

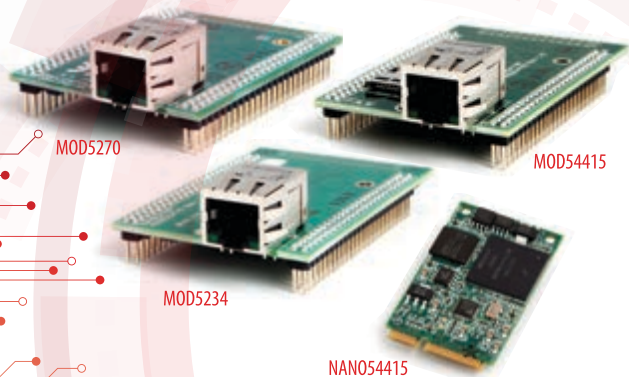


Engineer a Data Acquisition System



■ Electrical Engineering Tips (Update) | Q&A: FPGA Enthusiast ■ DIY Force-Sensing System | Build a Camera Stabilization Platform | Update to a Classic Electronic Game ■ Pooling Microarchitectural Resources | Variable Resistors | Improved PWM MOSFET Gate Driver | IR Transmissions ■ 3-D Printed Electronics

Ethernet Core Modules with High-Performance Connectivity Options



- **MOD5270**
147.5 MHz processor with 512KB Flash & 8MB RAM · 47 GPIO · 3 UARTs · I²C · SPI
- **MOD5234**
147.5 MHz processor with 2MB flash & 8MB RAM · 49 GPIO · 3 UARTs · I²C · SPI · CAN · eTPU (for I/O handling, serial communications, motor/timing/engine control applications)
- **MOD54415**
250 MHz processor with 32MB flash & 64MB RAM · 42 GPIO · 8 UARTs · 5 I²C · 3 SPI · 2 CAN · SSI · 8 ADC · 2 DAC · 8 PWM · 1-Wire[®] interface
- **NANO54415**
250 MHz processor with 8MB flash & 64MB RAM · 30 GPIO · 8 UARTs · 4 I²C · 3 SPI · 2 CAN · SSI · 6 ADC · 2 DAC · 8 PWM · 1-Wire[®] interface

Add Ethernet connectivity to an existing product, or use it as your product's core processor



The goal: Control, configure, or monitor a device using Ethernet

The method: Create and deploy applications from your Mac or Windows PC. Get hands-on familiarity with the NetBurner platform by studying, building, and modifying source code examples.

The result: Access device from the Internet or a local area network (LAN)

The NetBurner Ethernet Core Module is a device containing everything needed for design engineers to add network control and to monitor a company's communications assets. For a very low price point, this module solves the problem of network-enabling devices with 10/100 Ethernet, including those requiring digital, analog and serial control.

MOD5270-100IR.....\$69 (qty. 100)	NNDK-MOD5270LC-KIT.....\$99
MOD5234-100IR.....\$99 (qty. 100)	NNDK-MOD5234LC-KIT.....\$249
MOD54415-100IR.....\$89 (qty. 100)	NNDK-MOD54415LC-KIT.....\$129
NANO54415-200IR....\$69 (qty. 100)	NNDK-NANO54415-KIT.....\$99

NetBurner Development Kits are available to customize any aspect of operation including web pages, data filtering, or custom network applications. The kits include all the hardware and software you need to build your embedded application.

➤ **For additional information please visit**
<http://www.netburner.com/kits>

SUPERIOR **EMBEDDED** SOLUTIONS



DESIGN YOUR SOLUTION TODAY
CALL 480-837-5200

www.embeddedARM.com

TS-8820-BOX Industrial Controller Opto-Isolated Analog IO



Powered by:
TS-SOCKET
Modules

TS-8820-4700

Pricing starts at
\$659 \$608
Qty 1 Qty 100

TS-8820-4800

Pricing starts at
\$699 \$648
Qty 1 Qty 100

Features:

- 800 MHz ARM CPU
- 256 MB RAM
- 256 MB Flash Storage
- 1x microSD Socket
- 1x 10/100 Ethernet
- 2x USB Host
- 8x Opto-Isolated Inputs
- 6x Digital Inputs, 6x Out
- 16x ADC, 4x DAC Ports
- PWM, Counter, & H-Bridge

Benefits:

- Provides a variety of electrically isolated IO
- Metal enclosure with rugged screw connectors
- Convenient Power over Ethernet (PoE) powers SBC
- Customizable with programmable 5 KLut FPGA
- -40 to 85 °C industrial temperature range
- Boots Linux 2.6 in less than 3 seconds

TS-7680 Single Board Computer AC Power, Relays, WiFi, & Bluetooth



Pricing starts at
\$203 \$159
Qty 1 Qty 100

(Shown with all options)

Features:

- 454 MHz ARM CPU
- 128 or 256 MB RAM
- 2 GB Flash Storage
- 1x microSD Socket
- 2x 10/100 Ethernet
- Wireless 802.11b/g/n
- Bluetooth 4.0+EDR
- Wireless 802.11b/g/n
- 24 VAC or 8 - 24 VDC Power
- DIO, CAN, Modbus, Relays

Benefits:

- Low power and low cost industrial grade SBC
- Rugged 24-Pin screw terminal connector
- Flexible power inputs including AC and DC
- Wireless data acquisition via WiFi and Bluetooth
- 30 V tolerant DIO, analog IO, 3 A relays, and more
- Easy development w/ Debian and Linux 2.6



We've never discontinued a product in 30 years



Embedded systems that are built to endure



Support every step of the way with open source vision



Unique embedded solutions add value for our customers

Issue 290 September 2014 | ISSN 1528-0608

CIRCUIT CELLAR® (ISSN 1528-0608) is published monthly by:

Circuit Cellar, Inc.
111 Founders Plaza, Suite 300
East Hartford, CT 06108

Periodical rates paid at East Hartford, CT, and additional offices.

One-year (12 issues) subscription rate US and possessions \$50, Canada \$65, Foreign/ ROW \$75. All subscription orders payable in US funds only via Visa, MasterCard, international postal money order, or check drawn on US bank.

SUBSCRIPTIONS

Circuit Cellar, P.O. Box 462256, Escondido, CA 92046

E-mail: circuitcellar@pcspublink.com

Phone: 800.269.6301

Internet: circuitcellar.com

Address Changes/Problems: circuitcellar@pcspublink.com

Postmaster: Send address changes to

Circuit Cellar, P.O. Box 462256, Escondido, CA 92046

ADVERTISING

Strategic Media Marketing, Inc.
2 Main Street, Gloucester, MA 01930 USA

Phone: 978.281.7708

Fax: 978.281.7706

E-mail: circuitcellar@smmarketing.us

Advertising rates and terms available on request.

New Products:

New Products, Circuit Cellar, 111 Founders Plaza, Suite 300
East Hartford, CT 06108, E-mail: newproducts@circuitcellar.com

HEAD OFFICE

Circuit Cellar, Inc. 111 Founders Plaza, Suite 300
East Hartford, CT 06108
Phone: 860.289.0800

COVER PHOTOGRAPHY

Chris Rakoczy, www.rakoczyphoto.com

COPYRIGHT NOTICE

Entire contents copyright © 2014 by Circuit Cellar, Inc. All rights reserved. Circuit Cellar is a registered trademark of Circuit Cellar, Inc. Reproduction of this publication in whole or in part without written consent from Circuit Cellar, Inc. is prohibited.

DISCLAIMER

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

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

© Circuit Cellar 2014 Printed in the United States

INNOVATIVE DATA ACQUISITION SYSTEMS

It never fails. Each month, I find myself telling someone—sometimes it's a reader, other times it's a client—that the latest edition of Circuit Cellar is the most diverse to date. And each time I do, I truly believe it. Well, I'm doing it again. Why? In this issue we cover a variety of topics ranging from a DIY data acquisition system to a video camera stabilization system to an upgraded classic electronic game to 3-D printed electronics. By publishing articles on so many different topics, we're sure to pique your interest more than once.

We start the issue with an in-depth interview with Chris Zeh, a young hardware design engineer at STMicroelectronics who blogs about his interests and considers himself an FPGA aficionado (p. 6). Chris tell us about some of his most innovative projects, including the "HyperSniffer," which is an FPGA-based, application-specific logic analyzer.

On page 22, two Camosun College graduates explain how they engineered sensor-to-human data acquisition system. They built it to monitor the forces exerted by rowers on a crew team, but you can build a similar system for measuring, displaying, and logging force data for any number of other applications.

Next, a group of Cornell University students present their well-designed, microcontroller-based video camera stabilization platform (p. 32). After providing a high-level overview, they delve into the details about the project's hardware and software. They include information about how the system reads and processes data.

In "DIY RGB Game Design," Mitch Matteau details an upgrade to a classic electronic tic-tac-toe game system (p. 40). He describes his original design and then explains his recent upgrades.

Turn to page 48 for columnist Ayse Coskun's article on the topic of pooling microarchitectural resources. She explains how pooling your resources across applications can improve overall energy efficiency.

On page 54, George Novacek continues his series on resistors. In this installment, he takes a close look at variable resistors and covers their power ratings and different styles.

Mitch Matteau isn't the only engineer in this issue who revisits a previous design. This month, columnist Ed Nisley returns to his Arduino PWM MOSFET gate drive (p. 58).

Turn to page 66 for the second part in Jeff Bachiochi's article series on IR remotes. In this article, he tackles the topics of identifying, decoding, and reproducing IR transmissions.

Dr. Martin Hedges wraps up the issue with an essay on the future of one of the most exciting developments in our industry—3-D printed electronics (p. 80). As you'll see, this promising new field is poised to revolutionize the way engineers design and manufacture electrical systems.

C. J. Abate

cabate@circuitcellar.com



3-D chip packaging (Courtesy of Fraunhofer IKTS)

THE TEAM

EDITOR-IN-CHIEF

C. J. Abate

MANAGING EDITOR

Mary Wilson

ASSOCIATE EDITOR

Nan Price

ART DIRECTOR

KC Prescott

ADVERTISING COORDINATOR

Kim Hopkins

PRESIDENT

Hugo Van haecke

PUBLISHER

Don Akkermans

ASSOCIATE PUBLISHER

Shannon Barraclough

COLUMNISTS

Jeff Bachiochi, Ayse K.

Coskun, Bob Japenga, Robert Lacoste, Ed Nisley, George Novacek, Colin O'Flynn

FOUNDER

Steve Ciarcia

PROJECT EDITORS

Chris Coulston,
Ken Davidson,
David Tweed

CUSTOMER SERVICE

Debbie Lavoie

**US/UK**

Don Akkermans
+31 46 4389444
d.akkermans@elektor.com

**ELEKTOR LABS**

Wisse Hettinga
+31 46 4389428
w.hettinga@elektor.com

**GERMANY**

Ferdinand te Walvaart
+49 241 88 909-17
f.tewalvaart@elektor.de

**FRANCE**

Denis Meyer
+31 46 4389435
d.meyer@elektor.fr

**NETHERLANDS**

Harry Baggen
+31 46 4389429
h.baggen@elektor.nl

**SPAIN**

Eduardo Corral
+34 91 101 93 95
e.corral@elektor.es

**ITALY**

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

**SWEDEN**

Wisse Hettinga
+31 46 4389428
w.hettinga@elektor.com

**BRAZIL**

João Martins
+31 46 4389444
j.martins@elektor.com

**PORTUGAL**

João Martins
+31 46 4389444
j.martins@elektor.com

**INDIA**

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

**RUSSIA**

Nataliya Melnikova
+7 965 395 33 36
elektor.russia@gmail.com

**TURKEY**

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

**SOUTH AFRICA**

Johan Dijk
+31 6 1589 4245
j.dijk@elektor.com

**CHINA**

Cees Baay
+86 21 6445 2811
ceesbaay@gmail.com

OUR NETWORK

VOICE COIL

audioexpress

elektor

SUPPORTING COMPANIES

12th Int'l SoC Conference	45	IAR Systems	17
2015 International CES	37	Imagineering, Inc.	C4
Accutrace	7	Ironwood Electronics	78
Advanced Micro Circuits	78	Lemos International Co., Inc.	21
All Electronics Corp.	79	MaxBotix, Inc.	79
AP Circuits	39	Measurement Computing Co.	11
CadSoft Computer GmbH	27	microEngineering Labs	79
Custom Computer Services	78	MyRO Electronic Control Devices, Inc.	78
electronica 2014	25	NetBurner, Inc.	C2
Elektor	60, 61	R.E. Smith, Inc.	43
Elektor	69	Reach Technology, Inc.	78
Elprotronic, Inc.	47	Saelig Co., Inc.	43
EMAC, Inc.	21	Scidyne	78
ExpressPCB	35	Technologic Systems	1
Front Panel Express	39	TechTools	79
Humandata, Ltd.	79	Triangle Research International, Inc.	79

NOT A SUPPORTING COMPANY YET?

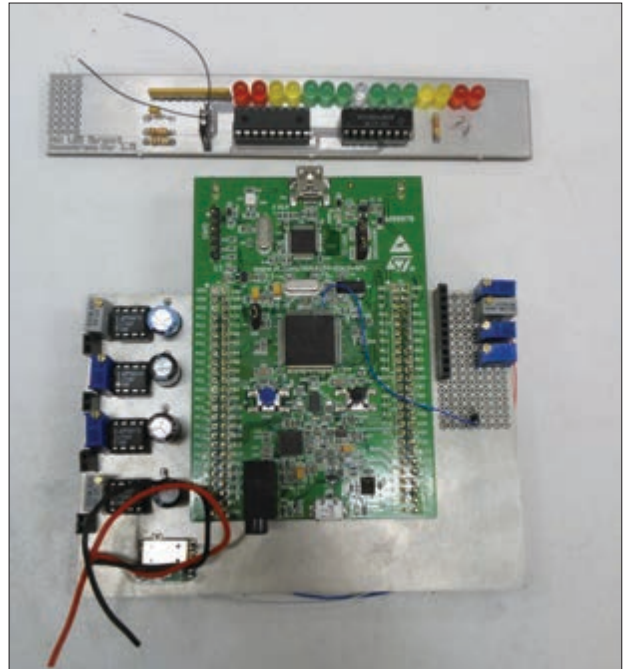
Contact Peter Wostrel (circuitcellar@smmarketing.us, Phone 978.281.7708, Fax 978.281.7706) to reserve your own space for the next edition of our members' magazine.

CONTENTS



SEPTEMBER 2014 • ISSUE 290

DATA ACQUISITION



DATA ACQUISITION SYSTEM PROVIDES FEEDBACK VIA 15 LEDs

CC COMMUNITY

06 : CC WORLD

08 : QUESTIONS & ANSWERS
 Engineer, Blogger, & FPGA Enthusiast
 By Nan Price

San Jose, CA-based engineer Chris Zeh on working with FPGA dev boards and a variety of projects

INDUSTRY & ENTERPRISE

14 : PRODUCT NEWS

21 : CLIENT PROFILE

Logic Supply (South Burlington, VT)



FPGA-BASED "HYPER-SNIFFER"—AN APP-SPECIFIC LOGIC ANALYZER

FEATURES

20 : Engineer a Force-Sensing System
 Sensor-to-User Data Acquisition Made Simple
 By Steven Liu and Albert Ruskey

Build a sensor-to-user data acquisition system for measuring, displaying, and logging information

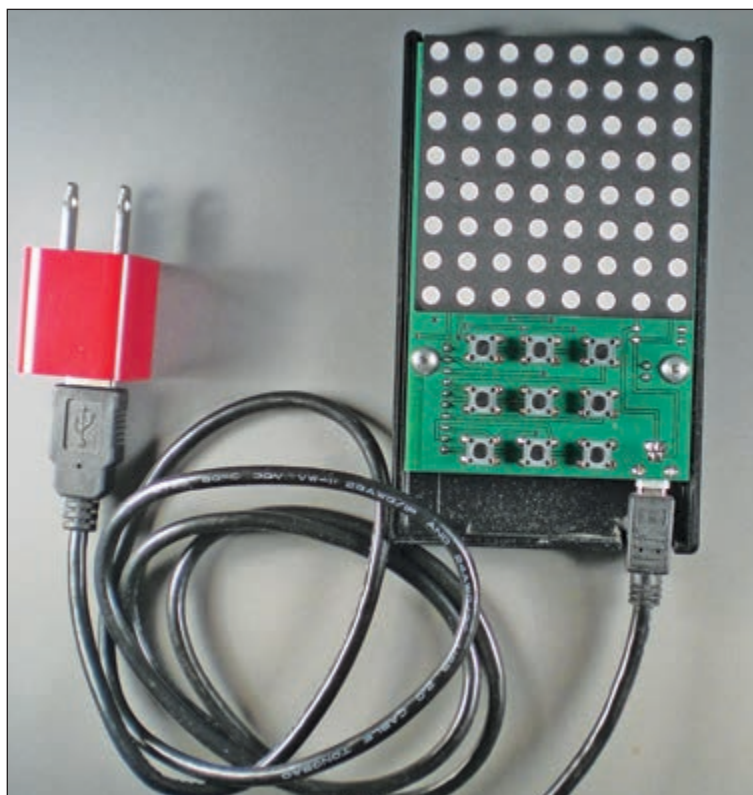
32 : Position Control
 Build a Microcontroller-Based Stabilization Platform
 By Evan Chen, Zequn Huang, and Geo Xu
 A DIY, hand-held, microcontroller-based stabilization system for camera control

40 : DIY RGB Game Design
 An Upgrade for a Classic Platform
 By Mitch Matteau
 The most recent iteration of an engineer's electronic Tic-tac-toe game system

COLUMNS

48 : GREEN COMPUTING
 Pooling Microarchitectural Resources
 Towards Flexible Heterogeneity
 By Ayse K. Coskun
 How pooling microarchitectural resources across applications improves energy efficiency

CONTENTS



UPGRADED TIC-TAC-TOE GAME SYSTEM



A MAX4544 EPOXIED ON A MOSFET



IR RECEIVER DEMODULATOR & LED TRANSMITTER

54 : THE CONSUMMATE ENGINEER**The Humble Resistor (Part 2)**

Variable Resistors

By *George Novacek*

A close look at variable resistors, including different styles and power ratings

58 : ABOVE THE GROUND PLANE**Improved Arduino PWM MOSFET Gate Drive**By *Ed Nisley*

Reworked Hall-effect LED current control circuitry and firmware

66 : FROM THE BENCH**IR Remotes (Part 2)**

IR Transmissions Explained

By *Jeff Bachiochi*

Tips on how to identify, decode, and reproduce IR transmissions



VARIOUS POTENTIOMETERS AND TRIMMERS

TESTS & CHALLENGES

74 : CROSSWORD

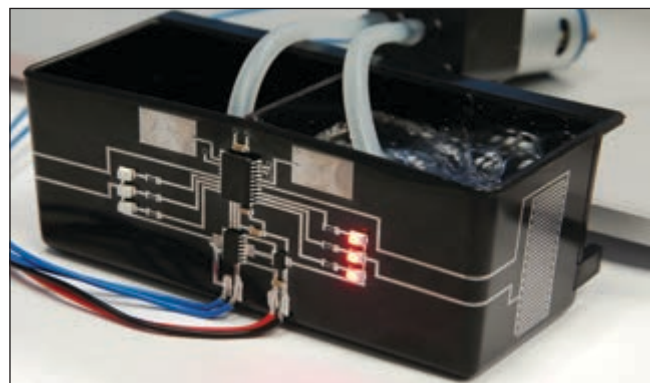
75 : TEST YOUR EQ

TECH THE FUTURE

80 : The Future of 3-D Printed Electronics

By *Dr. Martin Hedges*

An analysis of the potential for 3-D printed electronics to change the tech landscape and revolutionize production



3-D PRINTED ELECTRONICS DEMO (SOURCE: NEOTECH)

CC WORLD



EE TIPS UPDATE



By CC & EIM Staff (US & Netherlands)

A few months ago, the Circuit Cellar and Elektor team started posting electrical engineering (EE) tips on CircuitCellar.com. There are now several dozen on the site, and more are scheduled for the coming weeks. The EE Tips have been generating a lot of positive feedback. Two tips that have been shared the most during the past several weeks are “Embedded Security” (EE Tip #139) and “Don’t Trust Connectors, Solder, or Wires” (EE Tip #138).

EMBEDDED SECURITY

Embedded security is one of the most important topics in our industry. You could build an amazing microcontroller-based design, but if it is vulnerable to attack, it could become useless or even a liability.

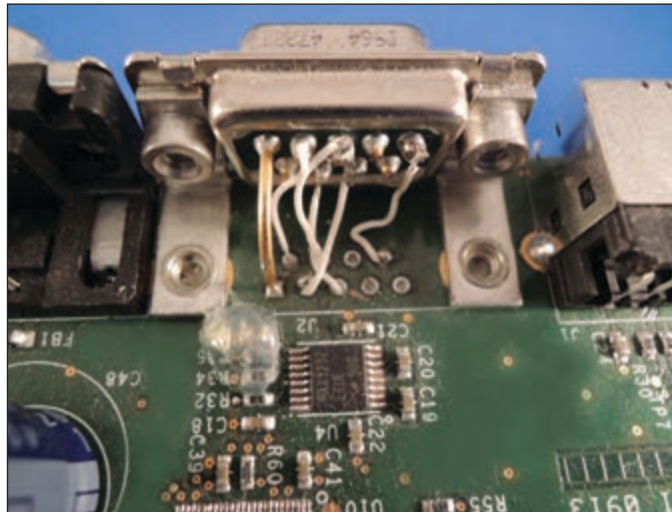
Virginia Tech professor Patrick Schaumont explains, “perfect embedded security cannot exist. Attackers have a wide variety of techniques at their disposal, ranging from analysis to reverse engineering. When attackers get their hands on your embedded system, it is only a matter of time and sufficient eyeballs before someone finds a flaw and exploits it.”

So, what can you do? In CC25, Patrick Schaumont provided some tips:

As design engineers, we should understand what can and what cannot be done. If we understand the risks, we can create designs that give the best possible protection at a given level of complexity. Think about the following four observations before you start designing an embedded security implementation.

First, you have to understand the threats that you are facing. If you don’t have a threat model, it makes no sense to design a protection—there’s no threat! A threat model for an embedded system will specify what an attacker can and cannot do. Can she probe components? Control the power supply? Control the inputs of the design? The more precisely you specify the threats, the more robust your defenses will be. Realize that perfect security does not exist, so it doesn’t make sense to try to achieve it. Instead, focus on the threats you are willing to deal with.

Second, make a distinction between what you trust and what you cannot trust. In terms of building protections, you only need to worry about what you don’t trust. The boundary between what you trust and what you don’t trust is suitably called the trust boundary. While trust boundaries were originally logical boundaries in software systems, they also have a physical meaning in embedded context. For example, let’s say that you define the trust boundary to be at the chip package level of a microcontroller. (Go to <http://bit.ly/1xXTx71> for the rest of his tips.)



Using the wrong pinout for a connector is a common error, especially on RS-232 ports where it’s approximately 50% probable that you’ll have the wrong RX/TX mapping. Swapping the rows of a connector (as you see here) is also quite common.

DON’T TRUST CONNECTORS, SOLDER, OR WIRES

Engineer Robert Lacoste is one of our go-to resources for engineering tips and tricks. When we asked him for a few bits of general engineering advice, he responded with a list of more than 20 invaluable electrical engineering-related insights. One of our team’s favorite “Lacoste tips” is this: don’t trust connectors, solder, or wires. Read on to learn more.

One of my colleagues used to say that 90% of design problems are linked either to power supplies or to connector-related issues. It’s often the case. Never trust a wire or a connector. If you don’t understand what’s going on, use your ohmmeter to check if the connections are as planned. (Do this even if you are sure they are.) A connector might have a broken pin, a wire might have an internal cut, a solder joint might be dry and not conductive, or you might simply have a faulty wiring scheme. (Go to <http://bit.ly/UNpUZk> for the rest of his tips.)

We’re always looking for new EE tips. If you have any tips, tricks, or suggestions, please send them to our editorial department. Thanks!

Send us your EE tips!



E-mail your tips and advice to editor@circuitcellar.com



Tweet your tip and advice to [@circuitcellar](https://twitter.com/circuitcellar). #eetips

PRINTED CIRCUIT BOARDS

THINK YOU CAN FIND PCB PRICES THAT BEAT OURS?

WE DARE YOU.

**BEST
PCB PRICES
IN THE
INDUSTRY**

OUR GUARANTEE:

We are so confident in our PCB pricing,
we dare you to find lower prices!
If you do, we will match the price AND
give you \$100 towards your next order!

*Visit us at www.PCB4u.com
and see why our pricing can not be beaten!*

- From same day quick turn prototype to production in under 10 days
- Full CAD and CAM review plus design rule check on ALL Gerber files
- Materials: Fr4, Rigid, Flex, Metal Core (Aluminum), Polyimide, Rogers, Isola, etc.
- HDI Capabilities: Blind/Buried Microvias, 10+N+10, Via-in-Pad Technology, Sequential Lamination, Any Layer, etc.
- Our HDI Advantage: Direct Laser Drilling, Plasma De-Smear Technology, Laser Micro via Conductive Plate Shut, etc.

Accutrace[®] inc.

www.PCB4u.com • sales@PCB4u.com • (408) 748-9600



QUESTIONS & ANSWERS



Engineer, Blogger, and FPGA Enthusiast

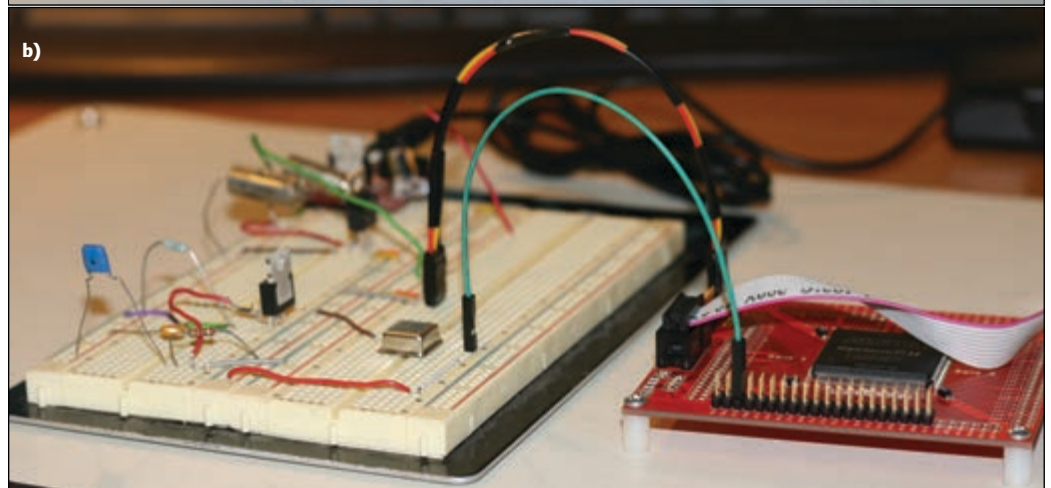
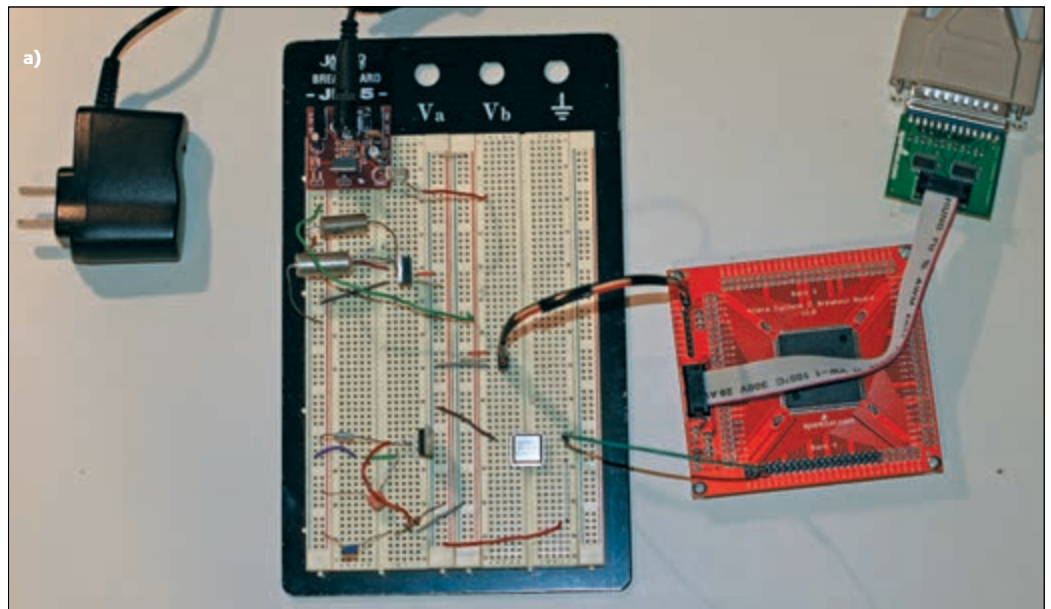
An Interview with Chris Zeh

San Jose, CA-based hardware design engineer and blogger Chris Zeh enjoys working with FPGA development boards, application-specific integrated circuits, and logic analyzers. We recently discussed some of the projects he is involved with at STMicroelectronics and on his own.—Nan Price, Associate Editor

NAN: Tell us about *Idle-Logic.com*. Why and when did you decide to start a blog?

CHRIS: I started blogging in the winter of 2009, a little more than a year after I graduated Colorado

State University with a BSEE. I realized that after graduating it was important to continue working on various projects to keep my mind and skills sharp. I figured the best way to chronicle and show off my projects was to start a blog—my



a—Chris's Saturn board prototype includes an Altera Cyclone II FPGA and JTAG FPGA programmer, two linear regulators, a 5-V breadboard power supply, and a 24-MHz clock. **b**—A side view of the board

QUESTIONS & ANSWERS

little corner of the Internet. Initially, I didn't really expect many people to visit the site, thus the main purpose was to act as a personal journal for the projects I was working on.

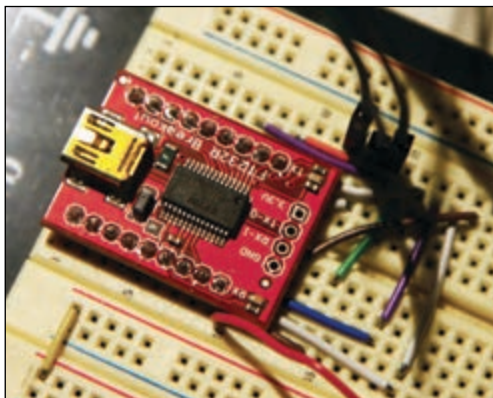
During the summer of 2009, Katie, who was then my girlfriend and is now my wife, was finishing her Master's degree in Biomedical Science and about to start four more years of schooling to get her DVM. I figured a good way to be a supportive significant other was to join her in the library. While she furiously studied cardiology, oncology, and all the other "ologies," I was busy putting together my website and working on projects (at the same time I also started University of California Berkeley Extension's Integrated Circuit Design and Techniques certificate program).

NAN: What types of projects do you feature on your site?

CHRIS: I like working on a wide range of different types of projects, varying from software development to digital and analog design. I've found that most of my projects highlighted on *Idle-Logic.com* have been ones focusing on FPGAs. I find these little reprogrammable, multipurpose ICs both immensely powerful and fascinating to work with.

My initial plan for the blog was to start a development project to create an FPGA equivalent to the Arduino. I wanted to build a main board with all the basic hardware to run an Altera Cyclone II FPGA and then create add-on PCBs with various sensors and interfaces. My main FPGA board was to be named the Saturn board, and the subsequent add-on "wings" were to be named after the various moons of Saturn.

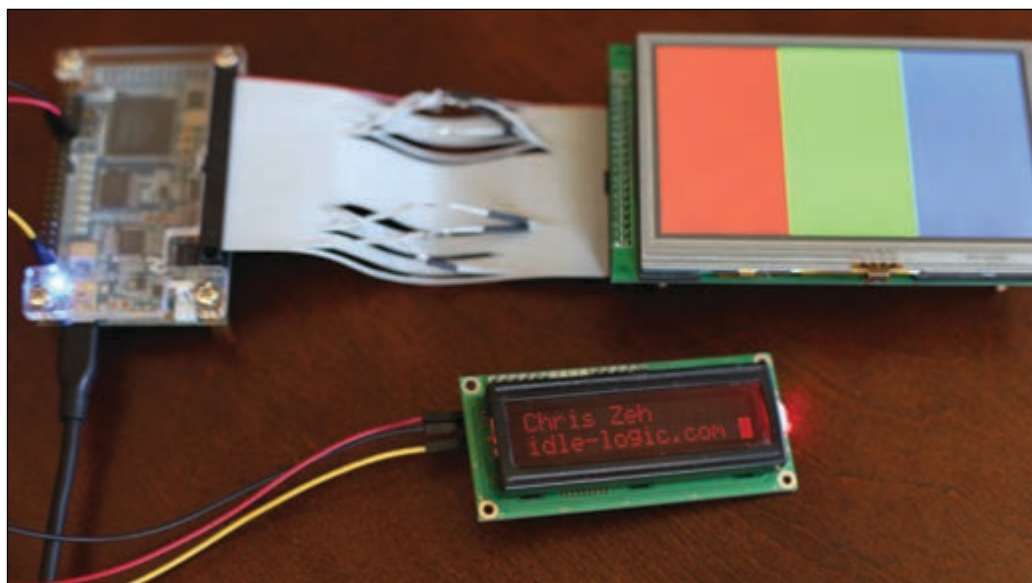
The project proceeded nicely. I spent some time brushing up on my Photoshop skills to put together a



Chris used a USB-to-UART technique to bit bang an FPGA configuration file.

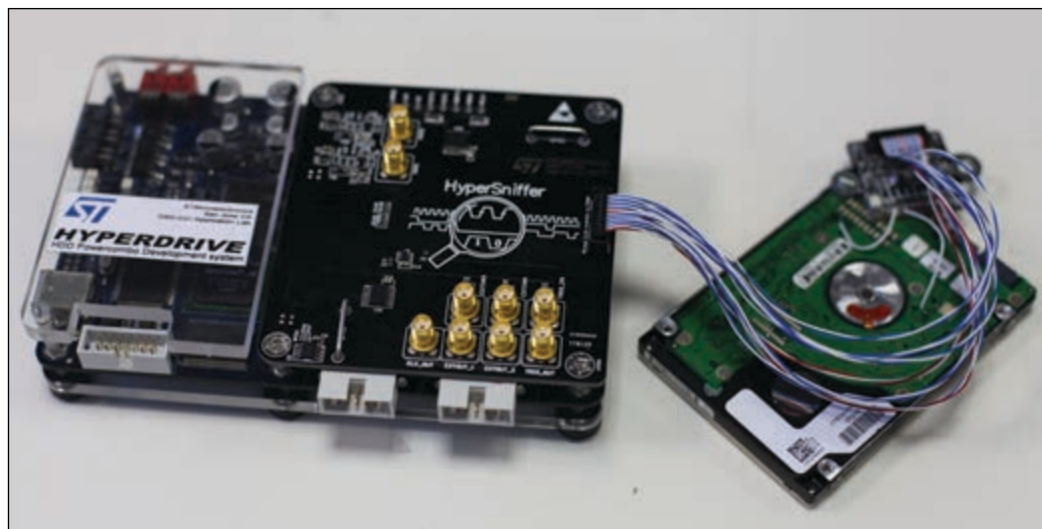
logo and came up with a minimized BOM solution to provide power to the nine different voltage supplies, both linear regulators and switched-mode supplies. One aspect of FPGAs that can make them costly for hobbyist is that the programming JTAG cable was on the order of \$300. Fortunately, there are a few more affordable off-brand versions, which I used at first. After many weeks of work, I finally had the total solution for the main FPGA board. The total cost of the prototype system was about \$150. Eventually I came up with a way to bit bang the FPGA's programming bitstream using a simple \$15 USB-to-UART IC breakout board driven by a tiny Python application, eliminating the need for the pricey cable. This Future Technology Devices International FT232RL USB-to-UART IC also provided a clock output enabling me to further reduce the component count.

The project was a success in that I was compelled to completely digest the FPGA's 470-page handbook, giving me a solid grasp of how to work with FPGAs such as the Cyclone II. The project was a failure in that the FPGA breakout board I wanted to use for the project was discontinued by



Chris is working with a Terasic Technologies DE0-Nano evaluation board to support a full-color TFT LCD touchscreen display.

QUESTIONS & ANSWERS



Chris designed the HyperSniffer logic analyzer, which is shown with the HyperDrive main board. (The PCB was designed by Vincent Himpe and Albino Migliaro.)

the manufacturer. Creating and fabricating my own four-layer board and hand soldering the 208-pin package was both prohibitively expensive and also a little daunting.

Fortunately, at that time Terasic Technologies introduced its DE0-Nano, a \$79 commercial, \$59 academic, feature-packed FPGA evaluation board. The board comes with two 40-pin general I/O plus power headers, which has become a perfect alternative base platform for FPGA development. I now intend to develop add-on “wings” to work with this evaluation board.

NAN: What do you enjoy most about working with FPGAs?

CHRIS: The FPGA is such an amazing invention. The possibility to create a digital design using hardware description language (HDL) and immediately see your creation working on your bench is fantastic. The ability to create multiple functional blocks that operate in parallel all in one device is so powerful.

I can recall how perplexed I was when first learning about the FPGA. The idea of creating various tasks that operated in parallel seemed so foreign compared to a microprocessor’s more familiar sequential operation. The ability to create complex digital designs without having to fabricate an application-specific integrated circuit (ASIC) unlocked a whole world of possibilities. Working with an FPGA comes with a steep learning curve, but I believe it is definitely worth the time and effort.

NAN: Tell us more about how you’ve been using Terasic Technologies’s DE0-Nano development and education board.

CHRIS: The main project I’ve been working on

lately with the DE0-Nano is creating and adding support for a full-color 4.3” (480 × 272 pixel) thin-film transistor (TFT) touchscreen LCD. Because of the large pin count available and reconfigurable logic, the DE0-Nano can easily support the display. I used a Waveshare Electronics \$20 display, which includes a 40-pin header that is almost but not quite compatible with the DE0-Nano’s 40-pin header. Using a 40-pin IDC gray cable, I was able to do some creative rewiring (cutting and swapping eight or so pins) to enable the two to mate with minimal effort. Eventually, once all the features are tested, I’ll fabricate a PCB in place of the cable.

There are many libraries available to drive the display, but for this project I want to develop the hardware accelerators and video pipeline from the ground up, purely through digital logic in the FPGA. I recently picked up an SD card breakout board and a small camera breakout board. Using these I would like to start playing around with image processing and object recognition algorithms.

NAN: What is your title at STMicroelectronics. What types of projects are you working on?

CHRIS: My official title is Senior Hardware Design Engineer. This title mainly comes thanks to the first project I worked on for the company, which is ongoing—an FPGA-based serial port capture and decoding tool named the HyperSniffer. However, my main role is that of an application engineer.

I spend most of my time testing and debugging our prototype mixed-signal ASICs prior to mass production. These ASICs are built for the hard disk drive industry. They provide several switch-mode power supplies, linear regulators, brushless DC motor controllers, voice coil motor actuation, and a shock sensor digital processing chain, along with the various DACs,

Great Products Great Prices • Great Support

USB-1608G

16-Channel, 16-bit DAQ



Average Rating: ★★★★★ 4.8 out of 5

"Easy to use. Good support." – NoxSC

"Great value for the money." – Frank260

Only \$399

USB-201

8-Channel, 12-Bit DAQ



Average Rating: ★★★★★ 4.8 out of 5

"Great product, great price point." – LabGuy

"Great value and ease of use." – David52

Only \$99

USB-TC

8-Channel, 24-Bit Temperature



Average Rating: ★★★★★ 4.7 out of 5

"Easy, accurate, reliable." – Patrick

"USB-TC is bullet proof." – Greg

Only \$359

USB-500 Series

Stand-Alone Data Loggers



Average Rating: ★★★★★ 4.6 out of 5

"Easy-to-use, understandable results." – Ltmmom

"Out-of-the-box great value!" – DrTom53

From \$49

mccdaq.com



The Value Leader in Data Acquisition

Contact us
1.800.234.4232

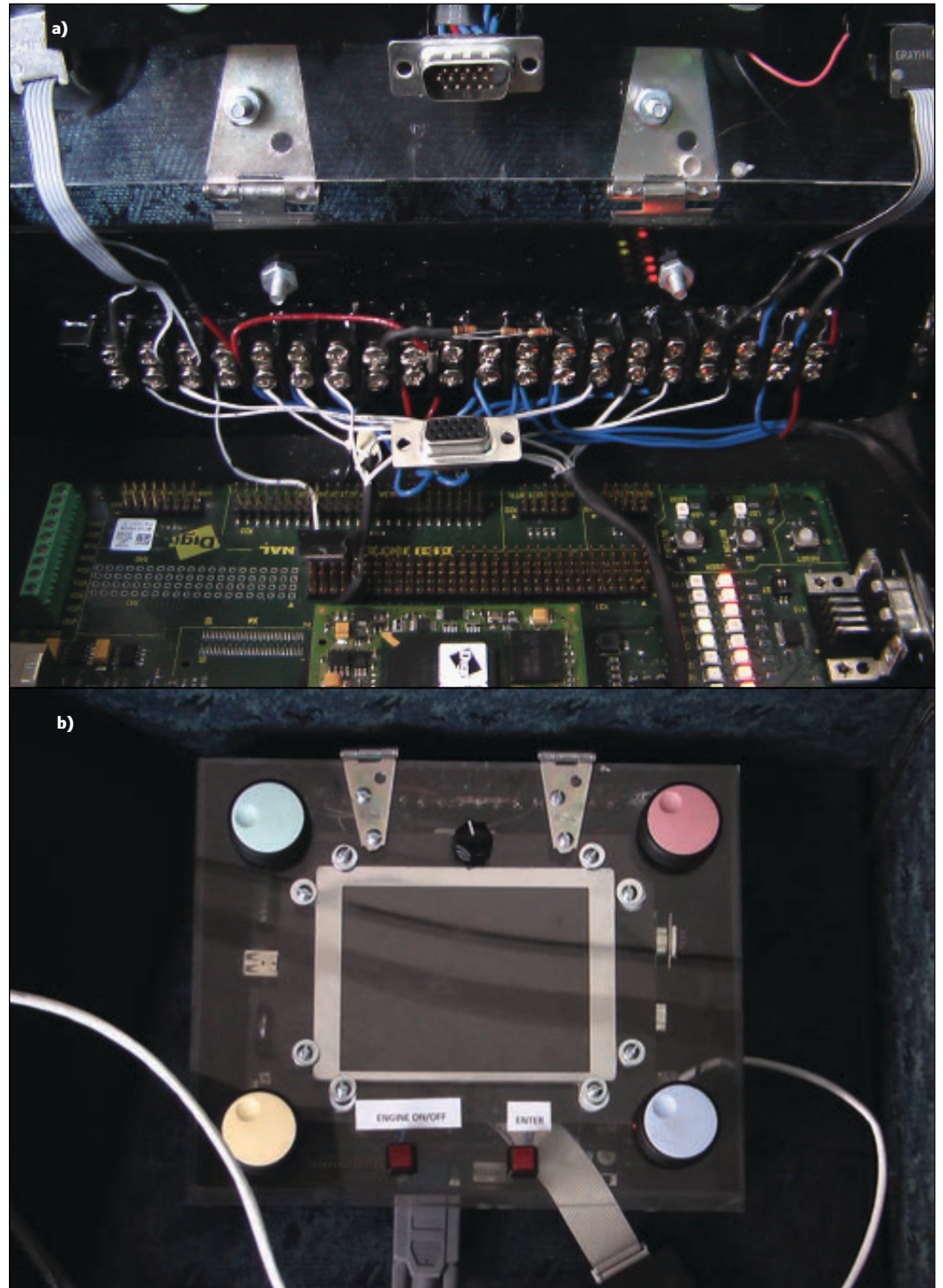
QUESTIONS & ANSWERS

ADCs, and monitoring circuits all integrated into a single IC.

Our ASIC's huge feature set requires me to stay sharp on a wide variety of topics, both analog and digital. A typical day has me down in the lab writing scripts in Python or Visual Studio, creating stimuli, and taking measurements using my 1-GHz, 10-GSPS LeCroy WavePro 7100A oscilloscope, several 6.5-digit multimeters, dynamic signal analyzers, and noise injection power supplies among other instruments. I work closely with our international

design team and our customers to help discover and document bugs and streamline the system integration.

A few years back I was able to join my colleagues in writing "Power Electronics Control to Reduce Hard Disk Drive Acoustics Pure Tones," an Institute of Electrical and Electronics Engineers (IEEE) paper published for the Control and Modeling for Power Electronics (COMPEL) 2010 conference. I presented the paper, poster, and demonstration at the conference discussing



For Chris's senior design project, he worked with a team at Colorado State University to design a prototype engine calibration controller. **a**—This inside view shows the terminal blocks and video graphics array (VGA) connector, which enabled the team to quickly swap different encoders and displays. **b**—The top view shows the 5.7" thin-film transistor (TFT) LCD and various rotary encoders on the custom acrylic enclosure, which swings open to reveal a Digi International ConnectCore 9 evaluation board.

QUESTIONS & ANSWERS

a novel technique to reduce acoustic noise generated by a spindle motor.

NAN: Tell us more about the HyperSniffer project.

CHRIS: The HyperSniffer project is an FPGA-based digital design project I first created right out of college. (My colleagues Vincent Himpe and Albino Migliaro did the board design and layout.) The tool is basically an application-specific logic analyzer. It enables us to help our customers troubleshoot problems that arise from serial port transmissions between their system-on-a-chip (SoC) and our ASIC. Through various triggering options it can collect and decode the two or three wire data transmissions, store them on on-board memory, and wait for retrieval and further processing by the application running on the PC. One of this tool's nice features is that it is capable of synchronizing and communicating with an oscilloscope, enabling us to track down problems that happen in the analog domain that arise due to commands sent digitally.

NAN: Tell us about your software developer internship at Nestle Purina. Can you share any interesting experiences?

CHRIS: I took this internship right out of high school, in the summer before starting at Colorado State University. Initially, I was hired to help convert 300 database-backed web applications from an old programming language to a new one.

I've been programming since I was very young, so I had a talent and a knack for writing software to make life easier. I was able to automate the conversion process and finished a full two months early. After demonstrating my abilities, the scope of my internship was expanded to include a full redesign of the internal website. I continued to work for Nestle Purina for the following two summers and holiday breaks, writing applications to process manufacturing data and help automate report generation. It was a fun introduction to the working world. I had a lot of autonomy and they really let me take on any project I wanted. I spent a lot of time learning how to write clean, reliable, and professional code—an experience that benefits me to this day.

NAN: You also did an internship at Xilinx. Can you elaborate on the FPGA research you conducted?

CHRIS: This was a short internship for a few months between semesters. My research mainly focused on evaluating the accuracy of the power planning tools integrated into the Xilinx ISE design suite, comparing the projected power consumption vs the actual. Additionally, I was tasked to compare the power performance of several designs on hardware from the various

FPGA manufactures. Much of the work was spent creating tools to collect data and generate visualizations for comparisons and marketing purposes.

NAN: How long have you been designing embedded systems? When did you become interested?

CHRIS: The first real embedded system I worked on was during my senior design project at Colorado State University. I was the project leader for a group of four students. Our goal was to develop an embedded system that would provide a clean user interface to assist in the calibration of the internal combustion engine. The device was developed to communicate with an engine controller to modify parameters such as fuel quantity and spark timing.

The team built a prototype enclosure that housed a Digi International ConnectCore9 evaluation board, a 5.7" TFT LCD, several quadrature encoders, and various buttons. The ARM microprocessor ran Digi International's NET+OS RTOS. We were able to quickly develop a GUI using the wxWidgets cross-platform library. The project was a success and we placed third in a year-end competition among all the other senior design project groups.

NAN: What do you consider to be the "next big thing" in the industry?

CHRIS: Judging by the number of people I see walking down hallways or sidewalks staring down at their smartphones, I have to believe that optical head-mounted displays (e.g., Google Glass) are going to be ubiquitous one day soon. In my opinion, the biggest drawback to these types of wearable computers, aside from cost, is the interface. Not only do you stand out in a crowd just by wearing the glasses, but you further alienate yourself by having to verbally control the hardware. I would imagine that adding small tactile remote control would be a big improvement. Maybe even just adding a few buttons (up/down/left/right, enter, and back) on the reverse side of a smartphone—which you could use without having to look down—would improve the head-mounted display's usability and reduce the awkwardness factor.

The market for wearable embedded systems for fitness really seems to be taking off. The ability to track and monitor fitness goals using motion-tracking armbands seems to be a real hit. I think the gamification of fitness using these wearables really helps to encourage healthy habits. For many, the desire to achieve their target step count or distance each day while having their accountability literally strapped to them is quite motivating. Some of the new technologies on the horizon capable of tracking even more metrics (e.g., heart rate) are really intriguing. 

PRODUCT NEWS

DUAL-PHASE BOOSTS STEP-UP EFFICIENCY

Linear Technology Corp. recently introduced the **LTC3124** two-phase, 3-MHz current-mode synchronous boost DC/DC converter. It features output disconnect and inrush current limiting. Dual-phase operation has the benefit of reducing peak inductor and capacitor ripple currents. This allows equivalent performance to be achieved in the power supply design with smaller valued inductors and capacitors.

The LTC3124 incorporates low resistance MOSFETs with an RDS(ON) of 130 mΩ (N-channel) and 200 mΩ (P-channel) to deliver efficiencies as high as 95%. The output disconnect feature allows the output to be completely discharged at shutdown and reduces switch-on inrush. An input pin can be used to configure the LTC3124 for continuous frequency mode to give low-noise operation. Additional features include external synchronization, output overvoltage protection, and robust short-circuit protection.

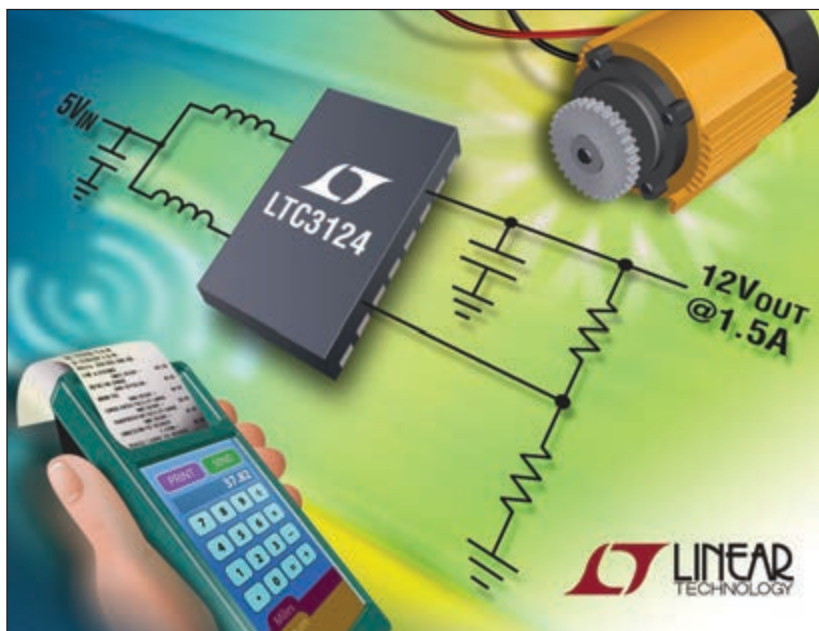
The LTC3124's main features include:

- VIN Range: 1.8 V to 5.5 V, 500 mV after start-up
- Adjustable output voltage: 2.5 V to 15 V
- 1.5-A Output current for VIN = 5 V and VOUT = 12 V
- Dual-phase control reduces output voltage ripple
- Output disconnects from input when shut down
- Synchronous rectification: up to 95% efficiency
- Inrush current limit
- Up to 3-MHz programmable switching frequency synchronizable to external clock
- Selectable Burst Mode operation: 25-μA IQ

- Output Overvoltage Protection
- Internal soft-start
- Less than 1 μA IQ in shutdown

The LTC3124EDHC and LTC3124EFE are both available from stock in 16-lead 3 mm x 5 mm DFN and thermally enhanced TSSOP packages, respectively. One-thousand-piece pricing starts at **\$3.26** each.

Linear Technology
www.linear.com



EIGHT-CORE, 64-BIT PROCESSOR TARGETS MOBILE DEVICES

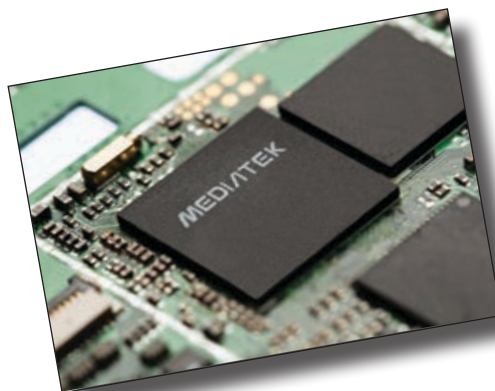
MediaTek recently announced the **MT6795**, which the company is targeting at the high-end Android 4G smartphones and tablet segment. According to the company, the eight-core processor also supports 2560 x 1600 resolution displays, FDD/TDD LTE technology, 802.11ac Wi-Fi, Bluetooth, GPS, FM Radio, and 2G and 3G wireless networks.

The chip also supports video recording and playback at Ultra HD (4K2K) resolution using the H.265, H.264 and VP9 formats, supporting high-speed 1080p video recording at up to 480 frames per second allowing slow-motion playback on screens with 120 Hz refresh. An integrated 16-MP camera image signal processor handles video input and MediaTek's ClearMotion technology eliminates motion jitter to ensure smooth video playback at 60fps.

The MT6795 uses eight ARM Cortex-A53 processors, based on a 28-nm process that clocks at 2.0 GHz and a Mali-T760 GPU to handle display control. MediaTek also supplies its CorePilot technology, which provides multicore processor performance and thermal control of the chip. The MT6795 also supports dual-channel LPDDR3 memory at 933 MHz.

According to MediaTek, we can expect to see 4G smartphones using MT6795 chips before the end of 2014.

MediaTek, Inc.
www.mediatek.com



PRODUCT NEWS

EMBEDDED SOM WITH LINUX-BASED RTOS

National Instruments has introduced an embedded **system-on-module (SOM) development board** with integrated Linux-based real-time operating system (RTOS).

Processing in the 2" x 3" SOM comes from a Xilinx Zynq-7020 all programmable SOC running a dual core ARM Cortex-A9 at 667 MHz. A built-in, low-power Artix-7 FPGA offers 160 single-ended I/Os and its dedicated processor I/O include Gigabit Ethernet USB 2.0 host, USB 2.0 host/device, SDHC, RS-232, and Tx/Rx. The SOM's power requirements are typically 3 to 5 W.

The SOM integrates a validated board support package (BSP) and device drivers together with the National Instruments Linux real-time OS. The SOM board is supplied with a full suite of middleware for developing an embedded OS, custom software drivers, and other common software components.

The LabVIEW FPGA graphical development platform eliminates the need for expertise in the design approach using a hardware description language.

National Instruments
www.ni.com



RASPBERRY PI MODEL B+

The Raspberry Pi foundation announced what it calls "an evolution" of the Raspberry Pi SBC. Compared to the previous model, the new **Raspberry Pi Model B+** has more GPIO, and more USB ports. In addition, it uses Micro SD memory cards and improved power consumption.

The GPIO header is now 40 pins, with the same pinout for the first 26 pins as the Model B. The B+ also has four USB 2.0 ports (compared to two on the Model B) and better hotplug and overcurrent behavior. In place of the old friction-fit SD card socket is a better push-push micro SD version.

In line with today's electronic concepts, the new board also lowers power consumption. By replacing linear regulators with switching ones, the power requirements are reduced by between 0.5 W and 1 W. The audio circuit incorporates a dedicated low-noise power supply, enabling better audio applications.

The new board is well organized. The USB connectors are aligned with the board edge, and the composite video now has a 3.5-mm jack. The corners are rounded with four squarely placed mounting holes.

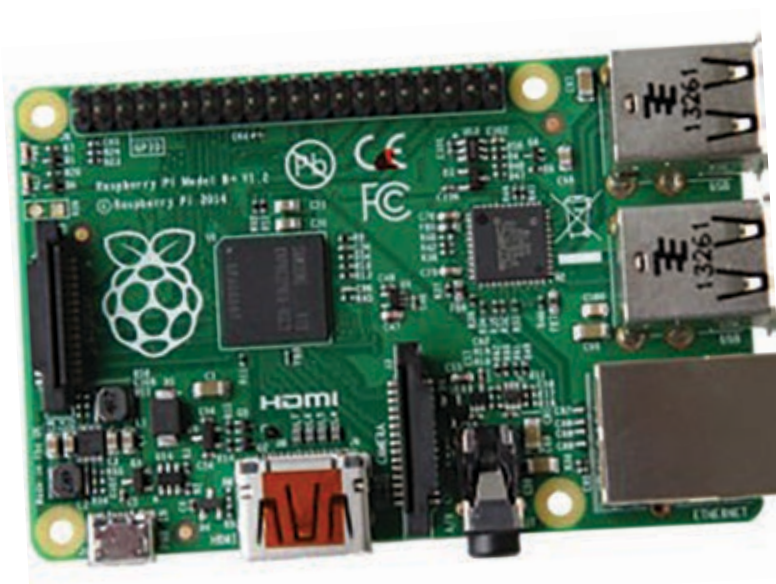
The Raspberry Pi Model B+ uses the same BCM2835 application processor as the Model B. It runs the same software and still has 512-MB RAM.

If you want to adapt a current project to the new platform, be sure to study the new GPIO pins and mechanical specs. To ensure continuity of supply for industrial customers, the Model B will be kept in production for as

long as there's demand for it.

At **\$35**, the new model B+ is the same price as the older model B and is already available from Farnell/element14/Newark and RS/Allied Components.

Raspberry Pi Foundation
www.raspberrypi.org



PRODUCT NEWS

JANSR+ 100KRAD TRANSISTORS FOR RADIATIVE ENVIRONMENTS

STMicroelectronics recently announced it is bringing into the JANS system the innovation released last year within the European Space Components Coordination (ESCC) program. Called **JANSR+**, the innovation consists of a series of 100krad JANSR high-dose-rate bipolar transistors with an additional 100krad low-dose-rate (100 mrad/s) test performed on each wafer.

Furthermore, ST has announced it will complete its JANSR+ offer with data from very-low-dose-rate (10 mrad/s) tests, demonstrating the outstanding robustness to radiation effect of its technology.

As a result, ST's JANSR+ series gives access to products with superior performance in radiative environments, with complete test data to support the claim. These products

can be used without any up-screen cost and lead time, thus dramatically raising the bar in the industry.

All parts are housed in advanced hermetic UB packages and are available in sample and volume quantities.

STMicroelectronics
www.st.com

NEW DUAL-CHANNEL FUNCTION/ARBITRARY WAVEFORM GENERATORS

B&K Precision recently launched its new **4060 Series** line of dual channel function/arbitrary waveform generators. The series includes three models that generate sine waveforms up to 80 MHz (4063), 120 MHz (4064), and 160 MHz (4065). Featuring an advanced pulse generator and high-performance 512k-point arbitrary waveform generator on one channel, these instruments are ideal for use in applications requiring high signal fidelity with extensive modulation and arbitrary waveform capabilities at a value price point.

Unique to the 4060 Series are its advanced pulse generation capabilities. The instruments can generate pulses with low jitter less than 100 ps and output 12 ns width pulses at frequencies as low as 0.1 Hz—a feature not typically found in DDS generators. Rise and fall times are also adjustable within a large range, e.g. users can output pulses with 6 ns rise times combined with 6 s fall times.

All models provide two independent output channels with a large 4.3" color LCD display, rotary control knob, numeric keypad, and intuitive function keys to make waveform adjustments quick and easy. Other standard features include a built-in counter, Sync output, trigger I/O terminal, and external 10 MHz reference clock input and output for synchronization of the instrument to another generator.

The 4060 Series offers linear and logarithmic sweep function and a wide variety of modulation schemes for modulated signal applications: amplitude and frequency

modulation (AM/FM), double sideband amplitude modulation (DSB-AM), amplitude and frequency shift keying (ASK/FSK), phase modulation (PM), and pulse width modulation (PWM).

Equipped with a high performance 14-bit, 500 MSa/s, 512k-point arbitrary waveform generator, the 4060 Series provides users 36 built-in arbitrary waveforms and the ability to create and load up to 32 custom arbitrary waveforms using the included waveform editing software via standard USB interface on the rear. A front panel USB host port is available for users to conveniently save waveforms and setups on a USB flash drive or to connect the optional USB-to-GPIB adapter for GPIB connectivity.

Available immediately, B&K Precision's 4060 Series products are backed by a standard three-year warranty at the following list prices:

- 4063 — 80 MHz — \$1,350
- 4064 — 120 MHz — \$1,670
- 4065 — 160 MHz — \$1,990

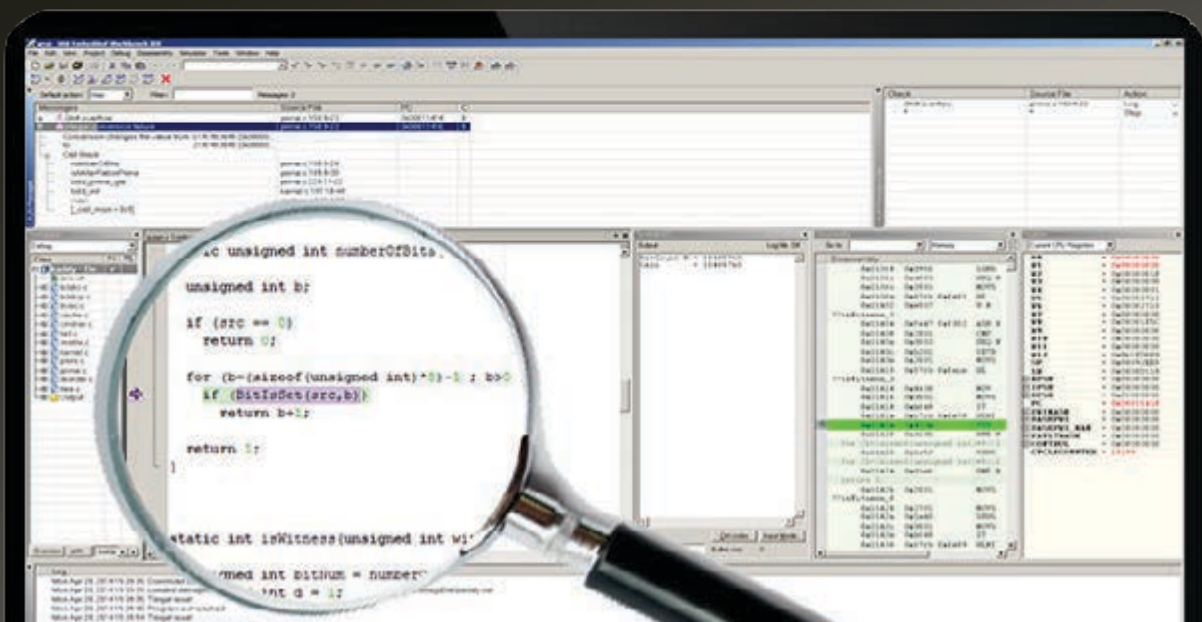
For additional technical specifications, accessories, photos, and support documents, visit B&K Precision's website.

B&K Precision Corp.
www.bkprecision.com



EXPLORE C-RUN FOR ARM

Runtime Analysis Simplified



C-RUN is a high-performance runtime analysis add-on product, fully integrated with world-leading C/C++ compiler and debugger tool suite IAR Embedded Workbench.

C-RUN performs runtime analysis by monitoring application execution directly within the development environment.

The tight integration with IAR Embedded Workbench improves development workflow and provides each developer with access to runtime analysis that is easy-to-use.

www.iar.com/crun



PRODUCT NEWS

USB-230 SERIES: NEW LOW-COST 16-BIT DAQ

Measurement Computing Corporation recently announced the release of two, 16-bit, multifunction USB DAQ devices with sample rates up to 100 ksps. The **USB-230 Series** includes low-cost 16-bit multifunction USB devices. Each device features eight single-ended/four differential analog inputs, eight digital I/O, one counter input, and two 16-bit analog outputs. Removable screw-terminal connectors make signal connections easy.

The USB-231 costs **\$249** and has a 50-ksps sample rate. The USB-234 offers a 100-ksps sample rate and is available for only **\$424**.

Included software options for the USB-230 Series include out-of-the-box TracerDAQ for quick-and-easy logging and displaying of data, along with comprehensive support for C, C++, C#, Visual Basic, and Visual Basic .NET. Drivers are also included for DASyLab and NI LabVIEW.

Measurement Computing
www.mccdaq.com



SDK FOR OPENCL DEV FLOW

Altera Corp. has simplified a programmer's ability to accelerate algorithms in FPGAs. The **Altera SDK for OpenCL version 14.0** includes a programmer-familiar rapid prototyping design flow that enables users to prototype designs in minutes on an FPGA accelerator board. Altera, along with its board partners, further accelerate the development of FPGA-based applications by offering reference designs, reference platforms and FPGA development boards that are supported by Altera's OpenCL solution. These reference platforms also streamline the development of custom FPGA accelerators to meet specific application requirements.

Altera is the only company to offer a publicly available, OpenCL conformant software development kit (SDK). The solution allows programmers to develop algorithms with the C-based OpenCL language and harness the performance and power efficiencies of FPGAs. A rapid prototyping design flow included in the Altera SDK for OpenCL version 14.0 allows OpenCL kernel code to be emulated, debugged, optimized, profiled and re-compiled to a hardware implementation in minutes. The re-compiled kernels can be tested and run on an FPGA immediately, saving programmers weeks of development time.

Altera and its board partners further simplify the experience of getting applications up and running using FPGA accelerators by offering a broad selection of Altera-developed reference platforms, reference designs and FPGA accelerator boards. Altera provides a variety of design examples that demonstrate how to describe applications in OpenCL, including OPRA FAST Parser for finance applications, JPEG decoder for big data applications and video downscaling for video applications.

Design teams that want to create custom solutions that feature a unique set of peripherals can create their own custom FPGA accelerators and save significant development time by using Altera-developed reference platforms. The reference

platforms include an SoC platform for embedded applications, a high-performance computing (HPC) platform and a low-latency network-enabled platform that utilizes IO Channels.

One notable enhancement is production support for I/O Channels that allow streaming data into and out of the FPGA as well as kernel channels allowing the result reuse from one kernel to another in a hardware pipeline for significantly higher performance and throughput with little to no host and memory interaction. Another enhancement is production support for single-chip SoC solutions (Cyclone V SoC and Arria V SoC), where the host is an embedded ARM core processor integrated in the FPGA accelerator.

Altera's SDK for OpenCL allows programmers to take OpenCL code and rapidly exploit the massively parallel architecture of an FPGA. Programmers targeting FPGAs achieve higher performance at significantly lower power compared to alternative hardware architectures, such as GPUs and CPUs. On average, FPGAs deliver higher performance at one-fifth the power of a GPU. Altera's OpenCL solutions are supported by third-party boards through the Altera Preferred Board Partner Program for OpenCL. Visit www.altera.com/opencl.

The Altera SDK for OpenCL is currently available for download on Altera's website (www.altera.com/products/software/opencl/opencl-index.html). The annual software subscription for the SDK for OpenCL is **\$995** for a node-locked PC license. For additional information about the Altera Preferred Board Partner Program for OpenCL and its partner members, or to see a list of all supported boards and links to purchase, visit the OpenCL section on Altera's website.

Altera Corp.
www.altera.com

PRODUCT NEWS

NEW BRUSHLESS DC MOTORS

Maxon's new **brushless** drive with a diameter of 19 mm has been specially designed for high speeds and features low heat development and extremely quiet and low-vibration running. This little powerhouse has applications in centers for miniaturized processing, hand-held tools and medical technology. When combined with planetary gearheads, many possible variations are available for applications in these key markets.

With its modular construction, this new brushless DC motor is available in three performance classes: the cost-optimized EC 19 at 60 W, the strong and high-speed EC 19 at 120 W and the sterilizable high-end version EC 19 at 120 W. The internals of the drives are responsible for their individual characteristics:

- The EC 19 at 60 W is equipped with a magnetic circuit that is designed for speeds of up to 80,000 rpm. The maximum continuous torque is 12 mNm and the motor contains no Hall sensors.
- The EC 19 at 120 W has been designed for speeds of up to 100,000 rpm and delivers a high continuous torque of 24 mNm. Versions with and without Hall sensors are available.
- The EC 19 Sterilizable at 120 W may be sterilized in an autoclave 1000 times. With a maximum permissible speed of 100,000 rpm and 24 mNm continuous torque, this drive also withstands high temperatures and harsh steam cycles. This is achieved through careful selection of materials and through protection of the components such as in the hermetically sealed rotor magnet.

With Maxon's winding technology, it is possible to offer various voltage versions. The stator has been designed without slots. Therefore, no cogging torque occurs resulting in excellent control properties and extraordinarily smooth running. Together with an optimally balanced rotor, low-noise and low-vibration operation is easily achieved.

Maxon's new sterilizable planetary gearhead, the GP 19 M is ideal for applications requiring very high speeds – input speeds of up to 40,000 rpm. To accommodate such high speeds, the toothing and materials have been especially designed to minimize friction. A key demand placed on the material is 1000 sterilization cycles. The gearhead contains a special shaft seal. Worth noting is the no-need to disassemble the drive unit for the sterilization process.

The new 19-mm drives are particularly suitable for surgical and dental devices, such as arthroscopic shavers and bone drills. There is also a focus on respirators and CAD/CAM spindle drives.

Maxon Precision Motors, Inc.
www.maxonmotorusa.com



PRODUCT NEWS

KERNEL RTOS EVALUATION KIT

eSOL has started offering the **eT-Kernel Evaluation Kit** for Xilinx's Zynq-7000 All Programmable SoC, which combines the dual-core ARM Cortex-A9 MPCore processor with Xilinx's 28-nm programmable logic fabric. The Evaluation Kit integrates eSOL's eT-Kernel Multi-Core Edition real time operating system (RTOS), its dedicated eBinder integrated development environment (IDE), middleware components, and device drivers. This complimentary 30-day Evaluation Kit permits developers to easily and quickly evaluate the performance and quality of Xilinx Zynq-7000 All Programmable SoC and eT-Kernel. Since eT-Kernel inherited the functions and architecture of uITRON, the most popular RTOS in Japan and Asian countries, developers can reuse their uITRON-based software assets without further work.

Runtime software in the Evaluation Kit includes the eT-Kernel Multi-core Edition, eSOL's PrFILE2 FAT file system, the SD memory card driver, and the

HDMI display driver, all of which are integrated and immediately run on the Zynq-7000 All Programmable SoC Evaluation Board. The eBinder IDE is available for eT-Kernel Multi-Core Edition-based software development. Besides ARM's genuine compiler, eBinder offers various development tools and functions to strongly support multi-programming, debugging, and analysis for complex multi-core software development.

Zynq-7000 All Programmable SoC tightly integrates two ARM Cortex-A9 MPCore processors and FPGA fabric. The hardware and software programmability of Zynq-7000 AP SoC enables system development with high performance, flexibility, and scalability, while achieving lower power consumption and cost. The eT-Kernel/Zynq-7000 All Programmable SoC Evaluation Kit allows developers to jump-start their evaluation using packaged device drivers, which saves the time and money of developing them. Zynq-7000 All

Programmable SoC and the eT-Kernel Platform are an ideal combination for advanced embedded systems in the automotive, industrial, and medical arenas, including Automotive Driver Assistance Systems (ADAS), high-resolution graphic systems, machine vision systems, and network systems.

eSOL Co., Ltd
www.esol.com

24-CHANNEL DIGITAL I/O INTERFACE FOR ARDUINO & COMPATIBLES

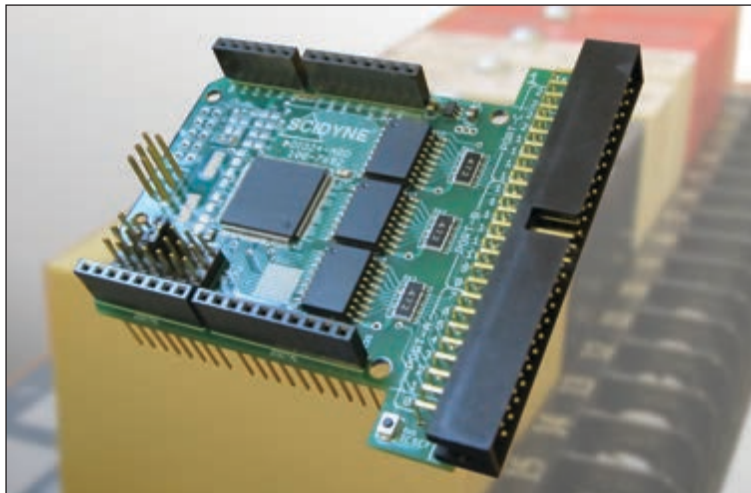
SCIDYNE Corp. recently expanded its product line by developing a digital I/O interface for Arduino hardware. The **DIO24-ARD** makes it easy to connect to solid-state I/O racks, switches, relays, LEDs, and many other commonly used peripheral devices. Target applications include industrial control systems, robotics, IoT, security, and education.

The board provides 24 nonisolated I/O channels across three 8-bit ports. Each channel's direction can be individually configured as either an Input or Output using standard SPI

library functions. Outputs are capable of sinking 85 mA at 5 V. External devices attach by means of a 50 position ribbon-cable style header.

The DIO24-ARD features stack-through connectors with long-leads allowing systems to be built around multiple Arduino shields. It costs **\$38**.

SCIDYNE Corp.
www.scidyne.com



CLIENT PROFILE

Logic Supply

www.logicsupply.com

35 Thompson St., South Burlington, VT 05403

CONTACT: Erika Austin (erika.austin@logicsupply.com)

PRODUCTS/SERVICES: Logic Supply is an industrial and embedded computer provider offering end-to-end hardware solutions and OEM services to a wide range of industries including Data Acquisition, Digital Signage, Industrial Control, Surveillance & Security, Networking, Manufacturing, and Transportation. Its rugged, compact, energy-efficient, wide temperature Data Acquisition gateways offer the ideal solution for data collection applications in any environment. Logic Supply DAQ hardware is well suited for remote data collection in the oil and gas industry, medical imaging and DICOM applications, renewable energy monitoring, RFID inventory management, or test and measurement data handling. Its DAQ systems feature powerful Intel dual- and quad-core processing, as well as a wide array of 10 options, including USB 3.0/2.0, RS-232/422/485 COM, DVI-1, DVI-D, VGA, CFast slots, SATA connectors, GPIO pin headers, and Gigabit LAN.



Circuit Cellar prides itself on presenting readers with information about innovative companies, organizations, products, and services relating to embedded technologies. This space is where Circuit Cellar enables clients to present readers useful information, special deals, and more.

INDUSTRY & ENTERPRISE

Wide Temp. Panel PC

PPC-090T

- Vortex86DX2 933MHz Fanless
- Up to 1GB of Onboard Memory
- Low Power Consumption
- 2 RS232/422/485 serial ports
- 3 USB 2.0 Host Ports
- 10/100 BaseT Ethernet
- PS/2 KB port, Audio Out
- Compact Flash Slot / SD Slot / SATA
- 9 inch 1024 x 600 WSVGA TFT LCD
- Resistive Touch Screen
- DC-IN +12VDC / +24VDC
- IP65 Front Panel
- Wi-Fi (Optional)
- Giga Ethernet (Optional)
- Wide Temperature (Optional)

3.xx KERNEL

The PPC-090T comes ready to run with the Operating System installed on flash disk. Apply power and watch the Linux X-Windows desktop user interface appear on the vivid color LCD. Interact with the PPC-090T using the responsive integrated touchscreen. Everything works out of the box, allowing you to concentrate on your application rather than building and configuring device drivers. Just Write-It and Run-It... Pricing starts at \$495 for Qty 1.

<http://www.emacinc.com/sales/cc9>

Since 1985
OVER
29
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

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

RF Specialists

"Making your RF ideas into profitable products."

Affordable RF Modules

Introducing the LMX-ISM-242-SR and LMX-ISM-242-LR

MURS

Multi-Use-Radio-Service Band Products SHX1 - Long Range, High Power Multi Channel Radios

ZigBee Pro

OEM Modules, USB ZigBee Sticks, Home Automation, Range Finders, Lighting Applications

Industrial Bluetooth

OEM, Modules, Wireless Device Servers, RS-232 Long range options, low cost

Low Power, High Performance Semiconductors

Excellent Sensitivity, Very Efficient Power Amplifiers

RF Design Services

Prepared to work with your in-house engineers, or support your RF project from initial design to implementation.

Industrial • Military • Space • Medical • Smart Grid Metering • SCADA • Lighting Control

LE MOS

INTERNATIONAL

Tel: 1.866.345.3667
orders@lemosint.com
www.lemosint.com

Engineer a Force-Sensing System Sensor-to-User Data Acquisition Made Simple

FEATURES

Need a data acquisition system for gathering and sending analog data to a user in real time? This article details the process of building a sensor-to-user data acquisition system. Although this project is intended to monitor forces exerted by rowers, you can use it as a guide to build a system for measuring, displaying, and logging forces for a variety of applications.

By Steven Liu and Albert Ruskey (Canada)

You have probably stood on a digital bathroom scale where your first thought was, "That must be wrong!" Maybe your second thought was, "I wonder how

that works?" The seemingly simple ability to take a measurement, digitize it, and present it to a user can be quite a mystery to the uninitiated. We set out to not only



PHOTO 1

This is the finished force-sensing unit mounted and attached to rowing machine for demonstration. You can see the LED and Android displays.

understand how such a system works, but also to tailor it to our own application.

While studying at Camosun College in Canada, we were introduced to a physics professor with close ties to the Victoria rowing programs. She turned us on to the idea that systems for monitoring forces exerted by rowers—and then relating that data to things like acceleration—were extremely limited. The idea was to develop a system to monitor forces in real-time and provide feedback to rowers for efficiency analysis. To do so, we wanted to monitor the force at a rower's feet and relate it to the boat's acceleration. Additionally, we wanted track the rower's balance of motion to the right or to the left. Our end-to-end, sensor-to-user data acquisition system was built specifically with rowing in mind, but you could easily modify it to measure, display, and log forces for any number of applications (see **Photo 1**).

GAUGING THE SITUATION

Our first priority was to determine how we would detect the forces being applied by a rower. We went through several options for force sensing which included strain gauges, conductive foam, and piezoelectrics. After testing each option and weighing the benefits against the cost for each option, we determined that strain gauges would provide the best mix of linearity, repeatability, precision, and stability for our purposes. Camosun College generously donated several Vishay/Micro-Measurements CEA-13-240UZ-120 120- Ω strain gauges to our project. These gauges are thin-film resistive elements that change their resistance when put under tension or compression.

The physical mounting of strain gauges can be quite a challenge in and of itself. It is important that it properly bonds to the surface as any unexpected tension or compression can greatly skew the results. We used a thin steel bar sanded down with fine, 800-grit sandpaper and bonded with Gorilla cyanoacrylate glue. The mounting bar then had to be oriented in such a fashion that there

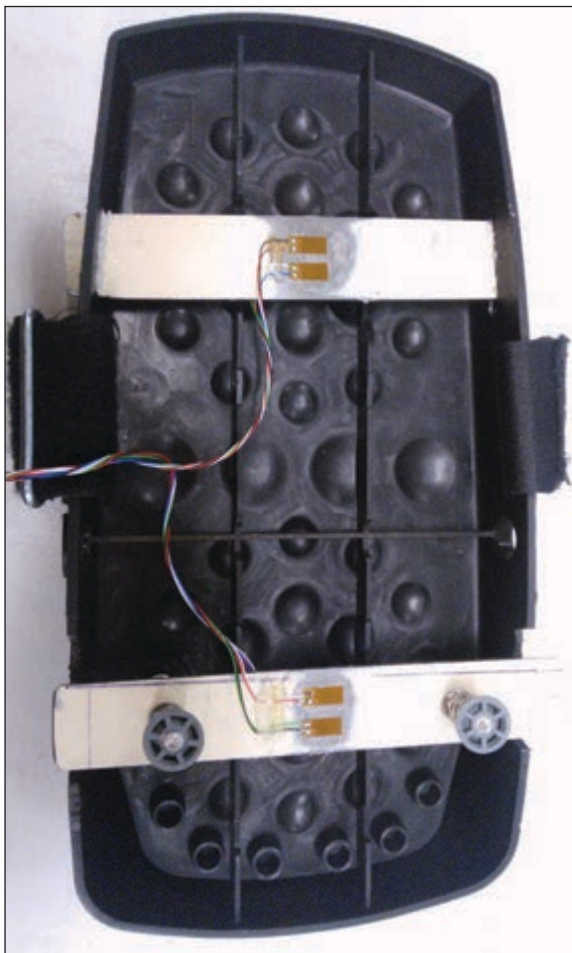


PHOTO 2

We mounted strain gauges on a rowing footplate. They were attached to steel bars with cyanoacrylate glue and soldered on 30-gauge wire.

would be flexion on the bar as the user rowed. **Photo 2** shows our final mounting strategy. We used some of the footpad's existing support architecture to apply even pressure across the strain gauges.

Since the elements are thin metal film that changes with compression and tension, they are particularly susceptible to temperature. To avoid any possible issues, especially because this system was designed with an outdoor rowing scull

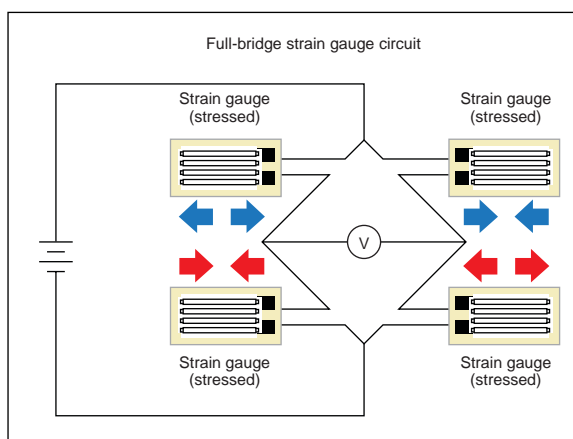
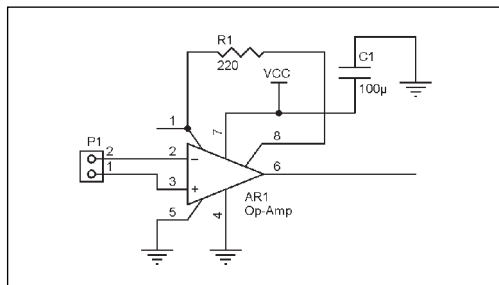


FIGURE 1

Here you see the bridge configuration for temperature compensation and linearity in data. As the strain gauges stretch and compress, their electrical resistance changes, which produces an output voltage that relates to an applied force.

FIGURE 2

The conditioning circuit is designed to amplify small signal inputs using an LT1167 instrumentation amplifier. This was used to amplify the minor voltage changes coming from our strain gauge circuits to the ARM board.



in mind, we had to create temperature compensation. Fortunately, this was accomplished without much difficulty by placing the strain gauges in a Wheatstone

bridge configuration (see **Figure 1**).

The very thin elements in the strain gauge do not pass a very large current. In order to produce a measurement that we could read on our STM32F4 board, we had to do some signal conditioning. This conditioning was done with an LT1167 op-amp circuit shown in (**Figure 2**). The amplifier uses a potentiometer as a tuning resistor in order to adjust the sensitivity of the strain gauge. This tuning allows for a wide range of forces to be measured.

ACQUIRING ANSWERS

With the strain gauges mounted in place,

```
void ADC_Configuration(void)
{
    ADC_CommonInitTypeDef ADC_CommonInitStructure;
    ADC_InitTypeDef ADC_InitStructure;

    /* ADC Common Init */
    ADC_CommonInitStructure.ADC_Mode = ADC_Mode_Independent;
    ADC_CommonInitStructure.ADC_Prescaler = ADC_Prescaler_Div2;
    ADC_CommonInitStructure.ADC_DMAAccessMode = ADC_DMAAccessMode_Disabled;
    ADC_CommonInitStructure.ADC_TwoSamplingDelay = ADC_TwoSamplingDelay_5Cycles;
    ADC_CommonInit(&ADC_CommonInitStructure);

    ADC_InitStructure.ADC_Resolution = ADC_Resolution_12b;
    ADC_InitStructure.ADC_ScanConvMode = ENABLE;
    ADC_InitStructure.ADC_ContinuousConvMode = ENABLE;
    ADC_InitStructure.ADC_ExternalTrigConvEdge = ADC_ExternalTrigConvEdge_None;
    //ADC_InitStructure.ADC_ExternalTrigConv = 0;
    ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
    ADC_InitStructure.ADC_NbrOfConversion = 4;
    ADC_Init(ADC1, &ADC_InitStructure);

    /* ADC1 regular channel configurations*/
    ADC_RegularChannelConfig(ADC1, ADC_Channel_11, 1, ADC_SampleTime_480Cycles);
    ADC_RegularChannelConfig(ADC1, ADC_Channel_12, 2, ADC_SampleTime_480Cycles);
    ADC_RegularChannelConfig(ADC1, ADC_Channel_14, 3, ADC_SampleTime_480Cycles);
    ADC_RegularChannelConfig(ADC1, ADC_Channel_15, 4, ADC_SampleTime_480Cycles);

    ADC_DMARequestAfterLastTransferCmd(ADC1, ENABLE);

    /* Enable ADC1 DMA */
    ADC_DMACmd(ADC1, ENABLE);

    /* Enable ADC1 */
    ADC_Cmd(ADC1, ENABLE);
}
```

LISTING 1

This setup is for the ARM board's built-in ADCs. It was created for continuous 12-bit conversions on data acquired from four different GPIO pins.



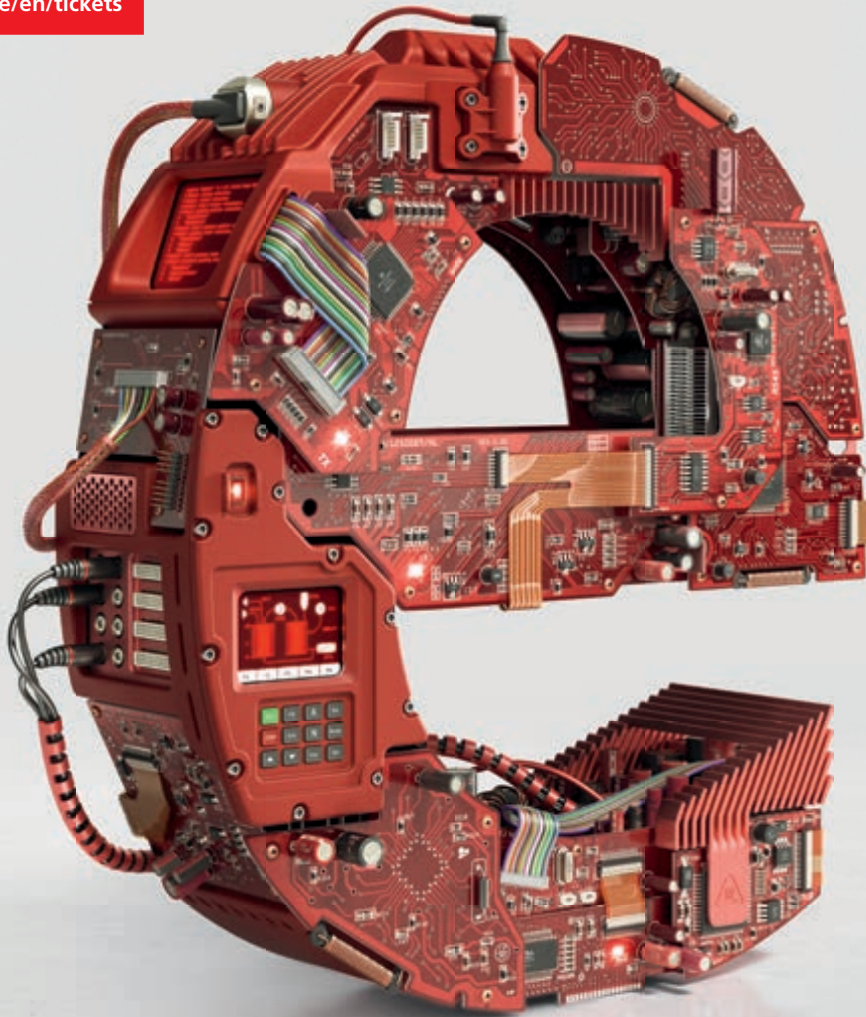
Messe München
International

Connecting Global Competence

Welcome to Planet e.

The entire embedded universe at a single location!

Tickets & Registration
www.electronica.de/en/tickets



26th International Trade Fair for
Electronic Components, Systems
and Applications
Messe München
November 11–14, 2014
www.electronica.de

50 years
electronica



electronica 2014
inside tomorrow

```

static void DMA_Configuration(void)
{
    DMA_InitTypeDef DMA_InitStructure;

    DMA_InitStructure.DMA_Channel = DMA_Channel_0;
    DMA_InitStructure.DMA_Memory0BaseAddr = (uint32_t)&ADCConvertedValues[0];
    DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)&ADC1->DR;
    DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralToMemory;
    DMA_InitStructure.DMA_BufferSize = 4; // Count of 16-bit words
    DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
    DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
    DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
    DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
    DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
    DMA_InitStructure.DMA_Priority = DMA_Priority_High;
    DMA_InitStructure.DMA_FIFOMode = DMA_FIFOMode_Enable;
    DMA_InitStructure.DMA_FIFOThreshold = DMA_FIFOThreshold_HalfFull;
    DMA_InitStructure.DMA_MemoryBurst = DMA_MemoryBurst_Single;
    DMA_InitStructure.DMA_PeripheralBurst = DMA_PeripheralBurst_Single;
    DMA_Init(DMA2_Stream0, &DMA_InitStructure);

    /* DMA2_Stream0 enable */
    DMA_Cmd(DMA2_Stream0, ENABLE);
}

```

LISTING 2

This setup is for the ARM board's DMA controller. It was created with a base address in memory for storing ADC conversion values. The DMA controller enabled us to perform multiple ADC conversions.

we were finally ready to start reading some values. We knew that we would need a microcontroller to collect and process the data before passing it out for display. To accomplish this, we decided to use a board that we had some experience with, the STMicroelectronics STM32F4 Discovery Board (see **Figure 3**). It includes all of the interfaces we needed: two ADCs with multiple channels, two USARTs, and multiple general-purpose input-output (GPIO) ports. This would enable us to collect data from the toe and heel of each foot, and then transmit out to a Microchip Technology PIC18F1330 microcontroller and a Bluetooth module.

To configure the ADCs, we had to take a number of steps. The ADCs can convert values from either the GPIO pins or from on-board peripherals. We were interested only in acquiring data on four pins from our strain gauge circuits; we had no interest in the peripherals. This meant that the first step was to configure the GPIO pins for analog use.

```

GPIO_InitStructure.GPIO_Pin =
    GPIO_Pin_1 | GPIO_Pin_2 |
    GPIO_Pin_4 | GPIO_Pin_5;

```

```

GPIO_InitStructure.GPIO_Mode =
    GPIO_Mode_AN;

```

Special care had to be taken ensure the correct pins were selected for use. We used PC1, 2, 4, 5, but it is easy to mistakenly use PC3, which is not an available ADC pin. Once we had the GPIO code setup, we needed to enable the ADC itself to start reading values. Most of the ADC configuration is similar to the setup for a single ADC, but with a few important differences. We knew we were going to set up the Direct Memory Access (DMA) to handle the multiple conversions and we had to set up the ADC with that in mind. We had to set the modes to enable multiple conversions and then set the number of conversions we wanted to perform on each servicing of the ADC. Lines 146 to 149 in **Listing 1** point the ADC at the correct GPIO pins and also determine the order in which the conversions will be made. After all this, we almost had usable data!

The final step for our ADC configuration was handling where it would be stored in memory. That's where the DMA came into play. The DMA enabled us to assign

The new EAGLE has landed!

Version 7

now
available



For more information, visit www.cadsoftusa.com

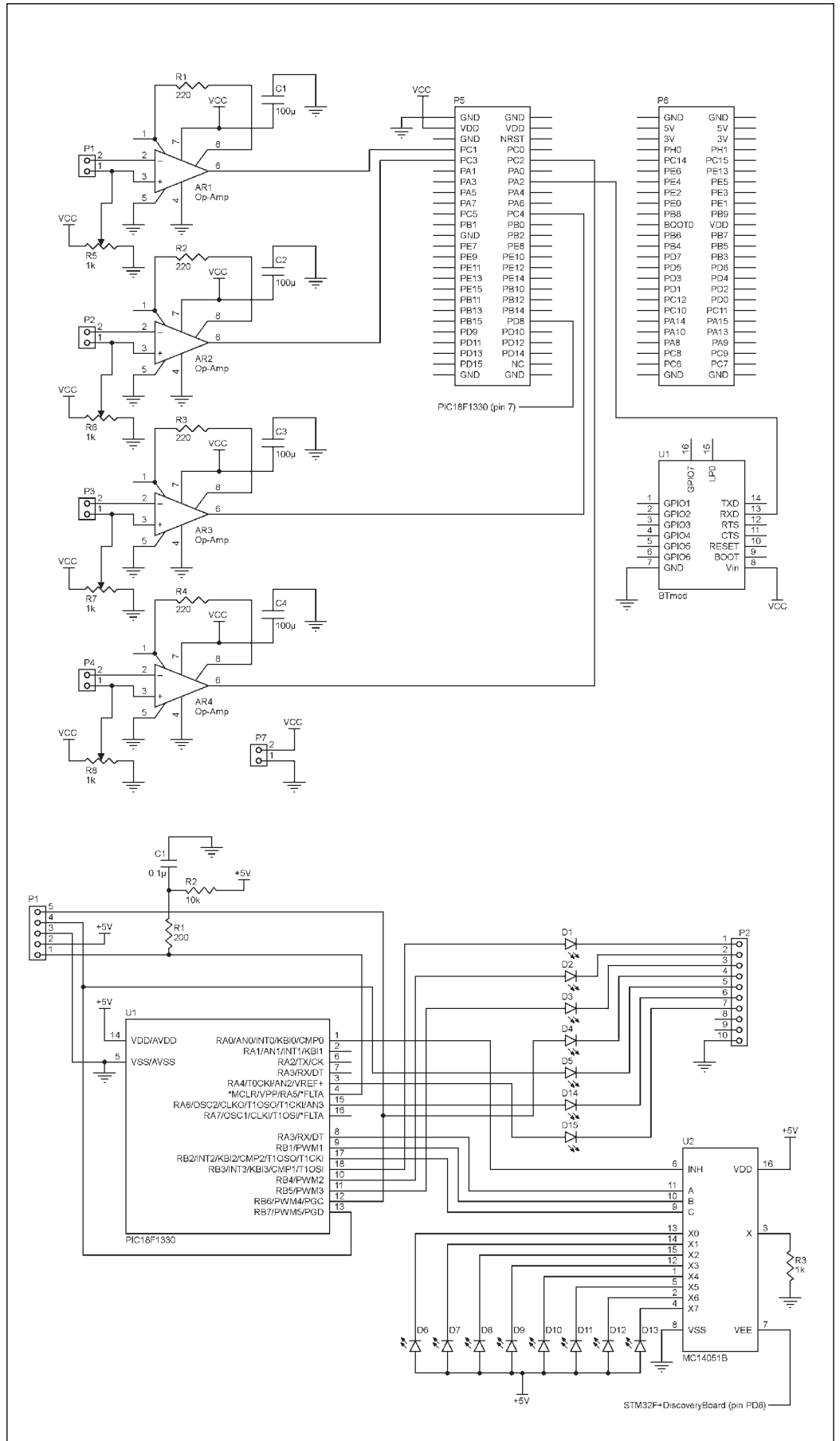


FIGURE 3

This is the complete project featuring the ARM and PIC circuits. The ARM board and conditioning circuits are located at the top. The PIC and LEDs are displayed at the bottom.

a location in memory to store all our converted values. Finally, we had our data in a usable form. But it still seemed to appear as a value between 0 and 4096 (see **Listing 2**). This value appears because of how the ADC works. In our ADC configuration, we chose a 12-bit resolution so our range of values was from 0 to 2^{12} . To interpret this as a voltage, we took our maximum known voltage of 5 V and divided by 2^{12} , which gave us our least significant bit and a way to relate back to our analog unit.

TALK TO US!

After we acquired our data, we needed to communicate with the world. We provided two ways for a user to see feedback from their data. We created an Android application that accepted information via Bluetooth, and we also created a 15-LED array controlled by a PIC18F1330 microcontroller that received

information over a serial connection.

To configure the USART on the STM32F4 Discovery Board, we again had to set up the GPIO pin that we wanted to use, but this time in Alternate Function (AF) mode.

```
GPIO_InitStructure.GPIO_Mode =
    GPIO_Mode_AF;
GPIO_PinAFConfig(GPIOD,
    GPIO_PinSource8, GPIO_AF_
    USART3);
```

Next was to setup the USART itself. We used standard settings for an 8-bit serial communication with our data rate selected depending on which device we were communicating with. To transmit out via Bluetooth, we used an STMicroelectronics SPBT2632C2A Bluetooth module. We were able to port our serial communication directly out over Bluetooth. The module comes preconfigured to accept serial in at 115,200 bps (data rate), so configuring our

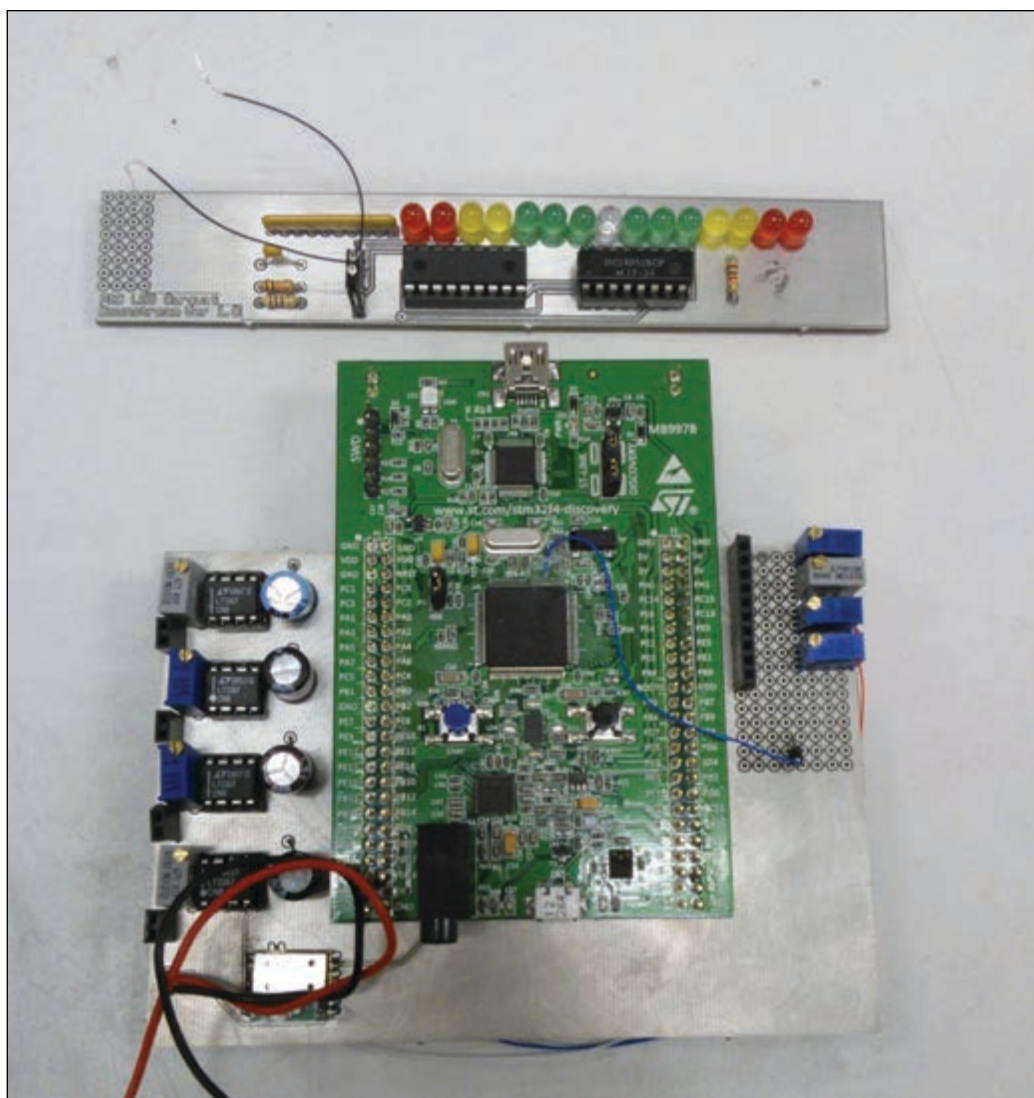


PHOTO 3

Here you see ARM and PIC PCB boards with op-amp conditioning circuits and MUX included. The overall system on separate ARM-based (bottom) and PIC-based (top) PCB boards. We used Altium Designer to create the schematics and PCB layouts.

The end goal for our project was to see our data. There are numerous options for communicating with the user, such as LCDs, buzzers, and LEDs ... We implemented an array of 15 LEDs that would move back and forth in order to provide additional information to the rower ...

USART to the correct data rate and then passing data out was dead simple. Creating Android applications is a broad topic that's outside the scope of this article. We'll leave it to you to investigate it further.

PIC US

Much simpler than creating an Android application from scratch was our second USART configured to be used for communicating with our PIC18F1330 microcontroller. The PIC18F1330 is limited to a data rate of 9,600 bps at the given internal oscillator speed. This limitation meant that, unfortunately, we were not able to drive both the PIC18F1330 and the Bluetooth units from a single pin on the ARM.

When creating these chip-to-chip connections, we also had to pay special attention to our timing. They say that in comedy timing is everything, but we think

it's even more important in electronics! Due to the difference in processor speed, 8 MHz for our PIC18F1330 compared to 180 MHz for the ARM, we have to be very aware of overflow errors when communicating between our chips. Overflow occurs when new data arrives before the receive register has had a chance to clear. To get around this, we introduced a delay between each USART communication. A 10-ms delay was more than enough to slow our system down to avoid any chance of overflow, while still giving us a high enough sample rate for our data to be useful.

SHOW US THE WORLD

The end goal for our project was to see our data. There are numerous options for communicating with the user, such as LCDs, buzzers, and LEDs. We needed something that could provide instant feedback without requiring the user to focus too long. To this end, we implemented an array of 15 LEDs that would move back and forth in order to provide additional information to the rower about their performance so that they could correct their left-right balance (see **Photo 3**). The LEDs would act as a bubble level and would be color coded to indicate just how in or out of sync the rower's feet were.

The PIC18F1330 configuration was not too difficult to figure out using the built-in USART library. The PIC18F1330 had a maximum internal oscillator speed of 8 MHz, and as a result a maximum data rate of 9,600 bps. The setup for the PIC18F1330 USART is:

```
OpenUSART(USART_TX_INT_OFF
& USART_RX_INT_OFF
& USART_ASYNCH_MODE
& USART_EIGHT_BIT
& USART_CONT_RX
& USART_BRGH_HIGH, 51);
```

Once we had the data on the PIC18F1330, it was time to do something with it. Processing the raw data was done on the ARM board first to convert the value into a smaller number. This processing was done first on the ARM due to the serial communication only being 8 bit and our data being a 12-bit value. The PIC18F1330 simply receives a value that is easily parsed to between 0 and 14, which then runs a function to switch on an LED. This of course creates some complications as the PIC18F1330 does not come equipped with enough I/O pins to handle 15 LEDs!

The final piece of this puzzle was to add in a multiplexer to add to our usable I/O

PROJECT FILES



circuitcellar.com/ccmaterials

To download the code, go to circuitcellar.com/ccmaterials.

SOURCES

Altium Designer software
Altium | www.altium.com

PIC18F1330 Microcontroller
Microchip Technology, Inc. | www.microchip.com

SPBT2632C2A Bluetooth Module and STM32F4 Discovery Board
STMicroelectronics | www.st.com

CEA-13-240UZ-120 120-Ω Strain gauge
Vishay/Micro-Measurements | www.vishaypg.com/micro-measurements/

ABOUT THE AUTHORS

Albert Ruskey (aruskey@shaw.ca) studied at Camosun College in Victoria, BC, where he earned Electronics and Computer Engineering Technology diploma. He is pursuing for a degree in Computer Engineering, with a focus on embedded systems, at the University of British Columbia. Previously, Albert worked as a marine technologist for Ocean Networks Canada and an electronics technician for the Department of National Defense.

Steven Liu (stevenliu20@hotmail.com) earned an Electronics and Computer Engineering Technology diploma from Camosun College in Victoria, BC. Steven is pursuing an Electrical Engineering degree at the University of Victoria and plans to specialize in mechatronics. He is also CompTIA A+ certified.

pins. Using an 8-to-1 multiplexer as a low-side switch, we were able to get four additional I/O pins for controlling LEDs. Then all that was left was to set up the routine that would turn on and off each LED. Fortunately, controlling digital I/O on the PIC18F1330 requires less work than on the STM32F4Discovery. Simply by setting the individual bits on each port, you are able to assign a logic high, which is enough to drive an LED.

The last challenge was to ensure that only the specified LED was on. To that end, every time we called the LED function, we first turned off all of the LEDs and then lit the appropriate one.


Now if you are thinking that you could drive LEDs from the STM32F4 board's GPIO pins and get rid of the PIC18F1330 entirely, you're absolutely correct! However, for our application, we wanted a solution that would give us the flexibility to add additional features later, such as making the display wireless or perhaps adding additional gauges to the STM32F4 board. We felt it would be advantageous to be able to make modifications while only creating a minimum amount of extra work initially.

With the PIC18F1330 circuit in place, we accomplished an end-to-end, sensor-to-user system. As an added bonus, due to the way that the data is acquired individually and processed, it would be easy to change or add functionality. For example, changing the LED display to display the overall force being applied would simply require changing a small amount of math in the ARM code. And that actually would be simpler than what's required for the left-right balance function.

TURN IT ON

The only thing left to do is turn it on and push off. Turning it on might be a problem though because although electricity often feels like magic, it does in fact require a source of power to work. For our application in a rowing scull to be effective, we could not increase the weight of the scull by more than about 1 lb. This ruled out things like bulky battery packs and multiple power sources. With that in mind, all of our components—the ARM, op-amps, PIC18F1330, and the MUX—were chosen to be able to run off of 5 V. To keep the weight down further, we decided to use an LM7805 linear voltage regulator to regulate 9 V, which provides the greatest voltage for the lightest weight of any standard alkaline battery available.

Hopefully, this article has as provided some insight into what is required to create a data acquisition system. Building a system to measure world around you can be both easy and inexpensive. If we ignore the strain gauges, which were donated, the entire cost of the project was close to \$50!

We created our setup with rowing in mind, but you'll find it an excellent starting point for any data acquisition system in which the end goal is to get analog data back to a user in real time. It's easy. Simply remember your acronyms: ARM to GPIO to ADC to DMA to USART to PIC to MUX to LED! 



BECOME
a member of
Circuit Cellar!

Project articles and design applications
.....
Embedded industry news and announcements
.....
12 issues annually
.....
CC.Post e-newsletter

 circuit cellar

MCU-BASED COLOR DATA ACQUISITION

circuitcellar.com/
subscriptions

Position Control

Build a Microcontroller-Based Stabilization Platform

FEATURES

You can engineer a way to taking better videos. This handheld self-stabilizing camera system consists of a gyroscope and an accelerometer mounted on a platform. When sensors detect a change in the platform's position, a microcontroller controls the motors to counteract the disturbance.

By Evan Chen, Zequn Huang, Geo Xu (US)

As avid outdoor enthusiasts, we often find ourselves taking action shots of our adventures, only to discover a very shaky and nauseating video of our endeavors at the end of the day. So, when we had to decide

on a project for our microcontroller design course (ECE 4760) at Cornell University, we chose to build a handheld camera stabilizing platform. While idea of a handheld camera stabilizer is not new, this project proved to be a challenge when it came to both the software and hardware. With a budget cap of \$100, we had to make design-related tradeoffs to keep costs down yet retain overall performance. In this article, we'll present our final design. We'll detail the specific hardware, software, and mathematics we used to complete the project.

HIGH-LEVEL OVERVIEW

As you can see in **Photo 1** at the core of our design are two servo motors, a main board containing an Atmel ATmega1284p microcontroller, a three-axis accelerometer, and a three-axis gyroscope. The microcontroller board and servo motor are attached to an L-shaped wooden frame that enables you to hold the apparatus in different ways. A battery pack is on the other side of the wooden frame. The servo motor attached to the main L-shaped wooden frame controls the platform's roll. Its shaft is directly screwed into a wooden support frame, which supports the second servo motor that controls the pitch of the platform. The shaft of the second servo motor is connected to a metal bracket supporting the actual platform that contains the test camera, three-axis accelerometer, and three-axis gyroscope.

When the device is switched on, the platform is initialized to the neutral position (level to the ground). Using specific algorithms

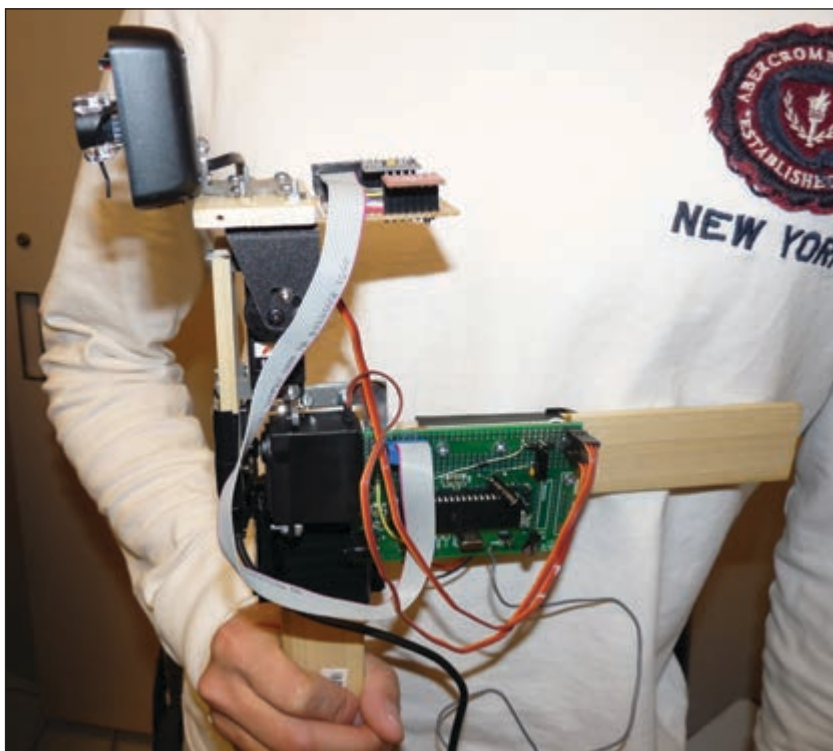
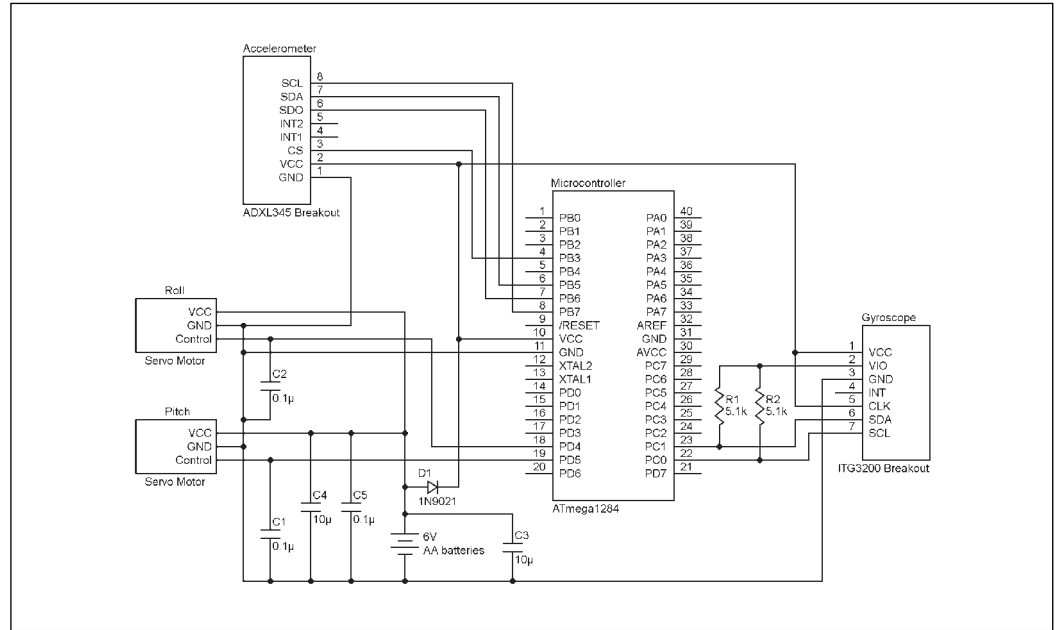


PHOTO 1

This is the basic setup for our self-stabilizing platform. The main board contains an ATmega1284p microcontroller with an AA battery pack mounted behind the handle. A servo motor controls the roll of the platform. The servo motor controls the pitch of platform. Also included are a triple-axis gyroscope, a triple-axis accelerometer, and a webcam attached to the platform for testing.

FIGURE 3

Take a look at the complete circuit. The accelerometer is connected to the SPI input pins. The gyroscope is connected to the microcontroller's I2C input pins.



lines to the servo motor. As a result, the device now functions smoothly without ever having the microcontroller reset unintentionally.

SOFTWARE OVERVIEW

Our microcontroller program is written using the multitasking kernel called TinyRealTime (TRT), which was written by Dan Henriksson and Anton Cervin. This enables us to achieve the concurrent execution of several necessary tasks for our design.

TRT Task 1 interacts with the PC keyboard to take user input. It then set up proportional integral derivative (PID) controller parameters to control the servo motors.

TRT Task 2 runs the PID control loop about 50 times per second using x-axis and y-axis tilt angles measured from another task. It then assigns the PID calculation result to the PWM output to drive the servo motors.

TRT Task 3 acquires the data from the gyroscope and the accelerometer. It then calculates the corresponding x-axis and y-axis tilt angles at about 50 Hz.

To implement TRT Task 1, we used predefined function `fscanf()` to get the input string from the UART port, which is the PC keyboard in this case. Also, two TRT semaphores, `SEM_RX_ISR_SIGNAL` and `SEM_STRING_DONE`, are required for serial communication. The former is used in UART receiving interrupt service routine (ISR). The latter is for detecting whether or not the user typed <enter> key. After getting a valid input, we assign the input value to corresponding variable such as PID parameters.

While this task is not necessary for the actual function of the device, this enabled us to fine-tune the default PID parameters, and

it allows the user to fine tune how fast and steady the user wants the device to respond to perturbations. For example, the proportional term and derivative term can be increased to make the design respond faster, although this results in less stability as the motors may overshoot the target position.

PID & MOTOR CONTROL

To implement TRT Task 2, which drives the servo motors, we tried two approaches to control the servo motors. The first was direct angle control and other was PID control, which we ultimately settled upon. The direct angle control method simply involved changing the PWM output to make the servo motor rotate a certain angle that was equal in magnitude to the current tilt angle and but opposite in direction. This, however, provided less user control than using a PID.

Our program's PID control algorithm is a standard algorithm provided by Cornell Professor Bruce Land (ECE 4760: Laboratory 4—"Tachometer and Motor Controller," 2013). The motor is modeled as a second-order system when it's powered and a first-order system when it's coasting. In this algorithm, we use the calculated pitch and roll angles as the feedback to the PID. The algorithm's core is the following line of code in the program:

```
m_input = pid_p * c_error + pid_i *
pid_integral / 1000 + pid_d *
(c_error - c_error_prev);
```

`Pid_p`, `pid_i`, and `pid_d` are the PID parameters for the proportional term, integral term, and derivative term. `C_error`

is the difference between current tilt angle and desired angle, which is level. `c_error_prev` is the previous tilt angle from the last PID cycle. `pid_integral` is the integral of the previous error terms.

To achieve smoother leveling of the platform, we made some modifications to the original algorithm. First, we put a limit on the integral term's maximum value to prevent instabilities when the integral changes sign. We also scaled the integral term by 1,000-fold smaller to make the `pid_i` parameter easier to input. In addition, we introduced an additional variable, `SCALE`, to scale the maximum value of the total PID calculation result `m_input` to control the change rate of the control PWM signal. Finally, the calculation result is mapped to the PWM output control registers `OCR1A` and `OCR1B`, which output to the microcontroller's pin `D4` and `D5`, respectively.

To determine the exact value for the PWM output, we had to view the specs for our particular servo motors, FEETECH model FS5106B servo motors purchased from SparkFun. In particular, the circuitry of the servo expects a constant 50-Hz PWM signal (period of 20 ms). The actual input we send the servo is the width of the high signal duration in the whole period. As shown in **Figure 4**, the width of the high signal determines the position of the servo correspondingly.

Since the proper frequency for the PWM signal is 50 Hz, we could not use the fast PWM mode, which generates a very high-frequency signal. So, we decided to use the Phase and Frequency Correct mode of `Timer1`. Only `Timer1` and `Timer3` support the Phase and Frequency Correct mode and have 16-bit timers. Since `Timer3` outputs its PWM to Ports `B3` and `B4`, which the accelerometer needed for the SPI protocol, we use `Timer1` for the Phase and Frequency Correct mode. `Timer3`, in the meantime, is used for the TRT kernel.

In Phase and Frequency Correct mode, we first set the Prescaler value, `N`, to 8, and we then calculated our `ICR1` (Input Capture Register 1 = TOP) register value from the equation below.

$$f_{\text{PWM}} = \frac{f_{\text{CLK}}}{2 \times N \times \text{TOP}}$$

After we set the `ICR1` (TOP) value to 20000, the PWM frequency is 50 Hz. The `OCR1A` register value determines the width of the high signal. We can vary `OCR1A` from 1000 to 2000, which should correspond to 10- and 20-ms signal widths, respectively. For our particular servo motor, we use more extreme values from 770 and 2300 in order to get the

maximum 180° rotation.

READING DATA

For TRT Task 3, we obtained data from the accelerometer and gyroscope. We used is an Analog Devices ADXL345 low-power, three-axis accelerometer. We used the standard SPI protocol to communicate between the accelerometer and the microcontroller. To read and write to the ADXL345 accelerometer's registers, we utilized and modified code provided by SparkFun. To convert the raw data to angular acceleration, we use the following equation:

\$51 For 3 PCBs
FREE Layout Software!
FREE Schematic Software!

- 01 DOWNLOAD our free CAD software
- 02 DESIGN your two or four layer PC board
- 03 SEND us your design with just a click
- 04 RECEIVE top quality boards in just days

expresspcb.com

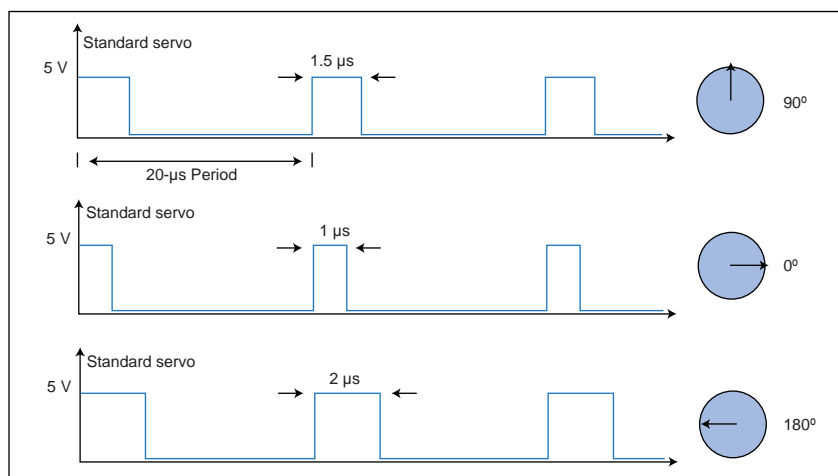


FIGURE 4

This is a servo motor PWM timing diagram and the corresponding angle of shaft position. The period is 20 ms and pulses vary from around 1 to 2 ms.

$$\text{Acc} = \text{rawData} \times \frac{8}{2^{10}} \frac{\text{LSBs}}{\text{g}}$$

In addition, to determine the pitch and roll of the platform, we used the following equations:

$$\theta_x = \tan^{-1} \left(\frac{A_{x,\text{OUT}}}{\sqrt{A_{y,\text{OUT}}^2 + A_{z,\text{OUT}}^2}} \right)$$

$$\theta_y = \tan^{-1} \left(\frac{A_{y,\text{OUT}}}{\sqrt{A_{x,\text{OUT}}^2 + A_{z,\text{OUT}}^2}} \right)$$

While the accelerometer alone can determine these angles without the aid of a gyroscope, it is rather noisy when used to measure the gravitational acceleration, especially when the platform moves back and forth. So, we also decided to include a gyroscope, which, if used by itself, will produce readings that drift over time.

We used an InvenSense ITG-3200 triple-axis gyroscope. We used the standard

I²C protocol to communicate between the gyroscope and microcontroller. Once again, to read and write to registers of the ITG-3200 gyroscope, we utilized and modified code provided by SparkFun.

The gyroscope's output is degrees per second. To translate this into degrees, we can integrate the gyroscope data over time provided the frequency is high enough. So, we used the following equation to convert the raw data to angular acceleration:

$$\text{AngularAcc} = \text{rawData} \times \frac{14.375 \text{ LSBs per } ^\circ}{\text{s}}$$

To further reduce noise in our measurements, we also sample multiple readings from the gyroscope and the accelerometer. However, with our initial tests, we were sampling too many readings, which negatively impacted the speed of our device. In particular, we were averaging eight samples of the gyroscope measurements and eight samples of the accelerometer. With this amount of integration, we observed that the motors were visibly adjusting at discrete intervals, resulting in a very jittery response. The motors seemed to adjust about twice a second. This occurred since our TRT Task 3 was unable to meet the deadline that we set. In particular, to compute the angles from the gyroscope, the microcontroller runs many computations on long integers, so we reduced the number of samples to two for the gyroscope and four for the accelerometer. With these new values, functions were able to be executed much more frequently, resulting in smoother transitions.

In order to combine the data from the gyroscope and the accelerometer, we looked into two different methods—using a Kalman filter and using a complementary filter. The Kalman filter operates by producing a statistically optimal estimate of the system state based upon the measurements. To do this it needs to know the noise of the input to the filter, which is called the measurement noise, and it needs the noise of the system itself, which is called the process noise. To do this, the noise has to be Gaussian distributed and have a mean of zero. For us, most of the random noise in our measurements has this characteristic.

The algorithm for the Kalman filter is rather complicated, but the code for the Kalman filter function can be broken down into the following four steps in a loop:

- Update estimation error covariance and project the error covariance ahead.
- Calculate discrete Kalman filter measurement update equations and

ABOUT THE AUTHORS

Evan Chen (ec488@cornell.edu) studies Electrical and Computer Engineering at Cornell University. Most of his work experience has been focused on digital electronics, but he retains a broad interest in a variety of engineering-related areas.

Zequan Huang (huangzequan.china@gmail.com) studied Electrical Engineering at Pennsylvania State University and is currently pursuing a Master's degree at Cornell University. Although Zequan mainly focuses on embedded systems design, he has completed many digital and analog circuit designs.

Jianglu (Georald) Xu (geoxu@gmail.com) is pursuing a Master's degree at Cornell University. He is currently focused on embedded applications and computer architecture. He is also working on a start-up project focused on a Wi-Fi remote controller and is seeking cooperation.



FOR FOUR DAYS, WHAT'S NEXT IS HERE NOW.

The 2015 International CES®. With the who's who of innovation, business and technology, it's the one show on Earth that promises the future. And delivers it in ways you never imagined. Register today at CESweb.org.

2015 International
CES
THE GLOBAL STAGE FOR INNOVATION

PRODUCED BY
 CEA

JAN. 6-9, 2015
LAS VEGAS, NEVADA
CESWEB.ORG
#CES2015

calculate Kalman gain.

- Calculate angle and bias and update estimate with measurement in k state.
- Calculate and update estimation error covariance.

The original Kalman filter code we used was from SparkFun. We modified it to suit our microcontroller and sensors.

COMPLEMENTARY FILTER

The complementary filter provides another way to combine different sensor data and it is much easier and more intuitive. In fact, this filter manages both high-pass and low-pass filters simultaneously. The low pass filter filters high frequency signals (such as the accelerometer in the case of vibration) and low pass filters that filter low frequency signals (such as the drift of the gyroscope). In its simplest form, the filter looks as follows:

$$\begin{aligned} \text{filterAngle} &= 0.98 \times (\text{previousAngle} + \text{gyrData} \times \text{dt}) \\ &+ 0.002 \times \text{accData} \end{aligned}$$

The complementary filter we used in our program is a second order filter which is more complicated. The algorithm looks like PID and is given as follows:

```
errorTerm = (newAngle - previousAngle) * k * k
integralTerm = integralTerm + errorTerm * dt
PIDTerm = integralTerm + (newAngle - previousAngle) * 2 * k + newRate
filterAngle = PIDTerm * dt + previousAngle
```

k is the time constant. *dt* is the derivative constant. *newAngle* is the tilt angle calculated from the accelerometer. *newRate* is the angular acceleration calculated from the gyroscope. In our code, we made several modifications to this algorithm to make the tilt recovery process faster. We introduced a variable to control the proportion of *newRate* in *PIDTerm*. We also scaled *errorTerm* in *integralTerm* to make it proportional to the *newRate*.

We currently use the complementary filter to calculate the tilt angles because the result from the complementary filter was very close to the one calculated by the Kalman filter. Also, the complementary filter is more microprocessor-friendly because it requires a small number of floating-point operations.

PERFORMANCE

With our default PID settings, we see relatively fast and smooth transitions. The motors can transition from -90° to 90° within 1.2 s. We can decrease the transition time by increasing the proportion term and the derivative term. However, this results in rougher transitions, and the platform will sometimes oscillate as a result of overshooting the target position. In any case, it is possible to reduce the response time, and the theoretical minimum response time seems to be limited by the slew rate of the motors.

While the platform remains in a relatively neutral position, the platform occasionally oscillates back and forth. This error seems to be caused by multiple sources. On the

PROJECT FILES



circuitcellar.com/ccmaterials

LetsMakeRobots.com, "Kalman filter vs Complementary Filter," 2011, <http://letsmakerobots.com/node/29121>.

SparkFun, "Triple Axis Accelerometer Breakout - ADXL345," 2013, <https://www.sparkfun.com/products/9836>.

———, "Triple-Axis Digital-Output Gyro-scope—ITG-3200," <https://www.sparkfun.com/products/9793>.

RESOURCES

S. Colton, "The Balance Filter," 2007, <http://bit.ly/1jLr9OZ>.

B. Land, "A Preemptive Kernel for Atmel Mega1284 Micro-controllers," Cornell University, 2013, <http://people.ece.cornell.edu/land/courses/ece4760/TinyRealTime/index.html>

———, "ECE4760: Laboratory 4—Tachometer and Motor Controller," Cornell University, 2013, <http://people.ece.cornell.edu/land/courses/ece4760/labs/f2013/lab4.html>.

SOURCES

ADXL345 Accelerometer
Analog Devices, Inc. | www.analog.com

ATmega1284p Microcontroller
Atmel Corp. | www.atmel.com

ITG-3200 Triple-Axis Digital-Output Gyroscope Servo—Generic High Torque (Standard Size)
SparkFun (distributor) | <https://www.sparkfun.com/products/9793>


software side, since we had to reduce the number of samples taken from the gyroscope and accelerometer, accuracy suffered where we saw variations in angles of about a degree or two. In addition, on the hardware side, the bearings on the motor that controls the roll of the platform became a bit loose. In particular, its shaft is directly connected to a wooden support that holds the weight of the second motor and the entire platform. After many trials and tests, the rather heavy loads that we had placed on it have probably caused the bearings to loosen. Combined, these sources of error make the platform vary at around $\pm 2.5^\circ$ from the neutral position.

When tested with a webcam, the platform kept the video oriented in the landscape position as we tilted the apparatus to the extremes of $\pm 90^\circ$ in both roll and pitch. However, when the device was held in place without any perturbations, due to the sources of error mentioned previously, the platform would continue to adjust its position up to $\pm 2.5^\circ$ from the neutral position.

Overall, the self-stabilizing platform works as expected. However, while it is able to keep a camera oriented in a relatively neutral

position, it experiences slight vibrations. So, while it can be suited to taking steadier action video shots, it is not as suited to taking still pictures, and many improvements can be made, especially on the hardware side.

For example, to reduce the load on the shaft of the motor controlling the roll, we can distribute the load by adding another support panel having a second point of contact on the opposite end of the motor. In addition, instead of mounting the gyroscope and accelerometer on top of two separate pin socket connectors, we can try placing both together flush against the platform. This may help increase accuracy in the angle calculations. Of course, if budget was not an issue, motors with better torque and slew rates can be used instead. A larger and more refined platform can also be attached so that it can hold a larger variety of cameras and objects.

Authors' Note: Full code is available on Circuit Cellar's FTP site. The code and further details are also available at <http://bit.ly/1mLH4MX>. A video demonstration of our camera stabilizing platform is also available at <http://bit.ly/1qRYtZJ>. 



AP CIRCUITS
PCB Fabrication Since 1984

As low as...

\$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com



The Convenient All-in-One Solution for Custom-Designed Front Panels & Enclosures



You design it
to your specifications using
our FREE CAD software,
Front Panel Designer



We machine it
and ship to you a
professionally finished product,
no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

**FRONT PANEL
EXPRESS**

FrontPanelExpress.com
1(800)FPE-9060

DIY RGB Game Design

An Upgrade for a Classic Platform

FEATURES

Since the 1960s, engineers and electronics enthusiasts have been building electronic versions of the Tic-tac-toe game. This article details a recent iteration of one engineer's electronic system. It's a microcontroller-based design that drives matrix LED displays.

By Mitch Matteau (US)

I've always been fascinated by the simplicity of Tic-tac-toe (or "Noughts and Crosses"). About 40 years ago, *Popular Science* published the article, "Electronic Tick-Tack-Toe in a Cigarette Box," which detailed an interesting electronic version of the classic game. Since then, I've designed a few variations of that electronic system. In this article, I'll present

my most recent version. I'll explain how I brought the game to life with a modern upgrade—a red-green-blue (RGB) 8 × 8 LED matrix display.

My original interpretation of the *Popular Science* game was based on a 40-pin DIP Motorola 68HC705C8A microcontroller and bipolar LEDs. The green wires shown in **Photo 1** go to off-board push-button switches. The controller was atypically mounted on the solder side of the PCB. And you'll probably notice that the single-sided PCB was drawn out freehand and manually etched. Ah, the good old days! The software counter-move algorithm used in that version was based on the one described in the original *Popular Science* article. That same algorithm was ported to the newer microcontroller for the current implementation, which saved a bit of coding time. I'll explain that algorithm later on.

SYSTEM OVERVIEW

The latest incarnation of the game is shown in **Figure 1**. The heart of the design is a Freescale Semiconductor HC908JB16 (U1) microcontroller, which I chose for a few reasons. One, I've been coding Freescale processors in assembler for years, and I'm comfortable with Freescale devices and software tools. Two, this particular variant includes a simple USB interface for in-circuit programming (ICP). And three, there were enough port pins available to connect to the input/output peripherals, requiring only a small number of outboard devices.

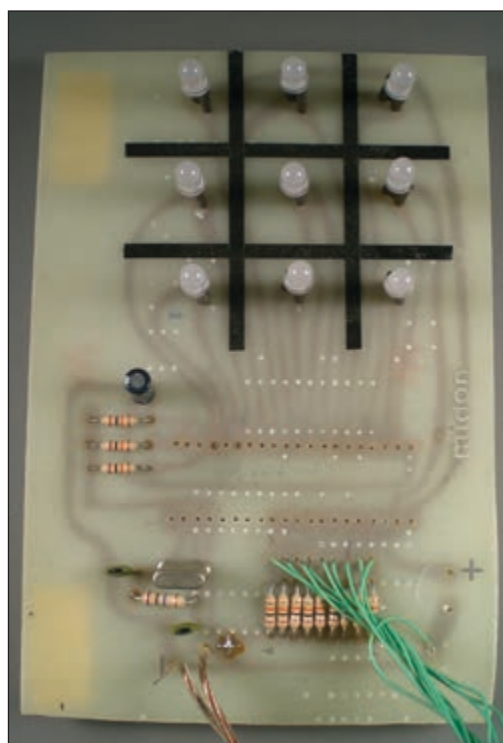
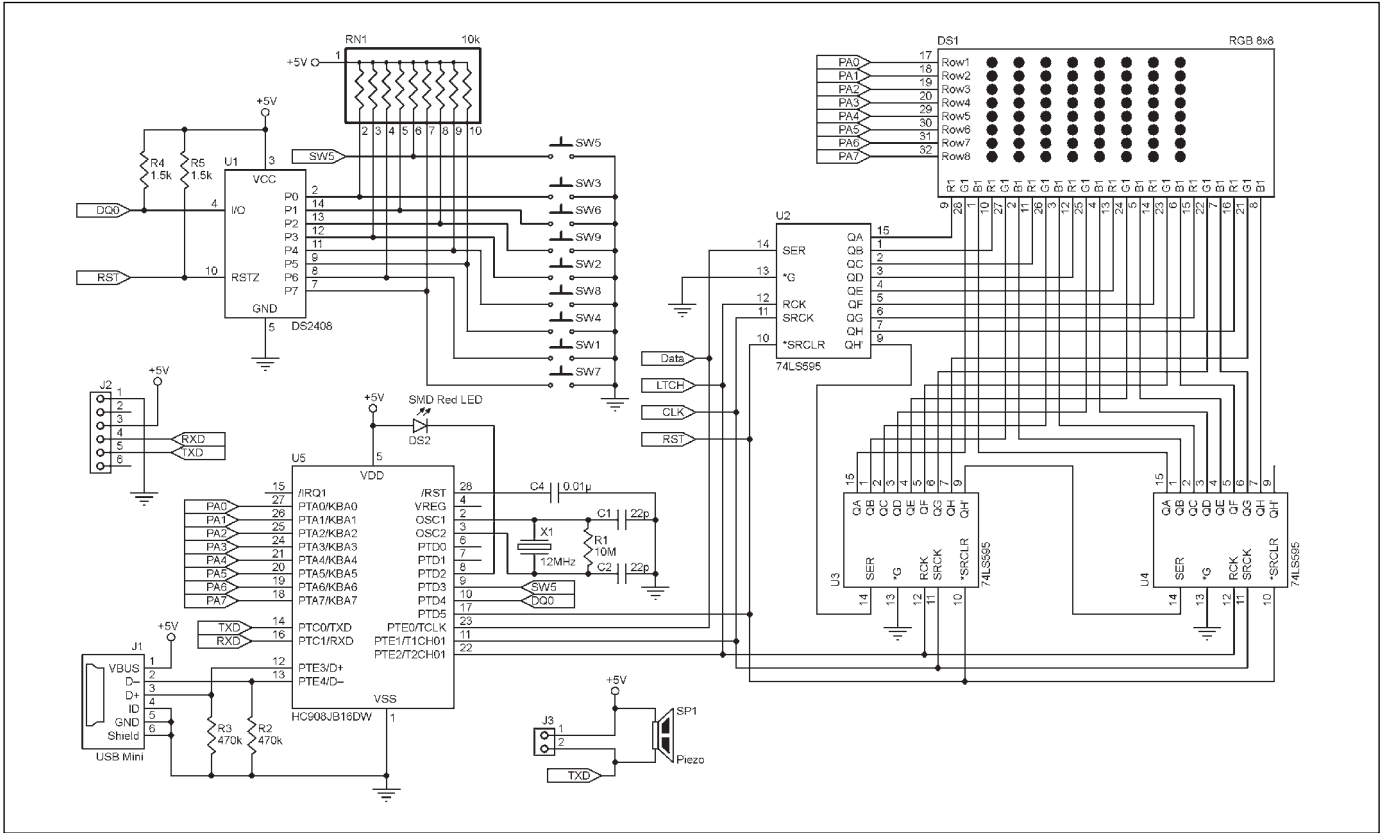


Photo 1

This design, built in the 1980's, used a 68HC705C8A microcontroller and bipolar LEDs to directly interpret the original game algorithm.



Surrounding the microcontroller are the interface components. Microcontroller U1 port A directly drives the display rows. U1 port E drives U2, U3, and U4, which are 74HC595's used as display column drivers. Port E also serves as the USB interface, which is needed for ICP. U1 port D provides the 1-Wire interface to U5, which is the Maxim Integrated

Products DS2408 used as the keyboard input. Port D also directly interfaces to SW5, which is used as the center switch of the Tic-tac-toe game as well as the control mechanism for ICP during reset. U1 Port C is used only for a debugging serial interface; it isn't used in the final product. U1 does not have a port B in the package style that I selected.

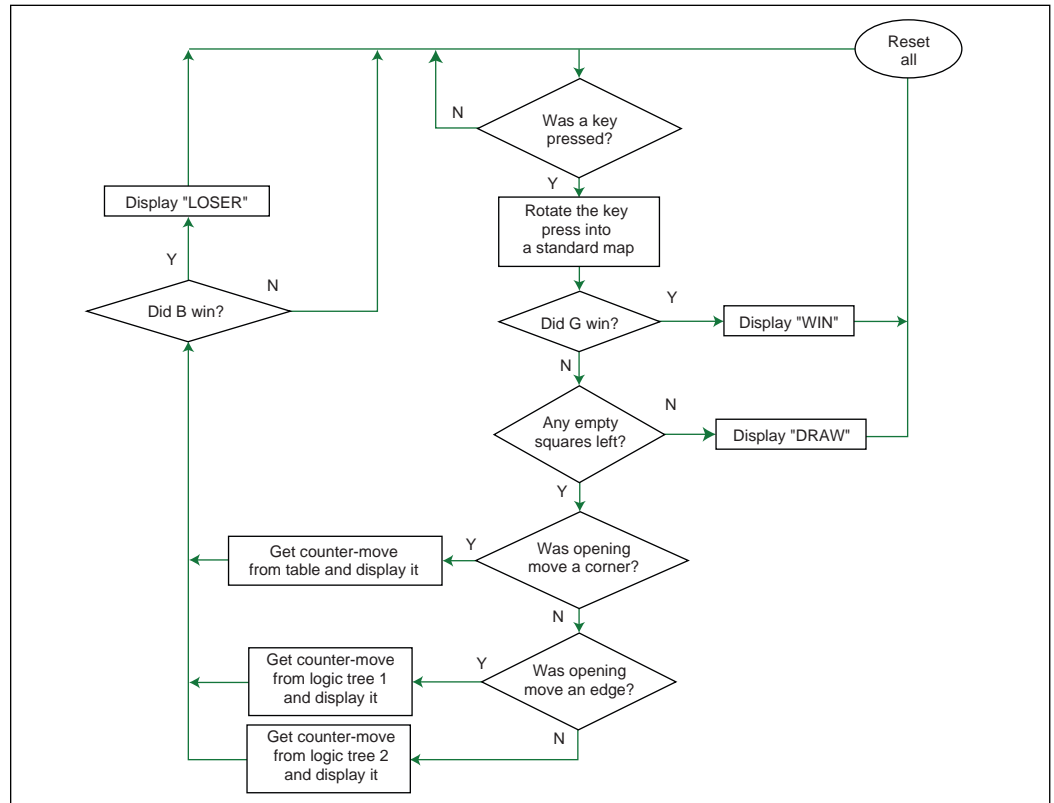
Figure 1
This is the RGB Tic-tac-toe game's circuitry. A Freescale 68HC908JB18 (U1) is the heart of the game.

```
* Parameter Equates
ICP_FLAG EQU $F7FE ; ICP flag
MON_USB_ICP EQU $FA19 ; Monitor USB ICP routines
*-----
* Main Program
*-----
ORG $F800 ; hard-coded location for this routine
ICP_Reset_Init:
    BCLR 3,DDRD ; set PTD,3 as input (i.e., use SW5 for this app)
    BRCLR 3,PTD,USB_ICP ; enter ICP if switch is pressed
    jmp RESET ; else jmp to the application program
*-----
* USB ICP
*-----
USB_ICP:
    sei ; disable interrupts
    mov #00000011,CONFIG ; enable STOP and disable COP
    mov #00000100,UCR3 ; enable USB pullup
    jmp Mon_USB_ICP ; check for new flash program
```

LISTING 1
This is the Power-On ICP check. A simple check for SW5 being depressed (logic zero) is made at power-on time, and the result determines whether or not to enter ICP mode, which is a call directly into the microcontroller's factory ROM code.

Figure 2

As you'd expect, the main game algorithm is straightforward. Wait for a move from the player, check for a counter-move, see if the game outcome has changed, and then do it again.



I chose the DS2408 as a keyboard interface because it offers a very simple processor connection, only one port pin, and I have a very extensive software library that I have developed over the years for interfacing to 1-Wire devices. The 74HC595's were ideal for driving each color of the display columns with only a three-wire interface required to communicate with all of them. A fourth pin from port D is used to reset the 74HC595s and the DS2408 during power-on.

Nine input switches are available, one for each position of the Tic-tac-toe display. The DS2408 decodes eight of them, and, as I mentioned previously, the center switch is connected directly to a microcontroller port pin. The bootloader also uses the center switch to determine if Normal mode or In-Circuit Programming mode is required.

The ICP circuit comprises SW5, the USB connector, J1, capacitor C4, and the two 470-k Ω resistors, R2 and R3. Except for a 12-MHz crystal, which is required for proper timing of the USB interface, no other components are required for ICP. A Windows XP program, provided by Freescale, handles the flash memory programming. (If anyone can provide me with a working Windows 7 variant of this software, I will gladly donate a working Tic-tac-toe game!) Power-on reset capacitor C4 is an absolute necessity for ICP as it handles the processor's reset circuitry. When first starting to use this controller, I left

out the reset capacitor, but soon found out that the controller would not always boot into ICP mode properly without it.

Listing 1 shows the simplicity of the ICP software interface. As stated in the Freescale application note AN2399: "Routines embedded in the monitor ROM area (\$FA00-\$FDFF and \$FE10-\$FFCF) are available to simplify the ICP process. A USB communications handler is already in the monitor ROM."^[1] The code shown in Listing 1 makes use of those built-in monitor ROM routines after checking for the request for ICP. This check is done at power-up by looking at switch SW5's state. If SW5 is depressed during the power-up process, the ICP monitor ROM code will be entered. If it is not, normal program code is entered. By chip design, if the program flash memory is blank, the ICP monitor code will be entered immediately at power-up, which is handy for a surface-mounted device.

The last major component is, of course, the RGB matrix display, DS1. I found mine on the Internet. Others are readily available from a variety of sources.

I added a piezo beeper SP1 to provide some feedback on key entries and wins/losses. The USB connector J1 is used as the power interface to the circuit, as well as the programming interface. Since 5-V USB power adapters are so common and inexpensive these days, I was able to simplify the power circuit.

The prototype originally included a header J2 that brought out the TXD and RXD signals from the microcontroller. This was tied to a plug-in TTL-to-USB converter—which allowed a serial connection from a PC—and was wired on the PCB so it could be connected to a serial-to-USB “friend.” I used this interface extensively during the software debug process, but it is not required in the finished product.

FIRMWARE

At start-up, following the ICP check, a sequence is entered that resets the peripheral devices, sets up all microcontroller registers, and then displays a “Welcome” message that scrolls across the LED matrix display. A simple flash pattern of Xs and Os is then shown. That was originally put in the code as a lamp test feature. It could have been deleted, but I left it in for the nice effect.

Figure 2 shows firmware’s main loop. The code is extremely linear and simple since I wanted to finish the project fairly quickly. In the flow diagram, the player is G (green) and the software is B (Blue).

The software continuously scans for a key press. Once any key has been pressed, a check is made to see if the pattern indicates

Move	Counter Move	Note
1	5	Mandatory starting move
2	3	
3	2	
4	7	
5	N/A	Counter-move to the mandatory starting move
6	8	
7	4	
8	6	
9	8	

1	2	3
4	5	6
7	8	9

Table 1

Take a look at the original *Popular Science* algorithm. This is a simple look-up table with one constraint: the first move must be in position 1.

Table 2

This is the way that the software interprets the position of any move. All moves made are mapped into this format before checking for counter moves.

SCOPE DEALS

PASSPORT-SIZE PC SCOPES
Great scopes for field use with laptops. Up to 200MHz bandwidth with 1GSa/s, high speed data streaming to 1MSa/s, built-in 1GSa/s AWG/wfm gen. - PS2200A **\$162+**

30MHz SCOPE
Remarkable 30MHz, 2-ch 250MS/s sample rate oscilloscope. 8-in color TFT-LCD and AutoScale function. Includes FREE carry case • 3 yr warranty! - SDS5032E **\$289**

60MHz SCOPE
Best selling 60MHz 2-ch scope with 500MSa/s rate • huge 10MSa memory! 8" color TFT-LCD. Includes FREE carry case! - SDS6062 **\$349**

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

70-300MHz SCOPES
Fast, versatile 2-ch 2GSa/s scopes with 8" WVGA LCD, integrated generator, 14Mpt memory, very low noise floor. - DS2000A series **\$839+**

INCREDIBLE LOW PRICES, FREE TECHNICAL SUPPORT
GREAT CUSTOMER SERVICE **SAELIG.COM**

RS485/422/232/TTL

ASC24T
RS232<=>RS485 ATE Converter



\$45.00 board only

IBS485HV
5 Port Isolated
RS485 Repeater



\$349.00

Enclosures, Cables, Power Supplies and Other Accessories

INTRODUCING THE SMFCOMX!

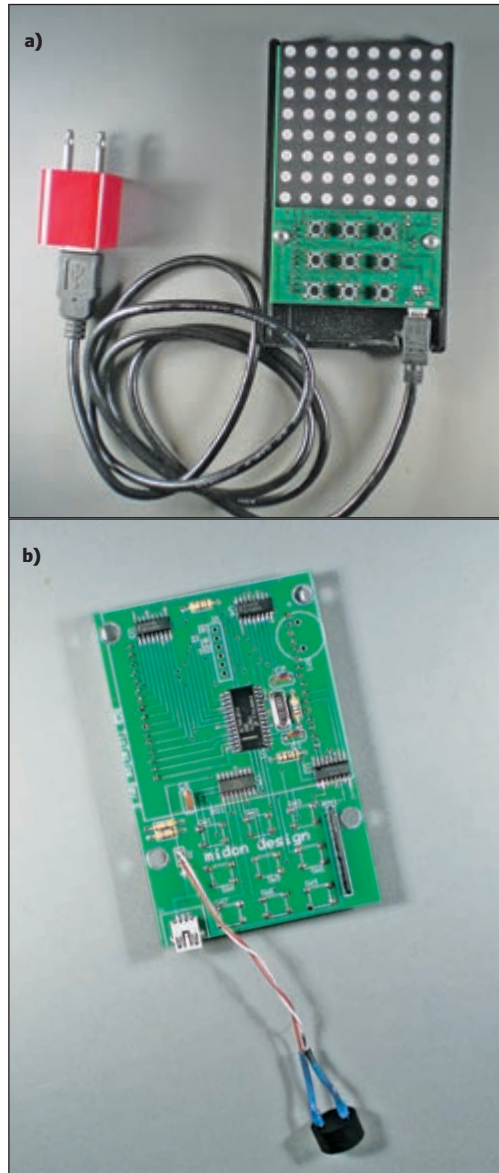
- Converters
- Extended Distance Units
- Repeaters
- Serial to Digital I/O
- Multi-Repeaters
- Large Multi-Drop Networks
- Hubs
- Custom Units & Smart Units
- Fiber Optics
- Isolators
- Industrial, 3.0 KV Isolation



Call the RS485 Wizards
513-874-4796
www.rs485.com

PHOTO 2

a—Take a look at the final product in its case, which is the bottom half of a standard PacTec case. **b**—This is the component side of the final product.



a win by the player. If not, a response is calculated and, after an intentional 1-s delay, the counter-move is presented. If a win, loss, or draw occurs, a corresponding "Win", "Loser," or "Draw" message is scrolled across the display and the program resets all registers, after which it returns to a known starting point.

The counter-moves are calculated in two different ways. The first is the simplest. If the starting move is a corner square, a table look-up is performed. This table is based on the algorithm used in the original *Popular Science* article. That algorithm requires a corner starting move and was devised by the original authors to permit a win by the player every now and then, although not as often as the player would like! A draw is by far the more likely outcome.

The second method checks for counter-moves to a starting move in either the center

(logic tree 1) or an edge (logic tree 2). This is a simple state check of each move based on the history of moves up to this point. Counter-moves are designed to force a draw or loss whenever possible. There are some moves that cannot be countered easily. Those are left to you to determine.

Each move, either by the player or by the controller, is recorded in an array. That array is used by the second method to check for the next best move. The array is also kept by the first method (table look-up); however, it isn't used to calculate a counter move. The table look-up method is stateless and does not require the program to understand what moves have already been made, except to determine if a win, loss, or draw has occurred.

Table 1 shows the counter-moves for the table look-up method. Note that there is no counter-move for a center square. The original authors did this intentionally since they required the first move to be a predetermined corner, position 1 in **Table 1** and **Table 2**, and the counter-move for that was the center. **Table 1** is quite ingenious. I commend the original authors for the simplicity caused by forcing the first move. I have only adapted that algorithm to allow for any corner to be equivalent to a move from position 1.

THE HARDWARE

I chose to implement this game on a custom PCB, which let me use surface-mount components for the most part. This kept the game small enough to be hand-held.

The PCB was sized to fit into a standard PacTec enclosure, which I happened to have on hand. Only half of the enclosure is used for simplicity.

The interesting aspect of the PCB design is that the component side was used for all the surface-mount and through-hole components; however, the solder side was designed to permit the mounting of the display and switches. I did this to keep the visible portion of the game as clean as possible. This made for an interesting exercise in mirroring the pinouts of the display matrix while designing the PCB, but that didn't take too long.

Photo 2a shows the final product in its case, which is the bottom half of a standard PacTec case. In the time I had, coming up with a clean way to have the display visible while still allowing the switches to be pressed prevented the use of a full case. **Photo 2b** shows the component side of the final product.

THEORY OF OPERATION

My game's operation is simple. A player always begins with X and is represented by green LEDs. The Player presses a switch (any switch) and then chases the Os (blue



12th International System-on-Chip (SoC) Conference, Exhibit & Workshops

October 22 & 23, 2014

University of California, Irvine - Calit2

www.SoCconference.com

CALL FOR SPEAKERS AND SPONSORS. . . Don't Miss Out!

This Year's Theme: *"Innovative SoCs Empowering the Communications Market."*

Platinum Sponsors

- FinFET Technology & Design
- Analog & Mixed-Signal Designs
- Sub 14nm Designs & Beyond
- 3-D ICs Designs
- IC Security & Challenges
- Multicore Software Development
- SoC Design & Verification
- Innovative EDA Tools
- Complex IP Subsystems
- Low-Power Techniques
- Memory Trends & Technologies
- Table-Top Exhibit (Free Passes)



- Complex Mixed-Signal SoCs
- SOI vs. CMOS
- RF Design
- FPGAs –Trends & Designs
- High-Speed I/Os
- Smarter Mobile Devices
- Multicore SoC Platforms
- Network-on-Chips (NoCs)
- Informative Panels
- IEEE Student Design Contest
- Networking Opportunities
- And Much More. . .

**Promote Your Technology, Products & Services at
The Most Informative, Targeted & Educational
IC & IP Design Conference, Exhibit & Workshops of the Year!**

Keynote Speakers



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



Skyworks Solutions
Dr. Peter L. Gammel,
Chief Technology Officer
(CTO).

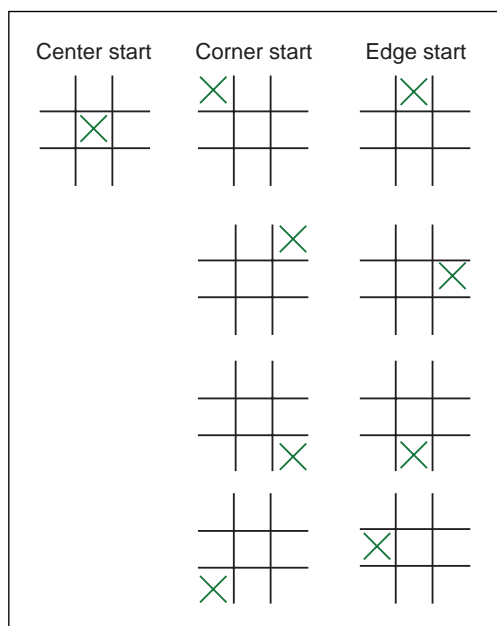
The SoC Conference brings together a very targeted and sophisticated audience from leading-edge technology companies as well as key universities for two exciting days devoted to groundbreaking products, technologies, and business disclosures, focused exclusively on designing complex System-on-Chip and the related technologies driving the SoC/ASIC/ASSP/FPGA/Foundry industry.

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

www.SavantCompany.com & www.SoCconference.com

Figure 3

There are only three possible starting moves if you eliminate symmetrical ones: the center, a corner, and an edge.



LEDs) around the game. There is nothing to configure, and no options to pick.

How does this all work? Well, the game theory is pretty straightforward. Let's look at some details.

First, you need to know the total number of the game board's possible variations. According to Wikipedia, there are 255,168 total combinations of the game, 51% of which are wins by starting player and 31% of which are wins by the second player.^[2] The remainders are draws. However, this includes all possible states. According to the same post, if symmetrical moves are removed (i.e., moves that are the same if the board is rotated 90°, 180°, or 270°), then this leaves only 26,830 possible combinations. Whew! That still seems like a lot. However, these can

be broken down even further if you eliminate what I call "stupid moves." Those would be moves that lead to an easy win by either player. Non-stupid moves are what I deemed to be moves by both players playing to win.

There are only three possible starting moves if you eliminate symmetrical ones: the center, a corner, and an edge (see **Figure 3**). Since I chose to use the original authors' algorithm (see Tables 1 and 2) for a corner starting move, this left me with the design of an algorithm to counter center and edge starting moves. According to the original algorithm, only a starting move from a corner was permissible, which was another reason to develop algorithms for the other starting moves. In Table 1, the only permissible starting move is for position 1. (Refer to Table 2 for the numbering of each position in the Tic-tac-toe matrix.) In software, however, we can rotate the board so that starting moves in position 3, 7, and 9 are all equivalent to a starting move of 1.

After spending hours analyzing the many possible game combinations, I determined that numerous combinations could be eliminated by removing "stupid moves" and by trying to force the player (X) to play where the computer (O) could either win or draw. Empirically, this left me with a small combination of moves to inspect for center starts and to inspect for edge starts. In software, I determined that these combinations could easily be checked via a simple look-up, check, and branch routine. Counter-moves that would try and force a loss or draw at each X player's move were devised for each combination and hard-coded in to the routines. Since each successive move determines the type of inspection to make for the next move, a move history table is used as part of the look-up, check, and branch software. The hard-coded responses from my study then become the counter-moves.

Moves that are not on my list of empirical combinations are simply ignored and a sort of random counter-move is provided in their place. This is probably the best way for X to win since these random counter-moves don't account for any strategy. Of course, I won't reveal those moves.

OTHER OPTIONS

When originally designing my electronic game, I had aspirations to use 1-Wire devices throughout for the interfaces to the display and keys. However, after building the first prototype, which used all 1-Wire devices, I pretty quickly proved that the 1-Wire protocol is just too slow for multiplexing 64 LEDs. If I had spent more time analyzing the 1-Wire timing, I would have known that and saved

PROJECT FILES



circuitcellar.com/ccmaterials

REFERENCES

[1] D. Lau, "In-Circuit Programming of FLASH Memory via the Universal Serial Bus for the MC68HC908JB16," AN2399, Freescale Semiconductor, 2003.

[2] Wikipedia, "Tic-Tac-Toe," <http://en.wikipedia.org/wiki/Tic-tac-toe>.

RESOURCE

B. Bawer and W. J. Hawkins, "Electronic Tick-Tack-Toe in a Cigarette Box," *Popular Science*, December 1970.

SOURCES

HC908JB16 Microcontroller

Freescale Semiconductor, Inc. | www.freescale.com

DS2408 1-Wire Switch

Maxim Integrated | www.maximintegrated.com

myself a redesign and PCB layout change. Lesson learned: read the freaking manual (RTFM). That oversight led to changing the column drivers from DS2408s to 74HC595s. The HC595s are driven with a 3-ms interrupt to keep the flicker of the display to a minimum. 1-Wire timing is certainly fast enough to detect switch presses, so I left a DS2408 in as the keypad scanner.

What I did not think through ahead of time was that the LEDs' forward voltage varies so much between the colors of the chosen 8 × 8 matrix display (from 1.9 to 3 V). This difference leads to the dimming of green LEDs in a row that also has red LEDs on. Use of a constant current supply, such as provided with a MY9221 or MY9268 from MY-Semi might have been an option; however, the time I had available and the space available on the PCB did not permit the luxury of such a major change to the circuit. Thus, I changed the color scheme for the game from green and red to green and blue, because the blue and green LEDs have similar forward voltages. The red LEDs are still used for text output since there is only one color used for text.


I had originally planned on using a standard USB-B connector for this design, but it was so large (relatively) that it prevented

a proper fit in my available enclosure. That resulted in changing the standard USB-B connector to a mini-B type of connector.

The beeper that I had on hand was also a physical problem. The PCB size did not leave me with enough space to mount it on the solder side, and the available vertical space wasn't enough to mount it on the component side. So, I squeezed the connection into a spot on the component side and extended the connection via wires. Adding a bit of length to the PCB would have left me the space I needed to add the beeper below or beside the switches. Of course, this would've resulted in an enclosure change, but that would've been easy enough if planned ahead.

Lastly, if I'd had more time, I could've designed the software more elegantly—that is, the look-up, check, and branch routines could have been contained within a routine accessing a look-up table or group of tables.

RESULTS

Modernizing my original microcontroller design was a lot of fun and the resulting game looks pretty nice. Hopefully, this article and the accompanying software will help you understand a bit about the game theory and driving matrix LED displays. 

ABOUT THE AUTHOR

Mitch Matteau (mitch@midondesign.com) founded Midon Design, a company that provides hobbyists with 1-Wire control solutions. He holds a Bachelor's of Applied Sciences from Ottawa University in Ottawa, Canada. Mitch enjoys both the hardware- and software-related aspects associated with designing embedded devices.

CONNECT WITH

Circuit Cellar

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

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

circuitcellar.com

 @circuitcellar
@editor_cc

 circuitcellar




FlashPro430 FlashPro-CC FlashPro2000 GangPro430 GangPro-CC


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

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

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



One PC and 8 programmers



www.elprotronic.com

GREEN COMPUTING

Pooling Microarchitectural Resources

Towards Flexible Heterogeneity

3-D stacking enables novel ways of “resource pooling” in a processor owing to the short wiring distances among the layers in the stack. This article investigates how pooling microarchitectural resources intelligently across applications improves energy efficiency.

By Ayse K. Coskun (US)

Majority of computing systems today include multicore processors. These multicore systems typically include a “homogeneous” set of cores, meaning that all the cores within the processor are of the same kind and have the same size and types of microarchitectural components. In addition, in such homogeneous systems, the external resources given to each core, such as the L2 caches or network interfaces are also typically identical.

While such symmetrical homogeneous multicore design provides ease of circuit design, placement, and routing, there are potential limitations as well. First, resources on a multicore system are often under-used as the systems are not necessarily designed for the typical or average use case, but instead they are designed with some “headroom” considering that some applications may need a larger amount of resources. This headroom indicates over-design such as placing a large amount of on-chip cache that may be under-utilized for a significant portion of the time. Obviously, over-design increases the area and power costs.

Over-design is less of an issue when a multicore architecture is highly optimized for a specific application, such as in the case of an embedded processor that is responsible of performing a few tasks only. Most systems today, including many embedded systems, however, run a variety of tasks throughout their lifetime. In addition, applications often have different phases (e.g., a single application

may go through a number-crunching phase that is heavily using the arithmetic units followed by a memory-intensive phase including lots of reads and writes to/from caches and memory).

Overall, different software applications, or different tasks within an application, thrive (i.e., achieve high performance) under different architectural configurations. For example, one application may benefit from a larger L2 cache while another one may not need much cache space and instead may operate better with a larger instruction issue queue.

Considering most systems are homogeneous by design but run a heterogeneous set of applications, how can we design efficient architectures that can provide desirable performance and power tradeoffs for a variety of applications?

One approach is to design the processor in a heterogeneous way, such as by employing different cores in the same chip (e.g., ARM’s big.LITTLE architecture) or by including GPUs or accelerators along with a set of homogeneous CPU cores (e.g., AMD’s APU systems). (Note that almost all processors include heterogeneity as they include many different components. Here heterogeneity specifically refers to including different types of cores in a multicore processor.) Heterogeneous design has its challenges, too, as it may bring higher design complexity and potentially longer time-to-market. Nevertheless, I expect to see a larger set of heterogeneous multicore systems in commercial products in the near

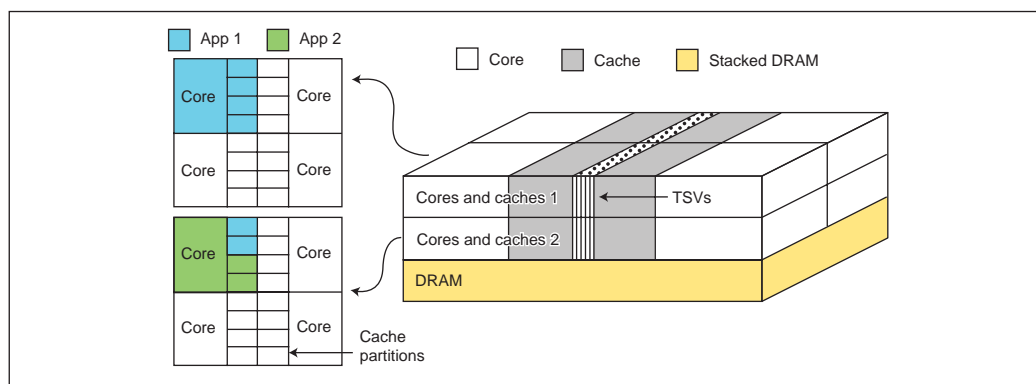


FIGURE 1

Cache pooling on a 3-D multiprocessor system. Each core has four dedicated cache banks by default. Each bank, however, can be pooled to the core in the adjacent layer if needed. In this example, Application 1’s performance benefits strongly from using a larger L2 cache. Thus, Application 1 is given a larger amount of private cache banks by borrowing cache banks from the core right underneath.

future as such systems have the potential to achieve better energy efficiency.

Instead of designing the system with a heterogeneous set of cores, this article discusses a novel design alternative: flexible heterogeneity. The goal of flexible heterogeneity is to achieve higher energy efficiency through low-cost, simple reconfigurations of the homogeneous system based on applications’ needs. This low-cost reconfiguration “pools” the performance-critical architectural components among the cores and allocates more hardware resources to cores that need them. At the same time, it limits the resources given to cores that are running less intensive applications. In this way, a pool of resources can be shared more efficiently. Also, in this way, it is possible to reduce the need for over-design.

FLEXIBLY HETEROGENEOUS PROCESSORS

The key idea of flexible heterogeneity is to determine the application needs while the system is running and allocating the architectural resources in a way that matches this demand. Which resources should we pool among the cores? In my research lab at Boston University, my students and I developed a cache pooling technique for two reasons.^[1] One, caches significantly affect the performance of many software applications. And two, applications strongly differ in their cache needs (or in other words, in their “cache-hungriness”).

Many processor architectures today have private L1 and L2 caches for the cores, and some also have large shared L3 caches. L1 caches are small-sized and are often highly utilized as a result. L2 usage among the applications, however, changes dramatically. L2 cache is also where over-design occurs more often to accommodate for the application diversity. A 1- or 2-MB private L2 cache per core is fairly common in high-performance desktop and server processors. As a result, caches occupy a large portion of the silicon area.

Note that designing a system with a shared L2, where all cores are connected to the same cache, reduces the cache coherence overhead and provides a more flexible distribution of cache resources to the cores, compared to assigning a private cache to each core. (Private caches require hardware or software coherence protocols that maintain correct program operation and data consistency.) Still, shared cache design at the L2 level is often not preferred due to several reasons. First, applications generally perform better with dedicated resources as they do not interfere with each other’s cached data in this way. Second, shared caches are larger and may have longer access times as a result. Most importantly, shared L2 cache approach is not scalable to support a large number of cores because of the high access latency and routing overhead that occur when each core needs to reach a centralized resource.

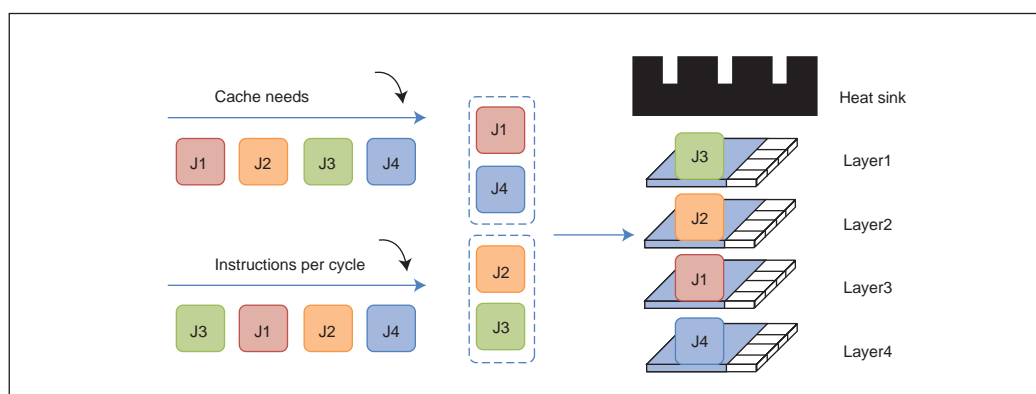


FIGURE 2

Job allocation policy to maximize resource pooling efficiency. Jobs (J1–J4) are ranked based on their cache needs and IPC, which is an indicator of power consumption. Jobs with contrasting cache needs are then placed on adjacent layers. Higher IPC job pairs are placed closer to the heat sink to ease heat removal.



ABOUT THE AUTHOR

Ayse K. Coskun (acoskun@bu.edu) is an assistant professor in the Electrical and Computer Engineering Department at Boston University. She received MS and PhD degrees in Computer Science and Engineering from the University of California, San Diego. Ayse's research interests include temperature and energy management, 3-D stack architectures, computer architecture, and embedded systems. She worked at Sun Microsystems (now Oracle) in San Diego, CA, prior to her current position at BU. Ayse serves as an associate editor of the *IEEE Embedded Systems Letters*.

Private caches generally provide better performance and easier design, but they restrict the cache size given to each core, as discussed above. Thus, if we can pool the L2 cache resources dynamically according to application resource needs, each core can then get a private but appropriately-sized cache.

Cache pooling can be performed at the granularity of cache banks, where each bank can be used by its dedicated core or can be borrowed by another core when needed. As the number of cores and caches increase, this approach is not easily scalable in traditional design, simply because a core would need to access cache banks potentially at the opposite side of the chip. Long access distance means longer wire length, which indicates larger latency. For example, sharing microarchitectural resources across a short distance incur only a few picoseconds of additional delay, while accessing a unit across the chip may cause hundreds of nanoseconds.^[2]

3-D stacking, where smaller chips are stacked vertically instead of building a single big chip, provides low-latency resource sharing opportunities across the different layers in the stack. This is because the through-silicon vias (TSV) that connect two architectural components on adjacent layers are only tens of microns apart (as opposed to mm-scale wire-length to connect two units across the chip). In addition to the advantage of the short distance, it is possible to place a larger number of TSVs that can be clocked at a high rate, and thus, provide a wide bandwidth for data transfer.

Figure 1 demonstrates the cache pooling idea. In this example 3-D system, there are four cores on each layer and each core has four private cache banks. A cache bank can be configured to be used by its main dedicated core, or it can be taken over by the core right above/under the main core. In this example, Application 1 thrives when running with a larger L2 cache size, so it is given a total of six cache banks. Application 2, on the other

hand, does not need much cache space and therefore is given only two banks. As the applications running on the system change, the cache allocation can be reconfigured.

An essential step in resource pooling is to determine the application demand. In cache pooling, one needs to estimate the cache-hungriness of each application. One way to do this is to first determine how much instructions-per-cycle (IPC) improvement any application should get from using an additional cache bank in order to achieve better energy efficiency (e.g., lower energy delay product, or EDP). This threshold of improvement can be determined by the designer through architectural performance and power simulations using tools such as gem5 (www.gem5.org/Main_Page) and McPat (www.hpl.hp.com/research/mcpat/).

After determining a threshold, it is possible to iteratively converge on the best cache bank distribution among two applications running on cores above/below each other on adjacent layers as follows. First, run each application using a small number of banks. Then, add another cache bank to each core, run the application for a short interval, and see if IPC improves beyond the predetermined threshold. If it does, add another cache bank. If the performance improvement is less than the predetermined threshold, then use the last cache bank setting for that application. If both applications continue to thrive with a larger number of cache banks, then one approach would be to favor the application that is getting larger performance benefits. Any unused cache banks can be selectively turned off to save energy.

Clearly, one would need to add some hardware into the traditional processor architecture to be able to pool the cache banks across the layers. To ease the design of this reconfigurable system, my students and I restricted the cache pooling mechanism such that only a pair of cores that are right above/below each other on a 3-D stack can share resources. The hardware we added mainly included registers (status bits) for keeping track of the cache bank assignment, selection logic (multiplexers) for determining which layer to read from/write to, and TSVs to enable data transfer.^[1] Overall, the total area overhead is on the order of a few thousand transistors and a few hundred TSVs. This overhead is negligible for a multicore processor.

Another source of overhead is the cache flushing that is needed after a cache reconfiguration to ensure the data in the caches are relevant. In our experiments, we used a 100-ms interval to check for the need of cache pooling adjustment and

REFERENCES

[1] J. Meng, T. Zhang, and A. K. Coskun, "Dynamic Cache Pooling for Improving Energy Efficiency in 3D Stacked Multicore Processors," in Proceedings of the IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), 2013.

[2] H. Homayoun, H., V. Kontorinis, A. Shayan, T.-W. Lin, and D. M. Tullsen, "Dynamically Heterogeneous Cores Through 3D Resource Pooling," in Proceedings of IEEE 18th International Symposium on High Performance Computer Architecture (HPCA), 2012.



circuitcellar.com/ccmaterials

Get Started with Advanced Control Robotics

By C. J. Abate (Content Director, EIM)

I met Hanno Sander in 2008 at the Embedded Systems Conference in San Jose, CA. At the time, Sander was at Parallax's booth demonstrating a Propeller-based, two-wheeled balancing robot. When I saw his interesting balance bot design and his engaging way of explaining design to interested engineers, I knew that Sander would be an excellent resource for future *Circuit Cellar* content. I was right.

Several months after the conference, we published an article he wrote about the balancing robot project (*Circuit Cellar* March 2009). Today, Sander runs OneRobot with the aim of "building high-quality, affordable products by pushing off-the-shelf components to their limits."¹

THE FUTURE IS NOW

When it comes to robotics, the future is now. With the ever-increasing demand for robotics applications—from home control systems to animatronic toys to unmanned planet rovers—it's an exciting time to be a roboticist, whether you're a weekend DI-

Yer, a computer science student, or a professional engineer.

It doesn't matter whether you're building a line-following robot toy or tasked with designing a mobile system for an extraterrestrial exploratory mission: the more you know about advanced robotics technologies, the more you'll succeed at your workbench.

Advanced Control Robotics is intended to help roboticists of various skill levels take their designs to the

next level with microcontrollers and the know-how to implement them effectively.

THEORY & BEST PRACTICES

Advanced Control Robotics simplifies the theory and best practices of advanced robot technologies. You're taught basic embedded design theory and presented handy code samples, essential schematics, and valuable design tips (from construction to debugging).

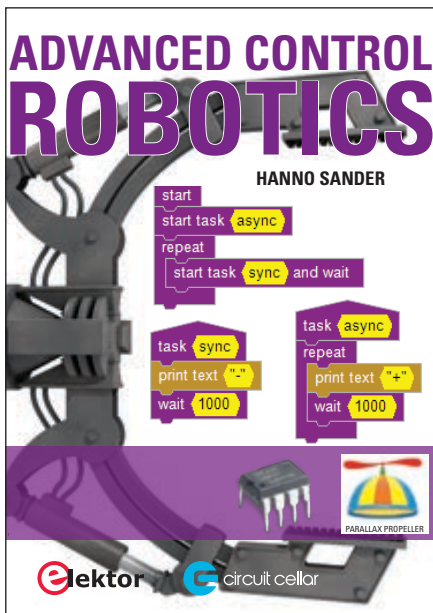
LEARN THEN DESIGN

The principles described and topics presented in **Advanced Control Robotics** are immediately applicable. With the book at your side, you'll be innovating in no time.

Sander covers:

- Control Robotics: robot actions, servos, and stepper motors
- Embedded Technology: microcontrollers and peripherals
- Programming Languages: machine level (Assembly), low level (C/BASIC/Spin), and human (12Blocks)
- Control Structures: functions, state machines, multiprocessors, and events
- Visual Debugging: LED/speaker/gauges, PC-based development environments, and test instruments
- Output: sounds and synthesized speech
- Sensors: compass, encoder, tilt, proximity, artificial markers, and audio
- Control Loop Algorithms: digital control, PID, and fuzzy logic
- Communication Technologies: infrared, sound, and XML-RPC over HTTP
- Projects: line following with vision and pattern tracking

Are you ready to start learning and innovating? Order **Advanced Control Robotics** today.



Title: Advanced Control Robotics

Author: Hanno Sander

Publisher: Elektor/Circuit Cellar

Year: 2014

Buy: www.elektor.com/advanced-control-robotics

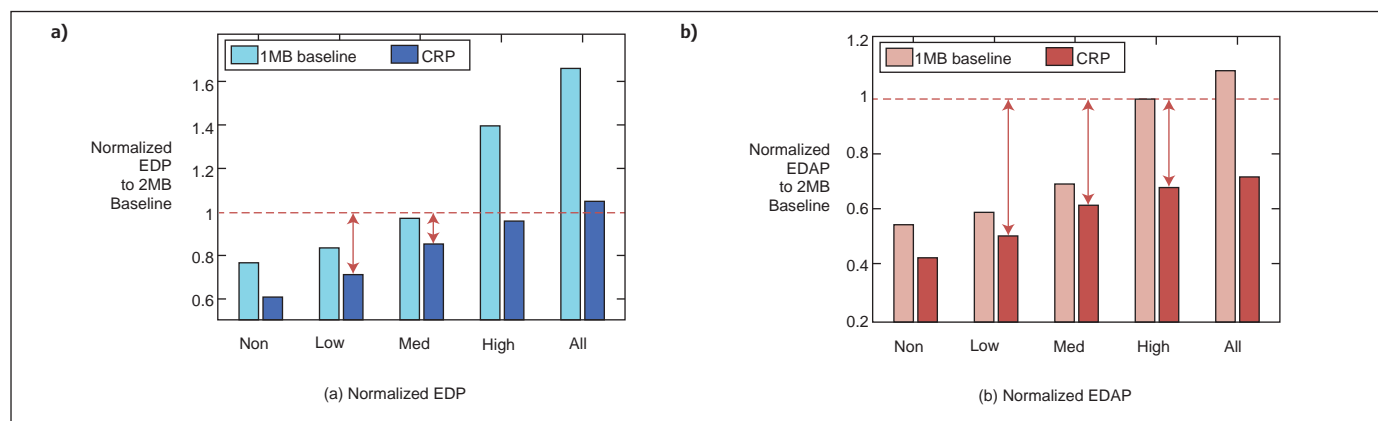


FIGURE 3

Energy-delay product (EDP) and energy-delay-area product (EDAP) results for using 1-MB private caches and for using cache resource pooling (CRP) on a 3-D stacked system with equivalent total cache size. X axes provide the cache-hungriness of the application mix, increasing from left to right. Lower EDP and EDAP are more desirable.

reconfigure if needed. Flushing the caches causes a “cold start” as the application needs to bring its relevant data from memory. My students and I measured the cold start overhead to be less than 1 ms for SPEC CPU 2006 applications using the gem5 simulator. Thus, cache pooling causes at most 1% delay when it is run with a 100-ms period. This delay is easily compensated with the performance improvement achieved by efficiently distributing the cache banks. Cache pooling frequency can be adjusted as needed to capture application phases at a negligible overhead for other application suites.

INTEGRATING JOB ALLOCATION & CACHE POOLING

Job allocation impacts the potential benefits of cache pooling as it determines on which core each application runs. For example, if the OS allocates two cache-hungry applications on adjacent cores on top of each other, cache pooling benefits may be limited. On the other hand, placing applications with contrasting cache needs on top of each other would increase the pooling efficiency.

In my lab, we designed a job allocation policy that maximizes benefits of cache pooling. **Figure 2** demonstrates the main idea of this policy. We first estimate the cache needs of each application using L2 misses and L2 replacement rates over a given number of cycles. We also observe the IPC of each application as IPC is often an indicator of the power consumption. We then pair the applications with contrasting cache demands, and use the IPC measurements to place potentially higher power applications closer to the heat sink.

Placing applications with higher power consumption closer to the heat sink eases cooling and reduces the thermal hot spots. Note that IPC alone may not always be sufficient for accurate power estimation as the instruction mix also determines the power consumption (e.g., a floating-point

intensive application may consume higher power than an integer application even when they have the same IPC). It is possible to use a larger set of performance counter measurements to improve power estimation accuracy if needed.

High temperatures are among the important challenges of 3-D stacking. I discussed some of the temperature modeling and management issues in 3-D systems in my January 2014 and March 2014 *Circuit Cellar* articles. If high-power cores are integrated into a 3-D stack, thermally-aware job allocation may not be sufficient and novel cooling solutions such as liquid cooling may need to be considered. In our cache pooling experiments, we used 1-GHz medium-power cores. For these experiments, the proposed job allocation method was sufficient to keep the temperature below critical levels.

If we integrate a larger amount of cores into the 3-D stack, each layer will likely have multiple cores. In this case, one can envision the 3-D stack as a collection of “columns of cores,” where a column is simply cores stacked on top of each other. Within each column of cores, one can apply the job allocation and cache pooling policies described above. Across these columns, it is possible to balance the number of cache-hungry jobs using additional job migrations to maximize pooling efficiency.

Figure 3a shows the EDP when using a 1-MB static private cache configuration and when using cache resource pooling (CRP) on a four-core 3-D system. The results are normalized with respect to the EDP results for a system with 2-MB private L2 caches. The x-axis shows the cache-hungriness of the application mix running on the system. For example, *medium* means some of the applications are cache-hungry, *none* refers to the case where none of the applications are cache-hungry, and *all* means all of the applications are cache-hungry.

Lower EDP is more desirable as it indicates lower energy consumption and lower delay

(i.e., higher performance). When all or most of the applications are cache-hungry and using the caches intensively, there is not much motivation for cache pooling as expected. For low and medium cases, CRP achieves over 30% reduction in EDP compared to using 2-MB static caches. Real-life workloads often include a mix of cache-hungry and non-cache-intensive applications, resembling the “low” and “medium” cases in the figure.


When area is also considered as a factor, the CRP results are even more impressive. Figure 3b demonstrates the energy-area-delay product (EDAP) results for the same experiment and indicates over 50% reduction in EDAP. This means that CRP can simultaneously achieve lower energy and good performance at a much lower area compared to using large static L2 caches.

OTHER POOLING OPPORTUNITIES

Caches are not the only resource on a multiprocessor that can be dynamically pooled. For example, recent work by Houman Homayoun et al.^[2] introduces resource pooling for performance-critical microarchitectural units such as reorder buffer, register file, load/store queue, and instruction queue. As 3-D stacking enables placing on-chip stacked

As 3-D stacking enables placing on-chip stacked DRAM together with the processor, as shown in Figure 1, DRAM banks and the on-chip memory bandwidth are also candidates for dynamic pooling.

DRAM together with the processor, as shown in Figure 1, DRAM banks and the on-chip memory bandwidth are also candidates for dynamic pooling.

I believe there are tremendous efficiency improvement opportunities in designing a flexibly heterogeneous computing fabric that can be configured using simple and low-cost hardware and software methods. For example, envision a set of cores, a set of poolable caches, a group of scratchpad memories, and on-chip DRAM banks, whose partitions and bandwidth can be allocated to applications based on their demands and characteristics. In this way, each application can be provided with a customized set of resources and energy efficiency of the processor can be substantially improved. I think 3-D stacking technology is a great candidate to realize this vision. 

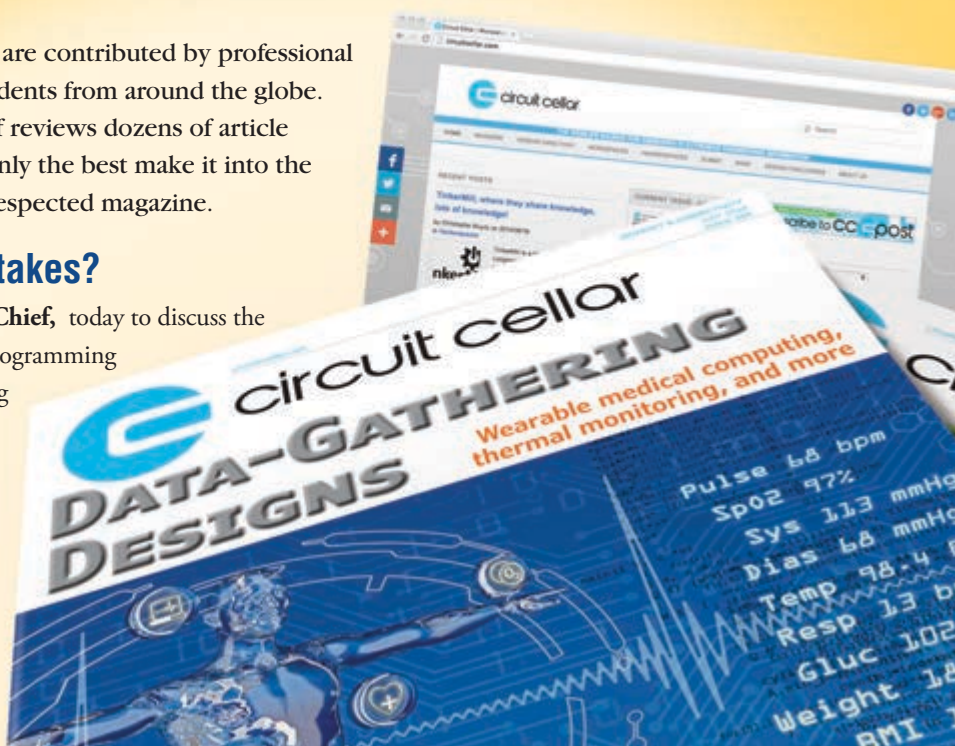
Get PUBLISHED. Get NOTICED. Get PAID.

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

Do you have what it takes?

Contact **C. J. Abate, Editor-in-Chief**, today to discuss the embedded design projects and programming applications you've been working on and your article could be featured in an upcoming issue or online at circuitcellar.com.

Email: editor@circuitcellar.com



THE CONSUMMATE ENGINEER

The Humble Resistor (Part 2)

Variable Resistors

Fixed resistors were covered in the first part of this article series. Now we'll turn to variable resistors, which come in different styles and power ratings. This article covers potentiometers, rheostats, and more.

By George Novacek (Canada)

In the first part of this article series, I introduced the fixed resistor and then covered the topics of thick and thin film resistors. Now let's turn our attention to variable resistors.

VARIABLE VARIETIES

Variable resistors come in many different sizes, power ratings, and values: potentiometers or rheostats, single or multturn, and so on. They comprise a resistive element, a sliding contact (wiper) that moves along the resistive element, making a contact to it. They also include some form of a mechanism and housing to keep the parts together, move the wiper, and keep out the dirt. Refer to **Photo 1** for a few examples of variable resistors.

POTS

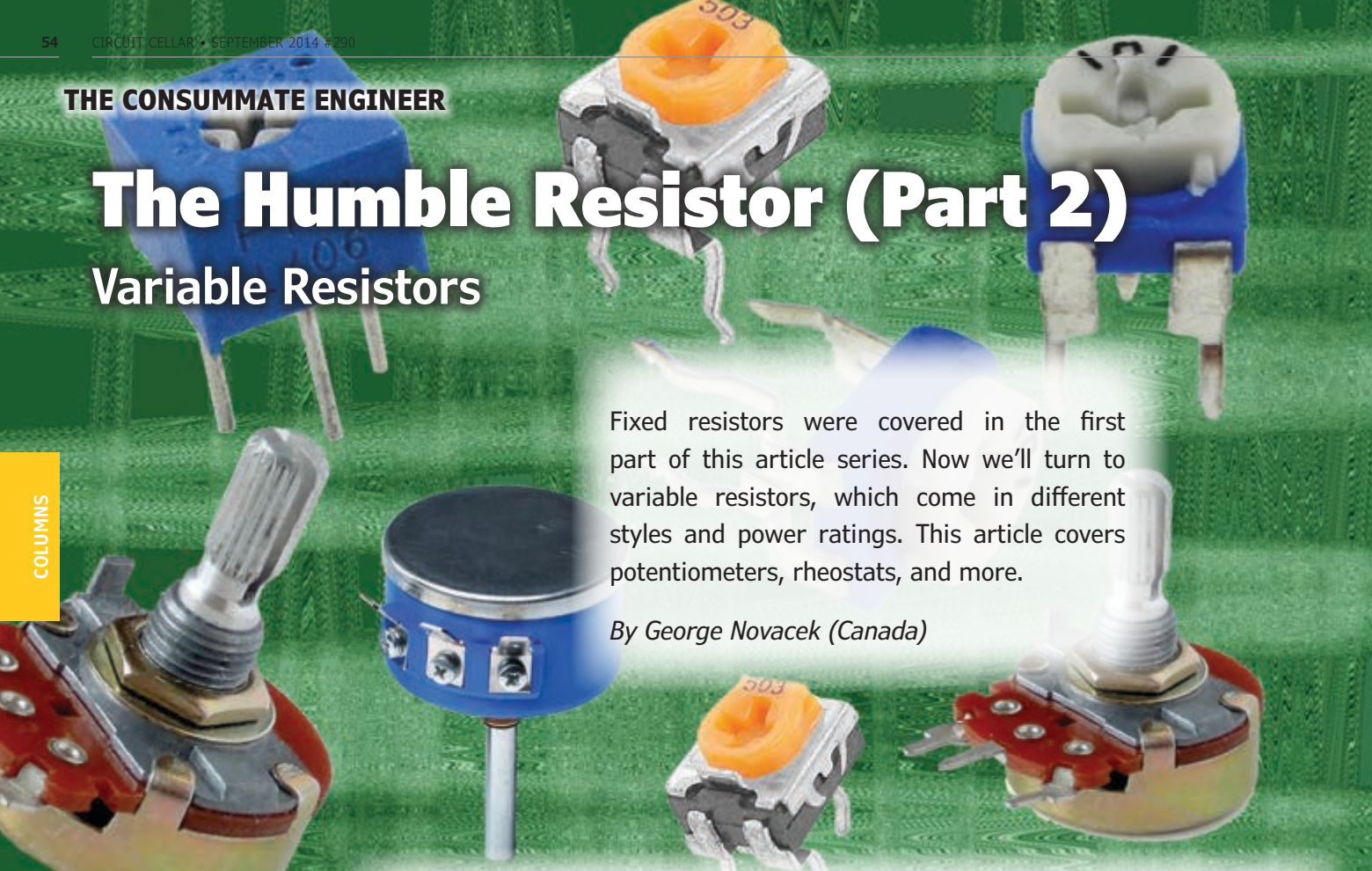
A potentiometer, or "pot," has three terminals and perhaps some taps along the resistive element. Potentiometers are used to vary output voltage by the wiper position on the resistive element (see **Figure 1**). Potentiometers should not be loaded—that is, the load resistance should be at least 10× that of the potentiometer. However, potentiometers can be loaded intentionally to modify their otherwise linear output. Most potentiometers' output is linear with respect to the position of the wiper on the resistive element. But there are potentiometers—such as those for audio volume control—with,

typically, logarithmic characteristics and a tap or taps for frequency compensation at low audio levels.

In terms of their mechanical arrangement, there are potentiometers with a shaft to be fitted with a knob for frequent adjustments, such as the volume control. Multiple potentiometers (i.e., two or more pots on a single shaft) are used in stereo equipment or dual-redundant control systems. Some are rotary. Others, such as those popular in professional audio equipment, have a linear sliding track. An important characteristic of multiple potentiometers is the accuracy of their mutual tracking.

Potentiometers referred to as trimmers, trimpots, or presets mostly have just a slot for a screw driver to adjust their outputs. They are usually set only once to trim a parameter, such as an amplifier offset or a reference voltage. Typical pots are made of a plastic base covered with a resistive track. The wiper rotates along the track. All rotary, single turn potentiometers and trimmers have about 270° rotational angle. Depending on the quality grade, some potentiometers' resistive elements are graphite, plastic with embedded carbon particles, or cermet. Wire-wound potentiometers are used when high quality or power dissipation is desired.

Multiturn potentiometers, instead of the typical 270° travel of single-turn pots, require five to 20 turns for the full range. This enables you to set the desired output more accurately



and with much greater long-term stability than single-turn devices. Some multiturn pots have a linear resistive element while the wiper moves across it by a lead screw. Others, mainly wire-wound pots, have a helical resistive element and the wiper moves across the helix.

RHEOSTATS

Rheostats adjust current flowing through a circuit (see **Figure 2**). Simply put, their resistance is added to the load, thus modifying the current through it and, in effect, the voltage across the load. Before power semiconductors, such as thyristors, made their way into power control, high-power rheostats had been used to dim lights in theaters or control the speed of electric motors.

Rheostats need only two terminals, although potentiometers with three terminals can be wired as rheostats too, provided their power rating is adequate. As you can see in Figure 2, in the presence of the third terminal, it is a good idea to connect it to the wiper as shown. Wiper contacts can become flakey due to the wear and tear, vibration, corrosion, and so forth. Because the rheostats' resistance when used to trim value of a current is often only a small percentage of the total resistance, a small overall resistance increase in case of the wiper contacts failure may be preferred to complete loss of the current.

This is an important consideration for all trimmers, be it potentiometers or rheostats. Their adjustment range should not be any greater than needed. For example, an integrated circuit (IC) analog amplifier, due to manufacturing tolerances, may require some adjustment of its bias current. A 50-k Ω resistance has been calculated to provide the nominal current value. You calculate the needed adjustment that can be achieved with a 1-k Ω pot. Don't use a larger one! You'll also have to analyze the repercussions of the potentiometer's wiper failing and thus the bias then determined by an additional 1 k Ω . You may find that the probability of this happening is very small or that the effect is not serious. Otherwise, you may take additional steps to minimize the effects of the trimmer failing or consider some other way of trimming.

TIPS

Packaging is very important for potentiometers of all kinds. It has several fundamental functions: it provides reliable means of varying the resistance, mechanical stability, mounting of the potentiometer assembly, and protection from the elements. In terms of quality, you usually get what you



PHOTO 1

Various potentiometers and trimmers

pay for.

I try to limit the use of variable resistor trimmers. They are simple to implement, small, some types are fairly inexpensive, but their reliability is not great compared with other resistive components and methods of trimming. Their setting can change with age due to vibration or accidental movement. When I do use them, I minimize their adjustment range to the absolute minimum needed. Often a series/parallel combination of fixed resistors can perform the trim with excellent long term stability.

Potentiometers are still used in feedback control systems for position command as well as feedback. For critical applications, rotary or linear variable differential transformers (RVDT or LVDT), or some of many digital encoders are, in my experience, better, more reliable solutions. It depends on the application, its safety requirements, and the cost.

I like to use solid-state potentiometers. They have many advantages over their mechanical brethren, such as stability, accurate tracking and so forth. You can find them in a lot of audio equipment in volume and tone control circuits. Essentially, they are a string of resistors with taps connected through solid-state switches to the output. A number of manufacturers make them, such as Intersil, Maxim Integrated Products, and Analog Devices. They have some idiosyncrasies compared to electromechanical resistive products. For instance, they do not provide

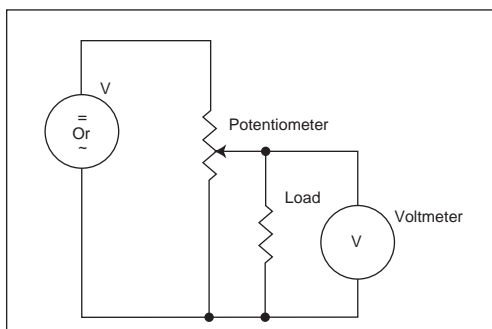


FIGURE 1

Potentiometer usage



ABOUT THE AUTHOR

George Novacek 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. Contact him at gnovacek@nexicom.net with "Circuit Cellar" in the subject line.

the "infinitely fine" adjustment of pots, but one can usually work around those. Read their specifications carefully. One criticism I heard was that the volume or tone adjustment was unacceptably coarse. I don't have "golden ears," but the alleged shortcoming seems to be hardly worrisome. The dynamic range of human hearing is about 140 dB. Live music range does not exceed 80dB. The electrically reproduced music's range is less; but let's assume it's also 80 dB. Humans can discern an approximately 2-dB change in volume. Thus, 1,024 steps available in many digital pots should provide 0.078 dB steps, which is more than sufficiently fine adjustment for

even the pickiest music aficionado.

Another peculiarity of the digital pots is that they need a DC operating voltage, typically 2.7 to 5.25 V. None of the pot's terminals can be exposed to a voltage outside this limit.

In some microelectronic devices (e.g., Micrel QwikRadio™ IC receivers), switched capacitor technology is used to simulate resistors. Also, FETs can act like variable resistors when their drain to source voltage is kept within their specified "resistive region." In other words, the drain current modulated by the gate voltage is reasonably linearly proportional to the drain voltage. This can be used, for example, to modulate operational amplifier gain. Similar nonlinear dependencies can be found in PN junctions.

THERMISTORS & VARISTORS

When discussing variable resistors we must not forget thermistors and varistors. Thermistors are temperature-dependant resistors. Most common thermistors, called NTC, have a negative temperature coefficient. Their resistance diminishes with temperature. They are mostly made of sintered metal oxide, and their common use is for temperature sensing.

Posistors, or PTCs, have positive temperature coefficient. Their resistance grows with temperature, and they're used, for example, to stabilize operating characteristics of circuits or as fuses. They are made of doped polycrystalline ceramic containing barium titanate (BaTiO₃). There are other types of temperature sensors, such as Silistors, made of silicone. Unlike thermistors, their resistance changes linearly with temperature.

In a harsh environment, such as a jet engine, the temperature is sometimes measured by resistance temperature detectors (RTDs) in addition to thermocouples. RTDs are essentially wire-wound resistors made out of platinum, nickel, or copper wire, with a highly predictable temperature characteristic/resistance. You can use them up to about 1,112°F (600°C).

Metal-oxide varistors (MOV) are made typically of zinc-oxide or silicon carbide. They begin to conduct current when the clamping voltage (from volts to kilovolts) of either polarity is exceeded. They are used mostly in commercial equipment for transient and overvoltage protection.

NOT SO HUMBLE


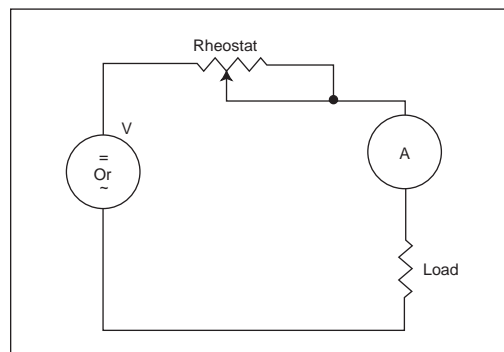
So that's it for variable resistors. Come to think of it, none of the resistors we discussed were particularly "humble." 

FIGURE 2
Rheostat usage



circuitcellar.com/ccmaterials

Intersil, "Digital Potentiometers(DCPs)," www.intersil.com/en/products/data-converters/digital-potentiometers--dcps-.html.

Maxim Integrated Products, "10-Bit, Dual, Nonvolatile, Linear-Taper Digital Potentiometers," 19-3562, 2006, <http://datasheets.maximintegrated.com/en/ds/MAX5494-MAX5499.pdf>.

TT Electronics, "Ultra-High Value Precision Resistors," www.welwyn-tt.com/pdf/data-sheet/3810.PDF.

Vishay, "Metal Film Resistors, Military/Established Reliability, MIL-PRF-39017 Qualified, Type RLR," 31023, 2013, www.vishay.com/docs/31023/erl.pdf.

RESOURCES

Analog Devices, "AD5227: 64-Position Digital Up/Down Control Potentiometer," www.analog.com/en/digital-to-analog-converters/digital-potentiometers/ad5227/products/product.html.

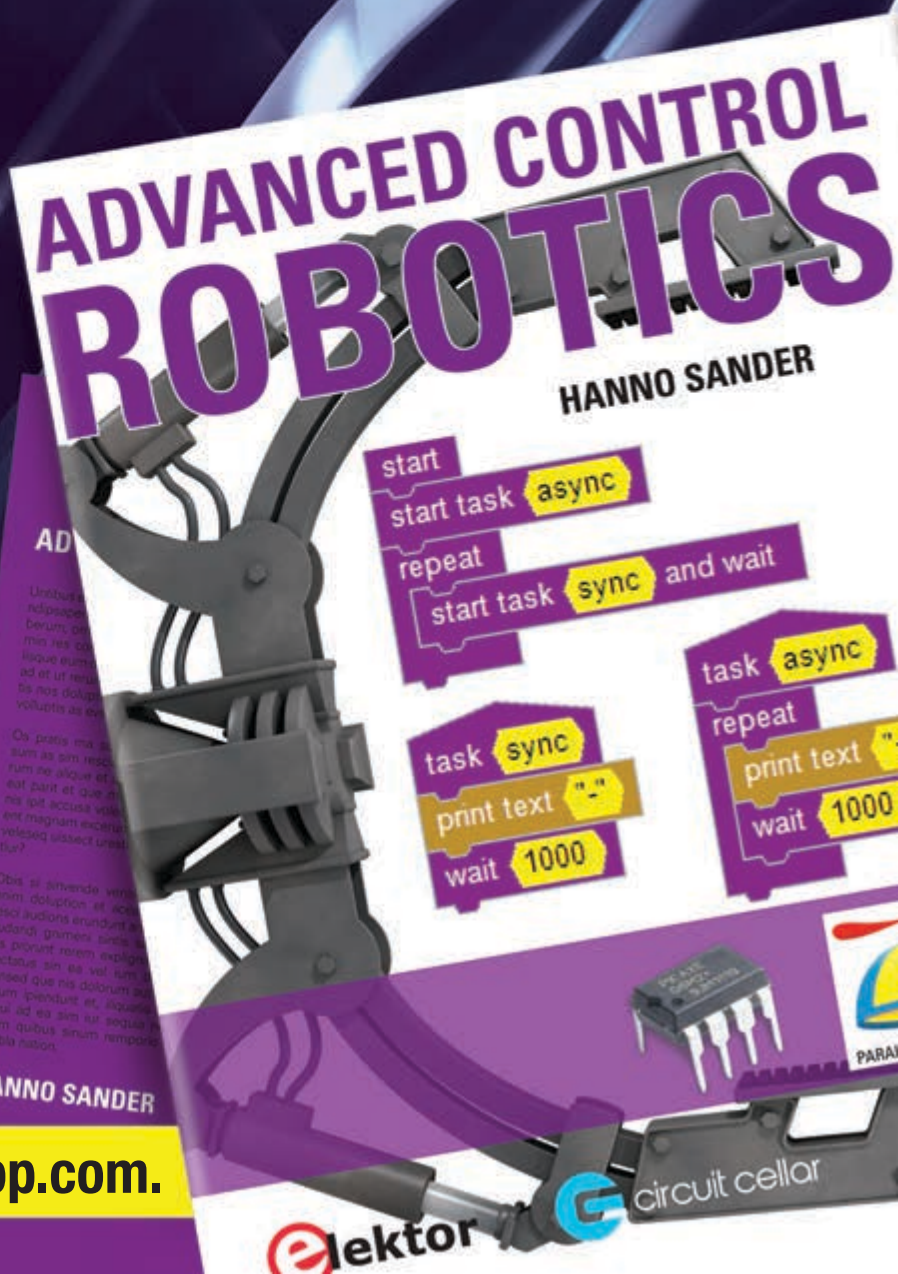
When it comes to robotics, the future is now!



From home control systems to animatronic toys to unmanned rovers, it's an exciting time to be a roboticist. *Advanced Control Robotics* simplifies the theory and best practices of advanced robot technologies, making it ideal reading for beginners and experts alike. You'll gain superior knowledge of embedded design theory by way of handy code samples, essential schematics, and valuable design tips.

With this book, you'll learn about:

- Communication Technologies
- Control Robotics
- Embedded Technology
- Programming Language
- Visual Debugging... and more



Get it today at ccwebshop.com.

ABOVE THE GROUND PLANE

Improved Arduino PWM MOSFET Gate Drive

COLUMNS

Two heads often produce better projects than one, as shown by a reader's commentary on one of Ed's previous articles.

By Ed Nisley (US)

Shortly after the March 284 *Circuit Cellar* appeared reader Carl Hage sent an evaluation of my Hall-effect LED current control circuitry and firmware. Although

readers often comment on my design decisions (and, alas, blunders), they rarely send half a dozen pages of detailed analysis! Hage has both a killer LED headlight on his

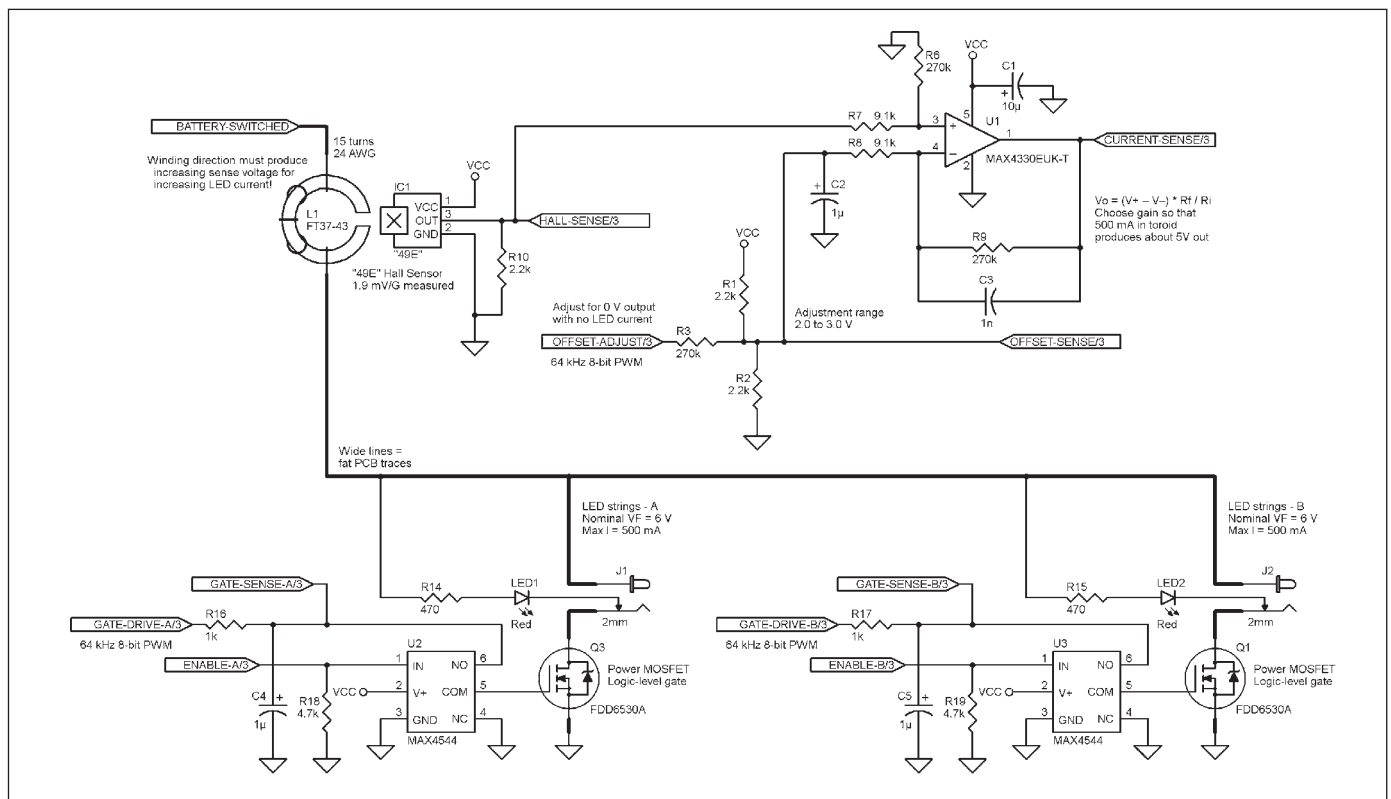


FIGURE 1

Adding Maxim Integrated MAX4544 analog switches to the schematic from my "Arduino PWM vs MOSFET Transconductance" (*Circuit Cellar* 284, 2014) article improves the MOSFET gate waveforms. The analog switches connect the MOSFET gates to either ground or the PWM filter output, so the pulse edges no longer depend on the filter response. The code can also change the PWM voltage with the gate grounded, thus allowing more time for the voltage to stabilize.

```
// Timer 1: PWM 9 PWM 10 - Hall offset
  TCCR1A = B10000001; // Mode 5 = fast 8-bit PWM with TOP=FF
  TCCR1B = B00001001; // ... WGM, 1:1 clock scale -> 64 kHz

// Timer 2: PWM3 PWM11 - MOSFET gate drive A, B
  TCCR2A = B10100011; // Mode 3 = fast PWM with TOP=FF
  TCCR2B = B00000001; // ... 1:1 clock scale -> 64 kHz

analogWrite(PIN_SET_VGATE_A,0); // force gate voltage = 0
analogWrite(PIN_SET_VGATE_B,0);
```

LISTING 1

The Arduino's default counter-timer setup produces relatively slow PWM. These statements set 1:1 clock division and 8-bit Fast PWM Mode for Timer1 and Timer2, and then set their output voltage to zero.

bicycle and deep knowledge of Atmel's AVR microcontrollers, so his suggestions obviously came from experience.

In this article, I'll take a second look at a design I'd considered nearly finished and, by blending some of Hage's suggestions with a few tweaks of my own, explore how the result improves on the original design.

As before, the circuit uses a Hall-effect sensor to measure the current flowing from the battery to the LEDs, then controls that current with PWM outputs that set the MOSFET gate voltages. A single-supply op-amp removes the sensor's $V_{CC}/2$ offset and amplifies the result for the Arduino's analog inputs. **Figure 1** presents the key part of the revised circuit; you can download the entire schematic from *Circuit Cellar's* FTP site.

FASTER PWM = LOWER RIPPLE

My original design selected the 1:1 clock prescaler for Timer1 and Timer2, which produced a 32- μ s PWM period. Hage pointed out that the default Arduino PWM configuration uses the dual-slope Phase-Correct mode that works well for motor drivers, but runs at half the maximum possible rate. In contrast, Fast PWM mode uses a single-slope comparison that instantly doubles the PWM frequency.

I'd used Fast PWM mode in other projects for exactly that reason, so his comment produced an immediate face palm: why didn't I think of that?

Listing 1 shows the statements that set both timers into Fast PWM Mode with 1:1 clock prescalers. Although Timer1 can operate with 8, 9, 10, or 16 bits of resolution, the Arduino firmware supports only 8 bits and I left it that way for reasons I'll discuss later.

I'd used a frequency-domain analysis to figure the ripple at the output of the PWM filters, with a 1-k Ω resistor and 1- μ F capacitor producing a 1-ms time constant that corresponds to a 160-Hz cutoff. Because an RC filter has one pole, the response rolls off at 6 dB/octave: doubling the frequency reduces the ripple by 6 dB. Hage suggested using the time-domain equivalent: applying

1 mA through the resistor will change the output by 1 V/ms. Reducing the PWM period by a factor of two reduces the voltage change by a factor of two: exactly -6 dB.

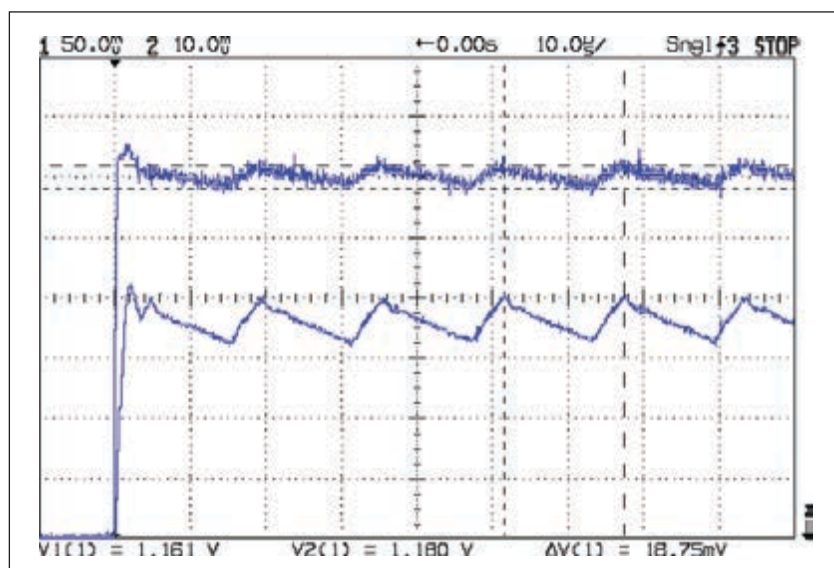
No matter how you figure it, doubling the PWM frequency or halving the period should reduce the ripple by about a factor of two. Comparing the 100 mA of ripple in Photo 3 from my March article ("Arduino PWM vs MOSFET Transductance") with the 40 mA of ripple in **Photo 1** here shows that the reduction is about a factor of two. A close look shows that slightly different PWM values in the two photos affect the exact ripple amplitude.

In all of the oscilloscope photos, the LED current trace comes from a Tektronix AM503 Hall-effect current probe scaled at 50 mA/div, with a bandwidth far exceeding my circuitry: you can depend on those traces!

The Arduino timer configuration allows only 256 PWM values: 0 through 255, corresponding to the range of each timer's eight bits, with about 20 mV separating successive PWM values at the filter output. As I showed in my March article, a 20-mV change in gate voltage produces a 50-mA change in drain current when the MOSFET

PHOTO 1

Less than 20 mV of ripple in the PWM filter voltage (upper trace) produces 40 mA of LED current variation (lower trace, 50 mA/div). Doubling the PWM frequency to 62.5 kHz reduced the current ripple by about a factor of two. The 0-V level for the gate voltage trace is far off screen, with the cursor readout showing its 1.2-VDC level.



Elektor.STORE

The world of electronics
at your fingertips!



NEW!

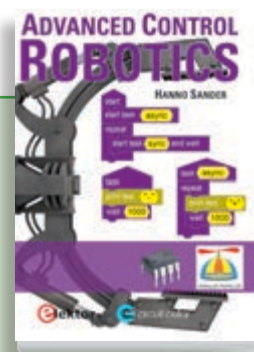
Fun to build and use projects!

Create 30 PIC Microcontroller Projects with Flowcode 6

This book covers the use of Flowcode® version 6, a state-of-the-art, all-graphical based code development tool, for the purpose of developing PIC microcontroller applications at speed and with unprecedented ease. Without exception, the 30 projects in the book are fun to build and use. A secret doorbell, a youth deterrent, GPS tracking, persistence of vision (POV), and an Internet webserver are just a few examples of projects in the book waiting to be explored and mastered. This makes the publication a perfect source of projects constantly challenging your hardware and software skills as you progress, resulting in advanced microcontroller applications you can be proud of. All sources referred in the book are available for free download, including the support software.

232 pages • ISBN 978-1-907920-30-1 • \$48

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



Theory and best practices

Advanced Control Robotics

It doesn't matter if you're building a line-following robot toy or tasked with designing a mobile system for an extraterrestrial exploratory mission: the more you know about advanced robotics technologies, the better you'll fare at your workbench. Hanno Sander's Advanced Control Robotics simplifies the theory and best practices of advanced robot technologies. You're taught basic embedded design theory and presented handy code samples, essential schematics, and valuable design tips (from construction to debugging).

160 pages • ISBN 978-0-963013-33-0 • \$54



The RPI in Control Applications

Raspberry Pi Hardware Projects

This book is about the Raspberry Pi computer and its use in control applications. Dogan Ibrahim explains in simple terms, with examples, how to configure the RPI, how to install and use the Linux operating system, how to write programs using the Python programming language and how to develop hardware based projects.

290 pages • ISBN 978-1-907920-29-5 • \$54

Books



Bestseller!

Learning to Fly With Eagle

Eagle V6 Getting Started Guide

The book is intended for anyone who wants an introduction to the capabilities of the CadSoft's Eagle PCB design software package. The reader may be a novice at PCB design or a professional wanting to learn about Eagle, with the intention of migrating from another CAD package. After reading this book while practicing some of the examples, and completing the projects, you should feel confident about taking on more challenging endeavors. This book is supplied with a free copy of Eagle on CD-ROM for MS Windows, Linux and Mac.

208 pages • ISBN 978-1-907920-20-2 • \$47



The Ultimate Guide!

The LTSpice IV Simulator

In a sturdy, hard-cover format, The LTSpice IV Simulator describes the operation of the program, all available commands, the various editors, dealing with SPICE models, the use of non-linear components and more. This book is more than just a manual. It also offers a variety of tips, methods and examples, all carefully illustrated using almost 500 drawings, diagrams and screenshots on high-quality paper. The book is designed so that it is suitable for both beginner and veteran SPICE users.

744 pages • ISBN 978-3-89929-258-9 • \$67



110 Elektor Editions, Over 2500 Articles

DVD Elektor 2000 through 2009

This DVD-ROM contains all circuits and projects published in Elektor magazine's year volumes 2000 through 2009. The 2500+ articles are ordered chronologically by release date (month/year), and arranged in alphabetical order. A global index allows you to search specific content across the whole DVD. Every article is printable using a simple print function. This DVD is packed with ideas, circuits and projects that are ideal for any electronics enthusiast, student or professional, regardless of whether they are at home or elsewhere.

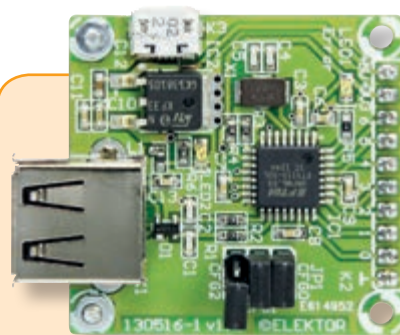
ISBN 978-1-907920-28-8 • \$121

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/store



Elektor US
111 Founders Plaza, Suite 300
East Hartford, CT 06108
USA
Phone: 860.289.0800
Fax: 860.461.0450
E-mail: order@elektor.com



Android Breakout Board

The FTDI FT311D is a flexible bridge that can interface your circuit to an Android smartphone or tablet. This Elektor Android Breakout Board offers options for seven digital outputs, four PWM outputs, asynchronous serial and I2C and SPI interfaces. The board is compatible with Android 3.1 (Honeycomb) or higher (Android Open Accessory Mode should be supported).

Ready-built module

Art.# 130516-91 • \$41



New!

IO-Warrior Expansion Board

Don't throw out your old PCs and notebooks or leave them gathering dust in the basement! They can be a useful resource: by adding this universal interface card an old PC can be pressed into service as a measurement and control hub. An IO-Warrior module on the I/F board takes care of USB communication, and source code is available that works with the free version of Visual Studio.

Ready-built module

Art.# 130006-91 • \$54

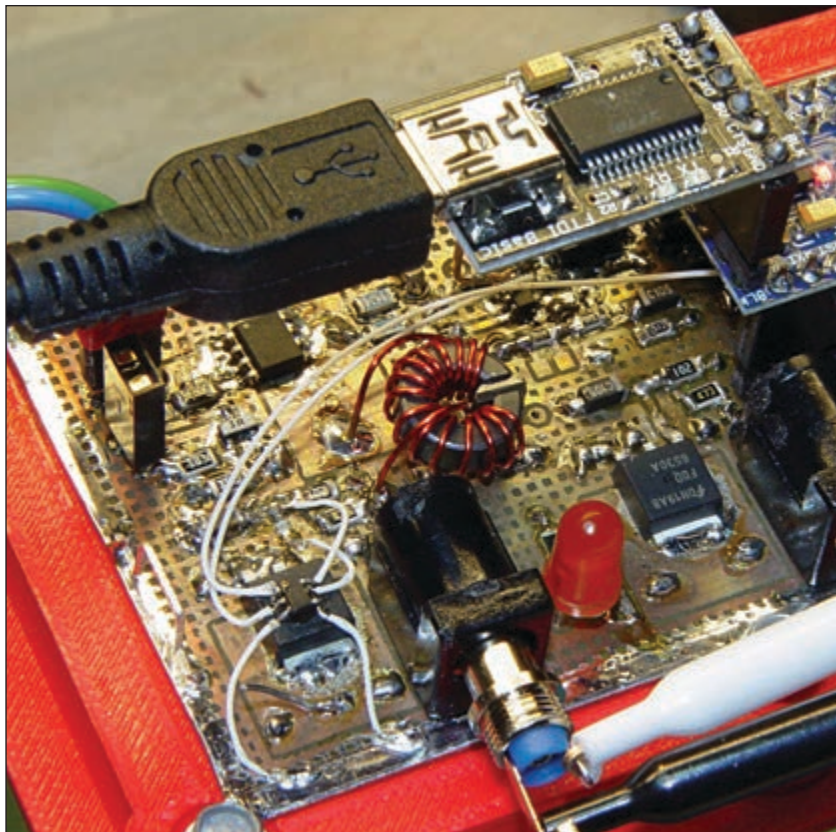


PHOTO 2

A Maxim Integrated MAX4544 epoxied atop the MOSFET with air-wired connections validated the idea, but it certainly won't survive actual use! The MOSFET on the right side hasn't been modified yet.

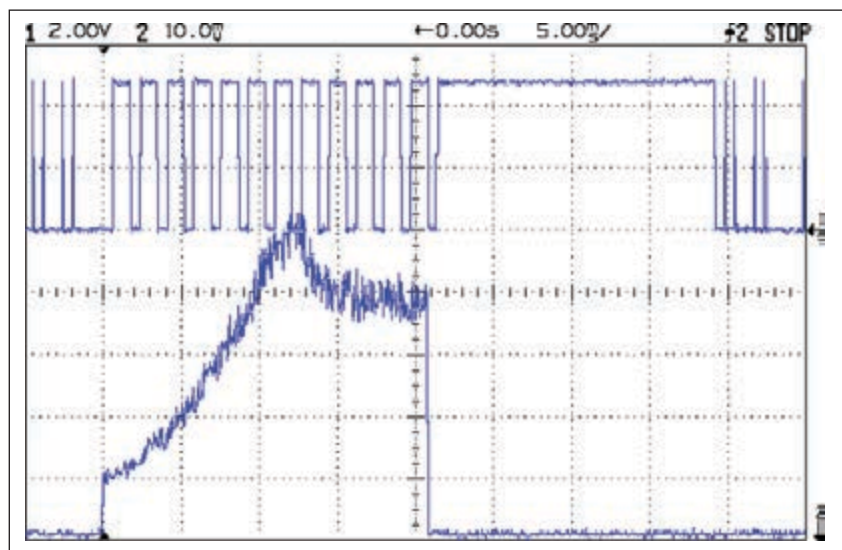


PHOTO 3

The lower sections of the upper trace show when the firmware measures the LED current, with the PWM changing by one count immediately afterward. The main loop's path length limits the maximum update rate to 1.7 ms, so slewing the LED current by 150 mA requires 10 ms.

conducts about 200 mA. Obviously, increasing the number of PWM values would enable finer control of the LED current.

The Timer1 hardware can operate with more than eight bits, but each additional bit reduces the PWM frequency by a factor of two. Given the same PWM filter hardware, each additional bit increases the ripple by a factor of two, giving finer control of an output with more variation. Because I doubled the PWM frequency, another bit of resolution would produce the same ripple as before.

Hage suggested a solution that shows his familiarity with wringing ultimate performance from limited microcontroller hardware: use firmware to dither the PWM value while the timer runs! For example, an 8-bit timer cannot produce a non-integral PWM value of 64.25, but running a PWM value of 64 for three periods and 65 for one period yields an average of 64.25 and the corresponding "fractional" voltage from the filter.

He suggested setting the hardware timer to count 5 bits, then using firmware to dither the counter value of eight successive periods. In effect, the firmware provides three high-order bits and the hardware provides five low-order "fractional" bits, but those high-order bits can change on the fly. Got that?

However, this requires formidable programming ability and produces extremely fragile code. As you saw earlier, the timers run from the 1:1 clock prescaler, which means they would require attention every 32 instructions. Normally, you'd use an interrupt handler, but a handler saving and restoring registers at that rate would cripple the CPU's throughput. As Hage noted, "You wouldn't want the Timer0 interrupt messing up the PWM dithering anyway, so [you] don't need Timer0 while in the PWM."

While I agree that his method is *possible*, it certainly requires more effort than seems warranted. It's the sort of code you write when confronted with changing requirements and frozen hardware: you must solve the problem without changing the BOM. I'll leave that as an exercise for the very motivated reader!

CRISP GATE VOLTAGE TRANSITIONS

Photo 2 in my March article showed the relatively slow transitions on each edge of the LED current pulse due to the filtered PWM at the MOSFET gate. Hage suggested a solution that draws on the power of a microcontroller: calculate the transition time based on applying a constant 0 or 5 V, turn off the PWM, set the output bit low or high for the appropriate time, and then restart the PWM. The filter

```

if (millis() >= (EventStart + (unsigned long)Events[EventIndex].duration)) {
    EventStart = millis(); // record start time

    Events[EventIndex].drive_a = VGateDriveA; // save drive voltages
    Events[EventIndex].drive_b = VGateDriveB;

    if (++EventIndex > MAX_EVENT_INDEX) // step to next event
        EventIndex = 0;

    VGateDriveA = Events[EventIndex].drive_a; // restore previous drives
    SetPWMPinVoltage(PIN_SET_VGATE_A, VGateDriveA);

    VGateDriveB = Events[EventIndex].drive_b;
    SetPWMPinVoltage(PIN_SET_VGATE_B, VGateDriveB);

    delay(PWM_Settle);

    digitalWrite(PIN_ENABLE_A, Events[EventIndex].en_a); // enable gates for new state
    digitalWrite(PIN_ENABLE_B, Events[EventIndex].en_b);

    if (!(Events[EventIndex].en_a || Events[EventIndex].en_b))
        FindSensorNull(); // both LEDs now off -> null sensor
}

```

LISTING 2

This code times each pulse, stores the PWM drive voltages, and fetches the next outputs. When both LEDs will be off, it nulls the Hall-effect sensor output.

would limit the slew rate, but the transition time would still be much shorter than waiting for the filtered PWM voltage.

Alternatively, rather than reducing the PWM all the way to zero, leave it slightly below the MOSFET's 0.9-V threshold to reduce the overall transition time. Hage also suggested turning the output on, monitoring the LED current until it approaches the desired value, then enabling the PWM output.

I decided to solve the problem with solder, by splicing a single pole, double throw (SPDT) analog gate between the PWM filter and the MOSFET gate (see **Figure 1**). The normally closed input connects to 0 V and turns the MOSFET completely off, with the pull-down resistor on the Arduino control signal ensuring that the pin stays low until the firmware gains control after a hardware reset.

The gate voltage in the upper trace of **Photo 1** rises from 0 to about 1.2 V in less than 1 μ s and the corresponding drain current transition in the lower trace requires about 1 μ s. The direct PWM drive solution I'd been using required nearly 10 ms for the same transitions.

Photo 2 shows that the hardware doesn't look quite as impressive as its output. Rather than build an entirely new PCB, I simply epoxied a Maxim Integrated

```

struct event {
    boolean en_a; // true = enable LED A output
    boolean en_b; // true = enable LED B output
    word duration; // milliseconds until next event
    float current; // amps - total current in LED A + B
    float drive_a; // volts - previous drive for LED A
    float drive_b; // ... LED B
};

event Events[] = {
    {true, false, 10, 0.200, 1.0, 0.0},
    {false, false, 20, 0.000, 0.0, 0.0},
    {true, false, 10, 0.150, 1.0, 0.0},
    {false, false, 20, 0.000, 0.0, 0.0},
    {true, false, 10, 0.100, 1.0, 0.0},
    {false, false, 20, 0.000, 0.0, 0.0},
    {true, false, 10, 0.050, 1.0, 0.0},
    {false, false, 20, 0.000, 0.0, 0.0},
    {true, false, 10, 0.025, 1.0, 0.0},
    {false, false, 370, 0.000, 0.0, 0.0}
};

#define MAX_EVENT_INDEX ((sizeof(Events)/sizeof(struct event)) - 1)

```

LISTING 3

An array of structures defines each LED pulse's current and duration. The pulse generation code saves the PWM output required to produce each pulse and restores it when that pulse happens again.

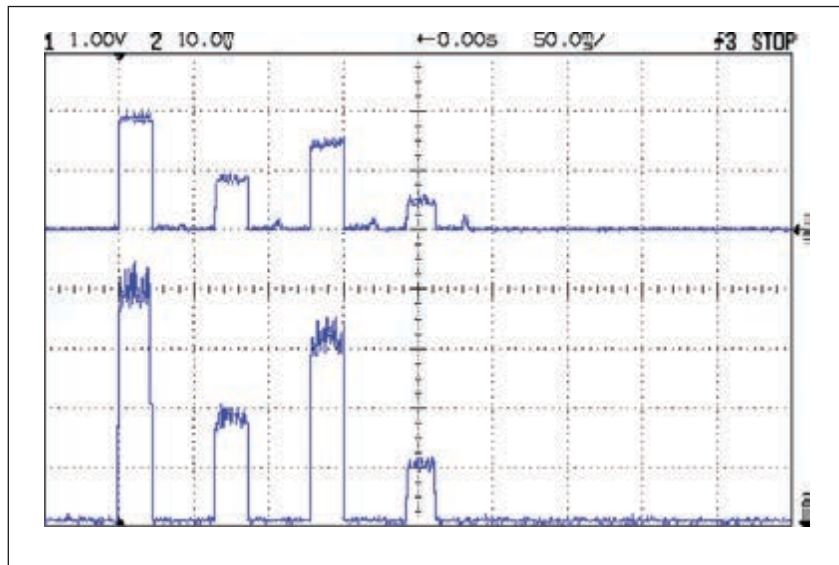


PHOTO 4

Presetting the PWM outputs at the start of each pulse produces sharp-edged LED blinks. The op-amp output in the top trace scales the Hall-effect current sensor to about 1 V/100 mA, with the small bumps between pulses showing the sensor null operation. The lower trace shows the actual LED current at 50 mA/div, with the PWM ripple clearly visible.



ABOUT THE AUTHOR

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

PROJECT FILES



circuitcellar.com/ccmaterials

——, Softsolder.com, "Hall-Effect Current Control PCB: First Light," <http://softsolder.com>.

——, Softsolder.com, "Hall-Effect Current Control PCB: Voltage Variations," <http://softsolder.com>.

SOURCES

MAX4544 SOT23-6 Package
Maxim Integrated | www.maximintegrated.com

AM503 AC/DC current probe amplifier
Tektronix, Inc. | www.tek.com

RESOURCES

E. Nisley, "Arduino PWM vs MOSFET Transductance" *Circuit Cellar* 284, 2014.

——, "Low-Loss Hall-Effect Current Sensing," *Circuit Cellar* 280, 2013.

MAX4544 SOT23-6 package atop the MOSFET, cut the gate trace, and soldered the airwires. This obviously won't withstand actual use, but it suffices for the prototype and shows that the idea works well. The second MOSFET to the right of the red LED remains unmodified.

Although the analog switch produces crisp transitions, it can't improve the speed of gate voltage changes. The main loop measures the LED current and adjusts the gate voltage by one PWM count on each iteration, which limits both the maximum slew rate and the loop's response to overshoot. The lower trace in **Photo 3** shows a single LED pulse, with the current starting at 50 mA and ending at just under 200 mA; the slew rate and overshoot are painfully obvious.

However, at the end of each pulse, the firmware has adjusted the current close to the desired level, so I modified the code, as shown in **Listing 2**, to save that PWM value and restore it when the pulse occurs again. The first `if` statement detects the end of the pulse, saves and restores the PWM values that control the gate voltages, and then enables the appropriate analog switches. The `delay()` function allows time for the PWM filters to settle at their new levels, so that the gates see a step transition when the analog switch selects a different input.

PULSE SHAPING

Producing a single blink requires little more than two timing loops that produce the on and off durations, but that code doesn't scale well for a complex sequence of blinks with different durations, amplitudes, and delays. The easiest way to generate a series of events is to put the complexity in a table (or array) and use a simple routine that steps from one entry to the next.

The `struct` event in **Listing 3** defines the values required to produce each pulse, including its duration and total current, as well as the delays between successive pulses. Because the hardware includes one current sensor and two MOSFET drivers, it can sense only the total current in both LED strings. The `en_a` and `en_b` values determine which outputs will be active during each pulse. If both are active, the firmware must adjust both gate voltages.

Each entry of the `Events[]` array is a single `struct` pulse defining a single pulse or delay. The first `if` statement in **Listing 2** steps the `EventIndex` variable to the next entry at the end of each pulse.

The Arduino runtime copies the values forming the `Events[]` array into RAM during startup, so you could change the pulse durations and currents on the fly.

Unfortunately, that means a random glitch can trash the array. Good practice suggests splitting it into two arrays: one in flash ROM for the fixed constants that define the pulses and another in RAM for the variables holding the dynamic PWM values.

Homework: make it so!

Photo 4 shows the complete solution in action, with the Hall-effect sensor output in the top trace matching the actual LED current (from the Tek probe) in the bottom trace, albeit with a lower bandwidth that suppresses the PWM current ripple. The analog switch produces crisp edges with the preset PWM values requiring little or no adjustment at the start of each pulse.

The small bump following each pulse in the top trace occurs when the code adjusts the offset value that nulls the Hall-effect sensor output. When both LED outputs are off and no current flows through the Hall-effect sensor, the nulling routine ensures the op-amp output is exactly zero. Even though the sensor output doesn't drift enough to require nulling after every pulse, there's no harm in doing so.

If the Hall-effect sensor output drifts below the offset value, the op-amp output remains at zero, so the code begins the nulling process by reducing the offset until the op-amp output goes slightly positive and then increasing the offset until it becomes zero again. Hage suggested nulling the output to a slight positive offset to eliminate the need for pre-adjustment, then subtracting that zero-current offset from subsequent measurements. I think either method will produce the same results, so I reused the existing routine.

Homework: implement Hage's suggestion, perhaps with a deadband around the setpoint.

With all that in mind, the code in **Listing 4** closes the current feedback loop. On each iteration of the main loop, it reads the LED current and, if the values fall outside the deadband, adjusts the gate voltages upward or downward to compensate. Because V_{Step} equals the voltage corresponding to a single PWM count, the gate voltage changes by the smallest possible amount.

I reduced the deadband's value to 25 mV in **Photo 5**, which shows the loop hunting back and forth on each side of the 200-mA setpoint. In this situation, none of the possible gate voltages produce an LED current that lies inside the deadband, so the current alternates between 200 and 250 mA on successive passes through the main loop.

I think the value of I_{Gain} that converts the op-amp output into current is a bit too low, because the current should settle at 200 mA. Fixing that is a simple matter of software, of course.

```
ILEDsense = ReadCurrent();           // Sample LED current

// Adjust gate drives to maintain LED current setpoint

if (ILEDsense < (Events[EventIndex].current - IDeadBand/2.0))
    PWMStep = VStep;
else if (ILEDsense > (Events[EventIndex].current + IDeadBand/2.0))
    PWMStep = -VStep;
else
    PWMStep = 0.0;

VGateDriveA += Events[EventIndex].en_a ? PWMStep : 0.0;
SetPWMPVoltage(PIN_SET_VGATE_A, VGateDriveA);

VGateDriveB += Events[EventIndex].en_b ? PWMStep : 0.0;
SetPWMPVoltage(PIN_SET_VGATE_B, VGateDriveB);

delay(PWM_Settle);
```

LISTING 4

The LED current control feedback loop closes through the firmware's main loop, which adjusts the PWM value by one PWM count when the measured LED current falls outside the deadband around the desired setpoint.

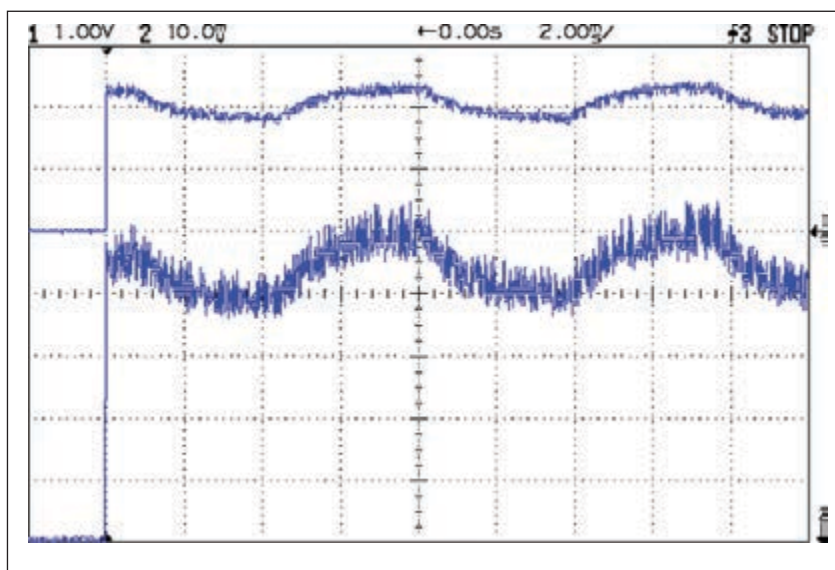


PHOTO 5

With the deadband set to 25 mA, the current control loop hunts back and forth on each iteration, because a single PWM count changes the current by about 50 mA. The top trace shows the current sense amplifier's output. The bottom trace shows the actual LED current at 50 mA/div.

CONTACT RELEASE

Hage thinks the Arduino hardware hasn't come close to its limits, but he has definitely pushed it much closer to the edge. In any event, the results you see here work better than my original design did, so I'll call it a success.

The information on Circuit Cellar's FTP site contains the new firmware and schematic; I haven't revised the PCB layout to accommodate the new parts. [E](#)

FROM THE BENCH

IR Remotes (Part 2)

IR Transmissions Explained

The first article in this series introduced IR technology and IR formats. This article details how to identify, decode, and reproduce IR transmissions.

By Jeff Bachiochi (US)

Appliance manufacturers are always looking for new ways to make their products user friendly and keep up with the latest technology trends. Lately, I've noticed more appliance manufacturers have adopted infrared (IR) technology as an inexpensive way for users to interface remotely, without having to use their local controls. While this doesn't help if you don't have the IR remote handy or lose it between the couch cushions, it does tend to make even an ordinary ceiling fan kind of techie.

Many universal remote controls (URCs) on the market offer the ability to integrate multiple remotes into a single device or just allow one to have an extra. This way, you don't have to fight over who has control of the remote. The universal remotes can handle almost any format a manufacture wants to use in its equipment. So you need only choose the secret four-digit code from a device/manufacture list and enter it into the

URC. An internal look-up table then provides the correct format parameters for the URC to be able to emulate specific IR remote. What does the list look like? For the most part, it isn't public knowledge; it's kept secret by the manufacturer. Usually, the same code on another URC will not produce the same format. There has been no standardization of these codes or the formats they represent. This also means that any new format that is introduced cannot be handled by an older URC, since the device won't be in the list. If only there was a way to describe the format in a more dynamic way, changes might be made without necessarily breaking everything. I ended last month with a short description of the infrared protocol (IRP) that attempts to do this. The protocol is well thought out and both simple and complex. A single line of text, punctuation, and values contain the complete description of an IR transmission by breaking it into three pieces: GeneralSpec, BitSpec, and IRStream.

The GeneralSpec contains basic info like modulation frequency, order of data (MSB/LSB first), and time units like microseconds or cycles. The BitSpec describes the relationship



between a bit = 1 and a bit = 0 in terms of Mark (modulated) and Space (idle) time units. While most formats use a Mark followed by a Space of one of two possible lengths to represent a single bit (1/0), some formats may use multiple bits (e.g., one of four space durations to represent 2-bit possibilities or Manchester encoding, which uses Mark/Space or Space/Mark to represent bit values). Lastly, the IRStream describes the data to be transmitted using the BitSpec format. This consists of a list of parameters, special bits, variables, and constants. Special bits might be used in a header or other identifier, while variables might be key press data and constants product identifiers. Each value isn't necessarily 8 bits, so it has a bit count associated with it and sometimes a bit starting position.

Initially, manufacturers favored transmission formats that offered minimum IR modulation times. The main object was to conserve battery life, so unnecessary bit transmissions would be wasteful. Today manufacturers are more likely to send more data per transmission. Not only is consumption less of an issue, but today's remotes are more functional than early remotes and require more data.

LEARN BY DOING

Some URCS use brute force to mimic remotes that they can't duplicate. This requires a vast amount of memory since you must save every transition of IR that you see in order to replicate it. While a single transmission may not be overburdening by itself, multiply this by the number of keys you need to mimic and you will quickly run out of memory space.

It wasn't until I was well into this project that I found the discussion of the IRP. I only needed to figure out a few remotes that I had hanging around, so I wasn't thinking in such general terms. However, as I dug through the IRP spec, it became clear that incorporating this format could ultimately simplify the code and be more useful. So, even though I had started by coding each manufacturer with its own specialized routines, I rewrote the code with a more generic twist.

The most difficult part was interpreting each IRP format. Each protocol is stored in a table as a list of characters. My first entry is for the Sharp protocol. Let's look at this IRP.

```
Sharp{38k,264}<1,-
3|1,-7>(D:5,F:8,1:2,1,-
165,D:5,~F:8,2:2,1,-165)+
```

You can easily separate this into five parts:

1. Sharp
2. {38k,264}

3. <1,-3|1,-7>
4. (D:5,F:8,1:2,1,-165,D:5,~F:8,2:2,1,-165)
5. +

Placing the protocol name (1) along with the IRP format not only makes it easy to identify the format in the code listing, but also provides identification for any display message you may be creating. Last month, I talked about the IR transmission itself, which is separated into a number of modulated and unmodulated time periods. Note that the modulation is clearly spelled out in the curly bracket section (2) as indicated by using the "k" thousand multiplier. Unless we find the string MSB in part 2, we can assume that the order is LSB first. Any value remaining will define a unit time multiplier, 264. This defaults to microseconds without any value suffix. The suffix "p" means the value is a cycle multiplier and not a time unit (picosecond) as one might suspect. A lack of any unit time should default to a value of 1.

Part 3 indicates two bit timings (separated by "|"). The first bit (zero) equals a Mark time = 1 (times the unit time multiplier) × 264, or 264 μs, and a Space time = 3 × 264, or 792 μs. The second bit (one) equals a Mark time = 1 × 264, or 264 μs, and a Space time = 7 × 264, or 1848 μs. A positive value is a Mark or modulated time, while a negative value denotes a Space time. The Sharp format is the simplest, using pulse distance modulation, a constant duration Mark (264 μs) followed by a variable duration Space 792 or 1848 μs.

Now comes the data format sequence in part 4. There are 10 elements to this data:

- Variable D consisting of 5 bits
- Variable F consisting of 8 bits
- Constant 1 consisting of 2 bits
- Mark with a length of 1 × 264 or 264 μ
- Space with a length of 165 × 264 or 43,560 μs
- Variable D consisting of 5 bits
- Complement of Variable F consisting of 8 bits
- Constant 2 consisting of 2 bits
- Mark with a length of 1 × 264 or 264 μs
- Space with a length of 165 × 264 or 43,560 μs

Each Sharp transmission is actually two nearly identical transmissions. The variable F is complemented in the second sequence. This is Sharp's way of checking that the F (function) data has been correctly received.

The last part (5) indicates whether the transmission is normally repeated as long as the key is depressed. The plus sign (+) indicates it is repeated one or more times.

MODULATION

While using an IR demodulator front

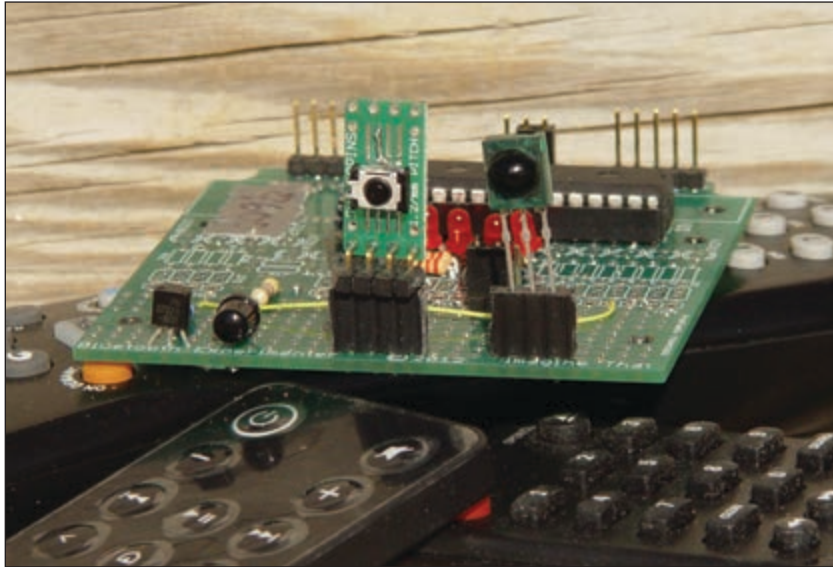


PHOTO 1

This is my prototype with (from right to left) an IR receiver demodulator, IR receiver, IR LED transmitter, and drive transistor all mounted along the front edge of the PC board.

end (e.g., Vishay Intertechnology TSOP322) simplifies your life tremendously, you do not “see” the modulation, so you can’t find out what modulation frequency your remote uses by its output. To measure the actual frequency you need a device like a Vishay

Intertechnology TSMP6000 that will sense the IR and pass the signal through without demodulating it. The TSMP6000 is an SMT part that can be easily mounted on an eight-pin SOIC SchmartBoard for easy handling (see **Photo 1**). **Figure 1** shows the project’s schematic, which has a jumper that enables you to switch between devices.

To begin capturing an IR transmission, select F (Frequency) from the menu screen (see **Photo 2**) using the serial terminal application RealTerm. A prompt instructs you to press one of the remote’s buttons (or hit any key to cancel the operation). This sampling routine waits for the first transition (a falling edge) and measures the time between falling edges (whole modulation periods). We know that all transmissions begin with modulation that last for a number of cycles. Eventually, the time measurement will exceed a minimum of, say, 33 μ s, which is 30 kHz. This sample can be tossed out and all previous samples are summed with the total being divided by the number of samples to get an average sample time. The modulation frequency is the reciprocal of this sample time.

I could have used two separate devices for this project. In fact, I began with an IR demodulator and added the TSMP6000 just to measure the frequency. The project now uses only the latter, as a demodulated signal can

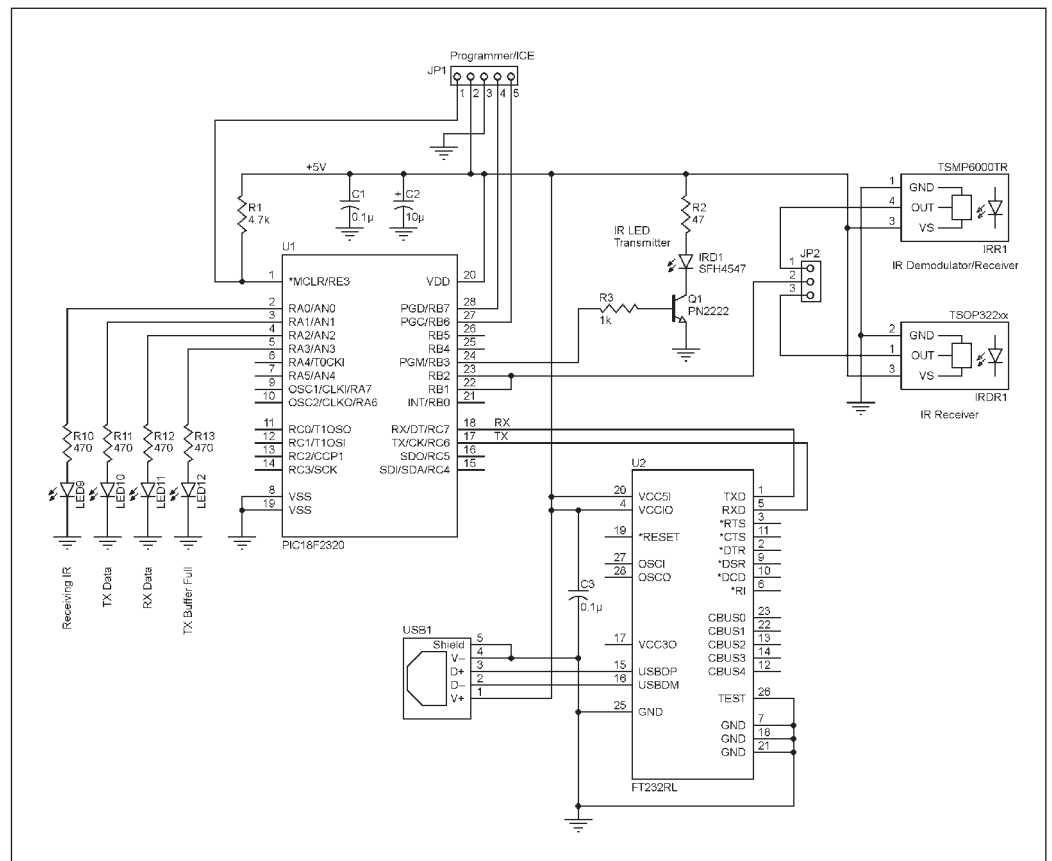


FIGURE 1

My project’s schematic includes a few LEDs that display application status along with an IR demodulator receiver, IR receiver, and IR LED. The USB connection is used to display user functions like those shown in Photo 2.

Join The Elektor Community

Become a GREEN Member Now!

Your GREEN Membership Comprises:

- 10 Editions of Elektor magazine in digital format (pdf)
- Direct access to Elektor.LABS; our virtual, online laboratory
- Direct access to Elektor.MAGAZINE; our online archive for members
- A minimum of 10% discount on all products in Elektor.STORE
- Elektor.POST newsletter sent to your email account each week
- 25 Extra Elektor projects per year (through Elektor.POST)
- Exclusive GREEN Membership card containing a state-of-the-art Mifare Ultralight RFID/NFC chip usable with NFC-compatible smartphones



EXCLUSIVE OFFER

FREE E-BOOK on AVR/Software Defined Radio with an Elektor GREEN Membership!*

Order Today at www.elektor.com/membership

* Available through www.elektor-magazine.com after you have received your magazine download login details.



Take Out Your GREEN Membership Now at www.elektor.com/green-membership

Connect with us!



www.facebook.com/elektorim



www.twitter.com/elektor

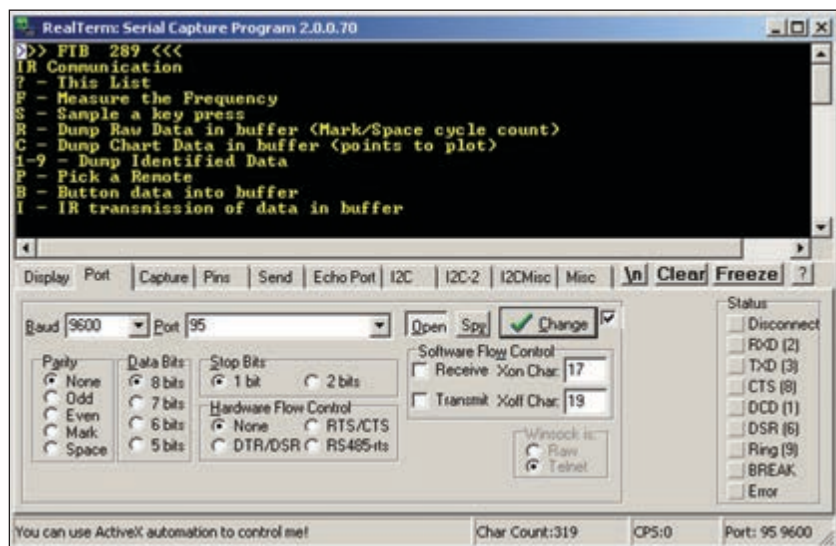


PHOTO 2

This screenshot shows the menu of functions that you can select using a serial terminal program like RealTerm via the project's serial port.

be derived through code. You can simplify the code by using both input devices if you want. Note that most IR demodulators can be purchased with their filters centered on a specific frequency; however, they can be used at other frequencies with a decrease in sensitivity.

The first check you can make on each format is the modulation frequency. This quickly eliminates protocols that use other modulation frequencies. **Photo 3** shows display output when an Acer tablet remote

was sampled for F (Frequency) and S (Sample) menu items.

FORMAT RECOGNITION

Last month, I went through the process of sampling the IR with an IR demodulator using edge triggered interrupts to measure the time periods between falling edge indicating the start of modulation (Mark) and a rising edge denotes the end of modulation and the start of the idle period (Space) using a IR demodulator. Now I'm using the TSMP6000 so I'm not seeing just the presence and absence of IR modulation, but the modulation itself. I need to do a bit more finagling to turn this input into Mark and Space timings. The frequency routine above is the key to this. That routine gave us the time for one modulation cycle. This time can be used as a window of when to expect either another modulation cycle or the lack of one. This would indicate a transition between a Mark (continuous modulation) and Space (a lack of modulation). The Mark and Space times can be adjusted slightly to account for a lack of fortune telling. The S (Sample) differs from the F (Frequency) in two ways. First, times recorded are separated into Mark times and Space times. And second, the times are not per modulation cycle, but the total consecutive modulation time and the total lack of modulation time.

Note that IR communication uses a return to zero (RTZ) data format. Thus, both a Mark and a Space is required for each data bit. The sample buffer holds records of the Mark and Space duration timing using a 4-byte sequence for each sample. "M", byteU, byteH, and byteL are for each Mark sample. "S", byteU, byteH, and byteL are for each Space sample.

To determine if an IR remote is using the Sharp protocol (or any other protocol), each protocol entry in the IRP table must be compared (one at a time) to the Sample buffer that holds the timing information of the protocol in question. To do this, an IRP is read from the table and used to build a number of buffers that will hold things like modulation rate, bit sequence, Mark and Space bit timing, and the data elements.

Every element of a protocol is made up of Marks and Spaces defined in the BitSpec section. Mark and Space times are calculated for each potential bit value. The Sharp protocol uses single bits; therefore, we have two values, bit1 and bit0. Previously we calculated these in the following way: bit0 as Mark=264 μ s and Space = 792 μ s and bit = 1 as Mark = 264 μ s and Space = 1848 μ s. This tells us that a Mark can be no shorter than 264 μ s and no longer that 264 μ s (with an average time of 264 μ s, more on this later) and a Space can

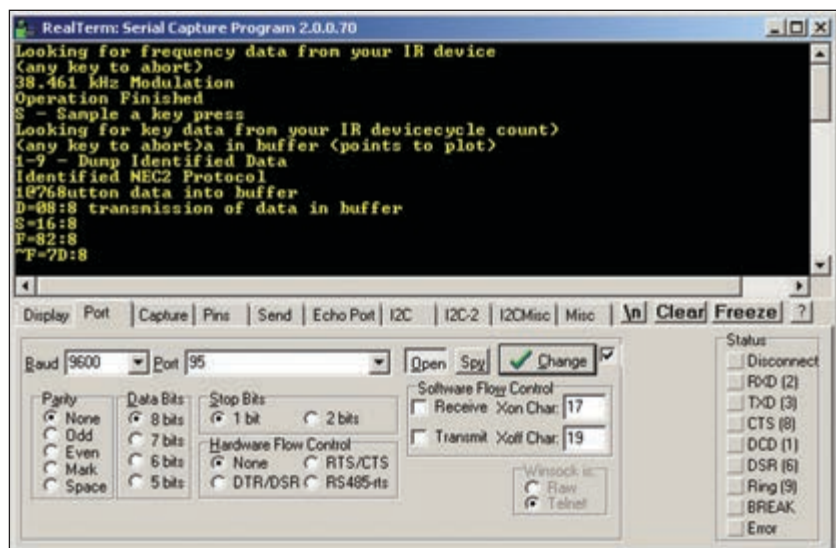


PHOTO 3

Selecting F (Frequency) will watch for an IR transmission and measure the frequency of the modulation. This can be performed with only an IR receiver that does not demodulate the received signal. Selecting S (Sample) attempts to recognize an IR transmission and proceeds to list the variable data found within that protocol according to the IRP.

```

IRPSize equ 0x80
;
IRP0      org 0x7000
data "Sharp{38k,264}<1,-3|1,-7>(D:5,F:8,1:2,1,-165,D:5,~F:8,2:2,1,-165)+",0
;
IRP1      org IRP0+IRPSize
data "NEC2{38.4k,564}<1,-1|1,-3>(16,-8,D:8,S:8,F:8,~F:8,1,-78)+",0
;
IRP2      org IRP1+IRPSize
data "Nokia32{36k,msb}<164,-276|164,-445|164,-614|164,-783>(412,-276,D:8,S:8,X:8,F:8,164,^100m)+",0

```

LISTING 1

This is a code snippet for the IRPs I placed in my application. IRP0 is for the Sharp TV. IRP1 is for the Acer tablet. IRP2 is for the AT&T cable box.

be no shorter than 792 μs and no longer than 1,848 μs (with an average of 1,056 μs). I use a tolerance value along with each expected Mark or Space time to calculate acceptable limits for each sample. If a sample falls within the calculated acceptable limits, the process proceeds with the next sample and next element (or part thereof). Any out of tolerance failure is flagged and the process for the present protocol is abandoned.

The protocol lists each element that makes up its transmission in the IRStream section. Previously we saw that the Sharp protocol uses two variable elements (D and F), a constant element, and Mark and Space elements. Mark and Space elements are straight forward and we can easily determine whether a sample time falls within a Mark or Space tolerance. But first we need to deal with the variable and constants. A variable element has a bit count associated with it. The first element D in the Sharp protocol has 5 bits. This means that we will need to check for 5 Mark and Space samples for this variable. The next element, variable F has a bit count of 8. So the next 8 Mark and Space samples must be checked to cover this element. The third element is a constant value of 1 with a bit count of 2. With a variable we don't know if any particular bit will be a bit1 or a bit0 so we must test for maximum and minimum times. With a constant we know exactly what the bits should be and where, so we can test accordingly. A value of 1, which is represented by two bits, would be 01, a bit0 followed by a bit1. Next, the protocol has a Mark element and a Space element. You may have observed that the same elements are repeated again with slight differences in variable F and the constant. Finally the + indicates that this whole shebang (transmission) could be repeated again. In fact, if sample buffer's size is sufficient, multiple transmissions will be sampled and can be decoded. One last thing to point out here. The Sharp protocol defaults to an LSB first sequence for data bits within each element. You must reverse the

bits according to the GeneralSpec section.

If we make it through a protocol without any samples being out of tolerance, we have identified the protocol. Next, we will want to know what data is produced for each button press. It turns out we are extremely close to this with the code we have just used. If we add one more routine to this process, we are there. We just need to gather the actual data we find in the sample buffer as we are going through and testing each data bit to see if it's within acceptable tolerance. It turns out we can do this within the tolerance testing routine for variables. Remember when I mentioned the average bit time above? After testing a sample for tolerance, a second test is done on the sample using the average

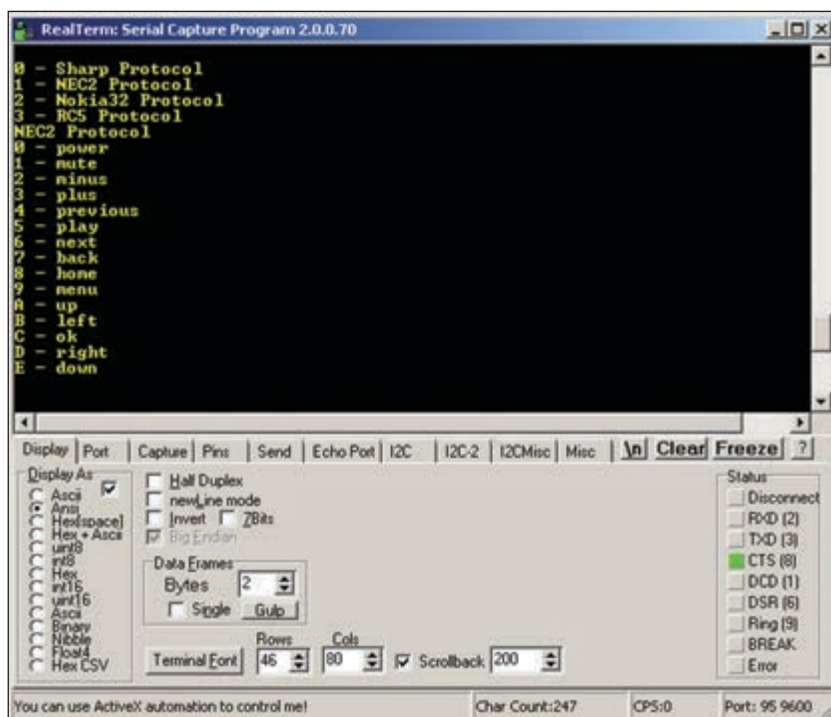


PHOTO 4

Tables for each remote were added to the application after sampling all of the remote's buttons. Here's the data listed for the Sharp TV based on sampling those buttons. Finally, you can use P (Pick) and B (Button) menu selections to make use of the data, first to pick a remote and then a button to emulate.



ABOUT THE AUTHOR

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@imagine-thatnow.com or at www.imaginatethatnow.com.

time. Any sample that is less than average is a bit0. More than average is a bit1. We just need to store this into the right variable at the right bit position and continue. Like variables should have equal data. Complemented variables should have complemented data. Now, if a protocol is identified, you already collected all the variable data that corresponds to a particular key press.

BUTTON DATA

The C (Chart) function lets me make good use of Michael Vogt's DatPlot. Using DatPlot, I can visually see what an IR device is sending and deduce which protocol I needed to support. With the general outline for IRP interpretation complete, adding additional protocols sometimes requires a minor tweak to the identification routines. I can't imagine trying to do this without the benefit of having actual hardware to work with.

Once my S routine was recognizing a protocol and reporting variable data, I needed to create a list of all the keys on the remote, use the S routine on each button to map the variable data values for each. I'll need a list for each remote I want to mimic and add it to this application to be able to reproduce each remote's button-related data transmission. The IRP describes a remote's protocol but does not indicate which keys and associated data values make up a remote (see **Listing 1**). I suppose if you don't have a remote, you could use trial

and error by sending out transmissions with each possible data value and waiting to see if it produces a function on the equipment. You may even find some undocumented function that a remote would not normally give access to! For this project, I started with a list of any buttons for a remote that produced an IR transmission for a specific protocol. For instance, my URC has a number pad and the buttons produce output for the television and not for the cable.

Producing IR requires adding an IR LED to the hardware. While the Microchip Technology PIC18F2320 microcontroller I am using can provide 25 mA of current, I added a drive transistor capable of supplying more current to the OSRAM Opto Semiconductors SFH4547 IR transmitter. This device can handle 100 mA and up to 1 A at modulated frequencies. Up to this point, I haven't specifically mentioned IR duty cycle, just modulation frequency. Early remote manufacturers noted they could improve battery life by shortening the "on" portion of the modulation duty cycle. Since current consumption is directly proportional to the amount of time an IR LED is on, reducing the duty cycle from a nominal 50% to 25% doubles battery life (at least that portion directly attributed to transmissions). While an IR receiver may still operate under these conditions, the maximum transmission distance will be affected. I am using the microcontroller's PWM function to provide a modulated output to the IR LED, so I can easily set the duty to any value of choice. The PWM function is enabled to provide modulated output for Mark timing and disabled providing no output for Space timing.

Two additional routines were added to this application. The first, P (Protocol), enables you to choose from a list of the supported protocols. **Photo 4** shows the serial display for the menu item P, the list of protocols, followed by a list of button options using menu item B. P interprets the IRP, as I discussed earlier, in the S function used to try and match a protocol against a recorded sample. The same information is required to reproduce an IR transmission (e.g., modulation rate, bit sequence, bit timing, and a list of elements). This time we will not be gathering data values; we will be providing data values. These values come from the second function, B (Button).

This function brings up a list of buttons associated with the chosen protocol. **Listing 2** is a code snippet that defines the Sharp remote functions. Once a button choice is made, the data elements associated with the chosen button are loaded into their related variables. Next, we construct bit timings using the sample buffer in the same format as S transmissions, M, byteU, byteH, byteL and S, byteU, byteH,

PROJECT FILES



circuitcellar.com/ccmaterials

SOURCES

DatPlot
AeroPerf (Michael Vogt) |

www.datplot.com.

PIC18F2320 Microcontroller
Microchip Technology, Inc. | www.microchip.com

SFH4547 IR Emitter
OSRAM Opto Semiconductors | www.osram-os.com

TSOP322xx and TSMP6000TR IR Receiver
modules
Vishay Intertechnology, Inc. | www.vishay.com

byteL. By following the list of elements prepared by interpreting the IRP, we should be able to reproduce the original transmission. Once all the timings information is present, a final routine reads through the buffer enabling and disabling the PWM function, producing Mark and Space times that create an authentic IR transmission.

I was not surprised to find that my transmissions did not operate any of my equipment. Where do I start? Since the format of the sample buffer data is identical for the S and B functions, the C function can be used to display what's there. If I reproduced data correctly, the graphs of both should be the same. Again, I was not surprised when I saw two different graphs. It didn't take too much work to figure out what was going on.

The hard part is done: identifying the protocol from a graph. Knowing the protocol allows you to identify actual data values from a graph. By comparing the data values each graph represents, you should be able to determine the error's location. For example, my error was in the bit sequencing. While I shifted the data correctly, note that the earlier discussed Sharp format consists of a 5-bit variable D. I forgot to shift 0s for the last 3-bit used bits. My value 00001 looked like 00001000 and not 00000001.

SUCCESS

Until I started this project, I wasn't using my Acer 7" tablet solely as a storage device for my music collection because it came with an IR remote and I thought I'd make use of it. I added the NEC32 format it uses to this project. With Realterm running on the PC, I can instruct the project to select and play music from the Acer. All right! Now for the real test, the cable and TV located in my den.


I often sit in front of the tube and pluck on a laptop while watching cable programming on our ancient CRT TV. (I've yet to purchase an HDTV. If it ain't broke, don't replace it.) My wife Beverly seems to be getting a little hard of hearing. She likes flipping the channels and upping the volume past my comfort level. So, tonight I connected up this project to my laptop and kept the circuit out of sight while I did my normal investigative surfing. Every once in a while, I issued a few VOL- commands. It didn't take her long to boost it back up, but this went on for a while before she made a comment about the cable company. I let these pass and changed tactics. This time I kept sending commands to change to my favorite channel, "NASA select." Wowsers! This was my downfall. I should have selected something like the shopping channel. She quickly realized that I was somehow involved with this evil trickery. She's a good sport.

```
IRPButtonSize equ 0x10
IRP0Buttons equ D'14'
IRP0_Power org 0x7400      ; Sharp
    db "D",0x01, "F",0x16,0
;
IRP0_VOLPLUS org IRP0_Power+IRPButtonSize
    db "D",0x01, "F",0x14,0
;
IRP0_VOLMINUS org IRP0_VOLPLUS+IRPButtonSize
    db "D",0x01, "F",0x15,0
;
IRP0_ENTER org IRP0_VOLMINUS+IRPButtonSize
    db "D",0x01, "F",0xF7,0
;
IRP0_MUTE org IRP0_ENTER+IRPButtonSize
    db "D",0x01, "F",0x17,0
;
IRP0_TV   org IRP0_MUTE+IRPButtonSize
    db "D",0x01, "F",0xFC,0
;
IRP0_INFO org IRP0_TV+IRPButtonSize
    db "D",0x01, "F",0x1B,0
;
IRP0_DOWN org IRP0_INFO+IRPButtonSize
    db "D",0x01, "F",0x58,0
;
IRP0_UP   org IRP0_DOWN+IRPButtonSize
    db "D",0x01, "F",0x57,0
;
IRP0_LEFT org IRP0_UP+IRPButtonSize
    db "D",0x01, "F",0xF5,0
;
IRP0_RIGHT org IRP0_LEFT+IRPButtonSize
    db "D",0x01, "F",0xF6,0
;
IRP0_OK   org IRP0_RIGHT+IRPButtonSize
    db "D",0x01, "F",0xF7,0
;
IRP0_MENU org IRP0_OK+IRPButtonSize
    db "D",0x01, "F",0x20,0
;
IRP0_TV_VIDEO org IRP0_MENU+IRPButtonSize
    db "D",0x01, "F",0x13,0
```

LISTING 2

Here is a code snippet of the button entries for the Sharp TV. When used with the cable box, only volume controls are necessary. The TV_Video button cycles through the auxiliary TV inputs, which is useful for selecting the DVD player.

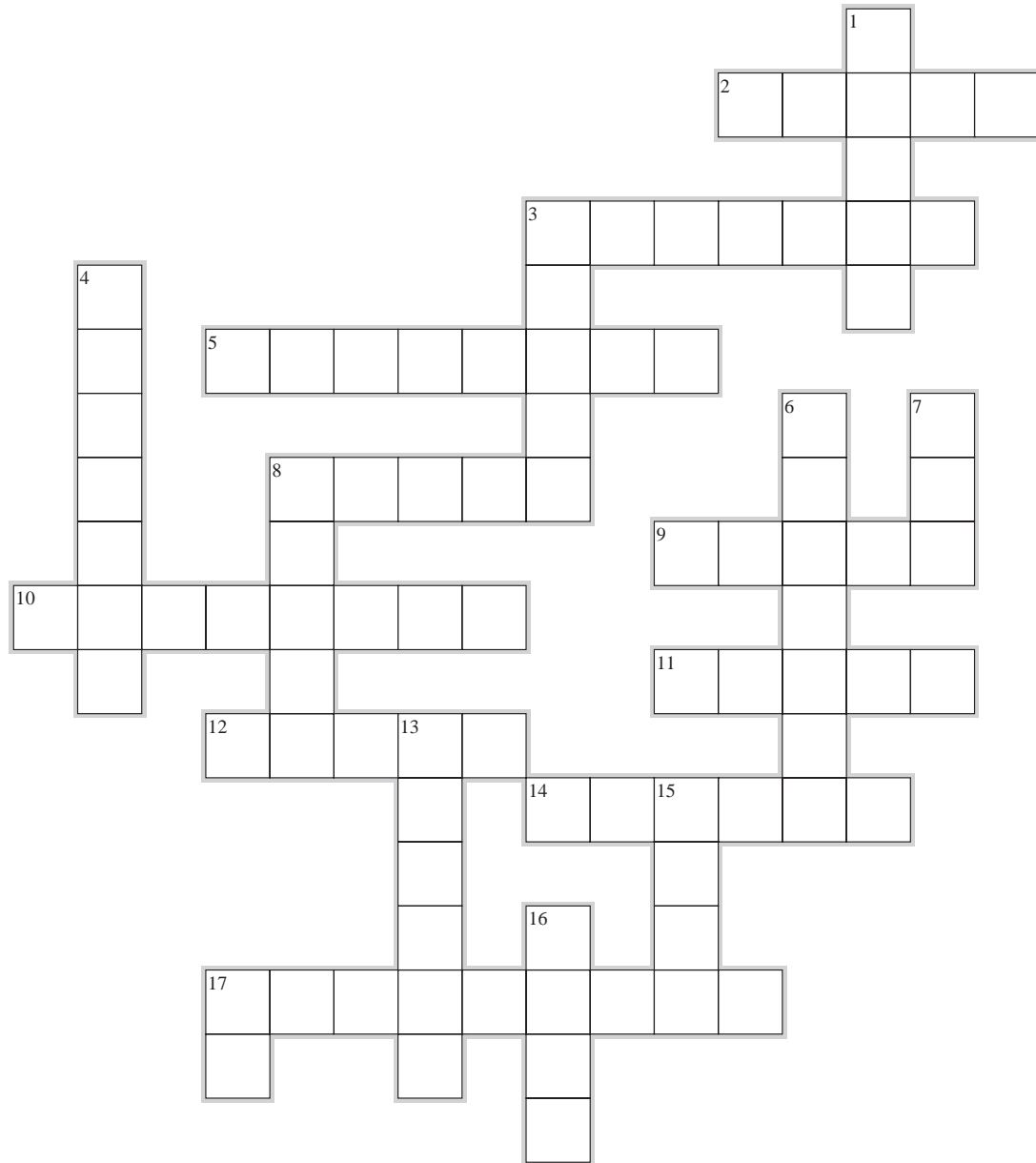
To reduce the workload on this project, I added only the protocols for which I have remotes. This means that this project does not tout compatibility with every remote out there. It is an educational tool developed for my requirements and may require extra work if you want to add additional formats. There is certainly plenty of room for that.

This IR project, while not the end of this story, stands on its own. I am planning to morph this project into something else. Are you curious? Stay tuned. 

CROSSWORD

SEPTEMBER 2014

The answers will be available at circuitcellar.com/crossword.



ACROSS

2. Trivalent valence
3. Kilby's Noble Prize in 2000
5. Converts DC to AC
8. BAT file
9. Founded ARRL in 1914
10. If you are AFK, what are you away from?
11. University that housed the ENIAC in a 30' x 40' room
12. 1 cycle per second
14. 4 bits
17. Asimov was the great what?

DOWN

1. PCB path
3. Quick fix
4. Used to monitor network traffic
6. A Gauss is one of these per square centimeter
7. "Big Blue"
8. "An Investigation of the Laws of Thought" (1854)
13. Move from setting A to B
15. Screen of death
16. A nonet is a group of what?
17. Exawatt

What's your EQ? The answers are posted at www.circuitcellar.com/category/test-your-eq/. You can contact the quizmasters at eq@circuitcellar.com.

TEST YOUR EQ

Contributed by David Tweed

PROBLEM 1

What is an R-C snubber, and what is a typical application for one?

PROBLEM 2

How do you pick the resistor value in an R-C snubber?

PROBLEM 3

How do you pick the capacitor value in an R-C snubber?

PROBLEM 4

What additional concern is there with regard to an R-C snubber when switching AC power?



CIRCUIT CELLAR

Sign up today and **SAVE 50%** • Sign up today and **SAVE 50%** • Sign up today and **SAVE 50%** • Sign up today and **SAVE 50%**

Now offering student SUBSCRIPTIONS!

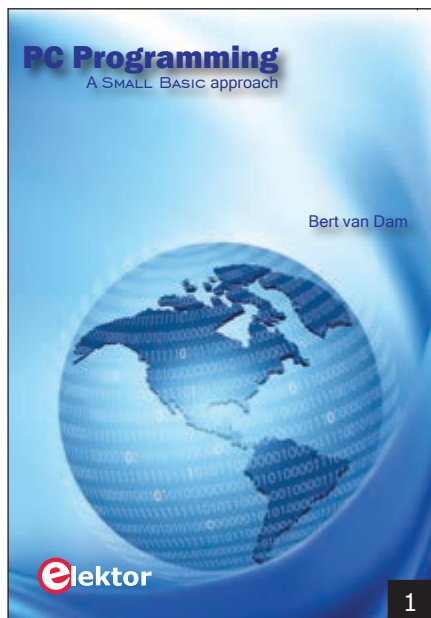
When textbooks just aren't enough, supplement your study supplies with a subscription to *Circuit Cellar*. From programming to soldering, robotics to Internet and connectivity, *Circuit Cellar* delivers the critical analysis you require to thrive and excel in your electronics engineering courses.

Sign up today and **SAVE 50%**
Sign up today and **SAVE 50%**
Sign up today and **SAVE 50%**

www.circuitcellar.com/subscription

CIRCUIT CELLAR
EMBEDDED DEVELOPMENT
Build an MCU-Based Light Timing & Control System
Electronic Computers: Fit Compensation & Calibration
S/D Port Examination
System-Level for Storage
MCSPY: JavaSPY

CC SHOP

**1 PC PROGRAMMING**

Many different PC programming languages are available. Some have beautiful names, some have easy-to-use development tools, others have incredible power. They all have one thing in common: they assume that you have, or want to have, a knack for technology and difficult-to-read commands. Using SmallBASIC, you will have an application up and running in a matter of minutes.

Author: Bert van Dam

Item #: BK-ELNL-978-1-907920-26-4

4 RFID MIFARE AND CONTACTLESS CARDS IN APPLICATION

RFID technology is now being used in many areas in which barcodes, magnetic strips, and contact smart-cards were previously used. This book provides a practical and comprehensive introduction to MIFARE, which is the most widely used RFID technology. The initial chapters cover physical fundamentals, relevant standards, RFID antenna design, security considerations, and cryptography.

Author: Gerhard H. Schalk and Renke Bienert

Item #: BK-ELNL-978-1-907920-14-1

**2 ANDROID APPS: PROGRAMMING STEP-BY-STEP**

Many smartphones and tablet computers are powered by an Android OS. These portable devices' speed and computing power enable them to run applications that would have previously required a desktop PC or custom-designed hardware. *Android Apps* introduces you to the programming required to design apps for Android devices. Operating the Android system is explained step-by-step to show how personal applications can be easily programmed.

Author: Stefan Schwark

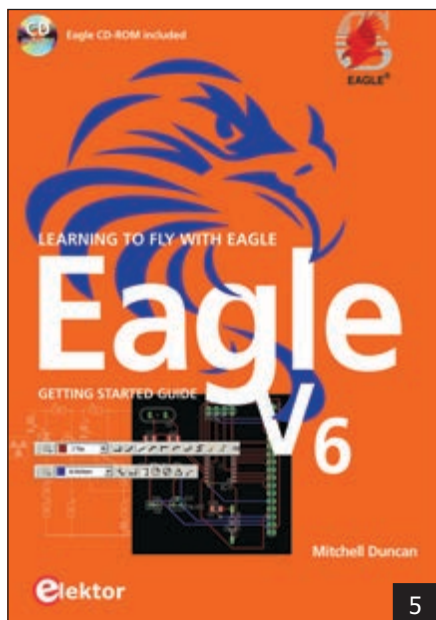
Item #: BK-ELNL-978-1-907920-15-8

**3 CC VAULT**

CC Vault is a pocket-sized USB that comes fully loaded with every issue of *Circuit Cellar* magazine. This comprehensive archive provides an unparalleled amount of embedded hardware and software design tips, schematics, and source code. CC Vault contains all the trade secrets you need to become a better, more educated electronics engineer.

Item #: CCVAULT

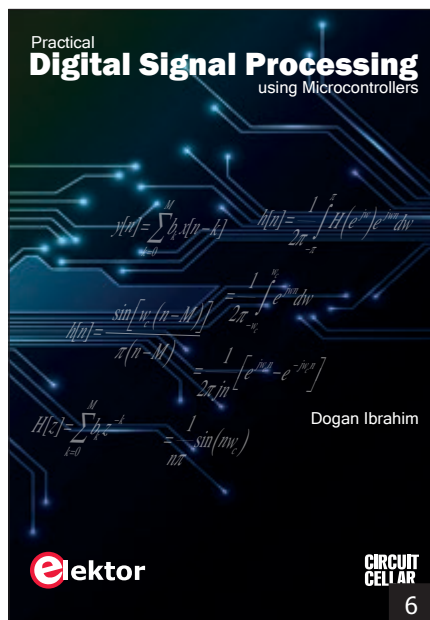
CC SHOP



5 LEARNING TO FLY WITH EAGLE V6

EAGLE is a user-friendly, powerful, and affordable software package for efficient PCB design. It can be used on most main computing platforms including: Microsoft Windows, Linux, and Apple Mac OS X. *EAGLE V6* will benefit novices and professionals who are eager to learn about EAGLE or may be migrating from another computer-aided design (CAD) package. From schematic and layout editing tools to project completion, this book will help you achieve your PCB fabrication goals.

Author: Mitchell Duncan
Item #: BK-ELNL-978-1-907920-20-2



6 PRACTICAL DIGITAL SIGNAL PROCESSING USING MICROCONTROLLERS

Digital signal processing (DSP) reflects the growing importance of discrete time signals and their use in everyday microcontroller-based systems. In this book, the author presents the basic theory of DSP with minimum mathematical treatment and teaches you how to design and implement DSP algorithms using popular PIC microcontrollers. You'll be up and running in no time!

Author: Dogan Ibrahim
Item #: BK-ELNL-978-1-907920-21-9



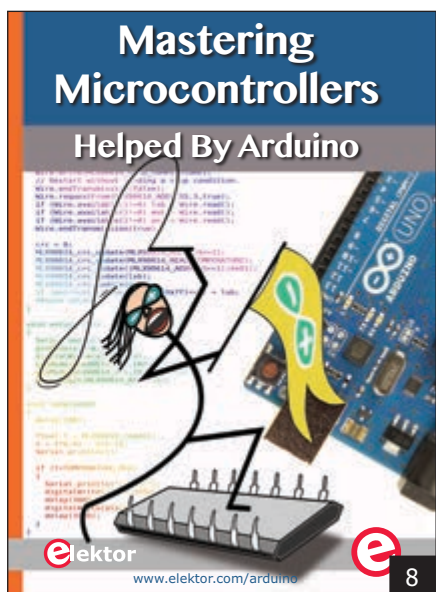
7 CIRCUIT CELLAR ISSUE PDFS

Have you ever missed an issue of *Circuit Cellar* and didn't know where to find it? Grab a digital version! From 1998 to the present, dive into more than 100 issues of embedded electronics insights that you may have missed. Can you envision the evolution and development across a decade? With technology advancing every day, don't fall behind. Learn about wearable wireless transceivers from Mathew Laibowitz and Joseph Paradiso or a DIY single-board computer (SBC) from Oscar Vermeulen and Andrew Lynch. The topics and projects are endless.

8 MASTERING MICROCONTROLLERS: HELPED BY ARDUINO

Arduino boards have become hugely successful. They are simple to use and inexpensive. *Mastering Microcontrollers* will teach you how to program microcontrollers and help you turn theory into practice using an Arduino programming environment. Become a master today!

Author: Clemens Valens
Item #: BK-ELNL-978-1-907920-23-33



Further information and ordering

www.cc-webshop.com

CONTACT US:

Circuit Cellar, Inc.
 111 Founders Plaza, Suite 300
 East Hartford, CT 06108
 USA

Phone: 860.289.0800

Fax: 860.461.0450

E-mail: custservice@circuitcellar.com

IDEA BOX

the directory of PRODUCTS & SERVICES

AD FORMAT:

Advertisers must furnish digital files that meet our specifications (circuitcellar.com/mediakit).

ALL TEXT AND OTHER ELEMENTS MUST FIT WITHIN A 2" x 3" FORMAT. E-mail adcopy@circuitcellar.com with your file.

For current rates, deadlines, and more information contact Peter Wostrel at 978.281.7708 or circuitcellar@smmarketing.us.

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

\$20 for 5PCBs

2 layer, 4x4inch, FR4(RoHS), 0.063", 1oz, 2LPI, Green, 1SK, Lead free HASL

Standard PCB: Promotion code:

CC14061

PCB & PCBA

Small to Mass QTY

INSTANT QUOTE AT:

www.myropcb.com

OR CALL:

1-888-PCB-MYRO



Add A Touch Screen

- Get a Prototype Up in Days
- Move Smoothly to Production
- Development Kits Start at \$349
- 100% Satisfaction Guarantee

Learn more at www.reachtech.com or contact us at 510-770-1417 or sales@reachtech.com.



REACH
TECHNOLOGY INC.

Stream Data through the PIC® MCU ICSP™ Interface

• Eliminate a diagnostic serial port & save money by streaming data through the ICSP™ interface

• Set the Serial #'s & Calibration values in production



Download a FREE
Full Demo Today

www.ccsinfo.com/CC814

sales@ccsinfo.com
262-522-6500 x 35

CCS

PIC® MCU is a registered trademark of Microchip Technology Inc.

GHz Bandwidth Sockets for DSP's in BGA

Industry's Smallest Footprint

- Pitch - 0.3mm to 1.27mm -
- BGA - QFN (MLF)
- Bandwidth to 40+ GHz
- Six different bid options
- Optional 500,000 insertions
- Heatsinking to 100 watts

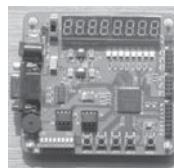


RoHS

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

LEARN FPGA PROGRAMMING USING THE ALTERA QUARTUS II AND VERILOG

The course comes with a Cyclone board, as shown, and an USB Blaster. Also included is a 175-page manual containing ten chapters that are full of examples and tutorials. This in-depth course is a must for the hobbyist, as well as, for schools teaching FPGAs. The manual is easy to understand and correct.



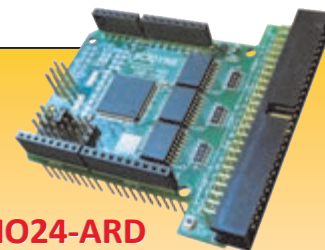
\$55 + \$12 shipping

ADVANCED MICRO CIRCUITS

443-817-2557

www.amcmicros.com

information@amcmicros.com



DIO24-ARD

Digital Interface
for Arduino and Compatibles

- 24 Programmable Input/Output Channels
- Connect I/O Racks, LEDs, Switches, and More
- 85ma Sink Output Capability
- Uses Standard SPI Software Library Functions
- Jumper Selectable and Off-Board SPI Enable
- Long Lead Stack-Through Connectors
- Industrial Operating Temperature Range
- RoHS Compliant

Learn More Details at . . .

SCIDYNE®
www.scidyne.com

Call Toll-Free
1-877-SCIDYNE
(1-877-724-3963)

Join The
INTERNET OF THINGS
REVOLUTION



PIONEERING IoT SINCE 2001

TRI
Programmable Logic Controllers

Powerful & Easy Ladder
+BASIC Programming
Ethernet integrated
MODBUS TCP/IP
DI/Os & AI/Os integrated

OEM Prices as low as \$119
for full-feature Nano-10 PLC

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



TRI TRIANGLE
RESEARCH
INTERNATIONAL

FPGA Boards **HUMAN DATA**
from JAPAN

SAVING COST=TIME with
readily available FPGA boards

Altera
ACM Series
Cyclone V, Arria II,
MAX II and other
many FPGA boards
are available

xilinx
XCM Series
Kintex-7, Spartan-6, Virtex-5
and other many FPGA boards
are available

PLCC68 Series

- Designed for 68-pin PLCC socket
- Very small size (25.3 x 25.3 [mm])
- 50 I/Os (External clock inputs available)
- 3.3V single power supply operation

www.hdl.co.jp/CC/

MaxBotix
High Performance Ultrasonic Rangefinders

Save 10%
Online Web-Order Code:
C143009
Valid thru October 1, 2014

HRXL-MaxSonar®-WR™

- High noise tolerance
- IP67 rated
- 1 mm resolution
- Multi-Sensor operation
- Calibrated beam pattern
- Starting at \$109.95



XL-MaxSonar®-EZ™

- Great for UAV's and robotics
- Incredible noise immunity
- Small in size
- 1cm resolution
- Automatic calibration
- Starting at \$39.95



Phone: 218-454-0766 Email: sales@maxbotix.com
www.maxbotix.com

Real Logic Analyzers from **\$349.00**

with Decoders & Plug-in Kit included



- 9 to 36 Channels
- 100 Msps to 400 Msps
- Store & Forward Architecture
- Real-time Smart Compression
- Plot & Measurement Features
- Powerfull Search Tools
- Plug-ins for full Custom Protocol integration

by **TechTools** www.tech-tools.com/real.htm • (972) 272-9392

ALL ELECTRONICS
CORPORATION

Electronic and Electro-mechanical Devices, Parts and Supplies.
Many unique items.


We have what you need for your next project.



www.allelectronics.com
Free 96 page catalog 1-800-826-5432

microEngineering Labs, Inc.
www.melabs.com 888-316-1753

Programmiers for Microchip PIC® Microcontrollers



PC-Tethered USB Model (shown):

- Standalone software
- Command-line operation
- Hide GUI for automated use
- Override configuration with drop-downs

Stand-Alone Field Programmer:

- Power from target device or adapter
- Program file stored on SD-CARD
- Programming options stored in file
- Single-button operation

Starting at \$79.95

Program in-circuit or use adapters for unmounted chips.
Zero-Insertion-Force Adapters available for DIP, SOIC, SSOP, TQFP, and more.

PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.



The Future of 3-D Printed Electronics


By Dr. Martin Hedges

Three-dimensional (3-D) printing for prototyping has been around for nearly three decades since the introduction of the first SL systems. The last few years have seen this technology receiving considerable attention to the point of hype in the mainstream media. However, there is a new emerging 3-D printing market that is increasing in importance: 3-D printed electronics (3-D PE). Whilst traditional 3-D printing builds structural parts layer by layer, 3-D PE prints liquid inks that have electronic functionality on to existing 3-D components. 3-D PE is achieved by combining advanced printing technologies, such as Aerosol Jet, with specially designed five-axis systems and advanced software controls that allow complex print motion to be achieved. The integrated print systems allow the full range of electronic functionality to be applied: conductors, semiconductors, resistors, dielectrics, optical, and encapsulation materials.

These can be printed on to virtually any surface material of almost any shape. Once deposited the inks are post processed: sintered, dried, or cured to achieve their final properties. Multiple materials can be printed to build up functionality, or surface mount devices (SMD) can be added to make the final electro-mechanically integrated system (see **Photo 1**). In this example, two capacitive sensor structures have been printed on the ends of an injection-molded PA6 tank. The sensors are connected by a printed circuit (conductive Ag) and SMD components are added to complete the device. When water is pumped into the tank, the sensors register the water level as it rises, lighting the LEDs to indicate the fill level. When the tank compartment is full, the circuit senses the water fill level and reverses the pump direction.

3-D PE has the potential to provide enormous technical and economic benefits in comparison to conventional electronics based on 2-D printed circuit boards. It allows the combination of electronic, optic, and mechanical functions on shaped circuit carriers. Therefore, it enables entirely new product functions and supports the miniaturization and weight-saving potential of electronic products. By eliminating mechanical components, process chains can be shortened and reliability is increased. As a digitally driven, additive manufacturing process materials are only applied where needed, improving the ecological balance of electronics production. With no fundamental limitation on substrate material, the user is able to select low-cost, easy-to-recycle and more environmentally friendly materials. The novel design and functional possibilities offered by 3-D PE and the potential for rationalization of production steps indicate a potential quantum leap in electronics production.

Advances in this field have been rapid since the first developments that focused on 3-D chip packaging. In this field, printing is conducted over small changes in z-height to connect SMDs. **Photo 2** shows an example where wire bonds are replaced by printing interconnects, from the PCB, up the side of a chip, and over onto the top contact pads. Such applications only require relatively simple print motion. The current "state of the art" is to use five axes of coordinated motion to print high complex shapes. This capability enables the production of truly 3-D PE systems, such as a 3-D antenna for mobile devices (see **Photo 3**). This application is well advanced and moving towards high-volume mass production driven by the benefits of a flexible manufacturing, novel design capabilities, and cost reduction compared to the current methods based on wet chemical plating processes. 3-D PE is also being scaled to print on large components beyond the size range possible with current manufacturing methods. For example, in the automotive field, 3-D prints of heater patterns are being developed for molded PC windscreens of up to 2 m × 1 m in size.

Currently, 3-D PE applications are mainly limited to circuits, antennas, strain gauges, or sensors using conductive metal as the print media with additional electronic functionality being added as SMDs. However, the technology also has the potential to leverage new material and process developments from the printed and organic electronics world. In this field, many different material systems are currently being applied on planar surfaces to create multi-material and multi-layer devices. Functionality such as resistors, capacitors, sensors, and even transistors are being incorporated into fully printed 2-D electronic systems. As these print materials and processes mature, they can be adapted to 3-D applications. It is expected that the coming years will see a rapid increase in the range of fully printed 3-D electronic devices of novel functionality. 

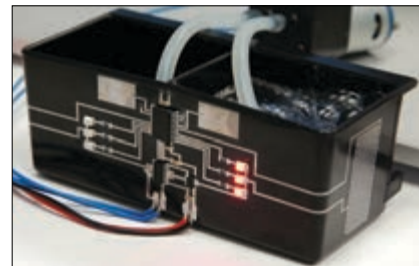


Photo 1

3-D PE demonstrator—Tank-filling sensor produced in the FKIA project funded by the Bavarian Research Foundation (Courtesy of Neotech AMT GmbH)

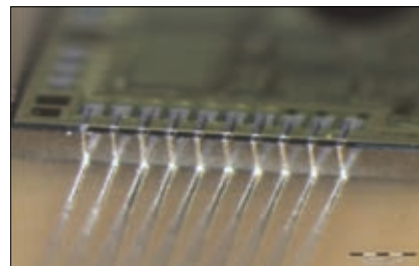


Photo 2

3-D chip packaging (Courtesy of Fraunhofer IKTS)



Photo 3

3-D printed antenna (Courtesy of Lite-On Mobile)

Dr. Martin Hedges (mhedges@neotech-amt.com) is the Managing Director of Neotech AMT GmbH based in Nuremberg, Germany. His research includes aspects of additive manufacturing, materials and processes. His company projects focus on the development of integrated manufacturing systems for 3-D printed electronics.

engineering electronics
 DISCUSS **embedded**
 design tips tutorial software
 engage **engineering** tools
 contests system **audio** business
 data **networking** **media** COMMUNITY
social media talk
 information product news
 projects

Want to talk to us directly?

Share your interests and opinions!

Check out our New Social Media Outlets for direct engagement!



CIRCUIT CELLAR / AUDIOXPRESS / ELEKTOR



MULTIPLE PCBA LINES **NO MORE WAITING** ACROSS THE GLOBE!

Imagineering utilizes the Jet Printer Technology which has been proven successful in five continents. This breakthrough innovation allows us to achieve high-precision solder deposits at speeds of more than one million dots per hour.

High-Speed Non-Contact Jetting Nozzle • Complete Volume Control • 100% Software Driven
More Components, More Possibilities • High-Performance Platform



imagineering inc.

www.PCBnet.com

(847) 806-0003 • sales@PCBnet.com

**NEW CUSTOMERS:
GET \$500 TOWARDS
FREE PARTS!**