

CIRCUIT CELLAR

THE MAGAZINE FOR COMPUTER APPLICATIONS

#227 June 2009

COMMUNICATIONS

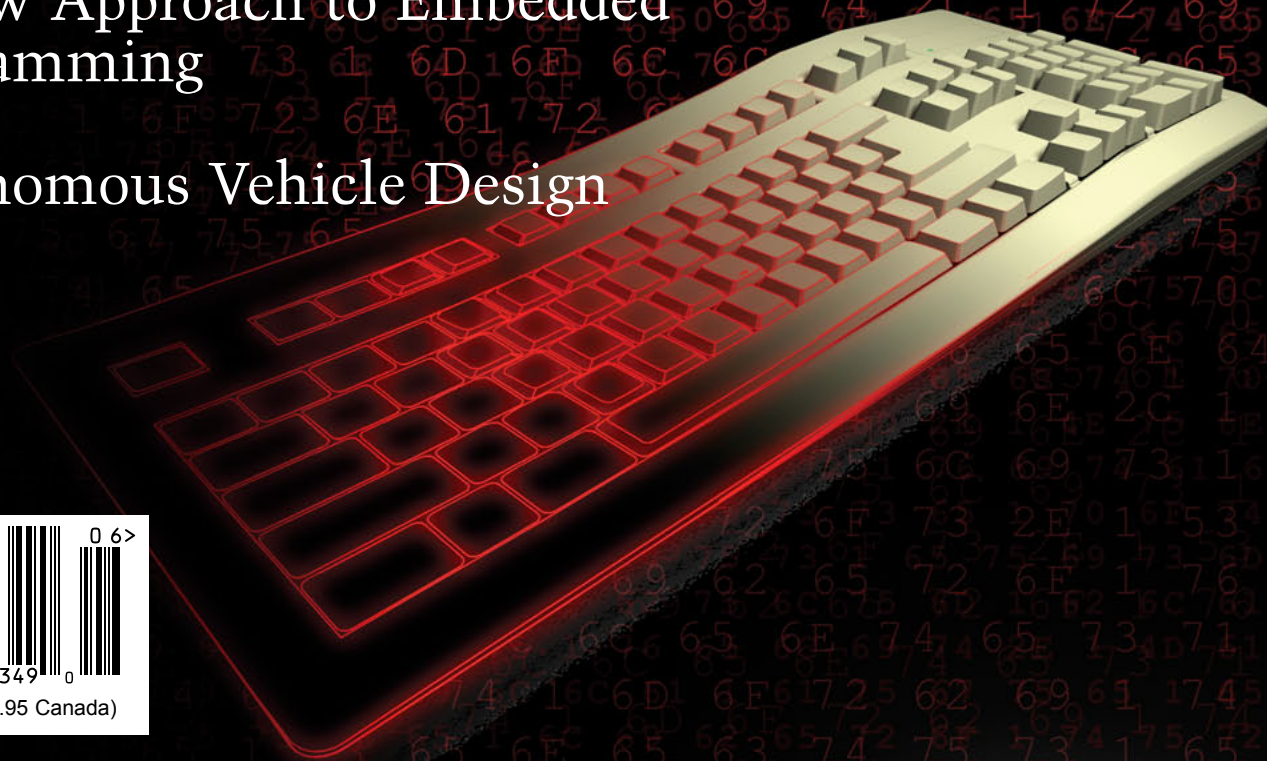
Virtual Keyboard Communication
Made Simple

Understanding High-Speed Signal
Transmission

Anisotropic Magnetoresistance
Explained

A New Approach to Embedded
Programming

Autonomous Vehicle Design



SECURE SERIAL-TO-ETHERNET SOLUTION



Low-cost



32-bit Performance



SSH/SSL Secured

Need a custom solution?

Customize with the NetBurner SB70LC Development Kit for only \$99.

Customize any aspect of operation including web pages, data filtering, or custom network applications.

Kit includes: platform hardware, ANSI C/C++ compiler, TCP/IP stack, web server, e-mail protocols, RTOS, flash file system, Eclipse IDE, debugger, cables and power supply

Kit enables communication 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

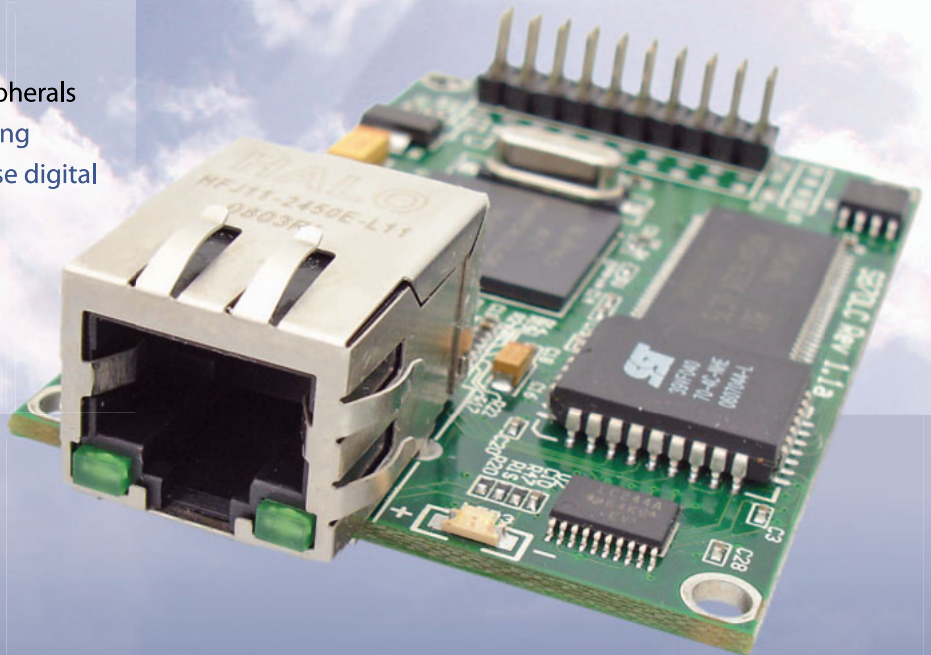
The complete, **secure** hardware and software solution

- Simple Ethernet connectivity for serial devices
- Works out of the box - no programming is required
- Enable data encryption to prevent unauthorized monitoring
- Customize to suit any application with development kit

Features:

- 10/100 Mbps Ethernet
- SSH/SSL/TCP/UDP modes
- DHCP/ Static IP support
- Web-based configuration
- Two TTL serial ports

SB70_{LC}
\$49⁰⁰ Qty. 100



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



5 Competitive Advantages

Overseas Manufacturing

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

CAM USA

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

Quick-Turn Production

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

Overseas Manufacturing

Shipping Logistics

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

Significant Price Saving

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

Quick-Turn Production

Door to Door Delivery

Significant Price Saving

CAM USA



For details please visit our website or call us

Capabilities

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

ITAR, ISO 9001 : 2008

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



Imagineering Inc.

847-806-0003

www.PCBnet.com

email: sales@PCBnet.com

24 YEARS IN BUSINESS...AND STILL GOING STRONG

Featured Products and PC/104 Peripherals

NEW!



\$449
qty 1

TS-TPC-7390

200MHz ARM9 Touch Panel Computer

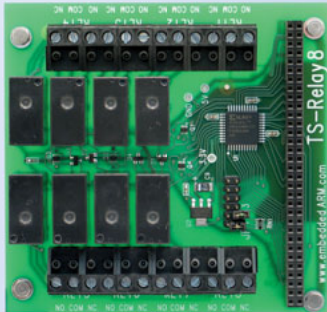
- Low Power, Industrial Quality Design
- Mountable aluminum frame
- 64MB SDRAM (128MB opt)
- 512MB Flash w/ Debian Linux
- Programmable FPGA- 5K LUT
- 7" Color TFT-LCD Touch-Screen
- 800x480 customizable video core
- Dedicated framebuffer- 8MB RAM
- Audio codec with speaker
- Boots Linux 2.6 in about 1 second
- Unbrickable, boots from SD or NAND
- Runs X Windows GUI applications
- Runs Eclipse IDE out-of-the-box

NEW!

TS-RELAY8

Eight Software Controlled Relays

- 8 SPDT relays
- Up to 277 VAC @ 5A
- Up to 30 VDC @ 5A
- Software controlled
- 40mA draw per coil
- I/O jumpers



\$89
qty 1

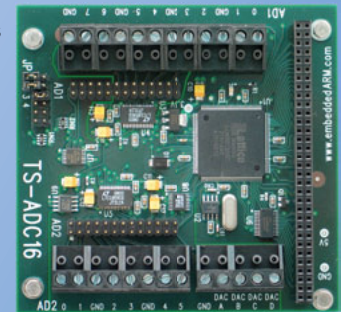
NEW!

TS-ADC16

ADC, DAC and Digital I/O

- 16 16-bit ADCs
- Up to 100Ksps (10us)
- 1KB ADC RAM-FIFO
- 4 ADC voltage ranges
- Prog. pacing clock
- Externally triggered
- 4 inputs, 1 output
- 4 12-bit DACs
- 4 16 bit counters

\$169
qty 1



see our website for x86 SBCs, peripherals and option details



We use our stuff.

Visit our TS-7800 powered website at
www.embeddedARM.com

Forward-Thinking Designers

I've said here before that we think of each project described in these pages as a key contribution to 21st-century technological development. Each design is the result of thousands of previous projects, and in turn each will surely lead to exciting new advances in the embedded design industry and beyond.

This month, some of the projects we present represent exciting new contributions to some of the most important scientific topics of our time. From Cambridge to Silicon Valley to Tokyo, engineers and scientists are devoting their work lives to the important topics presented in this issue: virtual reality designs, autonomous vehicle construction, wireless navigation, precision sensor technologies, high-speed signal transmission, solar power acquisition, and more. Our authors are truly forward-thinking designers!

In "Keystroke Communication," Naweed Paya and Venkat Ganesh describe how they designed a customizable virtual keyboard around a microcontroller and some affordable parts (p. 14). This flat keyboard design is the size of a piece of standard paper, which makes it perfect for use on virtually any flat surface.

Starting on page 22, Ed Nisley presents the second part of his "Solar Data Logger" series. With a redesigned layout, better power supplies, and a superior inductor, Ed's data logger project is back on track. Here he presents the new circuitry and covers the topics of inductor core saturation, solar panel output, and more.

Do you need an SSI controller to retrieve sensor readings? Before you shell out a lot of money to buy one, try your hand at Robert Bowen's project on page 30. Now you can build your own SSI readout display for position sensors.

On page 42, a team of engineers from Camosun College describes an autonomous vehicle design. They cover every aspect of the project, from the embedded system to motor control and sensor technology. This project can serve as a prototype for future military, industrial, or healthcare autonomous applications.

Data transmission speed is an important topic to everyone from engineers working on industrial applications to editors trying to move megabytes of photos to coworkers. Speed is king. In "High-Speed Signal Transmission," Robert Lacoste covers signal transmission theory (p. 52). He introduces the important topic of transmission on low-bandwidth cables, and in doing so, he covers the topics of preemphasis and equalization.

GPS technology isn't always a good fit for an embedded application that requires navigation capabilities. Check out Jeff Bachiochi's article "Location Notification" for a description of an accessible alternative to GPS technology (p. 60). He introduces the topic of magnetoresistance and explains how to use an anisotropic magnetoresistance (AMR) sensor in a design.

Having trouble with the software side of an embedded design project? Tom Cantrell presents a new chip and a web-based technique for software development (p. 68). It's an alternative approach to embedded programming that's sure to make your future design endeavors a lot easier.

cj@circuitcellar.com



CIRCUIT CELLAR®

THE MAGAZINE FOR COMPUTER APPLICATIONS

FOUNDER/EDITORIAL DIRECTOR

Steve Ciarcia

CHIEF FINANCIAL OFFICER

Jeannette Ciarcia

MANAGING EDITOR

C. J. Abate

MEDIA CONSULTANT

Dan Rodrigues

WEST COAST EDITOR

Tom Cantrell

CUSTOMER SERVICE

Debbie Lavoie

CONTRIBUTING EDITORS

Jeff Bachiochi

Ingo Cyliax

Robert Lacoste

George Martin

Ed Nisley

CONTROLLER

Jeff Yanco

ART DIRECTOR

KC Prescott

NEW PRODUCTS EDITOR

John Gorsky

GRAPHIC DESIGNERS

Grace Chen

Carey Penney

PROJECT EDITORS

Gary Bodley

Ken Davidson

David Tweed

STAFF ENGINEER

John Gorsky

ADVERTISING

860.875.2199 • Fax: 860.871.0411 • www.circuitcellar.com/advertise

PUBLISHER

Sean Donnelly

Direct: 860.872.3064, Cell: 860.930.4326, E-mail: sean@circuitcellar.com

ADVERTISING REPRESENTATIVE

Shannon Barraclough

Direct: 860.872.3064, E-mail: shannon@circuitcellar.com

ADVERTISING COORDINATOR

Valerie Luster

E-mail: val.luster@circuitcellar.com

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

PRINTED IN THE UNITED STATES

CONTACTS

SUBSCRIPTIONS

Information: www.circuitcellar.com/subscribe, E-mail: subscribe@circuitcellar.com

Subscribe: 800.269.6301, www.circuitcellar.com/subscribe, Circuit Cellar Subscriptions, P.O. Box 5650, Hanover, NH 03755-5650

Address Changes/Problems: E-mail: subscribe@circuitcellar.com

GENERAL INFORMATION

860.875.2199, Fax: 860.871.0411, E-mail: info@circuitcellar.com

Editorial Office: Editor, Circuit Cellar, 4 Park St., Vernon, CT 06066, E-mail: editor@circuitcellar.com

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

AUTHORIZED REPRINTS INFORMATION

860.875.2199, E-mail: reprints@circuitcellar.com

AUTHORS

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

CIRCUIT CELLAR®, THE MAGAZINE FOR COMPUTER APPLICATIONS (ISSN 1528-0608) is published monthly by Circuit Cellar Incorporated, 4 Park Street, Vernon, CT 06066. Periodical rates paid at Vernon, CT and additional offices. **One-year (12 issues) subscription rate USA and possessions \$23.95, Canada/Mexico \$34.95, all other countries \$49.95. Two-year (24 issues) subscription rate USA and possessions \$43.95, Canada/Mexico \$59.95, all other countries \$85.** All subscriptions orders payable in U.S. funds only via Visa, MasterCard, international postal money order, or check drawn on U.S. bank. **Direct subscription orders and subscription-related questions to Circuit Cellar Subscriptions, P.O. Box 5650, Hanover, NH 03755-5650 or call 800.269.6301.**

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

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

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

Entire contents copyright © 2009 by Circuit Cellar, Incorporated. All rights reserved. Circuit Cellar is a registered trademark of Circuit Cellar, Inc. Reproduction of this publication in whole or in part without written consent from Circuit Cellar, Inc. is prohibited.

What's the difference?



Price!

Electronic components work no matter what price you pay. Jameco carries everything you expect at prices below what others charge. But the price savings don't stop there. Jameco offers additional savings with its array of house brand and factory-overrun products.

The Jameco difference begins with the industry's highest quality catalog and is backed by the industry's longest warranty plus much more.



- Over 100,000 skus
- 99% of catalog products are in stock right now
- Low price guarantee

JAMECO[®]
ELECTRONICS

1-800-831-4242

◀◀ Order your FREE catalog today at www.Jameco.com/Price

INSIDE ISSUE

227

June 2009 • Communications

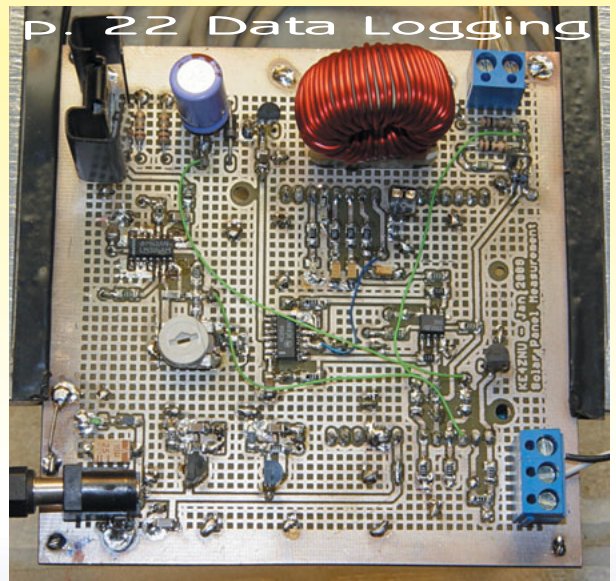


p. 14 Virtual Keyboard Design



p. 30 SSI Controller

- 14** **Keystroke Communication**
Design a Customizable Virtual Keyboard
Naweed Paya & Venkat Ganesh
- 30** **SSI Controller for Linear-Position Sensors**
Robert Bowen
- 42** **Autonomous Vehicle Design**
Embedded Systems, Sensor Technology, and Motor Control
Chris Britney, Paul Green, Andy Heath, Stephen Lau, and Kylee Lathrop



p. 22 Data Logging

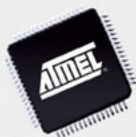
- 22** **INTELLIGENT ENERGY SOLUTIONS ABOVE THE GROUND PLANE**
Solar Data Logger (Part 2)
Data Points
Ed Nisely
- 52** **THE DARKER SIDE**
High-Speed Signal Transmission
From Eye Diagrams to Preemphasis and Equalization
Robert Lacoste
- 60** **FROM THE BENCH**
Location Notification
A Look at Anisotropic Magnetoresistance Sensors
Jeff Bachiochi
- 68** **SILICON UPDATE**
Easy (E)mbed
An Alternative Approach to Embedded Programming
Tom Cantrell

- TASK MANAGER** **4**
Forward-Thinking Designers
C. J. Abate
- NEW PRODUCT NEWS** **8**
edited by *John Gorsky*
- CROSSWORD** **78**
- INDEX OF ADVERTISERS** **79**
July Preview
- PRIORITY INTERRUPT** **80**
Lawns and Ponds
Steve Garcia

Hammer Down Your Power Consumption with picoPower™!



THE Performance Choice of Lowest-Power Microcontrollers



Performance and power consumption have always been key elements in the development of AVR® microcontrollers. Today's increasing use of battery and signal line powered applications makes power consumption criteria more important than ever. To meet the tough requirements of modern microcontrollers, Atmel® has combined more than ten years of low power research and development into picoPower technology.

picoPower enables tinyAVR®, megaAVR® and XMEGA™ microcontrollers to achieve the industry's lowest power consumption. Why be satisfied with microamps when you can have nanoamps? With Atmel MCUs today's embedded designers get systems using a mere 650 nA running a real-time clock (RTC) and only 100 nA in sleep mode. Combined with several other innovative techniques, picoPower microcontrollers help you reduce your applications power consumption without compromising system performance!

Visit our website to learn how picoPower can help you hammer down the power consumption of your next designs. PLUS, get a chance to apply for a **free AVR design kit!**

<http://www.atmel.com/picopower/>



INDUSTRIAL USB-TO-RS-232/422/485 ADAPTER

The **USB-FLEXCOM4** offers users the choice of four field-selectable RS-232, RS-422, or RS-485 protocols per port. This serial module was designed for use in systems where there are never enough serial ports to communicate with legacy devices as well as applications where old-school serial protocols are still the right choice for the design of new systems.

The USB-FLEXCOM4 conforms to the USB/104 form factor for flexible mounting either in PC/104 or USB/104 stacks. The device is available either packaged in a small, rugged, industrial enclosure, or as an OEM board (no enclosure). It is fully compatible with both USB 1.1 and USB 2.0 ports. Hot-plug functionality allows for quick connect/disconnect whenever you need additional I/O on your USB port.

The USB-FLEXCOM4 is supported for use in most operating systems and includes a free Linux (including Mac OS X) and Windows 2000/XP/2003-compatible software package. This package contains sample programs and source code in Visual Basic, Delphi, and Visual C++ for Windows. Also incorporated is a graphical set-up program in Windows. Third-party support includes a Windows standard DLL interface usable from the most popular application programs, and includes example LabVIEW VIs. Embedded OS support includes Windows XPe.

The USB-FLEXCOM4 costs **\$254**.



ACCES I/O Products, Inc.
www.accesio.com

NEW PRODUCT NEWS

Edited by John Gorsky



We add value to PCBs when others just sell it.

- One Stop Manufacturing Service
- \$10 ea. for 5pcs 4"x4" 2L
- Free Solder Mask & Silkscreen
- Free Components
- Free Assembly

Design Prototype Production



Designing Service

3D Enclosure Designing Virtual Assembly PCB Design



Assembly



Components



Pcbs



Enclosures



Fpcs



Keypads

www.EzPCB.com Email: sales@ezpcb.com

Please Visit www.machinepier.com For Machining

LOW-PROFILE PCB CONNECTOR

The POWERMOD PSX product line is a low-profile family of connectors which is 30% smaller than comparable products. The family of connectors is perfect for when board space is at a premium and overall mating heights must be minimized. The connectors are ideal for power connections to embedded AC/DC power supplies or for board-to-board connections in servers, storage enclosures and communications equipment. Target markets include telecommunications, datacom/network backbone, and data storage.

The streamlined design of the POWERMOD PSX offers excellent performance with power contacts rated to 60 A and signal contacts rated to 1 A (ratings per UL1977). The series is available in either power and signal versions or power-only to provide power distribution and control in a single connector body. The combined power and signal connectors offer 80 signal contacts, in addition to either two or four power contacts. The power-only connectors are available in two-, four-, six-, and eight-contact configurations. Both versions offer male connectors in vertical and right-angle configurations and a vertical female configuration to deliver support for coplanar, backplane, or mezzanine connections.

Pricing is dependent upon configuration, so please contact the manufacturer or a distributor for pricing.



Anderson Power Products
www.andersonpower.com

NPN

Expand Your I/O with Rabbit® RIO

Versatile Device with I/O Options

- Add functionality without costly processor changes
- Multiple Processor Interfaces
- Add 38 I/O
- Configure I/O for PWM, TRIAC, input capture, or decoder



Rabbit RIO™ Programmable I/O Application Kit for **\$299**

RABBIT™



Order Online At
rsappkits.com

08033

RF Specialists

RF Modules

From Part 15 to Part 90 Compliant
 Narrow Band FM, UHF Multi-Channel



GSM/GPRS

M2M Solutions
 GSM/GPRS modules and modem series



Industrial Bluetooth

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



Data Loggers

Stand Alone and
 Wireless Mesh Networking Logger



GPS

OEM Modules and USB ZigBee Sticks,
 Mesh Networks



ZigBee Pro

OEM Modules and USB ZigBee Sticks,
 Mesh Networks



Wi-Fi

RS232 / 422 / 485 to Wi-Fi Adapter

Connect Data Acquisition Equipment through Serial Port to Wi-Fi network

LE MOS
 INTERNATIONAL

www.lemosint.com
 866.345.3667
sales@lemosint.com

Celebrating Our 10th Year Anniversary

Get All Your Test Equipment In A Quick and Speedy Shipment!

20% Sale On Entire Selection Of Test Equipment!!


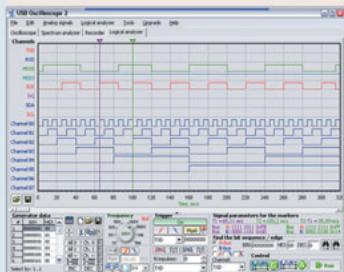


Oscilloscopes
Probes
Digital Multimeters
Function Generators
Voltage Testers
Capacitance Meters
Analog Multimeters
Test Leads
& More...


DesignNotes.com
What Your Electronic Hobby Stores Used To Be

1-800-957-6867 www.DesignNotes.com

USB Oscilloscope for \$169.50
Logic and Spectrum Analyzers, Generator.
www.HobbyLab.us

CIRCUIT CELLAR
FOR COMPUTER APPLICATIONS



Make sure you're signed up to receive Circuit Cellar's monthly electronic newsletter. *News Notes* will keep you up to date on Circuit Cellar happenings. Stay in the loop!

Register now. It's fast. It's free.
www.circuitcellar.com/newsletter/

USB RS-232 ADAPTER

The **SeaLINK+232-DB9** serial adapter is a single-port USB-to-RS-232 serial adapter incorporating a ruggedized, overmolded enclosure. This enclosure improves reliability and durability in industrial and mobile applications, such as GPS navigation systems, bar code readers, signature input devices, serial printers, scales, and similar applications.

The serial port appears as a standard COM port to the host computer to enable easy set-up and provide compatibility with legacy software. The SeaLINK+232-DB9 features programmable baud rate and data formats with 128-byte transmit and 384-byte receive buffers. The USB serial adapter is compatible with all standard PC baud rates and supports high-speed communication to 921.6 kbps. The adapter is powered by the USB port. Status LEDs molded into the enclosure indicate serial data activity and connection to the host.

All the company's USB serial adapters ship with the SeaCOM suite of Windows drivers and diagnostic utilities. Also included is WinSSD, a full-featured application providing powerful testing and diagnostic capabilities. Use WinSSD for Bit Error Rate Testing (BERT), throughput monitoring, loopback tests, and transmitting test pattern messages.

Standard operating temperature range for the adapters is 0° to 70°C, and extended operation (-40° to 85°C) is available. The attached 44" cable is fully shielded to protect the adapter from RF and EMI interference, which are common in mobile and industrial environments. The SeaLINK+232-DB9 serial adapter is available immediately. It costs **\$79** (Item# 2105R).

Sealevel Systems, Inc.
www.sealevel.com



ANTI-VANDAL SWITCH

The **PV8** is a newly available anti-vandal switch. This switch features a metal body and actuator with a 25-mm diameter panel cutout. Multiple functions are available with the PV8 and it is available with or without illumination. The illuminated option includes the choice of a dot or a ring lens style. The switch is sealed to IP65 standards.

The switch offers an electrical life of 50,000 cycles minimum and a mechanical life of 1,000,000 cycles minimum. It is available in 1P2T, 2P2T, momentary, and illuminated configurations. The PV8 offers a contact rating of 2 A at 48 VDC and a contact resistance of 50 MΩ (initial max) at 1 A, 12 VDC with an insulation resistance of 1,000 MΩ (min. at 500 VDC). Dielectric strength is rated at 2,000 VAC for 1 min.

Markets for the PV8 Series anti-vandal switch include elevators, public ticket dispensers, and security panels, among others. Pricing starts as low as **\$12** in volume quantities.

E-Switch, Inc.
www.e-switch.com



NPN

VCO OPERATES AT 1,560 MHZ

The model ZRO1560A1LF is a new RoHS-compliant voltage-controlled oscillator (VCO) for use in the L-band. The ZRO1560A1LF operates at 1560 MHz with a tuning voltage range of 0 to 5 VDC. This VCO features a typical phase noise of -121 dBc/Hz at 10 kHz offset and a typical tuning sensitivity of 0.7 MHz/V.

The VCO is designed to deliver a typical output power of 0 dBm at 5-VDC supply while drawing 18 mA (typical) over the temperature range of -40° to 85° C. This VCO features typical second-harmonic suppression of -30 dBc and comes in an industry-standard MINI-16 package measuring 0.5" x 0.5" x 0.22". It is available in tape and reel packaging for production requirements. The ZRO1560A1LF is also ideal for automated surface-mount assembly and reflow.

ZRO1560A1LF is well-suited for fixed wireless and test and measurement applications that require superior second-harmonic suppression and ultra low phase noise performance.

The VCO is priced at **\$29.95** with a five-piece minimum.

Z-Communications, Inc.
www.zcomm.com



HIGH-POWER DC OUTPUT SSRs

The new D1D and D2D series of FET output DC solid-state relays—formerly rated up to 40 A at 100 VDC and 12 A at 200 VDC—now includes new models rated at 60, 80, and 100 A at 100 VDC, as well as 40 A at 200 VDC. The new models—offered in the industry-standard "hockey puck" SSR package—add to the present offering, which includes the D06 rated to 100 A at 60 VDC, the D2D and D4D series rated to 12 A, and D5D models rated to 10 A at 200, 400 and 500 VDC, respectively.

The new D1D relays feature a low on-state resistance FET output capable of carrying 60, 80, or 100 A when the SSR is mounted on an appropriately rated heat sink. Likewise, the new D2D40 features a similar FET design rated to 40 A with appropriate heat sink. Control for these DC output SSRs is by a transformer isolated 3- to 32-VDC logic-compatible input providing 2,500-VAC input to output isolation. The outputs are diode-protected and specified to operate up to 1 kHz. DC output SSRs are ideal for heating and lighting control, as well as motor control in commercial, industrial, medical, aerospace, rail, and automotive applications.

Pricing is available upon request. Please contact the manufacturer directly.

Crydom, Inc.
www.crydom.com



NPN



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



**Embedded & Network
Computing Technologies**

Imagine YOUR systems
with OUR ready-to-use
boards!



Embedded Computer
TNY-A9G20
for data processing.



Daughter board
DAB GPRS
for data transmission.
(Pass-through technology)



Daughter board
DAB GPS
for data
acquisition.

Now, with pass-through technology,
you can plug Daughter boards
one on top of the other.

www.calao-systems.com



TOWER LIGHTS

EZ-LIGHT TL50 tower lights are exceptional for highly visible operator guidance and indication of equipment status. Displaying up to five colors in one tower and allowing multiple colors to be lit simultaneously, these preassembled and pre-configured multi-segment indicators replace conventional stack lights, which often require time-consuming assembly and complex wiring. In addition, models featuring audible alert with adjustable intensity are available for applications requiring sound indication.

The tower lights require less than 2 W of power, compared to 15 W for competitive units, making them an energy-efficient solution. Their aesthetic shape, along with a water- and oil-tight industrial housing, makes it



possible to mount directly to machines, while standoff pipe and adapters are available for elevated mounting. Green, yellow, red, blue, or white LED colors are offered to meet a variety of application needs.

The lights shed debris and moisture with an aesthetic design and feature a prewired 2 m attached cable or Euro-style quick-disconnect, depending on model. They offer bimodal NPN or PNP operation depending on hookup and are compatible with PLC or other logic-level control outputs. The TL50 delivers IP65 protection rating for general-purpose models and an IP50 rating for audible models.

Pricing is dependent upon configuration.

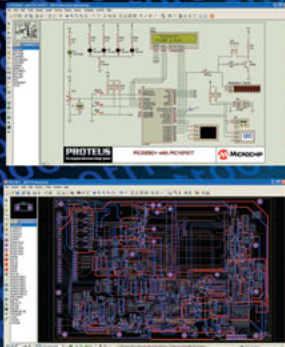
Banner Engineering Corp.
www.bannerengineering.com

NPN

THE FUTURE OF EDA SOFTWARE IS HERE! AND IT'S AFFORDABLE

SCHEMATIC CAPTURE, SIMULATION, PCB LAYOUT

PROTEUS DESIGN SUITE FEATURES:



- Easy to use, standard Windows interface.
- Publication quality schematics.
- Fully configurable bill of materials.
- Large component libraries for schematic simulation and PCB layout.
- Mixed mode SPICE circuit simulation.
- Co-simulation of PIC, AVR, 8051, ARM7 & BasicStamp microcontroller firmware.
- Integrates with MP-LAB and AVR Studio.

- Automatic component placement and gateswap optimization.
- Highly configurable design rules.
- Interactive design rule checking.
- Polygonal and split power planes.
- World class shape based autorouter.
- Automatic mitre/unmitre commands.
- Industry standard CAD/CAM & ODB++ output.
- Integrated 3D Viewer with 3DS and DXF export.

All levels of the Proteus Design Suite now include a world class, fully integrated shape based autorouter at no additional cost — prices start from just \$249.00!

R4 Systems Inc.

www.r4systems.com * Toll-Free 866.499.8184 * info@r4systems.com

USB MULTI-FUNCTION RELAY, ADC, & OPTO-IN MODULE

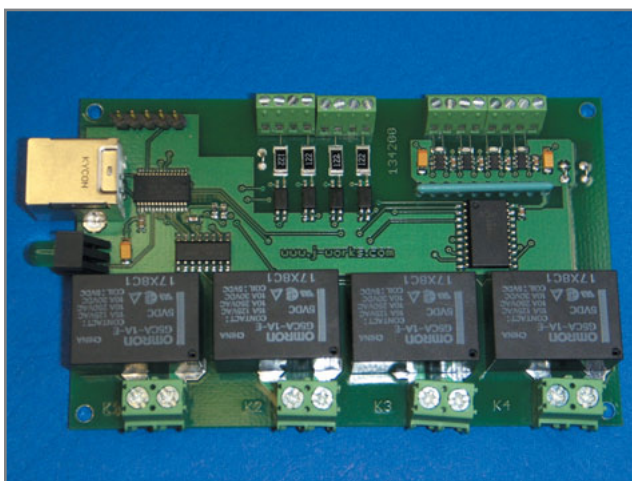
The model JSB-380 is a USB 2.0 controlled multi-function module with four relays, four ADC channels, and four digital optoisolated inputs. Each ADC channel can measure -5 to 5 V with an optional 12- or 16-bit resolution and single-ended or differential inputs. There is also a user-controlled PGA in 1, 2, 4, 8 steps. The optoisolated inputs can support two ranges of inputs, one trip is at 3.3 V with a 15-V range, and the other is 10 V with a 30-V range. Relays are SP5T 15 A 240 VAC. Screw-type terminal blocks are on the module to provide a quick and easy connector to all the IO. This small form factor module replaces internal PC-based plug-in cards or up to three USB modules for use in various test, control, monitor, and measurement applications.

The USB module programming interface is easy to use from Visual Basic, Visual C#, and Visual C++ via a standard windows DLL and a managed class library with sample source included. Any programming language that supports calls to a Win-

dows's DLL can also be used to control the JSB380 module. Each module has a unique serial number string which allows multiple USB devices to be controlled by the host USB controller as individual devices.

Single unit price starts at **\$160** for the 12-bit version and **\$190** for the 16-bit version.

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



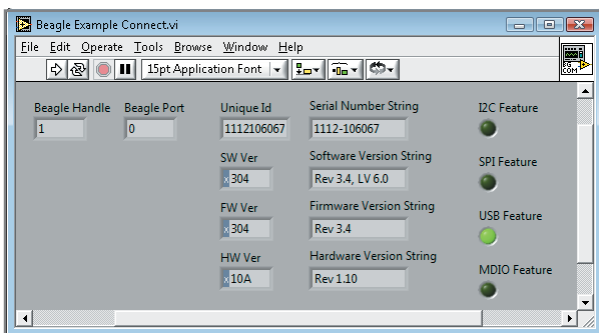
LABVIEW DRIVERS FOR CHEETAH ADAPTER & BEAGLE ANALYZER

LabVIEW drivers for the **Cheetah SPI host adapter**, the **Beagle I²C/SPI protocol analyzer**, and the **Beagle USB 12 protocol analyzer** are now available. These new drivers are available as free downloads on the Total Phase web site.

The new LabVIEW drivers add another dimension to the versatile functionality of these tools. The Cheetah SPI host adapter currently features gap-less SPI communication at speeds of up to 40-plus-MHz. The Beagle I²C/SPI protocol analyzer provides a high-performance bus-monitoring solution in a small, portable package that can display and filter I²C, SPI, and MDIO protocols in real time. Similarly, the Beagle USB 12 protocol analyzer can non-intrusively capture, display, and filter full- and low-speed USB traffic in real time.

The Cheetah SPI host adapter costs **\$350**, the Beagle I²C/SPI protocol analyzer is **\$300**, and the Beagle USB 12 protocol analyzer is **\$400**.

Total Phase, Inc.
www.totalphase.com



Software Trials Available

JTAG (J-Link™)

+++ ARM7/9 Cortex M3 +++

- USB to JTAG
- Fast 720kb/s Download Speed
- Serial Wire Debug (SWD) Support
- Multicore Debugging Support
- Auto JTAG Speed Recognition

The J-Link can be coupled with a number of available software modules to fit your application needs.

J-Flash is a stand-alone application used with the J-Link to program internal and external flash devices.

J-Link RDI permits the use of the J-Link with an RDI compliant debugger.

J-Link GDB Server is a remote server for GDB.

J-Link Flash Download is a module used to download your program into flash even if your debugger does not have a flash loader.

J-Link Flash Breakpoint permits you to set an unlimited number of software breakpoints while debugging in flash.

J-Link SDK is a standard DLL that extends the full functionality of the J-Link to your proprietary application.



We also offer a JTAG isolator which can be used to offer electrical isolation between your target hardware and the J-Link. This is essential when the development tools are not connected to the same ground as the application. It is also useful to protect the development tools from electrical spikes that often occur in some applications, such as motor control applications.



Special Offer

Includes the J-Link, J-Link GDB Server, and the J-Link Flash Download.
www.segger-us.com/ncu.html

J-Link Non-Commercial (NCU) Bundle

www.segger.com

NPN

Keystroke Communication

Design a Customizable Virtual Keyboard

You can build a customizable virtual keyboard around a microcontroller and some additional affordable parts. This flat keyboard design is the size of a piece of standard paper, which makes it perfect for use at a desk or on an airplane tray table.

I usually like to travel without any carry-on luggage, especially on long international flights. But on a flight from Chicago to Frankfurt last winter, it seemed like a good idea to bring my laptop along. I imagined I'd finally be able to finish my overdue research report during all that free time onboard. Unfortunately, I realized all too soon that the tray table was undoubtedly too small for the job. There just wasn't enough space to keep my 14" laptop LCD positioned at the right angle and still type properly without contorting my fingers in awkward positions. And even after I somewhat managed that, I was still worried my eight-year-old neighbor would spill the flimsy glass of soda all over my keyboard.

I first got the idea for a virtual keyboard while thinking

about how to address the dire needs of a modern high-tech traveler. The idea was simple. Imagine if the screen in the front of your plane seat was essentially a PC with some basic software and the tray table had a keyboard printed on it that couldn't get ruined by spilled liquid like soda. A solution as simple as that would certainly improve the in-flight experience of every passenger, right?

Luckily, I had a chance to implement part of this idea with my partner Venkat Ganesh at Cornell University. After learning the basic functionality of an Atmel ATmega32 microcontroller and working through some hands-on lab sessions, Venkat and I designed a customizable virtual keyboard in about five weeks.

We replaced a physical keypad with a customizable keyboard printed on a standard letter-size surface. "Keystrokes" on this printed surface are read and translated into real input (see [Photo 1](#)). We can place the virtual keyboard on flat surfaces such as desktops, airplane tray tables, and countertops. In theory, it can be interfaced with any computing device that requires text entry. This eliminates the need to carry around cumbersome electronic devices. Lamination protects the paper and prevents any mechanical damage in harsh environments. In addition, we can reconfigure the device's buttons on the fly to create a new keyboard layout using a GUI we built in Java. We can then transfer the data to the device via a PC's serial port.

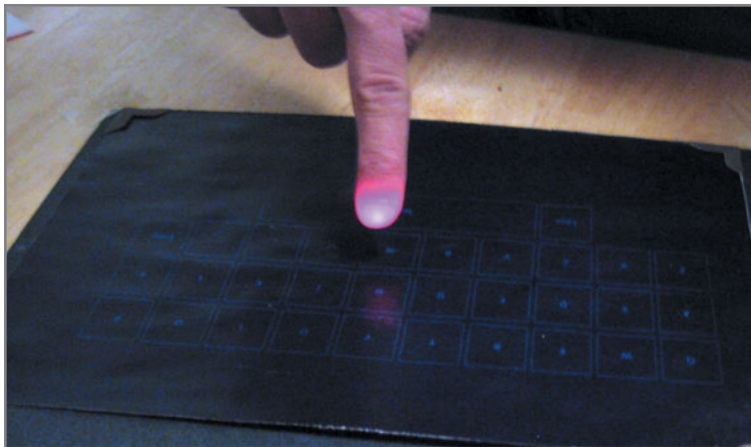
FOUR-PART DESIGN

The virtual keyboard has four main components: a laser, a camera, a microcontroller, and a keyboard printout.



Photo 1—This is the finished project. You can use it on a flat surface such as a desktop or tray table on a plane.

Photo 2—A finger shines bright red when passing through the invisible plane of red light hovering a few millimeters above the typing surface. This plane of light is generated by a 635-nm red laser module with a line-generating diffractive optical element attached to it.



The conventional, off-the-shelf red laser has a line-generating diffractive optical element attached to it. This assembly generates an invisible plane of red light hovering a few millimeters above the typing surface. When a finger passes through this plane, it shines bright red in that region (see **Photo 2**).

The CMOS camera continuously captures images of the region where the printed keyboard is supposed to be placed, and it transmits the data to an ATmega32 microcontroller. The microcontroller then checks these images for red color data that is above a preset threshold and satisfies other key-press detection criteria. The threshold concept works in this case because the laser shining on a typical human finger generates saturating values of red color data, which is easily distinguishable from its surroundings.

The keyboard is simply a standard letter-size sheet with a custom keyboard layout printed on it. We used a PC's serial port and a GUI we developed in Java to program the printed keyboard layout into the device. This GUI gives us a blank grid of buttons. We can assign any button to any letter or number we want.

MICROCONTROLLER & LASER

We started by soldering the ATmega32 microcontroller and corresponding parts onto a prototype board designed

by Professor Bruce Land (see **Figure 1**). The board includes a power supply, 16-MHz crystal clock, RS-232 interface for serial communication, and various bypass capacitors. In addition, a six-pin header allows flash memory programming from an STK500 Development board.

Our original plan was to use an invisible infrared laser to detect button presses using the CMOS camera, but after initial experimentation we quickly realized that user safety would be a major issue. A user would never know if he was staring directly at the laser, so there would be no way to prevent eye damage. We also realized that the OmniVision OV6630 CMOS camera was not too effective at detecting infrared light. So, we settled on a Class II 635-nm red laser instead.

Our laser module came with a built-in driver, so we didn't have to worry about biasing the laser properly to make it operational. All we had to do was connect the laser to a 3-V power source, which we obtained by connecting a 3-V voltage regulator to the 9-VDC power supply. The laser module also came with a line-generating

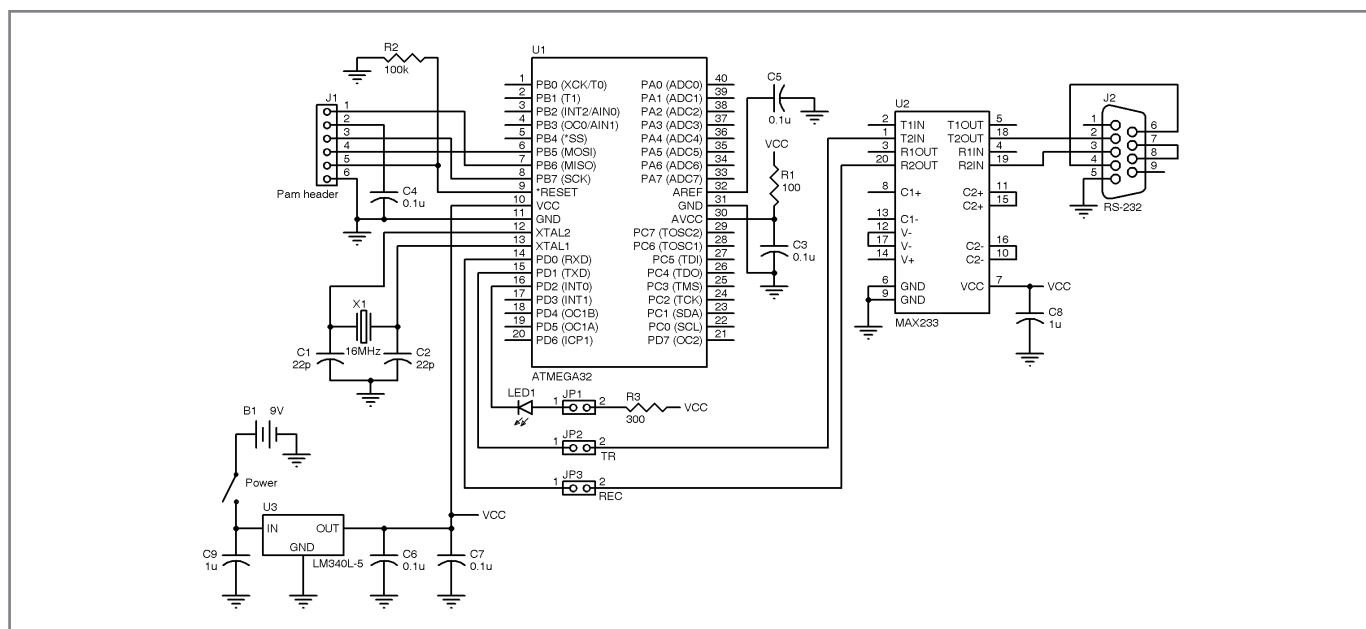


Figure 1—Cornell University professor Bruce Land supplied us with this schematic of the ATmega32 prototype board.

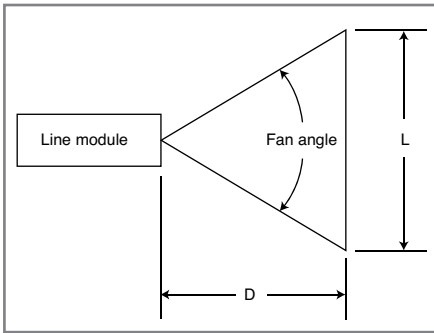


Figure 2—Using the geometry shown for laser line generation, we calculated our laser module's fan angle to be 44 degrees. Thus, to cover the entire width of our keyboard, we had to place the laser approximately 13" away.

diffractive optical element (DOE) attached to it. However, we didn't know the fan angle for this DOE, so we had to experiment with various distances in order to obtain a line length of at least 10.5", which was required to cover the entire width of our printed keypad. The fan angle was eventually calculated to be 44 degrees; therefore, we had to place

the laser at a distance of approximately 13" to obtain good results (see [Figure 2](#)).

CAMERA & CASING

We used a C3038 1/4" color sensor module with digital output, which uses an OmniVision Technologies OV6630 CMOS image sensor. We chose this camera module for its affordability and its ability to output image data in progressive scan mode. Progressive scanning was an important consideration for us because the computational power available on a 16-MHz ATmega32 microcontroller simply is not enough to process entire frames at once at reasonable speeds. Nonetheless, it was certainly possible to process images line by line. After rigorous testing and a lot of research, we discovered that instead of using the entire RGB output from the camera, we could work with only the red channel data and still be able to identify keystrokes rather accurately. Hence, we connected the 8-bit red channel output from the camera (UV[7:0]) to PORTA[7:0] on the ATmega32 (see [Photo 3](#) and [Figure 3](#)).

The hardware assembly for our device—which was made out of a combination of scrap wood and cardboard—holds the camera at a fixed position such that it looks over the appropriate region of the printed keyboard (see [Figure 4](#)). In addition, it also holds the laser module at a fixed position such that the plane of red light completely covers the area above the printed keyboard. We permanently attached a piece of black poster board to the assembly and mounted four photo-corners on it (see [Photo 2](#)). This ensures that new custom-printed keyboards can be readily swapped in and out of the device while maintaining proper distances.

SOFTWARE & I²C COMMUNICATION

Based on prior experience and our interests, we first split the software component of this project into five main tasks: implementing the I²C protocol to communicate with the camera; reading image data from the camera to obtain at least six frames



It takes more than a top program to win the Product of the Year Award five times running*

- ▶ For instance, reasonable prices without hidden costs.
- ▶ A fair update policy without mandatory maintenance fees or costly add-on contracts.
- ▶ A competent hotline available free of charge for every customer.
- ▶ A program development and optimization process based on an open discussion with the users.

In other words: EAGLE is one of the world's highest rated schematic capture and PCB-board layout packages because for the last twenty years we have been treating our customers the way we would like to be treated ourselves.

An example: In the course of product development we implemented a

Follow-me Router

which saves an enormous amount of time during the manual routing process. This new function is available free of charge to every customer as are all new features within a main EAGLE release. In this particular case you need to have the autorouter module of version 5.

EAGLE
Version 5
Schematic - Layout - Autorouter
for Windows® Linux® Mac®

* Annual award of the leading German electronics magazine. The winners of eleven categories are determined by the reader's votes.

Pick the level that is right for you — pay only the difference for upgrades!

	Light	Standard	Professional
Max. number of schematic sheets	1	99	999
Max. board size	4x3.2 inch	6.4x4 inch	64x64 inch
Max. # of signal layers	2	4	16
Layout or Schematic Editor		\$249	\$498
Layout and Schematic Editor		\$498	\$996
Layout Editor and Autorouter		\$498	\$996
Layout Editor and Schematic Editor and Autorouter	\$49	\$747	\$1494

Standard and Light Editions have full functionality except for the limitations mentioned in the table.

You can use EAGLE Light for evaluation and non-commercial applications without charge. Download it from our web site.

www.cadsoftusa.com
800-858-8355

CadSoft Computer, Inc., 19620 Pines Blvd., Suite 217, Pembroke Pines, FL 33029
Hotline (954) 237 0932, Fax (954) 237 0968, E-Mail: info@cadsoftusa.com

Windows / Linux / Mac are registered trademarks of Microsoft Corp. / Linus Torvalds / Apple Computer, Inc.

per second; processing the images to obtain the coordinates of a pressed key; converting the pressed key into a scan code, which is then transmitted to the PC using the PS/2 protocol; and sending serial data from a PC running our Java application to update the array of scan codes in the microcontroller.

The camera is essentially the device's backbone, so a big part of our challenge was to implement the I²C communication protocol to properly read and write to the camera. The protocol was not easy to work with, and there were about 92 registers we could use to change camera settings. At first, we considered using the Two-Wire Interface (TWI) protocol provided by CodeVision to communicate with the camera, but we were unable to do so. Consequently, we modified and used a version developed previously by our classmate Ruibing Wang.

According to the specifications, two wires in an I²C bus—serial data (SDA) and serial clock (SCL)—carry information between the devices connected to the bus. Each device is recognized by a unique address and can operate as either a transmitter or receiver, depending on its function. In addition to transmitters and receivers, devices can be considered as masters or slaves when performing data transfers. A master is the device that initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.

For our purposes, the microcontroller provides the clock signal to the camera. Its frequency is given by the following formula found in the ATmega32 manual:

$$SCL \text{ Frequency} = \frac{\text{CPU Clock frequency}}{16 + 2(TWBR) \times 4^{TWPS}}$$

TWBR is the TWI bit rate register. TWPS represents the prescaler bits in the TWI status register. We decided

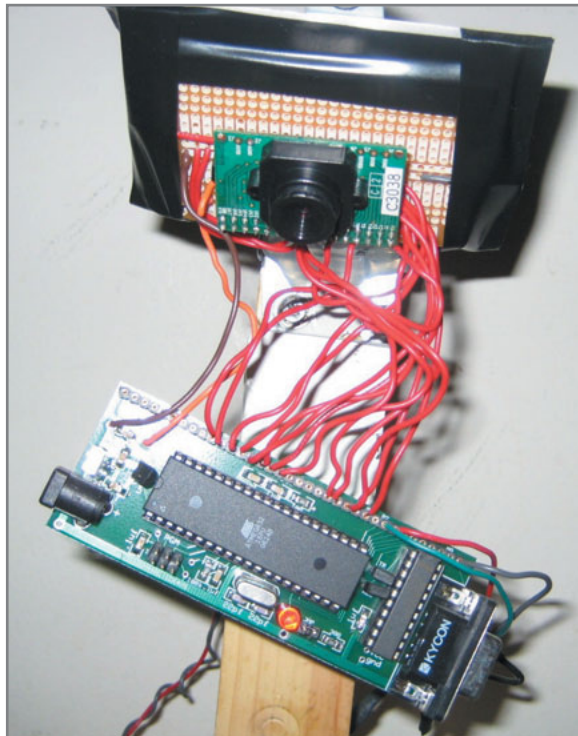


Photo 3—A C3038 image sensor module is mounted on a custom PCB and connected to the prototype board containing an ATmega32 microcontroller. The camera is mounted at an angle of approximately 52 degrees from the horizontal so it can view the entire printed keyboard area.

the fastest solution would be to satisfy the minimum frequency requirement, which was to set the bit rate register on the ATmega32 (TWBR) to 0x72 and the TWI status register (TWSR) to 0x00. The rest of the code followed the standard I²C protocol originally defined by Philips. After a bit of trial and error, we learned that the camera registers could be written in a three-step process: write a start bit, followed by a target register address, and then the target data. There was no need to read from the camera registers except in the initial testing phase when we had to make sure the protocol was working properly.

MAIN OPERATION

At first, we initialized PORTA on the ATmega32 to receive image data from the camera and PORTC to communicate with

We Listen. Think. And Create.

Serial I/O

Digital I/O

Distributed I/O

Industrial Computing

HMI

Create a powerful distributed data acquisition system with Sealevel's SeaI/O™ family of digital and analog control modules.

SeaI/O Solutions Offer:

- Digital & Analog I/O
- Connection via Ethernet, USB, RS-485, or RS-232
- Daisy Chain Expansion
- Field Removable Terminal Blocks
- Compact, Rugged Form Factor
- DIN Rail or Table Mount Design
- Robust Development and Configuration Tools

FOCUS

On Success
Call Today!

sealevel.com > sales@sealevel.com > 864.843.8067

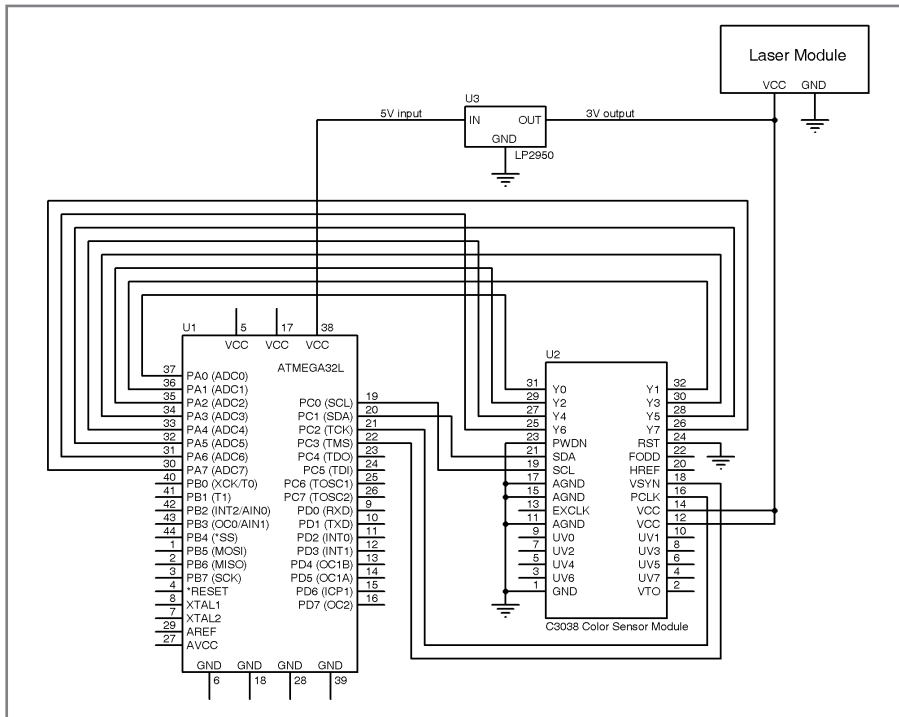


Figure 3—Here is the circuitry for the customizable virtual keyboard project. To prevent clutter, we didn't include the complete details of the ATmega32 connections in this schematic.

the camera over the I²C interface. The data rate was set to 19,200 bps to allow for sufficiently fast yet reliable serial communication. We then ran a calibration function that checked the keyboard area to determine a distinguishable value for red color threshold. This sort of calibration was important in this case because different ambient lighting conditions could have affected the device's operation, as we discovered after some initial testing.

We then called an `init_cam` function, which performed a soft reset on the camera before writing the required settings to appropriate camera registers. These registers changed the image frame size to 176 × 144 (the minimum resolution required to view the entire keyboard area in sufficient detail), turned on auto white balance, set the frame rate to 6 fps, and set the output format to 16 bits in the Y/UV mode. In this mode, the camera outputs images in the standard Bayer color format, which gives green color data on the 8-bit Y line (Y = G G G G) and blue and red color data on the 8-bit UV line (UV = B R B R). The code then enters an infinite loop, which continuously checks for the status of

the PS/2 transmission and tries to process the next captured frame if the transmitting queue is empty. If the queue is not empty, it is updated with the most recent key press detected and the PS/2 transmission is allowed to continue.

IMAGE PROCESSING

The `getRedIndex` function in our code captures rows of data from the camera and processes them one at a time. Before capturing meaningful image data, however, we first have to wait until a negative edge is seen on the VSYNC signal, which indicates

the arrival of new frame data. Subsequently, we also have to wait for a single pulse on the HREF signal to go by. Pulses on HREF indicate the arrival of each new line of valid data in a given frame and the first row of data happens to be invalid. At this point, we can begin capturing 176 pixels of data from each of the 144 vertical lines in the incoming image.

In this mode of operation, the UV line receives blue (B) and red (R) data. However, as we already mentioned, we discovered that we could work with only red data without sacrificing too much accuracy. Therefore, we stored only an array of 88 pixel values per line instead of the total 176 values by capturing the data on the UV line every two clock cycles. In addition, we tried to further streamline the image-processing work by trying to take advantage of the fact that the camera outputs information about the same set of pixels for consecutive rows. Our idea for optimization in this case was to completely drop data for the even rows, but we soon realized this was not going to save us anything. All our processing could be done anyway in the time interval between the end of one row of data and the beginning of the subsequent row.

After each vertical line of valid data, HREF stays negative for about 0.8 ms and the camera data becomes invalid. This gives us ample time to process one line worth of data. The first step in data processing entails looping through each of the 88 captured pixel values to check for values higher than the red

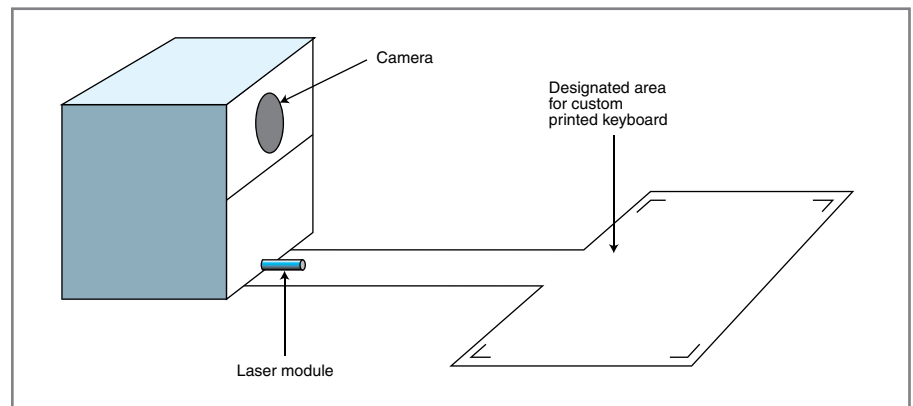


Figure 4—This is a diagram of the hardware casing. It was fabricated out of scrap wood and cardboard to minimize cost and save time.

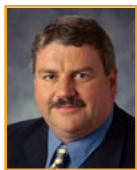
Standards Make Sense

Standards improve quality and enable designers to share components across different projects. Today, ARM® Cortex™-M profile processors, combined with the Cortex Microcontroller Software Interface Standard (CMSIS) and optimized middleware from the industry's largest ecosystem, are setting the hardware and software standards for microcontrollers.

These standards enable leading vendors such as Luminary Micro, NXP, and STMicroelectronics to supply advanced microcontrollers, while maximizing code reuse across multiple platforms.

Cortex-M3 Microcontrollers Make Sense

"The strengths of ARM processor-based NXP microcontrollers are fundamentally changing digital products by combining ease-of-use with high connectivity and low power consumption."



Geoff Lees
Vice President and General Manager,
Microcontroller Product Line



For more information visit
www.onARM.com

ARM

The Architecture for the Digital World®

© ARM Ltd. AD158 | 01.09

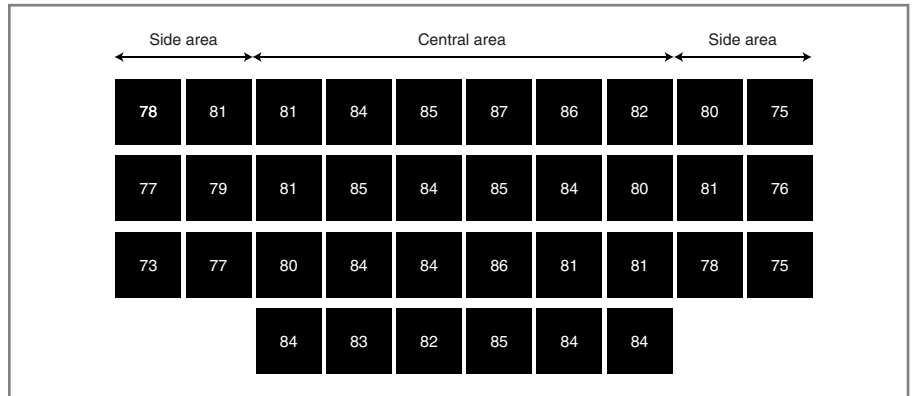


Figure 5—These are the testing results for keystroke detection accuracy. Note the difference in accuracy in the central region versus the peripheral region.

used to reset the waiting. The protocol allows a range of frequencies (10 to 16.7 kHz) that a computer would understand. We used a clock time of 250 and wait time of 700. Based on the 16-MHz microcontroller clock, these numbers translate to approximately 16 and 44 μ s, respectively. When the Timer1 compare A interrupt is fired, it transmits the bits in the order mentioned above when the clock is set to high. Otherwise, the clock state is updated. The rest of the code simply maintains a queue which holds the elements to transmit as characters. The queue also has "queueGet" and "queuePut" methods that update two pointers in the queue based on the number of existing elements and free spaces remaining.

PROGRAM THE EEPROM

One of the virtual keyboard's advantages is that it's customizable, which essentially means that key assignments can be changed on the fly. This feature can be quite useful considering that an entire QWERTY keyboard is not always necessary for all applications. For instance, a child might want to use the device as a simple gamepad, whereas a heavy-duty machine operator might want to have an entirely different set of keys. To accommodate both scenarios, we imagined the easiest approach would be to store the array of scan codes corresponding to each key in the microcontroller's EEPROM and turn on the RS-232 receive interrupt. Since the EEPROM is rewritable through a RS-232 port, unlike the

microcontroller's on-board flash memory, this enables us to instantly rewrite the scan codes based on the user's needs. And to make the entire process of assigning new keys even more user-friendly, we wrote a simple java applet which the user can run on any PC, enter scan codes of the keys they desire, and transmit the entire layout to the microcontroller through a standard COM port on the PC (see [Photo 4](#)).

TESTING

As a result of the limited viewing angle of the camera and positioning of the laser, we had to design and calibrate with various keypad layouts to make sure we could detect all of the buttons with reasonable accuracy. After extensive experimentation with various colors, we settled on a black background and blue letters for the printed keyboard. Because our device doesn't use its own light source, proper contrast was extremely important to distinguish the typing finger from the surrounding areas in various lighting conditions.

Figure 5 is our final design for the generic keypad and the testing results (percentage accuracy) for this layout. For the testing, we tried 100 keystrokes per key and set the acceptance threshold at 70% for side areas and 80% for the central area (see [Figure 5](#)). This means that if we can recognize a certain key accurately at least 70 or 80 times, respectively, out of the 100 times that it's pressed, that key passes the test.

Although an accuracy of 70% to 80% might sound unreasonably low for a keyboard, we also had to finish the entire project within five weeks. Thus, we could not afford to spend a lot of time perfecting the device's operation.

RESULTS

This project was a great learning experience. We successfully built a proof-of-concept device in less than five weeks on a \$75 budget. The keyboard works as we predicted, but typing speed is only about 60 characters per minute due to the ATmega32 microcontroller's limited processing capabilities.

Looking ahead, we are thinking about increasing the theoretical maximum typing speed. To do so, we might use another microcontroller in parallel, or maybe even an external FPGA to perform extra image processing. In addition, we might include multiple-key detection, sound effects for keystrokes, and a dynamic calibration algorithm that can be used to orient the custom-printed keyboard in any direction. This sort of functionality would require performing 2-D image transforms on the fly, which is not feasible with the current microcontroller. Last but not least, we might try to upgrade our current keystroke detection algorithm to improve typing accuracy. 📄

Maweed Paya (nap33@cornell.edu) is double-majoring in Electrical & Computer Engineering and Biological Engineering at Cornell University. He is passionate about embedded design and plans to combine his knowledge of physiology and digital design to develop human-implantable systems for biomedical diagnostics and treatment.

Venkat Ganesh (vsg3@cornell.edu) is double-majoring in Electrical & Computer Engineering and Computer Science at Cornell University. As a programmer, his interest is in the development of efficient software for microcontroller-based applications. Venkat enjoys playing soccer and solving Sudoku puzzles in his free time.

PROJECT FILES

To download the code and additional files, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2009/227.

RESOURCES

A. Chapweske, "The PS/2 Mouse/Keyboard Protocol," 2003, www.computer-engineering.org/ps2protocol/.

Cornell University, "RS232 Serial Communications with AVR Microcontrollers," Electrical Engineering 476, 2004, <http://instruct1.cit.cornell.edu/courses/ee476/Serialcom/>.

C. Tomasi, A. Rafii, and I. Torunoglu, "Full-Size Projection Keyboard for Handheld Devices," *Communications of the ACM*, Vol. 46, No. 7, 2003.

SOURCES

ADP3300ART-32 Voltage regulator
Analog Devices, Inc. | www.analog.com

ATmega32 Microcontroller
Atmel Corp. | www.atmel.com

C3038 Camera module
Electronics123.com, Inc. | www.electronics123.com

OV6630 CMOS Digital camera
OmniVision Technologies | www.ovt.com

THE ORIGINAL SINCE 1991
PCB-POOL
Beta LAYOUT

Servicing your complete PCB prototype needs :

- **Low Cost - High Quality**
PCB Prototypes
- **Easy online Ordering**
- **Full DRC included**
- **Lead-times from 24 hrs**
- **Optional Chemical Tin finish**
no extra cost

FREE LASER STENCIL WITH ALL PROTOTYPE PCB ORDERS

Watch "ur" PCB®
Follow the production of your PCB in
REALTIME

email : sales@pcb-pool.com
Toll Free USA : 1 877 390 8541
www.pcb-pool.com

ZADG TARGET Decipher PROTELS NATIONAL INVESTMENTS
Easy-PC Sprint Layout

Beta LAYOUT

Solar Data Logger (Part 2)

Data Points



With a redesigned layout, better power supplies, and a superior inductor, Ed's solar data logger project is back on track. Here he presents the new circuitry and covers the topics of inductor core saturation,

PWM filtering (and ripple), and solar panel output.

ABOVE THE GROUND PLANE

by Ed Nisley

One of software development's key maxims tells you to "build one to throw away," meaning that your first attempt provides an opportunity to figure out what you really wanted to do and how to make it work. Hardware designers have long known the wisdom of breadboarding a new circuit; indeed, that term dates back to radios built on actual bread-kneading boards.

In my April 2009 column, I explored some problems arising from poor layout and an ill-chosen inductor (*Circuit Cellar 225*). This time, I can discuss the solutions and show some actual observations.

This one works!

NEW CIRCUIT & NEW LAYOUT

The new board layout in **Photo 1** separates the panel current-sink and voltage booster circuitry from everything else. Although the ground plane is common to the entire top and bottom surfaces, the panel's relatively high currents follow the lowest-impedance

path. Notice that the rear edge of the board has a wider solid-copper area, with the top and bottom planes stitched together, connecting the panel input ground terminal in J1 with the low side of the sense resistors and Q2. That's a tidier replacement for the heavy wire I soldered across the planes on the previous layout.

An Arduino Diecimila microcontroller board plugs into the two rows of pin headers near the center of the board. In a more formal and permanent arrangement, mounting screws and standoffs would join the boards at the two holes in diagonal corners of the Diecimila board.

The schematic in **Figure 1** shows the key circuit changes. I added separate power-supply regulators for the digital and analog circuits, with provision to power the Diecimila board from the same wall wart. The ground connection between the two boards now uses only two adjacent pins in the power header, isolating the analog reference ground pin, and preventing any digital currents

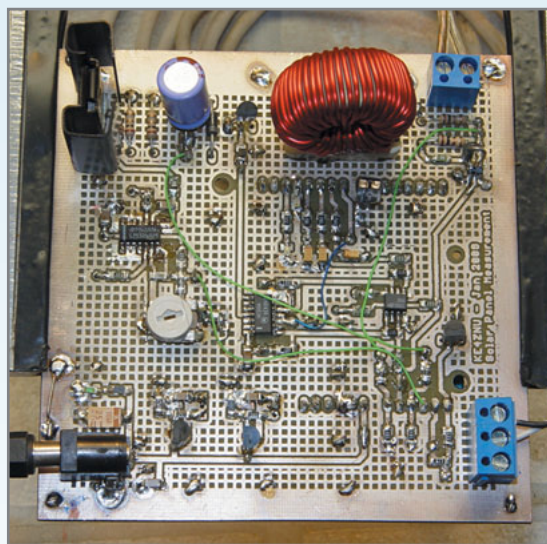


Photo 1—The expanded board layout and larger inductor core eliminated the previous problems. The Arduino Diecimila microcontroller attaches below the board on the two rows of pin headers and derives its power from the wall wart jack on the lower left.

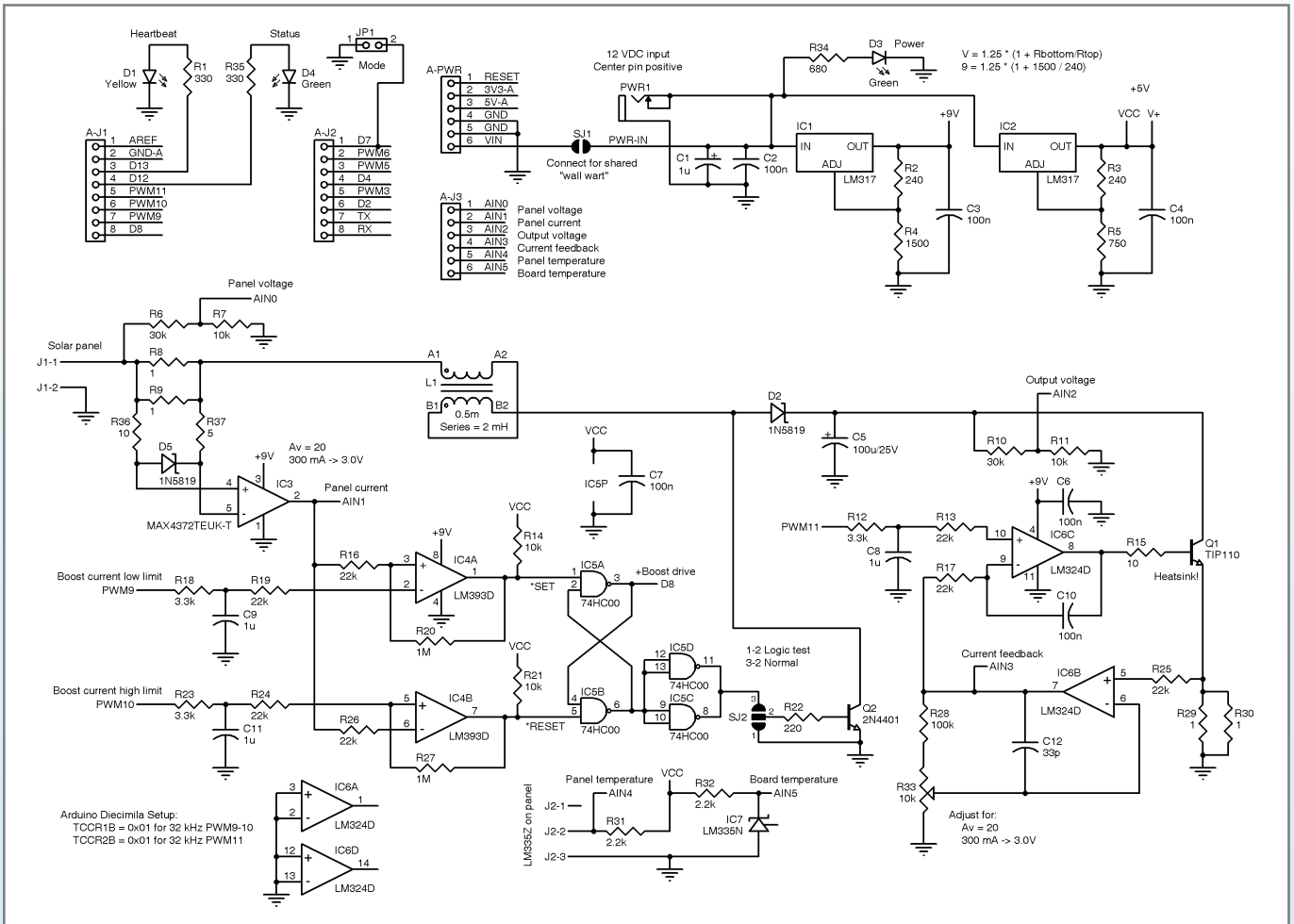


Figure 1—The new circuit includes a better inductor, separate power supplies, and protection for the fragile MAX4372 input circuitry. The improved board layout shown in Photo 1 eliminated the previous instabilities and had a straightforward bring-up process. The components have been renumbered from the schematic in the April 2009 column.

from crossing the analog terrain.

The moral of this story is that you cannot neglect analog design constraints when you're building a "simple" board attached to a microcontroller. The usual digital rules can get you in trouble, even if the schematic looks like it should work.

PWM ANALOG RIPPLE

The Decimila board uses the Atmel ATmega168's hardware PWM modules to produce analog output voltages. Of course, your board must apply a filter to the digital PWM signal that extracts its DC value. I elected to use a simple RC filter specifically to show the problems that can crop up from that choice.

PWM outputs, at least of the sort produced by the Decimila firmware, consist of a fixed-frequency rectangular pulse train with a duty cycle proportional to the desired analog value. The signal's fundamental frequency component is thus the PWM frequency, with additional components mostly at odd multiples of the fundamental. Surprisingly, the default Decimila firmware configuration sets a very low 500-Hz PWM frequency.

The PWM outputs have an 8-bit resolution over a 5-V

range, so 1 LSB = 19 mV or -48 dB. Ideally, you want to reduce the fundamental frequency component in the output to less than 1 LSB, so you need a PWM filter with at least 48 dB of attenuation at 500 Hz.

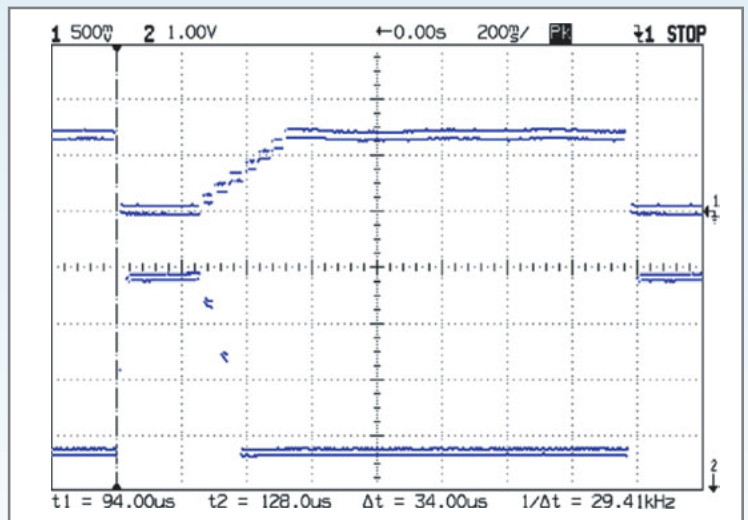


Figure 2—No matter what the firmware may call for, the actual current depends on the solar panel's illumination. Winter sunlight through two panes of glass tops out at 70 mA (Trace 1), with the voltage (Trace 2, measured at C5) dropping from 4 V.

Listing 1—This snippet of firmware steps the current sink load, measures the resulting values, stores them in arrays, and prints a single line to the serial output for use by an external logging program. The arrays simplify finding the maximum power point for the illumination on the panel, but obviously aren't needed for this simple application.

```
for (LoadIndex = 0; LoadIndex < LOAD_STEP_COUNT; ++LoadIndex) {
    PrintCSV(LoadIndex, ','); // array index

    LoadCurrent_Set[LoadIndex] = LoadIndex * LOAD_STEP_I; // Figure load current

    analogWrite(PIN_LOAD_I, PWM_SCALE_I(LoadCurrent_Set[LoadIndex])); // Set output load current
    delay(15); // wait of settling
    PrintCSV(LoadCurrent_Set[LoadIndex], ',');

    LoadVoltage[LoadIndex] = AI_SCALE_V(analogRead(PIN_LOAD_V)); // Read load voltage
    PrintCSV(LoadVoltage[LoadIndex], ',');

    LoadCurrent_Measured[LoadIndex] = AI_SCALE_I(analogRead(PIN_LOAD_FB_I)); // Read load current
    PrintCSV(LoadCurrent_Measured[LoadIndex], ',');

    PanelVoltage[LoadIndex] = AI_SCALE_V(analogRead(PIN_PANEL_V)); // Read panel voltage
    PrintCSV(PanelVoltage[LoadIndex], ',');

    PanelCurrent[LoadIndex] = AI_SCALE_I(analogRead(PIN_PANEL_I)); // Read panel current
    PrintCSV(PanelCurrent[LoadIndex], ',');

    PanelPower[LoadIndex] = (unsigned long int)PanelVoltage[LoadIndex] *
        (unsigned long int)PanelCurrent[LoadIndex] / 1000UL;
    PrintCSV(PanelPower[LoadIndex], ',');

    BoardTemperature[LoadIndex] = AI_SCALE_T(analogRead(PIN_BOARD_T)); // Read board temperature
    PrintCSV(BoardTemperature[LoadIndex], ',');

    PanelTemperature[LoadIndex] = AI_SCALE_T(analogRead(PIN_PANEL_T)); // Read panel temperature
    PrintCSV(PanelTemperature[LoadIndex], ',');

    Serial.println();

    ToggleLED(PIN_STATUS);
}

analogWrite(PIN_LOAD_I, 0); // Turn off load
```

A single-pole RC filter, the simplest and cheapest filter available, rolls off at 6 dB per octave (20 dB per decade) above the corner frequency at:

$$f_0 = \frac{1}{2\pi RC} = \frac{1}{2\pi\tau}$$

In order to be down 48 dB at 500 Hz, the corner frequency must be eight octaves lower, at:

$$f_0 = \frac{500}{2^8} = \frac{500}{256} = 1.95 \text{ Hz}$$

Combining those two equations tells you that you need rather high-valued resistors and capacitors.

Worse, an RC filter's response to a step input is an exponential, so you must wait about three RC time constants to get within -48 dB of the final value. That works out to 250 ms for a 1.95-Hz corner frequency, a rather leisurely pace that may come as a surprise.

Increasing the PWM frequency therefore pays off in both smaller components and faster response. I twiddled the ATmega168 hardware configuration bits to select a 32-kHz PWM frequency, a factor of 64 higher than the standard set-up,

which requires a filter corner frequency of 125 Hz. I paired 3.3-k Ω resistors with 1- μ F capacitors for a time constant of about 3 ms, a settling time under 10 ms, and a corner frequency of 50 Hz.

The oscilloscope screenshot in [Figure 2](#) shows that both the current and voltage traces consist of two parallel lines. It turns out that both are the result of a 32-kHz sine wave modulating the DC output.

Each current step in the upper trace represents 10 mA. The full-scale response is 500 mA, so 1 LSB = 2 mA and each step is 5 LSB. The 32-kHz modulation is roughly the same magnitude as one step, about -34 dB, so it seems the actual RC filter response isn't nearly as good as you'd expect from the equations.

Does that come as any surprise?

As nearly as I can tell from a quick measurement, the equivalent series resistance of those 1- μ F capacitors at 32 kHz is high enough to invalidate the simple ideal-capacitor model assumed by all the usual equations.

The moral of this story is that you must choose the PWM frequency based on your project's requirements, make sure your project can tolerate the effects of a given PWM frequency, and perhaps spend more on a better ripple filter. In

Do your MCUs suffer from bloated code?



HI-TECH C[®] PRO Denser code, better performance

HI-TECH C[®]
C O M P I L E R S
by Microchip Technology

HI-TECH C PRO compilers feature Omniscient Code Generation™ (OCG)

OCG is a whole-program compilation technology that can determine the usage of every register, stack, pointer, object and variable in your program, and generate fine-tuned code accordingly, effectively doubling your memory capacity, letting you fit more code in your MCUs.

Denser code, better performance: www.htsoft.com/ocg

HI-TECH C PRO compilers are available for Microchip PIC10/12/16/18/32 MCUs, from www.microchipdirect.com and Microchip distributors.



MICROCHIP

www.microchip.com/HI-TECH

any case, you must measure the results, verify them against the original equations, and perhaps adjust your simulations to match reality.

In this application, however, it's close enough to get some useful information from the panel.

CURRENT SINKHOLE

Figure 2 shows that the current sink is now stable, a pleasant change from the previous results. Notice that IC6C closes its negative feedback path through Q1 and IC6B, which means there's enough delay to impose a fairly low bandwidth on the whole circuit. C10 is thus far larger than the usual tens-of-picofarads that will generally stabilize even the most recalcitrant circuit.

Trimpot R33 adjusts the gain of IC6B so that 1 V from the Diecimila's (filtered!) PWM11 output produces 100 mA of current in Q1. The firmware can then read the actual output current through analog input AIN3. That circuitry resides in the upper-left corner of the board in Photo 1, benefiting from the lack of high-current return paths around its components.

The setup is simple: measure the voltages at Q1's emitter and the output of IC6B, then tweak R33 for a voltage gain of exactly 20. Because the sense resistors generate 0.5 V per amp, the overall gain is 10 V/A or, more conveniently, 1 V/100 mA. I wrote a short program that generates a repetitive current ramp from 0 through 300 mA, triggered the oscilloscope on one of the LEDs, and tweaked the pot.

LOAD TESTING

With the current sink circuitry up and running, it's a simple matter of software to measure the panel's characteristics, much as I did by hand in my February column. Listing 1 shows the core part of that routine: step the current sink load by 10 mA, measure all the analog inputs, dump the data to the microcontroller's serial output, and repeat for the next step. The Diecimila includes a serial-to-USB converter that's essentially



Figure 4—The panel's maximum power output varies dramatically due to clouds, tree shadows, and angle of incidence. Each data point represents slightly more than one minute of elapsed time.

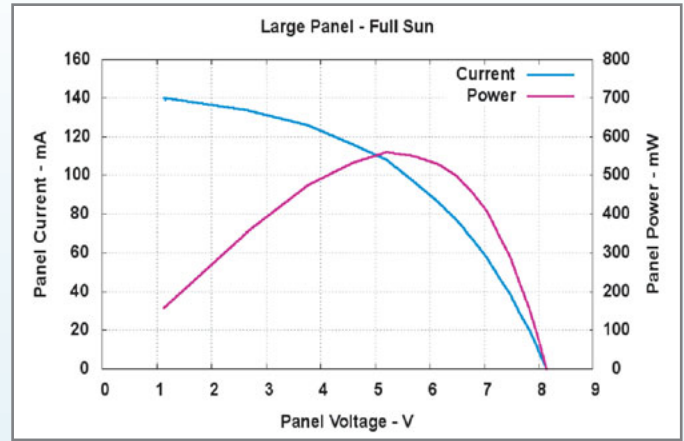


Figure 3—Full winter sun on the panel produces 550 mW at a panel temperature of 12°C.

transparent to the software on both ends of the connection.

I used `minicom` to capture the stream of data to a text file and a variety of GNU/Linux command-line filters to extract the relevant sections for the graphs.

Figure 3 shows the result of one such test, which closely resembles the manual results you saw earlier. Under early February sunlight, the large panel produces its maximum power at about 110 mA and just over 5 V.

The saturation voltage of Q1, a Darlington power transistor, plus the voltage drops across two pairs of current sense resistors, limits the low-end voltage to about 1 V even under maximum load. As you can see from Figure 3, however, that end of the performance curve lies far from the panel's maximum power point (MPP). You obviously need a different circuit configuration to characterize very low-voltage panels.

An automated system can take repeated measurements, presumably without getting bored, and Figure 4 shows a series of MPP measurements recorded later that afternoon. Another chunk of firmware locates the MPP of each load test and prints a line with those values.

The abrupt drop after 15 measurements (about 15 minutes) occurred as the sun moved behind a bare tree. Even though the diffuse shadows were barely visible, the power available from the panel dropped dramatically.

Figure 5 presents the MPP values in a different way: each point shows the current-voltage combination that produced the maximum power for that load test. The horizontal stripes show the varying voltages at each 10-mA current step.

The current measurements in Figure 5 come from the MAX4372 and appear to be about 10% lower than the nominal value set at the current sink. An additional bit of software, which I'll leave as an exercise, can compare the panel current values with the current sink values and create a look-up table or calibration equation for later use.

That difference brings us to the last section of the circuitry: the DC-to-DC converter.

INDUCTOR CORE CONCERNS

Extracting the maximum power from a solar panel requires a load that can operate at exactly that power level: it cannot draw more power than the available sunlight provides. Of course, it may use less, but a typical solar photovoltaic

installation doesn't have much spare power.

I think most solar power panels charge batteries, which generally require either a constant voltage or a constant current during much of their charging profile. For example, NiMH cells require a constant-current charge at a varying voltage and lead-acid (and, oddly enough, lithium) batteries require a current-limited constant-voltage charger.

As you can tell from Figure 5, the current and voltage available at the MPP vary dramatically, even without those pesky tree shadows, and probably won't match up with the battery's requirements over most of its charging profile. The solution is an intermediate stage between panel and charger that extracts the maximum power from the panel and presents either a constant voltage or a constant current to the charger. The simplest case provides a constant output voltage, so that's what I used in this circuit.

The circuitry in the lower-left quadrant of Figure 1 implements a simple boost-mode DC-to-DC converter. I

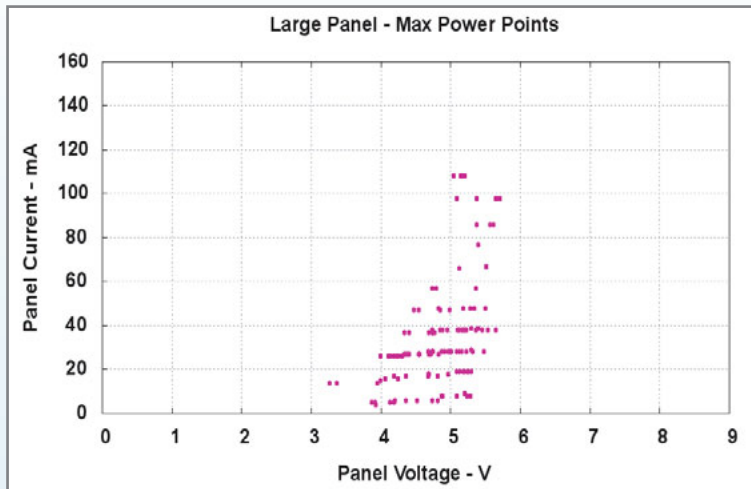


Figure 5—These points show the maximum power point for each sample in Figure 4. I have not corrected the panel current values for the gain error described in the text.

charges that energy into storage capacitor C5, with Schottky diode D2 preventing reverse current into the panel. The output voltage on the capacitor will be higher than the input voltage from the panel, with the load current proportionally lower.

Because the panel's power output drops dramatically on either side of its maximum power point, the inductor must conduct an average DC current equal to the panel's MPP current. Its core must have the desired inductance even at that DC bias, which is what killed my previous circuit design: the inductor had high inductance, but the core

picked a very large inductor to make the switching frequency low enough to show what's going on. A real application would use a commercial IC that switches at a much higher frequency with a much smaller inductor. There really isn't much need to build discrete-part DC-to-DC converters any longer, other than for pedagogic purposes.

The converter stores energy from the panel in inductor L1, then dis-

System on Module Internet Appliance Engine

SoM-9307

- EP9307 ARM9 200Mhz CPU
- 3 Serial Ports & 2 SPIs
- Up to 40 Digital GPIOs
- 3 USB 2.0 Host Ports
- I2S Audio Interface
- 10/100 BaseT Fast Ethernet
- SD/MMC Flash Card Interface
- Up to 64 MB Flash & 128 MB RAM
- Graphic LCD Interface with 2D Acceleration
- Linux with Eclipse IDE & WinCE 6.0
- 8 12-Bit A/Ds & 4 16-Bit Timer/Counters
- Small, 144 pin SODIMM form factor (2.66 x 2.38")



2.6 KERNEL



The SoM-9307 uses the same small SODIMM form-factor utilized by other EMAC SoM modules, and is the ideal processor engine for your next design. All of the ARM9 processor core is included on this tiny board including: Touchscreen Interface, Flash, Memory, Serial Ports, Ethernet, I2S Audio Interface, PWMs, Timer/Counters, A/D, Digital I/O lines, and more. Like other modules in EMAC's SoM product line, the SoM-9307 is designed to plug into a custom or off-the-shelf Carrier board containing all the connectors and any additional I/O components that may be required. The SoM approach provides the flexibility of a fully customized product at a greatly reduced cost. Single unit pricing starts at \$150.

<http://www.emacinc.com/som/som9307.htm>

Since 1985
OVER
24
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.
EQUIPMENT MONITOR AND CONTROL

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

WIRELESS MADE SIMPLE®

BRING YOUR PRODUCT QUICKLY AND LEGALLY TO MARKET

RF Modules

Add INSTANT wireless analog / digital capability to your product.

Low-Cost TX, RX & TRX Modules



Long-Range Modules

Multi-Channel Modules



OEM Products

FCC PRE-CERTIFIED & ready to customize for your application.

Handheld TXs



Keyfob TXs

Function Modules



Feature Products

A closer look at Linx innovation

LOW-COST • LONG-RANGE TRANSCIVER

REMOTE CONTROL TRANSCODER IC

- Direct serial interface
- Low power consumption
- PLL-synthesized architecture
- RSSI and power-down functions
- Compact surface-mount package
- No external RF components (except antenna)



- Up to 8 inputs
- Bi-directional control
- Transmitter ID output
- Automatic confirmation
- Secure 2²⁴ possible addresses
- Latched and/or momentary outputs



800-736-6677

159 Ort Lane · Merlin, OR 97532

www.linxtechnologies.com

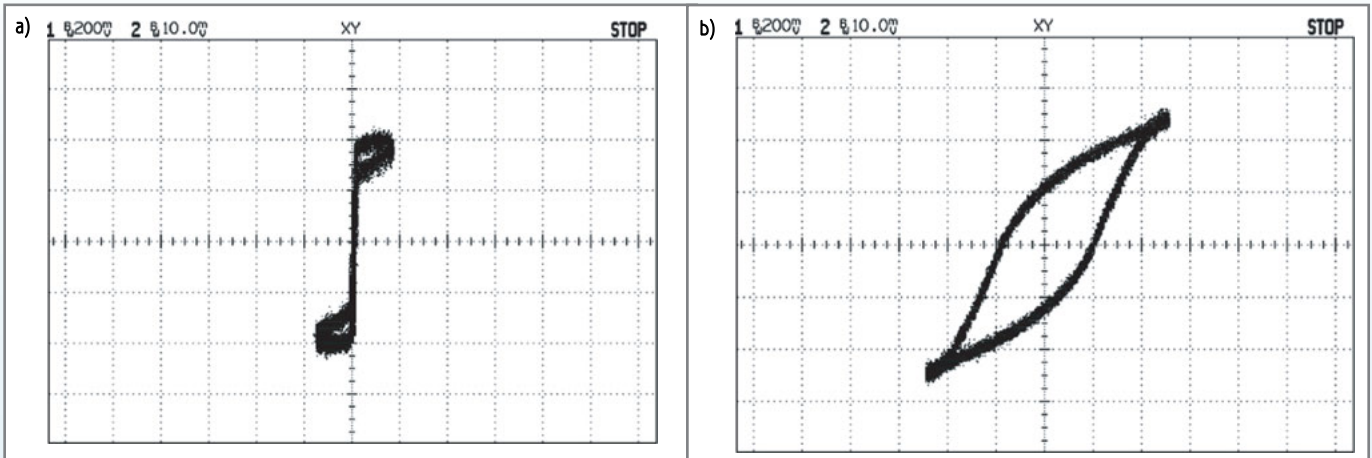


Figure 6a—The B-H curve for the common-mode choke inductor I used in the earlier circuit shows a very high inductance and saturates at a low DC current. **b**—The inductor in this circuit has a lower inductance, but doesn't saturate at a few hundred milliamps of DC current.

material couldn't support the DC current.

The two scope shots in **Figure 6** show the BH curves for the two inductors. I measured these using the same circuit as for the much larger transformer described in my February 2008 column ("Transformers," *Circuit Cellar* 211), but with a dozen turns of fine magnet wire as the secondary winding. Even without exact calibration, the general shape of the BH curves shows you what's happening.

The x-axis is proportional to H, the magnetizing force applied to the core, which is proportional to the current in the primary winding. The y-axis is proportional to B, the core flux density, which is proportional to the integral of the secondary voltage. The slope of the BH curve at any point is directly proportional to the inductance at the applied H.

The original inductor was a common-mode line filter choke, normally operated with balanced and opposing currents in its two windings. As a result, its core need not support a DC flux and, as shown in **Figure 6a**, it most certainly doesn't. Near the origin, the BH curve is nearly vertical, which means it has a very high inductance. For any nonzero H (and, thus, any non-canceling winding current), the core saturates with the slope of the BH curve becoming nearly horizontal.

I harvested the inductor shown in **Photo 1** from a scrapped switching power supply. As you can see by comparing it with the photo of the inductor in my April column, its core is much larger. While that does not guarantee tolerance of higher core flux density, it's a good sign.

The lower scope shot in **Figure 6** shows its more normal BH curve. The inductance is relatively low, but the core doesn't saturate abruptly.

The area inside an inductor's BH curve is proportional to its core power loss, so the larger core has a much higher loss. When you design a switching converter, particularly for a solar power project, reducing its core loss must be on your to-do checklist. Fortunately,

I can leave that as an exercise for the interested reader!

Figure 7 shows the inductor in action, switching between 100 mA and 200 mA limits; I used a bench power supply and an 8-Ω resistor to simulate a solar panel. The top trace is proportional to the inductor current, as measured by the MAX4372 high-side current amp, at 1 V = 100 mA. The lower trace is the base drive signal for Q2, the converter's current switch.

The current in this inductor now varies almost linearly while charging and discharging. The slopes also show good agreement with the applied inductor voltages, according to the familiar equation:

$$v = L \frac{di}{dt}$$

BOOSTER SETTINGS

After the firmware determines the MPP, it sets the PWM9 and PWM10 outputs to bracket the MPP current by about

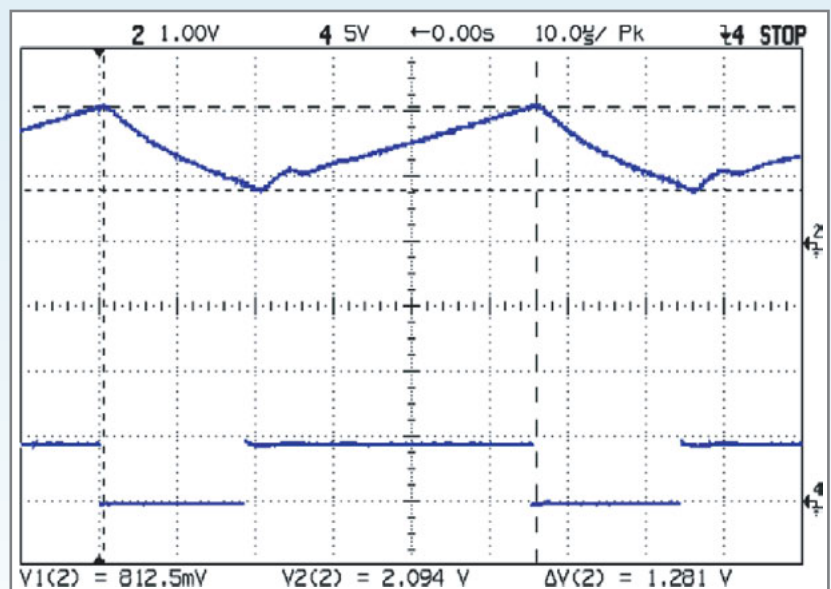


Figure 7—The inductor current now behaves as expected due entirely to the larger core. The slower transitions allow the MAX4372T output to accurately represent the inductor current.

±10%. When the current exceeds the high limit, Q2 turns off and the current begins dropping. When the current falls below the low limit, Q2 turns on and the current begins increasing. That cycle repeats until the firmware runs another load test and determines the new current setpoints.

The cross-connections between IC5A and IC5B form a set-reset flip-flop driving Q2 through two paralleled gates. The flip-flop remembers the transistor drive state between comparator transitions. You'll see the need for this as you probe the circuit: the comparators change state only briefly as the current goes beyond the setpoints, while the transistor must remain active or inactive until the current hits the other limit.

For the current limits shown in Figure 7, the booster runs at 18 kHz, which drives my golden-eared daughter out of my basement laboratory; it's completely inaudible to me. A much lower inductance, with a much higher di/dt, will run at a much higher frequency, but will also require much faster comparators and higher transistor drive. Remember, however, that the inductor core must support the flux required to drive the panel at its MPP: tiny surface-mount inductors probably won't suffice for an 80-W panel!

In addition to setting the inductor current limits, the firmware also sets the load current to match the power available from the panel. The boosted voltage on C5 can be any value you like, as long as it's higher than the panel voltage and less than 20 V (which is limited here by the Diecimila's 5-V analog inputs through the input scaling resistors). I set it to three times the panel voltage at the MPP or 15 V, whichever is less.

Then, knowing the desired output voltage and the available power, the firmware simply sets the load current in Q1 to their ratio. For example, with 540 mW available at a MPP of 5.4 V and 100 mA, the booster's output voltage will be 15 V and the load current will be:

$$36 \text{ mA} = \frac{540 \text{ mW}}{15 \text{ V}}$$

I implemented a very simple voltage-regulation algorithm, if you can glorify it with that name, which adjusts the load current once each second to maintain the output voltage as the solar input changes. After a minute of that the firmware performs another load test, recalculates the MPP, and picks the best load current for the new conditions.

Obviously, you can be far more clever, but ... that's another exercise!

INPUT PROTECTION

I used a MAX4372 to measure the panel current for the boost converter because it has a 28-V common-mode rating for the current-sense resistor, even with a logic-level power supply. After two of the chips mysteriously died on my bench, I sat down with the datasheet to find out why.

It turns out that the absolute maximum differential-mode voltage across the sense resistor must not exceed a mere 300 mV, most likely because that will turn on the internal ESD protection diodes and burn out the part. My solar panels can supply only a few hundred milliamps, which the sense resistor converts into half that number of millivolts and that seemed safe enough.

Perhaps you'll recall that I mentioned a "real" design must prevent the booster transistor Q2 from remaining on and driving the inductor core into saturation. I believe that's exactly what killed those two MAX4372s: after L1 saturates, the panel voltage divides between its internal resistance and the sense resistor, with Q2's saturation voltage accounting for only a few hundred millivolts.

Worse, I initially drove the circuit from a bench power supply through an 8-Ω resistor to simulate a solar panel in the darkness of my basement lab. Even though I had the supply current limit set to 300 mA, I think brief transients killed the MAX4372s before the limiter kicked in.

R36, R37, and D5 now hard-limit the voltage presented to the MAX4372 sense inputs and have successfully protected the chip from the rigors of my test setup. We live and learn.

CONTACT RELEASE

I don't have room to describe the process of extracting and formatting the relevant parts of the firmware's output to make the graphs, but the GNU/Linux command line will be your friend. Judicious application of `grep`, `cut`, `sort`, and `sed` to the log files will move you a long way toward the goal of valid input for `gnuplot`. ☒

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 additional files, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2009/227.

RESOURCES

Arduino project, <http://arduino.cc/>.

BH Curve information, E. Nisley, "Transformers," *Circuit Cellar* 211, 2008.

SOURCES

Gnuplot Plotting utility

Gnuplot | www.gnuplot.info

LTspice IV Design simulator

Linear Technology Corp. | www.linear.com/designtools/software/ltspice.jsp

Arduino Diecimila Microcontroller

SparkFun Electronics | www.sparkfun.com/commerce/product_info.php?products_id=666

SSI Controller for Linear-Position Sensors

For many applications, retrieving sensor readings requires an SSI controller. But it can be expensive. Robert presents an affordable solution for building your own SSI readout display for position sensors.

I like to develop innovative solutions for design problems rather than rely exclusively on expensive commercial products. For instance, I recently spent a weekend constructing a synchronous serial interface (SSI) readout display for position sensors (see [Photo 1](#)). In this article, I'll describe how I built the handy controller, which includes an LCD that enables stand-alone operation.

SENSOR EVALUATION

I am a firm believer in the following idea: If it is already invented, use it. I'm not a "roll-your-own" kind of engineer. If my application requires an RTOS or kernel, I will use a commercial product. If my controller requires an

embedded PC or SBC, I will shop around for one that meets my application's requirements. In my field, time to market is the key to success.

Recently, I was asked to evaluate a new linear position sensor, which is used for measuring displacement and velocity. Think of it as a high-precision electronic ruler. Linear position sensors—which you'll find in linear motor applications, CNC/PLC machines, servo-hydraulic closed-loop systems, and packaging machines—come in different flavors. Two popular but different technologies include magnetostrictive and glass scale encoders.

The MTS Systems Temposonics III R-Series linear position sensor in [Photo 2](#) uses magnetostrictive technology.

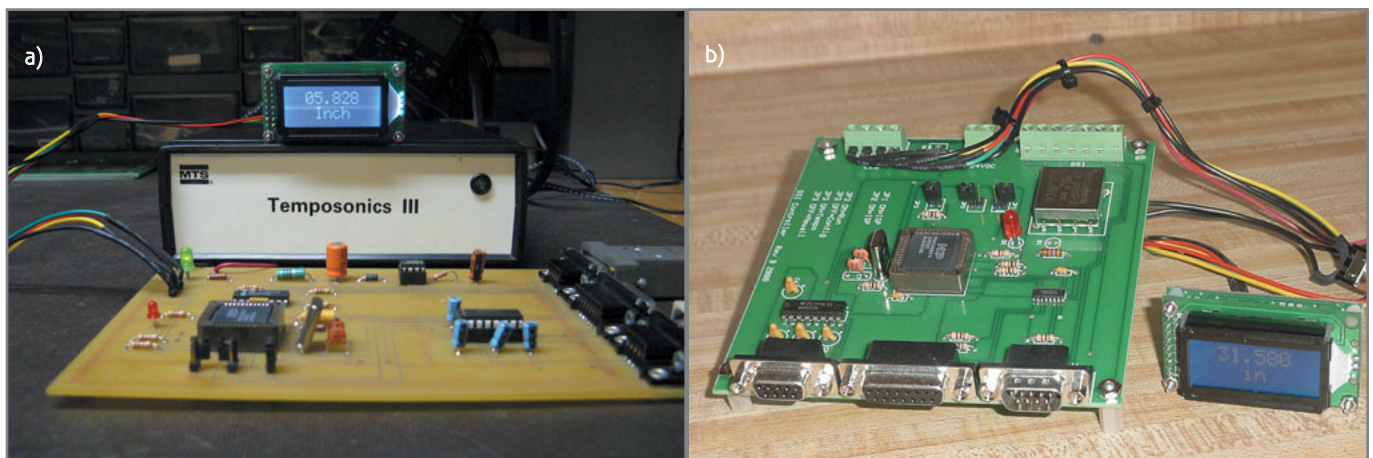


Photo 1a—The SSI controller supports both PC and Stand-Alone mode. In the latter, the sensor readings are displayed on the LCD screen. In PC Mode, the sensor data counts are sent to the PC's RS-232 serial port. **b**—An NXP P89C668 microcontroller sits at the heart of the design.



Photo 2—Unlike other sensors, the Temposonics III R-series linear position sensor can be reprogrammed in the field. Programmable options include: changing the sensor's resolution, output data format and the type of communication interface, synchronous or asynchronous.

These transducers convert the energy from a magnetic field into mechanical energy. This is accomplished by the interaction of two magnetic fields. One field is produced by a magnet that moves along the outside of the sensor's waveguide. The other field comes from a current pulse applied to the waveguide or tube. The interaction of the two magnetic fields produces a strain pulse. The magnet's position is calculated by measuring the elapsed time between the application of the interrogation pulse and the arrival of the resulting strain pulse.

Unlike the Temposonics sensor, glass scale encoders employ an optical glass and a photoelectronic reading head. The optical glass consists of a series of evenly distributed chrome lines. The reading head uses a photodetector to sense the chrome line patterns on the glass scale. As the reading head slides across the glass scale, it produces an electronic signal or pulse. For example, a 20- μm precision glass scale has 1,270 accurately placed chrome lines per inch. As the reading head is moved across the glass scale, the photoelectronic reading head detects each chrome line. When the reading head travels 1", it produces a count of 1,270.

Common among these devices is the SSI, which is a

widely used serial interface between an absolute position sensor and a controller. An SSI uses a clock pulse train from a controller to initiate a reading from the sensor. Position data is continually updated by the sensor and made available to the controller. **Figure 1** illustrates the basic requirements from an SSI controller (often referred to as digital readout display).

SSI OUTPUT

I was a little puzzled when my new linear position sensor arrived without an SSI controller. Instead I received a list of available readout displays. I didn't need anything elaborate, but I had a few requirements. The SSI controller had to be small and durable. I planned to use the device in the field, and it had to be able to withstand

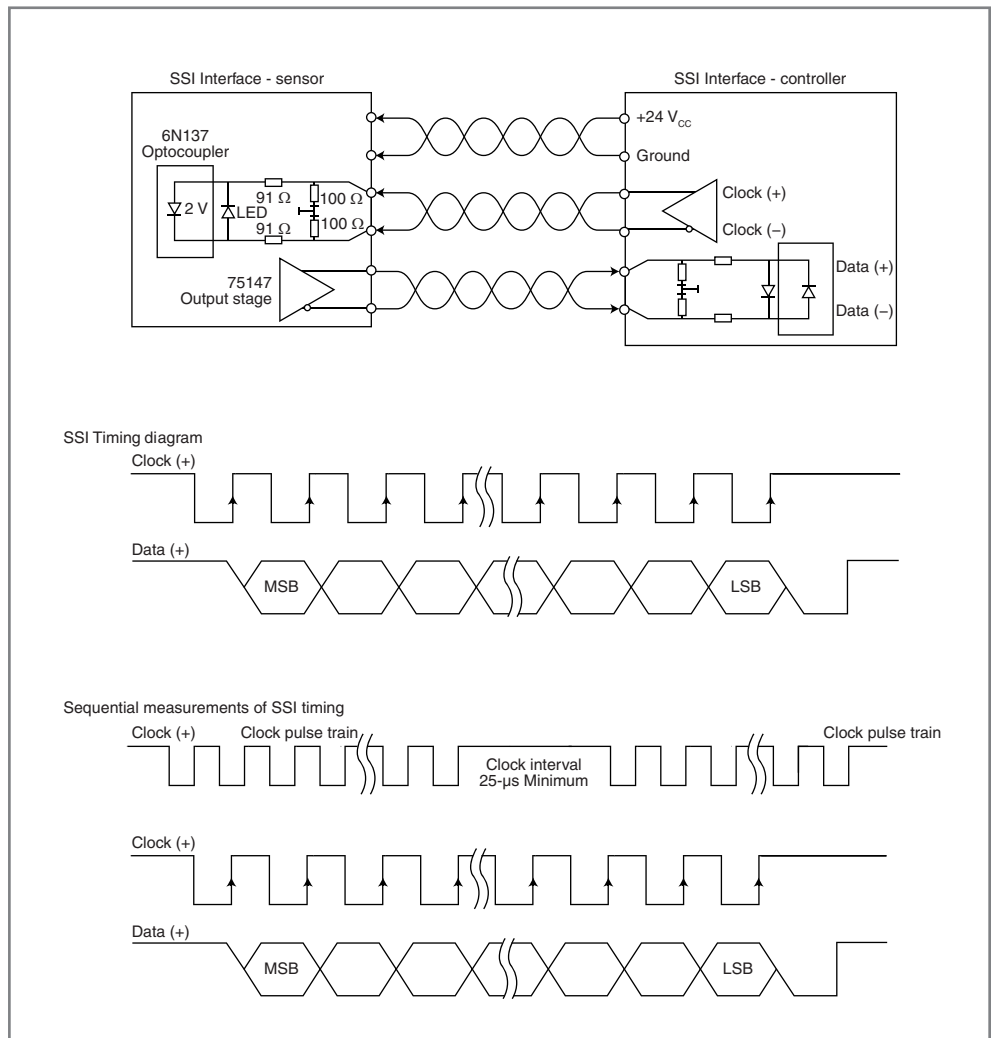


Figure 1—An SSI controller must comply with the timing diagram illustrated in the datasheet (MTS Systems, www.mtssensors.com/fileadmin/media/pdfs/R5SeriesSSI.pdf). It is responsible for supplying the 24 VDC to the sensor and must provide either an asynchronous or synchronous clock to the sensor. The Temposonics sensor requires a minimum dwell time of 25 μs between consecutive readings.

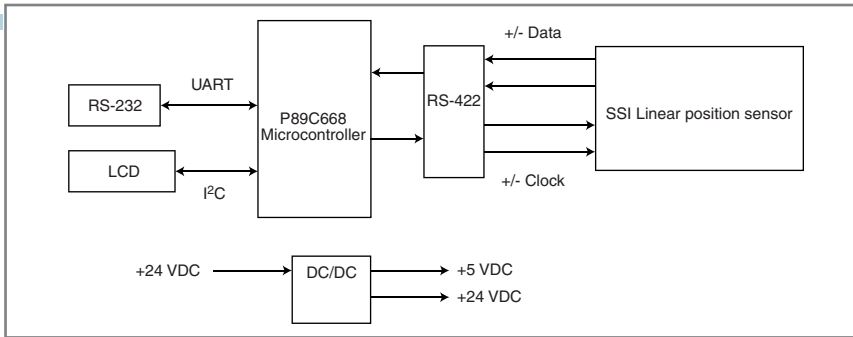


Figure 2—This is an SSI controller's basic configuration. All that is required are a clean and stable power source and a microcontroller for retrieving the data from the position sensor. Common among SSI sensors is the RS-422 interface, which provides the clock and data bus interface.

the punishment of being shipped from customer to customer. Additionally, I wanted the controller to be able to operate independently, as well as with a PC for data acquisition requirements.

One of my options was an Advanced Micro Controls NEXUS SSI controller (see Photo 3). This compact controller included support for linear, rotary, and laser position sensors. Each unit came standard with a four-channel SSI data interface and provided options for ControlNet, DeviceNet, Ethernet IP, and Modbus TCP/IP. Unfortunately, it lacked a display. This meant I would need a PC to operate the device. Additionally, my position sensor

et application. I decided it was time to “roll-my-own” SSI controller.

Photo 3—The Nexus SSI Interface Controller supports up to four SSI devices and has options for ControlNet, DeviceNet, Ethernet IP, and Modbus TCP/IP. Unfortunately, it lacks an LCD. This SSI controller requires a PC to view readings from the position sensor.



Cellular and GPS capable
Data Mover



- Flash file System
- 4 Chan 12-Bit A/D
- 1MB SRAM
- 512KB FLASH
- 4 Isolated Inputs
- 4 Hi Current Outputs
- 2 External RS-232
- 2 Internal RS-232
- Bat Backed Clk/Cal
- Cell Modem Option
- Internal GPS Option
- Metal Case Option
- 1ma Standby Option

It's easy and cost-effective to do mobile or solar-powered data collection and asset monitoring with the JK micro's **Data Mover**. With the ability to integrate a Cellular Modem, GPS and DOS-based embedded controller in a single rugged enclosure, you can capture and transmit your data quickly, easily and at low cost. Inexpensive development kits including Borland C/C++ and PowerBasic are available now. Call or email us for more details.

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

JK microsystems

Got Serial, Need Network?



Bluetooth Qty 1 \$145

Ethernet Qty 1 \$99

Wireless Qty 1 \$199

Volume Discounts Available

gridconnect™
www.gridconnect.com
 +1 800 975-4743

Listing 1—The firmware simply bit-bangs the Temposonics to acquire the 25-bit data count. After each high-to-low transition, data is retrieved from the sensor. The BCD_32() processes the readings and then they are sent to the LCD and to the RS-232 serial port.

```

;*****
; READ_TEMPOSONICS():
; This routine communicates
; directly with the Temposonics III,
; provides the required clock
; pulses and reads the 25-bit
; binary count result.
; The 25-bit binary result is
; stored in (MSB)R4,R5,R6,
; R7(LSB) which is passed to
; the BCD_32() function.
;*****
READ_TEMPOSONICS:
    SETB    SSI_CLK1 ; CLK HIGH
            FOR 25uS DWELL TIME
    MOV     R1,#250
    NOP
    NOP
    DJNZ   R1,$
    CLR    SSI_CLK1 ; CLK LOW
    MOV    ANSWER,#0
    MOV    A,#0
    CLR    C
    MOV    R1,#1 ; 1ST READING
            MSB OF TEMPO. RDNG.

MSB:   SETB    SSI_CLK1
    CLR    SSI_CLK1
    MOV    C,SSI_DATA1
    RLC    A
    DJNZ   R1,MSB
    MOV    ANSWER,A ; SAVE RDNG
            IN ANSWER.

    MOV    R1,#8 ; READ NEXT 8
            BITS OF TEMPO. RDNG.
HBYTE: SETB    SSI_CLK1
    CLR    SSI_CLK1
    MOV    C,SSI_DATA1
    RLC    A
    DJNZ   R1,HBYTE
    MOV    ANSWER+1,A ; SAVE
            RDNG IN ANSWER+1.

    MOV    R1,#8 ; READ NEXT
            8 BITS OF TEMPO. RDNG.
MBYTE: SETB    SSI_CLK1
    CLR    SSI_CLK1
    MOV    C,SSI_DATA1
    RLC    A
    DJNZ   R1,MBYTE
    MOV    ANSWER+2,A ; SAVE
            RDNG IN ANSWER+2.

    MOV    R1,#8 ; READ LAST 8
            BITS OF TEMPO. RDNG.
LBYTE: SETB    SSI_CLK1
    CLR    SSI_CLK1
    MOV    C,SSI_DATA1
    RLC    A
    DJNZ   R1,LBYTE
    MOV    ANSWER+3,A ; SAVE
            RDNG IN ANSWER+3.

;
; SETB    SSI_CLK1
;
; RET

```

After a weekend of soldering and coding, I designed and built my SSI interface controller for less than \$100. The controller includes an LCD screen that enables the unit to operate in Stand-Alone mode. It also includes an RS-232 interface for sending the sensor readings to a PC for data acquisition.

For testing, I purchased a Temposonics III R-Series linear position sensor. It has an SSI, a resolution of 0.001 mm, and stroke ranges from 50 to 10,060 mm. Because it is field-programmable, I can configure the sensor's SSI parameters. I can set the data length (24 or 25 bit), set the data format (binary or grayscale), set the resolution range (from 0.001 to 0.005 mm), select either synchronous or asynchronous measurement, and select either speed or position measurement.

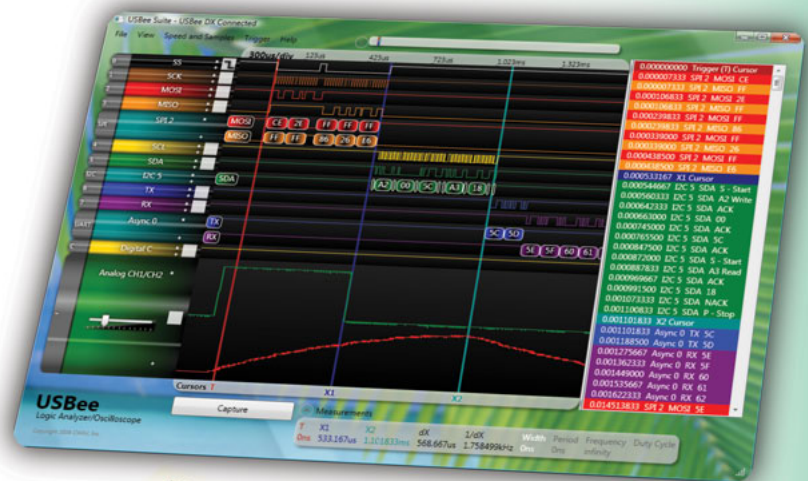
I also purchased the field-programming kit, which enabled me to reconfigure the sensor's parameters and thoroughly test the digital readout display. The kit came with a serial

converter, power supply, cables, and software. After running the installation CD, I connected the Temposonics and launched the MTS R-Series SSI Utility application. The software established a connection with the sensor and then displayed its settings in a user-friendly format. At that point, I could choose to change any of the available parameters. After clicking the Apply Changes command button, the new parameters were written to the sensor's on-board processor.

CONTROLLER HARDWARE

The SSI controller's configuration is depicted in Figure 2. The controller must provide the 24 VDC and system clock to the SSI sensor. An RS-422 dual-differential line driver and receiver would satisfy the ±Data line interface requirements from the sensor.

My first prototype used a 24 VDC at 500 mA wall transformer for the power source. The 24 VDC would satisfy the power requirements for



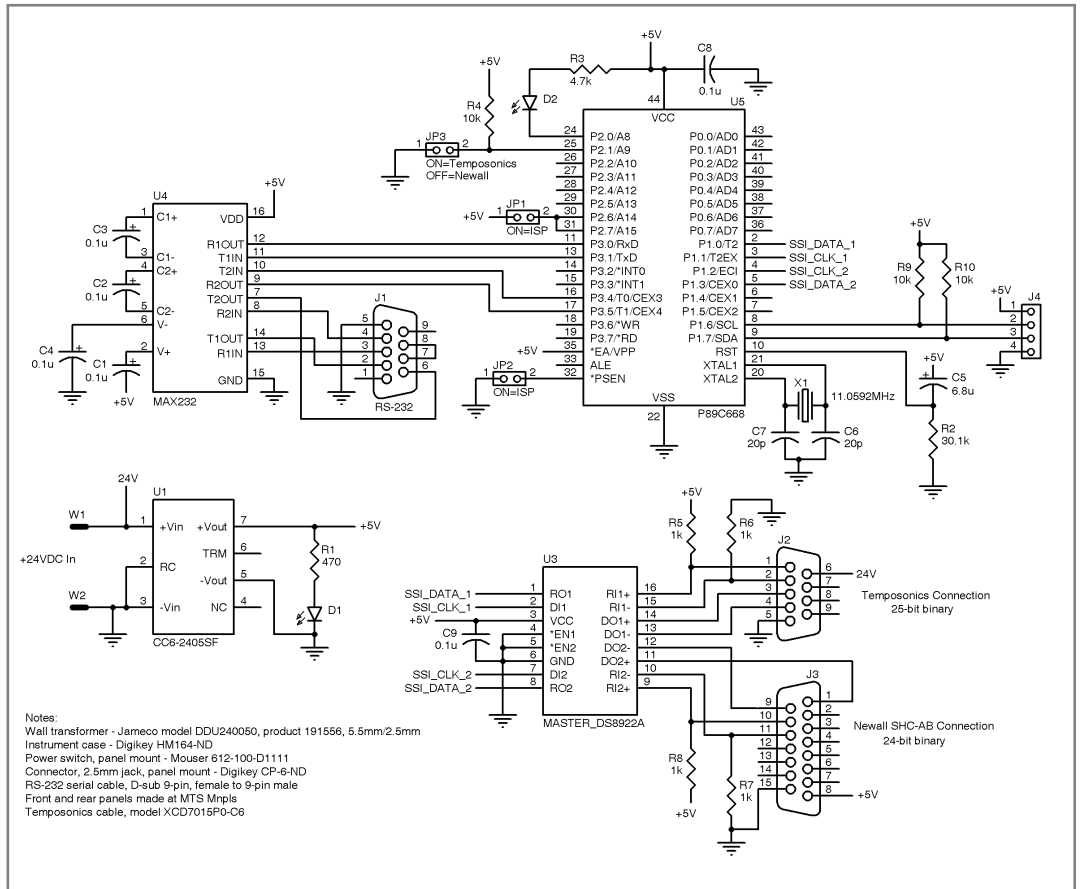
See It & Solve It

with
USBee Test Pods

- Logic Analyzers
- Oscilloscopes
- Signal Generators
- Protocol Analyzers
 - I2C, SPI, ASYNC, CAN
 - 1-Wire, PS/2, USB
 - I2S, SMBus, Serial
- Configurable
- Programmable
- High Speed USB
- PC-Based
- And start at \$139

www.USBee.com  USB based Electrical Engineer

Figure 3—To make a truly compatible SSI controller, you can add a few jumpers and extra connectors. The firmware can be written to detect which sensor is connected. I included both a nine- and 15-pin DSUB to the SSI controller.



Easy Embedded Linux

\$169
Qty 1

- 16MB FLASH / 32MB RAM
- 200Mhz Arm9 CPU
- 16 Digital I/O
- Watchdog
- 10/100 Ethernet
- Battery backed Clock/Calendar
- Audio In/Out
- 2 USB
- 2 Serial Ports

We brought you the world's easiest to use DOS controllers and now we've done it again with Linux. The **OmniFlash** controller comes preloaded with Linux and our development kit includes all the tools you need to get your project up and running fast.

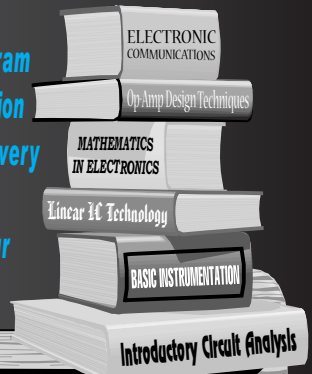
Out-of-the-box kernel support for USB mass storage and 802.11b wireless, along with a fully integrated Clock/Calendar puts the **OmniFlash** ahead of the competition.

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

JK microsystems

PROFESSORS

The Circuit Cellar college program puts quality engineering information in the hands of your students every month. Sign up now to get Circuit Cellar distributed to your class this semester.



To update your professor account or to find out more about our college program, visit www.circuitcellar.com/products/collegeprogram/

Listing 2—The LCD is connected to the microcontroller through an I²C hardware interface.

```
;*****;  
; LCD_DISPLAY_INCH() - Writes "Inch" on Row 2 Column 3  
;  
;*****  
LCD_DISPLAY_INCHES:  
  
MOV    S1CON,#040H    ;Enable I2C - no acknowledgements will be generated  
SETB  S1CON.5        ;STA = 1  
JB    S1CON.3,$      ;Wait for interrupt  
SETB  S1CON.5        ;STA = 1  
MOV    SIDAT,#254    ;cmd to LCD  
JB    S1CON.3,$      ;Wait for interrupt  
;  
MOV    S1CON,#040H    ;Enable I2C - no acknowledgements will be generated  
SETB  S1CON.5        ;STA = 1  
JB    S1CON.3,$      ;Wait for interrupt  
SETB  S1CON.5        ;STA = 1  
MOV    SIDAT,#71     ;cmd number to LCD  
JB    S1CON.3,$      ;Wait for interrupt  
;  
MOV    S1CON,#040H    ;Enable I2C - no acknowledgements will be generated  
SETB  S1CON.5        ;STA = 1  
JB    S1CON.3,$      ;Wait for interrupt  
SETB  S1CON.5        ;STA = 1  
MOV    SIDAT,#3      ;COLUMN = 3  
JB    S1CON.3,$      ;Wait for interrupt  
;  
MOV    S1CON,#040H    ;enable I2C - no acknowledgements will be generated  
SETB  S1CON.5        ;STA = 1  
JB    S1CON.3,$      ;WAIT FOR INTERRUPT  
SETB  S1CON.5        ;STA = 1  
MOV    SIDAT,#2      ;ROW = 2  
JB    S1CON.3,$      ;WAIT FOR INTERRUPT  
;  
MOV    S1CON,#040H    ;Enable I2C - no acknowledgements will be generated  
SETB  S1CON.5        ;STA = 1  
JB    S1CON.3,$      ;WAIT FOR INTERRUPT  
SETB  S1CON.5        ;STA = 1  
MOV    SIDAT,#049H    ;Letter "I"  
JB    S1CON.3,$      ;WAIT FOR INTERRUPT  
;  
MOV    S1CON,#040H    ;Enable I2C - no acknowledgements will be generated  
SETB  S1CON.5        ;STA = 1  
JB    S1CON.3,$      ;WAIT FOR INTERRUPT  
SETB  S1CON.5        ;STA = 1  
MOV    SIDAT,#06EH    ;Letter "n"  
JB    S1CON.3,$      ;WAIT FOR INTERRUPT  
;  
MOV    S1CON,#040H    ;Enable I2C - no acknowledgements will be generated  
SETB  S1CON.5        ;STA = 1  
JB    S1CON.3,$      ;WAIT FOR INTERRUPT  
SETB  S1CON.5        ;STA = 1  
MOV    SIDAT,#063H    ;Letter "c"  
JB    S1CON.3,$      ;WAIT FOR INTERRUPT  
;  
MOV    S1CON,#040H    ;Enable I2C - no acknowledgements will be generated  
SETB  S1CON.5        ;STA = 1  
JB    S1CON.3,$      ;WAIT FOR INTERRUPT  
SETB  S1CON.5        ;STA = 1  
MOV    SIDAT,#068H    ;Letter "h"  
JB    S1CON.3,$      ;WAIT FOR INTERRUPT  
RET
```

the Temposonics sensor. The Linear Technology LT1107-5 DC/DC converter would be used to produce 5-VDC at 300 mA. This would have been suitable power for the microcontroller, LCD, and the on-board control logic. However, I discovered that most linear position sensors required at least 5 VDC at 350 mA. This would require a slight change to the power supply circuit.

I replaced the LT1107-5 with a TDK-Lambda CC6-2405SF-E DC/DC converter, which boasts a 5 VDC at 1.2 A output. It has a compact footprint and supports both SMT and through-hole versions. This made laying out the PCB easy, and it eliminated the extra passive components.

An NXP P89C668 microcontroller is the heart of the SSI controller (see [Figure 3](#)). A variant of the 8051 microcontroller family, it comes packed with a lot of goodies, including on-board flash memory and an I²C hardware interface. In addition, it supports in-circuit serial programming (ISP), which is a useful feature for field updates and firmware development. Remember the old days when you had to remove the 8751 and erase it under ultraviolet light for 20 minutes? Firmware development took forever. Today, thanks to flash memory and ISP technology, you can be more productive.

I wanted the digital read-out display controller to operate in both Stand-Alone and PC modes. In the former, you could connect the

linear sensor and view the output displacement readings directly on the LCD screen. The PC could be optional and you could use it for data acquisition or converting the sensor data to velocity readings.

Although I hate to admit it, I spent more time searching for the perfect LCD than it took to develop the firmware. This was a low-budget project, so I had to consider cost, display size, power requirements, contrast, and, lastly, the type of interface. After several hours of navigating the Internet, I ended up at Matrix Orbital's website, where I found the LCD0821 display, which had all the right ingredients for my application. This tightly packed LCD had a small footprint measuring 58 mm × 37 mm × 25 mm (see [Photo 4](#)). It had a back-light display that could be controlled via software. It required only 5 VDC at 30 mA, and provided both an I²C and RS-232 communication interface. All of this for less than \$40!

FIRMWARE

The P89C668 microcontroller is responsible for communicating with the SSI linear-position sensor. It provides the necessary clock pulses and reads the 25-bit binary data code. The binary output code is a data count value representing the sensor's position or displacement. The position sensor is programmed to produce a count, which represents the amount of travel. For example, the Temposonics has a resolution of 0.005 mm. This equates to 5,080 counts per inch. To calculate a displacement measurement from the device, you would take the binary count value/5,080:

$$\begin{aligned} \text{Temposonics resolution} &= \\ \frac{1}{0.005} &= 200 \text{ counts/mm} \\ 200 \times 25.4 &= 5,050 \text{ counts/inch} \end{aligned}$$

The firmware is written in assembly language, and I used the Keil IDE MicroVision compiler. If assembly language is not your preferred programming language, the Keil IDE supports C language. For simple projects that require only a few pages of

Listing 3—The firmware is responsible for responding to the I²C status codes. The status code must be used to vector to an interrupt service routine (ISR).

```

;*****
; Received I2C ISR
;
; Interrupt Service Routine(s) for all I2C Interrupt conditions.
;*****
I2C_ISR::START CONDITION TRANSMITTED. STATUS CODE 08H
    MOV     R7,S1STA
    CJNE   R7,#08H,STATUS_CODE_10H
    CLR    S1CON.5      ;STA = 0
    CLR    A
    MOV    A,#050H      ;LCD ADDRESS
    ANL   A,#0FEH
    MOV    S1DAT,A
    CLR   S1CON.3      ;SI = 0
    RETI

STATUS_CODE_10H::REPEATED START CONDITION TRANSMITTED
;
    CJNE   R7,#010H,STATUS_CODE_18H
    CLR    S1CON.5      ;STA = 0
    CLR    A
    MOV    A,#050H      ;LCD ADDRESS
    ANL   A,#0FEH
    MOV    S1DAT,A      ;SEND DEVICE ADDRESS AND START SIGNAL
    RETI

STATUS_CODE_18H::SLAVE ADDRESS + WRITE TRANSMITTED - ACK RECEIVED
;
    CJNE   R7,#018H ,STATUS_CODE_20H
    CLR    S1CON.5      ;STA = 0
    CLR    S1CON.3      ;SI = 0
    RETI

STATUS_CODE_20H::SLAVE ADDRESS + WRITE TRANSMITTED - NO ACK RECEIVED
;
    CJNE   R7,#020H,STATUS_CODE_28H
    CLR    S1CON.5      ;STA = 0
    CLR    S1CON.3      ;SI = 0
    RETI

STATUS_CODE_28H::DATA BYTE TRANSMITTED - ACK RECEIVED
;
    CJNE   R7,#028H,STATUS_CODE_30H
    SETB   S1CON.4      ;STO = 1
    CLR    S1CON.3      ;SI = 0
    RETI

STATUS_CODE_30H::DATA BYTE TRANSMITTED - NOT ACK RECEIVED
;
    CJNE   R7,#030H,STATUS_CODE_38H
    SETB   S1CON.4      ;STO = 1
    CLR    S1CON.3      ;SI = 0
    RETI

STATUS_CODE_38H::ARBITRATION LOST OR DATA BYTES
;
    CLR    S1CON.3      ;SI = 0
    RETI

```

Status code	Status of I ² C bus
08H	A start condition has been transmitted.
10H	A repeated START condition has been transmitted.
18H	SLA+W has been transmitted. ACK has been received.
20H	SLA+W has been transmitted. NOT ACK has been received.
28H	Data byte in S1DAT has been transmitted. ACK has been received.
30H	Data byte in S1DAT has been transmitted. NOT ACK has been received.
38H	Arbitration lost in SLA+R/W or data bytes.

Table 1—The I²C interface is interrupt-driven. The application is responsible for responding to the I²C status codes. Creating a look-up table in the firmware allows the microcontroller to be responsive when activity is detected on the I²C bus.

code, I prefer using assembly. Let's examine the source code in [Listing 1](#).

The timing diagram illustrates that data is shifted out when the sensor receives a clock pulse from the controller. Data is valid on each high-to-low clock transition. To acquire the 25-bit data count from the sensor, the controller must provide 25 clock cycles. For asynchronous communication, bit-banging is a viable option. It is important to consult the sensor's datasheet. Some position sensors support only synchronous communication. Thus, trying to communicate with it in the wrong mode could result in data loss.

As you can see in the source code, I created four loops for acquiring the 25-bit result. The first loop is a single clock to acquire the first MSB. The next three loops each consist of eight high-to-low clock transitions. The 25-bit result is stored in the following format: MSB – bit 25; HBYTE – bit 24-17; MBYTE – bit 16-8; and LBYTE – bit 7-0.

The firmware contains two binary coded decimal functions, `BCD_24()` and `BCD_32()`. These functions are used to convert a 24-bit binary and 32-bit binary numbers to BCD format. Once converted, the result is sent to serial port.

For the LCD, a little more work has to be performed with the BCD result. Displaying a count value on the LCD screen would not be very useful to the user. To provide a meaningful output, the BCD value is converted to engineering units—inches, in this case. To accomplish this, the firmware utilizes a few integer math routines to complete the conversion:

$$\text{Engineering units (inches)} = \frac{\text{BCD Value}}{5,080}$$

This result represents the sensor's absolute position and is ready to be sent to the LCD.

In previous projects, I implemented the I²C protocol in firmware. This time, I decided to take advantage of the I²C hardware interface on the P89C668 microcontroller to communicate with the LCD. Before examining this part of the source code, I will briefly review the highlights of the I²C protocol as it pertains to the P89C668

microcontroller.

The I²C (Inter-IC Bus) interface uses two wires (SDA and SCL) to transfer data between devices connected to the bus. The P89C668 uses four special function registers: S1CON (control register), S1STA (status register), S1DAT (data register), and S1ADDR (slave address register). The CPU can operate in either Master or Slave mode. In this case, the CPU is configured to be the master. The master device generates all of the serial clock pulses and the START and STOP conditions. To send data to a slave device (the LCD screen in this case), the master first transmits a byte containing the slave's address and the data direction bit. After each byte is transmitted, an acknowledgement bit is received while the START and STOP conditions are output to indicate the end of the serial transfer.

[Listing 2](#) illustrates how to write the word "Inch" on the second row of the LCD screen. [Table 1](#) displays the Master Transmitter response table. After a byte has been sent to the slave, an interrupt is generated and it is up to the application to respond accordingly to the response codes. [Listing 3](#) illustrates how to implement this in the source code.

DIGITAL READOUT APP

It's convenient to have the linear-position sensor display its readings on the LCD screen. Data collection or manipulating the sensor readings to display velocity/speed is work best left to the PC. The digital readout display controller automatically sends the sensors data count to the serial port.

All that is left to do is develop an application to listen on the computer's serial port and present the data to the user. There are several tools available for developing a user interface application. For data acquisition applications, I prefer using either National Instruments's LabVIEW or Microsoft's Visual Basic. LabVIEW has truly revolutionized the ease of developing test and measurement applications. The LabVIEW IDE comes with a suite of tools that enables you to create a professional look and feel. Creating real-time chart recorders and displaying panel meters for instrumentation readouts are standard tools included within the LabVIEW environment.

To complete the SSI controller, I created a simple GUI in Visual Basic (see [Photo 5](#)). The application enables you to select the type of readout, either actual encoder counts or engineering units. In the Counts/Inch section, you can enter the linear-position sensor's resolution.



Photo 4—The LCD0821 has a 2 × 8 line alphanumeric liquid crystal display. It supports both RS-232 and an I²C communication interface.

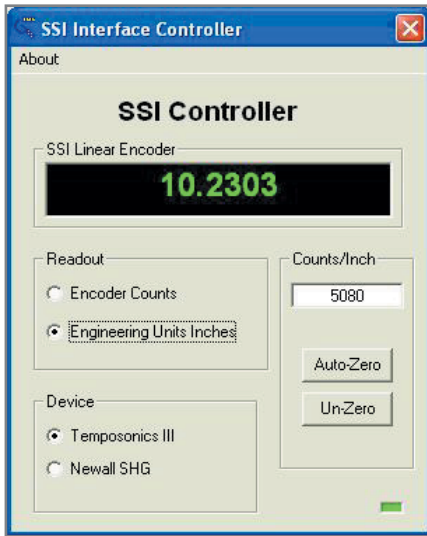


Photo 5—The SSI controller is responsible for communicating with the sensor. All data is sent to the RS-232 port. Creating a GUI application is effortless. The GUI simply listens on the serial port and can manipulate the data and present it to you in engineering units.

wireless modules are relatively inexpensive, and many wireless technology companies have made incorporating this equipment painless. When you start your next project that requires an SSI sensor, remember to consider the following requirements: power consumption, data speed options, and the connection type.

or PC solution. My design accommodates both options for a fraction of the cost.

If you're like me, you tend to look for off-the-shelf solutions for your projects. However, there are times when your budget will dictate how you tackle a specific task. And, yes, sometimes this means reinventing the wheel. If it can be built for less and it provides a smoother ride, then it makes sense. 📌

I chose the RS-232 interface for this design for the reasons of cost and convenience. But it would be easy to implement the design to support either USB or an Ethernet interface. Actually, I've been considering implementing a wireless solution. Today,

Robert Bowen (robert.bowen@mts.com) holds an Associate's degree in electronic technology and electronic engineering and has worked for AT&T Consumer Products designing test equipment. Currently, Robert works as a senior field service engineer for MTS Systems Corp., where he designs automated calibration equipment and develops testing methods for customers involved in the material and simulation testing fields. In his spare time, he enjoys amateur radio (call sign N0PAU), and he recently started playing the guitar with his 8-year-old son, Elijah, and his 11-year-old daughter, Madison.

The application uses this value to calculate the displacement readings properly from the sensor. Finally, the Device section provides a convenient way to support multiple sensors. This is useful if each sensor's output is scaled differently. The application will know how to handle the data accordingly.

Communicating with the position sensor begins when power is applied to the SSI controller. The sensor's readings, in engineering units, are automatically sent to the LCD, while the sensor's data count value is simultaneously sent to the RS-232 serial port. If a PC is connected, you may choose to view either the data count or the sensor's absolute position.

A LOW-COST SOLUTION

In this article, I presented a low-cost solution for building an SSI readout display for position sensors. There are many different sensing technologies. Each has its advantages. Typically, the environment and resolution required for an application will dictate the correct choice. Selecting the proper sensor for your application is only half the battle. Retrieving the sensors readings requires an SSI controller. While there are several manufacturers that provide commercial products, they are often expensive and require modifications for connecting the sensor to the SSI controller. Additionally, I found that most commercial SSI controllers provide either a stand-alone

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/227.

RESOURCES

Matrix Orbital, "LCD0821 Technical Manual," Rev. 2.0, www.matrixorbital.ca/manuals/LCDVFD_series/LCD0821/LCD0821_rev_20.pdf.

MTS Systems Corp. (Sensors Division), "Magnetostriction: Principle, Technology, and How It Works," www.mtssensors.com/technology/how-magnetostriction-works/index.html.

SOURCES

MicroVision Compiler

Keil | www.keil.com

LT1107-5 DC/DC Converter

Linear Technology | www.linear.com

LCD0821 LCD

Matrix Orbital | www.matrixorbital.com

Temposonics III R-Series linear position sensor

MTS Systems Temposonics | www.mtssensors.com

P89C668 Microcontroller

NXP | www.nxp.com

CC6-2405SF-E DC/DC converter

TDK-Lambda Americas, Inc. | www.us.tdk-lambda.com

Sweet!

Introducing the MiniCore™ Series of Networking Modules

Smaller than a sugar packet, the Rabbit® MiniCore series of easy-to-use, ultra-compact, and low-cost networking modules come in several pin-compatible flavors. Optimized for real-time control, communications and networking applications such as energy management and intelligent building automation, MiniCore will surely add sweetness to your design.

- **Wireless and wired interfaces**
- **Ultra-compact form factor**
- **Low-profile for design flexibility**
- **Priced for volume applications**

Wi-Fi and Ethernet Versions



MiniCore Module Development Kits

From **\$49** Limited time offer.



Buy now at: **1.888.411.7228** • rabbitwirelesskits.com

1.888.411.7228
rabbitwirelesskits.com
2900 Spafford Street, Davis, CA 95618



MP3P DIY KIT, Do it yourself

(Include Firmware Full source Code, Schematic)

• myPIC



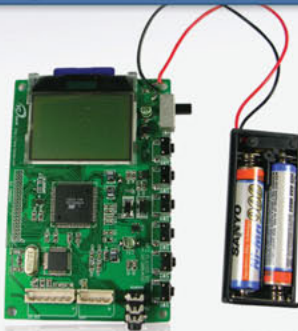
Only

\$160 **\$220**

qty 100

qty 1

• myWave (MP3 DIY KIT SD card Interface)



Only

\$150

qty 100

\$200

qty 1

• myAudio (MP3 DIY KIT IDE)

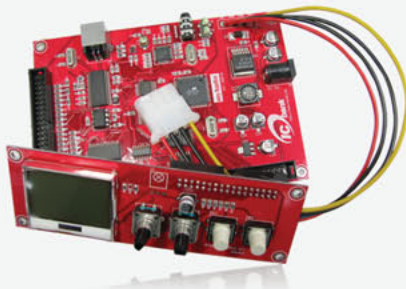
Only

\$180

qty 100

\$220

qty 1



Powerful feature

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

Specification

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

Powerful feature

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

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

Powerful feature

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

Autonomous Vehicle Design

Embedded Systems, Sensor Technology, and Motor Control

Autonomous vehicles and mobile applications are becoming increasingly incorporated in the military, industrial, and healthcare industries. This autonomous vehicle design implements the full potential of embedded, sensor, and modular electronic technologies.

Autonomous vehicles are sprouting up everywhere, from military bases (e.g., unmanned surveillance systems) to luxury car dealerships (e.g., self-parking cars) to home appliance stores (e.g., robotic vacuums). With such designs in mind—and inspired by the DARPA Grand Challenge, where autonomous cars race a long-distance course through the desert—our design team at Camosun College created an autonomous vehicle called the Interurbanator (see [Photo 1](#)).

The recent development of satellite navigation and video imaging enables the development of inexpensive autonomous vehicles. We didn't have the budget to implement LIDAR or gas-powered technologies, so we used a small hobby vehicle as our chassis, which includes two small 12-VDC motors and hollow plastic wheels. We used mostly off-the-shelf and recycled components for this do-it-yourself, modular-style project.

In this article, we will explain how we built our autonomous vehicle system, which features a modular Linux "brain" that runs a custom C++ program, a differential GPS and algorithm, a custom serial protocol between microcontrollers, low-level sensors, and a webcam laser rangefinder. We used Open Source Computer Vision (OpenCV) for the image-processing aspect of this application.

SYSTEM OVERVIEW

The vehicle uses many different sensors to reach its destination (see [Figure 1](#)). At the center of everything is

the CPU, which is a Pentium 4 computer running a Linux operating system. This gives us the ability to time slice and run multiple processes at the same time. The vehicle can approach its destination by avoiding all the objects in its way. When an object is detected and avoided, the system reroutes itself to the target.

We implemented infrared (IR) range sensors, an ultrasonic sensor, and webcam laser rangefinding to detect obstacles. From the CPU's perspective, each group of sensors has different priorities. The IR range sensors detect surfaces using analog voltage to represent range to a surface. The IR sensors detect only objects 30 cm away. If something is that close to the vehicle, action needs to happen immediately.

The webcam laser rangefinding system is designed for avoiding objects. Although it is accurate up to 6 m, it is mounted to the front of the vehicle and can't see what's behind it. The ultrasonic sensor is mounted to a stepper motor so it can rotate 360° to sense objects in all directions.

The vehicle's drive train consists of two 12-VDC motors driven by a pulse-width-modulated (PWM) signal. The wheel RPMs are then picked up by two infrared sensors in the motor gear boxes. From this we can choose multiple speeds and keep track of how far the vehicle has traveled. A Firgelli Technologies linear actuator enables steering control. This is a simple and easy replacement for a steering wheel. A digital compass provides steering feedback.

We used microcontrollers for low-level sensor data acquisition. This includes IR sensors, stepper motor control, ultrasonic sensors, digital compass, steering control, and driver motor control. Each MCU collects and controls each device at all times. The computer gives a command or requests data stored in the MCU. We designed a serial protocol for this communication to work. This protocol enables each thread in the brain to access each device without interfering with any other device. **Figure 1** is a block diagram of the system.

CENTRAL PROCESSING

Running a Linux operating system enables us to run multiple processes at one time. We use individual threads to divide up the brain's tasks. Threads are used for computing calculations, obstacle avoidance, and navigation. All autonomous vehicles require some form of obstacle avoidance and navigation control. We use a Linux kernel running a C++ program to control the autonomous vehicle. To simplify code management, we produced a series of files. Each file contains the relevant code and variables required to control and read information from a particular system. By using multiple threads, we allow the different parts to run and leave timesharing for the computer.

Each thread communicates with the others through a series of global structures. Primary avoidance is left to the webcam laser rangefinder. IR sensors manage secondary avoidance. The ultrasonic rangefinder isn't particularly accurate; therefore, it is only used as a secondary system to the webcam laser rangefinder. Nevertheless, its versatility is useful. Mounted on a stepper motor, it can see 360° around the vehicle.

SERIAL PROTOCOL

Figure 2a is the general communication format. **Figure 2b** is a special packet for IR/sonar data. FF is the header used for the easy identification of packets. Length H is one of two bytes given for the length of the packet. Length L is the second of two length bytes. TYPE is for device identification, so the brain can tell different devices apart. DATA contains information such as the sensor number, angle, and distance. CHKSUM is the checksum for error checking. **Figure 2c** is a motor control packet. Note that FF, Length H, Length L, TYPE, and CHKSUM are the same as they are for the IR/sonar data packet. Speed sets the speed of the motors. Direction sets the motors' direction (forward or back). Brake sets the motors to brake.

The vehicle has numerous integrated subsystems. To enable communication, we needed a standard method of passing data and commands between devices. To handle this task, we used a serial connection and a custom protocol. Early on, we decided a master/slave relationship



Photo 1a—Take a look at the Interurbanator in action. Here it's traversing fairly rugged terrain. **b**—IR range sensors enable the vehicle to avoid obstacles like people (seen here) and stationary objects. **c**—The design features two 12-VDC motors and a simple chassis. Everything fits nicely inside.

would work best and that the brain would be the master and each of the subsystems would be designed as slaves.

To pass data or a command between the brain and one of the microcontrollers, the brain initiates communication. The microcontroller then replies with either the

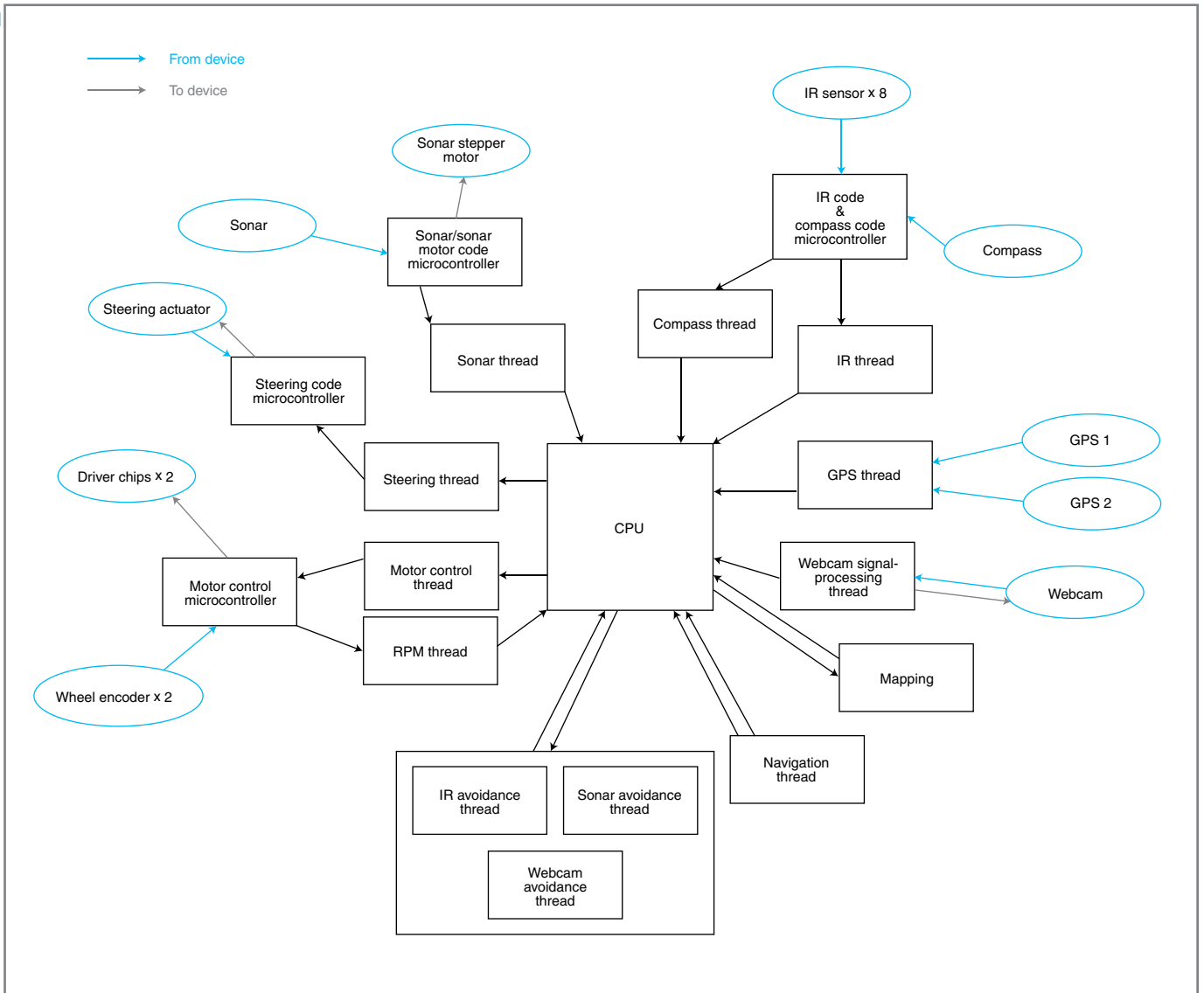


Figure 1—Each module is connected to the CPU by either a USB or a serial connection running one of the protocols in Figure 2a. As you can see, the various threads are a part of the CPU system and communicate as the master in a master-slave relationship with each module.

required data or an acknowledge command (ACK) to confirm reception. If the command is not received correctly, a non-acknowledge (NACK) is returned to the master. An ACK/NACK allows for a simple method of passing a yes/no message to indicate if data is understood. The protocol helps eliminate errors and, at a minimum, recognize erroneous data.

The basic protocol is shown in Figure 2a. The important parts common to each subsystem protocol are LENGTH, TYPE, DATA, and CHECKSUM. The length is communicated so that the receiving device knows how much data is being sent.

The type identifies the device that the brain is communicating with, which is important because some microcontrollers have more than one device connected to it. Each device is set to receive only the correct type or data. Data is the information passed to and from the device. Distance (to an object) and motor speed are examples of data. The checksum is added to the end for the microcontroller. The brain easily identifies bad data, which then results in an ACK or NACK.

Most of our devices had a serial connection to the motherboard for design flexibility. This gives us the ability to remove and troubleshoot a

broken board without having to compromise the entire system, which worked out well for a prototype design. The protocol is also modular enough for each device to be on one serial bus. The TYPE field in the protocol enables each device to determine if a packet is destined for it.

In summary, the master (brain) initializes all communication. The slave device (microcontroller) then responds with an appropriate reply. When bad data is received, the slave responds with a NAK. Depending on the system, bad data is either requested again or simply ignored. Most of our systems (e.g., IR) do not

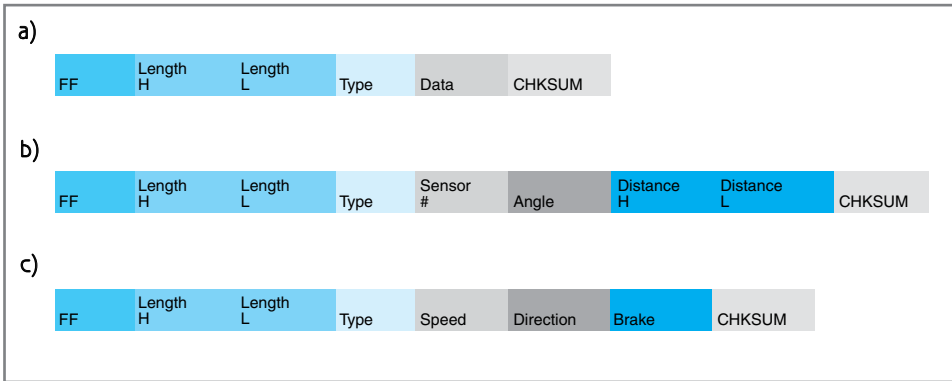


Figure 2a— This is the general protocol design with length and checksum. Type identifies the subsystem that the data is being communicated with. Data size varies with the system in question and the checksum provides a means of error checking. **b**—The Infrared/Compass/Ultrasonic Communication protocol layout. This shows the specific data for these subsystems. **c**—The Motor Control Communication protocol layout. This shows the specific data for these subsystems.

retransmit in the event of bad data because that data would no longer be valid in avoiding obstacles. There are slight amendments to the general protocol for each device (see Figure 2b and Figure 2c).

MODULE DESIGN

We took a modular approach to the hardware design process. We felt it was important to develop each system as a single entity. This enabled us to concentrate on each subsystem until it was fully realized. A Microchip Technology PIC18F4620 8-bit microcontroller is our subsystem's processor (see Figure 3).

All of the vehicle's systems are assembled on custom PCBs that we created in-house with state-of-the-art milling equipment. Each board receives a 12-V supply from the vehicle's deep-cycle battery. Regulators are mounted to each board to give the necessary voltage requirements to each subsystem.

MOTOR CONTROL SYSTEM

The motor control system, for both steering and the drive wheels, comprises a PIC18F4620 MCU that controls an STMicroelectronics VN2SP30

fully integrated H-bridge motor driver. The voltage and current requirements of the power wheels' stock motor, and that of the linear actuator, were well within the capabilities of the driver chip. This is an ideal motor controller for small mobile robotics applications. It has a low on-state resistance that enables it to run reliably without any additional heat sinking.

We made provisions on the motor

control board for adding current-sensing and fault detection using the VN2SP30's additional outputs (see Figure 4). We took advantage of the microcontroller's on-board PWM module to provide smooth speed control for the drive wheels. Our custom serial protocol controls motor speed, direction, and braking commands. A two-bit parallel interface controlled directly to the motherboard provides the steering input control. For feedback for the drive wheels, we drilled a small hole through the first set of reduction gears in the

motor housing. We use an integrated IR photosensor and matching light-emitting diode to achieve a reliable pulse count for wheel rotation. The linear actuator's built-in potentiometer provides an analog reference voltage that's interpreted by the microcontroller's on-board A/D module.

INFRARED SYSTEM

The infrared system includes eight

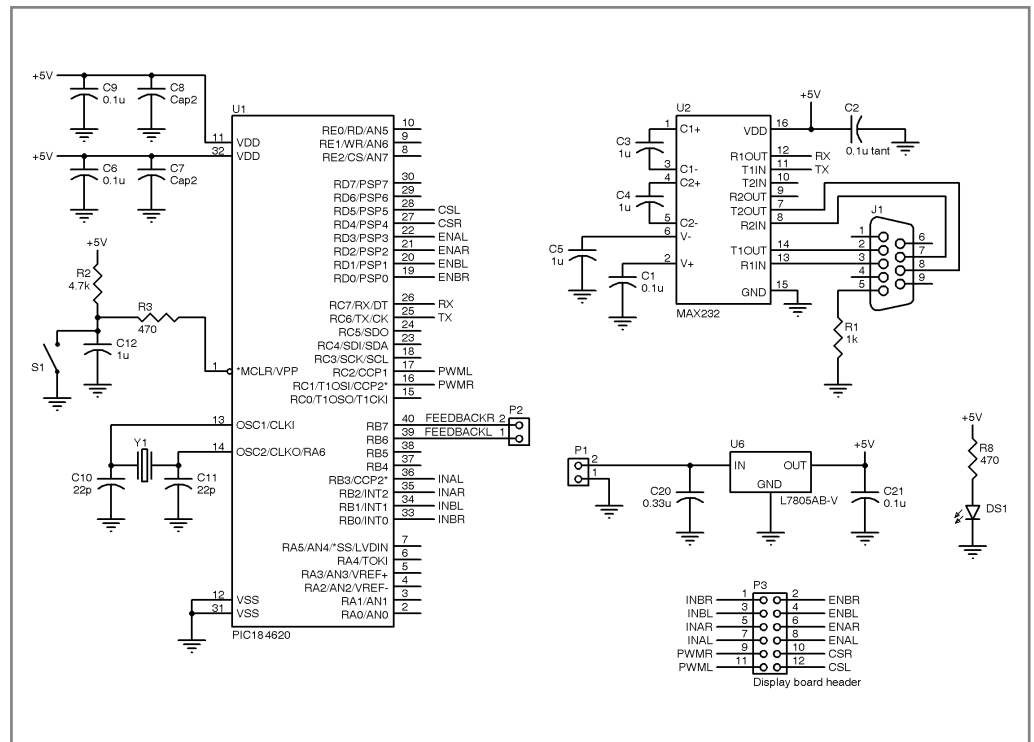


Figure 3—This board connects to the motor controller board. It receives a serial package from the CPU "Brain" and produces a PWM output with a PIC for controlling each motor individually. It includes feedback and support hardware for the rotary wheel encoders.

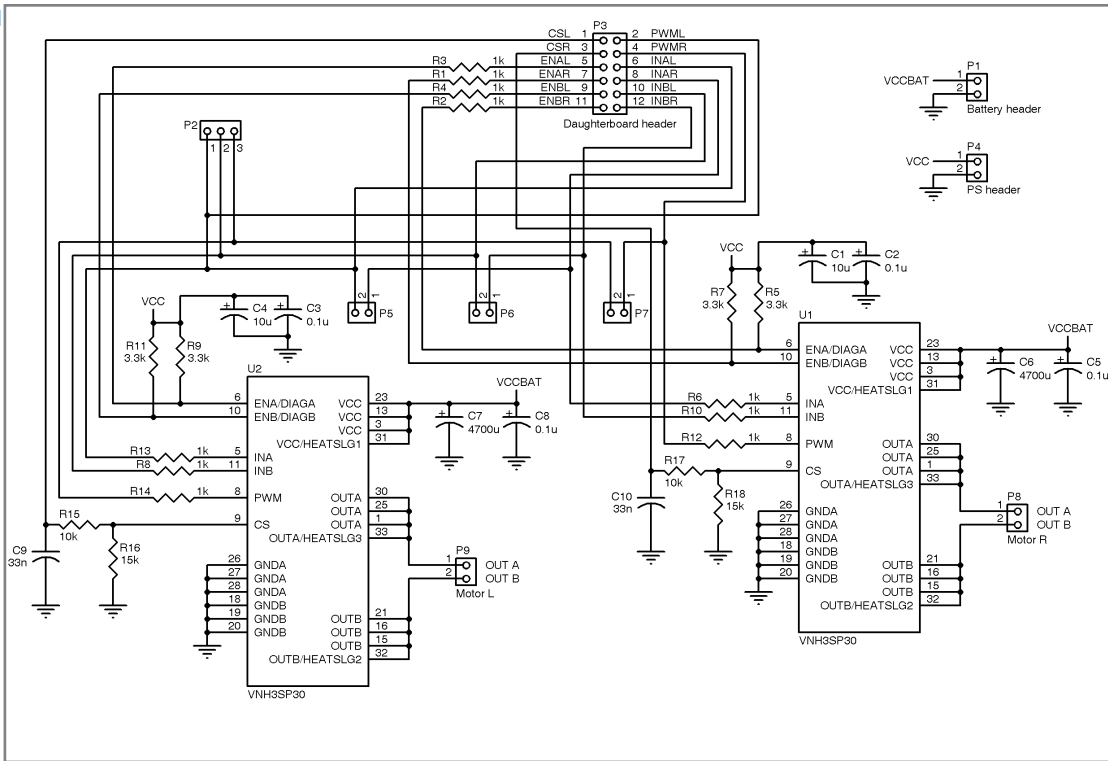


Figure 4—The motor driver board includes the H-bridge with built-in current protection. This board receives a signal from the motor daughter board and can also be controlled directly from Ports 5, 6, 7 for testing purposes.

Sharp GP2Y0A21YK sensors controlled by a PIC18F4620 microcontroller (see **Figure 5**). These sensors produce a 0- to 3.2-V analog output, which is then run through the PIC18F4620's 10-bit ADC module. Six of our sensors are mounted vertically and slightly angled down for object detection (four in the front and two in the back). The other two are mounted horizontally to detect significant drops in the terrain.

The two horizontal sensors are far enough away from the vehicle that if something lies ahead (e.g., a cliff) the vehicle has time to stop. Because the webcam laser rangefinder cannot deal with drops in terrain, the IR system must work perfectly to avoid a drop. The vertically mounted sensors on the left and right are used to detect objects on the left and right (e.g., walls). **Figure 6** shows how the IR sensors

are mounted.

These sensors proved to be susceptible to noise during the testing

process. The output was only 0 to 3.2 V, so the 100-mV of noise caused large amounts of error, particularly

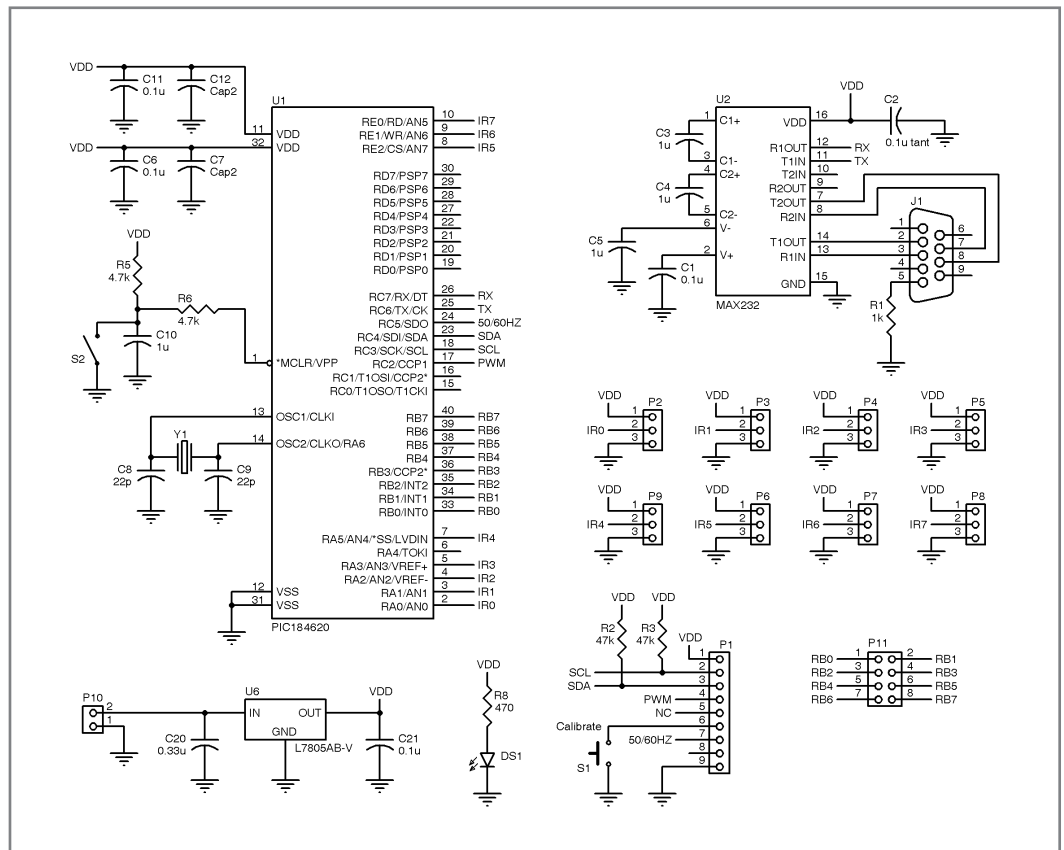
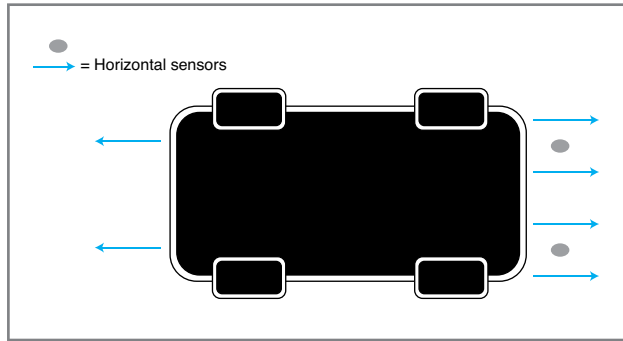


Figure 5—This board has a PIC that collects data from the compass and IR sensors. It then packages the data into the serial protocol shown in **Figure 2** and forwards it to the CPU "brain" when requested.

Figure 6—This is the mounting configuration of the IR sensors. The dots are downward sensors for "Cliff" detection.



at longer ranges where 100 mV equated to a difference of 10 cm. In addition to the noise, these sensors also faced interference from each other, as well as from ambient light. To deal with these situations, we integrated hysteresis and a look-up table into the infrared thread. The

PIC18F4620 outputs one of two states for each sensor to show either something is there or nothing is there.

During the final steps of testing, we tied the IR MCU to the motor driver MCU to increase our stopping time. This enables the vehicle to stop

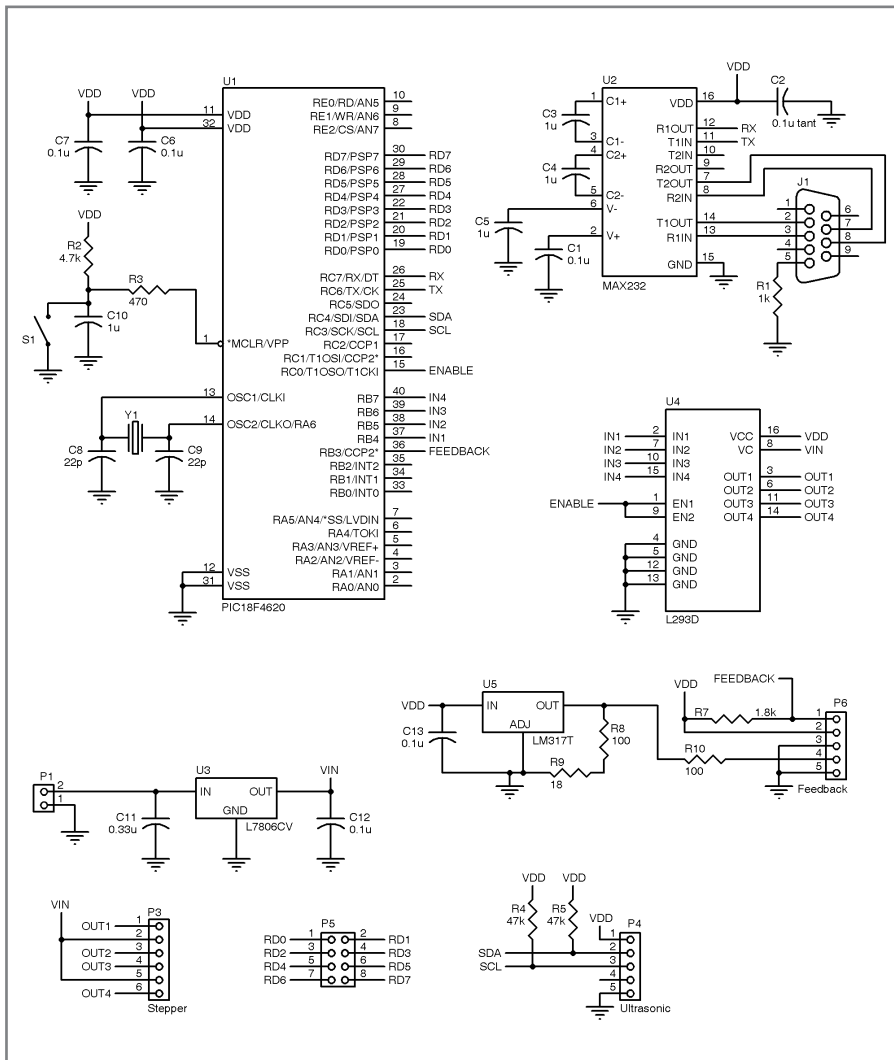


Figure 7—This board has a PIC that queries the ultrasonic sensor via I²C and controls the stepper motor through an H-bridge chip. It sends data to the CPU "brain" via the serial protocol shown in Figure 2 upon request.

Pololu

Robotics & Electronics

Robot Kits
Line followers
Robot arms
Hexapods
Chassis

3pi Robot
\$99.95

High-performance, C-programmable, ATmega168-based robot (with Arduino support!)

Mechanical Components
Motors, servos
Wheels, ball casters

Motion Control
Motor controllers
Servo controllers

Robot Controllers *with Arduino support!*

voltage regulator, dual H-bridge, power LED (green), red user LED, trimmer pot, ATmega48/328 microcontroller, 20 MHz clock, programming connector

Solder Paste Stencils
Use our low-cost solder paste stencils to quickly assemble your surface-mount designs.
From \$25

Custom Laser Cutting
From \$25

Cut your own custom chassis, front panels, and more!

1-877-7-POLOLU
www.pololu.com
6000 S. Eastern Ave. 12D, Las Vegas, NV 89119

Development Solutions for
 ARM, 8051 & XE166
 Microcontrollers

Microcontroller Development Kits

C and C++ Compilers

Royalty-Free RTX Kernel

µVision Device Database & IDE

µVision Debugger

Complete Device Simulation

Examples and Templates

Keil PK51, PK166,
 and MDK-ARM
 support more than 1,700
 microcontrollers
www.keil.com/dd

RTOS and Middleware Components

RTX Kernel Source Code

TCPnet Networking Suite

Flash File System

USB Device Interface

CAN Interface

Examples and Templates

Keil RL-ARM and ARTX-166
 highly optimised, royalty-free
 middleware suites
www.keil.com/rtos

www.keil.com
 1-800-348-8051

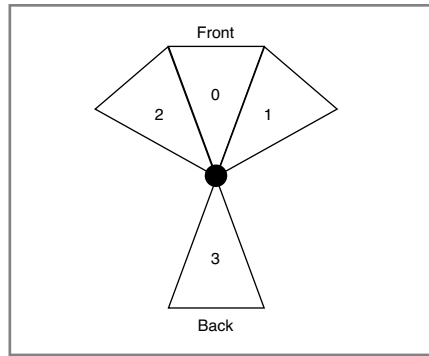


Figure 8—Sections 0, 1, and 2 are used for forward object avoidance. Section 3 is used for backwards object avoidance. The brain handles the avoidance routine similarly to the IR avoidance algorithm.

without having to command through the brain, which takes much longer. The system's reaction time is extremely fast. The IR-avoidance thread deals with the data in a simple way. For instance, if the brain sees something on any of the horizontal sensors, the vehicle stops immediately, turns its wheels left, backs up 1 m, and the reassesses its environment using other sensors. The short program uses simple algorithms.

ULTRASONIC RANGEFINDER

For object detection, we implemented a Devantech SRF08 ultrasonic rangefinder in conjunction with a stepper motor (see Figure 7). Both the motor and sensor are interfaced via a PIC18F4620 microcontroller. Having the motor and ranging control on the same PIC enabled us to easily synchronize the turning of the motor with the rangefinder's timing constraints. The SRF08 features an on-board EEPROM for easy interfacing. Data reading and command issuing to the rangefinder is achieved by reading from and writing to the I²C bus. Commands from the CPU enable the motor to change the scan direction to provide object detection when it's traveling backwards, as well as forwards. The main SRF08-related limitation in this application is its inherent 55° span of detection.

The module is mounted to a mast extending from the top of the vehicle. A stepper motor housed in the lower portion of the vehicle rotates it. The system enables us to take distance measurements at precise angles,

which generates a thorough rendering of the environment. A custom serial protocol relays the distance and angle information to the CPU. An L293D quadruple half H-bridge driver controls the stepper motor with input from the microcontroller. A rotary encoder mounted to the stepper motor's armature provides position feedback.

When the vehicle moves forward, the sensor scans forward in only 7.5° increments of the stepper motor. This data is communicated to the brain's ultrasonic-avoidance thread. The thread divides the data into sections as shown in Figure 8.

WEBCAM IMAGING

We first ordered the ultrasonic sensor thinking it would work well. But while performing the actual testing, we realized that we needed a more accurate method of object detection. Fortunately, some 'Net searching led us to Todd Danko's excellent tutorial on webcam laser rangefinding, "Webcam Based DIY Laser Rangefinder" (Drexel University, 2004). This technique turned out to be a great option because it was easy and affordable. The first few webcam tests did not go well. Some of the webcam's documentation was lacking, so it was hard to find and manipulate the files to make the images work properly. We ended up using a Philips SPCNC900 webcam after reviewing documentation. With the Philips driver interface, we could fully control the brightness and resolution through OpenCV, which is a library of programming functions aimed at real-time computer vision. There is not much

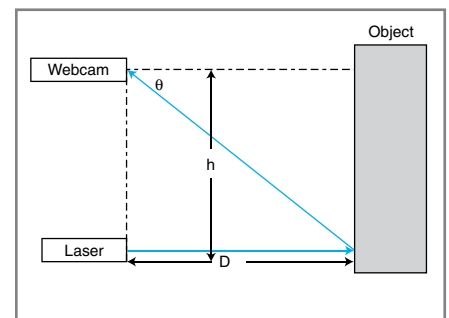
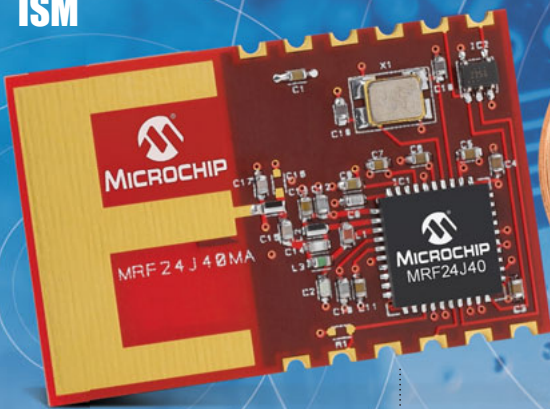


Figure 9—The laser webcam rangefinder can use trigonometry to calculate a distance to an object.

The Newest Wireless Solutions

ZigBee®
RFID
Bluetooth™
GPS
WLAN
Cellular
ISM



MRF24J40MA Radio-Frequency
Transceiver Module
www.mouser.com/MRF24J40MA/

New Products from:



MC13224V ZigBee™ Platform-
in-Package
[www.mouser.com/freescale_
mc13224v/](http://www.mouser.com/freescale_mc13224v/)



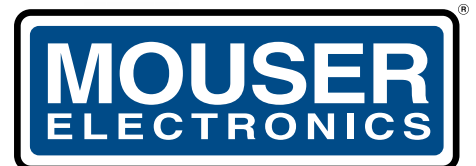
RF Front End Range Extender:
CC2591
www.mouser.com/TICC2591/



CYRF7936 CyFi™ Transceiver
[www.mouser.com/
cypresscyfi/](http://www.mouser.com/cypresscyfi/)

The ONLY New Catalog Every 90 Days

Experience Mouser's time-to-market advantage with no minimums and same-day shipping of the newest products from more than 390 leading suppliers.



a tti company

**The Newest Products
For Your Newest Designs**

(800) 346-6873



www.mouser.com

Over A Million Products Online

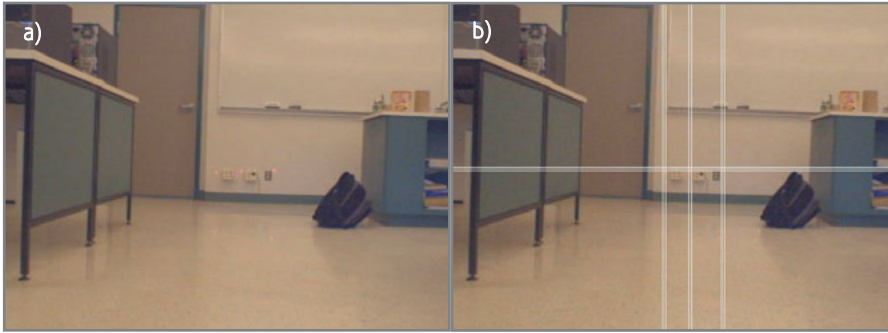


Photo 2a—This is the original laser image taken by the webcam. **b**—White lines indicate where the brightest points in each section are located.

up-to-date documentation on the 'Net for OpenCV, so it was time-consuming to figure out some of the functions. The functions did, however, prove to be useful and easy to use. They were a big help in getting our webcam laser rangefinder up and running. Using the OpenCV libraries enabled us to implement such image processing as Canny edge detection. Canny produced impressive results. It essentially output a black background with white lines at the edges of objects in the image. It would have been an

excellent addition to our project if we'd had more time to implement it.

The webcam's main purpose is to function as a laser rangefinder. An array of three lasers is mounted parallel to the camera. It is important that these lasers don't move and remain perfectly parallel to the camera. If they don't, the image processing won't work as it should. The distance of the three points is calculated based on the number of pixels below center, along with some simple trigonometry. Each laser should be tested individually to ensure

accuracy within the program. The basic theory is depicted in [Figure 9](#). The distance is:

$$D = \frac{h}{\tan \theta}$$

D is the distance to the target. h is always constant, because we always have the lasers mounted at the same height below the camera. We now find theta using:

$$\theta = (\text{pfc} \times \text{rpc}) + \text{ro}$$

pfc is the number of pixels from the center of the focal plane. rpc is the radians per pixel pitch. ro is the radian offset, which compensates for alignment errors. ro and rpc are always constant. This needs to be done and set in code. With these constants, we wrote the code to calculate pfc from the image because it changes in each condition. This is done by counting the number of pixels from the center of the image to get the maximum column and row for the brightest point of the picture, which should be the lasers. We can then easily calculate the distance.

Our program finds the brightest pixel in three sections because there are three lasers. The sections can be made by scanning only the bottom half of the image and making three vertical sections. Each laser will be in its own section, and this will make the code less prone to error. When the brightest point is found, the program will compare that position in the 2-D image to the expected pixel position based on calculations of where the laser should be. If they are equal, then we use the data. If they are not equal, we discard it. Why do we do this? If there is a brighter object than the laser, or if the laser point is lost, we don't get faulty data. By using OpenCV and our own auto-brightness algorithms, we can under-expose images to make it easier to detect the lasers we are transmitting (see [Photo 2a](#) and [Photo 2b](#)).

We found this system to be extremely accurate. For objects at 0.5 to 3.5 m, the error is always less than 0.5%. The algorithm for object avoidance using the webcam laser rangefinder is performed using the data from the three sections. The distance returned is then divided into one of three states: close, midrange,

GENERAL CIRCUITS CO., LTD


QUALITY PCB & SERVICE PROTOTYPE TO PRODUCTION

instant online quote
shopping cart ordering system
China competitive prices
free electrically test

web <http://www.pcbcart.com>
E-mail sales@pcbcart.com
Tel +86-571-87013819
Fax +86-571-87036705
Add No.76 GuCui Road, Hangzhou, China

WWW.PCBCART.COM

CHINA PCB SUPPLIER



and far. Again, this simplifies the object-avoidance algorithm.

TESTING & RECOMMENDATIONS

The modular hardware design of each subsystem, as well as the modularity of the brain's threads, we easily modified the code and hardware for on-the-spot testing. This is excellent for testing individual systems when others may not be working.

Budgetary and time limitations prevented us from fixing a few problems. When we tested the laser system outside on a bright day, it did not work very well. We can deal with this situation by using the ultrasonic rangefinder and infrared for object avoidance. This is why we have back-up systems. A solution to this problem would be a better laser. We used 5-mW laser modules, which were about \$20 each. It is important to get a laser module because it comes with casing and prealigned lenses, but it is more costly than laser diodes. A laser at a higher power such as 80 mW would better suit outdoor conditions.

When the vehicle is near metal objects, the digital compass may return incorrect and possibly harmful data. If we know ahead of time that there will be an excess of metal in the area, we can calibrate the compass to the specific environment.

FUTURE APPS

This project may seem like a cool toy, but when you investigate the system's capabilities, it is easy to imagine many new applications for this technology. Our system's modularity and affordability make it a potential development tool for future applications in the defense, industrial, and healthcare industries. 📄

Chris Britney (csb114@hotmail.com) is a computer engineering technologist working toward a degree in computer engineering. His interests include programming, robotics and microcontrollers.

Paul Green (vacuumpaul@gmail.com) graduated from Camosun College's electronics engineering technology program. He is currently completing his electrical engineering degree at the University of Victoria. His interests include power systems and renewable energy.

Andy Heath (hidden13@gmail.com) is a computer engineering technologist whose interests include embedded systems, robotics and programming. He currently resides in Nanaimo, BC.

Stephen Lau (toshi38@gmail.com) earned a computer engineering technology diploma from Camosun College. He is continuing his education at the University of Victoria, where he looks forward to researching new electronic and robotic technologies.

Kylee Lothrop (kyleelothrop@gmail.com) graduated from the electronics engineering technology program at Camosun College. She is currently working toward a degree in electrical engineering at the University of Victoria. Kylee is interested in green energy as well as embedded and control systems.

PROJECT FILES

For additional schematics and to download code and a project video, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2009/227.

RESOURCES

T. Danko, "Webcam Based DIY Laser Rangefinder," Drexel University, Philadelphia, PA, 2004, www.pages.drexel.edu/~twd25/webcam_laser_ranger.html.

Devantech Ltd (Robot Electronics), "SRF08 Ultrasonic Range Finder (Technical Specification)," 2008, www.robot-electronics.co.uk/html/srf08tech.shtml.

A. Geiger, "Using Philips Webcam Features with OpenCV and PWC," 2007, www.rainsoft.de/projects/pwc.html.

Microchip Technology, Inc., "PIC18F2525/2620/4525/4620 Data Sheet: 28/40/44-Pin Enhanced Flash Microcontrollers with 10-Bit A/D and nanoWatt Technology," 2008.

OpenCV Basic Structures and Operations Reference, www710.univ-lyon1.fr/~bouakaz/OpenCV-0.9.5/docs/ref/OpenCVRef_BasicFuncs.htm#decl_cvSumhttp.

Open Source Computer Vision Library, 2008, <http://sourceforge.net/projects/opencvlibrary/>.

SOURCES

Linear actuator

Firgelli Technologies, Inc. | www.firgelli.com

PIC18F4620 Microcontroller

Microchip Technology, Inc. | www.microchip.com

SRF08 Ultrasonic rangefinder

Devantech Ltd (Robot Electronics) | www.robot-electronics.co.uk

SPCNC900 Webcam

Philips | www.webcam.philips.com

GP2Y0A21YK Sensor

Sharp Electronics Corp. | www.sharppusa.com

VNH2SP30 H-Bridge motor driver

STMicroelectronics | www.st.com



High-Speed Signal Transmission

From Eye Diagrams to Preemphasis and Equalization

When it comes to high-speed signal transmission, there's more than meets the eye. Robert simplifies the topic in this thorough explanation of signal transmission theory. He describes transmission techniques on low-bandwidth cables, and covers the important subjects of preemphasis, equalization, and inter-symbol interference.

Welcome back to The Darker Side. You are used to connecting your RJ-45 plugs to the nearest router and assuming they will work without any problems—and they usually do. You understand that wireless networks like Wi-Fi can be a little trickier, yet even reputed electrical engineers think that wired communications are, well, just wires. But have you ever thought twice about the black magic that actually enables you to send

and receive 100 Mbps, 1 Gbps, or even 10 Gbps on these small twisted-copper pairs? Similarly, do you know how to use a low-cost FR4 PCB and connectors costing less than \$1 to build your PC motherboard with its 2.525-Gbps PCI Express 2.0 busses?

This month I will present basic techniques for high-speed signal transmission on low-bandwidth cables. I will cover concepts ranging from eye diagrams to preemphasis and equalization. You may find this information useful for your next high-speed design.

HIGH-SPEED ISSUES

Suppose you are tasked to design a transmitter and a receiver to send a binary datastream through a roll of twisted-pair wire. As an engineer, you want to ensure your design will work, so you first measure the wire's actual transmission channel. A simple technique involves injecting a sine signal with a sweeping generator at one end of the cable and then connecting a receiver, oscilloscope, or power meter at the other end. Since you take care of impedance

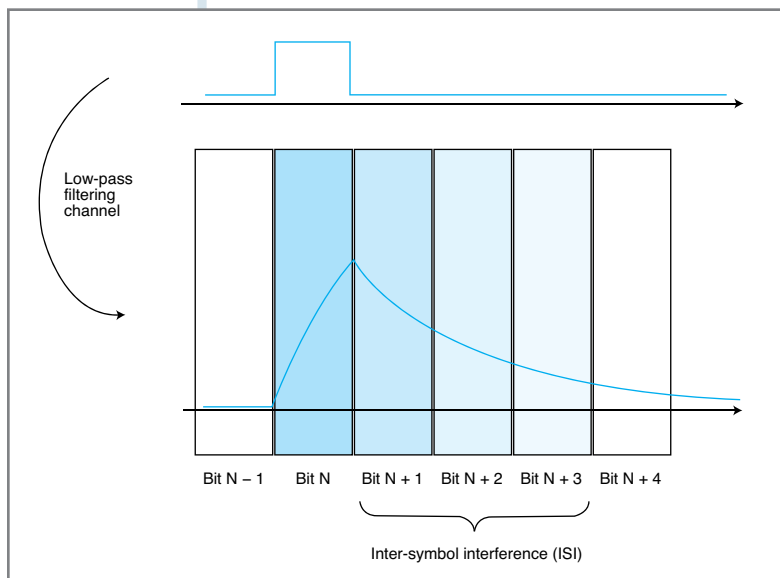


Figure 1—Inter-symbol interference appears due to the transmitting cable's relatively long RC time constant. The line voltage is perturbed several bits later than the original bit. This spreading makes reception far more difficult.

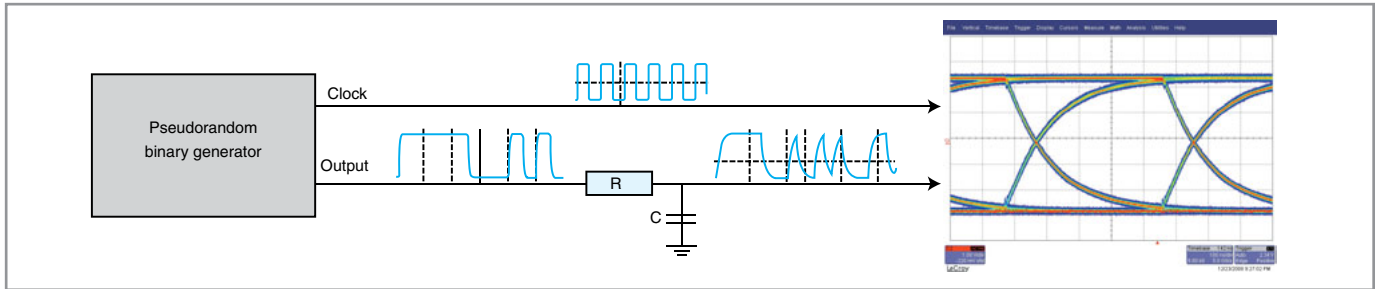


Figure 2—The test setup for an eye-diagram display requires only a pseudorandom generator and an oscilloscope. The required trick is to trigger the scope with a reference clock, either from the transmitter or a clean clock regenerated locally.

matching, you can easily measure the cable's 3-dB cutoff frequency, which is the frequency at which the received signal decreases to 50% of its original power. Imagine that you are lucky enough to find a 100-MHz cutoff frequency, which is already good for a standard copper pair.

You want to transmit binary signals and not just pure sines, so you need to transmit frequencies significantly higher than the bit rate. Namely, you must cope with a couple of harmonics, in order to have well-shaped bits, meaning more or less rectangular waveforms. So, you must use a bit rate reasonably lower than 100 MHz—10 times lower than the cutoff frequency, giving $100 \text{ MHz}/10 = 10 \text{ Mbps}$, may be a reasonable guess. Therefore, you will go back to the marketing department and announce that the system will be 10 Mbps. You know what will happen next, right? The reply will be something like: "Are you joking? We need 1 Gbps." And then your life becomes far more difficult as a result of an unpleasant phenomenon: inter-symbol interference (ISI).

Figure 1 shows where ISI comes from. Basically, a cable used closer or higher to its cutoff frequency behaves like a highly damping RC network. The rising and falling edges of any transmitted bit are reshaped with exponentially decaying slopes, with a time constant that could be significantly longer than the bit duration. That's a problem. But the most critical consequence is that the received voltage for a given transmitted bit will not only depend on this specific bit, but depend also on the previously transmitted bits. This will inevitably generate erroneous bits on the output

because a binary receiver is basically nothing more than a comparator to a given threshold. For example, if you

send alternating zeros and ones, the output signal could be distorted but will still cross the comparator

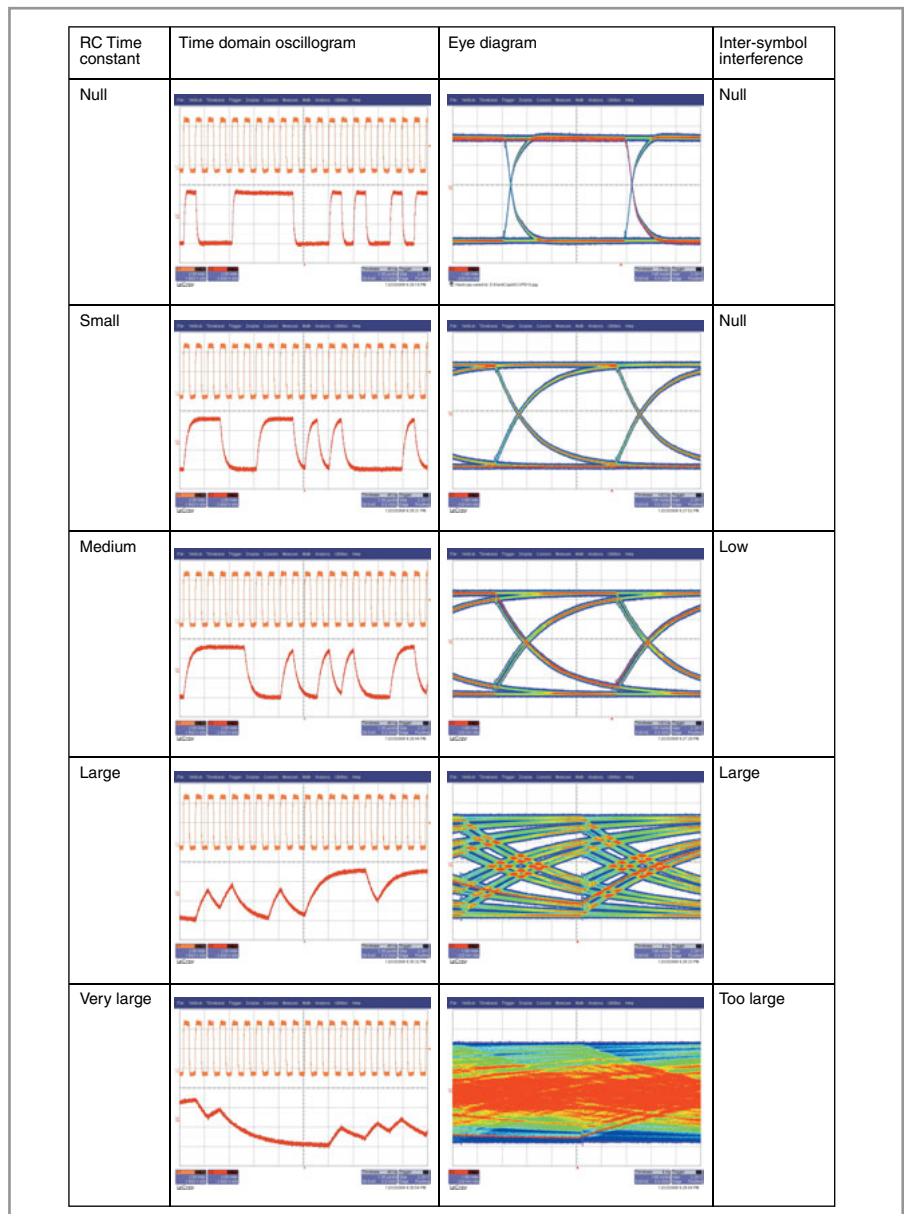


Figure 3—These are actual signal waveforms corresponding to the eye diagram when the line's RC constant is increased step by step. The ISI percentage is the ratio, on the center horizontal line, between the width of the transition region (where plenty of lines are superposed) and the bit period. It ranges from zero at the top to 100% at the bottom.

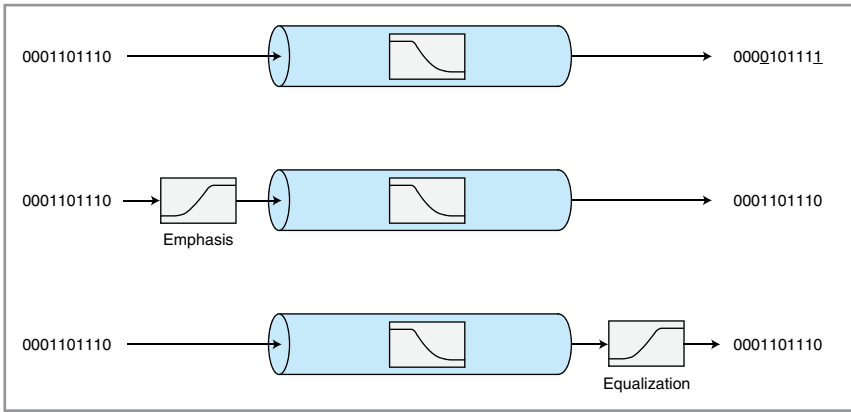


Figure 4—The correction of the channel transfer function can be done either on the transmitter side (pre-emphasis), on the receiver side (equalization), or on both.

threshold each time and give a solid transmission. However, if you send a long string of ones, the data line voltage will reach its maximum even if the RC time constant is long. Then, if you send only one zero followed by new set of ones, the line voltage will decrease little, but not enough to be identified as a zero by the receiver. This “history” phenomenon is the basic form of ISI, and it could actually make error-free reception nearly impossible.

DO YOU SEE THIS EYE?

The inter-symbol issue is not new. It was a problem associated with telegraphic lines right from the start. Telecom workers needed a way to visualize and evaluate a signal’s

ISI level, so they invented the eye diagram. This powerful, intuitive method needs just two pieces of equipment. First, a pseudorandom binary sequence generator must be connected to the transmitter’s input to ensure all possible ISI scenarios are sent through the channel. Such a pseudorandom generator is available as part of many telecom test sets, but you can also build one with a shift register and a couple of XOR gates. If you search the Internet for “Linear Feedback Shift Register” (or “LFSR”), you’ll find plenty of schematics. You also need a simple single-channel oscilloscope. It must have an external trigger input, and its bandwidth must be significantly higher than the signal’s bit rate.

You shouldn’t think that “high-speed techniques” are only for multi-Gbps systems that will inevitably require a fancy high-end digitizing scope. It could be the case if you’re working on Gigabit Ethernet or PCI express, but “high speed” can simply mean your required bit rate is higher than what your wire could support (as in my example). So, these techniques are also valuable with medium-rate transmissions (e.g., a few megabits per second), at least if you have to use a long or poor transmission line.

The eye diagram test setup requires you to connect the scope input to the signal to be measured and to trigger it with the original clock (see [Figure 2](#)). If you can’t—for

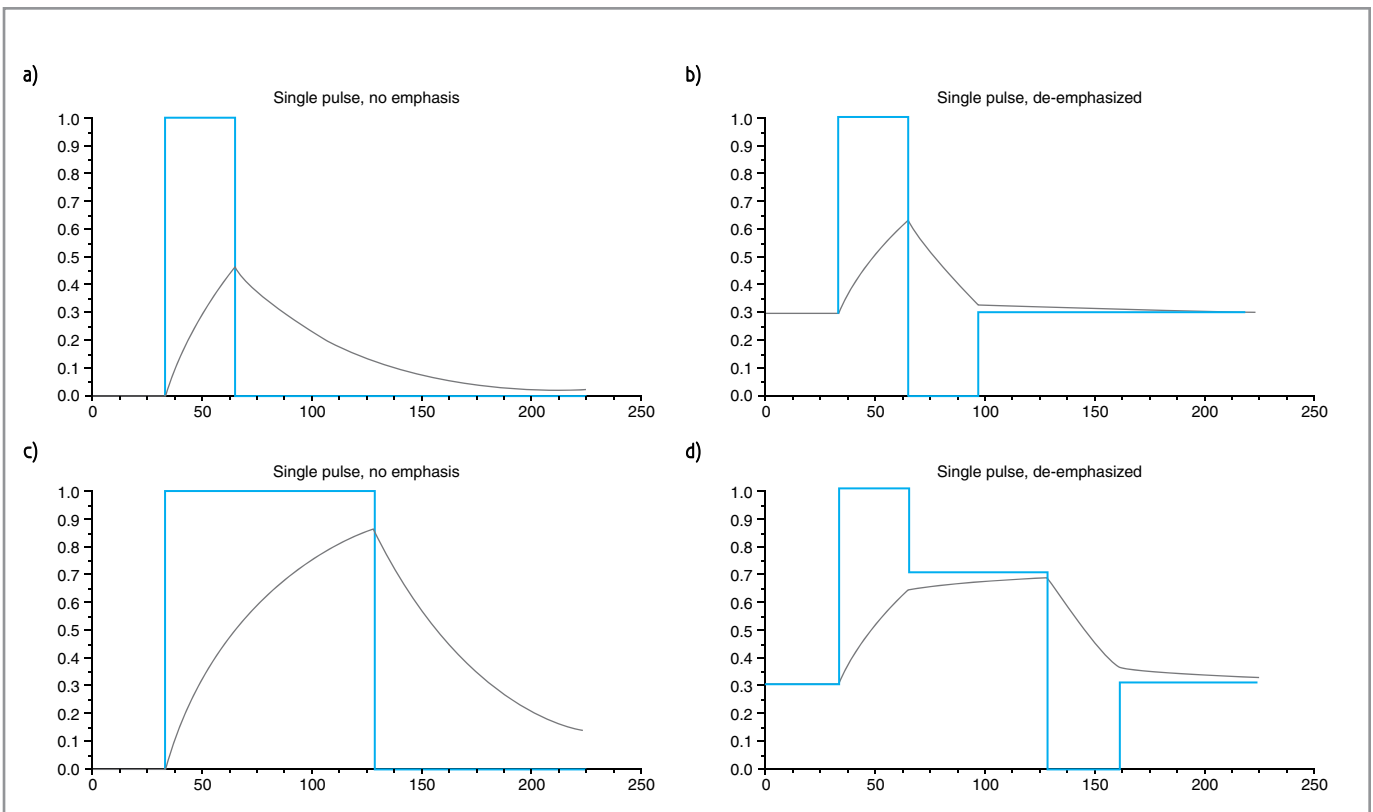


Figure 5—This is a Scilab simulation of pre-emphasis. The blue curves are the transmitted waveforms. The dark curves are the received waveforms. The left column is without pre-emphasis, respectively with a single bit (a) or a sequence of ones (c). It is clearly visible on the right side (b and d). The preemphasis enables the drastic reduction of the ISI.

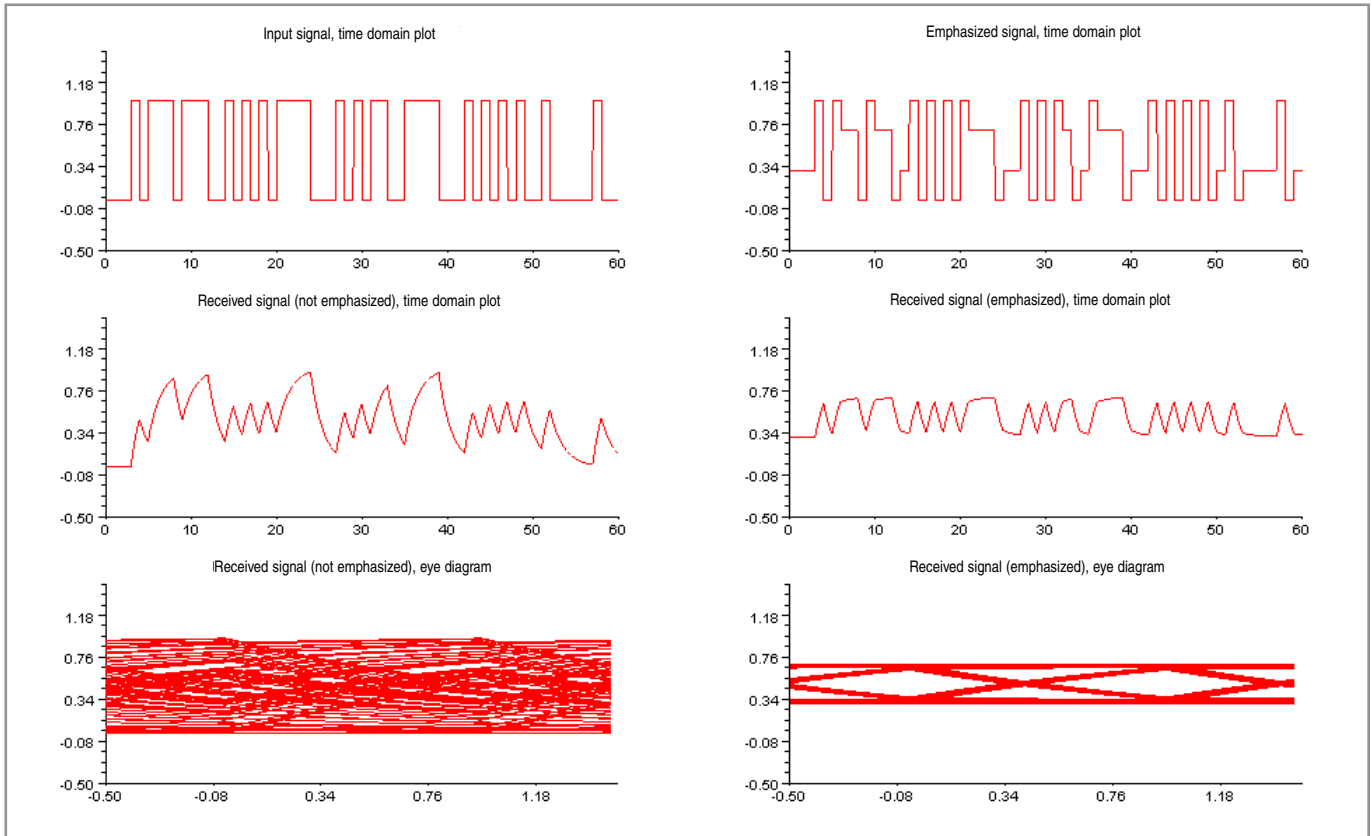


Figure 6—This simulation shows the impressive effect of preemphasis at least with a transmission channel modeled as a RC network. Left is without emphasis, right is with it. The eye diagram looks better, doesn't it? However, keep in mind that a real-world situation could be a little more complex than this simulation.

example, if both ends of the cable aren't in the same room—you can regenerate the clock locally through a phase-locked loop (PLL). The oscilloscope's horizontal time setting is set to twice the bit duration, and the trigger delay is adjusted to have a single bit centered on the display. What would you see on the oscilloscope's display? Basically a superposition of bit transition waveforms, which will include every possible bit sequence thanks to the pseudorandom generator.

I used my Lecroy Waverunner 6050 oscilloscope. It was driven by the pseudorandom generator section of an old Schlumberger 7703B 2-Mbps bit-error tester through a simple adjustable RC filter. The results are shown in [Figure 3](#). If the RC time constant is low—meaning that the transmission channel is good in comparison to the bit rate—then the shape of the data signal is rectangular and the eye diagram shows only a simple image made with the superposition of the four possible bit transitions in the

pseudorandom stream (0 to 1, 1 to 0, 0 to 0, 1 to 1). What happens if you send this signal to a comparator-based receiver? Ideally, the threshold will be centered vertically and you will get an ideal bit stream as an output, without any time jitter. The ISI is 0%.

As long as the RC time constant stays low in comparison to the bit period, the image looks the same. The center hole in the eye diagram, the center of the "eye," is said to be fully open, which means that bit recovery by the receiver will be easy. However, if you increase the RC constant, the eye diagram starts to be more complex (the third case in [Figure 3](#)). The raising and falling fronts are so damped that the signal can't reach their steady level before the arrival of the next bit, which means that there will be some ISI. The ISI percentage can be evaluated as the ratio, on the horizontal line, between the width of the transition region (where plenty of lines are superposed) and the bit period. It is there around 5%. The more the RC

time constant is increased, the more shut the eye. The fourth test case, a comparator, even with a perfectly centered threshold, shows a very high level of time jitter on its output, meaning a higher risk of bit error. There is around 50% ISI.

At a given point the eye is fully shut, which means that a comparator-based detector will no longer be able to reliably detect if the bit is a zero or a one. The transmission no longer works, ISI is 100%, and the eye is fully shut.

The eye diagram enables you to quickly evaluate the ISI level and the link's reliability. Using the eye diagram, I can also give you more data than just the ISI level. For example, the channel noise margin is simply proportional to the eye's height. The allowed level of timing errors, including clock jitters, is proportional to the eye's width.

PREEMPHASIS

Assume that you are in a bad situation: the eye diagram is fully closed

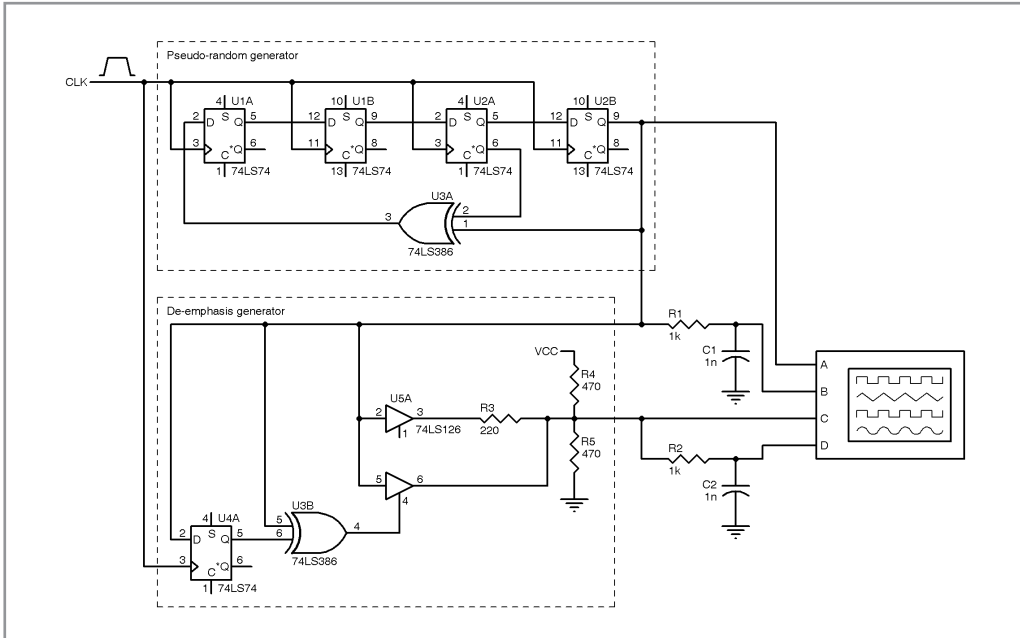


Figure 7—This is a discrete preemphasis circuit. The top section is a simple 16-step pseudorandom sequence generator made with four D latches and an XOR gate. The bottom portion uses one more D gate and an XOR gate to compare the current bit to the previous one. A weak signal is generated through R3 if it is the same. The XOR gate enables a strong signal if not. Both outputs are sent to a virtual scope for simulation (directly and through an RC network).

sent after a one, the first will be a strong zero (e.g., 0 V) and the next bits will be weak zeros (approximately 1 V). This way a little more energy is given during each transition thanks to the strong bits. As a consequence, the rising is quicker and the long-term damping is reduced, and it can be nearly zeroed with a proper emphasis ratio. **Figure 5** shows you a simulation of emphasis done with the Scilab simulator. (The source code for all the simulations is posted on the *Circuit Cellar* FTP site.) If you look at the frequency domain, you see that such a preemphasis scheme increases the signal's high-frequency

when you have a mandatory wire and bit rate. How can you still build a reliable transmission system without compromising the requirements? Think twice. You are in trouble because the channel (your wire) has a low-pass-shaped transfer function with too low of a cut-off frequency. The answer seems obvious. You need to amplify the high frequencies in order to get back to an overall flat frequency response end to end.

As you can see in **Figure 4**, you have two main options. You can amplify the high frequencies on the transmitter side with a technique called “preemphasis,” or you can do it at the receiver side with a technique called “equalization.” Many systems use both techniques together for optimum results.

Let's focus first on the TX side with preemphasis. The ISI comes from the shape of the pulses, which are too slow to rise or fall. It means there isn't enough high-frequency energy during the bit transitions. The most usual and clever

trick for signal emphasis uses only a limited amount of electronics. Whenever there is a bit transition (i.e., whenever a transmitted bit is different from the previous one), you can simply boost its voltage. This works by using not just two but four voltage levels: strong high, strong low, weak high, and weak low. For example, if the line is zero and you have to transmit a couple of ones, your circuit first sends a strong one for the first bit (e.g., 5 V) and then a lower voltage for the next ones (weak one, 4 V or so). Similarly, when a couple of zeros have to be

components—exactly what you are looking for.

Just to be exhaustive, note that two variants of preemphasis can exist. In the first, the voltages of the strong bits are actually increased in comparison to the standard logic voltages. This has an important drawback because you will need higher-than-normal and negative voltages on your transmitter. The most usual way is to use standard logic voltages for the strong bits and reduced voltage swing for the weak ones, with a fixed and preset ratio that's optimized depending on the

transmission line and bit rate (e.g., by looking at an eye diagram).

If you read my October 2007 column, “No Fear with FIR” (*Circuit Cellar* 207), you understand that such a pre-emphaser is actually equivalent to a two-tap FIR filter. More evolved emphasizing techniques are possible with more evolved FIR filters. The idea is in fact to add to the waveform to be transmitted

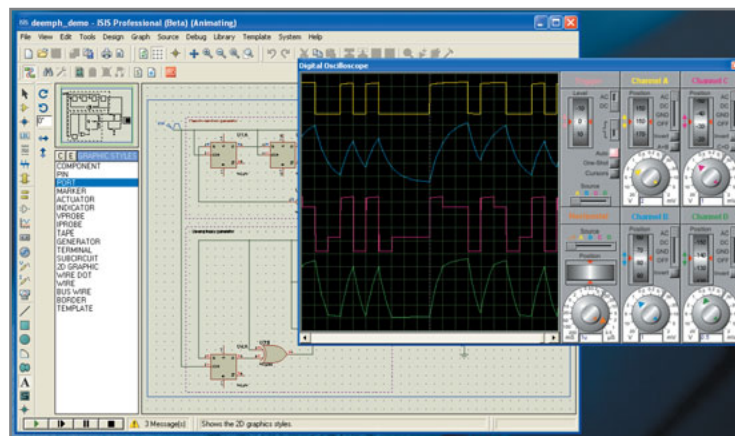


Photo 1—This is the result of Proteus VSM simulation of the schematic provided in Figure 7. I have not built the circuit, but I am pretty sure the actual measurements would be close.

a delayed and reversed copy of the erroneous channel response. But I won't cover that in this column.

Can a simple preemphasis solve the problem in my RC filter example? Yes. Refer to the Scilab simulation in [Figure 6](#). When using a preemphasis transmitter, the received signal level is obviously lower, because we are reducing the level of the weak bits, but it has a far lower ISI, which means a far better bit error rate.

You may wonder how to actually build a preemphasis circuit. I will describe existing dedicated chips. But you can do it yourself too, at least if the signal speed stays reasonable. [Figure 7](#) shows a possible solution that uses old 74xx series chips. As I drew the first version of this schematic with the Proteus CAD suite (a *Circuit Cellar* staff editor redrew it for this article), I clicked the Simulate icon and got what you see in [Photo 1](#). Nice, isn't it?

EQUALIZATION

Equalization is an option that involves building some kind of selective high-frequency booster on the receiver side. Equalization is usually less of a magical solution than pre-emphasis, because a high-pass filter has the disadvantage of amplifying any high-frequency noise on the line. This is why equalization is usually used as a complement to pre-emphasis. Preemphasis does 80% of the job, then equalization finishes it. On the hardware side, an equalizer is often simply a digital FIR filter and called a feed-forward equalizer (FFE). More complex nonlinear algorithms exist like the distributed feedback equalizer (DFE), but a FIR is usually enough.

Equalization provides a great bonus: a receiver actually sees the degraded received signal, which a transmitter can't, at least without a dedicated feedback information channel. It is then theoretically possible for the receiver to adjust the equalization filter to compensate for the channel's behavior and implement a "perfect" channel. In that

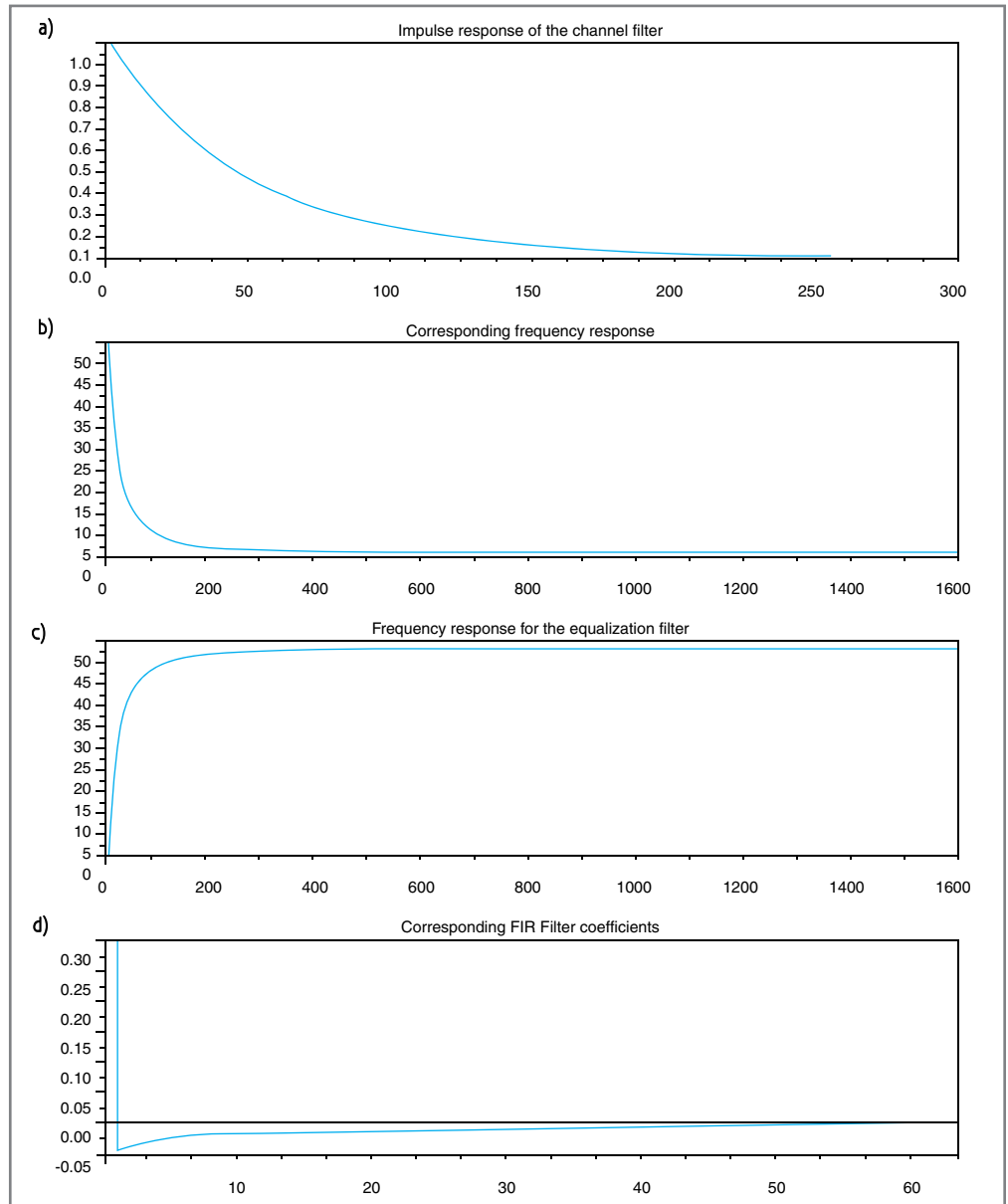


Figure 8—This Scilab simulation shows how an equalization filter could be generated. The top plot is the time domain impulse response of our reference RC channel. Take its FFT and you get the second plot. The horizontal axis is the frequency. This is the transfer function of the transmission channel. Negate it in order to get an ideal compensation network (third plot). You can then take its inverse FFT to get the required impulse response of the equalization filter, which is nothing less than the coefficients of the equalizing FIR filter.

case, if you know the problem, you can fix it. This can even be done if the channel's behavior changes over time, as is the case on a wireless link. This technique, adaptive equalization, is used in many modern wireless systems.

Without using a complex adaptive filter, how can you build an equalization filter for my example RC simulation? As you can see in [Figure 8](#), you need to calculate or measure the channel's frequency response (including preemphasis, if any). It's then easy to deduce the amplification factor over frequency required for a flat response. A simple Fourier transform then gives you the coefficients required to implement the corresponding FIR filter. (Once again, please refer to my column on FIR filters to refresh your memory.) I did it in Scilab (see [Figure 9](#)).

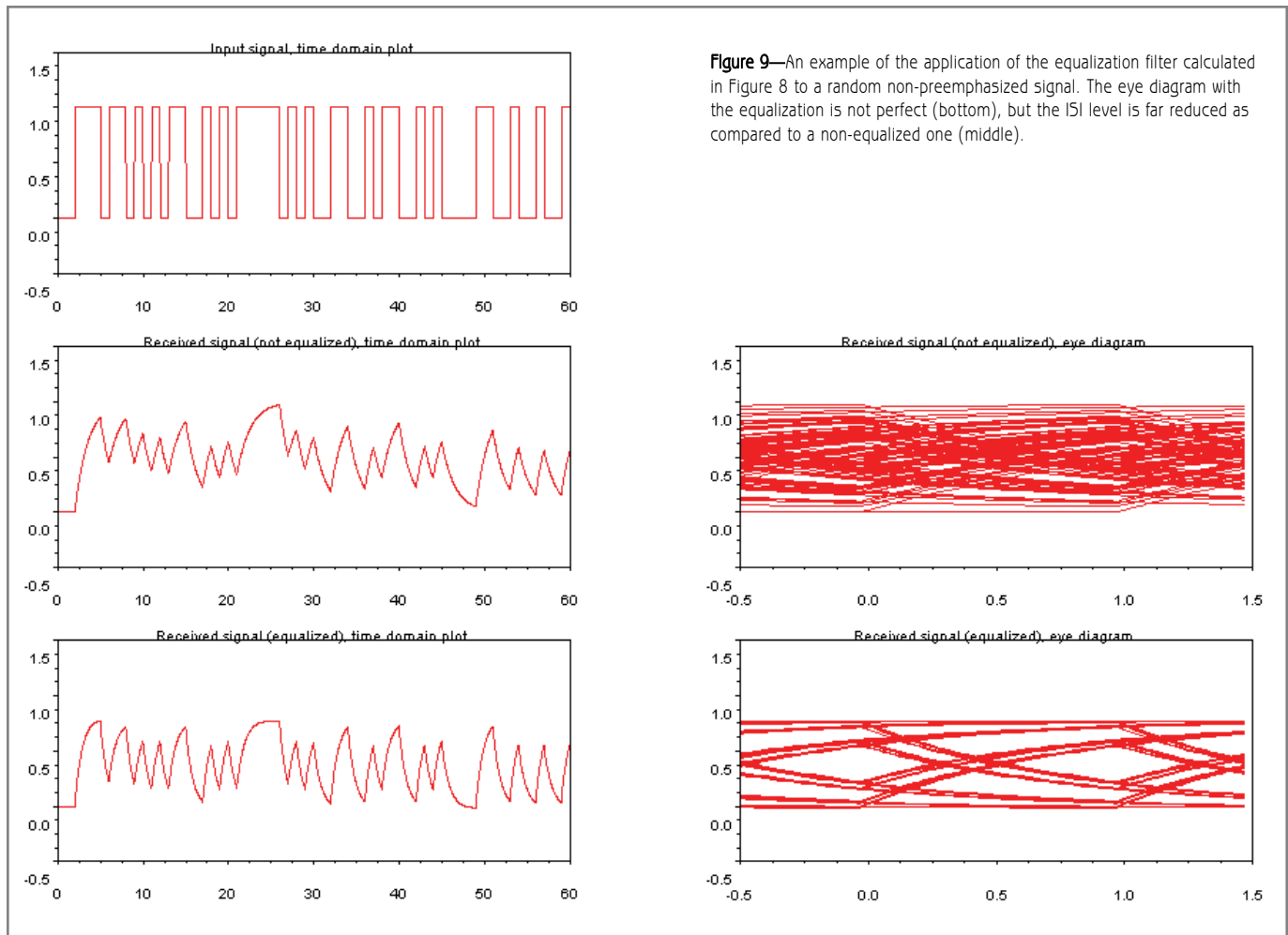


Figure 9—An example of the application of the equalization filter calculated in Figure 8 to a random non-preemphasized signal. The eye diagram with the equalization is not perfect (bottom), but the ISI level is far reduced as compared to a non-equalized one (middle).

The eye diagram is not perfect because of the filters, but at least the signal is improved.

SOME SILICON

Of course, silicon suppliers have ready-made chips for implementing preemphasis or equalization. Few solutions exist for relatively low-speed data links, but notable exceptions are Maxim Integrated Products's MAX3291 and MAX3292. These chips look like classic RS-485/422 transceivers, except they include preemphasis circuitry to allow higher bit rates on long cables. The MAX3291 is programmable for preset data rates of 5 and 10 Mbps, while the MAX3292 is configurable up to 10 Mbps through an external resistor. The corresponding Maxim application note (AN643) states that these chips enable you to double the bit rate in comparison to standard RS-485 drivers.

Conversely, you'll also find dedicated

high-speed equalizer chips, but they are scarce. Examples are National Semiconductor's EQ50F100n and DS64EV100. The latter is an equalization filter that improves data links up to 10 Gbps on cables or on a standard FR4 PCB through a choice of eight preset filter settings. A National DS40MB200 is a dual 4-Gbps buffer with both programmable preemphasis and a fixed equalization filter, which is available from Digi-Key if you want to play with it.

However, the majority of preemphasis/equalizer applications are done with FPGAs and not dedicated chips. This makes sense because these techniques are mainly useful for high-speed systems, and FPGAs are the de facto choices for such projects. You will find preemphasizing and equalizing blocks (or IP) from most FPGA vendors. A good example is Altera's Stratix II GX family of FPGAs. These monsters integrate no less than 20 high-speed (up to 6.375 Gbps) ports

and a built-in adaptive equalizer block called the adaptive dispersion compensation engine (ADCE), which uses an impressive mixed-signal approach. In a nutshell, a capacitor-based booster amplifies bit transitions on each input, with a boosting effect controlled by a low-speed DAC. Your application can then tune the DAC for a minimal bit error rate.

A last word: Equalization can be analog too. Nothing prevents you from designing an analog high-pass filter that could more or less exactly compensate the frequency behavior of a given cable. It can be difficult to do, but it is possible. For example, Tyco Electronics Corp. proposed a line of high-speed interconnect cables with a built-in equalization PCB through its high-speed serial data connector (HSSDC) family. These cables are specified for no less than a 2-Gbps transmission up to 30 m away on a copper cable. Try to do

that without equalization.

HIGH-SPEED COMMUNICATION

The techniques I covered in this article are used in high-speed communication systems. Roughly speaking, a simple “strong/weak bit” preemphasis technique can boost the high frequencies. It could be a good fit for the less complex situations you encounter. Equalization, and adaptive equalization in particular, enables you to fine-tune a channel for optimal performance. It’s a must whenever a given channel must be used at its best. This means quite often nowadays.

You won’t need these techniques for every design. If you need them to implement standard high-speed transmission protocols (e.g., Gigabit Ethernet or PCI-Express), you’ll probably get them as part of a chipset, so it should be more or less transparent for you. But if you need to work on an exotic data transmission project and can’t achieve more than 70% of the target’s performance, knowing the basics of preemphasis and equalization may enable you to come up with the missing 30%. Don’t forget that “high speed” just means quicker than the maximum the transmission channel can natively accept, which may be quite slow. And don’t forget the marvelous eye-diagram investigation method.

I hope you found this review of high-speed transmission techniques interesting and useful. Be sure to keep one eye in the time domain and the other in the frequency domain. If you do, preemphasis and equalization shouldn’t be on the darker side anymore. 📧

Robert Lacoste lives near Paris, France. He has 18 years of experience working on embedded systems, analog designs, and wireless telecommunications. He has won prizes in more than 15 international design contests. In 2003, Robert started a consulting company, ALCIOM, to share his passion for innovative mixed-signal designs. You can reach him at rlacoste@alciom.com. Don’t forget to write “Darker Side” in the subject line to bypass his spam filters.

PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2009/227.

RESOURCES

Agilent Technologies, Inc., “Understanding the Effects of Limited Bandwidth Channels on Digital Data Signals,” *Agilent Measurement Journal*, Issue 5, May 2008, www.agilent.com/go/journal.

Lattice Semiconductor Corp., “Transmission of Emerging Serial Standards Over Cable,” 2005, www.latticesemi.com/lit/docs/generalinfo/Transmission_whitepaper.pdf.

Maxim Integrated Products, Inc., “Pre-Emphasis Improves RS-485 Communications,” AN643, 2001, www.maxim-ic.com/appnotes.cfm/an_pk/643.

National Semiconductor Corp., “Setting Pre-Emphasis Level for DS40MB200 Dual 4Gb/s Mux/Buffer,” AN-1389, 2005, www.national.com/an/AN/AN-1389.pdf.

R. Stephens, “Answering Next-Gen Serial Challenges,” *Tektronix Knowledge Series*, Parts 1–3, Tektronics, Inc., www.tek.com/Masurement/programs/americas/serialdata_webinar/.

W. Wong et al, “Digitally Assisted Adaptive Equalizer in 90 nm with Wide Range Support from 2.5 Gbps to 6.5 Gbps,” Altera Corp., 2007, www.altera.com/literature/cp/cp-01026.pdf.

J. Zhang and Z. Wong, “White Paper on Transmit Pre-Emphasis and Receive Equalization,” Mindspeed Technologies/Conexant, 2002, www.analogzone.com/io_shoot_mindspeed.pdf.

SOURCES

Stratix II GX FPGA

Altera Corp. | www.altera.com

Scilab

INRIA | www.scilab.org

Proteus VSM mixed-signal simulator

Labcenter Electronics | www.labcenter-electronics.com (North American distributor: www.r4systems.com)

Waverunner 6050 digital oscilloscope

Lecroy Corp. | www.lecroy.com

RS-485/RS-422 Transceivers with preemphasis

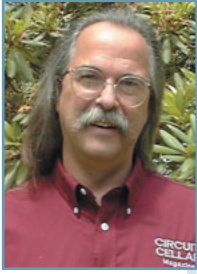
Maxim Integrated Products, Inc. | www.maxim-ic.com

EQ50F100n Equalizer, DS64EV100 Equalizer, and DS40MB200 mux/buffer

National Semiconductor Corp. | www.national.com

HSSDC Cable assemblies

Tyco Electronics | www.tycoelectronics.com



Location Notification

A Look at Anisotropic Magnetoresistance Sensors

Electronic compasses are an affordable alternative to GPS technology for some embedded designs. After introducing the topic of magnetoresistance, Jeff describes how he used an anisotropic magnetoresistance (AMR) sensor to add navigation capabilities to a recent design.

I'd looked at Honeywell products in the past. But to be honest, I'd kept away from them because they were costly and hard to find. More recently, however, many of the distributors I use have been carrying them with more favorable pricing structures.

Last fall, for instance, while reading a news release about a Honeywell three-axis magnetoresistance sensor, I was interested enough to download the datasheet and take a closer look at the part. I remember getting excited as I read through the specs until I came to the packaging reference: a 4-mm-square leadless chip carrier (LCC). Whoa, leadless! I didn't like the sound of that. I can deal with tiny leadless contacts that extend partially up the sidewalls of a chip. But this leadless package had no side contacts (only on its bottom surface, like a BGA). For those of you steady-handed hand solderers, you know this is usually a dead end.

There has been a lot of serious experimentation with using the typical kitchen toaster oven to solder SMT PCBs. I've always wanted to give this a try, and now I had a reason. Although I haven't gone through the steps of creating a microcontrolled oven (if you think

this is worthy of a column, or if you have input, let me know), I graphed my oven's temperature profile to see if it fit within reasonable parameters. After comparing what I saw, I felt there was a good chance for success.

My first prototype PCB has a single SMT—a Honeywell HMC5843—right in the center. I won't go into the process here other than to say I first pinpointed the solder paste melting point on a blank PCB so I knew what to look for when the HMC5843 was placed on the PCB. I left the heat applied for another 30 s after reaching that point and then let it cool completely before moving the PCB. It was impossible to see what had actually happened to all the solder joints under the chip, so I needed a good deal of faith.

THE SENSOR

The HMC5843 incorporates three anisotropic magnetoresistance (AMR) sensors, each of which is aligned to an axis (x, y, or z). Anisotropic, or directional dependence, describes the sensor's response to the Earth's magnetic lines of force. A magnetoresistance sensor is made from nickel-iron (Permalloy), which is deposited as a thin film on a silicon

wafer in a bridge configuration. Each sensor outputs a voltage—corresponding to the strength of the field it measures—and is oriented to a different axis. Thus, each measures a percentage of the field depending on its orientation to the field. The bridge output is positive when the magnetic field runs with the direction of sensitivity. It is negative when the field is reversed.

You, of course, can measure a field's strength with a single sensor, but it must be aligned with the field. With a three-axis sensor, you can measure the field without alignment by mathematically adding each sensor's outputs. To prevent the sensors from accumulating any biasing magnetic properties, the sensors have a mechanism to condition the magnetic domains. This is done by sending and reversing a current through an internal strap on each sensor, thereby degaussing the sensor.

ASIC

In addition to its three magnetoresistance sensors, the HMC5843 also has an array of circuit elements that applies signal conditioning to the

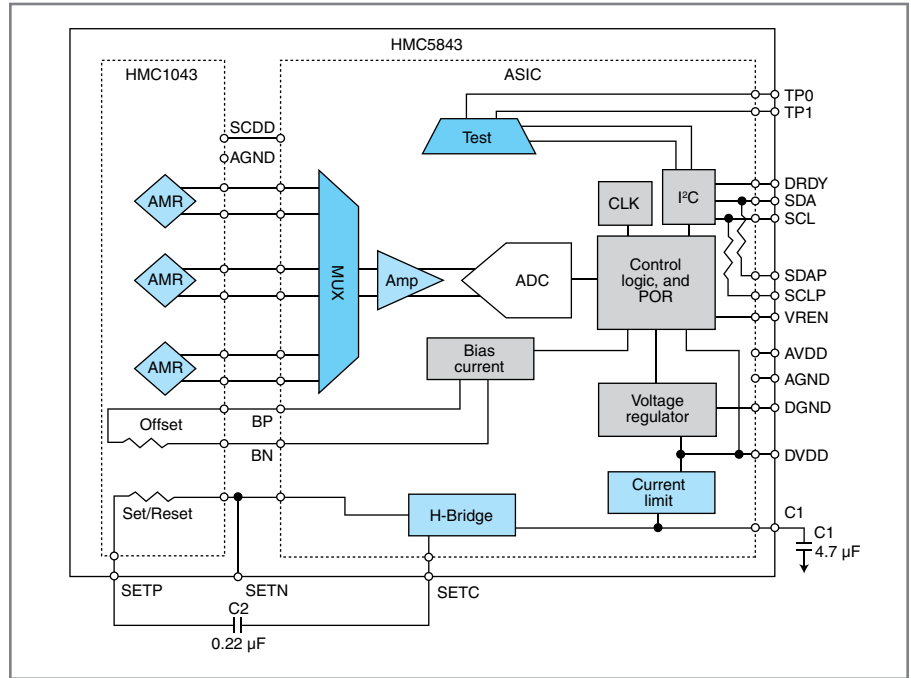


Figure 1—The HMC5843 is a multi-chip module containing a HMC1043 AMR sensor and ASIC support circuitry. It measures a mere 4-mm square.

sensors and provides a user-friendly interface to the measurement results and property configuration (see [Figure 1](#)). Each sensor's differential outputs are multiplexed into an instrumentation amplifier (IA, providing gain control)

and a 12-bit ADC (A/D, providing a digital representation of the three measured field strengths). A logic block controls the system timing, I²C interface, degaussing function, and sensor bias.

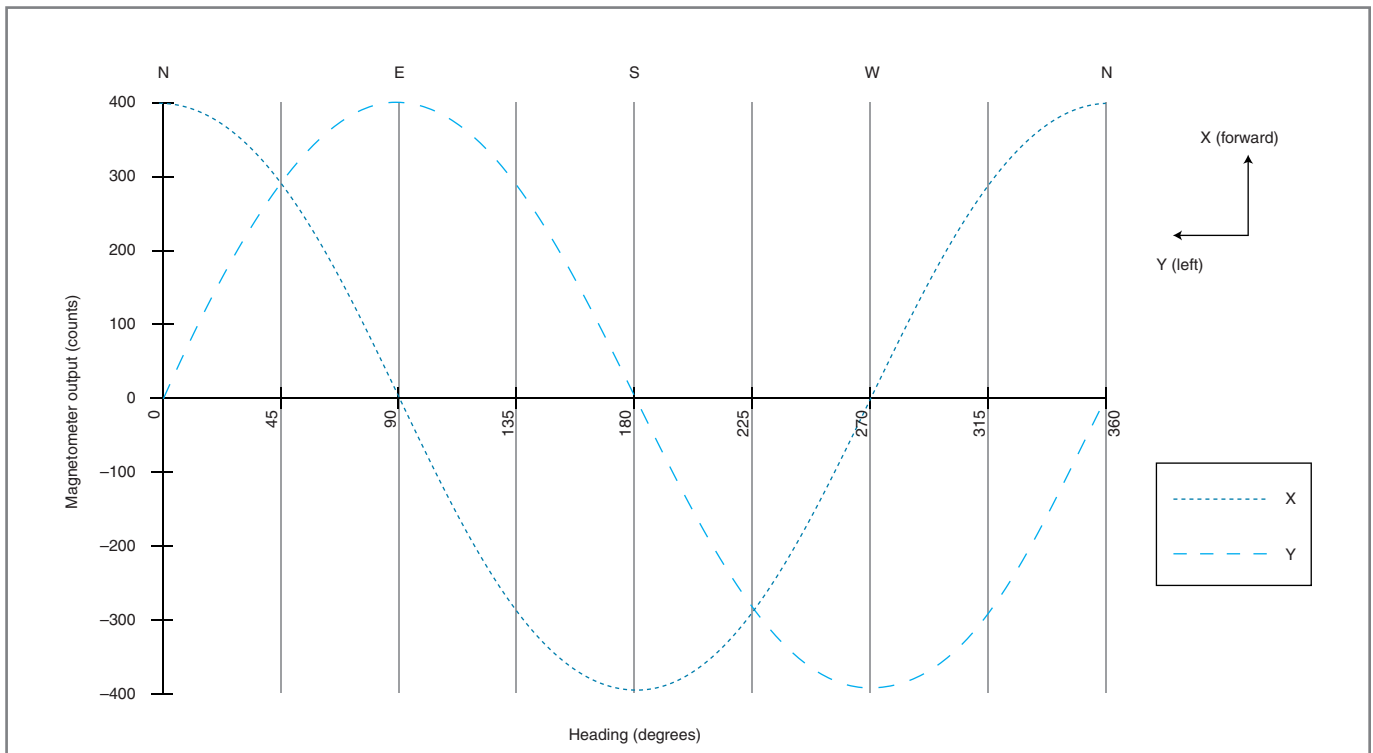


Figure 2—AMR sensors mounted at right angles will output magnetic measurements in a sine-cosine relationship with each other, as they are rotated 360° through the Earth's magnetic field.

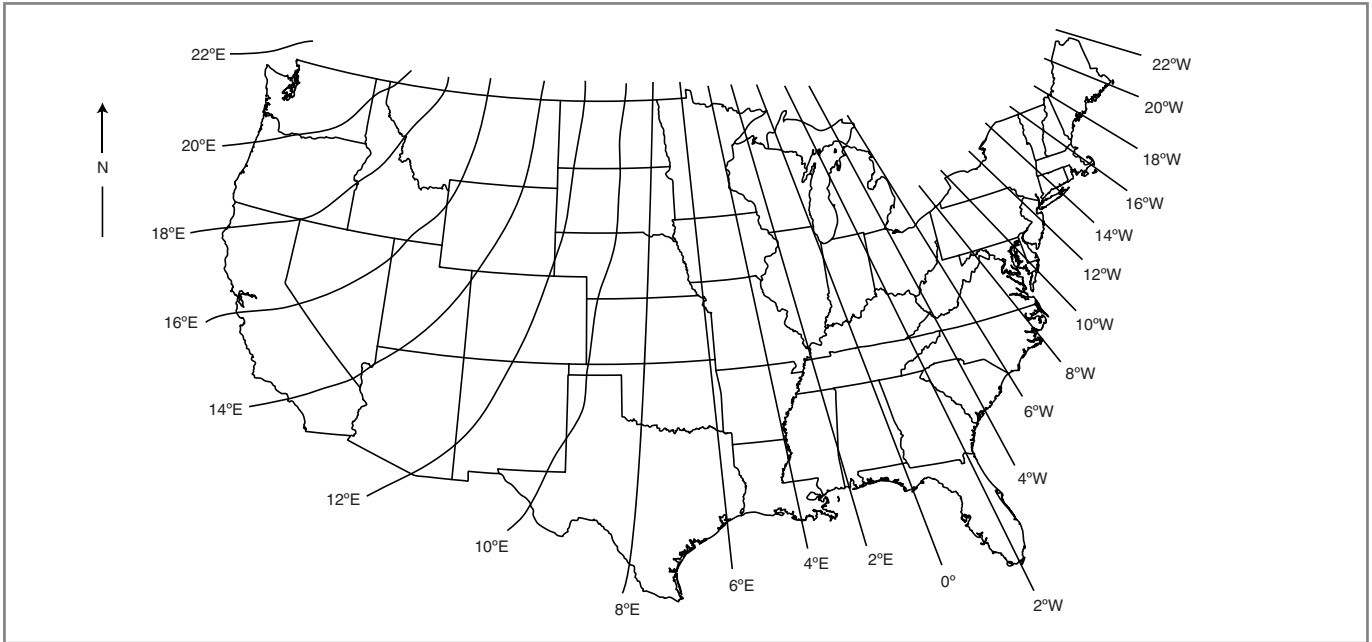


Figure 3—This map shows the declination error between the (true) north pole and the Earth’s magnetic north pole. The dark line (0°) is a place where the magnetic and true north poles are aligned (the magnetic pole being presently approximately 800 km south of true north pole along this line).

The HMC5843’s operating voltage is 2.4 to 3.3 V. Internally, it requires 1.8 V (for the digital section). This voltage can be self-generated via an internal regulator, or it can be supplied externally as a second supply voltage.

The IA, referenced to AVDD/2, amplifies the AMR bridge’s differential output. The Earth’s magnetic field strength is approximately 0.6 G (the AMR bridge). The HMC5843’s seven possible gain settings cover a range of 0.7 to 6.5 G. The ADC converts full-scale outputs to $\pm 2,047$ counts. If you have the gain too high and the ADC’s input level is exceeded, the conversion value is forced to 0xF000. Note that external ferrous metals (iron) can distort the Earth’s magnetic field by providing an easier path than through the air. This draws the field toward itself. And magnetized materials can influence the Earth’s magnetic field by adding a “fake” source. This not only pushes the Earth’s field, but it can also overpower the sensors.

ARCTAN

Two of AMRs are parallel to the PCB’S surface. If you rotate the PCB so the x-sensor’s positively sensitive axis aligns toward the north magnetic pole, the x-sensor can measure a maximum amount of the Earth’s magnetic field. The y-sensor is perpendicular to the field so it measures a minimum amount of the Earth’s magnetic field. As the chip is rotated clockwise from 0 to 360°, the x and y measurements have a sine-cosine relationship (see Figure 2). The angle or heading can be calculated using the arctangent of the y-sensor measurement divided by the x-sensor’s measurement.

When the x-sensor’s measurement equals zero, you have a divide-by-zero error. A check for $x = 0$ can have

two possible headings: one at 90°, if y is greater than zero, and one at 270°, if y is less than zero.

Arctan transforms measurements into a heading of 0–90° four times in one 360° rotation. The polarity of the x and y values indicates the quadrant of a circle to which the measurement refers. You must add (or subtract) the arctan heading from a quadrant offset to get the actual heading. After you have a heading value of 0 to 359°, you know the direction to the magnetic north pole. The top

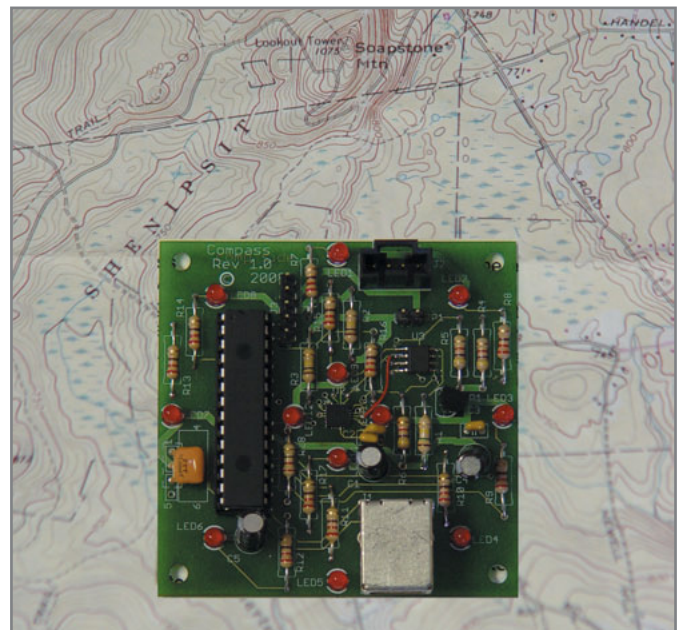
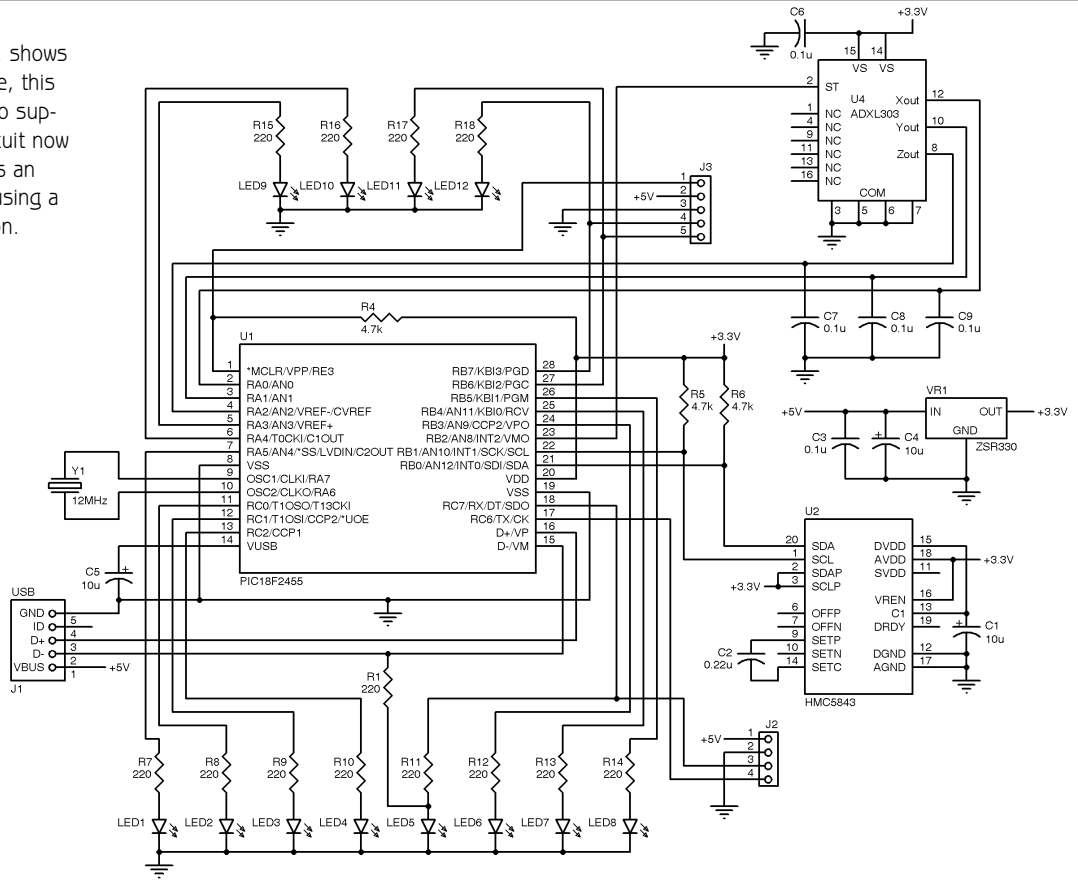


Photo 1—This circuit will turn on the most northern LED (or LED pair) to give you a general northern direction. When used as a peripheral, this circuit provides the actual heading via a serial port.

Figure 4—While Photo 1 shows a through-hole prototype, this circuit was redesigned to support SMT parts. This circuit now operates on 3.3 V. R1 is an alternate to R11 when using a serial vs. USB connection.



of most maps is oriented toward true north. Aligning the north/south longitudinal lines to magnetic north can throw off the map from +20° to -20°, depending on your location (see Figure 3). Most maps indicate this difference between magnetic and true north as a declination constant (in the map's key). Correct map orientation is achieved using the declination constant.

If you want your compass to point to true north instead of magnetic north, you need to add the declination constant for your area to the magnetic heading. For example, I live in Connecticut, which has a declination constant of approximately -15°. This means magnetic north is -15° (CCW) from true north. So, I must add 15° to my magnetic heading to get a true north heading.

IRON ERRORS

I covered how the environment can distort the Earth's magnetic lines of force. Measuring a distorted field will not provide accurate results unless you can know how the field is

Windows CE Touch Controller

CUWIN™

CUWIN4300K ▶
\$499 / Qty.1

- 7" / 10.2" wide color TFT display
- 800 x 480 resolution, 260K colors
- SD card & Ethernet support
- RS232 x 2 / RS485 x 1 or RS232 x 3
- Mono Speaker and Stereo Jack
- Real time clock (Battery backup)
- Visual Basic and Visual C++ support
- USB I/F (ActiveSync)
- ARM9 32bit 266MHz processor
- Keyboard or Mouse support

CUWIN3200 ▶
Qty.1 / \$399

- * CUWIN3200
- * 7" Bezel type case
- * CUWIN3500
- * 7" Cover Case (Front waterproof)
- * CUWIN4300K
- * 10.2" Black Bezel type case
- * CUWIN4300S
- * 10.2" Silver Bezel type case

COMFILE TECHNOLOGY www.cubloc.com
Toll-Free: 1.888.928.2562

distorted. If the compass is mounted in something like a car or robot, ferrous materials will introduce distortion that you can easily remove with a calibration routine. All of this distortion-altering content rotates along with the compass while the Earth's magnetic field remains constant, so you can take calibration measurements by rotating this in a circle. Distortion causes the measurements to stray from ideal conditions, but you can create adjustments based on what you see.

Under perfect conditions, each axis has the same range—that is, the x-sensor's maximum count should equal the y-sensor's maximum count. A difference in range can be adjusted by applying a gain adjustment to each axis. The axis with the largest range (positive count + absolute value of negative count) is given a gain equal to one. The other axis is given a gain of whatever is necessary to make its range the same as the first: $\text{gain} = x$. In this case, x is the largest range/smallest range (i.e., $2,000/1,800 = 1.11$). A sensor's gain adjustment must be applied to each of the sensor's measurement values.

In addition to the range adjustment, each sensor's maximum positive count should be equal to its maximum negative count. If a distortion is pulling the measured values one way or the other, the maxima won't be the same. If the x-sensor's positive maximum is, say, 100 counts more than the x-sensor's negative maximum, you must add an adjustment of -50 counts to each measurement taken by the x-sensor. The y-sensor measurements must be treated the same way. Calibration is necessary whenever the environment changes. The calculated gain and offset adjustments can be stored and used each time a sensor measurement is made. These are compensated measurements:

$$\text{x-sensor value} = (\text{x-sensor gain} \times \text{x-sensor measurement value}) + \text{x-sensor offset}$$

$$\text{y-sensor value} = (\text{y-sensor gain} \times \text{y-sensor measurement value}) + \text{y-sensor offset}$$

TILT

Tilt is a measurement of pitch and roll about an axis. (Here I reference the compass's x-axis.) If you imagine the x-axis as the nose and tail, and the y-axis like the wings of a plane, roll is the wingtip swing

Register address	Name	Access
0	Configuration A	R/W
1	Configuration B	R/W
2	Mode	R/W
3	X-axis MSB	R
4	X-axis LSB	R
5	Y-axis MSB	R
6	Y-axis LSB	R
7	Z-axis MSB	R
8	Z-axis LSB	R
9	Status	R
0A	Identification A	R
0B	Identification B	R
0C	Identification C	R

Table 1—These 13 registers are available via an I²C interface between the HMC5843 and the PIC microcontroller. The result of a measurement (± 2047) is stored in register pairs for each axis.

up or down about the x-axis, while the pitch would be the nose (tail) up or down about the y-axis. Either of these movements will produce an amount of tilt from level. Level is considered the plane that is perpendicular to a vector directly toward the center of the Earth (Earth's gravity vector).

Using the z-axis's relationship to the compass's x-axis and y-axis, you can calculate the Earth's magnetic vector—that is, the direction of the Earth's magnetic field. I previously mentioned that the Earth's magnetic field is aligned toward the north/south magnetic poles. But if you look more closely, you will notice that while the field is relatively perpendicular to gravity at the equator, as you move closer to a pole,

the field eventually becomes parallel to the gravity vector (as it enters the Earth directly into the pole). For the relationship of the x-axis to y-axis (or heading calculation to be accurate), they must remain as level as possible. A tilt may change this relationship. If it does, it will

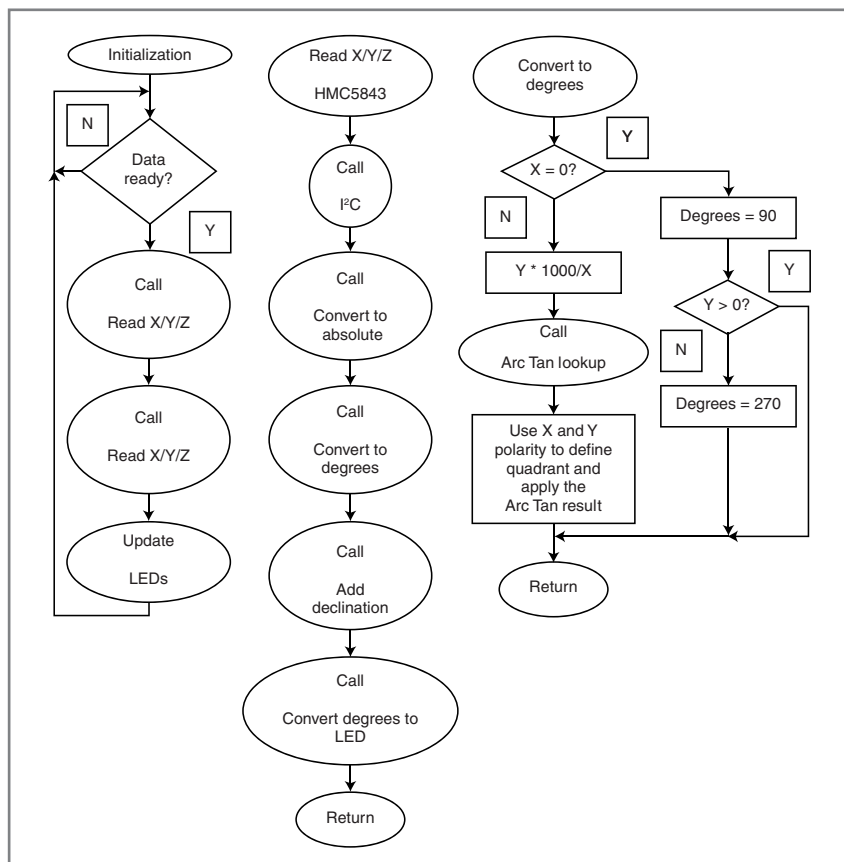


Figure 5—The flow chart on the left shows a simple main loop. The HMC5843 routine in the center illustrates how the data is converted. One of the conversions is charted on the right. It uses integer math and a table lookup to determine heading from the individual sensor measurements.

introduce an error. A particular tilt will cause an increasing error as the compass moves further north or south from the equator, so keeping a compass level is more important to those of us living in New England.

I've used Analog Devices iMEMS accelerometers in past projects to reference the direction of Earth's gravity ("What's the Slope?: Use an Accelerometer to Measure Slope," *Circuit Cellar* 202, 2007). In this project, the accelerometer's z-axis measures maximum Gs (1), while the compass (PCB) is level. The accelerometer's x-, y-, and z-axes are aligned with the AMR's. Therefore, as the accelerometer's z-axis begins to register a tilt, the z-axis and y-axis can be used to determine the direction of the tilt, and one of the four inside LEDs will indicate which direction in which you need to tilt (actually, anti-tilt) the PCB.

PROTOTYPE

My first prototype was through-hole, except for the HMC5843 (see [Photo 1](#)). The original idea was to use the compass's z-axis as a leveling indicator. Unlike the documented declination constant, the downward component to the Earth's magnetic field is not well documented. Because the magnetic pole and true north are not the same, I couldn't just use the latitude to calculate the z-axis component. So, I redesigned the circuit to include a three-axis accelerometer to better determine tilt. I designed with SMT parts and was able to reduce the PCB to 2" square (see [Figure 4](#)). At that point, it made sense to reduce the working voltage to 3.3 V. It also reduced the complexity by eliminating some level translation components.

The HMC5843 uses an I²C interface while the accelerometer has analog outputs. To interface with external circuitry, the PCB can be set up to use a serial UART or USB interface. In this column, I'm focusing on the AMR and how its measurements relate to magnetic (or true) north. Using the I²C interface, you can access 13 registers within the device (see [Table 1](#)). These registers can be divided into three groups. The first group (registers 00–02) has read/write access and can configure how the chip operates. The second group (registers 03–08) holds the measurements taken from each of the three axes. Each register pair holds signed values from 0 to 2047. The last group (registers 09–0C) contains status and identification information.

The device has four operating modes: Sleep, Idle, Single Measurement, and Continuous Measurement modes. The lowest current consumptions (a few microamps) come from Sleep mode (supplying dual voltages, no internal regulator). Normal operations (using the internal regulator for 1.8 V) consume approximately 1 mA. The mode register determines operational modes. After a single conversion, the device returns to Sleep mode. The configuration registers set up sensor gain, conversion delay, and continuous conversion rate. By reading the status register, you can determine when the compass is done updating the axis measurement registers. In addition, after you

begin reading them, the compass will hold off updating any of these six registers until you've finished reading all of them. The Identification registers have fixed values that relate to the device itself. In this case, the registers hold the ASC values H34.

COMPUTATIONS

Math is always a concern when you're writing code. Independent of the language you're using, you can expect that math support may often require more code space than the rest of the application. Floating-point routines require the most space. Integer math requires considerably less space but may not be able to provide adequate accuracy. Tables also can be used to reduce the need for floating-point support, although they can be potentially large.

In this application, I used function $\arctan = y/x$ to convert from axis measurements to degrees. I used a table holding precomputed values for degrees versus the arctan equivalent. You can find tables for arctan values in many reference books. This is a list of degrees with its equivalent arctan value: 0° (an arctan = 0) through 90° (an arctan = infinity). The values in my reference table are listed with a resolution of thousandths. Noticing that 89° (the largest value) = 57.290, I can multiply all values by 1,000 and keep them within a 16-bit word table entry without decimal headaches. Thus, when I divide y by x to get an arctan value, this result needs to be 1,000× larger to relate to the table. So, my math routine must use the form $\arctan = Y \times 1,000/x$. Two integer math routines will satisfy my needs: a 16-bit × 16-bit multiply and a 32-bit × 16-bit divide.

As you can see in [Figure 5](#), the main loop consists of three routines: read the HMC5843, read the AXL330, and the update of the LEDs. The read HMC5843 routine has four routines to read and convert the data: I²C routines are used to retrieve axis measurements for the device; each measurement is converted into an absolute value along with a sign polarity notation; the measurements are converted to a degree equivalent, and a pre-stored declination constant modifies the heading; and the heading in degrees is used to determine which LEDs should be on to indicate true north. The declination constant adjustment can be skipped if you want to point to magnetic north.

Let's look closer at the Convert to Degree routine. A check for divide by zero ($X = 0$) is made to prevent a computational error. Depending on y's polarity, when $X = 0$, degrees will equal 90 to 270. Otherwise, you want to convert the x-axis and y-axis measurements into an arctan value that can be used for table look-up. The table consists of arctan values for degrees 1 to 89. Beginning with the first table value (for 1°), compare the table's value with the computed value. If the computed value is bigger, increment the degree counter and continuing checking the next table entry. The degree counter is the converted equivalent. However, this is a measurement of position within one quadrant (0–90°) of a circle (360°). Using the polarity flags for the x-axis and y-axis measurements, you can determine

which quadrant the conversion reflects and make an adjustment relative to 0 or 180° by adding or subtracting the conversion from 0 or 180°.

While the microcontroller holds the actual heading in degrees and can be read through a serial or USB connection, eight LEDs on the PCB are used to indicate the heading to true north. Note that 360° divided by eight LEDs equals 45° for each LED. To improve the resolution, multiple LEDs can be used to imply a heading between the two, giving a resolution of 22.5°. The LEDs are meant to show you a general direction.

Holding the compass level is important to accurate measurements. Four additional LEDs are referenced to the x- and y-axes. The ADXL330 accelerometer measures the gravity vector. If the accelerometer measurements determine a non-level situation in either the x-axis or the y-axis, LEDs illuminate to show how to correct the tilt.

PAYING FOR MORE

As designers, we must constantly review the buy-versus-build choice. GPS can give you heading information, but this is based on movement.

Author's note: Thanks to Mark Amundson at Honeywell Aerospace and Robert Cook at ProComp Associates for their assistance.

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.

It must use multiple positions to determine direction of travel. Electronic compasses will cost far less than including a GPS in your design just for heading information. At the opposite end of the cost spectrum are individual magnetic sensors that you can combine with your own amplifiers, multiplexers, ADCs, and microcontrollers to create what the HMC5843 offers. However, \$20 seems like a pretty good bargain for all of that. Don't want to deal with computations? You can buy a two-axis version (HMC6352) that internally converts the heading for you for \$30. But you still must make sure the device is level! To step up to a device that contains a three-axis AMR, a three-axis accelerometer, and internal algorithm computations, you can use the HMC6343 at a mere \$175.

I've yet to receive the second (SMT) prototype, but I want to add routines like calibration. I think this circuit is flexible enough to be useful in future projects. However, the most fulfilling aspect of this project was overcoming my fear of staying away from those SMT parts that can't be hand-soldered. I might need to go shopping for a toaster oven. ☹

THE SERIAL PORT LIVES!

Everything you need to know about COM ports, USB virtual COM ports, & asynchronous serial ports for embedded systems.



Hardware & software for RS-232 & RS-485. Wireless options and more.

Serial Port Complete
Second Edition
Jan Axelson

ISBN 978-1-931448-06-2 \$39.95
Lakeview Research LLC www.Lvr.com

From the author of USB Complete

PIC-SERVO
MOTION CONTROL

MOTION CONTROLLERS FOR BRUSH, BRUSHLESS AND STEPPER MOTORS.

- controller chips
- controller boards

www.picservo.com
JEFFREY KERR, LLC

MACH64
PROGRAMMABLE LOGIC
STARTER KIT

Based on the Lattice ispMach 4064.



Includes 250 page lab manual.

Learn CPLDs the fun way with the MACH64! This complete kit comes with everything you need to take you from mystery to mastery with CPLDs and programmable logic. Learn to turn *software* into *hardware*!

www.XGAMESTATION.COM

RESOURCE

Honeywell International, Inc., "Compass Heading Using Magnetometers," AN-203, www.honeywell.com/sites/portal?smap=aerospace&page=Literature&theme=T5.

SOURCES

ADXL330 Accelerometer

Analog Devices | www.analog.com

HMC5843 Compass IC

Honeywell | www.honeywell.com/magneticsensors

PIC18F2450 Microcontroller

Microchip Technologies, Inc. | www.microchip.com



Easy (E)mbed

An Alternative Approach to Embedded Programming

Thanks to the march of silicon designing hardware these days is as simple as choosing which of the latest and greatest wonderchips to use. Ah, but software, well, that's another story. There's no Moore's Law for software and it isn't getting any easier. Maybe it's time for a change.

After all these years in the chip biz, one trend is clear to me. Ironically, it's the "hardware" that's gotten "easy," while writing "software" is "harder" than ever.

Actually, the irony goes deeper than that. After all, at a certain level—namely, chip design using Verilog or VHDL—creating hardware is little different from creating software. As I've said before, if you see an engineer scratching their head staring at a screen, the only way to tell whether they're designing software or hardware is to check and see if they're wearing shoes.

It's the users of the chips (i.e., the board- and box-level hardware designers) that have it easy. Thanks to Moore's Law, hardware design for the vast majority of embedded designs involves little more than dropping a few do-it-all chips onto a PCB. I was reminded of the fact when I recently fired up my first computer, a circa-1970s IMSAI. (Refer to my October 2008 article, "The Way We Were," which appeared in *Circuit Cellar News Notes*, Vol. 4.) With hundreds of chips and thousands of connections, that was "hardware" the hard (and expensive) way. These days, doing a similar hardware design would involve little more than choosing which 50-cent MCU to use.

Meanwhile, software stays hard. Yes, the tools get incrementally better and reuse is an easy way out. But the problem is that the aforementioned growth in hardware capability has simply fueled demand for more software features. You've heard the old saying that software, like a gas, expands to fill all available space. That old-timey, 8-bit micro chewing through a 1,000-line program has evolved into a fancy 32-bit MCU that hungers for a million lines of code. Feed me!

It's no surprise that in this month's column you'll see the usual miracle-working MCU. But this time, the story is also about the software tools as much as the silicon. Is it time for a change we can believe in?

MACRO MICRO

The LPC236x lineup is just one horse in NXP Semiconductors's stable of ARM-based flash memory MCUs (see [Figure 1](#)). Keep my

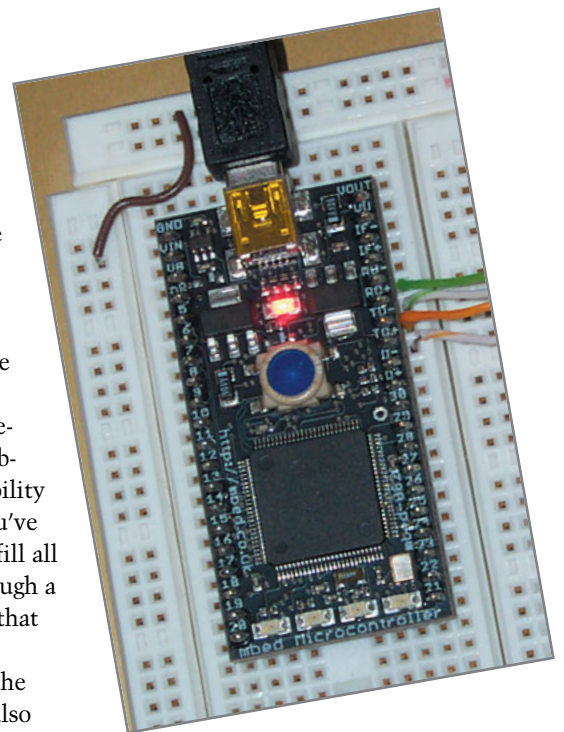


Photo 1—The LPC263x-based mbed module uses a DIP form factor, a great option for easy prototyping. Note the "transformerless" Ethernet connection on the right side.

“hardware is easy, software is hard” premise in mind as you contemplate the long list of features packed onto this 100-pin hotrod.

Under the hood, we find a venerable ARM7 core—not the newest micro on the block, but nevertheless running at up to 72 MHz, which is more than adequate for blue-collar embedded apps. Different models come with 128, 256, or a whopping 512 KB of flash memory, which is more than enough to keep the programmers busy. The smaller versions (i.e., with 128-KB flash memory) come with 8 KB of general-purpose RAM, while the larger ones boost that to 32 KB.

Different members of the LPC236x lineup offer a mix of high-end I/O capabilities including 10/100 Ethernet MAC and a full-speed (12 Mbps) USB device controller. These bring their own RAM to the party (16 KB for Ethernet and 8 KB for USB), which is also available for general-purpose use by the CPU. There’s a fancy DMA controller and dedicated bussing to keep the data flowing fast with minimal overhead. Other notable I/O options in the lineup include a CAN controller and multimedia card interface.

Along with the big-ticket peripherals, the LPC236x crams in virtually every flavor of serial I/O (UART, I²C, SPI, I²S,

and more), nearly a dozen timers including high-resolution (32-bit), PWM, a watchdog, and an RTC (which adds another 2 KB of battery-backable RAM into the memory mix). The analog bases are covered with a fast (2.4- μ s) six-channel, 10-bit ADC and a 10-bit DAC.

Going beyond the major bullets in the datasheet, I’m impressed with NXP’s attention to embedded details. For example, the memory interface features a sophisticated 128-bit prefetch mechanism to reduce the inevitable flash bottleneck. Bit-twiddling operations were never ARM7’s forte, so NXP includes dedicated bit set and reset registers for I/O ports. The clock generation and power distribution scheme is fully integrated so you can tweak all the dials to minimize power. For example, clocking can be driven on-chip (built-in RC oscillator) and off-chip (main and RTC oscillators), and each peripheral’s clock can be individually controlled (i.e., disabled or divided).

CHIPS & DIP

Although the LPC263x is impressive, the chip itself isn’t really the focus of this month’s column. The fact is, it isn’t even the newest arrow in NXP’s quiver, and it’s no secret all the major MCU suppliers offer parts with similar capabilities. The most amazing thing about the LPC263x is that it really isn’t all that amazing in this day and age. Referring back to my opening premise, rather than the chip itself, I’d like to focus on a unique approach to software development that that comes with it.

The “mbed” module shown in [Photo 1](#) highlights how hardware has become “easyware.” Just plop down the MCU and a few LEDs, and voila, instant EV board. Actually, there are a couple of extra housekeeping chips

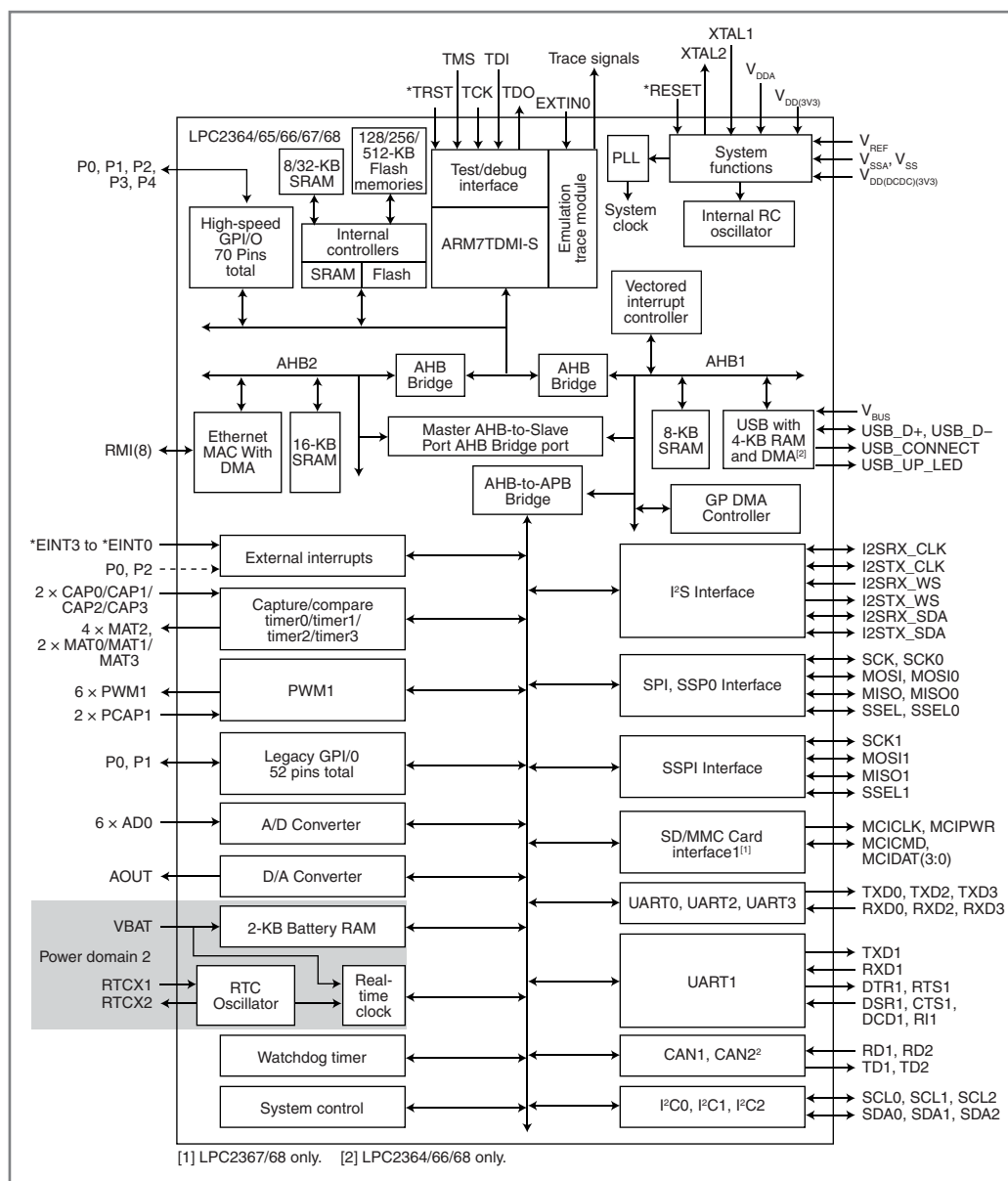


Figure 1—In the old days, the hardware design for a 32-bit micro with almost half a meg of memory, Ethernet, USB, and a bunch of I/O was “hard.” Now, thanks to flash MCUs like the NXP LPC263x, it’s easy. Software? That’s another story.

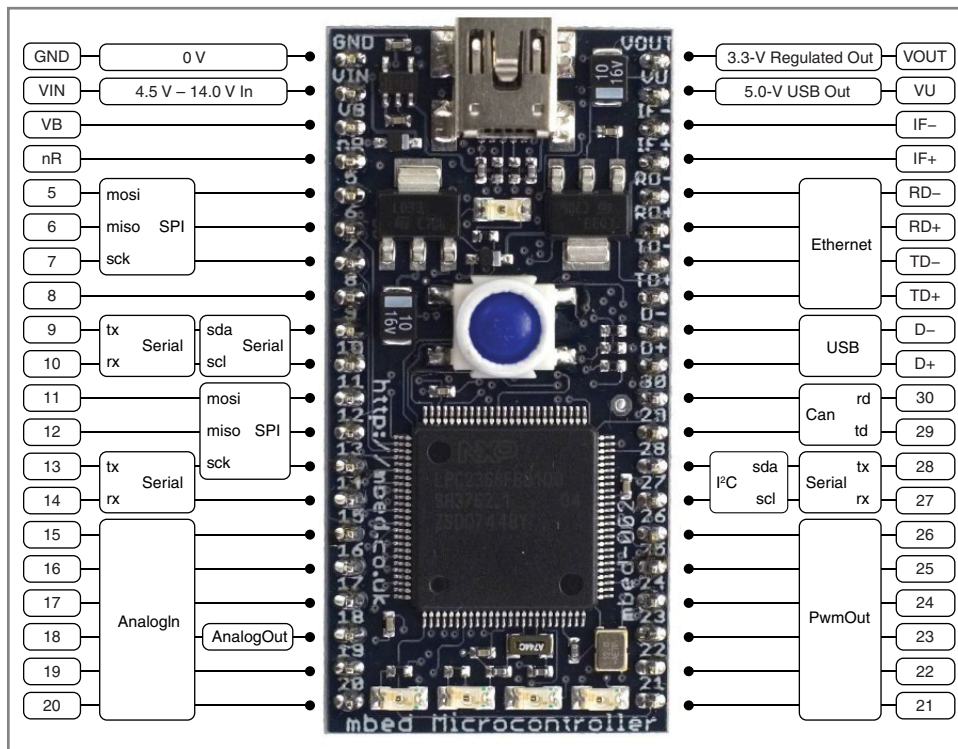


Figure 2—The mbed module DIP form factor makes hardware experimenting and prototyping, well, easy.

on the underside of the board starting with a separate LPC MCU devoted to handling the USB interface with the host PC. The extra MCU frees up the main LPC USB interface for your application and minimizes the intrusiveness of host PC communication. There's also an Ethernet transceiver (aka "PHY") from National Semiconductor (DP83848). Getting on the 'Net can be as easy as adding an RJ-45 connector, especially since the National part has a "transformerless" option that uses capacitors, instead of a transformer, for coupling.^[1]

Over the years, I've seen a lot of "evaluation boards," usually one of two varieties, the first being the older-school larger boards and more recently all manner of cute USB gadgets. To my mind, the mbed module, with its DIP plug-in approach, is a smart way to go (see [Figure 2](#)). Although small and inexpensive like other USB plug-ins, it is breadboard-friendly for prototyping and experimentation, and it could even make sense for low volume production.

Usually, it's about now that I'm taking a coffee break while gigs of the typical "C" IDE get installed on

my PC. And it's here where mbed thinks outside the box, literally. To make a long story short, there's zero software to install because the tools are web-based.

Here's how it works. All you do is connect the mbed module to a USB port and it is recognized by your PC as a generic mass-storage device just like a typical thumbdrive. Open the mbed

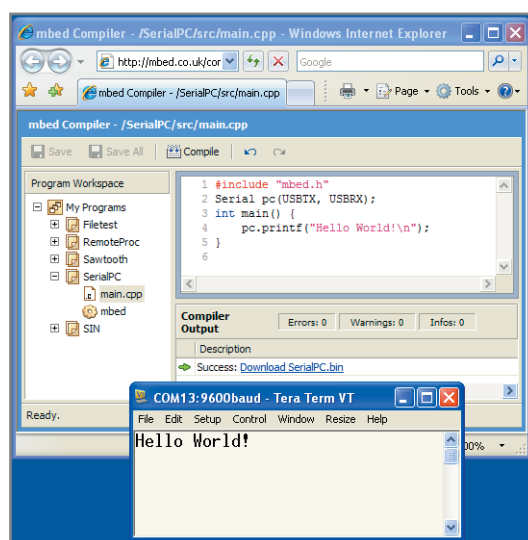


Photo 2—Absent a debugger, you'll need to fall back on PRINTF to test your program. Fortunately, there are built-in functions that implement a virtual serial port via USB so you can use a terminal emulator on the PC to see what's going on.

"drive" and look in the directory to find the "mbed.html" file. Double-click it and it will open with your browser just like any other web page. Voila, you're on the air and ready to start programming with nary a byte of software installed on your PC.

Invoking the "Compiler" link takes you to a web page that acts as an extremely simplified GUI. I guess you could call it a "WUI" (i.e., "Web-based User Interface") that pretty much boils down to basic file operations (create, open, save) and a button that says "Compile." Click that and a moment later a link appears that allows you to download the compiled code to your mbed "drive." Then push the Reset button on the mbed module and the LPC263x flashes itself with the new code and you're off to the races.

Needless to say, this is a different way of doing things that has rather profound implications worthy of further contemplation. But for now, let's hold those thoughts while we take a closer look at a web-based vision of what might be called "Embedded Programming 2.0."

IT'S ALL "C" TO ME

Now that you've compiled and downloaded your code, it's time to get up to speed with the debugger. That doesn't take long because there isn't one. Time to resurrect the old ways and use PRINTF for debugging. To that end, the mbed library includes a feature that routes serial traffic across the USB connection to a terminal emulator running on your PC (see [Photo 2](#)).

Other handy built-in routines are shown in [Table 1](#). For example, AnalogOut allows you to set the level on the LPC263x DAC output pin (see [Photo 3](#)). For convenience, two versions of the AnalogOut function are provided. The one used here expresses the rail-to-rail output as a fraction between 0 and 1, while another version uses unsigned 16-bit integers between 0x0000h and 0xFFFFh.

The LocalFileSystem function

enables you use standard “C” file operations (fopen, fprintf, fclose, etc.) on the mbed flash drive. This is noteworthy because it allows straightforward interchange of data between your program and a PC via the FAT file system (see [Photo 4](#)). For instance, it would be easy to use an mbed module as a data logger, creating a log file on its own flash memory that could subsequently be opened with a PC application (e.g., word processor, spreadsheet, etc.). Similarly, a PC could be used to “configure” an mbed module by writing a data file to the flash drive which the mbed application could query after being disconnected from the PC.

Going beyond the built-in functions, the mbed webpage has a “Cookbook” link that takes you to a bunch of add-on projects and examples. Many of these show how to connect the mbed module to popular I/O devices (e.g., LCDs, sensors, motors, etc.) and there are some handy utilities (e.g., a routine to set the real-time clock).

I found some interesting dishes in the “Cookbook,” starting with a simple web server that exercises the mbed Ethernet interface (see [Photo 5](#)). Taking advantage of the National transceiver’s “transformerless” feature mentioned earlier, I was able to jack-in simply by cutting one end off an Ethernet cable and connecting the transmit and receive wires directly to the mbed module.

Another demo takes advantage of the mbed “Remote Procedure Call” (RPC) capability. In essence, RPC is a mechanism that provides hooks to the functions shown in Table 1. The example in [Photo 6](#) shows how to assign a service name (“pot”) to an mbed function (AnalogIn). Once the logical connection is established, the mbed functions are exposed to the outside world and you can use the terminal to interact with them.

NEW IDEA

Now that you’ve got a glimpse under the hood, let’s contemplate the big picture and examine the various factors

Routines	Uses
PwmOut	Set a PWM output period, frequency, duty cycle
SPI	SPI master configuration (frequency, format) and data transfer
AnalogOut	Set analog output voltage level
Wait	Wait specified number of milliseconds/microseconds
Timer	General-purpose timer (seconds/milliseconds/microseconds)
DigitalIn	Read the state of an input pin
LocalFileSystem	Use C file functions to access mbed disk drive
Debug	Output error message to PC and debug LEDs
Ticker	Call a function at a recurring interval
DigitalOut	Set the state of an output pin
BusOut	Set the state of a collection of output pins
AnalogIn	Read analog input voltage level
Serial	UART configuration (bitrate, format) and data transfer
RTC	Set and read the real-time clock
Timeout	Call a function at a future time
BusIn	Read the state of a collection of input pins
SPI3	3-wire SPI master configuration (frequency, format) and data transfer
Stackheap	Functions return code, data, stack and heap addresses
I2C	I ² C master configuration (frequency) and data transfer

Table 1—The mbed library includes built-in functions that make programming easier. The more software someone else writes, the less you have to.

comprising the altogether unique mbed experience. “Keep it simple” is advice that’s clearly taken as gospel by the mbed folks. I must say I found it refreshing that I didn’t have to deal with the clutter and complexity of the usual big-ticket IDE. There’s no doubt it lacks the features and options that a professional programmer is used to. But on the other hand, the mbed approach makes things easy for anyone who just wants to get a simple app working without a lot of fuss and bother. Recalling my opening “hardware easy, software hard” premise, it seems to me that a straightforward solution to the “software crisis” is simpler tools that enable more folks to write their own code.

Simplicity is nice, but it isn’t really what makes mbed stand out from the crowd. Rather, it’s the fact the entire experience is web-based that is totally unique and would seem to have rather significant implications.

For instance, the fact the GUI is a web-browser feels quite liberating. Notably, it sweeps away all religious issues related to your platform of choice. PC, Mac, Linux—have it your way. Heck, I suppose you could figure out a way to program mbed with your iPhone or any other web-capable gadget.

Another thing that felt very liberating was being totally relieved of “IT” responsibilities. Keeping up with revising, upgrading, and migrating (e.g., XP to Vista) a pro toolchain can be a chore, one I’m glad to let the mbed folks take care of. It’s all the more liberating, for your wallet, to realize you’re getting access to some pretty fancy software (ARM’s own Realview compiler) for free.

On the other hand, I found my browser (Explorer) had some trouble with GUI basics. Keyboard and screen updates are a bit sluggish and the interface can be quirky or lacking compared to a real editor

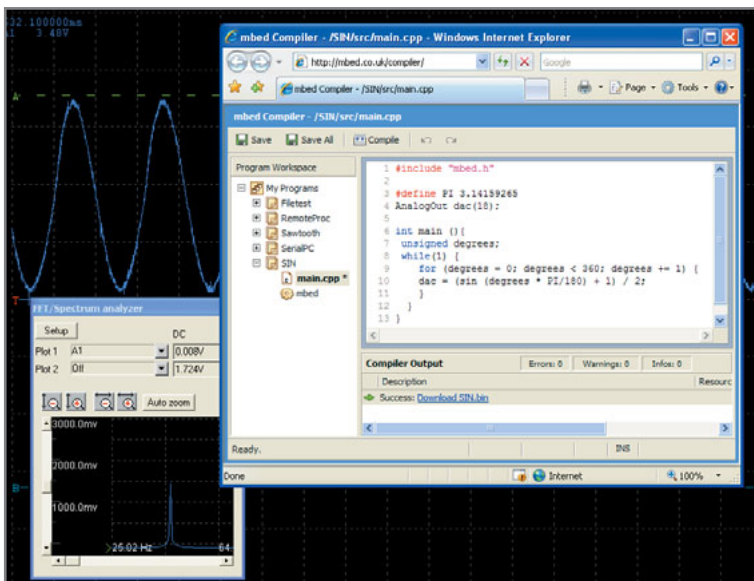


Photo 3—Another built-in function, AnalogOut, makes it easy to drive the MCU’s on-chip DAC.

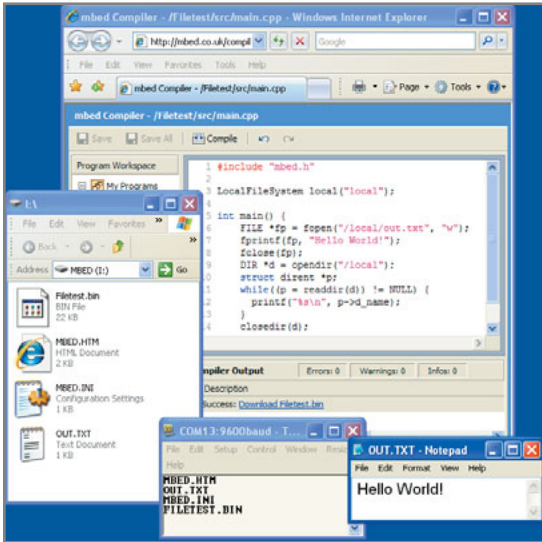


Photo 4—From the PC’s perspective, the mbed module appears as a typical USB thumbdrive. The LocalFileSystem function takes advantage of that by making it easy for an application program to exchange file data with the PC.

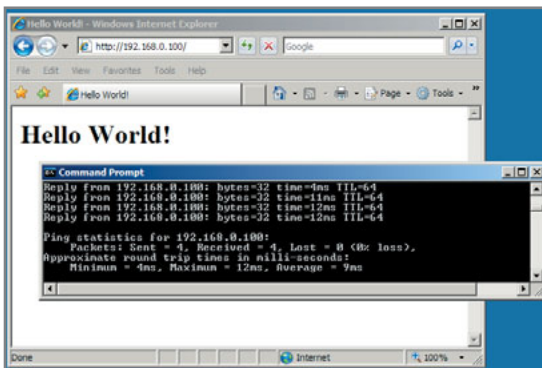


Photo 5—One of the projects in the mbed “Cookbook” is a mini-web server that made it easy to check out the “transformerless” Ethernet feature.

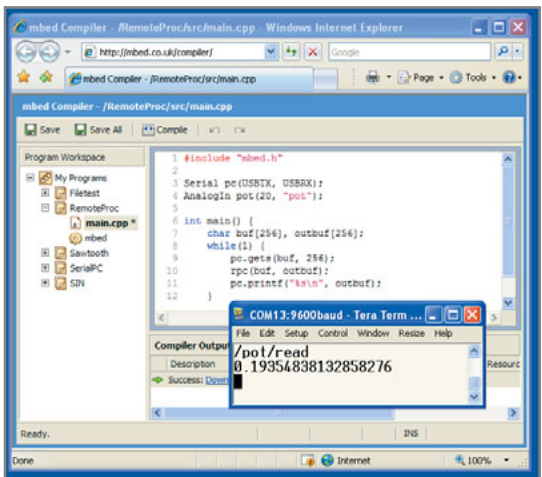


Photo 6—Remote Procedure Call (RPC) is an interesting feature that provides hooks to access the mbed built-in functions. This example shows the AnalogIn function being invoked via the console.

when it comes to things like cut and paste, find and replace, etc. And though intuitive, having to click through Windows hoops (e.g., “Replace existing file?”) every time you compile gets old really fast.

The web angle puts an interesting spin on issues like community, collaboration, commerce, reuse, and support. In principle, a web-based solution like mbed could evolve to become a comprehensive programming “portal.”

Of course, there are issues like security and intellectual property that can’t be overlooked. It’s easy to forget everything you’re doing is getting sent across the Internet. In fact, as best as I can tell, the programs you enter and save aren’t even stored on your own PC. If Big Brother is watching me hack away, so be it. Maybe he’ll even help me get my program working.

FORWARD TO THE PAST?

It’s kind of ironic how similar mbed’s “leave the driving to us” web-based approach is to the way things worked back in the days of mainframe

computers, albeit now on a much grander I-Way scale. After the mainframes came the “Personal Computer,” which itself felt liberating, at least at the time. But when it comes to wrestling with complicated toolchains and IT chores, I’m not sure today’s PC experience feels that liberating anymore.

Of course, the issue of web-hosted software goes beyond the niche of embedded programming to encompass all types of applications. Indeed, when you consider the likes of Google or iTunes, it’s apparent a transition to web-based functionality is already underway.

Keep in mind I’ve been working with a beta pre-release version of the mbed hardware and software. Things may change by the time you read this, so check the NXP web site (see the Sources below) for the latest information. Sure there are lots of questions and no doubt some bumps in the road ahead. But my gut feeling is that the mbed folks are onto something here. The web-based approach won’t totally replace the old way of doing old things (i.e., the traditional toolchain running on the PC), but it doesn’t need to. Instead, consider the mbed solution an alternative approach to embedded programming that may not be industrial strength, but is easy and accessible for all. ☒

Tom Cantrell has been working on chip, board, and systems design and marketing for several years. You may reach him by e-mail at tom.cantrell@circuitcellar.com.

REFERENCE

[1] D. Miller, D. Seely, and T. Ngo, “DP83848 PHYTER Transformerless Ethernet Operation,” Application Note 1519, 2006, www.national.com/an/AN/AN-1519.pdf.

RESOURCE

T. Cantrell, “The Way We Were,” *Circuit Cellar News Notes*, Vol. 4, 2008, www.circuitcellar.com/newsletter/1008.html#1.

SOURCE

DP83848 Transceiver

National Semiconductor Corp. | www.national.com

NXP LPC263x “mbed” module and web-based C IDE

NXP Semiconductors | www.nxp.com/microcontrollers




CAP™

CUSTOMIZE YOUR MCU!

ARM Standard Product + FPGA = CAP

www.atmel.com/products/AT91CAP



NEW Production Mate
the PIC® MCU
PRIME8
Production Programmer
8 Port In-Circuit
Gang Programmer

- SD Card Reader
- 2MB Internal Memory
- Voltage Configuration
- Stand-Alone LCD & Control Buttons
- Use CCSLOAD Programmer Control Software

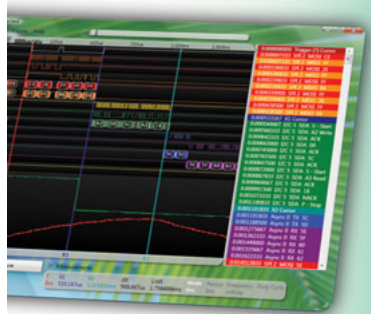
\$899

www.ccsinfo.com/eight



Discounts available for CCS compiler owners

Need something more portable? **LOAD-n-GO \$199**

See It & Solve It
with
USBee Test Pods

- Logic Analyzers
- Oscilloscopes
- Signal Generators
- Protocol Analyzers
 - I2C, SPI, ASYNC, CAN
 - 1-Wire, PS/2, USB
 - I2S, SMBus, Serial
- Configurable
- Programmable
- High Speed USB
- PC-Based
- And start at \$139

USBee™
USB based Electrical Engineer

www.USBee.com



Embedded & Network Computing Technologies

Let your Calao's Embedded Computer communicate with the world!



- 1 Wire, Spi, R232, I2C, Micro SD, Spi RTC
- Magneto 3 axis, Accelero 3 axis, Gyro 3 axis, Micro SD, RTC
- WiFi, Micro SD, RTC
- Bluetooth, Micro SD, RTC

www.calao-systems.com

Add USB to Your Designs

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



USB Complete The Developer's Guide
Fourth Edition
Jan Axelson

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



Mechanical parts
Wheels & casters
Motors & servos
Robot chassis

RP5 Tracked Chassis
item #1060: **\$49.95**

Electronics
Controllers & sensors
Cables & batteries
Discrete components

TReX Jr Dual Motor Controller
item #767: **\$59.95**

1-877-7-POLOLU
www.pololu.com
6000 S. Eastern Ave. 12D, Las Vegas, NV 89119

USB Bus Analyzers



Packet-Master™ - best value in USB1.1/2.0 USB analyzers and generators. Identify USB problems fast, fine-tune performance, easily view Host Commands, emulate host/device sequences, etc.

USB12 (USB1.1)	\$699
USB480+ (USB1.1/2.0)	\$1199
USB500AG (USB1.1/2.0/Gen)	\$1399

Saelig.com
UNIQUE electronics
1-888-7SAELIG
info@saelig.com
www.saelig.com



TPC-43A
Fanless Wall-mountable 32bit 200Mhz ARM9 CPU SDRAM/FLASH with Linux 2.6.2x with drivers & GUI



RTC w/battery
16bit Audio
10/100 Ethernet
RS232, IrDA
USB Host
USB Device
SD/MMC card
Optional: Java
- Bluetooth
- ZigBee
- WiFi

4.3" WQVGA (480x272) x 65K Color Touch Screen LCD w/backlight

From \$339

www.medallionsystem.com




Accelerate Your Debugging

USB, I2C, and SPI Development Tools

www.totalphase.com

Use code **DZXAXC** for 10% off
expires: 6/30/2009

IDEA BOX

THE DIRECTORY OF PRODUCTS AND SERVICES

AD FORMAT: Advertisers must furnish digital submission sheet and digital files that meet the specifications on the digital submission sheet. **ALL TEXT AND OTHER ELEMENTS MUST FIT WITHIN A 2" x 3" FORMAT.** Call for current rate and deadline information. E-mail adcopy@circuitcellar.com with your file and digital submission or send it to IDEA BOX, Circuit Cellar, 4 Park Street, Vernon, CT 06066. For more information call Shannon Barraclough at (860) 875-2199.

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

USB

Add USB to your next project—it's easier than you might think!

- USB-FIFO up to 8 mbps
- USB-UART up to 3 mbps
- USB/Microcontroller boards pre-programmed with firmware
- 2.4GHz ZigBee™ & 802.15.4
- RFID Reader/Writer

Absolutely NO driver software development required!

www.dlpdesign.com



The PLC with everything you ever wished for!
Lead the Green Revolution with our Smart Buildings control model.

TRI
Triangle Research International, Inc.

From the Programmable Logic Controller specialists who brought you the Super M-series, now **The Ultimate F-series**

Infra-Red Remote Control - A/V equipment control

OEM Prices below \$400 (display not included)

12 Relay Outputs 24VAC/DC or 250VAC

1x RS232
2x RS485

12 Analog I/Os (12-bit, 0 - 10V)

4 Super Outputs - PWM, Steppers, Light-dimmer

ETHERNET - Modbus/TCP
Internet - Programming, Emails, PLC to PLC

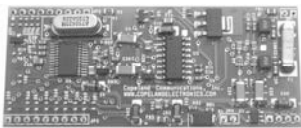
16 Digital Inputs 24VAC or DC - 3x hi-speed encoders

XBEE® Socket for ZIGBEE Interface Option
XBEE is a reg. trademark of Digi International

Battery-Backed Real Time Clock

www.tri-plc.com/ccl.htm At prices below most other basic PLCs! Call 1 877 874-7527

Low Cost Embedded OEM Modems & Device Servers



Low Cost 56K V.92 MNP4 Modems, Dialup Device Servers, Ethernet and WiFi—all in one form factor!

- 300 - 56K baud V.92
- V.42 - V.42bis - MNP4
- Global Compliance—5KV
- FCC, CE and TBR21 approvals
- AT command set
- SocketModem™ compatible
- Low Power—35mA
- Small Footprint 1.045" x 2.54"
- Wireless Modules—902-928MHz - 33Kb
- Auto Error Detection & Flow Control
- Modules are H/W & S/W compatible
- Custom antennas
- USB 2.0 Modules—Adapt legacy RS-232 DB9 to USB
- WiFi, Ethernet available soon!

www.copelandcommunications.com

CCI COPELAND COMMUNICATIONS, INC.
(614) 475-1690

I2C SPI 1 Wire

PC to PC Adaptor
I2C2PC
www.i2cchip.com/i2c2pc.html

Normal Galvanic Isolated Bluetooth

3 Separate Buses 5V & 3V
Simple ASCII Interface
Cross Platform - All OS

www.i2cchip.com

Embedded Ethernet
ONLY \$98

NEW! picoFlash

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

Development Kit Includes:

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

\$129 Development System

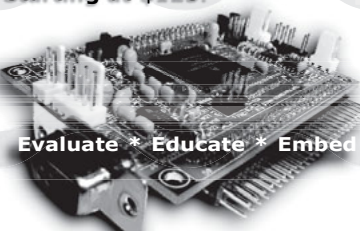
Call 530-297-6073 Email sales@jkmicro.com
On the web at www.jkmicro.com

JK microsystems

Adapt9S12XDP512
Modular Prototyping System

- * Robotics and Mechatronics
- * Electronic Fuel Injection
- * Freescale 9S12XDP512
- * RTOS-capable

Starting at \$125!



Evaluate * Educate * Embed

Program in
Assembler, BASIC, C, and Forth

www.TechnologicalArts.com

TECHSOL
TECHNICAL SOLUTIONS INC.

TPC-43A

Fanless Wall-mountable 32bit 200Mhz
ARM9 CPU SDRAM/FLASH with Linux
2.6.2x with drivers & GUI



RTC w/battery
16bit Audio
10/100 Ethernet
RS232, IrDA
USB Host
USB Device
SD/MMC card
Optional: Java
- Bluetooth
- ZigBee
- WiFi

4.3" WQVGA (480x272) x 65K Color
Touch Screen LCD w/backlight

From \$339

www.medallionsystem.com



Weather Instruments
for PCs

AAG
electrónica

www.aagelectronica.com

ALL ELECTRONICS
CORPORATION

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

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

PRINTED CIRCUIT BOARDS

QUALITY PRODUCT
FAST DELIVERY • COMPETITIVE PRICING

- Aluminum-Backed PCB
- Single & Double Sided
- SMOBC, RoHS
- LPI Mask
- Through Hole or SMT
- Nickel & Gold Plating
- Routing or Scoring
- Electrical Testing
- Artwork or CAD Data
- Fast Quotes
- Flex Circuits

10 pcs. (3 Days)
1 or 2 Layers \$249

10 pcs. (5 Days)
4 Layers \$695
(up to 30 sq. in. each)
Includes Tooling, Artwork,
LPI Mask & Legend

PROTOTYPE THROUGH PRODUCTION

PULSAR, INC.
9901 W. Pacific Ave. • Franklin Park, IL 60131
847-233-0012 • Fax: 847-233-0013
www.pulsar-inc.com • Email: sales@pulsar-inc.com

Giga-snap BGA Sockets and Adapters

Giga-snap BGA SMT Adapters
allow affordable socketing

- Ultra low insertion force
- Up to 2000 pins
- GHz bandwidth
- 0.75 to 1.27mm pitch
- Same CTE as PCB
- Industry's toughest socket
- Maximum solderability

RoHS



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

Order online at: www.melabs.com **microEngineering Labs, Inc.** Phone: (719) 520-5323
Fax: (719) 520-1867
Box 60039
Colorado Springs, CO 80960

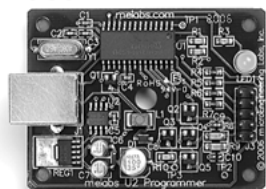
Development Tools for PIC® Microcontrollers

USB Programmer for PIC® MCUs \$89.95 (as shown)

RoHS Compliant

Programs PIC MCUs including low-voltage (3.3V) devices

Includes Software for Windows 98, Me, 2K, & XP

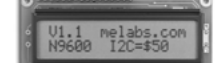


With Accessories for \$119.95:
Includes Programmer, Software, USB Cable, and Programming Adapter for 8 to 40-pin DIP

Serial LCDs

2-line x 16 \$39.95
4-line x 20 \$49.95

Quantity Discounts Available!



LAB-X Experimenter Boards

Pre-Assembled Boards Available for 8, 14, 18, 28, and 40-pin PIC® MCUs


2-line, 20-char LCD Module
9-pin Serial Port
Sample Programs
Full Schematic Diagram

Pricing from \$79.95 to \$349.95

PICPROTO™ Prototyping Boards

Double-Sided with Plate-Thru Holes
Circuitry for Power Supply and Clock
Large Prototype Area
Boards Available for Most PIC® MCUs
Documentation and Schematic

Pricing from \$8.95 to \$19.95



BASIC Compilers for PICmicro®

Easy-To-Use BASIC Commands
Windows 9x/Me/2K/XP Interface

PICBASIC™ Compiler \$99.95
BASIC Stamp 1 Compatible
Supports most 14-bit Core PICs
Built-In Serial Comm Commands

PICBASIC PRO™ Compiler \$249.95
32-bit signed variables and math operations
Supports Microchip PIC10, PIC12, PIC14, PIC16, PIC17, and PIC18 microcontrollers
Direct Access to Internal Registers
Supports In-Line Assembly Language
Interrupts in PICBASIC and Assembly
Built-In USB, I2C, RS-232 and More
Source Level Debugging

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

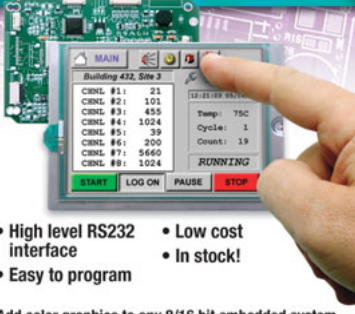


Free Training for Rabbit Developers

Learn More at
Rabbit-U.com

08078

Add a color touch interface to your embedded product!



CINCL #1:	21
CINCL #2:	101
CINCL #3:	455
CINCL #4:	1024
CINCL #5:	39
CINCL #6:	200
CINCL #7:	5660
CINCL #8:	1024

Temp: 75C
Cycle: 1
Count: 19
RUNNING

START LOG ON PAUSE STOP

- High level RS232 interface
- Easy to program
- Low cost
- In stock!

Add color graphics to any 8/16 bit embedded system. Easy, fast and flexible. Up and running in hours!

REACH
TECHNOLOGY INC.

www.reachtech.com • 510-770-1417

842 Boggs Avenue • Fremont, CA 94539

World Leading Driver-Free Win/Mac/Linux USB Chips



Looks like Flash Drive

A/D - I/O - SPI - I2C
BASIC programming
USB Status & Control

USB-DAQ

- USB-DAQ / FileSys sensing & logging
- USB-to-UART/I2C/SPI and I/O expanders
- No microcontroller programming required
- Add USB to your products in a day!

www.hexwax.com - Mouser - Farnell - Digikey

Amazing PIC programmer

Most devices supported
ICSP, SQTP, & copy limits

\$32 at Digikey & Mouser

www.flexipanel.com

Actual size - patents pending

RIGOL Beyond Measure

NKC ELECTRONICS

Electronic 2008 PRODUCT of the YEAR AWARD

From \$595

Rigol Technologies DS1000E series
Up to 1GSa/s and 1Meg memory
50MHz and 100MHz models - TFT LCD - USB
Advanced triggering: Edge, Pulse, Video, Slope, Alt

Zeroplus LAP-16032U Logic Analyzer
16-channel - 100MHz - USB 2.0
SPI - i2c - UART - 7-segment
2 free additional protocols

\$120

Open Source Hardware
Arduino - Freeduino - Seeduino boards
Arduino Shields
BlinkM

Arduino Duemilanove \$30

www.nkcelectronics.com

OFFICIAL RIGOL DISTRIBUTOR GLOBAL SHIPPING

★ SURVEY SPONSOR

What's the missing component?


JAMECO ELECTRONICS



Take the challenge at
www.Jameco.com/missing

CIRCUIT CELLAR

FOR COMPUTER APPLICATIONS



Make sure you're signed up to receive Circuit Cellar's monthly electronic newsletter. *News Notes* will keep you up to date on Circuit Cellar happenings. Stay in the loop!

Register now. It's fast. It's free.
www.circuitcellar.com/newsletter/

Full Speed It writes your USB Code!

NEW! HIDmaker FS for Full Speed FLASH PIC18F4550

Creates complete PC and Peripheral programs that talk to each other over USB. Ready to compile and run!

- Large data Reports
- 64,000 bytes/sec per Interface
- Easily creates devices with multiple Interfaces, even multiple Identities!
- Automatically does MULTITASKING
- Makes standard or special USB HID devices

NEW! "Developers Guide for USB HID Peripherals" shows you how to make devices for special requirements.

Both PC and Peripheral programs understand your data items (even odd sized ones), and give you convenient variables to handle them.

PIC18F Compilers: PICBASIC Pro, MPASM, C18, Hi-Tech C.

PIC16C Compilers: PICBASIC Pro, MPASM, Hi-Tech C, CCS C.

PC Compilers: Delphi, C++ Builder, Visual Basic 6.

HIDmaker FS Combo: Only \$599.95

Trace SYSTEMS, Inc.

DOWNLOAD the HIDmaker FS Test Drive today!
www.TraceSystemsInc.com
301-262-0300

BREAK THE TIES, GO WIRELESS!

CCS: WIRELESS

ember ZIGBEE™

PIC® MCU DEVELOPMENT KIT

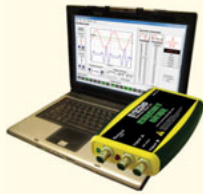
NEW APPLICATIONS & ADDITIONAL EXAMPLES

PRICES START AT \$249

262.522.6500 x35

WWW.CCSINFO.COM/EMBER3

7 in 1 Scope !



CircuitGear CGR-101™ is a unique new, low-cost PC-based instrument which provides the features of seven devices in one USB-powered compact box: 2-ch 10-bit 20MSa/sec 2MHz oscilloscope, 2-ch spectrum-analyzer, 3MHz 8-bit arbitrary-waveform/standard-function generator with 8 digital I/O lines. It also functions as a Network Analyzer, a Noise Generator and a PWM Output source. What's more – its open-source software runs with Windows, Linux and Mac OS's!

Only \$180

Saelig
unique electronics

1-888-73AELIG
info@saelig.com
www.saelig.com

Solve complex signal acquisition problems...

- positioning & control
- environmental
- acceleration
- transients
- pressure
- vibration
- sonar
- GPS
- Linux Driver
- Guaranteed in stock
- Many newly added features
- 16-bit analog inputs and outputs
- Million sample FIFO eliminates interrupts
- Wide analog input and output ranges
- -40°C to +85°C Standard
- Order 24/7, fast and easy.



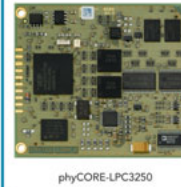
www.stx104.com
Apex Embedded Systems
help@stx104.com • 608-256-0767 x24

PHYTEC

*** Inside great products. Behind great ideas.

phyCORE® System on Modules:

- shorten time-to-market
- reduce development costs and avoid substantial design issues and risks
- Windows® Embedded CE and Linux BSPs (processor-dependent)
- \$129/unit benchmark price at 1K for ARM9-based SOM
- Design Services available to assist with deployment into target applications



phyCORE-LPC3250

ARM11: LMX35, LMX31
ARM9: LMX27, LPC3250, LPC3180
Cortex M3: STM32F103
ARM7: LPC2294
XScale: PXA270
x86: Z510, Z520, Z530 (Atom®)
Blackfin: ADSP-BF537
Coldfire: MCF5485
PowerPC: MPC5554, MPC5567,
MPC5200B, MPC565, MPC555

phyCORE® Rapid Development Kits include SOM, Carrier Board, LCD (kit specific), schematics, software, free BSP for applicable kits and a start-up guarantee. The Carrier Board serves as a target reference design, allowing the SOM to easily port to the user's target hardware.



www.phytec.com | 800.278.9913 | www.phycore.com

www.can232.com

Only \$108 €89

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

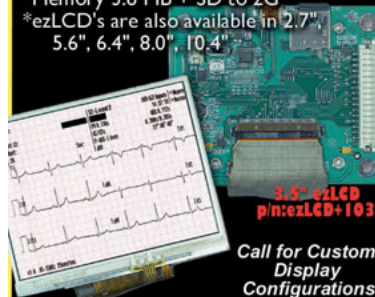
CANUSB Features:
Free ActiveX component
PC, MAC & Linux support
Both 11 & 29 bit ID support
Simple CAN logger included
Free Threaded Windows DLL
Firmware upgradable via USB
Sample programs in C, C++, VB, Delphi, C#, PureBasic etc.
No need for external power
Works up to 1Mbit CAN
Supports RTR Frames
USB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

Only \$154 €129

www.canusb.com

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

- *Versatile Programming LCD module
- *USB, SPI, RS232/TTL Interfaces
- *Bright 350 Nit LED Display (320 x 240)
- *Integrated Touch Screen
- *LUA Scripting Language capable- For stand alone embedded apps
- *Memory 3.8 MB + SD to 2G
- *ezLCD's are also available in 2.7", 5.6", 6.4", 8.0", 10.4"



3.5" ezLCD
p/n: ezLCD+193

Call for Custom Display Configurations

www.earthlcd.com

I²C™/SMBus

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



MCC
Micro Computer Control

FC is a trademark of Philips Corporation

www.mcc-us.com

CAN controllers

- Embedded, Standalone, C/C++ Programmable
- CAN to Ethernet, USB, CF, TFT, RS232/485, ADC, DAC, I/Os, Relays,...
- CAN-Engine™ Controller Area Network (CAN) with variable baud rate, low level register access, Ethernet, CompactFlash, ADCs...



\$39 OEM

\$99 Qty 1

60+ Low Cost Controllers with TFT, 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



Going Wireless is Easy!

Zigbee™

Wireless Mesh Network

NEW USB Zigbee™ Stick offers an easy way to Zigbee™ enable PC's

- ETRX2 USB Based on Ember Single Chip Zigbee™ 802.15.4 Solution
- No need for RF Design Experience
- 2.4 ISM Band

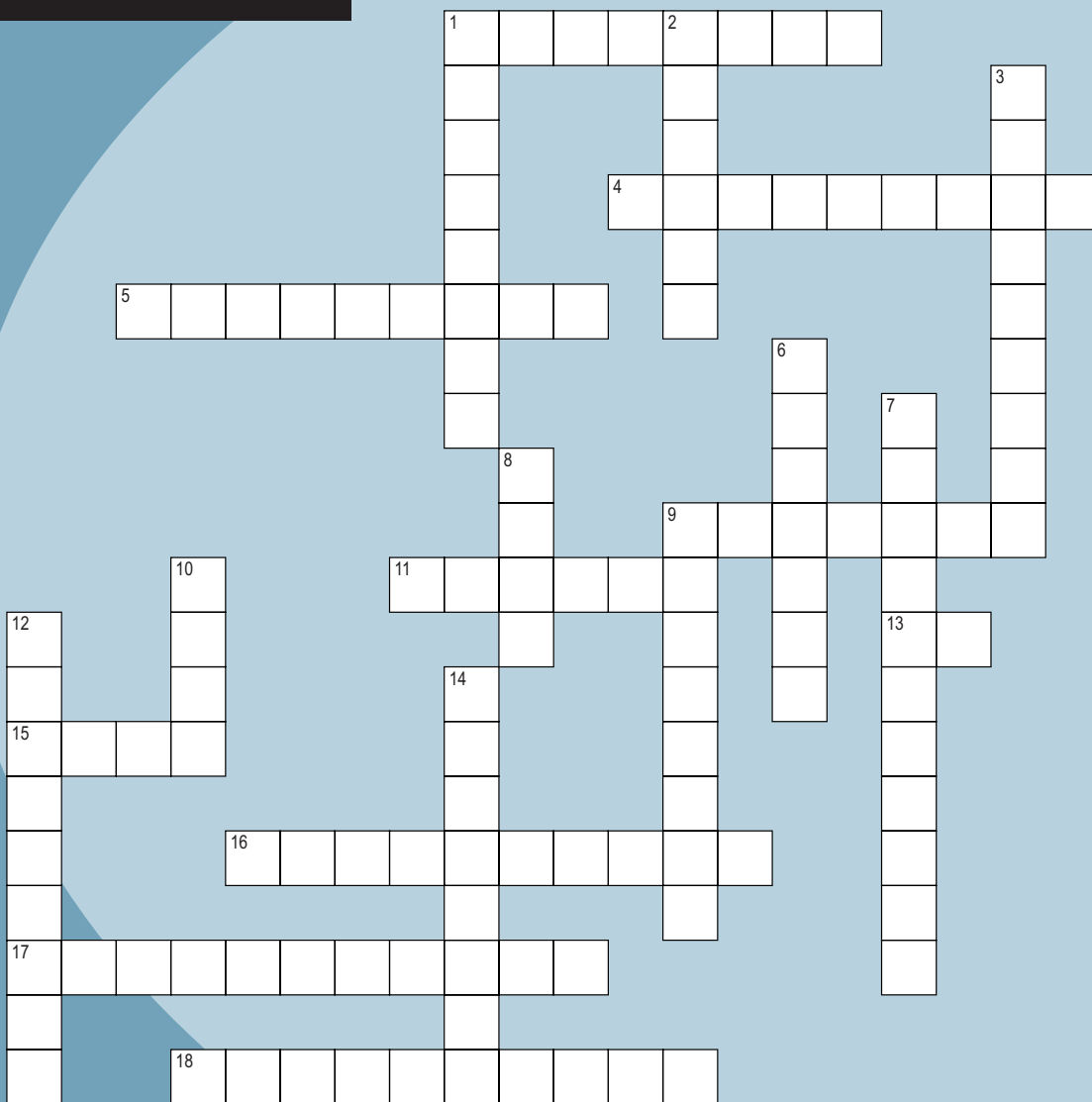


LE MOS
INTERNATIONAL sales@lemosint.com

Call Toll Free 1-866-345-3667 or email at

www.lemosint.com

CROSSWORD



Across

1. xfr
4. 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
5. No change in polarity
9. Damaging or problematic programs like viruses or spyware
11. Example: "mcu_fanatic2009"
13. Carbon copy
15. "P" in P2P
16. A test to determine if current is moving
17. Signal strength reduction
18. *Circuit Cellar News Notes* is an electronic what?

Down

1. 10^{12} flops
2. Character series
3. A board with sockets for other boards
6. An autoresponder
7. statC
8. Electronic mag
9. Data about data
10. The second square number
12. Software not released
14. Convert analog to digital

The answers are available at
www.circuitcellar.com/crossword.

INDEX OF ADVERTISERS

The Index of Advertisers with links to their web sites is located at www.circuitcellar.com under the current issue.

Page		Page		Page		Page	
75	AAG Electronica, LLC	19	ExpressPCB	67, 73	Lakeview Research	9, 39	Rabbit, A Digi International Brand
11	AP Circuits	8	ezPCB	77	Lawicel AB	76	Rabbit, A Digi International Brand
20	ARM	76	FlexiPanel Ltd.	9, 77	Lemos International Co. Inc.	76	Reach Technology, Inc.
75	All Electronics Corp.	50	General Circuits	27	Linx Technologies	73, 77	Saelig Co.
77	Apex Embedded Systems	32	Grid Connect, Inc.	77	MCC (Micro Computer Control)	17	Sealevel Systems
7, 73	Atmel	25	HI-TECH Software, LLC	75	microEngineering Labs, Inc.	13	SEGGER Microcontroller Systems LLC
33, 73	CWAV	10	HobbyLab, LLC	49	Mouser Electronics	C3	Tech Tools
16	CadSoft Computer, Inc.	74	I2CChip	76	NKC Electronics	73, 75	Technical Solutions, Inc.
11, 73	Calao Systems	40, 41	ICbank Inc.	C2	NetBurner	2, 3	Technologic Systems
63	Comfile Technology, Inc.	1	Imagineering, Inc.	67	Nurve Networks LLC	75	Technological Arts
74	Copeland Communications	75	Ironwood Electronics	21	PCB-Pool	77	Tern, Inc.
76	Custom Computer Services, Inc.	32, 34	JKmicrosystems, Inc.	C4	Parallax, Inc.	73	Total Phase, Inc.
74	DLP Design	74	JKmicrosystems, Inc.	77	Phytec America LLC	76	Trace Systems, Inc.
10	DesignNotes	5, 76	Jameco	47, 73	Pololu Corp.	74	Triangle Research Int'l, Inc.
27	EMAC, Inc.	67	Jeffrey Kerr, LLC	75	Pulsar, Inc.		
77	Earth Computer Technologies	48	Keil Software	12	R4		

PREVIEW of July Issue 228

Theme: **Internet & Connectivity**

Internet Weather Display

Master Control: Implement an I²C Master Bus Controller in a FPGA

Web Camera

iMash: Build an Ethernet-Controlled HERMS

LESSONS FROM THE TRENCHES C Start-Up: Get a C Program Up and Running

FROM THE BENCH Embedded Speak: A Text Library for Allophone Translations

SILICON UPDATE LiOn King: A Look at "Battery-in-a-Chip" Technology

ATTENTION ADVERTISERS

August Issue 229 Deadlines

Space Close: June 11
Material Close: June 19

Theme
Embedded Development

Bonus Distribution
Hot Chips; Flash Memory Summit

Call Shannon Barraclough
now to reserve your space!
860.875.2199
e-mail: shannon@circuitcellar.com

PRIORITY INTERRUPT



by Steve Ciarcia, Founder and Editorial Director

Lawns and Ponds

During college I stayed with out-of-state relatives. One of my jobs in “being part of the family” was that I got to mow the lawn every week during the summer. This wasn’t just any lawn, mind you. It was a bloody golf course! No, seriously, it was about an acre of rocks, outcroppings, tree stumps, and hills. OK, I didn’t have to cut it with a manual push mower, but 5 hours behind a non-self-propelled gas mower wasn’t much better. Their attitude was that all that sweat-equity would “build character,” and the hotter and more humid the weather, the better the experience. My perception was that I was “cheap labor.”

I never wanted to deal with this short green stuff ever again, and as absurd as it sounds, I’ve been true to this directive for the last 30-plus years of home ownership. If you look back at the photos in my solar system article series, you’ll see that I don’t have a lawn. My house is completely surrounded by hemlock mulch (spreading 60 yards of mulch may seem equally absurd to some people). :-)

A couple years ago, we bought a “cottage” for vacation getaways. It’s in a very nice “gated community,” where part of the ownership understanding is that people are to maintain their property to specific “community standards” (kinda like a condo agreement). The major concession for me was that these standards dictated that I keep a well-manicured lawn instead of a brown blighted-looking quarter-acre mulch smudge among the rows of green encircled houses. At least that’s how the community’s “little old lady patrol” would have viewed my territorial preferences.

Of course, I completely miss the rationale where people try to grow lawns where none should normally exist—like deserts, swamps, and arid areas. At the cottage, I have a service that cuts the grass weekly, but unless I pour a thousand gallons of water on this lawn two or three times a week, it will be like the Mohave desert. As you might have guessed, along with all my neighbors, the cottage needs an automatic lawn irrigation system.

For those of you readers living in temperate climates, let me just say that a lawn irrigation system is simply a bunch of buried pipes separated into various areas called zones. Each zone is connected to the water supply through a solenoid “zone” valve. The irrigation controller is typically nothing more than an electromechanical timer that turns on a water pump and sequences through the zones on a preset schedule.

Since I’m basically an absentee owner, I have to trust that things still function properly when I’m not there. Certainly I can always count on a call from the management if the lawn starts looking like a dried up oasis or a small pond, but emergency repairs from a distance are both aggravating and expensive. The preventive course was to add supervisory monitoring and remote activated controls like I have in the Connecticut house. When I am at the cottage, I can check oil tank levels, room temperatures, web cams, and more in Connecticut. It made perfect sense to install a similar system in the cottage.

The irrigation controller, albeit electromechanical, seemed straightforward. I monitor the control lines that tell me the power is on, the rain sensor is on or off, and that the zone valves are on or not. Since I know which days it is supposed to run, I can look at the monitoring log and see that all the irrigation control outputs trip in the right sequence. Or, if I really need more assurance, I can turn on a web cam and see the sprayers running in real time (at 5 AM) and that the lawn is still green. The irrigation system and my electronic oversight has worked fine for 2 years.

I always start the system before packing up to check that all the spray heads are clear. Then I put it back on automatic. This time was no different. As expected, the controller turned the water pump on, sequenced through the four zones, and shut off. But wait, the pump and zone 4 were still on. My voltage sensors said the controller and zone valves were powered off, but the water was still pouring out. “Help!” I thought. The irrigation company was there in less than a half hour. By then I had pulled the breaker on the water pump, but I couldn’t help but wonder about the consequences of this happening a couple days later when I was back in Connecticut.

Well, welcome to the world of plumbing control versus electronic control. You’d think that the irrigation controller would actually control the water pump. Closed loop control? Yeah, right. Guess again. Apparently, the water pump is only controlled by a pressure switch. When the irrigation controller turns on a solenoid valve, the pressure switch turns the pump on to maintain pressure in that zone. Turn the valve off and the pressure goes up to where the pump shuts off again. So, what happens when a little tree root grows into the underground zone valve that causes it to stick open? How about Niagara Falls! Who the heck designed this thing?

Cleaning the area around all the zone valves restored some confidence, but I know I have to close the control loop. I have to keep the present electromechanical controller because workmen access the system in my absence, but I can’t deal with such “open loop” uncertainties. You can bet your life that very soon the water pump will have an HCS-controlled power relay that coordinates the zone actuation and pump runtime. Lawns are bad enough. I don’t need a pond to go with it.

steve.ciarcia@circuitcellar.com

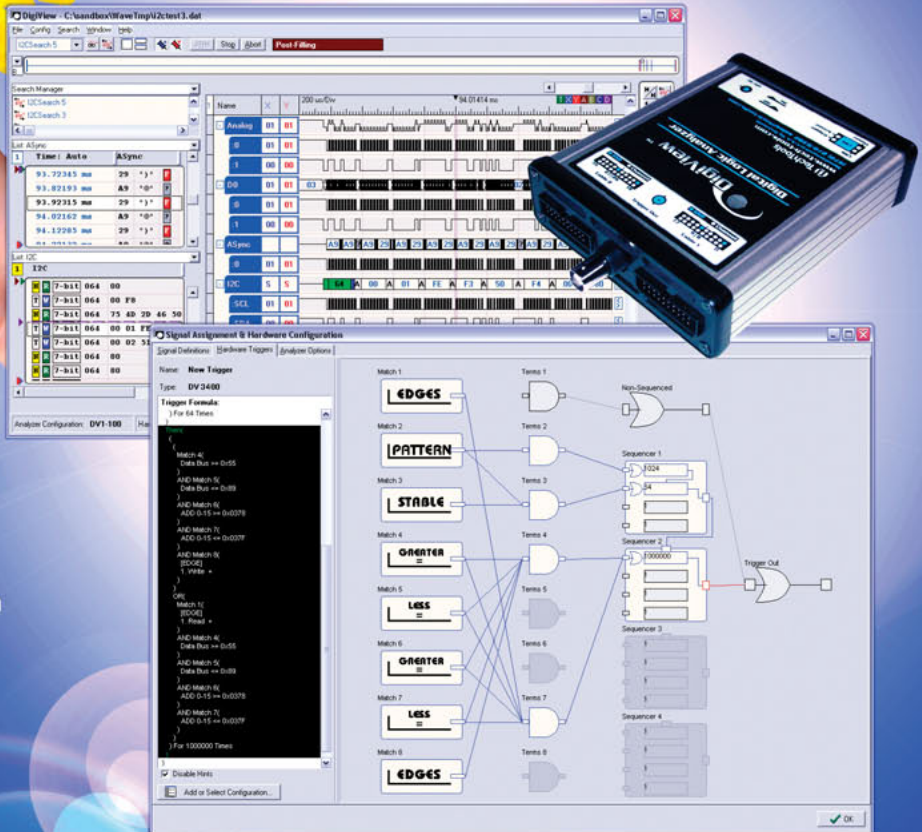
A handwritten signature in black ink, appearing to read 'Steve'.

PC Based Logic Analyzers

NEW Model DV3400

from \$499.00

- Faster Sampling (200/400 Msps)
- More Channels (36)
- More Memory (4x)
- Advanced Match circuits (8)
- ±6V Adjustable Threshold (x2)
- Cascadable Sequencers (4)
- Enhanced Compression



Professional Hardware Capture + Software Analysis

- Automatic Real-time Hardware Compression eliminates the need to reduce resolution. Our newest version makes "dead time" insignificant.
- Edge and Pattern Triggers on all models. Advanced Model also includes Range (>, <, =, !=, >=, <=) and Stable Matches with Duration, and 4 flexible cascadable sequencers with pass-counters.
- Pattern Searches with Match & Duration.
- Specialized Sequential Searches for Serial and State Mode Signals.
- Display I²C, Synchronous (SPI), Asynchronous (RS-232), State, Boolean, Bus and Analog Data.
- Single or Dual Waveform Views.
- Resolution Zoom in Wave Views.
- Time based Link Groups for all views.
- Specialized Exports from Data Tables and List Views.
- Multi-Signal Data Tables.
- Drag & Snap Markers.
- "Click to Center" function.
- "Snap Previous/Next" function.
- Print or Save Images with comments.
- USB 2.0 (480 Mbps).
- USB 1.1 compatible (12 Mbps).

Very flexible, "easy to configure" Advanced triggering.

PICmicro[®] MCU Programmer

Multi-Function, In-Circuit & Gang Operation

only \$199.00



See our Memory Emulators from \$179.00

TechTools
www.tech-tools.com
 (972) 272-9392 • sales@tech-tools.com

THE PARALLAX PERSPECTIVE

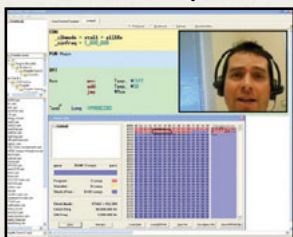
JUNE 2009 - BASIC STAMP, AND SX MICROCONTROLLERS, AND THE PROPELLER CHIP



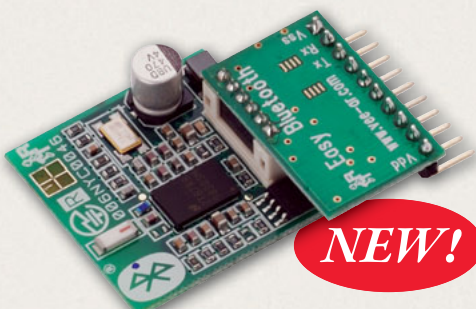
Free Educator's Courses

Parallax is offering free pilot courses at our Rocklin, CA, office for teachers interested in getting started with or furthering their BASIC Stamp knowledge and skills. Visit www.parallax.com/education for details and to request a date that fits your busy schedule. The next course is June 22-23, 2009.

More Webinars on the Way



Having run our first series of successful Propeller Webinars, Parallax is planning more. Attend Webinars from the comfort of your own computer and get your questions answered. Check www.parallax.com/go/webinars for the schedule. Space is limited to maximize attendee experience. Webinar archives are also available online.



Easy Bluetooth Module

The Easy Bluetooth Serial Module (#30085; \$69.99) is an effective and low-cost solution to free your hardware applications from wires. The module is small in size, and with its SIP header design, it can fit on any 0.1" spacing breadboards for rapid prototyping. The module can be powered with various voltages since it has an on-board regulator, allowing it to be used with all Parallax microcontrollers. The RBT-001 module is removable, so various other applications can be designed without the SIP header attached if desired. 1.x & 2.0 Bluetooth Compliant.



CO Gas Sensor Module

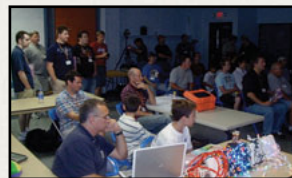
The CO Gas Sensor Module (#27931; \$29.99) is designed to allow a microcontroller to determine when a preset CO gas level has been reached or exceeded. Interfacing with the sensor module is done through a 4-pin SIP header and requires two I/O pins from the host microcontroller. The sensor module is mainly intended to provide a means of comparing carbon monoxide sources and being able to set an alarm limit when the source becomes excessive. Compatible with all Parallax microcontrollers.

Propeller Chips Now Only \$7.99!

Parallax is pleased to announce that we have reduced the price on Propeller chips from \$12.99 each to \$7.99. Our costs on all three chips (P8X32A-D40, P8X32A-Q44, and P8X32A-M44) have reduced and we are passing the savings on to our customers! Be sure to take a look at our high-quantity pricing as well.



Attend an Unofficial Propeller Expo



Propeller Discussion Forums contributor Jeff Ledger (aka "OldBitCollector") has organized two more Unofficial Propeller Expos: UPEW at Parallax headquarters in Rocklin, CA, on June 27-28, 2009, and the UPENE in Norwalk, OH, on August 22, 2009. Get more details and sign up to attend at www.warrantyvoid.us/expo/.

Order online at www.parallax.com or call our Sales Department toll-free at 888-512-1024 (Monday - Friday, 7 a.m. - 5 p.m., PDT).

BASIC Stamp is a registered trademark of Parallax Inc. Propeller, Parallax, and the Parallax logo are trademarks of Parallax Inc. Prices are subject to change without notice.

PARALLAX 
www.parallax.com

Time-Triggered Systems (Part 3)

Tackle Data Acquisition

In their last article, Mike and Lizie described how to add a GUI to a co-operative scheduler and then obtain detailed timing information while developing a multitask audio project. Now they present a revised Theremin system design. They also describe how make a co-operative scheduler recover from various data acquisition issues.

After seeing the initial design for our modernized Theremin musical instrument in the second part of this article series, a member of my fourth-year capstone design team at the University of Calgary said, “Dr. Smith, you have no soul.” The comment was unexpected because I have always seen myself as a warm and considerate professor (in a highly demanding sort of way). However, the student explained that he meant the music generated by the proposed Theremin instrument had no background “beat,” so the music had no rhythm.

“Rather than adjusting the Theremin’s volume by calculating how much light falls on a light sensor, why don’t we build a beat stick for you?” another team member asked. The idea for a beat stick was apparently the generic (home-built)

equivalent of a Wii stick using an accelerometer to sense the movement of the Theremin-ist’s hands. By waving the stick around, you can add a beat (a sudden volume change) to the music.

Currently, a pulse-width modulated (PWM) signal stream from a TAOS light sensor is used to control the volume. It would be figuratively “5 minutes work” to replace this PWM stream with one from an Analog Devices ADXL213 accelerometer. Recalibrate the volume scaling factors calculated in the code, and the Theremin-ist will be up and playing again!

However, the second aspect of the team’s idea was a little more complicated. First, they would write a DSP program to determine the basic beat from the generated Theremin

music. Then a second output audio stream would be generated by changing the frequency of the beats in some music having a good basic beat to be the same as that of the Theremin music. A team member’s parting comment was: “So the Theremin-ist will get automatic accompaniment to the music—all done in real time, of course.”

Figure 1 shows their new proposed design. The hardware was not a problem. A Blackfin BF533 evaluation board (used for teaching at the University of Calgary) is easily capable of handling multiple audio inputs and outputs. The BF533 has considerable DSP capability. The software needed to support the new student project would be, in principle, easy to modify from the original Theremin code (covered in our past articles) already running using

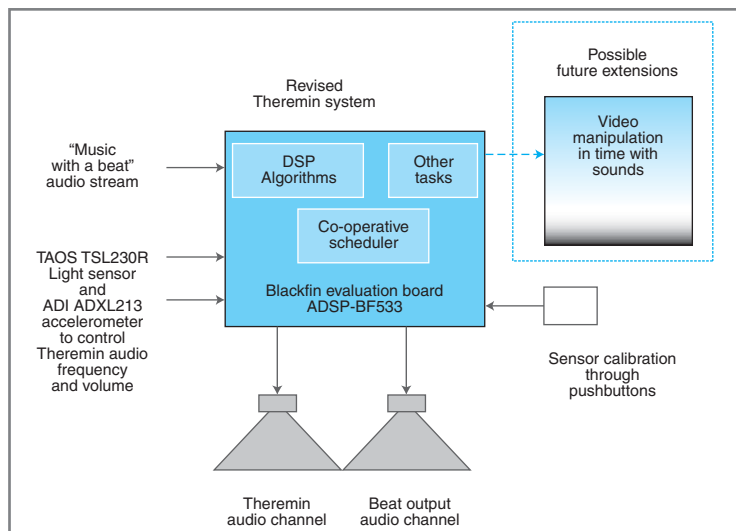


Figure 1—The revised plans are to give “a soul” to the Theremin. An accelerometer adds beats to the Theremin’s musical output. DSP algorithms manipulate the tempo of an input audio stream to provide a “beat” background music accompaniment.

Listing 1—The scheduler code for the revised Theremin musical instrument requires that two existing tasks will need minor modification (ModifiedOutputAudio() and AccelerometerSensorVolume()), and two new tasks need to be added (InputAudioBeat() and DSPTasks()).

```

1  #include "../Theremin.h"
2
3  int main(void) {
4      TTCOS_Init(TICK_TIME_IS_10us);
5      InitHardware( );
6
7      TTCOS_AddTask(ModifiedOutputAudio, NO_DELAY, EVERY_2_TICKS);  /***** MODIFIED *****/
8      TTCOS_AddTask(InputAudioBeat, NO_DELAY, EVERY_2_TICKS);      /***** NEW TASK *****/
9
10     TTCOS_AddTask(LightSensorFrequency, ONE_TICK_DELAY, EVERY_10_TICKS);
11     TTCOS_AddTask(AccelerometerSensorVolume, THREE_TICK_DELAY, EVERY_10_TICKS); /***** MODIFIED *****/
12
13     TTCOS_AddTask(AdjustParameters, FIVE_TICK_DELAY, EVERY_MS);
14     TTCOS_AddTask(DSPTasks, 10, 2 * EVERY_MS);                    /***** NEW TASK *****/
15     TTCOS_AddTask(AdjustLightSensitivity, SEVEN_TICK_DELAY, EVERY_TENTH_SECOND);
16
17     TTCOS_Start();
18
19     while (1) {
20         TTCOS_GoToSleep( ); // Wait, in low power mode, for an interrupt
21                             // The interrupt service routine calls TTCOS_Update( )
22
23         TTCOS_DispatchTasks( ); // Run all the tasks in the system according
24                                 // to whether their delays have expired
25     }
26
27     return 0; // Make compiler happy
28 }

```

an operating system based around a co-operative scheduler (see Listing 1).

Switching from controlling the volume of the Theremin using a PWM pulse stream from a light sensor LightSensorVolume() to a PWM pulse stream from an accelerometer AccelerometerSensorVolume() (Line 11) is essentially a recalibration issue. The old audio task OutputAudio() requires minor modification (Line 7) to handle two output channels rather than one. A second audio task InputAudioBeat() (Line 8) picks up the new input audio stream.

Developing the DSP algorithm DSPTasks() (Line 14) to determine the “beats” present in the two audio streams would be a challenge for a fourth-year university design team. However, given sufficient time, the team could make the system work even with the additional complication of having to smoothly integrate the two audio streams together. However, as the system state history snapshot shown in Photo 1 indicates, there is a serious data acquisition problem to be overcome. The moment DSPTasks() cuts in (top line), the processor becomes fully occupied with this task. No other task can now be executed in the co-operative scheduler, so every other task blocks (shown as orange) and the system dies. How do you fit a long DSP task into the operation of a co-operative scheduler that is running many sampling tasks that need to be serviced at regular 20- μ s intervals?

LENGTHY HARDWARE TASKS

We could easily switch back to a preemptive scheduler to provide a solution. In that case, we’d place the DSP

tasks as part of the loop in main() and let all the audio and measuring tasks performed by the light sensor and accelerometer be fired off in the background as part of a number of interrupt service routines, or ISRs (see Listing 2).

However, Lizzie and I didn’t think constructing this project was about making the “eerie Theremin music” that was so attractive to the fourth-year project team. For us, it had more to do with rediscovering the advantages of designing embedded systems with “highly predictable

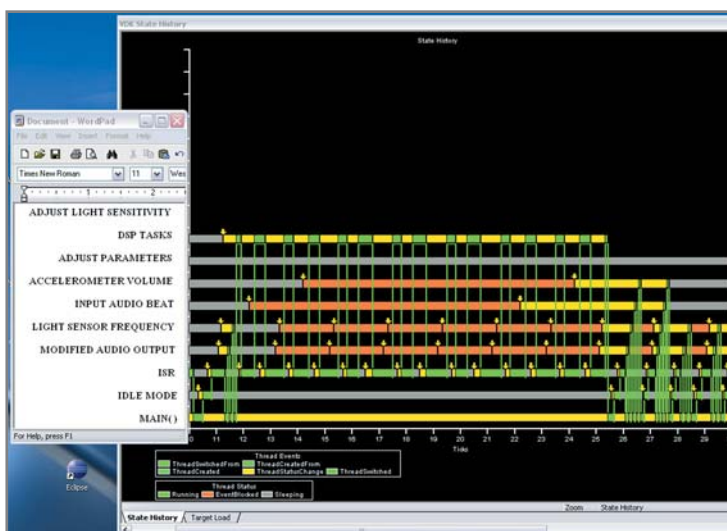


Photo 1—The moment that the DSPTasks() task starts to execute, all the other tasks start to block (shown in orange). When the DSPTask() completes execution, all the blocked tasks start to “play catch-up” and execute with every scheduler tick rather than at their proper times. This would first block and then distort the audio signal.

Listing 2—The code for a pre-emptive scheduler involves a low-priority task running within a while() loop in main() with higher priority tasks running as part of the interrupt service routine.

```
int main(void) {
    SetupInterrupts( );
    ActivateInterrupts( );
    while (1) {
        DSPTask( );
    }
}

EX_INTERRUPT_HANDLER(EverythingISR) {
    AcknowledgeInterrupt( );
    AudioInput( );
    AudioOutput( );
    etc.
}
```

performance” present when using a co-operative scheduler.

In the second article in this series, we described how a similar problem arose when there were a number of scheduler tasks being used to determine the frequency of the output signals from a light sensor. While the processor was waiting for the changes in the sensor signals, these support hardware tasks prevented the time-critical audio output tasks from running. The solution was to break the long running task into a large number of shorter tasks that interacted in a co-operative fashion with other tasks.

LENGTHY SOFTWARE TASKS

The original Theremin code design had tasks “waiting” for an external hardware signal to change its state. This required tasks with durations between 10 μ s to 30 ms depending on the signal’s frequency. Such a hardware-oriented task can be easily broken up into a series of “tiny” tasks (300 ns in duration and a period of 10 μ s) that “poll,” rather than “wait” for, the external signal changes.

Imagine trying the same approach with a lengthy software-oriented task. For the code in Listing 1 to actually work, the capstone project team would have had to write DSPTasks() as a series of code overlays that were swapped in and out of memory between the 10- μ s timer interrupts. That sounds like a really practical, time-efficient solution!

If this rediscovery of using co-operative schedulers in embedded systems is not to die a sudden death, a number of things need to be tackled to make the revised Theremin design work. What code development process will make it possible to simultaneously service a number of highly time-critical “fast” data acquisition tasks without these tasks being blocked by an equally important (but long-running) DSPTasks() routine?

There are now multiple tasks that interact with each other. How do you modify the operation of a co-operative scheduler to ensure that you can recover from one of the tasks failing? Sure, the Theremin does not need to meet safety-critical requirements, but we would like to be able

to continue in a default mode if a cable is temporarily disconnected.

FIND TIME FOR DSP

Executing a “long DSP task” using a co-operative scheduler is obviously a pretty standard thing to do. Rather than reinvent the wheel, we went back to Michael Pont’s *Patterns for Time-Triggered Systems* (2001) to see what he suggests to handle these issues. (The book is available for free at www.tte-systems.com/books.) He recommends two approaches. The first is to avoid frequent software polling. The second is to allow the scheduler to execute a single “preemptive task.”

Consider hardware polling. Currently, the LightSensorFrequency() task examines (polls) the input light sensor signal by monitoring a general-purpose I/O (GPIO) pin. Each signal transition causes the task either to move between one of four states (HIGH1, LOW1, HIGH2, or

Listing 3—Code that “waits” for a hardware event to occur will block other tasks in a co-operative scheduler. The blocking disappears if the code “polls” the hardware to see if an event has occurred, but absolute accuracy is lost.

```
enum {HIGH1, LOW1, HIGH2, LOW2};
void LightSensorTask( ) {
    // Set system state the first time this task runs
    static int currentState = HIGH1;
    int newState = currentState;
    // Determine level of sensor input
    // either HIGH or LOW
    int pinState = GPIOPin( );

    switch (currentState){ // SOFTWARE poll
    // Find leading edge of the pulse
    case HIGH1: // Test for sensor going low
        if (pinState == LOW)
            newState = LOW1;
        break;

    case LOW1: // Test for sensor going high
        if (pinState == HIGH) {
            newState = HIGH2;
            time1 = CurrentTime( );
        }
        break;

    // Find leading edge of the next pulse
    case HIGH2: // Test for sensor going low
        if (pinState == LOW)
            newState = LOW2;
        break;

    case LOW2: // Test for sensor going high
        if (pinState == HIGH) {
            newState = HIGH1;
            sensorPeriod = CurrentTime() - time1;
        }
        break;
    }

    // Update the task state
    currentState = newState;
}
```

Listing 4—Off-loading the measurement of a signal to a timer in “captured pulse period mode” both increases timing accuracy and reduce overall processor load.

```
enum {CAPTURE_NOT_COMPLETE, PERIOD_CAPTURED}

void LightSensorFrequency( ) {

    static int currentState = CAPTURE_NOT_COMPLETE;
    int newState = currentState;

    // Check if timer has captured pulse period
    int timerState = TimerCountDone( );
    switch (currentState) { // HARDWARE poll
        case CAPTURE_NOT_COMPLETE:
            // Measuring completed?
            if (timerState == DONE)
                newState = PERIOD_CAPTURED;
            break;

        case PERIOD_CAPTURED:
            // Read Timer period register
            sensorPeriod = *pPERIOD;
            ResetTimer( );
            newState = CAPTURE_NOT_COMPLETE;
            break;
    }

    // Update the task state
    currentState = newState;
}
```

LOW2) or to return from the task and free up the processor for other tasks to run.

The software polling task needs to be executed at 100- μ s intervals to ensure that the frequency is determined with a high enough accuracy to ensure that the Theremin does not produce sour notes (see Listing 3). As an alternative to performing frequent software polling, we could set up the processor for hardware polling. This approach requires polling the done flag of a general-purpose timer set to monitor the PWM stream via a pulse-width count and capture mode. In this mode, the timer automatically watches for the beginning and end of the light sensor pulse and records the elapsed time in its PERIOD register (see Listing 4).

Switching from software polling to hardware polling using the general-purpose timer offers a number of advantages. First, we have switched from a frequent time-critical task to an infrequent non-time-critical task. The time constraint on the duration of DSPTasks() has shifted from hundreds of microseconds to tens or hundreds of milliseconds. That’s a load of cycles on the DSP-capable Blackfin processor.

The second advantage is that we are getting much higher precision time measurements of the sensor signals, even though we are performing less frequent time-measuring tasks. In this Theremin application, poor timing accuracy is best avoided as it translates into terrible tonal quality of the music!

PERMIT A PREEMPTIVE TASK

A Blackfin ADSP-BF533 processor has three general-purpose timers that can be used in the Theremin project: one for monitoring the accelerometer signal (volume control),

one for monitoring the light sensor signal (frequency control), and an extra one for a future feature. However, switching from software polling to hardware polling does not really solve anything. There are still the frequent (10 μ s) tasks needed to handle the audio input and output samples, and these will still be prevented from running by DSPTasks(). To solve this problem, Pont suggests making a minor modification to the co-operative scheduler’s ISR to allow two types of scheduler tasks.

Standard co-operative tasks (Listing 5, Lines 24 to 29) are run by TTCOS_DispatchTasks(), which is part of the Theremin’s main loop (Line 37). These “standard tasks” can be interrupted by a single “preemptive task” (Line 22) that is set up to run as part of the co-operative scheduler’s ISR (Listing 6, Lines 60 to 68).

The audio manipulation tasks involve moving one or two words to and from the Blackfin’s audio CODECs, so they take very few processor cycles. Thus, moving all these tasks into a single preemptive task should have negligible impact on the overall performance of the system.

Thus, the short PreEmptiveTask() will meet the restrictions on the preemptive task used in a co-operative scheduler—restrictions that are stronger than those placed on the tasks (threads) present in a pre-emptive scheduler. There must be only one preemptive task in order to maintain the predictability of the co-operative scheduler. It is not necessary that the preemptive task be executed every time the timer interrupt occurs. However, it must be guaranteed that the preemptive task will complete in a shorter timer than that of the scheduler’s timer interrupt.

The Theremin preemptive audio manipulation task will probably meet these requirements without any re-coding. However, given the expected small size of these tasks (Listing 5, Lines 1 to 14), attempts to inline this task to further improve system performance is unlikely to cause any significant code bloat that would over tax the embedded system’s memory.

Permitting this single preemptive task has given back to the co-operative scheduler much of the flexibility present in handling tasks with a preemptive scheduler. However, the total predictability of the co-operative scheduler is retained.

RECOVERY FROM SOFT ERRORS

In a typical embedded project, minor data acquisition errors could be easily introduced by a sensor signal disappearing as a cable is temporarily unplugged. In this situation, we want to make a soft recovery. For example, if the volume sensor has been temporarily removed, then we want to set the system to a safe volume. When the signal reappears, the system should automatically resume normal operation. This situation can be handled by introducing software watchdog code that permits a soft system recovery. This is illustrated in Listing 7, where the LightSensorFrequency() task code has been modified to prevent the system from having convulsions. This means the code will not produce unbearable high-pitched sounds if the light sensor becomes unplugged.

The SOFWATCHDOG_SETUP macro (Line 84) establishes a

Listing 5—Combining a preemptive task (Line 24) with hardware polling (Lines 26 and 27) removes the need for the signal manipulation code `DSPTasks()` to meet very strict time constraints.

```
1  #pragma always_inline
2  inline void ModifiedOutputAudio( ) {
3  // Unchanged Task Code;
4  }
5
6  #pragma always_inline
7  inline void InputAudioBeat( ) {
8  // Unchanged Task Code;
9  }
10
11 void PreEmptiveTask(void) {
12     ModifiedOutputAudio( );
13     InputAudioBeat( );
14 }
15
16 int main(void) {
17     int maxNumberTasks = 7;
18
19     TTCOS_Init(TICK_TIME_IS_10us);
20     InitHardware( );
21
22     TTCOS_AddPreEmptiveTask(PreEmptiveTask, NO_DELAY, EVERY_2_TICKS);  /***** NEW TASK *****/
23
24     TTCOS_AddTask(LightSensorFrequency, ONE_TICK_DELAY, EVERY_10_MS);  /***** HARDWARE TASK *****/
25
26
27     TTCOS_AddTask(AccelerometerSensorVolume, THREE_TICK_DELAY, EVERY_10_MS; /***** HARDWARE TASK *****/
28     TTCOS_AddTask(AdjustParameters, FIVE_TICK_DELAY, EVERY_MS);
29     TTCOS_AddTask(DSPTasks, 10, 2 * EVERY_MS);
30     TTCOS_AddTask(AdjustLightSensitivity, SEVEN_TICK_DELAY, EVERY_TENTH_SECOND);
31
32     TTCOS_Start(maxNumberTasks);
33     while (1)
34         TTCOS_GoToSleep( );           // Wait, in low power mode, for an interrupt
35                                       // The interrupt service routine calls TTCOS_Update( )
36
37         TTCOS_DispatchTasks( );      // Run all the tasks in the system according
38     }                                  // to whether their delays have expired
39
40     return 0;                          // Make compiler happy
41 }
```

number of static variables. In this example, the software watchdog timeout condition is set to be a `QUARTER_SECOND`, and the watchdog is directed to record `LIGHTSENSOR_ERR` type errors. Finally, the system is instructed to move on to recovery task state `RECOVERY` (Line 102, which in this case leaves the note unchanged) if the software watchdog ever times out because the task can't be completed.

Upon a watchdog timeout, the error condition is recorded in the global `errorRecorder` variable, the task is directed to move to the `recoveryState`, and the watchdog timer is reset. See [Listing 8](#) for macro details.

In Line 79, the task's state variable is initialized so that the first time the task is run it will immediately call the new `FEED_SOFTWATCHDOG` case statement (Line 87). There the macro `SOFTWATCHDOG_HANDLER` (see [Listing 8](#)) resets the software watchdog and causes a move to the `NOT_READY` state, which was the original `FirstTaskState` for the `LightSensorFrequency` task.

The final state in the original task (Line 99) is now directed to call `FEED_SOFTWATCHDOG` to reset the watchdog timer each time the task has run to completion. This intuitive approach of initializing a software timer and adding a new `FEED_WATCHDOG` case is applicable

for adding a software watchdog timer to any data acquisition task in the system.

RECOVERY FROM HARD ERRORS

A major error might occur if one co-operative scheduler is being driven as a slave by an interrupt signal from a second processor (shared clock scheduling). Suppose the master processor suffers a hard error so that the shared interrupt signals stop arriving. The slave co-operative scheduler will complete its current task and then "hang." In this situation, we must be able to activate a "back-up" interrupt that would allow the slave co-operative scheduler to run tasks and put the system into a safe or shutdown mode.

Listing 6—Many of the issues with using a co-operative scheduler with “long” software tasks vanish if support for a single “short-duration” preemptive task is added to the co-operative scheduler’s interrupt service routine.

```

43 EX_INTERRUPT_HANDLER(TTCOS_Interrupt) { // Only ISR operative in the co-operative scheduler
44   Acknowledge_TTCOSInterrupt( );
45   TTCOS_Update( ); // Update the Run-Me-Now variables for each task
46 }
47
48 void TTCOS_Update(void) { // Part of scheduler interrupt service routine
49   int taskIndex;
50
51   for(taskIndex = 0; taskIndex < NUMBERTASKS; taskIndex++) {
52     if (Tasks[taskIndex].pointerToTask != DOLIST_ENTRY_EMPTY) { // Is there a valid task in the todoList?
53
54       if (Tasks[taskIndex].taskDelay != 0) // If taskDelay is non-zero then task is not ready to be run
55         Tasks[taskIndex].taskDelay--; // simply decrement the remaining delay time
56
57       else { // If the task is READY_TO_RUN
58         Tasks[taskIndex].RunMeNow++; // then increment the RunMeNow flag
59
60         // New code to handle a pre-emptive task
61         if (Tasks[taskIndex].pre_emptiveTask == true) { // Check to see if pre-emptive task
62           (*Tasks[taskIndex].pointerToTask)( ) // If pre-emptive task then run the task
63           Tasks[taskIndex].RunMeNow--; // and decrement the RunMeNow flag
64
65           // Remove 'RUN_ONCE' pre-emptive tasks from todoList
66           if (Tasks[taskIndex].taskPeriod == RUN_ONCE)
67             TTCOS_DeleteTask(taskIndex);
68         }
69
70         if (Tasks[taskIndex].taskPeriod != RUN_ONCE) // Prepare periodic tasks to run again
71           Tasks[taskIndex].taskDelay = Tasks[taskIndex].taskPeriod - 1;
72       }
73     }
74   }
75 }

```

Listing 7—The use of a several global defined macros makes it straightforward to add software watchdog capability to each task.

```

76 enum {CAPTURE_NOT_COMPLETE, PERIOD_CAPTURED, SOFTWATCHDOG_STATES}; //** CHANGED **/
77
78 void LightSensorFrequency(void) {
79   static int currentState = FEED_SOFTWATCHDOG; //** CHANGED **/
80   int newState = currentState;
81   int timerState = TimerCountDone( );
82
83   // Indicate software watch dog timeout, error code, and recovery state
84   SOFTWATCHDOG_SETUP(QUARTER_SECOND, LIGHTSENSOR_ERR, RECOVERY); //*** NEW ***/
85
86   switch (currentState) {
87     case FEED_SOFTWATCHDOG: //**** NEW ***/
88       SOFTWATCHDOG_HANDLER(CAPTURE_NOT_COMPLETE); /* Next state used after feeding complete */
89       break;
90
91     case CAPTURE_NOT_COMPLETE: //** Check if hardware counting complete */
92       if (timerState == DONE)
93         newState = PERIOD_CAPTURED;
94       break;
95
96     case PERIOD_CAPTURED:
97       sensorPeriod = ReadTimerCounter( );
98       ResetTimer( );
99       newState = FEED_SOFTWATCHDOG; //** CHANGED **/
100      break;
101
102     case RECOVERY: //**** NEW ***/
103       DoRecovery( );
104       newState = CAPTURE_NOT_COMPLETE;
105     break;
106   }
107
108   currentState = newState; // Update the task state
109 }

```

Listing 8—These macros provide the functionality to place a software watch timer inside every task to ensure that the lack of an input signal does not cause the system to end up in an unsafe state. The variable `errorRecorder` uses a bit to signify errors associated with each individual task. Errors can be identified and a recovery procedure initiated as part of the preemptive task that is run at every tick of the scheduler.

```

52 #define SOFTWATCHDOG_STATES FEED_SOFTWATCHDOG = 0x70000000, NO_ERROR
53
54 #define SOFTWATCHDOG_SETUP(timeOut,taskErrorBit,recoveryState) \
55     static unsigned int watchDogReset = timeOut;\
56     static unsigned int watchDogTimeOut = 0xFFFFFFFF; /* Avoid immediate time out */\
57     \
58     if (CurrentTime( ) > watchDogTimeOut) {\
59         errorRecorder |= (1 << taskErrorBit);\
60         currentState = recoveryState;\
61         watchDogTimeOut = CurrentTime( ) + watchDogReset;\
62     }
63
64 #define SOFTWATCHDOG_HANDLER(originalFirstTaskState) \
65     currentState = originalFirstTaskState;\
66     watchDogTimeOut = CurrentTime( ) + watchDogReset;
67
68 // Each software watchdog error sets an error bit in the errorRecorder variable
69 enum {LIGHTSENSOR_ERR, ACCELEROMETER_ERR /* .etc... */};

```

This can be handled by activating a hardware watchdog timer and adding a `FeedHardwareWatchTimer()` task to the system's to-do list. Provided the shared interrupt keeps arriving, the co-operative scheduler would be able to run `FeedHardwareWatchTimer()`, which ensures that the watchdog timer interrupt does not occur. If the external signal stops, then the watchdog timer interrupts start occurring and can be used to allow the co-operative scheduler to produce a system-safe condition.

THE NEXT DESIGN PROBLEM

In this article, we have looked at how to allow a co-operative scheduler to recover from hard and soft data acquisition issues—missing interrupts or external sensor signals. We have also shown that using hardware, rather than software, polling, combined with a single pre-emptive task, gives the co-operative scheduler a great deal of flexibility.

The capstone design project team has indicated that they plan to prototype their system by pairing the Theremin music stream with a recording of a snare drum. The repetitive beat in the snare drum music should be identifiable by using a correlation algorithm. They will then use this beat time, and a

sample of the beat note itself, to establish a new music stream whose tempo can be manipulated (see [Figure 2](#)).

It sounds like they will need to use multiple buffering techniques where they gather audio signals into one buffer, process a second, and place the DSP results into a third buffer, while outputting previous calculations from a fourth. That means they will need the Blackfin's direct memory access (DMA) capability to move the large amount of audio data stored in the evaluation board's external memory into the core for processing while not disrupting the basic `DSPTasks()` operation. How all this is handled within a co-operative scheduler environment might be the subject of a future DSP-related article that the team submits. 📧

Mike Smith (mike.smith@ucalgary.ca) has been contributing to Circuit Cellar since the 1980s. He is a professor in computer engineering at the Schulich School of Engineering, University of Calgary, Canada. Mike's main interests are developing new bio-medical engineering algorithms and moving them onto multicore and multiple-processor embedded systems in a systematic and reliable fashion. He recently became a convert to the application of Agile Methodologies in the embedded environment. Mention "test-driven development" and his eyes light up. In 2008, Mike had his Analog Devices University Ambassadorship renewed for the eighth straight year.

Lizie Dunling-Smith is a fourth-year undergraduate student at the University of Alberta, Canada. She will graduate in 2009 with a B.Sc. in engineering physics, specializing in electrical engineering.

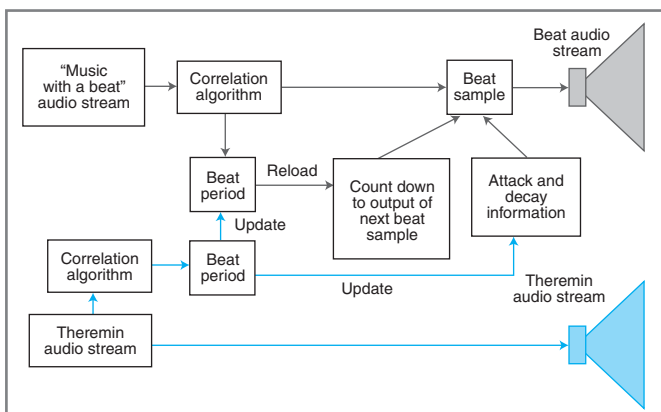


Figure 2—This is the proposed “adaptive drum tempo” DSP algorithm.

SOURCES

ADSP-BF533 Blackfin processor and ADXL213 accelerometer

Analog Devices, Inc. | www.analog.com

TTSL230R Sensors

TAOS, Inc. | www.taosinc.com