HOW TO: Handle Embedded Design Issues
LOCATION: United States
PAGE: 44

PROJECT DEVELOPMENT: Compliance & Reviews
LOCATION: Canada
PAGE: 48

INSIGHT: Charging with Photovoltaic Cells
LOCATION: United States
PAGE: 68

CIRCUIT CELLAR

THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

AUGUST 2012
ISSUE 265

EMBEDDED DEVELOPMENT

Build an MCU-Based Light Timing & Control System

Electronic Compass: Tilt Compensation & Calibration

I/O Port Examination

System-Level RF Design

MOSFET Tester Circuitry

PLUS

The Mind of a Roboticist

// Innovative Robotics Projects
// Inside the Design Lab
// Intelligent Use of MCUs & Sensors



$9.00US $10.00CAN
08
0 74470 75349 0

www.circuitcellar.com

The NetBurner SB70LC



$49.00
Qty. 100
SB70LC Board

The complete hardware and software solution

- Low-cost
- 10/100 Ethernet
- Two Serial Ports
- 32-bit Performance
- Digital I/O
- SSH/SSL Option
- Flash Card Ready
- I²C • SPI Interface
- Example applications



$99.00
SB70LC Development Kit

The SB70LC Development Kit

The NetBurner SB70 LC Development Kit is available to customize any aspect of operation including web pages, data filtering, or custom network applications. The kit includes platform hardware, ANSI C/C++ compiler, TCP/IP stack, web server, e-mail protocols, RTOS, flash file system, Eclipse IDE, debugger, and cables. The kit enables you to communicate with peripherals that use SD/MMC Flash Card (including SDHC), SPI, I²C, or the general purpose digital I/O interface. The NetBurner security suite option includes SSH v1, v2 and SSL support.

NetBurner
Networking in One Day!

Board Part Number | SB70LC-100IR
Development Kit Part Number | NNDK-SB70LC-KIT
Information and Sales | sales@netburner.com
Web | www.netburner.com
Telephone | 1-800-695-6828

freescale™
Alliance Member

NEWHAVEN DISPLAY
INTERNATIONAL

NEW
Capacitive Touch TFTs

- 3.5", 4.3" and 5.0" Capacitive Touch TFTs
- Built-in controller with I²C interface
- 6H hardness, >1 million touches
- 5 simultaneous touches (including lift, zoom & touch gestures)
- 6mA at 3.3V supply, 0.03mA sleep mode; 60HZ refresh rate; >100dpi resolution

NEWHAVEN DISPLAY INTERNATIONAL

Newhaven Display International Inc. now offers projective mutual capacitive touch TFTs with a built-in I²C interface controller. This allows for quick time to market while keeping your development costs low. The design of the capacitive touch TFTs will be available in 3.5", 4.3" and 5.0" size displays. The capacitive touch panel is the ideal tool for enhancing the user interface experience with any TFT application. These touch TFTs have multi-touch ability and are extremely durable in most environments including withstanding water contact, dust and grease. Capacitive touch TFTs activate with either a bare finger or a capacitive stylus and have an excellent dragging performance.



For more information visit: circuitcellar.newhavendisplay.com
Contact us at: 847-844-8795

RoHS Compliant

TASK MANAGER

Embedded Systems Abound

I recently read on CNN.com the transcript of an interview (May 9, 2002) with arachnologist Norman Platnick who stated: "You're probably within seven or eight feet of spider no matter where you are. The only place on earth that has no spiders at all—as far as we know—is Antarctica." It didn't take long for me to start thinking about embedded systems and my proximity to them. Is the average person always within several feet of embedded systems? Probably not. But what about 50% or 60% of the time? E-mail me your thoughts.

Embedded systems are becoming ubiquitous. They're in vehicles, mobile electronics, toys, industrial applications, home appliances, and more. If you're indoors, the temperature is likely monitored and controlled by an embedded system. When you're engaged in outdoor activities (e.g., hiking, golfing, biking, or boating), you probably have a few MCU-controlled devices nearby, such as cell phones, rangefinders, pedometers, and navigation systems. This month we present articles about how embedded systems work, and our authors also provide valuable insight about topics ranging from concurrency to project development.

Mark Pedley kicks off the issue with a revealing article about a tilt-compensating electronic compass (p. 16). Now you can add an e-compass to your next MCU-based project.

Turn to page 24 for an in-depth interview with Italy-based engineer Guido Ottaviani. His fascination with electronics engineering, and robotics in particular, will inspire you.

Have you ever come across a product that you know you could have designed better? Scott Weber had that experience and then acted on his impulse to build a more effective system. He created an MCU-based light controller (p. 32).

If you want to ensure a microcontroller works efficiently within one of your systems, you should get to know it inside and out. Shlomo Engelberg examines the internal structure of an I/O pin with a pull-up resistor (p. 40).

Bob Japenga continues his series "Concurrency in Embedded Systems" on page 44. He covers atomicity and time of check to time of use (TOCTTOU).

On page 48 George Novacek presents the second part of his series on project development. He covers project milestones and design reviews.

Ed Nisley's June 2012 article introduced the topic of MOSFET channel resistance. On page 52 he covers MOSFET tester circuitry and provides test results.

If you read Robert Lacoste's June 2012 article, you now understand the basics of frequency mixers. This month he presents high-level design methods and tools (p. 58).

Jeff Bachiochi wraps up the issue with an examination of a popular topic—energy harvesting (p. 68). He covers PV cell technology, maximum power point tracking (MPPT), and charge management control.

cj@circuitcellar.com



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

EDITORIAL CALENDAR

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

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

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

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

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

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

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

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

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

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

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

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

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

UPCOMING IN CIRCUIT CELLAR

FEATURES

Miniature Accelerometers, by Mark Csele

FAT Cache for Fast SD Card Access, by Kerry Imming

MCU-Based Environmental Data Logger, by Brian Beard

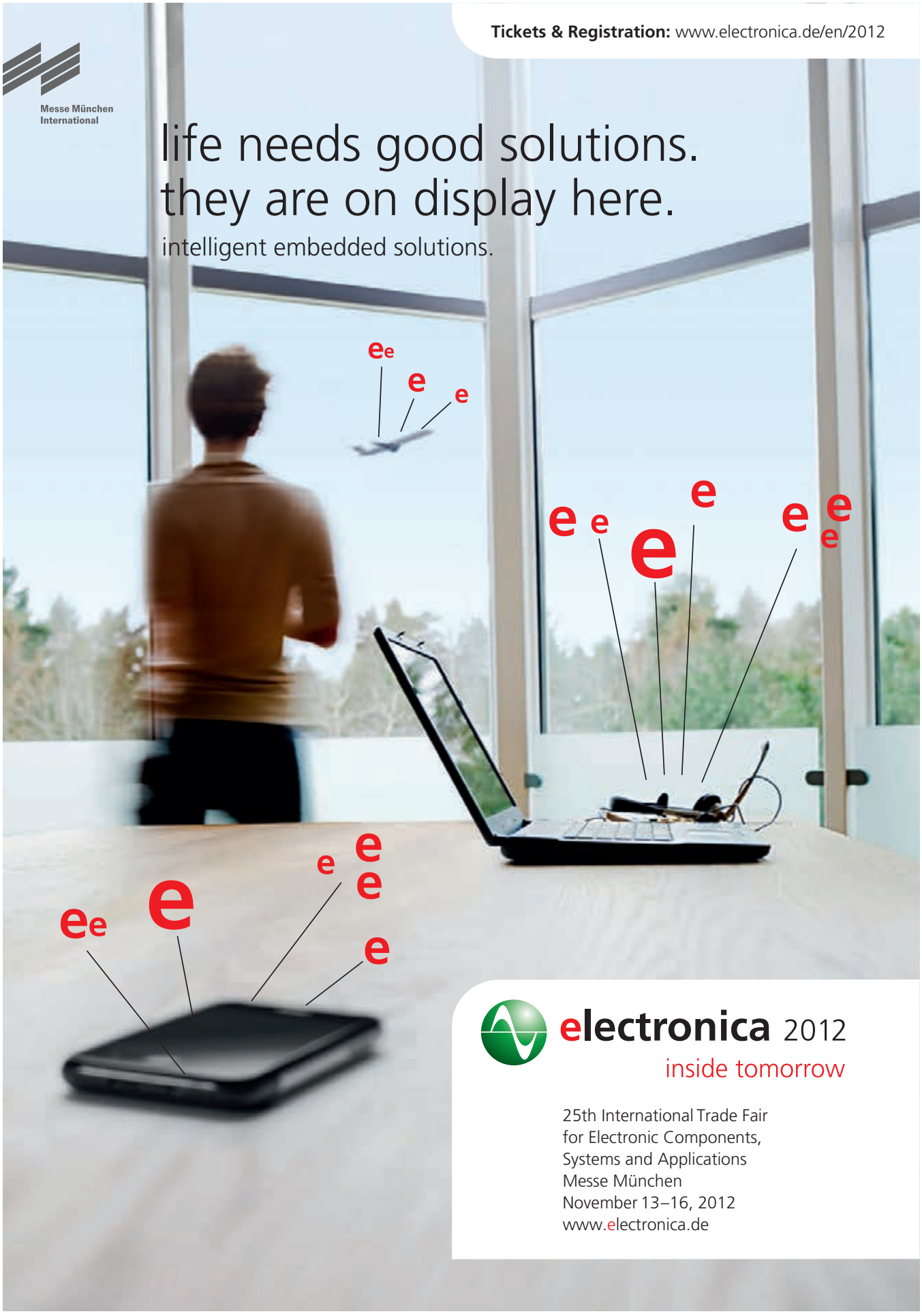
COLUMNS

Flowcharting Made Simple, by Jeff Bachiochi

Software & Design File Organization, by George Martin

Hardware-Accelerated Encryption, by Patrick Schaumont

Project Configuration Control, by George Novacek


Tickets & Registration: www.electronica.de/en/2012
Messe München International
life needs good solutions.
they are on display here.
intelligent embedded solutions.
electronica 2012
inside tomorrow
25th International Trade Fair
for Electronic Components,
Systems and Applications
Messe München
November 13–16, 2012
www.electronica.de

INSIDE ISSUE 265

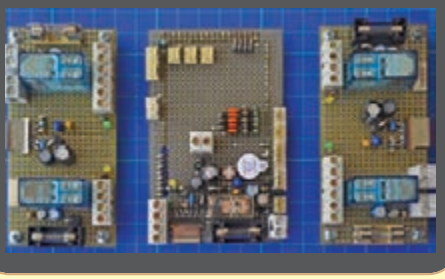
August 2012 • Embedded Development

16 **eCompass**
Build and Calibrate a Tilt-Compensating Electronic Compass
Mark Pedley

32 **MCU-Based Light Control**
Longer Serial Communication on Differential Wires
Scott Weber

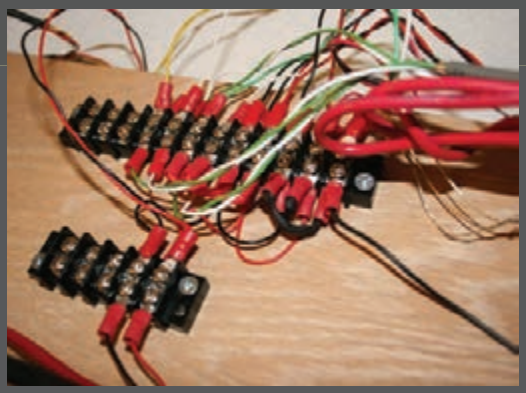
40 **Examining an I/O Port**
The Internal Structure of an I/O Pin with a Pull-Up Resistor
Shlomo Engelberg

Embedded System Integration in Robotics p. 24

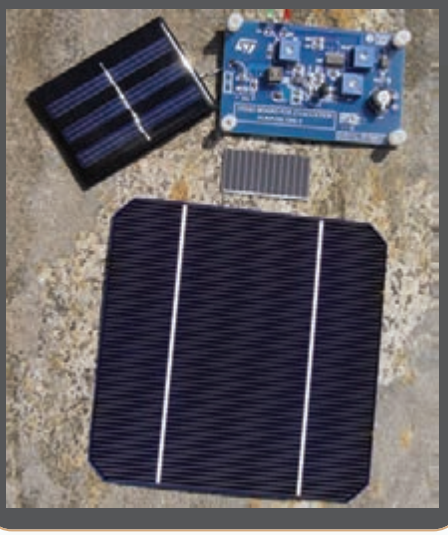


DIY Light Controller p. 32





Power Up with PV Cells p. 68



44 **EMBEDDED IN THIN SLICES**
Concurrency in Embedded Systems (Part 2)
Atomicity and TOCTTOU
Bob Japenga

48 **THE CONSUMMATE ENGINEER**
Project Development (Part 2)
Development Process, Milestones, and Design Reviews
George Novacek

52 **ABOVE THE GROUND PLANE**
MOSFET Channel Resistance
Tester Hardware
Ed Nisley

58 **THE DARKER SIDE**
System-Level RF Design
Robert Lacoste

68 **FROM THE BENCH**
Charging with PV Cells
Jeff Bachiochi

TASK MANAGER 2
Embedded Systems Abound
C. J. Abate

NEW PRODUCT NEWS 10

MEMBER PROFILE 14

TEST YOUR EQ SOLUTIONS 15

QUESTIONS & ANSWERS 24
The Man Behind the dsNavCon Design
An Interview with Guido Ottaviani
Nan Price

CROSSWORD 76

PRIORITY INTERRUPT 80
What Were They Thinking?
Steve Ciarcia


mouser.com
Distributing semiconductors and electronic components for design engineers.
Authorized Distributor
GO
MORE.
mouser.com
The widest selection of the newest products.
The Newest Products for Your Newest Designs®
MOUSER ELECTRONICS
a tti company
Mouser and Mouser Electronics are registered trademarks of Mouser Electronics, Inc. Other products, logos, and company names mentioned herein, may be trademarks of their respective owners.

The team

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

The network



Our international teams

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Issue 265 August 2012 ISSN 1528-0608

CIRCUIT CELLAR® (ISSN 1528-0608) is published monthly by Circuit Cellar Incorporated, 4 Park Street, Vernon, CT 06066. Periodical rates paid at Vernon, CT and additional offices. One-year (12 issues) subscription rate USA and possessions $50, Canada $65, Foreign/ROW $75. All subscription orders payable in U.S. funds only via Visa, MasterCard, international postal money order, or check drawn on U.S. bank.

Cover photography by Chris Rakoczy—www.rakoczyphoto.com

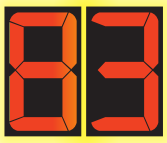
Subscriptions
Circuit Cellar, P.O. Box 462256, Escondido, CA 92046
E-mail: circuitcellar@pcspublink.com
Phone: 800.269.6301, Internet: www.circuitcellar.com
Address Changes/Problems: circuitcellar@pcspublink.com

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

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

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

MEMBERSHIP COUNTER

We now have  members in  countries.

Not a member yet?

Sign up at www.circuitcellar.com

SUPPORTING COMPANIES

2012 International CES 73
All Electronics Corp. 78
AP Circuits 43
ARM. 31
Atria Technologies, Inc. 77
Beta Layout, Ltd. 15
BusBoard Prototype Systems. 78
Comfile Technology 35
Cleverscope. 15
Custom Computer Services 78
Electronica 2012 3
Elektor 66, 67
Elektor 75
EMAC, Inc. 70
ESC Boston, a DESIGN East Summit 39
ExpressPCB. 71
Flexipanel, Ltd. 77
FTDI Chip. C3
Grid Connect, Inc. 43
HannoWare. 63
Holtek Semiconductor, Inc. 51
Humandata, Ltd. 65
Imagineering, Inc. C4
Ironwood Electronics 78
Jeffrey Kerr, LLC. 63
JK microsystems, Inc. 13, 77
Labcenter Electronics 29
LinkSprite 78
MaxBotix, Inc. 79
MCC, Micro Computer Control 77
Microengineering Labs 79
Mosaic Industries 78
Mouser Electronics, Inc. 5
NetBurner C2
Newark element14 19
Newhaven Display International. 1
NKC Electronics 79
Pico Technology, Ltd. 62
Pololu Corp. 47
Reach Technology, Inc. 78
Rigol Technologies 79
Renesas RL78 Green Energy Challenge 21
Saelig Co., Inc. 60
Sealevel Systems. 27
SiliconRay 13
Technologic Systems 8, 9
Tern, Inc. 77
Triangle Research International, Inc. 78
UltraTek 63

Not a supporting company yet?

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

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

Copyright Notice
Entire contents copyright © 2012 by Circuit Cellar, Inc. All rights reserved. Circuit Cellar is a registered trademark of Circuit Cellar, Inc. Reproduction of this publication in whole or in part without written consent from Circuit Cellar Inc. is prohibited.

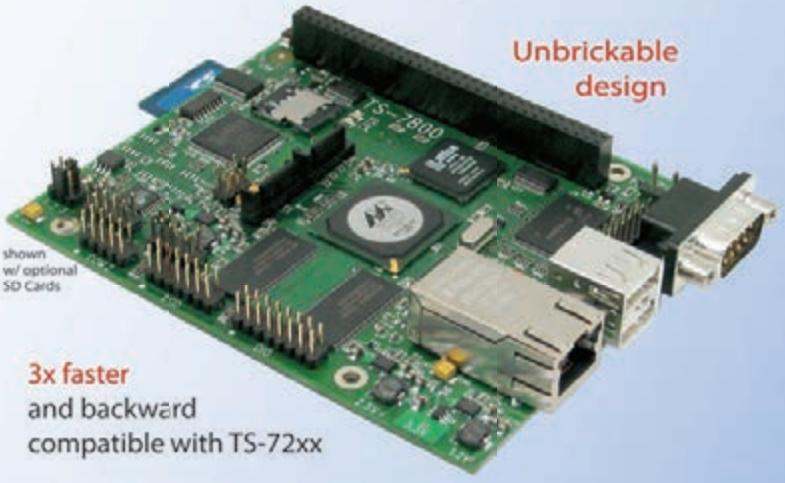
Disclaimer
Circuit Cellar® makes no warranties and assumes no responsibility or liability of any kind for errors in these programs or schematics or for the consequences of any such errors. Furthermore, because of possible variation in the quality and condition of materials and workmanship of reader-assembled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

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

© Circuit Cellar 2012 Printed in the United States

Embedded Systems

High-End Performance with Embedded Ruggedness



TS-7800
500MHz ARM9

- Low power - 4W@5V
- 128MB DDR RAM
- 512MB high-speed (17MB/sec) onboard Flash
- 12K LUT customizable FPGA
- Internal PCI Bus, PC/104 connector
- 2 host USB 2.0 480 Mbps
- Gigabit ethernet
- 2 SD sockets
- 10 serial ports
- 110 GPIO
- 5 ADC (10-bit)
- 2 SATA ports
- Sleep mode uses 200 microamps
- Boots Linux 2.6 in 0.7 seconds
- Linux 2.6 and Debian by default

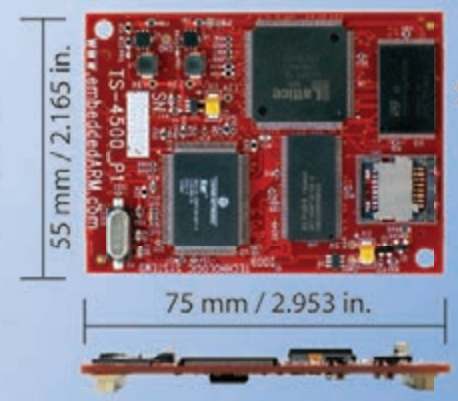
$229 qty 100
$269 qty 1

TS-SOCKET Macrocontrollers
Jump Start Your Embedded System Design

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

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

series starts at
$92 qty 100
$139 qty 1



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

- Over 25 years in business
- Open Source Vision
- Never discontinued a product
- Engineers on Tech Support
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products stocked and available for next day shipping

Design your solution with one of our engineers (480) 837-5200

New Products

Touch Panel Computers
800MHz with Video Acceleration

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



Fully enclosed TPC available Q3

NEW!



series starts at
$415
qty 100

$479
qty 1



picture of TS-8820-BOX

NEW!

series starts at
$199
qty 100

$229
qty 1

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

Industrial Controllers
Powerful, Rugged, Affordable

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

Technologic Systems

We use our stuff.

Visit our TS-7800 powered website at

www.embeddedARM.com

TRANSDUCERS FOR RMS VOLTAGE & CURRENT MEASUREMENT

The **MVTR** and **MCTR** are single-phase, RoHS-compliant transducers designed to measure voltage and current in applications that need a true RMS calculation of the transducer input. The MVTR units measure true RMS voltage. The MCTR models measure true RMS current. All MVTR and MCTR models produce a fully isolated 4-to-20-mA output proportional to the related input voltage and current, respectively, for the sinusoidal and non-sinusoidal inputs commonly found among variable-frequency applications. Instrument power is supplied via a control industry-standard 4-to-20-mA loop power.



The MVTR supports three full-scale voltage range inputs (150, 300, and 600 VAC). It has a 200-ms response time in 150-V and 300-V models, and a 500-ms response time in the 600-V model. The MCTR supports four input ranges (0 A to 0.1 A, 0 A to 1 A, 0 A to 5 A, and 0.333 V). Both transducers feature a 50-to-400-Hz frequency range and a –20-to-60°C operating temperature range.

Contact Ohio Semitronics for pricing.

Ohio Semitronics, Inc.
www.ohiosemitronics.com

RF BUILDING BLOCK QUADRATURE MODULATOR/DEMODULATOR

The **CMX973** is an integrated quadrature IF/RF demodulator and quadrature modulator. This addition to CML's RF building block range is well suited for wireless data terminals; HF, VHF, and UHF mobile radio; avionics radio systems; telemetry modems; high-performance professional radios; satellite terminals; and software-designed radio (SDR).



The CMX973 combines a flexible IF/RF quadrature modulator and quadrature demodulator into a single chip that forms the basis of a high-performance RF transceiver system. A wide signal bandwidth at RF, 0-dBm output, low noise, and serial bus control help the demodulator/modulator deliver maximum flexibility and high performance at a low operating power.

The demodulator/modulator is designed for superheterodyne architectures with 20-to-300-MHz IF frequencies. It can be used in low-IF systems or those converting down to baseband. The demodulator/modulator converts directly from baseband to the desired 20-MHz-to-1-GHz transmit frequency and features quadrature phase correction to minimize unwanted spectral components.

The CMX973 demodulator/modulator's additional features include: low-power operation (3 to 3.6 V), less than 1° I/Q phase matching, less than 0.5-dB I/Q gain matching, and 0-dBm modulator output level.

The CMX973's quadrature architecture supports a range of modulation formats and standards. The differential inputs are ideal for direct connection to standard modems and interface ICs such as CML's CMX7163, CMX7164, CMX910, CMX981, and CMX7861.

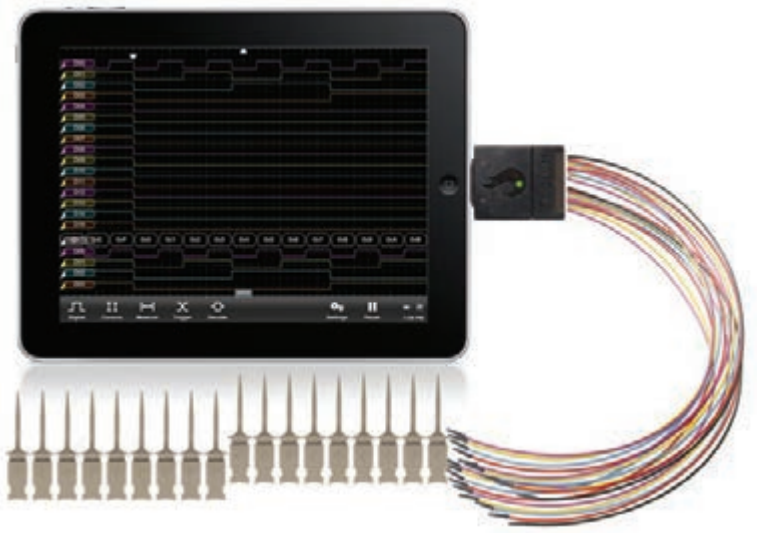
The CMX973 is available in an RF-optimized, 32-lead VQFN package that measures only 5 mm × 5 mm × 1 mm. Contact CML Microsystems for pricing.

CML Microsystems
www.cmlmicro.com

TRANSFORM AN iPAD INTO A PORTABLE LOGIC ANALYZER

LogiScope is a logic analyzer with an oscilloscope's real-time data analysis capabilities. It is ultra-portable and specifically designed for the iOS family of products (i.e., iPhone, iPad, and iPod Touch).

The analyzer transforms an iPad into a 100-MHz, 16-channel logic analyzer. Traditionally, a logic analyzer records a buffer that needs to be downloaded and searched. LogiScope's triggering enables decoded data to be viewed live. This eliminates the need to capture, pause, and then view.



The touchscreen-based iOS platform features a simple and intuitive display. For example, to change the timescale, simply zoom into a smartphone picture and adjust the delay. The logic analyzer's intuitive interface also provides immediate feedback for signals that are too fast for the timescale by changing the waveform to red. There is no need to wait for a complete reading.

Logiscope version 1.0.12 is available for free download from Apple, Inc. The LogiScope app is made for iPod Touch (3rd, and 4th generation), iPhone 4S, iPhone 4, iPhone 3GS, iPad 3, iPad 2, and iPad. LogiScope hardware costs **$389.99**.

Oscium
www.oscium.com

ULTRA-LOW POWER, ULTRA-COMPACT SENSOR HUB MCU FOR SMARTPHONES

The **ML610Q792 Sensor Hub** is a small, ultra-low-power microcontroller designed for the integrated, low-power control of multiple sensors in smartphones. The microcontroller enables the deployment of new cloud-based features and services in smartphones that would otherwise be restrictive due to increased power consumption. It logs user environment data and detects user activity without disturbing the host processor. The ML610Q792's low-power consumption prolongs battery life and enables support for wireless communication in compact products (e.g., pedometers and smartphone accessories).



The ML610Q792 microcontroller is equipped with LAPIS Semiconductor's high-performance 8-bit RISC core and 16-bit coprocessor for arithmetic calculations. Dual interfaces enable both sensor connection and connecting to the main chipset. The ML610Q792 microcontroller's additional features include: an ultra-compact, 48-pin wafer-level chip-scale package (WL-CSP) form factor (3.1 mm × 3 mm) package for small designs; comprehensive software support, including sensor drivers and key algorithms; the ability to control multiple sensors installed on smartphones; low-power consumption (0.6 µA or less in HALT mode); and built-in 64-KB flash ROM.

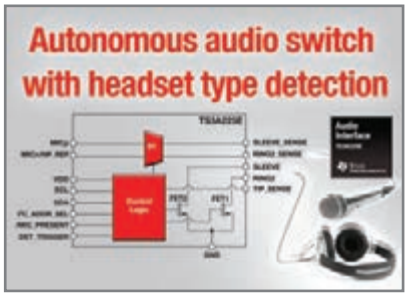
A development board containing multiple sensors and drivers is provided with a software development kit (SDK) that includes sample source code and libraries for data logging, calorie calculation, pedometer function, and device drivers to interface with the host processor. Driver firmware for each sensor is provided along with sample programs for data logging, pedometer, and activity monitoring with activity detection (i.e., walking, running, and riding). The SDK also includes software development tools and the uEASE emulator, an on-chip debug emulator for software debugging that uses the microcontroller's debugging function to run/stop programs.

ML610Q792 samples cost **$3.73**. The SDK costs **$869.57**.

LAPIS Semiconductor Co., Ltd.
www.lapis-semi.com/en

DETECTION & CONFIGURATION IC

The **TS3A225E** is an audio headset detection and configuration switch integrated circuit (IC) that enables universal headset support for audio applications in a single chip. The IC is designed to eliminate detection, leakage, and click-pop problems caused by using discrete components in end equipment (e.g., smartphones, tablets, laptops, audio docks, and home audio applications).



The TS3A225E supports three-prong (without microphone) and four-prong (with microphone) audio headsets from all manufacturers. The IC detects the configuration of the microphone and ground pins on a four-prong headset and appropriately routes them via a 2 × 2 crosspoint switch. In addition, the IC provides control over system operation and debugging by reporting headset information over an I^2C interface.

The TS3A225E is offered in 16-pin QFN and WCSP packages and costs **$0.32** in 1,000-unit quantities.

Texas Instruments, Inc.
www.ti.com

MODULE FOR RUGGED APPLICATIONS

The Ampro by ADLINK **Express-IBR** is a COM Express Type 6 module that supports the quad-core and dual-core third-generation Intel Core i7 processors and Mobile Intel QM77 Express chipset. The modular, power-efficient solution is designed for applications running in space-constrained, extremely rugged environments.

The Express-IBR is powered by a quad- or dual-core third-generation Intel Core processor and provides support for USB SuperSpeed 3.0, PCI Express (PCIe) Gen 3, and up to three independent displays.



The Express-IBR with dual-core processor is validated for reliable performance in extended temperatures ranging from –40 to 85°C and features a 50% thicker PCB for high vibration tolerance.

Contact ADLINK for pricing.

ADLINK Technology, Inc.
www.adlink.com

CONNECTIVITY SOLUTIONS FOR RASPBERRY Pi MINI COMPUTER

The **TTL-232R-RPi** is a USB-to-TTL serial UART debug cabling solution that incorporates FTDI's FT232RQ high-performance interface IC. It is complemented by the **RPi USB Hub Module**, which is designed to connect to a Raspberry Pi (RPi) unit and expand the number of interfaces that can be accessed downstream from a host.

RPi is a simple, ultra-low-cost, compact computer board that runs the Linux operating system. The board utilizes a 700-MHz, ARM-based processor and has 256-MB SDRAM memory.



When using a laptop or desktop PC, you can use FTDI's debug cable to interface with the RPi board and perform in-depth analysis on developing software applications. The 1-m long cable connects directly to the RPi's serial port, which outputs kernel debug messages for the PC to acquire. The FT232RQ USB-to-Serial UART interface IC within the TTL-232R-RPi cable provides a USB-to-asynchronous serial data transfer path that supports data rates from 300 bps to 3 Mbps at 3.3-V TTL levels. The chip handles all the USB signaling and protocol requirements.

The RPi USB Hub Module utilizes a USB-to-UART/MPSSE IC along with a GL850G USB hub controller IC. It provides engineers with up to four downstream USB 2.0 high-speed (480 Mbps) ports and supports asynchronous serial data transfer rates from 300 bps to 12 Mbps at 3.3-V TTL levels. Synchronous serial data rates of up to 30 Mbps can be handled on JTAG, SPI, and I^2C interfaces. The module's dimensions are 33.45 mm × 65 mm.

An operational temperature range covering –40°C to 85°C is supported by both the cable and the hub module, which are both RoHS-compliant.

TTL-232R-RPi debug cables cost approximately **$15** each. RPi Hub Modules cost approximately **$34** each.

Future Technology Devices International, Ltd. (FTDI)
www.ftdichip.com

ZERO-DRIFT OP-AMPS FOR SIGNAL CONDITIONING, INSTRUMENTATION, & PORTABLE SENSOR APPLICATIONS

The **MCP6V11** and the **MCP6V31** zero-drift, op-amps operate with a single supply voltage as low as 1.6 V and a quiescent current as low as 7.5 µA. The op-amps enable the development of portable medical products integrated with higher efficiency and signal-conditioning hardware and software. Industrial applications—such as portable sensor conditioning and instrumentation requiring low power, small form factors, and temperature considerations—can benefit from the op-amps' optimized performance, low quiescent current, and low operating voltage.

The devices utilize Microchip's CMOS technology, which requires less current to operate the amplifier while delivering longer battery life and minimal thermal-related challenges. The MCP6V11/31 family's self-correcting architecture provides a maximum input offset voltage of 8 µV for ultra-low offset and low-offset drift, enabling maximum accuracy across time and temperature. The MCP6V11 offers 80 kHz of gain bandwidth product, with a low typical quiescent current of only 7.5 µA. The MCP6V31 provides 300 kHz of gain bandwidth product, combined with a low typical quiescent current of 23 µA. The MCP6V11 and MCP6V31 op-amps are both available in small five-pin SOT-23 and five-pin SC-70 packages.

Pricing for the MCP6V11 starts at **$0.82** each in 5,000-unit quantities. Pricing for the MCP6V31 starts at **$0.81** each.

Microchip Technology, Inc.
www.microchip.com



NPN

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

USB STC 8051 MCU Programmer
Free For 8051 Developers
$0 Add to cart

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

(L3G4200D) 3 Axis Digital Gyroscope
Integrate Gyroscope Easily
$16.5 Add to cart

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

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

What's your flavor
Linux or DOS?

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

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

Made in the USA - Distributed Worldwide

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

MEMBER PROFILE: Mark Csele



Member Name: Mark Csele

Location: Welland, Ontario, Canada (in the Niagara region, about 30 minutes away from Niagara Falls)

Education: BS Physics (honors), BEng Computers, and PEng (in Ontario)

Occupation: Full-time professor at Niagara College, Canada

Member Status: Mark has subscribed since 1994, but he said he read the "Circuit Cellar" column regularly when it appeared in *Byte* magazine. It was his favorite column.

Technical Interests: At Niagara College, Mark has taught embedded systems courses employing both "normal" PIC processors as well as DSPs. He said he also "moonlighted" as a professional engineer building embedded power-control systems (primarily three-phase and controlling loads of up to 30,000 hp).

Mark's current full-time teaching load consists entirely of courses on laser engineering. His book, *Fundamentals of Light Sources and Lasers*, was published in 2003 (Wiley-Interscience).

Mark said he has two older solid-state pinball machines (circa 1980 and 1991) that he keeps running. He also has a small collection of "classic" mini and micro computers including PDP-8s, PDP-11s, and several Ohio Scientific (6502-based) machines.

Most Recent Embedded Tech-Related Purchase: Oddly enough, a fluorescent desk lamp with a powerful magnifier. With chip packages shrinking, prototyping has become more challenging often requiring a chips to be mounted on an adapter even for basic prototyping, Mark said. Mark's most recent "actual hardware" acquisition was an ICD-3 debugger (which has proven quite fast for debugging when viewing large blocks of registers). Although he has a decently equipped workshop, Mark has homebuilt equipment, including his logic analyzer, which is actually based on a 1995 design presented in *Circuit Cellar*. Mark's has been updated several times and now sports a 50-msps rate and a native USB interface (which uses a Microchip Technology PIC18F4550). That analyzer has proven invaluable when prototyping and debugging MCU-based projects, he said.

Current and Recent Projects: Mark is working on a TEA laser controller featuring a Microchip PIC18F452 MCU for the laser lab at the college. Programmed in Microchip C, this device runs as a state machine and controls the safety aspects of the laser (e.g., interlocks) as well as the voltage on the main capacitor bank in the laser (which can charge to 40 kV and can store up to 200 J of energy).

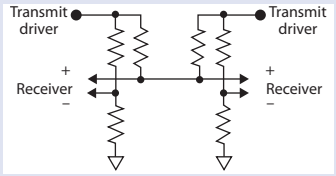
Mark's most recently completed project (see the nearby photo) is a digital oscilloscope. This project was designed for students to complete in a college workshop course in which they manufacture the board and case, mount components (including many SMT packages), and test/debug the completed project. The hardware was designed by a colleague, while Mark produced the firmware. Written in MPASM and running on a Microchip dsPIC33FJ MCU, the scope features two channels and can sample at 1 msps. Mark said the design is very simple. Aside from the MCU (with on-chip ADC), it uses only a graphical LCD and several op-amps for signal conditioning.

Thoughts on the Future of Embedded Technology: Like most engineers, Mark's "wish list" for an embedded device includes more memory, more MIPS, and more peripherals. Manufacturers have delivered over the past few decades, Mark said. He has seen many changes since his introduction to embedded technology in the 1980s from UV-EPROM technology to flash, incorporation of both RAM and ROM on chips, and the addition of a large number of peripherals, including ADCs right on these chips. Mark noted that recent offerings even include a DMA controller (a boon when moving huge amounts of data). He can now do, on a single chip, what would have taken a board full of chips only two decades ago. With MCUs embedded into almost everything, Mark hopes that software technology will keep up. Concerns like "software safety" still dominate many applications, Mark said.

Edited by David Tweed

Test Your EQ

ANSWERS for Issue 264



Answer 1a—Why not? Hybrids work just as well at DC as they do for audio; you just need a receiver with balanced inputs, like an RS-422 buffer (as shown in the figure to the right).

All resistors are the same value (e.g., 4,700 Ω) and the transmit driver needs to be a voltage source (low impedance). If the transmitter switches between, say, 0 and 5 V, the opposite receiver will see a voltage differential of 0 and 2.5 V, respectively, while the local receiver will just see 0 V.

For long lines, you'll probably want to use lower resistances and you'll want to limit the slew rate of the transmitter so that the receiver doesn't produce glitches on the transitions of the local transmitter.

If the RS-422 receiver is replaced with an op-amp differential amplifier with a gain of 2, then any analog voltage transmitted by one end will be reproduced at the other end.

Answer 1b—No. A hybrid is just a bridge circuit, with one arm of the bridge replaced by the line and the termination at the far end. The transmit signal is applied to two opposite corners of the bridge and the receive signal is taken from the other two corners.

In order to provide the Tx/Rx isolation, the bridge must be balanced, which in the example above, means the lower resistor on each side must match the impedance of the line/far end combination. For DC and short lines, a simple resistor suffices. At audio frequencies and with the long unshielded twisted pairs used in telephony, a more complex matching impedance is required.

Transformers are used only because it's the easiest way (and the only passive way) to get a balanced drive and/or receive signal—the transmit driver and receiver cannot share a ground. In order to mass produce phones that were dirt cheap, yet simple and reliable, the phone company figured out how to use a multi-winding transformer to provide the both the isolation and the balanced/unbalanced conversion in both directions, usually with a single resistor and capacitor to provide the line matching. As noted, modern electronic phones use active electronics to achieve the same things. As always, the theory is simple, but the practical implementations can get complicated.

Answer 2a—If we restrict the discussion to unit vectors at various angles A, the x component is cos(A) and the y component is sin(A), and the correct magnitude is 1. Furthermore, let's concentrate on angles between 0 and 45°—then we know that both cos(A) and sin(A) are positive and that cos(A) > sin(A). (The absolute value and compare operations provide the symmetry that covers the rest of the unit circle.) The approximation then gives the result:

$$\text{Magnitude} = \cos(A) + \sin(A)/3$$

Graphing this shows that this is most negative (0.943) at 45° and most positive (1.054) at approximately 18.4°. (The actual angle is given by atan(1/3). Can you show why?) The peak error is therefore –5.7%, +5.4%.

Answer 2b—Yes. One more multiplication operation gives a result with less than 4% error:

$$\text{Magnitude} = 0.960433 \times \max(|x|, |y|) + 0.397826 \times \min(|x|, |y|)$$

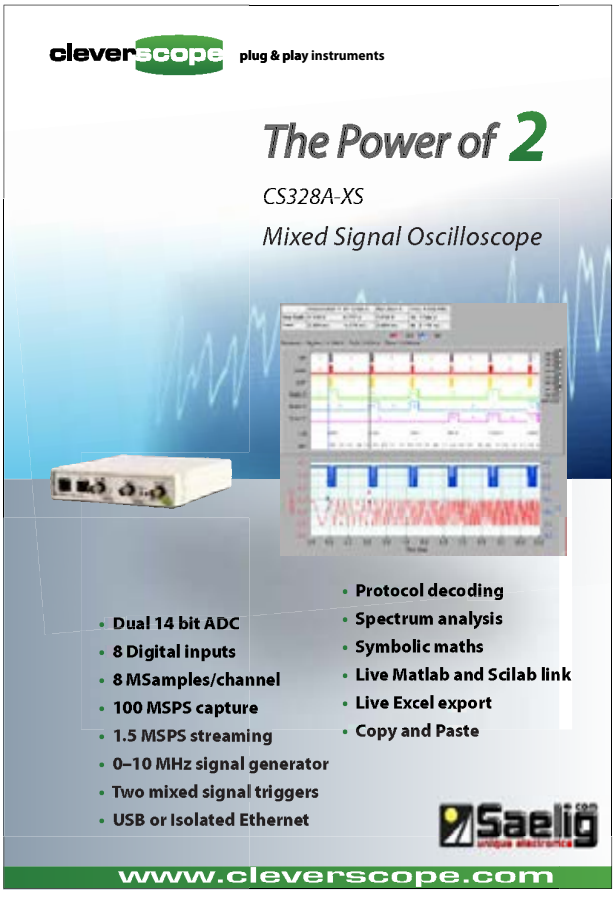
This function is most negative at 0 and 45°, and most positive at 22.5°. The error is ± 3.96%. This form is well suited to DSPs that have multiply-accumulate units. The two constants can be expressed as 62943/65536 and 26072/65536, respectively.

Contributed by David Tweed

What's your EQ?—The answers are posted at www.circuitcellar.com/eq/

You may contact the quizmasters at eq@circuitcellar.com





by Mark Pedley (USA)

eCompass

Build and Calibrate a Tilt-Compensating Electronic Compass

A modern smartphone contains a built-in electronic compass (eCompass). How does the tilt compensation work, and how is the eCompass calibrated for the magnetic interference from the circuit board? This article describes how you can use the high-performance consumer accelerometers and magnetometers developed for the smartphone market to add a tilt-compensated eCompass to your own microcontroller project for less than $5.

Whenever a new high-volume consumer electronics market develops, the semiconductor companies are never far behind, providing excellent components at surprisingly low prices. The market for sensors in consumer products is a recent example. It all started with an accelerometer used to select between portrait and landscape display orientations and then, with the addition of a magnetometer, evolved into the electronic compass (eCompass) used to align street maps to the smartphone's geographic heading or to enable augmented reality overlays. As a result, high-volume pricing for smartphone accelerometer and magnetometer sensors is now less than $1 each.

A magnetometer sensor alone cannot provide an accurate compass heading for two reasons. First, the magnetic field measured at the magnetometer varies significantly with tilt angle. Second, the magnetometer requires calibrating not only for its own offset but also against spurious magnetic fields resulting from any nearby ferromagnetic components on the circuit board. This article describes how the accelerometer is used to compensate the magnetometer for tilt and includes a simple technique for calibrating the magnetometer.

SENSOR SELECTION

The accelerometer should be three axis and capable of operating in the ±2-g range with a minimum of 10 bits of resolution. The output of a 10-bit accelerometer operating in the ±2-g range will change by 512 counts as the accelerometer is rotated 180° from pointing downward to upward, giving an average sensitivity of one count per 0.35° change in tilt. This is more than adequate for tilt-compensation purposes.

It is important to check the accelerometer datasheet for the "0-g offset accuracy" which is the output when the accelerometer is in 0-g freefall. Since this value is a constant additive error on each accelerometer channel, it adds a bias in the calculated tilt angles, so look for accelerometers where this parameter does not exceed 50 mg.

The magnitude of the earth's geomagnetic field is typically about 50 µT with a horizontal component that varies over the earth's surface, from a maximum of about 40 µT down to 0 at the geomagnetic poles. If an eCompass is required to operate in horizontal geomagnetic fields down to 10 µT (in arctic Canada, for example) with a noise jitter of ±3°, then a back-of-the-envelope calculation indicates that a magnetometer with a maximum noise level of 0.5 µT is needed.

Most of my projects have used Freescale's MMA8451Q Xtrinsic three-axis, 14-bit accelerometer and MAG3110 three-axis magnetometer. The MMA8451Q is supplied in a 3-mm × 3-mm × 1-mm, 16-pin QFN package and provides a 14-bit data output with ±30-mg, 0-g offset accuracy. The MAG3110 magnetometer is supplied in a 2-mm × 2-mm × 0.85 mm, 10-pin DFN package and provides a measurement range of ±1,000 µT with 0.1-µT resolution and a noise level down to 0.25 µT. Both parts operate with a supply voltage between 1.95 V and 3.6 V.

Similar sensors are supplied by Asahi Kasei (AKM), Kionix, STMicroelectronics, and other manufacturers. Your best strategy is to go to the manufacturers' websites and make a list of those that provide samples in single units or low-volume

Listing 1—C source code for the tilt-compensated eCompass

```
/* multiplicative conversion constants */
#define DegToRad 0.017453292F
#define RadToDeg 57.295779F

/* global variables */
float Psi, The, Phi;              /* yaw, pitch, roll angles in deg */
float Vx, Vy, Vz;                 /* hard iron calibration coefficients */

/* NED tilt-compensated e-Compass function */
void eCompass(float Bx, float By, float Bz, float Gx, float Gy, float Gz)
{
float sinAngle, cosAngle; /* sine and cosine */
float fBfx, fBfy, fBfz;  /* calibrated mag data in uT after tilt correction */

/* subtract off the hard iron interference computed using equation 9 */
Bx -= Vx;
By -= Vy;
Bz -= Vz;

/* calculate roll angle Phi (-180deg, 180deg) and sin, cos */
Phi = atan2(Gy, Gz) * RadToDeg;           /* Equation 2 */
sinAngle = sin(Phi * DegToRad);           /* sin(Phi) */
cosAngle = cos(Phi * DegToRad);           /* cos(Phi) */
/* de-rotate by roll angle Phi */
Bfy = By * cosAngle - Bz * sinAngle;      /* Equation 5 y component */
Bz = By * sinAngle + Bz * cosAngle;       /* Bz=(By-Vy).sin(Phi)+(Bz-Vz).cos(Phi) */
Gz = Gy * sinAngle + Gz * cosAngle;       /* Gz=Gy.sin(Phi)+Gz.cos(Phi) */

/* calculate pitch angle Theta (-90deg, 90deg) and sin, cos*/
The = atan(-Gx / Gz) * RadToDeg;          /* Equation 3 */
sinAngle = sin(The * DegToRad);           /* sin(Theta) */
cosAngle = cos(The * DegToRad);           /* cos(Theta) */
/* de-rotate by pitch angle Theta */
Bfx = Bx * cosAngle + Bz * sinAngle;      /* Equation 5 x component */
Bfz = -Bx * sinAngle + Bz * cosAngle;     /* Equation 5 z component */

/* calculate yaw = ecompass angle psi (-180deg, 180deg) */
Psi = atan2(-Bfy, Bfx) * RadToDeg;        /* Equation 7 */

return;
}
```

packs of up to five devices. With a bit of luck, you may be able to get both the accelerometer and magnetometer sensors for free. Add a handful of decoupling capacitors and pull-up resistors and you should be well within the $5 component cost.

Each reader has a preferred microcontroller to read the raw data from the two sensors and implement the eCompass. This article assumes the microcontroller provides an I²C bus to interface to the sensors, supports floating-point operations whether natively or through software emulation libraries, and has a few spare bytes of program and data memory to add the 15 executable lines of eCompass code in Listing 1.

LAYOUT & BOARD BRING-UP

Once you've selected your sensors, the next step is to design the accelerometer and magnetometer daughterboard with I²C bus connection to the microcontroller. Reference schematics for the MMA8451Q and MAG3110 are provided in the sensor datasheets and reproduced in Figure 1.

Don't waste any time rotating the accelerometer or magnetometer packages to align their x-, y-, and z-sensing directions to each other since this will be fixed later in software. But do ensure the sensor board will not be mounted in the immediate vicinity of any ferromagnetic materials since these will produce a constant additive magnetic field termed the "hard-iron field." The most common ferromagnetic materials are iron, steel, ferrite, nickel, and cobalt. Non-ferromagnetic materials are all safe to use (e.g., aluminum, copper, brass, tin, silver, and gold).

The calibration process described later enables the estimation and software subtraction of any hard-iron field, but it's good practice to minimize hard iron interference at the design stage. Remember, a current trace will create a cylindrical magnetic field that falls off relatively slowly with the inverse of distance, so place the magnetometer as far away from high current traces as possible. A 0.1-A current trace at 10-mm distance will produce a 2-µT magnetic field, four times our 0.5-µT error budget, only reducing to 0.5 µT at a 40-mm distance. More detailed layout guidance is provided in Freescale Semiconductor's application note AN4247: "Layout Recommendations for PCBs Using a Magnetometer Sensor."

You'll be surprised at the number of features implemented in the latest consumer sensors (i.e., freefall detection, high- and low-pass filtering options, automatic portrait and landscape detection, etc.), but disable all these since you simply want the raw accelerometer and magnetometer data. Configure the accelerometer into a 2-g range and check that you can read the x, y, and z accelerometer and magnetometer data (in units of bit counts) from the sensors' internal registers at a sampling rate of between 10 Hz and 50 Hz. Smartphones commonly use IDH3 to minimize power consumption while anything above 50 Hz is overkill. Check the accelerometer datasheet for the conversion factor between counts and g (4,096 counts per g for the MMA8451Q in ±2-g mode) and use this to scale the x, y, z accelerometer readings into units of g. Do the same for the x, y, z magnetometer data again taking the conversion factor from the magnetometer datasheet (10 counts per µT for the MAG3110).

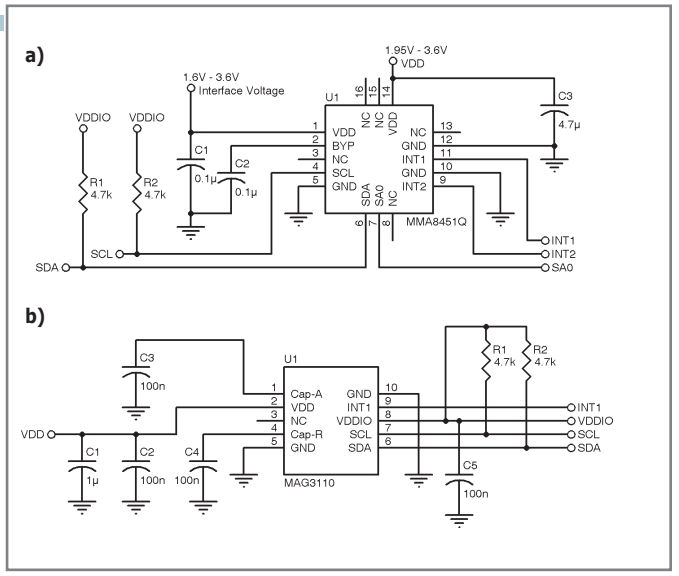


Figure 1—Schematics for **(a)** MMA8451Q and **(b)** MAG3110 sensors

COORDINATE SYSTEM

The equations and C software in Listing 1 use the "aerospace," or "x-North y-East z-Down," coordinate system depicted in Photo 1. This defines the initial eCompass orientation to be where the x-axis points north, the y-axis points east, and the z-axis points downwards. The three orientation angles, roll (ϕ), pitch (θ), and compass heading, or yaw (ψ)—are defined as clockwise rotations about the positive x, y, and z axes— respectively. Photo 1 also shows the earth's gravitational vector which points downward with magnitude of 1 g or 9.81 ms^{-2} and the earth's geomagnetic field vector, which points downward from horizontal (in the northern hemisphere) by the inclination angle δ to give a horizontal component $B_0 \cos\delta$ and a downward component $B_0 \sin\delta$. Based on how your eCompass housing will be held, you should be able to assign the compass-pointing direction or x-axis, the downward or z-axis, and the y-axis, which should point to the right to complete a right-handed coordinate system.

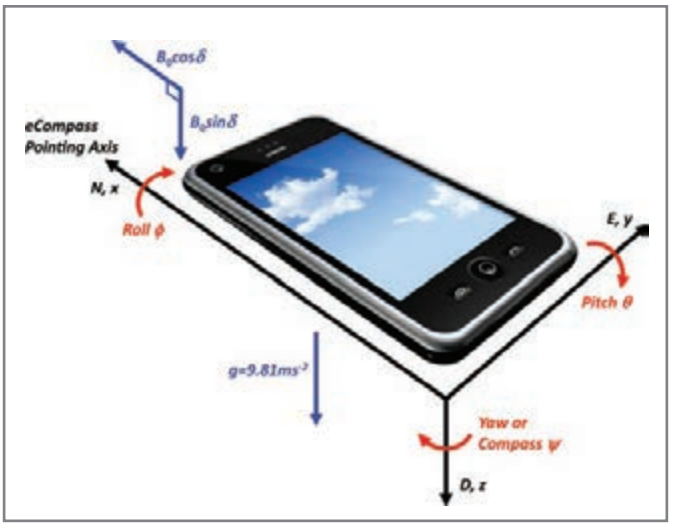


Photo 1—The aerospace north-east-down coordinate system

AXIS ALIGNMENT & MAGNETIC CALIBRATION

You now need to align the sensor data to the aerospace coordinate system. As with all work with magnetometers, this should be performed on a wooden table well away from any laboratory power supplies or steel furniture. Place the eCompass flat and upright so the z-axis points downward and is aligned with gravity. Check that the accelerometer z-axis reads approximately 1 g and the x- and y-axes are near 0. Invert the eCompass so its z-axis points upward and check that the z-axis now reads approximately −1 g. Repeat with the x- and y-axes pointing downward and then upward and check that the x- and y-axis accelerometer readings are near 1 g and −1 g, respectively. It's not important if the accelerometer readings are a few tens of mg away from the required reading since all you're doing here is correcting for gross rotations of the sensor packages and the sensor daughterboard in multiples of 90°. Any needed correction will be unique for your board layout and mounting orientation but will be no more complicated than "swap the x- and y-accelerometer channels and negate the z-channel reading." Code this accelerometer axis mapping into your software and don't touch it again.

Figure 2 may help explain this visually. The accelerometer sensor measures both gravity and linear acceleration and, in the absence of any linear acceleration (as is the case when sitting on a desk), the magnitude of the accelerometer reading will always equal 1 g, and therefore, lie on the surface of a 1-g sphere, irrespective of the orientation. The six measurements lie on the vertices of an octahedron inscribed within the 1-g sphere and the axis mapping simply rotates and reflects the octahedron as needed until the accelerometer channels are correctly aligned.

The magnetometer axis alignment is similar to that of the accelerometer, but makes use of the geomagnetic field vector. Place the eCompass flat, upright, and pointing northward and then rotate in yaw angle by 270° to the east, south, and finally west. The x-channel magnetometer reading should be a maximum when the eCompass is pointed north and a minimum when pointed south. The y-channel magnetometer reading should be a minimum when the eCompass is pointed east and a maximum when pointed west. The z-channel reading should be approximately constant since the vertical component of the geomagnetic field is constant irrespective of rotation in yaw.

Then invert the eCompass on the desk and repeat the process. As before, the magnetometer x-axis reading should be a maximum when the eCompass is pointed north and a minimum when pointed south. But now, because of the inverted position, the magnetometer y-axis should be a maximum when the eCompass is pointed east and a minimum when pointed west. The magnetometer z-axis reading should still be constant but, in the northern hemisphere, lower than the previous upright readings since the magnetometer z-axis is now aligned against the downward component of the geomagnetic field vector.

Figure 3 shows upright and inverted magnetometer measurements taken in the northern hemisphere with a 270° compass rotation. The maximum and minimum of the x- and y-axis magnetometer measurements occur at the expected angles and the z-axis measurement is less when inverted than when upright. These magnetometer axes are therefore

COMPLETE ENGINEERING SOLUTIONS

Start here.

Get direct, one-on-one technical support from real engineers with no go-between and no waiting. Access industry, manufacturer and legislative experts on our community. And find thousands of technical documents, videos & tools— all in one source. Engineering expertise starts at Newark element14.

9 out of 10 customers recommend Newark element14 Technical Support

– Customer feedback studies

Newark™ element14



HOW MAY WE HELP YOU TODAY?

TECHNICAL SUPPORT: 1.877.736.4835
COMMUNITY: element14.com
WEBSITE: newark.com
LEARN MORE: newark.com/together

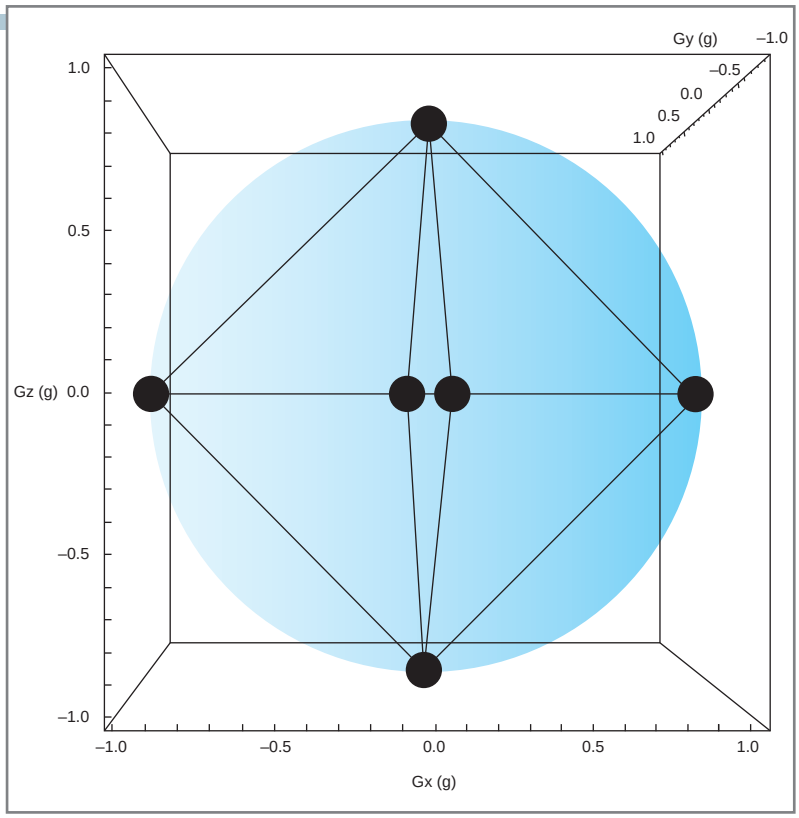


Figure 2—Accelerometer axis alignment points

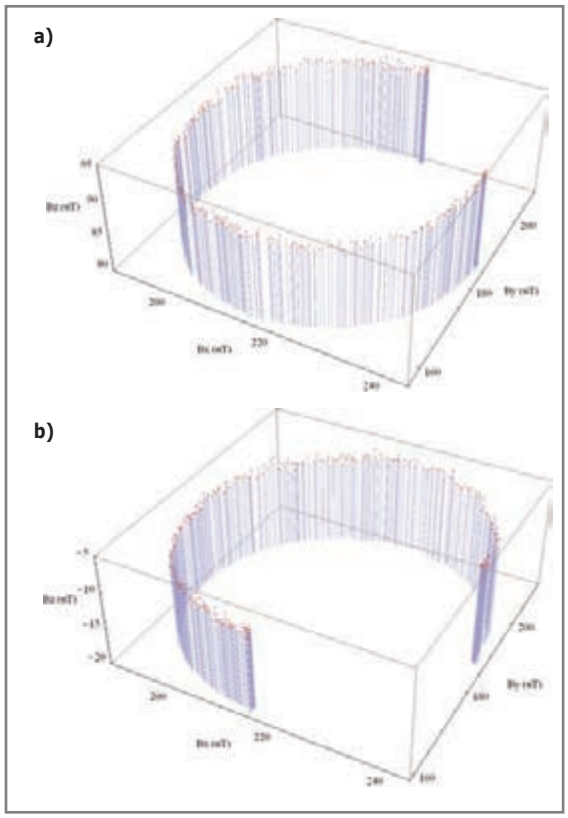


Figure 3—The upright (a) and inverted (b) magnetometer measurements

correctly aligned but, as with the accelerometer correction, swap and negate the measurements from your three magnetometer channels as needed until correctly aligned and then lock down this part of your code.

A lot can be learned by closely looking at the measurements in Figure 3. The x- and y-magnetometer measurements lie on a circle with radius of approximately 25 µT enabling us to deduce that the horizontal geomagnetic field is approximately 25 µT. But the measurements are offset from zero by the magnetic "hard iron" interfering field, which results from both permanently magnetized ferromagnetic materials on the circuit board and from a zero-field offset in the magnetometer sensor itself. Consumer sensor manufacturers long ago realized it was pointless to accurately calibrate their magnetometers when their target market is smartphones, each with a different hard-iron interfering field. The magnetometer sensor offset is, therefore, calibrated together with the circuit board hard-iron magnetic field. For now, simply note that the x and y components of the hard iron offset have values of approximately 215 µT and 185 µT. A simple method to determine all three hard-iron components is described later.

CALCULATION OF THE ROLL & PITCH ANGLES

In the absence of linear acceleration, the accelerometer reading g after arbitrary rotation in yaw (ψ), then pitch (θ), and finally roll (ϕ) from the smartphone starting position (flat, upright, and pointing to magnetic north) is simply the earth's gravitational field vector (1 g downward in the positive z direction) after application of roll, yaw, and pitch rotation matrices R_X (ϕ), R_Y (θ), and R_Z (ψ). The resulting accelerometer reading is:

$$G = \begin{pmatrix} G_X \\ G_Y \\ G_Z \end{pmatrix} = R_X(\phi)\, R_Y(\theta)\, R_Z(\psi) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \quad [1]$$
$$\begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\sin\theta \\ \cos\theta \sin\phi \\ \cos\theta \cos\phi \end{pmatrix}$$

It's worth emphasizing the point that the mathematics of the tilt-compensated eCompass assume that the accelerometer reading is entirely the rotated gravitational field without any linear acceleration. This is a perfectly reasonable assumption for the smartphone market, where users will naturally hold the phone steady to see the compass display, but readers should be aware that they will get erroneous compass headings from an eCompass located in a radio-controlled airplane pulling high g aerobatics or even when shaking the eCompass.

It should be no surprise that Equation 1 has no dependence on the yaw (or compass heading) angle ψ. All accelerometers are unaffected by rotations about a vertical axis. Of course, if accelerometers could measure rotation about the vertical gravitational field, we would not need a magnetometer to compute the compass heading.

Equation 1 can easily be solved for the roll angle ϕ and pitch angle θ from the x, y, and z accelerometer channel readings G_X,


RL78 GREEN ENERGY CHALLENGE
The RL78 Green Energy Challenge ends August 31!
Save energy, improve efficiency, and make the world a greener place. Develop an innovative, low-power design solution using Renesas' RL78 MCU and you could win $8,000, plus a trip to Renesas DevCon!
Don't delay. The RL78 Green Energy Challenge ends soon!
For complete details, visit
www.circuitcellar.com/RenesasRL78Challenge
Second Prize $5,000
First Prize $8,000 & a trip to Renesas DevCon!
Third Prize $3,000
Five Honorable Mentions $500
RENESAS
RL78
In association with
Elektor and Circuit Cellar
@RenesasAmerica / www.facebook.com/renesasamerica

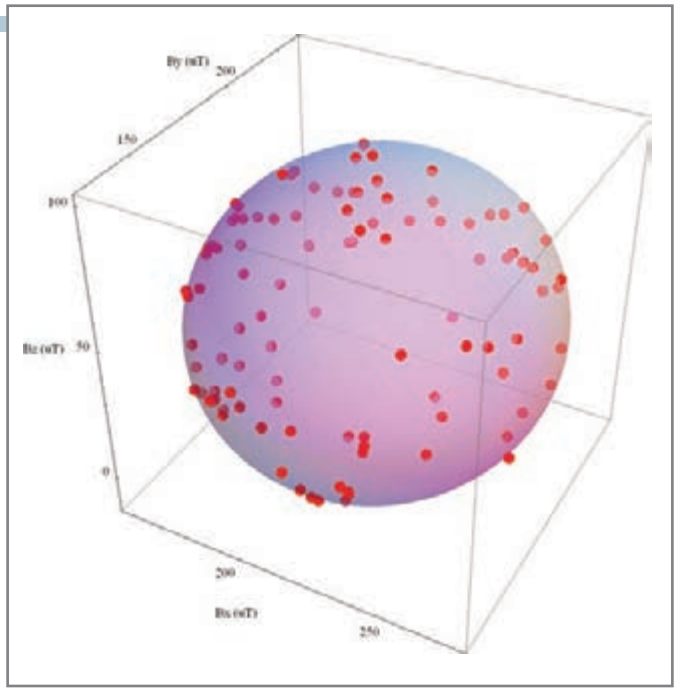


Figure 4—Distribution of magnetometer measurements under arbitrary rotation

G_Y, and G_Z. Simple algebra gives:

$$\tan\phi = \frac{G_Y}{G_Z} \quad [2]$$

$$\tan\theta = \left(\frac{-G_X}{G_Y \sin\phi + G_Z \cos\phi}\right) \quad [3]$$

The range of the roll angle ϕ is −180° to 180° so the C atan2 function is used in Listing 1 to calculate θ. The pitch angle θ only varies between −90° and 90°, so the standard C atan function is used in Listing 1 to calculate θ. Don't try to simplify the algebra by noting that $G_X = -\sin\theta$ in Equation 1 and computing $\theta = \sin^{-1}(-G_X)$. Even without linear acceleration, sensor noise will make G_X occasionally go outside the range of −1 g to 1 g, making the arcsine calculation nonsensical. The arctangent calculations in Listing 1 are stable in comparison since the atan and atan2 functions can take any argument from −∞ to ∞.

The next step is to program up Equations 2 and 3, remembering that most programming languages return angles in radians requiring a multiplication by 180/π, or approximately 57.3, to convert to degrees. Place the eCompass flat on the table and check that the calculated roll and pitch angles are approximately 0°. Rotate the eCompass upward and check that the computed pitch angle is positive, reaches 90° when pointing vertically upward, and then decreases to 0° when the eCompass is inverted. Repeat with the eCompass pointed slightly downward and check that the computed pitch is negative. Then apply a positive roll rotation by raising the left side of the eCompass and check that the computed roll angle is positive, passes through 90° when the eCompass is on its right edge, and finally reaches 180° when the eCompass is flat but inverted. Repeat the roll test by lowering the left side of the eCompass and check that the compute roll angle is negative. If you find the angles are swapped or negated from what is expected, you'll need to return to the accelerometer axis alignment procedure.

CALCULATING THE COMPASS HEADING ANGLE

In a similar manner as Equation 1, the magnetometer reading B can be modeled as the sum of the rotated geomagnetic field plus the hard-iron interfering field vector V to give:

$$B = \begin{pmatrix} B_X \\ B_Y \\ B_Z \end{pmatrix} = R_X(\phi)\, R_Y(\theta)\, R_Z(\psi)\, B_0 \begin{pmatrix} \cos\delta \\ 0 \\ \sin\delta \end{pmatrix} + V \quad [4]$$

If you subtract off the hard-iron interference V from both sides of Equation 4 and premultiply by the inverse roll and then inverse pitch rotation matrices, you obtain:

$$R_Z(\psi)\, B_0 \begin{pmatrix} \cos\delta \\ 0 \\ \sin\delta \end{pmatrix} = B_0 \begin{pmatrix} \cos\psi\cos\delta \\ -\sin\psi\cos\delta \\ \sin\delta \end{pmatrix} = R_Y(-\theta)\, R_X(-\phi) \begin{pmatrix} B_X - V_X \\ B_Y - V_Y \\ B_Z - V_Z \end{pmatrix}$$
$$= \begin{pmatrix} (B_X - V_X)\cos\theta + (B_Y - V_Y)\sin\theta\sin\phi + (B_Z - V_Z)\sin\theta\cos\phi \\ (B_Y - V_Y)\cos\phi - (B_Z - V_Z)\sin\phi \\ -(B_X - V_X)\sin\theta + (B_Y - V_Y)\cos\theta\sin\phi + (B_Z - V_Z)\cos\theta\cos\phi \end{pmatrix} \quad [5]$$

The x and y components of Equation 5 are shown below in Equation 6:

$$B_0\cos\psi\cos\delta = (B_X - V_X)\cos\theta + (B_Y - V_Y)\sin\theta\sin\phi + (B_Z - V_Z)$$
$$\sin\theta\cos\phi - B_0\sin\psi\cos\delta = (B_Y - V_Y)\cos\phi - (B_Z - V_Z)\sin\phi \quad [6]$$

Taking the ratio of the y and x components in Equation 6 gives the tilt-compensated compass heading angle ψ as:

$$\tan\psi = \left(\frac{-(B_Y - V_Y)\cos\phi + (B_Z - V_Z)\sin\phi}{(B_X - V_X)\cos\theta + (B_Y - V_Y)\sin\theta\sin\phi + (B_Z - V_Z)\sin\theta\cos\phi}\right) \quad [7]$$

We know the roll ϕ and pitch θ angles from the accelerometer so all that remains to compute the compass heading angle ψ from Equation 7 is to find a simple way to calculate the components of the interfering hard-iron magnetic field V. But, if you subtract V from both sides of Equation 4 and then calculate the resulting magnitude squared, you obtain Equation 8. The left-hand side is expanded using Pythagoras's theorem and the right-hand side is simplified by observing that the geomagnetic field magnitude remains B_0 irrespective of the rotations applied.

$$\left\| \begin{matrix} B_X - V_X \\ B_Y - V_Y \\ B_Z - V_Z \end{matrix} \right\|^2 = \left\| R_X(\phi)\, R_Y(\theta)\, R_Z(\psi)\, B_0 \begin{pmatrix} \cos\delta \\ 0 \\ \sin\delta \end{pmatrix} \right\|^2$$
$$\Rightarrow (B_X - V_X)^2 + (B_Y - V_Y)^2 + (B_Z - V_Z)^2 = {B_0}^2 \quad [8]$$

Equation 8 simply states that under arbitrary rotations in roll,

pitch, and yaw, the magnetometer readings lie on the surface of a sphere with radius B_0 centered at the hard-ron interference V_X, V_Y, and V_Z. Figure 4 shows actual magnetometer measurements taken at random orientations together with a sphere fitted to the measurements with radius 57.1 µT and centered at V_X = 216.4 µT, V_Y = 183.6 µT, and V_Z = 44.0 µT. To obtain your own hard-iron calibration, simply record and plot the magnetometer readings under random orientations and estimate your own hard-iron correction V_X, V_Y, V_Z from the center of the resulting sphere. A simple but effective technique is to rotate the eCompass in a figure of eight twisting motions for a few seconds, record the minimum and maximum magnetometer readings, and compute the hard-iron calibration from their average using Equation 9.

$$\begin{pmatrix} V_X \\ V_Y \\ V_Z \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \max(B_X) + \min(B_X) \\ \max(B_Y) + \min(B_Y) \\ \max(B_Z) + \min(B_Z) \end{pmatrix} \quad [9]$$

Equation 9 completes the information needed in Equation 7 to compute the compass heading ψ. Readers interested in a more sophisticated algorithm for least squares fit to the hard-iron calibration should refer to Freescale Semiconductor's Application Note AN4246: "Calibrating an eCompass in the Presence of Hard and Soft-Iron Interference" and visit Freescale's eCompass website for additional C source code and details.

The entire eCompass with tilt-compensation is provided as C code with just 15 executable lines in Listing 1. Readers wishing to add a digital low-pass filter to reduce jitter on the computed angles should take care to use modulo 360° arithmetic to avoid averaging angles jittering between −179° and 179° (pointing south) to 0° (pointing north). Freescale's Application Note AN4248 "Implementing a Tilt-Compensated eCompass Using Accelerometer and Magnetometer Sensors," contains further details and C source code for those interested.

Mark Pedley (mark.pedley@freescale. com) has a Physics degree from Oxford University and now works on sensor fusion algorithms for Freescale Semiconductor in Tempe, Arizona.

RESOURCES

Asahi Kasei Corp., www.asahi-kasei.co.jp/asahi/en.

Freescale Semiconductor, Xtrinsic eCompass Software, www.freescale.com/ecompass.

———, Application Note AN4247: "Layout Recommendations for PCBs Using a Magnetometer Sensor," 2012, www.freescale.com/files/sensors/doc/app_note/AN4247.pdf.

———, Application Note AN4246: "Calibrating an eCompass in the Presence of Hard and Soft-Iron Interference," 2012, http://cache.freescale.com/files/sensors/doc/app_note/AN4246.pdf.

———, Application Note AN4248: "Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors," 2012, http://cache.freescale.com/files/sensors/doc/app_note/AN4248.pdf.

Kionix, Inc., www.kionix.com.

ST Microelectronics, www.st.com.

SOURCES

MMA8451Q Xtrinsic accelerometer and MAG3110 magnetometer
Freescale, Inc. | www.freescale.com

NEED-TO-KNOW INFO

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

Mini Rover 7
Electronic Compassing for Mobile Robotics
by Joseph Miller
***Circuit Cellar* 165, 2004**
Electronic compassing is an excellent way to provide absolute heading information for a mobile robot. Joseph details how it factors in his Mini Rover 7 robot design, which he modeled after the NASA/JPL Rocky 7 Mars rover. Topics: Electronic Compassing, Robot, Heading, Navigation

Digital Compassing
Build a Tilt-Compensated Digital Magnetic Compass
by Vadym Grygorenko
***Circuit Cellar* 188, 2006**
Vadym built a PSoC-based digital compass with a three-axis magneto-resistive sensor featuring tilt compensation.The system stores calibration data in the PSoC's internal EEPROM. Topics: Tilt Compensation, Compass, Calibration

Find a New Direction
A Low-Power Digital Compass
by Jess Marroquin
***Circuit Cellar* 253, 2011**
You can use a low-power microcontroller to build a digital compass. This article details the planning, design, and programming processes. Topic: Digital Compass, Magnetic Sensor, Low-Power Consumption

Launching the "Roboat"
Navigation by GPS and Digital Compass
by Riccardo Rocca
***Circuit Cellar* 115, 2000**
Riccardo wanted to test the feasibility of driving an object using a GPS and a digital compass. In this case, the object is the "Roboat" model boat that can sail autonomously along a course planned in advance. Topic: Digital Compass, GPS, Navigation, Proportional Servo

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



The Man Behind the dsNavCon Design

An Interview with Guido Ottaviani

Guido Ottaviani worked as an analog and digital developer for several years for an Italian communications company. Today, he is a technical manager for a large Italian editorial group. In his spare time, Guido designs autonomous robots, shares his experiences with other self-professed "electronics addicts," and evangelizes robot design. In June, Guido and I discussed his fascination with robotics, his favorite microcontrollers, and some of his innovative projects, such as the dsNavCon system.—Nan Price, Associate Editor

NAN: Where do you live?

GUIDO: I live in wonderful Rome, Italy, the eternal city. Very chaotic but very stimulating, according to many points of view.

NAN: How did you become interested in electronics?

GUIDO: I was 13 at junior high school. A substitute teacher went out of the standard program (technical drawing) and started explaining the Crystal Set radio to us. I was so fascinated by that new world that, four years later, I got a radio amateur license, after a period of SWL activity and the construction of some tube devices as audio amplifiers and long-wave band receivers.

I knew all my youth friends on air, all electronics addicts like me. We spent most of our time in experimenting something.

A little anecdote, just to understand who we were: one afternoon after school, there were four of us in the laboratory trying to make an HF transmitter function. It was built by installing some tubes on an upside-down aluminum baking pan used as the base. Eventually, the transmitter started outputting some power to the load, an incandescent lamp used instead of the antenna. While we were still celebrating our success, looking at the lamp glowing to the rhythm of the modulating voice, our friend's mother came into the room and asked, "Why are you so happy? All this stuff just to light up a lamp?"

That episode didn't decrease our passion, and this passion wasn't an obstacle in my regular life: my wife is the sister of one of those fellows.

NAN: What was the first MCU you worked with? Where were you at the time? Tell us about the project and what you learned.



Guido's workstation

GUIDO: The very first one was not technically an MCU, that was too early. It was in the mid 1980s. I worked on an 8085 CPU-based board with a lot of peripherals, clocked at 470 kHz (less than half a megahertz!) used for a radio set control panel. I was an analog circuits designer in a big electronics company, and I had started studying digital electronics on my own on a Bugbook series of self-instruction books, which were very expensive at that time. When the company needed an assembly programmer to work on this board, I said, "Don't worry, I know the 8085 CPU very well." Of course this was not true, but they never complained, because that job was done well and within the scheduled time.

I learned a lot about how to optimize CPU cycles on a slow processor. The program had very little time to switch off the receiver to avoid destroying it before the powerful transmitter started.

Flow charts on paper, a Motorola developing system with the program saved on an 8" floppy disk, a very primitive character-based editor, the program burned on an external EPROM and erased with a UV lamp. That was the environment! When, 20 years later, I started again with embedded programming for my hobby, using Microchip Technology's MPLAB IDE (maybe still version 6.xx) and a Microchip Technology PIC16F84, it looked like paradise to me, even if I had to relearn almost everything.

But, what I've learned about code optimization—both for speed and size—is still useful even when I program the many resources on the dsPIC33F series.

NAN: What is your current occupation?

GUIDO: After a good period of electronic designing in that electronics company, a colleague of mine and I decided to face up to a challenge. There was the need to build up the technical department for a newspaper company from scratch. Since then, I've learned a lot about editorial systems, picture editing, page layout, computer to plate, paper, ink, web presses, and every technical issue needed to produce and print newspapers and magazines. Now I'm managing a company that takes care of all the technologies for the entire editorial group.

NAN: You worked in the field of analog and digital development for several years. Tell us a bit about your background and experiences.

GUIDO: Let me talk about my first day of work, exactly 31 years ago.

Being a radio amateur and electronics fan, I went often to the surplus stores to find some useful components and devices, or just to touch the wonderful receivers or instruments: Bird wattmeters, Collins or Racal receivers, BC 312, BC 603 or BC 1000 military receivers and everything else on the shelves.

The first day of work in the laboratory they told to me, "Start learning that instrument." It was a Hewlett-Packard spectrum analyzer (maybe an HP85-something) that cost more than 10 times my annual gross salary at that time. I still remember the excitement of being able to touch it, for that day and the following days. Working in a company full of these kinds of instruments (the building even had a repair and calibration laboratory with HP employees), with more than a thousand engineers who knew everything from DC to microwaves to learn from, was like living in Eden. The salary was a secondary issue (at that time).

I worked on audio and RF circuits in the HF to UHF bands: active antennas, radiogoniometers, first tests on frequency hopping and spread spectrum, and a first sample of a Motorola 68000-based GPS as big as a microwave oven.

Each instrument had an HPIB (or GPIB or IEEE488) interface to the computer. So I started approaching this new (for me) world of programming an HP9845 computer (with a cost equivalent to 5 years of my salary then) to build up automatic test sets for the circuits I developed. I was very satisfied when I was able to obtain a 10-Hz frequency hopping by a Takeda-Riken frequency synthesizer. It was not easy with such a slow computer, BASIC language, and a bus with long latencies. I had to buffer each string of commands in an array and use some special precaching features of the HPIB interface I found in the manual.

Every circuit, even if it was analog, was interfaced with digital ports. The boards were full of SN74*xx* (or SN54*xx*) ICs, just to make some simple functions as switching, multiplexing, or similar. Here again, my lack of knowledge was filled with the "long history experience" on Bugbook series.

Well, audio, RF, programming, communications, interfacing, digital circuits. What was I still missing to have a good background for the next-coming hobby of robotics? Ah! Embedded programming. But I've already mentioned this experience.

After all these design jobs, because my knowledge started spreading on many different projects, it was natural to start working as a system engineer, taking care of all the aspects of a complex system, including project management.

NAN: You have a long-time interest in robotics and autonomous robot design. When did you first become interested in robots and why?

GUIDO: I've loved the very simple robots in the toy store windows since I was young, the same I have on my website (Pino and Nino). But they were too simple. Just making something a little bit more sophisticated required too much electronics at that time.

After a big gap in my electronics activity, I discovered a newly published robotic magazine, with an electronic parts supplement. This enabled me to build a programmable robot based on a Microchip PIC16F84. A new adventure started. I felt much younger. Suddenly, all the electronics-specialized neurons inside my brain, after being asleep for many years, woke up and started running again. Thanks to the Internet (not yet available when I left professional electronics design), I discovered a lot of new things: MCUs, free IDEs running even on a simple computer, free compilers, very cheap programming devices, and lots of documentation freely available. I threw away the last Texas Instruments databook I still had on my bookshelf and started studying again. There were a lot of new things to know, but, with a good background, it was a pleasant (if frantic) study. I've also bought some books, but they became old before I finished reading them.

Within a few months, jumping among all the hardware and software versions Microchip released at an astonishing rate, I found Johann Borenstein et al's book *Where Am I?: Systems and Methods for Mobile Robot Positioning* (University of Michigan, 1996). This report and Borenstein's website taught me a lot about autonomous navigation techniques. David P. Anderson's "My Robots" webpage (www.geology.smu.edu/~dpa-www/myrobots.html) inspired all my robots, completed or forthcoming.

I've never wanted to make a remote-controlled car, so my robots must navigate autonomously in an unknown environment, reacting to the external stimuli.

NAN: On your website, you include information and pictures from several robotics events and competitions. Do you enter competitions or are you involved in organizing the events?

GUIDO: I've participated in some of the competitions, but that is not my main goal. The first challenge is always with me. I set my target and do my work until it is achieved. The first target was the SR04 robot from Anderson. It included most of the concepts you must know to make an autonomous robot.

I first tried to build a robot for the Explorer category competitions. You can see some slide shows on my website. This was very popular among the events organized by technical



Massiomo Banzi (left) and Guido Ottaviani at the Romecup 2011

schools, both in the junior (no programming) and the senior version (programmable robots). But that didn't fit my target; it requires more speed than precision. I tried, together with some friends, to involve more people in organizing an "indoor precision competition." We also defined the name, the logo, and the rules. It was RTC, which stood for Robo-Tolomeo Cup, with the name of the big greek cartographer, to prove that this challenge could be a simpler precursor for the better known, more complicated, outdoor Robo-Magellan from Seattle Robotics. But this was not so spectacular, and we were not able to involve enough people.

I won my personal competition when I was able to complete a full UMBmark (the video is on my website) with a centimeter precision. Then, that project was over and I had a new target: the Anderson's jBot (always him!). Right now, I'm designing the foundation of an outdoor robot, Lino.

But I usually participate—at least as an observer—in every competition in Italy I am able to. It is always stimulating, and I always learn a lot over there. Furthermore, I've had the opportunity to meet again the many people I've known in the past years, often thanks to the popularity *Circuit Cellar* gave to me. We are a bunch of robotics fans, spread all over Italy, and in touch every day via the 'Net. Those are the only moments we can meet personally. Each meeting, after a lot of robotics, always ends up in a celebration around some food.

This is one of the other disciplines typically involved in robotics: friendship.

NAN: Speaking of robotics events in Italy, can you tell us about the electronics "scene" in Italy? For instance, in the U.S., hackspaces and "circuit cellars" are cropping up everywhere, from New York to small towns in the southern states to Silicon Valley. These small groups comprise enthusiastic electronics designers of all ages and backgrounds.

GUIDO: The scene is very confused right now. The interest in robotics (much more than just electronics) is growing very quickly, and a lot of different facts pop up every day. One common element is actually the schools. A lot of teachers are involving more and more students on these projects. They understand the potential of robotics. The appeal it has for young people is very useful to stimulate them in a lot of different disciplines. It's easier to explain math, physics, science, and a lot of other things, by immediately using that theory in a real, practical robot. Some university students are showing the teachers how they can teach to the younger students. Those projects are new for most of the current teaching staff, and they are putting in a lot of personal effort to be up-to-date. I've talked with many teachers, one uses his own garage to receive the students in the afternoon and build their robots for an upcoming competition. The school structures have no money to let the laboratories open after hours.

Some association started organizing. "Mondo Digitale" is a group of people who are trying to define the way schools can proceed. The "Romecup 2012" was crowded with a lot of students of all ages, most of them very enthusiastic about robots. It was amazing to listen to a very young boy who precisely explained to me everything about sensors, actuators, and central processors for his robot.

Also Massimo Banzi, participating in these events, evangelizing Arduino and creating the "Officine Arduino" is a very good promoter of robotics.

The parallel reality is the most virtual. There are a lot of forums, groups, and mailing lists about these projects, with many participants, each with his own lab and many ideas to share. The "circuit cellar" is a dream for every one of us, but we have not yet reached the critical mass in any town to organize one.

NAN: Robotics is a focal theme in many of the articles you have contributed to *Circuit Cellar*. One of your article series, "Robot Navigation and Control" (224–225, 2009), was about a navigation control subsystem you built for an autonomous differential steering explorer robot. The first part focused on the robotic platform that drives motors and controls an H-bridge. You then described the software development phase of the project. Is the project still in use? Have you made any updates to it?

GUIDO: After I wrote that article series, that project evolved until the beginning of this year. There is a new switched power supply, a new audio sensor, the latest version of dsNav dsPIC33-based board became commercially available online, some mechanical changing, improvements on telemetry console, a lot of modifications in the firmware, and the UMBmark calibration performed successfully.

The goal is reached. That robot was a lab to experiment sensors, solutions, and technologies. Now I'm ready for a further step: outdoors.

NAN: You wrote another robotics-related article in 2010 titled, "A Sensor System for Robotics Applications" (*Circuit Cellar* 236). Here you describe adding senses—sight, hearing, and touch—to a robotics design. Tell us about the design, which is built around an Arduino Diecimila board. How does the board factor into the design?

GUIDO: That was the first time I used an Arduino. I've always used PICs, and I wanted to test this well-known board. In that case, I needed to interface many I^2C, analog sensors, and an



An interface between PIR sensors and an Isadora program

I²C I/O expander. I didn't want to waste time configuring peripherals. All the sensors had 5-V I/O. The computing time constraints were not so strict. Arduino fits perfectly into all of these prerequisites. The learning curve was very fast. There was already a library of every device I've used. There was no need for a voltage level translation between 3.3 and 5 V.

Everything was easy, fast, and cheap. Why not use it for these kinds of projects?

NAN: You designed an audio sensor for a Rino robotic platform ("Sound Tone Detection with a PSoC Part 1 and Part 2," *Circuit Cellar* 256–257, 2011). Why did you design the system? Did you design it for use at work or home? Give us some examples of how you've used the sensor.

GUIDO: I already had a sound board based on classic op-amp ICs. I discovered the PSoC devices in a robotic meeting. At that moment, there was a special offer for the PSoC1 programmer and, incidentally, it was close to my birthday. What a perfect gift from my relatives!

This was another excuse to study a completely different programmable device and add something new to my background. The learning curve was not as easy as the Arduino one. It is really different because... it does a very different job. The new PSoC-based audio board was smaller, simpler, and with many more features than the previous one. The original project was designed to detect a fixed 4-kHz tone, but now it is easy to change the central frequency, the band, and the behavior of the board. This confirms once more, if needed, that nowadays, this kind of professional design is also available to hobbyists.

SEALEVEL®

Computing/HMI | Serial | I/O

RISC computing power. A wealth of I/O features.

The R9 family is an application-ready platform for your next product design. R9 products deliver RISC computing power using the latest Windows® Embedded CE 6.0 software environment and unmatched I/O features that extend capabilities beyond traditional RISC applications. Choose from single board computers, rugged embedded systems, and panel mount flat panel touchscreen computers. All R9 solutions offer small size, wide operating temperature range, and flexible I/O connectivity.

R9 Features:

- Atmel AT91SAM9263 ARM® Processor
- On-board Serial, Digital, and Analog I/O
- Compatible with Windows Embedded CE 6.0 and Linux
- Rugged Mechanical Construction
- Compact Designs as Small as 4.9" x 3.9"
- -40°C — +85°C Operating Temperature Range

Powered by Windows® Embedded



Custom RISC solutions available – contact us for more information.



sealevel.com > sales@sealevel.com > 864.843.4343

Learn more about RISC computing solutions at **sealevel.com/cir/risc** or scan this QR code with your smart phone.

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

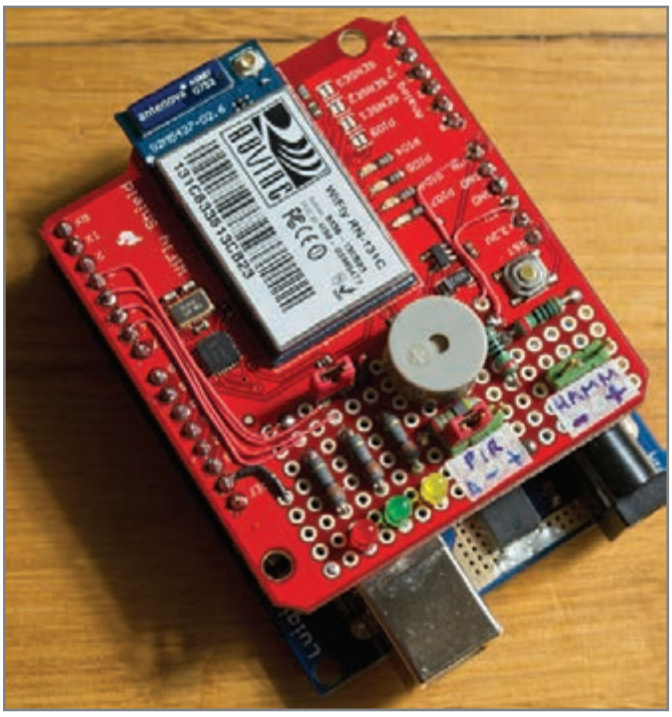
NAN: The Arduino Project originated in Ivrea, Italy. Have you used it? If so, what do you think?

GUIDO: Yes, I've used it and I use it often. I became an Arduino fan after the aforementioned sensor board project. Before then, I was very skeptical. I love all the openness philosophy and the community behind Arduino. It is an idea out of the ordinary, an international collaboration and… I'm proud it is an Italian idea. It is not so powerful in hardware, but you can easily solve most of your electronics issues with it.

I've read a lot about Arduino on many sites and magazines such as Wired Italia (it was on the first published number).

I've also personally met Massimo Banzi at the Romecup 2011. Everything I've read was confirmed by those hours I spent talking with him. He is very friendly and he is an explosion of ideas. In that meeting he fascinated many teachers who started using Arduino too in their lab courses.

I've included several pictures of some of my Arduino-based projects.

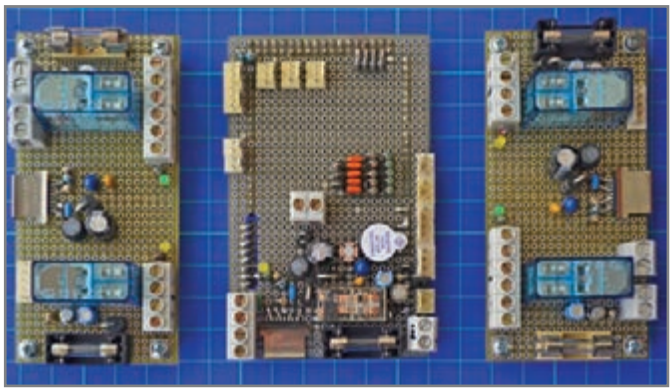


Using a WiFly shield to connect an Arduino to the 'Net

NAN: What are your go-to microcontrollers and embedded platforms these days? Do you have favorites, or do you use a variety of different chips?

GUIDO: I was brought up (in terms of robotics) with Microchip devices. I've used at least one device for almost each PIC family, PIC10F, PIC12F, PIC16F, PIC18F, dsPIC30F, and dsPIC33F. I have an ICD3 and I'm very happy now that the new MPLABX also runs natively on Mac OS X. The flexibility of those devices is very high and you can also develop very sophisticated systems. In one of my latest projects (modifying a parking sensor to obtain a quad sonar) I confirmed once more (if needed) the power of the peripherals also with a "simple" PIC18F2620. Almost the entire project is realized just configuring and connecting peripherals. The software is just simple glue that further connects the peripherals. The exact control you can have on every single bit or execution cycle gives you the opportunity to obtain whatever you want. I use them for all my critical projects, when I need to squeeze the maximum performance by applying all the optimization concepts I know.

On the contrary, if you need to quickly build something very simple, is not so easy with standard Microchip tools. You have no reliable, already-done libraries to configure the peripherals that often differ enough among devices. You spend most of your time just searching for that register's bit that will solve your problem. There is a huge amount of very good documentation but… you have to spend a lot of time.



A homemade shield for an Arduino mega used to create a subsystem that controls batteries and power supply status for Guido's forthcoming robot

In my opinion, this is one of the reasons for the Arduino success. Sometimes we are lazy, or we need short time-to-market (even for a hobbyist), being able to obtain something quick but not dirty, even if not extra-optimized. So, in some more relaxed projects I've used some of the many *xx*-ino boards.

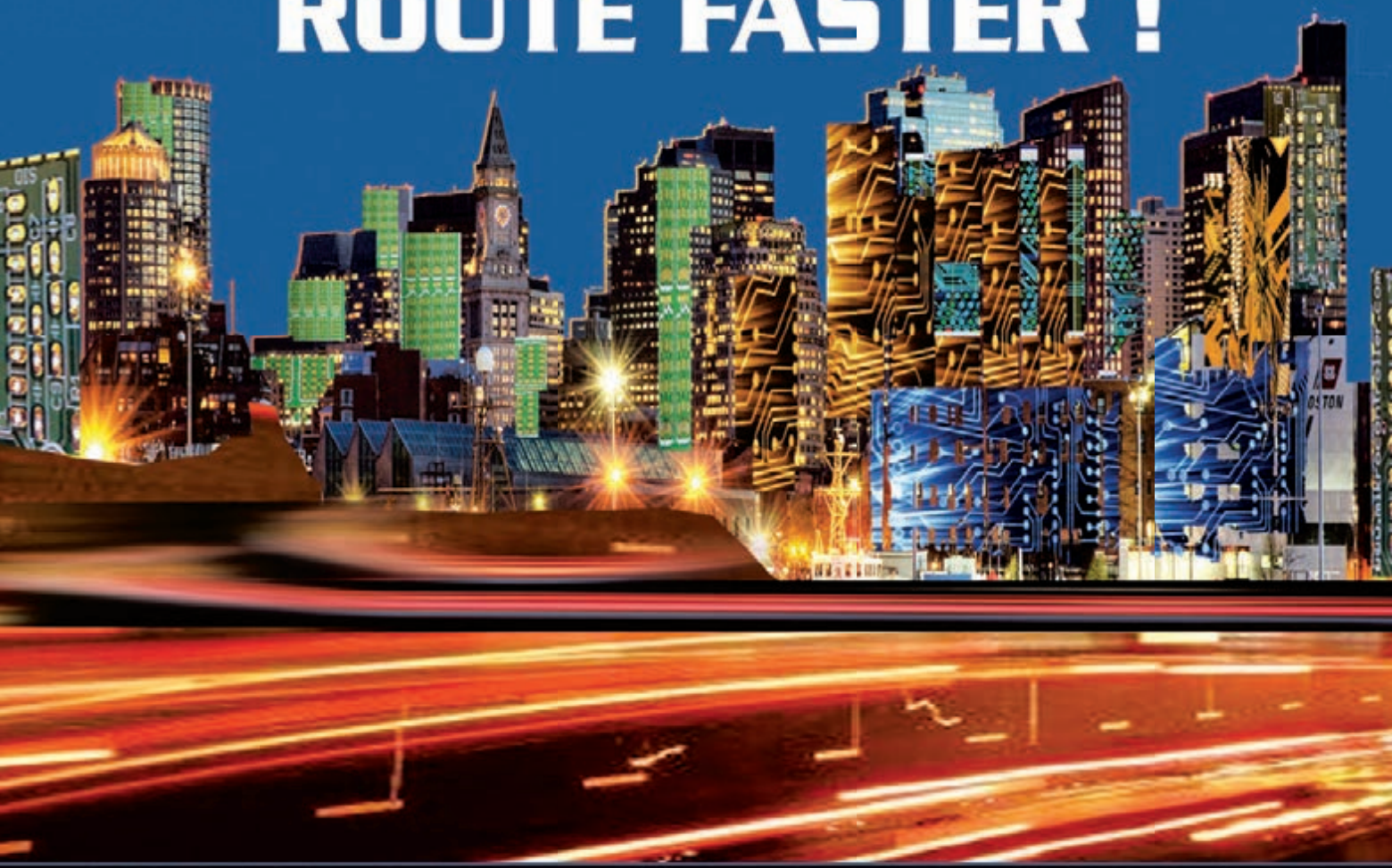
At some point, when I've substituted any of my old friends, SN74*xx* or NE555, with an MCU of whichever type, I wanted to do the same also for my even older analog friends: op-amps. When I need some analog circuit, amplifiers, filters and similar, it is easier to use a PSoC in place of a bunch of transistors, op-amps, capacitors, resistors, and so forth. The modification of a diagram or some line of codes on an IDE is much more friendly than soldering.

I've started studying the amazing world of GPS, in many of its details. I've found some free libraries to do all the not-so-simple calculi for distance, bearing, and so on, but I needed a processor more powerful than a dsPIC to execute, in real time, all that trigonometric math. An mbed gave me the power of an ARM with the easiness of an Arduino, once again with free IDE and compiler.

In conclusion, I always prefer my "first love:" PICs, but there are so many other devices that I can find the one that best fits any project on my mind.

Furthermore, because this is a hobby for me—and I've got a lot of knowledge from many contributors on magazines, websites, and other papers—I like to share back what I've learned. Using different solutions give me the opportunity to do this on the same media: paper, website, or forums. All of my projects are publicly available. Thanks again, Internet!

ROUTE FASTER !



WITH PROTEUS PCB DESIGN

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

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

PROTEUS DESIGN SUITE

- Hardware Accelerated Performance.
- Unique Thru-View™ Board Transparency.
- Over 35k Schematic & PCB library parts.
- Integrated Shape Based Auto-router.
- Flexible Design Rule Management.
- Polygonal and Split Power Plane Support.

Features:

- Board Autoplacement & Gateswap Optimiser.
- Direct CADCAM, ODB++, IDF & PDF Output.
- Integrated 3D Viewer with 3DS and DXF export.
- Mixed Mode SPICE Simulation Engine.
- Co-Simulation of PIC, AVR, 8051 and ARM7.
- Direct Technical Support at no additional cost.

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

labcenter Electronics

www.labcenter.com

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

NAN: Do you have any advice for *Circuit Cellar* readers who are considering building their own autonomous robots?

GUIDO: I love robotics because it involves so many disciplines. Each one of us can find their own specialization, but we must also know something about other fields. I receive a lot of odd questions in the forums coming from people who, knowing just high-level programming, believe they should immediately be able to build an autonomous robot.

Allow me to use the words of one of my gurus, David P. Anderson. Speaking about a robot competition, he said that you must accomplish in advance the very first test: make the robot run forward and backward for some minutes:

"Seasoned robot builders will tell you that getting to this stage is most of the battle. To successfully run this first simple exercise, the robot builder must have some sort of functioning robot platform with wheels and motors and batteries attached, all mechanically and electrically sound; some sort of H-bridge or other means of controlling the motors from a microcontroller; a microcontroller with the necessary I/O all wired up and working; a software development environment set up and working, with the ability to connect to the robot and download code; and a robust enough implementation that it can run for 10 or 15 minutes without crashing, resetting, coming apart, or having motor EMF spikes brown-out the CPU."

You have only a few ways to do that: One, work in a team with distributed specializations; two, buy a kit with most of those jobs already done; or three, study, practice, study, practice, study and… study again. That's all!

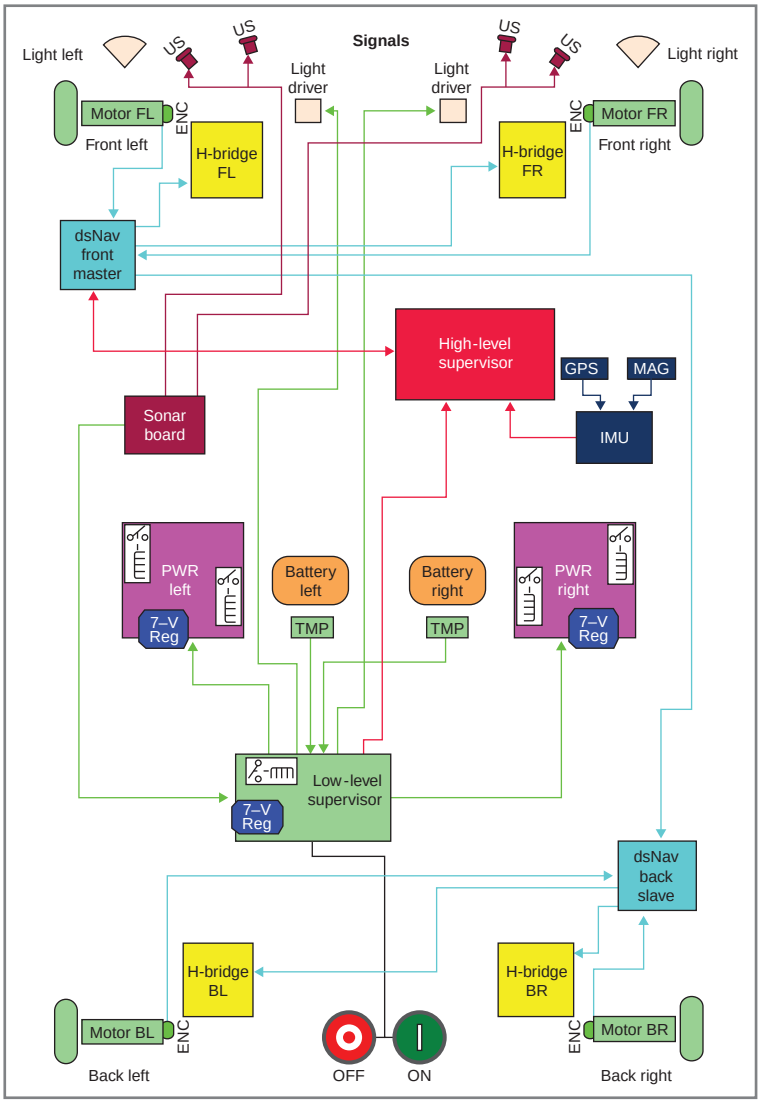
As you can see from my lab picture, I've chosen the third way. I excluded the first way, because this is a hobby performed just in my spare time (that I am not able to schedule), I can only share experiences sometimes and/or remotely with my friends. I excluded the second way because I want to build everything by myself, but this is only a preference of mine. There are a lot of kits that enable you to have a working robot with little or no mechanics effort: LEGO MINDSTORMS and VEX supply building blocks, Parallax and Pololu have already done robotics platforms of many sizes, just to talk about the most famous ones. It must be clear that this one is not the cheapest path and you must know something about electronics anyway.

On my workbench you can see at least three disciplines from left to right: mechanics, electronics, and programming. But they are not enough. You require physics, math, and more. The third way is not the easiest one, but it's the one with the most satisfaction.

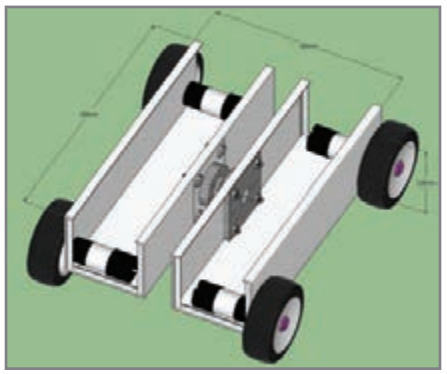
NAN: Are you currently working on or planning any robotics-related projects? Can you tell us about them?



A glimpse at Gudio's workbench reveals at least three disciplines, from left to right: mechanics, electronics, and programming



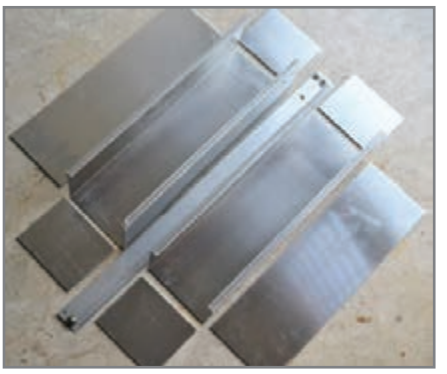
A block diagram of the boards that will be inside Guido's outdoor robot



The design phase of Guido's Lino robot: a 3-D model created with Google Sketchup



A mockup of the Lino made with corrugated paper to test the validity of the project



An aluminum sheet that has been cut and folded for the final version of Lino

GUIDO: Yes, I am.

Once I wrote in a forum, "I'll never build a robot that cannot fit in a shoebox." This was because my lab is in an apartment, and I have limited mechanical capabilities. Luckily, the human mind can change an idea during its lifetime. After my indoor studies were over, I started thinking about an outdoor robot. This must be simple, cheap enough, and realizable by anyone who is patient enough to follow my explanations. To make a full suspended four-wheel drive robot without using expensive RC cars parts that are not easy to assemble, I was inspired by the award-winning Flexi Flyer, articulated robot.

More pictures are available at my Lino website (www.guiott.com/Lino/Sito/Lino.html).

Right now, I'm in a very early stage. I want to reuse the already acquired knowledge for the "low-level" subsystem (batteries, sensors, and motor control), because the problems are very similar to the indoor experience. After everything works on this part (Anderson docet) I'll start experimenting with something new for me: gyros, accelerometers, magnetometers, and GPS. In short, inertial measurement unit (IMU).

I don't know when it will be ready, but it will be ready. Stay tuned.

NAN: What do you consider to be the "next big thing" in the embedded design industry? Is there a particular technology that you've used or seen that will change the way engineers design in the coming months and years?

GUIDO: As often happens, the "big thing" is one of the smallest ones. Many manufacturers are working on micro-nano-pico watt devices. I've done a little but not very extreme experimenting with my Pendulum project. Using the sleeping features of a simple PIC10F22P with some care, I've maintained the pendulum's oscillation bob for a year with a couple of AAA batteries and it is still oscillating behind me right now.

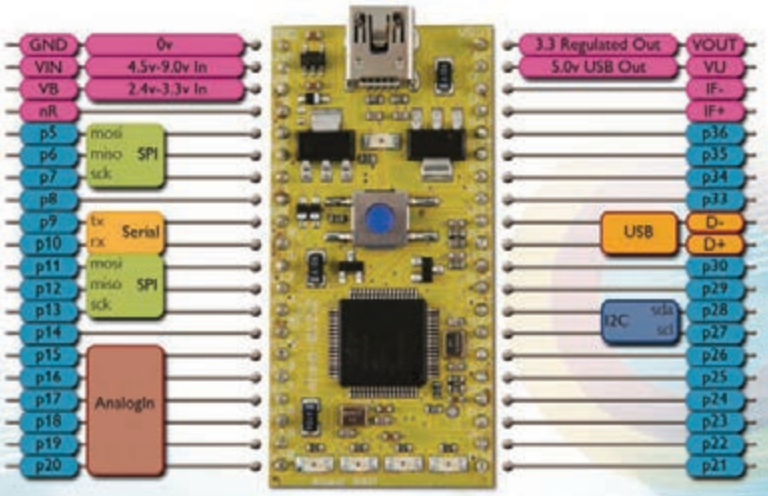
Because of this kind of MCU, we can start seriously thinking about energy harvesting. We can get energy from light, heat, any kind of RF, movement or whatever, to have a self-powered, autonomous device.

Thanks to smartphones, PDAs, tablets, and other portable devices, the MEMS sensors have become smaller and less expensive.

In my opinion, all this technology—together with supercapacitors, solid-state batteries or similar—will spread many small devices everywhere to monitor everything.

mbed

mbed NXP LPC11U24 Microcontroller



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

http://mbed.org

by Scott Weber (USA)

MCU-Based Light Control

Longer Serial Communication on Differential Wires

You don't have to purchase expensive off-the-shelf home control equipment. For instance, you can use a microcontroller and a few other components to build a custom system to control lights on a predetermined schedule. This design is intended to replace a wall switch.

The phrase, "I could do *that* better!" seems like the driving force behind most projects undertaken by engineers. In my case, the project was replacing a wall switch with a simple little timer that could be scheduled to turn exterior lights on or off. The units gave me no end of frustration since the day I installed two of them. Without naming the manufacturing culprit, I disassembled the unit to determine the problem. It turns out the unit was a collection of small plastic gears driven by a small motor. These gears turn a cam that presses a lever switch that opens or closes the AC line. In a perfect world, that might be okay, but variations in temperature and humidity caused changes in friction that couldn't be overcome by the little underpowered motor. Actually, there were two lever switches. The second one provided feedback to the microcontroller telling it when to stop. As the gears failed to turn, the unit displayed an error condition and caused an associated rise in my blood pressure. Of course, all this functioned even less reliably in a more hostile garage environment instead of in the comfort of the house. And I would not like to see the plastic gears' reliability after a few years of wear on the plastic shafts.

I decided that it was time to build a better mousetrap by designing a new microcontroller-based controller to replace the timer (see Photo 1). Had this been a simple little timer, there wouldn't be much of a story. However, I decided this initial event would enable me to start putting networked devices all over the house. This is the journey of creating some novel devices, choosing a way to interconnect them, and writing some programs in a variety of languages. Although the initial goal was to come up with a reliable controller, this became the opportunity to create other devices to accomplish tasks as needed.



Photo 1—Take a look at the final assembly. The board is mounted on the side, and two wires are pigtailed from the relay to wire nuts that control the AC power to the light. The four-conductor harness in red is the RS-485 line and 5-V power. This installation certainly won't endear me to any building code inspectors, but it works reliably.

Along the journey, I wrote a small PC Windows program that enabled me to send messages to the devices and monitor their replies. This was a big help with debugging devices and the code within them. The Windows program still has some useful functionality, but I am slowly moving away from it and adding those functions to other devices. All the code, including the PC program, is available on *Circuit Cellar*'s FTP site.

TALKING ON TWO WIRES

The first question was how to communicate with a collection of little microcontrollers located around the house. Wireless or wired? Serial or parallel? I chose wired, as I didn't want to deal with using batteries or wall warts to power the remote units. Also, Wi-Fi's complexity was a learning curve I didn't feel like undertaking at this stage. And, if the method was going to be wired, I could also supply power as needed. Fortunately,

my house has a little room I use as a wiring closet. I have an unused four-conductor run from the wiring closet to various locations, including close to the point where I needed to replace one of the light timers.

Wired communication across distances is normally done through some kind of serial connection, so the next decision was what to use. Some options for two-wire serial communications were I²C, TTL serial, or RS-232. These all suffer from distance limitation due to the voltages that can drop across long runs of wire. As an example, RS-232 is limited to 50', despite its full-swing specification of ±12 V. Clearly, running a distance of 50 or more feet will require a nice differential solution.

Such a differential solution is available in RS-422 and its sister specification, RS-485. Both of these use a simple two-wire differential signal that is more immune to noise and voltage drops incurred by long distances. In fact, the design enables distances of up to 4,000', which is almost a mile. The main difference between RS-422 and RS-485 is RS-422 supports full-duplex communication over four wires while RS-485 supports half-duplex communication over two wires. When using just two wires, RS-422 is essentially a broadcast topology whereby a single node can send information to up to 10 receiving nodes. The receivers cannot talk back to the broadcaster because the broadcaster is always asserting its signal on the transmission line. Implementing a true bidirectional communication requires implementing return paths, which would mean adding two new wires for each path. This quickly complicates the communication wiring.

An enhancement to the one-way topology of RS-422 is the specification of RS-485. RS-485 uses the same differential signal RS-422 uses, however it enables multiple devices to connect to the same pair of wires with no one unit asserting a signal on the line until it has something to say. In order to do this, each node uses a tristate line driver, and they all keep their output in high-impedance mode until they're ready to talk.

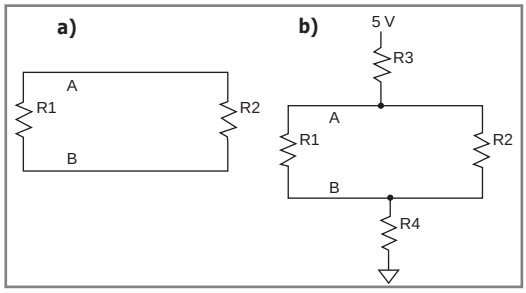


Figure 1a—Here is the basic RS-485 line with no devices attached, just the termination resistors. **b—**This shows the lines with the bias resistors so the bus has a constant "mark" on it when it's not in use.

WHICH WAY IS UP?

The two wires that make up the bus used to carry the differential signal are called—for lack of anything more interesting—A and B. While idle, the microcontroller keeps the serial output level at a mark condition, which is a logic-high state, and transitions to a space condition, or a logic low, to indicate the start bit. The datasheet for National Semiconductor's DS1487 RS-485 transceiver indicates that when the input logic is high, the output A becomes high and B becomes low. Of course, the opposite applies when the input logic is low. The RS-485 requirement is that the differential between A and B is greater than 200 mV. The condition is a valid space or mark, depending on the polarity. Keep in mind that with any AC signal running down a pair of conductors, the ends need to be properly terminated. In this case, about 100 Ω is adequate. Also, both ends need to be terminated because the bus doesn't really have a source and destination point like normal transmission lines. Figure 1a is a simple representation of the RS-485 line.

So far, so good. However, because all the nodes are essentially disconnected from the bus when they are not talking, the bus becomes a pair of floating wires, unconnected to any signal. This makes the bus one really big antenna and wildly susceptible to noise. The solution is to make sure there is a mark or Logic 1 asserted on the bus when no one is talking. In other words, keep the A line slightly more positive than the B line. In order to accomplish this, use Ohm's law. Keeping a desired differential of 200 mV across two 100-Ω resistors in parallel means about 4 mA flowing across the resistors (i.e., 200 mV/50 Ω) is needed. To bias these wires so they have a nominal difference of 200 mV when the system is powered with 5 V, place bias resistors R3 and R4 on the line (see Figure 1b). Next, determine that, if 4 mA flowing from top to bottom is required, ∑ R = 5 V/4 mA, so the total resistance in circuit B needs to be about 1,250 Ω. Since we know the terminators combine to form 50 Ω, the two remaining resistors need to add up to 1,200 Ω, or about 600 Ω each.

OFF THE PAPER, INTO THE ATTIC

I bring my telephone, Internet, alarm contacts, and anything else into the wiring closet, so this RS-485 network should not be an exception. Because the network needs to run continuously with terminations on each end, I started simply with two runs. Each run has its termination at its far ends and is connected to a barrier strip in the wiring closet. This also provides a point where I can insert the bias voltage. I also added some transient voltage suppressors between the power lines and ground that protect the power bus from spikes. The barrier strip that connects to the power supply is shown in the lower portion of Photo 2.

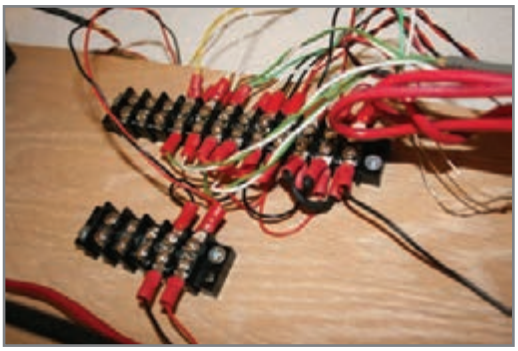


Photo 2—The barrier strip (top) that terminates the RS-485 lines in the wiring closet. All the power lines (black and red) connect to the right end. More difficult to see are the red shrink-wrapped bias resistors to the left of the MOVs. The bottom barrier strip is the 5-V power, which comes from a surplus Power One module.

One side of the bus starts at a location in the garage and then visits the light switch location that handles the accent lights outside my garage doors. From there, it visits a wall box near my desktop PC so I can attach my PC to run any testing. Then, it terminates on the barrier strip shown in Photo 2. From there, it leaves again and makes a straight run to the wall switch by my front door, which controls the exterior porch light. Devices should be attached to the bus as close

Addr Range	Usage
x20	Broadcast
x21-x25	Control panels
x26-x2f	Lights
x30-x37	Reserved
x38-x3f	Doors
x50	Time server

Table 1—This is how reserved blocks of addresses for various types of devices are assigned

Command	Payload	Purpose
S	none	Used to request the device to report its status. The status returned will have different meanings depending on the device being interrogated.
Z	none	Request a report of the reasons for MCU reset stored in the EEPROM. This is helpful to determine if a device is undergoing excessive watchdog resets or brownouts.
Y	none	Request the device clear all EEPROM counts of reset reasons, defined above.
T	hours, minutes, seconds	Request the device to set its time to this value, in 24-hour format. Devices that do not have internal clocks can ignore this command.

Table 2—The commands used to define what communication can be used by all devices

as possible, avoiding drops that have long feeders or a "star" layout. If I need to add any new devices, I insert them into the bus with a line that runs out and back again. This completes the setup needed for the RS-485 connections. Now to create a protocol.

COMMAND & CONTROL

To determine how these devices would talk to each other, I had to define the limits of my needs and then create an accommodating protocol. So I decided on these parameters and limitations with the goal of keeping things as simple as possible.

I wanted the message structure to be short, simple, and a variable length, so I used the ASCII value SOH (0x01) as a start marker and EOT (0x04) as an end marker. Knowing the beginning and end, I can stick a variable amount of data in between those two values, provided I don't use the values SOH and EOT again. Otherwise, anyone listening to the data will become confused trying to find a message's start or end. I needed an alternate way to represent integer values when sending payload data across the line.

Following the SOH byte is the device address I wanted to talk to and the device address of the message it came from. This is so that any time one unit wanted to talk to another, the target unit could easily build a return message by just swapping the two bytes. Each of these addresses only needs to be 1 byte. Removing the two marker bytes as address possibilities leaves 253 other addresses, which is more than enough for a home design. Taking a page from the class addresses originally used by the Internet, I decided to reserve blocks of addresses for various types of devices. Table 1 shows how these were assigned.

Next came the actual message payload. The messages being sent to assorted devices on the network will vary depending on the device. Each message begins with a command consisting of a single uppercase ASCII letter. If values are needed in its payload, they are encoded in ways that will prevent them from conflicting with the start and end marker bytes. Before defining communication to specialized devices, there is a collection of messages to define that can be used by all devices (see Table 2). For example, if I wanted to tell a unit that it needs to set the time to 9:55:45 A.M. on a device, I would send the command "T095545." More specialized commands will be explained in detail in the section describing those devices.

Other types of data include simple integers and Booleans (true/false) flags. Small-value integers are values less than 200. I chose this threshold because I can just add a value of 0x20 to the integers without worrying about converting an ASCII string back to a numeric value. What's more, they still won't overlap with the start and stop bytes. Wherever a Boolean value is needed, those values are sent as ASCII "0" or "1" (i.e., things that need to report as on-off or open-close values).

Finally, in cases where commands are sent to a device and a response is needed, the ASCII command letter is translated to lowercase in the reply message, indicating that this is a reply, not an instruction.

LET THERE BE LIGHT

I wanted my replacement light timer design to be solid state. I also didn't want it to consume excess power when the unit was just idling, so using a good latching relay seemed in order. I also wanted it to be small so it could be inserted into a single-gang wall box without taking up too much space. This also meant, when ordering circuit boards, I could place multiple copies on

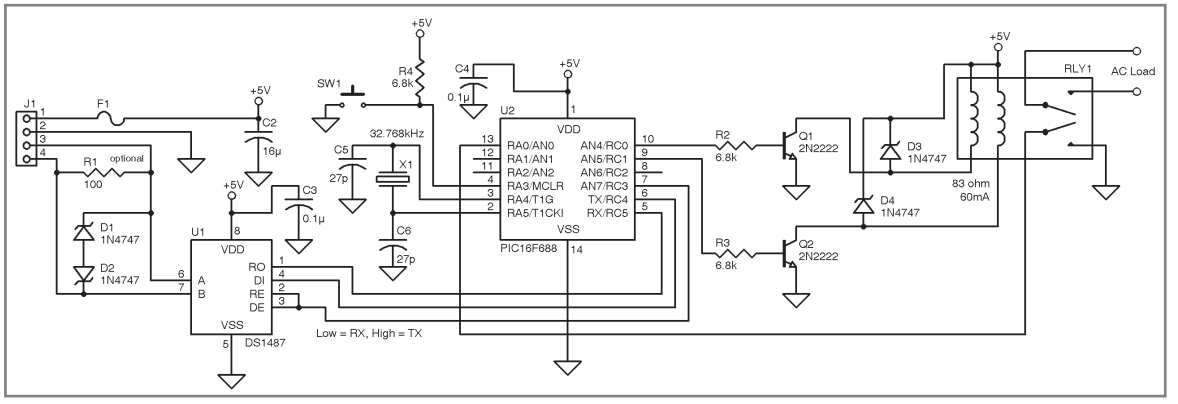


Figure 2—The light control board. The differential signal is translated to 5-V levels by U1 and sent to the PIC MCU U2. The termination resistor R1 is optional and only used when the unit is at the end of a run.



Windows CE 6.0 Touch Controller

CUWIN

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

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



C-Programmable Modular Industrial Controller

MOACON

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

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

BASIC with LADDER LOGIC CONTROLLER



New

Integrated CUBLOC Controller and I/O Board

CB210

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

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

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

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

CUPC

Industrial Touch Panel PC with ATOM Processor



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

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

CUSB

Integrated Industrial Controllers



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

a single PCB and cut them apart, giving me more units per dollar.

HARDWARE

Figure 2 is a schematic of my light controller. When I ordered three circuit boards from ExpressPCB, I actually combined the layout with another device so I could simply cut them in half, getting six boards for the price of three. The schematic and double-sided boards were created using the ExpressPCB software suite. Photo 3 shows views of both sides of an extra board, unpopulated. This was one of my first attempts at using surface-mount device (SMD) components. Using a fine-point soldering iron, small tweezers, and a magnifier, I'm thankful I'm not a coffee drinker.

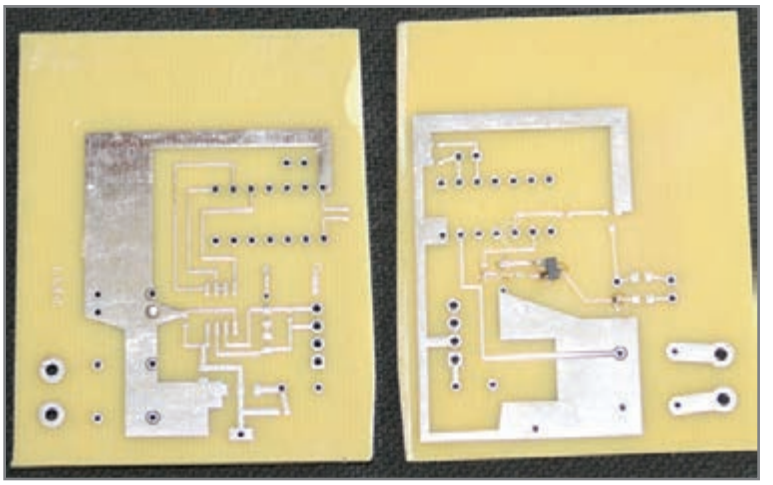


Photo 3—These are the two sides of the circuit board. There is more blank space than there is actual circuit. The large traces on the far right are where the AC lines attach to connect to the relay.

I used a Microchip Technology PIC16F688 microcontroller, which receives both power and data from the four-wire RS-485 bus connected to J1. The power is fuse-protected and bypassed to protect the bus. The differential data lines (A and B) are protected with back-to-back Zener diodes to protect from over-voltage damaging the differential transceiver. A termination resistor is also included. However, only one of the units represents the end of the network line. That unit used the termination resistor, while it was left off the other units. The differential signal is fed to the differential transceiver U1, a DS1487 RS-485 transceiver, which is a tristate driver enabling a device to insert or remove itself from the bus. The driver is switched between Transmit and Receive mode by I/O pin RC3. The receiver translates the differential signal to TTL-level serial signals and the TX and RX lines are connected to the microcontroller UART on Pins 6 and 5, respectively. By connecting the receive enable (RE) and output enable (~OE) together, the driver disables the receiver when it is transmitting, preventing the half-duplex echo where the information transmitted is echoed back to the microcontroller on the receive line.

I used a DPST-latching relay so it doesn't consume any power when it is idle. One pole is used to feed information back to the microcontroller so it can confirm that the relay is on or off. The other contact is used to switch the AC line for the exterior light. The NPN transistors Q1 and Q2 are common emitter configurations that provide the necessary current to energize the relay coils. The relay coils have Zener diodes across them to keep the spikes out of the power supply lines. I used the Panasonic Electric Works DSP1-L2 5-V relay with 5-A contacts, which is more than adequate to switch a 60-W bulb, or the more trendy equivalent 15-W CF.

The microcontroller needs some way of maintaining a real-time clock (RTC) so it can function as a timer. I used a 32,768-Hz quartz watch crystal attached to Timer 1 inputs.

They say it's easier to ask forgiveness than to seek permission, but clearly, this was not meant to apply to circuit boards. Originally, I left off a connection for a manual switch, went back to the board, and added a two-pin header for a small momentary push button. That meant drilling holes in a populated board and using a jumper wire to connect the header to the microcontroller. The debouncing is accomplished in software.

SOFTWARE

The microcontroller's code was written in PIC Assembler using Microchip's MPLAB integrated development environment (IDE). The project contains six source code modules to encapsulate the functionality, making it easier to maintain. However, like all Assembler programs should be, many instructions are commented. This helps remind me what it is doing, because the code was written more than three years ago. By keeping the program as lean as possible, it only uses 739 instructions, or 16% of the microcontroller memory. Now I'll describe the function of each source module.

Upon reset, the program starts in the source files main.asm. Here it sets the state of the microcontroller, enables the watchdog timer (WDT), and records the reason for the reset into EEPROM. It then sets the I/O ports on A and C to enable Timer 1 to run and to enable control of the latching relay. Next, the UART and Timer 1 are enabled and set to cause an interrupt when something happens. Finally, the program enters an infinite loop waiting to see if a message arrives from the network, if the push button was pressed, or if the time now matches an event that needs to turn the relay on or off. The entry point to the interrupt service routine (ISR) is located in a separate module called interrupt.asm. All it does is determine the cause of the interrupt and dispatch the execution to either timer.asm or comm.asm, depending on the type of interrupt.

The comm.asm module handles all incoming characters from the RS-485 line. It checks to ensure the arriving bytes can be placed into the receiver buffer, which wouldn't be permitted if the main loop is still processing the last message received. When it

Command	Payload	Purpose
R	1-byte event	Return the contents of the event # in EEPROM
E	4 bytes	Set event # with HH:MM and on or off
L	1-byte light	Tell the unit to turn the light on or off immediately "1" or "0"

Table 3—Here are additional communication messages understood by the light controller. These are used to read/write events to the EEPROM and enable it to set a number of actions throughout the day (or night).

Listing 1—This is the Assembler code that examines the incoming messages from other devices. The "msg_buffer" is defined as the area where the ISR has already placed the incoming data from the UART.

```
;_____________________-
;       Test to see if we got a message, and
;       determine what to do based on the message
rx_message
        btfss machine_stat,RX_MSG_BIT     ; A new message?
        return                                          ; no, bail

                                ;  Test it for universal address
        movlw 20                ; Test the message for 20
        subwf msg_buffer,W
        btfsc STATUS,Z
        goto process_command

        movfw dev_addr          ; Test the message for our addr
        subwf msg_buffer,W
        btfsc STATUS,Z
        goto process_command
done            ; default is to reset the RX flag
        banksel machine_stat
        bcf machine_stat,RX_MSG_BIT       ; clear the RX flag
        return
;_____________________
;  At this point, we have a message with the SOM
;  Byte 1 is dst addr, byte 2 is src addr byte 3 is cmd
process_command
        movlw 'L'                         ; Test the message for x'L'
        subwf msg_buffer + 2,W
        btfsc STATUS,Z
        goto fliplight
                                ;  Test the message for our address
        movlw 'T'                         ; Test the message for 'T'
        subwf msg_buffer + 2,W
        btfsc STATUS,Z
        goto settime

        movlw 'S'                         ; Test the message for 'T'
        subwf msg_buffer + 2,W
        btfsc STATUS,Z
        goto sendstat

        movlw 'R'                         ; Test the message for 'R'
        subwf msg_buffer + 2,W
        btfsc STATUS,Z
        goto readevent

        movlw 'E'                         ; Test the message for 'E'
        subwf msg_buffer + 2,W
        btfsc STATUS,Z
        goto saveevent

        movlw 'Z'                         ; Test the message for 'Z'
        subwf msg_buffer + 2,W
        btfsc STATUS,Z
        goto sendresets

        movlw 'Y'                         ; Test the message for 'Y'
        subwf msg_buffer + 2,W
        btfsc STATUS,Z
        goto clearResets

        goto done
```

sees an SOH byte, it resets the counters and begins collecting subsequent bytes into the receive buffer, counting the arriving bytes as they appear. If the count exceeds the size of the receive buffer, it discards the remaining bytes. When it sees an EOM byte appear, it sets a flag to indicate a new message has arrived. This flag is detected and acted upon by the main program loop.

The other interrupt is handled by timer.asm, which simply increments the seconds—and the minute and hour, if needed—every time the timer expires. Because the counter is 16 bits and the oscillator is 32,768 Hz, the timer would normally fire every 2 s. By adding 32,768 Hz to the timer, the timer interrupts are accelerated to every second. This is accomplished by simply setting the high bit of the Timer 1 counter.

Meanwhile, back in the main loop, the code tests to see if the time has changed or if a message has arrived. If a message does arrive, it is processed by the module commands.asm. In here, it compares the address of the message to its own address or to the universal address, and discards the message if they don't match up. It then compares the command byte to the commands it understands. The basic commands were previously discussed. Table 3 shows the specialized commands the unit understands. Listing 1 shows how the commands are processed.

The program stores events in EEPROM, which is simply an entry of the time something needs to occur (i.e., whether to turn the relay on or off). The microcontroller checks the events in the EEPROM against the current time every minute. If there is a match, it determines if the event is needed. It does this by examining the condition of the relay and will change it if needed. If the relay is already in the position requested, it won't take any action. The code that handles all the reads and writes to EEPROM is in the module save_ee.asm.

The main loop also performs the poll for the push button and does some debouncing, which I performed backward. For decades, electronics people have been taught that, to debounce a contact, you to watch for an action then wait a while and see if it's still valid. Essentially, it's a low-pass filter. However, I took a front-loaded approach, which makes the button slightly more responsive. In the byte that monitors the state of the module, a flag reflects the state of the button bit (high or low). The software responds to any change and then starts a debounce timer. The timer simply prevents the module from reacting to any change for a determined amount of time. Furthermore, since I am

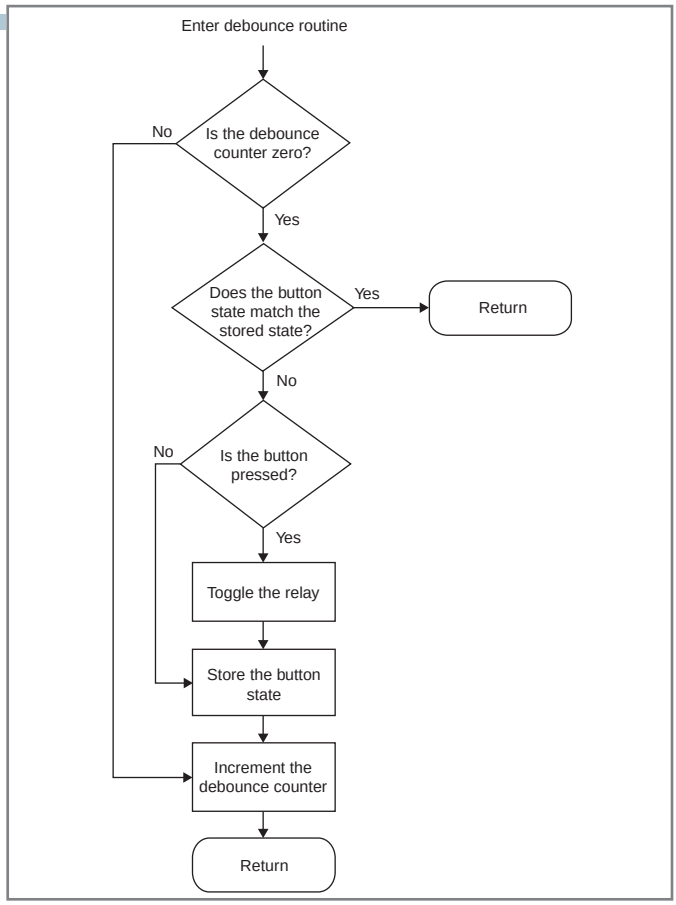


Figure 3—The debounce flowchart. First, check to see if the debounce timer is already running. If it isn't, check to see if there is a mismatch between the stored state of the push button and the button itself. Next, if the button was pressed, take action. Last, start the debounce timer, so this won't happen again until the timer expires.

only interested in button pushes, the code ignores any change where the button is released. The process is shown in Figure 3. The debounce timer is simply an 8-bit counter that wraps around back to zero after 255 counts. The debounce time could be shortened by seeding it with a predefined value instead of starting from one. In order to lengthen it, it must be expanded to a counter greater than 8 bits.

One common feature in all the devices I built for this project was that, during program start up, the reason for the reset was examined and a value for that reason was incremented in EEPROM. Those locations are EEPROM bytes 1–4 for power up, MCLR reset, brownout, and watchdog reset, respectively. For diagnostic purposes, the device will send these counts to another device by request, enabling examination of the unit to see the different reasons for restarting. I am proud to say there haven't been any watchdog-based resets, which tells me the program is relatively solid.

NOW WHAT?

I now have the control module installed and connected to the RS-485 network (see Photo 1). When covered with a signal-gang blank panel, the unit is completely unnoticeable.

The only thing I've described so far is this unit and my ability to send it instructions from my PC. By reviewing the described protocol, you can see that I plan to take advantage of this for other things. The first thing it needs to be told is the correct time. I could have built in a small user interface to set the time, but instead I decided to add a central time server that will broadcast the time across the network, so everyone can take advantage of it. In a future article, I will describe how I built a time server using data from a surplus GPS module.

POSSIBILITIES

Although I designed this to replace an inferior timer that operates a light on an AC line, the dry-relay contact the unit contains could be used for anything that needs to be turned on and off throughout the day. The nonvolatile memory that stores the on or off events provides me with a nice feature so I don't have to reprogram the on and off times when the unit comes up from a power outage.

Also, the flexibility of the RS-485 network—and the protocol I designed to enable devices to communicate—opens the door for the construction of other units. Something that comes to mind is a daylight sensor to tell the lights to activate when it gets dark, regardless of the actual time. Or maybe a Wiznet module that enables my PC to communicate on the network from any networked machine, or even my iPhone. Those are nice; however, I have already built other components I will describe in subsequent articles.

It is instructive to note that using multiple source code modules and grouping functionality into those modules, a project becomes much easier to maintain and enhance. It is also worth noting that developing and debugging this project convinced me more than ever to use a C compiler for any further development. So, for my next device, I'll describe how I did the programming in C and the issues I experienced with the compiler I used.

Scott Weber (scotty42@csweber.com) is a PC developer who is currently working slowly toward finishing a BS in Electrical Engineering at the University of Texas at Arlington. His primary goal is to get the sheepskin that justifies the knowledge he already has from 30 years of electronic tinkering. He currently works as a PC developer and is rapidly tiring of the direction the PC software world is taking. He lives with his wife and her beautiful garden in Texas.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/265.

SOURCES

ExpressPCB software suite
ExpressPCB | www.expresspcb.com

PIC16F688 Microcontroller and MPLAB IDE
Microchip Technology, Inc. | www.microchip.com

DS1487 Transceiver
National Semiconductor Corp. | www.national.com

Panasonic DSP1-L2 5-V relay
Panasonic Electric Works Corp. | http://pewa.panasonic.com

Register Now!

ESC

a designeast summit

ESC Boston returns this fall, bigger and better than ever with four days of non-stop technical training, education and product evaluation across 5 summits, all in one location.

September 17 - 20, 2012
Hynes Convention Center • Boston, MA
east.ubmdesign.com

DESIGN East is comprised of:

esc | android | design med

LEDs | sensors

DESIGN East is your one-stop shop and the most efficient way to update your skills & knowledge and discover the newest technologies & products — and it's your best opportunity to learn better & faster design techniques so you can create new solutions for your biggest challenges on the job

Sharpen the skills you need now:

- 5 Summits with 120+ educational & training sessions conference tracks
- Android™ Certificate Program
- Visionary Keynotes
- 150+ Exhibitors showcasing the newest products & technologies
- 30+ Sponsored Training Sessions & DESIGN East Theater
- Happy Hours, Product Teardowns, Giveaways and more!

Learn more at east.ubmdesign.com

by Shlomo Engelberg (Israel)

Examining an I/O Port

The Internal Structure of an I/O Pin with a Pull-Up Resistor

Understanding the internal workings of your microcontroller is essential. This article details how a simple experiment enables you to clearly "see" the structure of pins with internal pull-up resistors. The experiment also enables you to see the effect of each of the PMOS transistors that comprise the pull-up resistor.

Microcontrollers in general—and Intel's 8051 microcontroller and its descendants in particular—have input/output (I/O) pins. 8051-based microcontrollers have two types of I/O pins: pins with and pins without internal pull-up resistors. The pins are controlled by specific bits in the microcontroller's special function registers (SFRs). The internal structure of the pins with internal pull-up resistors is interesting. In this article, I'll explain how a simple experiment enables us to "see" that structure quite clearly.

Figure 1 depicts simplified descriptions of I/O pins with and without pull-up resistors, respectively. Pins with internal pull-up resistors can be used to output 0 V or V_{DD}. To output 0 V, close the internal switch. To output V_{DD}, open the switch and make certain not to load the output. To use a pin with an internal pull-up resistor as a digital input, its internal switch must be opened. To input V_{DD}, do not apply any voltage to the pin; let the pull-up resistor take care of that. To input 0 V, force the voltage on the pin to 0 V. (The microcontroller has circuitry that enables it to examine the voltage at a pin's output.) A specific bit is set in order to open the internal switch. The same bit is cleared to close the switch.

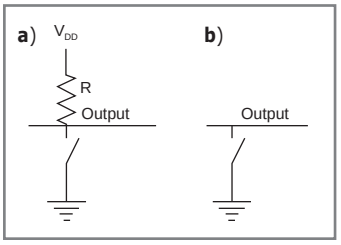


Figure 1a—This is a simplified model of an I/O pin with a pull-up resistor. **b—**This is a simplified model of an I/O pin without a pull-up resistor. In both cases, when a one is written to the bit associated with the I/O pin, the switch is opened. In Figure 1a, when the switch is open, as long as no voltage is being applied to the output pin, the voltage will be V_{DD}. When a zero is written to either pin, its output voltage becomes 0 V, and no voltage may be applied to the pin.

A good designer faces a dilemma when designing a circuit similar to Figure 1a. How big should the pull-up resistor be? On one hand, it should be as small as possible so that when the output transitions from 0 V to V_{DD}, the pin can quickly supply the current necessary to charge any parasitic capacitors. On the other hand, when a zero is input by forcing the voltage at the "output" to 0 V, any current that flows through the pull-up resistor is being wasted—so the resistor should be as large as possible. To address these and other issues, the microcontroller's designers chose to implement the pull-up resistor by using several transistors.[1] The pull-up resistor is not a resistor at all.

AN I/O PIN'S INTERNAL STRUCTURE

Figure 2 shows part of the internal structure of an I/O pin. There is one NMOS transistor, Q1, which acts as the switch

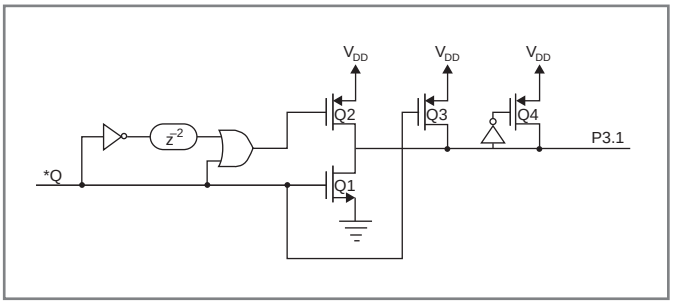


Figure 2—The internal structure of an I/O pin with a "pull-up" resistor. The pull-up resistor is not actually a single resistor. It is composed of the three PMOS transistors. (Here z^{-2} indicates a delay of two clock cycles.)

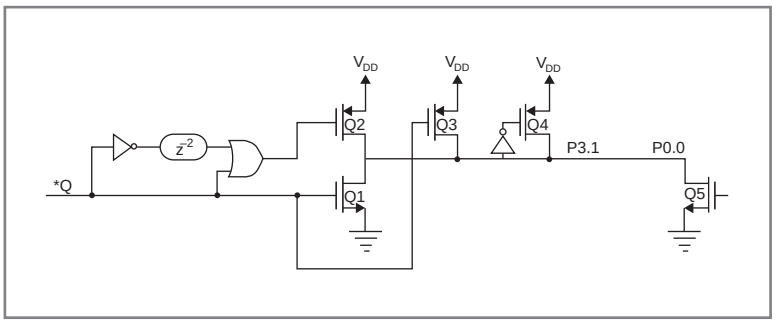


Figure 3—A pin with an internal pull-up resistor is connected to one without a pull-up resistor. The pin without a pull-up resistor is represented by Q5. The rest of the assembly is the pin with a pull-up resistor.

from Figure 1a, and three PMOS transistors: Q2, Q3, and Q4. Together, they constitute the "pull-up resistor." Q2 is a large transistor capable of handling a lot of current. Q3 is a small transistor. Q4 is a medium-size transistor. When the microcontroller is to output 0 V, it sets Q in Figure 1 to "logical zero" and the voltage associated with *Q is V_{DD}. This causes Q1 to conduct and all the other transistors to be cut off. The output voltage will be 0 V.

When Q goes from a logical zero to a logical one, the voltage associated with *Q goes from V_{DD} to 0 V. A close look at Figure 2 shows that Q1 will clearly be cutoff and Q3 will conduct. Additionally, for two clock cycles the very large transistor Q2 will conduct as well. Once the output voltage rises sufficiently, Q4 conducts too.

Q2 sources a large amount of current for a short time. This current is used to charge small capacitive loads quickly. Q2 makes it possible for the 0 V to V_{DD} transition to happen quickly. You now understand how the I/O pin can be used as an output. How can it be used as an input?

I/O pins have to be able to accept inputs of 0 V or V_{DD}. When used for input, Q is set to a logical one and Q1 is cut off. As the PMOS transistors make the voltage on the pin V_{DD}, you do not input V_{DD} to the pin; you let the pin float and V_{DD} is input "automatically." To input a zero, you ground the pin. Because Q2 is normally cut off, no current flows through Q2, the large transistor—and this saves us a lot of current. Once the output is pulled low, Q4 is cut off as well. Only Q3, the smallest transistor, is still conducting. In this way, very little current is wasted by the circuit when a zero is being entered.

What purpose does Q3 serve? Suppose that after entering a zero, you want to enter a one. After grounding the pin to enter a zero, you "unground" the pin, and Q3 acts to pull the voltage on the pin up and to charge any capacitances at the output. After a little while, the voltage on the output rises enough that the inverter at Q4's gate sees a logical one, and Q4 starts to conduct too. At this point, the output starts rising faster.

You can see that Q2 is there to make the zero to one transition faster when a pin is being used as an output. Q3 is there for a reason. If nothing is connected to a pin, when the pin's output should be a logical one, it is. Q4 is there to maintain a relatively low-resistance path to V_{DD} when the pin's output should be a logical one and nothing is pulling the pin low. There is no connection between the pin and 0 V.

I/O pins without internal pull-up resistors are essentially just NMOS transistors whose sources are tied to ground. In such pins, when the gate of the transistor is driven high, the transistor provides a path to ground, and when the gate is driven low, the transistor disconnects the pin from ground but does not connect it to anything else. In the 8051, the pins of port 3 have internal pull-up resistors, and the pins of port 0 are of the minimalist variety—they are open-drain pins.

I will now describe a simple way to *see* the effect of each of Q2, Q3, and Q4.

"I/O pins without internal pull-up resistors are essentially just NMOS transistors whose sources are tied to ground. In such pins, when the gate of the transistor is driven high, the transistor provides a path to ground, and when the gate is driven low, the transistor disconnects the pin from ground but does not connect it to anything else. In the 8051, the pins of port 3 have internal pull-up resistors, and the pins of port 0 are of the minimalist variety—they are open-drain pins."

THE EXPERIMENT

In order to see the effects of the three transistors, I tied an I/O pin with an internal pull-up resistor, pin 1 of port 3 (henceforth P3.1) to an I/O pin without an internal pull-up resistor, pin 0 of port 0, henceforth P0.0. Figure 3 is a transistor-and-gate-level schematic of this configuration. Q1–Q4 and their associated gates and connections are pin 1 of port 3. Q5 and its input describe P0.0. I had my 8051-family microcontroller (in this case, the Analog Devices ADuC841) check to see whether 0 V or V_{DD} was being input to a third I/O pin—to pin 2 of port 3, P3.2.

If V_{DD} was being input to P3.2, I had the microcontroller "open the switch" of P0.0 and "toggle the switch" of P3.1. If 0 V was being input, I had the microcontroller open the switch of P3.1 and toggle that of P0.0. Looking at the simplified models in Figures 1a and 1b, you would think that

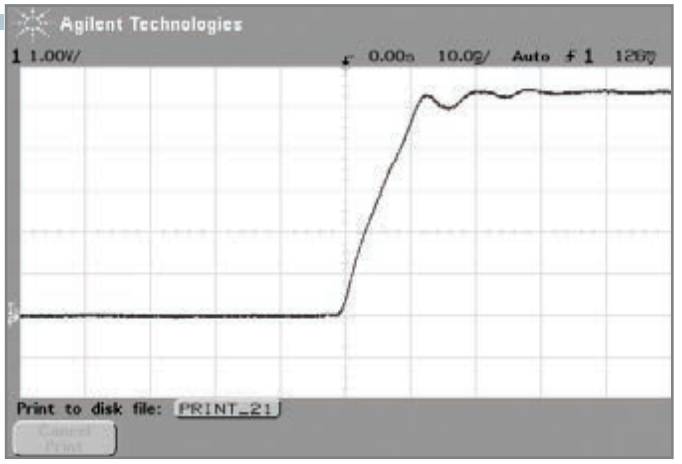


Photo 1—The 0-to-5-V transition when the I/O pin with the pull-up resistor is being toggled. The transition takes about 10 ns.

both methods would lead to a square wave at the (now) joint output of P0.0 and P3.1. You would be right, but you would not be 100% right. Interesting things happen during the 0-to-5-V transition.

First let's consider the case that V_{DD} was being input to P3.2. In this case, Q5 is always cut off and it plays a very little role in the operation of the circuit. Each time Q, the input at the far left of Figure 3, is set to zero, the joint output of the circuit is 0 V. When Q is set to one, then for two clock cycles—in our case about 200 ns—the large transistor, Q2, conducts and the output should transition quickly from 0 V to V_{DD} (which was 5 V in our microcontroller). Photo 1 shows the measured output of the circuit, and you can see that the transition is indeed quite fast.

Now let's consider the case that 0 V was being input to P3.2. Here, Q5 is used to ground the common output. For half of each cycle, it grounds the common output and for half the cycle it is cut off and enables the common output to float. When Q5 is conducting, the output of the circuit is (approximately) 0 V. What happens when Q5 stops conducting? At this point, the output is low, and it will remain low for a little while because of parasitic capacitances. (In this experiment, an oscilloscope was used to measure the output voltage. The oscilloscope itself has a small capacitance associated with it.) As long as the output is reasonably low, only Q3, the smallest of the transistors, conducts. This will charge the parasitic capacitances—but it will take a while. When the voltage at the output rises enough that the inverter at the gate of Q4 "sees" a one, the input to Q4's gate will go to 0 V and Q4 will start conducting. At this point, the joint output will start increasing faster.

Photo 2 shows the measured output of the circuit at a 0-to-5-V transition when P0.0 is being toggled. Initially, the output voltage rises linearly. It seems that Q3 is saturated and provides a constant current to the parasitic capacitances at the output. After a couple of microseconds, the output rises sufficiently to "turn on" Q4. At this point, the voltage rises faster. In total, the rise time for the 0-to-5-V transition is about 3 µs—several hundred times longer than when P3.1 is being toggled.



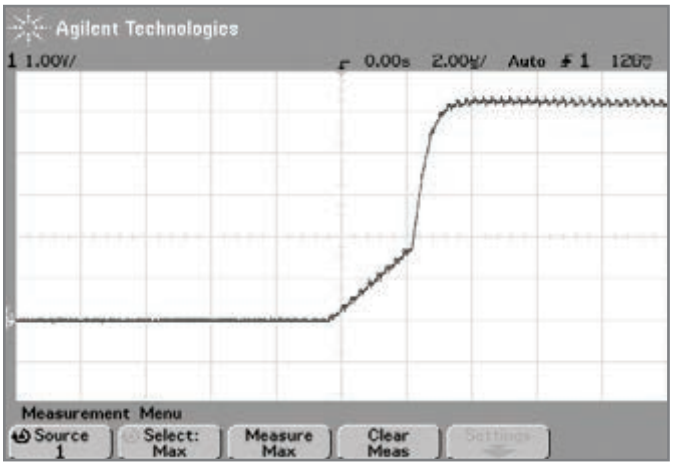


Photo 2—The 0-to-5-V transition when the I/O pin without the pull-up resistor is being toggled. Initially, the voltage rises linearly and slowly. In this initial stage, only Q3, the very small transistor, is conducting. After a bit more than 2 µs, Q4 turns on too, and the voltage rises faster. The total transition time is on the order of 3 µs.

WHAT'S INSIDE MATTERS

We've seen a fair amount about the internal structure of an I/O pin with a pull-up "resistor"—which is not really a resistor, of course. Most importantly, I described a very simple experiment that enables us to see the effect of each of the PMOS transistors that together comprise the pull-up "resistor." Finally, we see how important it is to understand the internal workings of your microcontroller if you want to make sure that it performs optimally.

Shlomo Engelberg (shlomoe@jct.ac.il) is former editor-in-chief of the IEEE Instrumentation and Measurement *magazine. He received bachelor's and master's degrees in Engineering from The Cooper Union and a PhD in Mathematics from New York University's Courant Institute. He is an associate professor in the Electronics department at the Jerusalem College of Technology. He is the author of many articles and several books, including* Digital Signal Processing: An Experimental Approach *(Springer, 2008) and* ADuC841 Microcontroller Design Manual: From Microcontroller Theory to Design Projects *(Circuit Cellar, 2011). Shlomo's technical interests include applied mathematics, instrumentation and measurement, signal processing, control theory, and 8051 microcontrollers.*

REFERENCE

[1] Intel Corp., "MCS 51 Microcontroller Family User's Manual," 1994, www.industrologic.com/MCS51FamilyUsersGuide.pdf.

RESOURCE

Rickey's World, "8051 Ebooks and Tutorials [ebooks and tutorials for 8051]," 2010, www.8051projects.net/download-d215-intel-mcs-51-8051-user-manual.html.

SOURCE

ADuC841 MicroConverter
Analog Devices, Inc. | www.analog.com

AP CIRCUITS
PCB Fabrication Since 1984

As low as...
$9.95 each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com

VISA MasterCard PayPal IPC MEMBER ASSOCIATION CONNECTING ELECTRONICS INDUSTRIES BBB

Gigabit Technology

- ARM-based
- System on a Chip
- Gigabit Ethernet
- Small, Cheap, Fast
- Both QFP and BGA Packages
- Standard Development Tools
- 10 Year Life Guarantee
- Royalty Free RTOS, TCP/IP V4/6

$10.00 each (Qty 100)

The gridARM™ System on a Chip (SOC) is a high performance, low cost, low power, highly integrated single chip with 10 / 100 / 1000 Mbps Ethernet, USB, CAN, Serial, SRAM Memory, SPI, I2C, RTC and internal peripherals designed to provide a complete solution for embedded applications.

THE NETWORKING EXPERTS

gridconnect.

800.975.4743 USA • 1 630.245.1445
gridconnect.com/gridarm.html

Leaders in the embedded and networking marketplace providing network hardware, high quality software and services

by Bob Japenga (USA)



Concurrency in Embedded Systems (Part 2)

Atomicity and TOCTTOU

The first part of this article series introduced the topic of concurrency in embedded systems. This article discusses two common problems associated with embedded system designs containing concurrency: using nonatomic operation in concurrent threads and time of check to time of use (TOCTTOU).

In the first part of this article series, ("Concurrency in Embedded Systems (Part 1): An Introduction to Concurrency and Common Pitfalls," *Circuit Cellar* 263, 2012), we defined concurrency as a common feature found in embedded systems. We saw that concurrency takes place any time two or more activities can happen in the same time segment. We saw that these concurrent operations only cause problems when they interact with each other. We looked at some common pitfalls that happen with such systems and discussed, in thin slices, how priority inversion can happen and what you can do to prevent it.

As I sat down to write this article, I thought about the other common pitfalls I mentioned last time and asked myself: "Which one do I want to talk about this month?" I decided to discuss how nonatomic operations wreak havoc in our embedded systems and how to prevent the chaotic situation. I'll also discuss a somewhat related issue: time of check to time of use (TOCTTOU).

DEFINITIONS

Let's start with some definitions. An atomic operation is any operation that cannot be interrupted by another operation. In a microprocessor, most single-assembly instructions cannot be interrupted until they complete their operation. If you design embedded systems, you should know which operations are atomic whenever you are operating on a shared resource. Using nonatomic operations on shared resources can lead to the generation and use of noncoherent data in your systems.

CASE STUDY #1: SHARED MEMORY

After a brief amount of preplanning for this article, I was called into a meeting to discuss a problem

HELPFUL DEFINITIONS

Atomic operation: An operation that is non-interruptible by any other operation and never presents partial results to an outside observer

Alpha testing: Testing performed by an independent team on a system installed at a place other than the targeted customer's site

Shared resource: Any provider of information that is read or written to by separate threads

KYZ pulse: In a mechanical electrical meter, a pulse that changes state every half rotation of the meter's disk and represents a quanta of energy

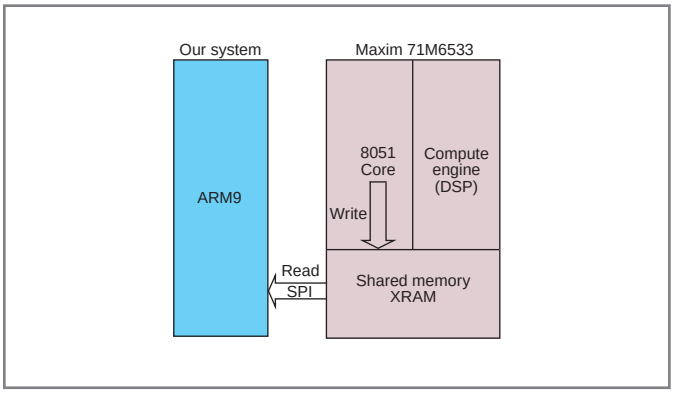


Figure 1—Our system used a SPI bus to talk to a Maxim Integrated Products 71M6533 Teridian energy-metering IC. Both systems access this shared memory interface at a 1-Hz rate.

we were having during the alpha testing of a new product. The problem manifested itself in one of our embedded systems by reporting bad data to its host. The system was measuring energy usage in kilowatt hours and reporting this data to a remote host. Our interface contained a Maxim Integrated Products 71M6533 Teridian energy-metering IC that we talked to over a SPI bus. This is a powerful energy-monitoring chip that can be used to measure power in three-phase electrical systems. This chip was programmed by another vendor to provide a memory interface we accessed over SPI. Both subsystems access this shared memory interface at a 1-Hz rate (see Figure 1). After some analysis, we concluded that the problem could be caused by a classic nonatomic concurrent operation. Let me explain by oversimplifying the details while maintaining the essence of the problem.

We read cumulative energy usage created by the Maxim energy chip with a 4-byte read of a shared memory region over SPI. An 8051 core inside the energy chip writes to this 4-byte region. Since both our 4-byte read and the 8051's 4-byte write were interruptible (nonatomic), at times we could get incoherent data. An example is shown in Table 1.

The end result is that, in a sequence of reads, we would obtain energy readings of:

255
511
256

The error is greater if the collision occurs on the rollover of the second, third, or fourth byte. Of course, those occur far less frequently than my example. This doesn't happen often, but with more than 2.5 million seconds in a month and more than 100,000 meters in the field, it only has to happen once in 250 billion reads to happen once a month.

CASE STUDY #2: INCREMENTING A COUNTER

Many years ago, we ported some code from a customer's energy-monitoring system to a new platform we were developing. The system consisted of a single loop with one timer interrupt routine. Thus, there were two concurrent threads. The most critical piece of information was the total energy read from each of four separate channels. In the timer thread, four KYZ pulses indicative of energy were polled and incremented in a 32-bit word each time a low-to-high transition was made. The software was written in C. The processor was an 8051 derivative. The timer thread would have code like what is shown in Listing 1. The background thread would send the `KYZ_counter` to the host, display them on the screen, or log them to a log file.

The problem was that occasionally the `KYZ_counter` data would be wrong when sent to the host on the display or in the log. How could this happen? This happens because incrementing the `KYZ_counter_1` (as well as counters 2–4) was not an atomic operation even though it was a single C instruction. Underneath this one C instruction would be a half dozen 8051 instructions—any of which could be interrupted. As we previously saw, the problem will occur any time one of the 8-bit portions of the 32-bit number rolls over from 0xFF to 0x00 after it has interrupted a nonatomic read of the data. If the same code was ported to a 32-bit ARM9 processor, which has atomic 32-bit operations, there would be no problem.

8051/Maxim Operation	Our SPI Read	Byte	Memory Location Contents
8051 writes byte 1 of 0x000000FF		0x00	0x00000000FE
8051 writes byte 2 of 0x000000FF		0x00	0x00000000FE
8051 writes byte 3 of 0x000000FF		0x00	0x00000000FE
8051 writes byte 4 of 0x000000FF		0xFF	0x00000000FF
1 s later, the 8051 writes byte 1 of 0x00000100		0x00	0x00000000FF
8051 writes byte 2 of 0x00000100		0x00	0x00000000FF
8051 writes byte 3 of 0x00000100		0x01	0x00000001FF
	Our system reads byte 1	0x00	0x00000001FF
	Our system reads byte 2	0x00	0x00000001FF
	Our system reads byte 3	0x01	0x00000001FF
	Our system reads byte 4	0xFF	0x00000001FF
8051 writes byte 4 of 0x00000100		0x00	0x0000000100
	1 s later, our system reads byte 1	0x00	0x0000000100
	Our system reads byte 2	0x00	0x0000000100
	Our system reads byte 3	0x01	0x0000000100
	Our system reads byte 4	0x00	0x0000000100

Table 1—Our 4-byte read and the 8051's 4-byte read. At times, both reads were nonatomic, resulting in incoherent data.

Listing 1—Time thread code

```
if (KYZ_1 != KYZ_1_last_value)
{
            KYZ_counter_1++;
            KYZ_1_last_value = KYZ_1;
}
```

CASE STUDY #3: READING A 16-BIT COUNTER ON AN 8-BIT PROCESSOR

I first ran into this problem while working with a 16-bit counter in a single-threaded program running on an 8-bit processor. As we saw in Part 1 of this article series, sometimes the concurrency occurs in hardware as we will see in this case.

The design had a 16-bit counter used to count bottles going down a conveyor. The counter was read by an 8-bit processor in two operations. The counter updated in nanoseconds and the processor read the counter in two 8-bit reads about 1 µs apart. Hopefully you see the scenario coming. The counter is sitting at 0x1FF when the processor reads the low-order byte (0xFF). By the time the processor has read the high-order byte, the counter has incremented to 0x0200. Thus the processor sees 0x02 and sets its internal variable to 0x2FF rather than either of the two correct results: 0x200 or 0x1FF. You just processed 255 extra bottles.

WHAT ARE SOME SOLUTIONS?

There are four basic solutions to avoid these kinds of pitfalls: *Don't share memory unless absolutely necessary.* Remember, concurrency problems only manifest when we share resources.

Make all operations on shared resources atomic (noninterruptible) by using some kind of locking mechanism. For example, in Case Study #2, you would want to lock out any reads while the data was being read or written (perhaps by disabling interrupts). Or in our simple single-loop system, you could create a snapshot of the data at the top of the loop with interrupts disabled and then freely use this coherent snapshot of the data throughout the loop. In Case Studies #1 and #3, that was impossible because the 8051 code was written by a vendor.

Build an indication of the coherency of the data into the structure. Using the hardware model, we could provide a busy bit on the data structures indicating that the data structure is being changed. Before changing the data, set the busy bit. After writing the data, clear the busy bit. It is also a good practice to define the maximum amount of time the busy bit can remain set. Again, we didn't control the 8051 in Case Study #1. As noted in the next section, we will see how this too must be carefully used. In Case Study #2, we could create a data structure with a header, the KYZ counter, and a footer sequence number.

Use multiple reads. If the writing of the data is slower than the rate you are reading it, you can perform multiple reads looking for coherent data. This is the least deterministic of the solutions. It would work well in Case Study #3.

TOCTTOU

The final concurrency issue we will look at this month sounds more like a South American bird. TOCTTOU (pronounced "tock too") vulnerability occurs when the data changes after it is read. Perhaps the simplest commonplace example of how this can happen in everyday use is in an airline reservation. The user looks at which flights are available at 7:00:00 (time-of-check) but then attempts to book the once-available flight at 7:01:00 (time-of-use) and is rejected because the flight has already been booked.

```
The data is 0x000001FF
SPI reads busy bit and it is not set (Time-of-check)
8051 sets the busy bit
SPI reads byte 1 as 0xFF (Time-of-Use)
8051 writes byte 1 as 0x00
8051 writes byte 2 as 0x20
SPI reads byte 2 as 0x20
8051 writes byte 3 as 0x00
8051 writes byte 4 as 0x00
SPI reads byte 3 as 0x00
8051 clears the busy bit
SPI reads byte 4 0x00
SPI reads busy bit and it is not set
```

Figure 2—A potential condition that could occur when the solution that involves building an indication of the concurrency of the data is applied to Case Study #1.

Let's take an in depth look at the third solution above (busy bit) and show how a faulty implementation of the busy bit solution fails because of TOCTTOU issues. If we wrongly apply this solution to Case Study #1, the condition that could occur is shown in Figure 2.

In this case, even with the busy bit, because of TOCTTOU, the SPI gets incoherent data (0x2FF instead of either 0x1FF or 0x200). The SPI reads the busy bit (time-of-check) and takes action on the busy bit (time-of-use) and the SPI has some bytes from the first sample and some from the second. When both threads of a system or two independent systems perform nonatomic reads and writes, a simple busy bit won't work.

WHAT ARE SOME SOLUTIONS TO TOCTTOU?

A simple solution to this would be to use a more complicated busy indicator, which is really an 8-bit counter where the low-order bit indicates busy.

The writing device would do the following:

```
flag |= 0x01;
WriteData();
flag++;
```

Listing 2—Results of the reading device

```
flag1 = ReadTheFlag();
if (flag1 & 0x01 == 0)
{
            ReadData();
            if (flag1 == ReadTheFlag())
            {
                        // Use the data
            }
            else
            {
                        // Wait and try later
            }
}
else
{
            // Wait and try again later
}
```

Note: OR a `1` into the flag instead of using `flag++` to guarantee synchronization. The reading device results are shown in Listing 2. Now we would be protected from incoherent data regardless of the overlaps of reading and writing the shared resource.

PROBLEMS HAPPEN

Concurrency issues can create nightmarish problems for embedded systems designers. The designer of the software in the energy chip did not carefully think through the implications of having three devices all reading and writing to a common shared memory when the interface was created. We were using the interface and didn't know that it was a shared-memory interface. The problems of nonatomic operations and TOCTTOU issues on shared resources can create problems that are like hurricanes in Hartford, England: they hardly ever happen, but when they do...

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

RESOURCE

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

SOURCE

71M6533 Teridian energy metering IC
Maxim Integrated Products | www.maxim-ic.com

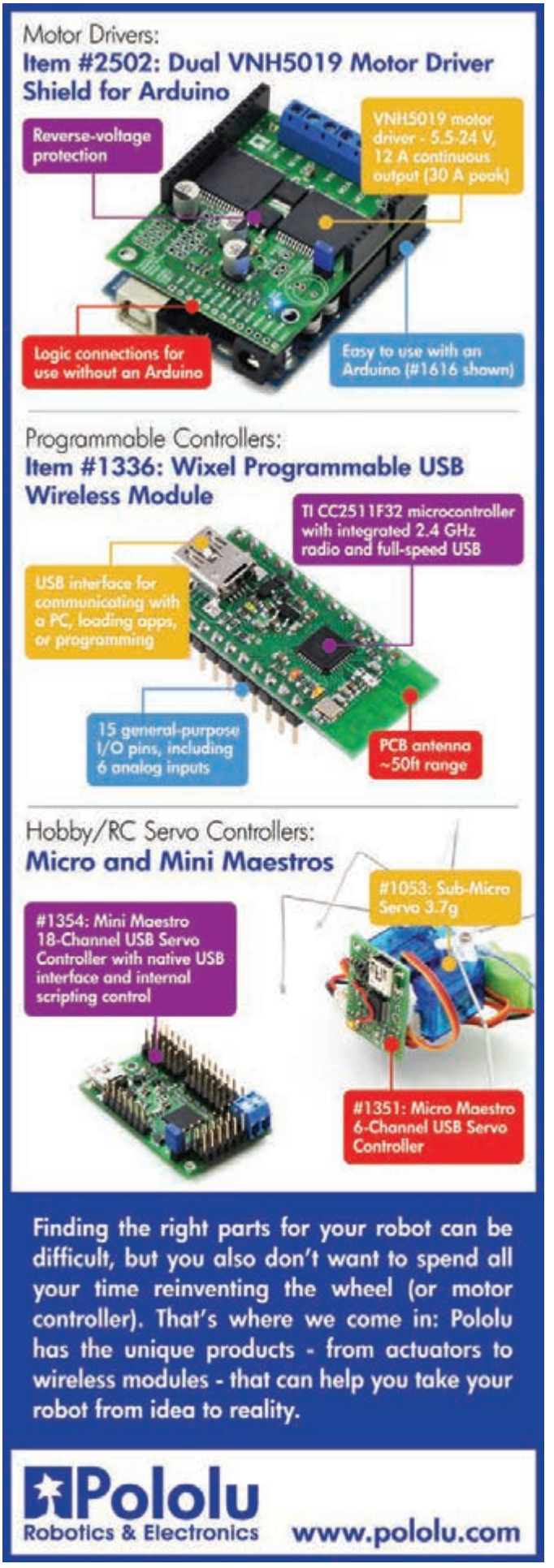
NEED-TO-KNOW INFO

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

Concurrent Small-C for PIC18 MCUs
by Andy Yuen
***Circuit Cellar* 188, 2006**

You can write concurrent programs for a PIC MCU with the Concurrent Small-C (CSC) programming language. This article covers the language's main features and then describes a complete multitasking CSC demonstration program. Topics: CSC, RTOS, Concurrency

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



by George Novacek (Canada)



Project Development (Part 2)

Development Process, Milestones, and Design Reviews

A competent electronics engineer must conceptualize and then follow a proper project development plan. To do so, the engineer must understand the essential topics of development milestones, design reviews, and configuration control.

In the first part of this article series, I detailed the project development process. This was graphically shown in Figure 1, which I once again include here for your convenience. Now I'll talk about the milestones (shown in green) and design reviews attached to them.

The first milestone—the project kickoff—isn't much more than just a green light to start work. Usually it's a telephone call or e-mail triggering a small celebration. A formal contract often follows at a later date. In a perfect world, before starting, you should have received a signed contract with a specification and a customer-approved statement of compliance, (i.e., compliance matrix), submitted by you in response to the customer's specification during negotiations. The compliance matrix is one of the most important contractual documents. An excerpt from one is shown in Table 1.

COMPLIANCE

The compliance matrix addresses every single paragraph of the specification, statement of work, or other contractual documents. It also describes how compliance will be achieved. Without careful review and response to compliance matrix documents, you may be unwittingly preparing grounds for future disputes and costly resolutions. This is not because the customer has some malicious intent. Many documents are prepared from templates or by "cut and paste." Consequently, some requirements may not apply or

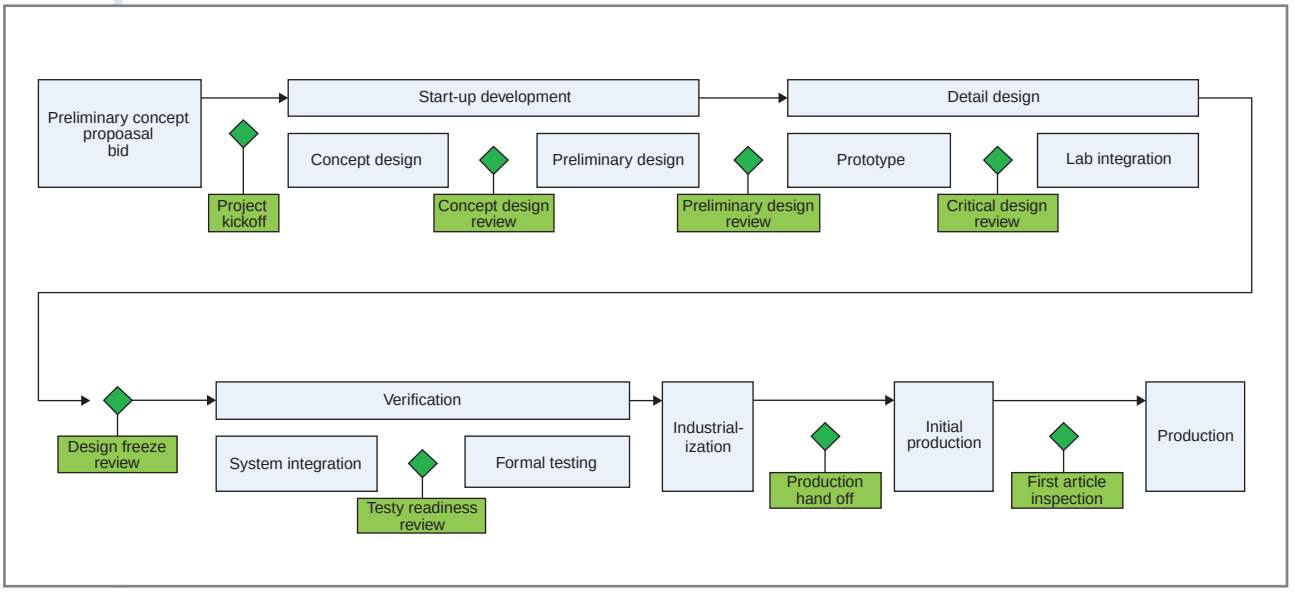


Figure 1—Project flow diagram from bidding to production

Requirement Heading	Para.	Comp.	Note
DC Power	7.2.1	No	Two primary supplies needed for 1-s power outage
Partial or total power failure	7.2.2	Yes	Compliance by test
Ripple amplitude and voltage transients generated by the equipment	7.2.3	Yes	Compliance by test
Ground separation within the equipment	7.2.4	No	Distinct grounds, except shielding/case ground, have a single connection point internal to the equipment
Electrical connections	7.2.5	No	COTS D-type connectors will be used
Electronic equipment	7.2.6	No	No mechanical keying of the boards is needed since they are interchangeable, no parts list will be provided at PDR

Table 1—An excerpt from a compliance statement

may even conflict with others. By clearly stating your noncompliance, you are forcing the issue to be addressed and resolved without delay at the component level. Once the project begins, you will have one point of contact—most likely a bureaucrat. The bureaucrat will hesitate questioning his own departments, because the contractual documents have been signed and their job is to ensure compliance.

REQUIREMENTS, REPORTS, & REVIEWS

Take, for example, a common requirement, copied and pasted for ages, that connectors exhibit a minimum of 2-MΩ resistance between pins, when 500 VDC is applied. This is to be confirmed by testing during the acceptance test procedure (ATP) prior to shipping. It can rarely be done. Even if the circuit boards were disconnected, there are electromagnetic interference (EMI) filters, electrostatic discharge (ESD), and lightning suppression devices directly attached to the connectors. These are sometimes integral parts of the connectors, often with 50-VDC maximum rating. There are ways to address this dilemma, but if you don't do it in the compliance matrix, you will eventually be begging a bureaucrat, who will think this "noncompliance" should result in your price reduction, for a waiver.

Although a part of your bid, your compliance response should also contain a list of documentation, testers, and prototypes (i.e., deliverables) confirming exactly what the customer will receive.

For example, a failure-mode and criticality analysis (FMECA) report can be prepared on a block diagram or a component-level basis. There is more than an order of magnitude difference in the amount of work to prepare a component-level FMECA. If left undefined, the customer will likely demand the component-level analysis. Similarly, testers, prototypes, or engineering support on site, if ambiguous, can quickly destroy the entire development budget.

Unfortunately, with concurrent engineering the specification is rarely signed early by the customer. This exposes you, the developer, to ever-increasing risk. It is somewhat mitigated during the concept design through the joint definition phase (JDP). Here, the system integrator and his subsystem suppliers work together to, in theory, nail down and finalize their respective specifications. Concept design review should then be the end of the JDP, with the subsystem developer starting work in earnest. The success of the JDP rests mainly on the integrator's leadership and the discipline of the attending engineers. I lived through several JDPs in their early days. Imagine a hall filled with more than 200 engineers, no clearly defined goals, and no leadership. The one-week planned activity stretched to three months and the results were dismal. Fortunately, we have come a long way since then.

Preliminary design review (PDR) is the milestone when the system architecture and partitioning of functions between hardware and software, shown by block diagrams and pseudo code, are reviewed. Preliminary reliability analysis, FMECA, FTA, weight, electrical load, and other analyses are critiqued by peers to ensure the design goals will be met. There are established entry criteria to define what is presented and what exit criteria needs to be approved. When approved, detail design can start. It is important to maintain minutes and assign responsibility for action items during design reviews and meetings. Before the meeting is adjourned, all are signed by the participating parties. Minutes and action items issued at later dates are always subject to differing interpretations and arguments.

The culmination of the detail design phase is critical design review (CDR). At this point, the design is essentially finished. Reliability and maintainability analyses have been done, the software has been documented and module-tested, the hardware has passed rudimentary environmental and EMC testing, the bill of material (BoM) has been issued to purchasing, component obsolescence has been addressed, orders have been placed, packaging and PCBs have been ordered, prototypes have been built, and laboratory integration becomes the major activity to ensure everything works as required.

Upon completing integration the design freeze follows, while prototypes are shipped to the customer for the system integration. It should come as no surprise that many times the "frozen" design doesn't work as intended. Thus, the seemingly straightforward activity shown in Figure 1 in reality looks more like many fuzzy loops. At some point, however, all bugs

"Preliminary design review (PDR) is the milestone when the system architecture and partitioning of functions between hardware and software, shown by block diagrams and pseudo code, are reviewed. Preliminary reliability analysis, FMECA, FTA, weight, electrical load, and other analyses are critiqued by peers to ensure the design goals will be met. There are established entry criteria to define what is presented and what exit criteria to be approved. When approved, detail design can start. "

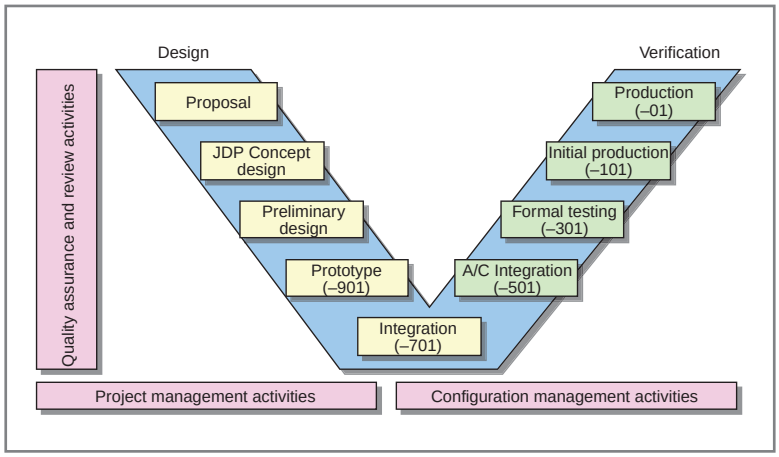


Figure 2—Development phases for aircraft equipment

In my experience, the process depicted in Figures 1 and 2, while it is required by software and hardware development standards, is somewhat of a wishful thought. The drive to meet the ever shorter deadlines and lower costs makes the sequential process impossible. The customer may be late delivering a specification detail, but will not accept it as your excuse for sliding the schedule. What you thought was a brilliant design idea may not work. This is where a competent engineer shines. By understanding the processes, the certification requirements, and the technical issues, the engineer manipulates the schedule, runs some processes in parallel or ahead of time, and, in the end, delivers on time.

Next month, I'll continue with a look at other project development activities.

are ironed out and the test readiness review confirms the product is ready for formal testing and approval.

DEVELOPMENT ENGINEERING

With the formal tests passed, industrialization takes place and the product is handed off to manufacturing. The first product made by the factory without engineering involvement undergoes the so-called first article inspection (FAI). If approved by the customer or the management, in case of an internal customer, production may commence.

Figure 2 shows development engineering activity in more detail. It's important to remember that each phase represents a lot of activity with one exit when the criteria are satisfied. The entire process should be supervised by quality assurance (QA) to prevent cutting corners and to maintain configuration control.

Configuration control is a critical activity. Companies have different approaches to it. Engineers should maintain log books with numbered pages instead of using loose paper for sketches and notes. All documents and drawings must have numbers assigned to them during the early stages of the program. I'll explain the process in my next article. Preliminary versions of those documents and drawings should not be signed but identified as preliminary. Although customers often insist on receiving signed documents, you should not sign them until signed specification and the compliance matrix approval have been received from the customer.

My system assigned breadboards produced in the lab and initial software series –901 version numbers. These would be bumped up to –903, –905, –907, and so forth to reflect even the smallest changes. The first prototype on a PCB would start version series –701 and once again higher numbers would reflect modifications. –501 series was good enough to be installed on an aircraft for the system integration. The version to undergo formal testing begins series –301 until passed and handed over to industrialization for initial production as series –101. Finally, following approved FAI, normal production series –01 commenced.

Why the odd numbers only? In some industries (i.e., aerospace) there are nearly identical systems, but one destined for the left and one for the right side. Even numbers are reserved for the right side only, while odd numbers are for the left as well as default, in case the right-side configuration doesn't exist.

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

REFERENCE

G. Novacek, "Project Development (Part 1): Plans, Schedules, and Task Management," *Circuit Cellar* 264, 2012.

NEED-TO-KNOW INFO

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

Smart Project Management
by George Novacek
***Circuit Cellar* 254, 2011**
Professional-level system specifications are imperative to any design project. Having project specifications in place before development saves time, cost, and effort. Here you learn why leaving things "to be determined" can lead to disaster. Topics: Project Planning, Specifications

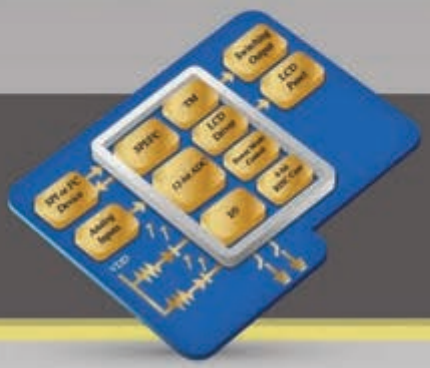
The Project-Ready Designer
A Refresher on Project-Essential Concepts
by George Martin
***Circuit Cellar* 248, 2011**
Project management should involve frequently reviewing design essential topics, reconsidering all tool options, and staying informed about new technology and design techniques. Topics: Project Management, Design, Hardware

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

HOLTEK
www.holtek.com

All In One

Ultra Low Power Consumption - Highly Functionally Integrated Flash MCU Solutions



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

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

STD Flash MCU HT66/67/68Fxx Series

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



Touch Flash MCU	**STD Flash MCU**	STD 8051 Flash MCU	USB STD Flash MCU	32bit MCU	Enhanced OTP MCU
TinyPower™ MCU	Power Management	UART MCU	Phone MCU	EEPROM	WLED Backlight

HOLTEK SEMICONDUCTOR INC.

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

by Ed Nisley (USA)



MOSFET Channel Resistance

Tester Hardware

Measuring MOSFET characteristics helps verify that the datasheet values will apply in an actual circuit. This Arduino-based tester applies known currents and voltages, and it sends the results to a PC for data logging and analysis. This article details the hardware design and verifies the overall calibration.

Although the ideal MOSFET equations and Spice models presented in my June article ("MOSFET Channel Resistance: Theory and Practice," *Circuit Cellar* 263, 2012) provide enough guidance to rough out a transistor circuit, there's no substitute for measuring real devices to see how they perform. The Arduino-based hardware shown in Photo 1 automatically characterizes a MOSFET over the range of currents and voltages relevant to my application by applying known inputs and measuring the outputs to produce a table of data points.

In this article I'll describe the MOSFET tester circuitry and show some of the results.

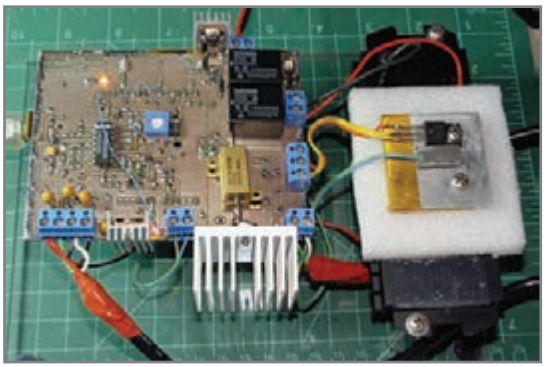


Photo 1—The MOSFET tester PCB completely hides the Arduino that runs the control program and communicates through the USB cable on the left edge. The large-finned aluminum heatsink in the foreground dissipates heat from Q1, the drain current source, during high-current parts of the test. The MOSFET under test and a thermistor assembly occupy the aluminum block mounted atop a Peltier module on the black CPU heatsink to the right.

MOSFET TESTER OVERVIEW

The MOSFET tester characterizes a transistor's behavior by computing the drain resistance at each combination of measured voltage and current. Photo 2, which resembles the graphs found in MOSFET datasheets, suggests how the tester must operate. The data emerged from a BUZ71A n-channel MOSFET from my parts heap.

The blue traces plot MOSFET drain current i_D against drain voltage v_{DS}, with each trace at a constant gate voltage V_{GS}. Producing those traces required setting the gate voltage, then either varying the drain voltage and recording the current or varying the drain current and recording the voltage.

Because I'm interested in the MOSFET's linear region of operation, where the device presents a low drain resistance that would draw a large current from the power supply for small drain voltages, I applied a limited drain current and measured the resulting

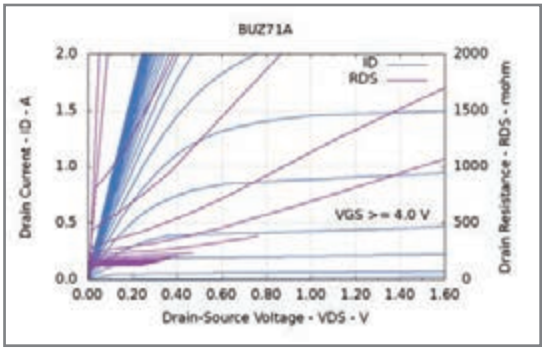


Photo 2—Testing a BUZ71A MOSFET shows the expected behavior as V_{GS} ranges from 2.0 V to 10.0 V in 1.0-V steps. The tester's 2-A maximum I_D and 1.6-V maximum V_{DS} restrict the data points to the lower-left corner of the graphs seen in MOSFET datasheets.

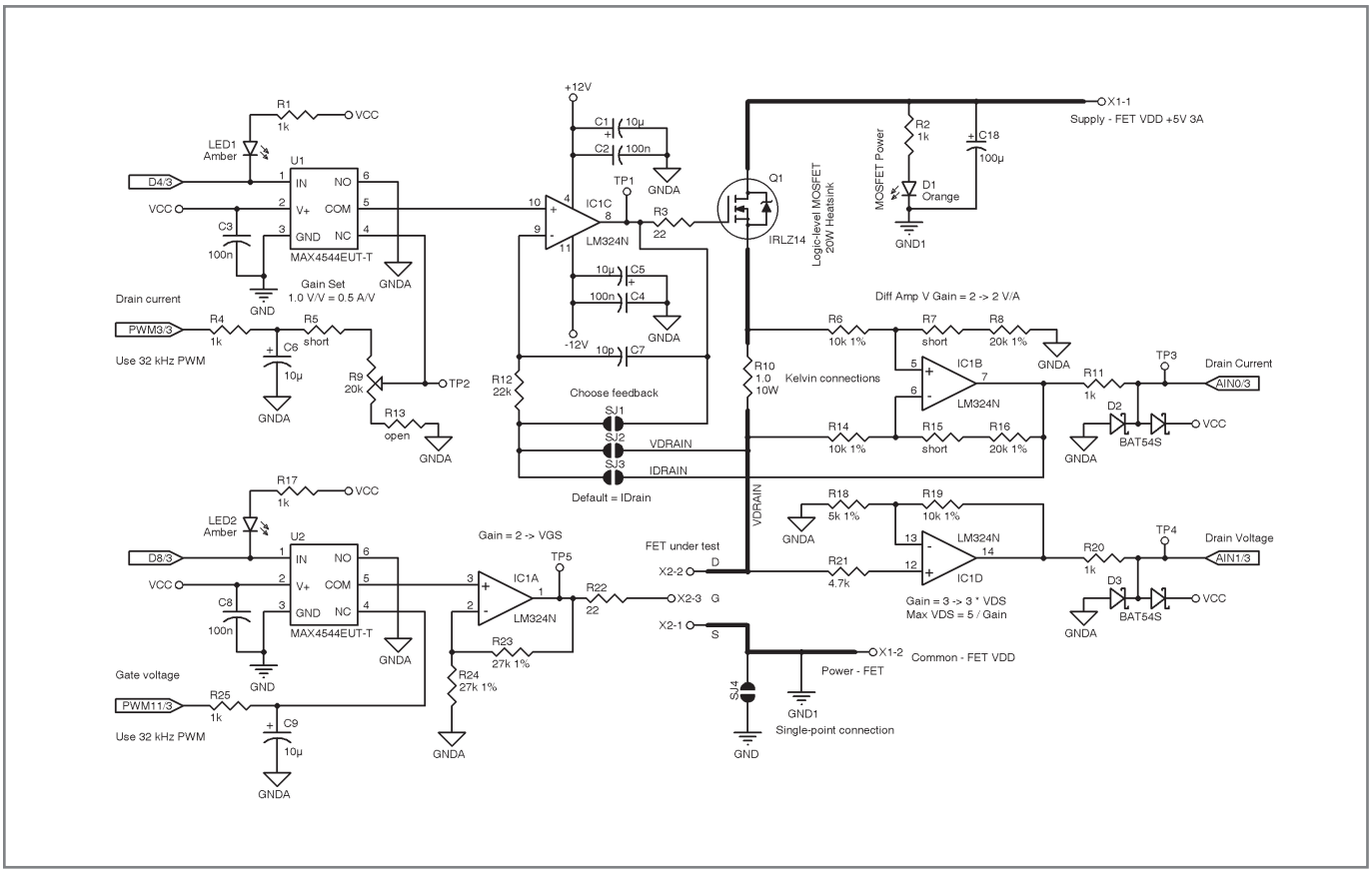


Figure 1—An LM324 quad op-amp provides the analog functions needed for the MOSFET tester. Q1 limits the MOSFET's drain current to the value set by the Arduino PWM3 output through IC1C. The 1-Ω, 10-W resistor near the center senses the actual drain current as a differential voltage that IC1B converts to a ground-referenced value for the Arduino A0 analog input and the IC1C current limiter.

drain voltage. That means the standard graphs have current, the independent variable, along the Y axis. Your math teachers told you to put the independent variable on the X axis, but we're not going to flip our transistor graphs just for them!

The actual current will be the lower of the current limiter's setpoint and the MOSFET's i_D for that V_{GS} and v_{DS}, because the current limiter cannot force more current through the MOSFET than the transistor can conduct, nor can the MOSFET draw more current than the limiter provides. Each trace in Photo 2 starts at the origin with the drain current limiter at 0.0 A and proceeds upward for successively higher current settings until the measured drain current exceeds the tester's 2-A limit or the drain voltage exceeds about 1.6 V.

Commercial transistor curve tracers have much larger current and voltage limits, of course, because they must test a wide variety of devices. You can extend the limits of my circuit with more complex hardware, which I'll leave as a project for the interested reader. Hint: there's a reason why curve tracers have big fan-cooled heatsinks!

The purple traces in Photo 2 plot the MOSFET drain resistance, computed from v_{DS}/i_D at each measurement point. The traces for the lowest resistances, around 100 mΩ for the BUZ71A, correspond to the steepest blue traces. You can match their endpoints with a vertical ruler. I computed the resistance in the tester's firmware, but many plotting programs can produce that quotient from the raw voltage and current samples.

The schematic in Figure 1 shows the key parts of the MOSFET tester's circuitry. An Arduino microcontroller provides all the control signals, as well as the analog outputs and inputs, through its normal header pins. Although the Arduino interface doesn't appear here, you can download the complete CadSoft Eagle schematic and PCB layout files from the *Circuit Cellar* FTP site.

ARDUINO ANALOG I/O

Although the Arduino has six "analog outputs," those pins don't produce true analog voltages, a peculiarity that often trips up new users. Instead, each one produces a digital PWM signal that requires a low-pass filter to extract its DC value. Unfortunately, the default Arduino startup code sets a fairly low PWM frequency that requires inordinately large capacitors to produce low ripple from the usual single-pole low-pass RC filter.

The tester uses the PWM3 output to set the drain current limit and PWM11 for the gate voltage, both of which use Atmel's ATmega168/328 Timer2 hardware. Changing the PWM frequency requires selecting a different clock source for Timer2 by changing the contents of a single register:

```
TCCR2B = 0x01;
```

That value produces a 32-kHz PWM frequency from the standard 16-MHz clock crystal. That's 64 times the default 488 Hz, high enough that the 16-Hz RC filters in Figure 1 can easily convert PWM to DC. A single-pole filter attenuates the signal amplitude by 6 dB/octave above the filter's –3 dB frequency, which works

out to 66 dB over the 11 octaves separating 15 Hz and 32 kHz. The PWM output's 5-V peak-to-peak signal will be about 2.5 mV after filtering.

Homework: Compute the ripple for the default 488-Hz Arduino PWM frequency for that filter.

I routed the filtered DC signals through Maxim Integrated Products MAX4544 SPDT analog switches, with their other input grounded, to inhibit unintended outputs. When the Arduino emerges from a hardware reset, all the output pins enter their high impedance state. The pull-up resistors on the multiplexer control inputs then select the grounded input, which ensures the MOSFET's gate voltage and drain current remain turned off until the firmware starts running and properly initializes the hardware.

The 10-ms analog filter time constant implies that each output voltage change requires about 50 ms (five time constants) to settle within 0.5%. The firmware imposes a delay after each analog output change to enable settling, which turns out to be the single most time-consuming part of the entire measurement cycle.

Pop Quiz: Compute the settling time for an RC filter that would provide 66-dB attenuation at 488 Hz.

Because the maximum gate voltage required for most MOSFETs will exceed the Arduino's 5-V output, IC1A amplifies the filtered DC voltage by a factor of 2. The classic LM324 op-amp I used can drive its output within 1.5 V of the positive supply, which works out to 10.5 V for the ±12-V supplies I used. You'll need a higher supply voltage, more voltage gain, and perhaps a better op-amp to extend the V_{GS} range beyond 10 V.

Arduino analog inputs have 10 bit resolution over their 0-to-5-V range, which works out to 4.9 mV per LSB. Because the drain voltages will be well below 1 V for typical power MOSFETs, IC1D amplifies the drain voltage by a factor of 3. Unfortunately, that gain means drain voltages above 1.6 V apply more than the 5 V allowed at the Arduino's ADC input. Although the firmware limits the voltage during the test, the op-amp will see the full MOSFET supply voltage when the MOSFET is completely turned off. Dual diode D1, two Schottky diodes in a single SMD package, protect the ADC input from that excessive voltage, as well as any transients below 0 V.

CURRENT CONTROL

The two op-amps in the upper half of Figure 1 limit the drain current into the MOSFET under test. IC1B converts the differential voltage across R10, the 1-Ω drain current sense resistor, into a ground-referenced voltage. The differential amplifier has a gain of 2, producing an overall gain of 2 V/A: a 2-A current produces 4 V.

A fault condition (or, perhaps, a firmware error) could produce more than the 2.5 A that would apply 5 V to the ADC input. Dual diode D2 clamps the ADC input voltage between ground and 5 V to protect the Arduino input.

"Although the Arduino has six 'analog outputs,' those pins don't produce true analog voltages, a peculiarity that often trips up new users. Instead, each one produces a digital PWM signal that requires a low-pass filter to extract its DC value."

The three solder jumpers below IC1C select its feedback connection: SJ1 turns IC1C into a voltage follower, SJ2 tracks the drain voltage, and SJ3 tracks the drain current. The firmware for this column assumes current feedback through SJ3 and, obviously, you should bridge only one jumper at any time.

IC1C drives the gate voltage on Q1 to make the feedback voltage match the DC voltage filtered from the Arduino's PWM3 output. Because IC1B has a gain of 2 V/A, the overall gain (more correctly, the transimpedance) from the microcontroller analog output is 0.5 A/V.

Although I included a voltage divider to adjust the voltage applied to IC1C, the circuit components shown simply pass the filtered PWM output directly to the multiplexer and then to the non-inverting input of IC1C. If you choose voltage feedback, select R5 and R13 to center the R9 trimpot voltage range around the nominal setpoint.

Because the negative feedback loop around IC1C closes through two op-amps, MOSFET Q1 and R10, I added C7 to roll off the high-frequency response and prevent oscillation. LM324 op-amps have a relatively low 1-MHz gain bandwidth product, but contemporary op-amps with higher GBW will require more aggressive compensation.

GROUND CONTROL

The PCB in Photo 1 completely conceals the Arduino Duemilanove plugged in below it. The Arduino controls all the measurements and sends logging data through the USB cable extending beyond the left edge of the photo. The conspicuous finned aluminum heatsink in the front dissipates heat from Q1, the drain current limiter, and the smaller heatsinks cool a voltage regulator and the Peltier PWM current driver. Everything mounts on a transparent acrylic sheet.

The MOSFET to be tested (more formally known as the device under test—DUT) sits on the aluminum block to the right of the PCB, beside the thermistor assembly that reports its temperature. A more accurate measurement would embed a thermocouple bead in the MOSFET package, but this simple method is good enough for my purposes. The white foam gasket, plus a foam top hat that's not in the picture, insulate the block from the surrounding air.

A Peltier module between the block and the black CPU heatsink can vary the MOSFET temperature between 0ºC and 50ºC during the test cycle, with a fan holding the heatsink near room temperature. The Arduino firmware stabilizes the temperature using a proportional integral (PI) control loop with feedback from the thermistor and power from the hardware in the right-rear corner of the PCB. I'll describe how that works in my next article, along with examples of how the MOSFET properties vary with temperature.

The heavy lines in Figure 1 mark the 2-A current flow through the test MOSFET. On the PCB, that current must coexist with op-amp circuitry handling analog signals with a resolution around 10 mV. Reducing that conflict required more-than-usual attention to PCB layout.

I used three separate wall-wart power

25th Anniversary SPECIAL EDITION

Circuit Cellar Archive on USB

Celebrate 25 years of Engineering Excellence with this Special Edition Circuit Cellar Archive packed with all these incredible features...*and a free gift!*

- Special Anniversary USB drive or our traditional CC GOLD USB drive
  Your choice!
- A USB memory upgrade from 16GB to a whopping 32GB!
- All Circuit Cellar magazine issues in print through date of purchase
- Article Code
- All the details from our previous Design Challenges*
- Two years of ELEKTOR Magazine Issues in PDF format (2010-2011)
- Two years of audioXpress Magazine Issues in PDF format (2010-2011)
- *Free Gift!* Circuit Cellar 25th anniversary hat



Save $25.00 for a limited time! Order today at:
www.cc-webshop.com

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

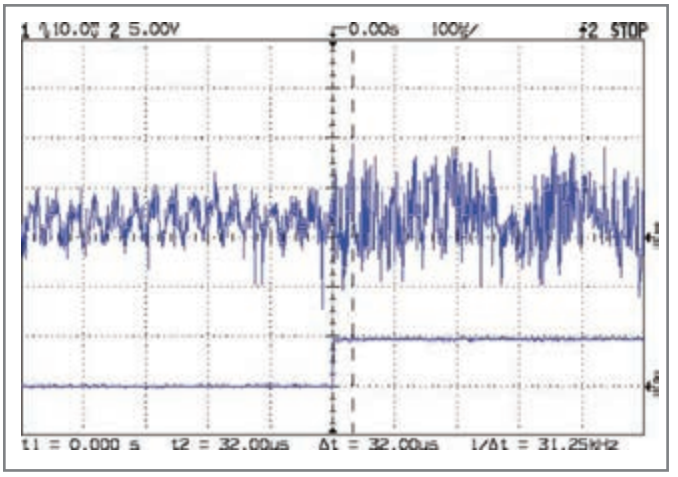


Photo 3—The drain current sense voltage noise has a strong 32-kHz component from the PWM outputs. Overall, the noise contributes ±4 counts to the measurement. The lower trace is a scope sync output from the test firmware.

Listing 1—Averaging three analog samples to produce a single result reduces uncorrelated input noise.

```
#define NUM_T_SAMPLES 3

float ReadAI(byte PinNum) {
word RawAverage;

     RawAverage = analogRead(PinNum);

     for (int i=2; i <= NUM_T_SAMPLES; i++) {
     RawAverage += (word)analogRead(PinNum);
     }

     RawAverage /= NUM_T_SAMPLES;
     return (float)RawAverage;
}
```

supplies to eliminate coupling at the source: a triple-output (+12, +5, −5 V) supply for the main circuitry, a +5-V supply for the MOSFET-under-test drain current, and another +5-V supply for the Peltier module. We tend to think of power supplies as ideal voltage sources, but real-world supplies generally aren't ideal at all. Separating the supplies prevents high current loads from inducing voltage transients in the measurement circuitry, a problem that's extremely hard to find and fix on a completed board.

The PCB layout uses four separate ground planes with single-point connections to eliminate voltage gradients caused by high ground current passing through the sensitive analog circuit components. The MOSFET terminals and current sense resistor in the lower right corner have one plane, the Peltier module driver in the upper right has another, and the analog circuitry between the Arduino header pins lives on a third. A fourth plane connected directly to the Arduino ground pins surrounds the analog plane.

The upper trace in Photo 3 shows that noise on the drain current sense voltage at the Arduino analog input pin amounts to about ±20 mV or ±4 counts; the lower trace is a scope sync output from a test firmware routine. The scope probe ground clips were attached to the analog plane that connects to the Arduino ground pin nearest the analog reference pin. Unfortunately, the Arduino PCB layout simply joins that pin to the same digital ground with all the other logic.

Listing 1 shows that the firmware averages three analog samples to further reduce the noise. Photo 4 shows the results from a 1-Ω resistor connected between the tester's drain and source terminals (the gate terminal is not used in this test), which closely match the expected values. The firmware forces the computed resistance to 0 Ω when the measured current is exactly zero, ensuring that the first point doesn't produce a divide-by-zero error.

Photo 5 shows a 0.1-Ω, 1% resistor that presents a more realistic load appears to be slightly too high at 110 mΩ. However, when I manually measured the voltage across the resistor due to a 1.00-A DC current, the connecting wires added about 6 mV and account for nearly the entire error. Below 50 mV, you can see the effects of the Arduino's 5-mV ADC resolution and the firmware's 50-mA drain current increments.

So it's fair to say the tester produces reasonably accurate results for drain currents below 2 A and drain voltages between 50 mV and 1.6 V, the entire range my MOSFETs will use.

MOSFET MEASUREMENTS

Photo 6a shows the results for an IRF630 MOSFET with a 400-mΩ maximum drain resistance. Photo 6b shows a closer

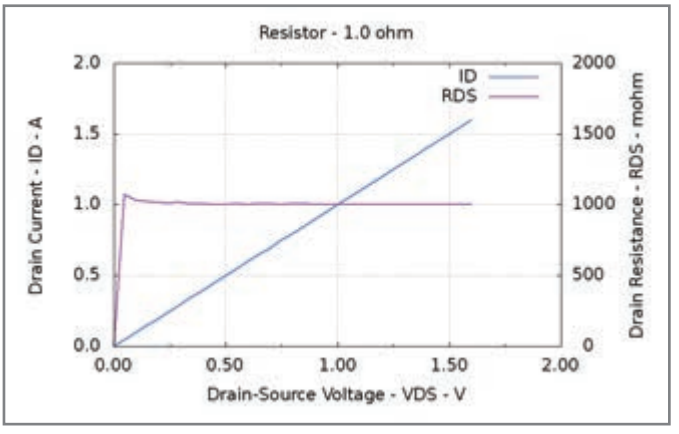


Photo 4—A 1-Ω, 1% resistor between the tester's drain and source terminals produces the expected result. The firmware forces the calculated resistance to 0.0 Ω when I_{DS} = 0 to avoid overflow.

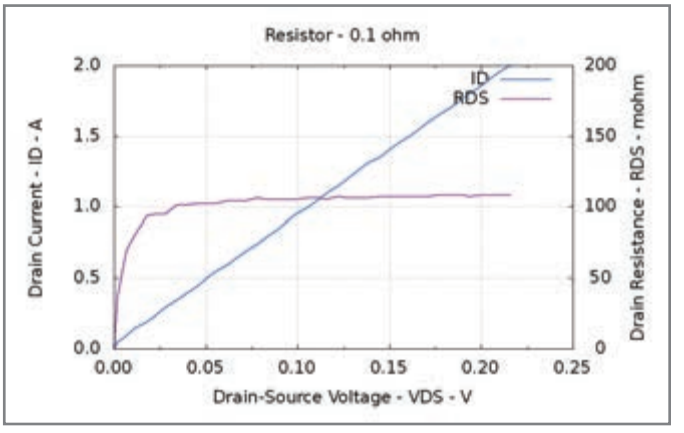


Photo 5—The tester reports a 0.1-Ω, 1% resistor as 108 mΩ, but the resistor's connection wires add 6 mΩ! The 5-mV ADC resolution become obvious at this graph scale.

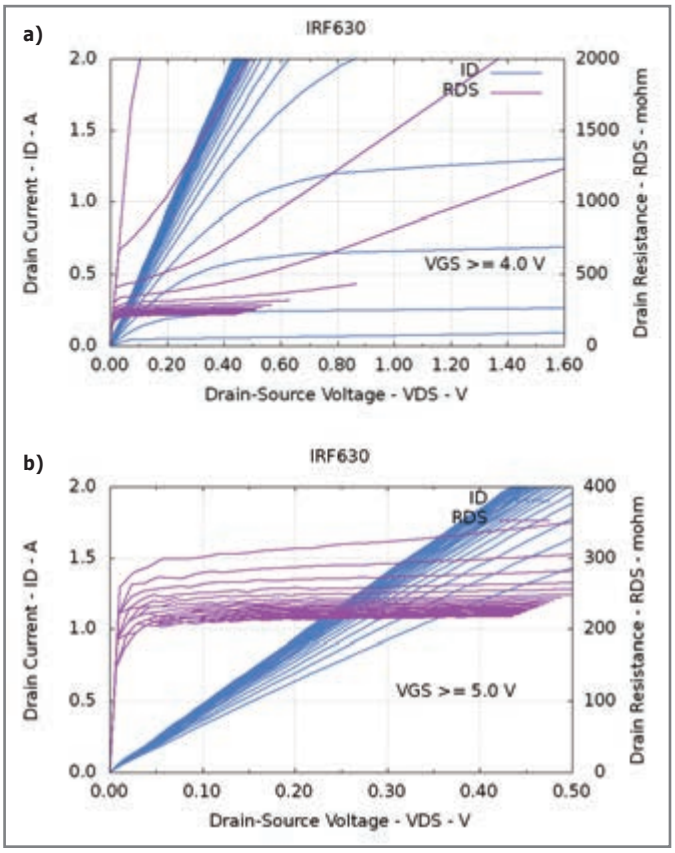


Photo 6a—This IRF630 MOSFET has a drain resistance well below the datasheet's specification of 400-mΩ maximum at V_{GS} = 10 V. The gate voltage steps by 250 mV for each trace. **b—**The IRF630 drain resistance remains nearly constant for currents above 100 mA with V_{GS} above 6 V. Between 5 V and 6 V, the resistance increases slightly with drain current.

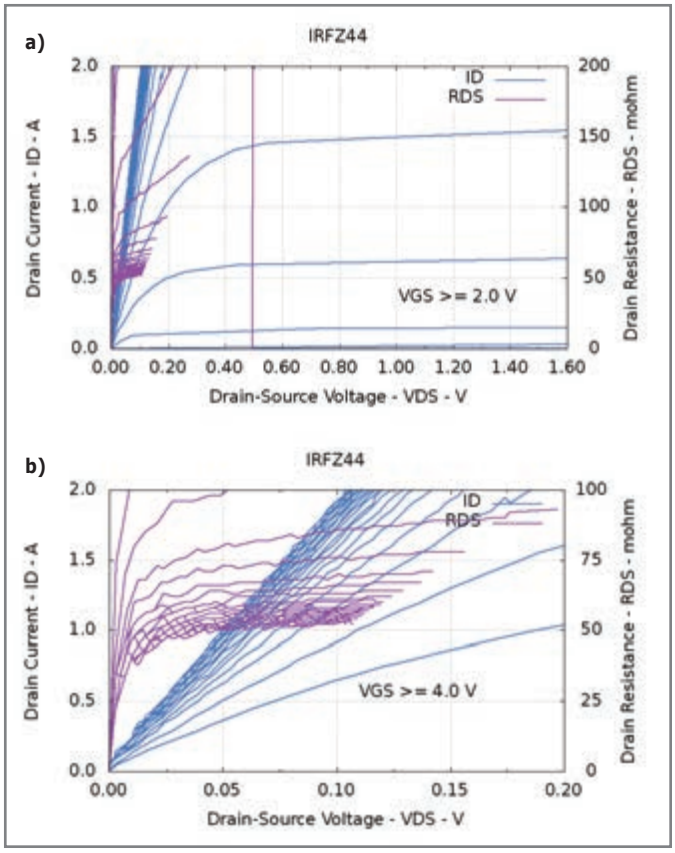


Photo 7a—The logic-level threshold of an IRFZ44 MOSFET produces drain current above 2 A for gate voltages above 3 V. The vertical purple line near V_{DS} = 0.50 V is a spurious value at V_{GS} = 2.0 V. **b—**Any gate voltage above 5 V produces a nearly flat drain resistance curve for this IRFZ44 MOSFET. Unfortunately, the resulting resistance may be too low for my purposes.

look at the linear region near the origin, where r_{DS} stabilizes near 220 mΩ for V_{GS} = 10 V. The IRF630 datasheet specifies the drain resistance at 4.5 A, but it seems reasonable to assume that this particular transistor falls at the lower end of the manufacturing tolerance.

The IRFZ44 MOSFET in Photo 7a has a logic-level gate threshold, so the drain current reaches 1.5 A for V_{GS} = 2.75 V. Photo 7b is a zoomed-in view showing that the drain resistance hits 50 mΩ with V_{GS} = 10 V. The specifications call for 28 mΩ at 31 A, but the connecting wires once again add several milliohms: the 20-mΩ difference probably has more to do with the test conditions than anything else.

In any event, such a low resistance wouldn't work well with the current levels I have in mind: a 1-A current produces only 50 mV, which may be too small for reliable sensing. Now that I can measure my MOSFETs, however, I can proceed with further experiments.

CONTACT RELEASE

The tester hardware applies only positive voltages, limiting it to n-channel enhancement MOSFETs. Some adroit switching could adapt the circuitry to test the p-channel MOSFETs often used for high-side switches, but I'll leave that as another exercise for the interested reader.

All the test results shown here maintained the transistor at 30°C with the Peltier module. In my next article, I'll describe that circuitry and the firmware that controls the MOSFET's temperature, then show how MOSFETs behave in a 0-to-50°C range.

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

PROJECT FILES

To download the schematic and PCB files, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/265.

RESOURCES

Gnuplot, www.gnuplot.info.

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

SOURCES

Arduino Duemilanove
Arduino | http://arduino.cc

CadSoft Eagle PCB Software
www.cadsoftusa.com

MAX4544 SPDT analog switches
Maxim Integrated Products | www.maxim-ic.com

by Robert Lacoste (France)



System-Level RF Design

Designing a radio frequency system—even using precharacterized modules or components—can be a daunting task without the proper methodology and tools. This article illustrates how system-level simulation tools can ease the process.

Welcome back to the Darker Side. In my last article, "Radio Frequency Mixers" (*Circuit Cellar* 263, 2012), I discussed the basics of frequency mixers. In a nutshell, these strange devices somehow translate a signal up or down in frequency thanks to the nonlinear effects of their internal diodes. More precisely, a mixer has two inputs (one signal input and one so-called "local oscillator" input) and one output. It generates the sum and difference of the frequencies F1 and F2 applied to its two inputs. On the nastier side, they also generate plenty of spurious signals in the form of n × F1 + m × F2. For example, mixing a 10-MHz signal and a 100-MHz local oscillator will generate mainly a 110-MHz signal (100 + 10 = 110) and a 90-MHz one (100 – 10 = 90); but also stranger frequencies such as 280-MHz (3 × 100 – 2 × 10) or 130-MHz (1 × 100 + 3 × 10), and so forth. Therefore a bandpass filter is usually used to select the output frequency required by the designer.

If you're not familiar with mixers, you may want to go back and read my last article, as this is kind of a follow up. I'll use the example of a frequency mixer and amplifier circuit to introduce some high-level design methods and tools that could be more than useful if you're working on RF designs. Don't be confused, I will not talk about classical electrical simulators like the ubiquitous SPICE, but I will present you some high-level system design tools. Just follow me through this introduction on RF system-level simulation!

REQUIREMENTS

Every project starts (or should start) with clear requirements. Let's assume you're working on a long-range wireless meter-reading project for a European customer. You have read the corresponding regulatory documents like the REC/ERC/70-03 European recommendation, which provides frequency bands that can be used for short-range, license-free applications on this side of the world. So, you discovered that the 169.4-to-169.475-MHz VHF band could be a great candidate. This VHF band is dedicated to meter-reading and asset-tracking applications and can be used with up to 500 mW of RF power. This is enough for several tens of kilometers of coverage even in dense urban environments.

Of course, all characters appearing in this column are fictitious, any resemblance to real projects, living or dead, is purely coincidental, even if my company has been working on such 169-MHz meter-reading projects for years.

Meter-reading transmitters are almost always designed around low-cost, high-performance integrated transceiver chips from companies such as Analog Devices or Texas Instruments, to name a few, but let's assume that this time you need to design a custom, handmade, transmitter chain.

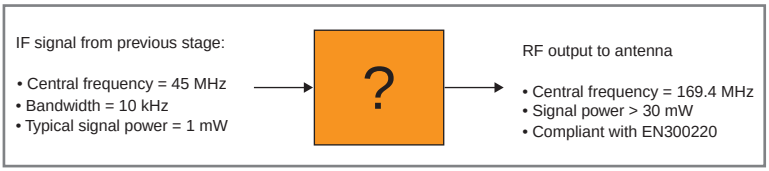


Figure 1—The goal is to design a frequency translation and amplification subcircuit for a VHF transmitter. The input is a 45-MHz intermediate frequency signal with a 1-mW typical power, which must be transformed into a 30-mW, 169.4-MHz signal in compliance with the applicable standards.

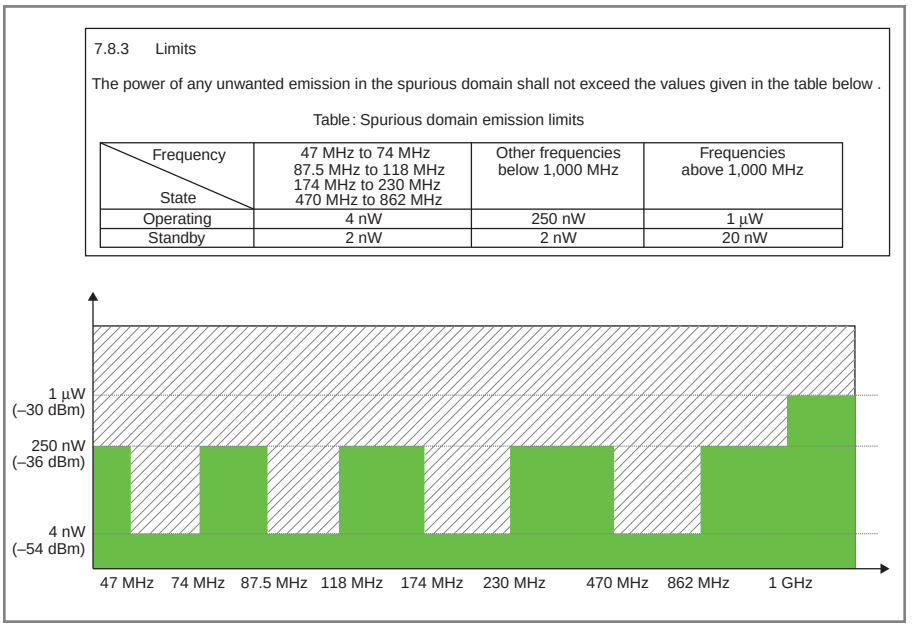


Figure 2—This is an excerpt from the EN 300 220 European standard, applicable to nonspecific RF transmitters below 1 GHz. The table shows the maximum level of acceptable spurious signals, depending on the frequency.

More precisely, and to keep my example as simple as possible, consider that someone else has already designed a great signal generation and modulation circuit. He provides you the signal to be transmitted as a narrow-band 45-MHz intermediate frequency modulated signal, with a 1-mW power. Your job is to design an up-converter and amplifier subcircuit, taking this 45-MHz signal as an input, moving it up to a 169.4-MHz carrier, and amplifying it to at least a 15-dBm power as shown in Figure 1. By the way, remember that a dBm is a power related to 1 mW and expressed in a decibel logarithmic form, so 15 dBm corresponds to 31.6 mW (i.e., 1 mW × $10^{15/10}$).

A last word on the requirements. Your product must be compliant with all the applicable regulations. You've already taken care to select a good frequency band, but CE approval (or FCC in the U.S.) implies other constraints as explained in my article, "CE Marking: A Process to Ensure Product Conformity," (*Circuit Cellar* 257, 2011). In particular, such a transmitter must be designed to comply with the EN300220 standard. Let's just focus on one of its aspects: spurious transmissions. The EN300220 states that a transmitter must not radiate any unwanted signals higher than 250 nW (equivalent to −36 dBm) up to 1 GHz, and not higher than 4 nW (yes, 4 nW, or −54 dBm) in some protected frequency bands like FM audio broadcast. Figure 2 summarizes this requirement. You must take this constraint into consideration for your mixing and amplifying stage.

HIGH-LEVEL DESIGN

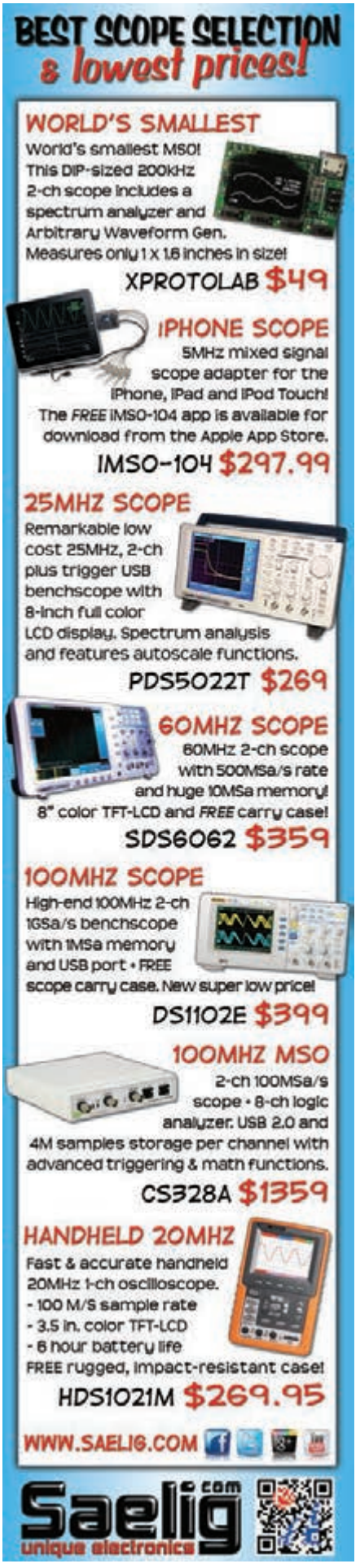
Let's start the design now. You will, of course, use a frequency mixer to move the frequency up from 45 to 169.4 MHz. You can use a local oscillator of 214.4 MHz (i.e., 169.4 + 45) or 124.4 MHz (i.e., 169.4 – 45). In this case, it is nearly equivalent, so just select the former. The mixer will then receive 45 MHz on its IF input and 214.4 MHz on its LO input, and will generate the sum and difference of the two frequencies, specifically, 259.4 MHz (i.e., 214.4 + 45) and 169.4 MHz (i.e., 214.4 – 45).

You will then need to browse catalogs of mixer suppliers to find a good match, meaning a mixer able to work with RF and LO frequencies of 169 MHz to 214 MHz and IF frequency of 45 MHz. As explained in my June 263 article, you must also take care to select a mixer with a LO power level high enough as compared to the input signal power. Here the input is 0 dBm, so as a first guess, let's be optimistic and state that a mixer with a 10-dB higher LO power is a strict minimum, so we will select a 10-dBm class mixer.

There are plenty of mixer manufacturers, but for this article, I used Mini-Circuits's ADE-1LH+ frequency mixer as an example. Table 1 shows its key specifications. The mixer is adequate for signals from 0.5 MHz to 500 MHz and has a reasonable 5-dB conversion loss. The specification also provides isolation figures, which means the level

Frequency (MHz)		**Conversion Loss (dB)**				**LO-RF Isolation (dB)**						**LO-IF Isolation (dB)**						**IP3 at center band (dBm)**
LO/RF	IF		Mid-band m			L	L	M	M	U	U	L	L	M	M	U	U	
$f_L - f_U$		*x	σ	Max.	Total range max.	Typ.	Min.	Typ.	Min.	Typ.	Min.	Typ.	Min.	Typ.	Min.	Typ.	Min.	Typ.
0.5–500	DC–500	5	0.1	6.5	8.2	65	50	55	35	47	26	52	40	45	22	34	20	15
1 dB COMP.: +5 dBm typ.						L = low range [f_L to 10 f_L]					m = mid band [10 f_L to f_U/2]						U = upper range [f_U/2 to f_U]	
						m = mid band [2 f_L to f_U/2]												

Table 1—An excerpt from the Mini-Circuits ADE-1LH+ frequency mixer. Understanding each of these figures is the key to success. (Source: Mini-Circuits)



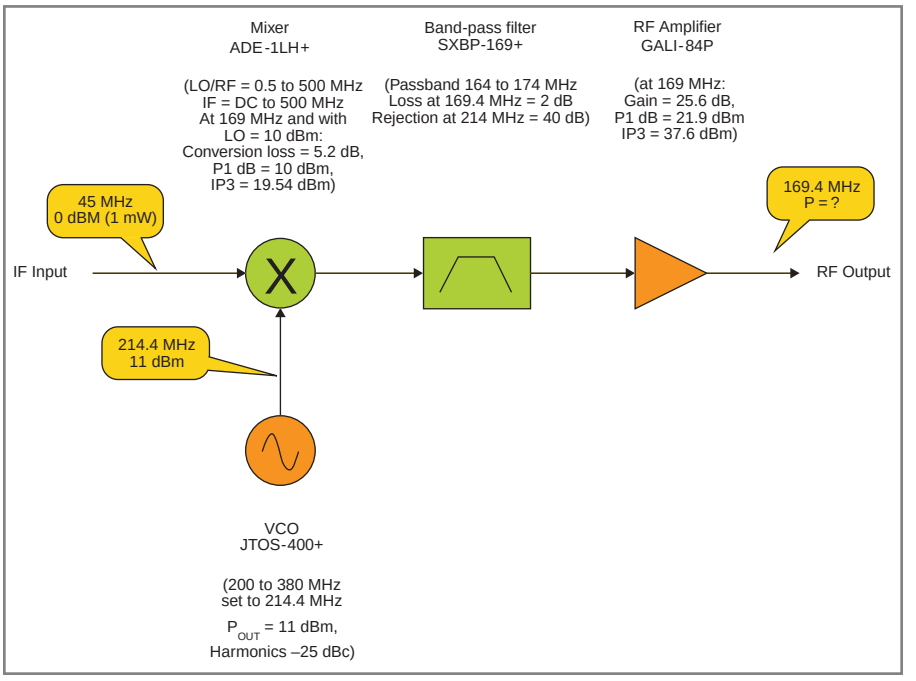


Figure 3—This block diagram shows the proposed architecture for the sub-circuit specified in Figure 1. The IF signal is first translated from 45 MHz to 169.4 MHz thanks to an ADE-1LH+. It is then filtered and finally amplified by a GALI-84+ amplifier, both from Mini-Circuits.

of unwanted signal leakage between the mixer ports. For example, its typical LO to RF isolation at 169 MHz is specified at 55 dB, meaning that the 214.4-MHz, 10-dBm LO signal will be present on the RF output at a level of –45 dBm (i.e., 10 dBm – 55 dB).

The last important parameter is the nonlinearity figure of the mixer, expressed here as IP3, which means "third-order intercept point." I'm afraid I would need another article for that topic, but that IP3 is closely linked to the nonlinear behavior of the mixer. The closer the input power is to the IP3, the higher the spurious signals will be on its output. Here, the IP3 is 15 dBm, which is just a little more than one order of magnitude higher than the 0-dBm input power, so we may anticipate some linearity problems.

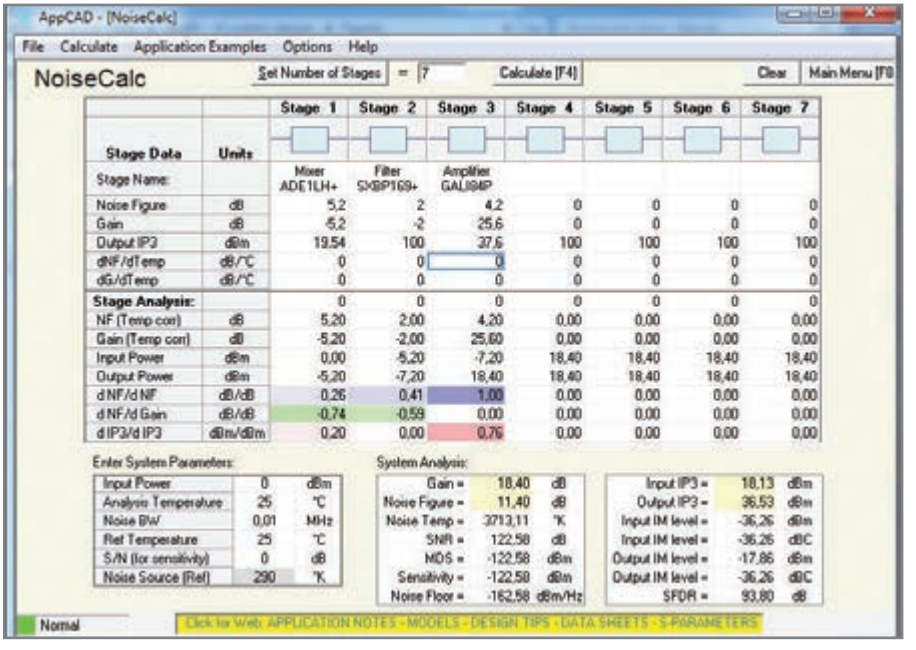


Photo 1—The AppCad software from Agilent is a quick way to calculate key characteristics of an RF chain by entering the key specifications of each stage (e.g., gain, IP3, etc.). Here, with an input power of 0 dBm (1 mW), the calculated output power is 18.4 dBm (70 mW), which is significantly above target.

CIRCUIT CELLAR

ADuC841 Microcontroller Design Manual:

From Microcontroller Theory to Design Projects

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



Now Just $35.00

Buy it today!

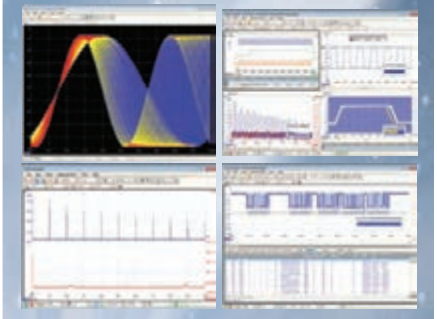
www.cc-webshop.com

pico Technology

GET MORE WITH THE
CLASS LEADING OSCILLOSCOPES
PicoScope 6000 Series
MORE MODELS • MORE CHOICE
MORE FEATURES AS STANDARD



Highest sampling rate and deepest memory of any scope for this budget



	6402A	6402B	6403A	6403B	6404A	6404B
Channels	4					
Bandwidth	250 MHz		350 MHz		500 MHz	
Sampling Rate	5 GS/s					
AWG or Function Generator	FG	AWG	FG	AWG	FG	AWG
Memory	128 MS	256 MS	256 MS	512 MS	512 MS	1 GS
Price $	3291.75	4116.75	4941.75	5766.75	6591.75	7416.75

All models include probes, carrying case, full software and a 5 year warranty. Software includes: measurements, spectrum analyzer, full SDK, advanced triggers, color persistence, serial decoding (CAN, LIN, I2C, SPI), masks, math channels all as standard with free updates.

For more info contact us at:
1-800-591-2796 or
www.picotech.com/pco476

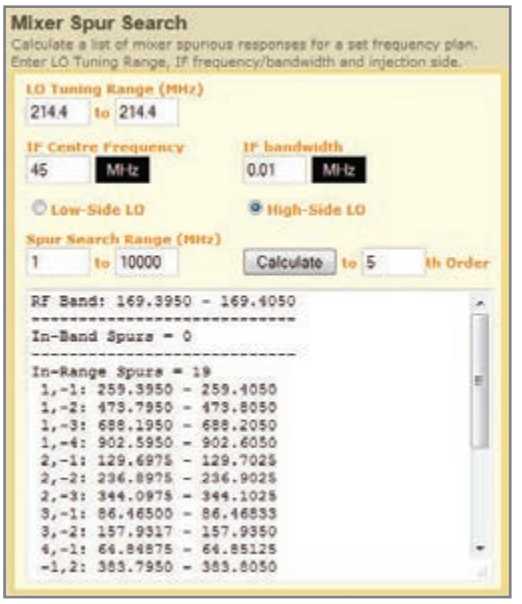


Photo 2—Online tools like a mixer spur chart calculator (www.leleivre.com) help you quickly determine the possible spurious frequencies received or generated by a frequency mixer. However, they cannot evaluate the frequencies' amplitude.

But let's continue.

Following the mixer, we will need a band-pass filter to isolate the desired 169.4 MHz from the other spurious frequencies generated by the mixer. We could design this filter with capacitors and inductors, but as I still had my Mini-Circuits catalog in hand, I found the SXBP-169+ bandpass filter. It has a passband of 164 MHz to 174 MHz with a 2-dB loss at 169 MHz, and rejection better than 40 dB below 122 MHz or higher than 240 MHz.

Lastly, we need to amplify the signal to the required output power. As an example, I used a Mini-Circuits GALI-84+ MMIC amplifier. This chip provides a 25.6-dB typical gain at 169 MHz. Its maximum output power is specified by its so-called P1dB of 21.9 dBm. P1dB is the 1-dB compression point, meaning the output power at which its gain is 1 dB lower than the small-signal gain as a result of compression. As this figure is significantly higher than the required 15 dBm output power, we can assume it will be enough.

Here we are. We have a preliminary design. Figure 3 shows the overall architecture of the RF subcircuit we have drafted.

CHAIN CALCULATION

Now we need to check if the performances of this chain match the requirements. The first step is probably to evaluate if the output power is in line with the target. You could easily do it by hand, just adding the gains or subtracting the losses of each stage in decibels. Let's try:

0 dBm (input power) – 5.2 dB (mixer losses) – 2 dB (filter losses) + 25.6 dB (amplifier gain) = +18.4 dBm

So, we are slightly above the target of 15 dBm, which is good. This calculation is simple, but you can also find free tools that will do this calculation for you and calculate other key characteristics of the chain. One of my favorites is the NoiseCalc module built into Agilent's AppCad software (which is available for free from its website). Just launch it, enter the key characteristics of each stage, and you will get the results as shown in Photo 1. As planned, the output power is calculated as 18.4 dBm, but this tool also provides you a calculated overall noise figure for the chain, here 11.4 dB. (Refer to my previous article, "Noise Figures 101," *Circuit Cellar* 249, 2011.) This tool also calculates the overall IP3 of the chain, enabling it to evaluate the maximum input signal power before nonlinear effects. Lastly, it provides the calculated power between each stage. You can then check if these powers stay significantly below the P1-dB compression point of each device, which is the case in that example.

WHAT ABOUT SPURS?

This quick chain-gain calculation shows that the design seems in line with our goals at least for the desired 169.4-MHz signal, but we have just done a small part of the job. The devil is hidden in the spurs. You need to calculate what would be the other signals generated by the circuit and check if they are safely below the regulatory limits. The main spur generator is the mixer, so you need to start with this one. First, you need to calculate the frequency of each spur that may fall in the transmitter pass band. Once again, you could do it by hand or use one of the mixer spur calculators available online. I used the mixer spur chart calculator at

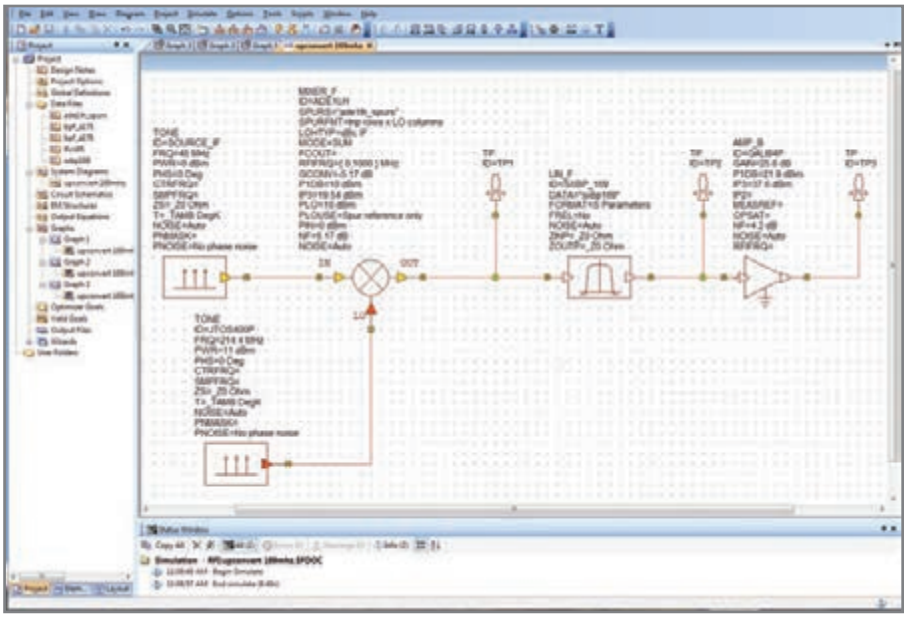


Photo 3—This is a system-level design of our subcircuit designed with AWR's Visual System Simulator (VSS). It is very similar to the block diagram shown on Figure 3, except here each component is associated with its system specifications.

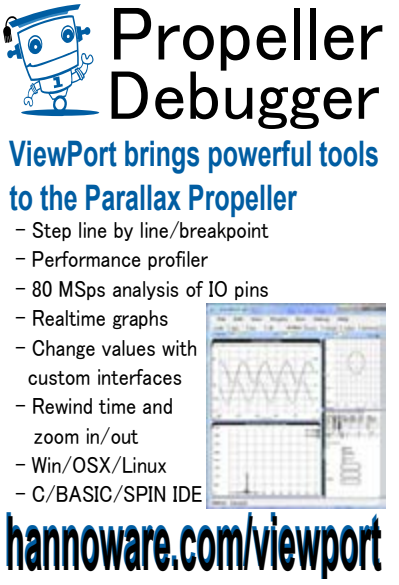
www.leleivre.com. The result is shown in Photo 2. Basically, this tool calculates all possible combinations of both input frequencies, and sorts them by frequency. You must also add to this list of possible spurs the leakage of the input signals (IF to RF and LO to RF leakage) and their harmonics.

You need to calculate the respective amplitude of each spur. This will be a boring step, as you will have to look through the detailed specifications of the mixers and extrapolate them for your application. This will not be easy as these amplitudes are linked to the input signal power and frequency, so an alternative method could be to buy an evaluation kit of the selected mixer and simply measure them with a spectrum analyzer.

With either method, you will know the mixer's most significant spurs. You will then need to simulate how they will be transformed by passing through the remaining parts of the chain. What is the attenuation of the band-pass filter and the gain of the output amplifier at the frequency of each spur? What are the possible new spurious signals generated by these components? This is mandatory in order to end up with a calculation of the output spectrum of the circuit and to be able to compare it with the EN300220 specifications. Unfortunately, these calculations are long and boring, so you will probably end up with a headache, even if a spreadsheet helps. Imagine, for example, that there are two mixers on the chain, which is typical. The number of potential spurious frequencies will be squared. That's why these calculations are usually done only for the couple of key spurious frequencies, with the risk to be unlucky with the first prototype if some other spur happens to be critical.

"Frequency mixers are strange devices that somehow translate a signal up or down in frequency thanks to the nonlinear effects of their internal diodes. More precisely, a mixer has two inputs (one signal input and one so-called 'local oscillator' input) and one output. It generates the sum and difference of the frequencies F1 and F2 applied to its two inputs. On the nastier side, they also generate plenty of spurious signals in the form of n × F1 + m × F2."







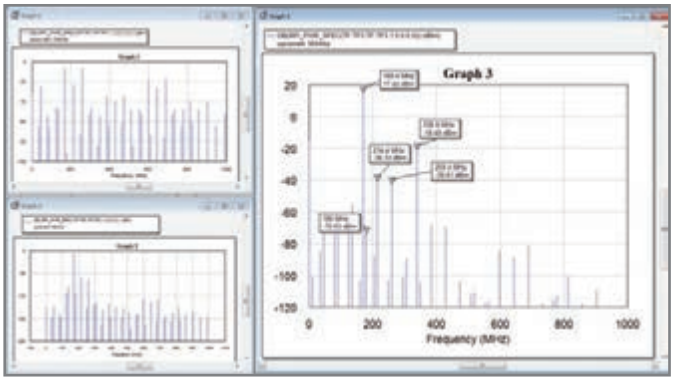


Photo 4—Click "run" and VSS shows the simulated output spectrum of the circuit. The top left plot is the output of the mixer, the bottom left is the output of the filter and output of the amplifier is on the right. The main output signal is as expected at 169.4 MHz with a power close to 18 dBm. Plenty of spurious signals are also expected to be generated, and VSS provides a simulated amplitude for each of them.

You definitively have to check the mixer image frequency (214.4 + 45 MHz) and the second and third harmonics of the output frequencies (2 × 169.4 MHz and 3 × 169.4 MHz).

VSS TO THE RESCUE

I want to present a more efficient solution using system-level RF simulator software. There are very few tools like it on the market, but my company is fortunate enough to have a license for one—AWR's Visual System Simulator (VSS) software. Caution: this is a costly tool.

VSS has a zillion features (e.g., time-domain modulation simulations); however, in this article, I will just demonstrate its most basic feature, which is system-level RF chain simulation. This is exactly what we need. Basically, VSS enables you to design an RF chain, to link "system" blocks such as mixers, filters, or amplifiers. You can define the characteristics of each block either at a behavioral model (i.e., this is an amplifier with a given gain, P1dB and IP3), through data tables from the manufacturer (i.e., this is a filter with these gains and phases through frequency), or even through circuit diagrams or electromagnetic models if you buy the other modules in AWR's Microwave Office tool suite. You can start with a high-level model, simulate it, refine some blocks down to an electromagnetic simulation of a microstrip PCB, and check the performance of the entire system. You can even link a VSS model with actual hardware through off-the-shelf test equipment, or pass a virtual sample of an actual signal (e.g., a CDMA frame) through the chain and check its impact on the modulation of bit error rate.

Let's stay with the basics and focus on our example circuit. Photo 3 shows the design drafted under VSS. Of course, you will need some training to efficiently use such a tool, but magical things happen when you click on the Simulate button (see Photo 4). VSS does exactly what I've explained above, but automatically and in a couple of seconds. Look at the results. The frequency output is as planned at 169.4 MHz, with a power of 17.42 dBm. This is 1 dB lower than we had previously calculated. This difference is due to the spurious frequencies, as a part of the available power is spread through these spurious signals.

However, VSS also provides all the simulated spurs with their respective levels. Remember the EN30220 limits shown in Figure 2? The transmitter must not generate any spurs higher than −36 or −54 dBm, depending on the frequency band. Here, unfortunately, several spurs are higher than these levels. The highest is at 338.8 MHz, with a calculated power of up to −18 dBm. VSS could show you where this frequency comes from with a mouse click, but here it is obvious: 338.8 is two times 169.4 MHz. This is the second harmonic of the carrier frequency. A tool like VSS is also useful to understand what's going on. Refer to the bottom-left plot in Photo 4. This spur is not significantly present at the output of the mixer, so it must be generated by the final GALI-84+ amplifier itself. We are probably too close to its compression point.

Other significant spurs are visible at 214.4 and 259.4 MHz. The former is the leakage of the local oscillator, which is not attenuated enough by the filter as it is quite close to its pass-band. The latter is the second mixing term (214.4 + 45 = 259.4), which is also not attenuated enough.

IMPROVEMENTS

What should you do now? The first action is to check for actual problems. In fact, the regulatory limits are based on the level of radiated spurs by the product, so the antenna is part of the system. As the antenna will be tuned for the desired 169.4-MHz frequency, its efficiency will be lower at the other spurious frequencies, reducing their actual impact. However, in that case, I'm afraid this will not be enough as the gap is significant. You will then have to iterate the design to optimize it.

The easy way would be to add a second band-pass filter at the output of the amplifier to filter out anything unwanted. This solution is usually too costly, but if you are not cost-constrained, this will drastically reduce the spur levels with a limited design effort. Checking the effects of such a design change is where using a simulation tool shines. I did it in 10 s on VSS. The result is shown in Photo 5. As you can see, the output spectrum is now very clean and fully within the specifications.

Another solution would be to try to reduce the power at the mixer's input. This will also significantly reduce the spurious level, but you will then need to add an extra gain stage to recover the required output power, so it may be out of specification with respect to harmonics, but these are quite

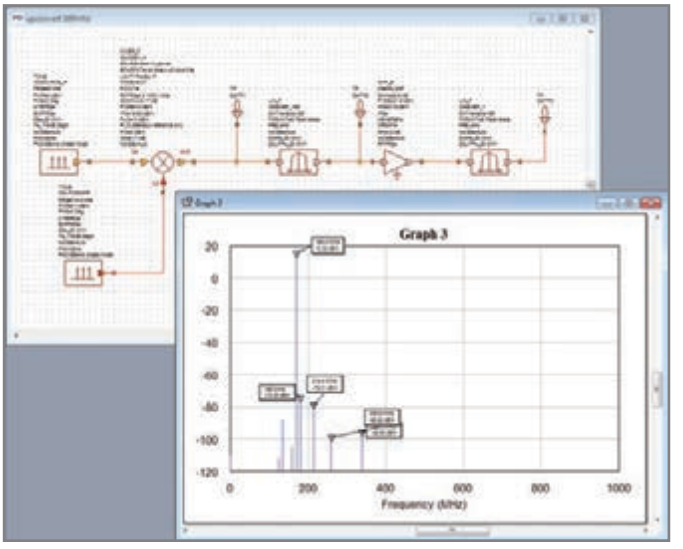


Photo 5—This simulation was done with another filter right after the last amplifier. This solution is costly, but greatly improves the output spectrum at the expense of 2 dB of extra losses. Compare it with Photo 4.

high in frequency and easily filtered with a low-cost passive low-pass filter. Lastly, you could select better parts (i.e., a mixer and an amplifier with higher IP3) and reiterate the design.

WRAPPING UP

My goal was not to provide you with a fully functional design. It was to present the key development steps you could use. Designing an RF system without calculating their key performances implies a risk of redesign close to 100%. You can either accept it, and reduce the redesign cost by using evaluation boards or similar, or minimize the risks by calculating or simulating the design as early as possible. System simulation tools like VSS (or its equivalent from other companies such as Agilent) are invaluable. They enable early simulations, even before the first schematic. Of course, there are risks associated with such complex tools. You may find wrong results due to a mistake in the model of a given part or due to the misuse of the tool itself. So good designers must always check if the results make sense prior to presenting them to their managers, but I'm sure all regular *Circuit Cellar* readers already know that!

Robert Lacoste lives near Paris, France. He has 22 years of experience working on embedded systems, analog designs, and wireless telecommunications. He has won prizes in more than 15 international design contests. In 2003, Robert started a consulting company, ALCIOM, to share his passion for innovative mixed-signal designs. You can reach him at rlacoste@alciom.com. Don't forget to write "Darker Side" in the subject line to bypass his spam filters.

RESOURCES

European Communications Office (ECO), European Community, Frequency Management, Regulatory Affairs and Spectrum Engineering Working Groups, "ERC recommendation 70-03 Relating to the Use of Short-Range Devices (SRD)," 2011, www.erodocdb.dk/docs/doc98/official/pdf/rec7003e.pdf.

European Telecommunications Standards Institute (ETSI), EN 300 220-1, "Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Short-Range Devices (SRD); Radio Equipment to be Used in the 25-to-1,000-MHz Frequency Range with Power Levels Ranging up to 500 mW; Part 1: Technical Characteristics and Test Methods," 2012, www.etsi.org/deliver/etsi_en/300200_300299/30022001/02.04.01_40/en_30022001v020401o.pdf.

———, EN 300 220-2, "Candidate Harmonized European Standard (Telecommunications Series)," www.etsi.org/deliver/etsi_en/300200_300299/30022002/02.01.02_60/en_30022002v020102p.pdf.

Le Leivre.com, Mixer Spur Search. http://leleivre.com/rf_mixerspur.html.

R. Lacoste, "CE Marking: A Process to Ensure Product Conformity," *Circuit Cellar* 257, 2011.

———, "Noise Figures 101," *Circuit Cellar* 249, 2011.

———, "Radio Frequency Mixers," *Circuit Cellar* 263, 2012.

SOURCES

AppCad simulator for Windows
Agilent Technologies, Inc. | www.agilent.com

Visual System Simulator (VSS) and Microwave Office RF/microwave design software
AWR Corp. | http://web.awrcorp.com

ADE-1LH+ frequency mixer, GALI-84+ MMIC amplifier, and SXBP-169+ band-pass filter
Mini-Circuits | www.minicircuits.com

HUMANDATA®

FPGA /CPLD Boards from JAPAN

SAVING COST=TIME with readily available FPGA boards

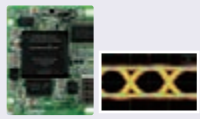
- Basic and simple features, single power supply operation
- Same board size and connector layout - ACM/XCM
- All stocked items are ready to be shipped immediately
- Over 100 varieties of FPGA/CPLD boards are available
- Free download technical documents before purchasing

ALTERA FPGA Board

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



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

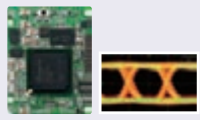


XILINX FPGA Board

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



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



PLCC68 series

XP68-03 Spartan-6 PLCC68 FPGA Module



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

AP68-04 Cyclone III PLCC68 FPGA Module



Cyclone III | PLCC 68
EP3C25U256C8N
16Mbit Configuration Device
Two User LEDs
One User Switch(Slide)
RoHS compliant

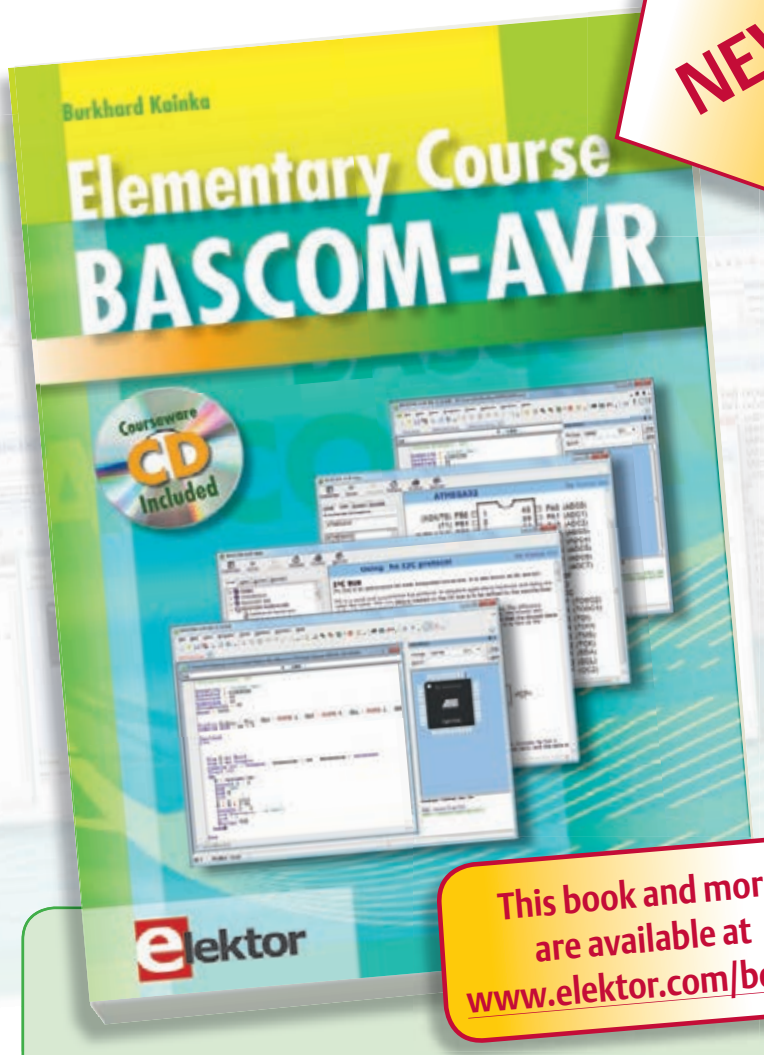
www.hdl.co.jp/CC/

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

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

Elektor Shop

The world of electronics at your fingertips!



NEW!

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

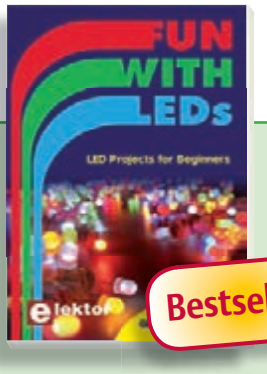
Books

Free Software CD-ROM included

Elementary Course BASCOM-AVR

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

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



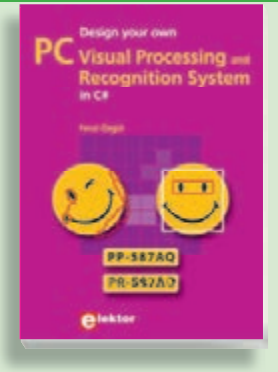
Bestseller!

Associated 60-piece Starter Kit available

Fun with LEDs

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

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



A comprehensive and practical how-to guide

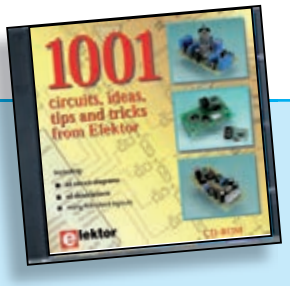
Design your own PC Visual Pro cessing and Recognition System in C#

This book is aimed at Engineers, Scientists and enthusiasts with developed programming skills or with a strong interest in image processing technology on a PC. Written using Microsoft C# and utilizing object-oriented practices, this book is a comprehensive and practical how-to guide. The key focus is on modern image processing techniques with useful and practical application examples to produce high-quality image processing software. All code examples used are available – free of charge – from the Elektor website.

307 pages • ISBN 978-1-907920-09-7 • $57.30

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

CD/DVD-ROMs



Circuits, ideas, tips and tricks from Elektor

CD 1001 Circuits

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

ISBN 978-1-907920-06-6 • $55.70



A whole year of Elektor magazine onto a single disk

DVD Elektor 2011

The year volume DVD/CD-ROMs are among the most popular items in Elektor's product range. This DVD-ROM contains all editorial articles published in Volume 2011 of the English, American, Spanish, Dutch, French and German editions of Elektor. Using the supplied Adobe Reader program, articles are presented in the same layout as originally found in the magazine. An extensive search machine is available to locate keywords in any article. With this DVD you can also produce hard copy of PCB layouts at printer resolution, adapt PCB layouts using your favourite graphics program, zoom in/out on selected PCB areas and export circuit diagrams and illustrations to other programs.

ISBN 978-90-5381-276-1 • $37.90



Embedded Linux Made Easy

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

Populated and tested Elektor Linux Board

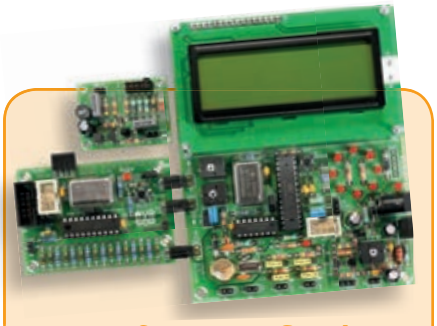
Art.# 120026-91 • $93.30

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

www.elektor.com/shop

elektor

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



AVR Software Defined Radio

This package consists of the three boards associated with the AVR Software Defined Radio articles series in Elektor, which is built around practical experiments. The first board, which includes an ATtiny2313, a 20 MHz oscillator and an R-2R DAC, will be used to make a signal generator. The second board will fish signals out of the ether. It contains all the hardware needed to make a digital software-defined radio (SDR), with an RS-232 interface, an LCD panel, and a 20 MHz VCXO (voltage-controlled crystal oscillator), which can be locked to a reference signal. The third board provides an active ferrite antenna. This bundle also includes the assembled and tested FT232R USB/Serial Bridge/BOB PCB!

Signal Generator + Universal Receiver + Active Antenna: PCBs and all components + USB-FT232R breakout-board

Art.# 100182-72 • $133.00



AndroPod

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

Andropod module with RS485 Extension

Art.# 110405-91 • $74.00

Kits & Modules

by Jeff Bachiochi (USA)



Charging with PV Cells

Energy harvesting is becoming more and more ubiquitous, but batteries—including rechargeable batteries—still play an important role in our day-to-day lives. This article discusses using photovoltaic (PV) cells, which are electrical devices that convert light's energy into electricity, as a means of keeping batteries charged.

Batteries were precious to me as a child. I was fortunate to grow up with the release of the first transistor radio. Prior to this, radios (and other equipment like the first computers) used vacuum tubes. Portable radios required very large batteries to power tube filament, plate, and grid voltages. Not only was the transistor tiny compared to the vacuum tube, but so was its battery requirement, which meant much lighter and smaller radios. This boom heralded the age of the consumer toy. Batteries were required to operate just about everything. "Batteries not included" was the common phrase of the day.

Alkaline batteries remain a popular consumable even today, although we now have rechargeable batteries. We've seen a progression from lead acid to NiCd to NiMH to Lithium-based cells (see Figure 1). Battery technology is high on the industry's priority list as electric and hybrid vehicles are becoming commonplace. However, there is not much difference between having to replace alkaline batteries or having to change rechargeable ones. When the device is out of reach—as some remote sensing devices can be—this can be a bother or next to impossible. Maybe that's why energy harvesting is growing in popularity. It would be great if we could pluck energy right out of the air, as needed. Today, we can get enough energy from things like sun, wind, and water. Unfortunately, it is only available sporadically and not 24/7. We must rely on a storage device to gather the harvested energy when its available to be used as needed. Assuming you can harvest as much as you use, you only need to store enough harvested energy to get you through the next energy drought. Let's take a look at using photovoltaic (PV) cells as a source for keeping batteries charged.

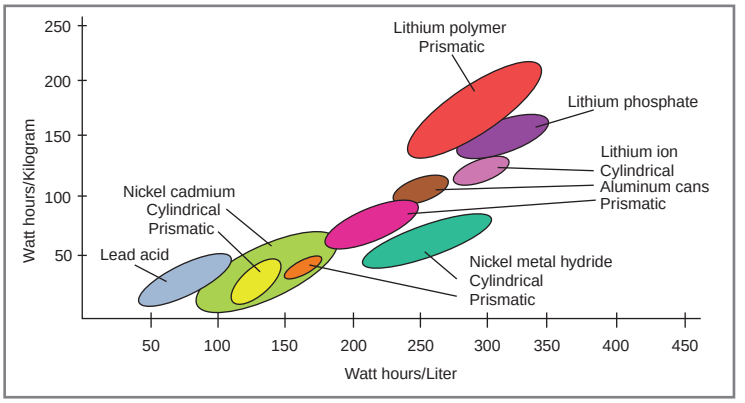


Figure 1—This chart shows the historical progression of the rechargeable battery, including improvements to both weight and increased energy density. (Source: Wikipedia.org)

PV CELLS DEFINED

A single PV cell is a translucent sandwich of P-type and N-type material forming a huge diode junction that can be exposed to a light source, such as the sun. Photons with energy greater than the band gap of the semiconductor create electron hole pairs in the intrinsic region of the diode. Instead of recombining immediately, the electrons and holes are

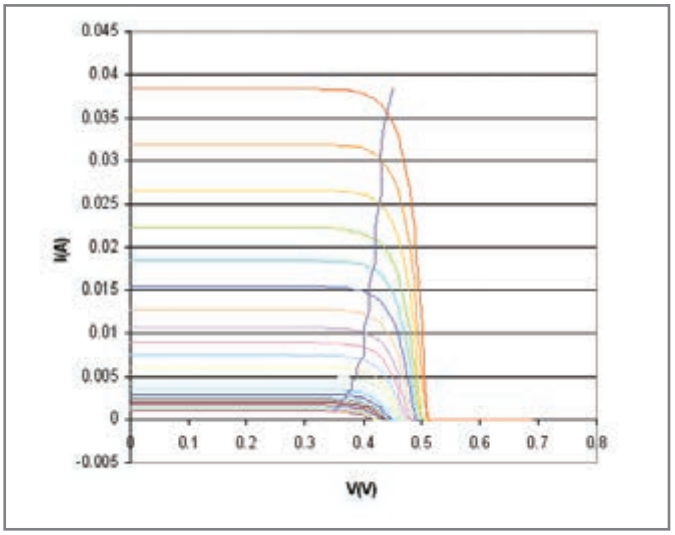


Figure 2—Optimum sunlight does not always fall on a solar cell. This chart shows the I versus V output for various amounts of light falling on a solar cell due to physical orientation or atmospheric conditions. (Source: Wikipedia.org)

swept to opposite sides of the intrinsic region by its inherent electric field. This extra charge can be drawn off by an external circuit, creating a current flow. A PV cell's diode junction has two properties associated with it. It has an "open circuit" voltage, which is the maximum potential it can produce when no current is flowing, approximately 0.5 V. Once current is flowing across the junction, this potential is reduced. As expected, the more current flowing across the junction, the lower the potential becomes. The cell can produce the most current when the potential is 0 V (short circuit).

This can be better visualized by graphing I versus V (see Figure 2). This graph shows a number of curves for varying levels of sunlight. The more intense the light, the higher the current and voltages become. Note the maximum voltage potentials are within the normal range for a PN junction. Also note the relationship is highly nonlinear. A quick power calculation of the two points just discussed results in zero for each. Maximum voltage × zero current = zero power. Maximum current × zero voltage = zero power. Makes sense, right?

This means that all power output falls somewhere between these two points. The power output can be calculated at every point on each curve in Figure 2. On any particular curve, maximum power occurs on the knee of the curve, as the fairly constant I value begins to decrease. Since the voltage goes up as the current goes down (and vice versa), the point for maximum power is a happy medium where the two values are both less than maximum. Note the almost-vertical line drawn through the knees of each curve. This is each curve's maximum power point.

The relationship between I, V, and maximum power point (MPP) varies with the amount of light falling on a particular cell. In past articles, I've discussed how the orientation of the panel to the light source (sun) will greatly affect the panel's output. Maximum output can only be obtained when the cell is perpendicular to the irradiance source. Any deviance off axis will degrade the effective maximum conversion, which, for most of today's cells, is still less than 20%. This discussion does not include cell positioning, but merely how to make the best use of whatever irradiance is available to the cell.

Other factors (beside irradiance) can also affect each cell's MPP, the manufacturing process control, and the temperature under use. While the manufacturing can affect the overall efficiencies (even cell to cell), the working temperature has the most impact. For a given irradiance as the temperature increases, the I increases and the V decreases. With the MPP constantly varying due to all these factors, how can you ensure you're operating at the MPP at any time?

SIMPLE OBSERVATIONS

A cell output voltage around 0.5 V is not very useful, so PV cells are often strung together into modules by adding cells in series to create higher voltages and in parallel to create higher currents. The least expensive way to trickle charge a battery may be grouping a number of cells together such that their output is larger than the battery voltage that needs to charge. A series resistor could be sized to limit the maximum trickle current. Depending on the number of cells in the module, this would only charge the battery when the module's output was greater than the battery voltage. This is not efficient by any means, but there is essentially zero cost beyond the price of the cells. However, because the cells costs are relatively high, using them efficiently might actually be less expensive in the long run.

Knowing where you are is always a good place to start. You must be able to measure the I and V of the solar module. To measure the module's voltage, you might employ a voltage divider across its output to measure a fraction of the output using an analog-to-digital converter (ADC). Module current can be measured by inserting a low resistance in series with the module's output. The voltage drop across this current-sensing resistor is proportional to the current. Two analog/digital channels can be used to obtain conversion values for I and V and calculate a P value. This tells you where you are.

One of the disadvantages of the inexpensive approach is that when the module's output is not optimum, it may be

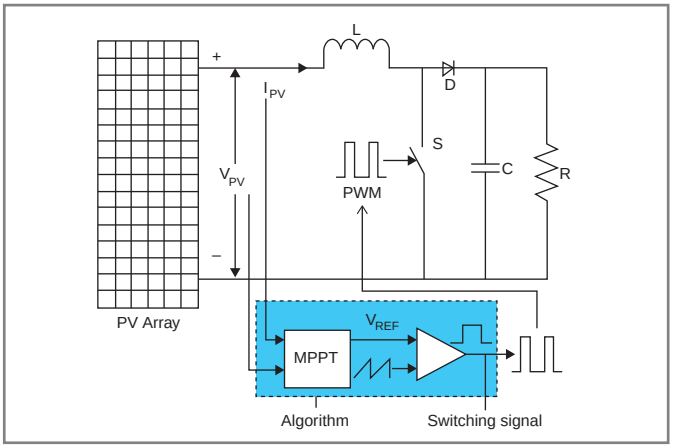


Figure 3—This simple diagram shows how the PWM output of a circuit can control the active component (switch) in a boost circuit. It effectively changes the load on the PV array and its operating point. The controlling algorithm adjusts the PWM duty cycle based on the calculated power output from the measured V and I of the PV array. (Source: Tung, Y., Hu, A., and Nair, N., "Evaluation of Microcontroller-Based Maximum Power Point Tracking Methods Using dSPACE Platform," 2006)

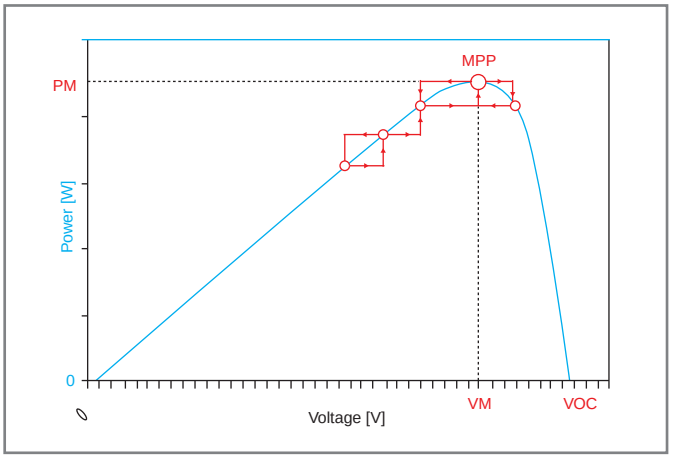
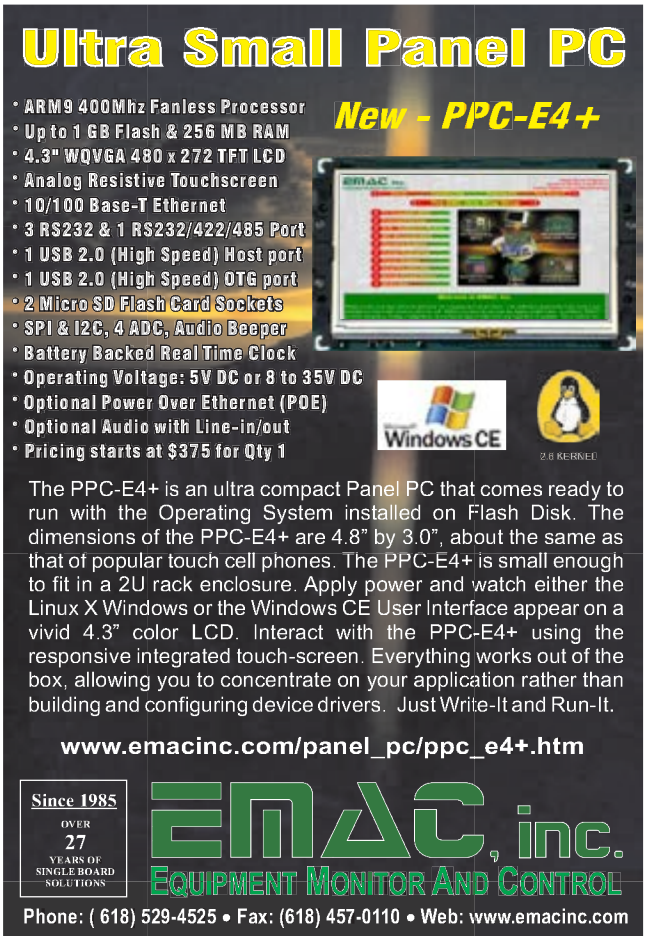


Figure 4—This power curve for a PV cell/panel shows how the perturb and observe algorithm compares calculated power for a particular PWM duty cycle with the calculated power after the duty cycle has been changed. The duty cycle is increased or decreased as needed to continually try to maximize the power. (Source: STMicroelectronics, "SPV1040: High-Efficiency Solar Battery Charger with Embedded MPPT," 18080, 2011)

below a voltage that cannot add any charge to the battery. Wouldn't it be nice if you could take advantage of any available output? Using a boost circuit (i.e., step-up regulators) can solve two problems with one solution. Boost circuits can control the voltage supplied to a load, which means that as long as the load is well-behaved (i.e., resistive or only mildly nonlinear), they can indirectly control the current as well. You've already seen that a PV cell's current output has a direct relationship to its output voltage. If you can control the I, you can control the V. So, let's take a closer look at boost circuitry.



ENERGY BOOST

Boost converters use the magnetics of an inductor to store energy that can be released into a load. Figure 3 shows an inductor (L) in series with a diode (D) between the PV module and a load (R). Assuming the PV module is in sunlight, in this steady state, the load is setting the PV module's current (I_{PV}) and the voltage (V_{PV}). If a switch is introduced between the inductor and the diode, the inductor can be placed directly across the PV module by turning on the switch. Current flows through the inductor and it looks like a heavy load. Energy is stored in the inductor in the form of magnetic energy. During this time, the diode is reverse bias so current can't flow out of the load, including a large capacitor (C) momentarily supplying the load's requirement. When the switch is opened, the inductor can release its magnetic energy as current, which adds to the PV module's current and is directed through the diode and into the load.

Clearly, to have any control over the current, the switch must be repeatedly opened and closed. The relationship of the inductor's charge time (switch on) to its discharge time (switch off) is the controlling factor here. This duty cycle determines the PV module's current and, ultimately, the voltage. A box labelled MPPT is located at the bottom of Figure 3. This box has I_{PV} and V_{PV} as inputs and a PWM signal as an output. The PWM signal controls the switch closures and its duty cycle is controlled by some magic algorithm (MPP) based on the I_{PV} and V_{PV} inputs.

There are a number of approaches to determining maximum power point tracking (MPPT). The algorithm most often used is the "perturb and observe" method. Take a voltage and current measurement to calculate the power. Next increase the PWM (perturb) and make another calculation (observe). If the power has increased, increase the PWM. If the power has decreased, decrease the PWM. The PWM is continually adjusted to obtain the perfect sweet spot or MPP (see Figure 4).

SPV1040

I used a couple of leaded parts from STMicroelectronics: an SPV1020 boost converter and an SPV1040 solar battery charger. As you might suspect from the package, the SPV1020 is capable of higher power (up to 40 V and 320 W) while the smaller SPV1040 handles up to 5.5 V and a few watts. As you can see in Figure 5, I used the SPV1040 because I was interested only in 3 to 5 V, which was enough for a few NiCd or NiMH cells or a single LiPo cell.

Many manufacturers include a good description of the MPP process and boost circuitry in their parts' datasheets. Formulas for determining parts' values are usually included as well as typical circuits that can be used, many times without having to do any calculations. I've discussed the major advantage of using a device with an MPP algorithm as a front end to energy collection using a solar cell(s), but this only gets us halfway to our goal of keeping batteries charged. The next step is to make good use of the collected energy without destroying the batteries you wish to keep in tip-top shape. If your energy supply is connected to the batteries, they may overcharge without any active control. If you

use a simple (and safe) trickle charge, you may not be able to bring a depleted set of batteries back up to snuff if there is a string of overcast days. A bit of smarts must be employed to take full advantage of any available energy.

Part of my goal was to come up with a design that didn't require a separate microcontroller to handle the task. While the individual devices may employ internal controllers to perform their intended functions, I wanted this initial circuitry to be void of programming. Again, I wanted to find a standalone part I could hand-solder and use to charge all battery chemistries. I came close with a few parts, but they couldn't be hand-soldered, handled only one battery type, or required a microcontroller for control. So, I limited my search to only NiCd and NiMH (as that was my initial goal). A Texas Instruments BQ2002 controller seemed to fit the requirement.

BQ2002

The BQ2002 is a charge-management controller that adjusts an external device used to supply charging current. When designed to supply maximum current, the BQ2002 modulates the external FET using PWM, which regulates the charging current applied to batteries. While a supply of energy exists (the SPV1040 is collecting energy), the BQ2002 adjusts its duty cycle to one of three rates: fast charge, top-off, or minimum charge. The TM input enables the designer to select one of three preconfigured charging schemes designated to tailor the maximum fast charging time, and subsequent duty cycle ratios for top-off and trickle-charging periods.

A charging cycle begins upon application of V_{CC} greater than 4 V or, if already operational, the voltage at BAT is less than 2 V. This begins with a fast charge until one of the following termination conditions: peak voltage detection, negative delta voltage, maximum voltage, maximum temperature, or maximum time. Peak voltage detection and negative delta voltage detection have to do with a special characteristic of this battery chemistry where, as the batteries reach optimum charge, their voltage actually goes down a bit. Maximum voltage is at the BAT input whose level is set by a voltage divider across the cells being charged. Maximum temperature is an optional voltage input where the voltage sensed is a resistor divider consisting of a pull-up resistor and thermister (negative temperature coefficient resistor in contact with the batteries). Maximum time is an internal timer that limits the duration of the fast charge period.

If the maximum voltage is reached prior to the maximum time, charging can continue at the optional top-off (reduced current) rate for the remaining time period. At this point, the charging is further reduced to the minimum (trickle) charge rate until V_{CC} is removed. (Or in our case, the energy supply is exhausted.)

The open-drain output (CC) remains a high impedance to request charging current and is pulled to ground to inhibit charging. I wanted to use a simple LM317 to control the charging current, but it device requires a minimum of 4-V overhead (V_{IN} to V_{OUT}) to operate, so I had to use a less elegant solution with a minimum drop. A MOSFET switch controls current flow. A series resistor is used to limit the



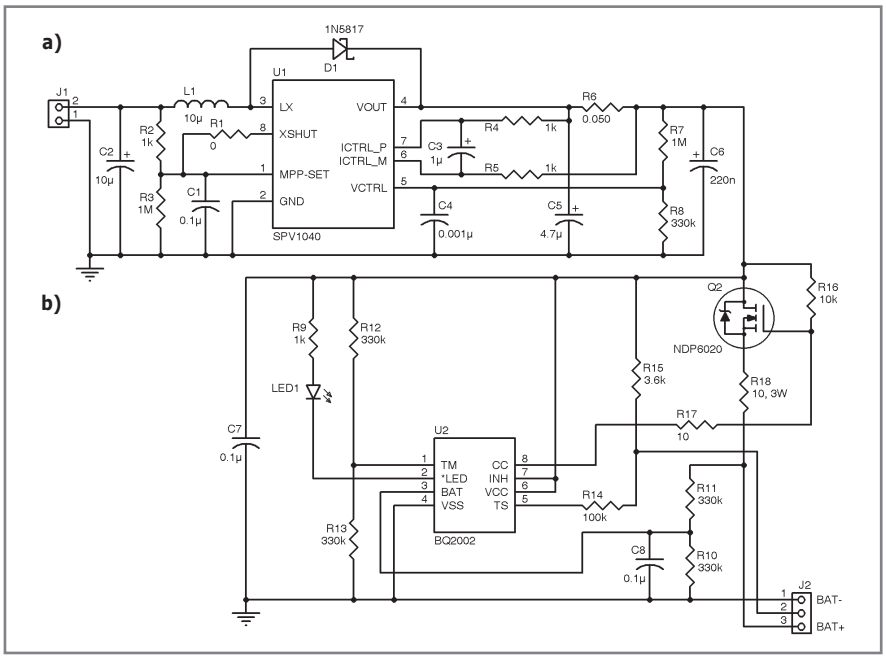


Figure 5a—The SPV1040 circuitry is used to gather energy from the solar cell using the MPP algorithm. **b—**This circuitry was implemented to provide fast charging capabilities for depleted batteries using a large solar cell. This was later determined unnecessary. A much smaller solar cell can be used if this requirement is relaxed, meaning precharged batteries would be needed.

short circuit current. Limiting short-circuit current to 1 A meant I needed to use an approximately 5-Ω, 5-W resistor. I decided to limit this to 0.5 A. This also meant that the charging current would reduce to approximately 250 mA as the cells' capacity is reached. Note that there is an inverter in the logic between the CC output and the external FET. I used an NPN transistor biased on to ensure that the FET is on while the BQ2002 is high impedance (not inhibiting charging current).

OVERKILL

Up to this point, I assumed the remote device was populated with dead batteries. I think this was a bit pessimistic. Having to initially charge batteries in any reasonable amount of time (say, approximately 2 h) requires a rather large solar cell, similar to Parallax's 750-00042 2.65-W solar cell (see Photo 1). This is really overkill if you look at the true requirement. For these batteries to last more than a year, some quick calculations can be made to show the per-hour usage is extremely small:

$$\text{Battery capacity} \sim \frac{2\text{ A}}{\text{year}}$$

$$\frac{2\text{ A}}{365\text{ days}} = \frac{5.5\text{ mA}}{\text{day}}$$

$$\frac{5.5\text{ mA}}{24\text{ hours}} = \frac{230\ \mu\text{A}}{\text{hour}} \text{ (required current)}$$

$$230\ \mu\text{A} \times 2.5\text{ V} = 575\ \mu\text{W}$$

Using a 2.65-W solar cell is a few thousand times overkill! If that assumption is changed to using precharged batteries, things take a big turn. Fast-charging current is no longer needed.

What do manufacturers say about trickle charging for extended periods? Recommended rates suggest not applying more than 1/40 C. (C is the 1-h rating of the particular cell.) This means a 2-Ah cell should have a trickle charge of no more than 50 mA (i.e., 50 mA × 2.5 V = 128 mW). Wait, that's still a few hundred times the operating current!

The shelf life of NiMH batteries wasn't even a year when they were first introduced. Improvements over the last few years have brought them up to acceptable levels. Today, these cells lose only about 20% capacity just lying around. So, if a set of batteries loses 20% of its 2-Ah capacity in a year, that's 400 mAh/year or roughly 1-mAh of capacity day! So, for a device that uses 5.5 mAh/day, you can expect to lose an additional 1 mAh/day from self-discharge.

We experience an average of about 12 h of sunlight each day. Assuming you mount our solar cell facing south (in the northern hemisphere) at an azimuth of your site's latitude is, at noon (on the equinox) your solar panel will be perpendicular to the sun. You can expect to gather the maximum amount of energy. Cells are rated assuming that 1,000 W/m^2 of illumination is falling on them. Figure 6 shows the typical solar insolation levels expected on a clear day in Winona, MN. At this location, the solar cell received about half the rated illumination (475 W/m^2). So, even under perfect conditions, the cell will only output half of its rated output. Also, since the sun is only perpendicular at noon, most of its potential for gathering energy is less in the morning and afternoon. The sine shape of the graph suggests that you might

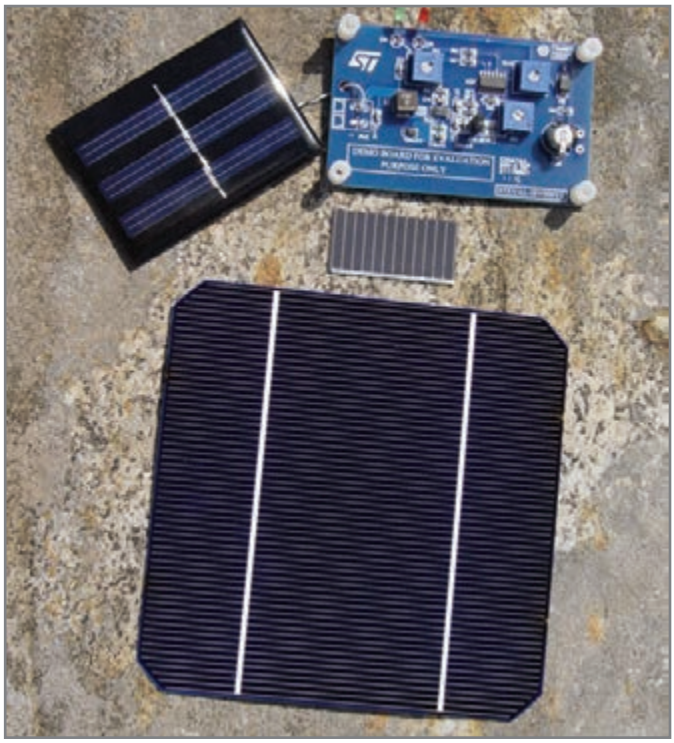


Photo 1—A great way to investigate MPPT for your design is to use an STMicroelectronics evaluation board, such as this STEVAL-ISV006V2 shown in the top of the photo. The smaller cell in the center is rated at 165 mW (0.55-V output at 0.3 A) measuring 1.5" × 0.75". At the bottom is a Parallax commercial-quality solar cell that is rated at 2.65 W (0.534-V output at 5.34 A) measuring 125 mm.

2013 International CES
THE GLOBAL STAGE FOR INNOVATION

PRODUCED BY CEA

Be the first to see what tomorrow will be talking about.



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



REGISTER NOW

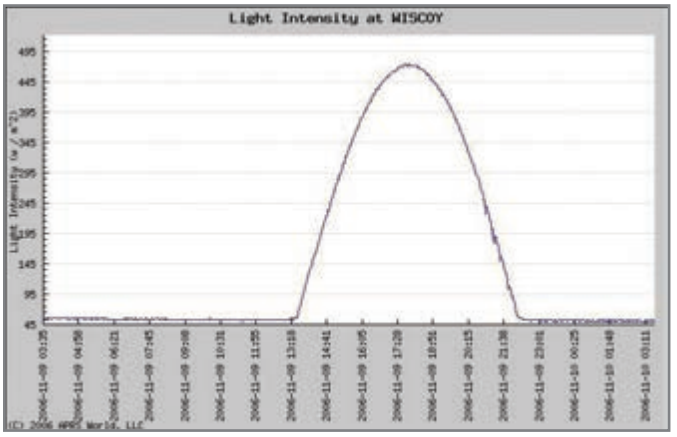


Figure 6—Solar insolation is the measurement of the sun's light intensity falling upon the earth. As expected, this is less than the 1,000W/m^2 used to specify the potential output of a solar cell/panel. It will vary by latitude as well as by time of year. This takes weather conditions into account. (Source: APRS World)

expect to get 0.707 of the maximum on average. So, over 10 h, that's approximately 3,300 Wh/m^2 (i.e., 10 h × 475 Wh/m^2 × 0.707) of illumination for the day. If you use Parallax's 2.65-W cell, you might see 3.3 h of capacity (i.e., 3,300 Wh/m^2/1,000 W/m^2) or 8.7 Wh (i.e., 2.65 W × 3.3 h) of energy over a one-day period. And that's under a clear sky!

So where does this get us? At my local electronics store, I found a small 1.5" by 0.75" solar cell that is rated at 300 mA at 0.55 V. That has a potential of 165 mW (i.e., 0.3 mA × 0.55 V). On clear days, I might expect to gather 544 mW (i.e., 165 mW × 3.3 h) of energy, which would be approximately 10 times the energy required for my remote device. Even with poor efficiency, once my load capacitor (supercapacitor) was charged up to 5 V, I could draw off current to trickle charge the batteries with a simple resistor. With a potential of 5 V and load of 2.5 V (battery voltage), a 250-Ω resistor would supply 10 mA for as long as I could keep the supercapacitor charged. Even an hour of sunlight a day could produce twice the daily requirements.

NATURE

Obviously, a good look at your locality's solar data is a must in order to be able to predict how insulation levels will affect the current-generating potential for a specific installation. While batteries may be sized to last a year, you may wish to extend their effectiveness by adding some support to keep them charged to extend their maximum lifetime. STMicroelectronics offers a couple of evaluation boards you can use to get started with a design (refer to Photo 1).

Energy harvesting from not just solar, but wind, water, thermal, and other methods is being researched today. Nature provides a plethora of possibilities. As stewards of our natural resources, we need to use them wisely, lest we tip the natural balance too far. There are some things we just can't live without.

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

RESOURCES

APRS World, "Solar Insolation (Light Intensity)," www.aprsworld.com/sensors/solarInsolation.

STMicroelectronics, "SPV1040: High-Efficiency Solar Battery Charger with Embedded MPPT," 2011, www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00287506.pdf.

Tung, Y., Hu, A., and Nair, N., "Evaluation of Microcontroller-Based Maximum Power Point Tracking Methods Using dSPACE Platform," Australian University Power Engineering Conference, 2006, http://itee.uq.edu.au/~aupec/aupec06/htdocs/content/pdf/165.pdf.

SOURCES

750-00042 2.65-W Solar cell
Parallax, Inc. | www.parallax.com

SPV1020 Boost converter and SPV1040 Solar battery charger
STMicroelectronics | www.st.com

BQ2002 NiCd/NiMH Fast-charge management ICs
Texas Instruments, Inc. | www.ti.com

NEED-TO-KNOW INFO

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

Solar-Powering the Circuit Cellar (Part 1)
Preparing the Site
by Steve Ciarcia
***Circuit Cellar* 209, 2007**
Have you considered installing a photovoltaic (PV) system? This series of articles describes the entire process, from site preparation to completion. Topics: Photovoltaic, Solar

Solar-Powering the Circuit Cellar (Part 2)
From the Ground Up
by Steve Ciarcia
***Circuit Cellar* 210, 2008**
This article covers the massive task of preparing two 11' schedule 80 poles (8" in diameter) to support two of the three solar arrays. Topics: Photovoltaic, Solar Arrays

Solar-Powering the Circuit Cellar (Part 3)
Wiring & Electronics
by Steve Ciarcia
***Circuit Cellar* 211, 2008**
Installing a major renewable energy source like a home photovoltaic system is no small feat. This article describes wiring and electronics. Topics: PV, Renewable Energy

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

Elektor Print

Confident reading: on paper



+

Elektor Digital

Contemporary & innovative reading: on PC, notebook or tablet

=

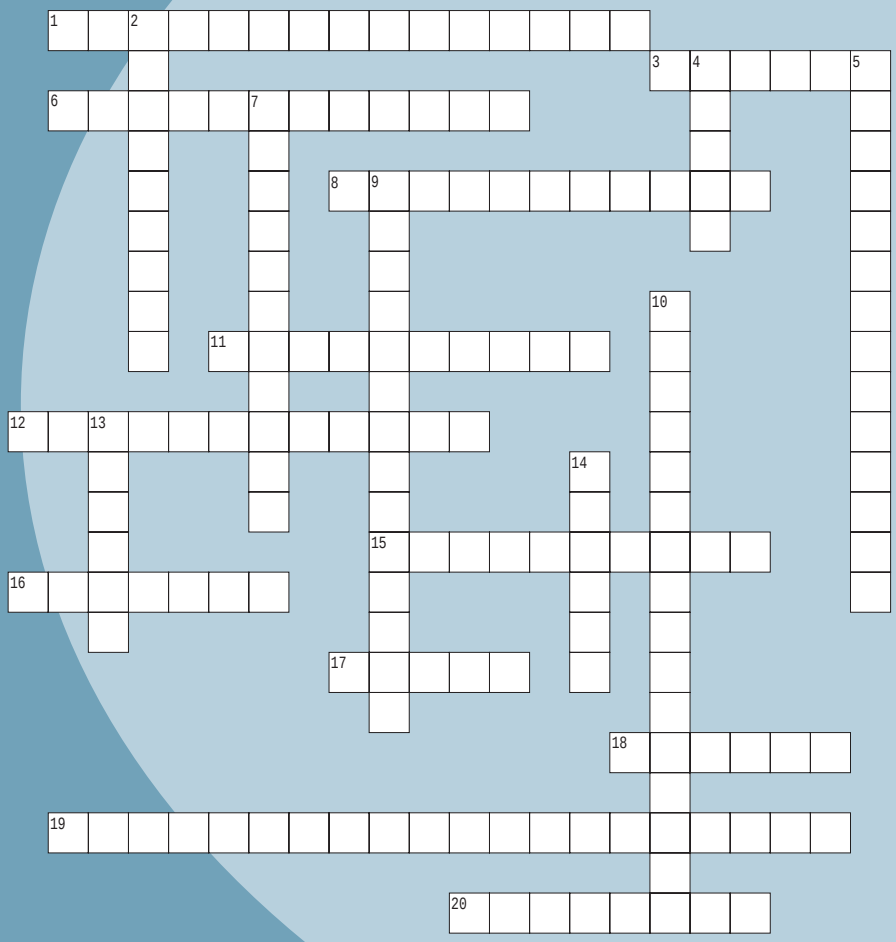
Elektor PLUS

Comprehensive reading: at home and on the road

Read Elektor with the cut-rate PLUS membership!

Join now or upgrade: www.elektor.com/member

CROSSWORD



Down

2. Software used to create other software, usually including a text editor, a compiler, a linker, and a debugger
4. A unit of inductance; abbreviation "H"
5. An amount of oscillation ranging from 3 kHz to 300 GHz [two words]
7. Miguel Sanchez's article, "Image Processing Development" (*Circuit Cellar* 263, 2012), used one of these along with an MCU and bipolar stepper motors to accomplish some computer vision-related tasks [two words]
9. In 1978, this company released one of its first products—a simple, MOS Technology 6502 microprocessor-based, single-board computer [two words]
10. According to Jeff Bachiochi's article in this issue, a single one of these is "a translucent sandwich of P-type and N-type material forming a huge diode junction that can be exposed to a light source" [two words]
13. Used as a volume control in audio amplifiers [two words]
14. SI unit of electric current

Across

1. The frequency at which a filter's output has fallen by 3 dB from the maximum level [two words]
3. Israeli cryptographer and one of the inventors of the RSA algorithm
6. V/n = k [two words]
8. Columnist Robert Lacoste's article, "Noise Filters 101" (*Circuit Cellar* 249, 2012), discussed how to determine and measure it in a radio frequency filter [two words]
11. ECMAScript-approved language that supports a lot of C-structured programming syntax
12. Lightening or a tripped circuit wire [two words]
15. Name of the radio show co-hosted by two recent *Circuit Cellar* Q&A interviewees [three words]
16. When a software thread relinquishes control of the processor to the operating system
17. American physicist (1923–2005) who worked with Robert Noyce to create the first integrated circuit
18. The interval between one musical pitch and another with half or double its frequency
19. A capacitive sensor that changes its polarization in response to a change in temperature [two words]
20. In a mechanical electrical meter, a pulse that changes state every half rotation of the meter's disk and represents a quanta of energy [two words]

The answers will be available in the next issue.

IDEA BOX

THE DIRECTORY OF PRODUCTS AND SERVICES

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

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

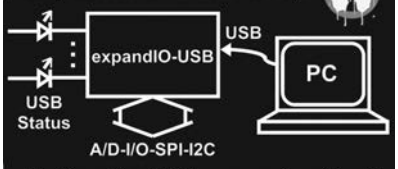
HUSB™
OEM $79
- Add a high speed USB port to a TERN controller.
- High performance USB stack chip (FT232H, FTDI)
- Ready to use, royalty free USB drivers
- USB 1.1 and USB2.0 compatible
- Data transfer rate to 8 MB/sec with D2xx driver
- 2.1"x1.3"



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

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

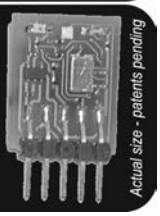
Revolutionary new expandIO-USB chip



- Ideal for adding USB to sensors & peripherals
- No drivers needed for Windows, Mac, Linux
- No microcontroller programming required
- Also check out our USB-232 USB to UART

www.hexwax.com - Buy from Mouser & Farnell

Amazing PIC programmer
Most devices supported
ICSP, SQTP, & copy limits
$32 at Digikey & Mouser
www.flexipanel.com

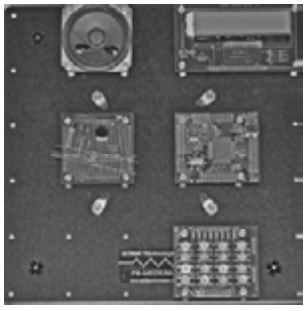


Disk On Chip
Ready for Delivery



From 16MB to 128MB Available!
Call 530-297-6073 Sales@jkmicro.com
www.jkmicro.com
jk microsystems, Inc.
International Orders Welcome

MICROCONTROLLER KITS

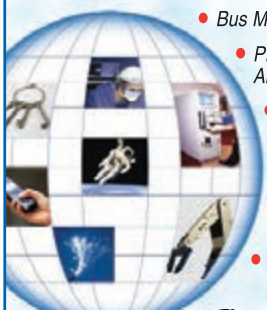


LEARN
CREATE
EXPLORE

BASIC ON BOARD

ATRIA Technologies Inc.
www.AtriaTechnologies.com

I²C™/SMBus



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

MCC Micro Computer Control
www.mcc-us.com



Add a Touch Screen to Your Embedded Product

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

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

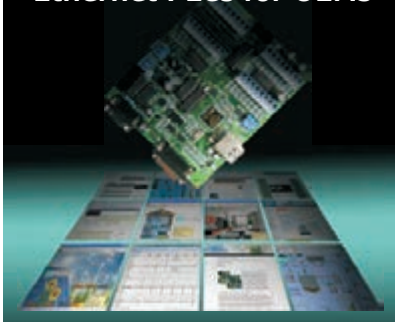


Development Kits include serial LCD controller board, display, touch screen, cables, sample images/code, power supply, technical support, 100% Satisfaction Guarantee.

REACH TECHNOLOGY INC.

LISTEN TO YOUR MACHINES

Ethernet PLCs for OEMs



FMD88-10 and FMD1616-10

Integrated Features :

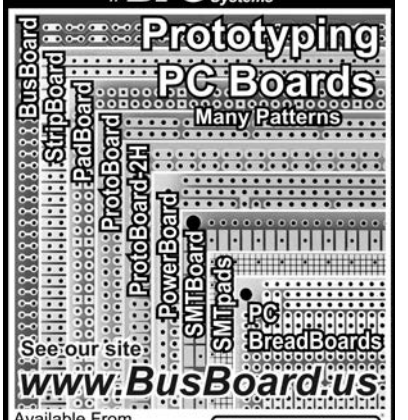
- ETHERNET / Modbus TCP/IP
- 16 or 32 digital I/Os
- 10 analog I/Os
- RS232 and RS485
- LCD Display Port
- I/O Expansion Port
- Ladder + BASIC Programming

$229 and $295
before OEM Qty Discount

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

TRI TRIANGLE RESEARCH INTERNATIONAL

BPS BusBoard Prototype Systems



Available From
JAMECO ELECTRONICS
MOUSER ELECTRONICS a ttl company
amazon.com amazon.co.uk

IC Device & Package Converters

Quick-Turn Custom Solutions

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

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



RoHS

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

Running out of time?



Let our team finish your custom project!

Engineers with over 20 years of software embedded experience

- Specializing in embedded designs include PCB design and prototype assemblies
- Staffed Software and Electrical engineers, full testing lab and technicians
- Quick code conversions and code completion services

Fixed Price Quotes & Contracts

www.ccsinfo.com/custom
sales@ccsinfo.com
262-522-6500 ext. 35

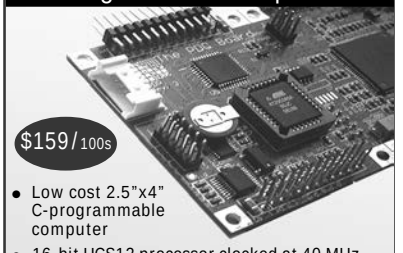
CCS

ALL ELECTRONICS CORPORATION

Electronic and Electro-mechanical Devices, Parts and Supplies. Wall Transformers, Alarms, Fuses, Relays, Opto Electronics, Knobs, Video Accessories, Sirens, Solder Accessories, Motors, Heat Sinks, Terminal Strips, L.E.D.S., Displays, Fans, Solar Cells, Buzzers, Batteries, Magnets, Cameras, Panel Meters, Switches, Speakers, Peltier Devices, and much more....

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

PDQ Board™ - A Fast I/O-Rich Single Board Computer

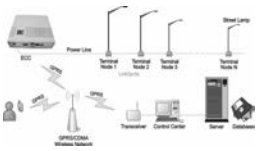


$159/100s

- Low cost 2.5"x4" C-programmable computer
- 16-bit HCS12 processor clocked at 40 MHz
- 8 PWM, 8 counter/timer, and 8 digital I/O
- 16 10-bit A/D inputs
- Dual RS232/485 ports, SPI and I^2C ports
- 512K on-chip Flash, 512K RAM with Flash backup
- Plug-in I/O expansion, including Ethernet, Wi-Fi, GPS, 24-bit data acquisition, UART, USB, Compact Flash card, relays, and more ...

Mosaic Industries Inc.
tel: 510-790-1255 fax: 510-790-0925
www.mosaic-industries.com

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







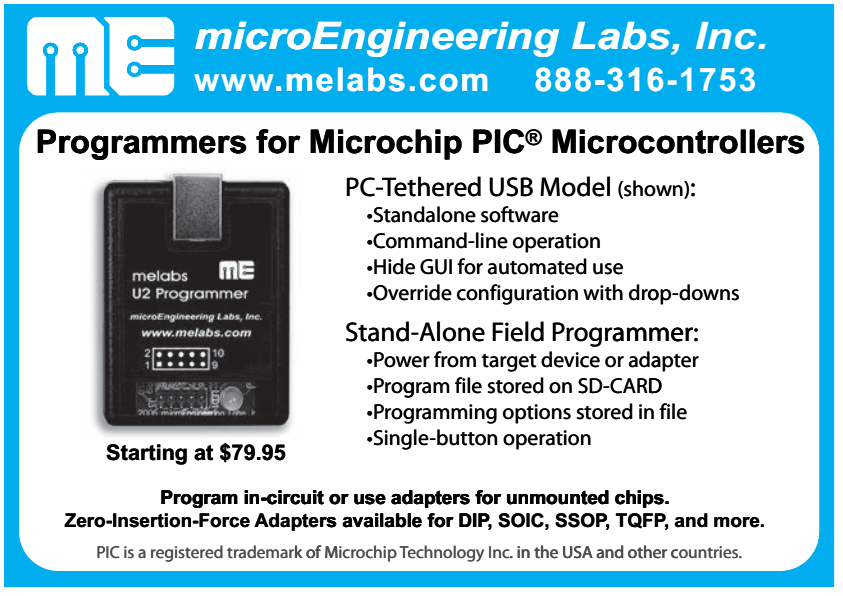
- Power-Line Communication Street Lighting System
- WiFi Street Lighting System
- Power-Line Communication Modules: Spyder and McLaren
- Arduino with WiFi: Diamondback, Redback
- Arduino Shields: Cupperhead and Juniper WiFi, PLC, GPRS, MIDI, Touch, gameduino
- Long Range UHF Gen2 RFID
- JPEG Serial Cameras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

www.linksprite.com
www.linkspritedirect.com











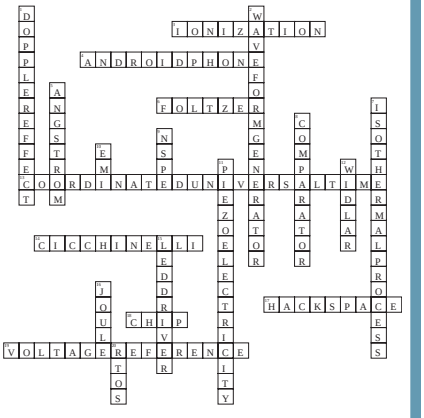
CROSSWORD ANSWERS from Issue 264

Across

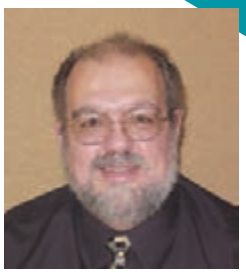
3. **IONIZATION**—Occurs when an atom or molecule gains either and positive or negative charge
4. **ANDROIDPHONE**—Matt Oppenheim's article, "Audio-Enhanced Touch Sensors" (Circuit Cellar, May 2012), said one of the stumbling blocks of using this for data collection is that it will try to recharge itself whenever you connect it to a USB port [two words]
6. **FOLTZER**—Circuit Cellar interviewee who participated in Motorola's IEEE-802 MAC subcommittee on token-passing access control methods [two words]
13. **COORDINATEDUNIVERSALTIME**—A method of keeping the world in sync [three words]
14. **CICCHINELLI**—Circuit Cellar published his book about a commonly used computer programming language in 2010
17. **HACKSPACE**—A synonym for "circuit cellar"
18. **CHIP**—Si plate
19. **VOLTAGEREFERENCE**—National Semiconductor's LM385 series is an example of an adjustable one. [three words]

Down

1. **DOPPLEREFFECT**—A phenomenon that occurs when a vehicle sounding a siren approaches, passes, and recedes from an observer [two words]
2. **WAVEFORMGENERATOR**—A device that produces electronic signals [two words]
5. **ANGSTROM**—Equals 1/10,000,000,000 m
7. **ISOTHERMALPROCESS**—$\Delta T = 0$ [two words]
8. **COMPARATOR**—A device that compares two voltages or currents and switches its output to indicate which is larger
9. **NSPE**—Organization formed in 1934 by bridge engineer David Steinman
10. **EMI**—Common cause of electronic data corruption and subject of Novacek's 257, 2011 *Circuit Cellar* article; acronym [two words]
11. **PIEZOELECTRICITY**—Occurs when crystals acquire a charge after being compressed, twisted, or distorted (e.g., quartz)
12. **WIDLAR**—American electrical engineer (1937–1991) and IC pioneer
15. **LEDDRIVER**—Circuitry that regulates or provides powers to a light source [two words]
16. **JOULE**—Symbolized by 10th letter of the alphabet
20. **RTOS**—Unscramble the following: IETEORGSEPSMNYMRLTIAEAT



PRIORITY INTERRUPT



by Steve Ciarcia, Founder and Editorial Director

What Were They Thinking?

I've mentioned this before, but the original BMW iDrive on my 2002 745iL was a prime example of overzealous design. Back then, somebody had the bright idea of condensing nearly all the switches and knobs formerly found on typical car dashboards down to a large knob, called iDrive, on the center console. Combined with a fancy graphics LCD, the joystick provided the driver with 3-D motion control for selecting specific subsystems and individual functions within that subsystem. The bad news was that zooming into and backing out of various control functions was so complicated it was a real driving hazard.

I'm guessing the iDrive designers got caught up in the process of creating a slick design and completely forgot about the basic reason you're in the car in the first place—to drive, not to run a computer! Let's face it, when you're driving, it's more expedient to reach for a single-function control you can locate out of the corner of your eye. The world was not ready to deal with a 3-D joystick, a complex decision tree on an LCD, and a dozen hand motions to tell a computer to accomplish the same thing. With the original iDrive, you either left the same Sirius station on forever with the hot air blowing in your face or learned to use the voice-response system to at least jump over some of the "slick" functionality and get directly to the heater or radio LCD screens.

It took until I got my 2010 X5 for them to get it right. (My three BMWs in between had iterative improvements.) The solution? They put most of the buttons back! Yeah, like most other cars these days, it still has a joystick knob and an LCD that controls individual settings, but a "real" button takes you directly to the right subsystem.

Circuit Cellar Project Editor Dave Tweed related another example to me while I was talking to him about this editorial idea. He told me about the Kawasaki intelligent proximity activation start system (KIPASS), which is an ignition system for some of Kawasaki's high-end motorcycles that's based on a "proximity fob" (RFID). If the physical key is in the ignition and you bring the fob within a few feet of the bike, you can start it (by turning and pushing the key). Sounds nice, right?

You can leave the key in the ignition all the time. When you walk away with the fob in your pocket, the key is captured by an electric latch so it can't be stolen. All's well and good. But it seems that most riders are in the habit of stopping the engine by using the "kill switch" instead of turning off the key. That's fine until you realize that the headlight is only controlled by the ignition key—the light stays on until you turn the key to the "off" position. On a normal bike, you would do this anyway in order to pocket the key before walking away. But with the KIPASS fob, you don't need the key and lots of bikers simply stop the engine and walk away—leaving the headlight on and killing the battery.

Here's where it gets fun. With a dead battery, you can't start the bike except by jumping it using another vehicle. The Allen wrench you need to open the panel you have to remove to get at the battery is in the tool kit under the seat, which is locked by the ignition key. In a real Catch-22, the ignition key is still "captured" by KIPASS even without power, and you can't remove it to unlock the seat! The only solution is to find a friend or a tow truck with both jumper cables and the correct Allen wrench.

Home theater equipment is another irritation. Forgiving for now that most functions can be accessed *only* from the remote control, you'd think the few buttons they do have would work correctly. For some reason, I'm finding the built-in buttons are a lot less responsive than the remote. I suspect the interrupt handling software for the IR receiver has a much higher priority than the hardwired switches. My Blu-ray player, in particular, virtually never responds to the physical change disc button on the front, and I almost always have to use the button on the remote. Even powering up the unit takes two to three firm panel button presses but only one click on the remote. My HDTV is similar. There's a manual button to change the input selection, and it always takes two to three presses before it actually starts to switch inputs. On the IR remote, it is instant. Grrrr.

The point is, when designing a new piece of equipment, don't get caught up demonstrating as much technology as possible to the exclusion of all other considerations. How hard can it be to design a piece of equipment to respond equally to a few hardwired buttons as well as the IR? Similarly, just because you have the computing power and software expertise, sometimes it's counterproductive to put all functional control only in the remote control or to put every function into one multi-option/multitasking joystick. As a designer, you have a responsibility to reduce hardware costs by eliminating excess manual controls, but you should always take the time to put yourself in the place of the end user who has to deal with your choices. More importantly, think about what happens after the dog eats the remote.

steve.ciarcia@circuitcellar.com

www.ftdichip.com

FTDI Chip

ENHANCED USB PERFORMANCE

Streamlined USB Bridge Solutions

X-CHIP

EXtensive Interfaces
UART, FIFO, SPI, I²C, FT1248

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

EXceptional Drivers
Windows, MacOS, Android, and Linux




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