity. The analog module (TWR-ADCDAC-LTC) features high-performance data conversion products and acts as a "playground" for engineers to evaluate and develop applications requiring high-resolution analog functionality.

The Kinetis Tower System kits (TWR-K40X256-KIT and TWR-K60N512-KIT) include their respective MCU modules and are packaged with the serial module (TWR-SER) and the elevator boards (TWR-ELEV). The Flexis Tower System kits (TWR-S08JE128-KIT and TWR-MCF51JE-KIT) include their respective MCU modules and are also packaged with the serial module and the elevator boards.

All of the modules and kits are economically priced allowing engineers to start designing without worrying about high start-up costs.

Freescale Semiconductor
www.freescale.com

NEW PRODUCT NEWS

Edited by John Gorsky

POWERFUL USB DIGITAL-PATTERN GENERATORS

The **GP-24116** and **GP-24132** combine a high-speed digital-pattern generator, logic analyzer, and bidirectional (SPI/I²C) host adapter with either 16 MB or 32 MB of internal memory. GP-241XX devices offer high-speed programmable bidirectional interfaces that allow the stimulation and analysis of digital electronic systems, providing flexible and powerful access to electronic devices under test.

GP-241XX devices are delivered with the 8PI Control Panel 2.xx software suite, including: Windows GUI, TCL/tk interface and access to C/C++ DLL. This new release allows multiple devices control, enabling the implementation of a complete stimulus-and-response loop with portable USB 2.0 devices. The C programming interface allows for developing custom test programs from any C-compatible environment to automate and repeat specific tasks.

With an internal embedded memory buffer and 100-MHz operation on all 16 digital lines, GP-241XX devices provide powerful general purpose and expandable PC-based tools that ideally complement existing system debug equipment, such as logic analyzers and high-speed oscilloscopes. The GP-241XX can be used as an arbitrary digital pattern generator, logic analyzer, or serial protocol master/analyzer with 16 address/data lines and six control lines, offering up to 100 MHz (200-MB/s burst) performance.

GP-24116 and GP-24132 are available now from Byte Paradigm and its network of distributors starting at approximately **$1,500** with standard options, software, and cables.

Byte Paradigm
www.byteparadigm.com

ZIGBEE REMOTE CONTROLS TEST SERVICE

Test, certification, and compliance specialist, TRaC, has further expanded its capabilities in ZigBee Testing to include **testing for ZigBee Remote Control** (ZRC) devices. The new service covers both the remote control devices and their target counterparts, with ZRC increasingly being employed in cutting-edge equipment such as HDTV, Blu-ray, Set Top Boxes, and other products using similar advanced technology.

The ZigBee Remote Control standard has been developed by the ZigBee Alliance, and defines an advanced RF remote control system that is built on an RF4CE platform. The two-way connection, which this new technology allows, provides considerably more functionality over traditional IR remote controls and removes the issues associated with line of sight control.

TRaC already offers a complete platform test service to RF4CE; however, the test specification for ZigBee Remote Controls has recently been published, allowing manufacturers of ZRC devices to seek profile certification and apply the ZigBee and ZRC logos.

The ZigBee Remote Control Profile Testing for both controllers and target devices is performed at the ZigBee Alliance-recognized test facility based in Hull, UK. TRaC's ZRC service provides valuable assistance at all stages of the design process, from initial concept through design, to final implementation. Both pre-testing and full certification testing are available.

TRaC Global Ltd.
www.tracglobal.com

ESD ROLL CREATES INSTANT "BLANKET OF STATIC PROTECTION"

EquaStat ESD Roll fabric creates an instant blanket of static protection on work surfaces or wherever static protection is needed on the spot. Economical, reusable, and effective, the fabric can be cut to length to provide a static-dissipative cover for any work surface. Technicians can cover a workbench prior to spraying spray flux or conformal coating on a PC board or electronic assembly. EquaStat ESD Roll fabric can also be used to cover shelves in storage areas where sensitive electronic assemblies are stored, or on benches where rework is to be performed when ESD-safe surfaces are not available. Simply measure out the desired amount of material to cover the work area, cut to size, perform the task, and then remove.

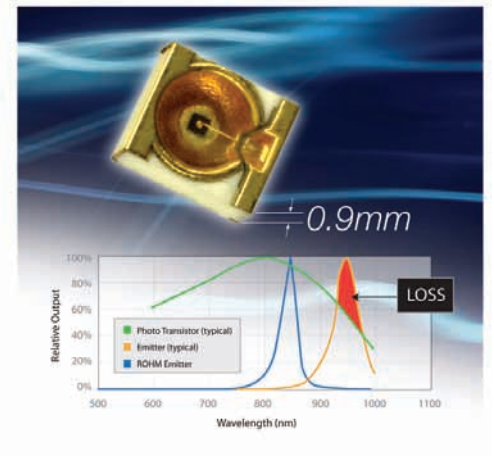
EquaStat ESD Roll is a heavy-weight fabric made from continuous filament polyester that is interwoven with DuPont Negastat carbon yarn to provide a continuous, tough, static-dissipative fabric. It can be measured and cut to length to provide an ESD-safe protective covering for a variety of work surfaces, including benches, storage shelves, trolley carts, spray booths, lab tables, desks, test stations, and demonstration areas. The same cut length of material can be re-used over and over until it becomes contaminated or torn, and then simply disposed of and replaced with fresh fabric. The material can be taped or fastened to the work surface. Each ready-to-use roll of EquaStat is wound on a 1.5" ID core tube that is 30" wide with 50 linear feet of material. The rolls are shrink-wrapped in an ESD-safe, static-dissipative film.

Contact JNJ Industries for pricing.

JNJ Industries, Inc.
www.jnj-industries.com

HIGH-PERFORMANCE INFRARED LEDs FOR PROXIMITY SENSING

A new series of high-performance infrared LEDs ideally suited for proximity sensor applications is now available from ROHM Semiconductor. The **SIM-030/031ST** and **SIM-040/041ST** surface-mount IR LEDs feature breakthrough IR wavelength technology providing peak output of 850/870 nm, compared to 950 nm for comparable devices. The 850/870 nm level is much closer to the peak wave sensitivity of phototransistor sensors, thus achieving higher-efficiency proximity sensing and energy savings of up to 66%. In addition, the small package footprint and low profile further enhance their application in cell phones and other portable devices.



The SIM-030/031ST, the smaller of the two devices, has a 2.3 mm × 1.95 mm footprint and a height of just 0.9 mm. With a forward current (IF) of 100 mA, the device delivers typical emission strength of 30 mW per steradian (mW/sr). The slightly larger SIM-040/041ST (3.1 mm × 2.25 mm × 1.6 mm) provides 40 mW/sr typical emission strength.

The ROHM SIM-030/031ST and SIM-040/041ST are the latest additions to the ROHM Optical Sensors Series of emitters, receivers, photo interrupters, tilt sensors, IrDA modules, and remote-control modules. Pricing starts at **$0.75** each for small OEM quantities.

ROHM Semiconductor
www.rohm.com

16 × 2 MULTIPLEXER AFFORDABLY CONNECTS EQUIPMENT

The new **Model QMUX16X2** multiplexer is designed expressly to permit test and measuring equipment to connect to many signals. The QMUX16X2 can measure two selected signals at the same time. This easy-to-use, affordable multiplexer features 16 inputs and two outputs. It can be powered and controlled by the user's PC. The multiplexer uses low-power reed relays and a USB controller in a DIN rail-mount enclosure. As such, it serves as the unique "missing link" in a complex test setup. It's suited to single or differential signals at speeds less than 10 MHz.

Because the device uses mechanical reed relays, you can use it as a de-multiplexer as well as a multiplexer. The de-multiplexing capabilities of the QMUX16X2 permit two input channels to be connected to any of the 16 output channels on a standard 40-pin IDC header. It is controlled through an easy-to-use software API over an industry-standard USB 2.0 full-speed port.

The QuickMux multiplexer is expandable too. The design includes a pass-through relay, which makes it possible to daisy chain multiple units without sacrificing an input channel. The company also offers a cable with built-in daisy chain connectors.

The QuickMux is entirely powered and controlled by a single, full-speed USB 2.0-compliant port. It costs **$249**.

Bitwise Systems
www.quickmux.com

PROPELLER-BASED PEN ROBOT

The **Scribbler S2** robot (part number 28136) is suitable for a variety of programming skills. The Scribbler robot arrives preprogrammed with eight demo modes, including light seeking, object detection, object avoidance, line following, and art. Place a Sharpie marker in the pen port and it will scribble as it drives. Next, use the graphical user interface (S2 GUI) tile-based programming tools, or modify the Propeller source code in the BASIC-like Spin language. Through the use of third-party tools, you can also program the S2 on a Mac or under Linux, in PropBASIC and C (resources for these languages will follow the release). The S2 is fully compatible with the Georgia Tech IPRE Fluke, too.

The S2 GUI is backward-compatible with the original S1 GUI. However, coders will use Spin for the Propeller instead of PBASIC, as they did for the BASIC Stamp in the S1. Examples make the transition easy. The benefits and flexibility of Spin in multi-core systems provides easy compartmentalization of S2 subroutines that run concurrently with shared memory. Controlling motors, managing sensors, and interfacing with the hacker port can be done concurrently even while playing sound. The Propeller makes it all possible.

The S2 costs **$129.99**.

Parallax, Inc.
www.parallax.com

ARM CORTEX-A8 AM3517 SYSTEM ON MODULE

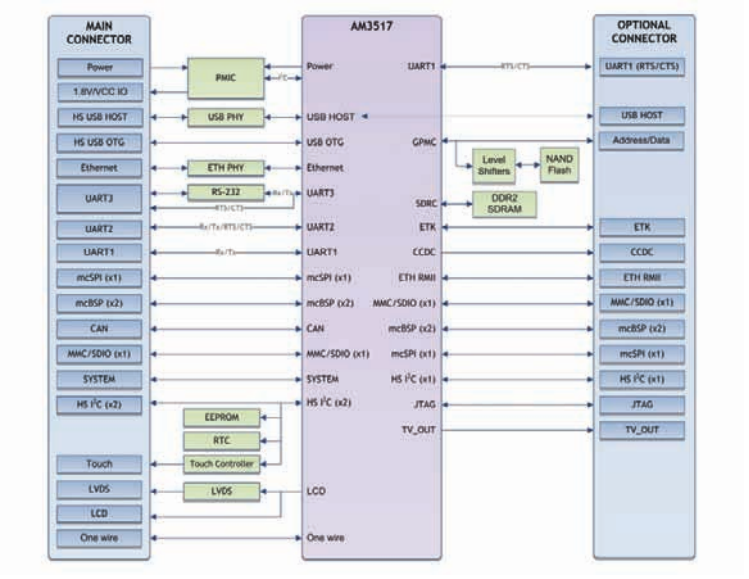
PHYTEC America, a leader in miniature, low-power, high-performance system on module technology announces the new **phyCORE-AM3517** System on Module (SOM), populated with Texas Instruments's Sitara ARM Cortex-A8 AM3517 microprocessor.

Excellent for medical and industrial control applications, the AM3517 MPU has a 600-MHz ARM Cortex-A8 Core with NEON SIMD coprocessor and Vector floating-point coprocessor. The AM3517 also boasts a POWER SGX graphics accelerator subsystem for 3-D graphics acceleration, a display subsystem enabling multiple concurrent image manipulation, and a programmable interface that supports a wide variety of displays, making the AM3517 ideal for high-performance video, imaging, and graphics-intensive applications. The phyCORE-AM3517 SOM also supports the AM3505 MPU.

The phyCORE-AM3517 SOM is available in PHYTEC rapid development kits (RDKs), which include the SOM, Carrier Board, LCD, and all accessories required for immediate start-up. The Carrier Board serves as a reference design for target hardware development and the RDKs provide an immediate platform on which to develop application code that can be seamlessly integrated along with the SOM into prototype, pre-production, and production systems. OEMs with aggressive timelines or limited resources can employ PHYTEC's full range of design services, which include embedded hardware design, Linux/Windows embedded design development, and complete turnkey designs.

Contact PHYTEC for pricing on a configuration to fit your needs.

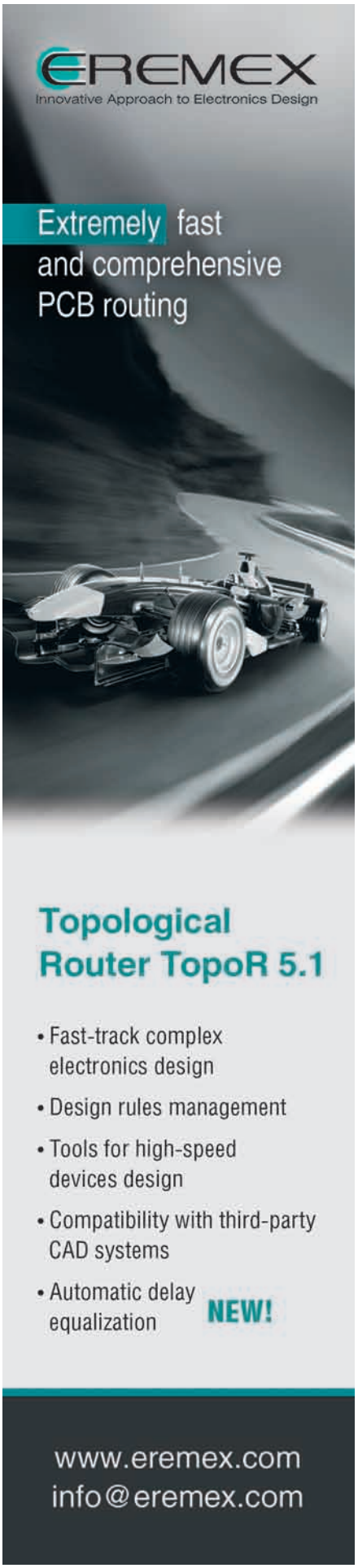
PHYTEC America, LLC
www.phytec.com



NPN

Easy Embedded Linux
OmniFlash
16MB FLASH / 32MB RAM
200 Mhz Arm9 CPU
16 Digital I/O
Watchdog
10/100 Ethernet
Battery backed Clock /Calendar
Audio In/Out
2 USB
2 Serial Ports
We brought you the world's easiest to use DOS controllers and now we've done it again with Linux. The **OmniFlash** controller comes preloaded with Linux and our development kit includes all tools you need to get your project up and running fast.
Out-of-the-box kernel support for USB mass storage and 802.11b wireless, along with a fully integrated Clock/Calendar puts the **OmniFlash** ahead of the competition.
Call 530-297-6073 Email sales@jkmicro.com
www.jkmicro.com
JK microsystems, Inc.
International Orders Welcome
MADE IN USA

RF Specialists
Distributors of Low-Power RF modules, Zigbee, Bluetooth, WiFi, GPS, GSM/GPRS
ZigBee Solutions
ProBee Series
High Performance and Robust Data Transfer Capability
ZigBee Solutions
Low Cost
Low Power
LEMOS INTERNATIONAL
www.lemosint.com
866.345.3667
sales@lemosint.com



28-nm FPGA PORTFOLIO

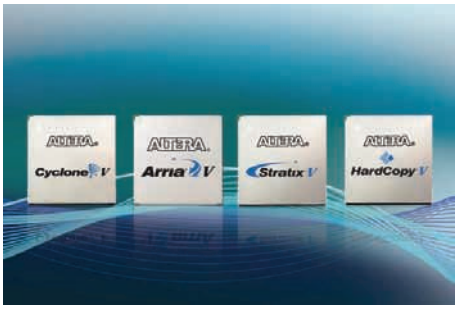
Altera Corp. has announced its new portfolio of **28-nm devices**. These devices leverage advantages in transceiver technology, product architecture, intellectual property integration, and process technology to optimize solutions that address diverse design challenges.

For applications such as motor control, displays, and software-defined radios, where low power and board space are concerns, the Cyclone V FPGA family is an ideal fit. The Cyclone V family offers 40% lower total power versus the previous generation devices, 12 transceivers operating at up to 5 GBps, hardened PCIe Gen2 ×1 blocks, and hard-memory controllers supporting LPDDR2, mobile DDR, and DDR3 external memory.

Targeting applications that require a balance of cost, low power, and high performance—such as remote radio units, in-studio mixers, and 10G/40G linecards—is the Arria V FPGA family. Offering 40% lower total power versus previous generation devices, Arria V devices include transceivers operating at up to 10 GBps, hard memory controllers supporting DDR3 external memory, and efficient systolic finite impulse response (FIR) filters with variable-precision DSP blocks.

Both the Stratix V FPGAs and HardCopy V ASICs have also been upgraded to 28-nm technology. The Stratix V family addresses a broad range of high-bandwidth applications such as advanced LTE base stations, high-end RF cards, and military radar. HardCopy V ASICs extend Altera's leadership in low-nonrecurring-engineering, low-risk transceiver-based ASICs.

Contact Altera or one of its distributors for pricing.



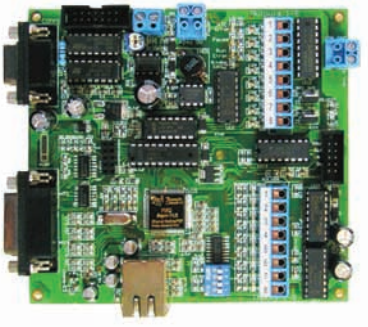
Altera Corp.
www.altera.com

EMBEDDED PLCs WITH ETHERNET

The **FMD88-10** and the **FMD1616-10** PLCs are Triangle Research International's latest Ethernet-equipped programmable logic controllers for OEMs. With the new FMD PLCs, Triangle Research now has a full range of highly integrated "Super PLCs" starting from the compact Nano-10 to the powerhouse F-series. This super PLC series combines the powerful and easy-to-use i-TRiLOGI Ladder+BASIC software with a wide array of features, including but not limited to: built-in digital and analog I/Os, PWM, PID, encoders, stepper controls, and on-board communication ports for connecting to other devices.

The FMD88-10 comes with eight digital inputs, eight digital outputs, and 10 analog I/Os. The FMD1616-10 comes with 16 digital inputs, 16 digital outputs, and 10 analog I/Os. Both models are equipped with an I/O expansion port, an LCD interface, RS-232 and RS-485 serial ports, and the Ethernet port, which has become increasingly indispensable today. Triangle Research's iTRILOGI client/server software and the support of MODBUS TCP/IP protocols not only make the FMD model PLCs remotely accessible for machine monitoring and OEM troubleshooting/reprogramming, but also enable their easy integration into mixed-brand PLC environments and networks.

The sub-$300 pricing of the FMD PLCs is rare for full-feature, Ethernet PLCs in this popular I/O range, making this PLC a particularly accessible choice for value-conscious OEMs. The FMD88-10 and the FMD1616-10 single-unit prices are **$229** and **$295**, respectively, and are further discounted with OEM quantity price breaks.

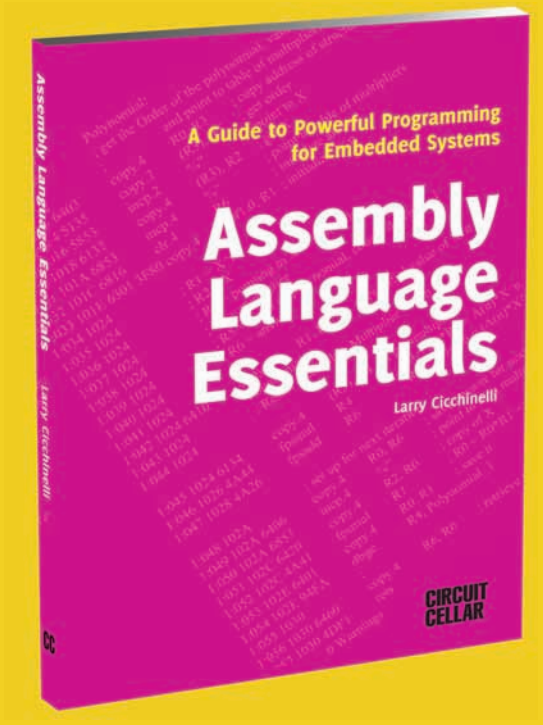


Triangle Research International, Inc.
www.tri-plc.com

NPN

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!**

NEW BOOK!

$41.00

(Special Introductory Price)

Pre-Order now thru April 31 and get FREE shipping!

www.cc-webshop.com

CIRCUIT CELLAR®
THE MAGAZINE FOR COMPUTER APPLICATIONS

*Offers valid until April 30, 2011. Books to ship by May 1.

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

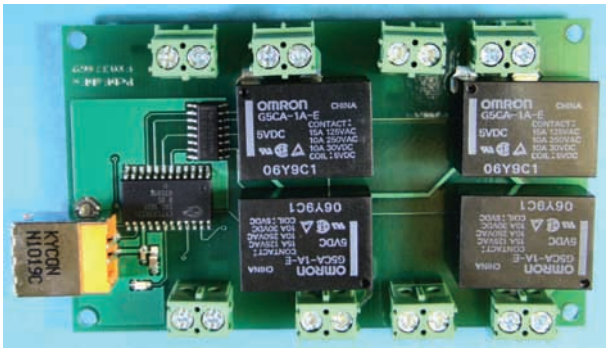
HUMANDATA
FPGA Boards from Japan
• Basic and simple features
• Single power supply operation
• Same size/connector layout - ACM/XCM series
• Free download technical documents before purchasing
• High quality and highly reliable board from Japan
• Stocked items are ready to ship immediately
Feel free to contact us to customize speed grade or any features.
0-Series
2-Series
3-Series
1-Series
ALTERA FPGA boards
ACM-023 CycloneIV E F484 FPGA board
Cyclone IV E MRAM
EP4CE55F23C8N
EP4CE75F23C8N
EP4CE115F23C8N
Credit card size (86x 54 mm)
RoHS compliant
EDA-004 Cyclone III USB-FPGA board
Cyclone III USB Config. USB Comm. MRAM
EP3C55F780C8N
FPGA configuration via USB interface
Credit card size (86 x 54 mm)
RoHS compliant
XILINX FPGA boards
XCM-018 Spartan-6 FGG484 FPGA board
Spartan-6 MRAM DDR2
XC6SLX45-2FGG484C
XC6SLX75-2FGG484C
XC6SLX100-2FGG484C
XC6SLX150-2FGG484C
Credit card size (86 x 54 mm)
RoHS compliant
EDX-006 Virtex-5 USB-FPGA board
Virtex-5 USB Config. USB Comm. MRAM
XC5VLX30-1FFG676C
FPGA configuration via USB interface
Credit card size (86 x 54 mm)
RoHS compliant
XCM-107 Virtex-5 LXT FPGA board
Virtex-5 Rocket IO
XC5VLX30T-1FFG665C
XC5VLX50T-1FFG665C
Compact size (43 x 54 mm)
RoHS compliant
Over 100 varieties of FPGA/CPLD boards are available.
See all our products at:
www.hdl.co.jp/CC/
HuMANDATA LTD.
E-mail: s2@hdl.co.jp
Fax: 81-72-620-2003

USB AC MONITOR AND RELAY

The **Model J5B375** monitors up to four AC lines. Along with the AC monitors there are four 15-A SPST relays. The AC monitor inputs sense AC voltage, and the current state of each input can be read from the USB port. The SPST relay outputs provide host control between a common line and normally open connection point.

Models support two or four monitors and relays. The user communicates with the module from any programming or test language that supports USB communications as shown in provided sample applications. This small form-factor module replaces internal PC-based plug-in cards in various test, control, monitor, and measurement applications. Complete product specifications are available at the company's website.

Single-unit pricing for the Model J5B370-02 (two relays) is **$95**. The Model J5B370-04 (four relays) is **$137.50**. Delivery is from stock.



J-Works, Inc.
www.j-works.com

SCAN-ON-DEMAND VIRUS DETECTOR FOR INDUSTRIAL COMPUTERS

Modern industrial systems increasingly use Windows OS-based off-the-shelf industrial computers. These systems are at higher risk of being attacked by the same viruses that affect office computers and can cause expensive plant shutdowns or service disruptions. Most industrial computers have only intermittent access to the Internet, if any, which prevents up-to-date virus definition and program version updates. In addition, with the highest priority placed on maximum performance, risk of system slowdowns due to real-time virus monitoring is another barrier to installation-type virus programs.

The **Vaccine USB** is designed to meet these challenges through its portable virus scan software that requires no software installation. Powered by anti-virus industry leader McAfee, up-to-date virus definitions can be downloaded to the Vaccine USB from any Internet-accessible computer so that users can perform virus scans with the latest definition file.

Upon insertion of the Vaccine USB, the virus scan program stored on its emulated CD-ROM partition will auto run, a significant benefit for industrial computers that may lack a keyboard, mouse, or monitor. The Vaccine USB default is set to perform virus "scan only," but can be configured to "scan and remove" the infected files as well. The scan results are provided by LED lights on the device; a red light indicates viruses detected, a blue light indicates a clean system. A detailed scan log is saved on the device, allowing further review by IT technicians.

The Vaccine USB is priced at **$499** for a unit with a one-year subscription.

Hagiwara Sys-Com Co., Ltd.
www.hsc-us.com

NPN

CIRCUIT CELLAR

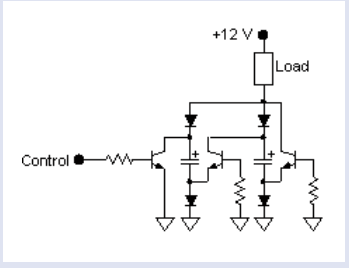
Test Your EQ

Edited by David Tweed

ANSWERS for Issue 248

Answer 1—Assuming the control signal has been off for a while, then C1 has been charged to 12 V through the load and the two diodes. When Q1 is turned on by the control signal, it pulls the upper end of C1 to ground, which forces the lower end of C1 to (almost) –12 V. Since the base of Q2 is tied to ground via a current-limiting resistor, it now turns on, pulling the low end of the load (temporarily) to –12 V and allowing the load current to discharge C1.

Once C1 has discharged to about 1 V—the exact value depends on the $V_{CE(SAT)}$ of Q1 plus the V_{BE} of Q2—then Q2 turns off again.



Answer 2—D1 conducts during the initial charging, but when Q2 turns on, the anode of D1 is pulled below ground, causing it to stop conducting. Once Q2 cuts off again, D1 resumes conducting, carrying the "hold" current of the load.

D2 conducts only during the charge cycle. As soon as Q1 turns on, the anode of D2 is pulled negative, and it doesn't go positive again until Q1 cuts off.

Answer 3—All of it. When Q2 is conducting during the first part of the active cycle, the current through it is passing through C1 and Q1 as well. When Q2 cuts off and D1 starts conducting, this current also passes through Q1.

Answer 4—A diode across the load wouldn't hurt anything, but it's probably superfluous. As long as C1 is large enough to store the inductive energy of the load without exceeding the supply voltage, the extra diode will never conduct. The inductive energy simply helps to recharge C1 for the next activation of the circuit.

Answer 5—The same principle can be applied with two or more stages, as shown here. The capacitors charge in parallel, but discharge in series.

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

Designed a Brand New Device?

Learn to search online databases to see if your device is cutting edge and may be eligible for a patent by using this new instructional DVD!

MIDWEST PATENT SERVICES

info@midwestpatentservices.com

www.midwestpatentservices.com

by Thomas Mitchell (USA)

Getting Started with Microprogramming (Part 1)

The Architecture, Programmer Model, and SMP

You can build a custom microprogrammed processor suitable for an FPGA. This article details how to define a microword, tie a microword definition to actual signals in the FPGA, write a microprogram, and merge a microprogram with a VHDL design.

In the 1980s, Intel sued Nippon Electric Corporation (NEC) for violating its copyrights on microcode used in the 8086 and 8088. NEC's competing processors, the V20 and V30, were alleged to have copied portions of the Intel microcode. Because of a conflict of interest, the first judge for the case was replaced by a new judge, Federal District Judge William Gray. Unfamiliar with computer terminology, Gray became impatient with the highly technical arguments of the Intel and NEC lawyers. In response, the companies gave him a two-day crash course in microprogramming, after which he ruled in NEC's favor because Intel had not properly copyrighted the microcode. While microprogramming can be a difficult skill to master, take heart. Consider that a judge with no engineering experience was able to comprehend complicated issues about microprogramming after a two-day course.[1]

Previously, I wrote on the subject of implementing microprogrammed machines in field programmable gate arrays (FPGAs). In my 2009 *Circuit Cellar* article, "Building Microprogrammed Machines in FPGAs," I gave a general description of microprogramming and a detailed description of a specific microprogram sequencer, the Advanced Micro Devices Am2910.[2] In this article series, I'll describe the process of implementing microprogrammed systems in sufficient detail so that anyone with a reasonable background in digital design should be able to design their own microprogrammed-based system.

In order to describe this process, I'll introduce another classic bit-slice device, the Am2901 4-bit register arithmetic logic unit (ALU). Using the Am2910 for the control path and the Am2901 for the data path, I will show how to build a simple 16-bit central processing unit (CPU) using microprogramming. On the way, I will show how to define the microword, how to tie the microword definition to actual signals in the FPGA, how to write the microprogram, and how to merge the microprogram with the VHDL design.

SIMPLE MICROPROGRAMMED PROCESSOR

The simple microprogrammed processor (SMP) is a 16-bit processor with eight 16-bit, general-purpose registers. It supports up to 65,536 words (64K × 16 bits) of memory for instructions or data, and it can also address up to 65,536 I/O locations. The SMP supports register direct, register indirect, and immediate memory access for almost all instructions. The 27 instructions can be grouped as: load and store, arithmetic, logical, shift, branch, and support. To simplify the design, all instructions operate on 16-bit operands and all memory and I/O accesses are 16-bit, as well.

Figure 1 shows the SMP block diagram. This diagram is helpful for visualizing the control and data paths, but it is important in determining the signals in the microword. The control path consists of the Am2910, the control store and pipeline register, the test input multiplexer, and the instruction register map multiplexer. The data path consists of the Am2901, the shift control, the status register, the interrupt register, the interrupt enable register, the instruction register, and the memory and I/O registers.

The function of the control path is to use state information and test inputs to determine the next state, and, therefore, the new control outputs. This is identical to the function of a random logic finite state machine (FSM). However, in a microprogrammed state machine, the control store is both the next-state logic and the output logic, and the outputs are all registered in the pipeline register. This feature makes a microprogrammed machine generally more efficient than a random logic FSM because the outputs are stable after

a clock-to-output-stable delay. In Mealy or Moore machines, the outputs are generated from the current state (and from the inputs, in the case of a Mealy FSM) by random logic, thus introducing additional combinatorial delay and the possibility of glitching.

The SMP control path further differs from the classic model of an FSM because of the instruction register map multiplexer. To explain why, first consider that the Am2910 has an instruction called "Jump Map." What this does is to take an external input and use this for the next microprogram address. What makes it powerful is that the external input is determined by the value in the instruction register. The map in the instruction register map multiplexer is a look-up table, addressed by a portion of the instruction value, which generates an address to a location in the microprogram. For example, say we want an instruction to add two numbers, then the value in the instruction register will point to a microprogram address stored in the instruction map. This microprogram address is the location of a piece of microcode that adds two numbers. The actual instruction register map multiplexer is a bit more complex than that. It has three maps, two for jumping to microcode that implements the different address modes, and the third for jumping to specific microcode for each instruction.

The core component of the data path is the Am2901 register ALU. Its function is to implement the registers of the processor and execute the arithmetic, logical, and shift functions (with the shift control logic) of the various processor instructions. The other components of the data path are the interrupt logic, the status register, the instruction register, and the memory and I/O registers. The function of the interrupt logic is to implement a maskable interrupt using the interrupt register, which synchronizes the external interrupt, and the interrupt mask register, which enables the interrupt input to the test input multiplexer. The status register holds the status of arithmetic and shift operations from the Am2901. The instruction register holds the currently executing instruction. The instruction register map multiplexer uses certain portions of the instruction word to jump to different sections of microcode as appropriate. The value of the instruction register is also used as alternate test select inputs (for the conditional jump instruction) to the test input multiplexer, and as alternate A and B address inputs to the Am2901. The latter is used to allow instructions to determine which registers are used for source and destination operands.

Not shown in Figure 1 are several multiplexers that determine the Am2910 D input, the Am2901 D input, the Am2901 CI input, the Am2901 A and B address inputs, and the test input multiplexer select input. Further, the control signals such as multiplexer select inputs, register load inputs, and

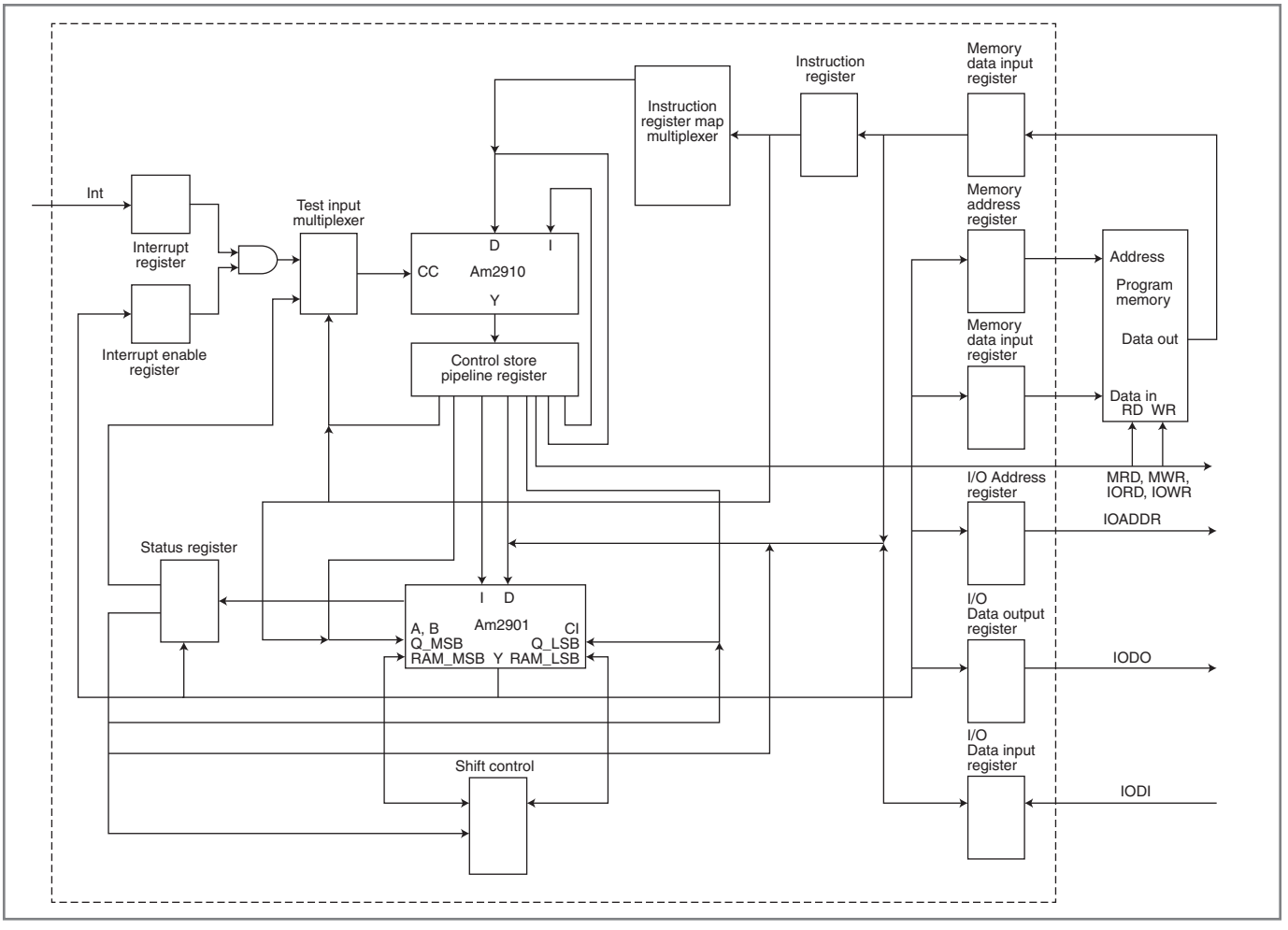


Figure 1—The components of the simple micoprogrammed processor (SMP)

instruction inputs for the Am2910 and Am2901 are not shown. All of these signals will be accounted for in the microword definition.

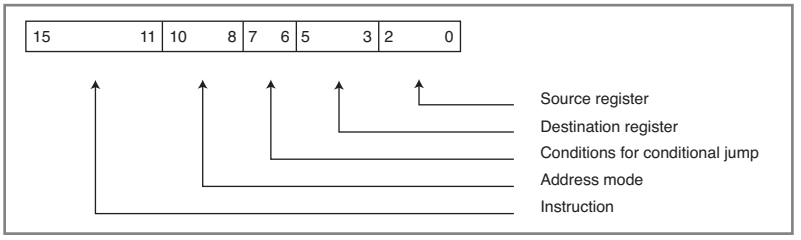


Figure 2—The structure of the SMP instruction word

SMP PROGRAMMER MODEL

The programmer model is the view of the hardware that the programmer "sees." I have already mentioned that the SMP has eight 16-bit registers, 27 instructions, and that all operations and addresses are 16-bit. The SMP also has a program counter (PC), a status register (SR), a stack pointer (SP), and an interrupt vector register (IV). The function of the microprogram is to implement the programmer model on the SMP architecture. Top-down design purists might argue that the design of the programmer model should have preceded the design of the SMP architecture, but my experience has been that nontrivial designs cannot be exactly specified without some consideration of the implementation. Besides, when I learned microprogramming, many of the design examples were processors, so I have had the general form of the architecture of the SMP in mind for many years.

Once I had the register layout of the programmer model developed, I needed to determine which instructions the SMP would support and the exact values for opcodes and operands. The latter is important for implementing the maps in the instruction register map multiplexer, but I will discuss that later. The first step in determining the instructions is designing the layout of the instruction word. I decided, for simplicity, that I would use fixed fields: the instruction type, the address mode, the condition for conditional branching, and the source and destination registers. I used fixed fields because had I used arbitrary values for opcode and operand values, I would have had to use a 64-K × 12-bit look-up table to map instructions to entry points in the microprogram. This is clearly impractical and an inefficient use of resources. With the fixed fields, I only need a 32 × 12-bit look-up table for the instruction map, and two 8 × 12-bit look-up tables for the two address mode maps. Figure 2 shows the layout of the instruction word. I allocated 5 bits for the instruction type, 3 bits for address mode, 2 bits for conditional jumps, and 3 bits each for source and destination operands. Table 1 shows the 27 instructions, their instruction word values, the mnemonic, and a description of the operation. Table 2 shows the address modes, their instruction word values, and how the destination and source operands are used. Finally, Table 3 shows the conditions for

the conditional jump instruction (CJMP).

THE AM2901

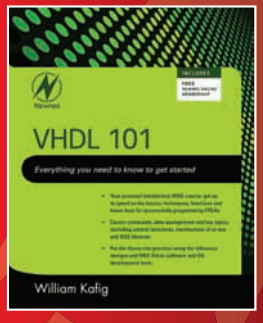
The Am2901 is a complicated component, easily more complicated than the Am2910. The hardware version of the Am2901 consistes of a 4-bit ALU with sixteen 4-bit registers. The Am2901 has a 9-bit instruction input that controls the sources of the operands, the ALU function, and the destination of the result. The Am2901 has four status signals: result is zero, most significant bit of the result (sign), overflow, and carry out. The Am2901 has generate and propagate carry signals for fast-carry look-ahead, but these signals are largely obsolete because modern FPGAs incorporate a "carry chain" that efficiently handles carries for counters and adders. In the HDL version, I made some changes to the original Am2901. First, I made the width of the "slice" variable so that custom width ALUs can be easily created. Second, in the original Am2901, the shift signals (Q0, RAM0, Q3, and RAM3) are bidirectional. In the HDL version, each shift signal becomes a signal "triple" (i.e., an input, an output, and a direction output). Because the slice width is variable, the shift triples are renamed: Q_LSB, Q_MSB, RAM_LSB, and RAM_MSB. The third change is subtle and involves a latch for the register RAM output. In

I15..I11	Mnemonic	Function
00000	LD <DST>, <SRC>	<DST> <= <SRC>
00001	ADD <DST>, <SRC>	<DST> <= <DST> + <SRC>
00010	ADDC <DST>, <SRC>	<DST> <= <DST> + <SRC> + Carry
00011	INC <DST>	<DST> <= <DST> + 1
00100	SUB <DST>, <SRC>	<DST> <= <DST> - <SRC>
00101	SUBC <DST>, <SRC>	<DST> <= <DST> - <SRC> - Carry
00110	DEC <DST>	<DST> <= <DST> - 1
00111	AND <DST>, <SRC>	<DST> <= <DST> AND <SRC>
01000	OR <DST>, <SRC>	<DST> <= <DST> OR <SRC>
01001	XOR <DST>, <SRC>	<DST> <= <DST> XOR <SRC>
01010	CMP <DST>, <SRC>	<DST> - <SRC>
01011	SRL <DST>	<DST> <= {<DST> >> 1 && 0x7FFF}
01100	SLL <DST>	<DST> <= {<DST> << 1 && 0xFFFE}
01101	SRA <DST>	<DST> <= {Carry << 15 || <DST> >> 1}
01110	SLA <DST>	<DST> <= {<DST> << 1 || Carry}
01111	IN <DST>, <ADDR>	<DST> <= IO[<ADDR>]
10000	OUT <ADDR>, <SRC>	IO[<ADDR>] <= <SRC>
10001	LDSP <SRC>	SP <= <SRC>
10010	LDIV <SRC>	IV <= <SRC>
10011	STSP <DST>	<DST> <= SP
10100	DI	IE <= 0
10101	EI	IE <= 1
10110	JMP <ADDR>	PC <= <ADDR>
10111	CJMP <ADDR>, <COND>	PC <= (<COND> : <ADDR> : PC)
11000	CALL <ADDR>	PC <= <ADDR>, MEM[SP--] <= PC + 1
11001	RET	PC <= MEM[++SP]
11010	RETI	PC <= MEM[++SP], SR <= MEM[++SP]

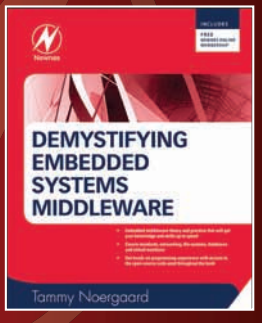
Table 1—The 27 instructions, their instruction word values, the mnemonic, and a description of the operation

Newnes Press

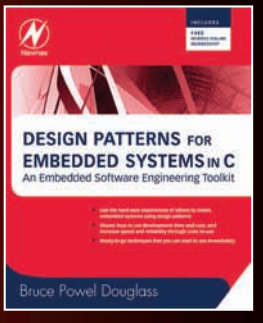
Education Never Ends. Everything you need to know to get started.



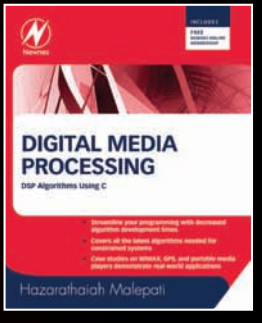
By: William Kafig
ISBN: 9781856177047



By: Tammy Noergaard
ISBN: 9780750684552



By: Bruce Douglass
ISBN: 9781856177078



By: Hazarathaiah Malepati
ISBN: 9781856176781

Save 20% on all titles when you purchase from www.newnespress.com.
Enter promotional code Newnes20 at checkout.

Newnes

Find us on Facebook!

Register for our e-news at newnespress.com
Receive our best discounts · Hear about books before they publish
Access to free sample chapters, video tutorials and more!

I10..I8	Destination	Source
000	Register	Register
001	Register	Indirect
010	Register	Immediate
011	Register	*
100	Indirect	Register
101	Indirect	Indirect
110	Indirect	Immediate
111	Indirect	*

Table 2—The address modes, their instruction word values, and how the destination and source operands are used

the original Am2901, the purpose of the latch was to hold the output of the register RAM stable while the value from the ALU operation was written to the RAM. This latch is unnecessary in the FPGA version because the RAM used will hold the output stable until the new data is written at the next clock edge.

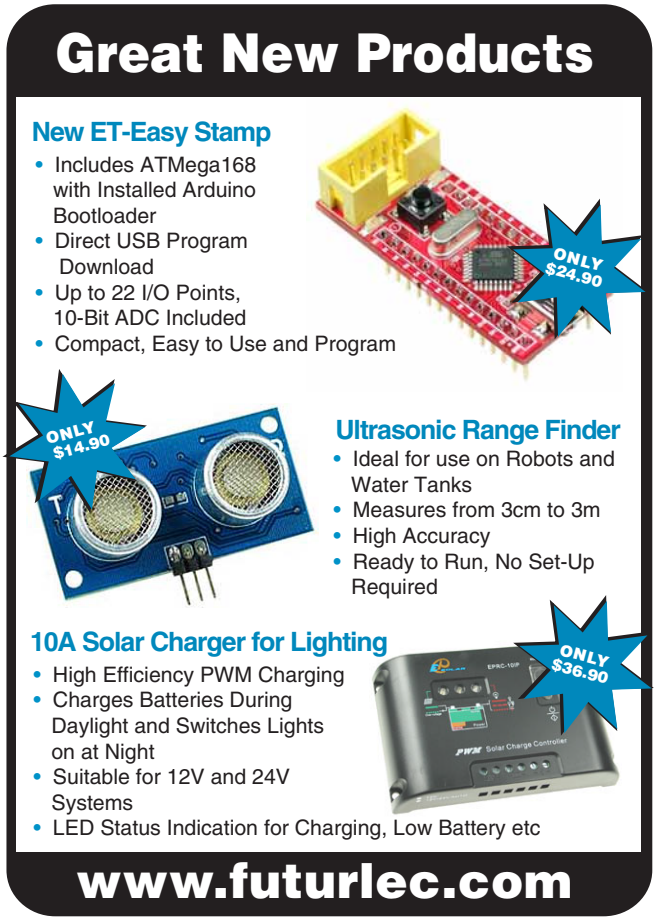
In Figure 3, you can see that the Am2901 is composed of a RAM shifter (for shifting values before writing them to RAM), the RAM that implements the registers, the ALU data source selector that controls the inputs to the ALU, the ALU, the Q shifter (like the RAM shifter but for the Q register), the Q register, and the output data selector. Notice that while the RAM has a single data input, it has two independent data outputs. The A and B address inputs are used to simultaneously access independent locations in the RAM. The two RAM outputs can be used as inputs to the ALU; but when writing the RAM, the B address is used.

HOW IT WORKS

The general operation of a processor is to fetch an instruction, decode it, execute the instruction, and then do it all again. If the processor supports interrupts, then there will also be a step to check whether an interrupt is active and to execute some operation to handle the interrupt.

Figure 4 shows the SMP instruction fetch and execution flow. The flowchart shows the start-up and interrupt decision, but what are the blocks with multiple branches? These are microprogram branches that use the Am2910 JMAP instruction. JMAP is the Am2910 instruction that uses the direct input as the next microprogram address. This direct input comes from the instruction register map multiplexer and is used to direct the microprogram sequencer to some instruction-specific part of the microprogram. There are three maps in the instruction register map multiplexer. They are the pre-operation, instruction, and post-operation maps. The function of the pre-operation map is to direct execution to functions that handle the eight possible address modes before the instruction is executed. Likewise, the function of the post-operation map is to jump to functions that handle the eight possible address modes after the instruction is executed. Why is this necessary?

The SMP can support register direct, register indirect, and immediate source operands, and register direct and register indirect destination operands. What this means is that



I7..I6	Conditions
00	Zero
01	Carry
10	Sign
11	Overflow

Table 3—The conditions for the conditional jump instruction (CJMP)

any two operand instructions need to support six different address modes (see Table 2), and a single operand instruction needs to support two or three modes. Rather than write multiple versions of instructions that do essentially the same function, I wrote one function for every instruction, but they use predefined registers for operands. For two-operand instructions, I use the source register RSRC (Am2901 register 12) for the source operand and the Am2901 Q register for the destination operand. Single-operand instructions use either RSRC or the Q register. Branch instructions use RSRC for the branch address. The I/O input instruction (IN) uses RSRC as the I/O address and the Q register as the destination operand. The I/O output instruction (OUT) uses the Q register as the I/O address and RSRC as the source operand. A few instructions (EI, DI, RET, and RETI) use no operands.

The function of the pre-operation is to load RSRC with the register, memory, or immediate value, specified by the source register, and address mode fields of the instruction. The Q register is loaded with the register or memory location, specified by the destination register, and address mode fields of the instruction. The function of the post-operation is to store the resulting value of the instruction—which is in the Q register—in the register or memory location, specified by the destination register and address mode fields of the instruction.

Each of the 27 instructions is implemented independently with the exception of the CJMP instruction, which uses the jump (JMP) instruction microcode if the branch condition is true. The status register is updated by arithmetic, load and store, and logical instructions. Shift instructions can use or modify the carry bit of the status. Because of the pre- and post-operation functions, most of the instructions are implemented with only one microword.

If an interrupt occurs and is not

SEALEVEL®

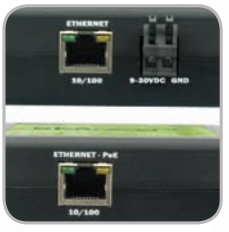
Computing/HMI | Serial | I/O

Compact Ethernet data acquisition modules support PoE.

Easily connect I/O to your network with Sealevel's new eI/O™ family of Ethernet data acquisition modules. eI/O provides system designers with a compact, cost-effective solution for monitoring and control with optically isolated inputs and Reed or Form C relays.

eI/O Solutions Offer:

- Ethernet Connected I/O
- DC or PoE Powered
- Variety of I/O Configurations
- Field Removable Terminal Blocks
- DIN Rail or Wall Mount Design



DC or PoE Powered

Power over Ethernet (PoE 802.3af) devices receive power and data on one cable and eliminate the need for additional power supplies. Ready for DIN rail mounting, eI/O modules include a removable plastic clip that snaps onto 35mm DIN rail.

sealevel.com > sales@sealevel.com > 864.843.4343

Learn more about eI/O Data Acquisition Modules at **sealevel.com/cir/eio** or scan this QR code with your smart phone.

© 1986-2011, Sealevel Systems, Inc. All rights reserved.

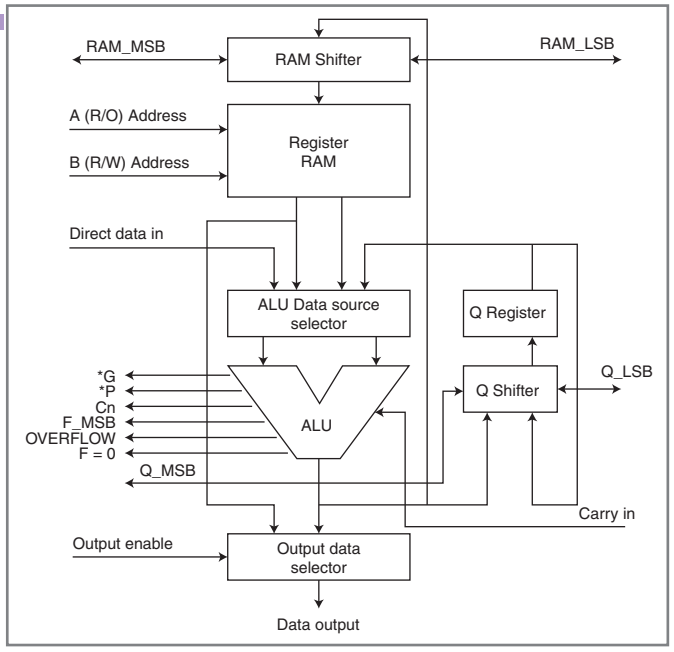


Figure 3—This is the block diagram of the Am2901 register ALU.

masked, then the SMP executes the interrupt microcode before executing the next instruction. The interrupt microcode stores the next program counter value on the stack and decrements the stack pointer. It then stores the current status register value on the stack and decrements the stack pointer again. Lastly, the program counter is loaded with the interrupt vector and the microprogram jumps to the fetch section of the microprogram. The RETI instruction does the reverse. The status register and program counter are restored from the stack and execution continues normally.

THE MICROWORD

Now that the programmer model and the SMP architecture have been determined, we can start to implement the microprogram. The first step in writing any microprogram is determining the microword's layout. I mentioned before that the SMP block diagram (see Figure 1) is the model for determining the microword, but I also said Figure 1 omitted certain details. The detailed version of the block diagram is available along with the source code. I mention this because some of the signals I am about to introduce may not be obvious from Figure 1.

The microword is typically divided into "sections" of related signals. Grouping related signals into sections makes the microcode easier to read and write. Table 4 shows the sections of the SMP microword in the column headed "Field Name." These sections, from top to bottom, are the Am2910 section, the Am2901 section, the register load (REGLD) section, the multiplexers (MUX) section, the read/write enable (RWEN) section, the shift control (SHIFT) section, the instruction map multiplexer (IMAP) section, and the processor state (PSTATE) section.

Table 4 shows the complete microword and the corresponding signals used in the VHDL source. The Am2910 section is 24 bits and is further subdivided into four subsections: Am2910 micro-instruction (4 bits); Am2910/Am2901 direct input (16 bits); test input control—that is, test input polarity (1 bit); and test input select (3 bits). The Am2901 section is 18 bits and is subdivided into six subsections: ALU destination (3 bits), ALU function (3 bits), ALU source (3 bits), A address (4 bits), B address (4 bits), and carry in (1 bit). The register load section is 9 bits, and each bit is a load-enable signal for a different register. The multiplexer section is 5 bits, where 3 bits control three 2-to-1 multiplexers, and the remaining two bits control a 4-to-1 multiplexer. The read/write enable section is 4 bits, which are the read and write enables for memory and I/O operations. The shift control section is 2 bits and controls what is shifted into the LSB and MSB of the RAM and Q shifters. The instruction map multiplexer section is 2 bits and controls the 3-to-1 multiplexer in the instruction register map multiplexer. The processor state section is 6 bits and is used to track the various operations of the processor, such as which instruction is executing. The processor state bits were originally added to fill out the microword, but during testing they were so useful that I left them in.

If you've been keeping count, you may have noticed that the width of the microword is 72 bits. Why 72 bits? The reason is that the target device is a Spartan-3 FPGA (specifically the XC3S200-4FT256), which has configurable block RAMs—that is, they can assume various widths and depths as long as the maximum number of bits (18 Kb) is not exceeded. The widest configuration is 512 bits deep by 36 bits wide. The Spartan-3 block RAMs can be as deep as 16,384 (with a width of 1 bit),

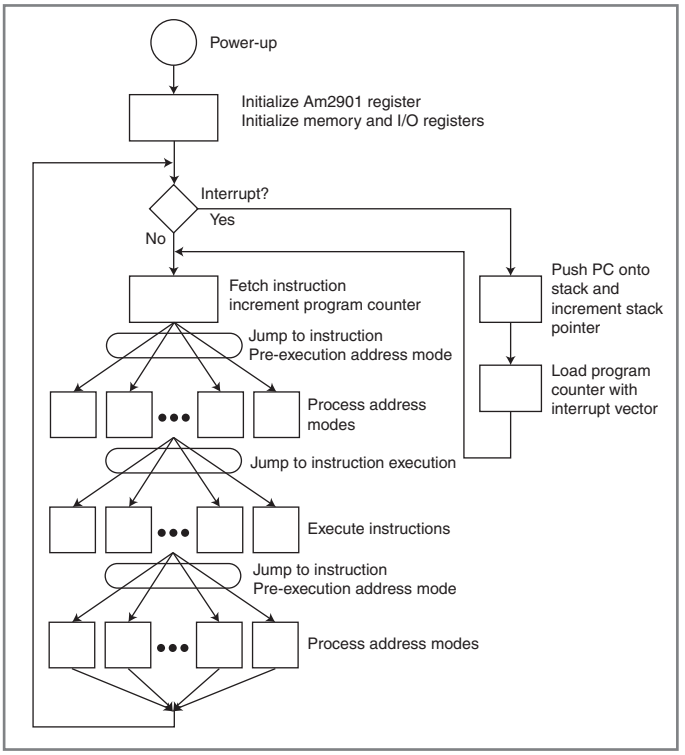


Figure 4—Instruction fetch and execution flow

Logical microword bit	Field name	Field description	Signal description	VHDL Signal name
71..68	AM2910	Am2910 instruction	Am2910 instruction bits 3..0	am2910_I(3 downto 0)
67..64		Am2901 direct input	Am2901 D bits 15..12	CS_am2901_D(15 downto 12)
63..52		Am2910/Am2901 direct input	Am2910/Am2901 D bits 11..0	CS_am2910_D(11 downto 0) CS_am2901_D(11 downto 0)
51		Test input control	Test input polarity	CS_Test_Pol
50..48			Test input select bits 2..0	CS_Test_Select(2 downto 0)
47..45	AM2901	ALU destination	Am2901 instruction bits 8..6	am2901_I(8 downto 6)
44..42		ALU function	Am2901 instruction bits 5..3	am2901_I(5 downto 3)
41..39		ALU source	Am2901 instruction bits 2..0	am2901_I(2 downto 0)
38..35		Am2901 A address	Am2901 A address bits 3..0	CS_am2901_A(3 downto 0)
34..31		Am2901 B address	Am2901 A address bits 3..0	CS_am2901_B(3 downto 0)
30			Am2901 carry In	CS_am2901_CI
29	REGLD	Register load	Memory data input register load	MDIR_Load
28			Memory data output register load	MDOR_Load
27			Memory address register load	MAR_Load
26			I/O data input register load	IODIR_Load
25			I/O data output register load	IODOR_Load
24			I/O address register load	IOAR_Load
23			Instruction register load	IR_Load
22			Status register load	SR_Load
21			Interrupt enable load	IE_Load
20	MUX	Multiplexer controls	Am2901 carry in Mux select	CI_Mux_Select
19			Am2901 A address Mux select	A_Mux_Select
18			Am2901 B address Mux select	B_Mux_Select
17			Test source Mux select	Test_Source_Mux_Select
16..15		Am2901 D Mux select	Am2901 D Mux select bits 1..0	am2901_D_Mux_Select(1 downto 0)
14	RWEN	Read/write enable	Memory read enable	MRD
13			Memory write enable	MWR
12			I/O read enable	IORD
11			I/O write enable	IOWR
10..8	SHIFT	Shift control	Shift control bits 2..0	Shift_Select(2 downto 0)
7..6	IMAP	Instruction map select	Instruction map select bits 1..0	IR_Map_Mux_Select(1 downto 0)
5..0	PSTATE	Processor state	Processor state bits 5..0	Pstate(5 downto 0)

Table 4—The complete microword and the corresponding signals used in the VHDL source

but I did not expect my microprogram to exceed 512 microwords, so width was more important than depth. Therefore, the SMP uses two 512 × 36-bit block RAMs.

We have covered a lot of ground. I worked out the architecture, programmer model, instruction set, and general operation of the simple microprogrammed processor. Along the way, I developed the Am2901 HDL model, which is the core of the processor. So far, I've only hinted at the FPGA and microcode implementation. In the next part of this series, I'll cover this topic, as well as the process of building a system using the SMP. See you then.

Thomas Mitchell (thmitche@gmail.com) is a registered professional engineer who has worked for the Department of Defense for the last 31 years. He graduated from the University of Delaware with Bachelor's degrees in Electrical Engineering and Physics. Thomas later received Master's degrees in Electrical Engineering and Applied Physics from The Johns Hopkins University. He has worked on numerous high-speed digital designs of components, boards, and systems. Thomas has implemented designs with ECL, TTL, and CMOS using discrete logic (SSI/MSI/LSI/VLSI), programmable logic (PALs, complex PLDs, and FPGAs), microprogram sequencers, and microprocessors.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/249.

REFERENCES

[1] A. Barnum, "Judge Makes Case for Self-Education, Intel-NEC Dispute Required Study," *San Jose Mercury News*, February 8, 1989.

[2] T. Mitchell, "Building Microprogrammed Machines with FPGAs," *Circuit Cellar* 233, 2009.

SOURCES

Am2910 Sequencer and Am2901 ALU
Advanced Micro Devices | www.amd.com

Spartan-3 FPGA
Xilinx | www.xilinx.com

by Mariano Lizarraga, Renwick Curry, and Gabriel Elkaim (USA)

Reprogrammable UAV Autopilot System (Part 1)

System Hardware and Software

This two-part article series covers the design, development, and testing of a reprogrammable UAV autopilot system. Here you get a detailed system-level description of the autopilot design, with specific emphasis on its hardware and software.

Unmanned aerial vehicle (UAV) usage has increased tremendously in recent years. Although this growth has been fueled mainly by demand from government defense agencies, UAVs are now being used for non-military endeavors as well. Today, UAVs are employed for purposes ranging from wildlife tracking to forest fire monitoring. Advances in microelectronics technology have enabled engineers to automate such aircraft and convert them into useful remote-sensing platforms. For instance, due to sensor development in the automotive industry and elsewhere, the cost of the components required to build such systems has fallen greatly.

In this two-part article series, we'll present the design, development, and flight test results for a reprogrammable UAV autopilot system. The design is primarily focused on supporting guidance, navigation, and control (GNC) research. It facilitates a frictionless transition from software simulation to hardware-in-the-loop (HIL) simulation to flight tests, eliminating the need to write low-level source code. We can easily make, simulate, and test changes in the algorithms on the hardware before attempting flight. The hardware is primarily "programmed" using MathWorks Simulink, a block-diagram-based tool for modeling, simulating, and analyzing dynamical systems.

The complete system, called the Santa Cruz Low-cost UAV GNC System (SLUGS), was developed by the

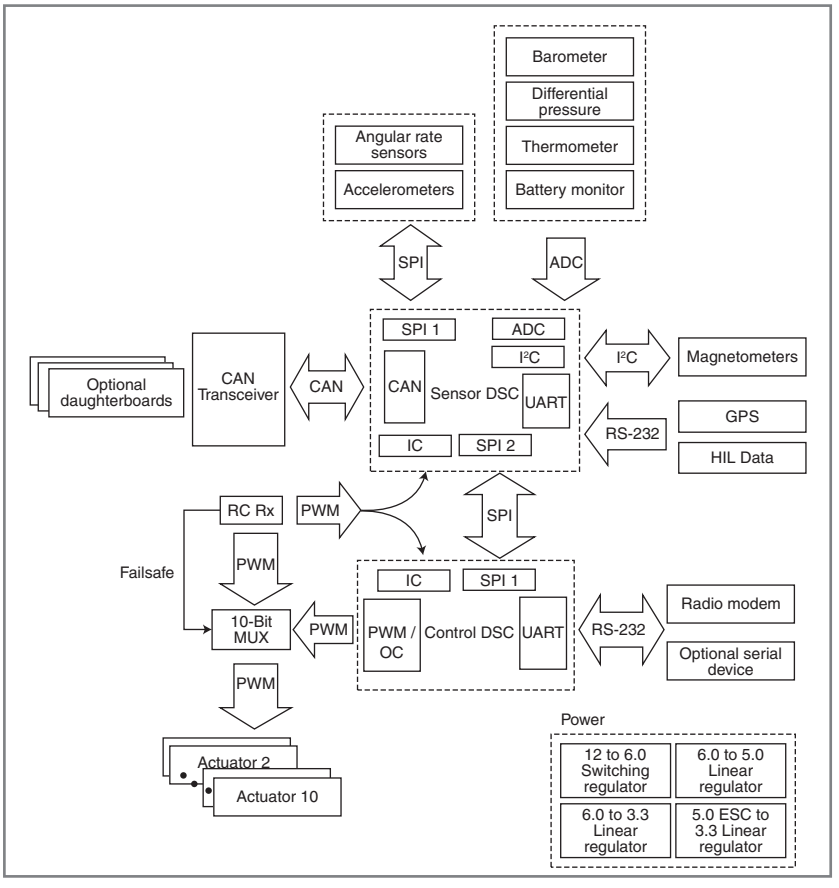


Figure 1—The autopilot hardware architecture

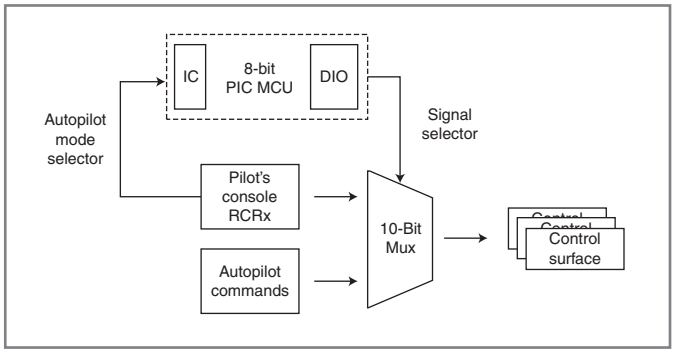


Figure 2—The multiplexor's commands

Autonomous Systems Laboratory at the University of California Santa Cruz and released under the MIT open-source license. Due to space restrictions, this article is focused exclusively on a system-level view of the autopilot, its hardware, and software. Refer to the Resources listed at the end of this article to learn more about the system's other components and the autopilot itself.

THE COMPLETE UAV SYSTEM

First, we'll provide a high-level description of the complete UAV system and how the components interrelate. On the aircraft, the autopilot takes readings from a variety of sensors and uses them to stabilize the aircraft in flight: holding altitude, maintaining velocity, tracking straight lines, or constant rate turns. It then uses these "inner loop" modes to navigate through a sequence of waypoints, either preloaded or communicated to the autopilot via an RF link from the ground station.

The autopilot has two operating modes: Autonomous and Manual. In the former mode, the autopilot directly commands the control surfaces and is fully responsible for the aircraft. In Manual mode, the autopilot relays the commands from an RC pilot directly to the control surfaces, leaving the pilot in command. (Essentially, the autopilot removes itself from the system and follows along, ready to take over if switched back to Autonomous mode.)

Controlling and monitoring the UAV is conducted from a ground control station (GCS), which receives and displays the UAV's sensor telemetry, controlling its various high-level functions. The ground equipment also includes a conventional RC transmitter for take-off, landing, and control of the aircraft in case of malfunction. All mission planning and control is exercised from the GCS, which consists of a laptop PC, a two-way RF link to the aircraft, antenna, batteries, and some level-shifting hardware to match the RF link to the PC ports.

The core component of the GCS is its software. The operator uses the GCS software to interact with the UAV and modify its mission and autopilot parameters while still airborne. The UAV sensor data is graphically displayed on the GCS, and the position of the UAV is shown in real time overlaid onto a geo-referenced map image.

The third component of the system is the HIL simulator. In this mode, the autopilot gets its sensor data from an external computer pretending to be the world (as opposed to the internal sensor suite). This enables the autopilot to operate using simulated data and thus "fly" the aircraft without ever leaving the ground and risking the airframe. The autopilot does not know that it is not flying a real UAV; the algorithms and outputs are running in real time on the actual flight hardware. This enables end users and researchers to safely test the system and verify its correct operation.

The essential component here is the autopilot. Its suite of custom avionics makes autonomous flight possible, while the hardware and software at the center control the aircraft in flight, guiding it along the desired flight path and navigating from waypoint to waypoint. This is the key difference between autonomous aircraft and remotely piloted ones.

AUTOPILOT HARDWARE

At its core, the SLUGS autopilot hardware has two Microchip Technology dsPIC33 digital signal controllers (DSCs) as its main processing units, each running at 40 MIPS. These are interconnected via SPI at 10 MHz and are called the "sensor" and "control" DSC, respectively. The sensor DSC reads the data from all of the sensors in the sensor suite, such as rate gyros, accelerometers, and static and differential pressure. The data is then fed into a complementary filter, which fuses the data and computes the aircraft's position and attitude (3-D orientation). This information is packed in a predefined communication protocol (described in the second part of this series) and sent via the SPI to the control DSC. The control DSC (based on the data received from the sensor DSC), the aircraft current location (established via GPS), and the current commands received from the GCS generate the commands for the control surfaces and schedule the data to be sent to the ground telemetry via the RF link.

The autopilot hardware has four key subsystems: the sensor DSC and its sensor suite, the control DSC, the RC commands multiplexor, and the power supply. Each of the individual integrated circuits that comprise the autopilot was selected based on three key deciding factors. The first was functionality. The component included features that reduced the required external components (e.g., capacitors

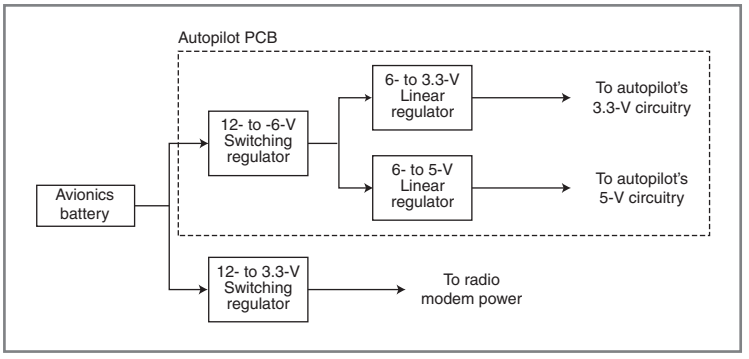


Figure 3—Avionics battery power distribution

HUGE discounts
on select RIGOL equipment
- for a limited time only!
DG1022 - 20MHz Function/Arbitrary Waveform Generator
Was $695 Now only $499
DS1052D - 50MHz Mixed-signal Scope 2ch + 16 Digital Channels
Was $1,095 Now only $899
DS1102D - 100MHz Mixed-signal Scope - 2ch +16 Digital Channels
Was $1,495 Now only $1,099
DG1022 + DS1052E
Basic Signal Analysis Package
20MHz Function/AWG + 50 MHz 2-Ch Digital Oscilloscope
Was $1,094 Now only $799
DS1102CA + DG2041A
Standalone Signal Analysis Package
40MHz Function/AWG + 100 MHz 2-Channel Digital Oscilloscope
Was $2,290 Now only $1,799
Hurry! This sale is only for a limited time!
Call us at 888-772-3544
or go to www.saelig.com today!
Love Saelig's Unique Electronics?
unique geek
Get a free "unique geek" t-shirt with your next $50 or more purchase!
(See website for details)
Saelig .com unique electronics
Follow us:

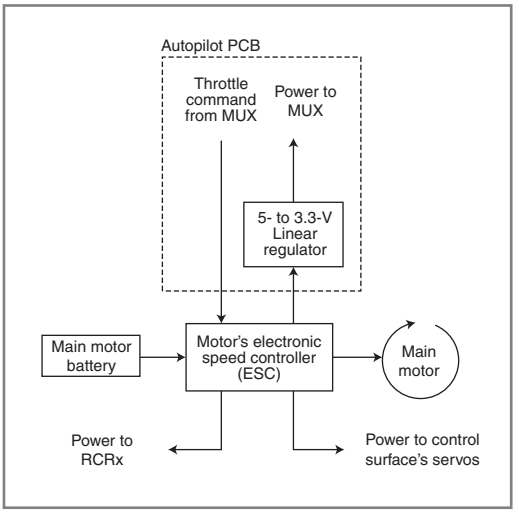


Figure 4—Main motor battery power distribution

and inductors), the required software for utilizing its data (e.g., factory calibration or temperature compensation), or the interface to communicate with the device (where digital was always preferred over analog for simplicity).

Size was the secondary key factor. Great care was taken to make the autopilot as small as possible without sacrificing functionality.

The last key factor was availability. All of the components were chosen so that they were easily available with no restriction to their purchase in countries outside the United States. The idea was to make the SLUGS platform available to the widest possible group of research laboratories.

SENSOR DSC

As its name implies, the sensor DSC reads data from all sensors available to the autopilot and performs low-level transformations (e.g., scaling, temperature compensation, null shifts, and data alignment) to generate engineering units. The position and attitude estimation algorithms then use these measurements to determine the aircraft's position and orientation at 100 Hz. This data is packed in several communication protocol sentences and delivered to the control DSC, making use of most of the peripherals available in the DSC.

The sensor suite read by the sensor DSC consists of several sensors. Refer to Figure 1 as we describe each.

One is a San Jose Navigation FV-M8 (formerly EB-85) GPS, which is used to obtain the UAV's latitude, longitude, height, speed over ground, and course over ground.

A tri-axis Honeywell HMC5843 I²C magnetometer is used in the orientation computation. Three-axis accelerometers and gyros are also included via the Analog Devices ADIS16405 AMLZ SPI tri-axis inertial sensor for the rate gyro and accelerometer readings.

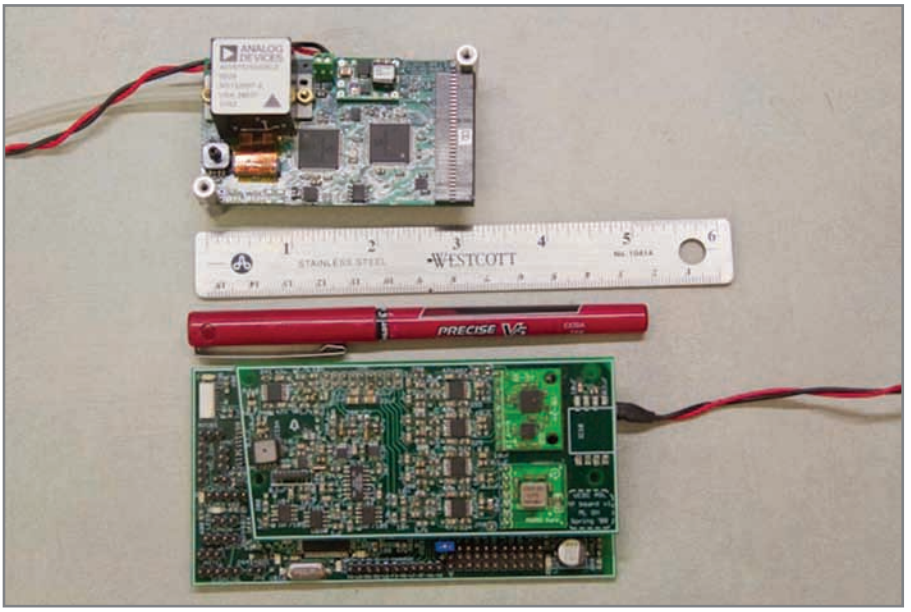


Photo 1—(Bottom) Version 1.0 of the SLUGS autopilot. It consisted of two four-layer boards stacked. (Top) Version 2.0 is a single four-layer board in a 3.25" x 1.75" board.

Faster CPU and More Memory?
The CUWIN5000 Now Shipping!

CUWIN

Need more sophisticated features in a touch screen?
Enter the CUWIN? A Revolutionary Touch Controller.

Programmable in C++ or any .Net Language (C#, VB.Net, etc...) for the creation of highly sophisticated applications involving data acquisition, machine control, sensor monitoring, and even web reporting, just to name a few.

Cheaper than a PC, and more features than an HMI touch screen.
The CUWIN? A Revolutionary Touch Controller
Already in use in thousands of applications throughout the world.

▶ CUWIN5200 ▶ CUWIN5500 ▶ CUWIN6300

- Samsung 2450 (533MHz) ARM Processor
- 128MB SDRAM / 128MB NAND Flash
- Windows® Embedded CE 6.0
- .Net Compact Framework 3.5
- Software Development: C++ or any .Net Language (C#, VB.Net, etc...)
- Ethernet 10/100Base-TX
- RS-232 x 3, RS-485 x 1
- USB 2.0 x 1, USB 1.1 x 1
- Audio (Line Out) / RTC
- Keyboard and Mouse (optional)
- CE, FCC, and KCC certified

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

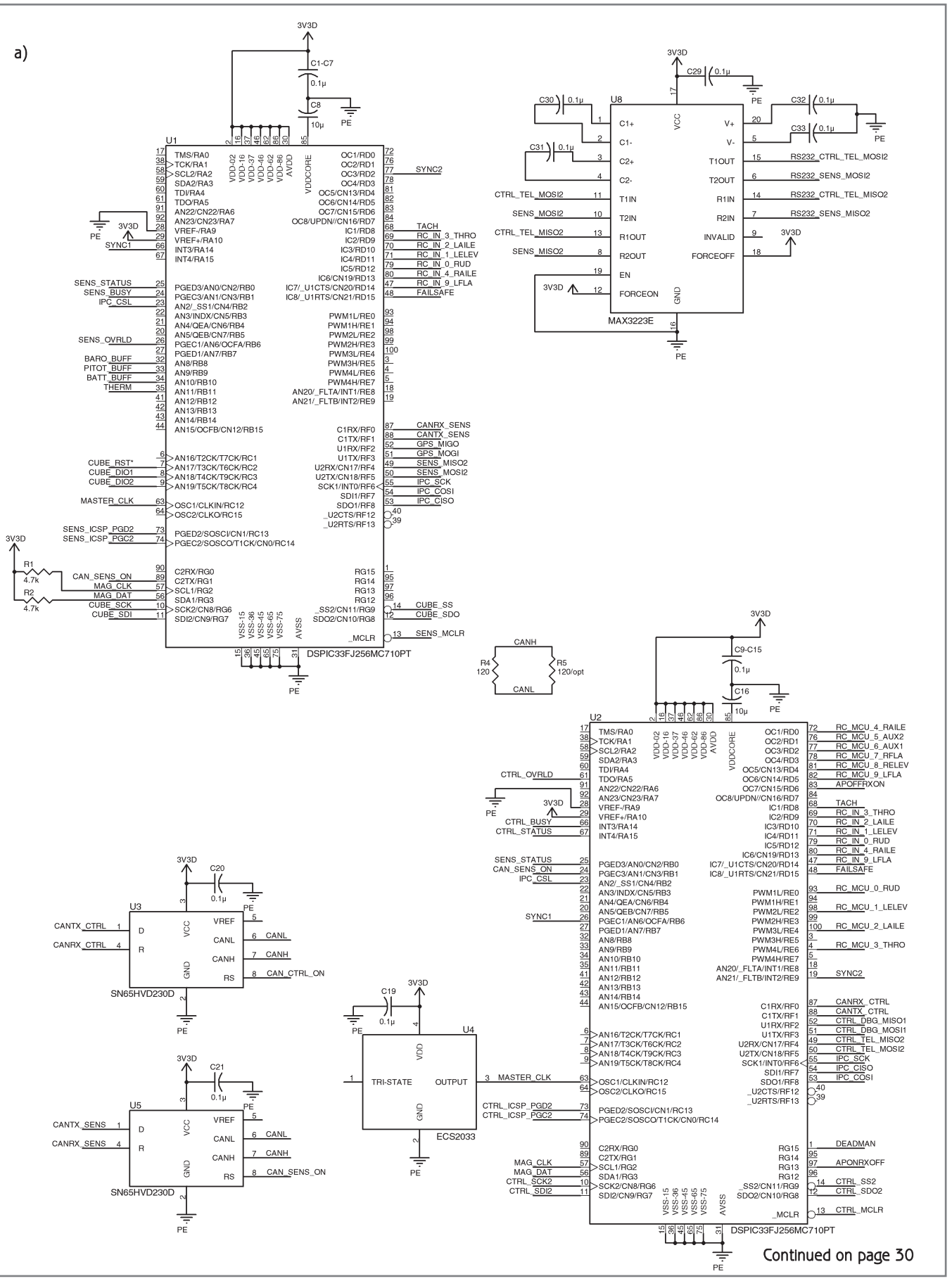


Figure 5a—The autopilot circuitry includes sensor and control DSCs. The rest of this schematic is on page 30.

Cortex-M0 a simple choice

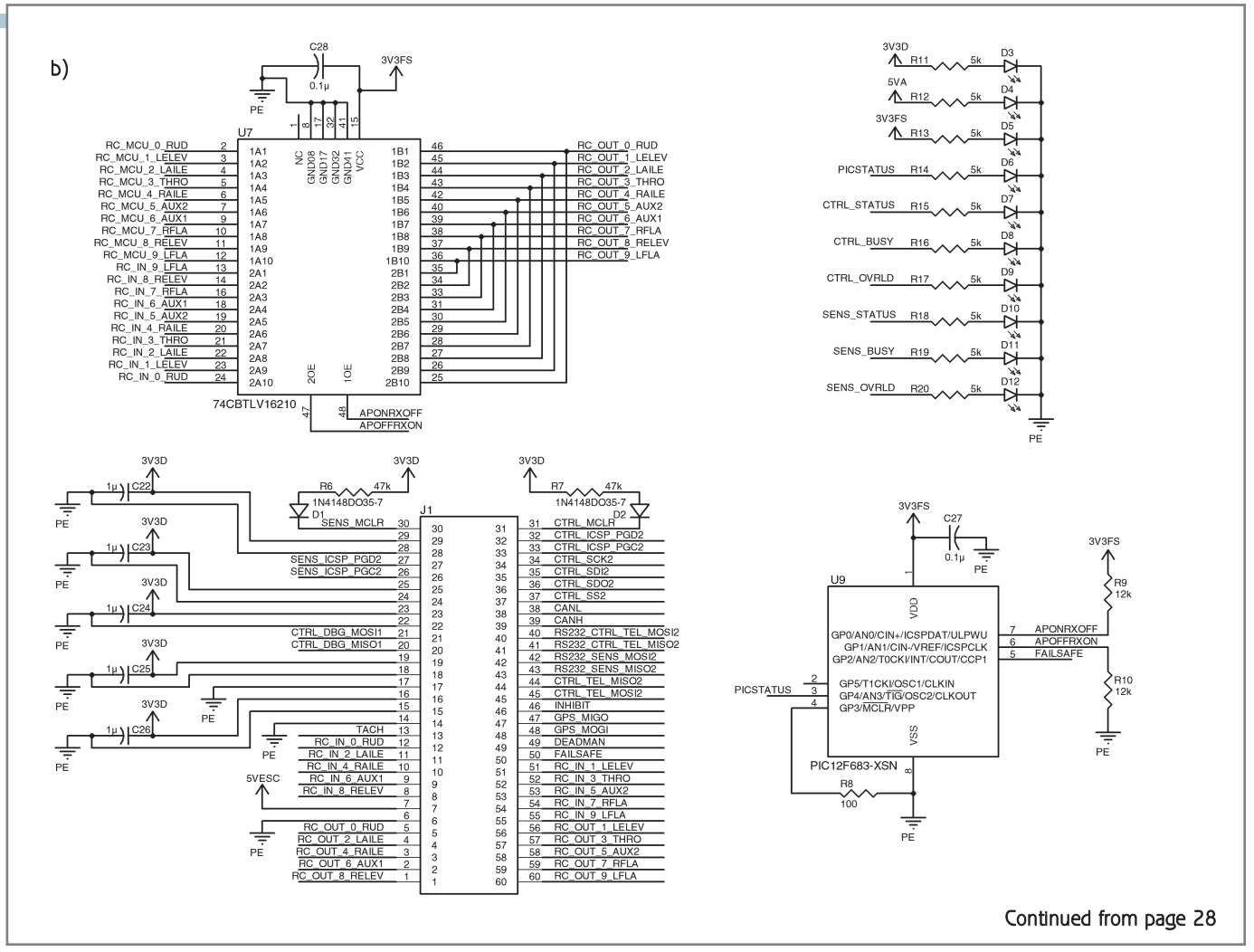
We're changing how engineers think about design, with Cortex-M0™ solutions that let you save power, reduce cost, shrink your design, and get to market fast.

- **Lowest active power** — as low as 130µA/MHz
- **Superior Code Density** — 50% less code for most tasks
- **Higher performance** — LCP1100 runs at over 45 DMIPS
- **Smallest size** — the LPC1102 has a footprint of 5 mm^2
- **Low-cost toolchain** — LPCXpresso for less than USD 30

Show us how you switched from your old 8-/16-bit habit to the new 32-bit NXP Cortex-M0 solution and we'll send you a FREE LPCXpresso development board.

www.nxp.com/cortex-m0

NXP



Continued from page 28

Figure 5b—The RC command multiplexor, status LEDs, and I/O interface

We included a Microchip Technology MCP9700 low-power linear active thermistor for temperature compensation of the sensors. To monitor the battery voltage, we used a voltage divider. A voltage divider built with 1% resistors connected to the dsPIC's ADC module monitors the avionics battery voltage.

A Freescale Semiconductor MPXV5004DP differential pressure sensor is used to compute the UAV's airspeed. The system uses a Freescale Semiconductor MPXH6115A6U absolute pressure sensor to compute altimetry for the UAV.

The battery voltage monitor, the absolute, and the differential pressure signals are first buffered (and some basic signal conditioning is applied) by a National Semiconductor LMV844MT quad op-amp in voltage-follower configuration. Additionally, the sensor DSC employs its input capture (IC) module to record the pilot's PWM commands to the servos, its remaining UART module (the other one is used by the GPS) to receive simulated readings during a HIL simulation, and the remaining SPI module (the other is used by the ADIS16355) to communicate with the control DSC.

CONTROL DSC

The control DSC receives commands from the GCS via its UART module (see Figure 1). It also gets attitude and position information from the sensor DSC through its SPI module. With these inputs, it runs its navigation and control algorithms to produce, as its name implies, the control surface commands. These commands are translated into the appropriate PWM duty cycle and are sent out by the dsPIC's PWM motor control module to actuate the UAV's control surfaces servos. As in the sensor DSC, the control DSC records the pilot's PWM commands. These readings are used to obtain the control surface trim conditions when transitioning from manual to autonomous flight. This prevents any unwanted large input disturbances when "throwing the switch."

Additionally, the autopilot uses Maxim Integrated Products's popular MAX232E driver/interface to have telemetry available in both signal levels, direct from the UART and RS-232 for its use directly in a PC. This driver also level shifts the signals to and from the HIL simulator.

For easy expansion (to other sensors, payloads, etc.), it is possible to connect to external daughterboards via a

controller area network (CAN). This provides extensibility without the need to redesign the hardware. For this, both DSCs are connected to a properly terminated CAN bus via two Texas Instruments SN65HVD230D CAN transceivers.

RC COMMAND MULTIPLEXOR

As with any vehicle, safety is critical. Because we are dealing with an aircraft, there is always a crash risk to the aircraft itself, as well as to people on the ground. There's a fundamental safety requirement. In the case of any malfunction, the safety pilot can retake control of the aircraft from the ground through the RC transmitter. This back-up system not only has to be reliable, but also completely independent (power, signal, and radio) from the autopilot. The pilot's commands are received onboard the aircraft by an off-the-shelf radio control receiver (RCRx). This receiver produces independent PWM outputs for each channel. Five channels are used in total: four contain the pilot's control surface commands (ailerons, rudder, elevator, and throttle) and the fifth is the commanded autopilot mode selector (autonomous or manual).

The commands multiplexor architecture in Figure 2 comprises a Microchip Technology PIC12F683 8-bit MCU and a Texas Instruments SN74CBTLV16210 20-bit FET bus switch (mux). The mux sits between the autopilot and the RC servos. The MCU's IC module reads the RCRx's mode selector PWM command and configures the bus switch with two digital input/output (DIO) lines. The bus switch, based on its configuration, routes the PWM signals from either the autopilot or the RCRx to the control surfaces' servos.

POWER SUPPLY

The power supply architecture was designed to provide power to the avionics independent of the airborne-critical systems and propulsion (see Figure 1). For this purpose, two Thunder Power Lithium-Polymer (LiPo) batteries are employed. A 1,325-mAHr battery powers the avionics and radio modem. A 5,000-mAHr battery powers the airborne-critical systems. This yields approximately 25 minutes of flight on our prototype UAV.

The avionics battery output is connected to a Texas Instruments PTN78000WAZ plug-in power module (in the autopilot PCB) that drops the voltage from 12 to 6 V using a switching regulator (see Figure 3). Further down the power line, two National Semiconductor LP2986 and LP38690SD low-dropout (LDO) linear regulators lower the voltage to 5 and 3.3 V for the circuitry in the autopilot. These regulators and their accompanying filter capacitors also serve the purpose of reducing some of the noise intrinsic to the output of switching regulators. An independent PTN78000WAZ plug-in power module board drops the voltage from 12 to 3.3 V to power the UAV's radio modem for telemetry.

The main motor battery powers an off-the-shelf 60-A electronic speed controller (ESC). The ESC performs a few essential tasks (see Figure 4). One, it provides 5-V power to the RCRx. Two, it provides power to the control surfaces' servos. Three, it provides power to another LP38690SD 5-to-3.3-V LDO linear regulator (in the autopilot PCB), which, in turn,

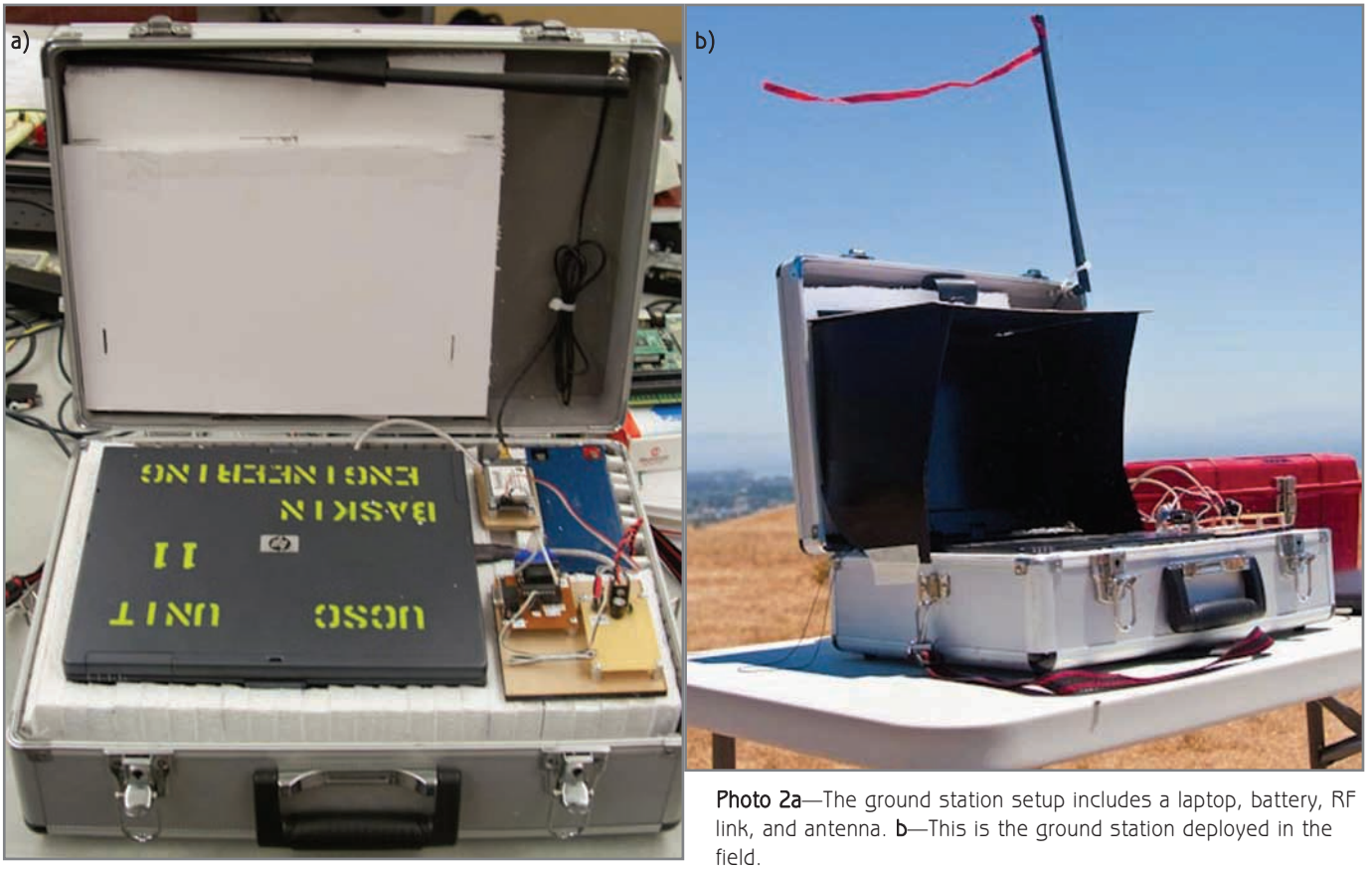


Photo 2a—The ground station setup includes a laptop, battery, RF link, and antenna. **b**—This is the ground station deployed in the field.

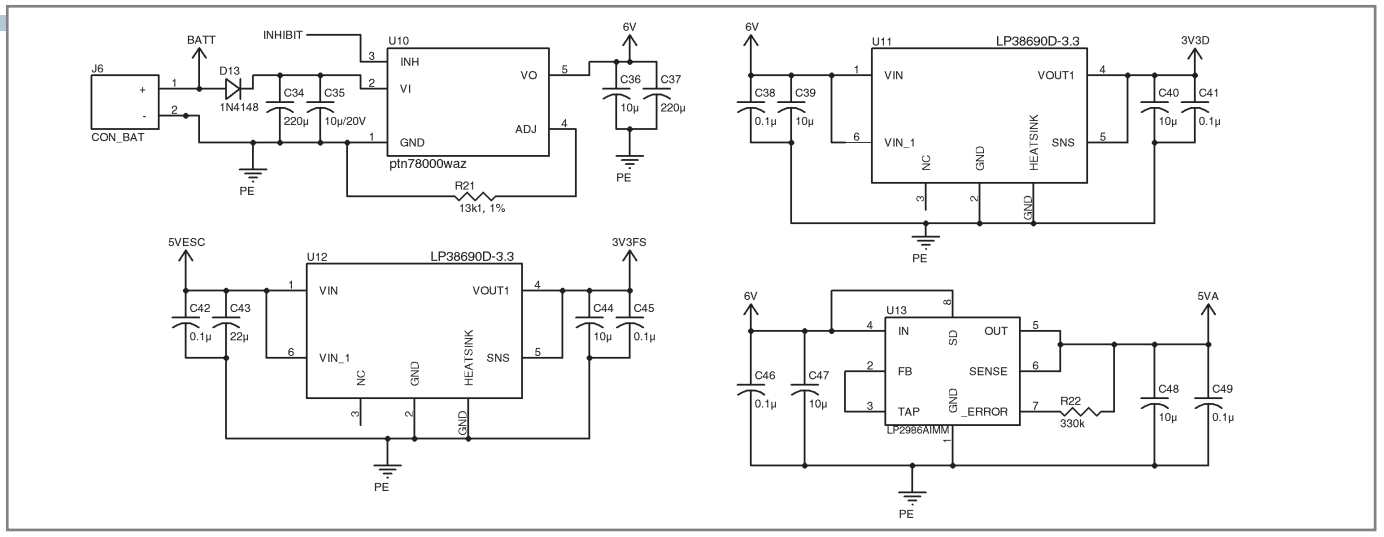


Figure 6—Here you see a Texas Instruments PTN78000WAZ power module and National Semiconductor linear regulators.

powers the commands multiplexor. And four, it receives throttle commands from the commands multiplexor and generates the necessary signals to drive the three-phase brushless DC motor at that desired setting. This architecture guarantees that in the case of an autopilot power loss or malfunction, the safety pilot can always retake control of the UAV.

The total power consumption of the autopilot and the commands multiplexor in normal operation is 3.1 W at 12 V. It measures 3.25″ × 1.75″ and weighs 50 grams when fully assembled. (This excludes the flight harness and the battery power cables.)

The first board prototype (Version 1.0), which you can see at the bottom of Photo 1, was sent for manufacture to Advanced Circuits and the components were later hand-soldered under a microscope. For the next batch of prototypes (Version 2.0, shown in Photo 1 at the top), the board size was significantly reduced, which made the component density grow. Thus, for manufacturing, we used Sierra Proto Express,

Easy Embedded Linux
OMNI-EP


• 200 Mhz ARM9 CPU
• 10/100 Mb Ethernet
• 32 MB RAM
• 16 MB Flash
• 16 Digital I/O Lines
• 2 Ports of USB 2.0
• SPI Bus
• AC97 Amplified Audio
• Battery Backed Clock
• 2 Serial Ports
• Low Power Consumption
• RoHS Compliant
Our newest ARM9 Linux controller, the OmniEP doesn't cost an arm and a leg. It delivers removable storage, amplified audio, ethernet and serial RS232 communication ports in a rugged and attractive enclosure. Models without enclosure and LCD available.
The OmniEP comes preloaded with Linux to jumpstart your development process, with LCD and pushbutton drivers supplied. Large capacity USB drives can be easily mounted in the USB port.
Call 530-297-6073 Email sales@jkmicro.com
www.jkmicro.com
JK microsystems, Inc.
International Orders Welcome
MADE IN U.S.A

AP CIRCUITS
PCB Fabrication Since 1984
As low as...
$9.95
each!
Two Boards
Two Layers
Two Masks
One Legend
Unmasked boards ship next day!
www.apcircuits.com
VISA MasterCard PayPal IPC MEMBER ASSOCIATION CONNECTING ELECTRONICS INDUSTRIES BBB

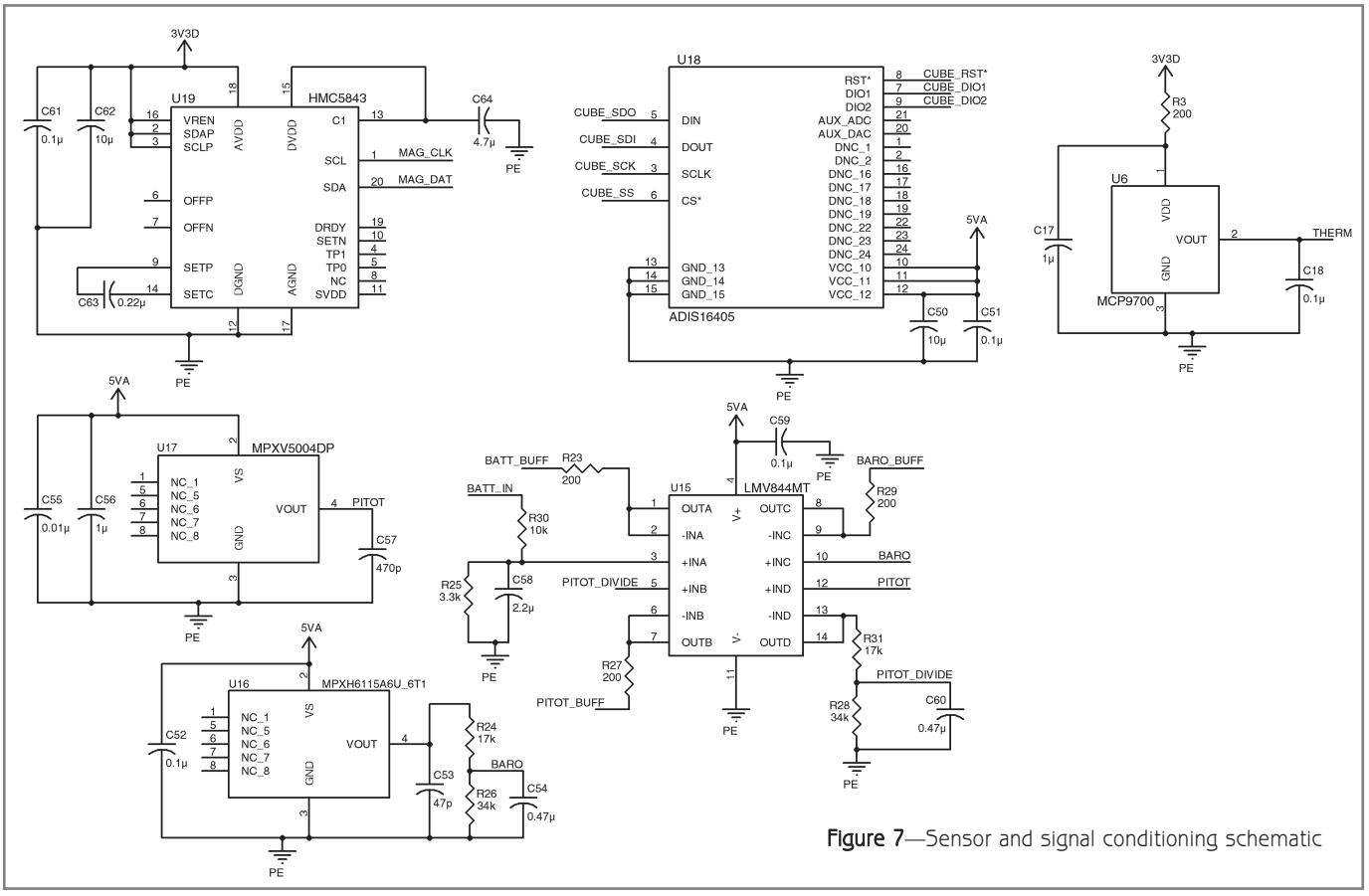


Figure 7—Sensor and signal conditioning schematic

which specializes in low-count orders. These too were hand-soldered under a microscope and used extensively during several months of flight tests. More recently, a third batch (Version 2.1) was sent for manufacture and population to Sierra Proto Express and will be used in a new UAV research project.

The SLUGS v2.x board has only two electrical connectors to the outside world, and a set of two narrow plastic hoses for static and dynamic pressure. To simplify moving the autopilot from the aircraft to the lab for testing, it has a miniature screw terminal to connect power and ground from the avionics battery and a Samtec CLM-112-02-L-D high-density 60-pin connector for the rest of the signals. Figure 5, Figure 6, and Figure 7 depict the hardware circuitry.

GROUND STATION HARDWARE

The GCS hardware consists of a conventional laptop (1.5-GHz Pentium M, 1-GB RAM, 160-GB hard drive) running Windows XP, QGroundControl, and Open Source ground station software. A USB-to-serial converter is used to connect to the radio. The GCS allows con-

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

Unlock TWO Decades of Engineering Excellence
CC Key fits on any keyring and contains the entire Circuit Cellar issue archive in searchable PDF format.
Keep it current with your Digital 2.0 subscription.
Buy now at www.cc-webshop.com

figuration and control of the UAV (which will be detailed in the second part of this article), and also allows the telemetry stream to be dumped straight to a file. A Digi International XTend 900-MHz radio module is connected to a Hyperlink half-wave whip antenna. A small custom-made board has the serial splitter, RS-232 level shifter, and power regulator for the radio modem. The entire ground station is housed in a small briefcase, including the antenna mount and a lead-acid 12-V battery (see Photo 2).

HIL SIMULATOR

In order to test the autopilot without risking the airframe, it is critical to simulate the algorithms in realistic conditions. The Simulink programming paradigm—which we'll describe in the second part of this article series—makes it easy to simulate the algorithms on the PC using MATLAB and Simulink, but this is purely a theoretical simulation. To test both the hardware and the software, it is necessary to run the algorithms on the actual hardware. Using the HIL simulator, these real-time simulations reveal flaws and untrapped errors that would destroy the airframe if they occurred in the field.

In the HIL simulator, a computer runs a physics-based simulation of the UAV (see Figure 8). After it is corrupted by noise, bias shifts, or other imperfections, the data from the simulation is pushed onto the autopilot, overwriting its own sensor data. From there, the autopilot flies the aircraft, but the generated commands are intercepted and sent back to the simulation, causing the simulation to react as if the autopilot were flying a real plane.

In our case, the physics engine is a full nonlinear, six degree of freedom (6DOF) simulation running in Simulink. The aircraft can be started in any environment and configuration, with complete control of the simulation. This means that sensors can be failed (GPS outage), gusts introduced, winds changed suddenly, and the actions of the autopilot observed. A simulation crash simply requires the simulator to be reset. While the specifics of matching the simulation to a given airframe model are beyond the scope of this article, more details can be found in the references.

The simulation generates the sensor packet and is sent via a UDP port, while the GCS program listens in the same port and interprets the data. This is then sent to the autopilot over a serial link. The GCS receives the autopilot control surface commands using the serial link, and then sends them back to the simulation via UDP. The advantage here is that a separate computer can run the simulation (processor intensive), while the GCS continues to display the results of the incoming telemetry as if the aircraft were flying. (HIL also enables the testing of the GCS.) It is very easy to plug our simulation results into standard flight simulators (FlightGear, X-plane) for visualization of the flight.

SOFTWARE DEVELOPMENT

In the next part of this article series, we'll present all of the software developed for the autopilot: the low-level drivers, a brief description of the position and attitude estimation filters, and navigation and control algorithms. We'll also present flight test results and discuss the convenience of using Simulink with SLUGS as a development platform compared to writing the code in a traditional programming environment, such as C or C++.

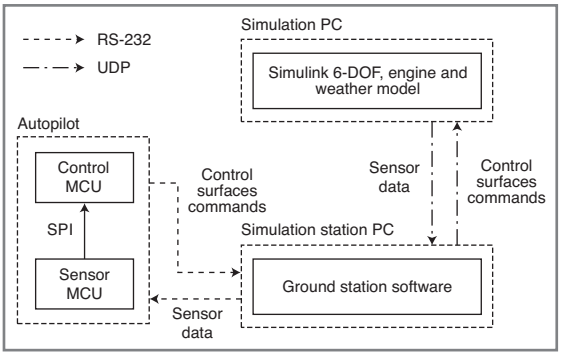


Figure 8—HIL signal flow paths

Authors' notes: We'd like to express our deepest gratitude to Greg Horn for working so hard on the board schematics and layout and hand-soldering the components of the first two batches of autopilots. He was also our safety pilot, and more than

once he saved us from having to build another UAV. We'd also like to thank Vladimir Dobrokhodov for all of his help in making SLUGS happen and lending his experience as a long-time UAV developer and end user. Finally, note that code for the SLUGS project is posted at http://github.com/malife/SLUGS.

Mariano Lizarraga (malife@soe.ucsc.edu) holds a BS in Naval Sciences Engineering from the Mexican Naval Academy, an MSEE and Electrical Engineer's degree from the U.S. Naval Postgraduate School, and a PhD in Computer Engineering from UCSC. He has more than eight years of experience writing software and designing hardware for UAVs. He is currently a Research Fellow in the Computer Engineering Department at the University of California Santa Cruz.

Renwick Curry (rcurry@ucsc.edu) earned an AB in Physics from Middlebury College, and he holds a BS, MS, and PhD in Aeronautics and Astronautics from MIT. He has worked as a faculty member (MIT and Cornell), a spacecraft guidance engineer (Mariner and Apollo projects), and an aviation safety engineer for NASA. Renwick is currently an Adjunct Professor in the Computer Engineering Department at University of California Santa Cruz, and president of AASI (www.aasi.com), a company specializing in aircraft flight-path optimization.

Gabriel Hugh Elkaim (elkaim@soe.ucsc.edu) holds a BSE in Aerospace Engineering from Princeton University, as well as an MS and PhD in Aeronautics and Astronautics from Stanford University. He is an Associate Professor in the Computer Engineering Department at University of California Santa Cruz, where he works on autonomous vehicle guidance, navigation, and control.

RESOURCES

SLUGS Project, http://slugsuav.soe.ucsc.edu

UC Santa Cruz Autonomous Systems Lab, http://asl.soe.ucsc.edu

SOURCES

ADIS16355 iSensor
Analog Devices | www.analog.com

XTend OEM 900-MHz RF module
Digi International, Inc. | www.digi.com

HMC5843 I²C Magnetometer
Honeywell International | www.honeywell.com

dsPIC33 DSC, MCP9700 Thermistor, and PIC12F683 microcontroller
Microchip Technology, Inc. | www.microchip.com

LP2986 and LP38690SD LDO Linear regulators
National Semiconductor | www.national.com

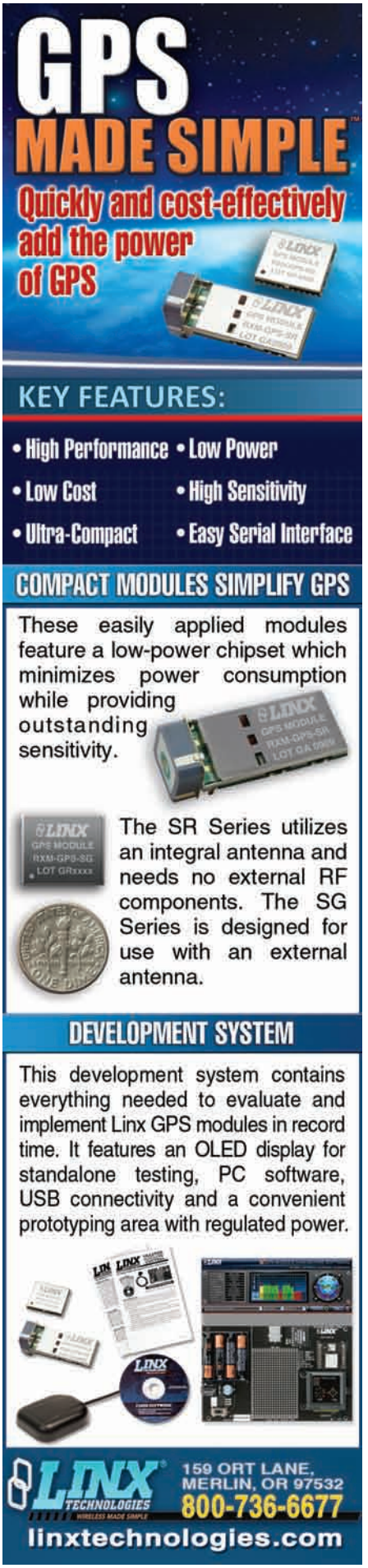
CLM-112-02-L-D High-density 60-pin connector
Samtec, Inc. | www.samtec.com

FV-M8 GPS Engine board
San Jose Technology, Inc. | www.sanav.com

PTN78000WAZ Power module and SN65HVD230D CAN transceivers
Texas Instruments, Inc. | www.ti.com

Simulink Simulation and model-based design
The MathWorks, Inc. | www.mathworks.com

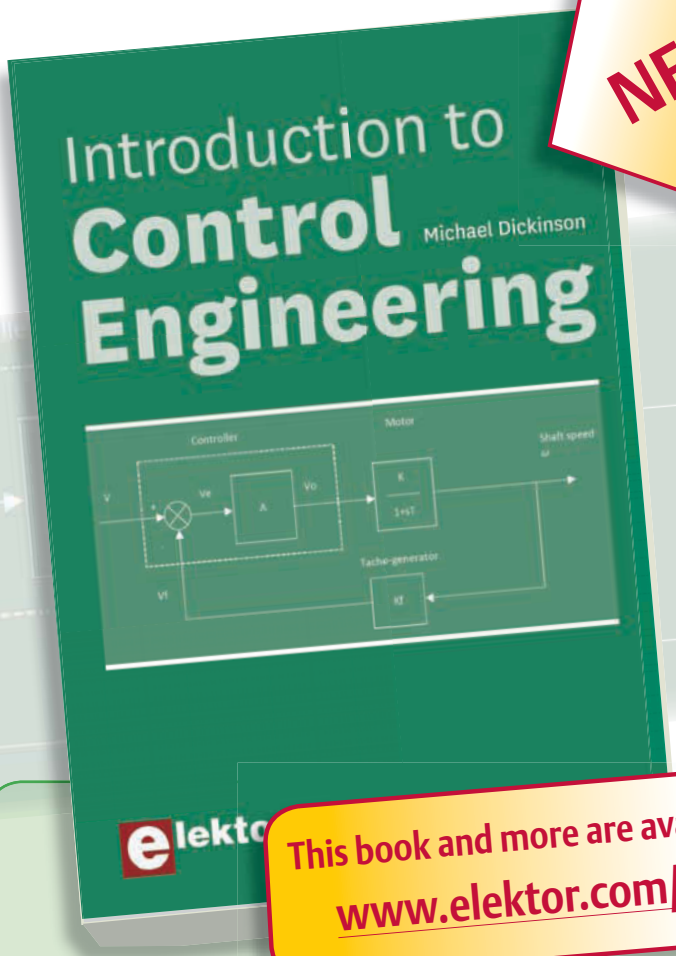
Lithium-Polymer (LiPo) batteries
Thunder Power RC | www.thunderpowerrc.com



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

Elektor Shop

The world of electronics at your fingertips!



NEW!

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

Books

A sourcebook of solutions for control system applications

Introduction to Control Engineering

This book is intended as a source of reference for hardware and software associated with instrumentation and control engineering. Examples are presented from a range of industries and applications. Throughout the book, circuit diagrams and software listings are described, typical of many measurement and control applications. The hardware and software designs may be used as a basis for application by the reader. The book contains examples of PIC, PLC, PAC and PC programming. All code samples are available to download free of charge from the support website. After an introductory section on control theory and modelling, the text focus is upon software for control system simulation and implementation, with appropriate reference to interfacing, electronic hardware and computing platforms.

164 pages • ISBN 978-0-905705-99-6 • $44.40

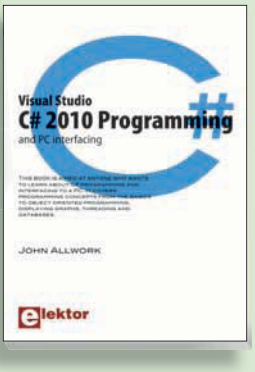
Bestseller!

Associated starter kit available

ARM Microcontrollers

This is the perfect book for people who want to learn C and who want to use an mbed ARM microcontroller in an easy and fun way. The mbed NXP LPC1768 uses cloud technology. This means you do not need to install software on your PC in order to program the mbed! The only thing you need is a browser and a USB port on your PC. No previous experience or knowledge required. You can get access to your project from any PC anywhere in the world and continue working on it. When you are done a few mouse clicks transfer the program to your mbed hardware.

250 pages • ISBN 978-0-905705-94-1 • $47.60



Visual Studio

C# 2010 Programming and PC interfacing

This book is aimed at anyone who wants to learn about C# programming and interfacing to a PC. It covers programming concepts from the basics to object oriented programming, displaying graphs, threading and databases. The book is complete with many full program examples, self assessment exercises and links to supporting videos. All code examples used are available – free of charge – from a special support website. Professional quality software tools are downloadable –also free of charge– from Microsoft.

305 pages • ISBN 978-0-905705-95-8 • $47.60

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

CD/DVD-ROMs

Packs 1.2 gigabytes worth of technical documents

DVD Wireless Toolbox

On this DVD-ROM you'll find a number of technical documents and tools that will enable you to add wireless data exchange to your electronics systems. In accordance with the principle of our Toolbox series, we've brought together technical documentation (spec. sheets, application notes, user guides, etc.) on various devices according to the frequency and/or protocol used. All of the documents are PDF files (in English). Browsing around the DVD is made easy by an HTML menu. Finally, this Wireless Toolbox DVD contains a collection of articles on this topic (RFID, xBee, DCF77, GPS, infrared, etc.) that have appeared in Elektor magazine.

ISBN 978-90-5381-268-6 • $46.00

A whole year of Elektor magazine

DVD Elektor 2010

This DVD contains all editorial articles published in Volume 2010 of the of the English, American, Spanish, Dutch, French and German editions of Elektor. The DVD is packed with features including a powerful search engine and the possibility to edit PCB layouts with a graphics program, or printing hard copy at printer resolution.

ISBN 978-90-5381-267-9 • $37.90

More than 75 power supply designs

CD The Power Supply Collection 1

This CD-ROM contains more than 75 different power supply circuits from the volumes 2001-2005 of Elektor. Highlights include the Cuk Converter, Automatic Battery Switchover, Battery Voltage LED, Digital Benchtop Power Supply, Lithium-Ion Charger and much more. Using the included Adobe Reader you are able to browse the articles on your computer, as well as print texts, circuit diagrams and PCB layouts.

ISBN 978-90-5381-265-5 • $28.90

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

Elektor US
PO Box 180
Vernon, CT 06066
USA
Phone: 860-875-2199
Fax: 860-871-0411
E-mail: order@elektor.com

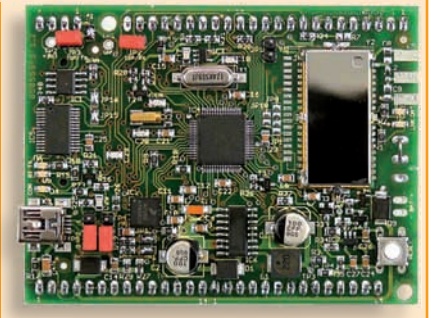
Kits & Modules

NetWorker

An Internet connection would be a valuable addition to many projects, but often designers are put off by the complexities involved. The 'NetWorker', which consists of a small printed circuit board, a free software library and a ready-to-use microcontroller-based web server, solves these problems and allows beginners to add Internet connectivity to their projects. More experienced users will benefit from features such as SPI communications, power over Ethernet (PoE) and more.

Module, ready assembled and tested

Art.# 100552-91 • $85.50



Reign with the Sceptre

This open-source & open-hardware project aims to be more than just a little board with a big microcontroller and a few useful peripherals — it seeks to be a 32-bit ARM7 fast prototyping system. To justify this title, in addition to a very useful little board, we also need user-friendly development tools and libraries that allow fast implementation of the board's peripherals. Ambitious? Maybe, but nothing should deter you from becoming Master of Embedded Systems Universe with the help of the Elektor Sceptre.

PCB, populated and tested, test software loaded

Art.# 090559-91 • $143.60

by Mark Csele (Canada)

DSP-Based Color Organ

Use the Convolution Technique to Create High-Performance Filters

Modern technology is employed to create a colorful light show. The project demonstrates the use of a classic DSP technique—convolution—to create high-performance filters.

"More colorful than a performance of Swan Lake," or so Don Lancaster describes the show from a color organ in his classic 1966 construction article on the "Musette" color organ. While the color organ was certainly popular in the psychedelic 1960s, the version I'll present here uses entirely modern technology (see Photo 1). But this is more than a project-focused article. I'll also describe a fundamental digital signal processing (DSP) technique that every engineer should know—convolution.

The color organ is a device that separates the musical spectrum into frequency bands (called channels) and pulses lamps in time with the music. In its simplest form, it can be a single channel in which a light simply blinks in time with the amplitude: louder musical programming results in a brighter lamp intensity. More complex devices separate music into low-, mid-, and high-frequency bands (or perhaps even more) with a number of colored lamps driven from each channel.

Originally, analog RC filters were employed to separate frequency bands, and later, op-amps improved performance. Today, a digital signal processor can easily accomplish the task of filtration and solve another fundamental problem that plagued earlier designs—handling signals that varied in amplitude. All previous designs required adjustment should the program material change in volume: turning the music down or simply playing a song with low volume meant adjusting the device until the lamps once again lit properly. (It might be worth mentioning, as well, that most early designs required tapping the speaker wires of the audio source—somewhat inconvenient for setup.) I address this shortcoming with a constant-volume control algorithm (another benefit of using a DSP chip). Other features include a microphone input and a light output, which is truly proportional to the audio amplitude by using a phase-control technique.

DIGITAL FILTERS

Multichannel color organs separate musical programming into frequency bands using filters. I chose frequency bands based on a traditional color organ design: low-channel (60 to 150 Hz), mid-channel (200 to 600 Hz), and high-channel (800 to 2,200 Hz). I later changed these in the final design.

I used digital filters to omit the need for analog filters altogether (except for the sampling filter). Convolution, the most general DSP technique, is employed here. Convolution filters can mimic the operation of an analog filter. In order to do so, the filter's impulse response must be known. (This is the only required information about the original analog filter.) The impulse response, as depicted in Figure 1, is simply the resulting output (amplitude samples) to an impulse applied to the input of the filter. Now, any arbitrary input signal can be decomposed into a series of individual impulses of varying amplitudes. With an analog signal sampled at a constant rate for each incoming sample (each an impulse itself), the output is known.

To see how convolution works, consider a simple square wave that's decomposed into simple impulse responses: a repeating series of three impulses of amplitude 0 followed by three impulses of amplitude 1 (see Figure 2). When the individual responses to each impulse (i.e., each sample) are considered (each shifted in time from the previous) and then summed, the resulting filter output may be computed.

A real input signal consists of a series of samples, each delayed in time (in my case, an array of samples beginning with the latest): these are the impulses used as

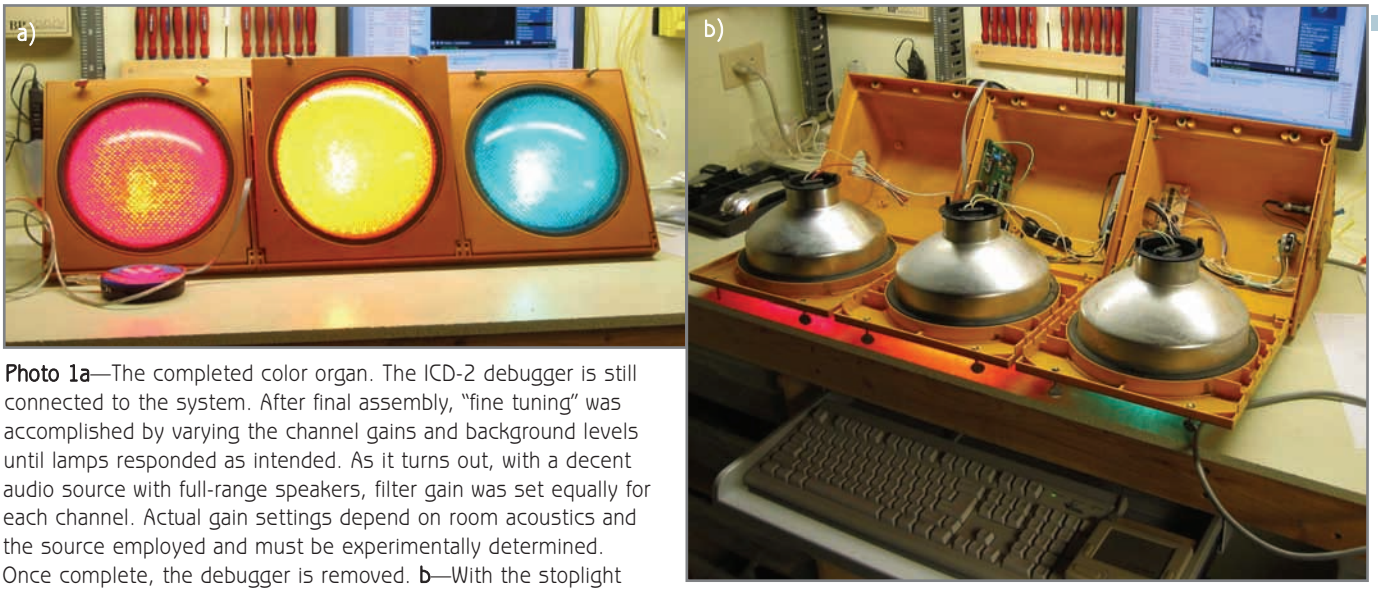


Photo 1a—The completed color organ. The ICD-2 debugger is still connected to the system. After final assembly, "fine tuning" was accomplished by varying the channel gains and background levels until lamps responded as intended. As it turns out, with a decent audio source with full-range speakers, filter gain was set equally for each channel. Actual gain settings depend on room acoustics and the source employed and must be experimentally determined. Once complete, the debugger is removed. **b**—With the stoplight open, you can see the entire assembly. The demo board (with MCU and analog preamp) is located in the middle (yellow) housing along with the 9-V power supply. The lower (green) housing holds the IEC power connector (with RFI filter), a fuse holder, and the AC driver board. The stoplight is mounted via a pipe from the top (which is why no boards were mounted into the top housing).

inputs to the filter. Each sample (impulse) of a real signal has a different amplitude, so the corresponding response of the filter also has a proportionally different amplitude but the same basic waveshape as the impulse response. We need only convolve the input sample array against the impulse response (now called the filter kernel) to compute the output value of the filter for the current sample. This operation requires computing the impulse response for each input sample and summing the resulting amplitude at each time interval. It also requires, of course, storage of past samples (with the number of samples equal to the size of the kernel). Convolution requires the use of a large number of multiplication and addition operations, a task for which DSP chips feature multiply and accumulate (MAC) instructions, the destination of which is a large register called an accumulator.

The filters employed here are rectangular window FIR type and use an output-side convolution algorithm in which the impulse response (which decays in time and is held in an array) is reversed to form the kernel. In the first prototype design, I used a kernel with a length of 64 samples (chosen arbitrarily).

While performance of the 64-tap filter was adequate for the mid- and high-frequency bands (the impulse responses decay quickly), performance of the low-frequency band was very poor, consisting of more noise than signal (see Figure 3)! The filter was improved by increasing the kernel size to 256 taps. (About 350 would be required for optimal performance—determined by observing the value of the impulse response of the filter as it decays in amplitude—but 256 was chosen arbitrarily and was found to be adequate.) While coefficients for a filter could be determined experimentally by applying an impulse to a real analog filter, a technique that could be used to reproduce the performance of a "black box" analog system, coefficients may also be calculated mathematically (as they were in the case of this project) by using a computer program that models simple analog filters. There are numerous design packages that enable a user to enter filter parameters, such as type (low, high, or bandpass), cut-off frequency, and filter order and type, which then produce the required coefficients. Coefficients produced by such utilities are typically floating-point numbers and must be multiplied by a large integer (e.g., 0xFFFF) in order to store these on an integer DSP as read-only data.

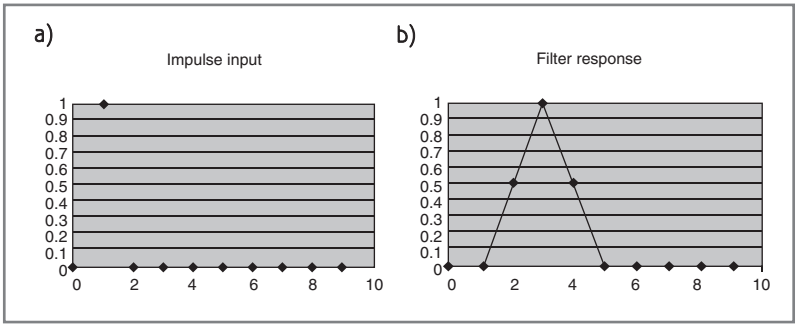


Figure 1—In this example, an impulse input to a filter **(a)** produces the output response seen in second graph **(b)**. This response approximates that of a simple low-pass filter.

IMPLEMENTING FILTERS ON A dsPIC

The kernel coefficients, stored as integers in program (X-space) memory as "HWORDs," are accessed via program space visibility (PSV) on the dsPIC, thereby allowing the use of a working register as a pointer. In order to reverse the order of the kernel, as required for the output-side algorithm, the pointer is aligned to point to

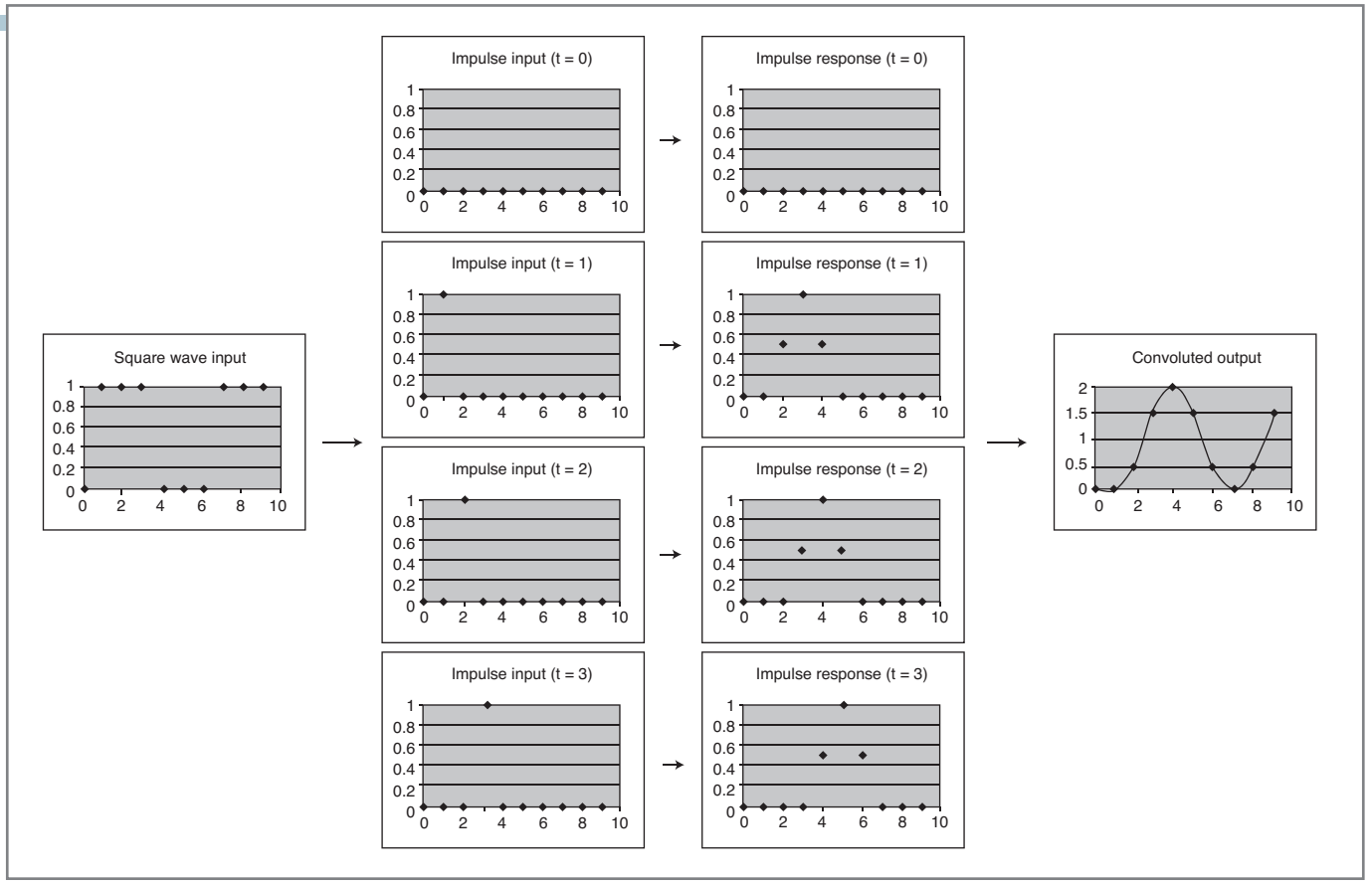


Figure 2—In this example, a square wave is decomposed into a series of impulse responses (only four are shown, but the actual wave decomposes into a string of impulses). The result is the sum of each impulse response at each point in time.

the last entry (word #255) in the kernel. To allow fast access, incoming samples (the last 256) are stored in registers in Y memory space and also accessed using a second working register as a pointer. Convolution is then accomplished by multiplying the sample values by the kernel values and storing the result in a large 40-bit accumulator register. To complete the convolution, 256 such multiplications are required with the results of each multiplication summed into the accumulator. The entire process is made easy by using the MAC instruction, which features a prefetch (so that each MAC instruction executes in one cycle) as follows:

```
clrA
repeat  #256
macw5*w6, A, [W8]-=2, W5, [W10]+=2, W6
```

The MAC instruction multiplies the contents of W5 (the kernel coefficient) and W6 (the sample value) and adds these with the value already in the accumulator (hence the opcode, "multiply and accumulate"). At the same time, the instruction fetches the next kernel coefficient (pointed to by W8) into W5 and the next sample (pointed to by W10) into W6. Finally, the pointer (W8) is decremented to point to the previous kernel entry (W10) and incremented to point to the next sample. (These are both words so each pointer is incremented/decremented by two.) Note that this instruction performs simultaneous access to two memory areas at the same time: X and Y memory spaces (which is why the kernel must be in one space, and the samples in another). The MAC instruction is repeated 256 times via the repeat instruction. When the convolution is complete, the contents of the accumulator (now a very large number) are scaled using the barrel shifter (the `sftac` instruction), which enables a shift of up to 14 bits in one instruction cycle. This code illustrates what makes a DSP chip a DSP chip: special instructions, such as the MAC instruction, that lend themselves well to algorithms such as this and can execute complex mathematical sequences in one machine cycle. Execution of these same functions on a "traditional" microprocessor would consume considerably more machine cycles (and have a much longer execution time).

In this project, three convolutions are completed, one for each frequency band, and the sample array is then updated by shifting all entries downward (with the newest incoming sample placed at the top of the array). With the filter algorithm working, filters were tested by routing the real-time output from each filter (separately) to a DAC so that an oscilloscope could be used to monitor each—the DAC on the demo board used as a platform for this project (an MCP4101) is only 8 bits, but this is sufficient for testing purposes. The demo board also has a 4-kHz reconstruction filter. In this manner filter performance was verified, and

verified to match that predicted by the utility used to design the filter coefficients. There were several problems identified with the original filter design though, notably that the low channel was sensitive to 60-Hz AC hum interference and that the low channel was not as responsive as the others. Apparently, there is less musical programming in the 60- to 150-Hz range in popular music than in the other bands, so the "low" output was hardly ever active compared to the other two. Both problems were solved by revising the design range of the low channel filter to 90 to 200 Hz, incorporating a wider range of musical instruments (including bass guitar and drums) as well as omitting the 60-Hz range where copious quantities of noise exist from the AC line. (Since the system incorporates triac drivers, there is bound to be a great deal of electrical noise generated.) It was also noted that the mid channel was too responsive, so the range of that filter was narrowed to 250 to 400 Hz (still covering a large range of musical instruments as well as vocals) and the high channel changed to 500 to 2,000 Hz. With the filters balanced in this manner, the activity on each filter channel was approximately the same. (Although with a real music source, it does vary. Naturally, some musical passages feature increased bass notes while other music features intense highs. It's this variation that makes watching the lamps interesting!)

Finally, the output of each filter channel was multiplied by a gain constant, allowing light activity to be increased or decreased. If, for example, the musical programming is weak in bass frequencies, a simple boost in this channel will improve activity. After multiplication into the accumulator, and shifting using an SFTAC instruction, the result is a value ranging from 0x00 to 0x0F in WREG0 (where 0x00 results in a dark lamp and 0x0F a lamp at full brightness). This value (with 4-bit resolution) was then used to determine the phase-angle at which to fire a triac to produce a lamp brightness proportional to the filter output.

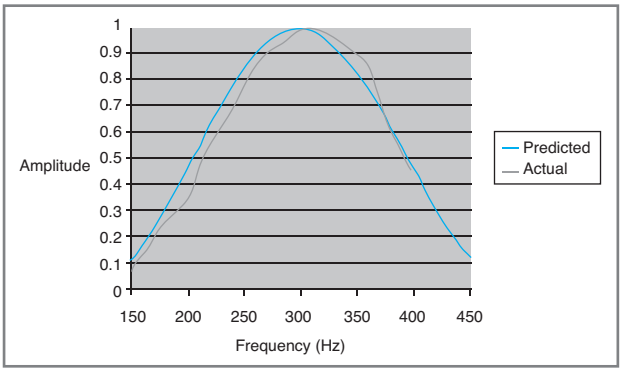


Figure 3—The predicted and measured response of a 64-tap digital filter (the prototype) shows just how closely the filter approximates the mathematically modeled version. The predicted response is the blue curve, while the measured response is the gray curve. In the final version, 256 tap filters were used with even closer response to that predicted.

CONSTANT VOLUME AMP

Input amplitude levels can vary over a large range, so an algorithm was required to adjust input gain to normalize the signal. This was a major problem with early color organ designs, since loud pieces of music would result in all lights being turned on continuously, and soft pieces would result in the lights barely blinking, requiring the operator to continually adjust the input level via a potentiometer.

Numerous algorithms exist to normalize the signal; however, the algorithm chosen is familiar to those who have used a cassette tape deck for recording: set the input gain so that peaks all fall within a known range (on most old tape

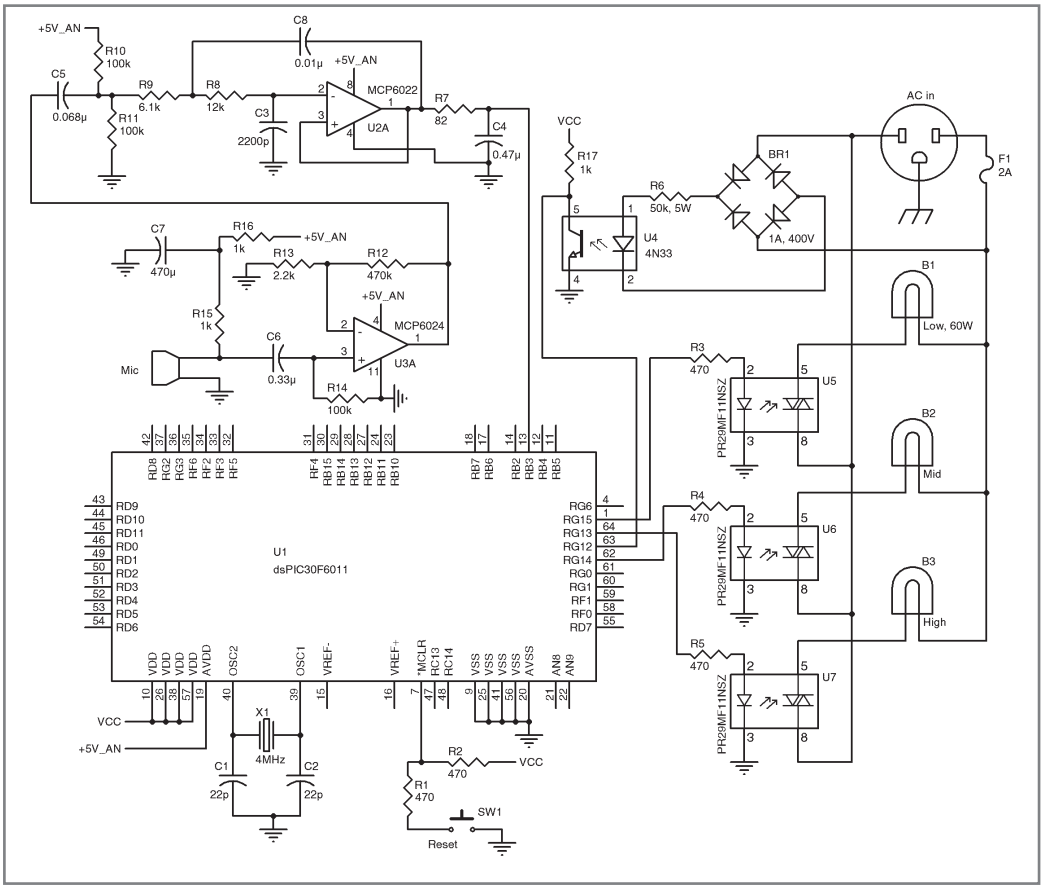


Figure 4—This basic schematic for the project shows only essential circuitry. The 4-kHz sampling filter is already on the demo board I use. If the demo board is not used, virtually any dsPIC will work, including a 28-pin version.

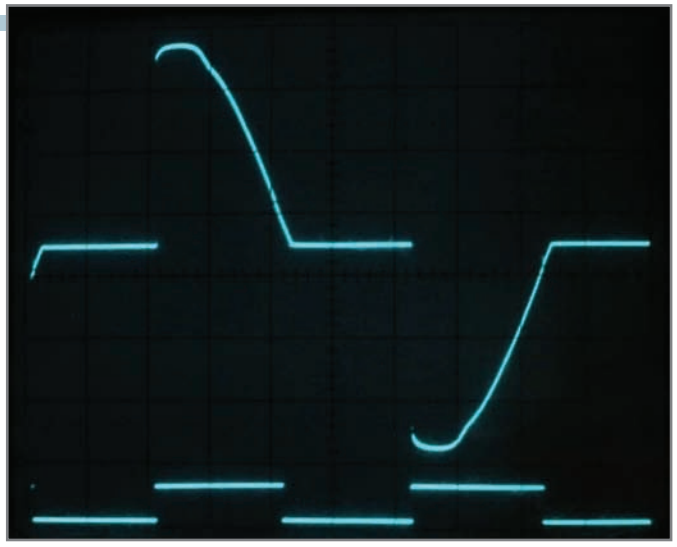


Photo 2—The triac triggering pulse (lower trace) and resulting AC output waveform (top trace) from the AC drivers. The SSR devices employed are nonzero cross devices that enable phase control.

decks, in the 0 to 3 dB range of the VU meter—the "red" area on the meters). This is a simple algorithm to implement with the only concern being the time constant used during which the peak amplitude is captured: during soft musical passages, the algorithm should not boost the signal so high that when the music becomes loud again the amplitude is enormous (so all lights would be on). This time constant must be determined experimentally and will likely depend on the user's musical preferences.

The simple algorithm for gain adjustment works as follows. Set up a counter for 32,000 samples (4 seconds). Clear the "peak" value at the beginning of this period. For each incoming new sample, update the peak to represent the maximum found (i.e., if the current sample is greater than peak, update peak). At the end of each period of samples, determine the gain by dividing the target peak by the determined peak. If, for example, the music peak for the last period was 0x300 and the desired peak was 0x4000, the gain would be calculated to be 0x4000/0x300 or 20, so all input samples for the next period are multiplied by 20 to boost peaks to 0x4000.

With very low volume levels, the signal-to-noise ratio is poor. One option to improve low-volume operation is the addition of an amplifier with a gain of 50 between the AN3 and AN4 inputs allowing the use of AN4 as a "low sound intensity" input. A simple algorithm would then switch to the AN4 input if the value on AN3 falls below a certain range, say, with peak amplitudes below 40—this will boost the input signal and improve the dynamic range. Whether or not this circuit is required depends on the volume range normally used in the listening room (in the prototype design a quad Microchip Technology MCP6024 op-amp was used allowing upgrades such as this, if required)—even at comfortably low volume levels the basic system was found to work well and the extra amplifier was not needed.

LIGHTING IT UP

Knowing that each filter has a digital output level ranging from 0x00 to 0x0F (after scaling), one of 16 possible light levels is generated to represent the intensity of sound in that band. A simple approach would be to divide the 120-Hz AC half-cycle into 16 equal steps and trigger the scr or triac, driving the light at that point, however, such a scheme results in a light output that is not proportional to the actual amplitude of each frequency band (since the slope of the sine wave is not constant). Linearization of light output (converting each step into a linear increase in light) is accomplished by a table in the PIC that translates each intensity step into a phase angle. Four bits (the four most significant bits from each filter output) are used to represent intensity. The table translates each of these steps into 16 firing angles (from 0 to 180°), which will bring about equal changes in lamp intensity (accomplished by numerically integrating the total area under the sine curve at each angular step).

The phase-control algorithm works by detecting zero-cross (twice per AC cycle) and triggering the appropriate output as required: triggering early in the AC cycle results in a bright lamp while triggering later results in a dim lamp. In the example in Photo 2, the triggering of the triac (seen in the lower trace) occurs in the first half of the AC cycle, resulting in a lamp with greater than 50% brightness.

Detection of the zero-cross of the AC line, required for phase-angle control, is accomplished using a 4N33 optoisolator. To avoid any phase-shift effects, the isolator connects directly to the rectified AC line via a large (50 kΩ) resistor. A high-going pulse is produced every 1/120 of a second, which defines "zero degrees" of the half cycle.

For simplicity, the same timer used for sampling is used to select a trigger point for the triac drivers. At 8,000 Hz (the sample rate), a 120-Hz half-wave can be triggered on one of 66 points (approximately 2.7° apart). Better resolution is possible if a higher frequency is used (e.g., 21,600 Hz would yield a resolution of one degree); however, the resolution employed is more than adequate for this application.

Each driver also features a background setting—the minimum light level that exists even with no sound present. This is simply a phase angle at which the triac triggers regardless (usually near the end of the AC half-cycle) and represents a minimum light level for the lamps. All three lamps feature an independent background level.

For hardware phase control of the triacs, nonzero cross optocouplers are required to enable the instant triggering of the triac gate. (The triac turns off when the next zero cross of the AC line occurs.) The standard approach would be to use a triac output optocoupler with external triac; however, an elegant solution is the use of small eight-pin DIP solid-state relays (SSRs) (e.g., a Sharp PR29MF11NSZF). These SSRs can drive a 0.9-A AC load directly and are nonzero cross devices allowing phase angle control of loads.

CONSTRUCTION

The actual device consists of two boards: the demo board (with additional analog circuitry) and an AC driver board. The demo board, a Microchip DM300016 dsPIC starter board, features a PIC30F6011 processor, 4-kHz sampling and reconstruction filters, an 8-bit DAC (for

testing filters), and a prototyping area large enough for all additional analog circuitry. With most of the required circuitry already on the demo board, only a microphone preamplifier was required.

The datasheet for the board, complete with schematics, is available on Microchip's website. An abbreviated version, showing only the key circuitry of the board, is shown in Figure 4. On the prototyping area, an MCP6024 quad op-amp was mounted for all analog circuitry. The MCP6024 is a rail-to-rail device designed for single supply operation at 5 V, and it's ideally suited for this application. At present, only one of the four amplifiers in the package is used for the microphone preamplifier, the other sections are reserved for future upgrades, including an amplifier for low volume usage (as previously described).

I used digital filters to omit the need for analog filters altogether (except for the sampling filter). Convolution, the most general DSP technique, is employed here. Convolution filters can mimic the operation of an analog filter. In order to do so, the filter's impulse response must be known.

Weak signals from the electret microphone are first decoupled from the DC bias used to power the microphone then amplified by a *215 amplifier. This amplification was determined by prototyping to ensure that the loudest normal audio levels encountered results in no more than a 5-V_{pp} signal at the output (i.e., without significant clipping). This amplified signal then passes through a second-order 4-kHz low-pass sampling filter (already on the demo board) and to analog input AN3 on the MCU.

All circuitry is embedded into a stoplight housing (see Photo 1b). (Stoplights such as these are commonly available in flea markets.) There is precious little room in the stoplight housing, so all boards must be designed to fit in the spaces available. The demo board (with the MCU and sampling filter onboard, and the microphone preamp on the prototype area) is mounted horizontally on the partition between the red and yellow lamp housings using two self-tapping screws, the 9-V power supply in the middle lamp housing, and the AC driver board (a separate board with the three SSR chips and the zero-cross detector) in the green housing mounted on two plastic extrusions originally meant for a terminal block. Since user controls aren't required, the only holes that must be cut into the housing are an IEC AC connector with integral RFI filter (on the rear) and a small hole for the microphone, which was placed on the front of the stoplight on the yellow lamp front. If you want to modify the gain or background level parameters, the demo board can be accessed by removing the yellow lamp (as is done for lamp replacement) and the ICD debugger may be plugged into the board via the modular connector.

CONVOLUTION

The device performs quite well. Frequency bands are suitably separated (with the low channel responding nicely to the kick of a bass drum and the high channel responding to the clash of cymbals). The constant volume feature is a welcome addition because it relieves you from having to make any adjustments during operation.

More importantly, perhaps, is the outline of the technique of convolution that I provided. A fundamental DSP algorithm, it is a powerful technique, and given the specialized MAC instructions of a DSP processor, it provides fast execution when implemented using integers as done in this project. Convolution enables the rapid implementation of digital filters in a host of scenarios. It could also be used to mimic or replace any analog system. For example, you could use it to provide the "classic" sound of a tube-based guitar amplifier to any guitar amp simply by determining the impulse response of that amplifier. Of course, that's another "fun" application. These filters can perform a variety of functions in industrial control applications as well.

Mark Csele is a professor at Niagara College, Canada. A scientist and professional engineer (P.Eng.), he teaches courses in embedded control systems, as well as laser technology, and has worked on various engineering projects involving industrial controls. Mark authored Fundamentals of Light Sources and Lasers *(Wiley, 2004). You can reach him at mcsele@computer.org. (Type "DSP" in the subject line to avoid spam filters.)*

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/249.

RESOURCES

FIR Digital Filter Design Applet, (Java 1.1 Version) www.dsptutor.freeuk.com/FIRFilterDesign/FIRFilterDesign.html

D. Lancaster, "Build the Musette Color Organ," *Popular Electronics*, 1966, www.swtpc.com/mholley/PopularElectronics/Jul1966/PE_Jul1966.htm

S. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, www.dspguide.com. (See chapter 6 for the design of the filters employed in this project.)

SOURCES

dsPICDEM Board and MCP6024 op-amp
Microchip Technology | www.microchip.com

PR29MF11NSZF SSR
Sharp | www.sharpsma.com

ICbank
Gateway to Electronics

MP3P DIY KIT, Do it yourself

(Include Firmware Full source Code, Schematic)

· myPIC

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

Powerful feature

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

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

· myWave (MP3 DIY KIT SD card Interface)

Only
$150 qty 100
$200 qty 1

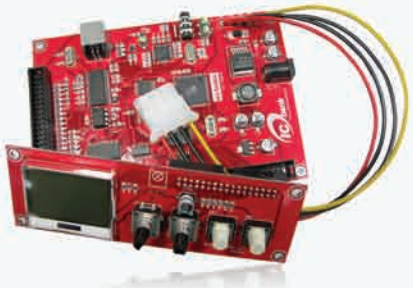
Powerful feature

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

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

· myAudio (MP3 DIY KIT IDE)

Only
$180 qty 100
$220 qty 1



Powerful feature

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

ICbank
Gateway to Electronics

www.icbank.com

ICbank
Gateway to Electronics

3 PORT INTERFACE
RS-485 to Ethernet Converter

Only
$170

RS-485 to Ethernet Converter

Powerful feature

- Protocol converter RS485 between Ethernet
- Offer TCP/IP Communication to Devices with RS485 I/F

Specification

Network	: TCP, UDP, DHCP, ICMP, IPv4, ARP, IGMP, PPPoE, Ethernet, Auto MDI/MDIX , 10/100 Base-TX Auto negotiation (Full/half Duplex)
Serial	: RS485 3 Ports, 1,200~115,200 bps, Terminal block I/F Type
Control program	: IP Address & port setting, serial condition configuration, Data transmit Monitoring
Accessory	: Power adapter 9V 1500mA, LAN cable
Etc	: - DIP Switch(485 Baud Rate setting) - LED: Power, Network, 485 Port transmission signal

by Robert Lacoste (France)

Noise Figures 101

What is the noise figure of an RF amplifier and how do you measure it? This article introduces the concept of a noise figure and presents how it's linked to a signal receiver's sensitivity.

Welcome back to The Darker Side. Let's start our journey with a small virtual experiment. Assume that you have a perfect resistor—say, a 1-kΩ resistor—without any RF perturbations. If you connect a DC voltmeter between its two legs, you will, of course, measure 0 V. And, if you switch the voltmeter to AC mode, you will still read 0 V, right? If you think this is true, then read this article, as I will explain why you're wrong.

This month I'll present some basic concepts related to thermal noise, describe the noise figure of an RF amplifier, and explain how to measure it. As you'll see, the noise figure is a fundamental concept that's closely linked to the sensitivity of any signal receiver.

THERMAL NOISE

Last winter in Paris, as I started working on this article, snow was piled up nearly 8" (around 20 cm). That's great weather for staying inside and writing for *Circuit Cellar*. However, in warm weather, I like to do things like walking, cycling, and sailing. Electrons are like us: They tend to move more when the temperature increases, even when there isn't any external voltage source. So, when you consider a conductor, there is statistically a small electrical signal that appears between its two ends, caused by the thermal agitation of the electrons (see Figure 1a). This thermal noise, also called Johnson-Nyquist noise, is proportional to the ambient temperature.

Thermal noise is white noise, meaning that it is equally spread across the full frequency range. The higher the frequency bandwidth, the higher the noise. More precisely, the RMS voltage U of the thermal noise that appears across a resistor R can be calculated with the following formula:

$$U_{THERMAL} = \sqrt{4kTR(BW)}$$

In this formula, U is in volts, T is the ambient temperature in Kelvins, R is the resistance in ohms, BW is the frequency bandwidth in hertz, and k is Boltzmann's constant, and equal to 1.38 × 10^{-23} J per Kelvin. If you make the calculation for a 1-kΩ resistor at room temperature, and with a bandwidth of 1 MHz, you'll calculate a voltage of 4 µV, which is small but measurable. This formula also indicates that larger-value resistors produce more thermal noise.

I've covered the topic of impedance matching in previous *Circuit Cellar* articles, so you know you can get the maximum power out of a source when you connect it to a receiver that's impedance-matched with this source. Here, this means you can get the most of the thermal noise if you connect this resistor R, considered as a source, to a receiver system that has the same impedance R (see Figure 1b). The noise voltage is then divided by two between the two resistors, and the noise power that you get in your receiver is calculated as usual with the Ohm formula: $P_{THERMAL} = U_L^2/R$, with $U_L = U_{THERMAL}/2$. Do the calculation and

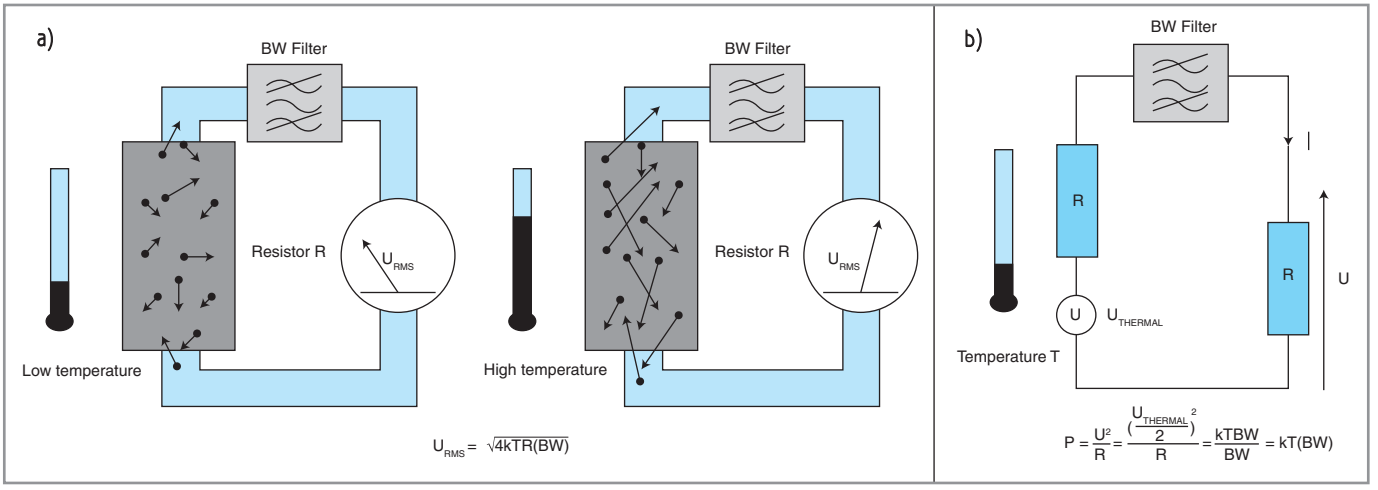


Figure 1a—Thermal white noise is generated across any resistive device: the higher the temperature, the higher the generated voltage. **b**—When the circuit is closed on a matching resistor, the power transfer is maximal. The thermal noise power is proportional to the temperature and to the frequency bandwidth, but independent of the resistor value.

you'll get the following result:

$$P_{THERMAL} = kT(BW)$$

The interesting point is that this power is independent of the value of the resistor and is simply proportional to the temperature and the frequency bandwidth. An ambient temperature (20°C, or 294°K) gives $P_{THERMAL} = 4.05.10^{-21}$ W per hertz of bandwidth. This also can be expressed as:

$$P_{THERMAL}(dB) = 10\log_{10}\left(4.05 \times \frac{10^{-21}W}{1\,mW}\right) = -174 \text{ dBm per hertz}$$

This is a fundamental result. The best noise floor of a perfect receiver will be –174 dBm per hertz of bandwidth, whatever its impedance. For example, this will result in –174 + 30 = –144 dBm if its bandwidth is 1 kHz, or –134 dBm for 10 kHz.

NOISE FIGURE?

Any system will suffer from thermal noise, proportional to the temperature and proportional to the frequency bandwidth. Suppose you have a signal coming from an antenna with a given signal-over-noise ratio (SNR). If you want to amplify it, the best you can do is amplify the signal and noise by the same amount, because the amplifier won't be able to distinguish between the signal and noise. The SNR at the output then would be the same as the SNR at the input. Unfortunately, perfect amplifiers don't exist because they are built with real components. These components also suffer from thermal noise, which adds to the input thermal noise. In a nutshell, the SNR at the output of any device will be lower (meaning worse) than the SNR at its input (see Figure 2). By definition, the ratio between these two SNRs is the device's noise factor:

$$F = \frac{SNR_{INPUT}}{SNR_{OUTPUT}}$$

This noise factor F will always be lower than one. As usual, RF engineers prefer using decibels. This gives us the device's noise figure, which is nothing but the noise factor expressed in a logarithmic form:

$$NF(dB) = 10\log_{10}(NF) = 10\log_{10}\left(\frac{SNR_{INPUT}}{SNR_{OUTPUT}}\right)$$

The logarithmic form of a division is the difference between both logarithmic forms. Thus:

$$NF(dB) = 10\log_{10}(SNR_{INPUT}) - 10\log_{10}(SNR_{OUTPUT}) = SNR_{INPUT(dB)} - SNR_{OUTPUT(dB)}$$

A device's noise figure is simply the difference, in decibels, between the SNR at its input and output, which is always positive. So, the noise figure is an indication of the device's performance. If a given receiver has a noise figure 1 dB better than another one, its output signal will be 1 dB better in terms of SNR for the same input signal. To be precise, this definition assumes that all parts of the system (source, device, etc.) are at the same fixed temperature, usually the standard 290°K ambient temperature. This is usually true, except for systems in outer space. But that topic would require another article.

CASCADED NOISE

Let's review some basic examples. The noise figure of a passive system, like an attenuator, is simply equal to its loss. This can be demonstrated mathematically, but you can also understand it in simple terms (see Figure 3). Suppose you add a 10-dB attenuator in front of a receiver that has a –130-dBm noise floor. This attenuator has no way to reduce the receiver's noise floor, so it will stay constant at best. However, any input signal will be reduced by 10 dB through the attenuator. Therefore, the system's SNR will be degraded by 10 dB, which means that the attenuator's noise figure will be 10 dB. This is just an example, but you get the idea.

What about an active device like an amplifier? The noise figure of an integrated amplifier is usually noted in its datasheet. As I already explained, this noise figure is always positive, and it's often in the range of a couple of decibels. For example, the noise figure of a common

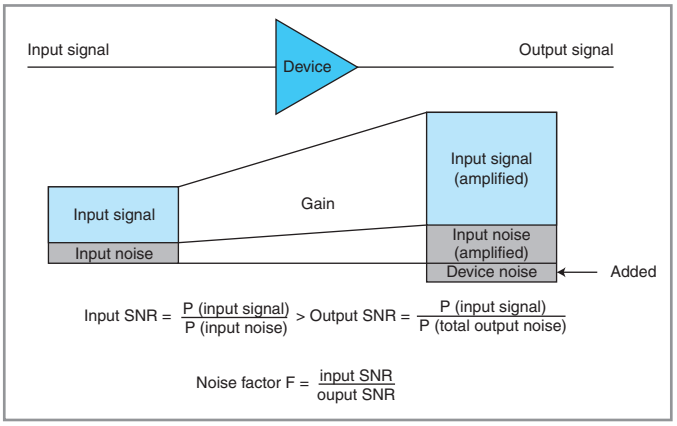


Figure 2—The noise factor of a device is the ratio of its input to output signal-over-noise ratios. Its noise figure is the same value expressed in logarithmic units (decibels).

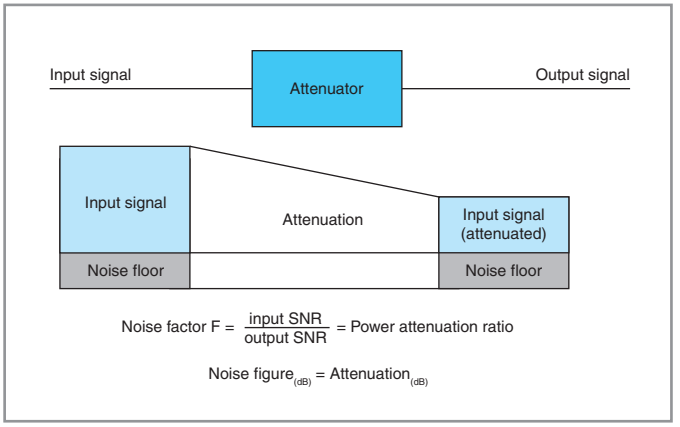


Figure 3—A passive circuit, like an attenuator, reduces the signal strength but not the noise floor: Its noise figure is simply equal to its losses.

ERA-2+ DC-to-6-GHz gain block from Mini-Circuits is a respectable 4 dB. So-called low-noise amplifiers (LNAs) have, of course, lower noise figures than standard amplifiers. A PMA-545+ LNA from the same supplier will show a noise figure of only 0.8 dB with the same 6-GHz maximum frequency, but with a different price. If you build a discrete amplifier, you have to calculate its noise figure by yourself. However, as a starting point, the noise factor of a bipolar transistor amplifier is roughly proportional to the collector current and to the collector-emitter voltage. Thus, the lower the better.

Building an actual electronic design means cascading several building blocks: amplifiers, filters, mixers, and more. What will be the noise figure of the entire system based on the noise figure of each block? The Friis formula provides the answer (see Figure 4):

$$F = F1 + \frac{F2-1}{G1} + \frac{F3-1}{G1 \times G2} + \frac{F4-1}{G1 \times G2 \times G3} + \ldots$$

In this formula, the F variables are the noise factors of each successive stage. The G variables are their respective

System on Module
Internet Appliance Engine
SoM-9G45
• AT91SAM9G45 ARM9 400Mhz CPU
• 4 Serial Ports & 2 SPIs
• Up to 40 Digital GPIOs
• 2 USB 2.0 Host/Device Ports
• I2S Audio Interface
• 10/100 BaseT Fast Ethernet
• SD/MMC Flash Card Interface
• Up to 1 GB Flash & 256 MB RAM
• Linux with Eclipse IDE & WinCE 6.0
• 8, 10-Bit A/Ds & 4 16-Bit Timer/Counters
• Graphic LCD Interface with 2D Acceleration
• Small, 200 pin SODIMM form factor (2.66 x 2.38")
Windows CE
2.6 KERNEL
The SoM-9G45 uses the same small SODIMM form-factor utilized by other EMAC SoM modules, and is the ideal processor engine for your next design. All of the ARM9 processor core is included on this tiny board including: Touchscreen Interface, Flash, Memory, Serial Ports, Ethernet, I2S Audio Interface, PWMs, Timer/Counters, A/D, Digital I/O lines, and more. Like other modules in EMAC's SoM product line, the SoM-9G45 is designed to plug into a custom or off-the-shelf Carrier board containing all the connectors and any additional I/O components that may be required. The SoM approach provides the flexibility of a fully customized product at a greatly reduced cost. Single unit pricing starts at $190.
http://www.emacinc.com/som/som9g45.htm
Since 1985
OVER
25
YEARS OF
SINGLE BOARD
SOLUTIONS
EMAC, inc.
EQUIPMENT MONITOR AND CONTROL
Phone: (618) 529-4525 • Fax: (618) 457-0110 • Web: www.emacinc.com

FlashPro430
FlashPro-CC
FlashPro2000
GangPro430
GangPro-CC
USB Flash Programmers for Texas Instruments' MCUs
MSP430, Chipcon CCxx, C2000 DSPs
Reliable and the fastest programmer on the market.
Perfect for production usage.
* can assign unique serial number
* up to eight programmers can be connected to one PC and program target devices simultaneously
Programmers
USB-FPA-1
USB-FPA-2
USB-FPA-3
USB-FPA-8
JTAG / SBW or BSL
MSP430Fxx
MSP430Fxx
MSP430Fxx
MSP430Fxx
USB
PC
One PC and 8 programmers
Elprotronic
Incorporated
www.elprotronic.com

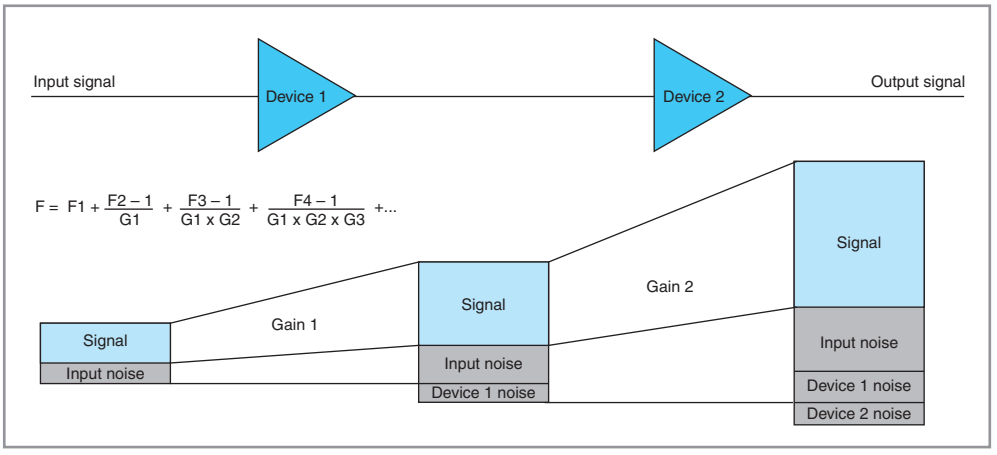


Figure 4—The Friis formula calculates the noise figure of a cascaded chain based on the gains and noise figures of each element of the chain. Usually, the first stages dominante the overall result.

gains. Be careful. This formula is expressed in noise factors (linear terms) and not in noise figures (logarithmic), making the actual calculation a little painful. Fortunately, you'll find plenty of cascade noise figure calculators on the Internet, either as spreadsheets or as more elaborate software, like the excellent (and free) Agilent AppCad tool (see Photo 1).

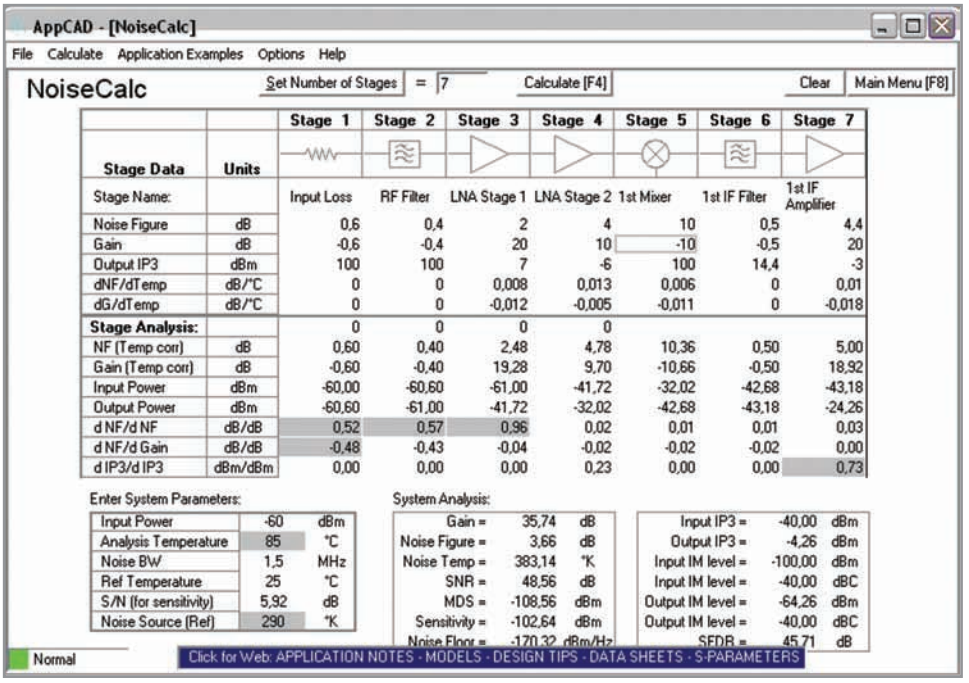
Refer back to the Friis formula to understand a very fundamental point: The successive stages usually have positive gains. This means the dividers of the successive factors of the equation are larger and larger, and as a result their contribution to the system's overall noise factor will be lower and lower. In engineering terms, the noise figure of a well-engineered receiver is mainly due to the noise figure of the first blocks—that is, to the components closest to the antenna connector. That's why it's a good idea to install an LNA close to your TV antenna and not at the other end of a lossy low-cost, 100-m coaxial cable.

FROM NOISE TO SENSITIVITY

The noise figure of any RF receiver is closely linked to its sensitivity, which is, of course, the key performance factor for a good receiver. Let's address how.

Globally, a receiver includes some RF amplifying, filtering, and conversion stages, followed by a demodulator, which enables you to recover the information for the RF signal (see Figure 5). This demodulator can be a digital demodulator, like in your mobile

Photo 1—Software like the free AppCad easily calculates the characteristics, including noise figure, of a chain. In this example of a classic heterodyne receiver, the overall calculated noise figure is 3.66 dB, mainly dominated by the input losses of 0.6 + 0.4 = 1 dB, plus the 2-dB noise factor of the first LNA, which totals 3 dB out of the 3.66 dB in the full chain.



phone, or the human ear, in the case of an old AM/FM audio receiver. In any case, the demodulator's performance can be expressed by a minimum SNR required for proper signal recovery. Let's call it $SNR_{DEMODULATOR,\ MIN}$. The preceding RF conversion stages have a given noise factor (NF_{CHAIN}). And lastly, the overall chain has a given RF bandwidth in hertz (BW).

What is the full receiver's sensitivity? As I described at the beginning of this article, the noise floor of a perfect receiver will be –174 dBm per hertz of bandwidth. Our receiver is not perfect, so we should add the noise factor of the receiver. We should also take into account the bandwidth, which is not 1 Hz, and the SNR margin needed by the demodulator, and *voilà*:

$$\text{Receiver sensibility (dB)} = -174\ \text{dBm} + NF_{CHAIN} + 10\log_{10}(\text{BW}) + SNR_{DEMODULATOR,\ MIN}$$

What does it mean? First, that a sensitive receiver is a receiver with a low noise figure, which means with low-noise first stages and as little attenuation in the early signal stages as possible. It also means that a narrowband receiver will always be more sensitive than a wide band one. Unfortunately, that's why high-bitrate transmission systems always have smaller coverage than low-speed ones. A typical Wi-Fi access point has a sensitivity of about –70 to –90 dBm, depending on the modulation mode; whereas a low-cost and low-speed Texas Instruments CC2500 2.4-GHz transceiver shows a –104-dBm sensitivity at 1,200 bps, mainly because the RF channel is far narrower. Lastly, this formula also shows that the minimum SNR needed for demodulation is also critical. That's why ham radio operators are still using Morse code from time to time, as nothing is demonstrated in a

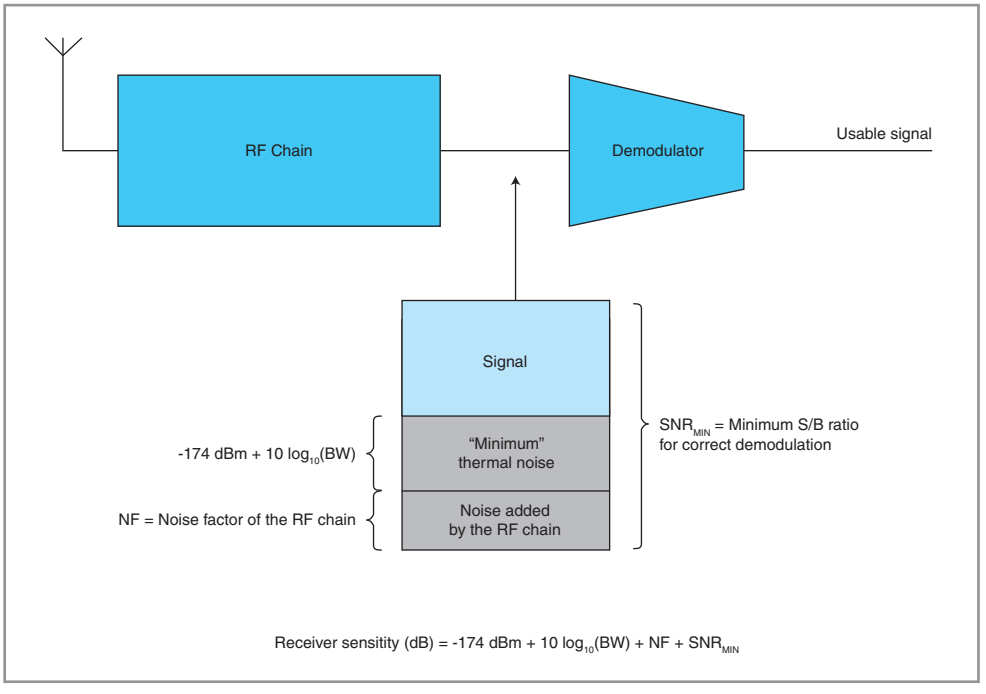


Figure 5—The sensitivity of a receiver is closely linked to its noise factor. The other two parameters a designer can work on are the frequency bandwidth (the narrower the better) and the minimum signal-to-noise ratio that the demodulator can accept.

better signal-to-noise demodulation than a trained pair of ears listening to *di-dah-dah-dits*.

MEASUREMENTS

This article is a little more theoretical than usual, so I'm sure you're anxious to get to the bench as soon as possible. Let's take any RF amplifier off the shelf. I found a Mini-Circuits ZQL-2700M SMA-connectorized LNA module I bought some time ago for a 2.4-GHz project. Its usable frequency band is 2,200 to 2,700 MHz, and its datasheet notes a minimum gain of 25 dB and a maximum noise figure of 1.3 to 1.5 dB, depending on the frequency. How do you measure this noise figure? There are, in fact, three methods.

The so-called "Y-Factor" method enables you to use more standard equipment. The idea is to measure the output noise of the amplifier when an external noise source is switched on and off at its input.

The easiest method is to use a dedicated instrument—a noise figure meter. The reference is the obsolete, but still very expensive, HP8970B. I assume you don't have such a tool. Neither do I! The second solution is to use the definition of the noise factor itself. If you know the amplifier's gain, just connect a load at its input and measure its output noise with a spectrum analyzer with a defined resolution bandwidth. You can calculate its noise factor assuming that the ambient temperature is close to 290°K:

$$NF = P_{NOISE\ MEASURED} + 174\ DbM/hZ - 10\log_{10}(RBW) - Gain$$

Next, you have to first measure the amplifier's gain with a given input signal and then the noise level without any signal at the input. This method, which is called the "gain method," has a major drawback: the measurement is often limited by the noise floor of the spectrum analyzer. For example, to measure an LNA like mine, the analyzer's noise floor must be lower than –147.5 dBm/Hz (i.e., 1.5 – 174 + 25), which is below the sensitivity of many spectrum analyzers.

Fortunately, there's another method. The so-called "Y-Factor" method enables you to use more standard equipment. The idea is to measure the output noise of the amplifier when an external noise source is switched on and off at its input. The difference, called Y, allows you to calculate the noise figure of the amplifier. This is, in fact, the method used by noise figure analyzers. To use this method, you need a noise source and a spectrum analyzer (even a very sensitive milliwattmeter may be used in place of the spectrum analyzer). The noise source, often called an excess noise ratio (ENR) source, is simply a precisely calibrated diode that generates strong

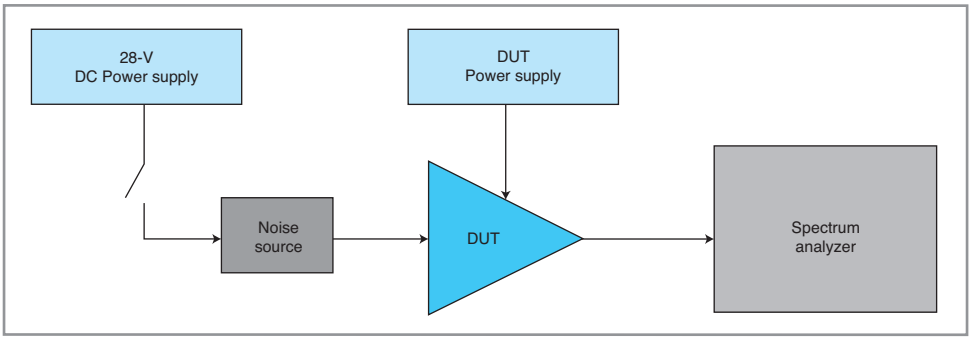


Figure 6—The Y-Factor method just needs a noise source and a spectrum analyzer, plus a pair of standard power supplies.

noise when a high DC voltage is applied on its terminals. This voltage, typically 28 V, could be automatically generated by the noise figure meter, or simply by a lab power supply. You can buy noise sources from brokers or on eBay. The best ones, like the classic HP346A/B, are unfortunately quite expensive, but they are very stable and ultra wideband (from 10 MHz to 18 GHz). However, you can find more affordable devices, like the old one I got years ago (see Photo 2). Such noise sources are very robust. Just be sure to buy one with its manual or at least with its specified ENR value. This figure (15.5 dB in my case) is in the extra noise figure of the noise source when the DC voltage is switched on, and it could be frequency-dependent. The test setup is simple, as shown in Figure 6 and Photo 3.

Here we are. For the moment, keep the DC power supply driving the noise source switched off. You now just need to correctly configure the spectrum analyzer. Remember my article about frequency domain equipment in *Circuit Cellar* 243? Here, you should select a center frequency equal to the carrier frequency band you are interested in—say, 2.4 GHz. You should also select the wider possible resolution bandwidth filter because a wider bandwidth will enable you to grab "more noise." Select 0-dB attenuation to increase the analyzer's sensitivity. And lastly, reduce the video bandwidth filter to optimize once more the sensitivity of the measurement. With this setup, you should be out of the analyzer's noise floor with any decent equipment. It's easy to check. If you disconnect the analyzer's input from the amplifier under test, you should see that the measured noise value goes down. If not, well, recheck your settings or your analyzer. When you're done, the actual measurement is easy: measure the current noise level, switch on the noise source, and measure the difference (see Photo 4). Here I found a difference of 11.46 dB.

Trust me. This is the calculation you will have to do:

$$\mathrm{NF}\ (\mathrm{dB}) = 10\log_{10}\left(\frac{10^{(\mathrm{ENR}/10)}}{10^{(Y/10)} - 1}\right)$$

So, for my example, we have:

$$\mathrm{NF}\ (\mathrm{dB}) = 10\log_{10}\left(\frac{10^{(15.5/10)}}{10^{(11.46/10)} - 1}\right) = 1.05\ \mathrm{dB}$$

The measured noise figure for this amplifier is even better than its specification (1.3 to 1.5 dB). That's quite a good amplifier.

WRAPPING UP

Noise calculations are a fundamental topic for all RF receiving chains, and this article was, of course, only presentation of the basics. I encourage you to read Agilent's application note, "Noise Figure Measurement Accuracy—The Y-Factor Method," which details the Y-Factor method's uncertainties, as well as important concepts like noise temperature. This application note supersedes the old, but always relevant, Hewlett-Packard application note, "Spectrum Analysis—Noise Figure

Photo 2—This old noise source has an unknown supplier. (Maybe Marconi?) It isn't very pretty, but it works. Fortunately, it was supplied with its ENR value (here 15.5 dB). Such a source has a DC voltage input, usually 28 V, through a BNC connector, and an RF SMA or N output connector.

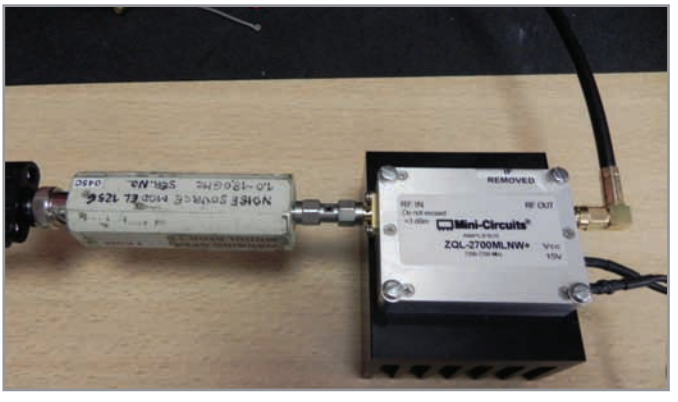


Photo 3—This is my test setup. The noise source is connected to a Mini-Circuits low-noise, 2.2- to 2.7-GHz amplifier. The output of the amplifier is connected to a spectrum analyzer. Lastly, two DC power supplies are driving the noise source and the amplifier, respectively.

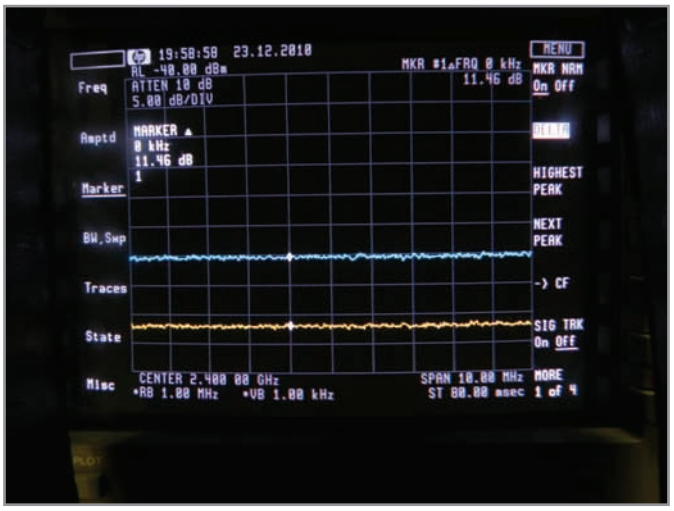


Photo 4—This is the measurement shown on my trusty HP70210C high-end spectrum analyzer. The yellow signal is the noise level measured when the noise source is switched off. The blue is when it's on. The difference, measured with delta markers, is 11.46 dB which allows us to calculate the noise figure of the amplifier under test.

Measurement," which is, unfortunately, now quite difficult to find. Lastly, Henry Ott's excellent book, *Noise Reduction Techniques in Electronic Systems*, is as always a must-read for anyone interested in noise-related topics. I hope you already have a copy of it.

What else? Well, maybe should I warn you that the noise figure-to-sensitivity relationship is a little more complex in modern receivers full of software-defined radios, clever digital signal processing, and antenna diversity features. Anyway, I hope you caught all of the basics and that noise is a little less "dark." Don't hesitate to experiment!

Robert Lacoste lives near Paris, France. He has 20 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.

REFERENCES

Agilent Technologies, "Noise Figure Measurement Accuracy—The Y-Factor Method," Application Note 57-2, 2001.

A. Bensky, *Short-Range Wireless Communication*, Newnes/Elsevier, 2003.

Hewlett-Packard, "Spectrum Analysis—Noise Figure Measurement," Application Note 150-9, 1976.

Maxim Integrated Products, "Three Methods of Noise Figure Measurement," Application Note 2875, 2003.

"Noise Figure," www.microwave101.com.

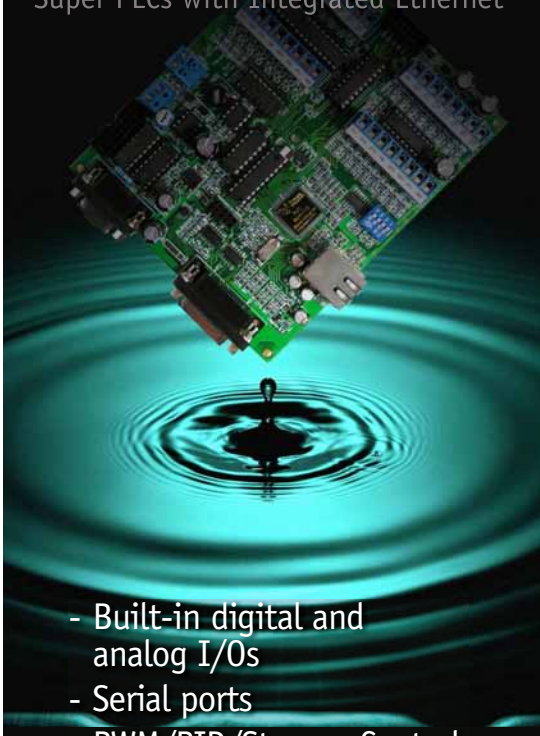
H. Ott, *Noise Reduction Techniques in Electronic Systems*, Second Edition, Wiley-Interscience, 1988.

SOURCES

AppCad Software
Agilent Technologies | www.hp.woodshot.com

ZQL-2700M Low-noise amplifier
Mini-Circuits | www.minicircuits.com

INTEGRATED
Super PLCs with Integrated Ethernet



- Built-in digital and analog I/Os
- Serial ports
- PWM/PID/Stepper Control
- MODBUS TCP/IP

Integrate your OEM machines into today's sea of Internet-connected devices with Super PLCs from Triangle Research International. The open-board style and advanced features of the Ethernet-equipped Super PLCs make them ideal for embedding into OEM applications.

Our acclaimed TRiLOGI Ladder+BASIC software reduces complex control task programming to a few simple keywords — to date, over half a million copies of TRiLOGI have already been downloaded by enthusiastic users worldwide. Be a part of our international network.

telephone: +1-877-TRI-PLCS
facsimile: +1-775-254-1816
web: www.tri-plc.com/cci.htm

TRIANGLE RESEARCH INTERNATIONAL

by George Novacek (Canada)

Reliable Programming

Work Toward Fault-Free Software

Whether you're a freelance developer or part of a corporate design team, you know that specification changes (by clients or team leaders) are common occurrences. In such situations, software requirements can fall through the cracks. Thus, developing reliable programming skills is essential.

I hate software bugs, especially those resulting from shoddy engineering. Fortunately, bad programming is rarely seen in embedded controllers anymore. The prevalent software bugs today are more hideous and often hard to discover. Sophisticated software design environments and test tools, together with controlled design processes, have helped eliminate many programming errors. What we're often up against are bugs resulting from incomplete specifications or misunderstood requirements.

One culprit is concurrent engineering, which is a reality in today's business environment. It forces a system's hardware and software to be developed very much in parallel, making the specification a moving target. With incessant requests for specification changes, it's no wonder some requirements fall through the cracks while others are misinterpreted. Producing reliable software under those conditions is difficult. Proving it to be bug-free is *mission impossible*.

UNCERTAINTY REMAINS

How can you prove software is fault-free? A scientific proof is not a legal proof. "Beyond reasonable doubt" just doesn't cut it in engineering. It has been mathematically shown that an exhaustive test of even a small piece of code would take more time than the age of the universe. And that does not include the validation of the design environment (e.g., compilers, debuggers, etc.). I'll refer to the famous Fermat's Last Theorem to illustrate the point.

In the 17th century, Pierre de Fermat postulated that no three positive integers a, b, and c can satisfy the equation $a^n + b^n = c^n$ for any integer n greater than 2. More recently, computers (using the sledgehammer approach) performed the calculations up to astronomical values, but that still didn't prove some other numbers satisfying the equation might not exist. The theorem remained just a conjecture until a proof was discovered by Andrew Wiles in the late 20th century. Similarly, until we find a way to scientifically prove software is perfect, we must come to terms with the fact that a degree of uncertainty remains. Therefore, we must design systems with this in mind.

SOFTWARE DESIGN METHODS

To minimize programming errors, the aerospace industry has relied heavily on a design methodology that's spelled out in the Radio Technical Commission for Aeronautics's DO-178B standard. (Other safety-critical industries, such as nuclear and medical, have their own similar standards.) A new revision, DO-178C, is slated for release in 2011; but whatever it contains, let's be clear about one thing: no standard can ever guarantee production of flawless software. It enforces the discipline and the process to take steps to avoid errors. But that's all.

DO-178B software design methodology resembles a waterfall. You don't enter the next step until the current one has been finished and formally accepted. You start the development by preparing plans, such as The Plan for the Software Aspects of Certification, a Configuration Control Plan, a QA Plan, test plans, and so on. Once the plans are in place, you analyze the customer's spec and prepare System Requirements. Next you prepare the Software Requirements, followed by the Software Design Description in pseudo-code. Then you program. You follow with software integration, hardware/software integration, system integration,

verification and validation (V&V) tests, and so on, until you release the final code accompanied by the Software Accomplishment Summary. Concurrently, traceability matrices are maintained. Every requirement must be traced from the customer specification, through the requirements, down to the source code, and every line of code must find its corresponding requirement in the spec. Your tests must show full "requirements coverage."

You can imagine this process is not quite compatible with the idea of concurrent engineering. And, even for a simple embedded controller, it burns tons of time. With the spec in continuous flux, every little change at the top level has to trace through all of the steps, down and up: all documents updated, every engineering change order properly justified and recorded, and every test failure report closed.

DO-178B is *trying* to enforce strict discipline into this process. I used the verb "trying" intentionally. Just the beginning of the development with the specs not yet finalized goes against the spirit of DO-178B. And things get worse as time is burned: budgets are blown; deadlines approach; and the customer demands more changes while remaining oblivious to their impact on cost and schedule. Consequently, corners are cut, and the waterfall methodology goes the way of the Dodo bird. Safety must not be compromised, but the required accompanying bureaucracy takes a second seat.

While I'm a supporter of DO-178, it's my humble opinion that the process it demands is now well within the realm of the law of diminishing returns. The questions are: How far do we have to go striving for unattainable perfection? And at what cost? Not long ago, customers were so frustrated they insisted that software not be used in avionic equipment, if at all possible, to avoid its certification cost and burgeoning bureaucracy. But then DO-254 was introduced, bringing a process similar to DO-178 into hardware design, and the costs escalated.

During my career, I've had two encounters with bugs. The first cost me and my team an entire Christmas and New Year's vacation. The software had been certified, underwent countless hours of successful V&V and flight tests, and then failed within the first five minutes on the first aircraft delivered in the Far East. At the time, there were no sophisticated debugging tools available. After manually inspecting line after line of the machine binary code, we discovered that the compiler would fail to issue disable interrupt (DI) instructions if there was a specific sequence of statements in the source code. Was the bug preventable? Hardly. Unless, by pure luck, during testing the conditions needing the interrupts to be disabled arose and we experienced a fault.

Another bug took us several years to nail down, although, strictly speaking, the software was doing exactly what it was supposed to do. Occasionally, the system declared a fault that could be cleared by reset. It had no impact on safety but was annoying and considered by the customer as a flaw we had to fix. The occurrence was very infrequent and the only background information we had were the pilots' laconic notes in the log that the system had to be reset. Instrumenting the aircraft and performing tests was out of the question. Eventually, working from the inside out, we concluded that the fault had to be caused by only one condition, which the system designer told us could never occur. He was wrong. Its occurrence was normal, albeit infrequent.

While the regulations keep pushing us further into diminishing returns, business realities demand exactly the opposite. Thus, the developer is squeezed in the middle with very few options left. In my view, we need to recognize that perfect software is the proverbial pot of gold at the end of the rainbow, determine when we achieve reasonable software reliability, and concentrate on overall system design to create safe, fault-tolerant embedded systems. To do that, we need some software reliability metrics. There have been attempts to develop them, but none have been generally accepted yet.

REDUNDANCY & ANALYSIS

Industry long ago conceded that hardware is never failure-free. Unlike hardware, software doesn't wear, but it can fail due to infrequent and unpredictable circumstances designers didn't foresee. With the help of statistics, we address hardware failures through redundancy. We can do exactly the same with software. When we use dissimilar hardware and dissimilar software in redundant channels, we prevent the same fault appearing simultaneously in all control channels. I saw an embedded, mission-critical system controlled by majority voting of three independent computers, each computer comprising three processing channels with dissimilar hardware and dissimilar software throughout. Redundancy, however, will not prevent systematic failures caused by bad system design. We cannot replace a thinking engineer with a machine or a standard.

Every system requires demonstration of achieving its availability and reliability requirements. Analyses, such as failure mode and criticality analysis (FMECA) and fault tree analysis (FTA) are used for this purpose. Establishing hardware reliability has a long tradition based on MIL-HDBK-217, as well as other methods, and is quite straightforward. Establishing reliability for software is a different story. To my knowledge, as I stated above, no generally accepted method exists. There are several approaches, but their respective protagonists seem at odds with each other. I have seen FMECAs and FTAs of certified embedded controllers taking no account of the software reliability. These were based on the assumption that, after all the software and system integration testing per DO-178, the software reliability was at least an order of magnitude better than the hardware reliability and, thus, inconsequential for the overall safety assessment. Can we prove it?

100% BUG-FREE?

Can we manufacture and prove software is 100% bug-free? That will remain *mission impossible* for some time. But building a reliable system to satisfy safety and availability requirements is well within our engineering capability.

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.*

by Ed Nisley (USA)

Thermal Performance

Understanding the principles of heat transfer and having the ability to work out heat transfer problems are essential skills for any serious electronics engineer. The techniques detailed in this article will enable you to figure out the origin of a project's heat and how to measure its progress.

Heat management doesn't pose a problem for most small electronics projects. Microcontrollers rarely sport heatsinks, desktop and laptop PCs need a fan or two, and only the most dense server racks demand liquid chillers.

A recent project forced me to work out some heat transfer problems: I had to build an oven that heated its contents to 120 °F (about 45 °C), hold that temperature for an hour, then cool it to room temperature. The reason I needed such a thing isn't relevant here, but I needed results in a hurry!

Although many of the simplifying assumptions for room-temperature circuitry don't apply inside an oven, the principles of heat transfer work the same way everywhere. With that in mind, here's how to figure out where your project's heat will come from and go to, plus how to measure its progress.

POWER PROJECTION

I built the oven shown in Photo 1 from a single 4-foot × 8-foot sheet of 1-inch thick aluminum-clad polyisocyanurate foam from a local building supply store. It's 36-inch long, 26-inch tall, and 24-inch wide, with inside dimensions two inches smaller, and wastes just a single 1-inch strip from the original sheet. I sealed the edges and joined the sides with stainless-steel foil adhesive tape.

The photograph shows the oven on the wood floor of our living room, although it's obviously not an object even an engineer would want there. I discovered that photographing a matte aluminum box against our basement's gray cement floor and concrete-block walls produced poor results: dull gray on gray.

A heater inside a closed box will increase the air temperature until the energy lost

Photo 1—A simple rectangular box serves as an oven. The box is 36 x 24 x 26-inch on the outside, with 1-inch thick walls made of aluminum-sheathed foam insulating board. The lid sitting inside the box seals against the top edge with two adjoining strips of felt insulation.

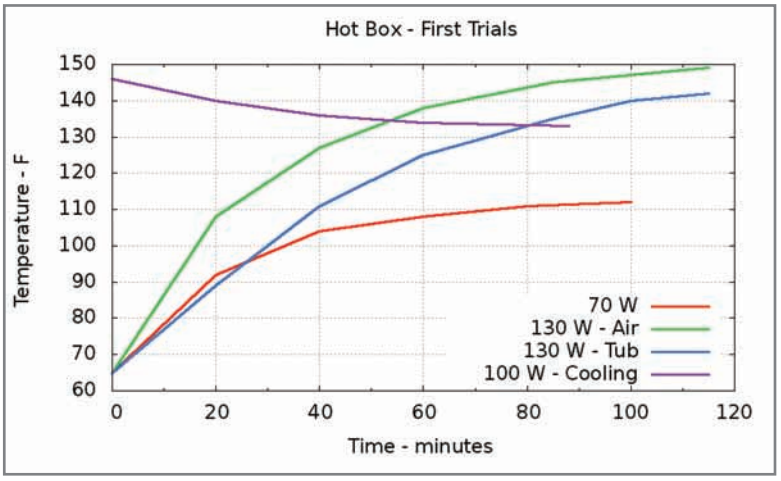


Figure 1—Plotting air temperature against time for a variety of power levels shows that the oven behaves very much as expected. The blue "tub" curve shows the temperature inside a tub of cloth, which lags the oven air temperature.

through the walls equals the energy from the source, at which point the temperature stabilizes. To a very good approximation, the temperature will change exponentially, with a time constant determined by the material being heated, although I'll consider only the steady-state response here.

Energy flows through each of the six sides at a rate proportional to the inside-to-outside temperature difference, the surface area, and the wall's thermal resistance. Because engineers in the United States must deal with odd units, the thermal resistance, known as the R-value, has units of:

$$\frac{\text{foot}^2 \cdot \text{hour} \cdot \Delta\,^\circ\text{F}}{\text{BTU}}$$

In the rest of the world, the R-value (sometimes known as RSI to indicate its SI-unit basis) has useful engineering units of:

$$\frac{\text{m}^2 \cdot \Delta\,^\circ\text{C}}{\text{W}}$$

Note that the R-value and RSI both depend on the temperature difference across the wall, not the temperature levels, so you must convert between Fahrenheit and Celsius without the usual offset:

$$\Delta\,^\circ\text{C} = \frac{5}{9} \times \Delta\,^\circ\text{F}$$

Your calculator's temperature unit conversion button will produce an incorrect answer because it converts temperature levels, not differences. I won't include the Greek delta symbol in all of the values, but keep it in mind.

The insulation board's datasheet provides R-values for several sheet thicknesses, albeit in specific conditions that probably won't exactly match your situation. As with all datasheet values, your results will differ and you must take measurements to verify the outcome.

The 1-inch thick insulating board I used has an R-value = 6.5 in U.S. units. The oven has an interior surface area of 30 ft^2, so it will lose heat at a rate of:

$$4.66\frac{\text{BTU}}{\Delta\text{°F} \cdot \text{hr}} = \frac{30.3\ \text{ft}^2}{6.5\ \text{ft}^2 \cdot \text{hr} \cdot \Delta\,^\circ\text{F/BTU}}$$

Although the final temperature of the objects inside the oven must be 120 °F, I decided to heat the air to 140 °F to increase the initial heating rate, then stabilize at 120 °F when the object's interior temperature reaches that level. The ambient temperature in my "basement laboratory" hovers near 60 °F in the winter, so the temperature differential between the oven's interior and the basement air can be 80 °F. That difference determines the heat transfer rate out of the oven:

$$370\frac{\text{BTU}}{\text{hr}} = 80\,\Delta\,^\circ\text{F} \times 4.66\frac{\text{BTU}}{\text{hr} \cdot \Delta\,^\circ\text{F}}$$

That awkward value becomes 110 W in the unit required for an electrical heater.

Homework: Verify the power using metric units, after converting the U.S. R-value to metric RSI. Remember to convert the temperature difference correctly!

A 110-W heat source inside the box can therefore maintain the interior at 80 °F above the ambient temperature outside the box, assuming that all six sides conduct heat equally. Obviously, the floor underneath the box conducts energy at a different rate than the air on the top, both of those surfaces differ from the four side walls, and the felt sealing strips along the top edge probably leak more energy than all the double-taped seams.

One measurement being worth a thousand models, Figure 1 shows the results of the first tests on the completed box. I used ordinary incandescent light bulbs as reasonably accurate power sources, plus an axial fan to circulate the air, with the basement providing the test environment.

The red trace shows that 70 W produced by a 60-W bulb and the fan raised the temperature to 110 °F in 100 minutes. A second test with two 60-W bulbs, a total of 130 W, pushed the temperature close to 150 °F in about the same time. The blue trace shows that the temperature inside a plastic bucket

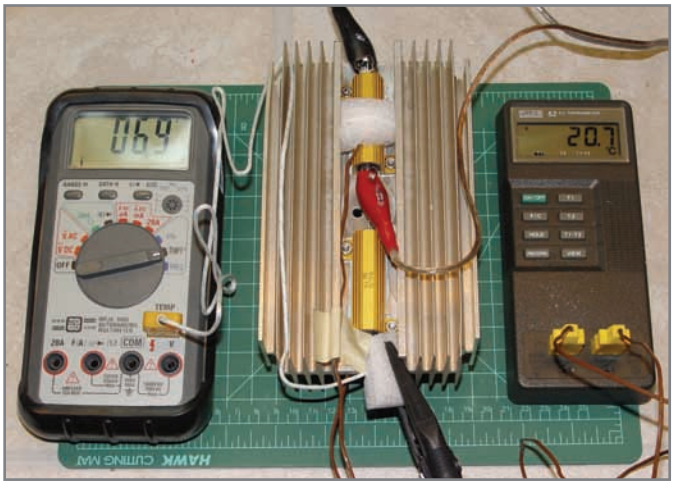


Photo 2—Two power resistors dissipate a known amount of power in the heatsink, with a trio of thermocouples recording the resistor and heatsink temperatures.

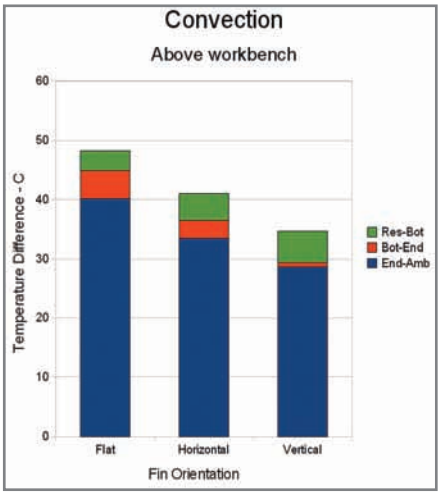


Figure 2—The temperature differentials in a convection-cooled heatsink show the poor heat transfer to the surrounding air. The dissipation is only 24 W to limit the temperature rise.

stuffed with cloth (to simulate an object to be heated) tracked the air temperature after a significant delay.

With the interior at 150 °F, I plugged the lamps into a Variac autotransformer and set it for a total of 100 W of power inside the oven. The purple traces show the temperature would stabilize near 130 °F, 63 °F above the 67 °F ambient temperature in my "basement laboratory" at the time.

Knowing the power and temperature difference, the actual R-value for the oven in those conditions is:

$$5.6 = \frac{30.3\ \text{ft}^2 \cdot 63\ \Delta\,^\circ\text{F}}{341\ \text{BTU/hr}}$$

That's 14% below the insulation board's nominal R = 6.5, which seems reasonable in view of the vagaries of construction, environment, and measurement. The power required to support a given temperature difference will be correspondingly higher.

I used light bulbs for those first measurements, but I wanted a much lower heater temperature to prevent local hotspots. In addition, using line voltage power inside a metal-lined box required far more safety interlocks than seemed reasonable.

DISSIPATION BY ORIENTATION

My parts stock produced three large-finned aluminum heatsinks that probably started life in an industrial-strength power supply. Each heatsink sported three TO-3 power transistors, which meant it could handle several hundred watts with adequate forced-air cooling and, to judge from the amount of dust decorating the fins, they'd been subjected to many years of high-speed air flow.

The usual thermal design process begins with a device's expected power dissipation and ends with the heatsink area (and, perhaps, air flow) required to keep the semiconductor junctions below their maximum-rated operating temperature at a given ambient temperature. In this case, I knew the total power dissipation, oven temperature, and had the heatsinks, so I worked backward to find the device temperature rise.

Just as insulating board has an R-value, heatsinks present a thermal resistance, known as R or θ, between the device and ambient air. Unlike the R-value, a heatsink's thermal resistance has units of Δ °C/W or Δ °F/W, because it has a fixed area. You can find the thermal resistance for a new heatsink from the manufacturer's specifications, but, lacking that information, I measured these. You should always verify your heatsink's performance in its actual operating conditions against predictions based on the datasheet values, too.

I decided to combine panel-mount power resistors with a standard PC ATX-size power supply that can provide several hundred watts of power with UL-listed AC line isolation. Molex-style connectors limit each pin to about 10 A, and the typical 18-AWG wires used in such supplies suggest an even lower limit, putting a definite upper limit on the power available for each resistor.

With plenty of room in the oven, I decided to use all three heatsinks, with each one holding two resistors dissipating a total of 50 W. ATX power supplies generally require a minimum load for proper regulation, so I divided the power equally between the +5-V and +12-V outputs. A 6-Ω resistor across the +12-V supply dissipates 24 W and 1 Ω across +5 dissipates 25 W, with a few watts of variation due to resistor and voltage tolerances.

Photo 2 shows the test setup I used to characterize the heatsinks: a pair of 50-W resistors mounted on the heatsink, with bench power supplies attached by clip leads. I adjusted the voltage to deliver a known current to each resistor, thus eliminating the effect of resistance loss in the wires and clips. I clamped thermocouples to the 6 Ω resistor body, the opposite side of the heatsink near the center of the resistor, and at one end of the thick central web, using flexible foam insulation to ensure the thermistors measured the temperature of the solid surfaces.

A thin layer of thermal compound between the resistors and the heatsinks improves the heat transfer. Both surfaces seem reasonably flat, but they're not lapped to a perfect fit, and the air gap forms a very good thermal barrier.

Although these heatsinks were obviously designed for forced-air cooling, I started by measuring the convection cooling performance while dissipating 24 W in the 6-Ω resistor and leaving the 1-Ω resistor unpowered to reduce the temperature rise. Figure 2 shows the results for three different orientations.

With the heatsink in the worst possible orientation, as shown in Photo 2, nearly the entire temperature difference occurs between the heatsink and ambient air: 40 °C. The

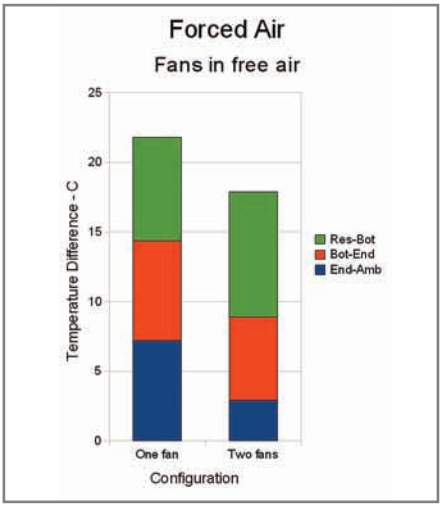


Figure 3—Forcing air across the heatsink fins dramatically improves its heat transfer. With 50 W total dissipation and fans on both sides, heat transfer between the resistor and heatsink becomes the limiting factor.

ESC
Silicon Valley • May 2-5, 2011

Join the industry's leading embedded systems event
(Translation: Can't miss event)

ESC brings together the largest community of designers, technologists, business leaders and suppliers in one place.

Keynote Speaker

Steve Wozniak

Co-Founder, Apple Computer, Inc. and electronics industry visionary delivers the opening keynote speech on Tuesday, May 3rd at ESC Silicon Valley!

Categories and Tracks that address the most relevant issues facing engineers and the industry. Take a moment to review this content, so that you can customize your educational experience.

Applications
- HMI and Multimedia
- Systems Architecture
- Reliability, Security, and Perforance
- Remote Monitoring and Wireless Networking

Embedded Software
- Linux/Android/Open Source
- Programming for Storage, I/O, and Networking
- Programming Languages and Techniques
- RTOS and Real Time Software
- Software Processes and Tools
- Windows for Embedded
- Quality Design and Untellectual Property
- Safety Design

Hardware for Embedded Systems
- Challenges and Solutions in Embedded Designs
- Connectivity and Security
- Memory in Embedded Designs
- Microcontrollers in Embedded Designs
- Powering Embedded Designs
- Programmable Logic in Embedded Designs

Tools and Best Practices
- Best Practices
- Debugging and Optimizing
- Design and Test
- Managing and Process
- Tools

Topics in Embedded-System Design
- Architecture Design
- DSP, Communication, and Control Design
- HW and Platform Design
- Quality Design and Intellectual Property
- Safety Design

Did you know that Steve Wozniak is an ESC Alumni?
"It will be good to reconnect with the engineering community that still drives so much electronics innovation."
Steve Wozniak
Co-Founder, Apple Computer, Inc. and Chief Scientist for Fusion-10

Register today and save 10% OFF any package when you enter the promo code: sp
www.embedded.com/cc

Learn today. Design tomorrow.
ESC
Silicon Valley • May 2-5, 2011

UBM Electronics

thermal resistance from the heatsink to the air is:

$$\theta = 1.7 = \frac{40\,^{\circ}\text{C}}{24\text{ W}}$$

With the heatsink on its long edge and the fins horizontal, the thermal resistance drops to 1.4 °C/W. Orienting the fins vertically improves the air flow, but the heatsink remains 29 °C above the air, with thermal resistance = 1.2 °C/W. Although that's certainly better, dissipating 50 W would raise the heatsink 60 °C above ambient!

The green bar segments in Figure 2 show that the temperature difference between the resistor body and the heatsink remains roughly constant at 3–5 °C. The temperature gradient across the heatsink, shown by the orange segments, shrinks as the air flow across the heatsink increases.

MOVING HEAT IN AIR

However, convection cooling can remove only a very limited amount of heat. Figure 3 shows the vast improvement caused by ordinary PC-case fans blowing air directly onto the heatsink, with the two resistors dissipating a total of 50 W. My bench supplies have a 3 A current limit, so I ran the 1-Ω resistor at 9 W and the 6-Ω resistor at 41 W.

A single fan reduces the heatsink-to-air temperature difference to 7 °C and two fans, one on each side, cuts it to 3 °C. Compared to the convection-cooled condition, the thermal coefficients drop by an order of magnitude: 0.14 °C/W and 0.06 °C/W, respectively.

The temperature difference between the 6-Ω resistor body and the heatsink increases due to the higher power dissipation. The thermal resistance of that junction with a single fan applied to the heatsink is:

$$\theta = 0.18 = \frac{7.4\,^{\circ}\text{C}}{41\text{ W}}$$

The coefficient is slightly higher with two fans, because the lower heatsink temperature increases the temperature difference through the heatsink web below the resistor. Even though my thermocouple mounting technique doesn't provide enough accuracy for high-resolution measurements, a value around θ = 0.2 should suffice for most purposes.

Homework: Calculate θ for the convection-cooled conditions in Figure 2.

Each resistor body will thus be 5 °C hotter than the heatsink while dissipating 25 W. The heatsink itself will be 7 °C hotter than ambient with a heat load of 50 W from both resistors.

The orange bars in Figure 3 show the temperature difference across the heatsink, measured directly below the 6-Ω resistor dissipating 41 W and the edge of the heatsink near the 1-Ω resistor dissipating 9 W. I expect this value would be lower when the two resistors dissipate nearly the same amount of power, but it's a reminder that aluminum is a relatively poor thermal conductor compared to, say, copper.

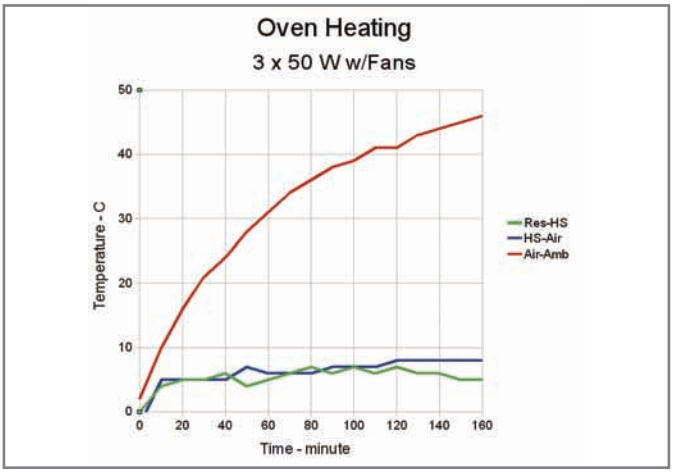


Figure 4—Heating the oven at 150 W will eventually raise the temperature about 50 °C over ambient. The resistor-to-heatsink and heatsink-to-air temperatures remain roughly constant, as expected.

The overall bar heights in Figure 3 sum the effect of those three interfaces to show that the resistor bodies will be about 20 °C hotter than the air temperature for 50 W total dissipation. Because the air ambient temperature inside the oven will be 140 °F = 60 °C, the resistors should run at 80 °C.

THERMAL DERATING

The datasheet specifications for nearly all electronic components give their values at 25 °C, the more-or-less normal "room temperature" for homes and offices. The datasheets also include temperature derating curves or coefficients that oven designers ignore at their peril: room-temperature ratings do not apply at high temperatures.

For example, the resistors can handle their 50 W rated power at 25 °C, but only 10% of that value at 250 °C. The derating curve for a resistor mounted to the datasheet's standard heatsink (a square foot of 1/16-inch aluminum: 291 in^2 surface area) slopes downward at –0.4%/°C = –0.2 W/°C, which means the maximum dissipation at 80 °C is:

$$39\text{ W} = 50\text{ W} - (80 - 25\,^{\circ}\text{C}) \times \left(\frac{-0.2\text{ W}}{^{\circ}\text{C}}\right)$$

My heatsinks have about 300 square inches of surface area, roughly comparable to the specified heatsink. However, the datasheet doesn't require either thermal compound or forced-air cooling, so I assume they anticipate dry mounting and convection cooling.

The right-hand column in Figure 2 shows a 35 °C rise above ambient with 24 W dissipation and convection cooling. Assuming the thermal resistance from the resistor to the ambient air remains constant at 1.2 °C/W, the resistors would rise 60 °C above ambient with a total of 50 W dissipation:

$$60\,^{\circ}\text{C} = 50\text{ W} \times 1.2\,\frac{^{\circ}\text{C}}{\text{W}}$$

Therefore, the resistors will run at 76 °C in the 16 °C ambient air on my workbench. That's permissible because the derating curve allows full-power operation

sensors expo & conference

Pre-Conference: June 6, 2011
Conference & Expo: June 7-8, 2011
Rosemont, IL • Donald E. Stephens Convention Center

25+ YEARS OF EXCELLENCE

www.sensorsmag.com/sensorsexpo



The Only Industry Event in North America Exclusively Focused on Sensors and Sensor Integrated Systems! Find the Solutions to Your Sensors and Sensing Technology Challenges!

Gain the knowledge you need from leading experts and peers in the sensors industry.

This year's Conference Program includes more Technical Sessions in Tracks covering the topics that are most important to you and your business, including:

- MEMS
- Energy Harvesting
- Wireless Sensor Networking
- Novel Approaches to Measurement & Detection
- Wireless Sensor Data
- Mobile Sensing – NEW for 2011!
- Cutting-Edge Technologies – NEW for 2011!

Identify specific solutions to your most difficult sensing, detection and control-related challenges on the expo floor.

Sensors Expo brings together the largest and best-in-class showcase of sensing technologies and systems for attendees to evaluate and make informed decisions.

Don't Miss these Expo Floor Highlights:

CO-LOCATED WITH:
Learn today. Design tomorrow.
ESC

ENERGY HARVESTING PAVILION

MEMS PAVILION

TECHNOLOGY UPDATE THEATER

REGISTER TODAY AT THE EARLY BIRD RATES & SAVE UP TO $400 ON YOUR CONFERENCE PASS! OR, REGISTER NOW FOR A FREE EXPO HALL PASS!

SUBSCRIBERS: Visit www.sensorsmag.com/sensorsexpo to register or call 866-817-5048. Use discount code A305E for an EXTRA $50 OFF a Gold or Main Conference Pass!

MEDIA SPONSOR: CIRCUIT CELLAR®

PRODUCED BY: QUESTEX MEDIA

OFFICIAL PUBLICATION: sensors

below 25 °C ambient.

Homework: Determine the maximum allowed power dissipation in a 60 °C oven. Bonus: Repeat that calculation for forced-air cooling.

Hint: That's why I used 50 W resistors to dissipate 25 W.

I've seen 10 W resistors used as heating elements in 200 °C environments, dissipating nearly an order of magnitude more powerful than their derating curve allows. While they certainly work at first, such abuse guarantees them a short, unhappy life.

COOKING WITH AIR

I set up a final heat-transfer experiment by putting three heatsinks, each dissipating 47 W from two resistors, plus three fans (another 13 W) directing air parallel to the fins, in the oven. Three thermocouples report the temperature at the 6 Ω resistor body, the end of the heatsink, and the air inside the oven.

Assuming the oven's R-value = 5.6, the 154 W = 525 BTU/hr heat load should boost the oven temperature above ambient by:

$$97\,°\text{F} = 525\frac{\text{BTU}}{\text{hr}} \times 5.6\frac{\text{ft}^2 \cdot \text{hr} \cdot \Delta\,°\text{F}}{\text{BTU}} \times \frac{1}{30.3\ \text{ft}^2}$$

Figure 4 shows the results: the oven temperature actually stabilizes somewhat over 50 °C = 90 °F above the (rather chilly) 17 °C ambient air outside the oven.

(Remember to convert the temperature difference correctly!) The target temperature rise was 80 °F above ambient, so there's plenty of heat to spare. It's always good to have slightly more power available than not quite enough: the controller can easily throttle down the heaters.

Homework: Compute the effective R-value for the oven under those conditions.

The temperature differences across the resistor-to-heatsink joint hovers around 7 °C, giving θ = 0.29. That is higher than the 0.2 °C/W measured earlier, because the bottom-side thermocouple position includes the temperature difference across the heatsink. The heatsink-to-air temperature difference remains about 7 °C and θ = 0.14 °C/W.

The temperatures near the right edge of the graph show that the resistor bodies run about 14 °C above the air temperature surrounding the heatsink. That's slightly less than predicted by the previous measurements, but close enough for my purposes, and well within their specifications.

Incidentally, each of those three fans move a nominal 64 ft^3/min with an insanely loud whine. Fortunately for me, they're running inside a sealed box. Unfortunately for them, they'll be near the high end of their 70 °C temperature specification.

CONTACT RELEASE

You'll often see gorgeous three-dimensional thermal models serving as advertisements for computational fluid dynamic simulator programs. Although you may need that level of precision, if not accuracy, for bleeding-edge designs, a few simple calculations and measurements can tell you how close your design comes to that edge.

Now to start work on the oven controller!

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 spreadsheet with experimental data, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/249.

SOURCES

Aluminum thermal grease and heatsink finder
Avvid Thermalloy | www.aavidthermalloy.com/products/options/interface.shtml

www.aavidthermalloy.com/products/extrusion/north american_extrusions.shtml

Tuff-R insulation board
The Dow Chemical Company | http://building.dow.com/na/en/products/insulation/tuffr.htm

Resistors
Vishay-Dale | www.vishay.com/company/brands/dale

Store Closing?
Don't search for Circuit Cellar, let it come to you!

It's not about where you read, it's about **WHAT** you're reading.

Innovative Project Designs!
Superior solutions!
Hot products!

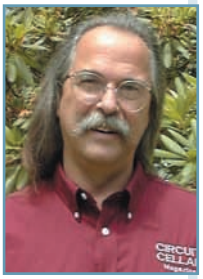
GET IT NOW, DELIVERED STRAIGHT TO YOU.

12¢/day

Use the enclosed subscription card or sign-up at CC-Access.com today!

CIRCUIT CELLAR

by Jeff Bachiochi (USA)



Boot Up

Bootloading Essentials

Flash-based microcontrollers enable upgrades via external media, such as giving a host program access your design. Thus, you have the ability to upgrade virtually any design and enhance its usability. This article details the basics of bootloading, and you learn about a technique that you'll find useful at your workbench.

Flash-based microcontrollers are wonderful. Some of you cavemen may be familiar with their prehistoric predecessors, the one-time programmable (OTP) chip. Or, if your design team back then could afford the quartz-windowed version, you could have UV-erased your parts between code changes. For the rest of us, it was crash (execute a bug) and burn (fix errant code and program a brand new device). What a waste. The long and winding road of incremental improvement has finally reached a point where every manufacturer now has flash-based products. In the past, software upgrades, if at all possible, required sending a product back for service. Socketed microcontrollers at least allowed for easy replacement. Including an in-circuit programming port (IPP) in the design meant you didn't have to pull out the chip to upgrade the software. But, unless the end user had the special hardware necessary for reprogramming the microcontroller, he or she couldn't upgrade the product.

The invention of the PC brought something new to the table. It included some undervalued gateways in the form of serial and parallel ports. While initially devised for peripheral communication, these ports took on a world of their own. Parallel ports had the advantage of byte-wide transmissions and provided direct access to individual I/O bits. Bidirectional serial communication, which required only three wires, became a source for adding all kinds of I/O devices, such as bar code scanners, pointing devices, GPS receivers, and the like.

Product designers immediately started using serial ports to extend their products' lifespans by allowing new features to be added (as well as hidden bugs exterminated). The process was dubbed *bootloading*, which is a word derived from the term *booting*, or the steps a system implements to reach an operational stage. Sometimes, as in a PC, this initial routine is but a small piece of code that lets the system find and load a larger, more powerful chunk, located on some external media. This enables fixed hardware to take on a totally new look, or operating system (OS), by simply swapping the media. On less powerful products, the microcontroller boots or runs initialization code and the application directly, without looking elsewhere for instructions. It has a fixed purpose and cannot

STX	0x0F (not to be confused with ASCII control codes)
ETX	0x04
DLE	0x05
CRCL	Low: 16-bit CRC
CRCH	High: 16-bit CRC
The command byte	
0x00	Read bootloader information
0x01	Read flash memory
0x02	Read CRC of flash memory
0x03	Erase flash memory
0x04	Write flash memory
0x05	Read EEPROM memory
0x06	Write EEPROM memory
0x07	Write configuration fuses
0x08	Branch to application

Table 1—Command bytes and data specific to each byte

be easily upgraded by the user.

Flash-based microcontrollers have opened up the possibility for upgrades via external media (in this case, a host program accessing the product through the serial port). Giving a product this potential is both powerful and dangerous. It requires careful consideration that starts with the initial design.

WHAT'S NEEDED

Let's consider some prerequisites. The product must use a flash memory-based microcontroller and have a serial port. The application can use the serial port or it can just stand by as a means of allowing an upgrade.

Four areas should be addressed in order to provide this bootloading mechanism: a host program, a trigger, the bootloading code, and how this will affect the application. The host program communicates with the bootloader via some communication channel (i.e., the serial port), providing it with the upgrade file. The trigger is a way of telling the bootloader whether to begin the application normally or execute code to begin the upgrading process. The bootloader handles the upgrade process of sending and receiving data to and from the host program. It also handles the process of erasing the old application and writing the new application from and to flash memory. Finally, when the upgrade is complete and verified, it needs to redirect execution to the newly upgraded application.

TRIGGERS

The code for a bootloader depends on the type of trigger that signals the need for taking alternate action. This might come from a hardware device, like holding a normally high input pin low or creating an unusual signal on the serial port, such as a "break." Or, it might be a special software command that is received through the serial device. By design, you can have the bootloader look for this special trigger during power-up or reset and take alternative action. If the trigger is missing within a reasonable time, you should assume normal operation.

When this trigger is an input pin, a timer doesn't need to be involved because you can require the user to hold the input low while turning the power on or resetting the microcontroller. The microcontroller can configure the input and then read its state to determine its next operation.

Many UARTs contain break recognition; however, you can still determine if a break signal has occurred without special hardware based on character overrun (no stop bit detected). A break character is defined as 12 data bits of all zeros. Assuming you are looking for characters at a known data rate, by definition, you must have a stop bit after the agreed upon number of data bits (normally eight). A transmission of 12 zero bits will not provide the necessary stop bit within the required time period, so it will be flagged as an overrun error. Actually, any serial transmission that keeps the data at logic low (RS-232 high) for at least this long will provide an overrun condition. At power-up or reset, the bootloader must configure the UART before the break can be detected. In this case, the bootloader might want to look for a break over some period of time so it won't accidentally miss it.

The hardware triggers mentioned above require the user to take some physical action of the product to initiate recognition of the trigger. If the UART is not used by the application, physically resetting the system might not be necessary. Any received characters (possibly even a special command) could be considered a trigger. In this case, the application must be prepared to monitor the UART for this trigger and take action by executing a branch to the bootloader code. It is dangerous to depend solely on the software trigger from within an application, as any application failure that prevents monitoring the UART will also prevent an upgrade. That's why it's a good idea to implement one of the two previous hardware triggers as a backup.

If the UART is used by the application but uses ASCII data (e.g., command/response protocol), then the software trigger is easily implemented. You might even require some kind of password besides recognizing the correct upgrade command to prevent unauthorized access to the bootloader.

When the serial port is being used in your application to pass binary data, all data values are already in use (legal data values). Therefore, there is no unique "command" that can be sent that would stand out from the data. In this case, a hardware trigger is the only solution.

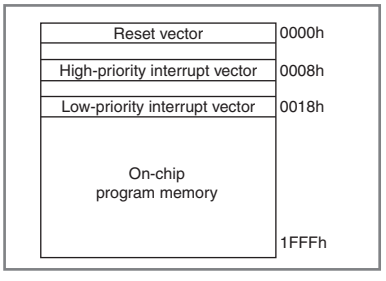


Figure 1—Here is the available on-chip program memory map for a PIC18F2320. The three vectors are entry points for executing the application code (reset vector) and any interrupts (high and low) the designer has chosen to implement.

HOST PROGRAM

You've applied power or done a reset, and because of a hardware trigger the

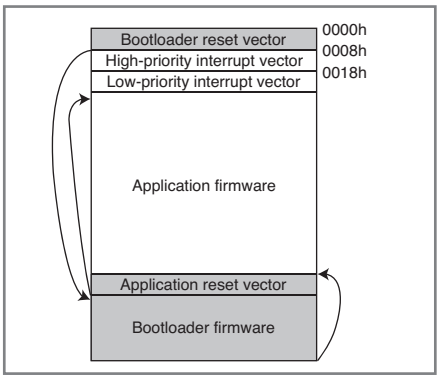


Figure 2—When the bootloader is located at the top of program memory, the host program running on the PC will automatically make some changes to the code being upgraded. It will replace the application's Reset Vector with the Bootloader Reset Vector (so it can take immediate control.) The host program will also place that original Reset Vector at the new location Application Reset Vector (so the bootloader can find the application.)

bootloader code has been redirected into the bootload mode. Or, during the application's execution, it has been redirected by a software trigger back into the bootload mode. How does this bootload mode work?

Building an application—whether it is written in assembler, C language, or a higher language—produces a hex file. The ASCII format of a hex file contains not only the code that will be executed, but also information about where the code will be placed within the microcontroller's flash memory. The host program has two tasks. First, it must be able to interpret the application's hex file. Second, it must be able to instruct the bootloader on what to do with the data it passes via the serial port.

You will find that all of the flash memory microcontroller manufactures have host programs available that support their own protocols for communicating with an associated bootloader, written specifically for their products. While much of the work has been done for you, it is a good idea to understand how it all works since you might need to alter it to make use of a specific triggering method. Your specific manufacturer's implementation may vary, but the basics remain the same. One typical host program uses the following protocol for communications[1]:

```
<STX>,<command>,<data>,….,<CRCL>,
  <CRCH>,<ETX>
```

Table 1 includes the command bytes and the data specific to each command.

This all looks pretty straightforward, right? Well, it's not as simple as it looks for a number of reasons. Flash memory is treated as address blocks. The size of an Erase block is not necessarily the same size as a Write block. For instance, in a Microchip Technology PIC18F2320 device, each Erase block is 64 bytes in length, while a Write block is only 8 bytes. You can read any number of bytes. Prior to writing new data to flash memory, it must be erased. A single Erase block of 64 bytes must be erased even if you're writing a single byte. (You cannot write just a single byte; you must write at least 8 bytes.) This means if you have code filling up a 64-byte Erase block, you must make note of what is stored in the complete block before erasing the block with one Erase block command. You can then make changes and replace the entire 64-byte block by doing eight 8-byte Write block commands.

Assuming the bootloader has been triggered for an upgrade, the host program attempts to make contact with the bootloader by requesting bootloader information (command 0x00). The bootloader responds with its size, version number, command mask, family ID, and starting address in flash program memory space. From this, the host program identifies the part number, its location in the microcontroller's flash program space, and where it ends—or, more importantly, where the normal application code begins. The host program then refers to its small SQL database of parts to find out more about the device. It pulls key parameters from this database, like the aforementioned Write and Erase block sizes, so it knows how to modify its commands to be compatible with the part with which it's communicating. This allows the host program to be used with any of the parts predefined in its database.

With all this specialized knowledge of the part, the host program now prepares a plan of attack to upgrade the microcontroller's present application code to the new code identified by the hex file. The plan includes comparing the old code and the new code within each block to determine if anything has changed. It skips those blocks that have no changes. Blocks are checked by performing a command 0x02 cyclic redundancy check (CRC) on a block. This takes less time than transferring the actual block of data for comparison purposes. The CRC in each command also ensures that communication strings are received without errors.

One of the first things to do in the plan is to clear the first block of application code. This destroys the beginning of the application so it can't be executed by mistake, potentially locking up the system. If the upgrade does not complete for some reason, you don't want to be jumping into an incomplete application. Therefore, the host program begins at the end of flash memory and works its way forward erasing and writing those blocks that require changes. When the first block has been replaced and all CRC values verify the part has been upgraded

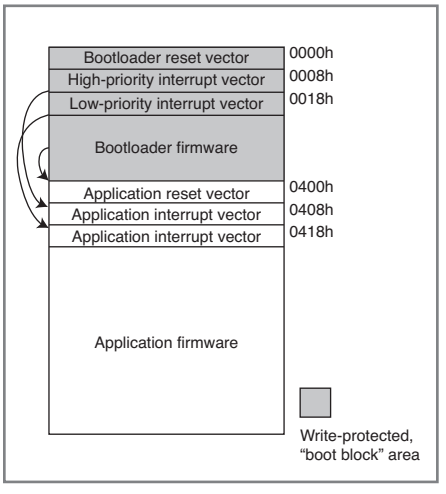


Figure 3—When the bootloader is located at the bottom of program memory, the host program does not need to alter anything. However, the application must have been generated using the new addresses for its three vectors.

NOW, THERE'S EVEN MORE TO DISCOVER.

NEW!

The upgraded Elektor PLUS subscription!

- All 11 issues including the Summer Circuits edition
- Included in your PLUS subscription: Annual DVD 2011
- **50%** cheaper than normal retail price
- Up to **40% discount** on selected Elektor products
- Elektor is delivered to your doorstep every month
- Read your copy before everyone else
- **NEW:** On your personalized Elektor PLUS website, you have permanent access to the three latest editions of the magazine in PDF format, as well as to a fast Elektor search engine!

NEW: exclusive access to www.elektor-plus-usa.com!

Your Elektor PLUS subscription gives you **exclusive access** to the new website **www.elektor-plus-usa.com** where the three latest editions of Elektor magazine are available in PDF files (i.e. the current issue and the two preceding ones). With a simple click you download the complete edition (front to back!) or any single article. **www.elektor-plus-usa.com** also supplies the most extensive Elektor search engine found on the web (all issues since 1998).

Subscribe now
www.elektor.com/usa • Phone 860-875-2199

elektor

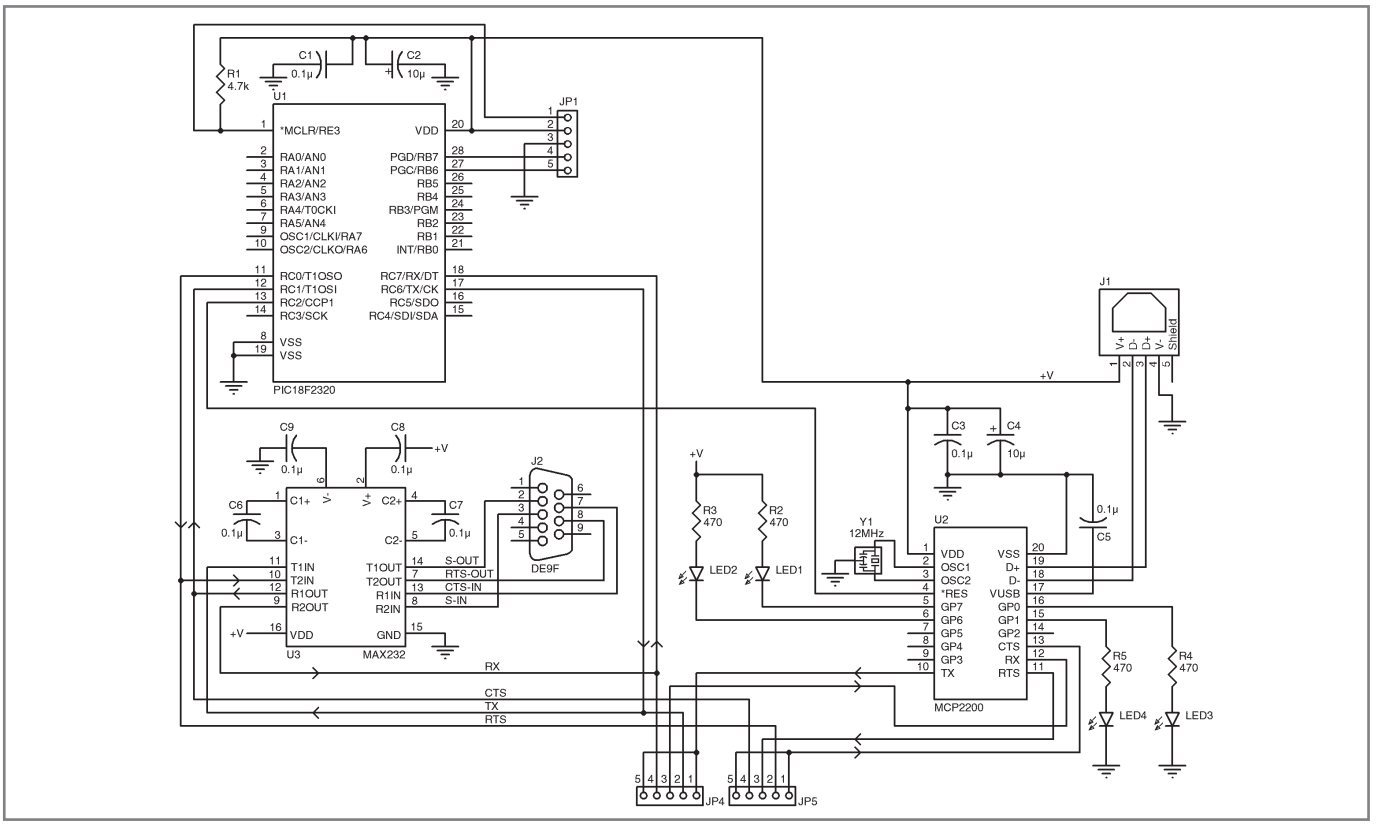


Figure 4—Here is a typical schematic for using a microcontroller's UART as a method for upgrading the application software via a PC. Note the schematic has alternate configurations for RS-232 and USB connections to the PC. If the microcontroller is left out of its socket, a USB-to-RS-232 interface is created. Can you say dongle?

correctly, the host program can use command 0x08 to branch to the application vector, normally the beginning of application code.

BOOTLOADER

For the most part, the act of checking for a trigger at power-up or reset is a simple task. Configuring a pin as a digital input does not require much support code. Initialize a few registers and you're good to go. When the trigger requires the use of a peripheral like a UART, things can get a little more involved. One of the nice things about working with a UART is that it can be used in a simple way without the need for interrupts or setting up data buffers. Receiving a character and updating status flags are all handled by the peripheral. This means little code is necessary to initialize and monitor the UART directly.

With a software trigger, you might need to receive complete command strings, which means setting up data buffers and password authentication. This is certainly moving away from the premise of keeping things simple in the bootloader. The bootloader must handle the process of reprogramming the microcontroller through the serial port without taking up a lot of valuable program space. So, this kind of triggering belongs in the application and not in the bootloader. After all, the main consideration here is to provide a mechanism for a running application to monitor for special upgrade commands and then, after authorization, to let the bootloader handle the actual upgrading process.

The communication protocol identified earlier shows that while the host program (running on a PC) is in charge of the upgrade process, the bootloader code performs the necessary functions of reading, erasing, and writing the flash program address space. It's up to the host program to direct the bootloader properly, as the bootloader is only capable of doing what it's told. However, there are a couple of things the bootloader can do on its own. Like all higher forms of life, it can have the intelligence of self-preservation. That is, it will not perform an Erase of, or Write to, any block it occupies, even if the host program requests it. In addition, its configuration can be protected. This is a little more involved because the configuration is only the last four bytes of the last block. So, the bootloader must allow erasure and writing of this area but prevent changes and put back the original configuration.

I mentioned earlier that the particular PIC microcontroller I was using didn't have EEPROM. What happens if the host program requests a legal function that is unsupported in the device? The bootloader must always return a response to every request. A simple acknowledgement response is used for any function not supported:

```
<STX>,<command>,<CRCL>,<CRCH>,<ETX>
```

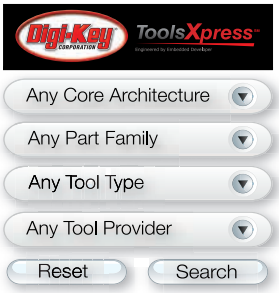
Before going on to how the application is affected by using a bootloader, I want to present two scenarios as to

www.DevtoolsXpress.com

Pick a Tool. Any Tool.

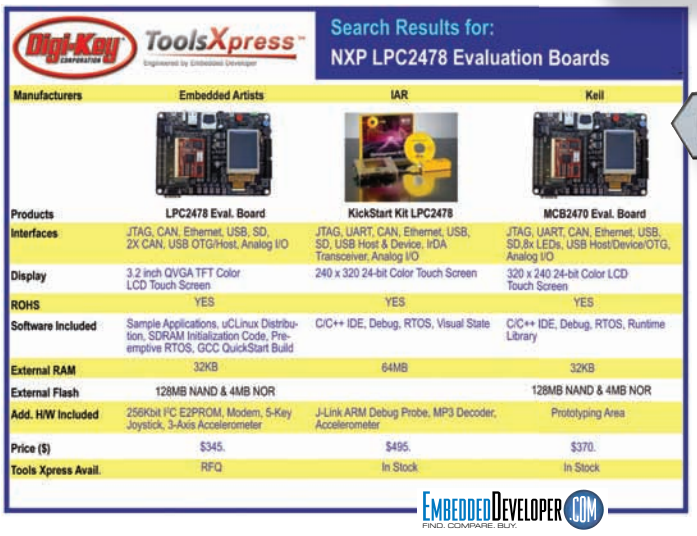
Find the Right Development Tool, Compare it to Other Tools, Evaluate It, and Buy It from Digi-Key Tools Xpress -- Without Leaving Our Site.

The Digi-Key Tools Xpress intuitive research engines are used by engineers worldwide to locate, compare and evaluate hardware or software development tools.



Digi-Key Tools Xpress, engineered by Embedded Developer, is the only site in the industry where engineers can quickly find, compare and buy the leading development tools.

Compare before you buy: tools are listed side-by-side, with features and performance specs, availability, and prices, so you can make an educated decision!



Digi-Key ToolsXpress — Search Results for: NXP LPC2478 Evaluation Boards

Manufacturers	Embedded Artists	IAR	Keil
Products	LPC2478 Eval. Board	KickStart Kit LPC2478	MCB2470 Eval. Board
Interfaces	JTAG, CAN, Ethernet, USB, SD, 2X CAN, USB OTG/Host, Analog I/O	JTAG, UART, CAN, Ethernet, USB, SD, USB Host & Device, IrDA Transceiver, Analog I/O	JTAG, UART, CAN, Ethernet, USB, SD,8x LEDs, USB Host/Device/OTG, Analog I/O
Display	3.2 inch QVGA TFT Color LCD Touch Screen	240 x 320 24-bit Color Touch Screen	320 x 240 24-bit Color LCD Touch Screen
ROHS	YES	YES	YES
Software Included	Sample Applications, uCLinux Distribution, SDRAM Initialization Code, Pre-emptive RTOS, GCC QuickStart Build	C/C++ IDE, Debug, RTOS, Visual State	C/C++ IDE, Debug, RTOS, Runtime Library
External RAM	32KB	64MB	32KB
External Flash	128MB NAND & 4MB NOR		128MB NAND & 4MB NOR
Add. H/W Included	256Kbit I²C E2PROM, Modem, 5-Key Joystick, 3-Axis Accelerometer	J-Link ARM Debug Probe, MP3 Decoder, Accelerometer	Prototyping Area
Price ($)	$345.	$495.	$370.
Tools Xpress Avail.	RFQ	In Stock	In Stock

EmbeddedDeveloper.com FIND. COMPARE. BUY.

Join the thousands of engineers worldwide who use Digi-Key Tools Xpress for their development tool needs.

Digi-Key® CORPORATION

ToolsXpress℠
Engineered by Embedded Developer

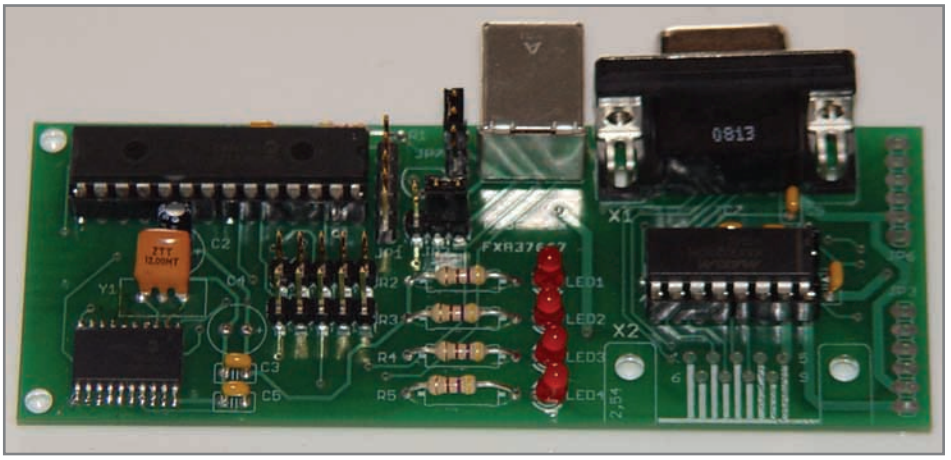


Photo 1—While both the serial and USB interfaces have provisions for data and control lines, only the data lines—TX and RX—need to be implemented in a minimal interface. The serial interface can be used by populating the MAX232 chip. The USB interface can be implemented by choosing the appropriate jumpers on JP4/5.

where the bootloader might be located. At power-up or reset, the microcontroller's program counter is forced to the reset vector. In most microcontrollers, this is normally zero, the very bottom of the flash memory program area. This location holds the first instruction of your application. Without the use of any interrupts, the subsequent locations will hold the remaining portion of your application. If you are using interrupts, then the assembler creates a jump instruction that branches to the actual beginning of your application and around the interrupt vectors. Interrupts have their own permanent vector address as well, a low-priority interrupt vector and a high-priority interrupt vector.

Figure 1 shows where these are located in the PIC18F2320. In this case, the microcontroller executes a jump instruction at the reset vector (0x0000) to the application code placed in the on-chip program memory. The bootloader must now take charge of the reset vector location. It gains control by permanently replacing the jump destination (normally the application address) with its own address. Execution now begins with the bootloader code. Figure 2 shows the structure when the bootloader is placed at the end of the on-chip program space. When the bootloader is placed at the top of flash memory, it places the original jump to the application at the end of the on-chip program space, just below itself. This creates a permanent place for a branch instruction to the beginning of the application (application reset vector).

Figure 3 shows the structure when the bootloader is placed at the beginning of the on-chip program space. In this situation, it places the original jump to the application at a permanent location in the on-chip program space, just above itself. Again, this creates a permanent place for a branch instruction to the beginning of the application (application reset vector).

Both structures have their advantages and disadvantages. Placing the bootloader at the top of flash memory means that the application can be written as normal (i.e., beginning at the reset vector). However, the interrupt vectors are controlled by the application. Placing the bootloader at the bottom of flash memory means the application must be located (ORGed) somewhere other than 0x0000. The interrupt vectors are now under bootloader control and must be remapped for use with the application. Let's look at how this affects the application.

APPLICATION

You can think of the bootloader and the application as two entirely

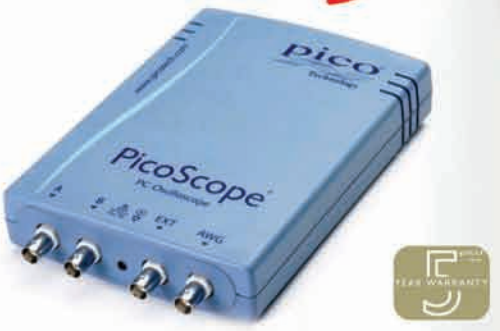
Tired of just getting the bare bones in your test equipment?

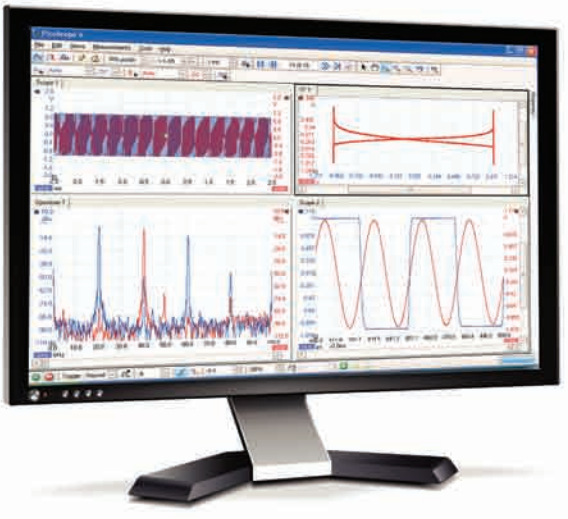
Check out the NEW

PicoScope 3000 Series...

All of the Extras
without breaking the budget!

- 200 MHz bandwidth
- 128 MS buffer size
- Spectrum analyzer
- Function generator/AWG
- USB-connected and powered
- Serial Decoding
- Mask Limit Testing
- Rapid Trigger / Segmented Memory
- 10,000 waveform buffer
- Unlimited Maths
- Advanced Digital Triggers





It's all in there + more.

From Pico Technology, the leader in PC oscilloscopes, the all-new 3000 series are the highest performance USB powered oscilloscopes on the market. Designed to be versatile and adaptive these multifunctional instruments come complete with the extras and without compromise.

"Small but Smart - I like that!"

For more information give us a call at 1-800-591-2796
or visit us online at www.picotech.com/pco446

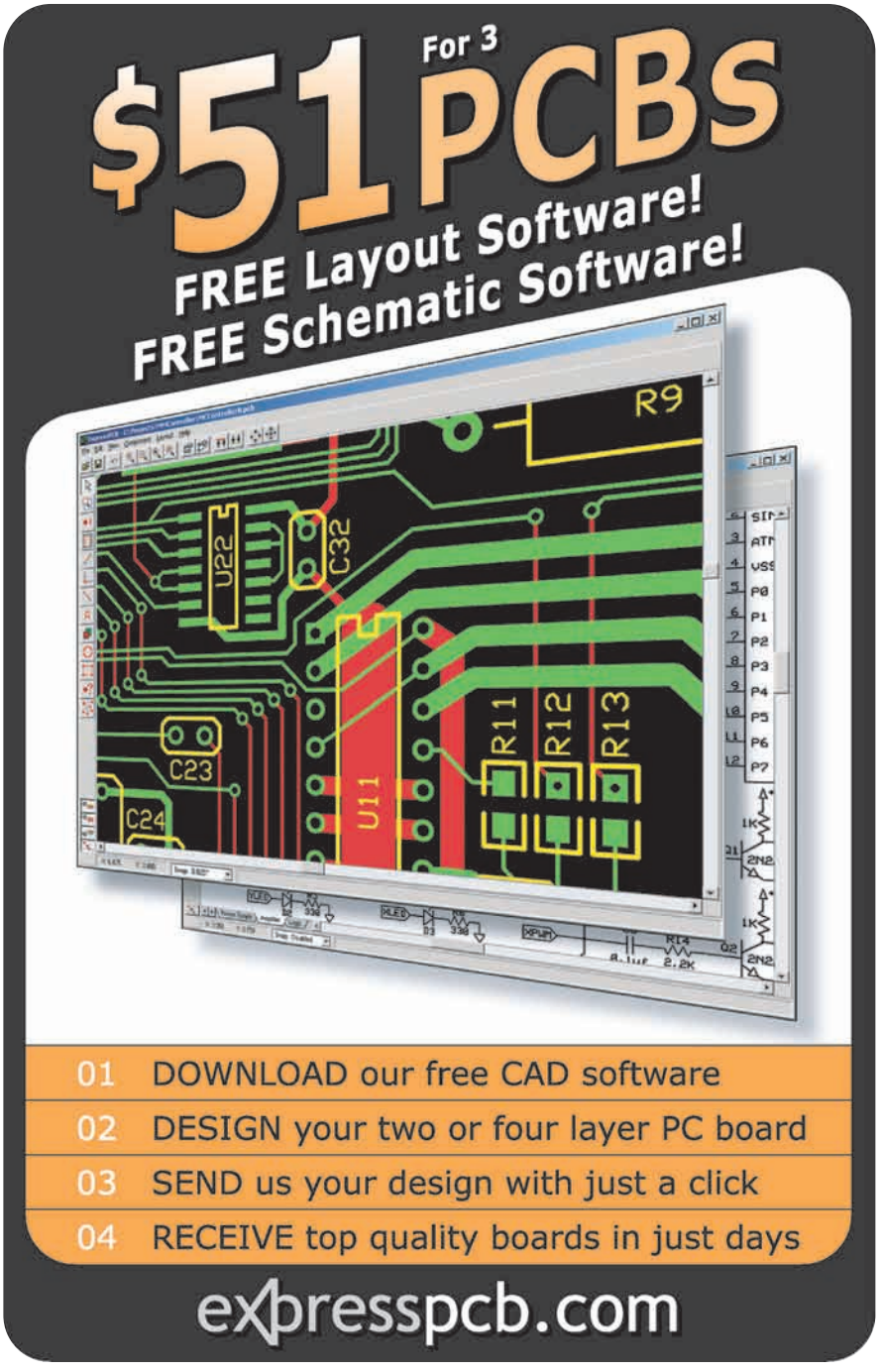
pico® Technology

separate programs. While they each require their own program space, since only one is in control at a time, they can share the same data memory and peripherals. When you write an application there are generally no restrictions on where it will reside. The assembler will organize your application code to begin at the normal reset vector. A power on or reset will therefore begin directly executing your application code. Referring back to Figure 2, a bootloader placed at the top of flash program memory will not interfere with the application, except that the maximum size of the application has now been reduced by the size of the bootloader. There is no change in the way the application handles the interrupt vectors. This will eliminate the possibility of using these from within the bootloader, as it has no control over them (unless you require some special coding of these in every application you write, which kind of defeats the cleanliness of this approach). If the host program sees the bootloader is high, it will make two necessary changes. First, it replaces the reset vector to force a jump to the bootloader code (to check for a trigger). And second, it places a jump to the application code just below the bootloader, so that the bootloader knows how to get to the application code.

If you decide to place the bootloader code at the bottom of flash program memory space, the application code must be handled a bit differently than normal. Refer back to Figure 3. In this instance, the bootloader will already have control over the reset vector so the host program doesn't need to make any changes to this. The interrupt vectors are in control of the bootloader, so use of these by the bootloader is possible. However, how can your application use these?

Instead of placing a return from interrupt (`RETFIE`) at the end of any interrupt code, the bootloader uses a jump to a predefined location (where the application is located), which allows the application to have access to the execution flow of each interrupt vector.

Let's look at this a little more closely. The length of the bootload code is known. Let's say it's 800 instructions long, which is 0x0320. You could start the application at address 0x0321, but it makes sense to use a nice even address based on block size, like 0x0400. Normal applications, and in this case the bootloader, use the three vectors: reset (0x0000), high priority (0x0008), and low priority (0x0018). You've determined that the application now begins at an application vector of 0x0400. In keeping with the original offsets for the interrupts, that would make the high-priority vector at 0x0408 and the low-priority vector at 0x0418. The low bootloader now knows that it can branch to the application at 0x0400 and uses branches to 0x0408 and 0x0418 at the end of its interrupt code. This means that every application written for use with this bootloader must use new addresses for the three associated vectors, an application reset vector at 0x0400, an application high-priority interrupt vector at 0x0408, and an application low-priority



interrupt vector at 0x0418. In addition, it must place a RETFIE instruction at each application interrupt vector, even if these are not used. This is necessary to complete the interrupt code started by the bootloader.

THE VANISHING SERIAL PORT

It's true. Computers are now made without serial ports—well, at least the serial ports as defined by the DB25 and DE9, which we have come to recognize as serial connectors. USB has totally replaced these. But I'll bet you have at least one USB serial (or parallel) dongle that you can plug into your USB slot to get back that serial port that just won't go away. So, adding a serial port to your product isn't going to make it a dinosaur! But wait, this column comes at a very opportune time. Microchip just released an inexpensive, easy-to-use, serial-to-USB IC that enables you to add USB capability to your UART-based microcontroller. And, it works perfectly with this bootloader and supporting host program.

Refer to Figure 4 and Photo 1. The schematic shows the serial-based circuitry and the USB-based circuitry connected to a microcontroller. Note that the cost in parts for the serial solution might actually be slightly more than the USB solution. And, the USB solution requires less real estate. Although I didn't use it for hand assembly reasons, the USB IC is available in a 5 mm × 5 mm 20-pin quad flat no-leads (QFN) package.

You may remember from a past column that I worked on an assembly translation of USB support for PICs that have an internal USB peripheral. First, let me say that those of you who use C have access to bootload code through Microchip's application libraries and those supported by most other manufacturers. Those of you who have been following my rantings over the years know that I have a nitty-gritty, need-to-know (some might even say "sadistic") desire to use only assembly language. Deciding to use an on-board USB peripheral changes the complexity of the bootloader code drastically. While support for the actual programming of the device remains constant, the support for the USB peripheral makes the bootloader size increase dramatically. In this case, the assembly code I reworked to make use of this bootload scheme increased by a factor of four. Now, instead of requiring an offset of 0x0400 for the application, it must be 0x1000. So, at least in small applications where it might seem like using a microcontroller with integrated USB peripheral could save money, the larger (and more complicated) bootloader might end up reducing the amount of room you have for the actual application.

UPGRADES POSSIBLE

You can use the basic bootloading technique I described to upgrade virtually any device or product. And having the ability to do so can lead to more profit when you're designing for your employer or client. Many of the applications you run on your PC have the ability to check for upgrades, download, and install themselves with little user effort. This is all thanks to an Internet connection. It only makes sense to provide your products with the same capabilities.

WHO KNOWS?

While Intel might own the prize for the first microprocessor in the 1970s, it was a quite a few years before peripherals were added along with a microprocessor in the same package. Initially, these microcontrollers had ROM as well. PROM (OTP) enabled manufacturers to one-time-program the parts as needed, while EPROM (windowed) UV-erasable parts were expensive and therefore used only for developing code. Who knows when and who introduced the first flash memory-based microcontroller?

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

PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/249.

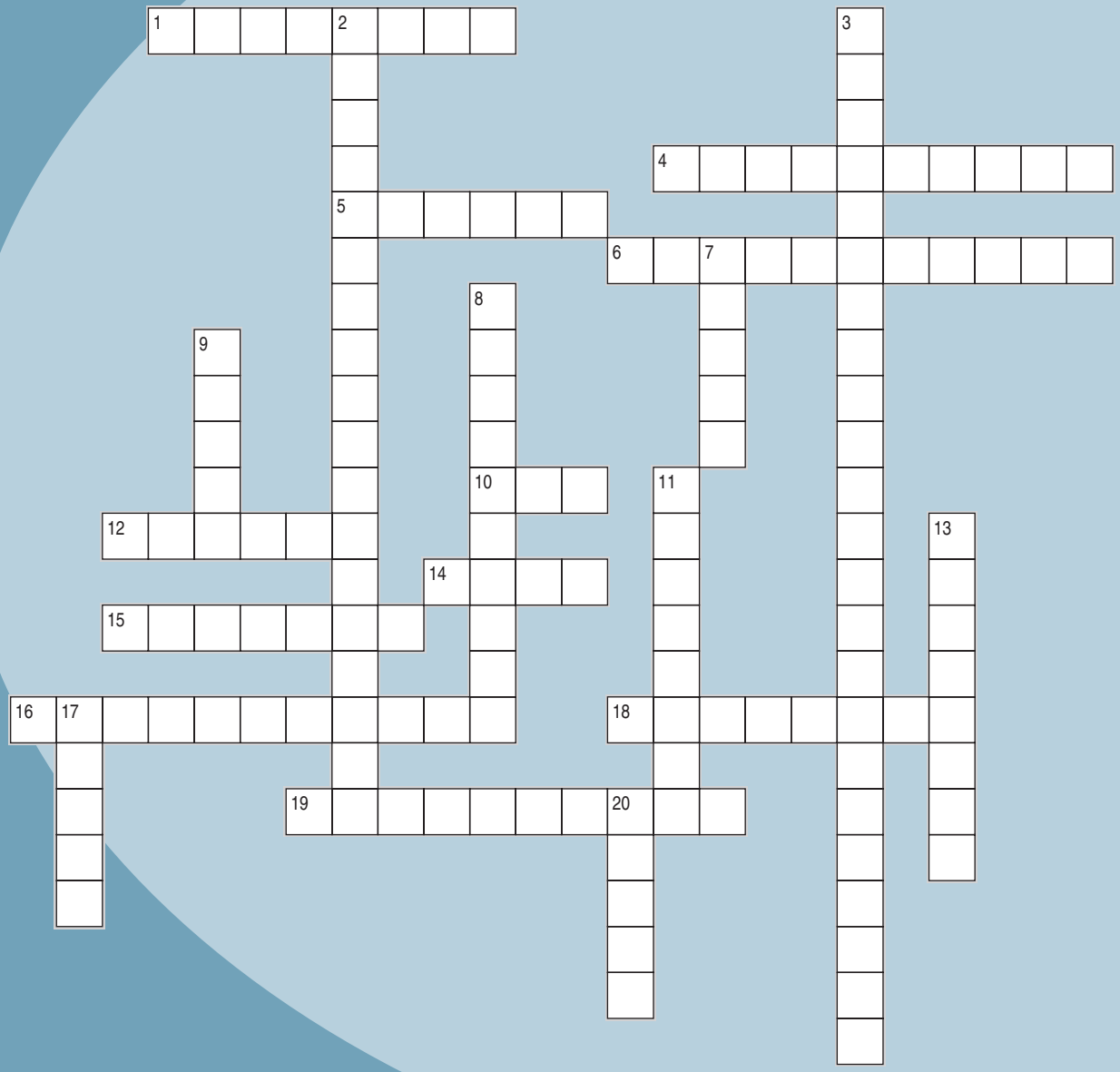
REFERENCE

[1] E. Schlunder, "High-Speed Serial Bootloader for PIC16 and PIC18 Devices," AN1310, Microchip Technology, 2010.

SOURCES

MCP2200 USB 2.0-to-UART Protocol converter
Microchip Technology, Inc. | www.microchip.com

CROSSWORD



Across

1. Makes direct connections [two words]
4. Social networking guru
5. A group of messages
6. Used to enclose text
10. Protocol allowing the exchange of data between networked devices
12. Between J and L
14. One way a computer reads data
15. Data, a series of lines [two words]
16. Another word for main board or system board
18. Creator of Unix-type operating systems
19. Permits current forward and backward

Down

2. Used to brighten things [three words]
3. Used to weave the Web [four words]
7. Mechanical, not real
8. 10^{-9} s
9. Common text message: "Rolling on the floor laughing"
11. Required to login
13. Creates fluidity, such as liquid or air
17. Ω
20. Used to build electronic circuits [two words]

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

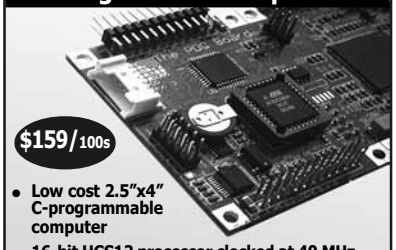
IDEA BOX

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, PO Box 180, Vernon, CT 06066. For current rates, deadlines, and more information contact Peter Wostrel at 800.454.3741, 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.

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

bob-4

Low-cost video data overlay module

Features:

Automatic text scroll and crawl
'TTL-232' and SPI control ports
Any size for custom fonts
Vector and bitmap graphics

NEW REDUCED PRICING

www.decadenet.com

DECADE ENGINEERING
503-743-3194 Turner, OR, USA

Embedded Ethernet

ONLY $98

- 10Base-T Ethernet
- 186 Processor @ 40 MHz
- DOS w/ Flash File System
- 16 Digital I/O • 5V DC
- Console / Debug Port
- Hardware Clock/Calendar
- Socket for DiskOnChip
- 512K DRAM & 512K Flash
- (2) Serial Ports
- (2) 16-bit Timers
- Watchdog Timer
- 3.75" x 2.50"

NEW! picoFlash

$129 Development System

Development Kit Includes:
- picoFlash Controller
- Borland C/C++ 4.52
- TCP/IP, PPP & Web Server
- Serial Driver Library
- AC Adapter and Cables

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

JK microsystems

Evaluate * Educate * Embed

Adapt9S12
Modular Prototyping System
For education & development

freescale Alliance Member

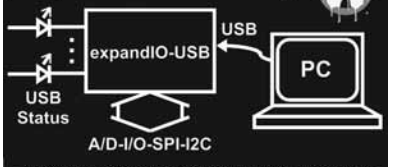
Topologies Supported:
Planar, Stack, Backplane, Tower

MCU Families Supported:
S12A, C, D, E, NE, XD, XE, XS

Toll-free (USA & Canada):
1-877-963-8996

www.TechnologicalArts.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

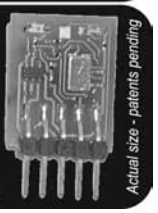
www.hexwax.com - Buy from Mouser & Farnell

Amazing PIC programmer

Most devices supported
ICSP, SQTP, & copy limits

$32 at Digikey & Mouser

www.flexipanel.com



Actual size - patents pending

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

Optimized C Compiler for PIC® MCU
· Decrease code size by 40%
· Use less expensive MCU with smaller program memory
· Save hundreds of dollars in component costs in your projects

CCS Compilers Save You Money

Compilers Starting at $150

CCS
www.ccsinfo.com/SAVE
sales@ccsinfo.com
262.522.6500 x35

ANALOG and DIGITAL for PC/104

GPIO-104
8 ANALOG INPUTS
4 ANALOG OUTPUTS
24 DIGITAL I/O

Need More Analog I/O?

ADIO-104
16 ANALOG INPUTS
8 ANALOG OUTPUTS
24 DIGITAL I/O
PULSE ACCUMULATOR

SCIDYNE®
1-877-SCIDYNE
info@scidyne.com
www.scidyne.com

Also Available
•96 DIGITAL I/O
•24 HIGH POWER I/O
•RELAY/OPTO I/O
•EXTERNAL EXPANSION
•PC/104 PROTOTYPING

CIRCUIT CELLAR®
CC-ACCESS
SUBSCRIBERS RESERVED AREA

Sign up for a CC 2.0 subscription today and receive automatic entry to CC-Access!

- Gain instant access to source code & project files
- Search for your favorite articles at supersonic speeds
- Quickly download your issue PDF when you are a CC 2.0 Digital or Combo subscriber
- Easily maintain account information
- Receive VIP treatment with special offers on products & services you can't live without!

All this and more at...
www.CC-Access.com
(Already a subscriber? Log in now!)

Add USB to Your Designs

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

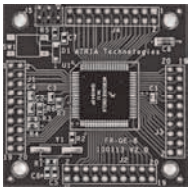
USB Complete
The Developer's Guide
Fourth Edition
Jan Axelson

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

Disk On Chip
Ready for Delivery

From 16MB to 128MB Available!
Call 530-297-6073 Sales@jkmicro.com
www.jkmicro.com

JK microsystems, Inc.
International Orders Welcome



Microcontroller Modules
- 8 & 32 bit
- Freescale Flexis
- MC9S08 series
- MCF51 series

Communication Modules
- Bluetooth
- RS-232
- USB

I/O Modules
- 2x20 LCD
- Keypad

GREAT For:
- Projects
- Breadboards
- Experiments
- Labs

Kits Available

ORDER NOW

BASIC On Board Eliminates Development Tools

ATRIA Technologies Inc.
www.AtriaTechnologies.com

MaxSonar
Perfect for OEMs & Engineers

XL-MaxSonar-EZ
• High acoustic power
• Real-time calibration
• Tiny size
• $39.95 USD qty. 1

MaxSonar-WR (IP67)
• Industrial packaging
• Weather resistant
• Standard ¾" fitting
• $99.95 USD qty. 1

LV-MaxSonar-EZ
• Power-up calibration
• Smallest MaxSonar
• Low power, 2.5V-5.5V
• $29.95 USD qty. 1

MaxSonar-WRC (IP67)
• Compact packaging
• Weather resistant
• Quality narrow beam
• $99.95 USD qty. 1

www.maxbotix.com

Add a Touch Screen to Your Embedded Product

• No special OS or Library Required.
• Programming GUI is Simple.
• Development Kit = Up and Running in Days.

Learn more at www.reachtech.com, call 510-770-1417 or email info@reachtech.com.

REACH TECHNOLOGY INC.

See us in booth 1245 at ESC San Jose.

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

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

www.earthlcd.com
Call for Custom Display Configurations

Exactly why is USB hard?

Part# QUSB2 - $149 qty 1

* 25MB/sec data bus, bit-bang, RS-232, I2C Master, SPI Master, FPGA port

* Software library and example code

* New, high-performance kernel drivers for Windows, Linux and MacOS

QuickUSB®
www.quickusb.com

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 ttl company
amazon.com amazon.co.uk

I²C™/SMBus

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

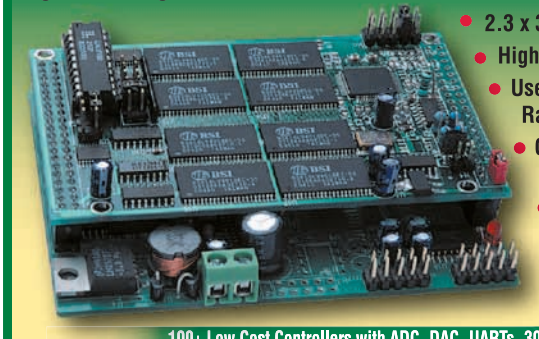
MCC
Micro Computer Control

I²C is a trademark of Philips Corporation

www.mcc-us.com

GR16™ Low Cost, High Speed, Analog Signal Grabber — Up to 10MHz SPS, 16-bit ADC, and 8 MB FIFO

Start at $149



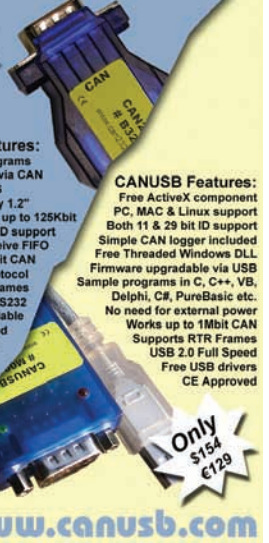
• 2.3 x 3.6 x 0.5 inches
• High speed 16-bit ADC (ADS1610)
• User Programmable Sample Rate, up to 10M SPS
• Configurable FIFO data size, up to 8 MB.
• Stackable multi units sharing Sync, and Trigger
• Driven by 16-bit TERN controllers

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

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

www.can232.com

Only $108 €89



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

CAN232 Features:
Free sample programs
8-15VDC supply via CAN
Timestamp in mS
Small size 2.7" by 1.2"
100% Bandwidth up to 125Kbit
Both 11 & 29 bit ID support
32 Message Receive FIFO
Works up to 1Mbit CAN
Simple ASCII protocol
Supports RTR Frames
Max 230Kbaud RS232
Firmware upgradable
No drivers needed
OS independant
CE Approved

CANUSB Features:
Free ActiveX component
PC, MAC & Linux support
Both 11 & 29 bit ID support
Simple CAN logger included
Free Threaded Windows DLL
Firmware upgradable via USB
Sample programs in C, C++, VB, Delphi, C#, PureBasic etc.
No need for external power
Works up to 1Mbit CAN
Supports RTR Frames
USB 2.0 Full Speed
Free USB drivers
CE Approved

Only $154 €129

www.canusb.com

UART-WiFi, UHF EPC Gen2 RFID Reader, Powerline Communication

Long Range UHF RFID reader (6 meters, 12 meters, and 25 meters) (ISO18000-6C EPC G2)

TTL UART to WiFi

JPEG Serial Camera

Single Board Computer with Win CE 6.0, Linux, and Android. Optional on-board Cellular, WiFi, GPS

- UART to Power line Communication Modules
- Bluetooth modules
- Cellular modules

www.linksprite.com

Control by USB or Web!

TCPmaker for control over the Web

Easy as 1-2-3:

1. **Define Your Data**, with variable names that YOU create.
2. **Lay Out Your Content** (with gorgeous web-ready screen controls that you can ***grab on to***) using TCPmaker's drag & drop Visual Page Designer.
3. **Generate Your Code**, for all Microchip C compilers, to "wire it all together."

NO PC PROGRAMMING AT ALL – just point your web browser at your device!

Simple Upgrade Management System

Low cost integrated system manages upgrades across different processors & connectivity types:

> Lets your end users upgrade your PIC firmware safely, simply, and securely.
> **Automatically delivers upgrade by email.**
> Simple for non-technical end users.
> **Encrypted system** protects your firmware from theft or from being programmed into the wrong device.
> **Grows with your product line:** for a new product w/ different processor or connectivity type, just add another SUMS bootloader.

Trace SYSTEMS, Inc.

From the makers of HIDmaker FS:

www.tracesystemsinc.com

888-474-1041

microEngineering Labs, Inc.

www.melabs.com 888-316-1753

PICBASIC PRO™ Compiler

Bridges the gap between ease of use and professional level results. The simplicity of the language allows both hobbyist and engineer to master it quickly, but the engineer appreciates what's happening under the hood. This is a true compiler that produces fast, optimized machine code that's stable and dependable.

Supports:
More than 300 PIC® Microcontrollers
Direct Access to Internal Registers
In-Line Assembly Language
Interrupts in PICBASIC and Assembly
Built-In USB, I2C, RS-232 and More
Source Level Debugging in MPLAB

PICBASIC PRO™ Compiler $249.95

USB Programmer for PIC® MCUs

$89.95 *(as shown)*

RoHS Compliant

Programs PIC MCUs including low-voltage (3.3V) devices. Includes Software for Windows 98, Me, NT, XP, and Vista.

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

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

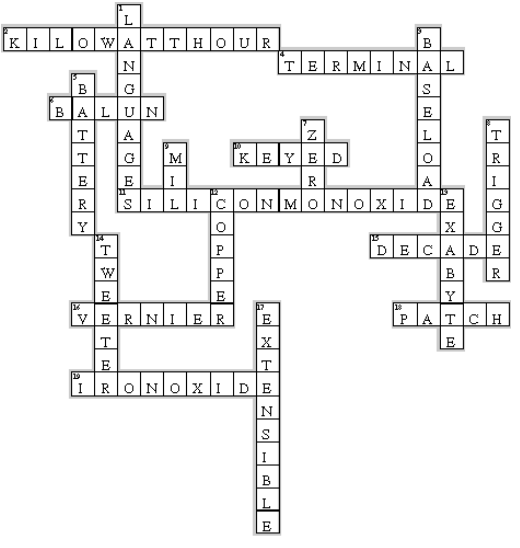
CROSSWORD ANSWERS from Issue 248

Down

1. **LANGUAGES**—Tcl, Lisp, Fortran
3. **BASELOAD**—Minimum power delivered
5. **BATTERY**—CR2032 is a standard
7. **ZERO**—Empty set size
8. **TRIGGER**—Schmitt
9. **MIL**—0.001″
12. **COPPER**—Element in a "ground island"
13. **EXABYTE**—1,024 Petabytes
14. **TWEETER**—Speaker for high-frequency sounds
17. **EXTENSIBLE**—X

Across

2. **KILOWATTHOUR**—1,000 Watthours [two words]
4. **TERMINAL**—Where electricity enters/exits a device
6. **BALUN**—Balanced/unbalanced
10. **KEYED**—OOK is on/off what?
11. **SILICONMONOXIDE**—SiO [two words]
15. **DECADE**—0 to 9, Counter
16. **VERNIER**—Measurement, Calibration, used with main scale
18. **PATCH**—Quick fix
19. **IRONOXIDE**—Rust [two words]



INDEX OF ADVERTISERS

PREVIEW of May Issue 250

Theme: Measurement & Sensors

Virtual Sensing: Build a Virtual Alarm System

Control Shaft Encoders

Reprogrammable UAV Autopilot System (Part 2): Software and Testing

Microprogramming a CPU (Part 2): A Look at the Meta-Assembler

Stereo Spectrum Analyzer

THE CONSUMMATE ENGINEER **Functional Upset**

LESSONS FROM THE TRENCHES **Design Development (Part 1):** Planning and Tools

FROM THE BENCH **Error Checking**

ATTENTION ADVERTISERS

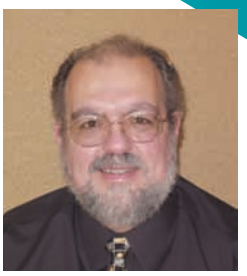
June Issue 251 Deadlines

Space Close: Apr. 13
Materials Close: Apr. 20

Theme
Communications

Call Peter Wostrel
now to reserve your space!
800.454.3741 or 978.281.7708
e-mail: peter@smmarketing.us

PRIORITY INTERRUPT



by Steve Ciarcia, Founder and Editorial Director

A Guess to Come

I predict that some day you won't have to read this dribble anymore. No, seriously, it seems that while I often have much to say about certain technologies, I don't remember ever just making general predictions about a bunch of concepts or products and their futures. I suppose that utterly failing to read *all the signs correctly* in the past has deterred me from speculating in public; but, then again, I must have done a few things right to get where I am. So, here goes, and don't shoot the messenger. ;-)

The Cloud: As much as I love the sound and smell of humongous quantities of computer hardware, there will be a vast metamorphosis in business away from owning physical IT hardware and instead leasing software and services "in the cloud." A cloud-based OS will arise that, if not from Microsoft, will mean certain end to the MS monopoly. However, without some content standardization, the rapid commercial adoption of the cloud will linger. In the end, everything will be browser-based and you will use a single logon (and, more importantly, a single subscription) for all of your services and devices. And remember the foreign call center support you hated in the past? It will be replaced by new "accent-less" e-mail support from the same people abroad. ;-)

Google vs. Facebook: Google seems to have a scattergun approach to business products and may ultimately lose the war. The fact that 99% of the planet is on Facebook and I'm not suggests that Facebook's future success is absolutely ensured. When I'm wrong, I'm always dead wrong! There is an opportunity to move rapidly up the corporate ladder for "quick learners" (obviously not me) who understand the enterprise value of using social networks to engage an audience about the company's message, products, and services.

Mobile Computing: While I'm not sure about the exact format, mobile devices will continue to expand their processing abilities and bandwidth. They'll ultimately replace PCs as the *most common route* for web access. Any business that still has a large-format homepage with a gigabyte of graphics isn't going to make it in a new small-screen format world. All computing devices will run some rendition of "apps," and more commercial enterprises will use smartphone near-field communication (like RFID) to make purchases and financial transactions. Android sales will outpace iPhone, but comparisons are irrelevant. Android's introduction put smartphone development and competition on steroids, but it doesn't ensure the same quality among all the products sharing the brand. A bit of "buyer beware" consciousness is needed when purchasing mobile devices: anything is wise, at least in the short term.

The Tablet Wars: The iPad and its future variants may end up dictating the market. iPad's avalanche of success was primarily about perfect timing. E-readers and netbooks had introduced the concept of portable entertainment/computing/presentation appliances, but they were too narrowly focused for majority interest. With the addition of location, motion, video, and a variety of other contextual inputs, tablets will dominate this market segment as netbooks go extinct and low-end generic notebooks become endangered species. Along with the world's evolving understanding of "apps," the tablet will become the new shared computer that anyone can pick up and use (as opposed to your private notebook or smartphone).

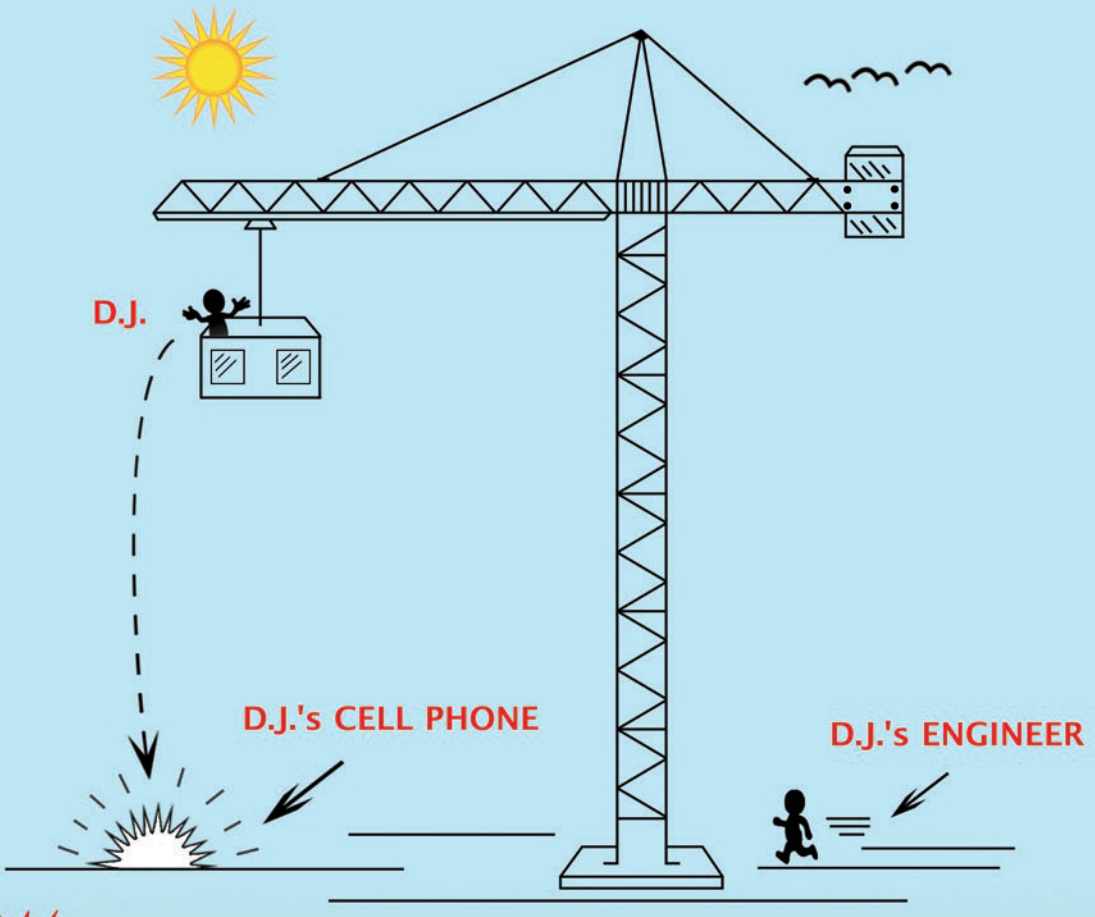
Ubiquitous Computing: Let's face it. You're already using a lot of computers. Your phone, your transportation, your work, and your entertainment all depend on them. I can't say how soon, but the next generation of implementation will involve putting processors into everything else that you wouldn't ordinarily think about. Clothes that automatically set their washing cycles, TV dinners that set their microwave cooking times, business cards that auto-update a recipient's database, etc. Welcome to a completely networked life.

And a few final quick observations: the disk drives in your next laptop will be flash; the DSLR you covet will be superseded by a cheaper compact camera with an interchangeable lens; you can toss your present e-readers because soon they should be color (before becoming extinct); basic cell phones will continue to exist along with high-end 3G and 4G smartphones; the Apple zombies will continue to worship the fruit and won't buy other brands (no matter what); engineering a new IP product with only IPv4 addressing will be career suicide; mobile device addiction will become a valid concern; Facebook will buy Skype; and web-connected TV will gain momentum.

So, that's it. I realize that some of these predictions are probably dead right and others are possibly total misses. The gambler's creed is to always bet on the winners. I trust you'll have better luck picking winners than I've had all these years. ;-)

steve.ciarcia@circuitcellar.com

Can You Solve This?



Sign up for our e-newsletter and get a FREE comic book.
www.Jameco.com/adventure1

Radio disk jockey D.J. Smith organized a stunt to raise donations for a charity by conducting his radio show for one week while suspended from a 150-feet tall crane. A cell phone with a solar charger served as D.J.'s only link to the radio station. His little home in the sky included foil-wrapped freeze-dried meals, bottled water, binoculars, pencil, notebook, paper clips, rubber bands, pocket knife, MP3 player, and a portable radio. The weather cooperated with light wind, clear skies and pleasant temperatures. All went well untill D.J. dropped his cell phone to the concrete below. The stunt rules prevented any physical contact between D.J. and the ground. How did he continue his show? (hint: D.J.'s engineer was an Alexander Graham Bell history buff). Go to www.Jameco.com/teaser10 to see if you are correct. While you are there, sign up for their catalog. Forrest M. Mims III

1-800-831-4242 | www.Jameco.com

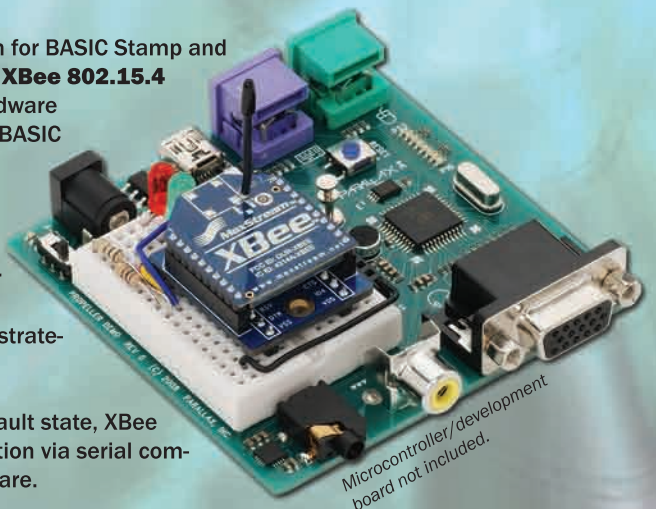
JAMECO® ELECTRONICS

XBEE STARTER KIT

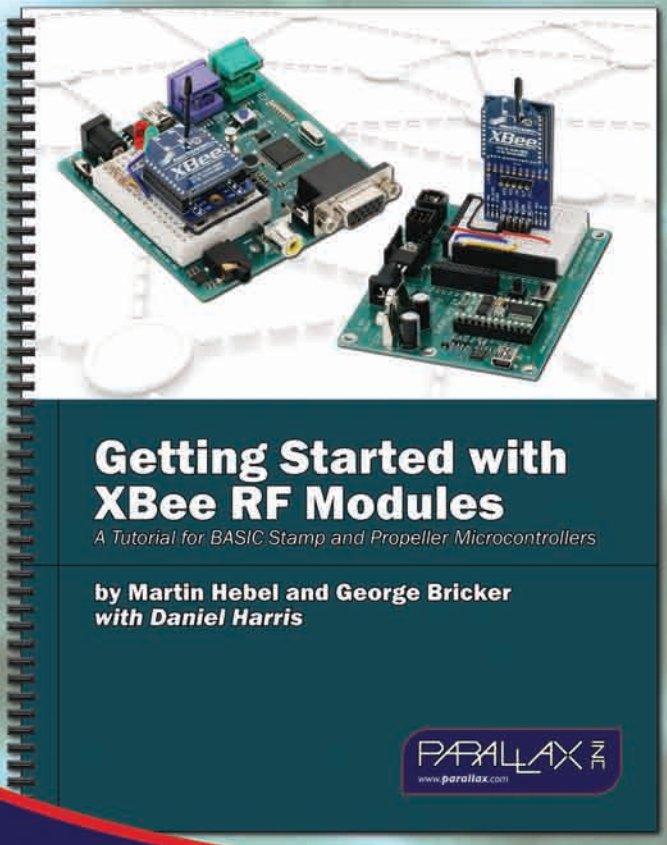
XBee modules provide an incredibly versatile RF communication solution for BASIC Stamp and Propeller microcontroller projects, and this kit helps you get started. The **XBee 802.15.4 Starter Kit (#32450; $179.99)** includes the printed tutorial and the hardware required to complete the activities in Chapters 1 through 6, on your own BASIC Stamp or Propeller microcontroller development boards (not included).

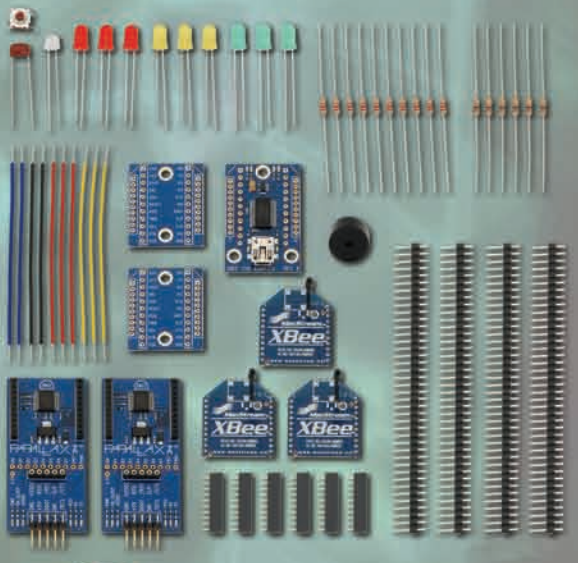
The tutorial introduces basic and advanced setups with XBee 802.15.4 projects you can do on your desktop. It features both BASIC Stamp (PBASIC) and Propeller (Spin) code examples, allowing you to see similar code running in both of these Parallax microcontrollers. Subjects include XBee testing and configuration, network topologies and communication strategies, and using XBee Reset, Sleep, and API modes.

Ready to use for bi-directional, point-to-point communication in their default state, XBee modules can also support complex networking with additional configuration via serial commands sent from your microcontroller, or through Digi's free X-CTU software.



Microcontroller/development board not included.





Note: soldering is required to assemble the XBee Adapter Board and XBee USB Adapter Board.

$179.99

Order the XBee 802.15.4 Starter Kit (#32450; $179.99) at www.parallax.com or call Sales toll-free at 888-512-1024 (M-F, 7 AM - 5 PM, PDT).

Friendly microcontrollers, legendary resources.™

Prices subject to change without notice. BASIC Stamp is a registered trademark of Parallax Inc. Propeller is a trademark of Parallax Inc. XBee is a registered trademark of Digi International Inc.

PARALLAX INC

www.parallax.com

"ParallaxInc" on Twitter, Facebook, and YouTube