TESTING: PWM Voltage Reference Technique
LOCATION: United States
PAGE: 18

DESIGN TIP: Protect Against EMI
LOCATION: Canada
PAGE: 46

Q & A: Embedded Theory & Practice
LOCATION: Israel
PAGE: 50

CIRCUIT CELLAR

THE WORLD'S SOURCE f OR EMBEDDED ELECTRONICS ENGINEERING INf ORMATION

NOVEMBER 2011
ISSUE 256

ANALOG TECHNIQUES

Analog Signal Management & Sound Tone Detection

MCU-Based Auditory Navigation System

Engineer an Alternative to Joystick Control

Electronics Design: Processors, Power, & Interfacing

A Look Inside an Ionization Detector

PLUS

"Spark" a Design

Implement 32-Bit Power & Arduino Compatibility

// Get Started with the chipKIT
// Program with a Modified IDE
// Shields Explained
// And More

7 25274 75349 9
$7.50 U.S. ($8.50 Canada)

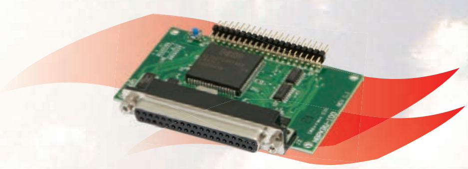
www.circuitcellar.com

The easiest finished product you will ever create

Start with the PK70 EX

+

Design your own, or use a NetBurner Personality Blade



=

Add your label to get a finished product

Personality Blades

Prototype Blade *(P/N NBPKBP-100CR)*........ **$25 ea.**
Create your own Personality Blade

CPLD Digital I/O Blade *(P/N NBPKD-100CR)*........ **$59 ea.**
Programmable Xilinx CPLD
32 digital I/O

Multi-I/O Blade *(P/N NBPKM-100CR)*........ **$99 ea.**
Eight 12-bit analog-to-digital inputs
Two 16-bit digital-to-analog outputs
16 digital I/O

FPGA Digital I/O Blade *(P/N NBPKX500-100CR)*........ **$169 ea.**
Programmable Xilinx Spartan 3E FPGA
High-density 62-pin connector

NTP Server with GPS Blade *(P/N NBPKX500-100CR)*........ **$99 ea.**
Network Time Protocol (NTP) server
Precision GPS time reference

Quad UART Blade: RS-232 Version
(P/N NBPKBU-232CR)........ **$99 ea.**
Network enable up to 4 RS-232 serial devices

Quad UART Blade: RS-422/485 Version
(P/N NBPKBU-485CR)........ **$119 ea.**
Network enable up to 4 RS-422/485 serial devices

Quad UART Blade: RS-232/422/485 Version
(P/N NBPKBU-MMSCR)........ **$245 ea.**
Network enable up to 4 RS-232/422/485 serial devices

PK70 EX & Development Kit

PK70 EX *(P/N PK70EX-100IR)*........ **$199 ea.**
32-Bit Freescale ColdFire 147 MHz CPU
8MB SDRAM • 4MB Flash Memory
SD/MMC Flash Card Support • I2C • SPI • Address and Data Bus
10/100 Ethernet • 2 UARTs

PK70 EX Development Kit
(P/N NNDK-PK70EX-KIT)........ **$299 ea.**
The Development Kit features NetBurner's Eclipse, an enterprise level professional IDE that provides editing, downloading and debugging in one environment.

NetBurner
Networking in One Day!

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

TASK MANAGER

Tech Evolution

Like many of you, I have shelves of bookended *Circuit Cellar* magazines dating back to the first issue. For work-related purposes, and out of curiosity, I frequently take time out of my busy schedule to thumb through back issues to review many of the embedded designs of the past 23 years. As I prepared for this issue, I returned to *Circuit Cellar* 224 (March 2009) to reread Guido Ottaviani's article, "Robot Navigation and Control," which was actually the starting point for his article about sound tone detection in this issue. (When you get to his article on page 30, you'll understand why I returned to his 2009 robotics article.) After reading the article, I turned to the Task Manager section in that issue, where I had written:

> *Most quality new designs are the sum total of dozens, perhaps hundreds, of earlier projects, applications, and programs. When you see a project in this magazine, you can think of it as a single point in the long timeline of technological evolution. Looking ahead, you should consider each project, idea, and program described in these pages as a contribution to future projects.*

I still believe this. Guido's article in this issue is an excellent example. The project he finished in 2009 was simply a prelude to his work with sensors today. And, as you'll see, the other projects described in this issue are also contributions to the evolution of various important embedded technologies. Let's review.

On page 18, David Ludington continues his series titled "High-Accuracy Voltage Reference Using PWM." This month he covers a hardware design to evaluate practical PWM circuits.

Turn to page 24 for an article about an auditory navigation system project completed by a team of students at Cornell University. The team describes how it harnessed the power of various cutting-edge technologies—from embedded to GPS to biological—to build a wearable auditory navigator. The design is ready for future upgrades.

On page 42, Clemens Valens introduces the exciting chipKIT Max32. This is the next phase of Arduino-compatible technology.

Embedded system protection is a constantly evolving field of engineering. In this issue, George Novacek provides helpful information about EMI, EMC, and designing successful systems (p. 46).

Turn to page 50 for an interview with Shlomo Engelberg, a professor and author who has helped the field of embedded design evolve with his projects, courses, and books since the mid-1990s.

On page 54, George Martin presents the fourth article in his series on product development. His step-by-step tutorial will help you bring a product from idea to production.

On page 60, Richard Wotiz covers the topic of ionization detection. His article will likely inspire many of you to develop new, more effective detection systems of your own.

Lastly, Jeff Bachiochi presents the next phase of his innovative fly-by-wire wheelchair project (p. 66). This project represents not only the evolution of a novel system design, but also the evolution of more than two decades worth of electronics control technologies.

cj@circuitcellar.com

C. Abate

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

PROJECT EDITORS
Ken Davidson
David Tweed

PUBLISHER
Hugo Van haecke

ASSOCIATE PUBLISHER
Shannon Barraclough

CUSTOMER SERVICE
Debbie Lavoie

CONTROLLER
Jeff Yanco

ADMINISTRATIVE COORDINATOR
Valerie Luster

ART DIRECTOR
KC Prescott

GRAPHIC DESIGNER
Grace Chen

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® (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.

INSIDE ISSUE 256

November 2011 • Analog Techniques

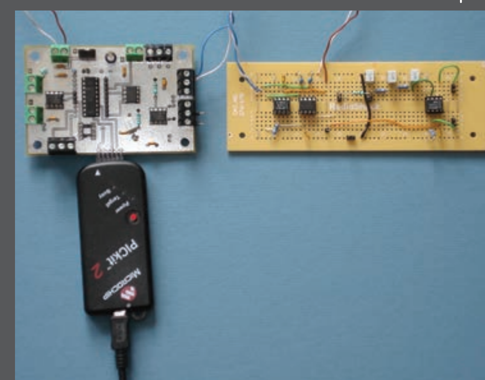
18 **High-Accuracy Voltage Reference Using PWM (Part 2)**
Hardware Design
David Ludington

24 **Auditory Navigator**
Garrett Phillips, Matthew Kinne, and Nick Annetta

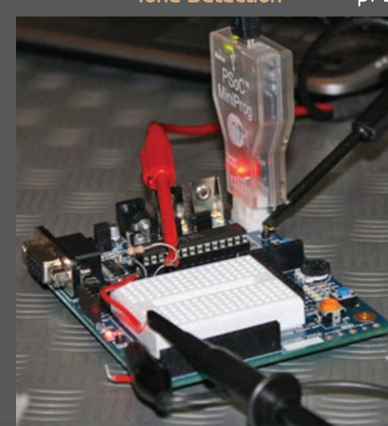
30 **Sound Tone Detection with a PSoC (Part 1)**
Analog Signal Management
Guido Ottaviani

42 **Super Arduino**
Getting Started with the chipKIT Max32
Clemens Valens
DesignSpark chipKIT Challenge Primer

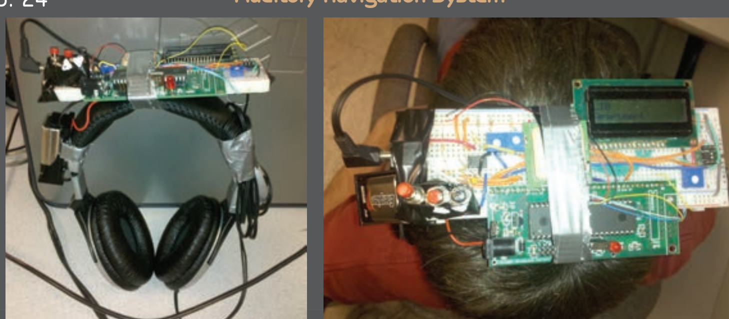
PWM Generator Board p. 18



Tone Detection p. 30



p. 24 Auditory Navigation System



46 **THE CONSUMMATE ENGINEER**
Shielding and Transfer Impedance
George Novacek

54 **LESSONS FROM THE TRENCHES**
Design Development (Part 4)
Processors, Power, and Interfacing
George Martin

60 **EMBEDDED UNVEILED**
Ionization Detectors
Richard Wotiz

66 **FROM THE BENCH**
Fly-By-Wire Wheelchair (Part 1)
Beyond Normal Joystick Control
Jeff Bachiochi

TASK MANAGER **4**
Tech Evolution
C. J. Abate

NEW PRODUCT NEWS **8**

TEST YOUR EQ **17**

QUESTIONS & ANSWERS **50**
Embedded Design Theory & Practice
An Interview with Shlomo Engelberg

CROSSWORD **74**

INDEX OF ADVERTISERS **79**
December Preview

PRIORITY INTERRUPT **80**
To 4G or Not to 4G?
Steve Ciarcia

HIGH-EFFICIENCY 3-A STEP-DOWN CONVERTER

The **AS1328** is high-efficiency 1.5-MHz synchronous DC-DC step-down converter. This buck regulator features efficiencies up to 96% and a quiescent current of only 25 µA.

The converter delivers up to 3 A of output current at output voltages as low as 0.6 V. It operates from an input voltage of 2.7 to 5.5 V over a –40°C to 85°C temperature range. Fixed-output voltage versions are available in 50-mV increments. Alternatively, the output voltage can be set from 0.6 V to the input voltage via an external resistor divider. The AS1328 converter incorporates a Softstart feature to minimize the initial inrush current. It is available with a Power Okay function or Low Battery Detection signal.

The converter conserves board space by incorporating internal compensation components as well as two switches with RDS(ON) of only 35 mΩ . Encased in a 3 mm × 3 mm TQFN package, the AS1328 features a high fixed-switching frequency of 1.5 MHz, enabling the use of small surface-mount inductors.

The converter's input voltage range makes it well suited for applications powered by a single li-ion battery. A small footprint enables the AS1328 to be used in portable devices, including medical instruments, mobile Internet devices, and tablet PCs. The AS1328 converter is also useful for applications requiring high currents such as solid-state drives, set-top boxes, base stations, telecom and network equipment, FPGA, DDR, and microprocessor core supplies.

The AS1328 costs **$1.30** in 1,000-piece quantities.

austriamicrosystems
www.austriamicrosystems.com

ANALOG RESISTIVE USB TOUCHSCREEN CONTROLLER

The **AR1100 Analog Resistive Touch Controller** is a high-performance, USB plug-and-play device that offers advanced calibration capabilities as a USB mouse or single-input digitizer. The controller is available as a turnkey chip or board product supporting four-, five-, and eight-wire touchscreens, with free drivers for most major operating systems. The AR1100 is designed for those who want a drop-in touch controller to universally support their entire standard-resistive-touch portfolio.

Touch input is fast becoming a standard user interface. Resistive touch provides the benefits of easy integration, low total system cost, and acceptance of finger, glove, or stylus input. The AR1100 is designed for touch-input mechanisms such as medical devices, industrial controls, and handwriting or signature capture.

USB communication is the industry standard for attaching peripherals to a computer. The easy-to-integrate AR1100 meets all of these needs in a single-chip solution. It also features advanced calibration options for alignment and linearization as well as accurate button pressing for critical applications with tight spacing.

In addition to the AR1100 controller, Microchip announced enhancements to its AR1000 Analog Resistive Touch-Screen Controllers, with new high-volume market pricing, a full suite of drivers—including those for the Windows CE, Linux, and Android operating systems—and a new 2-V minimum operating voltage.

The AR1000 targets low-cost embedded applications using I^2C, SPI, or UART communications. The controller is available in 20-pin SOIC, SSOP, and QFN packages for **$0.95** each, in 10,000-unit quantities. The AR1100 is also available as a board product for **$12** each in production quantities. The mTouch AR1100 Development Kit costs **$89.99** and includes an AR1100 production-ready printed circuit board (PCB) with a USB cable and a five-wire, analog-resistive touchscreen.

Microchip Technology, Inc.
www.microchip.com

NEW PRODUCT NEWS

QTouch CAPACITIVE-TOUCH CONTROLLERS

The **QTouch** family of capacitive-touch controllers is International Electrotechnical Commission (IEC) and European Standards (EN) 60730 ready for the home appliance market. IEC/EN 60730 standards cover mechanical, electrical, electronic, EMC, and abnormal operation of appliances. These certifications ensure safe operation within an appliance if a fault within the system is detected by shutting it down.

Selecting the right microcontroller increases the ease of passing the certification, which speeds a products time to market. With this certification, manufacturers can easily design household appliances that are IEC/EN-ready with Atmel capacitive-touch controllers for buttons, sliders, and wheels.

The QTouch family also features Atmel's conducted immunity (CI), which is the ability for a device to perform its operations despite unwanted "noisy" RF voltages and currents carried by external wires and cables.

The QTouch family of capacitive-touch controllers includes the Atmel AT42QT1244, AT42QT1245, AT42QT2640, and AT42QT1481. All devices are IEC/EN 60730 Class B certified and include failure mode effect analysis (FMEA) support to monitor self tests to help ensure that the system is running properly. In addition, all the new controllers include Atmel's patented Adjacent Key Suppression (AKS) mode which provides precise discrimination between closely spaced keys and moisture tolerance with Atmel's patented QMatrix technology for extreme environments. The AKS mode also supports high button counts through 24-, 48-, or 64- acquisition channels. All Atmel touch controllers are ready to use without microcontroller programming. Configurations are completed through UART, SPI, or I^2C interfaces.

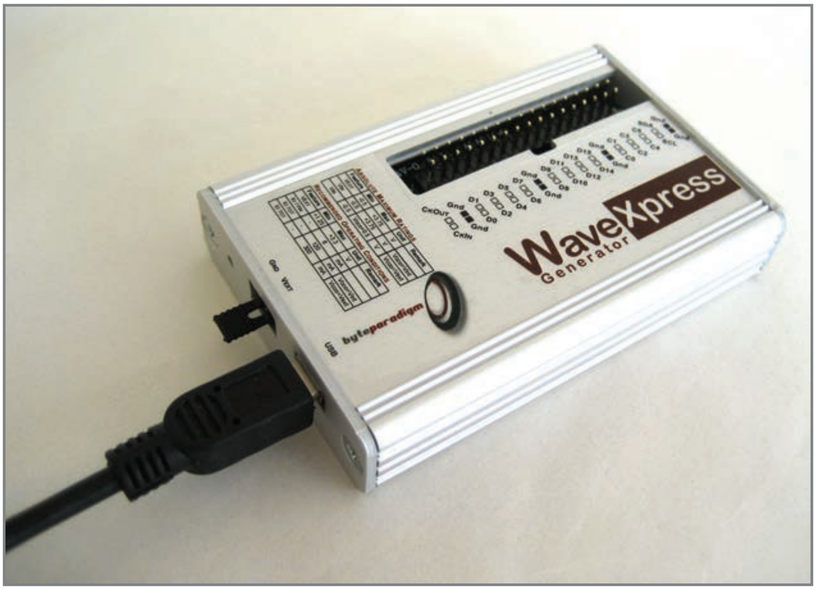
Pricing for the Atmel AT42QT1481, AT42QT1244, AT42QT1245, and AT42QT2640 starts at **$2.48** for 1,000-piece quantities.

Atmel Corp.
www.atmel.com

NPN

NEW STANDARD FOR USB-CONTROLLED DIGITAL-PATTERN GENERATORS

Byte Paradigm has upgraded the specifications of its entry-level digital-pattern generator. Since September 1, 2011, the **Wave Generator Xpress** has been delivered with a large 8-MB internal memory buffer and supports digital-pattern generation up to 100 MHz. With the specification upgrade, digital electronics engineers now have 500 times more memory for storing digital vectors and two times the speed of the previous model for pattern generation.



Wave Generator Xpress enables cycle-accurate 1-to-16-bit-wide digital-pattern generation. The USB 2.0 device supports 1.5- to 3.3-V LVCMOS I/O standards. It is PC controlled with BPI Control Panel software, which includes graphical user interface, TCL/tk scripting environment, and a free C/C++ API, which enables more complex and automated testing.

Possible applications include post-manufacturing ASIC/SoC manual bench functional testing; FPGA, digital board, and DAC input stimulus generation for functional testing and debugging; IP validation on a prototyping board; custom digital interface access; and VHDL/Verilog design environment-to-board prototype digital stimuli generation.

The Wave Generator Xpress costs **$699**.

Byte Paradigm
www.byteparadigm.com

NPN


1st in the Industry
PCB ASSEMBLY PCB ASSEMBLY
PCB ASSEMBLY PCB ASSEMBLY
Full Turnkey Online Quote!
Why suffer from Procurement Headaches!
Go to www.PCBnet.com and click on 'Online Turnkey Quote'.
Just upload your BOM
Define Part Number and Quantity
Enter few specifications for Bare Boards
...and you are DONE.
You will get Full Turnkey Price (Components, Labor & Bare Boards) within MINUTES!
imagineering inc.
SINCE 1985
www.PCBnet.com
847-806-0003 Fax: 847-806-0004
email: sales@PCBnet.com

CLASS-G HEADPHONE & CLASS-D SPEAKER AMPLIFIERS

The **FAB1200** headphone amplifier and the **FAB2200** audio subsystem are two products designed to make small speakers in mobile devices—such as smartphones, tablets and multimedia Internet devices—sound louder and better while minimizing the impact on battery life.

The FAB1200 is a stereo Class-G ground-referenced headphone amplifier with an integrated buck converter. It features a charge pump that generates a negative supply voltage. This enables the headphone output to be ground-centered and capacitor-free, which eliminates up to two external capacitors. An integrated inductive buck regulator provides direct battery connection and adjusts the supply voltage between two different levels based on the output signal level in order to reduce power consumption. The result of these features is reduced systems cost and extended battery runtime, with a high level of audio quality. Available in a 16-bump, 0.4-mm pitch, 1.56 mm × 1.56 mm WLCSP package, the device offers excellent audio performance for better sounding audio headsets and is ideal for mobile handsets, tablets/MIDs, and MP3 and portable media players.

The FAB2200 is an audio subsystem with a stereo Class-G headphone amplifier and a 1.2-W Class-D mono speaker amplifier. It combines a capacitor-free stereo Class-G headphone amplifier with a Class-D speaker amplifier. A proprietary integrated charge pump generates multiple supply rails for a ground-centered Class-G headphone output to reduce power dissipation and provide a high-power supply rejection ratio. The filterless Class-D amplifier can be connected directly to a speaker without the need for two external filter networks, which reduces the overall cost of the solution. The device features automatic gain control which limits the maximum speaker output levels to protect speakers without introducing distortion. It can also dynamically limit clipping as the battery voltage falls. Available in a 25-bump, 0.4-mm pitch, WLCSP package, the FAB2200 is ideal for cellular handsets, notebook computers, and tablets.

In 1,000-piece quantities, the FAB1200 costs **$0.36** and the FAB2200 costs **$0.57**.

Fairchild Semiconductor
www.fairchildsemi.com

NPN


PCB Fabrication PCB Fabrication PCB Fabrication
PCB Fabrication PCB Fabrication
5 Competitive Advantages
Quick-Turn Production
CAM USA
Door to Door Delivery
Overseas Manufacturing
1. Overseas Manufacturing
Imagineering, Inc. enjoys the reputation of being one of the most experienced & successful offshore PCB suppliers.
2. 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 analysis.
3. Quick-Turn Production
Imagineering offers small volume production in 5-6 days and medium to large volume production in 2-3 weeks.
Significant Price Saving
Capabilities
Up to 42 Layers
Blind Buried Vias
HDI Micro Vias
Di-Electric Thickness
Impedance Control (TDR Tested)
Plated Edge Holes
Up to 6oz Copper
6 mil Laser Drill
2 mil line width/spacing
Conductive Silver Filled Vias
Epoxy Filled Vias
Aluminum Metal Core Boards
ITAR, ISO 9001 : 2008
4. 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.
5. Significant Price Saving
Our global buying power combined with the capabilities of our overseas manufacturers translate into tremendous savings to our customers.
Over the past 5 years, 90,000 prototypes have been successfully delivered from overseas to over 5000 customers.
imagineering inc.
SINCE 1985
www.PCBnet.com
847-806-0003 Fax: 847-806-0004
email: sales@PCBnet.com

100-kHz DATA ACQUISITION MODULES

The **DT9812A**, **DT9813A**, and **DT9814A** are three new low-cost, USB-powered data acquisition (DAQ) modules. With the multifunction capability of Data Translation's ECON-series, the modules feature USB 2.0 compliance and fast sampling rates of up to 100 ksps.

Additional features include: power by USB for portable measurements; 12-bit resolution for analog input/output subsystems; 8-, 16-, or 24-SE analog input channels for flexibility; two analog output channels to generate sine, rectangle, triangle, or DC waveforms; one counter/timer to perform event counting, frequency measurement, edge-to-edge measurements, and to generate continuous-pulse output operations; and digital I/O lines. The modules also include software and drivers compatible with Windows XP/Vista/7.

The DT9812A costs **$475**, the DT9813A costs **$525**, and the DT9814A costs **$575**.

Data Translation, Inc.
www.datatranslation.com

PXIe EMBEDDED CONTROLLER

The **M9036A** is a modular PXIe embedded controller designed to take advantage of the x8 PCI Express links when using the Agilent M9018A PXIe chassis. When combined, the M9036A controller and M9018A chassis provide a high-throughput PXI test platform for peer-to-peer applications.

This platform is capable of integrating legacy PXI instruments into the hybrid slots of the chassis while providing up to 8 GBps of system bandwidth with dual x8 express links to handle demanding RF, uW, and streaming applications, such as transferring data between cards without involving the controller. The controller can also operate in a four-link configuration providing compatibility with existing PXIe chassis.

The M9036A was designed using an Intel Core i5 2.4-GHz dual-core processor with hyper-threading technology with a removable 160-GB solid-state drive and 4 GB of RAM (upgradable to 8 GB) in the standard configuration. The built-in front-panel connectors enable seamless integration with traditional standalone instruments to create sophisticated hybrid test systems. This is done using USB and LAN or DVI-I, GPIB, ExpressCard 34, and an SMB trigger connector.

The controller module provides a choice of an external controller or the new embedded controller for integrated, compact solutions. It is preloaded with operating system, controller and chassis drivers, soft front panel and Agilent I/O libraries including VISA, Agilent Connection Expert, and I/O monitoring. This enables users to quickly communicate with and control both modular and standalone instruments.

The M9036A starts at **$6,600**.

Agilent Technologies, Inc.
www.agilent.com

CIRCUIT CELLAR

Test Your EQ

Problem 1—Is the signal carried by an "ideal" coaxial cable immune from external electric and magnetic fields? Explain.

Problem 2—What happens if you "ground" both ends of the shield of a coaxial cable?

Problem 3—In what ways are real coaxial cables non-ideal?

Problem 4—How does Twinax, a form of shielded twisted pair, address some of the shortcomings of coax?

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

STATEMENT REQUIRED BY THE ACT OF AUGUST 12, 1970, TITLE 39, UNITED STATES CODE SHOWING THE OWNERSHIP, MANAGEMENT AND CIRCULATION OF CIRCUIT CELLAR, THE MAGAZINE FOR COMPUTER APPLICATIONS. Published monthly at 4 Park Street, Vernon, CT 06066. Annual subscription price is $45.00. Publisher: Hugo Van haecke. Editorial Director: Steven Ciarcia. Editor-In-Chief: C.J. Abate. The owner is Circuit Cellar, Inc., Vernon, CT 06066. The names and addresses of stockholders holding one percent or more of the total amount of stock are: Elektor International Media, LLC, 4 Park Street, Vernon, CT 06066.
EXTENT AND NATURE OF CIRCULATION: Average number of copies of each issue published during the preceding twelve months; (A) total number of copies printed, 18,302; (B.1) paid/requested mail subscriptions, 5,183; (B.3) sales through dealers and carriers, street vendors and counter sales, 3,723; (B.4) paid/requested copies distributed by other mail classes, 10; (C) total paid/requested circulation, 8,916; (D.1) samples, complimentary, and other nonrequested copies,1,857; (D.4) Nonrequested copies distributed outside the mail, 2,559; (E) total nonrequested distribution (sum of D.1 & D.4), 4,416; (F) total distribution (sum of C & E), 13,332; (G) copies not distributed (office use, leftover, unaccounted, spoiled after printing, returns from news agents), 4970; (H) total (sum of F & G), 18,302. Percent Paid Requested: 67% Actual number of copies of a single issue published nearest to filing date: (A) total number of copies printed, 15,600; (B.1) paid/requested mail subscriptions, 5,354; (B.3) sales through dealers and carriers, street vendors and counter sales, 3,318; (B.4) paid/requested copies distributed by other mail classes, 10; (C) total paid/requested circulation, 8,682; (D.1) samples, complimentary, and other nonrequested copies, 1,832; (D.4) Nonrequested copies distributed outside the mail,1,781; (E) total nonrequested distribution (sum of D.1 & D.4), 3,613; (F) total distribution (sum of C & E), 12,295; (G) copies not distributed (office use, leftover, unaccounted, spoiled after printing, returns from news agents), 3,305; (H) total (sum of F & G), 15,600. Percent Paid Requested 71%. I certify that the statements made by me above are correct and complete. Hugo Van haecke, Publisher.

by David Ludington (USA)

High-Accuracy Voltage Reference Using PWM (Part 2)

Hardware Design

The first part of this article series covered the basics of pulse-width modulation (PWM). This article details a hardware design to evaluate practical PWM circuits. Also covered is the test setup to evaluate the hardware along with performance test results showing the achieved voltage accuracy.

This is the second part of a two-part article series that demonstrates how pulse-width modulation (PWM) can be used to generate an accurate programmable voltage reference. In Part 1, I described basic PWM operation and identified the two PWM parameters (frequency and voltage) that are important for this application. I also identified analog parameters (time delays, offset voltages, bias currents, etc.) of the PWM generator and filter circuits that need careful attention during the design to minimize degradation of the output voltage accuracy.

In Part 2, I will be using the analysis developed in Part 1 to select components and also design and fabricate circuit boards to test this PWM technique. First, I will describe the hardware design for this project. It consists of a microcontroller PWM generator circuit board and a filter board (see Photo 1). Then, I will describe the software code used for testing reference voltage performance. Finally, I will describe the test setup and present performance test results.

HARDWARE DESIGN

So far, the discussion has been at a theoretical/conceptual level. In this section, we will choose components and design circuits to evaluate PWM voltage reference performance. The first component is the PWM generator. Most modern 16- and 32-bit microcontrollers have PWM circuits with clock speeds high enough to minimize delays and enough resolution steps to enable a wide range of test conditions. Another option is to design the PWM circuit in a field-programmable gate array (FPGA) where the clock speed is typically higher than a microcontroller giving shorter cycle times. However, FPGA design is not as flexible for changing test conditions. So, for this evaluation, a 16-bit microcontroller will be used.

The next component is the PWM driver. A single pole, double throw (SPDT) analog switch seems like an obvious choice to switch between 0 and 5 V. However, analog

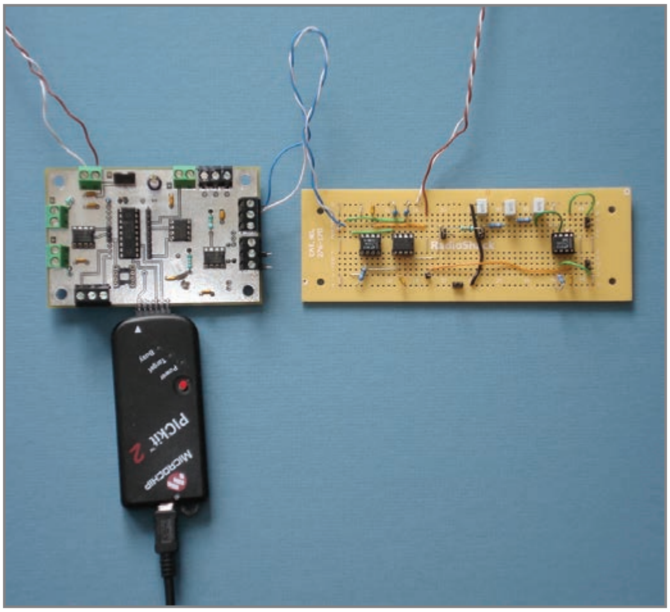


Photo 1—The PWM generator board and filter board

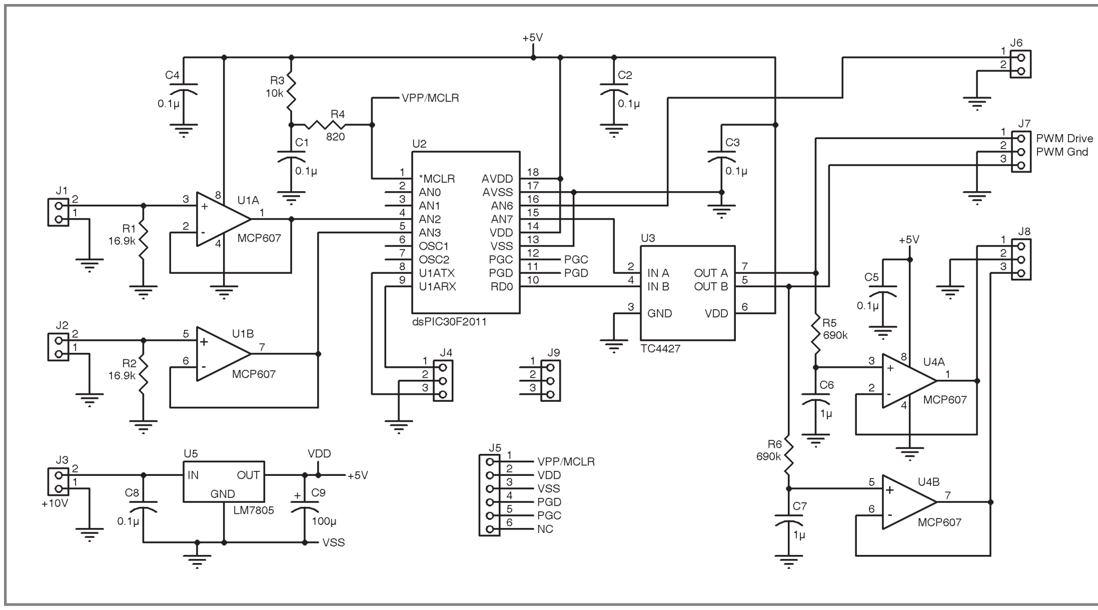


Figure 1—A circuit board with the PWM generation and driver circuits I am using to test the PWM voltage reference technique

switches are usually designed with a deliberate difference between on and off time delays to ensure break-before-make operation which could create a significant voltage offset. Another possibility is to use a high-speed MOSFET driver instead. It is used as a low-resistance SPDT switch with more evenly matched delay times. For this project a MOSFET driver will be used.

The design of the low-pass filter requires tradeoffs between the number of filter stages, the component values, and the buffer amplifier parameters. The performance goal for the low-pass filter is to maximize filter attenuation at the PWM frequency while at the same time causing the least degradation of the DC output voltage. In general, a multistage filter will give better performance than a single-stage filter. The choice of component values for the multistage filter depends on which parameter (capacitor leakage or buffer amplifier input bias current) affects the output voltage the most.

For example, using a 1-MΩ resistor and 1-µF capacitor, the single-stage filter time constant is 1 s and the filter corner frequency is 0.16 Hz. If the PWM frequency is 16 Hz, then the filter corner frequency is 100 times lower than the PWM frequency and the attenuation is 100 (40 dB) for a single-stage filter. For a two-stage filter (keeping the total resistor and capacitor values the same) the resistors would be 500 kΩ and the capacitors would be 0.5 µF. This creates a corner frequency for each filter section four times higher than a single-stage filter and the corresponding attenuation four times lower. However, the attenuation of each individual filter stage (25) is multiplied together to give a total filter attenuation of 625 (56 dB).

In this project I decided to go with a three-stage filter with the total filter resistance less than 1 MΩ, but using three 1-µF capacitors with a total capacitance of 3 µF since the capacitor leakage current should have less impact on the output voltage than the amplifier bias current.

Figure 1 is a schematic of a circuit board with the PWM generation and driver circuits I am using to test the PWM voltage reference technique (the schematic also contains other circuits that are not being used in this application). A Microchip Technology dsPIC30F2011 digital signal controller is used to generate the PWM test waveforms. The timers in this microcontroller are 16 bits with a prescalar divider. This provides a lot of flexibility in generating the test waveforms. There are two PWM

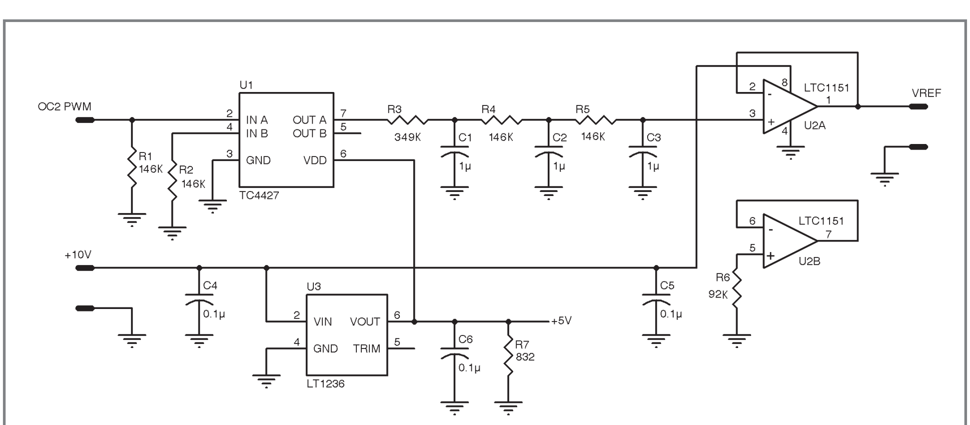


Figure 2—The filter board

circuits, OC1 and OC2, which are output on pin 10 and pin 15, respectively. The PWM signals are sent to a Microchip Technology TC4427A dual MOSFET driver. The output FETs in this driver have low on-resistance to V_{SS} and V_{DD} and are normally used to drive large-capacitive loads. In this application I am using this capability to provide a low-resistance source with matched delay times to the PWM filters. Also, the resistance values of the two output FETs are closely matched which will minimize offset voltages in the filtered output.

The outputs of the dual driver go to connector J7 and also to the inputs of two single-stage filters consisting of resistor R6 and capacitor C7 for the first filter and resistor R5 and capacitor C6 for the second filter. These filter outputs are connected to the inputs (at pin 3 and pin 5) of two buffer amplifiers in a Microchip Technology MCP607 dual-operational amplifier.

The on-board filter circuits (see Photo 1) were used in the initial testing, but as the PWM frequency was lowered to improve accuracy, these single-stage filters did not provide enough filtering and a separate circuit board with a three-filter stage was added (see Figure 2). The input for this filter comes from the OC2 PWM circuit (J7, pin1) in Figure 1 into another MOSFET driver U1 in Figure 2. This driver uses a more precise reference voltage provided by a Linear Technology LT1236 voltage reference chip. The driver output on pin 7 of U1 goes to the three-stage filter that consists of resistors R3, R4, and R5 and capacitors C1, C2, and C3. The output of the filter is buffered by a Linear Technology LTC1151 op-amp which has very low offset voltage.

Listing 1—Code for Microchip Technology's dsPIC30F2011 microcontroller

```
#include "p30F2011.h"

_FWDT(WDT_OFF);
_FOSC(CSW_FSCM_OFF & FRC);          //FRC directly

int main(void)
{
       IFS0bits.T2IF = 0;           //clear interrupt flag
       IEC0bits.T2IE = 1;           //set interrupt enable bit

       PORTC = 0x0000;              //Set PORTC output
       TRISC = 0x1FFF;              //Set PORTC pin directions

       OC2CON = 0x0000;             //Set OC2 parameters
       OC2RS  = 0x0000;             //    :
       OC2R   = 0x0000;             //    :
       OC2CON = 0x0006;             //Use Timer2

       TMR2 = 0;                    //Timer2 = 0
       PR2 = 49999;                 //Timer2 period
       T2CON = 0x8000;              //Timer2 on

       while(1)
       {
       }

       return 0;
}

void __attribute__((interrupt)) _T2Interrupt(void)
{
       PORTCbits.RC14 = 1;          //Turn on marker pulse
       OC2RS = 9000;                //Set OC2 duty cycle
       IFS0bits.T2IF = 0;           //Clear interrupt flag
       PORTCbits.RC14 = 0;          //Turn off marker pulse
}
```

SOFTWARE DESIGN

The software code used in the dsPIC30F2011 microcontroller to evaluate the performance of the PWM voltage reference is shown in Listing 1. The board I was using did not have any interfacing circuits, so for these first tests, I hard coded values into the program and recompiled them whenever I needed to change the PWM waveform or clock frequency. In the future, I plan on using a printed circuit board (PCB) that includes a PS/2 keyboard interface and a serial port connection to the computer.

The `_FOSC` macro at the beginning of the program is used to change the clock frequency. The internal oscillator in the dsPIC30F2011 microcontroller nominally runs at 7.37 MHz and the instruction clock is the oscillator divided by four. The PR2 variable sets the Timer2 period and determines the number of steps in the PWM waveform (PR2 + 1). The T2CON variable, which sets the T2CON register, includes bits to set the prescalar division ratio and lower the Timer2 clock relative to the instruction clock. The MSb of the T2CON register turns Timer2 on or off. The `PORTCbits.RC14` variable in the interrupt routine is used to turn a pulse on and off every time the Interrupt routine is called. The length of time spent in the interrupt routine can be determined by measuring the pulse width on an oscilloscope or frequency counter.

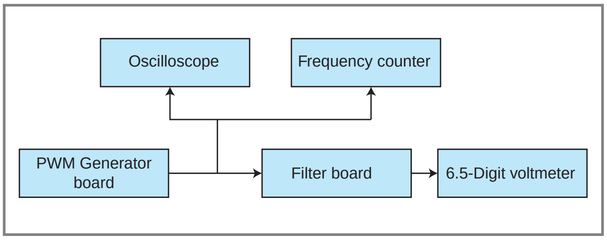


Figure 3—Voltage reference evaluation test setup

VOLTAGE REFERENCE TESTING

The setup for testing voltage reference

PWM Parameters*			Data			
PR2	OC1RS	#Steps	V_{REF}	Expected voltage	Measured voltage	Full-scale error %
4	1	5	5.00057	1.00011	0.93039	1.4
49	10	50	5.00086	1.00017	0.99322	0.14
499	100	500	5.00089	1.00018	0.99939	0.016
4,999	1,000	5,000	5.00099	1.0002	1.00006	0.0028
49,999	10,000	50,000	5.00089	1.00018	1.00009	0.0018

* Timer2 clock = 1.827 MHz

Table 1—Full-scale error versus number of PWM steps

performance is shown in Figure 3. The oscilloscope and frequency counter were used during the initial debug of the circuits to verify correct operation and during the performance testing to verify that the PWM and rate matched the PR2 and OC1RS parameters for the selected voltage output.

The main instrument used for evaluating performance was a Datron 1081 digital multimeter with a full scale count of 1,999,999. This enables the use of just one voltage scale (10 V) to measure both the PWM reference voltage and the derived output voltage with seven digits displayed. Thus, the full 5 V is displayed along with showing the digit.

The LT1236 voltage reference without any adjustment had a no-load output voltage of approximately 5.0015 V. I loaded the output with an 832-Ω resistor and the voltage dropped to about 5.00085 V and drifted between 5.00057 and 5.00099 V with time and room temperature changes.

When data was taken, the reference voltage was first measured. Then an expected output voltage based on that measured value, PR2 and OC1RS, was calculated and compared to the measured PWM output voltage. The first performance evaluation I performed used a fixed low-resolution five-step PWM period (PR2 = 4), a minimum-pulse width (OC1RS = 1), and varied the clock rate into the PWM generator. At higher clock rates, the filtered output voltage (nominally 1 V) was not that precise. The precision increased as the clock rate was lowered (the whole data set is not shown here, except for one value which is the first entry in Table 1).

For the second test, I fixed the PWM clock rate at 1.827 MHz (the dsPIC30F2011 internal RC clock rate divided by four) and changed the PR2 and OC1RS values to maintain a nominal 1-V PWM output while increasing the resolution. This data is shown in Table 1. Similar results to the first test were obtained. As the pulse width and period were

Nominal voltage	Expected voltage	Measured voltage	Full-scale error (%)
0.5	0.500085	0.49999	0.0019
1	1.00017	1.00006	0.0022
1.5	1.500255	1.50018	0.0019
2	2.00034	2.00018	0.0032
2.5	2.500425	2.50027	0.0031
3	3.00051	3.00035	0.0032
3.5	3.500595	3.50044	0.0031
4	4.00068	4.00052	0.0032
4.5	4.500765	4.50062	0.0029

* 50,000 PWM steps

Table 2—PWM voltage reference performance data*

increased (lower effective PWM rate), the accuracy of the filtered output voltage increased.

The final performance test results are shown in Table 2. Here the PWM clock is 1.827 MHz and the PR2 period is 49,999. The OC1RS variable was changed to generate different output voltages and evaluate performance across the output voltage range.

ACCURACY & PERFORMANCE

The data in Table 1 and Table 2 show that a programmable voltage using a PWM digital-to-analog converter (DAC) can be highly accurate. The full-scale error of the output voltage is less than 0.004% over the entire voltage range. With a 5-V reference and 50,000 PWM steps, the minimum step size is 100 µV. Any voltage between 0 and 5 V with 100-µV resolution can be obtained simply by changing the OC1RS variable to a value between 0 and 49,999. Thus, for a low cost, you can add a high-precision programmable voltage reference generator to your collection of test instruments.

The only negative is that to achieve this accuracy and resolution, the PWM rate is only 36.5 Hz. This means that to get greater than 100 dB of attenuation in the filter requires multiple stages and large-value components. In the future, I plan to investigate implementing the PWM generator in a FPGA. Since almost any modern FPGA can be clocked well above 100 MHz, the digital delays and rise/fall times should be less than from a microcontroller which is clocked at a lower rate. This should enable a higher PWM clock frequency than 1.827 MHz and PWM rate of 36.5 Hz while still giving the performance shown in Table 2.

This two-part article has demonstrated that PWM can be used to generate an accurate programmable voltage reference of less than 0.01% full-scale accuracy. The present data shows that the output voltage is still a little lower than the expected voltage. I hope to investigate this error in future work. I also have several ideas on improvements to minimize capacitor leakage and increase the PWM rate. ◾

David Ludington (davidludington@hotmail.com) is a retired electrical engineer with experience in low-noise analog design and infrared system design. He earned a BSEE at Michigan State University and an MSEE at Syracuse University.

SOURCES

1081 Digital multimeter
Datron Connection | www.datronconnection.com

LT1236 Precision reference and LTC1151 op-amp
Linear Technology Corp. | www.linear.com

dsPIC30F2011 Digital-signal controller, TC4427A dual MOSFET driver, and MCP607 dual-operational amplifier
Microchip Technology, Inc. | www.microchip.com

by Garret Phillips, Matthew Kinne, and Nick Annetta (USA)

Auditory Navigator

You can build an auditory navigation system that mimics a natural biological function to guide a human. This innovative design uses a GPS, a digital compass, and a microcontroller to generate sound based on the direction a user must turn in order to face a determined direction.

Navigation in the past has primarily relied on the use of a map, a compass, or another device that must be visually interpreted. This project demonstrates the ability to navigate a user based on synthesized directional audio that enables the user to move to a known location without the use of a visual aid (see Photo 1). The module uses a global positioning system (GPS), a digital compass, and an Atmel ATmega32 microcontroller to generate sound based on the direction a user must turn in order to face a determined direction. A schematic is shown in Figure 1.

Once the GPS has a lock, the system calculates the angle to the destination using the current latitude and longitude data. The compass then determines which direction the user's head is facing relative to magnetic North. These two angles are used to find the final direction (angle) relative to the user's direction and produce a sound relative to the difference. The left and right sound channels are delayed in time and varied in amplitude to give the user the sensation that the noise is coming from the target location.

THEORY & DESIGN

Sound perceived by the human ear includes a mixture of several spatial auditory cues such as pinna response, interaural time delay (ITD), and shoulder echo. When creating 2-D sound, only the ITD needs to be replicated because it is the feature that enables humans to track sound across the horizon. The ITD is defined as a phase shift that occurs as a sound reaches the left ear and the right ear at different times.[1] The phase shift is based upon the position of the sound source relative to the listener and the diameter of the user's head. If the speed of sound is approximately 13.2″/ms and the user's head is 6.2″ wide, then the maximum interaural delay would be about 463 µs.[2] The maximum delay would occur if the

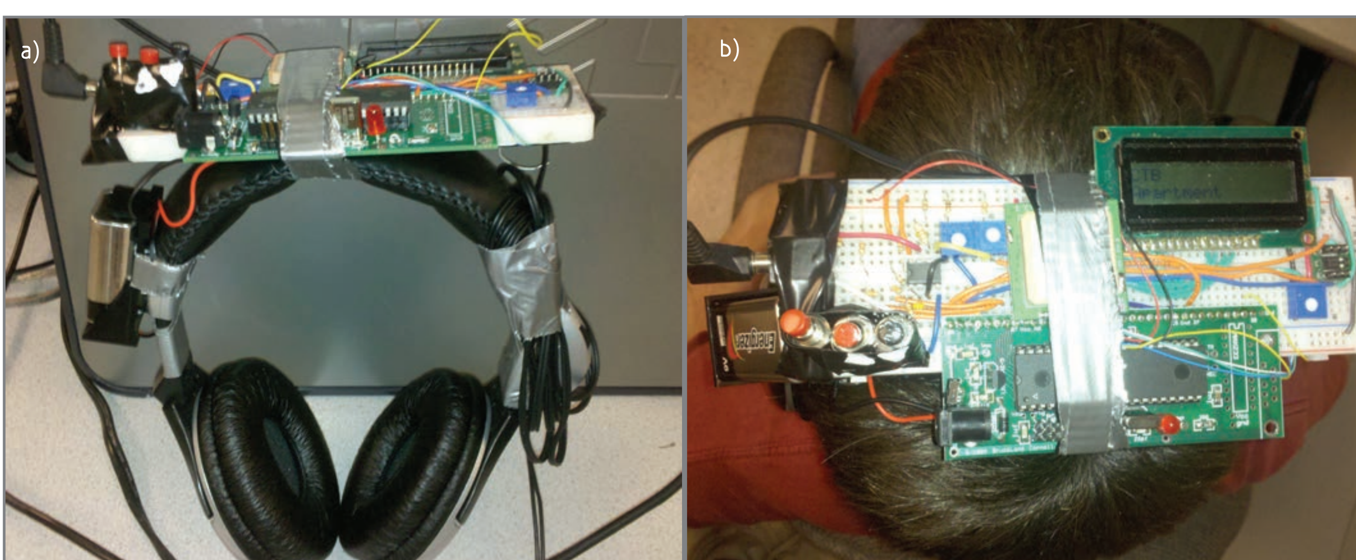


Photo 1a—The final device doesn't look too pretty, but it worked great. b—Notice the push buttons and LCD screen attached right to the top of the device making it compact and easy to use.

sound source were positioned 90° from the user's nose.

The equation for calculating the phase difference based upon the position of the sound source is shown in Figure 2. The quantities v and h are the speed of the sound in air and the width of the observer's head, respectively.[3] The angle is taken with respect to the nose of the observer.

The amplitudes of the sound waves between the two ears are varied to recreate the head shadow that occurs at the ear opposite from the sound source. The head shadow has a higher attenuation at frequencies greater than 1 kHz. The PWM is running at 62,500 samples per second, which is an output frequency of at about 244 Hz for an 8-bit DAC. Therefore, the attenuation should only be 2 to 3 dB.[4]

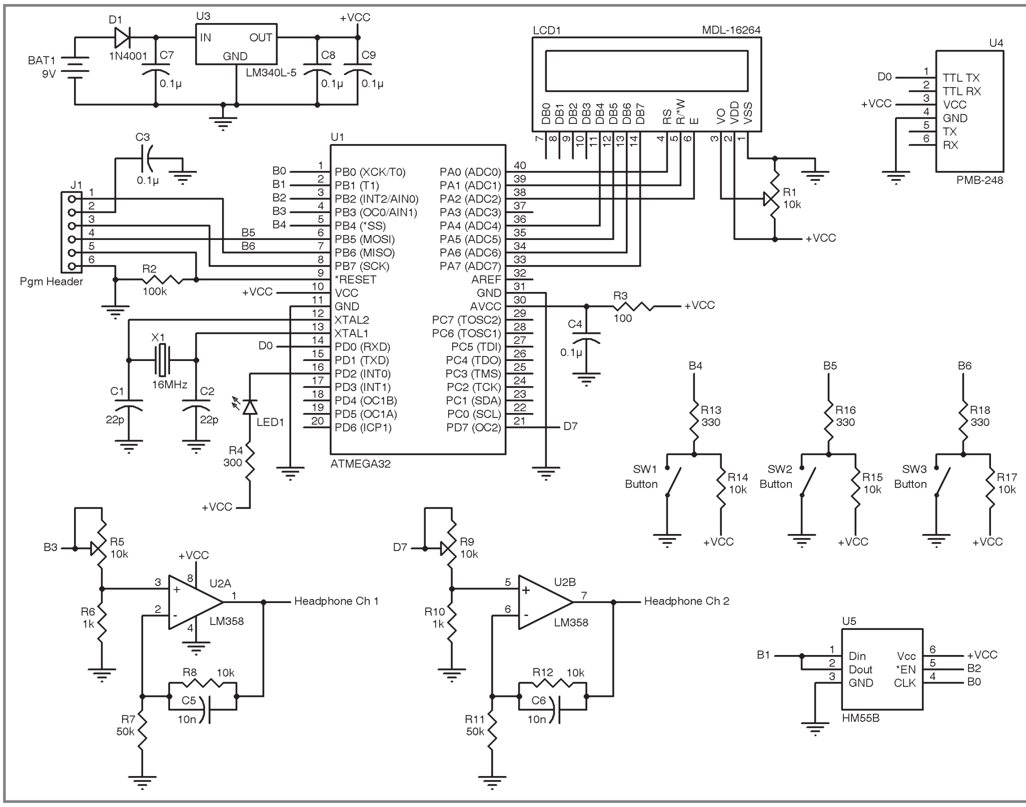


Figure 1—The schematic includes an Atmel ATmega32 microcontroller, a GPS, a compass, and supporting circuitry. The entire device runs off a 9-V battery, which is regulated down to 5 V. A couple of general-purpose amps are used to control the volume in each ear.

Figure 3 shows a high-level diagram of the system. Upon initial startup, the user selects from a number of predefined locations through the use of an LCD screen. We use a GPS and a digital compass to determine at what angle the user hears the sound pulses. Once the GPS has a lock, the system determines the bearing (angle from true North) the user must travel to get to the destination. This angle is compared with the compass output and a sound is made based on which direction the user must turn to face the final location. The sound consists of short pulses of different amplitudes that are delayed between the right and left side to give the effect of direction.

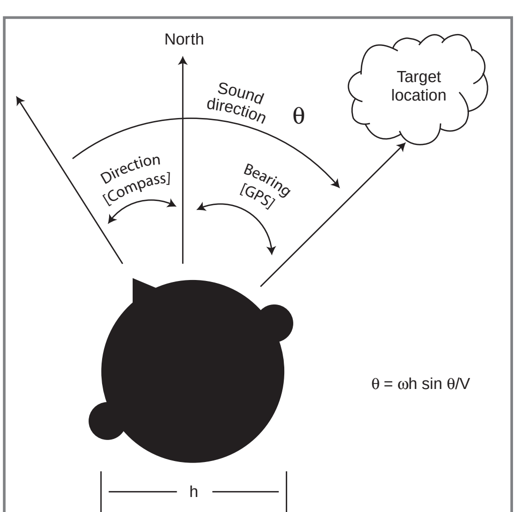


Figure 2—The compass and GPS data are used to determine an angle from true North. The phase difference equation is also shown, which is determined by the width of a human head.

HARDWARE

The digital compass module we used was a Hitachi HM55B from Parallax, mainly because it was around from previous projects. The communication between this device was done with three wires (clock, data, and enable) in a SPI-like fashion. The communication is based on the Basic Stamp function SHIFTIN/SHIFTOUT, which is a two-wire communication. The challenge was to implement this in C for the ATmega32. Timer1 on the device was used to control the timing of the function and three general-purpose input/output (GPIO) pins were used for the controls. A compare interrupt was used to generate the clock and data. This enabled the transmission/receive rate to be set at a fine resolution resulting in the fastest possible operation. The clock signal was generated at twice the data rate so data could be clocked on both the rising and falling edge. We wanted to keep this function as general as possible in case we needed to use it for future projects.

To get data, the ATmega32 pulses the Enable bit high for two clock cycles, then sends `0b0000` on the shared `data_in` and `data_out` line. After another enable toggle, `0b1000` gets sent to the compass signifying a "start measurement" command. The ATmega32 then sends `0b1100` and reads in 4 bits waiting for the data conversion to be done. When the compass sends back 0b1100, the data is ready. The ATmega32 then clocks in 22 bits (11 for x value, and 11 for y value) of

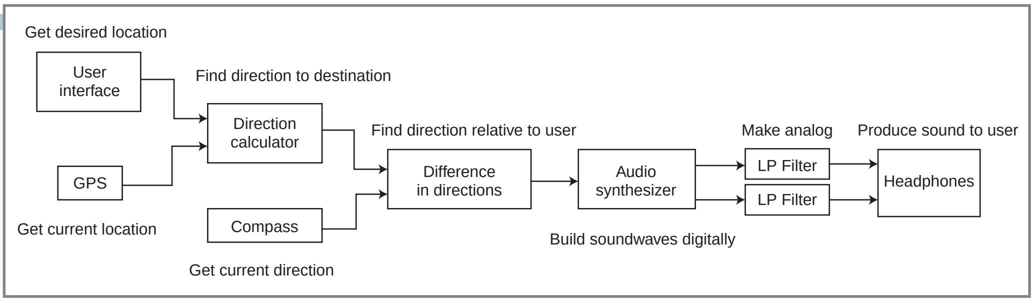


Figure 3—The flow is straightforward starting with the GPS data, followed by the compass data, and finally out to the headphones.

two's complement data most-significant bit (MSB) first. After formatting the data to get the proper sign, the arctangent of x/–y is used to get the resulting angle (from magnetic North). This is then converted from radians to degrees and compensated for the difference between true and magnetic North (subtracting 12° where we are, in Ithaca, NY). The whole function takes 60 ms to run. We determined this was fast enough compared with how quickly the human head can turn.

GPS COMMUNICATION

We chose the Parallax PMB-248 GPS receiver because it was relatively inexpensive and was readily available. We used interrupt-driven serial communication over the USART to receive data from the GPS receiver. The GPS receiver outputs NMEA sentences approximately once a second at 4,800 bps. An example of one packet of NMEA sentences is shown in Figure 4.

There are several cases of redundant data between NMEA sentences and, for the purposes of this project, we only need one set of longitude and latitude. We decided to only read in the recommended minimum content (RMC) sentence as a string and then extract the longitude and latitude from that string.

We use the USART character-ready interrupt which triggers as soon as a full character is received by the USART buffer. Once a character is ready in the USART buffer, the ISR writes the character to a string buffer and enters a state machine. The state machine successively checks for the characters that are expected to be seen at the beginning of the RMC sentence. If the program does not receive the expected header characters, it returns to the beginning of the state machine. If the input characters are `$`, `G`, `P`, `R`, `M`, and `C` in succession, the program records the rest of the sentence as data. The sentence is terminated with `\n`.

```
$GPGGA,144739,4251.9960,N,07806.0827,W,1,04,5.6,1898.0,M,34.5,M,,*61
$GPRMC,144739,A,4251.9960,N,07806.0827,W,1908.5,270.0,050510,5,E,A*2F
$GPGSV,8,1,32,01,12,205,-18,02,43,251,14,03,08,022,-22,04,05,271,00*75
$GPGSV,8,2,32,05,78,032,45,06,83,236,48,07,81,084,47,08,64,206,33*71
$GPGSV,8,3,32,09,90,086,52,10,42,202,13,11,26,284,-4,12,88,117,51*6C
$GPGSV,8,4,32,13,90,027,52,14,68,030,37,15,78,143,44,16,60,220,30*7E
```

Figure 4—Example of a packet of NMEA sentences

The data from the RMC sentence is written to a string buffer. Knowing the format of the RMC sentence, we use the sscanf function to extract the longitude and latitude. The longitude and latitude are sent in the format DDMM.MMMM where DD is in degrees and MM.MMMM is in minutes. We parse the data and recalculate the values in terms of only degrees. It should also be noted that when writing to the string buffer, we ignored decimal points due to difficulties with sscanf reading in floating-point numbers on the microcontroller. In order to compensate for this, we scaled the longitude and latitudes appropriately during the conversion into degrees and then radians.

The following equation was used to calculate the direction from the user's current location to the final destination:

$$\theta = \text{atan2}\left(\begin{matrix}\sin(\Delta\text{long}) \times \cos(\text{lat2}), \cos(\text{lat1}) \times \sin(\text{lat2}) - \\ \sin(\text{lat1}) \times \cos(\text{lat2}) \times \cos(\Delta\text{long})\end{matrix}\right)$$

where lat1 = current latitude, lat2 = target latitude, and dlon = difference in current and target longitude.[5]

SOFTWARE

The directional "beeps" the user hears are short bursts (approximately 20 ms) that are sine waves with varying amplitudes produced at 244 Hz. Direct digital synthesis (DDS) is used to create these sine waves through the use of pulse-width modulation (PWM) on the ATmega32. The amplitudes of the sine waves are also ramped linearly from zero to full scale (and vice versa on the other end) in an attempt to reduce the high-frequency "pop" that is generated from a sharp transient edge.

The PWMs from Timer0 and Timer2 are used to generate sound pulses. Each timer is set to operate in fast PWM mode with a prescaler of one, giving the PWMs a sample rate of 62,500 samples per second. The overflow interrupts of each timer are used to update the output compare registers (OCRs). The OCR indicates the top value or the value of the timer counter (TCNT0) that causes another interrupt.

Initially, Timer1 of the ATmega32 was used to generate two PWM waveforms. The idea was that since Timer1 has two output compare registers, it should be able to generate two PWMs. Timer1 was able to generate two PWMs, but since OCR1A was used as the top value and the PWM waveforms were not unique, the outputs of OCR1A and OCR1B toggled whenever the overflow interrupt was triggered. Using two timers prevented the overflow interrupt from toggling both pulses and also enabled one timer to be used as a clock based on when an interrupt was triggered.

The DDS process consists of a sine function quantized into a 256-entry table and a ramp

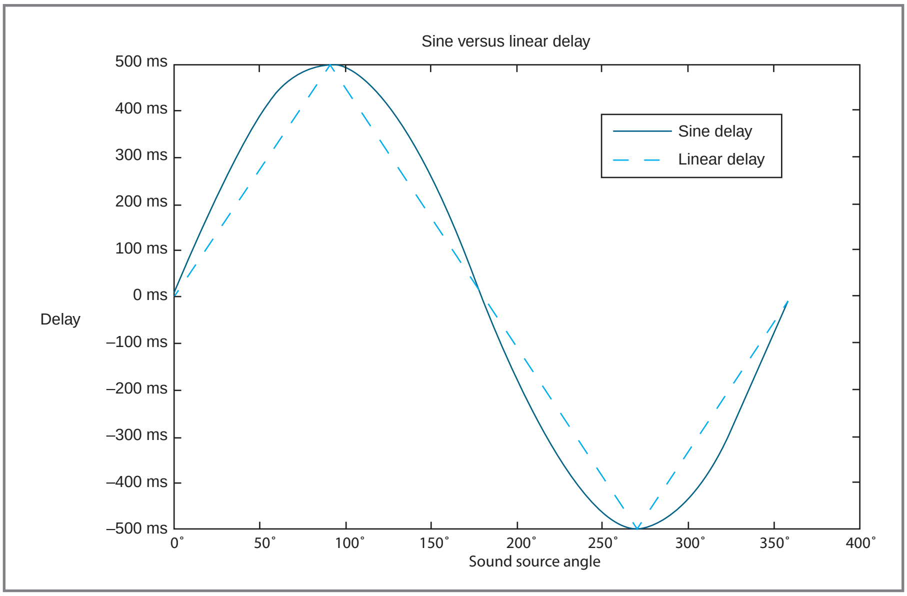


Figure 5—These are the two different delay functions used. The idea was to get more information to the user in the small angle facing forward.

table to linearly increase or decrease wave amplitude. Two more tables were used to represent the phase offset and amplitude of the wave for a given angle representing the sound location. These tables are precomputed to prevent extra computation during execution.

The phase offset table is used to represent the delay that is present at each degree. It was constructed to achieve no offset at a zero-degree heading (the source is directly in front of the user) and maximum offset at 90° or 270°. If the sound source is behind the user, then the channel of the ear furthest from the source will be fully delayed, encouraging the user to turn his or her head to better discriminate the location of the sound.

We mimic the head shadow by differing the amplitude of the sound waves between the two ears so that full intensity is delivered when the user faces the source with a 0° difference. As the user rotates away from the source direction, the amplitude of the PWM wave channel furthest from the source is reduced while the closer channel maintains full intensity.

According the equation in Figure 1, the phase difference between the ears should vary sinusoidally. However, for this project, the phase difference varies linearly. Accuracy is most important for angles directly in front of the user because that is what the user bases his heading upon. Sound sources extending beyond ±15 degrees do not have to be as accurate because the user will intuitively turn his head to gain a more accurate indication of the location of the sound source. The difference between increasing the phase linearly versus sinusoidally can be seen in Figure 5. By using a linear model, as opposed to a sinusoidal model, we gain better resolution in the ±15° range that is most important.

INTEGRATION & TIMING

Timing the entire system was a challenge as the data from the GPS module was continuous and the amount of data received from it depended on how many satellites were locked on. This meant that we had to receive GPS data, get the compass data, do the calculations, and output the audio all before the new GPS packet was received. The three parts (GPS, compass, and audio) were all run off of separate interrupts that each had their own timer. They were all staggered from one another so they would not overlap.

We start the timing on the first packet received from the GPS and let the USART interrupt run until the entire RMC sentence is received. Once the RMC sentence is received, we turn off the USART interrupt so we don't interrupt all the characters of

Photo 2—Dr. Bruce Land on his trek to an unknown destination. We're happy to note that he got within a meter or so of the actual location. From the looks we received from other students, we nicknamed the device "the nerd helmet."

the other NMEA sentences. We then enable the interrupts for the compass function which takes approximately 60 ms. As soon as the compass function ends, the sound function and its interrupts are run. The sound function will vary in length based on its output, but it should never take more than 150 ms. We then repeat the compass and sound functions again before turning on the USART interrupt, which will wait for the next RMC sentence. Running the compass and sound functions twice leaves enough time for the USART to be enabled before the next RMC sentence arrives. This also enables two sound pulses per second, which is more than enough to get a good sense of direction.

We used two active low-pass filters to produce the two lines of audio signals from the two PWM outputs. We made the filters active so they could drive the headphones. The wire routing was kept as compact as possible, meaning that data lines ran close to the audio, introducing a lot of noise. Using a custom printed circuit board (PCB) could remedy the problem, but that was out of the scope of this project. A potentiometer was used on each channel for the user to calibrate the volume upon start-up, ensuring equal volumes when pointing to the destination. Also, an LCD screen was used to select the initial destination. This was done with three buttons: Screen up, Screen down, and Select. These were implemented active low with a 10-kΩ pull-up resistor and a 330-Ω resistor current protection.

TESTING & RESULTS

The results of the project were successful. A GPS program on a smartphone was used to get desired GPS coordinates that were then programmed into our device. This enabled the user to select between a few different points around the Cornell campus through the use of the LCD screen and push buttons. Once the GPS got a satellite lock, users understood that they needed to follow where the beeps were coming from with almost no instruction.

The device was tested on multiple students with the final test directing Dr. Bruce Land, a senior lecturer in Electronics and Computer Engineering, from the Engineering Quad to the Big Red Barn—approximately 0.4 miles (see Photo 2). Without giving him instruction as to how the device worked, he was able to get within a meter of the desired location. The end point was apparent to him because the device began to circle around the desired location. Videos of his voyage can be found on the ECE 4760 course website (see Resources).

FOLLOW THAT SOUND

The inspiration for this project was the idea that we could mimic a natural biological function in order to guide a human in a way that is simple for the user to understand, without requiring any thinking to distract the user from other tasks. The hope was that the user would instinctively understand what the system was trying to communicate. We were pleased to see that exact outcome when testing the system. This was best exemplified by a student who happened to be passing us by outside while we were holding the device. When he inquired about it, we simply told him to put it on (see Photo 3). Without giving him any context of the situation, he immediately turned and pointed to where the sound was coming from, even though we never mentioned that there was a direction that he should be trying to find. It just seemed like his natural instinct to respond in the way that he did.

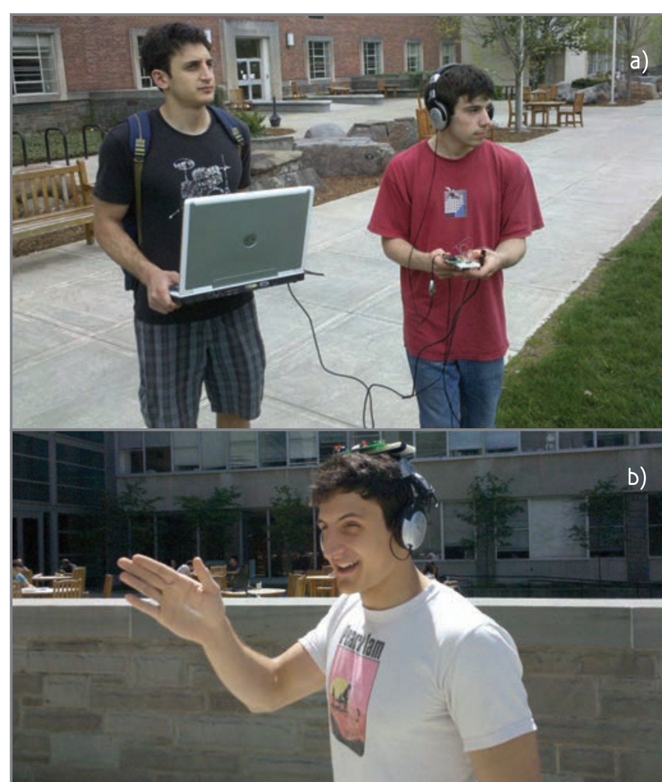


Photo 3a—Two students are using the GPS to direct a random student (b) who is wearing the "nerd helmet."

Moving forward, it's easy to imagine that many different features could be added to this, one being a more pleasing sound, such as a prerecorded voice. Also, waypoints could be added, as currently, the user is directed "as the crow flies." It is also possible to add a vertical axis that would enable the user to hear sound in three dimensions, not just two. Our sound model currently limits us to the 90°-to-270° range for the heading. Although it is not necessary, it would be an interesting but challenging improvement to add a complete 360° range. With a larger budget and more time, we would use a faster microcontroller to make the sound interval faster and smoother so the user can move faster and receive updates more quickly.

Garret Phillips (gp244@cornell.edu) received a BS in Microelectronic Engineering from Rochester Institute of Technology and an MEng in Electrical Engineering from Cornell University. He is an analog IC design engineer at Intel in Chandler, AZ. Garret's technical interests involve designing against non-ideal processing effects in tomorrow's CMOS technologies. In his spare time, he enjoys home brewing and developing embedded systems.

Matthew Kinne (mrkinne@buffalo.edu) is a Cornell University graduate, embedded programmer hobbyist, and devoted runner.

Nick Annetta (nva2@cornell.edu) earned a Bachelor's degree from University of Pennsylvania and a Master's degree from Cornell University, both in Electrical Engineering. He designs medical devices at Battelle and is a devoted pasta eater.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/256.

REFERENCES

[1] L. Rayleigh, "On Our Perception of Sound Direction," *Philosophical Magazine*, 1907.

[2] National Aeronautics and Space Administration, "Speed of Sound," www.grc.nasa.gov/WWW/K-12/airplane/sound.html.

[3] University of California Davis, Department of Electrical and Computer Engineering, "Psychoacoustics of Spatial Hearing," http://interface.cipic.ucdavis.edu/sound/tutorial/psych.html.

[4] F. Zeng, A. Popper, and R. Fay, *Cochlear Implants: Auditory Prostheses and Electric Hearing*, Springer, 2004.

[5] Movable Type Ltd., Movable Type Scripts, "Calculate Distance, Bearing and More Between Latitude/Longitude Points," www.movable-type.co.uk/scripts/latlong.html.

RESOURCE

Cornell University, School of Electrical and Computer Engineering, ECE 4760: Designing with Microcontrollers Final Projects, http://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/#s2010.

SOURCES

ATmega32 Microcontroller
Atmel Corp. | www.atmel.com

Hitachi HM55B Compass module graphical viewer software and PMB-248 GPS receiver
Parallax, Inc. | www.parallax.com

by Guido Ottaviani (Italy)

Sound Tone Detection with a PSoC (Part 1)

Analog Signal Management

The first part of this two-part series details the hardware part of an audio sensor. You learn how to mount the component to a printed circuit board (PCB) and configure the internal hardware modules to manage an analog signal.

My Rino robotic platform is continuously evolving. Being a development platform it will, perhaps, never be completed (www.guiott.com/Rino/index.html). As shown in my "Robot Navigation and Control" article series (*Circuit Cellar* 224 and 225, 2009), the platform can autonomously navigate in an unknown environment. In "A Sensor System for Robotics Applications" (*Circuit Cellar* 236, 2010), I explained how to build a dedicated subsystem to discover an obstacle on a robot path and find some targets using popular sensors readily available on the market. Now I want to focus on a specific kind of sensor, the most complicated one for people who are used to designing digital circuits: the audio sensor. Its design requires a good knowledge about analog circuits. Fortunately, modern op-amp chips help a lot in this type of design, reducing the complexity and instability typical of such circuits. But, you still need a good background in both practical and theoretical analog issues to avoid unwanted results. Once more, evolving technology helps us.

Cypress Semiconductor developed a programmable system-on-chip (PSoC) that contains several analog and digital blocks that easily connect together (see Photo 1). There are many "bricks" representing different kinds of functions and, like a Lego system, you have to arrange them in order to have a circuit that transforms the analog signal according to your needs. You still need some knowledge about analog electronic circuits; you must know the theory behind them. But, if you don't have enough practice building these kinds of circuits, you can still tackle such a project. You can use point and click instead of a soldering iron, you don't need to test many discrete passive components on a perfboard. You can simply reprogram your PSoC until the goal is achieved.

The PSoC is quite different from the usual microcontroller, CPU, or FPGA. It flashes a program inside to perform some tasks but it's not just a microcontroller. It mixes different electronic functional blocks, but it's not an FPGA. Its unique feature is that it has the potential to modify an analog signal keeping it analog and the characteristics of the analog blocks can be modified on-the-fly by the program. You have the alternative to use a digital-signal controller to sample your analog signal and apply

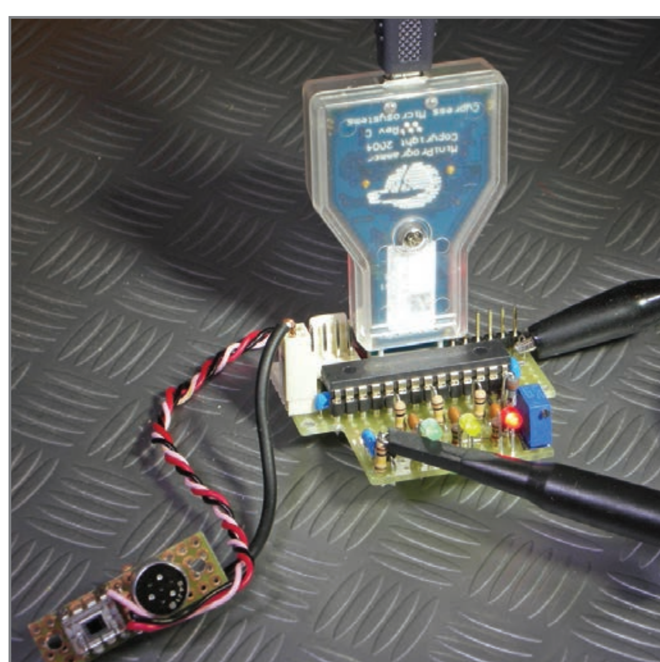


Photo 1—Cypress Semiconductor's PSoC and its programmer used in the real circuit

some mathematical transformations on it. But, you still need amplification, mixing, filtering, and amplification of the previous signal before sending it to an ADC for processing the signal by software. Often, a PSoC can do all this (and more) for you. The analog blocks are really analog! They are real operational amplifiers; they are not a software simulation of an analog behavior or a mathematical elaboration. The only digital stuff is that the capacitance can vary using switched capacitors technique. Adjusting the switching frequency and/or duty cycle, you can modify the capacitance, and therefore the parameters of the analog blocks, without adding external components. Looking at the electrical characteristics on the datasheets, you can find the same parameters of a normal op-amp datasheet: GBP, noise factor, common-mode rejection ratio, and so on.

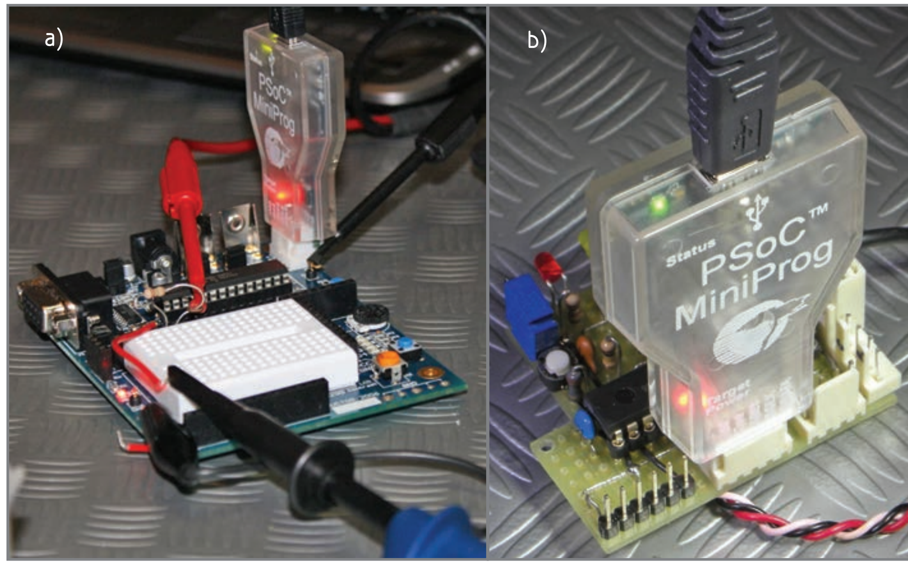


Photo 2a—You can buy a full evaluation kit with everything needed to start experimenting. b—Or you can buy the less expensive miniprogrammer to use with your own prototyping or final board. In both kits, there is a Cypress Semiconductor CY8C29466-24PXI PDIP PSoC device sample, which is perfect for our purposes.

As an example, there is a module that is unique on programmable devices—an instrumentation amplifier. This is one of the trickiest circuits to realize even with expensive op-amps. It requires good knowledge to obtain very high impedance, enough bandwidth, a good common-mode rejection ratio, and everything else needed to have a good measuring instrument. This is not achievable with a standard microcontroller.

I don't want to write a PSoC manual. I'm not going to explain every feature of PSoCs. I'll just explain how to build a tone-detector circuit with a Cypress device, focusing on some specific issues I've found. Even if you don't need a 4-kHz tone-detector circuit, the same basic principles I'm going to describe can be useful for many other types of projects. This could be considered "on-the-job training."

THE BASICS

Do you remember the circuit described in my sensor board article? In an explorer robotic challenge, the robot must autonomously find some sound sources that emit a 4-kHz tone. To achieve this goal, the robot has three microphones on three sides. The conditioning and revealing circuit is based on a quad op-amp and a tone detector. There is a first stage that mixes and amplifies the signals coming from the microphones, an active tunable band-pass filter, a second amplifier stage, and an NXP Semiconductors NE567 tone detector. The remaining op-amp in the chip is used as an active virtual ground reference for all the amplifiers, enabling higher impedance for their inputs.

At the very least, my first goal was to obtain the same functions with a single PSoC. I reached and exceeded that minimum goal, and I added some more functionality that I'll describe in this article.

First of all, what kind of PSoC? Let's start with the simplest and most affordable version, PSoC 1. This is an easy way to start understanding this device family without spending a fortune. The PSoC 1 series was the first series realized by Cypress and it's evolved noticeably both in analog module features and in digital block availability. There are many devices in this series, distinct for maximum clock frequency, RAM, and flash-memory quantity and for number of blocks. The latest one is the CY8C29x66 family—sold in different versions, from 24 to 64 pin—the only series available in a PDIP version, which is good for development boards (see Photo 2).

CONNECTING BLOCKS

Every book about PSoCs starts with, "Design and carefully test your circuit before you start writing the code." It's really true! When you build your project, after connecting all the blocks needed, Cypress Semiconductors's PSoC Designer integrated development environment (IDE) defines the template for the program you are going to write. It collects all the libraries needed and creates the basic main.c file and all the "include" files with the name you gave to each block. Changing the block configuration after you have written hundreds of lines of code using those names and libraries, even if possible, can be painful.

The good thing is that you can test your analog circuit without writing a line of code. Connect programmable amplifiers, filters, and other blocks via the common buses available and you can display the results of your work with a scope after injecting the signal to the input. It's not a simulation; it's real.

If you are used to working with microcontrollers, the first approach could seem strange. You don't have a fixed number

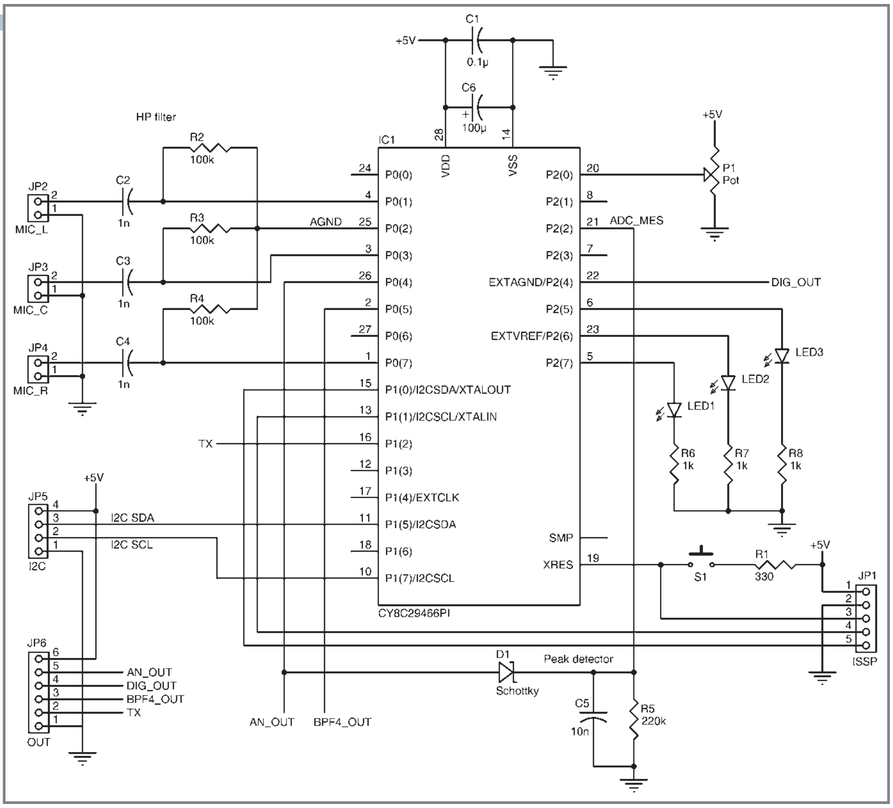


Figure 1—The "external" connections

of peripherals that can be connected to I/O pins according to your needs; you have a fixed number of boxes that can be filled with any kind of functional block. In this case, the limit is the number of boxes, not the number of blocks. The blocks can be connected internally with other blocks or externally with I/O pins. Not every combination is possible. There are a good number of buses and boxes, but you have to study the right compromise between your needs and the connections available. It can be confusing at the very beginning, but it becomes easier after a while. Be prepared to spend most of your time testing different connections and different configurations for each of the blocks before reaching the optimum solution.

Every operation is performed inside PSoC Designer. You can connect blocks, configure them, read the datasheet of each block, write and compile your program, flash the device, and more without going outside of the IDE. You can also configure everything manually editing the code; after all, it acts as a usual microcontroller with a huge number of registers for a lot of different configurations. You have to take care of all the compatibilities avoiding possible conflicts. Fortunately, PSoC Designer does all these things, creating the code for you.

At the end, you can also create an exhaustive report of the entire configuration with the block diagram and a detailed description for each block using the menu item "view-configuration datasheet." It creates a folder (ConfigDataSheet) with all the information and the picture for a complete documentation. Many of the pictures used in this article have been automatically generated with this function. It is interesting, even only as an exercise for the brain, to read some of the many application notes available on Cypress's website.

TRICKS & TIPS

Allow me to share some personal experiences with the hope of saving you some time and headaches. When copying text from one window to another, you cannot use standard ctrl-C and ctrl-V for copying and pasting. You must use drag-and-drop with a mouse between tabs. Fortunately, I'm a Mac OS X user, and drag-and-drop is more common than copy and paste for me, compared with Windows users. At least at the beginning, it is highly recommended to enable the "show allowed connections" feature (right click) to immediately know which connections are admitted for the line or module you are using.

Lastly, there is no big enough monitor when you are working with PSoC Designer, you have to zoom and pan often. To navigate more easily on the block diagram to place or connect modules and buses, you can temporary switch from standard to pan-mode view with an Alt key instead of clicking the menu item or using the right click contextual menu.

THE CIRCUIT

The "external" circuit is simply a bunch of passive components (see Figure 1 and Photo 3). On the microphone's side, there are three high-pass RC filters for DC decoupling and low-frequency noise filtering. Three current-limiting

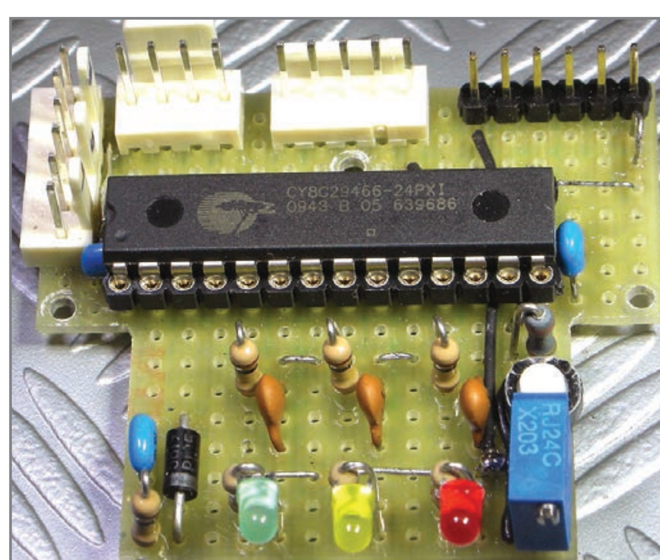


Photo 3—The circuit is simple. It is easy to wire it on a small piece of perfboard.

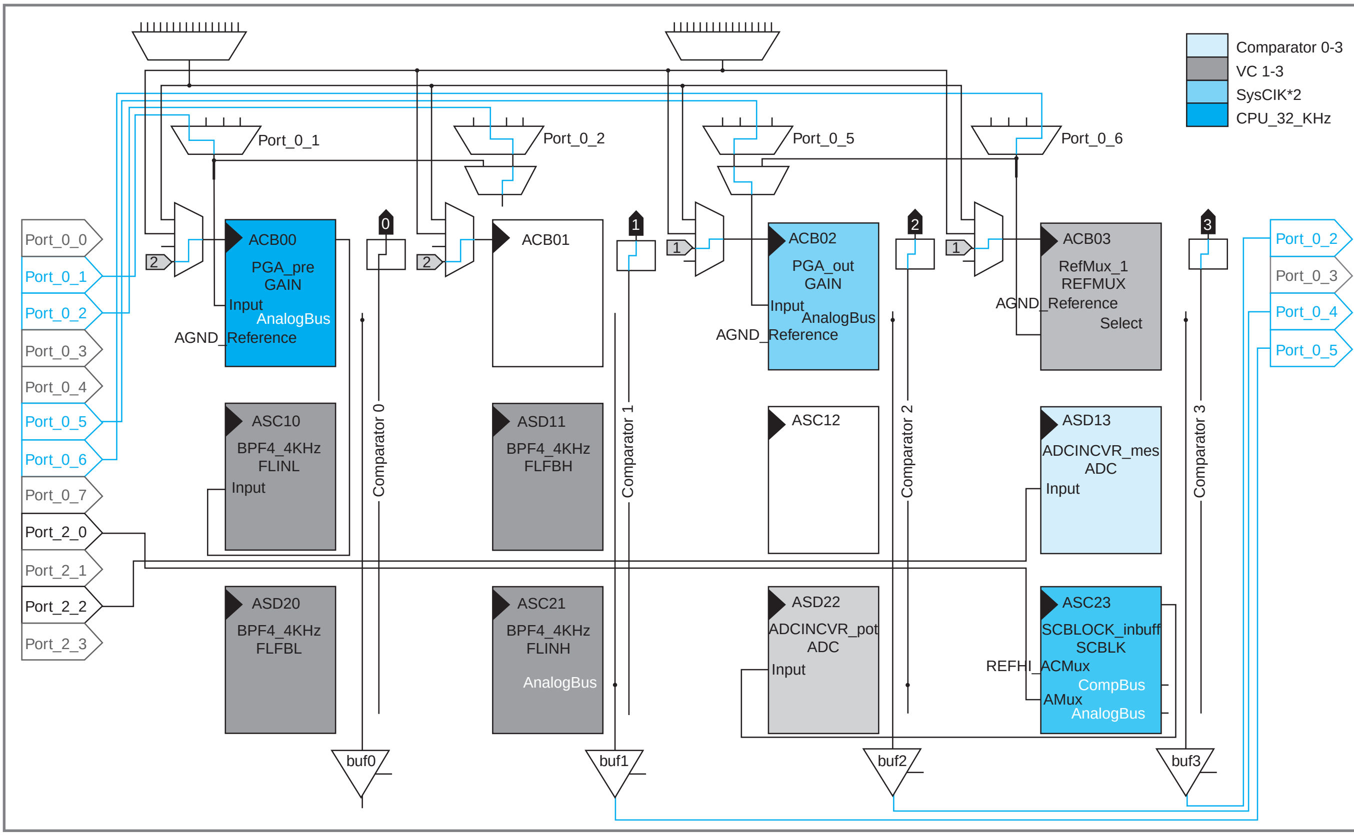


Figure 2—The analog blocks configuration with the connections to the internal buses and external pin drawn using PSoC Designer schematic conventions

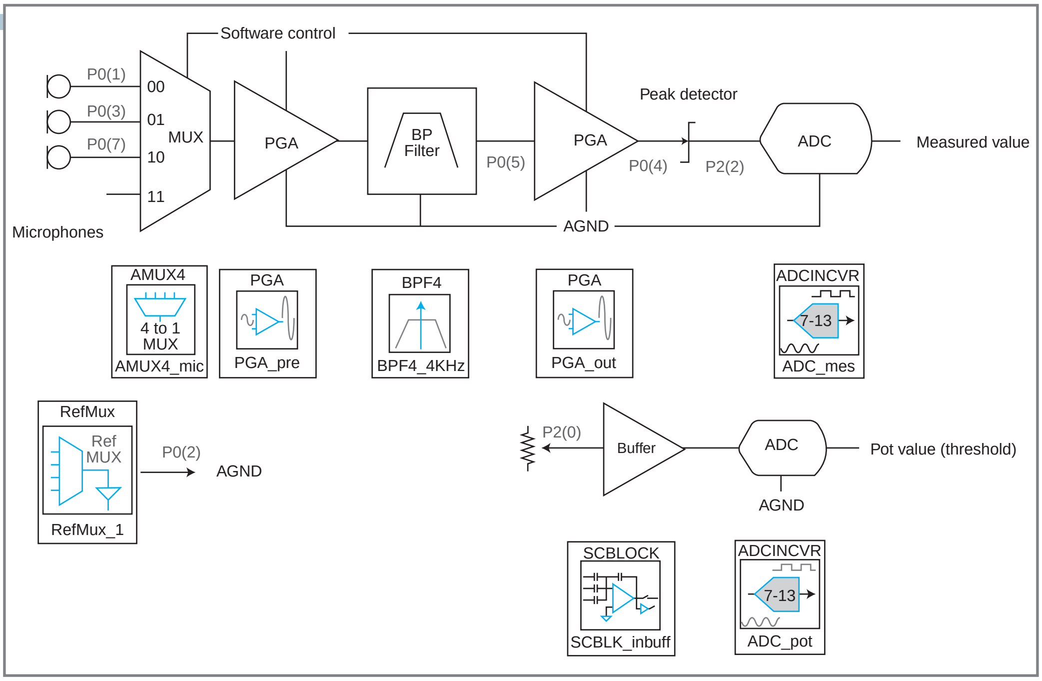


Figure 3—This is equivalent to Figure 2, but it's drawn using standard symbols.

specific I/O pin to the specific input of the block. Substituting a standard MUX with an AMUX4 or AMUX8, you can dynamically switch four or eight pins toward an input during the execution of the program using its API. This module doesn't fill up a block since it superimposes the function of an existing module.

In this configuration, ports P0(1), P0(3) and P0(7) can be sequentially addressed to PGA_pre.

PGA_pre refers to the programmable gain amplifier (PGA). The microphone's signals need a first amplification before entering into the band-pass filter. The ACB00 block accepts analog type-B module, its input is internally connected to the AMUX4_mic and its output can be internally connected to the analog switched capacitor block ASC10 without using any bus. It is perfect for our purposes. The characteristics of this amplifier are good for an audio-band amplifier—GBP up to 5.4 MHz for a 5-V power supply in high-power mode. The gain is programmable in several steps from 1/16 to 48. It is amazing to me to read about GBP, noise factor, bias, etc. in a programmable-device datasheet.

Using an amplifier with a software programmable gain enables us to realize an automatic gain control, expanding the dynamic of the amplification chain. This is another feature not present in the op-amp version of this circuit.

BPF4_4KHz refers to the band-pass filter. This is a

resistors drive the LEDs. A Schottky diode forms a peak detector together with a capacitor to measure the level of sound; this kind of diode causes a lower voltage dropout than normal rectifier diodes. The purpose of the R5 resistor is to decrease the discharge time of the C5 capacitor in order to enable a faster multiplexing of the three inputs. Those external components were chosen instead of a peak-detection block because all the internal available boxes are busy and this three-component network is an acceptable compromise. A trimmer is used to set up the threshold of the analog signal that triggers digital out. A trimmer? Why a trimmer when you can use an up/down digital circuit with an LED bar indicator of the position? Because it's much simpler. Sometimes I'm lazy.

Filtering capacitors and sockets complete the "bill of materials." The real complexity is in the "internal" part of the circuit, as expected using this PSoC device. Let's describe every single block and connection.

ANALOG BLOCKS

Figure 2 shows the analog blocks configuration with the connections to the internal buses and external pins drawn using PSoC Designer schematic conventions. Figure 3 shows the equivalent schematic diagram drawn using standard symbols.

The AMUX4_mic block is one of the improvements with respect to the older op-amp version. In that one, the microphones were all mixed in one single input, not enabling the identification of the sound source direction. With a multiplexer, we can use a single chain of filtering and amplification, assigning it to one microphone at a time.

All the analog continuous blocks (first row) are connected to the external world through a standard MUX or a combination of multiplexers. With PSoC Designer you can address a

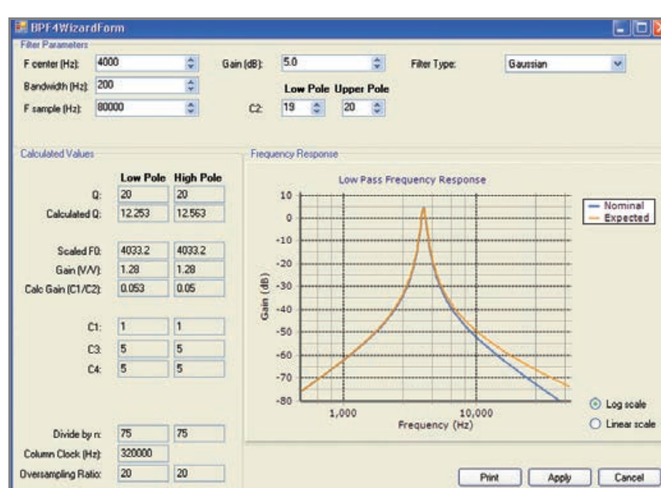


Photo 4—You can use the filter design to write down all the values for all the registers involved in your filter. Simply right click on a filter block to pop up the contextual menu.

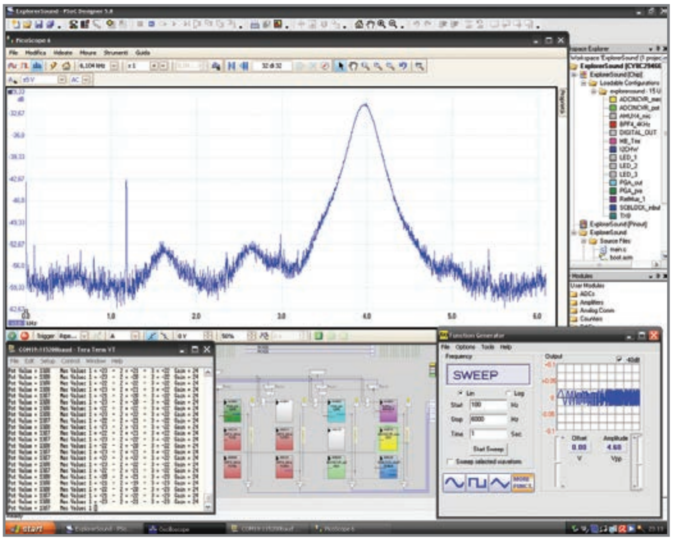


Photo 5—The graph captured by the DSO in Spectrum Analyzer mode, the function-generator window that sweeps a signal in the audio band, and a terminal window with the debugging values sent by the PSoC via the UART port. The background shows the PSoC Designer window with the project used for this test.

switched capacitor module that fills up four blocks. The output of BPF4_4KHz goes to AnalogBus 1 and out to Port_0_5. The Port_0_5 pin is internally connected also to AInMux_2 and routed to PGA_out for the final amplification.

These are the most interesting kind of modules. With switched-capacitor amplifiers you can build fully analog filters with the same characteristics of classic filters (e.g., Butterworth and Bessel).

Second-order filters fill up two switched-capacitor analog blocks (ASD and ASC). Fourth-order filters fill up four blocks. Remember, there are some connection constraints that limit the number of practicable filters. Only second and third rows accept switched-capacitor blocks. Only the first and third columns can be connected to an ACB block, the only one that enables connection to the world external to the device.

Not every filter topology is available on every PSoC 1 family. For example, Cypress's eight-pin CY8C27143-24PXI PSoC device, even with 12 analog blocks, admits only BPF2 and LPF2 filters. All available filters topology are band pass of second order (BPF2), band pass of fourth order (BPF4), elliptical low pass of second order (ELPF2), elliptical low pass of fourth order (ELPF4), low pass of second order (LPF2), and low pass of fourth order (LPF4).

We need some basic theory about the switched-capacitor technique to understand what we are going to do. Switched cap blocks use the same op-amp used in the continuous analog blocks. This op-amp then has some capacitors driven by switching signals placed in feedback, input, and output positions to serve various functions such as integrators, summers, and filters. The timed switching signals give the capacitors the ability to function the same as a resistor would in the circuit. The value of the capacitance along with the speed of the switching enables the user to vary the correlating resistance value.

This is the heart of the switched-capacitor technology. Since switching is involved, the output signal you would expect isn't continuously present and must be sampled at specific times in order to see the correct signal. The construction of the PSoC switched-capacitor blocks enables you to process these signals easily in filters and put them out to the analog output buffers or use the blocks as analog-to-digital converters without being a switched capacitor expert.

Once more, you can write down all the values for all the registers involved in your filter or use the practical filter design wizard by right clicking on a filter block to pop up the contextual menu (see Photo 4). This wizard suggests the capacitance values needed, after entering the characteristics of the filter you are going to realize. The resulting filter frequency response is shown in both theoretical and effective form. As happens in classical discrete component filters, you have to slightly adjust values to obtain the better compromise on frequency bandwidth, phase, and amplitude flatness.

One of the important things to note about designing a filter is that the cutoff frequency of the filter is based on the clock frequency being sent to the analog columns that contain the filter. The filter design wizard does not set up resources to generate this frequency for you. It simply tells you which frequency is important. The specified frequency must be sent to all switched-cap blocks used in the filter. You will be required to create a clock using digital resources to send to the analog column. This clock can come from timers, counters, and so on. If the clock is attainable by using the VC1, VC2, or VC3 signals, you can reserve your digital blocks for other uses. Since there are limits on what clocking frequencies can be set to the blocks, there are limits on what frequencies you are able to choose for your filters. That is the reason why you won't find a high-pass filter design in the PSoC. Unfortunately, this is the nature of switched-capacitor designs.

Photo 5 shows the response, in the frequency domain, of the realized band-pass filter. The screenshot shows the graph captured by the DSO in Spectrum Analyzer mode, the function-generator window that sweeps a signal in the audio band, and a terminal window with the debugging values sent by the PSoC via the UART port. The background shows the PSoC Designer window with the project used for this test.

Another PGA of the same type is PGA_out. This one amplifies the signal at the out of the BPF4_4KHz. It is also software controlled, which enhances the dynamic of the entire chain even more.

PGA_out is the output that is connected to AnalogOutBuf_2 and routed to Port_0_4. The filtered signal is therefore available on Port_0_4. Port_0_4 is connected to Port_2_2 through a Schottky diode and a capacitor in order to rectify the 4-kHz signal. The DC signal out of the rectification circuit on Port_2_2 is routed to ADCINVR_mes to measure via software the level of the 4-kHz signal present at the input.

With the scope placed on Port_2_2, you can see the signal (see Photo 6). The configuration is the same as in Photo 5, with the DSO in Time Domain mode (oscilloscope). The sampling frequency of the switched capacitor block is clearly visible, as previously described.

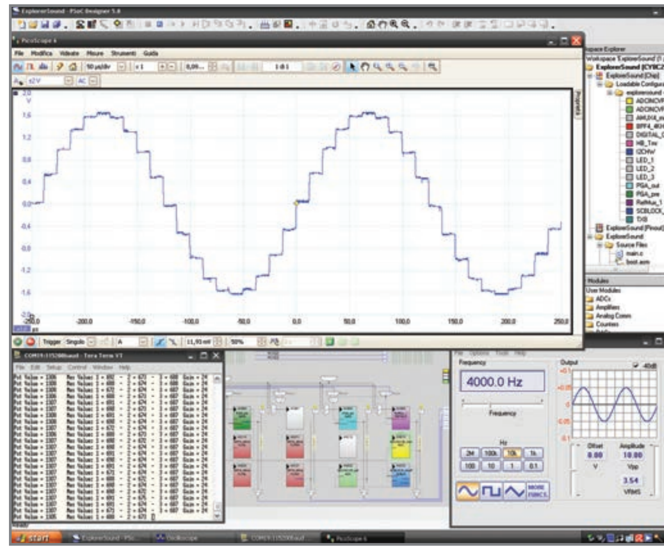


Photo 6—With the scope placed on Port_2_2, you can see the signal. The configuration is the same as in Photo 5, with the DSO in Time Domain mode (oscilloscope).

RefMux_1 refers to the analog virtual ground reference. This is routed to AnalogOutBuf_3 in order to obtain an AGND virtual ground reference for the inputs. It is connected to them through a 1-MΩ resistor. Another interesting module available on PSoC devices, this enables the generation of a virtual ground with a level between VDD and VSS. Anyone who has been involved in analog circuit design with a single power supply knows how much care must be dedicated to this part of the circuit. A bad design can damage the signal introducing distortion or noise. A passive network virtual ground reference can lower down the input impedance at unacceptable values.

Studying several application notes, something caught my attention. They always use a resistor between inputs and AGND reference. Why? All analog blocks should be automatically referenced to AGND internally, what is the purpose of that resistor? On a classic op-amp there is always some feedback resistor to the input that brings it to the right bias. A switched-capacitor amplifier input, decoupled with a capacitor to the external circuit, is really floating with the risk of having an unwanted DC bias or amplifying a noise. It needs a resistor between each input pin and the virtual ground reference to fix the bias.

AGND reference can be realized in different ways. As in normal op-amp, when no more pins are available on your device, you can use a voltage divider with two resistors and a capacitor from VDD (a Cypress Semiconductor AN16833 PSoC). In application note AN2247, Cypress simply uses an unused OutBuff connected to an output pin. If a block and a pin are available, the best way is what you can see on this circuit: configuring a RefMux with AGND as input and OutBuff as output to an external pin, this one becomes a clean and stable AGND reference for all of the circuit.

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)

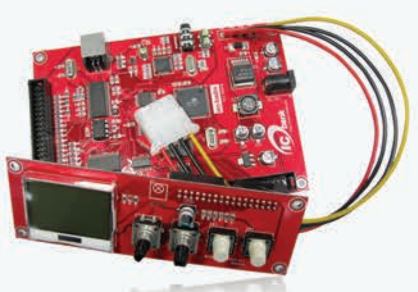
· myWave (MP3 DIY KIT SD card Interface)

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

· myAudio (MP3 DIY KIT IDE)



Only
$180 qty 100
$220 qty 1

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

Powerful feature

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

ICbank
Gateway to Electronics

www.icbank.com

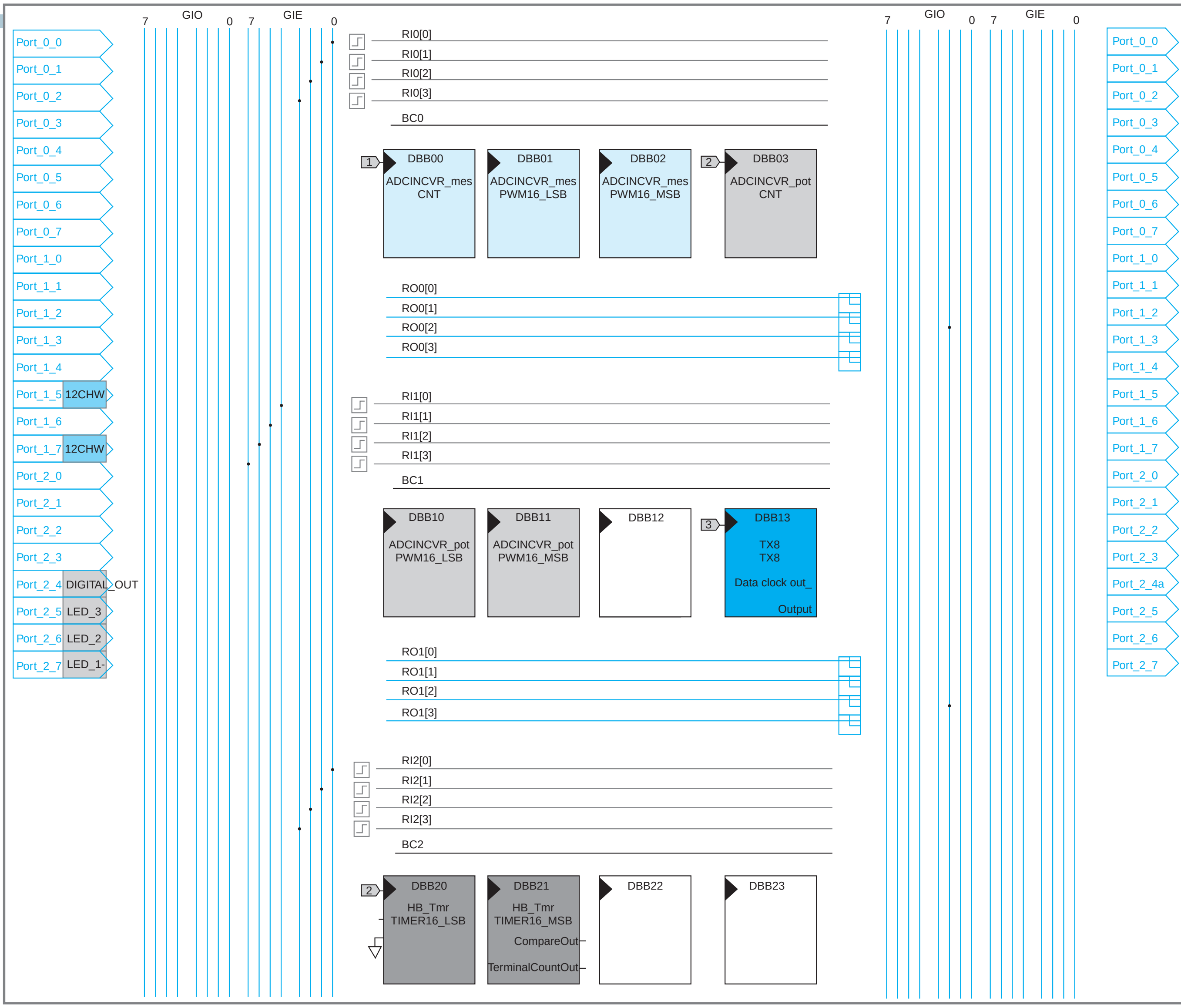


Figure 4—Digital blocks diagram

SCBLK_inbuff, the generic SCBlock, is just used as a buffer with a gain of one to connect the external Port_2_0 with ADCINCVR_Pot ADC. This is a trick that popped up after hours of tests on how to connect the port on which the trimmer is with an ADC that can only be placed on a fourth column because the other columns are already busy. It was also a good exercise for understanding an SCBlock. I had undervalued this generic block because it was too "generic," and also because the datasheet is too poor.

After looking at application note AN16833, where a four-channel mixer with independent adjusting for each channel is realized with a single PSoC 1, I found the AN2223 that better explains this module, with relations to standard op-amp characteristics. It's very useful. It's a real op-amp, programmable in several different modes.

Someone once said, "For those who are really adventurous, Cypress has created the generic switched cap block. This block gives you a visual configuration of all the settings of a switched capacitor block to allow you to create your own module easily." It's partially true, indeed. The placement is not trivial, not every connection is possible. There are several restrictions depending on the placement of the block and sometimes some external links are needed, involving an analog bus. But it's really versatile. Unlike classic op-amps, you can also have two inverting inputs.

ADC_mes are, of course, mixed blocks. Part of their function is in the analog section and part is in the digital section. There are many different kind of ADCs available on a PSoC 1. Choosing the one right for your purposes can be tricky. Fortunately, Cypress's documentation is exhaustive with datasheets and a lot of application notes. Discussing in detail the reasons for choosing one or the other is beyond the scope of this article. For this purpose, an ADCINVR with 11-bit resolution is the right compromise between needs and block occupation.

ADCINCVR_mes is used to read the input signal as aforementioned. The ADCINCVR_pot is used to convert the position of a trimmer in order to manually (and optionally) set a threshold for the trigger part of the circuit.

According to the formulas in the datasheet, both types

of ADCINCVR_ perform a full measure in 1.7 ms with a 4.8-MHz clock (VC1). These parameters are chosen to have an integration time that filters out some undesired signals. A spreadsheet for calculating parameters is available on the *Circuit Cellar* FTP site.

Like any other analog circuit, and because the maximum possible amplification gain is quite high, it is highly recommended to have a very clean power supply to avoid amplification of noise that could interfere with the real signal. A dedicated linear-voltage regulator or an LC low-pass filter could be used to filter out undesired signals, primarily if the sound circuit is on the same power line of digital circuits.

DIGITAL BLOCKS

Figure 4 shows the digital blocks diagram. These modules are probably more familiar to standard programmable device users. The diagram also uses not so standard conventions to show how the digital modules are connected to the internal buses or to the external pins. This is due to the "not so standard" way that Cypress approaches this issue and to the huge amount of possible connections the PSoC has. This could be a little bit confusing for designers used to working with other kinds of microcontrollers but, after some practice, it quickly becomes easy to manage.

Some boxes in the diagram are filled with the digital counterpart of the ADCs.

The HB_Tmr is a 16-bit timer. It fills up two boxes because it is a 2-B timer. It is the base clock for all of the timings in the program. Its timing is:

$$VC2 = \frac{VC1}{15} = \frac{\frac{SysClk}{5}}{15} = \frac{\frac{24\text{ MHz}}{5}}{15} = 320\text{ kHz}$$

$$\text{Timer Period} = \left(\frac{1}{VC2}\right) \times (\text{PeriodRegister} + 1) =$$

$$0.000003125 \times (3{,}199 + 1) = 10\text{ ms}$$

TX8 refers to the serial transmission module. It enables transmission via the UART of the analog values read from various inputs. It can be useful for debugging purposes using standard ASCII characters to a serial console or, using any kind of protocol (e.g., dsNav communication), it could be used to interface other boards alternatively to I^2C port.[1]

It is notable that even only the TX part of the UART module can be used if no data has been received, as in our case. This saves a box and an I/O pin that can be used for other purposes.

Here is some information about digital OUTs. To drive an LED or a generic I/O it is highly recommended to use an LED module. This doesn't fill up a box, it is just a kind of library that correctly drives the I/O port. It can be found under the MiscDigital section. Not using this module requires a direct control of the port using the shadow registers. The module can be configured in Active High or Active Low mode. The first mode must be used when the cathode of the LED is connected to VDD and a high level on the anode will switch it on. The second mode requires an LED with the anode connected to VSS and a low level to the cathode will switch it on. In both cases, the "LED_On" instruction is used to switch the LED on.

The I/O ports can be used even if they are not connected to a block, after all, this is a microcontroller too. A pin configured as "StdCpu" can be used in the program using a mnemonic name.

Every line can be configured in different modes, which enables a lot of flexibility. They accept Strong-high, Strong-low, High-impedance, Pull-up, Pull-down, Open-drain-high, and Open-drain-low modes. Almost everything is possible, even configuring a pin in both Input and Output mode. This requires good practice with PSoC that can be achieved in the field or by a carefully reading many application notes.

In a single PSoC family, there are many kinds of different devices with a different number of I/O pins, but all with the same number of analog and digital blocks. All other pins can be used as input/output like any standard microcontroller. All the outputs can directly drive devices that require several milliamps, such as LEDs or something similar.

EzI2Cs refer to the I^2C communication module. This doesn't fill up a box, you have just to assign a clock and data line to a couple of pins to start using this protocol.

The basic circuit described above performs a basic task, but this approach simplifies much of the design if compared with an equivalent fully analog circuit. In this article, there is a full description of how this specific board works, but there is also most of the information you need to start using PSoC devices when developing similar circuits. The complete and up-to-date PSoC Designer project is available on my Google Code space.

HARDWARE OR SOFTWARE?

It may sound bizarre defining software to hardware differences in a device completely configured through a computer. You are changing some physical parameters via a graphical user interface (GUI) on a computer window. In the same way you can change the flow of a program with a conditional jump, you can optimize the phase response of a Butterworth filter—without a screwdriver on a trimmer or a solder iron on a capacitor—just with mouse and keyboard. And it is not a simulation, it is real world.

But let me be conservative to better explain this innovative device. Let's use traditional terms. In the first part of this article series, I described the hardware part of the project, how the component can be mounted on a printed circuit board (PCB) or on a development board, and how the internal hardware modules can be configured to manage our analog signal in order to be useful for our needs.

Now you have the correct signal at a correct level. What can you do with it? Let's apply some "real" software, at least according to the traditional description of this term.

In the second part of this series, I'll describe the software capabilities of the PSoC 1 device and how to use them to measure the level of the signal, expand the dynamic of the amplifying chain, and reliably share this information in several different modes with the external world. ■

Guido Ottaviani (guido@guiott.com) has made electronics and ham radio his hobby since the "tube times." He turned the hobby into a job working as an analog and digital developer for several years for an Italian communication company. Many years ago, a big change made him a technical manager at a company that develops and manages graphic, pre-press and press systems, and technologies for a large Italian editorial group that publishes sports newspapers and magazines. Some years ago, he got the scope and the soldering iron out of the drawer to make autonomous robots. Currently, Guido is an active member in various Italian robotics groups who shares his experiences with other self-professed "electronics addicts" and evangelizes amateur robotics.

PROJECT FILES

To download the "ADCINVRcalc.xls" spreadsheet for the calculation of parameters, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/256.

REFERENCE

[1] dsNav Communication, www.guiott.com/Rino/CommandDescr/Protocol.htm.

RESOURCES

Cypress Semiconductor Corp., "PSoC 1 Overview," 2011, www.cypress.com/?id=1573.

———, Cypress PSoC 1 Application Notes Database, www.cypress.com/?app=search&searchType=keyword&keyword=&rtID=76&id=1573&applicationID=0&l=0.

Google, Explorersound, a PSoC-based system for 4-kHz tone decoder, http://code.google.com/p/explorersound.

G. Ottaviani, "Robot Navigation and Control (Part 1): Construct a Navigation Control Subsystem," *Circuit Cellar* 224, 2009.

———, "Robot Navigation and Control (Part 2): Software Development," *Circuit Cellar* 225, 2009.

———, "A Sensor System for Robotics Applications," *Circuit Cellar* 236, 2010.

SOURCES

CY8C29x66, CY8C29466-24PX, CY8C27143-24PXI, AN1683, and AN2223 PSoCs, CY3210-PSoCEval1 and CY3217-MiniProg1 evaluation kits, and PSoC Designer (IDE)
Cypress Semiconductor Corp. | www.cypress.com

NE567 Tone detector
NXP Semiconductors | www.nxp.com

by Clemens Valens (France)

DesignSpark chipKIT™ Challenge

Super Arduino

Getting Started with the chipKIT Max32

If you're looking for more computing power on an Arduino-compatible microcontroller board, Digilent's chipKIT Max32 may be just what you need. Digilent integrated a compiler, linker, and programmer for the Microchip Technology PIC32 processor used on the chipKIT boards into the Arduino IDE. Read on to find out more.

If you're interested in microcontrollers, you've probably heard of Arduino—maybe you've even worked with it. If you have, you may have run into the limits of this friendly 8-bit platform and wished for more computing power, memory, or I/O pins. This is, of course, easily possible, since there are microcontroller boards galore, but at the cost of learning new tools. Or so it was until recently, when Digilent introduced its solution for those who want more power without changing tools. Its chipKIT Max32 offers 32-bit computing power and some 80 I/O pins while being compatible with the Arduino environment (see Photo 1).

There have been other attempts at making 32-bit Arduino-compatible boards, but (as far as I know) these attempts only achieve Arduino form-factor compatibility, not tools compatibility. Some of these boards are supported by software libraries that offer Arduino-like functionality and syntax, but they need a different compiler and use other firmware loading techniques. Digilent has taken Arduino compatibility a step further by integrating the compiler, linker, and programmer for the Microchip Technology PIC32 processor they use on their chipKIT boards into the real Arduino 0022 integrated development environment (IDE). (For more information, refer to the sidebar, "Introducing the PIC32.") From an Arduino IDE point of view, the chipKITs are just targets, listed together with the classic 8-bit Arduino boards. Digilent even went so far as to create a website with a URL ending in .cc like Arduino (see Resources). Also, in the spirit of Arduino, the chipKIT is completely open source and hardware. The hardware design files (Eagle schematic and PCB files) are available for download and the software is all open source. Unlike Arduino, the chipKIT printed circuit board (PCB) is a four-layer board, so few people will be tempted to roll their own.

There are two chipKITs: the Uno32 and the Max32. Mechanically, the Uno32 is compatible with the classic Arduino Uno and the Max32 is compatible with the Arduino Mega, the longer version of the standard Arduino. Note that Digilent made the boards rectangular by straightening the strangely shaped short edge of the Arduinos so they are slightly larger. (I will offer a Max32 board to the person who can tell me the real reason why the Arduino boards are not rectangular.) I don't think this will be a problem for anyone.

The rest of this article will concentrate on the Max32. Here we go!

THE BOARD

The Max32 comes as a red four-layer PCB in a little mainly red and white box without anything else, no USB cable, no documentation, just a URL. For those who do not know the Arduino Mega dimensions by heart: the board measures 4″ × 2″. Three sides of the board are lined with connectors, like an Arduino Mega, except that the

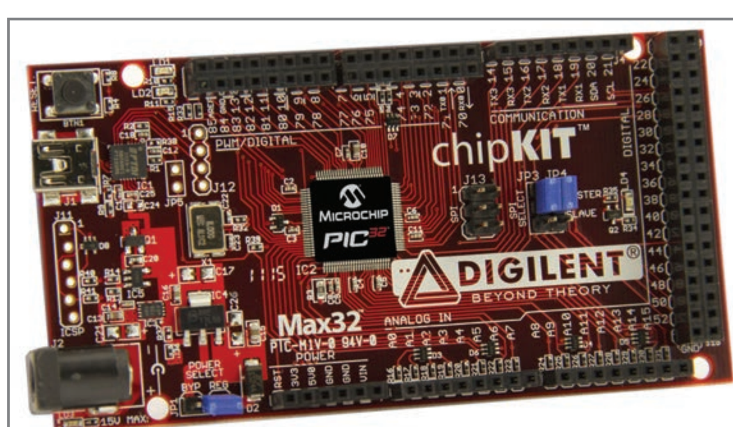


Photo 1—Digilent's chipKIT Max32

connectors for the digital pins 0 to 13 (Arduino speak) are double-row on the Max32. You will find digital pins 70 to 85 on the extra contacts. On the USB and power connector side there is room for a Microchip Technology ICSP connector. This connector has the special Sparkfun "out-of-line" placement of the pins so that a pin header will make good contact even without being soldered.

Powering the board can be done through the USB connector or using the power barrel jack that will enable input voltages up to 15 V. Jumper JP1 lets you bypass the 5-V regulator, so be careful what you do here or you may blow out some chips. When you power a virgin board, you will notice an annoyingly bright red LED next to the power jack indicating that the 3V3 line has a voltage on it and a green LED (LD4) blinking at some 3 Hz.

The Max32 is hardly more than a breakout board for the 100-pin PIC32 processor that is mounted close to the center of the board with a power supply and a USB serial port on the side. The 32-bit PIC32 compares pretty well to ARM's Cortex-M3 (see Sidebar). The board sports the largest PIC32 device currently available, Microchip Technology's PIC32MX795F512L. This chip sits in a 100-pin package and features 512 KB of flash memory with 128 KB of RAM and runs from an 80-MHz clock. It has USB on-the-go (OTG), an Ethernet MAC, and two CAN controllers.

You are familiar with the hardware. Now let's take a look at the software.

THE IDE

As I mentioned in the introduction, programming the board is done using a modified Arduino IDE that you can download as a free archive from Diligent's website. Installing the 128-MB file is simple. You only have to unpack it to a convenient place on a convenient disk on your computer. Unpacked, it takes up some 480 MB of disk space. To start the IDE, launch the executable mpide.exe in the root of the IDE's folder. The IDE is cross platform and will work on Windows, Linux, and MacOS, although you might have to install Java first.

By basing the Max32 on the Arduino IDE, the people at Digilent saved themselves a lot of documentation writing. Indeed, all you need to know to install and get started with Arduino is explained in detail on Arduino's website. You can also go to the website with your questions about the programming language syntax.

At the time of this article, the latest version of the IDE is 0022 (mpide-0022-chipkit-win-20110619, to be precise), the same as the current official Arduino IDE. According to Digilent, this IDE is exactly the same as the official Arduino IDE except that it has been extended with a PIC32 compiler/linker and libraries and it can be used to program 8-bit Arduino boards, too. Of course I tried this only to discover that my Arduino clone, a Seeed Studio Seeeduino v1.1, was not recognized: invalid device signature. This board works perfectly fine with the official Arduino 0022 IDE.

After installing the IDE and hooking up the Max32 to the computer, you can try the tools by compiling one of the simple examples included with the IDE and uploading it to the board. Don't forget to select the Max32 board from the Tools, Board menu and set the serial port properly (Tools, Serial Port). Once done, the BlinkWithoutDelay example (File, Examples, Digital) should work without modification, just click the Upload button to see the green LED LD4 start to flash at a rate of 0.5 Hz.

INTRODUCING THE PIC32

When asked about 32-bit microcontrollers most people will first mention ARM, then some ARM-core implementers such as Atmel, STmicroelectronics, or NXP and only a few will think about Microchip. Although it's true that many mobile phones are built on ARM technology, a lot of other common electronic devices (i.e., digital cameras and printers) contain MIPS-based processors. Now, I haven't done any serious checking on this, but it is said that there are far more 32-bit MIPS processors out there than there are ARM processors. Therefore, having some MIPS experience is a good thing for any serious microcontroller enthusiast and Microchip Technology's PIC32 is an excellent platform to get started.

There are currently five families: 3*xx*, 4*xx*, 5*xx*, 6*xx*, and 7*xx* (see Table 1). The 3*xx* and 4*xx* are considered general purpose, whereas the other three have more peripherals like CAN or Ethernet and can have more RAM. The devices are based on a 32-bit MIPS MK4 core with a five-stage pipeline and support clock frequencies up to 80 MHz. A maximum performance of 1.56 DMIPS/MHz (Dhrystone 2.1) is claimed, which is slightly better than an ARM Cortex-M3 that can do 1.25 DMIPS/MHz, according to ARM.

All families have up to 512-KB flash memory plus 12-KB boot memory and up to 32 KB RAM for the 3*xx*/4*xx* devices or up to 128 KB for the 5*xx*/6*xx*/7*xx* devices. They feature all the peripherals you would expect from this kind of microcontrollers (serial ports, PWM, ADC, etc.) but they also have multiple direct memory access (DMA) channels.

The families come in two "sizes," 64 pin (H suffix) or 100 pin (L suffix). Note that a 121 XBGA package contains a 100-pin device. The PIC32s are pin compatible with some PIC24 and dsPIC devices, so they integrate nicely in Microchip's big microcontroller family and development tools (MPLAB). A lot of software libraries are available on the website and a dedicated website exists for sharing PIC32 projects (www.mypic32.com). Datasheets and other documentation can be accessed at www.microchip.com/pic32.

Family	USB OTG	CAN	Ethernet	RAM
3*xx*	–	–	–	Up to 32 KB
4*xx*	1	–	–	Up to 32 KB
5*xx*	1	1	–	Up to 128 KB
6*xx*	1	–	1	Up to 128 KB
7*xx*	1	2 (1 for 764)	1	Up to 128 KB

Table 1—The five Microchip Technology PIC32 families

If you get this far without any problems, you're ready to develop real applications for the Max32. So read on, because there will be some hurdles you may want to be aware of.

Photo 2—Digilent's chipKIT Max32 IDE showing a `#define` to remove AVR-specific code

PORTING A SHIELD

Flashing an LED is nice, but not very satisfying. That's why I went on to try out my Arduino Ethernet shield (a shield is an extension card for an Arduino board) on the Max32. This shield is based on Microchip Technology's ENC28J60 stand-alone Ethernet controller with SPI. I know, the PIC32 has an Ethernet MAC inside, but I didn't have a shield handy with an Ethernet PHY and RJ45 connector. Digilent proposes such a shield (that includes some other things as well), but I didn't have the time to order one and wait for it to arrive. Besides, now I had a good opportunity to check out the Max32's compatibility. As you will see in what follows, it's not completely compatible....

My old Ethernet shield—I will call it "etherShield" from now on to distinguish it from the new official Arduino W5100-based Ethernet shield—is supported by a library and some examples. This shield and code work perfectly fine on my Seeeduino. Therefore, the first step was to install this library in the Max32 IDE (in the folder "libraries\") and see if it would compile. As you might expect, the answer was, no. The reason was not so much the code itself, but the fact that the compiler didn't seem able to find anything to compile. According to the Digilent website, where some things about porting Arduino code are explained, libraries are supposed to be handled the same way as Arduino does, but clearly not in my case. When I put the library in the folder "hardware\pic32\libraries\" (where you find the same items as in the folder "libraries\"), the compiler did find the code, but produced many errors and a warning saying that the code contained AVR-specific code. (Arduino boards are based on Atmel's AVR processors.) Great! Now I had clue.

The first thing you have to do when porting Arduino libraries is to get rid of the references to program memory. With an AVR, some special compiler directives are needed for accessing constants (e.g., strings, tables) located in program space. This is unnecessary for the PIC32 and these directives have to be removed. To keep your code Arduino compatible, it may be better to `#define` them away (see Photo 2). You can use the macro `_BOARD_MEGA_` defined by the Max32 IDE to do this (confusing, one would have expected something like `_BOARD_MAX32_`). Do the same for the AVR-specific `#include` directives.

This may not be sufficient. In my case, it wasn't, because the library may also refer to AVR registers that the PIC32 doesn't have. The SPI driver for the ENC28J60 Ethernet chip did this, probably because it is relatively old and the SPI library that is now part of the Arduino IDE did not exist yet when it was written. (It appeared only in version 0019, September 2010.) Therefore, I modified the old etherShield library to use the Arduino SPI library instead, and I tested this in the real Arduino IDE before trying it in the Max32 IDE.

This introduced new problems because the SPI library references pin functions that the Max32 compiler did not like. This turned out to be caused by the fact that my library was written in C and compiled as such, whereas the pin functions and the SPI library are written in C++. Files with the extension .c are compiled as C. Files with the extension .cpp are compiled as C++. So, back to the Arduino IDE it was for porting the complete etherShield library to C++ and testing it. This is not very difficult to do, but you may have to watch out for compiler directives such as `#extern "C" { … }` hidden in unexpected places. After this last modification, my etherShield library compiled without errors in the Max32 IDE *and* for the Max32 board. But did it work?

No. Of course not. This did not really come as a surprise as I had already spotted SPI problem posts on Digilent's website. But, the optimist in me kept hoping.

The main problem is incompatibility of Max32 I/O pins with AVR I/O pins. The Arduino digital pins 10 to 13 have SPI capability and pins 10 and 11 can also do PWM. The reason for this combination is simply the way the AVR I/O pins were laid out by Atmel. The PIC32 combines functions in a different way on its I/O pins and there are no exact equivalents for these AVR pins. Digilent has chosen to give priority to the PWM functions as they are used by Arduino's analogWrite functions, meaning that they could only connect part of SPI port 2 to these pins. However, they did find a way to connect SPI port 1 in an Arduino-compatible manner by using the Arduino ICSP connector which is connected to the same signals as the pins 10 to 13 (see Photo 3a). I had never considered this connector as part of a compatible shield, and I am in good company. But Seeed Studio, the maker of my etherShield, had put an ICSP pin header on the shield in the right place. Replacing it by a female header on the solder side only took a few minutes and restored SPI compatibility with the Max32. To prevent conflicts on pins 11 to 13 (MOSI, MISO, and SCK) I simply removed these pins from my shield (see Photo 3b). Now it would work, right?

Wrong. At this point, I fired up the oscilloscope because I suspected compatibility issues between the SPI protocol as produced by the PIC32 and the one accepted by the ENC28J60. Notably, the clock speed was worrying me a bit. The scope proved me right, because where the Arduino managed to reach a clock speed of about 610 kHz, the PIC32 was blasting away at 20 MHz. According to the ENC28J60 datasheet, this should be okay, but a later experiment when all was finally working showed me that 2.5 MHz was a more realistic value. For now, I just reduced the PIC32's clock

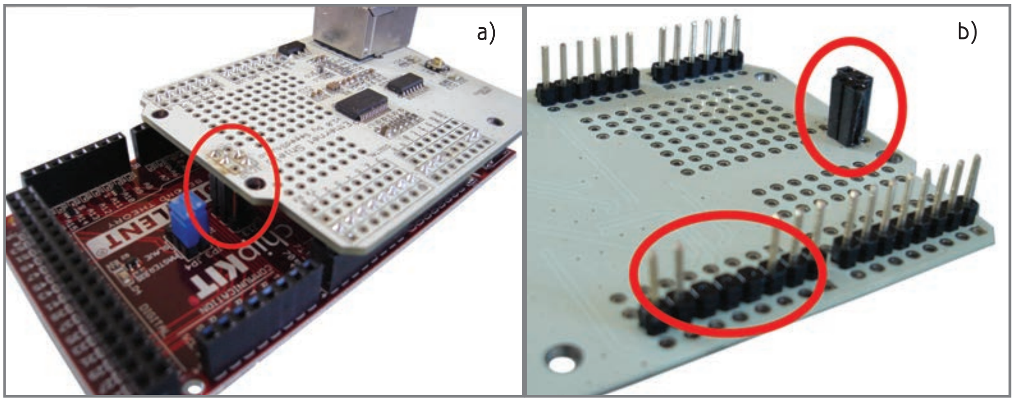


Photo 3a—A close-up of the SPI connection between the etherShield and the chipKIT Max32. b—These are modifications I made to the etherShield.

speed to an Arduino-like value: 625 kHz. With this change, the shield still didn't work, but I felt like I was getting close.

Nowadays, even the lowest budget digital oscilloscopes can record traces, including my $318 25-MHz Atten ADS1022C digital storage oscilloscope. This useful function showed me that there was a polarity/phase issue between the SPI clock and data lines. By carefully comparing transitions, I discovered that the shield needed SPI mode 1 on the Max32 whereas it used mode 0 on the Arduino. Did that mean that the shield was now finally working? It did. Phew. Refer to Photo 4.

USE AVAILABLE RESOURCES

Digilent has done a decent job of porting the PIC32 to the Arduino IDE. Although they did not achieve 100% compatibility, they definitely tried hard and managed to get pretty close. It's safe to assume that simple Arduino shields with simple libraries respecting Arduino coding rules and style will be easily portable, although you may run into some of the problems I encountered. The more complex shields exploiting AVR subtleties will definitely be more difficult and may require solid PIC32 knowledge. To make your life simpler, try to use the functions available in Arduino libraries as much as possible, and let Digilent do the hard work for you.

Digilent has started keeping a list of known-to-work shields, so look there first before starting a project yourself. Updates to the Max32 IDE correcting some of the issues mentioned in this article may be expected, so make sure you always use the latest version.

Photo 4—The result of a successful port: I can now connect to a little web browser running on the chipKIT Max32.

The only thing I haven't completely cleared up yet is the issue of where to put your own libraries. After thinking hard and experimenting a bit, I settled on the assumption that all files containing PIC32-specific code, such as low-level drivers, must be placed in the folder "hardware\pic32\libraries\" including the files that need these files. All other files, including the examples that use the library, must be placed in the folder "libraries\" to ensure that they are recognized by the IDE as examples. ■

Author's note: You can download the code associated with this article at www.elektor.com/110661.

Clemens Valens (c.valens@elektor.fr) is Editor-in-Chief of the French edition of Elektor. *He has more than 15 years of experience in embedded systems design. Clemens is currently interested in sound synthesis techniques, rapid prototyping, and the popularization of technology.*

RESOURCES

chipKIT, http://chipkit.cc.

Microchip Technology, Inc., PIC32 Design Community, www.mypic32.com.

SOURCES

Arduino 0022 IDE and Arduino Ethernet shield
Arduino | http://arduino.cc.

Cortex-M3 Processor
ARM Ltd. | www.arm.com

ADS1022C Digital storage oscilloscope
Atten Group, Inc. | www.attenelectronics.com

chipKIT Uno32 and Max32
Digilent, Inc. | www.digilentinc.com

PIC32 and PIC32MX795F512L Microcontrollers, ICSP connector, and ENC28J60 stand-alone Ethernet controller
Microchip Technology, Inc. | www.microchip.com

Seeeduino v1.1
Seeed Studio | www.seeedstudio.com

by George Novacek (Canada)

Shielding and Transfer Impedance

Electromagnetic interference (EMI) and electromagnetic compatibility (EMC) issues should be at the top of your list of concerns when designing both prototypes and finished systems. A clear understanding of shielding and transfer impedance, along with some careful planning, will help lead to a successful design.

Our world is permeated by electromagnetic (EM) fields. These fields are both man-made and due to natural forces in the universe. Man-made fields are the result of the operation of electrical equipment, some intentionally (e.g., by transmitters), others unintentionally as a by-product of the equipment operation.

From high school physics, we know that electrical energy will be induced in a conductor exposed to an EM field. If the conductor also carries a useful signal, the induced energy will corrupt. It may even obliterate this signal. We must, therefore, protect the integrity of signals by reducing the effects of the EM fields.

EM & SE

EM fields are made up of two elements: electrical and magnetic. This time we will focus on the effects of the electrical component, which is the most common source of interference.

Shielding interconnecting wires and cables is usually the only effective way to reduce the induced interference. Shielding effectiveness (SE) is one method of choice. SE is a measure of the unwanted signal attenuation in decibels (dB) which helps us to evaluate the circuit susceptibility to interference. Figure 1 shows the results of attenuation versus frequency measurements on a single- and a double-braided cable

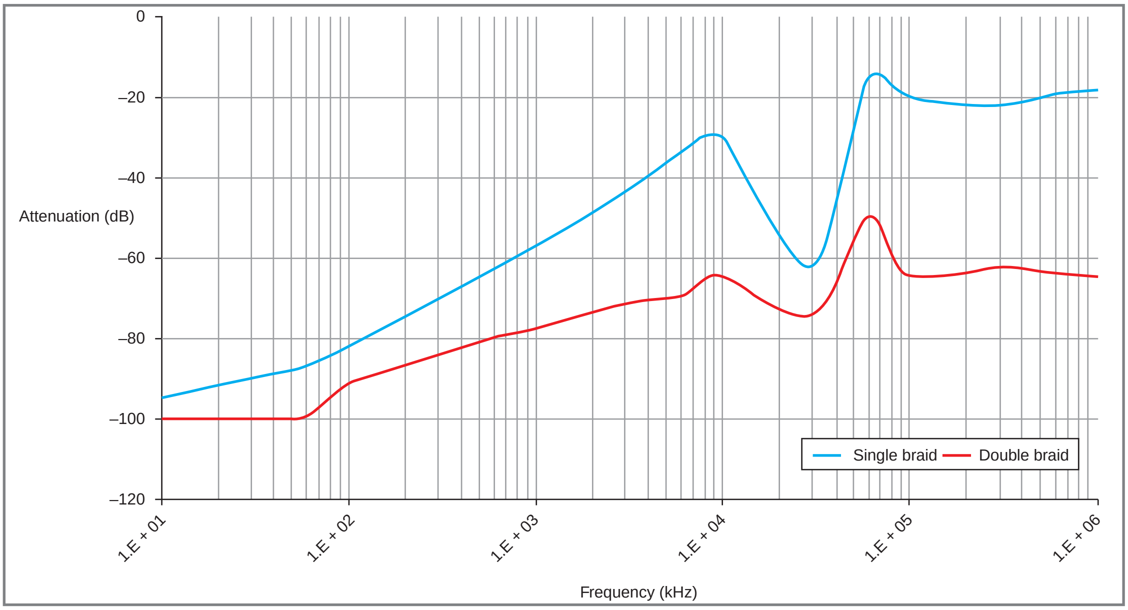


Figure 1—Shielding efficiency

as performed in a lab.

For better understanding, imagine you work in a darkroom. The EMI is an outside light and the shielding is an incompletely opaque window. You are interested in determining how opaque the window must be or how many such windows need to be stacked up to block enough light to prevent damage to a photographic material. As shown in Figure 1, SE is frequency dependent. A single-braided cable provides only about 26 dB (20×) attenuation at 10 MHz and a measly 20 dB (10×) from about 100 MHz up. This is something to keep in mind when working with fast computer interfaces and analog circuits even if their bandwidth is limited to a low frequency range, such as audio. Figure 1 also exhibits resonance around 30 MHz, which is often seen with shielded cables.

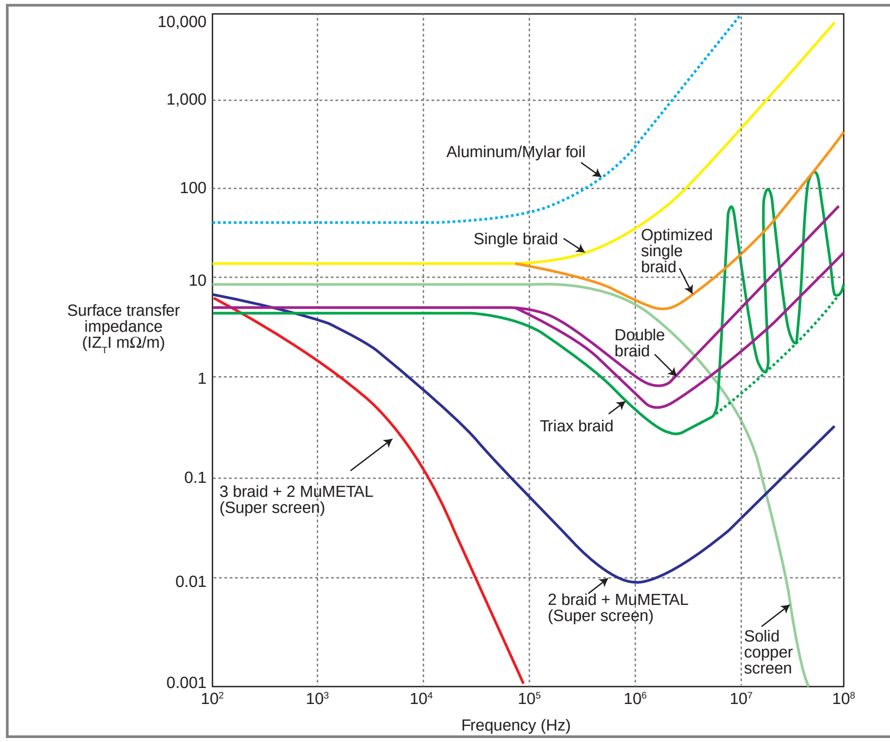


Figure 2—Transfer impedance of various cables versus frequency

ESTABLISHING SE

Manufacturers of shielded wires and cables rarely provide SE data, as it is dependent on installation. However, their cables usually comply with some industry standard, so their parameters will be essentially the same as for all cables in their category. Figure 1 is typical of what can be expected.

To establish the SE of a cable by measurement is a challenge requiring a well-equipped RF lab with an anechoic chamber. Worse, the test results of the same sample performed in a lab may vary significantly due to the test configuration. In real-life applications, the experienced SE may, once again, be different from the test, as it is affected by installation.

Here, transfer impedance Z_T comes to the rescue. The concept was introduced in the mid 1930s by S. A. Schelkunoff. It is a parameter that is intrinsic to the cable shield and nothing else. For the past several decades, transfer impedance Z_T has been the method preferred by electromagnetic compatibility (EMC) engineers to address EMC issues.

Z_T enables you to calculate the voltage developed on the shielded conductor as a result of the current flowing on the outer surface of the shield. This is due to current coupling through the shield thickness plus leakage inductance through the braid holes. When a solid metal conduit is used, with increasing frequency the diffusion quickly becomes immeasurable due to skin effect. Figure 2 shows Z_T for different types of shields, including a conduit, called solid copper screen. You can see that Z_T decreases—therefore the SE increases—with the increasing quality of the braid.

Z_T parameter of a specific shielded cable or wire is also difficult to obtain from the manufacturer, but once again, standardization comes to the rescue. Also, measuring transfer impedance Z_T of a given shielded cable is easier than measuring SE (see Figure 3). A current is injected into the shield by a current transformer, which is a standard piece of equipment for any company that performs conducted susceptibility and emission tests. The current will induce a small voltage between the shield and the center conductor. The voltage is measured and the transfer impedance, normalized to 1-m length, can be calculated:

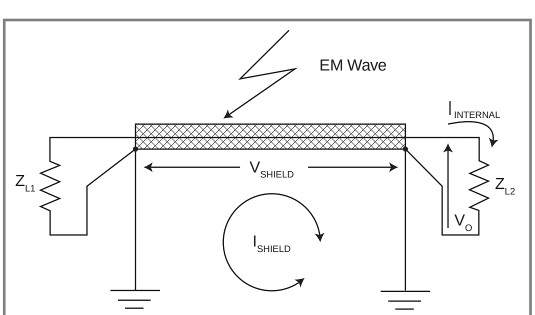


Figure 3—Transfer impedance calculation

$$Z_L = \frac{V_O}{I_{SHIELD} \times l} \left(\frac{\Omega}{m}\right)$$

V_O is the voltage induced between the inside conductor and the shield over length l. l is the length of the test cable in meters. I_{SHIELD} is the current injected into the shield.

If the cable is terminated at both ends with loads Z_{L1} and Z_{L2}, both equal to the cable characteristic impedance, each end will show V_O, which is one half of the fully induced voltage. Therefore:

$$Z_T = \frac{2 \times V_O}{I_{SHIELD} \times l} \left(\frac{\Omega}{m}\right)$$

The manner in which the shield is electrically terminated also affects the final value of Z_T. The shield termination methods range from the excellent 360° termination to very poor pigtails.

This impedance, usually determined empirically, needs to be added to Z_T. Figure 2 shows typical values of some commonly used cables.

Z_T below certain frequency—100 kHz for most cables—remains constant, as it is merely the shield's ohmic resistance, which can be easily measured. This changes at higher frequencies due to the leakage inductance L_T between the braid and the inner conductor. The inductance is usually determined empirically as well, based on the kind of braid and its weave. A good single-layer braid typically has a leakage inductance L_T of about 1 nH/m. As a result, above some frequency, 100 kHz typically, the transfer impedance becomes a complex number:

$$Z_T = R_{SHIELD} + j\omega L_T$$

What does all this mean in practical terms? Here is an example: Referring to Figure 3, shield current is determined based on the installation and the illuminating e-field. Below 400 MHz, however, the hardware designer doesn't have to worry about the field strength. The maximum shield current to be encountered (300 mA is considered to be the highest for rugged equipment), is provided in the specification by the system EMC engineers who select proper values based on standards, the installation, the harness length, and the intended operating environment. For instance, the specification states that a specific 4-m cable with a shield current of 30 mA from 0.5 MHz to 400 MHz must cause no upset to an embedded controller.

Let's say a single-braid cable, such as RG58, is used and you want to know the interfering voltage reaching the equipment at 15 MHz. Figure 2 indicates that the cable's Z_T at 15 MHz is about 0.15 Ω/m. With 30 mA flowing through its 4-m shield, the voltage induced on the center coax wire V_O will be:

$$V_O = Z_T \times 1 \times I = 0.15 \times 4 \times 30 \times 10^{-3} = 18 \text{ mV}$$

If the cable is terminated at both ends by 50-Ω resistors, load (the controller input) Z_L will see:

$$V_L = \frac{V_O \times 50\ \Omega}{(50\ \Omega + 50\ \Omega)} = \frac{18 \text{ mV}}{2} = 9 \text{ mV}$$

of the interfering 15-MHz signal. If the load impedance Z_{L2} is much higher than Z_{L1}—a more common situation, say 5-kΩ—the voltage V_L seen by the controller will be:

$$V_L = \frac{V_O \times 5\ k\Omega}{(5\ k\Omega + 50\ \Omega)} \approx 18 \text{ mV}$$

This is the level of interference your circuits must tolerate without losing data. You may ask why we need to terminate the shield at both ends? If

NEED-TO-KNOW INFO

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

—

Killing the EMI Demon
by Norman Rogers
***Circuit Cellar* 146, 2002**

Emissions regulations can make it expensive to get your microprocessor boards to the market. In an effort to help us reduce EMI testing costs, Norman reveals the EMI reduction tricks Rabbit Semiconductor developed for its second-generation processor. Topics: EMI, FCC, Rabbit 3000, Frequency, Harmonic, Pulses, Clock, Power Supply, Bus, Spectrum Spreader, OFDM

—

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

we terminate it at only one end, there should be no shield current flowing and thus no voltage developed as the result. This reasoning is true for low frequencies. But, when the cable length exceeds one tenth of the wavelength of the interfering field frequency, or the shield impedance is high (cheap cable), the voltage will build up along the cable length and capacitively couple to the inner conductor. Therefore, long cables should have shielding grounded at both ends at least, better in regular intervals along their length. It is not uncommon to see double-braided cables with one shield grounded at both ends and the other at only one end.

EMC: MYTH VERSUS REALITY

EMC is often viewed as some kind of black magic. But, like everything else, it is subject to the strict laws of physics. The practical problem is that it is next to impossible to define all parameters affecting EMC issues. Therefore, our calculations have to rely on approximation and experience. For design of EMC protection, referring to Figures 1 and 2 will be quite sufficient to get you started. With a healthy safety margin, your design will be successful with the first test. ■

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.*

RESOURCES

O. Hartal, *Electromagnetic Compatibility by Design*, R&B Enterprises, 1993.

M. Mardiguian, "Differential Transfer Impedance of Twisted Pairs,"*Interference Technology*, 2010, www.interferencetechnology.com/uploads/media/TDG10_Mardiguian_01.pdf.

QUESTIONS & ANSWERS

Embedded Design Theory & Practice

An Interview with Shlomo Engelberg

Shlomo Engelberg has been a member of the faculty of Jerusalem College of Technology's department of electronics since 1997. His research interests include the theory of ordinary and partial differential equations, control theory, signal processing, instrumentation, and measurement. Shlomo has written five articles for Circuit Cellar *and is the editor-in-chief of the* IEEE Instrumentation and Measurement Magazine. *In September 2011, I spoke with Shlomo about his background in mathematics, engineering, and teaching. We also discussed a few of his passions—most notably, math, teaching and explaining things, programming microcontrollers, and Analog Devices's microcontrollers and digital signal processors.*
—Nan Price, Associate Editor

NAN: Where do you live and work? What city, country?

SHLOMO: I live in Bet Shemesh, Israel, and my primary job is in Jerusalem.

NAN: Do you spend part of the year in the United States?

SHLOMO: I usually visit the United States a couple of times a year. During the summer, my family and I generally spend about a month visiting my parents in Far Rockaway, a neighborhood in New York City.

NAN: You have a PhD in Mathematics. That means you spent a lot of time hitting the math books. So, when did you first become interested in electrical engineering and how did you learn so much about the field while studying mathematics?

SHLOMO: Actually, I started off in electrical engineering. I earned a Bachelor of Engineering and a Master of Engineering from the Cooper Union (in Manhattan, NYC). Toward the end of my time as an undergrad, I realized that I wanted to study more mathematics. I went to NYU's Courant Institute for my graduate work in mathematics. During the 10 years I was studying in college and graduate school, the schools I studied at were within a few blocks of one another.

NAN: How long have you been designing microcontroller-based systems? What was your first design?

SHLOMO: I have been working with digital signal processors (DSPs) and microcontrollers since the mid-1990s. The first time I worked on a microcontroller project I was part of a team developing a tool that would measure the constituent parts of fresh milk.

> "When I was department chair, I always had a 'scope near at hand. I love math, but sometimes math is not enough, and I just have to see something actually work. I have a bit of equipment at home, and from time-to-time I try to get my 14-year-old interested in working with microcontrollers."

NAN: You've been involved at the Jerusalem College of Technology department of electronics for a number of years—first as a lecturer and most recently as an associate professor. You also run a lab at Bar Ilan University. What courses do you currently teach and where do you teach them?

SHLOMO: I enjoy teaching a wide variety of courses. I am currently teaching an electronics course for computer engineering students at the Jerusalem College of Technology. Next semester I will be teaching a linear systems course at the Jerusalem College of Technology and a microprocessor lab at Bar Ilan University. In the spring, I will be teaching an introduction to control theory at Bar Ilan University. (I'll be on sabbatical from the Jerusalem College of Technology in the spring; otherwise, I would be teaching more.)

NAN: What are your main areas of interest and research at this time?

SHLOMO: My research interests include the theory of ordinary and partial differential equations, control theory, signal processing, instrumentation, and measurement.

I am currently working on coding theory, and I spend a fair amount of time working on instrumentation and measurement issues. (I am the editor-in-chief of the *IEEE Instrumentation and Measurement Magazine*, and that takes up quite a bit of my time.) I am also working on a set of simple experiments/demonstrations to accompany the electronics course I am teaching. I enjoy trying to present things in a way that helps people understand what I am trying to get across, and I hope that these demonstrations help me achieve that goal.

"When I look at the projects in my book, *The ADuC841 Microcontroller Design Manual: From Microcontroller Theory to Design Projects*, I see the history of the subjects I was working on during the years the book was written. I had a good time working on all of the projects and I hope that my readers enjoy them too."

NAN: In addition to teaching, you've also done a fair amount of writing. In fact, *Circuit Cellar* has published five of your articles and will soon be publishing a book you authored. In 2003, we ran your article "Spread Spectrum: Theory and Practice" (*Circuit Cellar* 156), which was designed to help readers comprehend the science behind spread-spectrum techniques. Tell us how you came to write about that subject.

SHLOMO: At the time I wrote the article, I was working on the mathematical theory of spread spectrum. (I published an article about spread spectrum in an academic journal at about the same time as the article in *Circuit Cellar*.) Though I love mathematics, I also like to see how things work. Getting a simple version of spread spectrum to work in a DSP-based system led to my *Circuit Cellar* article which was written with Elihoshua Shekalim.

NAN: Some of the projects in your articles have been influenced by a member of the Israeli police force. In "Digital Watermarking" (*Circuit Cellar* 183, 2005), you discuss a method for marking files for authentication and security. In "Voice Changer Technology" (*Circuit Cellar* 187, 2006), you describe a system that can detect when someone is using a voice changer. Tell us about those two designs and how they came to fruition. Are they still in use?

SHLOMO: Over the years, I have done a fair amount of work with members of the Israeli police force. (Several of the Jerusalem College of Technology's graduates are police officers who do engineering work for the police department.) Though I do not believe that either of the projects you mention are still in use, I am currently working with a police officer on ideas that will hopefully enable the police to verify that recordings were made when they were claimed to have been made.

NAN: Tell us about the microprocessor laboratory at the Jerusalem College of Technology and Bar Ilan University.

SHLOMO: I used to run the microprocessor lab at the Jerusalem College of Technology. Now I am loosely associated with the lab, but I do not run it. At Bar Ilan University, I am in charge of a lab course that is required of all students in the school of engineering. On average, about 100 students pass through the lab each year. Teaching that course and running the lab have taught me quite a bit about managing a large lab course. The first thing I learned is that it is important to have good staff members. Thankfully, the lab has always been blessed with many excellent staff members.

NAN: Analog Devices's microcontrollers figure prominently in several of the projects you described in your *Circuit Cellar* articles. For example, the ADuC812 was featured in "Digital Watermarking." It was also highlighted in "Pseudo-Random Noise Theory and Applications" (*Circuit Cellar* 171, 2004), which focused on building an ADuC812-based noise-generation system that produces noise on demand. "Signal Processing with the ADuC812" (*Circuit Cellar* 174, 2005) details an alternative to digital processor chips, which can be difficult to interpret and low on peripherals. Why the these microcontrollers?

SHLOMO: The reasons for using the ADuC812 and the ADuC841 are connected to my love of the 8051 and my desire to have an onboard analog-to-digital converter (ADC) and digital-to-analog converter (DAC) on the microcontroller I use in the lab. I have used some of Analog Devices's DSPs too, and I generally enjoyed the experience.

NAN: Your forthcoming *Circuit Cellar* book, *The ADuC841 Microcontroller Design Manual: From Microcontroller Theory to Design Projects*, is based on a lab you teach. What compelled you to write the book? What can readers expect to learn?

SHLOMO: I love to explain things, and in this book I do my best to explain 8051-based microcontrollers to my students and readers. The book has a set of labs designed to familiarize the reader with the 8051 and several experiments that introduce the reader to using the onboard ADCs and DACs. It also has a wide assortment of more advanced projects.

When I look at those projects, I see the history of the subjects I was working on during the years the book was written. I had a good time working on all of the projects and I hope that my readers enjoy them too.

NAN: Are there any other "go-to" microcontrollers that you like?

SHLOMO: Not really. I am hoping to start a project with a new (and much more powerful) microcontroller soon, but I haven't started the project yet.

NAN: Are you a do-it-yourselfer—meaning, do you have a workspace at home where you work on projects in your free time? Or do you do most of your engineering at the university?

SHLOMO: I do most of my engineering work in my office at the Jerusalem College of Technology. Even when I was department chair, I always had a 'scope near at hand. I love math, but sometimes math is not enough, and I just have to see something actually work. I have a bit of equipment at home, and from time-to-time I try to get my fourteen-year-old interested in working with microcontrollers.

NAN: What embedded-design-related projects are you currently working on or planning? What can you tell us about them?

SHLOMO: I have an industrial job that may be starting soon. Other than that, I generally make changes to my microcontroller labs each year. As I try to never to give students a lab that has not been thoroughly tested, making the changes takes a fair amount of time. Luckily, I really enjoy programming microcontrollers.

NAN: As a professor, you have an interesting insight into what excites future engineers. What are the hot topics currently captivating your students? Robotics? FPGAs? Digital signal processing? Rapid prototyping (e.g., Arduino)?

SHLOMO: I have a hard time generalizing here. I know what I am interested in and I try to interest my students in areas that I think are interesting and important.

Right now, I have students working on digital signal processing topics. One of the students is working on implementing a pretty new and "hot" technique known as compressive sensing using an ADuC841. That is a nice combination of a mathematically sophisticated topic, some practical microcontroller work, and some MATLAB programming.

NAN: Last question. Let's say you had as much time as you wanted to work on any embedded-design project you wanted. Call it your "dream project." What would it be?

SHLOMO: I have a dream project, but it will probably seem a bit odd to most people. I seem to be surrounded by heaters and air conditioners that claim to maintain a space at some desired temperature. I ought to be able to set the temperature as I please and walk away for six months and return to a pleasantly cool home. Actually, I would never leave my air conditioner unsupervised for days or weeks.

My dream project is to put together a heater/air conditioner, a network of sensors, and, possibly, fans that would all be controlled by a microcontroller and that would actually keep a room at 20°C if that's what was requested of it. In this fantasy, I would be able to set the temperature once and be certain that, winter or summer, any time I walked into my house the temperature would be just what I asked for. Someday, I hope to actually have the time to design and build such a system.

[Book Excerpt]

The ADuC841 Microcontroller Design Manual
From Microcontroller Theory to Design Projects
by Shlomo Engelberg

The main prerequisite for reading this book is a basic knowledge of programming. The labs in this book will, hopefully, help the reader learn how to program in assembly language. They are not designed to teach programming from the ground up. Some of the later labs require some knowledge of electrical engineering. At Bar Ilan University, the lab is given to students in the first semester of their third year. A couple of years in an electrical engineering program should be sufficient to allow a student to understand the more electronics-oriented of the later labs.

Students entering the microprocessor lab often have no experience doing very low-level coding, and many of the students have grown up using desktop PCs running various operating systems. People who are accustomed to using such computers are apt to think of computers as capricious and hard to understand. This book's goal is to demonstrate that if you consider a simple microcontroller, it is possible to see exactly why the microcontroller does what it does. In the space of one semester, of about 10 labs, it is possible to become sufficiently familiar with the Analog Devices ADuC841 to feel that you truly understand it and many of its capabilities.

This book is slated for publication by* Circuit Cellar *in late 2011, early 2012.

by George Martin (USA)

Design Development (Part 4)

Processors, Power, and Interfacing

The first three parts of this series covered the processes for designing and viewing schematics and using unified modeling language (UML) to design a new product. Now it's time for a detailed look at processors, power, and interfacing to the transducers.

In this series of articles, I have described the process of designing a product from scratch. Part 1 set up the required documents and a list of requirements. Part 2 refined the requirements and presented a first pass at the hardware design. Part 3 presented a unified modeling language (UML) diagram for the software. In each part, I described what we knew about the project and what could be worked on, leaving areas in question for a later date. This is an example of the "*mañana* design approach" at work. I read about this a long time ago and can find no reference to it now. I find it a useful approach when there are a lot of unanswered questions. Now, I'm not claiming credit for the *mañana* design approach, so if you like it, it's yours.

I left the CPU, battery, memory card, and power supply for a later date because not enough was known about these topics to do a design. Well, the future is now. It's time to make some decisions.

CPU

Our system is a battery-powered data logger. From our other requirements we find that the CPU needs to have two I²C ports, one for the sensors and one for the SD memory card. Also, modern CPUs have USB and Ethernet built in. Without a firm requirement in the specifications or any input from the customer, I would go with Ethernet as an interface. Connecting to the world via the Internet would be a more universal approach. So, with little or no input, let's just go with an Ethernet interface. Speed is sufficient and we can get power over Ethernet. It might be too much to ask that the batteries run the unit while that unit is communicating over Ethernet. Low-cost CPUs are available with Ethernet interfaces built in. Be careful when you select a CPU, as some have little of the hardware built in while others have everything except the connector and magnetics.

This leaves several hundred—if not several thousand—devices to choose from. The power the CPU consumes is the next area I would explore. If we select a 3.3-V device, the power will be less than a 5-V device. I believe we can get a 16- or 32-bit device with RAM and flash memory built in. The operating speed is probably not important for data logging but may show up when we're running the Ethernet interface. I would look for the ability to program in C and a great integrated development environment (IDE). When we analyze power consumption we need to look at two distinct areas. The first is sleeping. The CPU is in a Low-power mode and waiting for an event to wake it up and start running. The second is power consumed when running and how fast the CPU operates. The sleeping power is straightforward and probably a number right out of a datasheet. The second is more complicated. The calculation is to take the power when running multiplied by the time the CPU is running. So a faster, more power-consuming CPU may have a lower operating power if it gets the work done faster.

Let's say that wake up, gather data, save to memory card, and go to sleep are our normal operations and we do that every 6 s. This is the same analysis we did for sizing the memory in my previous article. If the CPU ran for 1 s out of the 6 s then we could take:

$$\frac{(\text{operating power } + \text{ sleeping power } \times 5)}{6}$$

and get the average power. And, as we operate faster, the contribution of the operating power becomes less of the total power. I expect the operation will consist of waking up the sensors, taking readings, and writing the data to the memory card. We could probably double buffer the data so that waking up the sensors and reading the sensors would run concurrently with writing the previous readings to the memory card.

And, if this CPU took 1 s out of 6 s to do the work described above, we would have:

$$\frac{(80 \text{ mA} + 5 \times 0.55 \text{ mA})}{6} = \frac{(80 \text{ mA} + 2.75 \text{ mA})}{6}$$
$$= \frac{82.75 \text{ mA}}{6} = 13.79 \text{ mA}$$

At 3.3 V, that's 45.5 mW and 24 hours of operation is 1092.17 mW (i.e., 24 × 45.5) hours per day. If the CPU did its work in 0.5 s, we would have:

$$\frac{(80 \text{ mA} + 11 \times 0.55 \text{ mA})}{12} = \frac{(80 \text{ mA} + 6.05 \text{ mA})}{12}$$
$$= \frac{86.05 \text{ mA}}{12} = 7.17 \text{ m}$$

At 3.3 V, that's 23.7 mW and 24 hours of operation is 567.93 mW (i.e., 24 × 23.7) hours per day. So, faster CPUs can sometimes consume less power overall.

I had the Texas Instruments Stellaris datasheet handy, so I started there (see Figure 1). Let's look at Freescale Semiconductor's K60 Kinetis ARM Cortex-M4 microcontroller. This version has all the features we will need. It claims less than 500 nA in Stopped mode and 200 µA/MHz in Run mode. Say we run at 20 MHz. That's 4 mA at, say, 3.3 V for 13.2 mW for full-power operation—not much different than the Texas Instruments microcontroller. I would also recommend looking at the new Renesas offerings.

Running From Flash:
- Current Consumption 80 mA
- VDD = 3.3 V
- Code = `while(1){}` executed in flash
- Peripherals = All ON
- System Clock = 50 MHz (with PLL)
- Temp = 25°C

Sleep Mode
- Current Consumption 8 mA
- VDD = 3.3 V
- Peripherals = All clock gated
- System Clock = 50 MHz (with PLL)
- Temp = 25°C

Deep Sleep Mode
- Current Consumption 550 µA
- Peripherals = All OFF
- System Clock = IOSC30KHZ/64
- Temp = 25°C

Figure 1—Data from Texas Instruments's Stellaris datasheet

BATTERY

Let's jump to the battery to see what this calculated power consumption means. I'd like to select a battery that can be removed and connected to a separate charging device. That way we are not trying to build battery chargers. They have been known to catch fire and also cause batteries to fail (go boom). There's no need for us to take on those problems. If I do a quick search for batteries on Digi-Key's website, I see the available technologies are alkaline, lithium, nickel cadmium, nickel-metal hydride, zinc, and lead acid. One could make a career—perhaps even several careers—out of battery design and selection. Let's just assume that alkaline is not rechargeable and lead acid may tip over and spill. This would just eliminate these from consideration. I know that both types of batteries have their applications.

When looking for a battery, you need to consider capacity in ampere hours (Ah), voltage (V), discharge rates in amps (A), and shelf life usually in years. Some batteries have a high capacity, large discharge rates, and a short shelf life. These would be good for power tools that are recharged every evening. At the other end of the spectrum are batteries with a low discharge rate and long shelf life (seven years). These are good to back up the clock in your PC.

For our application, I would look for a battery with a capacity of 5,000 mAh (10 days of operation as the calculations currently stand), discharge rates from 0.08 to 8.00 A, voltages of 4 to 6 V, and a shelf life or self-discharge rate of at least one month.

This search came up with a Panasonic nickel-metal hydride battery pack. It isn't hazardous and doesn't require special disposal, except in California. I would, however, responsibly recycle any used up batteries. The specific rating of this battery pack is 3.6 V with 3,500-mAh capacity. This battery is made up from individual cells. If we ever need more (or less) capacity or voltage, it's straightforward to select a different battery pack using the same internal technology.

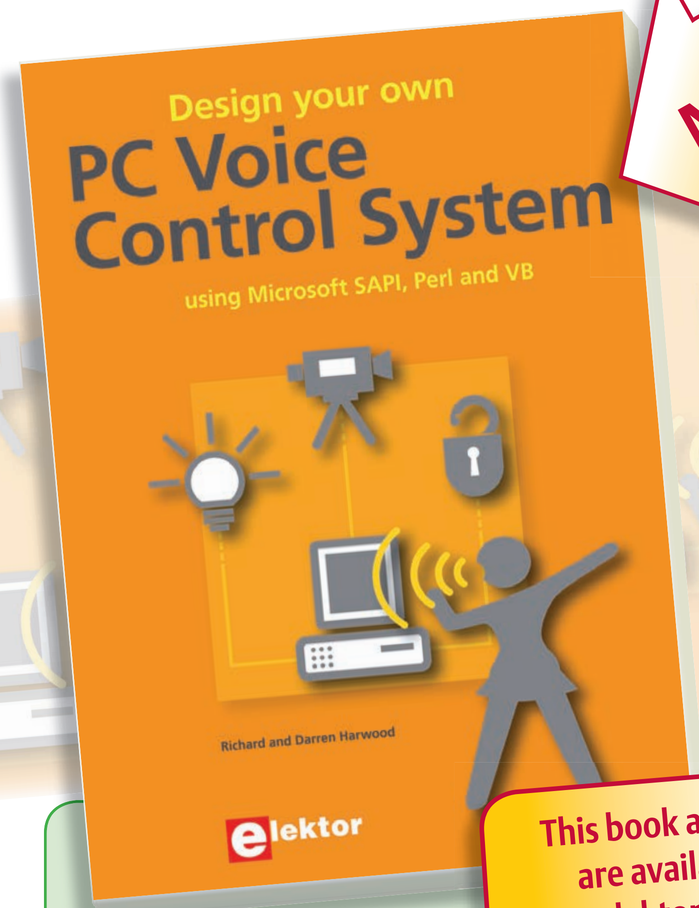
I believe the size of this battery pack is 2.4″ × 4″ × 1.3.″ So, when we size the case, we will need to have room for the battery.

MEMORY CARD

A lot has been written about SD memory cards and several articles have appeared in *Circuit Cellar*. I went to the website and did a search on "SD Memory cards." There are enough results in that search to fill up several days of reading. Basically, the SD card looks like a SPI device at power up. We have reserved one complete SPI interface in the microcontroller for the interface to an SD card. It would be great if we had a file system that could be called up rather that designed from scratch. So, go back to CPU selection and add a requirement of SD interface code. As I was looking through the *Circuit Cellar* discussion groups, I came across a reference to Brush Electronics. They have code for an SD card file system. Their code is for the PIC CPU. If

Elektor Shop

The world of electronics at your fingertips!



NEW!

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

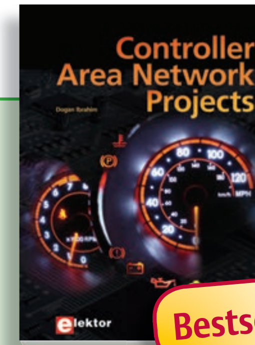
Books

User: "Computer, turn on hall lights" – Computer: "Hall lights on"

Design your own PC Voice Control System

Don't you feel it's just about time we could simply talk with computers and get them to handle some of the everyday house automation tasks - lights, doors, gates, curtains, televisions, music and other domestic appliances? Or even make the information super-highway slicker by talking with you and telling you about the information you requested to be retrieved from the web? Think of all the time and inconvenience you could save - no more need to install yourself at a desk and fumble about with keyboard and mouse. Well now you can! This book guides you through practical speech recognition, speech annunciation and control of really useful peripherals. If you are interested in the practical technology of interfacing with machines using voice, such as talking with your own PC, then this book is your guide.

Approx. 216 pages • ISBN 978-1-907920-07-3 • $47.60



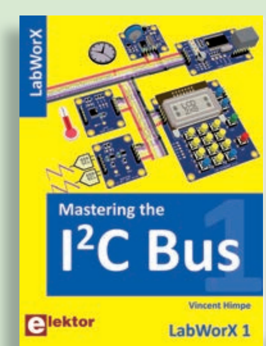
Bestseller!

Free mikroC compiler CD-ROM included

Controller Area Network Projects

The aim of the book is to teach you the basic principles of CAN networks and in addition the development of microcontroller based projects using the CAN bus. You will learn how to design microcontroller based CAN bus nodes, build a CAN bus, develop high-level programs, and then exchange data in real-time over the bus. You will also learn how to build microcontroller hardware and interface it to LEDs, LCDs, and A/D converters. This book comes complete with a free CD-ROM.

260 pages • ISBN 978-1-907920-04-2 • $47.60



LabWorX: Straight from the Lab to your Brain

Mastering the I²C Bus

Mastering the I²C Bus is the first book in the LabWorX collection. It takes you on an exploratory journey of the I²C Bus and its applications. Besides the Bus protocol plenty of attention is given to the practical applications and designing a solid system. The most common I²C compatible chip classes are covered in detail. Two experimentation boards are available that allow for rapid prototype development. These are completed by a USB to I²C probe and a software framework to control I²C devices from your computer.

248 pages • ISBN 978-0-905705-98-9 • $47.60

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

CD/DVD-ROMs

Elektor's Components Database

CD ECD 6

This CD-ROM gives you easy access to design data for over 70,000 components! The program package consists of eight databanks covering ICs, transistors, diodes and optocouplers. A further eleven applications cover the calculation of, for example, zener diode series resistors, voltage regulators, voltage dividers and AMV's. A colour band decoder is included for determining resistor and inductor values. All databank applications are fully interactive, allowing the user to add, edit and complete component data. This CD-ROM is a must-have for all Electronics enthusiasts!

ISBN 978-90-5381-258-7 • $40.20

Circuits, ideas, tips and tricks from Elektor

CD 1001 Circuits

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

ISBN 978-1-907920-06-6 • $55.70

Elektor OSPV

The Elektor Wheelie, which we launched a while ago, has been adopted by schools and private individuals who want to gain a thorough understanding of the technology behind self-balancing vehicles. This new Elektor OSPV is based on the same concept, but with the difference that it's for indoors, it's easy to steer, it's light and foldable and...it's open source. You can configure or modify it to suit your wishes. This self-balancing vehicle is perfect for factory halls, warehouses, hospitals and schools. Check www.elektor.com/ospv for further information!

Kit of parts

Art.# 110320-91 • $1570.00

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

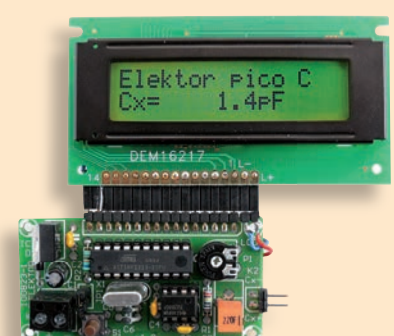
Elektor US
PO Box 180
Vernon, CT 06066
USA
Phone: 860-875-2199
Fax: 860-871-0411
E-mail: order@elektor.com

USB Long-Term Weather Logger

This stand-alone data logger displays pressure, temperature and humidity readings generated by I²C bus sensors on an LCD panel, and can run for six to eight weeks on three AA batteries. The stored readings can be read out over USB and plotted on a PC using gnuplot. Digital sensor modules keep the hardware simple and no calibration is required.

Kit of parts incl. PCB, controller, humidity sensor and air pressure sensor modules

Art.# 100888-73 • US $50.20



Pico C Meter

Even upmarket digital multimeters boasting a built-in capacitance meter are useless if you want to check out tiny capacitances like 2.7 pF or 5.6 pF. Usually, you're tied to a lowest measuring range of 2000 pF, which is a good laugh to RF designers and radio amateurs. Although at 3.5 digits the DMM's resolution is 1 pF, any measurement below 200 pF or so will produce coarse if not ridiculous results. Pico C does a far better job. Beating many DMMs hands down, this little instrument easily measures capacitances down to fractions of a picofarad.

Kit of parts incl. Elektor Project Case, programmed microcontroller, LCD and PCB

Art.# 100823-71 • $118.40

Kits & Modules

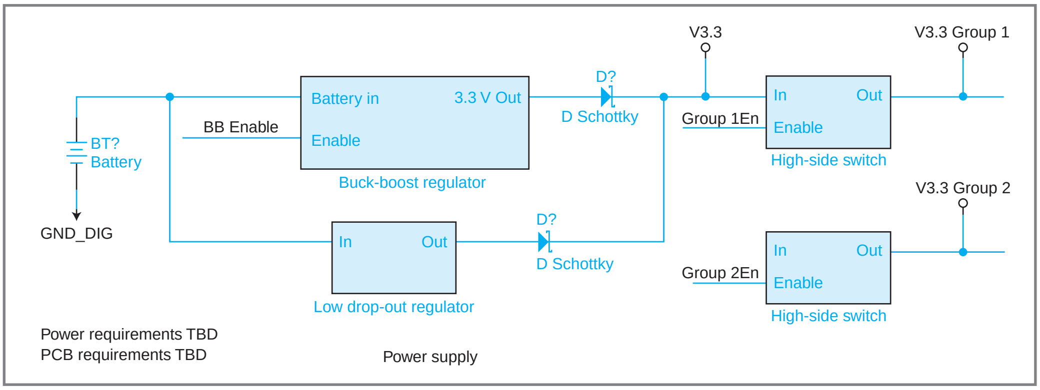


Figure 2—Power supply schematic

you're going to use a PIC, then you're all set. If not, then you'll need to keep looking, write one yourself, or purchase one and modify it. This last process might not be too difficult since it's a simple SPI interface. Perhaps just the low-level code (code that touched the hardware) needs to be rewritten.

We have to select connectors, decide if the user will have access to the card, and determine what size cards to support. But for now, let's stop here. I'm sure more questions will come up as we move through the design.

POWER SUPPLY

We need a power-supply design that will take the 3.6-V nominal battery output and make 3.3 V for the circuitry. That's easy. Just put in a low drop-out (LDO) regulator and we're all done. Hold your horses. It's never that simple. If we look at the discharge curve for the battery we see 3.6 V for 80% of its life and less than that for the other 20%. I think we need to extract that remaining 20%. One choice would be to use a four-cell pack. Digi-Key supplies a four-cell pack that gives 4.8 V at 6,500 mAh: Panasonic's nickel metal hydride batteries. So, that's the simple solution. It costs more money and is a larger size. The boss won't be happy; neither will the customer.

Another solution is to lower our operating voltage. This is probably a good approach. I'm finding many of the modern chips will work from 3.3 V down to 2.2, 2, or even 1.8 V. So, if this is your choice, you need to take a look at all the datasheets and see if the components will run with that low of a power-supply voltage.

Another choice is a buck-boost regulator. When the battery voltage is higher than 3.3 V, the regulator reduces that voltage to 3.3 V (buck). When the battery voltage is less than 3.3 V, the regulator increases the battery voltage (boost). Texas Instruments's UCC2942 buck-boost controller is one such device. This IC takes battery voltages as low as 1.8 V and as high as 8 V and will produce a 3.3-V regulated output. The advantage is that we could use the smaller/lower-cost three-cell battery. The disadvantage is that this device is complex to use and generates switching noise. That switching noise needs to be eliminated for the analog measurement portion of our design.

This certainly looks like chaos. (Refer to my article, "Design Development (Part 4): Software Design," *Circuit Cellar* 254, 2011.) We clearly have to balance each decision off all the datasheets of all the integrated circuits in our design. The power supply/battery combination needs to meet the requirements of the most restrictive device in our design. It may pay to reevaluate our initial parts selections now that we are more educated about our design.

THE POWER SUPPLY DESIGN

Let's try a first pass at the power supply design and see what happens. Let's assume we use the battery pack that is 3.6 V nominal. That's 4.2 V (i.e., 3 × 1.4) maximum and 3.3 V (i.e., 3 × 1.1) minimum if you look at the discharge curve.

Other buck-boost controllers to consider include Analog Devices, which has the ADP2504/2504, and Linear Technology, which has the LTC3440. Both of these devices can handle our input voltage range and can produce a 3.3-V output. Their prices are less than $3 at the 1,000-piece level. They both require an external inductor and several other components. Both have efficiencies in the 90% range. I would be cautious about basing too many decisions on this efficiency number at this point in the design process. And, both manufacturers offer evaluation boards for the devices. Using the evaluation boards you can be up and running with little effort on your part.

The CPU (yet to be selected) will have the capability to go into a low-power mode. What we need is for it to be able to turn off power to the other devices in the system. Shutting them down would save considerable power. The cost will be in the switching circuitry that actually shuts them down and the power lost during the wake-up time for these devices. If we shut down the accelerometer, we won't get any readings. So, that device may need to be continuously powered.

Looking for a high-side switch, I did another search on Digi-Key's website. I found several devices. Among them, ON Semiconductor's NCP380 single-channel current-limiting power distribution load switch. This device has several uses (current limiting) but it has an enable input. When that input is low, the device output is disabled. This input is meant to be driven from a TTL or CMOS signal. So, we could have one of these switches for each device or group of devices we want to control.

Now we have to look at the datasheets for the devices and see what happens when we shut down the power. For example, if we shut down each device one by one will the serial bus we are using for communications still work? Details like these will keep the unit from working the way we want it to.

Let's consider one more optimization we could make.

When we are powered down, the current draw will be very little, so little that we could shut down the buck-boost regulator and save more power. We could do this with a comparator and feed back to the buck-boost controller. But, there may be an even easier approach. If the CPU goes into Low-power mode and then shuts down the buck-boost regulator, we would save the power required to run that regulator. But, eventually, the power on the CPU and any capacitors attached would be used up. How about connecting a diode directly from the battery to the CPU? This would work to supply that low-power mode. Well, that's acceptable, but when the battery pack is fully charged, I think the voltage would be too high. So, how about a LDO regulator with a blocking diode on its output that would only supply power in the low-power operating mode?

I've implemented this approach in the power supply schematic (see Figure 2). All the Enable signals come from the CPU. With all Enables off, the LDO regulator is supplying power to the CPU and other sensors that are powered. It's not very efficient, but it should be a low power draw. The Schottky diodes are added to isolate the two main regulators. These introduce additional voltage drop (loss) in the power supply. Linear Technology has an application note that describes how to turn a MOSFET into a low-loss diode. It also offers LTC4411, which is a 2.6-A, low-loss, ideal diode in ThinSOT that uses that technique.

I have not shown where I would place the capacitors in this design. And I have not shown any filtering on the power supply. I think we will need both. We have devices that are converting the physical parameters into digital signals. And you may think we don't have to worry about noise in the power supply. Well, I bet we do. I would consider putting the filtering into the power supply. I believe the filtering we need is to eliminate the switching frequency that the buck-boost regulator has introduced.

THE SCIENCE OF DESIGN

Is product design a science or an art form? I think it's a science. It should be a science. But as you see from what we've just been through, there are a lot of subjective decisions you need to make as you design a product. In a former life, while working with Steve Ciarcia at an aerospace company, we watched as the system engineers would do the system design. That would result in a document describing all the functional components of a product with specific performance numbers attached when required. That was—and is—clearly a scientific process. In the commercial field, we need to move more rapidly with not all requirements and design questions answered, and that is considerably more art than science. In the end, a good product must work and meet the requirements of the market. I hope you enjoyed the past few articles. I'll return to more specific topics, but I'll use what we've just been over as I design new systems. In fact, I'm starting two new designs this week. Let me know how you approach system design and I'll share it with our readers. Next time, I'll present an article on A/D converters, switching power, supply noise, and filters. ■

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

RESOURCES

Brush Electronics, www.brushelectronics.com.

Panasonic Industrial Co., "Nickel Metal Hydride Batteries: Individual Data Sheet," 2005, http://panasonic.com/industrial/includes/pdf/Panasonic_NiMH_HHR650D.pdf.

SOURCES

ADP2504/2504 Buck-boost DC-to-DC converter
Analog Devices, Inc. | www.analog.com

K60 Kinetis ARM Cortex-M4 microcontroller
Freescale Semiconductor, Inc. | www.freescale.com

LTC3440 Buck-boost DC/DC converter and LTC4411 2.6-A low-loss ideal diode in ThinSOT
Linear Technology Corp. | www.linear.com

NCP380 Power distribution switch
ON Semiconductor | www.onsemi.com

Nickel metal hydride battery pack
Panasonic Industrial Co. | www.panasonic.com

Stellaris MCU Family and UCC2942 buck-boost controller
Texas Instruments, Inc. | www.ti.com

by Richard Wotiz (USA)

Ionization Detectors

It's time for a look inside a curious type of sensor that's used to bridge the gap between electricity and particle physics. This article covers a family of devices that measures ionizing radiation.

Welcome back to Embedded Unveiled! This month we'll journey inside an unusual type of sensor that's used to bridge the gap between electricity and particle physics. There are many types of detectors that use the principle of ionization to measure a variety of physical or chemical phenomena. I'll take a look at one family of devices that measures ionizing radiation. Then I'll peek inside an ionization smoke alarm to see how it works in the presence of varying amounts of smoke. Don't worry, no buildings were harmed in the creation of this column!

First, here's a quick lesson on the physics behind an ionization detector. There are many methods of producing ions, both naturally occurring (such as lightning), and man-made (such as a particle accelerator). A positive ion is created when an atom or molecule loses an electron, giving it a net positive charge. The negatively charged free electron can then combine with another atom or molecule, creating a negative ion. Since opposite charges attract, positive ions and either negative ions or free electrons in close proximity will be drawn together and recombine, neutralizing each other. The ionization detector works by creating an electric field between a pair of electrodes that attracts these charged particles before they recombine, causing a current to flow.

CATCHING RAYS

Three related types of ionization detectors are used for measuring radioactivity. They all use a sealed metal container with a pair of electrodes. Often the container itself acts as one of the electrodes. There is also a window that's transparent to the particular type of radiation

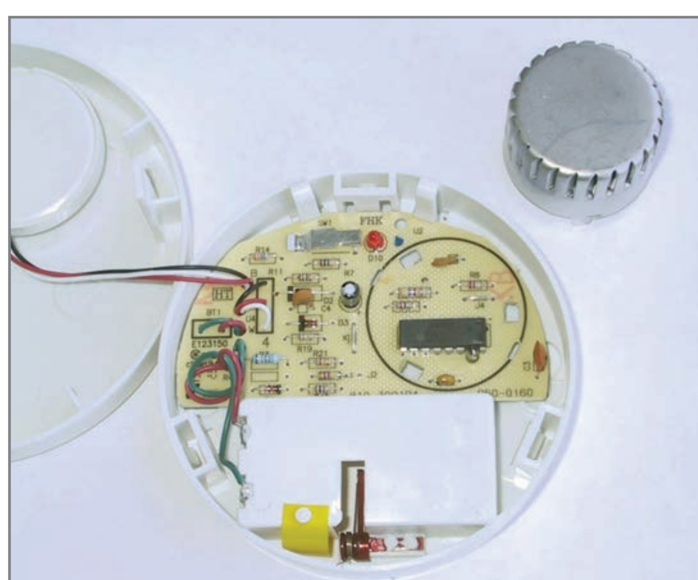


Photo 1—The inside of a smoke detector with the ionization chamber removed. You can see the test button at the top of the printed circuit board (PCB) and the alarm sounder mounted to the top cover at the far left.

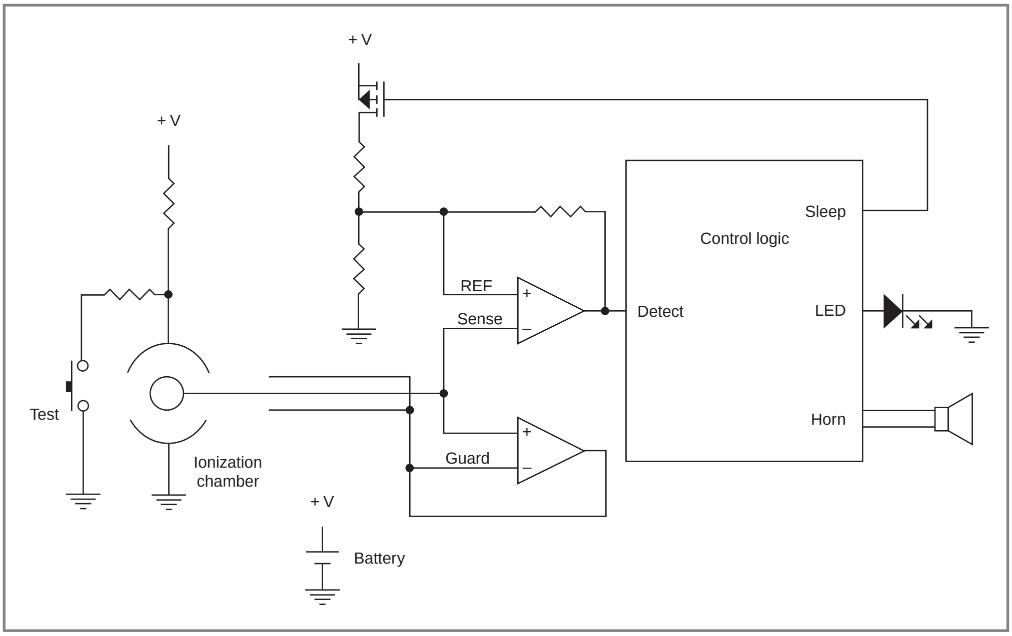


Figure 1—The key components of the smoke detector. The sense input circuitry and control logic are contained within a single integrated circuit.

being measured. When a radioactive particle enters the container, it may collide with an atom of the gas that is inside. If the particle has enough energy, it will knock off an electron, ionizing the atom.

The key difference between the detector types is the range of electrode voltages. The simplest, but also the least sensitive, is the ionization chamber. It can operate on a voltage as low as a few volts. When ions are created in the chamber, the electric field between the electrodes will draw positive ions to the negative electrode and negative ions to the positive electrode. This movement, called drift, is proportional to the strength of the electric field. It is a slow phenomenon, typically less than 1 m/s. The drift causes a current to flow that's proportionate to the number of ions reaching the electrodes. The current can be very small, typically in the picoampere range, so you need a sensitive amplifier with extremely high input impedance. This is not difficult with modern FETs, but in earlier days, you'd need an expensive electrometer to get a meaningful signal.

The second type of detector is known as a proportional counter. The physical construction is similar to an ionization chamber, but it uses a higher voltage. This causes an avalanche effect, where a free electron is accelerated enough by the electric field to knock off more electrons from additional atoms. The avalanche eventually dies out on its own. This results in an amplification effect, which produces an output signal that's proportional to both the electrode voltage and the energy of the radioactive particle that started the process.

The last type, which is the most sensitive, is the Geiger-Müller tube. Like the proportional counter, it depends on the avalanche effect. But the voltage is set high enough so that the entire gas volume in the container will ionize with each incoming particle. Therefore, the output signal does not depend on the energy of the incoming particle; it can only be used to count particles. This process results in the familiar clicking sound produced by a Geiger counter that has an audio output.

Ionization chambers are commonly used in smoke alarms that you can buy for your home. But instead of measuring external radiation, the chamber contains a weak radioactive source to create its own stream of ions. It's also open to the outside air, so smoke can enter and affect the movement of the ions. This changes the current flow through the chamber's electrodes, which sounds the alarm.

SMOKE ALARMS

The first battery-powered smoke alarms appeared on the market in 1969, but they didn't come into widespread use in the U.S. until the following decade.[1] That's when the national building codes first included a requirement for smoke alarms in new homes, and Underwriters Laboratories first developed standards for them. As a result, manufacturing of smoke alarms jumped from an estimated 50,000 units in 1971 to more than one million by 1974, increasing to more than 10 million annually by the mid-1980s.[2]

Photo 1 shows the inside of a typical ionization smoke alarm. I was quite surprised when I first opened one up; I was expecting to see a bit more complexity. It turns out almost all of the circuitry is contained in a single IC, in this case a Microchip Technology RE46C152 low-power CMOS ionization-type smoke detector IC. Figure 1 shows a simplified schematic of the unit. The ionization chamber has

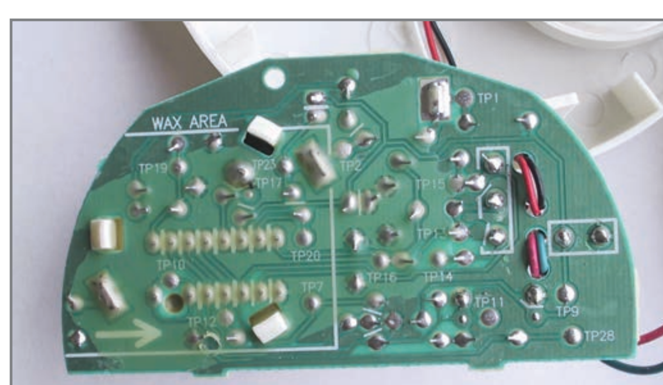


Photo 2—The bottom of the PCB. Part of the board is covered in beeswax to protect the high impedance paths from contamination that could lead to excessive leakage current.

three electrodes, which enables the sense signal to be measured independently from the chamber supply current. The chamber's output impedance is high enough that even the slightest leakage current will shift its voltage level. To combat this, the adjacent pins on both sides of the IC's sense input are guard pins, which are outputs driven to the same level as the input. This minimizes leakage across the IC package. The input pin itself isn't connected to the printed circuit board (PCB); rather it is bent upward and soldered directly to the chamber's sense pin. If you look closely near the bottom right corner of the IC in Photo 1 you can see a solder blob where I unsoldered the pin from the chamber. The rest of the circuitry isn't quite as high impedance as the sense input, but it can still be affected by humidity or dust settling on the PCB. Photo 2 shows the bottom of the board covered in beeswax, which is much less expensive than a conformal coating, but still does the job.

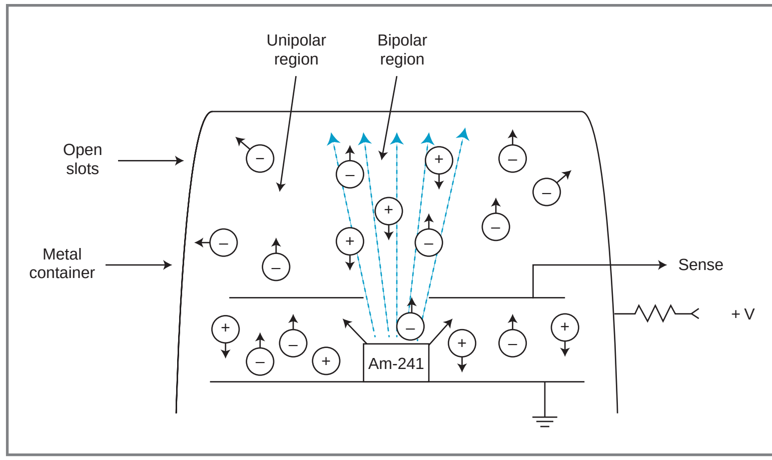


Figure 2—The ionization chamber contains a small radiation source to ionize the air inside. The ions are drawn to the opposite polarity electrode, causing a small current to flow.

IONIZATION CHAMBER

Figure 2 is a diagram of the ionization chamber. The radioactive source is specified on a label on the smoke alarm as 0.9 microcuries (µCi) of Americium-241 (Am-241), which emits a constant stream of alpha particles. The particles have an energy of 5.486×10^6 electron volts (eV), which will travel up to 4 cm in air. Alpha particles are the largest and slowest type of radioactive particle and can be stopped by a sheet of paper. But, their large size also gives them enough mass to impart significant energy to whatever they may bump into, which makes them extremely dangerous to come in contact with. I was careful to take proper safety precautions when removing the chamber from the PCB for the photo. I then replaced it right away without disassembling it any further.

The maximum current that can flow through the chamber is given by:[3]

$$I_S = e\left(\frac{E_m}{E_i}\right) F\,Q\,S, \text{ where:}$$

I_S = chamber saturation current (amps)

e = charge of an electron = 1.6×10^{-19} C

E_m = ionizing particle energy = 5.486×10^6 eV

E_i = ion pair formation energy $\approx$ 14 eV for air (nitrogen and oxygen)

F = efficiency factor of radioactive source (assume 1.0)

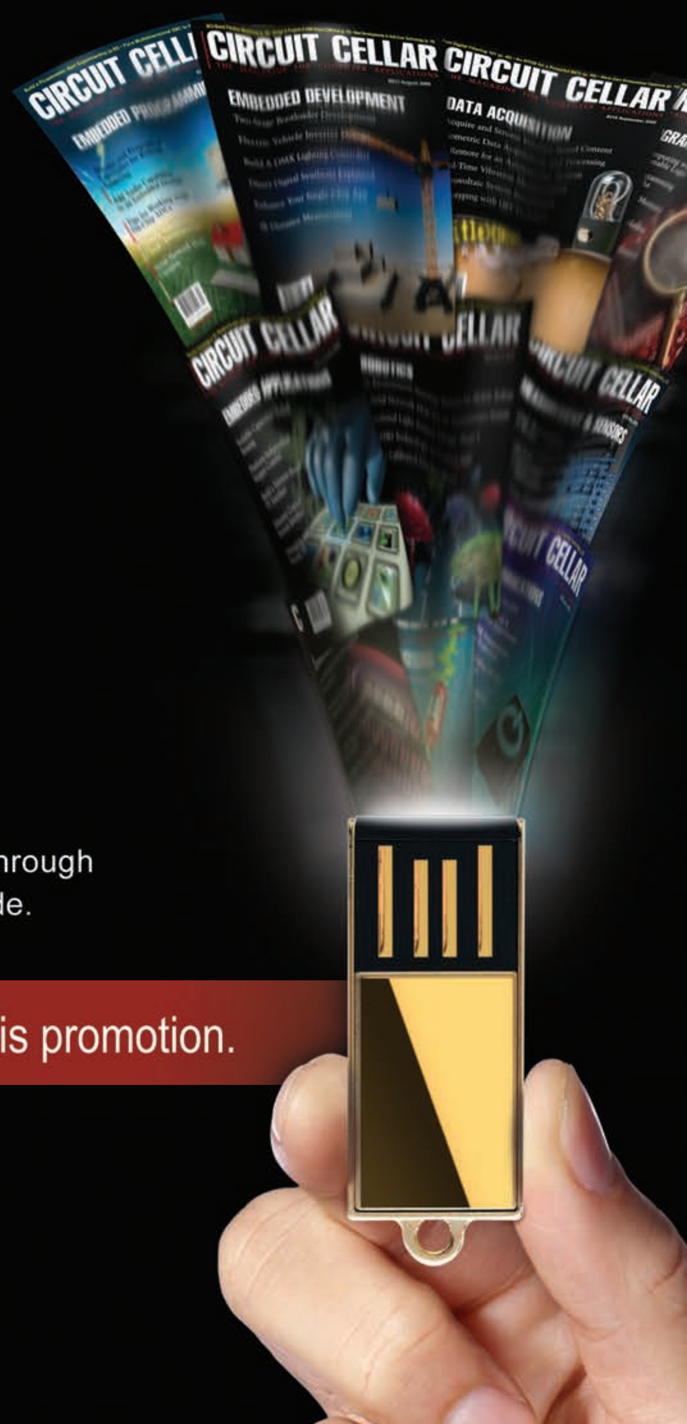
$$Q = \text{ionizing particles per second per curie} = \frac{3.7 \times 10^{10}}{\frac{s}{Ci}}$$

S = source activity = 0.9×10^{-6} Ci

This gives a saturation current of 2.1 nA. The chamber is supplied from the 9-V battery through a 1-M resistor, so the chamber is being operated at saturation. I used an efficiency factor (F) of 1.0, which assumes that there isn't any filtering of the alpha particles. It's possible that the source includes a thin layer of gold foil to reduce the particle energy, which would reduce F, but I don't have access to the proper equipment to find out. In any case, that would only reduce the saturation current, which wouldn't change how the chamber operates.

There are two basic configurations for an ionization chamber. In a bipolar chamber all of the air is ionized, so both positive and negative ions are present throughout its volume. A unipolar chamber is designed so the radiation source only ionizes part of the air. The rest of the chamber will only contain ions of a single polarity that have been pulled there by the electric field. A space-charge region, acting as a boundary layer, will form between the two areas. This will enable ions to travel to the unipolar area at a relatively constant rate if the chamber is being operated at saturation. The term "unipolar" may seem a bit misleading since the chamber does contain both positive and negative ions, but it refers to the fact that part of the chamber farthest from the radiation source only contains ions with one polarity. In both chamber types, ions will be attracted to uncharged smoke particles if they are present and will attach themselves to the smoke. The charged smoke particles move much more slowly than the ions, which will cause a reduction in current flow in the chamber. The unipolar chamber is more sensitive to smoke density, and less affected by environmental changes and air movement.[4]

The geometry of the chamber I examined appears to be designed to have a unipolar region, since the sense electrode partially shields the alpha source so it's not exposed to the entire chamber volume. In operation, no appreciable current flows to the sense input, so the sense electrode will take on an intermediate voltage between the other two electrode voltages. When smoke enters the chamber, it will have a much greater effect on the unipolar region, since its supply of ions is limited by the ion flow across the boundary layer. This will cause a larger voltage drop between the sense electrode and the positively charged metal container. The sense voltage will decrease by an amount that's approximately proportional to the smoke concentration times the average particle size, eventually triggering the alarm.



CIRCUIT CELLAR
CIRCUIT CELLAR
EMBEDDED DEVELOPMENT
CIRCUIT CELLAR
DATA ACQUISITION

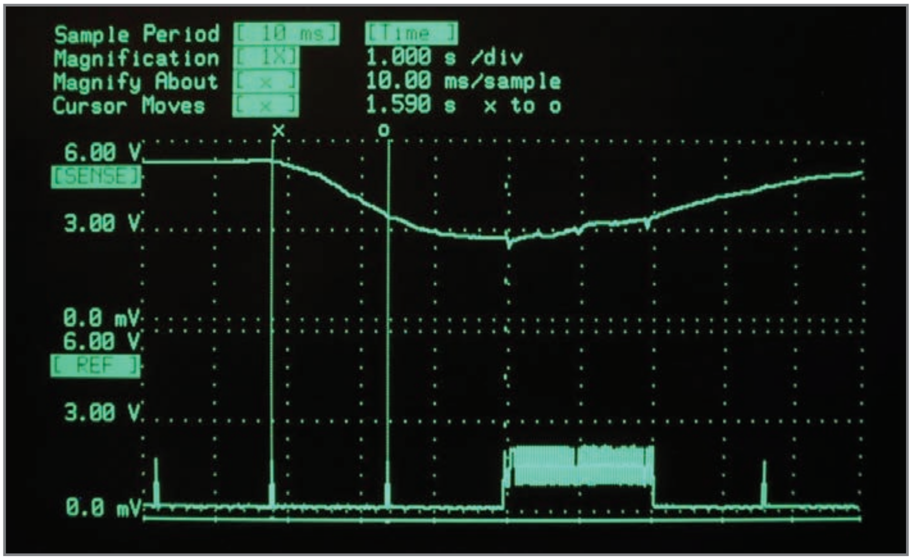


Photo 3—The system's response to a small puff of smoke. The scope's sampling rate is too slow for the REF amplitude to be accurate, but you can see the periodic sampling every 1.59 s until SENSE drops low enough to trigger the alarm.

TESTING

At this point, I wanted to try out the smoke alarm for myself. I opened one up that still had its ionization chamber intact, disconnected the piezo beeper, and connected some wires to the Guard and REF pins. That way I could look at the detection comparator signals while it was operating. Since the Guard pin is driven to the same level as the Sense input, it enables me to monitor the chamber's voltage without putting any additional loading on its high-impedance output. Photo 3 shows the system responding to a small puff of smoke. The comparator status is checked about every 1.5 s until smoke is detected. Then it's repeatedly checked until the smoke clears. While the alarm is sounding, the LED blinks once per second. You can see small blips on the Sense signal as the battery voltage dips slightly with each blink. You'll also see a corresponding gap in the REF signal, since the comparator isn't checked during the blink to avoid a false reading.

Photo 4 is a closeup of the alarm trigger point. You can see the REF signal more clearly as it gets enabled by the Sleep signal to take a measurement about every 40 ms. The smoke alarm also has a hush feature that temporarily silences a false alarm. It works by reducing the REF signal by about a volt, which makes the alarm less sensitive without entirely disabling it.

I wanted to see how sensitive the ionization chamber was to changing environmental conditions, since it can be affected by temperature, humidity, and atmospheric pressure. After leaving the unit in the refrigerator overnight, I found the Sense signal had dropped slightly to 5.07 V from its nominal room-temperature value of 5.30 V. Next, I subjected the alarm to the high-humidity environment of a hot shower, but I didn't see any measurable change in the output. I didn't have any practical way to test it at varying pressures—I could only imagine trying to explain to airport security why I was carrying a disassembled smoke alarm and some test equipment onto an airplane!

Residential smoke alarms like the one I tested are calibrated at the factory and are designed to stay within specifications for their 10-year operating lifetime. Their performance is specified by the UL217 standard (see Resources). I noticed that the units I examined each had a different resistor value installed at a particular location on the PCB that affected the REF voltage level. That's to compensate for unit-to-unit variations in the ionization chamber's output. Smoke alarms designed for non-residential buildings are subject to more stringent accuracy and testing specifications, both at manufacture and during their lifetime, and are evaluated using the UL268 standard. They're required to have regular testing to confirm that their sensitivity is within range, rather than just a simple Go/No-go functional test.[5] To get this level of accuracy and stability, the alarms have two identical ionization chambers. One is exposed to smoke and the other is protected so smoke can't enter. The alarm is triggered when the voltage difference between the two chambers exceeds a set threshold. Any environmental or lifetime-related drift

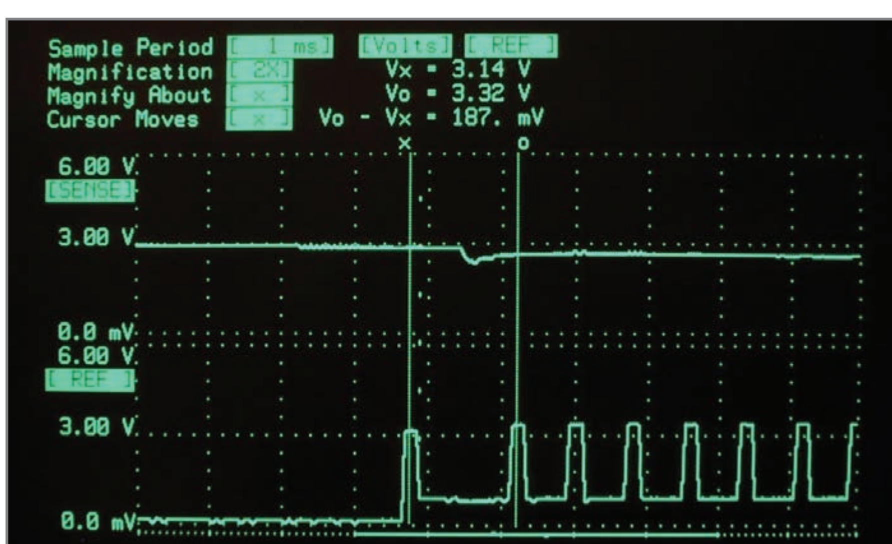


Photo 4—A closeup of the alarm trigger point. Note the 187-mV hysteresis on the REF signal under the cursors. The horizontal scale is 50 ms per division.

will affect both chambers equally and will only have a minimal effect on the alarm's sensitivity.

I hope you've enjoyed this journey into the world of smoke and physics. I certainly have a new appreciation for all the research that's gone into fire detection in the past 50 years, as well as what really happens when I hear that unmistakable beeping sound.

Richard Wotiz has been taking products apart ever since he was old enough to pick up a soldering iron. He's been helping others put them back together since 1991, when he started his design consulting business. Richard specializes in hardware and software for consumer products and children's toys. He can be reached at rw601@spiraltap.com.

REFERENCES

[1] R. Bukowski et al., "Performance of Home Smoke Alarms: Analysis of the Response of Several Available Technologies in Residential Fire Settings," *NIST Technical Note 1455-1*, National Institute of Standards and Technology, 2008.

[2] R. and G. Mulholland, "Smoke Detector Design and Smoke Properties," *NBS Technical Note 973*, National Bureau of Standards, 1978.

[3] C. Litton, "Optimizing Ionization-Type Smoke Detectors," *Fire Technology*, Vol. 15 No. 1, 1979.

[4] A. Scheidweiler, "Physical Aspects of Ionization Chamber Measurement Techniques (Unipolar and Bipolar Chambers)," Fire Detection for Life Safety, Proceedings of a Symposium, National Research Council (U.S.) Committee on Fire Research, 1975.

[5] National Fire Protection Association, "NFPA 72: National Fire Alarm and Signaling Code," 2010.

RESOURCES

R&E International, a Subsidiary of Microchip Technology, Inc., "RE46C152: CMOS Ionization Smoke Detector ASIC with Interconnect, Timer Mode, and Tone Select" Product Specification, 2009, ww1.microchip.com/downloads/en/DeviceDoc/22175A.pdf.

Underwriters Laboratories, "UL217: Single and Multiple Station Smoke Alarms," 2006.

———, "UL268: Smoke Detectors for Fire Alarm Systems," 2009.

SOURCE

RE46C152 CMOS Smoke detector IC
Microchip Technology, Inc. | www.microchip.com

NEED-TO-KNOW INFO

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

—

A Look at "Battery-in-a-Chip" Technology
by Tom Cantrell
***Circuit Cellar* 228, 2009**
Got Energy? Energy harvesting is all the rage, but like a real harvest you need a place to store the crop. This month, Tom introduces the next advance in thin-film rechargeable lithium battery technology: a battery-in-a-chip. Topics: Battery, Enerchip, Li-on, CBC050, Charge, Supercap, Capacity, CBC-EVAL-08, Ez430-RF2500, Energy Harvesting

—

Self-Powered Solar Data Logger
by Abigail Krich
***Circuit Cellar* 198, 2007**
Abigail designed a microcontroller-based, self-powered solar data logger that uses a photodiode to measure solar insolation levels. The system converts the analog signal to a digital value that's stored in flash memory. Topics: Solar, Data Logger, Insolation, Photovoltaic, ATmega32, STK500, Photodiode, ADC, LCD

—

iEthernet Bootcamp
Get Started with the W5100
by Fred Eady
***Circuit Cellar* 208, 2007**
Are you ready to join the Ethernet revolution? If so, it's time to start working with WIZnet's W5100 hardwired TCP/IP embedded Ethernet controller. In this article, Fred helps you get started on your first W5100-based design. Topics: W5100, Ethernet, TCP/IP, PPPoE, Development Board, PIC18LF8722, PICC-18 C Compiler, MPLAB

—

Multifunctional Wireless Alarm
by Carl Smith
***Circuit Cellar* 194, 2006**
Carl's multifunctional wireless alarm system can monitor everything from the doors to the sump pump in your house. The system features an MC13192 SARD board and several wireless sensors. When the alarm is activated, simply place a call to your house to obtain a status report. Topics: Alarm, Wireless, MC13192 SARD, MC9S08GT, Garage Door, Water Level, Phone, Temperature, DTMF, Emic, CH1837A, SSI204, MPX2010GS, Accelerometer, MMA1260D

—

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

by Jeff Bachiochi (USA)

Fly-By-Wire Wheelchair (Part 1)

Beyond Normal Joystick Control

Joystick-controlled motorized wheelchairs are useful to those with mobility issues. But what if a user doesn't have the dexterity required to control a joystick? You can add the circuitry in this design to a commonly used non-bus-type motorized wheelchair to override joystick control.

Many aging Americans use motorized wheelchairs to help them stay active in their communities. Joint replacement is becoming commonplace. It is no longer futuristic to expect that we may one day order up a replacement part from the local organ farm. I'm talking about growing organs not harvesting used ones. In this scenario, death might only come by accident rather than from a worn out or diseased body part. While this probably won't happen within my lifetime, I can remember when going to the moon seemed impossible.

Those members of society who were born mobility challenged or have become so as the result of an accident can benefit from the use of motorized wheelchairs. According to a study initiated by the Christopher & Dana Reeve Foundation, there are nearly one in 50 people living with paralysis of some kind—approximately six million people.[1] While those with the incapacity to walk or stand unassisted can gain a level of freedom from the use of motorized wheelchairs, there is a subgroup that cannot use the standard user interface: the joystick.

Most motorized wheelchairs require arm/hand dexterity to move a joystick as the only input to motor control. Joystick operation provides both a gas pedal and steering wheel in a single control. The XY joystick is auto centered meaning, if untouched, it returns to a centered Off position. A slight nudge forward, backward, left, or right applies a small amount of power to each wheel-driving motor. Corresponding larger movements (from center) increase this power, just like

Photo 1—Jet 3 Ultra motorized power wheelchairs, a popular alternative to the manual wheelchair, give users with limited capabilities the ability to address the world with minimal assistance. (Photo courtesy of Pride Mobility Products Corp.)

stepping on the accelerator. Directional movement determines the polarity of the voltage to each motor as well as a power relationship between the motors permitting directional control, like a steering wheel.

Users who are unable to move the joystick may as well have an unpowered chair. There are manufacturers that specialize in alternative input devices; however, these are not necessarily wheelchair manufacturers. As third-party enablers, they must align themselves with chair manufacturers that have designed their systems with proprietary architectures, much like a car manufacturer having its own proprietary bus. (Refer to my series "Vehicle Diagnostics," *Circuit Cellar* 251–253, 2011.) This month's project uses a popular non-bus-type motorized wheelchair and show how you can add some circuitry to override the normal joystick control.

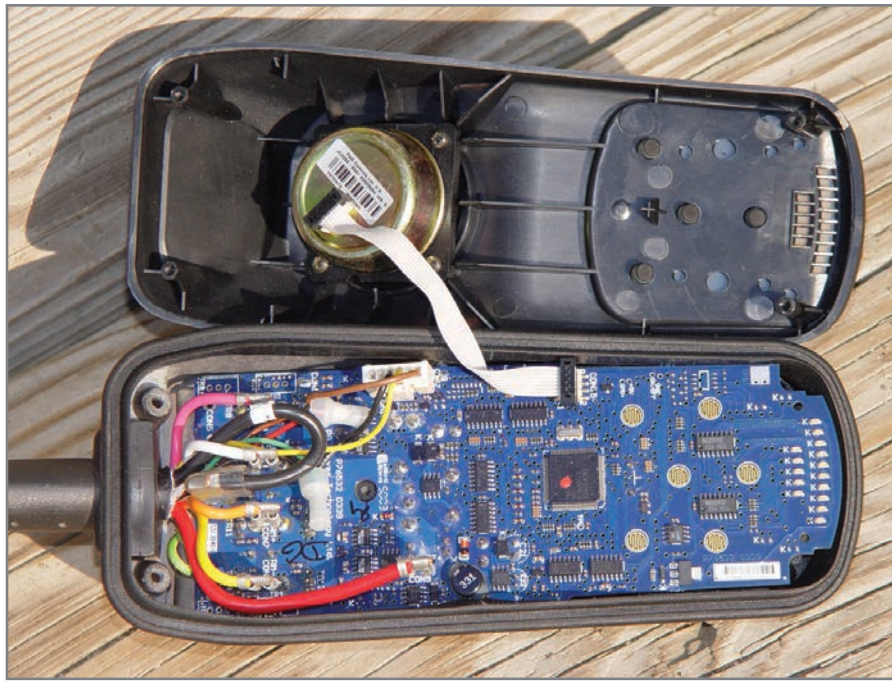


Photo 2—The VR1 controller is encased on a single PCB in the armrest of the power chair. Visual cues come from the surface-mount LEDs. Gold-plated push button contacts provide targets for several buttons on the rubber keypad. A 2 × 4 shrouded connector connects to the only external device, a commercial joystick.

COMMERCIAL JOYSTICK

The rugged frame of Pride Mobility's Jet 3 Ultra Motorized Power Wheelchair houses DC motors and batteries and offers a strong and stable mount for the padded driver's seat (see Photo 1). It uses a PG Drives Technology VR1 controller mounted in the chair's armrest. All drive and control circuitry is contained on a single printed circuit board (PCB). As you can see in Photo 2, the push button (contacts) and LED status are integrated on the PCB. The remaining I/O is a PG Drives JC2000 series off-the-shelf joystick that plugs onto the PCB.

I can see why this product is chosen for tough environment applications where precision fingertip control, safety, and long trouble-free life are primary requirements. The rubber-booted joystick handle is self-centering and requires a minimum amount of force to bring it to its maximum limit of ±20° (vertically and horizontally). The use of Hall effect sensors provides a non-contact design that yields extreme life expectancy. Dual outputs for each axis provide redundant information that can be used to determine failure issues. A center tap reference output offers a way to compare for a centered stick output.

While this joystick is available in XYZ format, the VR controller uses an XY implementation. The joystick requires a regulated 5-V supply. Idle (centered) output is 0.5 of V_{CC}. A center-referenced voltage, as well as dual-voltage outputs for each axis, are available at the (1 mm on center) shrouded eight-pin connection header.

The manufacturer repeatedly notes the need for safety precautions (those of the system architecture) to prevent unwanted movement. Of prime importance is a way to disable movement if necessary. Beyond the emergency situation, the system should carefully monitor all inputs for discrepancies. If the redundant axis outputs of the joystick don't remain within 5.6% of each other, joystick action should be disabled. This may also include a power-up to check that the joystick is not producing anything other than a centered (and idle) output.

FORTUNE TELLING

This project reroutes the joystick outputs so they can be disconnected from the chair's controller. Alternate voltage outputs from some external source can be substituted for the on-board joystick outputs. A switch input will force the outputs to a centered position. This "emergency" input will have a unique side effect; see if you can imagine what this would be.

My first thoughts were to add some relays that would switch between the main joystick outputs and some other alternate. Not only might this put an excessive draw on the 5-V power available, but the break-before-make (BBM) action of the contacts may add a glitch to the VR1 processor that may be seen as a error or malfunction. So, this design developed into an analog-to-digital (ADC) front end and a digital-to-analog (DAC) rear end. Initial operation is just to measure the joystick outputs and verify the values. Once the joystick is monitored correctly, the second phase would be to reproduce the joystick voltages using the DAC outputs for the VR1. Without putting unnecessary limits on what may be used as an external device to take command of the chair, I decided to use the basic serial port available on the microcontroller to provide the means of taking

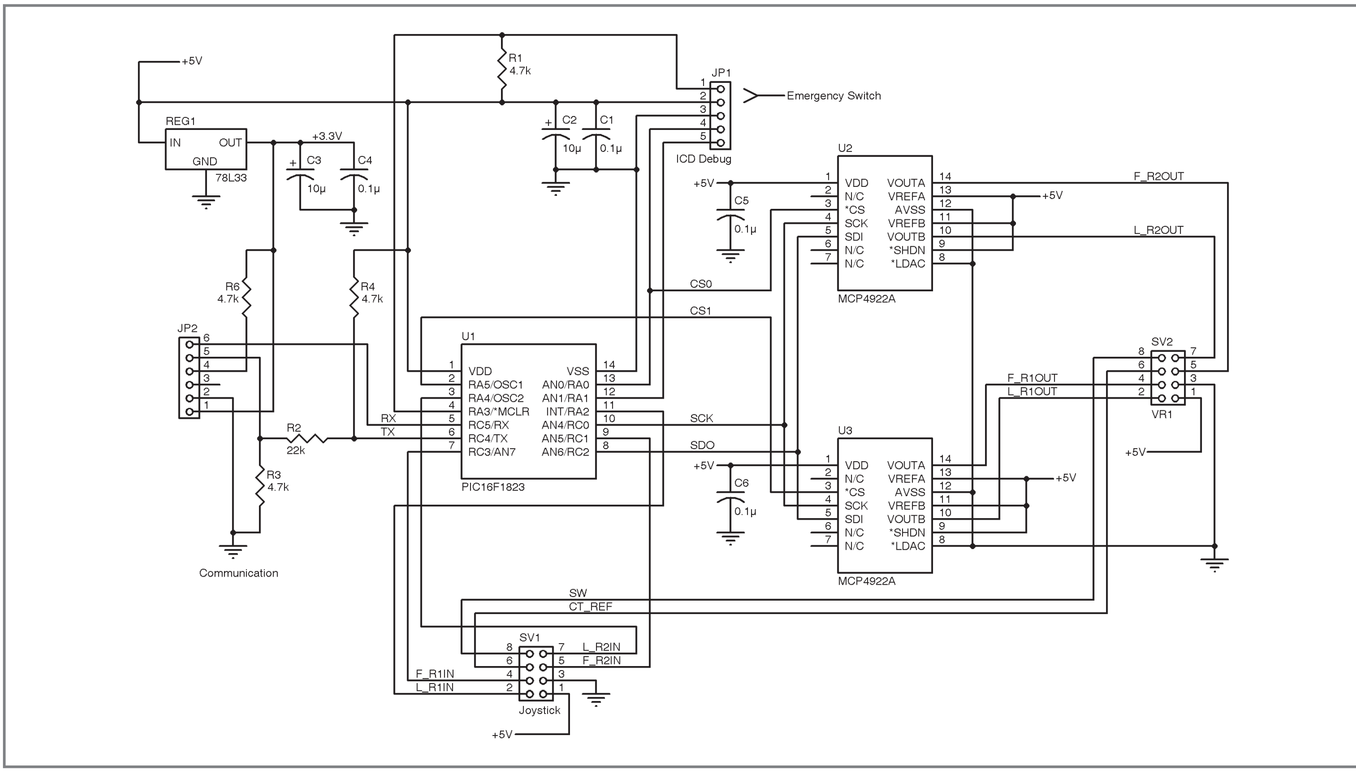


Figure 1—I used a small microcontroller that provides four A/D inputs (to convert the joystick's output voltages to a digital values) and twin D/As (to convert digital values back into four voltage outputs) as well as a serial communications interface. A switch contact input can provide D/A data forcing their voltage outputs to 0.5 of V_{CC} (centered joystick).

over control (more on this later).

The Hall-effect sensors are the largest contributor to noise on the outputs of the joystick. If we assume a full output swing of 5 V, the Hall-effect noise is ±20 mV or 0.04 V/5 V (0.8%). An 8-bit ADC has a resolution of about 0.019 V (over a 5-V range) so it would be plenty adequate. The microcontroller I used for this project has a 10-bit ADC, which is fine. In fact, I can probably toss away some resolution. We'll see. On the output side, I will be using external DACs as they are available with output amplifiers that can supply some output drive. The interface to the DACs is SPI, as shown in Figure 1. The Microchip Technology MCP4922 DACs I selected have a reference input that can be scaled by 2×. This could be used to keep the DAC referenced to that of the joystick actual reference value, but I didn't use this as all the circuitry would be running from the same supply (reference).

When you think of a joystick's travel, you probably think a circular maximum path. The axis outputs follow a sine/cosine relationship. However, the direction and distance of travel may be confined by some mechanical limits to the joystick's movement. There are other potential patterns that may be used in order to modify the axis outputs in relationship to one another. By creating a shaped hole or gate in a plate that the joystick sticks up through you can limit the joystick's travel in many ways. The circular pattern (just mentioned) is the most widely used. A square- or diamond-shaped hole yields square and triangle relationships that are 90° out of phase. While these patterns limit only the maximum excursions, they can be used in situations that require a non-linear relationship. Other shapes may include a vertical or horizontal slot to limit the joystick to a single axis. Or X or "+" slots where the axis are required to move together or alone.

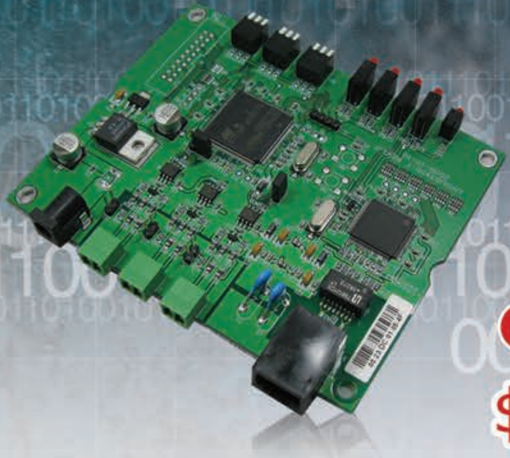
As previously mentioned, the idle position is centered and produces equal voltages of 0.5 of V_{CC} on each axis. As the joystick is moved further away from idle, toward its mechanical stop, it's like stepping on the gas. This is indicated by the amount.

As discussed earlier, with a noise factor of 40 mV, there isn't much sense in using any resolution less than 40 mV when the noise alone can vary this much. While the resolution of the microcontroller's ADC is 10 bits, you can just use the upper 8 bits of the conversion and still have a resolution that is less than half that of the noise (19 mV). The 8-bit conversion values for each of the joystick's voltage outputs can be transmitted out of the serial port as four two-digit hexadecimal values (X1X2Y1Y2) followed by a fifth pair and a <CR> (carriage return). Using ASCII characters instead of binary data enables the transmission string to be viewed. The fifth pair constitutes a control value. If nothing else, it can indicate the status of the "emergency switch." This makes each transmission string 11 bytes long with a fixed format.

Before the microcontroller can redirect control from the onboard joystick to some external device, let's set up some basic criteria. If the "emergency input" is closed, then

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

ICbank
Gateway to Electronics

www.icbank.com

the microcontroller will send idle voltage values to the DACs (as input to the VR1). If the on-board joystick is moved off center, the microcontroller will send its output voltages to the DACs (as input to the VR1). External data must be received at a sustained periodic rate for at least 1 s for it to be sent to the DACs (as input to the VR1). Deltas in DAC data can be limited (ramped up or down) so as not to exceed a safe acceleration.

COMMAND STREAM

We begin by monitoring the onboard joystick's output via the serial port. With a transmission string of 11 characters multiplied by 10 strings per second that's 110 characters or 1,100 bps, for a minimum transmission rate of about 1,200 bps. (I chose 38,400, for no particular reason.)

I consciously left off any drivers on the schematic serial interface, which allows for more possibilities. One of the unique features of this UART is the ability to configure the TTL serial interface as either the normal logic high idle or an inverted logic low idle format.

Most of the execution takes place in Interrupt routines. Refer to Figure 2. Notice that the main loop is only responsible for setting the DAC outputs. All other functions take place within individual interrupts. Received communication data is not buffered, so it must be taken care of as it arrives. A/D conversions proceed in the background with an interrupt taking place upon the completion of a conversion. Every TICK all joystick channels have been converted, a data string is transmitted, and the DACs are updated. This process then begins another round of ADC conversions and the Main loop is again waiting for a TICK.

I can plug a Bluetooth dongle I made using KC Wirefree's KC-21 Bluetooth data module right onto the JP2 6-pin connector (see Photo 3). This makes for a quick wireless serial interface that isn't permanently attached to any prototype. Since my desktop and laptop have Bluetooth dongles, I can quickly connect to projects. If I run a terminal program on my PC I will be able to see the streaming data output by my circuit, which is monitoring the joystick.

Viewing a list of numbers jumping around isn't exactly

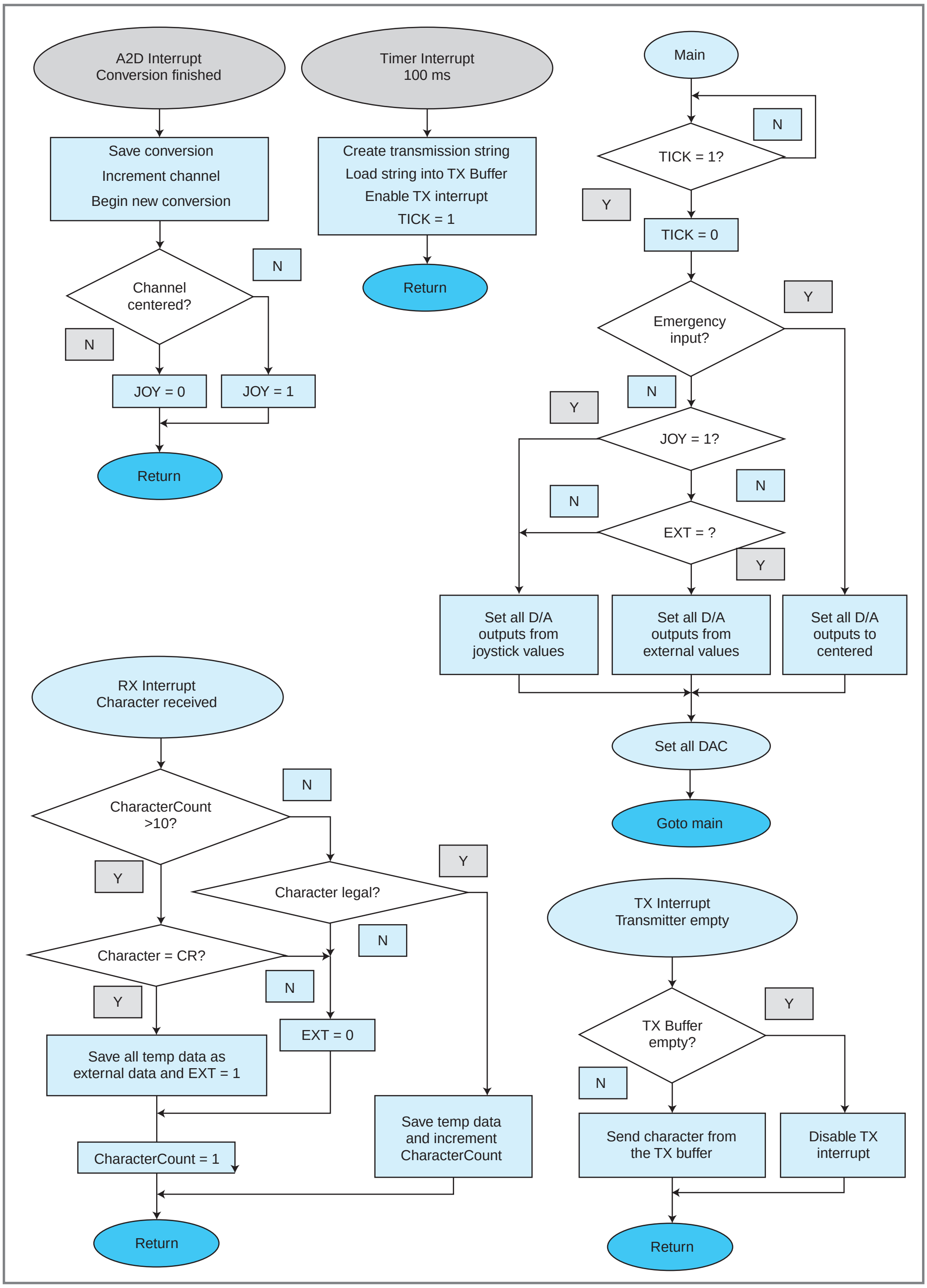


Figure 2—The interrupt routines handle the gathering and distribution of data from the joystick and between the microcontroller and the communication channel. The Main loop just has to wait for its flag (TICK) to set the DACs depending on state of the Emergency switch, JOY, and EXT.

exciting, so let's create a more expressive display. Using Liberty BASIC, a Windows-based BASIC programming language, I can create a simple graphic image of a joystick and show knob position based on the data being received from the circuit. Referring to Photo 4, you can see both a visual representation of the joystick's present position as well as the Y (vertical) and X (horizontal) axis values received via Bluetooth connection.

Photo 3—The joystick normally interfaces to the VR1 with a 0.1" 2 × 4 square pin header. This photo shows the project circuitry connected to the joystick as well as a KC Wirefree Bluetooth dongle used for wireless communication to any Bluetooth device.

VISUAL DISPLAY

Simple drawing commands enable a base graphic to be displayed consisting of circle commands that draw a filled in (red) circle along with some concentric (black) rings (see Photo 4). Drawing commands are pen based. Think of this like a pen plotter. You can move the pen around and raise (to move without drawing) and lower it (to draw). Everything here is based on "home," which in this graphic example is the center of the drawing space (graphic box). This basic graphic will be permanent, while the knob of the joystick wants to be able to move around. Therefore, it must be erased and redrawn for each frame (new position).

The knob position is based on two things, home and the data being received. To move relative to home, we need to know where home is. If we issue the Home command, we can then query this location to get a reference point in X (horizontal pixels from the left side of the graphic box) and Y (vertical pixels from the top of the graphic box). There is some debate on how the actual data should be presented (sent). The two basic alternatives are the actual ADC conversion values (simplest) and a direction and amount from center or idle (requires conversion). Since I'm a presentation kind of guy, I chose to send the actual conversion values and pack them into a single string to simplify transmissions.

The serial receiving portion of this application must parse the incoming data looking for 11-byte strings terminated with a <CR>. Once each string that is stripped off is format verified, the individual conversion values are converted into vertical (X1 and X2) data values and horizontal (Y1 and Y2) data values. These become offsets from home for the joystick knob. Depending on the scale in your application, you may need to apply a gain factor to keep this offset value within the range of the graphic box. In my case, I used none as the amount to pixel conversion of 1:1 was acceptable.

Each time a set of data is received, the knob drawing routine is called up to replace the current knob location with the new location based on the newly received data. By beginning this routine at the home position and moving the pen to the new location, the stem of the joystick is drawn from the center to the new position. Then a circle-filled command completes the knob located at the new position. The joystick position is based on a single set

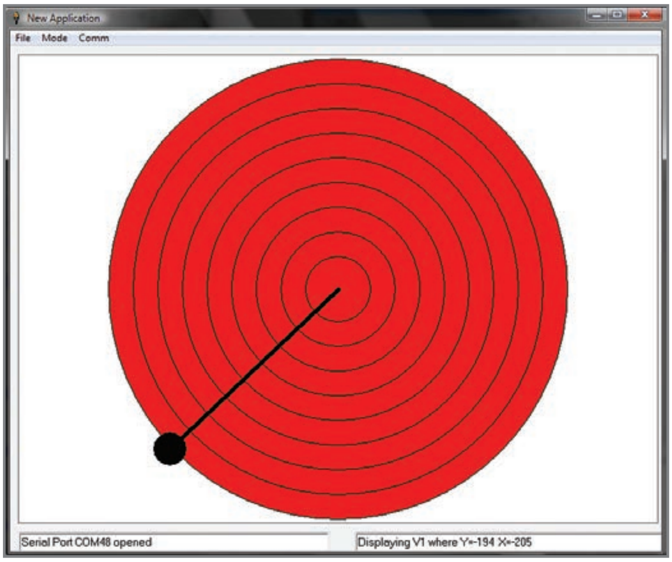


Photo 4—To verify A/D conversion through the communication channel, this PC application shows the joystick position based on received data. Its position should follow that of the physical joystick for both sets of outputs.

of X and Y positions. A pull-down menu enables the user to select between the two sets of outputs. They should be about the same if all is working well.

WHAT'S NEXT?

Next month I'll finish up this project with a closer look at the DAC operation, and I'll solve the mystery use of the emergency input. If you have some time to roam around the Internet, Google "power chairs" to see what's going on in the industry. You'll be surprised to find just how sophisticated these rigs are becoming. ■

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 the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/256.

REFERENCE

[1] Christopher & Dana Reeve Foundation, Paralysis Resource Center, www.christopherreeve.org/site/c.mtKZKgMWKwG/b.5184189/k.5587/Paralysis_Facts__Figures.htm

RESOURCES

J. Bachiochi, "Vehicle Diagnostics (Part 1): The CAN Protocol and OBD-II," *Circuit Cellar* 251, 2011.

———, "Vehicle Diagnostics (Part 2): Pre-CAN Protocol Standards," *Circuit Cellar* 252, 2011.

———, "Vehicle Diagnostics (Part 3): ISO 15765-4 (the CAN Protocol)," *Circuit Cellar* 253, 2011.

Liberty BASIC, www.libertybasic.com

SOURCES

KC-21 Bluetooth data module
KC Wirefree | www.kcwirefree.com

MCP4922 DAC
Microchip Technology, Inc. | www.microchip.com

VR1 Joystick module and JC2000 joystick
PG Drives Technology, Inc. | www.pgdt.com

Jet 3 Ultra motorized power wheelchair
Pride Mobility Products | www.pridemobility.com

NEED-TO-KNOW INFO

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

—

Programmable Power
Build a Simple USB DAC
by Yoshiyasu Takefuji
***Circuit Cellar* 213, 2008**
Yoshiyasu describes the step-by-step construction of a simple USB DAC around an ATtiny45 and a MAX517. You can use the system as a programmable power supply. Topics: USB, DAC, ATtiny45, NRZI, WinAVR, MAX517, AVR-USB, GNU C, Protocol Stack, Device Driver

—

Bluetooth-Based Display for GPS Data
by Jay Carter
***Circuit Cellar* 190, 2006**
Jay interfaced a Wintec Bluetooth module to an LCD and a PIC16F88, which connects to a miniature Bluetooth GPS receiver. The display shows the date, time, speed, direction, location, and number of the satellites used by the GPS receiver. Topics: Bluetooth, LCD, PIC16F88, GPS, Nemesis, Athena, BASIC

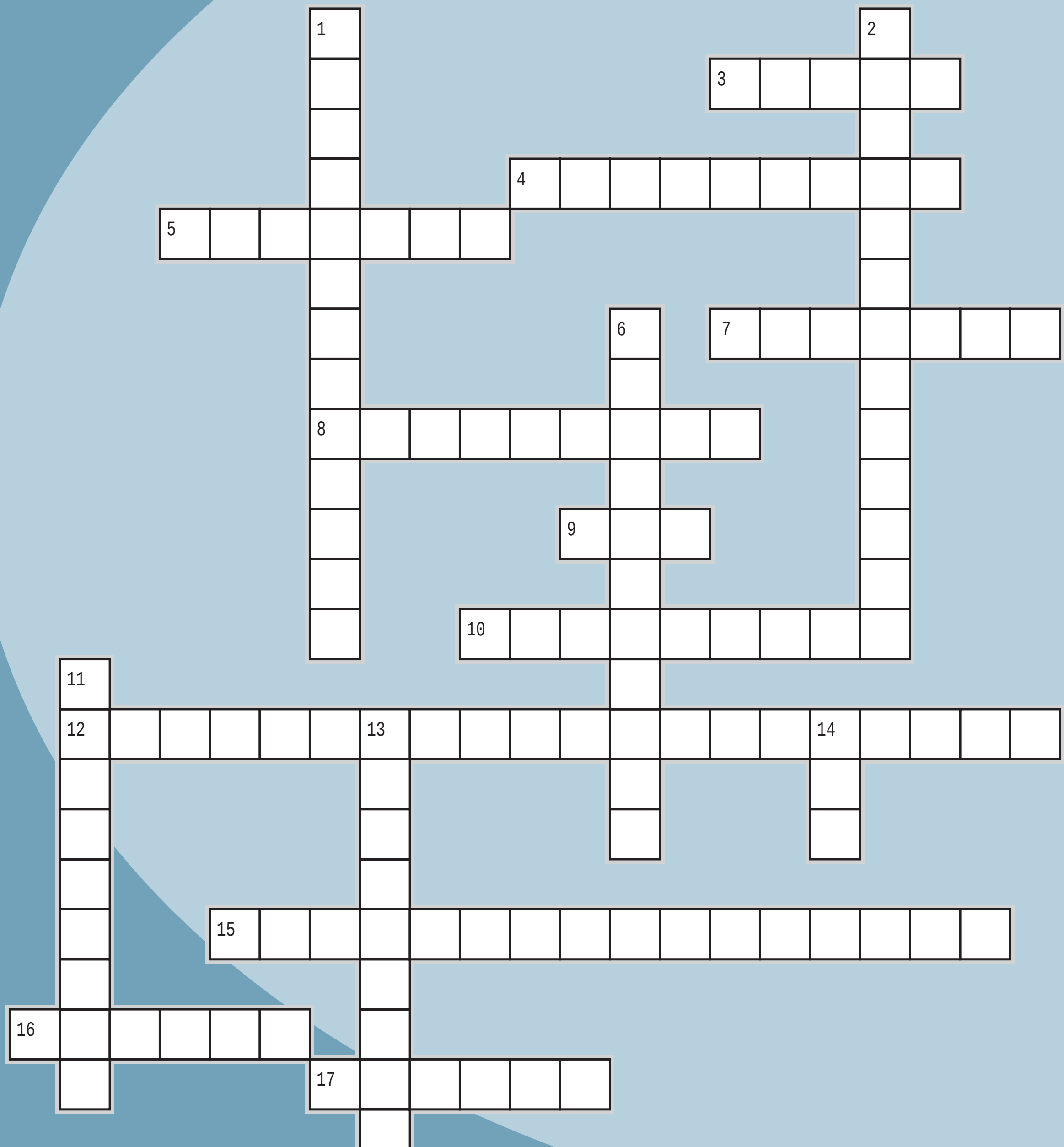
—

Joystick Technology
by Jeff Bachiochi
***Circuit Cellar* 176, 2005**
An analog joystick would be a great addition to some designs, but space limitations could present a problem. Fortunately, new devices like the JS1100AQ five-position joystick navigation switch can help you solve the space problem. Topics: Joystick, JS1100AQ, Analog, Digital, DS1869, PIC16F505, MCP42*xxx*, Resolution, PIC10F20x, PIC16C57

—

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

CROSSWORD



Across

3. Not visual
4. The foundation of an application
5. An online seminar
7. A series of lines of code
8. A device used to increase power
9. Fast movement, a sequence or course
10. A barrier of conductive or magnetic matter
12. LCD [three words]
15. IP [two words]
16. A unit of measurement
17. A plan or specification

Down

1. An arrangement of units
2. Decreases vibration
6. The amount of times a component is unsuccessful [two words]
11. A circuit with two stable states
13. Changes analog to digital and vice versa
14. Below

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.

CROSSWORD ANSWERS from Issue 255

Across

1. TIMESTAMP—The time recorded by a computer (not real time) [two words]

6. VARIABLE—Opposite of constant

7. RSSFEED—Used to automatically distribute frequently updated information [two words]

10. SCRIPT—A sequence, instructions

11. PULSEWIDTHMODULATION—PWM [three words]

13. PUSHBUTTON—Mechanism for controlling mechanical and electronic devices [two words]

16. SIGNAL—A time-varying or spatial-varying quantity

17. HENDRIKLORENTZ—Dutch physicist who first quantified the Lorentz force [two words]

18. SYSTEMONCHIP—Merges computer components into one IC chip [three words]

19. ROOTMEANSQUARE—RMS [three words]

Down

2. INTERFACE—Device used to connect a computer to a network

3. MICROSECONDS—μs

4. DIGITAL—Electronic, mathematical

5. LEASTSIGNIFICANTBIT—LSB [three words]

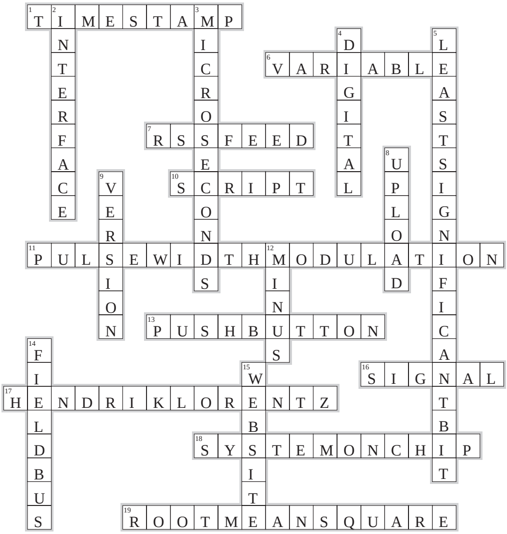
8. UPLOAD—Sends local data to a remote system

9. VERSION—Abbreviated as "V"; usually there is more than one

12. MINUS—Take away

14. FIELDBUS—Family of protocols for real-time distributed control

15. WEBSITE—Frequently ends with "com" or "org"



Index of Advertisers

Page		Page		Page		Page	
16	2012 International CES	56, 57	Elektor	11, 13	Imagineering, Inc.	C2	NetBurner
14	Abacom Technologies, Inc.	73	Elektor	76	Ironwood Electronics	C4	Parallax, Inc.
76	All Electronics Corp.	10	EMAC, Inc.	77	Jeffrey Kerr, LLC	27	Pico Technology Ltd.
29	Anaren	21	Embedded Developer	17, 33	JK microsystems, Inc.	35	Pololu Corp.
10	AP Circuits	41	ExpressPCB	75, 77	JK microsystems, Inc.	48	Saelig Co., Inc.
71	ARM	33	ezPCB	5	Labcenter Electronics	49	Sealevel Systems
77	Atria Technologies, Inc.	75	FlexiPanel Ltd.	76	Lawicel AB	2, 3	Technologic Systems
1, 78	Audio Amateur	C3	FTDI Chip	77	Maxbotix, Inc.	77	Tern, Inc.
15	Comfile Technology, Inc.	12	GHI Electronics LLC	76	MCC, Micro Computer Control	76	Triangle Research, Inc.
75	Custom Computer Services, Inc.	17	Grid Connect, Inc.	76	microEngineering Labs, Inc.	9	WIZnet Co., Inc.
75	Decade Engineering	22	Humandata Ltd.	76	Mosaic Industries, Inc.		
39	DesignSpark chipKIT™ Challenge	37, 69	IC Bank	7	Mouser Electronics, Inc.		

Preview of December Issue 257

Theme: Programmable Logic

The (VI)sualizer: A Smart Electronic Load

Sound Tone Detection with a PSoC (Part 2): The Software

Sleep-Stage Alarm Clock

THE CONSUMMATE ENGINEER **EMI and Data Integrity**

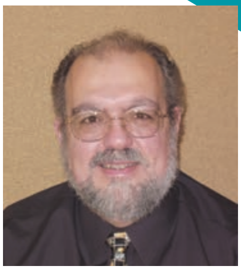
FROM THE BENCH **Fly-by-Wire Wheelchair (Part 2):** Programming and Operation

EMBEDDED IN THIN SLICES **Getting Started with Embedded Linux (Part 2):** Choosing a Platform for Your System

ABOVE THE GROUND PLANE **Stepper Motor Drives (Part 3):** Torques

THE DARKER SIDE **CE Marking:** A Process to Ensure Product Conformity

PRIORITY INTERRUPT



by Steve Ciarcia, Founder and Editorial Director

To 4G or Not to 4G?

While I don't do as much hands-on design work these days as I used to, it's hard to avoid the impact of changing technology coming from the world of smartphones and wireless gadgets. It was just a short time ago that I jumped onto mobile broadband. Now I can't live without it. Of course, the unique difference between discussing the latest technology at *Circuit Cellar* and a newspaper, for example, is that ultimately, the depth of discussion here is about implementing the technology, not just becoming an end user of it.

It's time for embedded system developers to start thinking about how to take advantage of the newly emerging mobile broadband fourth generation (4G) capabilities, which offer on the order of 100 Mbps download bandwidth and a cleaner "flat-IP" (Internet Protocol) architecture. Frankly, I'm having a hard time imagining embedded applications that need that kind of raw bandwidth—after all, even a high-definition video signal fits into about 25 Mbps—but I'm sure they must be out there.

In any case, there seem to be two standards that are being widely deployed, at least in the United States: mobile WiMAX (e.g., Sprint, Clearwire) and 4G Long-Term Evolution (LTE) (e.g., AT&T and Verizon). The latter isn't really up to 4G speeds yet; it's more like a 3G-plus (it's also been called 3.9G). The only difference moving forward will be an increase in performance. WiMAX is being developed by the non-profit WiMAX Forum, an organization of network operators and component and equipment vendors. LTE is being developed by the 3rd Generation Partnership Project (3GPP), a collaboration among telecommunications associations, which are themselves groups of manufacturers and standards bodies.

It's interesting to note that the two standards have arrived at the 4G arena from different directions. LTE is just what it says, the evolution of the cellphone network from 2G and 3G with steadily increasing bandwidth available to the end user, currently peaking out at around 42 Mbps. On the other hand, WiMAX was originally conceived as a fixed, point-to-point wireless service with bandwidth up to 1 Gbps, for "backhaul" Internet access in places where wired service isn't economical. The extension to mobile WiMAX is a relatively recent addition. In mobile applications, some reduction in bandwidth is required in order to accommodate issues such as variable path lengths and Doppler shifts, as well as the lack of directivity (and gain) in the antennas. So, the effective throughput is on the order of 100 Mbps.

A handful of module manufacturers have jumped on the 4G bandwagon to date, including Novatel Wireless and Sierra Wireless. Both companies offer modules that support both standards and all of the frequency bands currently in use, so there doesn't seem to be any issue with building hardware that's fairly network-agnostic. This will be important moving forward since, although WiMAX currently has the lead in terms of deployment, LTE, with the support of two large carriers, will quickly catch up. It's pretty clear that both technologies will remain important for end-user devices for the foreseeable future.

And it really isn't an either/or question anyway, at least not the way VHS and Beta was back in the 1980s. In this case, end users don't need to make an investment in a library of physical media, which is what forced everyone to make a choice before. With 4G, the radio part of the module is almost completely defined by software, so there's no recurring cost penalty for supporting both standards—just the nonrecurring cost of developing the firmware. This also makes the hardware relatively future-proof as well, as long as the issue of deploying firmware updates is taken into consideration.

There will certainly be a plethora of 4G products in the pipeline. A significant one right now is the 4G "hotspot"— a simple router with two wireless interfaces that functions as a bridge between the metropolitan-area 4G network and a local-area Wi-Fi network. NetComm and NETGEAR offer products like this. This is a quick way to set up a proof-of-concept for any new 4G application ideas you may have.

As originally conceived, 4G is intended to be "MAGIC" (i.e., mobile multimedia, anytime or anywhere, global, integrated, and customized) and I hope it is. If that remains true, I may also get my wish. I'm sure there will be many products I can't possibly conceive, but I only want one ASAP! Paying for a dozen individual broadband subscription fees (house, phones, cars, etc.) is absolutely driving me nuts! I just want one decent mobile 4G hotspot so I can at least get some of their hands out of my pocket.

steve.ciarcia@circuitcellar.com