#195 October 2006

www.circuitcellar.com

CIRCUIT CELLAR®

THE MAGAZINE FOR COMPUTER APPLICATIONS

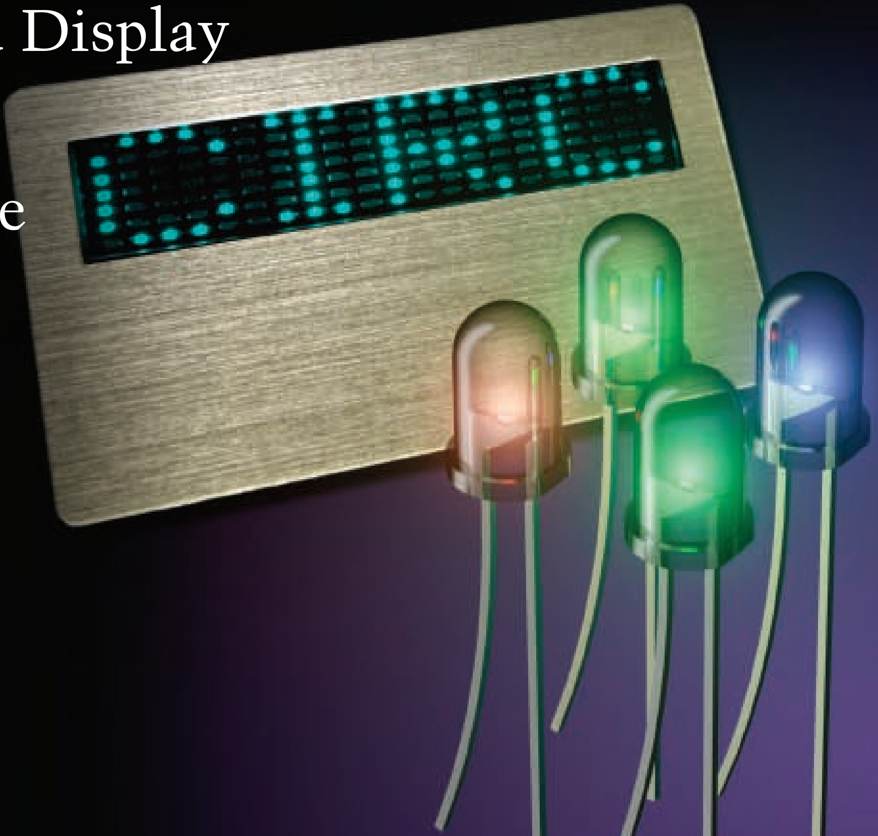
GRAPHICS & VIDEO

Video13 TV Interface

Internet-Connected Display

Video Without
Dedicated Hardware

PoE Integration



$4.95 U.S. ($5.95 Canada)

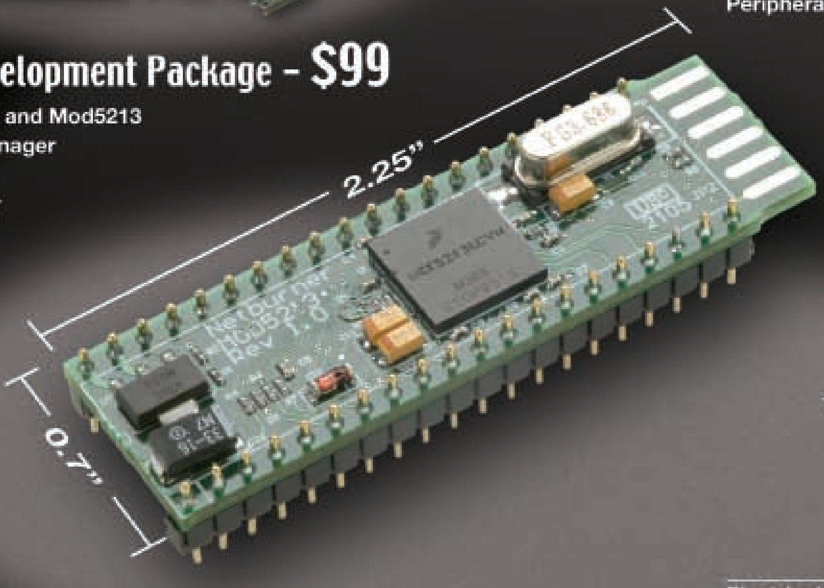
32-bit performance at an 8-bit price

Introducing the NetBurner Mod5213

for embedded applications

Complete Development Package – $99

Development Board and Mod5213
IDE with project manager
C/C++ Compiler
Graphical Debugger
Real-Time OS
Deployment Tools



Mod5213 – $39*

Processor	Freescale 32-bit MCF5213
Speed	66MHz, 62 MIPS
Flash Memory	256 KB
SRAM	32 KB
Size	2.25" x 0.7" 40-pin DIP
Peripherals	Up to 33 GPIO 8-channel 12-bit Analog to Digital **Serial Interfaces** 3 UARTs DMA capable SPI I2C CAN 2.0 **Timers** Four 32-bit timer channels with DMA capability Four 16-bit timer channels with capture/compare/PWM 4-channel 16-bit/8-channel 8-bit PWM generator Two periodic interrupt timers (PITs) **Special Features** Hardware accelerated Multiply, Accumulate and Divide 4-channel DMA controller Watchdog Timer
Electrical Specs	Input Voltage 4 -7 Volt Stop mode Current 130uA
Temperature range	-40°C to +85°C

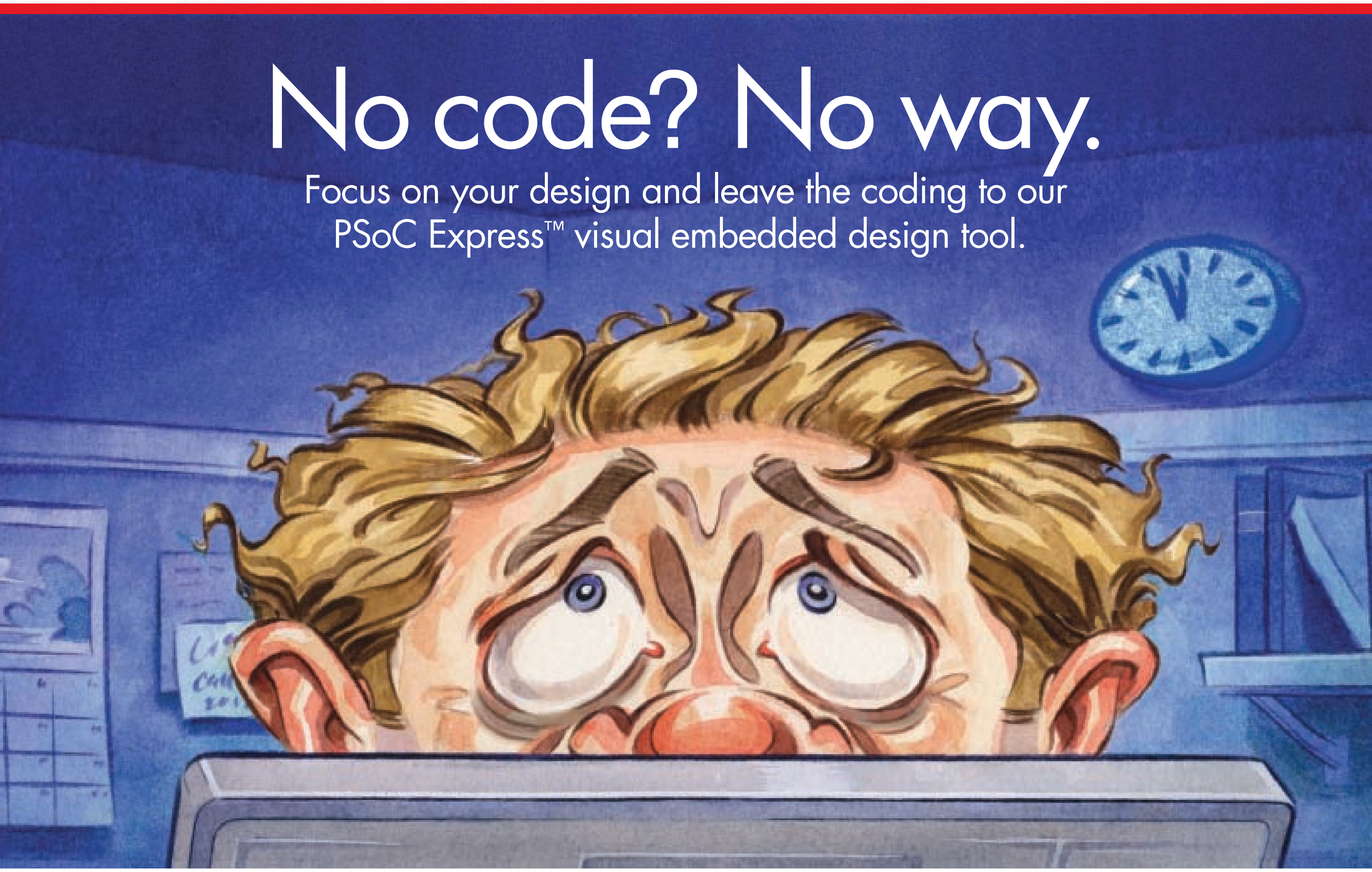
*quantity 1,000

NetBurner
Networking in 1 Day!
www.netburner.com • 1-800-695-6828

freescale
semiconductor

No code? No way.

Focus on your design and leave the coding to our PSoC Express™ visual embedded design tool.



Without writing a single line of assembly or "C" code, generate a complete custom design with PSoC Express. Here's what you get with Version 2.1 of our visual embedded software:

- *Support for our family of Programmable System-on-Chip™ (PSoC®) mixed-signal arrays—powerful, programmable digital and analog blocks with integrated MCU and flash memory*
- *Rich visual environment with simulation enables you to see your design and evaluate its performance instantly*
- *Built-in support for interdevice communication; seamlessly divide design problems into smaller pieces using multiple PSoC devices*
- *Retarget to any PSoC mixed-signal array at any time; design first and select device later*

TRY IT OUT NOW.

Whether you have a minute or an hour to invest, we have a way to get you working with PSoC Express:

- *View our online PSoC Express demo:*
  www.cypress.com/expressdemo
- *Request a FREE PSoC Express evaluation kit:*
  www.cypress.com/expresskit
- *Test drive PSoC Express at a live seminar near you:*
  www.cypress.com/expresstour
- *Download FREE PSoC Express software:*
  www.cypress.com/psocexpress

Make all your embedded designs fast and easy.
www.cypress.com/getexpress

CYPRESS
PERFORM

Cypress, the Cypress logo and PSoC are registered trademarks, and Programmable System-on-Chip and PSoC Express are trademarks of Cypress Semiconductor Corporation.
All other trademarks are properties of their respective owners. ©2006 Cypress Semiconductor Corporation. All rights reserved.

(Check out that Jameco Smile!)

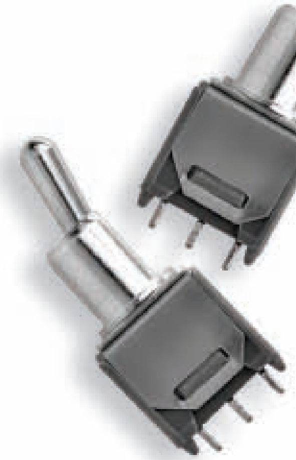
☑ **Big savings.**

10 to 20% off — on all of the industry's name-brand components. Every day. Across the board.

☑ **Or even bigger savings.**

20 to 30% off — on top-quality ValuePro and ReliaPro components. Totally guaranteed by Jameco.

Jameco gives me more choices and smarter ways to save!



Big names at best prices
For parts like memory and microprocessors you'll likely want name brands...

So Jameco gives you all of those at the lowest prices in the industry. *Guaranteed in writing!*

But that's just the start.

Save even more – whenever you want to...
In passives, interconnects, electromechanicals, power products, motors and similar parts, Jameco offers you a unique *additional* way to save...

In addition to name brands you can also choose from *ValuePro* and *ReliaPro* brands – and cut *20 to 30%* off of name-brand pricing!

Actual Jameco Savings vs. Other Major Catalogs

	Name Brands	House Brands
LED, T-1¾, Red, 1200mcd LEDTech LT1873-81-UR-P22	12.5–44.0% Savings	20.8–49.3% Savings
Connector, RF, BNC, 50Ω Amphenol 31-202	17.1–27.9% Savings	31.6–40.5% Savings
Cable/Wire, CAT 5E Belden 1000'	10–20.6% Savings	55–60.3% Savings
Molex Mini-Fit Jr™ R/A Gold Molex 39-30-1241, 24 pos.	14.7–37.3% Savings	22.1–70.7% Savings
Trimmer Pot, ¼" rnd, 1kΩ, .5W Bourns 3329P-1-102	15.6–28.9% Savings	24.6–36.5% Savings

See complete details at www.jameco.com/comparisons.

And all of these parts are Grade-A quality. Fully guaranteed by Jameco.

(Fact is, they're often made by the same manufacturers who supply the name brands. But Jameco buys direct and saves you all the middleman cuts!)

More choices, top savings
Name brands or house brands, you always get more ways to save at Jameco. Call 1-800-831-4242 for a free catalog. Or go direct to the website...

JAMECO® ELECTRONICS

Great Products. Awesome Prices. You'll Be Smiling Too!

So what are you waiting for? See for yourself, right now, at www.Jameco.com/cct

PLC / Embedded controller

CUBLOC

CUBLOC™ (kyoob-lŏk) ***n.*** 1. Easy-to-use Industrial Controller capable of running both BASIC and Ladder Logic. Simply add buttons, sensors, motors, relays, and an LCD to finish project on-time.
2. Could-this-be-true? **µ**-sized **B**asic and **L**adder **O**n **C**hip?

- Use BASIC with Ladder Logic language
- Pin compatible with BS2 (CB220 only)
- 80KB Flash User Program memory
- 2~24KB Data memory
- BASIC Speed : 36000 inst./Sec
- RS232, SPI, I2C, MODBUS supported
- Share Data memory between BASIC & Ladder Logic
- All Development Software Included
- Technical support to jump start your project

MODEL	CB220	CB280	CB290
FLASH	80KB	80KB	80KB
I/O	16	49	91
DATA	3KB	3KB	28KB
PRICE	$34	$43	$69

Consolidate your GUI before it's too LATE!

ALL-IN-ONE Type Touch Panel Controller

CUTOUCH

LCD + KEYPAD + PLC

COMFILE TECHNOLOGY

550 Pilgrim Drive Suite K Foster City, CA 94404
call : 1-888-9CUBLOC (toll free)
1-888-928-2562

* 14-Day Money-Back Guarantee

Log on to
www.comfiletech.com
for your FREE catalog!

TASK MANAGER

Reduce the Risk

The Internet is a venue for exchanging information, goods, and services. As a designer, you search the Internet on a daily basis for the hardware, software, and information you need to bring your projects to completion. In exchange, you often provide companies, organizations, and other designers with payments, contact information, and advice of your own. With each exchange, however, comes risk. Is the blog you're reading about that new line of MCUs truly reliable? Did the designer you're in contact with vet his code for bugs? Is that free software download riddled with glitches or infected with a virus? The upside to taking such risks is that you just might find what you're looking for. The downside is that you might receive the kind of shoddy code or bad advice that could destroy a project.

My brother recently took a similar risk when he bought a refurbished DVD player on a "one deal, one day" type of web site for about $200 less than the unit's original sticker price. Before he bought the unit, he had learned from a blogger that there was a known defect in the unit's firmware that was causing DVDs to skip and occasionally freeze. The blogger had said, however, that the marked-down system was still a good buy for an IT-savvy consumer willing to upgrade the DVD player's firmware on his own. After weighing the pros and cons, my brother bought the system.

A few weekends ago, the worst-case scenario played itself out. We were riding out a rainy Saturday with pizza and flicks when one of the movies began to skip. Rather than huff and puff about the problem, my brother shot into action. Fortunately, it didn't take him long to download the firmware (provided by the manufacturer and posted on the blogger's web site), load it on a CD, and pop it into the DVD player. In a matter of minutes, the system was working perfectly.

My brother was comfortable with the amount of risk he was assuming by purchasing a problematic DVD player and relying on a stranger to supply him with a solution to the problem. But when it comes to an important project on your workbench, just how much of your time and money are you willing to put on the line?

A great thing about *Circuit Cellar* is that we help take the risk out of the equation for designers working on projects that are confined by tight deadlines and budgets. Take the recent Atmel contest. You know that when you read through the winners' entries, you will be dealing with the work of serious engineers who put a lot of time and energy into their designs and code. These entries are great points of reference for future projects.

As for our editorial content, well, you already know that you're getting the best of the best. When you set out to design, say, your own version of Dale Wheat's Video13 system (p. 12), you have at your side a step-by-step guide written with a reader's needs in mind. If you're interested in power over Ethernet technology, you can follow Eddie Insam's advice (p. 60) and then refer to the documents listed at the end of his article for more information. All of the supplementary information is included for your benefit.

Another indispensable resource is the code we post on our web site. If you want to build a color STN display (p. 30), you're free to peruse Dejan Durdenic's code as you map out the flow of your own. You can do the same for many of the other projects covered in this issue.

The message here is to use all of *Circuit Cellar*'s resources as you work on your projects. Be sure to let me know how you do!

cj@circuitcellar.com

C. Abate

CIRCUIT CELLAR®
THE MAGAZINE FOR COMPUTER APPLICATIONS

FOUNDER/EDITORIAL DIRECTOR
Steve Ciarcia

MANAGING EDITOR
C.J. Abate

WEST COAST EDITOR
Tom Cantrell

CONTRIBUTING EDITORS
Jeff Bachiochi
Ingo Cyliax
Fred Eady
George Martin
Ed Nisley

NEW PRODUCTS EDITOR
John Gorsky

PROJECT EDITORS
Steve Bedford
Ken Davidson
David Tweed

CHIEF FINANCIAL OFFICER
Jeannette Ciarcia

MEDIA CONSULTANT
Dan Rodrigues

CUSTOMER SERVICE
Debbie Lavoie

CONTROLLER
Jeff Yanco

ART DIRECTOR
KC Prescott

GRAPHIC DESIGNER
Mary Turek

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.rakoczyphoto.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 subscription orders payable in U.S. funds only via Visa, MasterCard, international postal money order, or check drawn on U.S. bank. **Direct subscription orders and subscription-related questions to Circuit Cellar Subscriptions, P.O. Box 5650, Hanover, NH 03755-5650 or call 800.269.6301.**

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

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

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

Entire contents copyright © 2006 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.

Support **Across The Board.**™

Bringing Products to Life with Mixed-Signal FPGAs.

Ed Baca
Avnet Memec,
Sr. Field Application Engineer

Kyle Dando
Avnet Memec,
Technical Manager,
Western Region

Kevin Thompson
Actel,
SC District Sales Manager

At Avnet Memec, support across the board is much more than a tagline for us. From initial design through end of life — we are deeply committed to driving maximum efficiency throughout the product lifecycle.

The Challenge

How do you launch the first **Mixed-Signal FPGA** to the engineering masses? Actel's Fusion Programmable System Chip integrates configurable analog, large Flash memory blocks, comprehensive clock circuitry, and high performance programmable logic in a single chip. With so much to offer, the company wanted to quickly reach the engineering market. Enter Avnet Memec.

The Solution

Actel's leading distribution partner Avnet Memec underwent rigorous product training on the applications and benefits of Fusion. In the spring of 2006, Avnet Memec rolled out its SpeedWay™ full-day hands-on workshops and labs, highlighting the versatility of Fusion for OEM engineers. All this adds up to quite an education.

As OEMs look to implement Fusion into their designs, Avnet Memec and Actel will be there all the way — providing technical support in every phase of design development. It's true support across the board.

To learn more, request a free Fusion informational CD and enter to win one of **10 free Fusion Starter Kits**, go to:
www.em.avnet.com/actel/satb

Actel® FUS1ON ONECHIP

AVNET® Memec

Enabling success from the center of technology™

1 800 408 8353
www.em.avnet.com

Avnet Green Initiative

Supplier Authorized Distributor

©Avnet, Inc. 2006. All rights reserved. AVNET is a registered trademark of Avnet, Inc.©Actel 2006. All rights reserved. Actel logo and combinations thereof, and others are the trademarks of Actel Corporation or its subsidiaries.

October 2006: Graphics & Video

FEATURES

12 **Video13**
Build a Simple TV Interface
Dale Wheat

24 **Generate Video from Software**
René Geraets

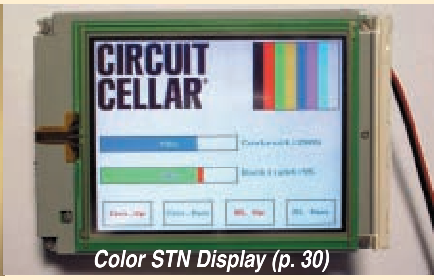
30 **Drive a Color STN Display**
Dejan Durdenic

40 **LED Message Display**
Peter Gibbs & Ramon Sargeant

52 **Power Generator for Portable Applications**
Dhananjay Gadre

60 **Power Over Ethernet Solutions**
Eddie Insam

Building the Video13 (p. 12)



Color STN Display (p. 30)


Simple Power Generator (p. 52)

COLUMNS

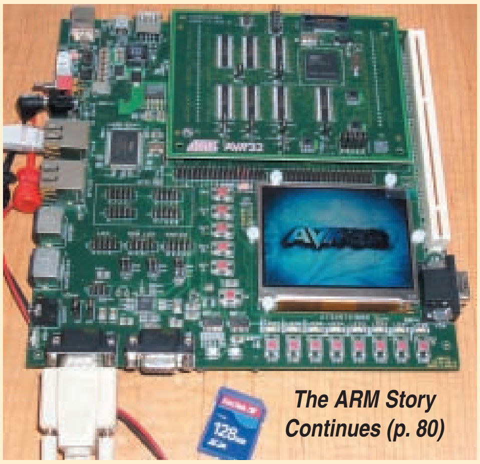
56 **ABOVE THE GROUND PLANE**
DIY Diodes
Ed Nisley

68 **APPLIED PCs**
Advanced USB Protocol Sniffer
Fred Eady

74 **FROM THE BENCH**
Accessing the USB Framework
A Quick Transition from UART to USB
Jeff Bachiochi

80 **SILICON UPDATE**
ARM Twister
Tom Cantrell

USB Protocol Sniffer (p. 68)


The ARM Story Continues (p. 80)

DEPARTMENTS

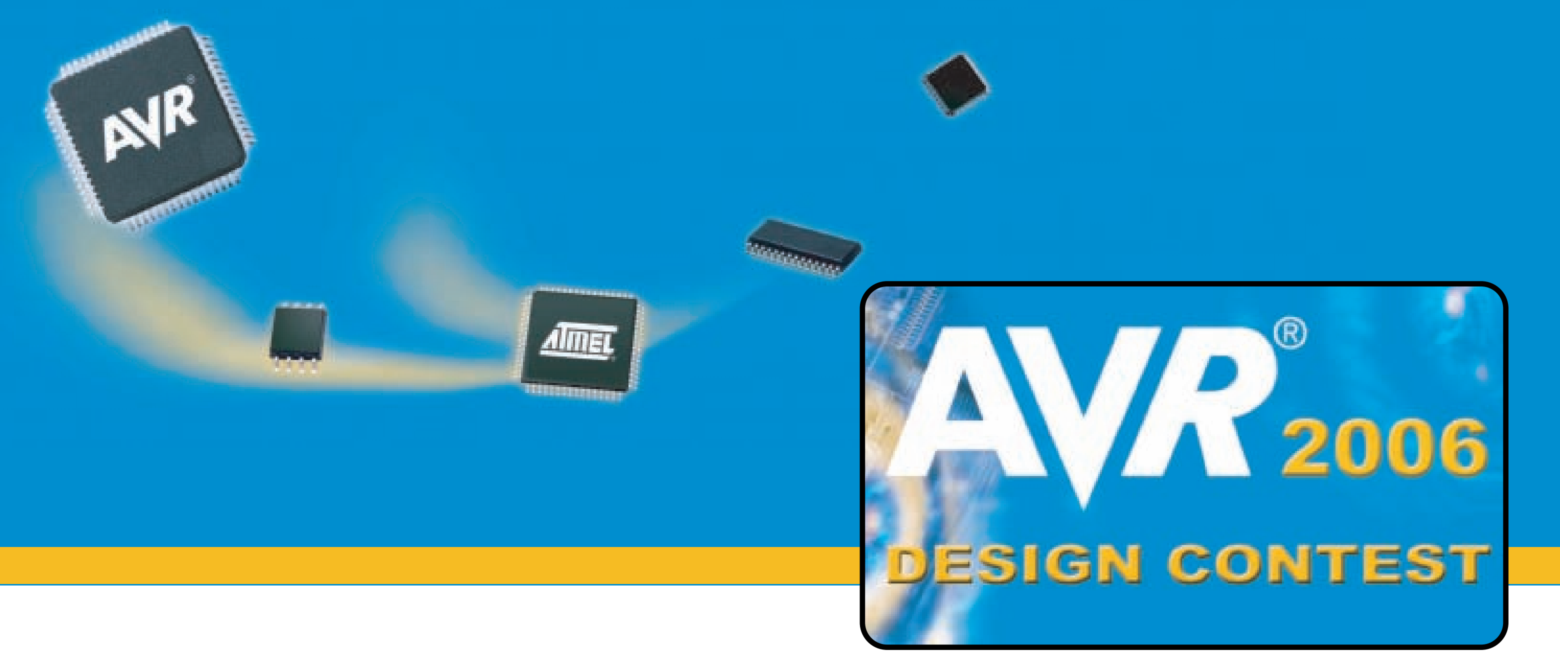
4 **TASK MANAGER**
Reduce the Risk
C.J. Abate

8 **NEW PRODUCT NEWS**
edited by *John Gorsky*

93 **CROSSWORD**

94 **INDEX OF ADVERTISERS**
November Preview

96 **PRIORITY INTERRUPT**
In Your Opinion
Steve Ciarcia



The Atmel AVR Design Contest 2006
is your vehicle to fortune and fame!

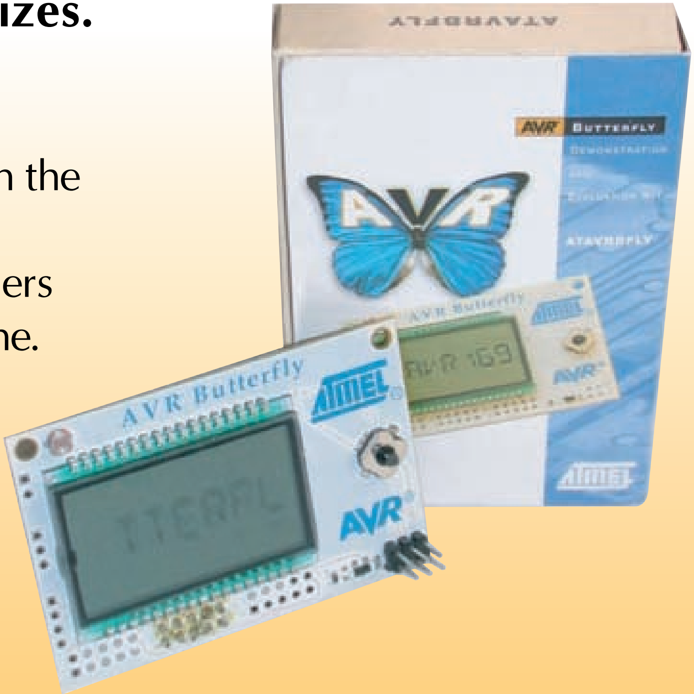
$15,000 in cash prizes.

Thank you to everyone who participated in the AVR design contest. The judges are now reviewing the contest entries and the winners will be announced in Circuit Cellar magazine.

For more information about this contest, visit www.circuitcellar.com/avr2006

CIRCUIT CELLAR®
www.circuitcellar.com

ATMEL®
www.atmel.com



NEW PRODUCT NEWS

Edited by John Gorsky

128 × 32 LED GRAPHICS DISPLAY WITH DRIVE ELECTRONICS

The industry's first 128 × 32 LED graphics display with drive electronics and a 5-V HC CMOS-level video interface is now available. Designed as a drop-in compatible replacement for the popular APD-128G032 plasma display module, the **LED-128G032** is designed to offer high brightness and superior viewing characteristics in a slim package.

The LED-128G032 LED has four times the life of a plasma display, which reduces maintenance and replacement costs. With a 12.75″ × 3.15″ viewing area, the LED display is optimized for low- to medium-level information content. It is suited for applications such as arcade games, process control, POS terminals, medical equipment, message centers, and automatic teller machines.

Featuring a very low 17.78-mm profile, the LED-128G032 LED display emits a brilliant orange color. The display has large 16.51 mm × 11.43 mm characters for long-distance readability, offers a high contrast ratio of greater than 30:1, and provides a wide viewing angle of greater than 150°.

The display costs **$259** in OEM volume quantities.

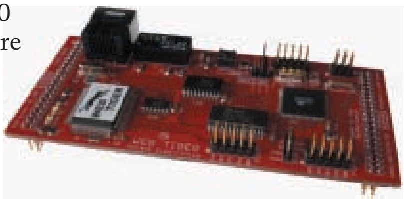
Vishay Intertechnology, Inc.
www.vishay.com

ETHERNET EXPANSION FOR THE ATMEL STK500

The **Web Tiger** is a module designed to fit on top of the Atmel STK500 development board. With this board, the STK500 is extended with an ATmega128 and an Ethernet interface. A Realtek Semiconductor RTL8019 acts as the Ethernet controller on the board.

The board also features 32 KB of RAM and a 32-kHz crystal for RTC applications. The Web Tiger can be programmed through its ISP header or via the parallel high-voltage method. This compact board is only 56 mm × 120 mm × 25 mm and weighs in at 45 grams. Because of the on-board Atmel ATmega128, the board can also be used as a stand-alone controller. An evaluation board that allows you to use the Web Tiger without an STK500 is also available.

The Web Tiger costs approximately **$88.** The evaluation board is approximately **$63** (including VAT). LCDs and programming tools that can be used with the STK500 and Web Tiger are also available.



Achatz Electronics
www.achatz.nl

Logic Supply offers a full line of Mini-ITX mainboards and systems for embedded applications.

VIA Authorized Distributor

Mini-ITX mainboards

with VIA, Pentium, or Pentium M processors
- x86 compatability
- full-featured SBC
- off-the-shelf availability

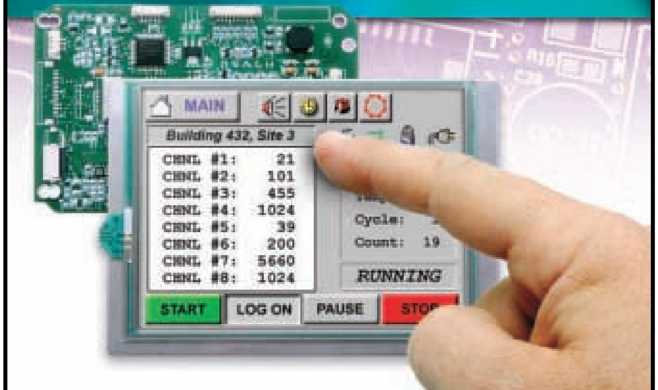
fanless Mini-ITX systems

Utilizing heat pipe technology, these completely fanless Mini-ITX systems offer the perfect solution for applied computing and embedded environments.

LEADERS IN MINI-ITX SOLUTIONS

LOGIC SUPPLY
LOGICSUPPLY.COM
info@logicsupply.com
802 244 8302

Add a color touch interface to your embedded product!



The Reach SLCD controller gives embedded systems a modern graphic display interface. No costly and complex Operating System is needed – just a serial port on any microcontroller. High level commands draw text, define buttons, and display bitmap graphics stored in flash memory. Use a PC to rapidly prototype and validate a human interface.

Choose from monochrome, color passive, or TFT panels from Hitachi, Kyocera, NEC, and others. Complete evaluation kit including 5.7" QVGA color touch panel for only $349.

For more information, call Reach today at **(503) 675-6464**, or see us on the web at **www.reachtech.com**

REACH TECHNOLOGY INC.

NKK®
SWITCHES

DESIGN FLEXIBILITY

GIVE YOUR DESIGNS THE GREEN LIGHT.

Design flexibility has never been more simple. NKK's SmartSwitch™ allows you to design what you want, any way you want. Our series of LCD programmable switches and displays are developed for use with a wide variety of applications. You've got the green light to explore design freedom.

IS Dev Kit-2

SMARTSWITCH™
ACTIVATIONS MADE SIMPLE

http://cc.nkksmartswitch.com

Power your design with NKK.
Visit us online to:

- **ORDER** Dev Kits to drive your creativity
- **DOWNLOAD** software to expedite your design process
- **PROTOTYPE** and test new design ideas cost effectively

NEW PRODUCT NEWS

ARM7X DEVELOPMENT BOARD

The **ARM7X development board** is designed for prototyping, laboratory, and OEM use. The board hosts an Atmel AT91SAM7X256 processor, running at 48 MHz, along with RS-232, RS-485, USB, CAN, and 10/100 Ethernet ports. It also has an on-board RTC, and a 16-pin Optrex compatible LCD interface.

Included with the board is a version of FreeRTOS that has been modified to support the I/O, PHY, and RTC, as well as other aspects of this ARM7X development board that will get your application up and running quickly. A 20-pin, polarized, JTAG connection is provided for in-system programming and debugging. The ARM7X development board can also be programmed through the RS-232 port using the ARMBL-PC bootloader application.

The on-board SD/MMC memory module connector can provide tremendous storage and can be accessed with MS-DOS compatibility using the FlashFile-ARM-SD source code package. On-board switch-mode regulation allows for power inputs from 8 to 38 VDC with an LED power indicator. Termination is provided for 5-VDC output with up to 2 A of combined load. There are even four connections for standard R/C servos for those RCoIP projects!

The development board costs **$199**. Quantity discounts are available.

Priio
www.priio.com

HIGH RETENTION USB INTERFACE

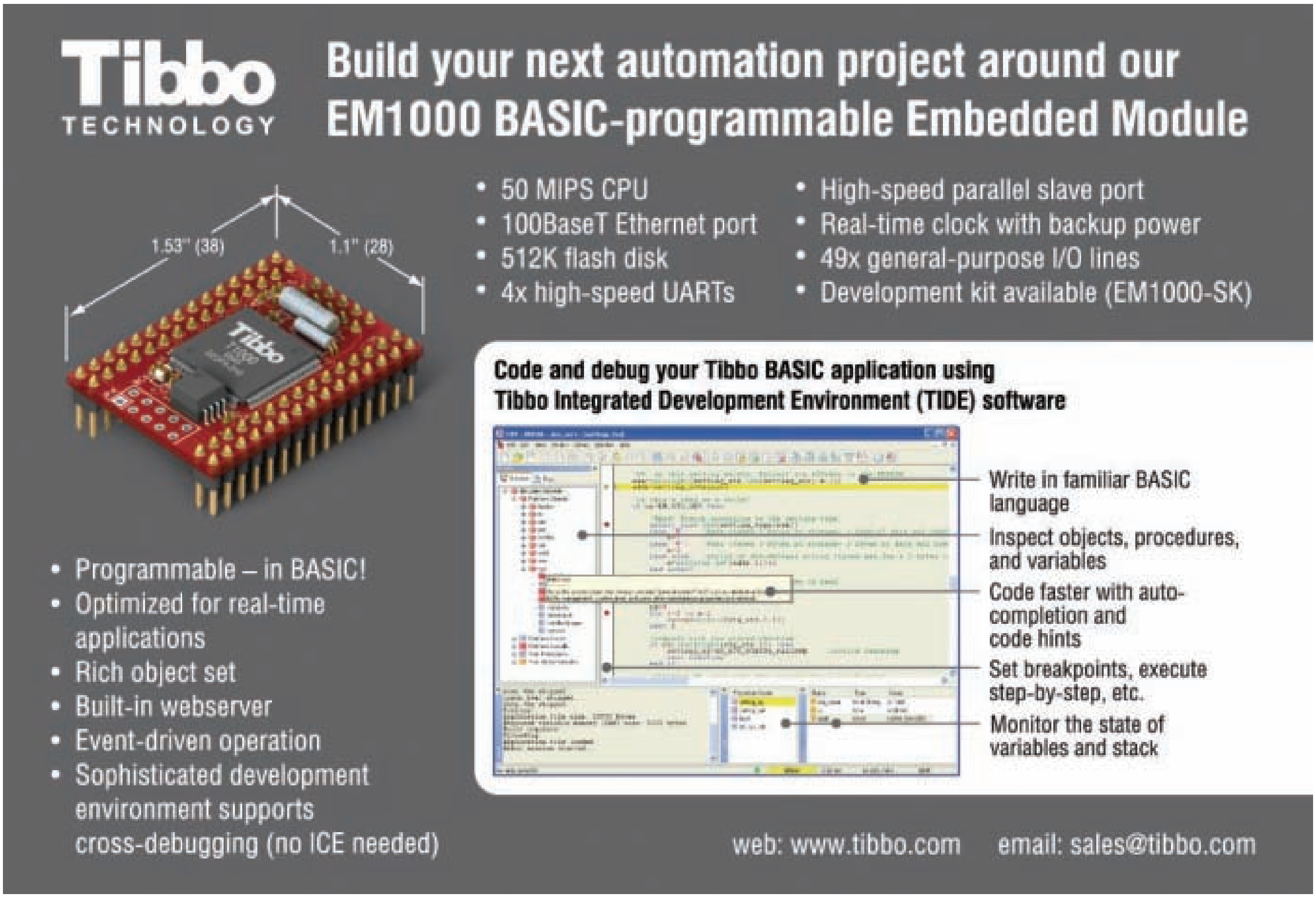
Samtec's high retention Type B USB interface (**USBR series**) complies with the class 1, Div II minimum withdrawal requirement of 15 N. This connector has an orange color-coded insulator to differentiate it from standard withdrawal force connectors (USB series) that are available with white or black plastic.

These USB connectors are available in surface-mount and through-hole designs with a choice of standard right angle or vertical top-entry orientation. In addition to B Type connectors, a full line of A Type and mini-USB interfaces and standard (USBC series) and custom cable assemblies are available.

Samtec's full line of I/O interfaces includes D-Subs, Mod Jacks, Firewire and USB, and Mini-DINs. In addition, they offer a large variety of panel-to-panel systems for high-speed and rugged I/O applications.

Pricing for the USBR series is approximately **$0.35** in production quantities.

Samtec, Inc.
www.samtec.com



Tibbo TECHNOLOGY

Build your next automation project around our EM1000 BASIC-programmable Embedded Module

- 50 MIPS CPU
- 100BaseT Ethernet port
- 512K flash disk
- 4x high-speed UARTs
- High-speed parallel slave port
- Real-time clock with backup power
- 49x general-purpose I/O lines
- Development kit available (EM1000-SK)

- Programmable – in BASIC!
- Optimized for real-time applications
- Rich object set
- Built-in webserver
- Event-driven operation
- Sophisticated development environment supports cross-debugging (no ICE needed)

Code and debug your Tibbo BASIC application using Tibbo Integrated Development Environment (TIDE) software

web: www.tibbo.com email: sales@tibbo.com

NEW PRODUCT NEWS

Visit www.circuitcellar.com/npn
for more New Product News.

SECURE BATTERY-BACKUP CONTROLLER

The **DS3600** is the industry's first comprehensive secure battery-backup controller for data protection in point-of-sale terminals. The DS3600 provides active tamper detection and rapid-erasure key memory. It also supports FIPS-140 security levels 3 and 4. The DS3600 is packaged in a CSBGA, a preferred choice for certification because no pins are exposed to the outside world. As a result, the package is resistant to tampering.

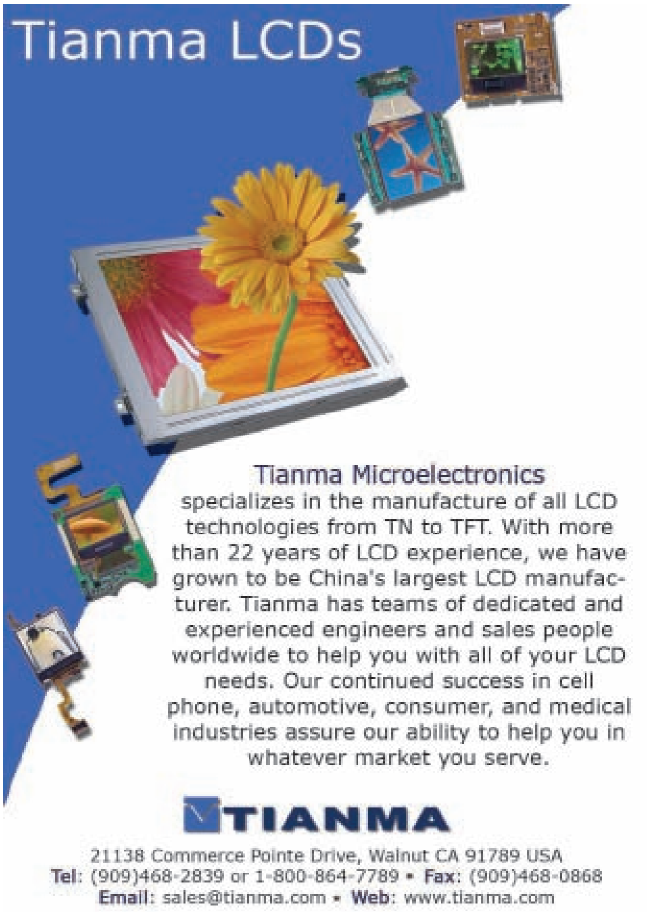
The DS3600's proprietary, on-chip nonvolatile SRAM is used for the storage of encryption keys. This memory architecture constantly complements the SRAM cells to eliminate the possibility of memory imprinting due to oxide stresses. As a result, this technology prevents the passive detection of data remnants in stressed memory cells.

When the DS3600 generates a tamper alarm, the entire 64-byte array is cleared within 100 ns. The erasure of data is this fast because the memory's high-speed direct hardwired clearing function and on-chip power source ensure active erasure.

The DS3600 has tamper-detection inputs to interface with system voltages, resistive meshes, external sensors, and digital interlocks. Using a low-power operation, it also monitors those components continuously. The device monitors the integrated real-time clock (RTC) crystal oscillator and will invoke a tamper response if the oscillator frequency falls outside the set threshold. The internal digital temperature sensor has a programmable rate-of-change detector that protects the DS3600's encryption key memory from thermal attacks. The DS3600 constantly monitors primary power. In the event of a power failure, an external battery power source is automatically activated to keep the SRAM, RTC, and tamper detection circuitry alive.

The DS3600 costs **$3.75** in 1,000-piece quantities.

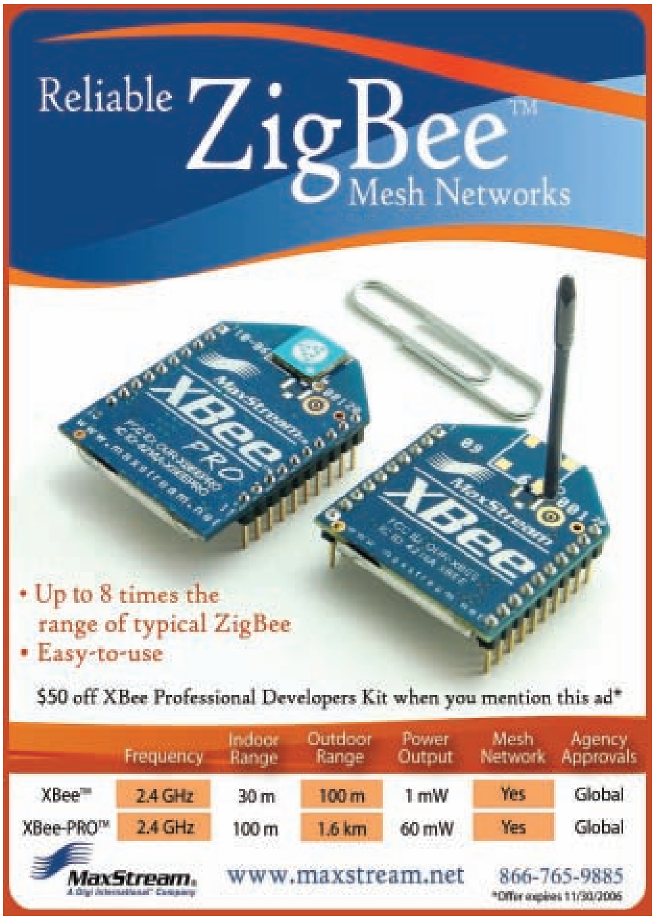
Dallas Semiconductor
www.maxim-ic.com



Tianma Microelectronics
specializes in the manufacture of all LCD technologies from TN to TFT. With more than 22 years of LCD experience, we have grown to be China's largest LCD manufacturer. Tianma has teams of dedicated and experienced engineers and sales people worldwide to help you with all of your LCD needs. Our continued success in cell phone, automotive, consumer, and medical industries assure our ability to help you in whatever market you serve.

TIANMA

21138 Commerce Pointe Drive, Walnut CA 91789 USA
Tel: (909)468-2839 or 1-800-864-7789 • **Fax**: (909)468-0868
Email: sales@tianma.com • **Web**: www.tianma.com



- Up to 8 times the range of typical ZigBee
- Easy-to-use

$50 off XBee Professional Developers Kit when you mention this ad*

	Frequency	Indoor Range	Outdoor Range	Power Output	Mesh Network	Agency Approvals
XBee™	2.4 GHz	30 m	100 m	1 mW	Yes	Global
XBee-PRO™	2.4 GHz	100 m	1.6 km	60 mW	Yes	Global

MaxStream. A Digi International™ Company www.maxstream.net 866-765-9885
*Offer expires 11/30/2006

by Dale Wheat

Video13

Build a Simple TV Interface

Need a simple TV interface? Dale's inexpensive circuit makes a great six-line by 10-character monochrome ASCII display.

In 1976, I saw an item for sale in Southwest Technical Products Corporation's catalog: the GT-6144. It was described as a graphic video adapter that would work with any video monitor or "slightly modified" television set. The GT-6144 offered a resolution of 64 horizontal pixels by 96 vertical pixels, all for the price of $98.50. What really caught my attention, to be perfectly honest, was the crude, bit-mapped image of the Starship Enterprise in orbit over the Earth. Atari's popular PONG video game had invaded our homes during the Christmas holiday season of 1975, so I decided that whatever they could do, I should be able to do better.

Thirty years later, this is not yet the case. I have, however, had some limited success in building a simple video interface that can be easily adapted to other uses, and I'm happy to share what I've learned with you (see Photo 1).

IT'S BEEN DONE BEFORE

The surface of the Earth cooled, oceans formed, and by 1980 most PCs came with built-in video interfaces. What began as dozens of integrated circuits (ICs) on a dedicated PCB (or boards) soon shrank to a mere handful of chips taking up only a corner of a computer. Reading Don Lancaster's *Cheap Video Cookbook* really got me excited about the possibility of building my very own video interface. Mere decades later, I met with limited success.

I read with fascination about Alberto Ricci Bitti's award-winning Video DVM circuit (http://www.riccibitti.com/dvm.htm), which used "direct video synthesis" to create a video signal from an inexpensive Atmel AT90S1200 microcontroller, using only a few discrete components. This was amazing! I had no idea that a humble microcontroller could produce such a high-bandwidth signal.

Next, while browsing around the AVR-freaks web site, I stumbled across Bruce Land's Designing with Microcontrollers (ECE476) course at Cornell University (http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/). His students were given some basic video generation information, and they came up with dozens of incredible and imaginative applications based on an Atmel ATmega32. Not only did I now know that it was possible, I had also been handed multiple, complete design documents, including schematics, source code listings, and photographs. My humble contribution to the extant body of work is to illustrate a minimalist implementation using what I consider to be the fewest parts.

Photo 1—*The Video13 TV interface adds old-school charm to any project. What better way to get your message across than with chunky, monochrome, uppercase text?*

VIDEO SIGNAL

The black-and-white NTSC video signal is defined by the Electronics Industries Association's (EIA) RS-170A standard. The standard specifies the video signal to be an analog line with a characteristic impedance of 75 Ω and various other timing specifications. The voltage range is 0 to 1 V, with 0 V volts indicating a synchronization level, approximately 0.3 V being black, and 1 V being white. This analog voltage can be generated with two of the microcontroller's digital outputs (PB0/VIDEO DATA and PB1/VIDEO SYNC) and combining them with a simple DAC built out of three discrete resistors: R1, R2 and R3 (see Figure 1 and Table 1).

R1 (91 Ω), in combination with the other resistors, helps to set the characteristic impedance of the line. R2 (1 kΩ) sets the appropriate voltage level for the video sync portion of the signal (~0.3 V). R3 (470 Ω) allows the video data portion of the signal to vary between ~0.3 V (black) and 1.0 V (white).

Now, for the purposes of this exercise, I'll omit a lot of the fine detail of the complete video signal specification. In particular, I'm bypassing a bit of work by skipping the color component of the signal entirely. Luckily, the present color composite video signal was designed to be backwards compatible with the previously popular black-and-white system. This allows you to generate a fair-

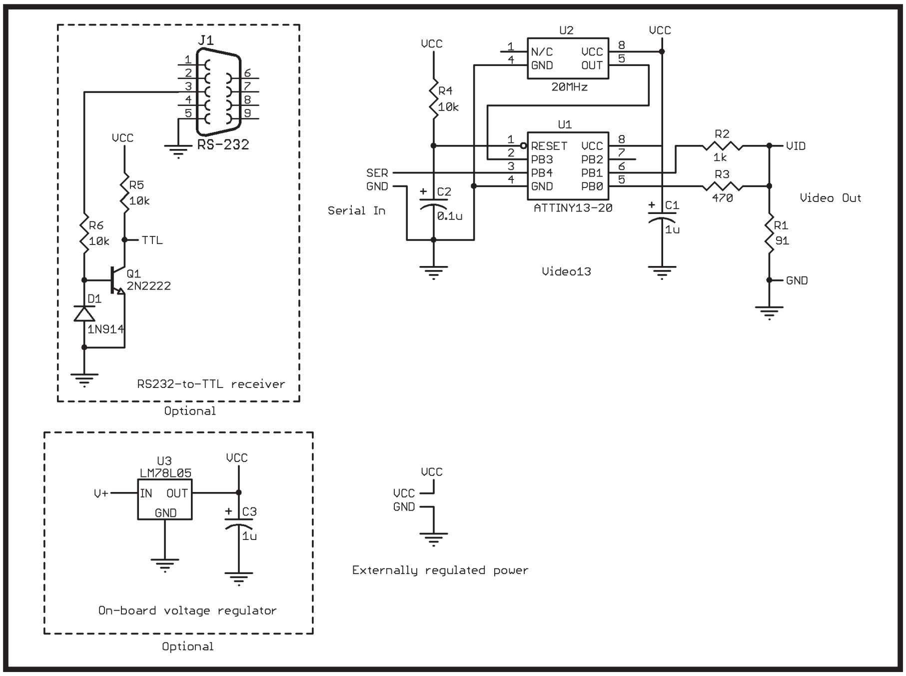


Figure 1—*Check out the entire Video13 TV interface, including all of the optional circuitry. It's very simple to build and has no critical layout requirements. The magic lies in the programming of the versatile Atmel ATtiny13 microcontroller (U1), which is also detailed in this article, and available for download.*

ly simple analog signal that will produce an acceptable image on any commercially available television or monitor that has a composite video input.

Another simplification is to generate a noninterlaced image instead of the more standard interlaced signal. This seems to work well, although the individual scan lines are somewhat visible instead of blurring and overlapping into each other as would normally occur.

ATtiny13

The heart of the circuit is the Atmel ATtiny13 microcontroller. Although it isn't the most humble of its lineage (that would be the ATtiny11, or perhaps the now obsolete AT90S1200), it's close. All of the specifics are described in the datasheet. A few characteristics are pertinent for this discussion. It comes in an eight-pin DIP package and operates at 5 V at 20 MHz. With high-performance RISC architecture, it features 1,024 bytes of reprogrammable flash memory, 64 bytes of RAM, 64 bytes of EEPROM, six I/O lines, and two PWM channels with interrupt capabilities. The microcontroller comes with free software development tools and simple programming hardware.

UNUSED FEATURES

This application does not use all of the ATtiny13's features, but I seriously doubt that any application ever will. The four-channel ADC, analog comparator, external interrupt, and watchdog timer are not used at all. Most of the 32 general-purpose registers and one of the two available PWM outputs are used.

The most disappointing unused feature was the on-chip calibrated oscillator. I had originally wanted to use this handy, built-in circuit, but the resulting signal output was unacceptable. The exact frequency of the clock was not critical, but the jitter it produced caused random noise and visible synchronization errors in the output signal. So, while the internal clock was good enough on average, the timing between individual clock pulses varied so much as to produce very visible and quite unacceptable timing anomalies and glitches on the screen.

I spent many hours tracking down the problem. This was the type of problem that required me to start questioning basic assumptions. I like my assumptions. I don't really like these kinds of problems.

Another common assumption is that the manufacturer's datasheet is correct. Who would know better than the manufacturer? Mistakes happen. More specifically, miscommunications, misunderstandings, and omissions happen. This article represents an intellectual effort that is at best 1% of 1% of the ATtiny13's design effort and probably not even that. Yet, even with that 10,000-fold statistical advantage, I would never presume that there aren't any errors or omissions in this little story. You have been warned.

While trying to pare down the firmware to the absolute minimum, I began looking for "optional" items to omit. The very first code I want a microcontroller to execute is usually to set up the stack pointer if there is one (e.g., the ATtiny11 has no SRAM and therefore no SRAM stack pointer, as such). Failure to properly set up the stack pointer is a common omission that's characterized by code that works until a subroutine is called or an interrupt is encountered. Both functions require a stack. According to the ATtiny13 datasheet, "The stack pointer points to the data SRAM stack area where the subroutine and interrupt stacks are located. The stack space in the data SRAM is automaticall [sic] defined to the last address in SRAM during power on reset."[1]

The diagram of the stack pointer register (SPL) also has the following notation:

Initial Value 1 0 0 1 1 1 1 1

This led me to believe that the stack pointer was automatically set when the device emerged from a reset. This would allow me to omit the ritual initialization, thus saving two instructions or 4 bytes of program space. This is simply not the case. In itself, it is not really that critical an issue, but it serves to illustrate my point. As James Bond's nemesis Goldfinger pointed out, it's foolish to be caught in a lie that can be revealed with a single phone call.

Data (PB0)	Sync (PB1)	Resulting voltage and function
0	0	0.0 V: Vertical and horizontal sync
0	1	0.3 V: Black level
1	1	1.0 V: White level

Table 1—*Combinations of digital outputs PB0/VIDEO DATA and PB1/VIDEO SYNC produce the required analog voltages at the output of the trivial DAC. Because the Video13 doesn't deal in shades of gray, only the sync, black, and white voltages are needed.*

Testing the theory that the stack pointer wasn't initialized took about 5 min. with the simulator that came with AVR Studio (as well as being consistent with empirical evidence). Perhaps a future generation of ATtiny13s will live up to its designers' aspirations.

An odd omission from the ATtiny13 feature set is a quartz crystal oscillator, which is present on almost all of the other AVR devices. This forced the addition of the external oscillator to the circuit. In theory, you can save money by using a different AVR that offers a built-in oscillator circuit that can facilitate the use of a simple quartz crystal instead of a complete oscillator circuit.

LEVERAGE APPLICABLE FEATURES

By far, the most effective use of the ATtiny13's built-in features for this application is to employ the versatile PWM peripheral subsystem. One of the two available PWM channels generates both the synchronization pulses of the output signal and the internal timing signal used to sequence the display logic.

At first glance, the video synchronization signal looks complex. It's composed of both horizontal and vertical sync pulses occurring at different frequencies and periods. The horizontal sync pulse is defined to be 4.7 µs long, occurring at the horizontal refresh rate of approximately 15,750 Hz, or nearly every 63.5 µs. The vertical sync pulse is defined to be about 180 µs in duration, repeating 60 times per second. Now add in the fact that this signal is intended to be broadcast over the air. Direct current (DC) levels don't travel well in that medium, and you have a need to add serration and equalization pulses to the vertical sync signal. Yikes, that looks complex, indeed! Note that the specifications for the original black-and-white video signal vary somewhat from today's color video signal. I trust true videophiles have noted these discrepancies already.

Would you believe that the PWM circuit in the ATtiny13 could generate either of the horizontal or vertical sync pulses nearly perfectly? Not only is this possible, switching between the two sync modes (after proper initialization) requires only the toggling of a single bit in the PWM control registers. I'll explain this in detail, but refer to the ATtiny13 datasheet for more information. A dollar buys a lot of functionality these days.

The PWM hardware supports two channels of 8-bit PWM. This is really a single 8-bit counter with two 8-bit compare registers (not two completely separate timers). Either channel may be associated with an output pin, allowing the PWM subsystem to set, clear, or toggle the associated pin in response to a set of conditions. Both channels share a PWM mode that controls their behavior.

I use a version of the Fast PWM mode (mode 7, which allows the PWM period to be set to an arbitrary value). This allows me to set both the period (and therefore the frequency) of the signal as well as the duty cycle. This is initially set up to produce the horizontal sync pulse of about 4.7 µs every 63.5 µs. The formulas for deriving the required initialization constants are given in the source code posted on the *Circuit Cellar* FTP site.

PUT THE POWER OF THE WEB IN YOUR PRODUCTS.

Imagine providing the ability to access, control, even diagnose and repair your products from virtually anywhere... at any time over a network or the Internet.

XPort® and WiPort™ device servers enable you to quickly build Ethernet or 802.11 b/g connectivity into your designs. With a robust operating system, built-in Web server and full TCP/IP stack, XPort and WiPort have everything you need to bring your products to market with lightning speed.

And they're secure. Both have 256-bit AES Rijndael encryption. WiPort features 128-bit WEP and WPA security.

Lantronix provides the networking expertise, so it's easier than you may think. WiPort is even FCC-certified, so you don't need separate certification. Best of all, design changes are usually minimal or unnecessary.

Call or visit our Web site to get a Development Kit and put the power of the Web in your products.

Visit www.lantronix.com for your free Device Networking white paper.

LANTRONIX®
Network anything. Network everything.™

©2006, Lantronix, Inc. Lantronix and XPort are registered trademarks, and WiPort and Device Server are trademarks of Lantronix, Inc.

www.lantronix.com | 800.526.8764

The PWM subsystem can also trigger an interrupt after every cycle. The interrupt handler implements a scan line counter. When the proper time comes (the vertical sync period at the end of each frame), the output polarity of the PWM signal is inverted by changing the PWM match action from clear to set. As a result, the horizontal sync pulse becomes a vertical sync pulse, with all the fancy serrations and equalization pulses that are required (or at least reasonable facsimiles). This continues to the end of the video frame, where the output is restored to horizontal sync duty. It's not exactly correct per the standard, but it works quite well.

The remaining hardware is optional, depending on your particular application. The video output would be considered more traditional if it were presented via a standard RCA socket. This would allow the use of standard video cables to connect to consumer television sets or video monitors. The serial input signal levels to the ATtiny13 pins are TTL. This is perfect for connecting the Video13 to a Parallax BASIC Stamp or any other microcontroller. Connecting directly to a RS-232 adapter would require the use of level-shifting and inversion circuitry. There are several commercially available single-chip solutions for doing this, but for something as simple as this circuit, I'll present an equally simplistic RS-232-to-TTL adapter that uses a single small signal transistor, two resistors, and a diode.

For low data rates (9,600 bps and below), this circuit works quite well. A regulated 5-V power supply is required at low current levels (typically less than 50 mA). A simple linear regulator can be used with appropriate filter capacitors. To allow subsequent programming of the ATtiny13, an in-system programming (ISP) connector can also be added.

Note that the PB0/VIDEO DATA line's low impedance (due to the connection to the three-resistor DAC) will prevent the ISP programmer from working. Adding a 1,000-Ω resistor in series with this signal and attaching the ISP signal directly from the programmer to the device will allow in-system programming, while slightly reducing the contrast of the resulting composite video signal. A shorting jumper across the resistor allows a simple bypass of the resistor when it is no longer required.

SOFTWARE

The video sync signals generated by the PWM subsystem and the discrete resistor DAC provide the necessary backdrop, much like the rhythm section of a band. Let's cover the process of adding some melody.

Monochrome video is modulated by raising the voltage of the video signal from the sync level (which, technically, is just below the official black level) all the way up to white (or 1 V). Setting the VIDEO_DATA line to a digital one does this. Setting the VIDEO_DATA line to a digital zero forms black dots. It doesn't get much simpler than that.

The timing of this stream of black and white pixels is broken up into frames and lines. The frame is the entire two-dimensional picture. It's made up of a series of scan lines, starting from the top of the display and proceeding downward. Each scan line is made up of a sweep of the electron beam (in genuine cathode ray tubes) from left to right. Each horizontal sync pulse designates the beginning of a scan line. The individual bits that make up the scan line must be shifted out with proper timing.

For a point-addressable graphics display (such as the GT-6144), each pixel must be stored in an accessible array. The ATtiny13 has a total of 64 8-bit bytes available as SRAM and 32 8-bit registers. Even if all of those resources could be used as a bitmapped array, that only makes up 768 individual bits of information, which would only make a 32 × 24 grid.

I pursued a different approach. I used the available SRAM to store symbols or characters. I then used the slightly less precious resource of the program memory to decode them. This eliminated the possibility of bit-addressable graphics and with it the image of the Enterprise in orbit over the Earth. The upside was a usefulness beyond a proof-of-concept curiosity.

I originally wanted to see an 8 × 8 character display, using all 64 bytes of

SeaLINK USB serial adapters are the fastest, most reliable way to connect peripherals to any USB-equipped computer.

SeaLINK USB Serial Adapters Provide:
- 1, 2, 4, 8, and 16-Port Models
- RS-232, RS-422, and RS-485 Serial Interfaces
- Data Rates to 921.6K bps
- State Machine Architecture to Reduce Host Processor Overhead
- Operation as Standard COM Ports to the Host Computer
- Lifetime Warranty

sealevel.com > sales@sealevel.com > 864.843.4343


TRI-M SYSTEMS
proudly distributes

Connect Tech Inc.
Industrial Strength Communications
PC/104-Plus to Mini PCI Adapter
AD006
Compatible with standard Universal, 3.3V or 5V PC/104-Plus stack
Mini PCI Type III 124 pin female socket
PC/104-Plus 120 pin stack-through connector
Extended temperature: -40°C to 85°C

TRI-M ENGINEERING
100Mhz PC/104 Module
MZ104
Featuring the new edition ZFx86 FailSafe® Embedded PC-on-a-Chip
Dual watchdog timers, Phoenix BIOS and FAILSAFE Boot ROM
Extended temperature -40°C to 85°C

msystems
FLASH SOLUTIONS
DiskOnChip family of products
Standard interfaces, including 32 pin DIP, UBS and IDE
Multiple capacities (16MB - 1GB)
Advanced Error Correction

www.tri-m.com info@tri-m.com
1.800.665.5600
HEAD OFFICE: VANCOUVER
tel: 604.945.9565 fax: 604.945.9566

available SRAM, but I soon figured out that I would have to allocate at least 2 bytes (and maybe four 4 bytes) as a hardware stack in the SRAM memory. I had to do this because the super-tight synchronization needed for a steady video display required the use of interrupts generated from the PWM timing subsystem. This narrowed down the possibilities to a 6 × 10 or 5 × 12 arrangement. I went with the 6 × 10 arrangement for a total of 60 bytes of display memory.

The appearance of each displayable character is defined as a sequence of 8 bytes in program memory. I could have used smaller encodings, such as 5 × 7, but the math to look them up and decode them would have been more complex, and program space was at a premium. I'd like to know if you find a simple way to do this on the ATtiny13. Using byte-aligned data reduces the complexity of the code significantly.

Eight bytes of data per displayable character quickly start to use up available memory. Uppercase and lowercase characters, numbers, and punctuation could easily add up to 96 characters (or 768 bytes of the 1,024 available), leaving only 256 bytes (or 128 program words) for the actual code. The basic video sync and data code would fit, but there'd be no room for any more features, and I do so love my features. I decided to eliminate the lowercase characters and limit the character set to 64, using half of the program memory as a character generator and leaving half to the program code. This has worked out well. It leaves enough room to add some interesting functionality to the device.

The source code is contained within a single file (video13.asm) and begins with some standard header-type information, including a few comments ("who/what am I"), useful definitions, and macros. The ATtiny13's characteristics are detailed in the tn13def.inc file, which is provided by the Atmel assembler, AVR Assembler 2.1.5. I used Atmel's free AVR Studio (AVR Studio 4.12 Service Pack 1) program to write, assemble, and debug the firmware for the Video13. The free (as in beer) software is available on Atmel's web site for only the Microsoft Windows platform.

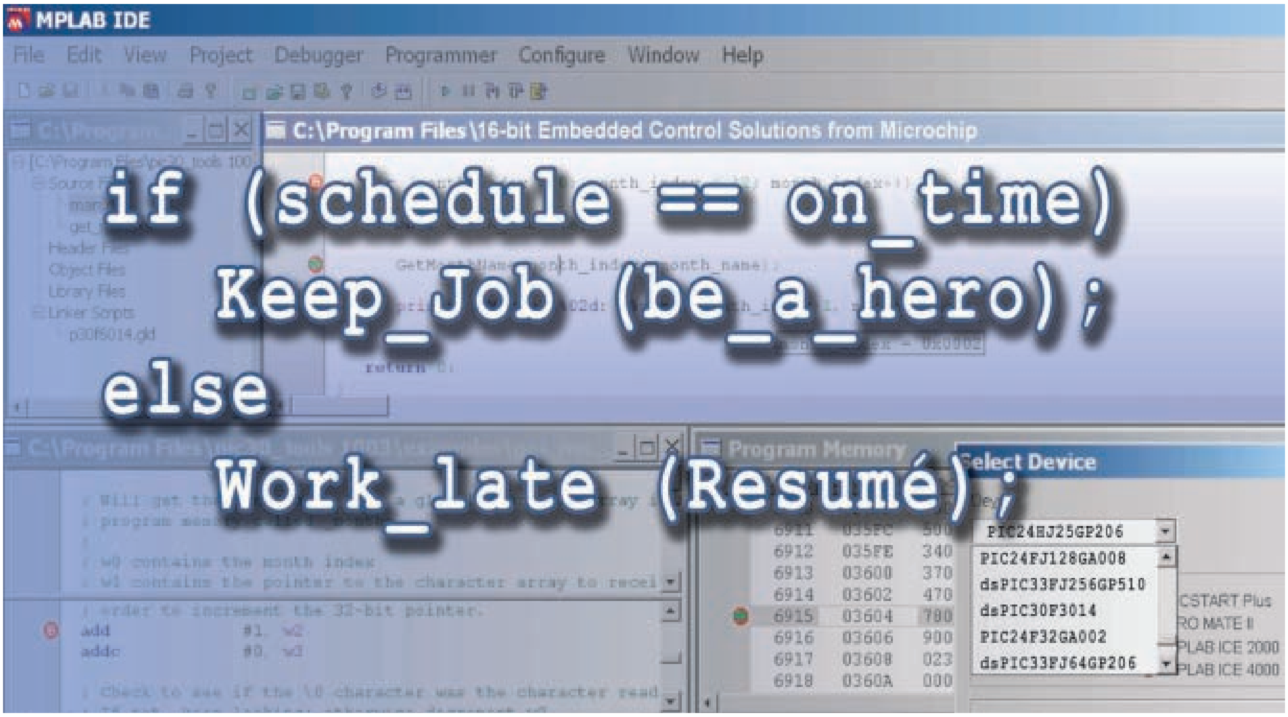
The 8-bit AVR architecture defines 32 general-purpose registers. I used most of them in this application. A series of .def statements defines them. This allows me to refer to the registers by (somewhat) meaningful names instead of register numbers.

The next section of the firmware defines the data to be stored in the device's 64 bytes of EEPROM, using the .eseg assembler directive. This is used as a page storage memory. Because the size of the EEPROM array matches that of the SRAM, a single page of text can be stored in EEPROM and retained even during power loss. The ability to save and subsequently retrieve the current contents of the screen is implemented in the firmware. I shamelessly take this opportunity to put up a brief advertisement and product demo at power-up. You can either edit this out in the source code or simply compose a different page of text and invoke the Save Page command. The EEPROM memory section also contains a persistent variable that is used as a flag to determine if the contents of the EEPROM should be copied to the display memory at power-on. If this flag is not set, the display memory is simply cleared.

The next section of code defines the data segment, using the ".dseg" assembler directive, which corresponds to the SRAM area of memory. Although you can't actually specify the contents of this memory area (only persistent memory, such as program code or EEPROM data, is supported), it serves to define the address of the data area in the memory map. Only one thing is stored there: the array of characters that constitute the display buffer. The display buffer is a linear array, with the first (leftmost) character of the first (top) line occupying the first location in the array. A definition allocating the remaining 4 bytes as a stack is there just as a reminder. It does nothing to actually allocate or initialize the stack.

Now begins the actual firmware by defining a partial vector table. The AVR architecture defines a vector table beginning at program memory

Stay on Schedule



```
if (schedule == on_time)
    Keep_Job (be_a_hero);
else
    Work_late (Resumé);
```

```
PIC24HJ25GP206
PIC24FJ128GA008
dsPIC33FJ256GP510
dsPIC30F3014
PIC24F32GA002
dsPIC33FJ64GP206
```

Envision: Add Features, More Performance, On Schedule

Microchip's 16-bit Microcontroller (MCU) and Digital Signal Controller (DSC) solutions rapidly speed your application development to code freeze… and to production. Microchip offers the industry's only unified 16-bit MCU and DSC architecture with 32-bit performance and 8-bit simplicity. The PIC24F, PIC24H, dsPIC30F and dsPIC33F allow migration from low cost to high performance to DSP capability within the same low cost design environment. ***Download our high quality development environment and test drive our C compiler for free today.***

Spend less time optimizing code!

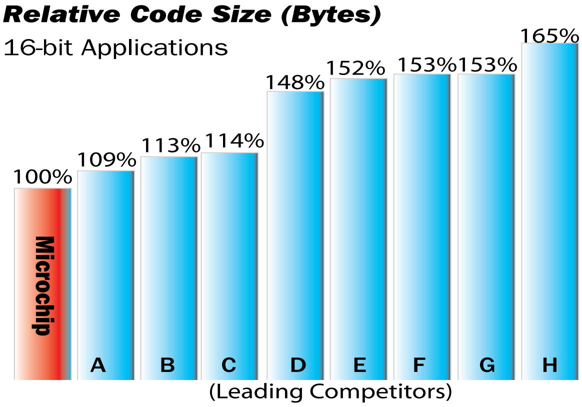
- Industry leading C code efficiency
- Selectable file level optimization for size or speed
- Architecture optimized for C

Libraries shorten your design cycle and reduce risk

- TCP/IP, encryption, FAT16 file management, math and more
- Speech, audio, motor control and communication libraries
- Integrated peripheral driver library for (UART, SPI and more)
- DSP function library (FFT, IIR and more)

Advanced software tools increase productivity

- Visual Device Initializer: auto-generated peripheral code
- DSP filters, motor control graphical user interface
- RTOS and scheduler



Purchase your 16-bit Embedded Control Solutions from…

microchip DIRECT

www.microchipdirect.com

*Get started with **FREE** MPLAB® IDE and MPLAB C30 C Compiler downloads at:*

www.microchip.com/envision

MICROCHIP

Microcontrollers • Digital Signal Controllers • Analog • Serial EEPROMs

The Microchip name and logo, the Microchip logo and MPLAB are registered trademarks of Microchip Technology Incorporated in the USA and in other countries. ©2006 Microchip Technology Incorporated.

address 0. The first vector is always the reset vector that contains code to be executed after a reset (which can have several sources) is encountered or a jump to a location that contains such code. The other vectors follow and generally correspond to the available interrupts that can be triggered by the peripheral hardware. One program word (2 bytes) is allocated for each vector.

The AVR allows a certain flexibility in defining this area. If you aren't using interrupts, your program code can simply begin at the 0 location. No reference to a proper vector table is required. You may also omit any unused vectors from the end of the table. They are not used. Instead of defining the complete list of possible vectors for the ATtiny13 (a total of 10 in all), I use the reset vector, the pin change interrupt vector, and the Timer/Counter0 overflow (PWM) interrupt vector. I also sneak in part of the initialization code in the space that would normally be used by the external interrupt vector because it isn't used in this application. This saves a couple of bytes of program memory.

The reset code follows where the device is initialized following a reset of any kind. On a more sophisticated system, you might pause to determine the cause of the reset and respond accordingly. It all means the same thing in this application, so no differentiation is needed. The code goes on to initialize all the major peripheral subsystems (stack pointer, I/O, PWM, and pin change interrupts) and system variables.

I designed the firmware so that the EEPROM contains a flag that indicates whether or not the data stored in the remainder of the EEPROM should be copied to the screen at power-up. If the flag so indicates, the data is read from the EEPROM and copied to the SRAM; otherwise, the screen buffer is filled with blanks. This flag can be modified using either the Set Restore Flag or Clear Restore Flag commands. The code then drops into the main program loop.

The main loop is a big one, and it does all the video data generation tasks. What it doesn't do is generate the various horizontal and vertical synchronization signals and handle the serial input. These tasks are accomplished with interrupts. The sync tasks are handled in the PWM overflow interrupt. The serial input is managed with the pin change interrupt. I would have preferred to use the external interrupt pin as the serial data input line (because it's more versatile), but the pin was already being used as the video sync output. Pin change interrupts are tricky because they trigger on any change in the state of the input line. The external interrupt pin can be configured to respond to only rising or falling edges, both, or any low-level input.

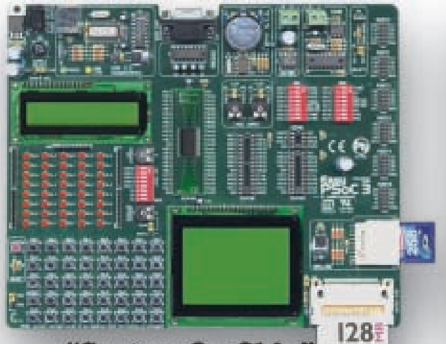
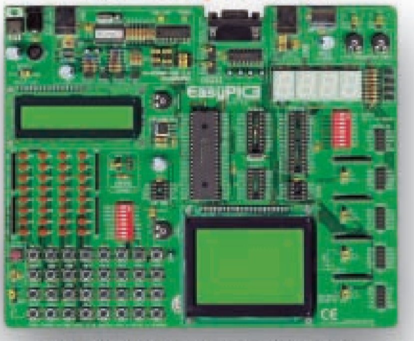
Back in the initialization section, the pin change interrupt is enabled and the pin change interrupt vector points to the handler routine. Should a connected serial device begin to transmit a character, the input line, which is normally at a logical high level, drops to a logical low level to signal the beginning of a character frame. This is known as the start bit. It invokes the pin change interrupt handler.

EMBEDDED DEVELOPMENT IS EASY!

mikroElektronika Development Tools & Compilers

EasyPSoC3
$199.00

EasyPIC3
$169.00

"System-On-Chip" Development System
USB2.0 Programmer - CY8C27843 - LCD
GLCD - DS1820 - CAN - RTC - RS485

Complete 8-40 Pin PIC Development System
USB2.0 Programmer - PIC16F877A
LCD - GLCD - DS1820....much more!

NEW! Easy-VR Stamp + RSC-4x C Compiler

Voice Recognition Development System

USB2.0 Programmer
VR Stamp Module
Microphone - Speaker
LCD - RTC - CF Slot...and more!

$249.00

SECURE ONLINE ORDERING....MAJOR CREDIT CARDS ACCEPTED

circuit-ED.com

Take a peek farther down at the interrupt handler code. The very first thing it does is check the present level of the input pin and immediately return if it determines that the interrupt is the wrong kind. Assuming that it's the right kind (i.e., a valid start bit), the routine preserves some registers and simply counts on its wee fingers to waste some time until it is centered up on the incoming bit cell. A short loop reads in a total of 8 bits, shifting them into a register in the correct order.

Once all the bits have been received, the routine waits for the input line to return to a logical high state, which indicates the stop bit. If the last data bit received happens to be a 1, then the routine doesn't have to wait any more because it has confidence that the stop bit is coming up immediately thereafter.

The reassembled serial character is saved in the recv_data register. The interrupt handler returns after restoring the previously preserved registers. Well-behaved interrupt handlers must save and subsequently restore any registers they want to use in their processing that might also be in use by the foreground task that has been interrupted. Otherwise, odd things happen.

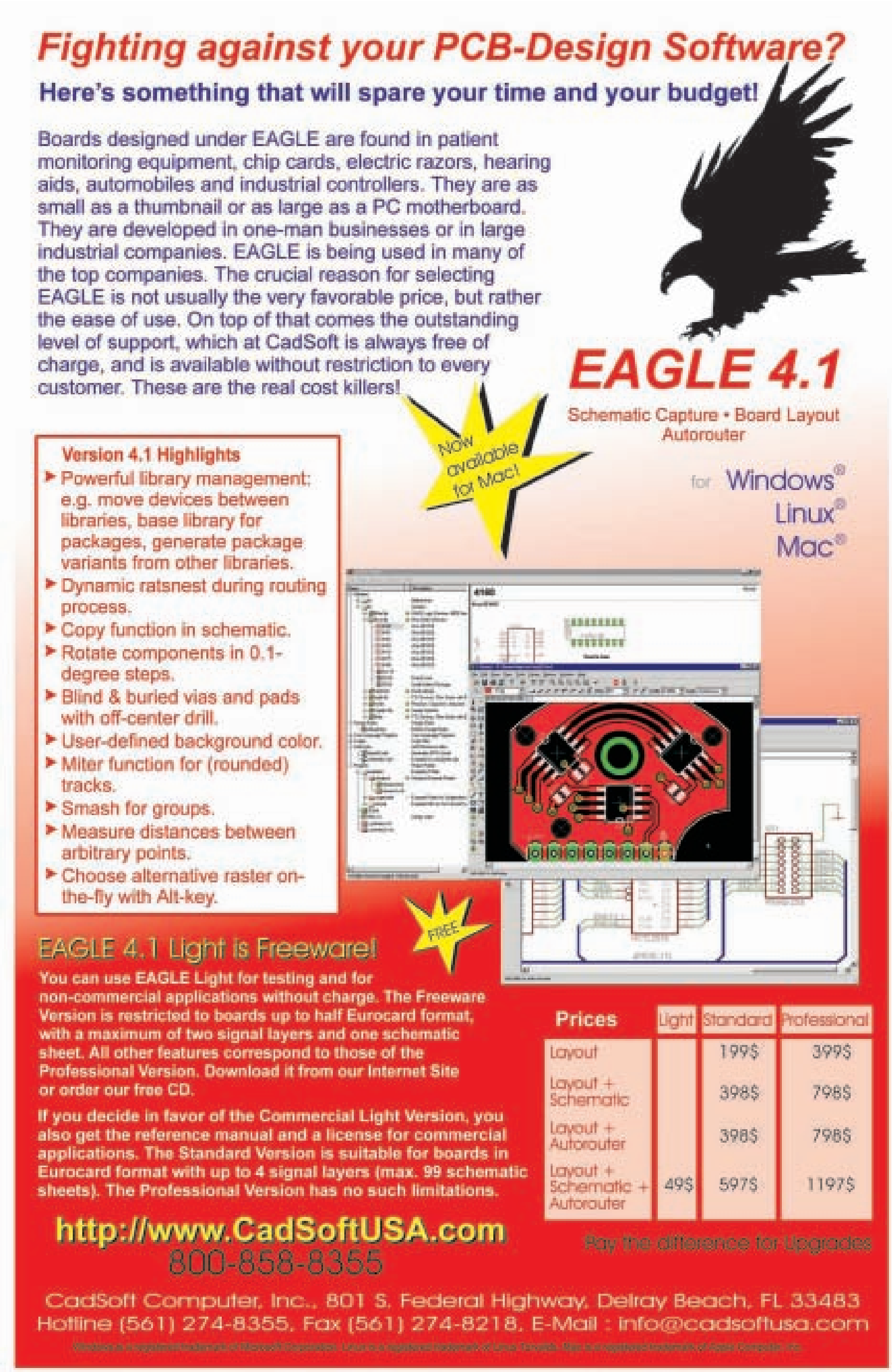
MAIN LOOP

Let's focus on the main loop. The first thing the main loop does is check the recv_data register to see if anything has come in yet. A 0 in this register is a special value that means no characters have been received. If a character has been received, it's copied to another location (the XL register), the recv_data register is cleared, and the incoming data is examined to figure out what should be done. If it is a printable character (i.e., not a control code or command), then the character is case-folded to uppercase and stored in the display buffer at the present cursor location. The cursor location is automatically advanced by virtue of the AVR post-increment addressing mode implied in the store instruction (ST Y+, XL). The cursor address is then examined to see if the end of the memory buffer has been passed. If so, it's pointed back at the beginning of the buffer.

If a nonprintable character has been received, it's assumed to be a control code or command of some sort. The venerable ASCII control codes have been preserved to an extent. Functions such as (nondestructive) backspace, carriage return, and line feed appear to do what they should and correspond to their traditional ASCII encodings. Note that the carriage return is only that. It doesn't automatically advance the cursor to the next line.

Other codes that might not be obvious are the form feed command and the start of heading control code. The former clears the screen instead of ejecting a piece of paper and lining up on a nice, new, blank piece of paper. The start of heading control code simply moves the cursor back to the upper-left corner of the screen, which is also known as the home position.

The remainder of the control codes implement the page memory function



Fighting against your PCB-Design Software?

Here's something that will spare your time and your budget!

Boards designed under EAGLE are found in patient monitoring equipment, chip cards, electric razors, hearing aids, automobiles and industrial controllers. They are as small as a thumbnail or as large as a PC motherboard. They are developed in one-man businesses or in large industrial companies. EAGLE is being used in many of the top companies. The crucial reason for selecting EAGLE is not usually the very favorable price, but rather the ease of use. On top of that comes the outstanding level of support, which at CadSoft is always free of charge, and is available without restriction to every customer. These are the real cost killers!

EAGLE 4.1

Schematic Capture • Board Layout
Autorouter

Now available for Mac!

for Windows®
Linux®
Mac®

Version 4.1 Highlights
- Powerful library management: e.g. move devices between libraries, base library for packages, generate package variants from other libraries.
- Dynamic ratsnest during routing process.
- Copy function in schematic.
- Rotate components in 0.1-degree steps.
- Blind & buried vias and pads with off-center drill.
- User-defined background color.
- Miter function for (rounded) tracks.
- Smash for groups.
- Measure distances between arbitrary points.
- Choose alternative raster on-the-fly with Alt-key.

FREE

EAGLE 4.1 Light is Freeware!

You can use EAGLE Light for testing and for non-commercial applications without charge. The Freeware Version is restricted to boards up to half Eurocard format, with a maximum of two signal layers and one schematic sheet. All other features correspond to those of the Professional Version. Download it from our Internet Site or order our free CD.

If you decide in favor of the Commercial Light Version, you also get the reference manual and a license for commercial applications. The Standard Version is suitable for boards in Eurocard format with up to 4 signal layers (max. 99 schematic sheets). The Professional Version has no such limitations.

Prices	Light	Standard	Professional
Layout		199$	399$
Layout + Schematic		398$	798$
Layout + Autorouter		398$	798$
Layout + Schematic + Autorouter	49$	597$	1197$

Pay the difference for Upgrades

http://www.CadSoftUSA.com
800-858-8355

CadSoft Computer, Inc., 801 S. Federal Highway, Delray Beach, FL 33483
Hotline (561) 274-8355, Fax (561) 274-8218, E-Mail : info@cadsoftusa.com

(see Table 2). The ASCII control codes define a group of device control function codes that are intentionally vague and device dependent. They are used to save and restore the screen, as well as to manipulate the flag that controls whether or not the saved page is brought up automatically at power-on.

Hex	Control	Description
0x01	^A	Start of heading (home cursor)
0x08	^H	Back space (nondestructive backspace)
0x0a	^J	Line feed
0x0c	^L	Form feed (clear screen)
0x0d	^M	Carriage return/enter
0x11	^Q	Device control 1 (save screen)
0x12	^R	Device control 2 (restore screen)
0x13	^S	Device control 3 (set restore flag)
0x14	^T	Device control 4 (reset restore flag)

Table 2—*The Video13 understands and executes these control codes. Traditional ASCII functionality is retained where possible, making it easier to think of the Video13 as a "glass teletype." Additional control codes can be added by the clever tinkerer.*

VIDEO GENERATION

Now comes the fun part: video generation. In fact, it's so exciting that the first thing you should do is take a nap. If this seemingly endless narrative hasn't done the trick already, then let's just have the processor go to sleep. Why? Because the video signal has to be synchronized exactly with the clock or else wavy and illegible images will appear. Remember that the timer was initialized for PWM duty and to generate an interrupt on a precise schedule (after every scan line).

If you use a polling technique to wait for the event, the response time could vary by as much as three or four clock cycles for every scan line. That's completely unacceptable. By going to sleep, you ensure that the processor wakes up and begins processing the video at the exact instant required.

To put the ATtiny13 to sleep, the sleep enable bit must be set in the microcontroller unit control register (MCUCR). This must be immediately followed (or at least within four clock cycles) by the SLEEP instruction. While setting the sleep enable bit, take the opportunity to set the Sleep mode bits to select which kind of sleep is desired. The ATtiny13 supports three distinct Sleep modes, each with different levels of sleepiness: Idle mode, ADC Conversion mode, and Power Down mode. Idle mode is the only one that allows the timer to continue to run (and to subsequently and periodically generate interrupts). That's the one you want. The other modes are useful for other applications.

So, time passes and eventually the moment arrives when the timer overflows and an interrupt is generated. This causes the processor to wake up and execute a jump to the appropriate interrupt vector in the table. It then resumes code execution from just after the SLEEP instruction. The first thing to do in the interrupt handler is to keep a count of where you are on the page. More specifically, you need to know which scan line is being displayed.

The scan line counter is composed of two 8-bit registers because the total number of scan lines is 262. This is just beyond the range of an 8-bit number. Luckily, 16-bit arithmetic is trivial with the AVR RISC core. Due to carry and zero propagation, many 16-bit operations can be performed using just two instructions. A good example is the scan line down counter. Down counting is used whenever possible because it's easier to test for zero than any other number. So, the scan line counter is decremented, and because the zero bit is already set (unless it isn't), a conditional jump to the code to handle the end of frame situation is put in place.

What if it's the end of the frame? Reset some variables (scan line counter register pair, character generator line counter, and display memory pointer), reconfigure the PWM control register to resume emitting normal (i.e., noninverted) video sync pulses, and the interrupt handler is finished.

If it isn't the end of the frame, it might be near enough to the end to be of concern. By this I mean it's time to start generating the "other" kind of video sync pulses. If the scan line count indicates that there are only 20 scan lines remaining, it's time to switch the PWM circuit into vertical Sync mode. Changing 1 bit in the PWM control register does this. Voila! Instant vertical sync pulses. Now return to the main loop and continue from there.

In case it's not the end of the frame or even nearly the end of the frame, it still might be a case of getting woken up for nothing. Even though there are 262 scan lines (actually, there are supposed to be 262.5, but that's inconvenient, so forget it), not all of them are visible. Some are past the top of the visible part of the display and others are likewise past the bottom. There's nothing you need to do there. The PWM circuit sends out all of the sync pulses automatically, so keeping up with the current scan line is all that's required. So, it's back to the beginning of the loop to wait some more.

After eliminating all of the other possibilities, you have arrived (like Sherlock Holmes) at what must be the truth. Well, not so much the truth, but what I like to call "show time." This is where the visible part of the video signal is generated. But wait! No, literally, you have to wait a bit.

Listing 1—*The characters displayed by the Video13 are defined in the ATtiny13's program memory and included in the source code. Ones produce white pixels and zeros produce black pixels. Sixty-four characters—including uppercase letters, numbers, and punctuation—are included. You can create your own character sets by reprogramming the device.*

```
; 0x30 0
; .oooooo.
; oo....oo
; oo....oo
; oo....oo
; oo....oo
; oo....oo
; .oooooo.
.db 0b00000000, 0b01111110, 0b11000011, 0b11000011, 0b11000011,
0b11000011, 0b11000011, 0b01111110
```

Just as there are vertical dead spots on the screen, there are similarly areas to the left and right of the screen that just can't be seen. Collectively, these areas are referred to as "overscan." So, close your eyes and count backwards from 90. It doesn't take an AVR running at 20 MHz long to do this. It takes about 13 µs in fact. This gets you right up along the left edge of the screen, time-wise. The electron beam begins its sweep. The video amplifiers in the monitor are eager to modulate the signal you want to send. What shall you do?

Initialize some variables, that's what! Make a copy of the current display memory pointer and reset the characters-per-line counter to 10. Now go into a loop. The fun stuff happens inside that loop.

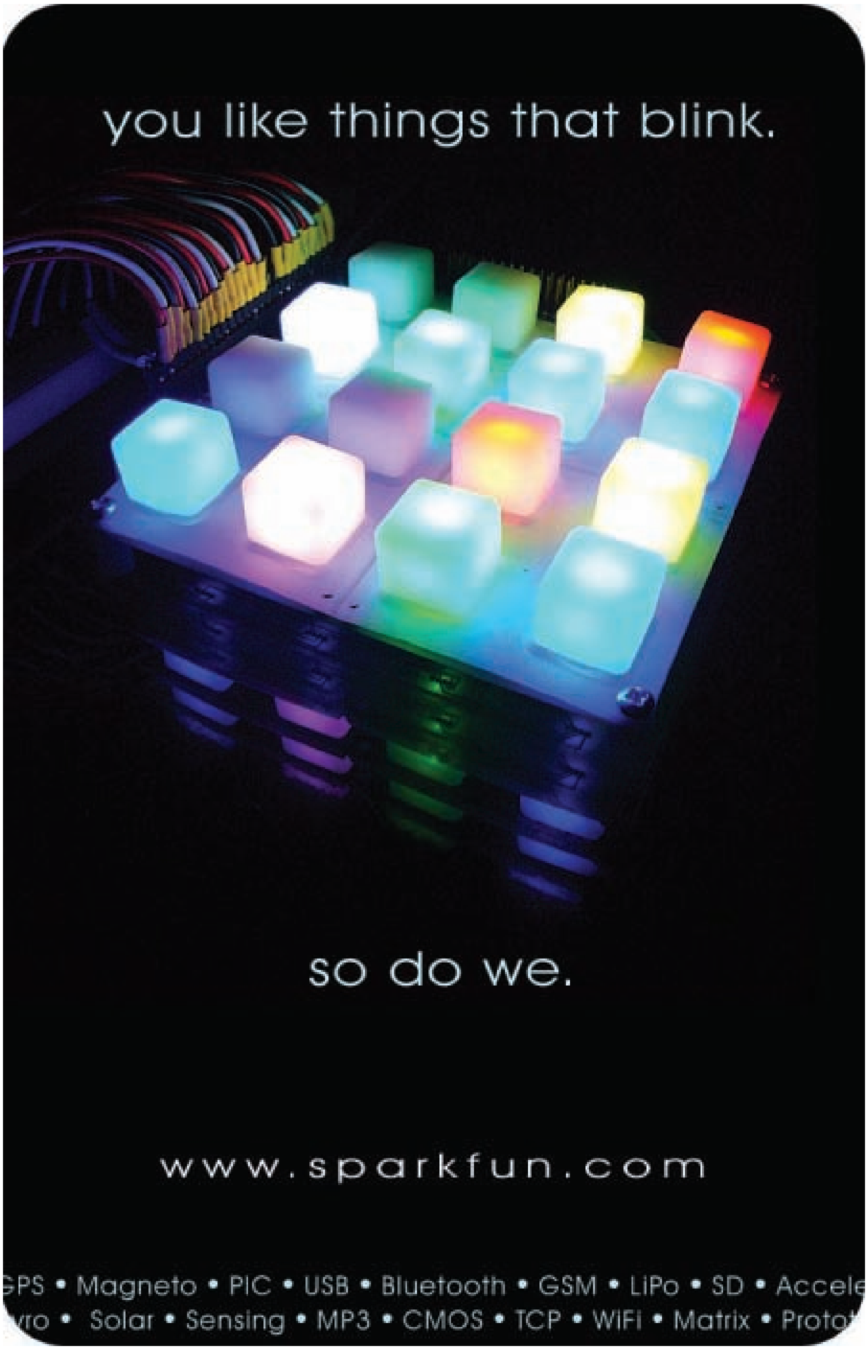
Let's take a trip through this loop. The novelty will wear off quickly. The loop begins by fetching the character code from the display memory. This character will have to be read out several times before you're done with it because the displayed image of the character is composed of multiple scan lines. In fact, each pixel of each character is repeated three times in the vertical dimension. That's why a copy of the memory pointer is used. You have to back up and do it again, over and over, until it's done. Only then can you go on to the next row of characters.

The lowest 32 ASCII codes are control codes and technically unprintable (i.e., invisible), so don't waste any character generator table space on them. The character code itself becomes an index into the character generator table. Instead of adjusting the character code index by 32, the address of the character generator table is offset by 256 (i.e., 32 × 8 bytes per character). The character code index is then multiplied by eight (left-shifted three times because the ATtiny13 lacks hardware multiplication), and the current scan line for this particular row of character generator data is added. The load program memory (LPM) instruction causes the data in the table (pointed at by the Z register) to be copied into register R0. That's the data that you want to shift out.

Like most repetitive tasks, shifting out the data is performed as a loop. A loop counter is reset and the bits are shifted out by writing to the output port (port B). The bits are then shifted. The loop counter is incremented and tested for the terminal value, which is eight in this case. A conditional branch has the loop run until it's done.

What's odd about this loop is that it's too fast. I hadn't counted on this. I was worried that the humble little AVR would have to be tricked and coerced into providing the processing speed necessary to generate a live video signal. I envisioned arcane performance hacks and having to take advantage of obscure modes and undocumented instruction side effects. Now I had to give it something to do: busy work.

The no operation (NOP) instruction is perfect for filling little timing gaps such as this. However, even that wasn't enough. I ended up with two NOPs in the loop. I used an up counter (instead of the more efficient down



counter) to get the timing just right. Another good time-waster, I found, was the fake subroutine call. Call a subroutine that does nothing but return. This takes seven processor cycles and only one word of program memory as opposed to seven NOPs. You can even use the return from subroutine (RET) instruction from a legitimate subroutine as the target for the fake subroutine call. Just label the RET instruction "fake" or something similar.

OK. Imagine doing that several more times and all the characters for that row have been scanned (sort of). That's only one scan line. Each pixel is three scan lines tall, so the process must be repeated three times for that particular line of pixels. Each character is made up of eight lines of pixels, so imagine a further level of nesting there.

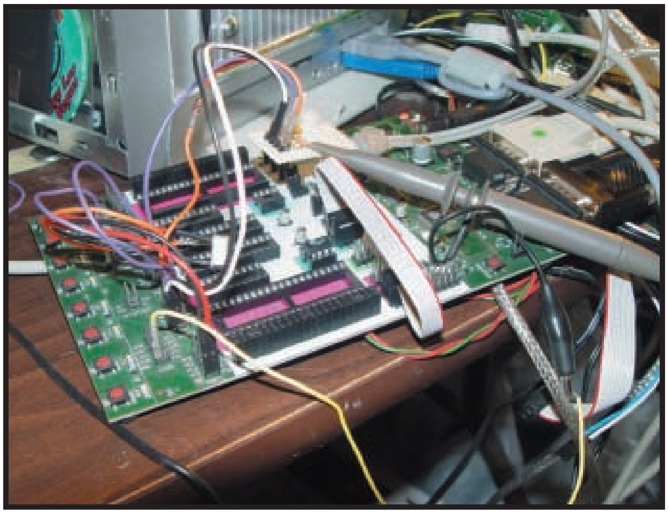


Photo 2—*The first incarnation of the Video13 was built using Atmel's STK500 development board. I recommend starting any AVR project with this board because it offers a lot of options while simultaneously reducing the number of variables in your new circuit.*

Oh, don't forget to clear the video data line at the end of each character to create some negative space between the characters. The last pixel shifted out could just as easily be a white one as a black one. Once they've all been shifted out, clear the video data line with the simple one-word clear bit in I/O port (CBI) instruction.

After all of the characters from the current row have been completely displayed, it's time to bump the memory pointer (affectionately named "W" in order to be in line with X, Y, and Z) by 10 so that it points to the next line of data to be displayed. And then repeat this ad infinitum. That's it for the code.

CHARACTER GENERATOR TABLE

Now let's focus on how the character generator table is constructed. I happen to like the appearance of the characters. This is the surest indicator that someone else won't. Each character is defined by a define byte (`.db`) assembler directive followed by eight data values that are expressed in binary format. I chose that format because it lets me see which bits are on and which are off.

Take a look at the definition for that most useful symbol for nothing: zero (see Listing 1, p. 20). The semicolon (`;`) introduces a comment according to the Atmel Assembler's syntax. I start by indicating which character I'm attempting to portray, along with its ASCII code in hexadecimal. Next comes the ASCII art rendition in an 8 × 8 cell. I use periods to indicate blank space and lowercase `o` characters to show white pixels. Following all that graphic goodness comes the actual data. Hopefully, you can see the correspondence. Another formatting option is to use the `\` line continuation character, which allows you to fold the long `.db` directive up so it looks more like the ASCII art version:

```
.db 0b00000000, \
0b01111110, \
0b11000011, \
0b11000011, \
0b11000011, \
0b11000011, \
0b11000011, \
0b01111110
```

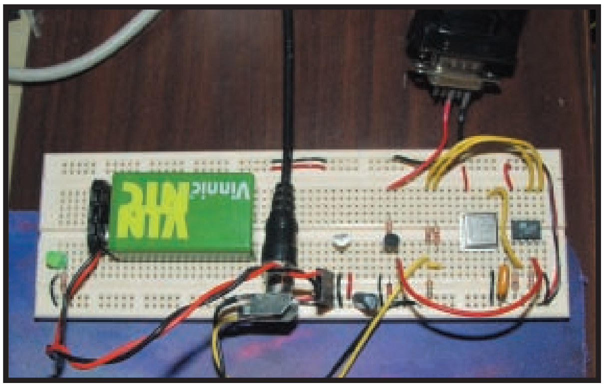


Photo 3—*After getting the basics of the Video13 system working, I moved the entire circuit from the STK500 to a solderless breadboard, one section at a time. I try not to make too many changes at once, and I check that everything is still working after any "improvements."*

You can't just put each byte on its own line with separate define byte directives. The assembler, which wants everything to be word-aligned, pads all 8-bit data with an extra byte of all zeros. This isn't what you want. If you squint or perhaps poke a finger into your eye edgewise, you can almost make out the zero character among the units and ciphers. I happen to prefer the ASCII art technique.

On my list of things to do, I've added a utility program to help design alternate fonts for the Video13. Perhaps it will be ready soon. If not, stick with the provided font. It's legible (for the most part).

BUILDING THE Video13

The original prototype was built on the firm foundation of Atmel's STK500 development system, along with many flying wires (see Photo 2). I built the video DAC on a small piece of perforated board and attached it to the STK500 with jumpers.

The second prototype was built on a solderless breadboard. I transitioned the circuit from the STK500 to the breadboard in stages. The first stage still relied on the STK500 for regulated power, RS-232 level shifting, and the flying video DAC assembly. I also used a special socket for the ATtiny13 that I had designed previously to route the ISP (programming) pins to a header to connect back to the STK500.

I eventually put the entire circuit

on the solderless breadboard with room to spare (see Photo 3). I think it clearly demonstrates just how few components are actually required to add a simple TV interface to a project.

UPDATES AHEAD

I enjoyed designing the Video13. At one point, I actually broke it. Fixing the system took five times the original number of "me" hours as the original design.

I also discovered late in the game that my humble jWIN JV-TV1010 portable black-and-white television with video input jack was ever so forgiving of video timing anomalies. Last-minute testing on other devices, including an aging Panasonic television and some Sony portable Watchman-style color TVs, showed a significant shearing effect at the top of the screen. I attributed this to the lack of a properly interlaced signal and reduced the number of vertical lines of resolution from 192 to 144. This enlarged the top and bottom overscan areas, giving all of the TVs plenty of time to synchronize to my slightly nonstandard video signal.

Due to the interruption of video processing to handle incoming serial data, the picture tends to jump a little as characters are typed. There is perhaps a clever resolution to this phenomenon, but it eludes me at the moment. I can't recommend the Video13 for applications that require a high update rate. But it's perfect for static displays and low update scenarios, like time and temperature or alarm indicators.

Author's note: I'll post Video13 updates on my web site (http://dalewheat.com). Preprogrammed Video13 chips are available for $5 plus shipping.

Dale Wheat (dale@dalewheat.com) is an old-fashioned tinkerer, inventor, musician, philosopher, and student of the human condition. He has no formal professional education. He enjoys reading science fiction short stories and playing a variety of musical instruments. In the summer, he enjoys mowing two acres of grass and in the winter enjoys not mowing it. In 2005 and 2006, Dale was the president of the Dallas Personal Robotics Group (http://dprg.org), one of the nation's oldest personal robotics interest groups.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2006/195.

REFERENCE

[1] Atmel Corp., "ATtiny13: 8-bit AVR Microcontroller with 1K Bytes In-System Programmable Flash," rev. 2535E-AVR, 2004.

RESOURCE

D. Lancaster, *The Cheap Video Cookbook*, Sams, Indianapolis, IN, 1975.

SOURCE

ATtiny13 Microcontroller and STK500 development system
Atmel Corp.
www.atmel.com

by René Geraets

Generate Video from Software

René uses an M16C/62P microcontroller to generate video signals without dedicated video hardware. The M16C/62P also produces a PAL or NTSC analog RGB color video signal.

The ever-increasing clock speeds and computational power of modern microcontrollers enable faster, more complex processing. The increased performance of such devices allows you to work on applications that were previously out of your reach. For example, the high throughput of modern microcontrollers enables them to process and generate video signals.

I recently designed a CD player with video output to a TV (see Photo 1). The system is fairly simple. A Renesas Technology M16C/62P microcontroller is connected to an ATAPI CD-ROM player so it can play audio CDs. At the same time, the microcontroller produces a PAL or NTSC analog RGB color video signal. A television displays the video signal and provides a GUI to operate the CD player (see Figure 1).

The M16C/62P microcontroller generates the video signal without any additional dedicated video hardware. All that's needed are a few simple resistors and transistors to transform the microcontroller's 5-V digital outputs into the voltage range for video. In this article, I'll describe the system.

VIDEO SIGNAL ANATOMY

Video is a sequence of still images (frames) that produces the illusion of motion when it is played fast enough. The European TV system (PAL) produces 25 frames per second. The American system (NTSC) produces 30 frames per second. Each frame consists of image lines that form the vertical resolution on your television. Each frame in PAL consists of 625 lines. NTSC frames have 525 lines. Therefore, although PAL has fewer frames per second than NTSC, it has more image lines per frame, and thus a higher resolution.

With respect to line frequency, the lower frame rate and higher number of lines per frame cancel each other out. PAL and NTSC signals contain about the same number of lines per second. In both PAL and NTSC, each video line is produced in 64 µs.

Showing video at 25 or 30 frames per second can result in visible flickering on traditional CRT televisions. Illuminated phosphor particles in the picture tube produce images. The particles slowly dim before they are illuminated again for the next image. The human eye is fast enough to notice these variations in light intensity if the image is updated only 25 (or 30) times per second. The solution is to update the screen twice as fast.

Photo 1—*The system includes a CD-ROM player, an M16C62P starter kit, an experiment board with the digital I/O-to-video level conversion, and an RGB monitor for displaying the video. The audio is rendered through the speakers in the monitor. Instead of the monitor, you can use the television in your living room.*

An image is divided into two fields. One consists of all even lines. The other consists of all odd lines (see Figure 2). Thus, each field spans the complete screen area but can be written to the screen in half the time it would take to write the entire image. This way the screen is updated 50 (or 60) times per second. This process is called interlacing.

A varying analog voltage can represent a black-and-white video signal. Note that 1 V represents white and 0.3 V represents black. The voltages between them represent corresponding shades of gray. The voltage changes over time to represent the different shades of gray that build up the image lines. The faster you update this voltage during an image line, the more features you can show horizontally. This determines the horizontal resolution.

It doesn't make sense to have a big difference between the horizontal and vertical resolutions. You know the vertical resolution (the number of lines per frame) and the duration of each image line, so you must be able to vary

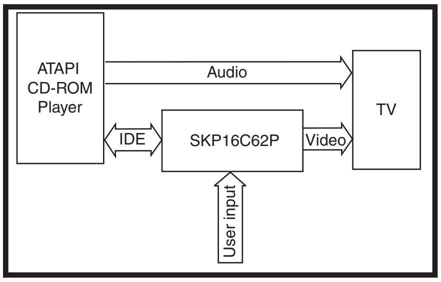


Figure 1—*The M16C62P starter kit board talks via an IDE bus to an ATAPI CD-ROM player to play audio CDs. The CD player's audio output is routed directly to a television, along with a video signal that's generated by the microcontroller.*

the voltage about 10 million times per second for a high-quality television picture. With its 24-MHz clock frequency, the M16C/62P is not fast enough to reach this 10 million mark, but a factor of three or four less still provides a reasonable resolution for small applications.

To make sure the picture is stable on your screen, you need a way of identifying exactly where each image line starts (i.e., horizontal synchronization) and where each field starts (vertical synchronization). H sync is encoded in the video signal by lowering the voltage below the 0.3-V black level for 4 µs. Figure 3 is a typical image line. The V sync pattern is more complicated. It consists of a series of pulses and also distinguishes between the even and odd fields. V sync patterns are shown in Figure 4 (p. 26).

Adding color to the black-and-white signal is accomplished by superimposing a 3.58-MHz (NTSC) or 4.43-MHz (PAL) carrier onto the black-and-white signal. The amplitude and phase of the carrier determine the color saturation and hue. The analog value that forms the black-and-white signal determines the color's intensity, or luminance. The resulting signal is called a composite baseband video signal (CVBS). This is the standard used for television broadcasts. It's also the kind of signal that comes out of a VCR.

Creating a CVBS signal from software is probably impossible for any one microcontroller on its own. Generating the 3.58- or 4.43-MHz carrier wave and varying its amplitude and phase with enough precision and resolution to create a nice-looking color signal would require additional dedicated video hardware. But that doesn't mean you can't create color video from software. With the advent of new types of video equipment, new video encoding standards are also implemented as inputs for television. For example, DVD players not only create a CVBS output signal, but also present the video in the form of an RGB signal. An RGB signal is much easier to create with software.

With the CVBS signal, all video information (luminance, color saturation and hue, and synchronization pulses) is transported via one wire. The RGB signal uses four wires. In the RGB representation, the color information is broken down into its red, green, and blue constituents. Each of the R, G, and B components is transported as a 0.7-V_{pp} signal over its own wire (much like the black-and-white video signal was). The fourth wire carries the synchronization pulses (and thus forms a 0.3-V_{pp} signal). By separating the color information over three wires, the need for the hard-to-generate 3.58- or 4.43-MHz carrier has disappeared. The RGB format brings color video back within the reach of the M16C/62P microcontroller. That's why I used it.

TRANSFORMING DIGITAL I/O

The M16C/62P microcontroller has 5-V digital outputs. To transform these into an RGB video signal, you need some additional circuitry. However, you can limit it to a couple of simple resistors and transistors.

First, you need to transform the M16C/62P microcontroller's 0- to 5-V output to the 0.7-V_{pp} video level. This is easily achieved with a two-resistor voltage divider. R8 and R10 form this voltage divider for the red output signal (see Figure 5, p. 27). R9 adds a small DC offset to it. Transistor T2 is connected as an emitter follower; with R11, it provides the signal with the 75-Ω output impedance that's required for video. The green and blue outputs are created with identical circuits. A television's input impedance is also 75 Ω, so the actual video signal level is half of the level present at the transistor's emitter. If you calculate the high and low values for the three color components, you'll find that the difference is 0.6 V. That's about 85% of the maximum level (0.7 V), and it produces nice bright colors.

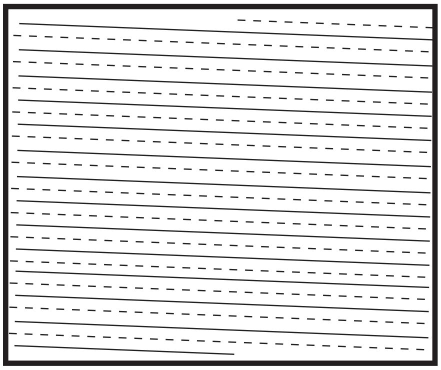


Figure 2—*The even field (solid) ends in a half image line. The odd field (dashed) starts with a half image line. The two fields form one frame. This is called interlacing. In reality, each field is made up of more image lines than what you see here. In PAL, a field has 312.5 lines. In NTSC, a field has 262.5 lines.*

A similar circuit is used for the sync output. To generate a pure RGB signal, it's sufficient to transform only the 0- to 5-V sync output from the M16C/62P microcontroller into a 0.3-V_{pp} signal like with the R, G, and B outputs. But I added something extra. The RGB output from a DVD player often contains a complete CVBS signal on the sync output. The synchronization pulses along with the R, G, and B signals are used when connected to an RGB input, and the complete CVBS signal from the sync output is used when connected to a CVBS input. I did something similar.

I can't create a complete color CVBS signal (it's too hard to encode the R, G, and B components into the 3.58- or 4.43-MHz carrier wave), but I can create an old-fashioned black-and-white video signal from it. That involves simply adding up the luminance of the R, G, and B components. In the analog world, adding voltages involves just tying them together with a few resistors.

Take another look at the circuit

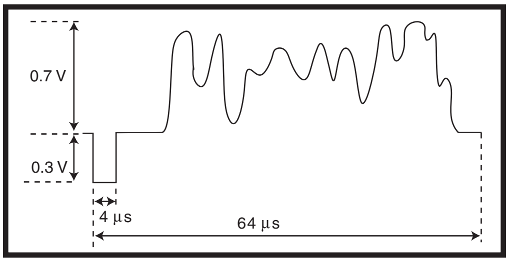


Figure 3—*This is a typical image line for a black-and-white video signal. The image line starts with a 4-µs sync pulse. The video data starts 8 µs after that. CRT televisions need this time to move the image-producing electron beam back to the left of the screen. A color video signal has a 3.58- or 4.43-MHz carrier wave superimposed on this holding the color information.*

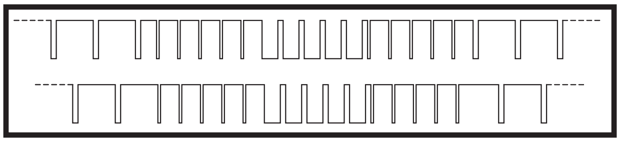


Figure 4—*Check out this V-sync pattern for even fields (top) and odd fields (bottom). The V-sync of the even field starts and ends at the image line start points. The V-sync of the odd field starts and ends in the middle of an image line.*

diagram in Figure 5. R1, R2, R3, R4, and R6 mix the color and sync signals. R5 creates small DC offset. Just like with the color signals, T1 and R7 create a 75-Ω video output. The three color components are not mixed in the same ratio. When converting to black and white, green is brighter than red, which in turn is brighter than blue. I mixed the green, red, and blue component in a ratio of 4:2:1. This results in eight evenly distributed shades of gray in the black-and-white signal.

So, if your television set does not have an RGB input, connecting the sync signal alone to the composite (CVBS) input provides a black-and-white image. Alternatively, by tying R1 and R3 to G instead of to R and B, the Sync output becomes a sync-on-green output that can be used along with the R and B outputs to feed into a television or monitor that doesn't have a separate sync input on its RGB input.

OUTPUT RESOLUTION

The processor supplies the R, G, B, and sync signals on four of its I/O pins. Using four pins on the same I/O port makes it possible to change all of the levels in just one output operation (writing to the port register). To obtain a usable horizontal resolution, it must be possible to change the value on the R, G, and B outputs quickly. Each image line in the video stream lasts only 64 µs. Only 52 µs form the visible part on the television screen. The other 12 µs are for the sync pulse and horizontal flyback. Most televisions even clip a bit from the left and right when displaying the image, so you shouldn't put important information near the edges of the screen. For this project, I assumed a guaranteed visible period of 45 µs on each image line.

To be able to show a reasonable amount of text on the screen, you need at least 120 pixels (20 characters) from left to right. Because the processor is clocked at 24 MHz, I aim for a pixel frequency of 3 MHz. This means the processor has eight clock cycles to output each pixel. Note that 45 µs at a 3-MHz pixel frequency gives a horizontal resolution of 135 pixels (i.e., 45 × 3). This meets the requirement of at least 120.

The number of lines in each field restricts the vertical resolution. A PAL image contains about 560 visible lines. An NTSC image contains about 470 visible lines. These lines are divided over the two fields of each frame. Thus, per field, it's 280 for PAL and 235 for NTSC.

It doesn't make sense to use a higher resolution in the vertical direction than in the horizontal direction. (Remember that the horizontal resolution is 135 pixels per line.) Using two video lines per pixel in the vertical direction gives a vertical resolution of 140 for PAL and 117 for NTSC, which roughly corresponds with the horizontal resolution.

It's where 8-bit and 32-bit become compatible.

For more go to freescale.com/continuum

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ©Freescale Semiconductor, Inc. 2006.

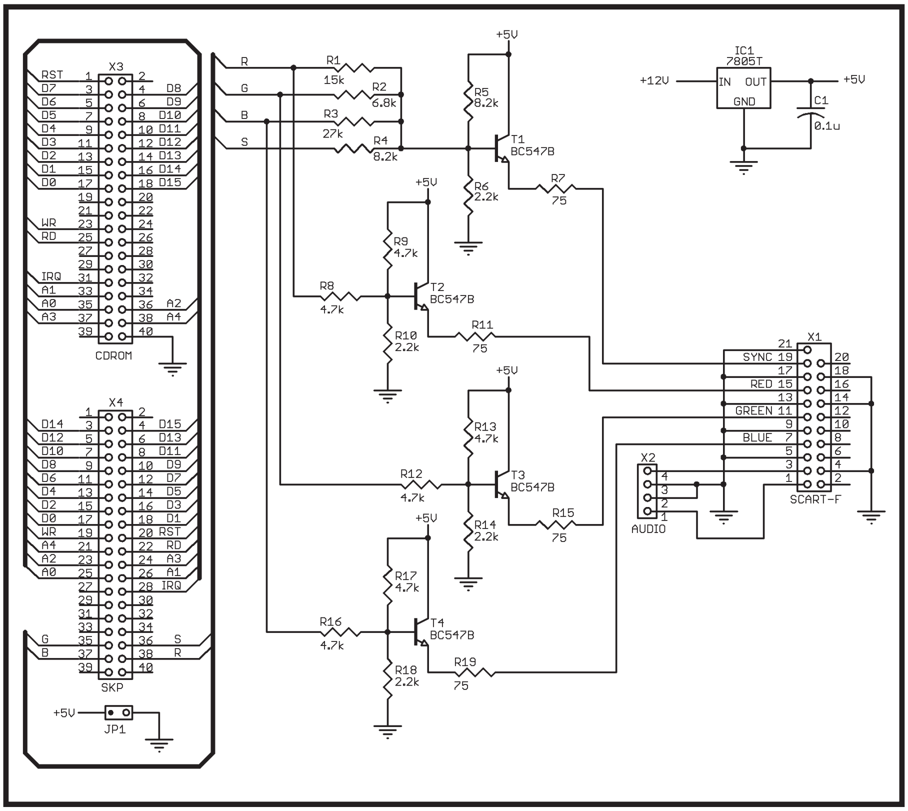


Figure 5—*The connectors on the left connect the CD-ROM player to the M16C62P starter kit board. They aren't needed if you want to build only the video production portion of the design. The SCART connector on the right is the audio/video connection on European televisions. For American televisions, replace it with cinch connectors for the individual signals.*

(In an ideal situation, you'd have square pixels, but this is close enough.) To be on the safe side (a television set might clip some more from the top and bottom), I restricted the area dedicated for output to the center 240 image lines for PAL and 200 image lines for NTSC. This resulted in a resolution of 135 × 120 for PAL and 135 × 100 for NTSC.

MEMORY RESTRICTIONS

Another reason to keep the resolution relatively low is to make the image fit in the M16C/62P microcontroller's internal memory. I need 4 bits per pixel (1 bit for each I/O pin allocated to video), but use a complete byte to store each group of 4 bits. This makes it possible to copy the pattern directly to the I/O port without any mask or shift operations on the bit pattern. This is important because I have a limited amount of time to spend on each pixel (only eight processor clock cycles!), and I use the DMA controller for the actual output streaming.

Given such choices, I can calculate the required amount of memory. Each

Welcome to the Controller Continuum

At Freescale, we've broken the bit barrier. The Controller Continuum is our product roadmap that combines the simplicity of working with 8-bit and the high-performance of 32-bit. Soon, we'll be rolling out pin-for-pin compatible devices that let you easily upgrade your 8-bit designs to 32-bit. By sharing common peripherals and tools, such as our now easier-to-use Fast Track services for the CodeWarrior® tool suite, you can accelerate design cycles and speed products to market. It's not only how to make smart things work. It's how to work smarter.

freescale™
semiconductor

video line contains 64 µs of data (useful picture content plus the sync pulse). At a 3-MHz pixel clock, the amount of memory to store each line amounts to 192 bytes (i.e., 0.000064 × 3,000,000). With 120 lines for PAL, the total amount for the image is 23 KB (120 × 192). It is slightly less for NTSC because I use only 100 image lines. The 23 KB fit well within the available 31 KB of memory in the M16C/62P microcontroller. This leaves enough room to spare for the other variables, data structures, and stack of the application.

USING DMA

The large amount of RAM (for this type of microcontroller) in the M16C/62P is not the only nice feature I use for video production. The M16C/62P microcontroller's DMA controller also comes in handy. The DMA controller can be configured to repeatedly copy bytes of data from one location to the other, where either the source address pointer or destination address pointer can be automatically incremented. The trigger for each successive copy action can be chosen from a variety of interrupt sources. And because the M16C/62P uses memory-mapped I/O, any peripheral or control registers can be the source or destination of the transfer.

An obvious application for the DMA controller is to create large communication buffers (e.g., to transmit a large block of data over the UART in one go). Set the source pointer to the beginning of your buffer, set the destination pointer to the UART data register, and select the UART transmit interrupt as the trigger for the DMA actions. Now each time a byte transmits over the UART, the DMA controller will automatically feed it the next byte.

Any memory address can be chosen as the output location. So that means not only communication registers like the UART data register, but also less obvious locations like the address of an I/O port. And, instead of the UART transmit interrupt, a timer interrupt can be chosen as the trigger for the DMA copy actions. Well, that's exactly what I need to produce the video data!

I set the source pointer to the 23-KB chunk of memory reserved to store the output pixels and have it increment after every copy action. I fix the destination pointer to the memory-mapped I/O address of the output port containing the video output pins. I then trigger the copy actions from a timer that's set to a frequency of 3 MHz (the pixel output frequency). Instead of having to feed each individual pixel to the I/O port (one action every 8 clock cycles), the processor now has to reconfigure the DMA controller only once per video line (one action every 1536 clock cycles). As a result, the DMA controller inside the M16C/62P does the actual video output. The main processor is then free to perform the other tasks, like interfacing to the CD player, filling the frame buffer with the strings, and checking and responding to user input.

PERFECT FIT

The M16C/62P microcontroller turned out to be a perfect fit for video production. The 24-MHz clock frequency makes it fast enough to generate video at a reasonable resolution. The 31 KB of embedded memory provides enough storage to hold the video picture. And, thanks to the embedded DMA controller, which handles the most intensive part of the video production, there's plenty of processing power left to run all kinds of useful applications. I interfaced it to a CD-ROM player, but the availability of video output is an enabler for many more interesting applications.

René Geraets (r.geraets@gmail.com) has a Master's degree in computer science. He has been working as an embedded-software engineer for the past 10 years. René now works in the advanced systems lab of a large semiconductors company. His hobbies include electronics and amateur rocketry.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2006/195.

SOURCE

M16C/62P Microcontroller
Renesas Technology Corp.
www.renesas.com

Add SD in a FLASH !
with our
Flash Card Library

- ☑ Easily exchange data between your product and a PC
- ☑ Create, Append and Delete PC compatible files
- ☑ Add GIGABYTEs of onboard or removable storage to your product
- ☑ Compatible with over 20 card styles SD, MMC, MicroSD, TransFlash, …
- ☑ No OS, RTOS or interrupts are needed
- ☑ Works with low cost Microcontrollers
- ☑ Support for FAT16 and FAT32 file systems
- ☑ Simple API calls with full C sources

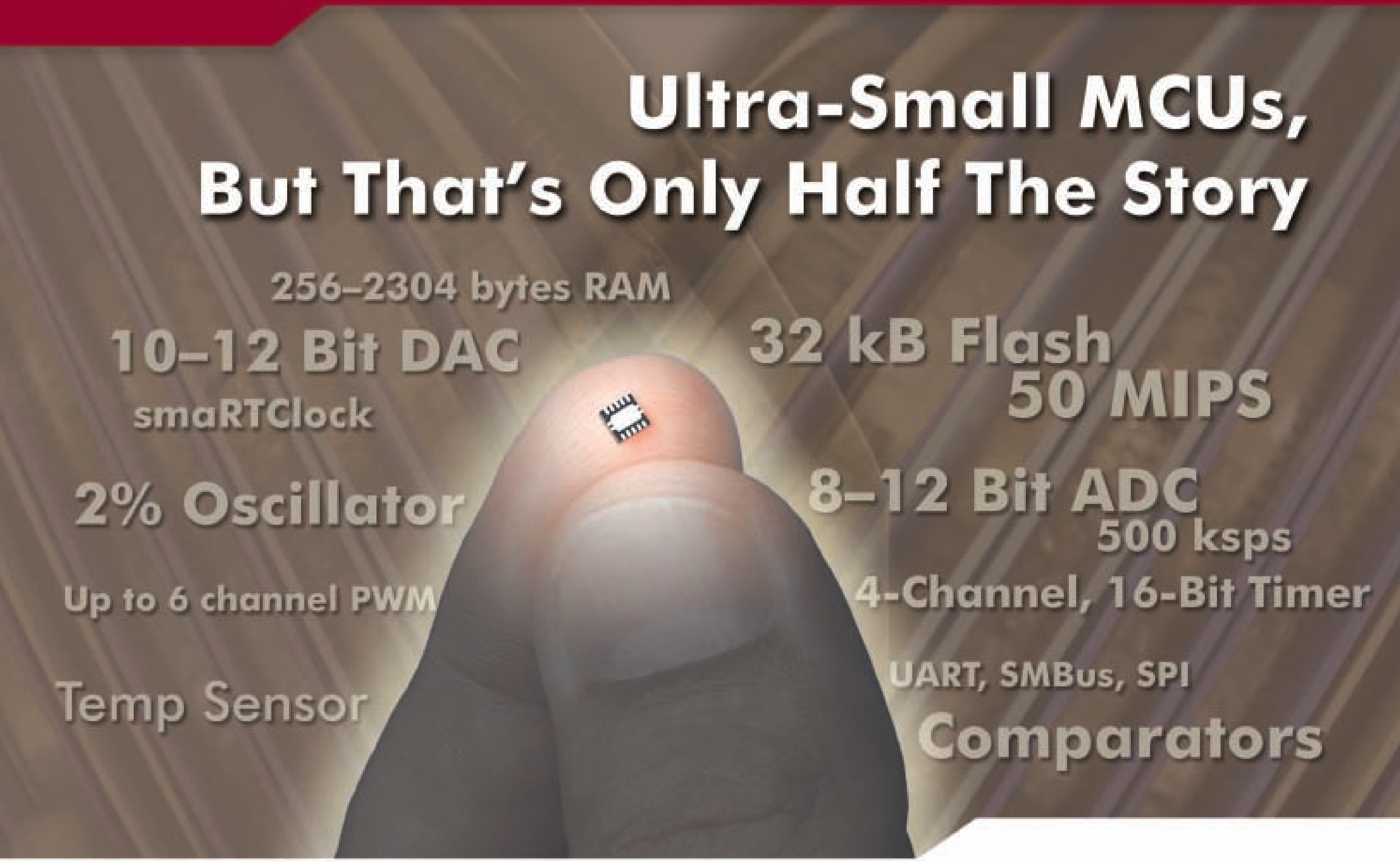
To get you up and running quickly, Development Boards and Adaptors are available for a variety of Micros.

Get More Details at www.ecs87.com/sfclib

EFFICIENT COMPUTER SYSTEMS, LLC
5 Emerson Drive Center Barnstead, NH 03225 • 603-776-3151
Offering MEGA solutions for the embedded MICRO world.

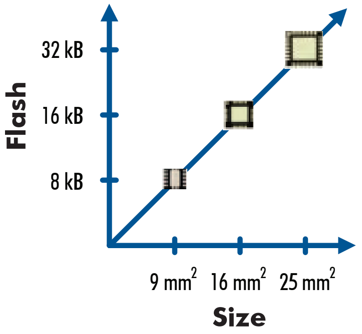
Introductory Prices starting at $139.95

Ultra-Small MCUs, But That's Only Half The Story



Fully Loaded, Small Form Factor MCUs

Silicon Laboratories' mixed-signal MCUs combine a high-speed 8051 CPU, Flash memory and best-in-class analog peripherals in ultra-small packages allowing designers to reduce component count while improving system performance. These highly-integrated, feature-rich devices are optimized for space-constrained, low-power, cost-sensitive applications.



Low-Cost, Professional Tools

Complete development kits (including in-system debug and integrated development environment) make development quick and easy. Low-cost evaluation kits and reference designs simplify evaluation and accelerate time-to-market.

Highest Functional Density

- 8–12 Bit ADC
- 2–32 kB Flash Memory
- 25–50 MIPS 8051 CPU

Product details: www.silabs.com/SmallMCU

MCUs | TIMING | POWER | BROADCAST | WIRELINE | WIRELESS

SILICON LABORATORIES
www.silabs.com

by Dejan Durdenic

Drive a Color STN Display

Why add push buttons to an application when you can easily design in a touchscreen? Check out Dejan's QVGA eight-color LCD controller. The C8051F120-based system features touchscreen support.

Recent developments in semiconductor technology have allowed old dogs to learn new tricks. Take the fast Silicon Laboratories C8051F120 microcontroller, which Tom Cantrell first described back in February 2004 ("'51 Flavors," *Circuit Cellar* 163). It can run at an amazing 100 MIPS, and it's packed with peripherals such as two ADCs (12 and 8 bits), a 12-bit DAC, and a programmable counter array (PCA), just to name a few. Those features, combined with the microcontroller's high speed, enable you to work on applications that were previously reserved for more powerful devices.

One such application involves driving a color QVGA super-twisted nematic (STN) display via port pins. In this article, I'll describe my design (see Photo 1).

STN DISPLAY

An LCD controller is typically used for generating timing signals and copying display data from display memory to an LCD panel. An STN color panel has a simple interface: 8 data bits, a data clock line, two sync inputs (line and frame), and a global display enable input (that disables the panel during initialization). That's pretty much it. Obviously, one and a half CPU I/O ports can generate all of the necessary signals. But what is the required timing and can the CPU meet it?

For this project, I used an Apollo Display Technologies F-51900NCU-FW-ACN-V365 5.7″ QVGA color STN display. It's a modified Optrex STN LCD module with a touchscreen that has a white LED backlight instead of a more common cold cathode fluorescent lamp (CCFL) tube. Rather than a DC/AC inverter for a CCFL backlight, an external current-limiting resistor drives the LED backlights.

Now let's focus on the timing. In the STN display, each pixel consists of three color pixels: red, blue, and green. By turning them on and off, you can create eight different color combinations. (Actually, there are six possible combinations. Two are black and white.) To increase color resolution, you can apply PWM to each of the color pixels. The result will be 256 or even 4,096 colors on some panels. However, I ruled out PWM because it couldn't be done in software at the required speed. So, I stayed with eight colors, which is enough for simple user interfaces.

Each row has 320 pixels (or 960 bits because each pixel has three color bits). The on-board display drivers drive one row of pixels at a time. The time to display one row is calculated from the frame frequency multiplied

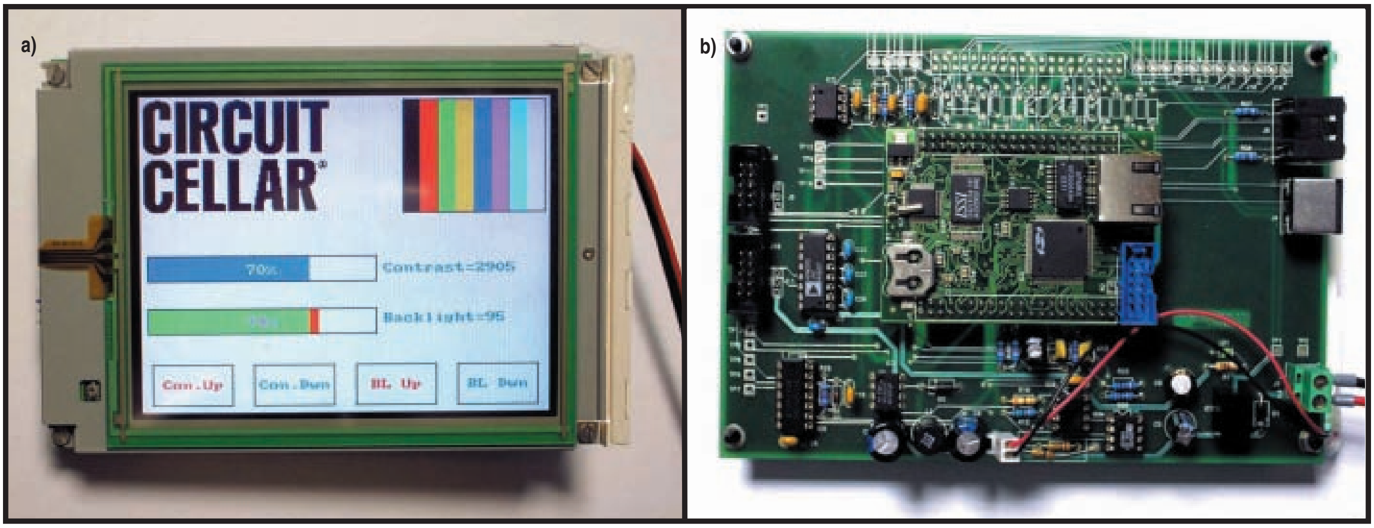


Photo 1a—*What do you think of my demo application? The GUI has numerous features (e.g., bitmap image, progress bar, and touch buttons).* **b—***The entire interface is mounted on the back of the LCD panel to form a compact solution.*

SCHEMATIC CAPTURE PROSPICE EMBEDDED SIMULATION PCB DESIGN

ISIS SCHEMATIC CAPTURE

A powerful capture package tailored for todays engineer and designed to allow rapid entry of complex schematics for simulation and PCB Layout.

PROSPICE MIXED MODE SIMULATOR

A customised implementation of the industry standard Berkeley SPICE 3F5 engine with extensive optimisations and enhancements for true mixed mode simulation and circuit animation.

VSM VIRTUAL SYSTEM MODELLING

The worlds first and best schematic based microcontroller co-simulation software. Proteus VSM allows you to simulate the interaction between software running on a microcontroller and any analog or digital electronics connected to it. This streamlines the project lifecycle and obviates the need for expensive hardware analysis tools.

ARES PCB DESIGN

A modern and professional layout package which seamlessly integrates with the ISIS capture software. Features such as autoplacement and autorouting, interactive DRC and an intuitive interface all serve to maximise productivity and reduce time to market.

LABCENTER ELECTRONICS LTD.

A technology pioneer in the EDA industry since 1988.
Technical support direct from the program authors.
Flexible packages and pricing tailored to customer requirements.

CONTACT US NOW

to discuss requirements or
request a **FREE** evaluation copy.

R4 Systems Inc.
1100 Gorham St
Suite 11B-332.
Newmarket Ont.
Canada L3Y 8Y8

Authorized Distributor of Proteus
Other products include:
Proton + Development Suite
Discovering PIC Books and Kits
PIC Educational Courses

Toll Free 866.499.8184
Tel: 905.898.0665
Fax: 905.898.0683
email: info@r4systems.com
www.r4systems.com

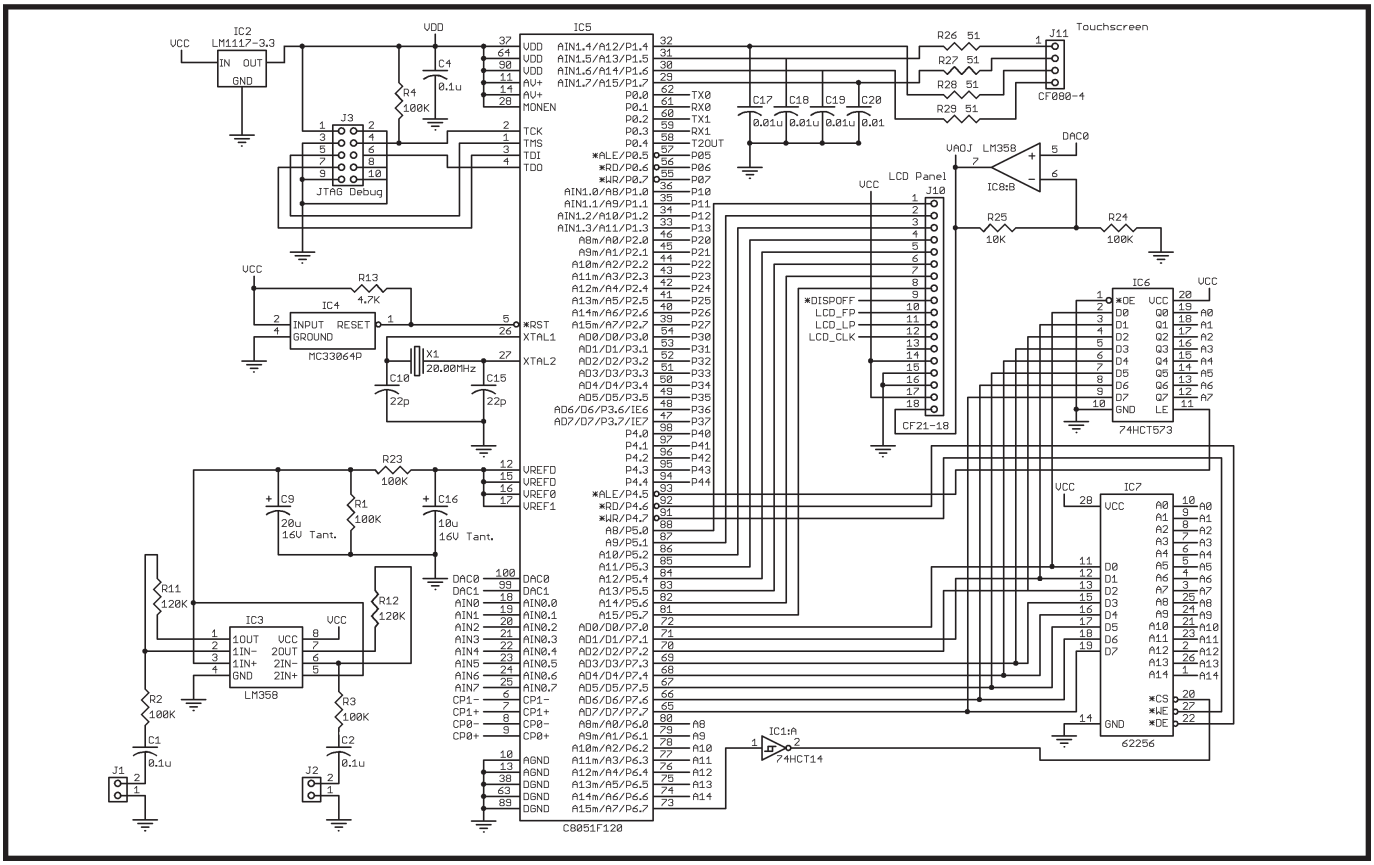


Figure 1—*The external memory interface makes the most of the circuit. The rest is really simple.*


Going Wireless is Easy!
Mesh Networking
ZigBee™
Bluetooth
with Dip-Switch options
Multi-Channel
900 MHz Wireless
RF Modules
LEMOS
INTERNATIONAL
Call Toll Free 1-866-345-3667 or email at sales@lemosint.com
www.lemosint.com

by the number of display rows (240 in the case of a QVGA display). To keep the display from flickering, the frame frequency should be 50 Hz or more. To be on a safe side, I chose 60 Hz.

The timing calculation is simple: 60 × 240. That's a 14,400-Hz line sync frequency, or 69.44 µs. That means the CPU has to move 120 bytes of data from its memory to the LCD panel every 69.44 µs. The data clock frequency can be up to 25 MHz. That means you can write 120 bytes of data in only 4.8 µs. The interface can obviously do it in the required time. But can the CPU do it? I wrote a short software routine to check. It simply takes a byte from the external memory, outputs it to a CPU port, and toggles the data clock line. After the process repeats 120 times, a check is performed to see if a frame pulse is needed. That's it.

The results are promising. Without any optimization, it takes about 50 MIPS for the display refresh at 100 MHz, leaving an ample 50 MIPS for the rest of the application.

In order to try the design with the

Listing 1—*Take a look at the Timer2 interrupt service routine. T2 is used as the LCD controller LP pulse generator at 14,400 Hz. The function is compiled in a small memory model to boost performance.*

```
#define X_PIXELS 320L
#define Y_PIXELS 240L

BYTE LCD_BUFFER[X_PIXELS*Y_PIXELS*3];

        Sfr   LCD_DATA = P5;
        sbit  LCDCLK = P4^0;
        sbit  FP = P4^1;

        data  BYTE lines;             // Refresh line counter
        data  BYTE xdata *image;      // pointer to image buffer

void lcd_line_interrupt(void) small interrupt 5 using 1
{
        data BYTE d;
        data BYTE chars;
        data BYTE xdata * tmp;         // Copy of image pointer
                                      // compiler will hold it in
                                      // DPTR register, increasing
                                      // performance

        TF2 = 0;                      // clear timer2 interrupt flag
        TMR2CF = 0x06;                // set T2 bit again
        SFRPAGE = CONFIG_PAGE;
        if (lines==0)                  // start of frame reached
          {
         FP = 0;
         lines=Y_PIXELS;              // reload line counter
         image=LCD_BUFFER;            // and frame pointer
         SFRPAGE = ADC2_PAGE;
         AD2BUSY=1;                   // start AD conversion
         SFRPAGE = CONFIG_PAGE;       // for touch panel
        }
        else
        {
         FP = 1;
        }
        lines–;                       // Decrement line counter
        tmp = image;                  // Fetch current display address
        d=*tmp;                       // Fetch first data byte
        chars=(X_PIXELS*3)/8;         // Set nr.of bytes per line
        do
        {
         LCD_DATA=d;                  // Set data for LCD panel
         tmp++;                       // Increment pointer
         LCDCLK = 0;                  // Assert data clock
         d=*tmp;                      // Fetch next data byte
                                      // Stretching clock pulse
         LCDCLK = 1;                  // Deassert data clock
        }
        while(–chars);                // Repeat for whole line
        image=tmp;                    // Store memory pointer
                                      // for the next interrupt
}
```

+++ NO ROYALTIES +++

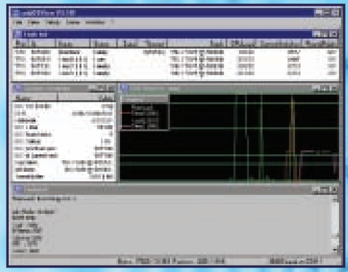
embeddedsoftware
solutions

Eval versions available

embOS® (RTOS)

+++ 8/16/32 bits +++

Preemptive multitasking
Zero interrupt latency
Easy to use start project included
Profiling support included
Object/source code available



emWin® (GUI)

+++ 8/16/32 bits +++

ANSI "C" source code, no C++ required
Supports b/w, grayscale and color
2D graphic library included
Variety of fonts included
PC simulation included
Window Manager/Widgets (opt)

emFile (File system)

+++ 8/16/32 bits +++

ANSI "C" source code
MS-DOS/MS-Windows compatible
FAT12, FAT16 and FAT32 support
Multiple media support
Non FAT file system available

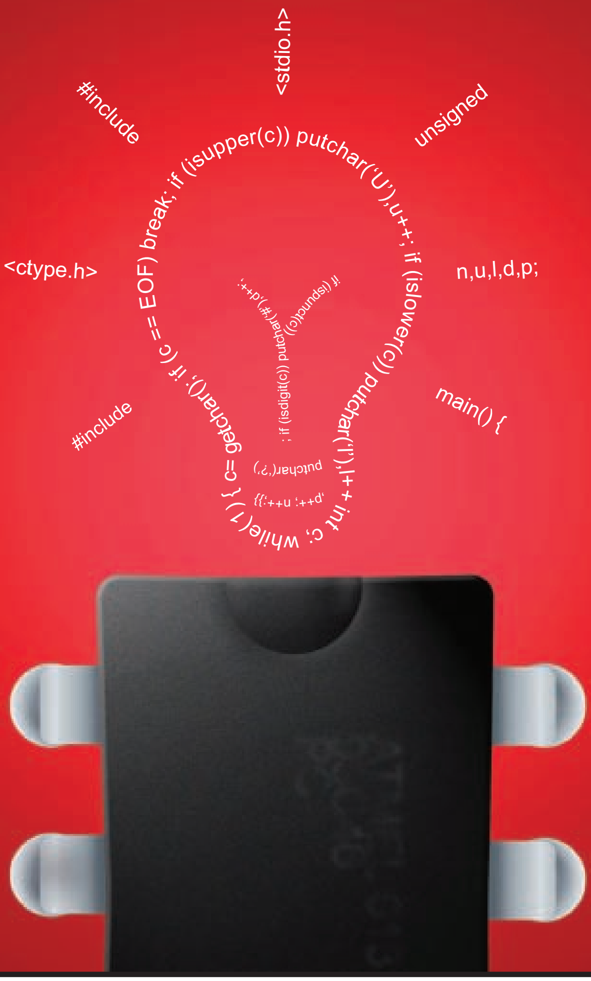
For ARM Chips:
JTAG debug solution
with flash programming

SEGGER

phone: 978-874-0299
www.segger.com

We make chips think

HI-TECH Software delivers the industry's most reliable embedded software development tools and compilers for over 12 different 8-bit, 16-bit, 32-bit, and DSP chip architectures!



As one of the top five compiler vendors with the largest variety of supported chip architectures, HI-TECH Software's product range is renowned for delivering cutting-edge technology and robust results for development teams worldwide.

With over two decades of industry experience, our long-term relationships with leading chip manufacturers ensure that our products are tightly attuned to new technological releases.

Whichever processor family you are targeting, HI-TECH Software's C compilers can help you write better code and bring it to market faster.

- HI-TECH PICC™ Enterprise Edition
- HI-TECH PICC™
- HI-TECH PICC-18™
- HI-TECH dsPICC™
- HI-TECH C® for ARM®
- HI-TECH C® for 8051
- HI-TECH C® for MSP430
- HI-TECH C® for HOLTEK MCU
- HI-TECH C® for ARClite™
- HI-TECH C® for XA
- HI-TECH C® for Z80
- HI-TECH C® for H8/300
- HI-TECH C® for 68HC11

To see how our compilers can improve your productivity, download a demo now at *www.htsoft.com/downloads/demos.php*.

HI-TECH C® is a registered trademark of HI-TECH Software. HI-TECH PICC™, HI-TECH PICC-18™ and HI-TECH dsPICC™ are licensed exclusively to HI-TECH Software by Microchip Technology Inc.
All other trademarks and registered trademarks are the property of their respective owners.

HI-TECH Software LLC 6600 Silacci Way Gilroy, CA 95020 USA
Ph: 800 735 5715   Web: http://www.htsoft.com/

HI-TECH
SOFTWARE

real hardware, I needed a CPU with at least 32 KB of external RAM (120 × 240 bytes for the display memory, or 28,800 bytes). To build the prototype, I used a C8051F120-based Dilogic CYF120 core module because it had the required memory onboard (see Figure 1, p. 32). The only thing I needed was a 5-V power supply. The rest of the interface was simple. I used one I/O port for eight data lines. I chose P5 because it doesn't share any peripheral functions and four control lines (CP, LP, FP, and *DISPOFF).

One of the coolest things about the C8051F120 is its JTAG debug interface. Combined with the free IDE, it provides a powerful debugging platform. This enables code downloads to internal flash memory, single stepping, and breakpoints. When the CPU stops, it stops completely (including all of the timers). As a result, it can't generate the timing signals for the LCD. As you know, LCDs require constant refreshing in order to prevent the dangerous DC bias that can lead to permanent segment damage. A watchdog is required. I used an FP signal to discharge capacitor C1 through diode D1. If the FP signal is running, the capacitor discharges quickly through the diode. If it stops (e.g., due to a JTAG breakpoint), the capacitor will charge slowly through resistor R1. After approximately 20 ms, the display is disabled via the *DISPOFF signal.

I needed to add external memory too. Although the C8051F120 has 8 KB of internal XDATA SRAM, it obviously wasn't enough. The image buffer needs 28,800 bytes. As a result, I added 32 KB externally. Although the memory interface allows for both the multiplexed and nonmultiplexed approaches, I opted for the former to save one I/O port, which otherwise would have been used for the lower eight address lines.

I routed the A15 line via the inverting gate to the RAM chip select input because the internal SRAM is located at 0x0000–0x1FFF. If the external RAM is connected in the same address range, all of the memory accesses in that range can go to either

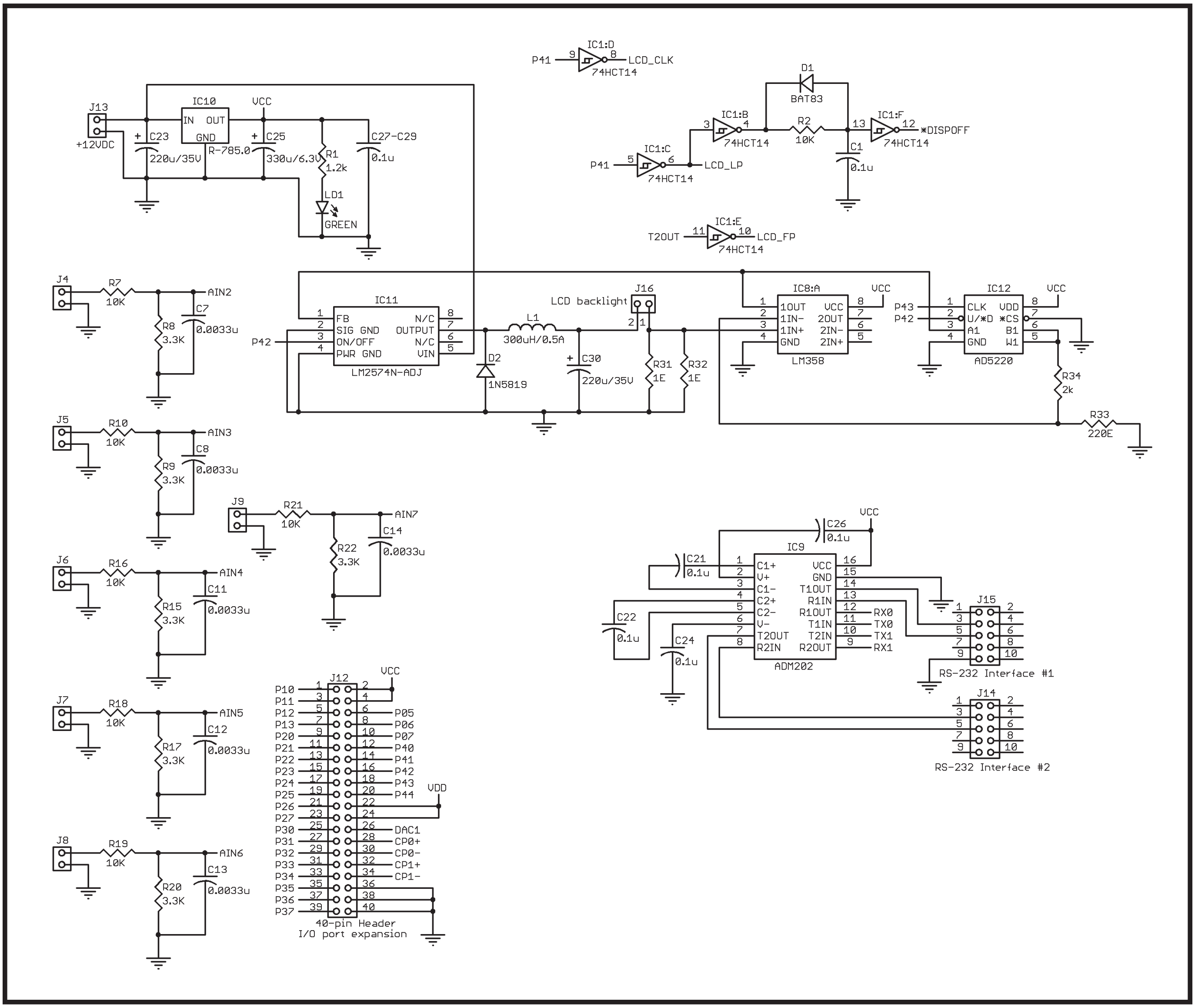


Figure 2—*The LM2574 is wired as constant-current source. The current is varied via an AD5220.*

the internal or external RAM depending on the internal configuration of the register setting. The internal RAM is much faster (no wait states are needed at 100 MHz), so it's a pity not to use it. On the other side, I needed all 32 KB of the additional memory. So, I simply located the external memory in the 0x8000—0xFFFF range. This way I have both at the same time. The gap between two areas is easy to handle in compiler settings. Although a 70-ns access time seems long for a 100-MHz CPU, the actual speed penalty isn't too high.

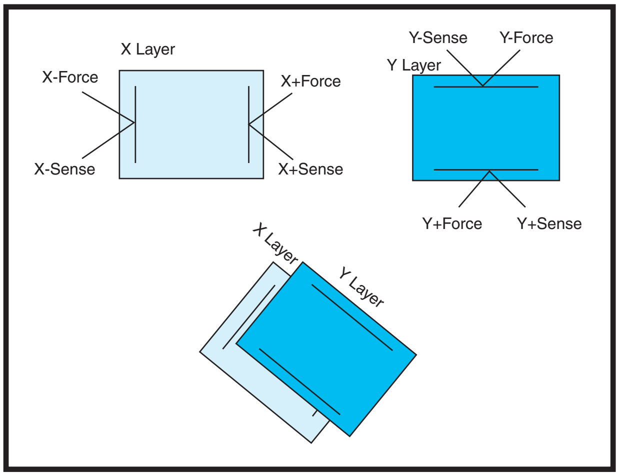


Figure 3—*Take a close look at this simplified representation of the touch-screen foils.*

Only the image buffer is located in the external memory. All of the other XDATA variables use internal SRAM. With the current software driver, I found that it takes 28 µs to update one display line. With the 15-ns memory, it takes about 23 µs. The line refresh repeats every 69 µs (i.e., 40 versus 33 MIPS) if the CPU runs with a 100-MHz clock. Operating external memory in nonmultiplexed mode speeds up the refresh for another few MIPS, so the maximum that can be pulled out is about 28 MIPS for the LCD refresh and 72 MIPS for the application.

UNDER CONTROL

With so many resources on hand, I couldn't resist adding some nice features like a software-controlled backlight and contrast control. The LCD panel has a white LED backlight that requires a constant current source. The current can reach 240 mA at 6.5 V. To reduce dissipation, I used a switch-mode regulator. I often use a National Semiconductor LM2574-ADJ regulator because it can supply 0.5 A at a wide range of output voltages (see Figure 2, p. 35). The only problem is that it's a voltage source. So, I modified the output circuit.

Backlight LEDs are connected in series with the current sense resistor. The voltage at the resistor is connected to the regulator's feedback input. To keep the dissipation low, I used two 1-Ω resistors in parallel. (I could have used a 0.5-Ω resistsor if I had one.) At 240 mA, the voltage drop is only 120 mV. Obviously, it wasn't enough. The IC required a 1.2-V reference voltage. So, I inserted an op-amp with variable gain. Gain can be varied from 10 to 55 by means of an Analog Devices AD5220 digital potentiometer. In terms of output current, it's approximately 44 to 240 mA.

The AD5220's control signals are connected to the C8051F120, so I can control the backlight intensity. I use the LM2574-ADJ's OFF input to turn off the backlight. To adjust the contrast, I use one of two 12-bit DACs on the C8051F120. Another half of IC4 boosts the 0- to 2.4-V DAC output to 0 to 2.64 V, which is enough for the display. (The recommended voltage for adjustment is 2 V.)

I can also control the contrast with software. Because the board was intended to serve as a proof-of-concept, I added a few things, just to play it safe. The C8051F120 has two asynchronous serial ports that I routed to the ADM202 RS-232 level shifter. By doing so, I could connect additional devices like a GPS receiver. A 12-bit ADC was tempting too, so there's another op-amp serving as the input buffer for the two input channels. Both are AC coupled with 0.5-V_{REF} DC offset, so I can measure AC signals up to 2 V_{pp}. In the future, I will try to turn the module into an FFT analyzer. The ADC's 100-kHz sample rate will provide a nice 50-kHz bandwidth for some audio devices. The remaining six channels are simply connected through dividers. The result is a 0- to 10-V input range.

The C8051F120 has a lot of I/O pins and ports. All of the remaining and unused pins are connected to the 40-pin header. I might need them in the future.

SOFT SIDE

Now let's focus on how the software works. Timer2 is used as a timebase that's programmed to interrupt the C8051F120 at 14,400 Hz. The frequency is calculated by multiplying the recommended 60-Hz frame rate with 240 display lines. On each Timer2 overflow, one row of display data (120 bytes) is transferred from the C8051F120's memory to the display and one LP pulse is generated. A counter keeps track of the number of the transferred lines. Every two hundred fortieth time, the FP pulse is generated and the process repeats.

Another detail worth mentioning is that the timing of the LP and FP puls-

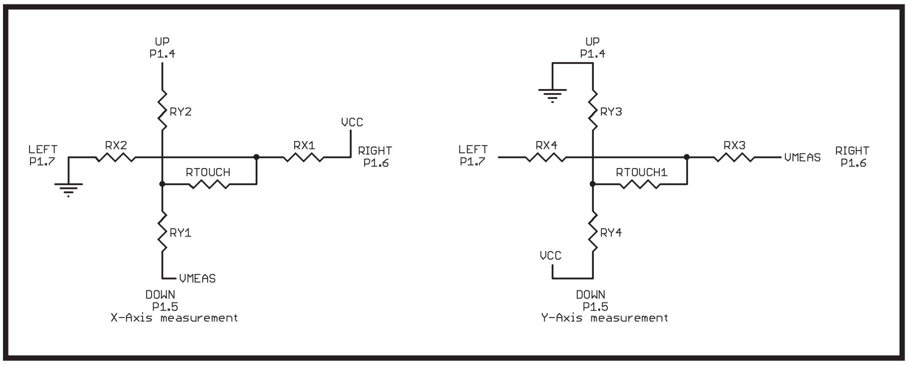


Figure 4—*Here you see the touchscreen in x- and y-axis measurement modes.*

Listing 2—*This is the touchscreen A/D end-of-conversion interrupt service routine. The A/D conversion is triggered after each LCD frame (60 Hz).*

```
#define    TOUCH_VALUE    240
  BYTE smTouchController;
  BYTE raw_X,raw_Y;      // Raw touch coordinates
  Bit Touched            // set by this routine, cleared when main
                         // code accepts new touch readings

void touch_interrupt(void) small interrupt 18 using 1
{
  AD2INT = 0;            // Clear conversion complete flag

  switch (smTouchController)
  {
    case 0:
    if(ADC2<TOUCH_VALUE) // Check if screen is touched
    {
      if (Touched==0)    // Is the previous result accepted?
      {
           AMX2SL = 4;   // Switch to x-axis measurement
           smTouchController=1;
           SFRPAGE = CONFIG_PAGE;
           P1=0x70;
           P1MDOUT=0xC0;
           P1MDIN=0xCF;
      }
    }
    else
    {
      if (Touched==0)    // Screen not touched
      {
           raw_Y=0;      // Clear conversion results
           raw_X=0;
           Touched=1;    // and set flag for main prog.
        }
    }
    break;

  case 1:
    raw_X=ADC2;
    if (ADC2<TOUCH_VALUE)
    {
           smTouchController++;
           AMX2SL = 6;   // Select UP electrode
           SFRPAGE = CONFIG_PAGE;
           P1=0xE0;      // P1.7,P1.6 and P1.4 -> 1
           P1MDOUT=0x30; // P1.4, P1.5 push-pull
           P1MDIN=0x3F;  // P1.6 i P1.7 are analog inputs
    }
    break;

  case 2:
    raw_Y=ADC2;
    if (ADC2<TOUCH_VALUE)
    {
        Touched = 1;
    }
    AMX2SL = 5;          // Select Down electrode
    SFRPAGE   = CONFIG_PAGE;
    P1MDOUT=0x00;        // Reconfigure I/O port for
    P1MDIN=0xFF;         // touch detection
    P1=0x30;
    smTouchController=0;
    break;
  }
}
```

es must be essentially jitter-free. Otherwise, there might be visible artifacts on the display. If you set the Timer2 interrupt's priority level to the highest level, you can have a clear and stable display. However, its rate of 14,400 Hz can cause problems when other interrupts are used. For example, the UART interface has only one character buffer. Thus, at high data rates, it must be serviced quickly to avoid data loss. And that's not to mention timer-controlled A/D and D/A conversions. Raising Timer2's priority level can cause significant latency on interrupt response times. On the other hand, if the priority is low, jitter can occur on LCD control signals.

The solution is simple. On the C8051F120 microcontroller, Timer2 has a mode where a port pin will toggle on timer overflow (independent of the interrupt response time). So, Timer2 is programmed to generate an interrupt AND to toggle the PORT pin during overflow. That pin is actually used as an LP signal. Now, when Timer2 overflows, an LP signal is immediately generated and an interrupt is generated. As long as it is serviced within a half period of 14,400 Hz (remember that it takes a long time to process the Timer2 interrupt), everything will run smoothly and the Timer2 interrupt priority level can be set to the lowest level. Note that Listing 1 (p. 33) is the complete timer interrupt service (frame refresh) routine.

FINAL TOUCH

Resistive touchscreens are normally used for cost-sensitive projects. Their simple designs and modest hardware and software requirements make them appealing choices. The LCD panel I used features a four-wire resistive touchscreen.

A four-wire touchscreen is made of two thin sheets of glass or plastic. The interior surfaces are coated with a thin layer of conductive material. Small glass beads that allow the upper sheet to bend slightly when you touch it separate the two conductive surfaces. It's connected to the lower sheet to make a contact.

A four-wire touchscreen has four

connected electrodes (two on each sheet). On one sheet, there's an electrode along the x-axis on the top of the foil. Another is along the x-axis on the bottom of the foil. The electrodes are along the y-axis on the second sheet (on the left and right sides), as you can see in Figure 3 (p. 36).

If you connect a voltage source to the x-axis electrodes (e.g., connect 5 V to the left electrode and GND to the right electrode), there will be a voltage drop on the resistive coating that's proportional to the distance from the right electrode. When you touch the screen, two foils connect. If you then measure the voltage on one of y-axis electrodes, it will be proportional to the x-axis position of the contact. If you reverse the connections in a way that the voltage is applied to the y-axis electrodes, the voltage on the x-axis electrodes will be proportional to the y-axis position of the touch. Therefore, you have a way to determine the position of the touched point anywhere on the surface. Four-wire touchscreen signals are usually named Left, Right, Up, and Down to indicate the appropriate electrode position. Figure 4 (p. 36) is the four-wire resistive touchscreen in x-axis and y-axis measurement modes.

You need an ADC in order to use a touchscreen. You can use a special chip for this purpose, but the C8051F120 can do it too. The C8051F120 has two ADCs. One is 12 bits with eight dedicated analog inputs. The other is 8 bits and shares pins with I/O port P1. The natural choice is the 8-bit ADC. First, 8 bits is enough for such a small screen. In addition, you can use the same pins for measurement and driving the touchscreen electrodes.

The pin function selection (A/D input or I/O port) is controlled completely by the software, so the only external components are simple RC filters used for filtering out noise pickup. The touchscreen electrodes cover the entire display area. As the image is constantly refreshed, a large amount of noise is capacitively coupled.

The software driver is simple (see Listing 2, p. 37). It waits until you touch the screen and then measures the voltages proportional to the x- and y-axis touch position. To detect a touch, you need a slightly different approach. If you pull one layer up with a weak pull-up resistor and then pull another layer with a strong pull-down resistor, the voltage on the pulled up layer will be inversely proportional to the force applied to the touch point. Because the layer resistance is much smaller (a few hundred ohms) in comparison to the pull-up resistor value, any touch causes a considerable drop in the sensed voltage (see Figure 5).

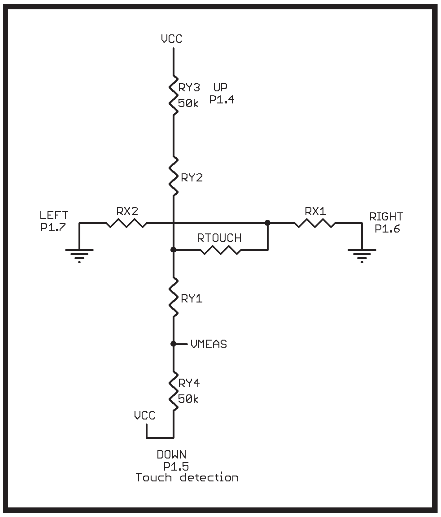


Figure 5—*Have a look at the touchscreen in the touch-detection mode.*

The I/O port has a particular configuration for touch detection. The P1.4 and P1.5 lines (connected to y-layer electrodes) are both set high. But because the push-pull output driver isn't enabled, only weak pull-up resistors are active. The x-layer electrodes (lines P1.6 and P1.7) are set to a logic low level (a strong one) because the lower transistors are always active.

The A/D multiplexer selects the P1.5 line as the A/D input. When the screen isn't touched, it measures almost all of the full supply voltage. When the screen is touched, the voltage drops because the x-layer foil is grounded through the P1.6 and P1.7 pins. As a result, the voltage is considerably lower (almost 0 V). Now, when a touch is detected, the I/O port configuration is changed. P1.4 and P1.5 are set to logic high and pull-ups are disabled. P1.6 is set high, and P1.7 is set low. (Both pins are in push-pull mode.) The x-axis left and right electrodes are driven low and high, respectively. The A/D input is routed to pin P1.5, allowing the measurement of the voltage proportional to the x-axis touch position.

Next, P1.4 and P1.5 are switched to push-pull mode and set low and high, respectively. P1.6 and P1.7 are now configured as inputs without pull-ups. P1.6 is used as A/D input. As the up and down electrodes are driven, you can measure y-axis touch position.

The C8051F120's ability to change I/O pin configuration on the fly enables this extremely simple interface. You can configure the I/O pin to be a strong push-pull output (capable of driving 20 mA), an open-drain output with a pull-up resistor, or an analog input (for A/D conversion). The low-level touchscreen driver is implemented as an A/D end-of-conversion interrupt service that executes a simple state machine once per frame pulse (at 60 Hz). A complete scan takes three frame cycles. The result is a 20-Hz touch panel scan rate.

The low-level driver sets two variables to the value proportional to the voltages on the touchscreen's x- and y-axes. What you need are actual screen coordinates in pixels. To calculate them, you first need to calibrate the touchscreen. There are several ways to do so, but a simple two-point calibration can do the job for this small design (it's only a 5.7″ display). The calibration routine draws two crosses in two opposite corners of the screen (at the known coordinates). When you press the middle of each cross, the calibration routine will calculate the required offset and gain settings to get the proper screen coordinates from raw touch values (see Photo 2).

To avoid 32-bit arithmetic, I sacrificed 1 bit of the x-axis resolution. As a result, the LSB of the x-axis position word is always set to zero. It has no practical influence on such a small display. Compare the tip of your finger with the screen's size. Insisting on the native 320 × 240 pixel resolution for such a touchscreen doesn't make too much sense.

KEEPING IT COMPACT

When the basic low-level drivers


Touchscreen calibration
Press cross in upper left corner

Photo 2—*Calibrating the touchscreen is simple.*

were finished, I needed a graphic library. I wanted to be able to draw simple objects (e.g., lines and rectangles) and add text. I also wanted user-defined buttons for touchscreen interfacing.

The code for this project code is extremely compact. The demonstration application compiles in less than 10 KB (including two fonts). The complete library with source code is posted on the *Circuit Cellar* FTP site.

I also added some cool features like a progress bar, scalable fonts, and bitmap image drawing. The only thing I don't have at this time is window management. It's in the works.

I will continue to develop the library. As I add new features, I will try to keep the code footprint as small as possible. After all, it's still an 8-bit microcontroller!

Dejan Durdenic (dejan@dilogic.hr) earned a B.S. in electrical engineering from Zagreb University in Croatia. He has been designing embedded systems for the past 25 years. Dejan owns Dilogic, a company that provides hardware and software consulting services.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2006/195.

RESOURCES

C. E. Vidales, "How to Calibrate Touch Screens," Embedded Systems Design, June 2002, www.embedded.com/story/OEG20020529S0046.

SOURCES

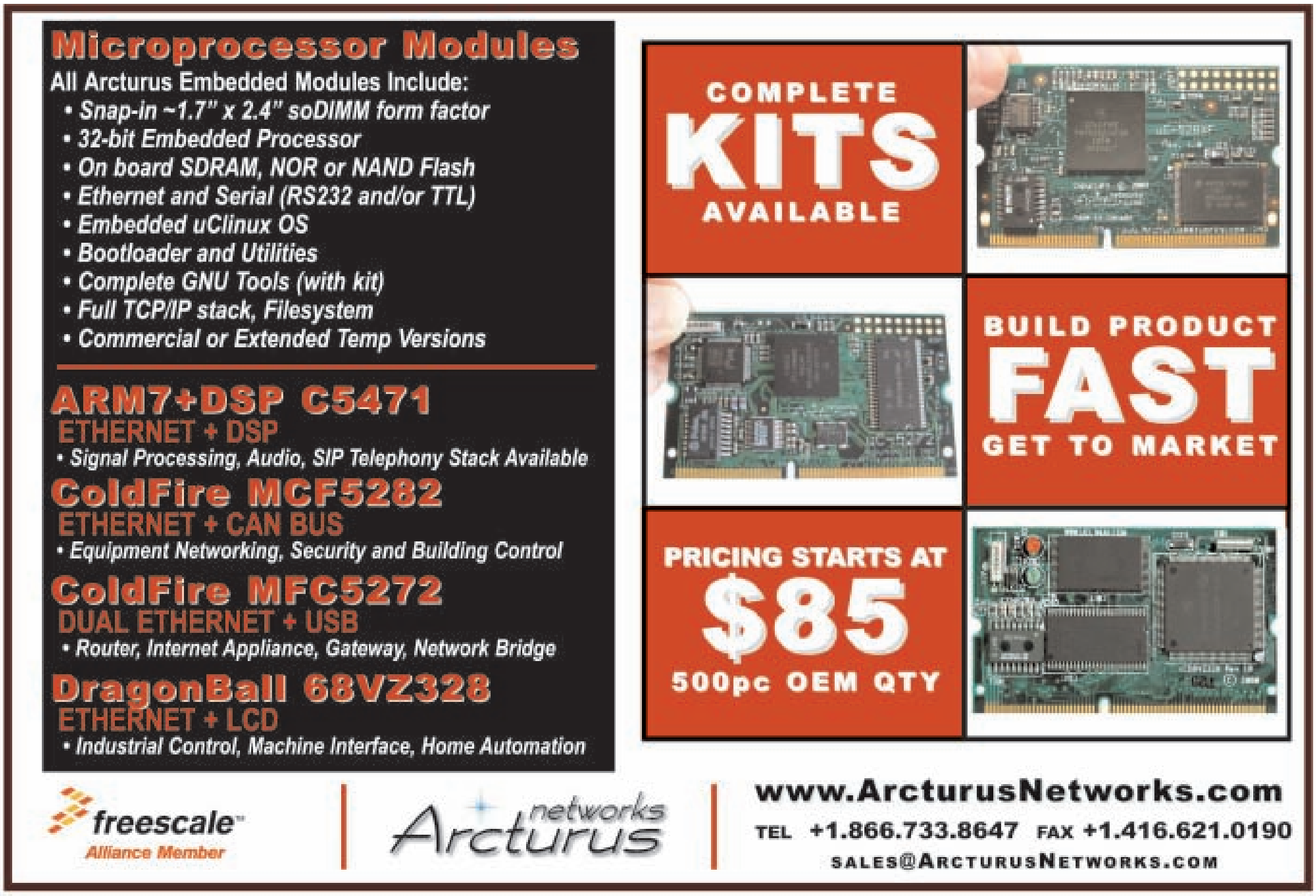
ADM202 Transceiver and AD5220 digital potentiometer
Analog Devices, Inc.
www.analog.com

F-51900NCU-FW-ACN-V365 STN Display
Apollo Display Technologies
www.apollodisplays.com

CYF120 Core module
Dilogic
www.dilogic.hr

LM2574-ADJ Regulator
National Semiconductor Corp.
www.national.com

C8051F120 Microcontroller
Silicon Laboratories, Inc.
www.silabs.com


Microprocessor Modules
All Arcturus Embedded Modules Include:
- Snap-in ~1.7" x 2.4" soDIMM form factor
- 32-bit Embedded Processor
- On board SDRAM, NOR or NAND Flash
- Ethernet and Serial (RS232 and/or TTL)
- Embedded uClinux OS
- Bootloader and Utilities
- Complete GNU Tools (with kit)
- Full TCP/IP stack, Filesystem
- Commercial or Extended Temp Versions

ARM7+DSP C5471
ETHERNET + DSP
- Signal Processing, Audio, SIP Telephony Stack Available

ColdFire MCF5282
ETHERNET + CAN BUS
- Equipment Networking, Security and Building Control

ColdFire MFC5272
DUAL ETHERNET + USB
- Router, Internet Appliance, Gateway, Network Bridge

DragonBall 68VZ328
ETHERNET + LCD
- Industrial Control, Machine Interface, Home Automation

COMPLETE KITS AVAILABLE
BUILD PRODUCT FAST GET TO MARKET
PRICING STARTS AT $85 500pc OEM QTY

freescale Alliance Member
Arcturus networks
www.ArcturusNetworks.com
TEL +1.866.733.8647 FAX +1.416.621.0190
SALES@ARCTURUSNETWORKS.COM

FEATURE ARTICLE by Peter Gibbs & Ramon Sargeant

LED Message Display

You don't need to spend thousands of dollars on a commercial display system. Peter and Ramon describe how to build an Internet-based LED display of virtually any size.

By the time the electronic scoreboard display Peter described in *Circuit Cellar* 184 was up and running, it was time to begin planning an improved version (P. Gibbs, "Large-Scale Electronic Display," 2005). The original system used electromechanical digits that operated on a reflective principal. External light was required so you could see the digits at night. There were other limitations too.

As you know, things that move eventually wear out. The flip digits on the display were no different. Someone had to intervene whenever a digit got stuck. The system was also noisy. It wasn't electronic noise, but the flipping and flopping digits made noise nonetheless! Another problem was that the digits showed only numeric data. As a result, names and messages couldn't be displayed for viewers. The system couldn't even generate crude graphics for advertising.

We built a new display to address all of these problems (see Photo 1). The new design is an emissive system, which means it features LEDs. We now have a noiseless display with alphanumeric and graphics capabilities.

We work at the University of the West Indies, so our system operates within a wireless Internet footprint. Using wireless technology allows us to place our operator at any distance from the system and still maintain control. Embedded systems at the microcontroller level support RS-232. With support chips, TCP/IP features can also be used. You can use a cable if wireless connectivity isn't available.

DISPLAY TECHNOLOGY

Large-screen displays are an important part of many local and international events. They provide useful information to viewers and generate a source of advertising revenue for the owners.

One of the captivating properties of such displays is their physical size. Although the screen sizes vary, most of these displays are larger than 1′. Another characteristic is that the entire display area can be manufactured as a single component (older systems) or a number of smaller panels that fit together (newer systems). Many older systems feature special incandescent light bulbs that work as the active display element. Modern large-screen displays use full-color (RGB) LEDs as the active display elements (pixels).

The major drawback to a large-screen display is its cost. Small large-screen displays can cost several thousands of dollars. Large state-of-the-art displays can cost millions. And that's not the only problem. Try obtaining the technical data for a large-screen display. It's akin to running into a brick wall. You can search the Internet all you want, but the systems' intricate designs are closely guarded secrets. They tend to be patented and protected.

Never fear. We're perfectly comfortable with revealing our method for designing and building a working system. As you'll see, our design is based on a collection of panels. You can now build a display of virtually any size.

LED-BASED DESIGN

A modular design offers advantages over a single system. Our display comprises a number of panels that work together as a single display. We can easily modify the display's size and configuration by adding, removing, or arranging panels as desired.

When we started this project, we were tempted to build a versatile system with full-color, real-time action, video replay, and various other bells and whistles. But reality quickly set in. We realized that such a system would require a lot of time and money, so we settled for a monochrome (red) system. The primary display element (pixel) was an ultra-bright LED.

Photo 1—*These two 1′-squared panels were constructed out of a sheet of double-sided PC copper board. We placed makeshift red filters in front to enhance the contrast.*

LEDs are a great choice for a system like this. First and foremost, they're long lasting. A typical LED will work for 100,000 h. LEDs are also useful because you can build a cluster to represent a single pixel. Resolution is proportional to pixel pitch. To achieve a satisfactory video resolution, we selected a pitch of 15 mm. We can improve the resolution if we reduce the pitch.

Internet-based devices are rapidly becoming an important part of our everyday lives. We're on the Internet

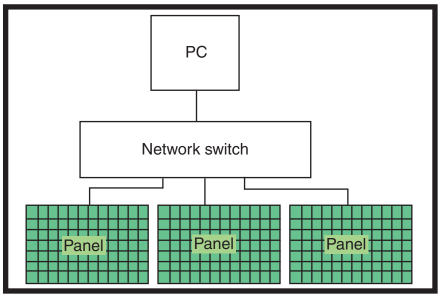


Figure 1—*The display system includes a computer, a network switch, and display nodes (panels). The pixel pitch determines the size of a panel.*

INTRODUCING THE NEW

GLK240128-25

The GLK240128-25 is a large graphic display that gives you the room to display many pictures and text all at the same time. Our graphic displays offer you high speeds and an easy to use command interface. This gives you the flexibility to quickly download high quality fonts in any size, style or language and create graphics using primitives such as bitmaps, pixels, lines, rectangles and bar graphs.

Get 5% off your next purchase online at Matrix Orbital. Simply log on to our site below and enter the code **MOCC0134** in the area provided at the checkout.

MATRIX ORBITAL

www.matrixorbital.com

BitScope

Pocket Analyzer

USB Oscilloscope & Logic Analyzer

8 Channel 40MS/s Logic Analyzer
✓ Capture digital signals down to 25nS with arbitrary trigger patterns.

3 Input 100MHz Analog DSO
✓ Classic Analog Scope using a standard x1/x10 BNC probe. Additional inputs on the POD for dual channel operation.

8 + 1 Mixed Signal Scope
✓ True MSO to capture an analog waveform time-synchronized with an 8 channel logic pattern triggered from any source.

Real-Time Spectrum Analyzer
✓ See the spectrum and waveform of analog signals simultaneously and in real-time

Waveform Generator
✓ Load up to 32K arbitrary waveform and replay via the onboard DAC (10MS/s) or a digital pattern from the POD (40MS/s)

The new generation Scope for the age of microelectronics.

Turn your PC or NoteBook into a powerful Scope and Logic Analyzer!

See inside your circuit in the analog and digital domains at the same time to make tracking down those elusive real-time bugs much easier.

Pocket Analyzer combines a high speed sample-synchronized storage scope and logic analyzer with a programmable waveform and logic pattern generator. Also included is an integrated real-time spectrum analyzer and powered "Smart POD" expansion interface so you've got all bases covered!

About the same size and weight as a Pocket PC, this USB powered BitScope needs no bulky accessories. It's the perfect low cost "go anywhere" test and debug solution.

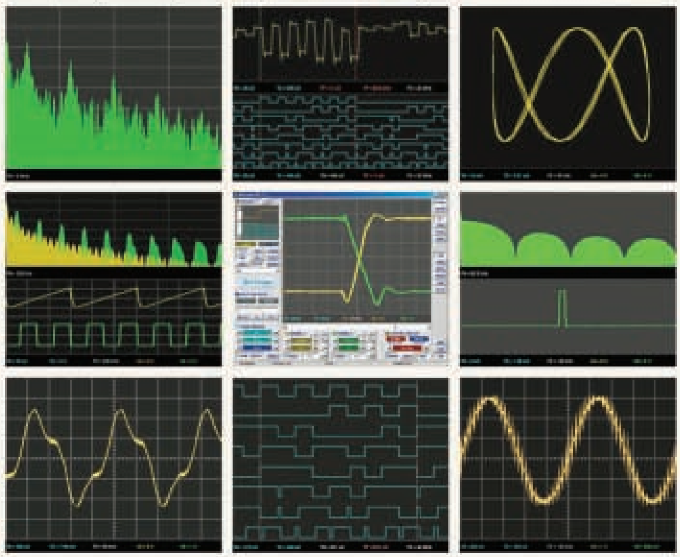
Standard 1M/20pF BNC Input
200uV-20V/div with x10 probe
S/W select AC/DC coupling
S/W select 50ohm termination
Arbitrary Waveform Generator

BitScope "Smart POD" Connector
8 logic channels, 2 analog channels
Dual channel capture from POD A/B
Async serial I/O for external control
Logic Pattern generator 32K 40MS/s

BUS Powered USB 2.0 Device
Single USB cable to your PC
Compressed data transmission
Simple ASCII control protocol
BitScope Scripting Language

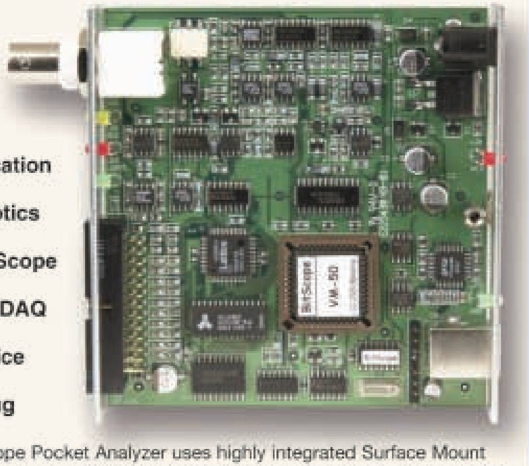
External/Passthru Power Supply
Auto senses an external supply - removes power load from USB for use with unpowered hubs.
Supplies up to 500mA via POD

BitScope and your PC provide an array of Virtual Instruments



BitScope DSO 1.2 software for Windows and Linux

- R&D
- Education
- Robotics
- Lab Scope
- Fast DAQ
- Service
- Debug



BitScope Pocket Analyzer uses highly integrated Surface Mount technology to provide functionality you would expect from scopes many times the size and price. Its programmable Virtual Machine architecture means new functionality can be added via software. For custom Data Acquisition, export directly to your spreadsheet.

www.bitscope.com

bandwagon. We coupled the speed and flexibility of a networked device with the reliability and simplicity of an embedded system to create this versatile, low-cost LED display board.

PC, SWITCH, & PANELS

As you can see in Figure 1 (p. 40), our display system consists of three main components: a PC, a network switch, and intelligent display nodes (panels). The power requirements for each panel depend on the type of LED you use. Our first panel measured 1′ × 1′ and included ultra-bright red LEDs. We chose to source and sink the columns and rows directly from the driver IC. A complete panel required approximately 800 mA at 5 VDC.

The two panels in Photo 1 were constructed out of a sheet of double-sided PC copper board. We also built a larger 3′ × 3′ panel, which uses an LED cluster to form a pixel with a pitch of 40 mm. Each cluster contains 15 surface-mount LEDs arranged as three parallel rows of five in series. Photo 2 shows the panel arrangement, which operates from 12 VDC. We inserted simple driver circuits between the IC and cluster for the larger panel. You can choose an adequate power supply for the system you build.

The majority of large display manufacturers use custom software. This is disadvantageous because it confines each application to a particular operating system. Incompatibility between PCs, Macs, and UNIX systems means that multiple versions of a single program are sometimes required. We don't have that problem because we use a web page for system control. We can use any browser-capable PC to control our display. This eliminates the need to maintain multiple versions of the software. It also means we can easily upgrade our control application.

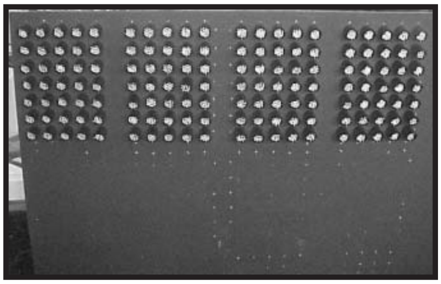


Photo 2—*This is a partially completed 3-ft² panel. Each pixel is a cluster of 15 red LEDs. It's easy to see this panel from far away.*

We built the entire system with off-the-shelf equipment (see Figures 2 and 3, p. 44). The panel control board is shown in Figure 4 (p. 45). Thus, we don't need to spend a lot of money on maintenance. When you start working on your display, remember that you can use something as simple as an eight-port, 10/100-Mbps network switch, which won't cost you more than $20 at your local electronics store. Your network switch's size will determine the number of panels your system can accommodate. For example, if we dedicate two ports for a control computer and DHCP server, six panels can be supported. If we use only a control computer, seven panels can be supported.

SYSTEM SPECIFICS

Figure 5 (p. 46) shows the main components of each display node and the flow of data from the source (the PC or another node) to the display panel. The first component is a MagJack connector, which is a special RJ-45 connector with integrated electronics (e.g., a transformer and resistors). The connector eliminates the need for external components and reduces the PCB footprint. It also provides the necessary signal conditioning and signal isolation for reliable communication.

As you can see, we also used a Realtek Semiconductor RTL8019AS networking IC. This is a full-duplex, 10-Mbps IC found on many old ISA network cards. It accepts raw bitstream data from the MagJack connector and stores it in an internal buffer. The 16-KB SRAM internal ring buffer is

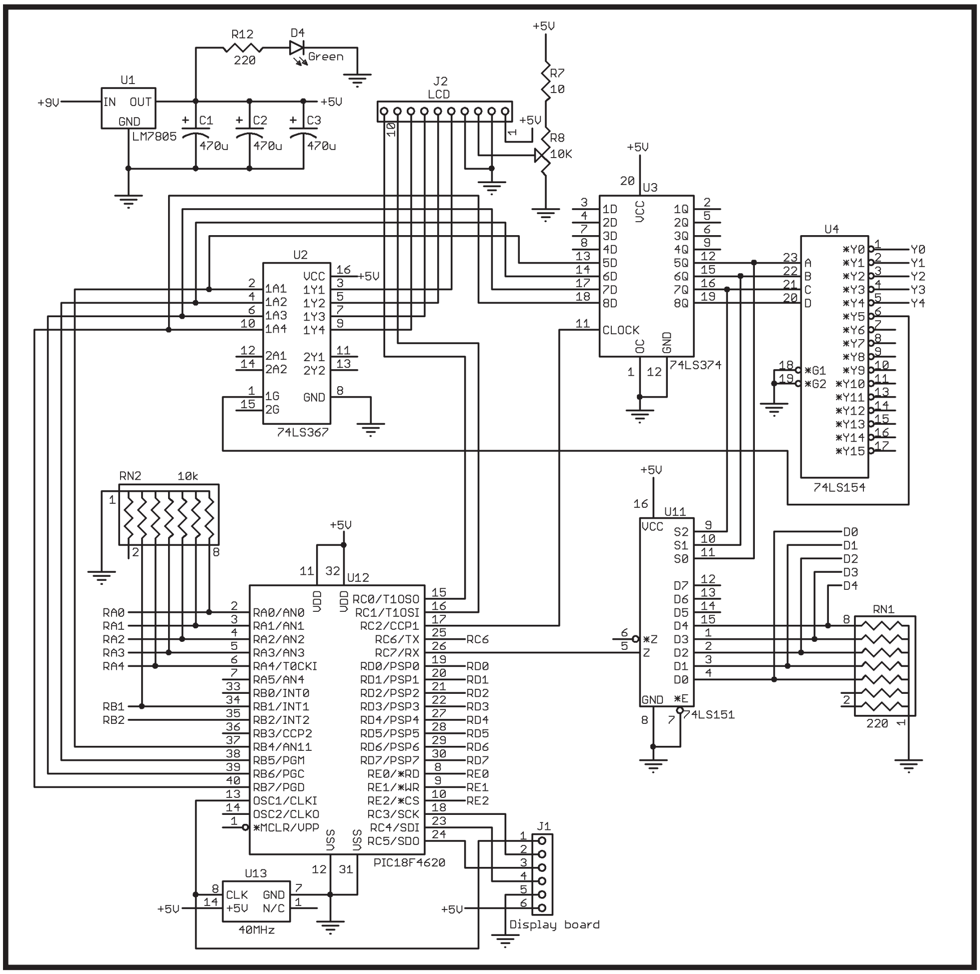


Figure 2—*The PIC18F4620 acts as the master microcontroller passing data to the display panel controller and handling network data from the network control board.*

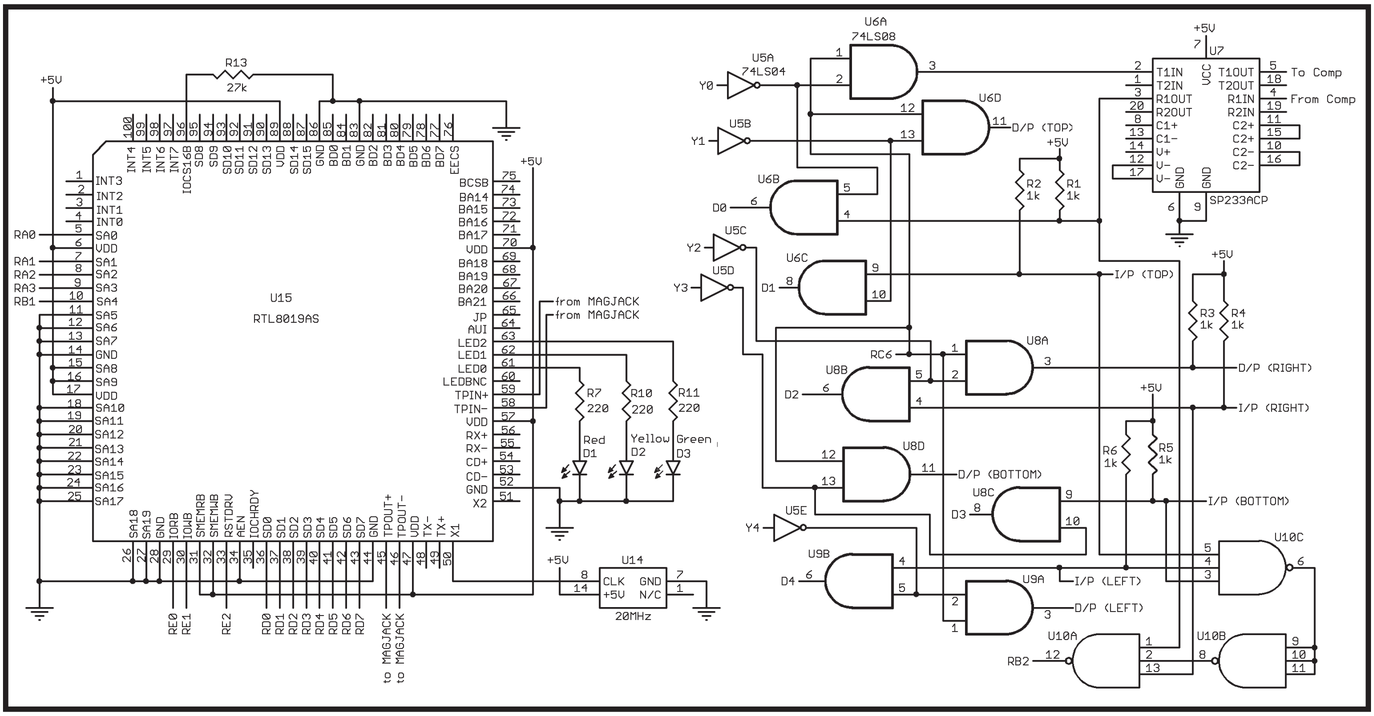


Figure 3—*The network control board is fairly simple. The green LED (D3) is for transmission. The amber LED (D2) is for reception. Red (D1) indicates collision detection.*

divided into 64 pages of 256 bytes each that store both transmitted and received data. The 64 pages are divided into transmit and receive buffers using four pointers. Two pointers indicate the start and end of the receive buffer section. The other two indicate the start and end of the transmit buffer section. Each packet is stored in an integral number of pages so one page can't contain data from more than one packet.

The RTL8019AS handles all of the data link layer functions for the system. As a result, the IC handles cyclic redundancy check (CRC) generation and verification automatically. Each received packet is prefixed with 4 bytes of header information. The first byte is the status byte, which is a reflection of the status register within the IC. The second byte is a pointer to the next free page in the ring buffer. The remaining 2 bytes are the total length of the received packet, including the 4-byte header.

Each received packet generates an interrupt that's fed to one of the external interrupts of a Microchip Technology PIC microcontroller. We use two PIC18F*xxxx* series microcontrollers in our design. A PIC18F4620 (or PIC18F4680) handles all the network procedures and acts as the master microcontroller. A PIC18F242, which is considered the slave microcontroller, formats and displays the received data. The two microcontrollers communicate via a simple message protocol over a four-wire, half-duplex SPI interface. Both microcontrollers operate at 10 MIPS, which is acceptable for the level of network traffic expected during normal operation.

The LCD modules display network information about the node such as its IP address, netmask, gateway IP address, and group IP address (see Photo 3, see p. 50). The first three addresses are standard on any PC (type "ipconfig /all" at a command prompt in Windows or "ifconfig" in UNIX). The group IP address is mainly found when using the Internet Group Management Protocol (IGMP), which allows routers to send information to groups of PCs (multicasting). An example is the streaming video of the World Series for paid subscribers. Using a group address allows a collection of panels to be grouped together for display purposes. The result is that a single display can be partitioned into multiple screens to display different content. For example, you can display information about two separate games on different halves of the same display board.

Note that all of the code is written in PIC assembly language. This reduces code size and increases execution speed. Currently, all of the code and embedded web pages for the PIC18F4620 fit in 24 KB of the total 32 KB of available program memory. Code for the PIC18F242 fits in 4.5 KB of the total 8 KB of available program memory. Thus, there's plenty of room for future development.

Next is the MAX6952 LED matrix display driver. This IC drives up to 140 LEDs that are arranged as four characters, each a 5 × 7 dot-matrix. Each panel in the design is driven by three addressable MAX6952s. Therefore, each panel contains 420 LEDs arranged in three rows of four characters. The MAX6952EPL simplifies LED control because of its internal font table and simple command format. The font comprises 104 characters in ROM and 24 user-definable characters. Our display is mainly for text, but there are some basic graphics in the character set. You can build up to 24 additional graphic characters. The PIC18F242 coordinates these drivers. It takes data sent from the PIC18F4620 and formats it to fit the display.

Let's consider an example. To display the word "Message," the microcontroller extracts the first four characters ("Mess") and sends them individually to the first MAX6952EPL driver. The driver then looks up the character in its font table and displays it (if it's found). Otherwise, it displays a null character (all of the LEDs are on). The microcontroller then extracts the final three characters ("age") and sends them to the second driver. The third driver isn't accessed in this example. For a single panel, if there are more than 12 characters in the message, only the first 12 characters are displayed. The others characters are simply discarded. A simple extension of this logic is used to handle multipanel displays.

DISPLAY CONFIGURATION

You must choose the display's configuration as either automatic or semiautomatic. Your choice will depend on whether a dynamic host configuration protocol (DHCP) server is connected to the system. If a DHCP server is present, the automatic configuration is used. In this mode, the DHCP server allocates all IP addresses for the controlling PC and the nodes. This is similar to what happens when you connect your PC to the network at your office.

If there isn't a DHCP server, the system reverts to a semiautomatic configuration. In this configuration, two events occur. First, all of the nodes try to generate a unique IP address for themselves from within the range of 192.168.0.1 to 192.168.254.240 with a netmask of 255.255.0.0. Second, you must manually select an IP address from 192.168.254.241 to 192.168.254.253 with a netmask of 255.255.0.0. There is a default gateway address of 192.168.254.254 because Windows XP allows both a primary and alternate configuration when setting up IP addresses.

The primary configuration is usually set to obtain an IP address automatically (DHCP). If this fails, the alternate configuration is used. We use this alternate configuration with the aforementioned addresses. Windows systems may require administrative privileges to change these settings.

SYSTEM OPERATION

At power-up, each node sets up all of the necessary control registers and initializes a software clock. The software clock is the basis for all of the program's timing functions. After clock initialization, the node performs a 500-ms delay to allow all of the components to stabilize before initializing the LCD, RTL8019AS chip, and USART module. Each node then checks the network for a DHCP server in order to obtain an IP address.

A number of conditional decisions now apply. If a DHCP server is present and there are enough available addresses, each node is assigned a unique IP address. On the other hand, if no DHCP server is present, each node generates its own address and

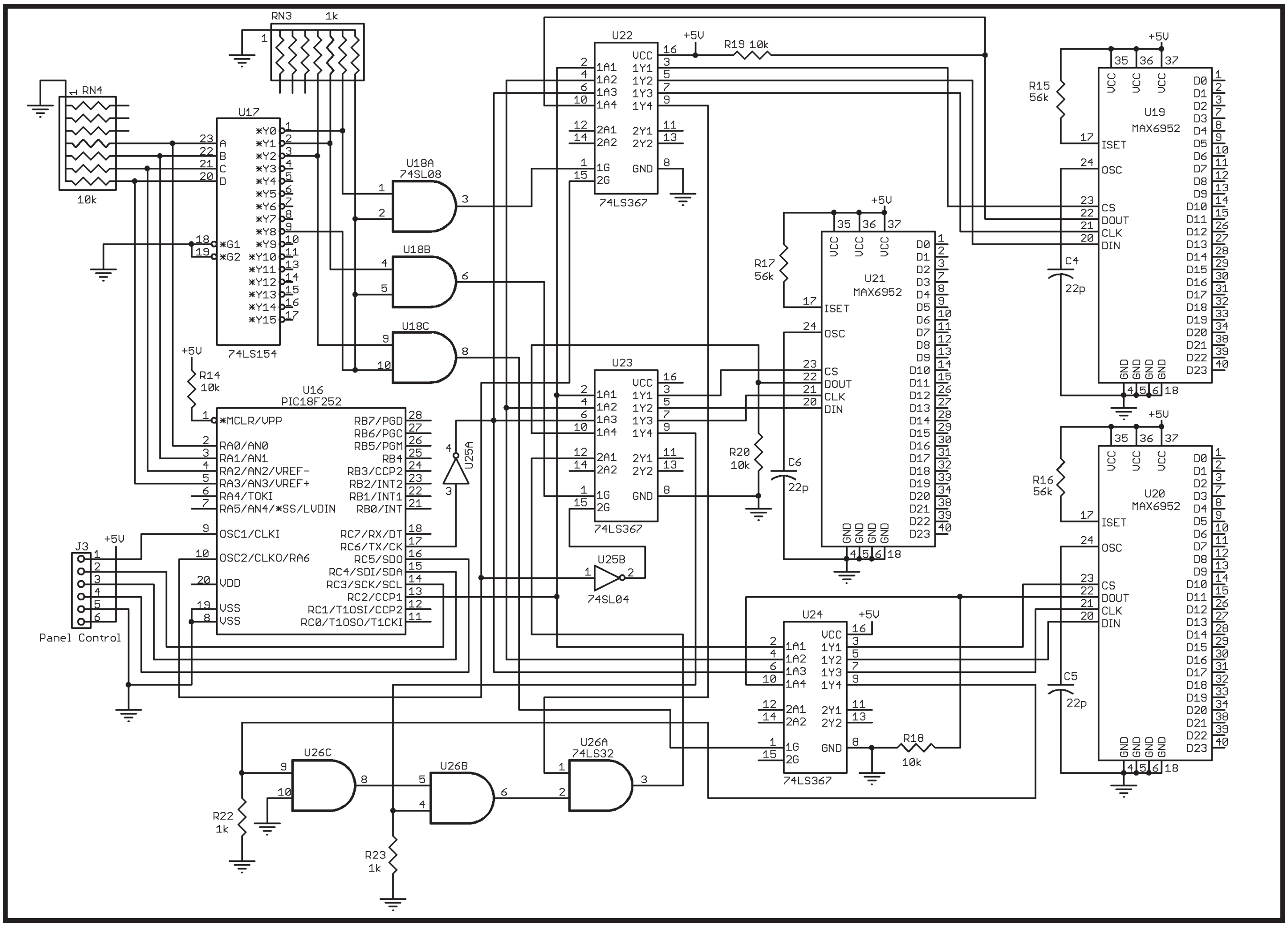


Figure 4—*Check out the panel control board. Outputs from the MAX6952 go directly to the 1-ft² panel. Larger panels require drivers.*

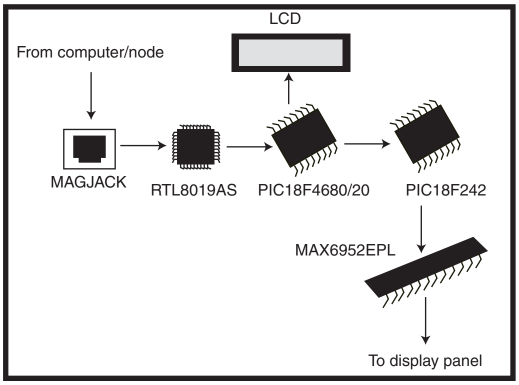


Figure 5—*Information flows through a panel. Large panels have a display driver after the MAX6952EPL.*

broadcasts it using an address resolution protocol (ARP) packet. If another node has already allocated that address, another is generated and the broadcast repeats. This continues until the node attains a unique IP address.

The final step in the initialization phase involves the built-in serial port, which checks if any neighboring panels are connected. Neighbor discovery is performed after a particular node has acquired an IP address. This process uses the microcontroller's EUSART port (in RS-232 mode) to communicate with its neighbors. This single port is multiplexed to provide five RS-232 ports at 115 kbps (one for a PC and one for each of the four neighboring panels). The node addresses each neighbor port in the following order: the top neighbor, the right neighbor, the bottom neighbor, and the left neighbor. Once addressed, the node sends each neighbor a GetNodeAddress command, which returns the neighbor's MAC address and IP address. This information is stored in a look-up table for future reference. If there is no neighbor, the request times out and the next neighbor in the sequence is addressed. The program then sits in an infinite loop and waits for packets and checks for neighbors.

When a new packet arrives, the PIC18F4620 firmware parses the packet header for the type of protocol being transported. The supported protocols are TCP, UDP, ICMP, NetBIOS, and SMB. Due to program memory constraints, complete protocols aren't supported. As a result, only essential commands are implemented. If a protocol isn't supported, the packet is simply discarded.

We designed the entire system to be user friendly and to support plug-and-play functionality. Each panel requires certain cables to operate properly. A power cable moves power from the power supply to the node. A network cable serves as a straight-through (patch) cable between the node and the network switch. Four RS-232 crossover cables connect the neighboring panels (one cable for every two panels). An RS-232 cable is optional. You can use it to control a panel if the NIC fails.

Each node has a unique name that any user can easily reference. The node names are structured as "Nxxxxx," where "xxxxx" is an integer ranging from 00001 to 65535 (inclusive). Node names are generated and assigned at start-up on a first-come, first-serve basis. The entire process is entirely transparent to a user.

After acquiring an IP address (DHCP or otherwise), each node generates a name for itself and broadcasts it. If the name has already been claimed by another node, the broadcasting node generates a new name and repeats the process until a unique name is established. The algorithm assumes the display is rectangular or square in size and orientation (no diamonds). The top-left panel is the panel whose top and left neighbors do not exist (i.e., IP addresses are zero and similar for the bottom-right panel). Therefore, the top-left panel does not necessarily have to be N00001, just the luck of the draw at start-up.

Unfortunately, PC and web site names have no significance for network components when delivering data. Networks use IP and MAC (hardware) addresses to correctly send data to the appropriate recipient. Thus, you have to associate the node name with its corresponding IP address. This is accomplished with the NetBIOS Name Service (NBNS) protocol in order to request a node's IP address given the node name. The result is that either the node name or the node IP address can be used to find a node.

But which node name should you use? Because the system is symmetrical, any node will suffice. And provided the system is powered and fully connected, there will always be a "N00001" node. Therefore, simply enter the address http://N00001 to access the system.

All commands to the system are issued from a simple web page that's served from the node on which you're logged. Due to the limited availability of SRAM on the PIC18F4620 (3,986 bytes), packet fragmentation isn't allowed. Therefore, all web pages (headers, etc.) must fit within a single 1,500-byte packet. As a security measure, you're required to enter a username and password when you first start the system. Once you gain access, you can issue commands from the command screen (see Photo 4, p. 50).

SYSTEM COMMANDS

The system can operate in either Local or Global mode. In the former, commands are only executed on the node to which you're currently logged. In contrast, commands executed in Global mode affect the entire display. Selecting a mode is an on-demand decision determined by selecting either the Local or Global button before sending a message.

Local mode is operationally simple. When you fill out the web page, the PC encapsulates the message in a TCP packet and sends it to the node. Due to the unknown length of messages, all text is transported in the body of the packet using the POST method instead of the common GET method. Using the GET method is similar to writing a message on a postcard. You just want to convey basic information to the recipient. Using the POST method is analogous to sending a letter to a friend. All of the information is in the letter that you place in an envelope and put in the mail. The letter can be one word or take up several pages.

For example, consider a system with four panels: N00001, N00002, N00003, and N00004. If you send a local message to the first panel (N00001), then only that panel is updated. In this case, your PC sends data in a TCP packet addressed to the first panel. The PC obtains the panel's address when you type http://N00001 in your browser's address bar.

Global mode follows the same ini-

How far will your design take you?

DESIGN
STELLARIS
2006

Challenge yourself against other top embedded engineers around the world in **DesignStellaris2006**, proudly sponsored by Luminary Micro, Keil, and Circuit Cellar.

Use any microcontroller in Luminary Micro's Stellaris™ family of ARM® Cortex™- M3 controllers with Keil's RealView® Microcontroller Development Kit (RVMDK) to create your design contest entry, and see how far your design will take you!

- No purchase necessary to enter.
- $10,000 in cash prizes!
- Entry deadline is February 7, 2007.
- Winners will be announced at the Embedded Systems show... Silicon Valley 2007.
- Submit your design today!

FOR COMPLETE DETAILS, VISIT
www.LuminaryMicro.com/ DesignStellaris2006

DESIGN
STELLARIS
2006

LUMINARY MICRO™

KEIL™
An ARM® Company

CIRCUIT CELLAR®

www.LuminaryMicro.com/DesignStellaris2006

LUMINARY MICRO
LM3S811 EVALUATION KIT
Visit our website
for more information!

THE LM3S811 EVALUATION KIT includes the Stellaris LM3S811 Evaluation Board, an evaluation copy of the Keil RealView Microcontroller Development Kit, USB cable, documentation, and programming examples.

Cortex
Intelligent Processors by ARM

PC/104 Single Board Computers

Low Price, Low Power, High Reliability using Linux development tools


TS-7200 shown with optional A/D converter, Compact Flash and RS-485

- options include:

  onboard temperature sensor, A/D Converter 8 channel 12 bit, Extended Temperature, Battery Backed Real Time Clock, USB Flash 256 M (with ARM Tool Chain), USB WiFi

200 MHz ARM9
Power as low as 1/4 Watt

- 4 boards, over 2000 configurations
- Fanless, no heat sink
- SDRAM - up to 128M
- Flash - up to 128M onboard
- 10/100 Ethernet - up to 2
- DIO lines - up to 55
- 2 USB ports
- COM ports- up to 10
- Linux, NetBSD
- Real Time extension

$119 qty 100

$149 qty 1

NEW!
- SD card option

Off-the-Shelf Solutions ready to design into your project using DOS development tools


TS-5600 Shown with optional flash modules, A/D, RS-485 and Merlin cellular modem

- options include:

  RS-485 Half and Full Duplex, A/D Converter up to 8 Channels at 12 bits, DAC up to 2 Channels at 12 bits, Extended Temperature

133 MHz 586

- Power as low as 800mA
- Fanless, no heat sink
- SDRAM - up to 64M
- COM Ports - up to 4 ports
- Ethernet Ports
- DIO Channels - up to 40
- PCMCIA II adaptor
- Compact Flash adaptor
- USB Ports (Except on TS-5300)

$229 qty 1

$259 qty 1

see our website for 33 MHz 386 configurations

- Over 20 years in business
- Open Source Vision
- Never discontinued a product
- Engineers on Tech Support
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products stocked and available for next day shipping

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

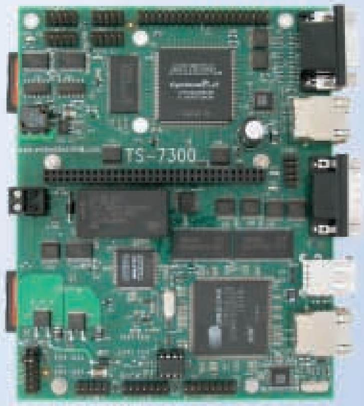
New Products and PC/104 Peripherals

High Security Linux FPGA Computer

NEW!

$189
qty 100

- 200 MHz ARM9
- Up to 128M SDRAM
- Programmable FPGA
- 2 SD card slots
- 2 10/100 ethernet
- VGA video out
- DIO - up to 55
- 2 USB ports
- RoHS compliant

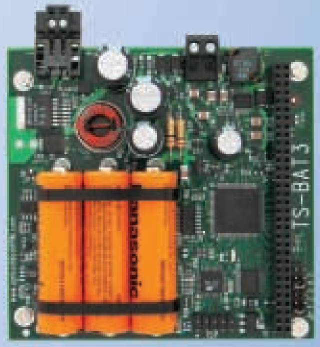


Intelligent Battery Back-up

NEW!

$119
qty 1

- Run your system for days with no external power source



Modems	33.6K baud, 56K baud, AT commands, caller ID, cellular using GSM and CDMA technologies
Non-volatile Memory	up to 2MB, 10 year lithium battery
Serial Ports	up to 4 serial ports with optional RS-485
Opto-isolated Serial Port	opto-isolated RS-485 on 2 ports
12 bit A/D, DAC	8 channel 12-bit A/D converter, optional 2 channel 12-bit DAC, A/D jumpered for 0-2.5V, 0-10V or 0-20mA
CAN Bus Controller	Philips SJA1000, opto-isolated, up to 1 megabit/sec selectable termination resistor, Ocera Linux driver
NEW! 64 Digital I/O	32 inputs, 32 outputs, 200 mA drive, optional 512 Kbyte or 1 MB battery-backed SRAM, stack up to four boards, RoHS compliant

see our website for more boards and option details

Technologic
SYSTEMS

We use our stuff.

Visit our TS-7200 powered website at

www.embeddedARM.com

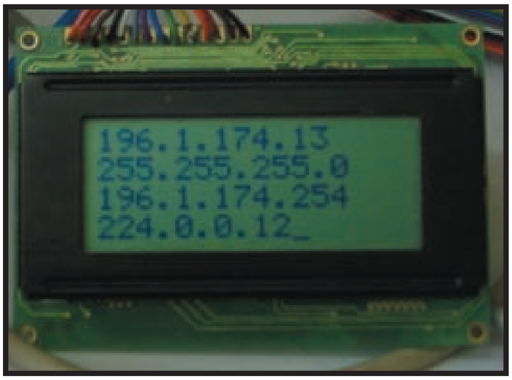


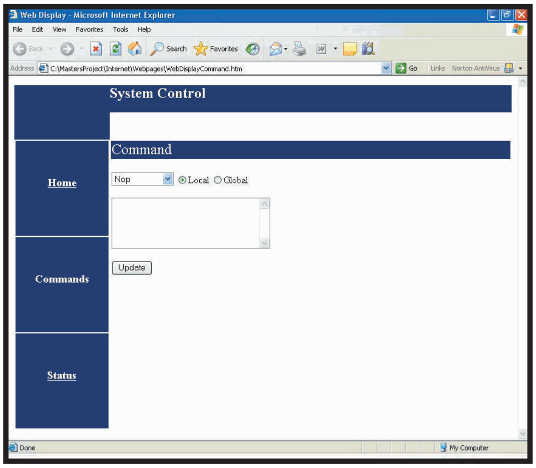
Photo 3—*This LCD module displays network information about the node such as the node IP address, netmask, gateway IP address, and group IP address.*

tial process as Local mode but goes a step further where the single message affects the entire display. After the first panel (N00001) receives the message, it copies the data from the TCP packet to a UDP packet and sends the data to its right and bottom neighbor panels. The neighbor addresses are retrieved from the look-up table populated by the `DiscoverNeighbour` routine.

All nodes use UDP port 6051 for global communication. These neighbors relay the message to their right and bottom neighbors until the message reaches the panels at the extreme top right and extreme bottom left of the display. Each relayed message increments a hop counter within the message header so each panel knows its general location within the display layout. The extreme panels then broadcast the message with the maximum hop counts to all of the panels in the display. This way, each panel obtains the general layout and their relative position within the display. As a result, they can determine which part of the message to extract and display using the hop counter as pointers into the message data.

Therefore, the panels sort out among themselves how to display the transmitted message. All display messages are limited to a maximum of 1,340 characters. This is because packet fragmentation is not supported. Thus, all PC-transmitted data must fit within one TCP packet. Future implementations can address this problem by using external RAM to store multiple packets.

When the packet arrives at the destination node, it's parsed to determine the type of protocol and method. The raw data is extracted and transmitted to the PIC18F242 for formatting and display. As a result, the PIC18F4620 doesn't perform any formatting.



Address C:\MastersProject\Internet\Webpages\WebDisplayCommand.htm

System Control

Home

Commands

Status

Command

Nop ● Local ○ Global

Update

Photo 4—*The control web page (written in HTML) is easy to expand.*

KEIL™
An ARM® Company
Microcontroller Tools
µVision® IDE
µVision Project Manager
C/C++ Compiler
Macro Assembler
C/C++ Libraries
Linker / Locator
µVision Debugger
Device Simulation
Target Hardware
RTOS
RTOS Kernel
TCP/IP Suite
TCP, UDP, PPP, SLIP
ARP, DNS,
Ethernet, DHCP,
HTTP, TFTP, SMTP
Flash File System
USB Device Interface
CAN Interface
Professional Tools for Over 1,000 Devices
- 8-bit: 8051 and Extended 8051 Variants
- 16-bit: C16x, XC16x, and ST10
- 32-bit: ARM7, ARM9, and Cortex-M3
New! RealView® ARM Compiler
✆ 800-348-8051 www.keil.com/xd

Enhance Your Connectivity
The RCM4000 — Ethernet And Much More
- Ethernet with royalty-free TCP/IP stack
- Up to 25 configurable GPIO
- On-board 12-bit A/D
- Up to 5 serial ports
- 512K flash (program), 32 MB NAND flash (data)
- Rabbit® 4000 at 58.98 MHz
New! Next Generation Rabbit 4000
RCM4000
Complete Development Kit
$149 Regular $239 Limited Time Offer
RABBIT Semiconductor
Order Online At rabbit4000.com
06227

Although this method may seem lengthy, each PIC18F4620 can generate a UDP message in less than 1 ms. Separate processors handle network communication and data display. Don't you have embedded parallelism here?

Currently, only four commands (Display Text, Clear Screen, Scroll Left, and Scroll Right) are supported. Additional commands can be easily incorporated into the system.

SYSTEM UPDATES

The current system works well, but it isn't perfect. Hardware and software improvements will enhance the system's functionality to put it on par with commercial products. We plan to add a faster NIC to increase the transmission rate from 10 to 100 Mbps. We will add a larger and faster microcontroller along with the faster NIC (e.g., a Microchip dsPIC or equivalent).

We also plan to add memory. The current 3,986 bytes aren't enough to process more than one incoming or outgoing packet at a time. More memory will enable us to store more packets while the microcontroller performs other actions.

On the software side, we plan to add a real-time operating system (RTOS). This will allow the microcontroller to run multiple programs concurrently.

We're considering a bootloader program because we currently have to perform all of the firmware updates manually. This requires us to remove the ICs from the circuit in order to program them in a PIC programmer. A bootloader program will allow us to update the firmware automatically and collectively via the network.

We will enhance the system to handle full motion and full-color displays. This won't be easy, but it will be extremely interesting to test the limits of microcontrollers.

Peter Gibbs (pgibbs@uwichill.edu.bb) is a senior lecturer in physics and electronics at the University of the West Indies (Cave Hill Campus) in Barbados. He holds a Bachelor's degree in physics and math, as well as a diploma of education, from the University of the West Indies. He earned a Master's degree in physics at the University of Guelph in Ontario. When he isn't working on embedded systems control applications, Peter enjoys long-distance swimming.

Ramon Sargeant is a graduate student studying electronics at the University of the West Indies. He holds a Bachelor's degree in physics and electronics from the University of the West Indies and a Master's in mechatronics from King's College, London. You may contact Ramon at sargeantram@hotmail.com

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2006/195.

RESOURCES

S. Anderson-Freed, *Weaving a Website: Programming in HTML, JavaScript, Perl and Java*, Prentice Hall, 2002.

P. Gibbs, "Large-Scale Electronic Display," *Circuit Cellar* 184, 2005.

Microchip Technology, "PIC18F2525/2620/4525/4620 Data Sheet," DS39626b, 2004.

———, "PIC18FXX2 Data Sheet," DS39564B, 2002.

G. Musgrave, "Very Large-Screen Video Displays," Conceptron Associates, January 1999.

A. S. Tanenbaum, *Computer Networks*, Prentice Hall, 2003.

Viola Systems, OpenTCP Ethernet Protocol and Driver for Realtek's RTL8019AS, /opentcp/Ethernet.c, www.opentcp.org/documentation/api/html/files.html.

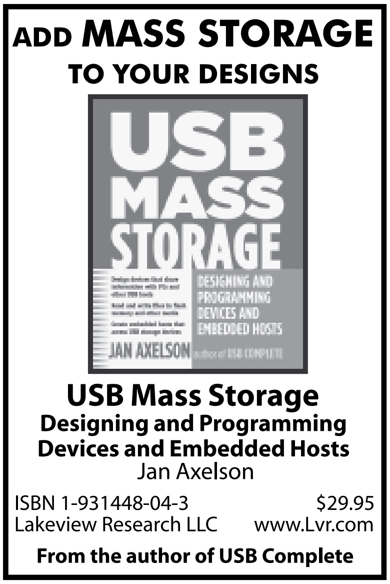
SOURCES

PIC18F242 and PIC18F4620 Microcontrollers
Microchip Technology, Inc.
www.microchip.com

MagJack Connector
Digi-Key Corp.
www.digikey.com

Ethereal network analyzer
www.ethereal.com

RTL8019AS NIC
Realtek Semiconductor Corp.
www.realtek.com.tw


ADD MASS STORAGE
TO YOUR DESIGNS

USB Mass Storage
Designing and Programming
Devices and Embedded Hosts
Jan Axelson

ISBN 1-931448-04-3 $29.95
Lakeview Research LLC www.Lvr.com

From the author of USB Complete


USB/ETHERNET DAQ

LabJack
UE9

Available now
for only ...
$399 qty 1

USB/Ethernet Data
Acquisition & Control

- USB 2.0/1.1 and Ethernet
- 14 analog inputs (12- to 16-bit)
- Stream input data up to 50 kHz
- Use with C, VB, LabVIEW, etc.
- Includes DAQFactory Express
- Operates from -40 to +85 deg C
- 2 analog outputs (12-bit)
- 23 digital I/O
- Up to 2 counters (32-bit)
- Up to 6 timers
- Approx. 3" x 7" x 1"

LabJack Corporation, Colorado, USA
info@labjack.com, (303) 942-0228

www.labjack.com


ANDRE LAMOTHE'S
XGAMESTATION
DO-IT-YOURSELF VIDEO GAME SYSTEM

Inspired by the
Atari 2600, Apple II &
Commodore 64!

SX52 @
80 MIPS

INCLUDES:
- Assembled XGS Micro Edition Unit!
- Complete Development Kit!
- Tools, Demos & Utilities!
- eBook on Designing the XGS Console!
- Cables and Power Supply Included!

WWW.XGAMESTATION.COM
(925) 736-2098 SUPPORT@NURVE.NET

by Dhananjay Gadre

Power Generator for Portable Applications

Battery-less LED flashlights have become quite popular because they offer a perpetual source of power. After studying how LED flashlights work, Dhananjay built a hand-held power generator for portable electronic applications.

Portable applications often require battery power. Solar panels are good sources of power, but they aren't reliable under some conditions. In this article, I'll describe a source of clean, affordable, and reliable power that's often ignored: muscle power! Although a muscle-powered solution may not be suitable for an application that requires long periods of unattended operation, my simple generator is perfect for a small system such as my TV's remote control (see Photo 1).

MUSCLE POWER

There are numerous advantages associated with using a muscle-powered generator. Imagine not having to worry about draining the battery in your favorite electronic application. And think about the benefits for the environment. By reducing your dependence on batteries (and even if they're rechargeable), you can help contribute to the goal of a cleaner planet.

Many LED flashlights on the market rely on muscle-powered generators. My power generator design is similar to the ones used in such flashlights. It consists of two main parts: a voltage generator and a voltage regulator. You can adapt the generator, which is based on Faraday's principle, with the help of large capacitors such as supercapacitors and suitable DC/DC converters. By doing so, you can meet almost any voltage/current requirement in a portable application.

Photo 1—*Drop the batteries and go "green." Simply connect my power generator to a regular TV remote control and shake.*

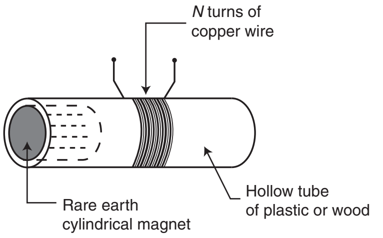


Figure 1—*Use your muscle power to shake the generator and move the magnet. The tube is sealed with cork. What a simple design!*

SYSTEM OPERATION

The voltage generator portion of my design consists of a long Mylar tube with a coil wound around it and a cylindrical magnet inside it (see Figure 1). The magnet can slide freely inside the tube. The ends of the tube are sealed tightly with 3-mm pieces of cork.

To operate my power generator, you must shake tube vigorously so that the magnet traverses the length of the tube in a back and forth motion. The voltage generated across the coil is governed by Faraday's principle:

$$E = -N\left(\frac{d\phi}{dt}\right)$$

E is the induced voltage, and N is the number of turns in the coil. $d\phi/dt$ is the rate change of the magnetic flux through the coil. ϕ is the magnetic flux. Note that $\phi = B \times A$, where B is the magnetic field, and A is the area of the coil. Thus, a larger magnetic field or larger coil area cross section produces a larger voltage for the same rate of change. Similarly, with all other things being constant, increasing the number of turns of the coil also increases the induced voltage. For this design, the magnet is approximately 1 T. The area of the coil's cross-section is about 6 square centimeters.

As the magnet enters the coil, the

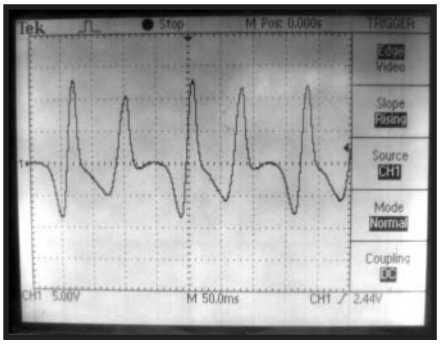


Photo 2—*I used an oscilloscope to capture the voltage generated by the shake generator. The horizontal scale is 50 ms/div. The vertical scale is 5 V/div.*

polarity of the voltage generated is opposite that of the polarity of the voltage when the magnet leaves the coil. Thus, a bipolar voltage is generated across the coil. This voltage is rectified with a bridge rectifier and a large capacitor is charged.

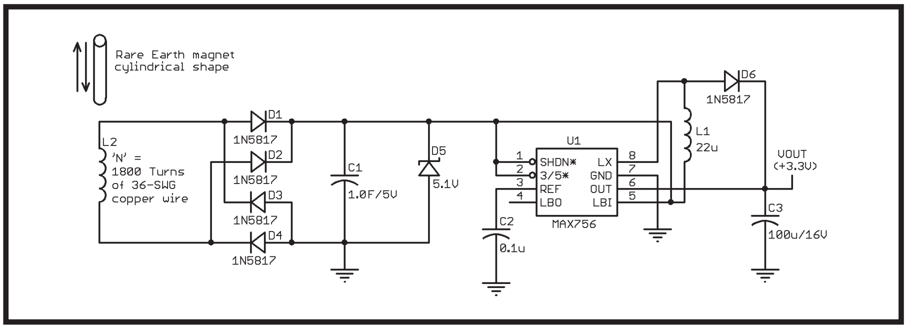


Figure 2—*The circuitry for my 3.3-V, muscle-powered generator is simple. Instead of 1N5817 diodes, you can use 1N4001 rectifier diodes.*

SYSTEM COMPONENTS

My power generator is a fairly simple system (see Figure 2). C1 is the charge storage capacitor. The larger the capacitor, the longer it takes to charge it. At the same time, a larger capacitor (such as a supercapacitor) will provide power for a longer amount of time for the same load. Zener diode D5 prevents the voltage across the capacitor from increasing beyond its voltage ratings.

The storage capacitor starts discharging as soon as load is applied to the capacitor. The voltage across the capacitor then starts dropping. If the application requires a fixed voltage, some sort of voltage regulation circuit must be used. For example, a microcontroller-based circuit often needs 5 or 3 V for operation. A step-up type of DC/DC voltage converter IC such as a Maxim MAX756 or MAX1722 can be used.

There is an advantage to using a step-up/down voltage converter. Even if the storage capacitor voltage drops below the required voltage of operation, the circuit will continue to provide output voltage until the storage capacitor voltage drops below the lower limit of the DC/DC converter IC's operating voltage. (In the case of a MAX1672, the limit is 1.8 V.) A converter with low quiescent current is of course preferred. Table 1 is a list of some of the available converters.

A supercapacitor is often rated at a much lower voltage than normal electrolytic capacitors. Common voltage ratings are between 2.3 and 5 V. Thus, if the output voltage required from the shake generator is 5 V, then a step-up type of DC/DC converter is suitable (instead of a step-up/down converter).

CHOOSE A CONVERTER

Choosing the right DC/DC converter for a given application is important. The MAX1672 and MAX710 are suitable when the open circuit voltage generated by the shake generator is more than the required stabilized output voltage. Typically, a small storage capacitor will rapidly charge to the open load voltage generated by the shake generator.

On the other hand, supercapacitors, which have large capacities (0.33 F and higher), have much less maximum voltage rating. Therefore, the voltage at which a supercapacitor can be charged should be less than its rated voltage. Simple DC/DC step-up converters (MAX756, MAX1722, or ZXSC100) are suitable for applications that require supercapacitors.

EXPERIMENTAL RESULTS

I made and tested several prototypes. The results from my experiments matched the output from Faraday's equation. As you can see in Photo 2, I recorded the bare output of a prototype with 1,800 turns of 36 AWG copper wire on an oscilloscope. The prototype can charge a 1-F, 5-V supercapacitor to 3 V in less than 100 shakes.

Device	Manufacturer	Converter topology	Minimum voltage for operation	Minimum startup voltage	Quiescent current
MAX1672	Maxim	Step-up/down	1.8 V	0.9 V	85 µA
MAX710	Maxim	Step-up/down	1.8 V	0.9 V	200 µA
MAX1722	Maxim	Step-up	0.91 V	0.83 V	1.5 µA
MAX756	Maxim	Step-up	0.7 V	1.1 V	60 µA
ZXSC100	Zetex	Step-up	0.92 V	1.01 V	150 µA

Table 1—*There are numerous converters on the market. Here are a few options.*

The same prototype was also used to charge a 0.5-F, 5-V supercapacitor, and a 1-kΩ resistance load was applied to the capacitor. The supercapacitor was charged to 3.75 V and allowed to discharge through the load. Figure 3a (p. 54) shows the plot of supercapacitor voltage as a function of time.

I tested the circuit in Figure 2 with a constant load of about 1.6 mA. The output remained stable for more than 6 min. The 1-F supercapacitor was charged to 3 V. The MAX756 DC/DC converter output was set to 3.3 V. The supercapacitor voltage was monitored as a function of time until the DC/DC converter output voltage was 3.3 V (see Figure 3b).

Because the supercapacitor must now provide constant power, its voltage drops:

$$V_{CAP} = \sqrt{V_{MAX}^{\;2} - \frac{2pt}{C}}$$

V_{CAP} is instantaneous voltage across the supercapacitor. V_{MAX} is the initial voltage across the supercapacitor. p is the power drawn by the load, and C is the capacitance of the supercapacitor. In simple terms, it means that if the square of the supercapacitor voltage is plotted as a function of time, you'll get a straight line (see Figure 3c). The DC/DC converter offers a constant power load to the supercapacitor.

APPLICATIONS

You can use my simple generator to power a variety of portable applications. A battery-free flashlight featur-

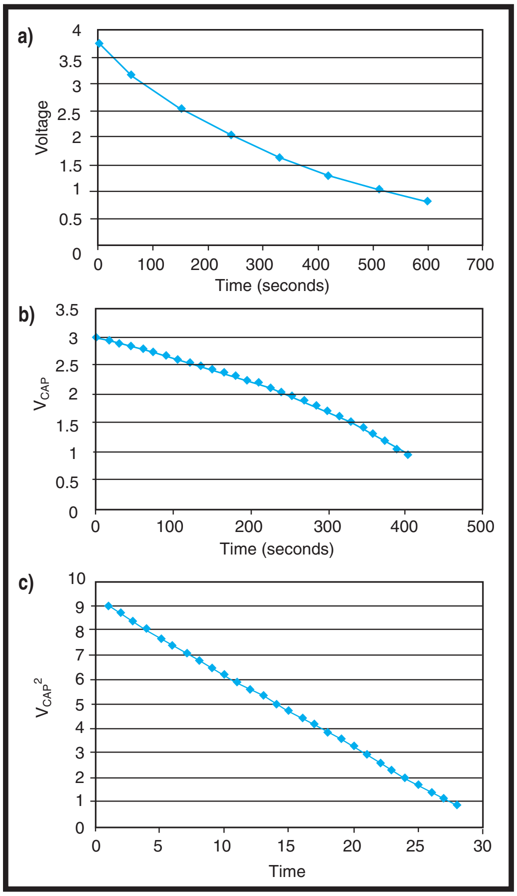


Figure 3a—*This is a plot of 0.5-F supercapacitor discharge voltage through a 1-kΩ resistor as a function of time.* **b**—*Check out 1-F supercapacitor discharge voltage through a constant power load of 5.28 mW as a function of time.* **c**—*Finally, take a look at* V_{CAP}^2 *as a function of time for a constant power load of 5.28 mW.*

ing bright white LEDs is a great example. You can use a supercapacitor to provide voltage to a DC/DC converter. The pulsed output of the DC/DC converter (without the output smoothening capacitor) will then drive the white LEDs. A simpler solution would be to use the supercapacitor's voltage to drive white LEDs via a simple circuit.

I use the generator to power my TV's remote control. You can do the same by integrating the generator with a custom TV remote or with the help of a 3-V DC/DC converter to power your existing remote. Wouldn't it be nice to get rid of your remote's batteries?

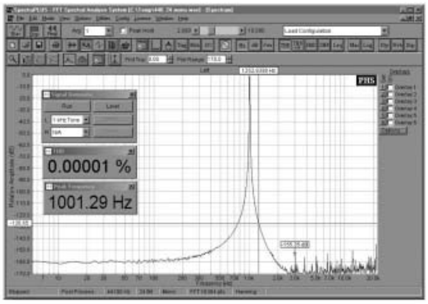
I opened the remote's battery compartment and then removed the batteries and connections to make more space for the PCB. I constructed the circuit on a small PCB (see Photo 3a). I then opened the TV remote casing and soldered the output wires from the converter PCB to the TV remote PCB (see Photo 3b). I now have a "green" remote. I like to think of it as my small contribution toward the goal of a cleaner environment. I have to shake the little beast to make it work, but I consider it part of my daily exercise regimen.

Author's note: I would like to thank Professor Prabhat Ranjan for introducing me to LED flashlights. I'd also like to thank my wife Dr. Sangeeta D. Gadre for her feedback and Mr. Satyaprakash for making the tubes and winding the coils and fabricating the circuit boards for this project.

Dhananjay V. Gadre holds a Master's degree in electronic science from the

Turn Your PC into a Real-Time Audio Spectrum Analyzer

Features
FFT to 1048576 pts
24 bit, 192kHz
1/96 Octave
Dual Channel
3-D Surface Plot
Spectrogram Plot
Digital Filtering
Signal Generation
RT-60, Leq, Delay
THD, IMD, SNR
DDE, Logging
Vibration Analysis



SpectraPLUS 5.0
FFT Spectral Analysis System

Download 30 day FREE trial!
www.spectraplus.com

Starting at $295

PHS
Pioneer Hill Software
24460 Mason Rd
Poulsbo WA 98370
360 697-3472 voice
360 697-7717 fax
pioneer@telebyte.com


EMBEDDED LINUX Made Easy!

Panel PCs
PC Compatible SBCs
SoM & PC/104 Modules
Custom Engineering
Embedded Servers

Turn-Key Solutions!

- ECLIPSE IDE with full Library and Module Support
- Integrated GDB Source Level Debugger
- Resident Menu Driven Setup Utility
- No Charge for Tools or Standard Linux Install
- Free Integration, Setup and Test

2.4 Kernel

Since 1985
OVER
21
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.
Phone 618 - 529 - 4525 Fax 618 - 457 - 0110
2390 EMAC Way, Carbondale, Illinois 62902
World Wide Web: http://www.emacinc.com

University of Delhi and a Master's degree in computer engineering from the University of Idaho. Since 2001, he has worked as an assistant professor in the ECE division at the Netaji Subhas Institute of Technology in New Delhi. Dhananjay is a licensed radio amateur (VU2NOX). You may contact him at dvgadre@gmail.com.

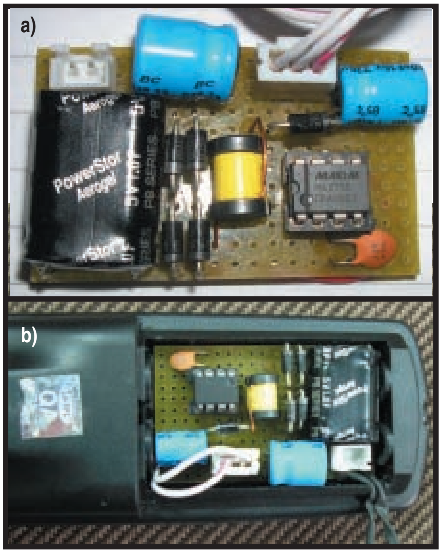


Photo 3a—*This PCB corresponds to the circuit shown in Figure 2.* **b**—*The generator's voltage converter circuit sits inside my remote control's battery compartment.*

PROJECT FILES

To download additional photos, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2006/195.

RESOURCES

A. Dutcher, "Single Cell Lights Any LED," *EDN*, 2001, www.edn.com/contents/images/70501di.pdf.

Maxim Integrated Products, "MAX1672: Step-Up/Down DC-DC Converter in QSOP Package," rev. 2, 19-1307, 2002, http://pdfserv.maxim-ic.com/en/ds/MAX1672.pdf.

———, "MAX1722/MAX1723/MAX1724: 1.5µA IQ, Step-Up DC-DC Converters in Thin SOT23-5," rev. 0, 19-1735, 2001, http://pdfserv.maxim-ic.com/en/ds/MAX1722-MAX1724.pdf.

———, "MAX710/MAX711: 3.3V/5V or Adjustable, Step-Up/Down DC-DC Converters," rev. 0, 19-1254, 1997, http://pdfserv.maxim-ic.com/en/ds/MAX710-MAX711.pdf.

———, "MAX756/MAX757: 3.3V/5V/Adjustable-Output, Step-Up DC-DC Converters," rev. 2, 19-0113, 1995, http://pdfserv.maxim-ic.com/en/ds/MAX756-MAX757.pdf.

R. Resnik, D. Halliday, and J. Walker, *Fundamentals of Physics*, John Wiley and Sons, Hoboken, NJ, 2000.

Zetex Semiconductors, "ZXSC100: Single Cell DC-DC Converter Solution," 2004, www.zetex.com/3.0/pdf/ZXSC100.pdf.

SOURCES

PowerStor Aerogel supercapacitor
Cooper Bussman
www.cooperet.com

10-F Capacitor
Spark Fun Electronics
www.sparkfun.com

ZXSC100 Converter
Zetex Semiconductors
www.zetex.com


HIGH TECHNOLOGY PROTOTYPE PCBs

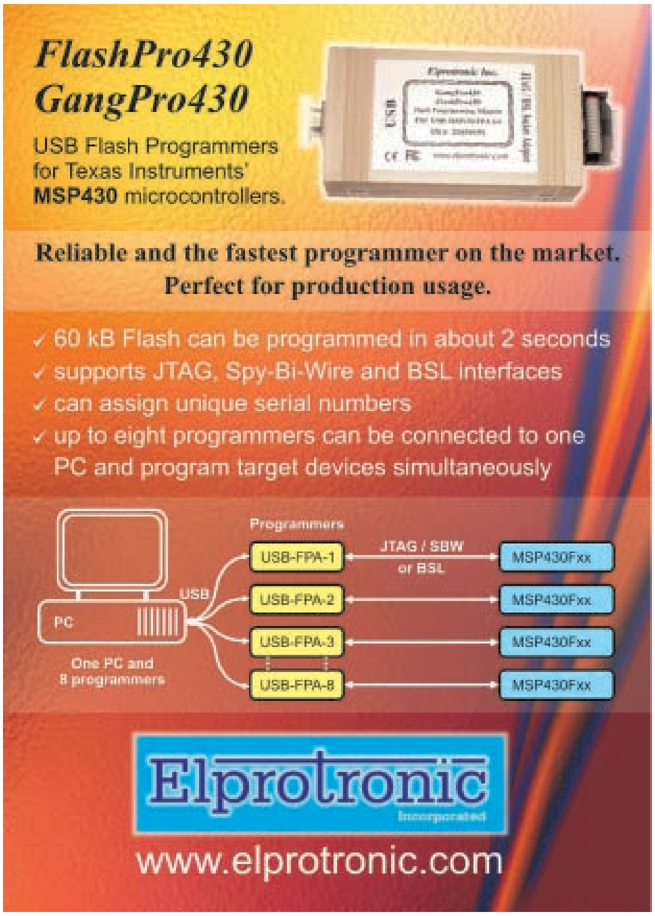
- MICROELECTRONICS—LINE WIDTHS DOWN TO 1.25 MIL. (0.00125")
- HDI—STAGGERED, STACKED AND FILLED MICRO VIAS
- HIGH RELIABILITY LEAD–FREE BOARDS
- HEAVY COPPER BOARDS

SUPPORTING THESE MARKETS:
• MEDICAL • MILITARY • SPACE • DOWNHOLE • HIGH RELIABILITY • SECURITY

SIERRA PROTO EXPRESS
800.763.7503
WWW.PROTOEXPRESS.COM

CELEBRATING 20 YEARS!

A DIVISION OF
SIERRA PROTO EXPRESS

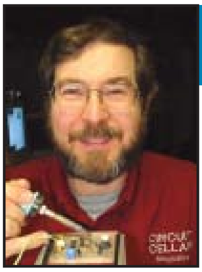

FlashPro430
GangPro430

USB Flash Programmers for Texas Instruments' **MSP430** microcontrollers.

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

✓ 60 kB Flash can be programmed in about 2 seconds
✓ supports JTAG, Spy-Bi-Wire and BSL interfaces
✓ can assign unique serial numbers
✓ up to eight programmers can be connected to one PC and program target devices simultaneously

Elprotronic
Incorporated
www.elprotronic.com



by Ed Nisley

DIY Diodes

Diodes started with do-it-yourself technology. Ed describes how those early diodes worked and explains how natural diodes can cause RF problems.

We think of diodes as standard devices, well defined by tidy datasheet tables, available in bulk, and indistinguishable from one another. It wasn't always that way and, back in the beginning, each diode had unique characteristics that might change from minute to minute.

The earliest diodes predate the electronics age, appearing as mineral lumps in "crystal radio" receivers powered by the incoming RF signal. Although circuits no longer depend on crystal diodes, you'll find similar effects in the strangest places, concealed by their very nature.

Let's take a look at how those first diodes behaved, because their properties remain relevant today. In particular, if your RF designs must work outdoors, this column's for you!

PREPARE FOR PROBING

The mineral crystals that act as diodes have characteristics that vary with chemical composition, surface condition, probe material, and the exact contact location and pressure. Every crystal radio description mentions finicky diode adjustments, with tedious probing to find a "sweet spot," because any vibration can shift the probe to an adjacent area that isn't a diode at all.

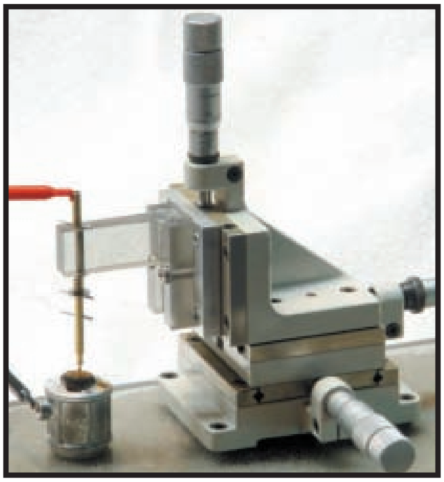


Photo 1—*A surplus optical positioner moves the probe accurately within a 0.5″ cube. A CNC-machined hole in the plastic baseplate puts the sample holders at an exactly repeatable location. Simple clip leads connect the sample to the electronics.*

I mounted my probe on the manual three-axis positioner shown in Photo 1 to minimize that problem for my measurements: the three micrometer screws cover a 0.500″ cube with resolution under 0.001″ and a spring-loaded brass tube ensures firm-but-gentle landings. A stereo zoom microscope helped me hit interesting sites on the tiny crystals.

I made point-contact probes by soldering rods of various metals to short brass tubes that friction-fit inside the spring-loaded tube. I made a carbon probe from a 0.7-mm HB pencil "lead" refill by drilling a 0.031″ hole in a solid brass rod and securing the carbon rod with a glue dot.

Photo 2 shows a close-up of a chalcopyrite (pronounced kal′-keh-pie′-right) sample mounted in a 0.75″ EMT fitting from the local home-repair superstore. A CNC-machined hole in the plastic base exactly matches two flats in the fitting's threaded bottom to ensure a precisely repeatable position. I sealed the fitting with an aluminum disk, seated the sample on some crumpled aluminum foil, and then cast Wood's Metal around it for solid positioning and good electrical contact.

The circuit in Figure 1 applies a bipolar current to the sample holder and converts the differential voltage across the crystal into a single-ended voltage for the oscilloscope. Set your function generator to produce a triangle wave centered around 0 V.

I used an ordinary LF411 op-amp that's limited to about ±10 V and ±10 mA. Feel free to use a power op-amp for more oomph, but even these tiny currents can make small sparks jump between separate crystals!

Any two-channel oscilloscope that can produce an XY display will work. A 200-Hz triangle produces a solid trace without flicker or inductive effects from the clip leads connecting the circuit and the sample holder.

DIODE ACTION

Before trying any mineral samples, you should verify that your circuit

Photo 2—*The outer brass tube slides in the plastic holder with a spring for soft landings. The probe tips friction-fit in the end against the bottom taper pin. I cast a copper wire into the sample holder as an electrical contact.*

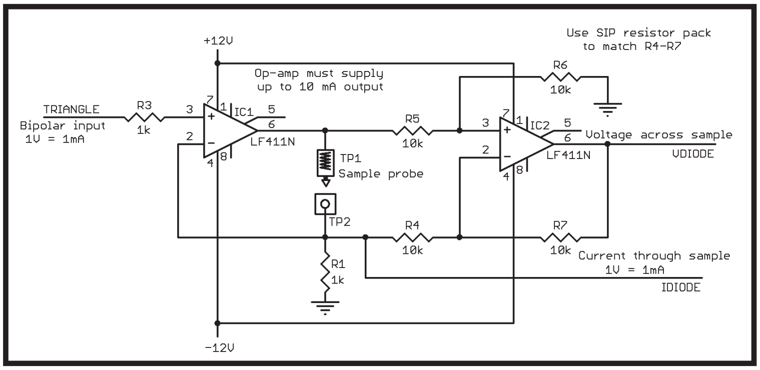


Figure 1—*This circuit applies a low-current bipolar signal to the crystal sample from a triangle-wave input. Connect the current output to an oscilloscope's X input, the voltage output to its Y input, and set the scope to 2 or 5 V per division for a reasonable trace size.*

works correctly with known components. Photo 3 shows a 1-kΩ resistor producing the expected 1 mA/1 V straight line and a small-signal silicon diode producing a nearly right angle curve at 500 mV.

The differential amplifier presents a 20-kΩ resistance across the sample probe because the LF411 maintains pins 2 and 3 at the same potential. The current through R4 and R5 masks very high sample resistances, such as the reverse-biased part of the silicon diode curve, by limiting the maximum slope. As you'll see, this isn't a problem for most of these DIY diodes.

My mineral grab bag included pyrite (FeS), chalcopyrite (CuFeS), gold ore, grey copper (Cu-As-S or Cu-Sb-S), and lead (which I think is actually galena, PbS), most of which displayed either simple linear resistance or no conductivity at all. Iron pyrite and chalcopyrite both had diode characteristics, with the former living up to its reputation as a good radio crystal.

Photo 4 shows the characteristics of three spots on the pyrite sample. I found that the probe's material had far less effect on the results than where it contacted the crystals, to the extent that any probe could have produced any of those curves. Although the preferred crystal-radio contact seems to be phosphor bronze wire, I think you can use (or at least try) nearly anything conductive.

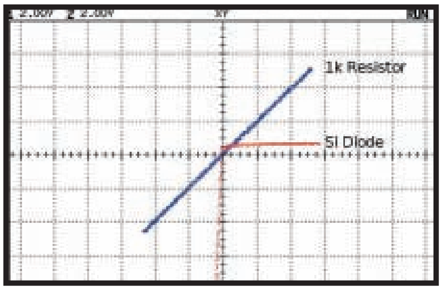


Photo 3—*A 1-kΩ resistor shows the expected 1 mA/V slope. A generic small-signal silicon diode has a nearly ideal right-angle characteristic, with the differential amp's input resistance putting a slight slope on the reverse-biased part of the curve.*

Compared to the silicon diode in Photo 3, iron pyrite isn't particularly good. However, the silicon diode requires 0.5 V of forward bias to begin conducting, while pyrite switches at almost 0 V. That critical difference explains why crystals work so well in low-voltage applications.

Some crystal radio designs use a resonant tank circuit to select a particular frequency, but the peak voltage from even a local transmitter generally won't exceed more than a few hundred millivolts. A simple Spice circuit (available in the download file on the *Circuit Cellar* FTP site) can show how diodes behave in those conditions with a 100%-modulated AM signal applied to a simulated diode and an RC filter mimicking a high-impedance earphone.

A reasonable diode model is a current-controlled switch with ON and OFF resistances taken from the sample's curves. For example, the green trace in Photo 4 has an ON resistance of:

$$2.6\ \text{k}\Omega = \frac{7.8\ V}{3\ \text{mA}}$$

and an OFF resistance of:

$$8.9\ \text{k}\Omega = \frac{-8\ \text{V}}{-0.9\ \text{mA}}$$

The red trace has resistances of 590 and 3.3 kΩ, making it a somewhat better diode. Achieving a lower ON resistance inevitably produced a lower OFF resistance, at least for the samples I measured.

Figure 2 (p. 58) shows that even relatively poor diodes do a reasonable job of demodulating the AM signal and that the "red trace" diode works slightly better. I tweaked the RC filter's capacitance to match each diode's characteristics, much as you would while optimizing a crystal radio's performance.

Surprisingly, the ragged violet trace is a standard Schottky diode. Even its forward drop simply isn't low enough for very small RF signals, so it does not demodulate the AM signal nearly as well as the crystals. As you might expect, a standard silicon diode is even worse: it produces an essentially flat line that's not shown here.

Now that you've seen a natural diode in action, you're ready to see some truly odd behavior.

NEGATIVE RESISTANCE

Negative resistance seems like an impossibility: the voltage across a resistor dropping as the current through it increases. Some materials behave exactly that way, at least for some values of voltage and current. For example, the voltage required to ignite an electric arc is much higher than its sustaining voltage, which is why gas-discharge tubes require a current-limiting ballast.

My chalcopyrite sample sometimes displayed a symmetric S-shaped curve with two negative resistance regions. While researching this effect, I found that overheated galvanized steel can have spots of a negative-resistance steel-zinc compound. Photo 5 (p. 58)

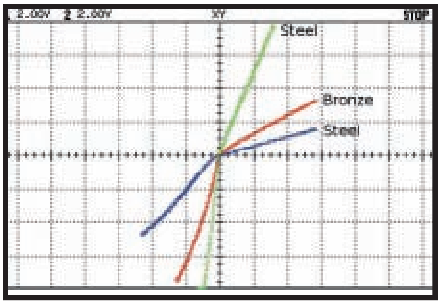


Photo 4—*Iron pyrite crystals exhibit fair to poor diode behavior. Their characteristics depend more on the crystal's surface properties than the probe material.*

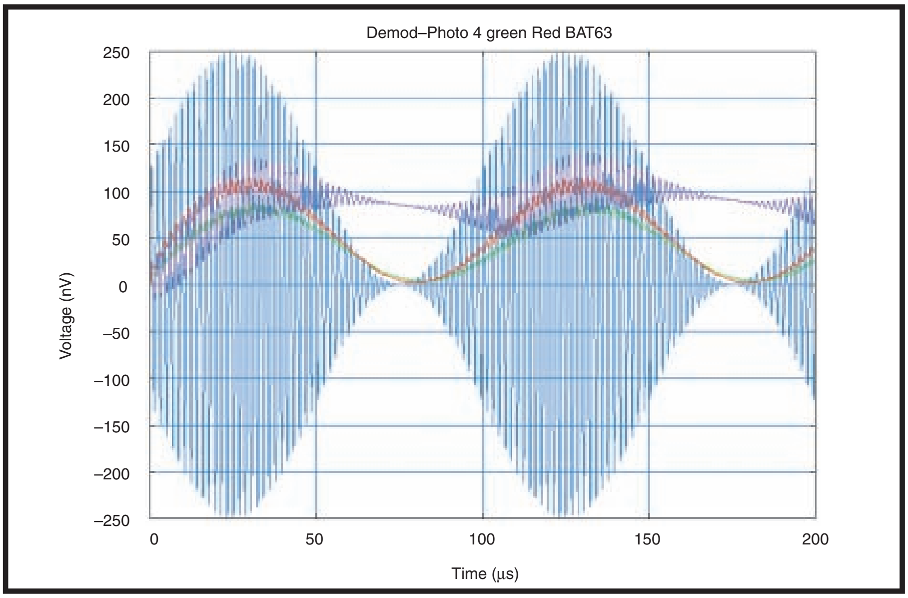


Figure 2—*Even a poor crystal diode can demodulate an AM broadcast-band signal. The red and green signals correspond to the pyrite diodes in Photo 4, violet is a Schottky diode, and blue is the incoming AM RF.*

shows three probe positions on such a sample.

The symmetric red trace in Photo 5 resembles my chalcopyrite sample, with two negative-resistance regions. As the current increases from zero, the voltage rises smoothly at 8 kΩ before peaking at 5 V. From there, the voltage drops as the current increases: negative resistance in action.

The blue trace in Photo 5 shows an asymmetric response, with normal (albeit nonlinear) positive resistance for positive current and negative resistance beyond about –2 mA. The green trace resembles the blue trace, but with a discontinuity at –2.2 mA where the voltage suddenly drops by nearly 2 V.

Notice that pyrite, chalcopyrite, and galvanized steel all include iron combined with other metallic or semimetallic elements. That combination typically requires high-temperature conditions, but you can convert plain old steel, which is mostly iron with a dash of carbon, into an electrically interesting substance without heat or dangerous chemistry. I put the bolt shown in Photo 1 head-down on a salt-water-saturated paper towel atop a copper sheet. A clip lead between the two completed a galvanic couple producing about 1 mA at 650 mV.

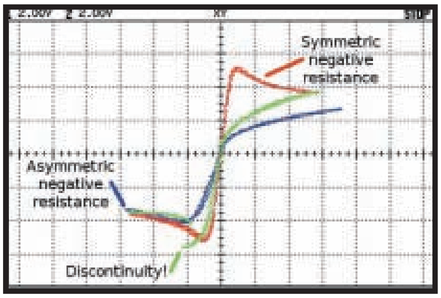


Photo 5—*Zinc alloy spots on galvanized steel can exhibit diode action, negative resistance, and abrupt voltage discontinuities.*

Although the bolt would rust in a few days, I added an AA cell to drive about 10 mA through the circuit and speed up the reaction. After 4 h, the bolt head sported a thick layer of corrosion that I air-dried using an LCD monitor's heat, and then mounted in an EMT coupling for probing.

The graceful blue negative-resistance trace in Photo 6 comes from that bolt. The red trace shows that corrosion can also behave as a nonlinear resistor. I didn't observe any pyrite-style diode behavior.

The negative resistance regions for both galvanized and plain steel samples certainly continue to much higher currents than available from the LF411 I used. Remember to keep the total power dissipated under the probe tip under control as you boost the current!

TROUBLE IN THE WILD

In my February 2002 column ("UHF Voice Radio," *Circuit Cellar* 139), I described nonlinear mixing with a simple diode circuit that produced a forest of mixing products from two input sine waves. I observed that such mixing also occurs in the corrosion sometimes found on antenna towers: the "rusty bolt" phenomenon. The combination of high RF power, sensitive receivers, galvanized steel structural components, and rainwater with the usual ionic contaminants can produce rather interesting results.

It might seem that even high signal levels couldn't push currents into the nonlinear parts of the curves in Photos 5 and 6, so they shouldn't produce strong mixing products. Unfortunately, corrosion isn't that simple.

Contemporary crystal radios sometimes include batteries that make modern diodes usable, a technique that piles anachronism atop anachronism. A few microamps will slightly forward-bias a silicon diode and dramatically improve its small-signal detection by shifting the operating point into the nonlinear region at the knee of the diode curve. That same principle also applies to naturally occurring diodes.

For example, just 300 µA of DC current through the probe tip producing the blue trace in Photo 6 would put a small AC signal near the nonlinear peak of the S-curve. Slightly more DC current would put the AC signal at the peak where negative resistance would produce some interesting effects.

Remember how that same bolt near a copper sheet in salt water pushed a milliamp through a wire? The exact voltage depends on the two materials and the available current depends on the surface area, but there's certainly enough power available to bias a natural diode.

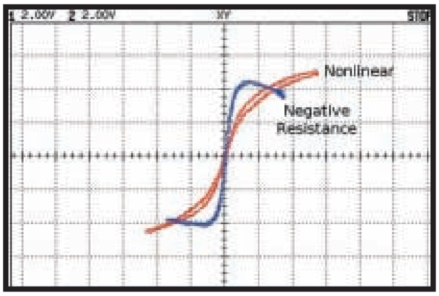


Photo 6—*Corroded steel also shows nonlinear behavior, as well as negative resistance. This is what a "rusty bolt" mixer on an antenna tower looks like!*

If you were depending on a known current to produce a specific bias, of course it wouldn't happen. Given enough time, however, corrosion will eventually create and bias a diode that mixes two RF signals together into a mixing product sitting squarely in the middle of your receiver's passband. You can depend on that!

CONTACT RELEASE

If you're inclined to experiment with negative resistance, a suitably tuned LC circuit with a resistor setting a DC load line intersecting the device's curve in two points will make a serviceable oscillator. The LC tank sets the frequency, and if you pay attention to details, the oscillation can reach microwave frequencies.

Nearly any probe material will work and you'll get good results if you have a stable, repeatable mechanical setup. That three-axis positioner is obviously overkill, but I had it in my collection.

I wrote a Kermit script to capture screenshots from my HP54602B scope's serial port and convert them to PNG format using the hp2xx utility, which eliminated the need to fire up Windows. ImageMagick composited the traces and The GIMP added callouts. Gnuplot converted the Spice simulator's results to a presentable graph.

Wood's Metal—a toxic alloy of cadmium, lead, tin, and bismuth—melts at 160°F. Field's Metal melts at a bit over 203°F by omitting the cadmium. Follow the usual rules for handling heavy metals: don't eat while you're working, and wash up afterward!

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 scripts, more photos, and Spice models, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2006/195.

RESOURCES

B. Bosch, "Crystal Set Analysis," www.oldradioworld.de/gollum/analysis.htm.

S. C. Dexter, "Galvanic Corosion," University of Delaware, 1999, www.ocean.udel.edu/seagrant/publications/corrosion.html.

hp2xx utility, www.gnu.org/software/hp2xx/.

Kermit, www.columbia.edu/kermit/index.html.

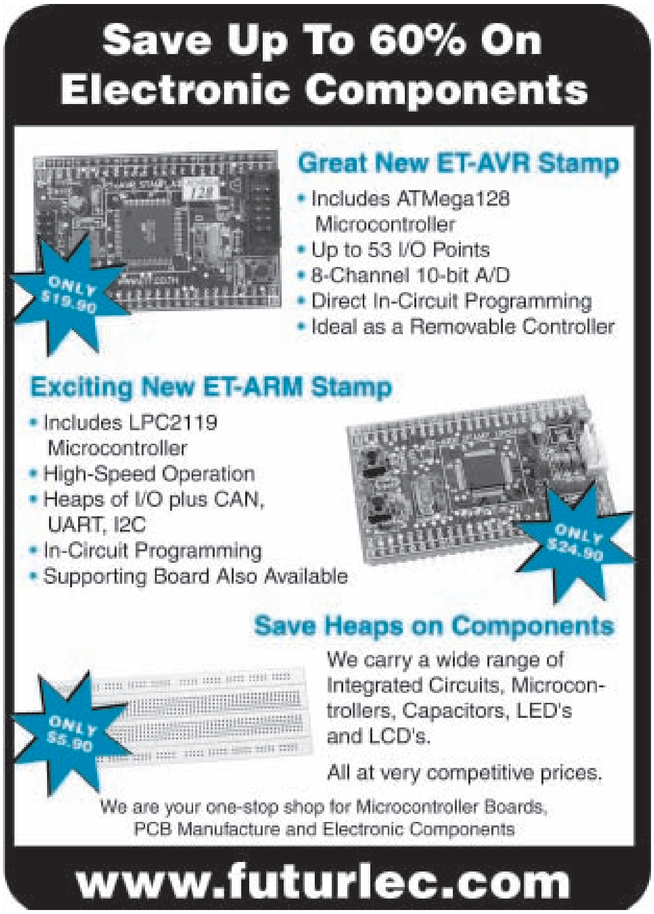
G. Steber, "I-V Curve Tracing with a PC," ARRL QEX, July/August 2006, www.arrl.org/qex/2006/07/qx7steber.pdf.

N. Steiner, "Zinc Negative Resistance Oscillator," http://home.earthlink.net/~lenyr/zincosc.htm.

SOURCES

Wood's metal, springs, and phosphor bronze sheet
Micro-Mark
http://micromark.com

Taper pins and small parts
S. LaRose, Inc.
www.slarose.com


Save Up To 60% On Electronic Components

Great New ET-AVR Stamp
- Includes ATMega128 Microcontroller
- Up to 53 I/O Points
- 8-Channel 10-bit A/D
- Direct In-Circuit Programming
- Ideal as a Removable Controller

ONLY $19.90

Exciting New ET-ARM Stamp
- Includes LPC2119 Microcontroller
- High-Speed Operation
- Heaps of I/O plus CAN, UART, I2C
- In-Circuit Programming
- Supporting Board Also Available

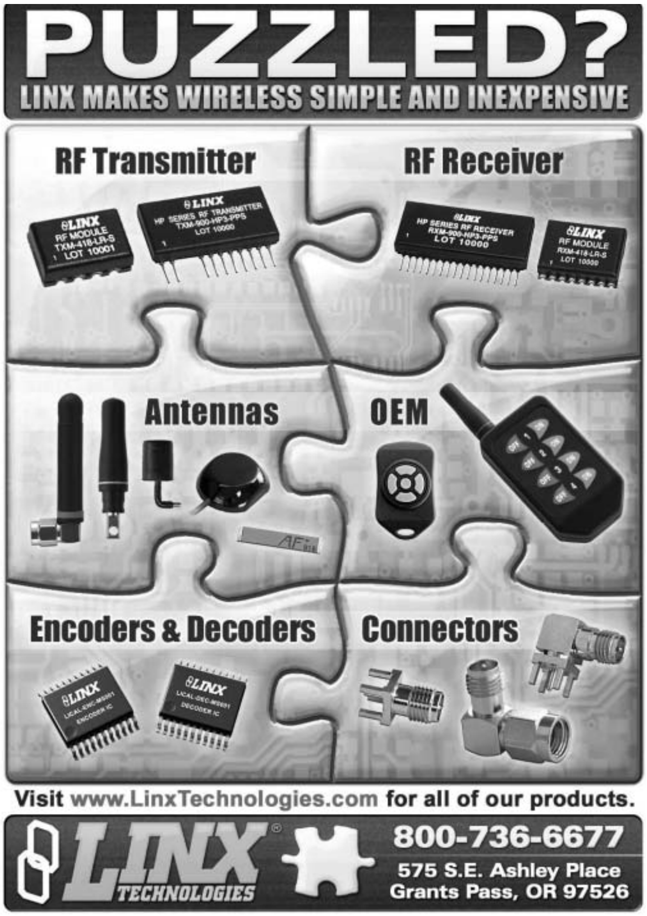
ONLY $24.90

Save Heaps on Components
We carry a wide range of Integrated Circuits, Microcontrollers, Capacitors, LED's and LCD's.
All at very competitive prices.

ONLY $5.90

We are your one-stop shop for Microcontroller Boards, PCB Manufacture and Electronic Components

www.futurlec.com


PUZZLED?
LINX MAKES WIRELESS SIMPLE AND INEXPENSIVE
RF Transmitter
RF Receiver
Antennas
OEM
Encoders & Decoders
Connectors
Visit www.LinxTechnologies.com for all of our products.
LINX TECHNOLOGIES
800-736-6677
575 S.E. Ashley Place
Grants Pass, OR 97526

FEATURE ARTICLE

by Eddie Insam

Power Over Ethernet Solutions

Powering devices over Ethernet cabling seems easy, but there's more to it than meets the eye. Eddie explains how it all works.

So you've designed a brand new Ethernet-based device. Perhaps it's a clock, a weather sensor, or an industrial controller device. You plan to hang it proudly on your wall and connect it to a RJ-45 wall socket. But how are you going to power it? Where will the system get its juice? Surely, you aren't going to disgrace your design with a brick wart. There must be a better way!

Why not feed power over the CAT-5 cable? Well, you're not the first person to consider this technique.

Standard CAT-5 cable has four pairs, and only two are used for data in a typical 10- or 100-Mbps installation (see Figure 1a). So, it sounds obvious to stick a few DC volts down the spare pairs. Oh, yes. But hang on, life is never so simple. This is technology, remember? There has to be a catch somewhere. So, sit down and relax, I have the story.

It may not come as a surprise that the wise men at the IEEE thought about this for a while and came up with a standard (IEEE 802.3af). This standard has been around since 1999, but progress has been relatively slow. It started to take off only recently, mainly because of the availability of inexpensive specialist components. Tom Cantrell and Jeff Bachiochi have covered some of the available components and modules (*Circuit Cellar* 165 and 187). A wide range of parts are now available, including dedicated switching transistors, isolation transformers, and high-quality nonsaturating magnetics, making power over Ethernet (PoE) a practical proposition.

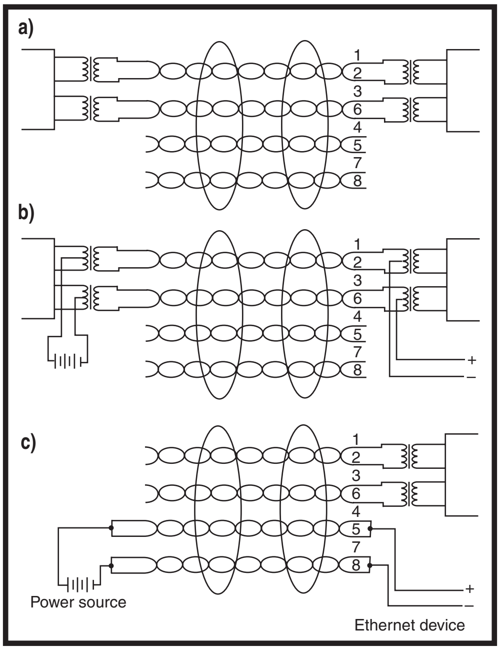


Figure 1—*Standard 10- and 100-Mbps Ethernet devices use just two of the four available pairs. The spare wires can be used to transmit power to the remote. Two possible methods are shown (**b** and **c**). But watch out! The power source must be smart enough to detect shorts and overloads and to avoid damaging components at the far end.*

TECHNICALITIES

The IEEE document covers two main methods for sending power down the CAT-5 wire. One involves using the spare pairs. The other involves sharing with the existing data lines using center-tapped transformers (see Figures 1b and 1c). The latter method is beneficial when spare cable capacity isn't available.

The method involving spare pins allows a decent amount of current to be drawn because the two spare pairs are paralleled together to increase capacity by reducing the total DC resistance. The present IEEE specifications allow up to 13 W of power to be transferred this way. This may not be enough for some heavy-duty devices, but it's quite acceptable for medium-size and small items such as TV cameras and VoIP phones. An updated PoePlus standard is currently being considered. This will allow for up to 30-W capacity, while still remaining backwards compatible.

Transmitting power with center-tapped transformers is more limited. Pulse transformers and other magnetics in the Ethernet controller must be designed to take the full DC power load current without saturating. That isn't an easy task for miniature surface-mounted components. The advantage of this alternative is that it leaves the extra pairs alone, an essential consideration in higher-speed gigabit Ethernet, which requires all four pairs to carry data.

POWER SUPPLY

Why can't you just stick any old power supply across the spare wires? Because you don't know what's at the remote end, and you may run the risk of blowing up sensitive equipment. If you don't believe me, take a look at Figure 2, which is a typical Ethernet terminator. This kind of circuitry is sometimes contained within a single metal enclosure called a MagJack. Note the two 50-Ω resistors R3 and R4 across the center taps of transformers T3 and T4. They are branched in series to form an effective 150-Ω DC

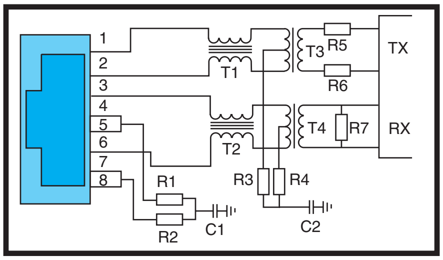


Figure 2—*This is a typical Ethernet termination. The resistors strapped to the spare data pins and center taps are there to balance the line and to reduce noise. They can quickly flash to smithereens in true Harry Potter style if any unmanaged DC power is placed on the cable.*

load across the input lines. Also note the two 50-Ω resistors R1 and R2 right across pins 7 and 8 and 4 and 5. These present a controlled impedance load to the otherwise non-terminated wires. They are there for robustness and noise reduction. This hookup is sometimes known as a Bob Smith termination.

If you connect a 48-VDC raw supply into such a socket, you will be driving a good third of an amp through these tiny resistors. This is guaranteed to vaporize them to kingdom come. Tiny SMD resistors are not built for such treatment.

Admittedly, some terminators and MagJacks have extra series capacitors to protect the resistors, and not all Ethernet devices use such extra networks. However, you don't want the power supply to blow the other devices that have them.

There are other potential problems that can be blamed on bad design or pure accident. For example, a wireman could accidentally short or swap the CAT-5 pairs. All possibilities have to be considered, and many are mentioned in a 1999 IEEE report entitled "DTE Power Problem Set and Solution Methodology."

Needless to say, the good people at the IEEE have devised cunning schemes to preempt the aforementioned challenges. In simple terms, the smart power supply can figure out what's happening at the load end. It does this by taking a number of graded impedance measurements before applying full power. These impedance signatures tell the supply whether or not it's safe to apply full power. Full power is applied only when it's safe to do so. Furthermore, the load is regularly monitored during normal operation to ensure nothing drastic has happened. This allows the supply to turn off the wick if it detects any suspicious problems, when the load fails, or when it is disconnected. This arrangement, of course, needs cooperating equipment at the load end to provide the right dummy impedances at the right time.

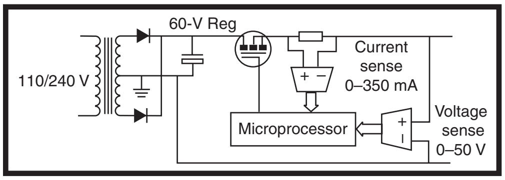


Figure 3—*A power supply will include a microcontroller in a standard design configuration to sense load current and generate output voltage levels accordingly.*

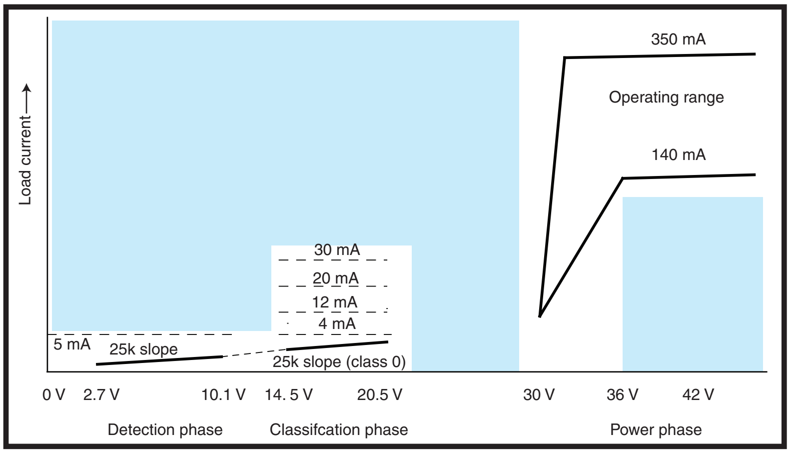


Figure 4—*There are three distinct phases. The simplest of loads will present a 24.5-kΩ resistance until the input voltage rises above 30 V, at which point the actual driven circuit will be switched into operation.*

Apart from the safety factor, the IEEE standard helps to reduce overall energy loss, because only those sockets that have a valid load can be programmed with power. During sensing, the supply knows the range of power loading taken by a load, and it ensures that the correct amount of current is delivered (within a reasonable range). No more, no less.

PC-CONTROLLED POWER?

So, does the power supply need to be computer controlled? Well, yes, but what isn't nowadays?

The operating algorithm is relatively straightforward, and even the tiniest microprocessor can handle it. You just need a power supply that can deliver a programmable voltage between 2 and 48 V, a means of sensing the load current, and a means of measuring its output voltage from which you can compute the load impedance and various other parameters. The rest is just software. Mind you, and as you would suspect, the IEEE standard is not that straightforward. Many options are included to cater for all eventualities. For example, there are options for sensing an AC load as well as the DC load, but many of these are just optional enhancements. You can get away with just sensing a plain DC resistive load. Figure 3 shows what the supply looks like.

How about the load end? The power source does its validation by sensing the impedance of the load at different source voltage levels. While this is taking place, the load needs to behave a bit like a nonlinear resistor, which is otherwise called a signature impedance (see Figure 4). The circuitry to do this is relatively simple, and there are a number of ICs that will do the job for you. The basic circuit is best described in terms of discrete components. Figure 5 (p. 62) shows the basic principle.

HOW IT WORKS

First, I need to introduce some jargon. Don't forget that I'm talking about IEEE standards, so the use of jaw-churning techno-speak is essential.

Power sourcing equipment (PSE) is a term for the source end, or power supply. Powered device (PD) is the equipment at the user end or load. An endpoint feed describes the arrangement or situation where the power supply is fitted inside the source box (e.g., inside an Ethernet router), so only one cable link is needed between the router and the PD.

A midspan feed unit (MFU) is a separate box that's added somewhere between the router and the PD to provide the power. This necessitates two CAT-5 links, one between the router and the MFU, and another between the MFU and the PD. You need to buy an MFU if you already have a router that doesn't provide PoE. If you start from scratch, you may prefer to buy a router with a built-in endpoint feed. Are you still with me? Don't go away. There's more.

The voltage level at the power supply is specified as between 44 and 57 V, whereas this

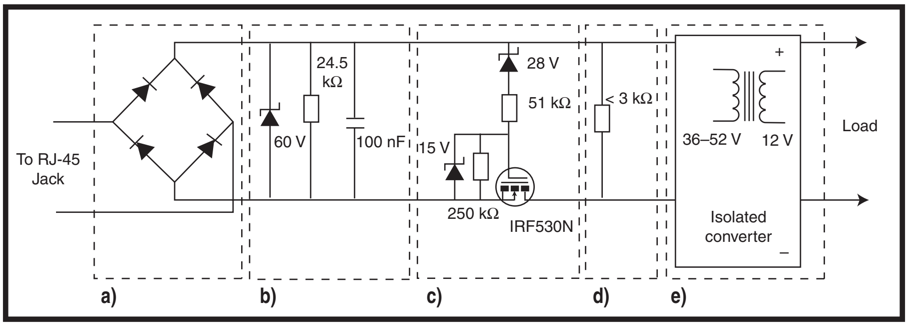


Figure 5—*Take a look at the operation of a typical PD in stages.*

is widened to 36 and 57 V at the user end to allow for reasonable ohmic drop down the CAT-5 cable. The PSE is allowed to supply up to 15.4 W of power with a maximum current limit of 350 mA. The maximum power consumption at the PD is about 13 W, which corresponds to a nominal current of 270 mA at 48 V. CAT-5 runs can be considerably long, and a lot of ohmic loss can be expected. This is one of the reasons why the standards suggest that pairs 4 and 5 and 7 and 8 should be paralleled together to halve the cable's resistance.

Although the specifications define which pin should be positive and negative, the load must not assume anything. Murphy's law! The PD must also ensure that the internal supply is floating with respect to the input power feed. So, it needs to include a bridge rectifier on the input plus a floating transformer-isolated power converter.

So, how does it work? Let's take it in stages. Take a look at Figure 4. When there is no load applied (i.e., when the user end PD is disconnected or during first power on), the source (PSE) repeatedly sits in a short loop sensing the line for an ohmic signature. This is the detection phase. It does this by placing at least two spot voltage levels between 2 and 10 V and then measuring the line currents drawn at these points. The current difference is taken rather than the absolute values because this makes for a more precise derivation of the signature impedance. It also compensates for fixed losses such as diode drops. A current limiter on the line ensures the load can draw no more than 5 mA just in case there is a short or similar problem.

The two test voltages are changed relatively slowly to avoid any glitches. The specifications suggest between 2 and 500 ms between readings. During the detection phase, the load has to present a 24.5-kΩ resistive component in parallel with a 0.1-µF capacitor. This is not a real component value; it's a theoretical average. You can't buy 24.5-kΩ resistors in the shops. To be more precise, any load between 23.75 and 26.25 kΩ is considered valid. Loads below 15 kΩ or above 33 kΩ are considered invalid. Loads outside of these two ranges are in no man's land and may or may not indicate the presence of a (possibly faulty) PD.

If this all sounds confusing, it's because this is the way standards tend to specify things that need to lie in ranges. Mere mortals like us need to know only that the resistance needs to be about 25 kΩ. The capacitor is required for an optional alternative AC load sensing method. I'll cover this later.

When the 24.5-kΩ resistor is detected, the PSE proceeds to the next stage: the classification phase. If at any point the load measures too low or too high, the PSE assumes there is no valid termination and removes the power altogether. It then waits a couple of seconds and then starts again from the detection phase, repeating the cycle forever. In the worst case, an incompatible or bad PD will see a maximum of 10 V or 5 mA applied across it and no harm will be done. This is somewhat more preferable than being hit with 48 V at full current!

The purpose of the classification phase is to determine the range of load currents the user device will need. In other words, the PD tells the PSE how much current it is going to need. The use of limited power ranges could be useful for loads that need critical monitoring or to avoid users connecting unauthorized devices to certain sockets. A main application for this is to allow limited resource PSEs to allocate different power levels to different outlets or to allow the PSE to enable only certain PDs in case of an emergency or other priority. In practice, however, this may create more problems than it can solve. Table 1 shows some of the available options.

During the classification phase, the PSE applies two or more voltages between 15.5 and 20.5 V (current limited to 100 mA) and measures the new signature impedance. The PD recognizes these new voltage levels and switches in a suitable load resistor according to its expected needs. Note that if the PD retains the original 24.5-kΩ resistor, it will be classified as Class 0 and default to full-power range, which is very convenient. In other words, the simple do-nothing option will give you the full power range. Who says committees never come up with practical ideas? The PSE will have a further chance of detecting improper loads or shorts during this stage. It will remove the power altogether if anything feels suspicious.

Class	Load by PD	Usage	Power range
0	0–4 mA	Default	0.44 to 12.95 W (full range)
1	9–12 mA	Optional	0.44 to 3.84 W
2	17–20 mA	Optional	3.84 to 6.49 W
3	26–30 mA	Optional	6.49 to 12.95 W

Table 1—*Take a look at the PD power classification scheme. This allows the supply to provide only as much power as the device demands.*

Having passed the classification phase, the PSE can now slowly ramp up to full power, so the voltage now goes up to the 48 V per 300 mA current limit. At the same time, the PD will connect the line to its internal circuits powering the user electronics. After this new stage and while providing full power, the PSE will constantly monitor the load for current drawn. The PD will guarantee to sink a defined maintain power signature (MPS). In other words, if the load current rises above 400 mA at any time or drops below 10 mA for than 250 ms, the PSE will assume the load has gone funny, kill the supply, and revert to its detection phase as before. There is a defined

You can start with U CAN

The unique fully functional 16-bit microcontroller
CAN Evaluation Tool

Smart, easy to use and excellent value

Key Features of the XC164CM U CAN START KIT:

- High-performance, real-time 16/32-bit MCU
- On-chip TwinCAN
- Single Cycle Instruction Execution
- 64 kBytes embedded Flash
- DSP support
- Enhanced power-saving modes

THE XC164CM U CAN START KIT is the only low-cost tool available with which you can test and evaluate not only the real-time embedded 16-bit XC164CM microcontroller but also your CAN program. It is a fully enabled CAN evaluation platform with a full-featured HITEX HiTop debugger and Tasking XC166 compiler. The TwinCAN feature of the XC164CM includes 2 CAN transceivers, allowing both CAN nodes to be accessed through a 10-pin header along with analog and digital I/O signals.

Simply plug the start kit into a USB port and install the GUI-driven CAN application program. Or U CAN verify and test your own CAN applications.

You can start with U CAN.

Order now at:

www.infineon.com/us-ucan

Infineon

Never stop thinking

back-off period of 2 s to avoid the entire thing going into wild oscillations.

A well-known scenario to be avoided is when a valid PD device has just been unplugged from an Ethernet wall socket and a legacy device is plugged in immediately after. If the PSE doesn't recognize this situation quickly, it can damage the legacy device because full power is still being applied to the line. This is where the alternative AC sensing method scores. A 500-Hz AC common-mode signal is superimposed on the DC. Any AC disconnection can be detected immediately, whereas a DC disconnection has to rely on slow voltage decays before it can be correctly detected. Note that the supply can optionally use either AC or DC sensing, but the load must include methods for supporting both. In practice, this is just a 100-nF capacitor in parallel with our beloved 24.5-kΩ resistor.

Photo 1—*The D-Link DWL-P50 is a ready-to-go module. Ethernet in, Ethernet out, and a choice between 12- and 5-VDC outputs.*

THE LOAD'S JOB

During initialization, the PD presents a variable impedance to the supply depending on the input voltage across its input pins. Between 0 and 10 V, the load looks like a 24.5-kΩ resistor (plus the voltage drop effects of the bridge rectifier). Between 10 and 20 V, it can still be a simple resistor, but it's calculated to give the current load specified in Table 1. Alternatively, it can keep the same 24.5 kΩ to respond for Class 0 and the full power range. As the input voltage ramps up between 30 and 42 V, the user load is switched in. If during full power the input falls below 36 V, the PD disconnects itself from the supply. This is known as under voltage lock out (UVLO).

It's the PD's responsibility to ensure that the load doesn't take more than the rated power or less than a minimum threshold current to make sure it doesn't get turned off. This minimum current is specified as 10 mA for at least 75 ms in every 325 ms. Unplugging the PD can then be easily recognized by the PSE as it sees the current drop below 10 mA.

The disadvantaged products in this scheme are low-power devices that need to include a bleed resistor just to ensure that the minimum current threshold is met. So much for energy conservation!

TYPICAL PD

Figure 5 shows a most basic PD. It has been divided into sections to show the relative responsibilities. Figure 5a shows a bridge rectifier. It's always good practice to use a bridge in case the wires have been swapped around. A PD can make use of both alternative sources by having two bridges, each connected to the two power options shown in Figures 1b and 1c.

Figure 5b shows the main 24.5-kΩ

Easy Embedded Linux
$169
Qty 1
16MB FLASH / 32MB RAM
200Mhz Arm9 CPU
16 Digital I/O
Watchdog
10/100 Ethernet
Audio In/Out
2 USB
2 Serial Ports
Battery backed Clock/Calendar

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

Circuit Specialists, Inc.
Programmable DC Power Supplies
•Up to 10 settings stored in memory
•Optional RS-232 interface
•Low output ripple & noise
•High resolution at 1mV

Model	CSI3644A	CSI3645A	CSI3646A
DC Voltage	0-18V	0-36V	0-72V
DC Current	5A	3A	1.5A
Power (max)	90W	108W	108W
Price	$199.00	$199.00	$199.00

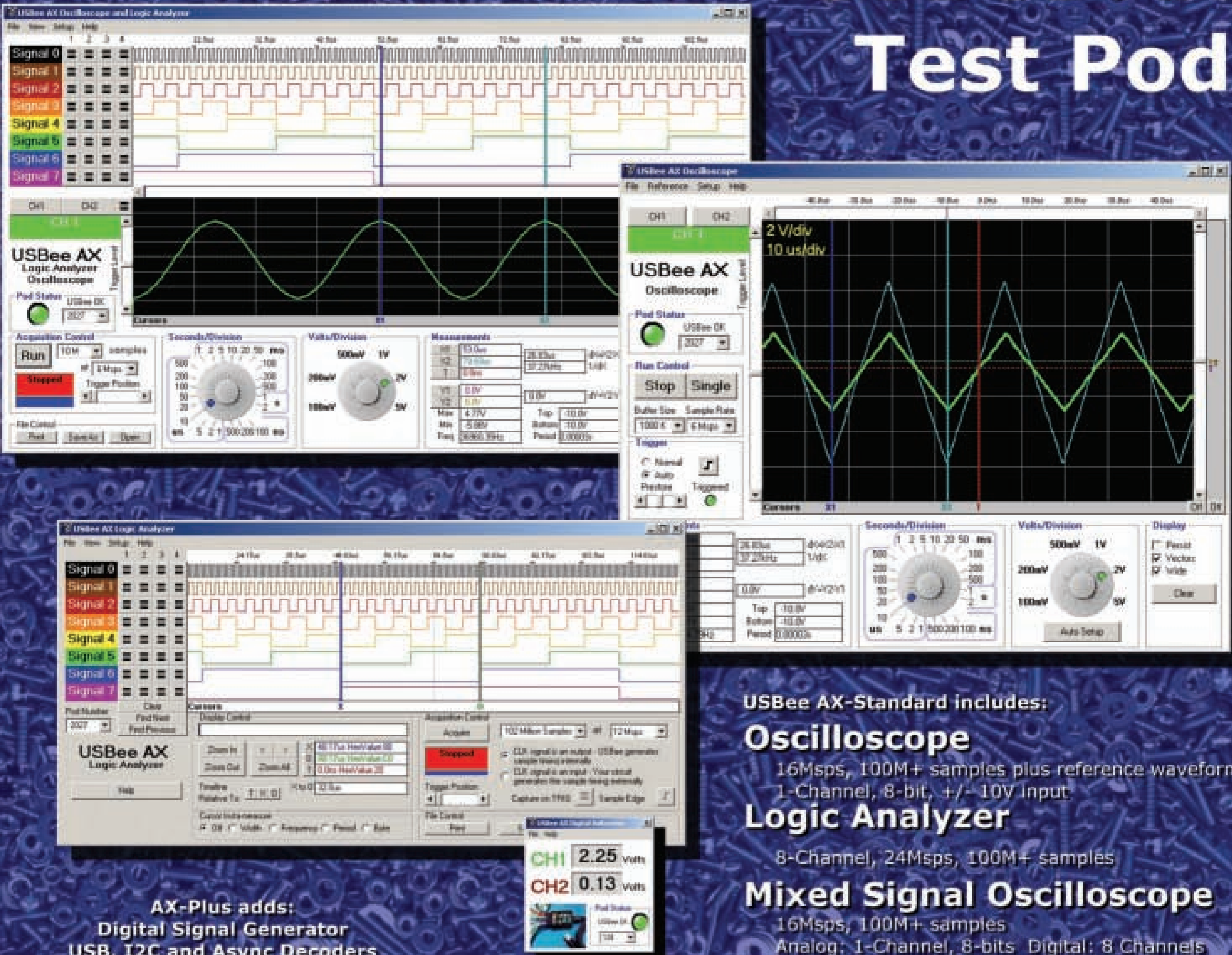
Full Details on Web Site!

INTRONIX LOGICPORT 34 CHANNEL LOGIC ANALYZER
34 Channel Logic Analyzer
500MHz Timing Mode Sample Rate
$389.00
Full Details on Web Site Item# LOGICPORT

A Sophisticated Scope Adaptor
200MHz USB DSO
$749.00
Full Details on web
Item#: 200DSO
A Price Breakthrough!

www.CircuitSpecialists.com
800-528-1417 / Fax:480-464-5824

USBee AX Test Pod



CH1 2.25 Volts
CH2 0.13 Volts

USBee AX-Standard includes:

Oscilloscope

16Msps, 100M+ samples plus reference waveform
1-Channel, 8-bit, +/- 10V input

Logic Analyzer

8-Channel, 24Msps, 100M+ samples

Mixed Signal Oscilloscope

16Msps, 100M+ samples
Analog: 1-Channel, 8-bits Digital: 8 Channels

Digital Voltmeter

AX-Plus adds:
Digital Signal Generator
USB, I2C and Async Decoders

AX-Pro adds:
Toolbuilder Source Code
plus 7 more debug tools

Data Extractors

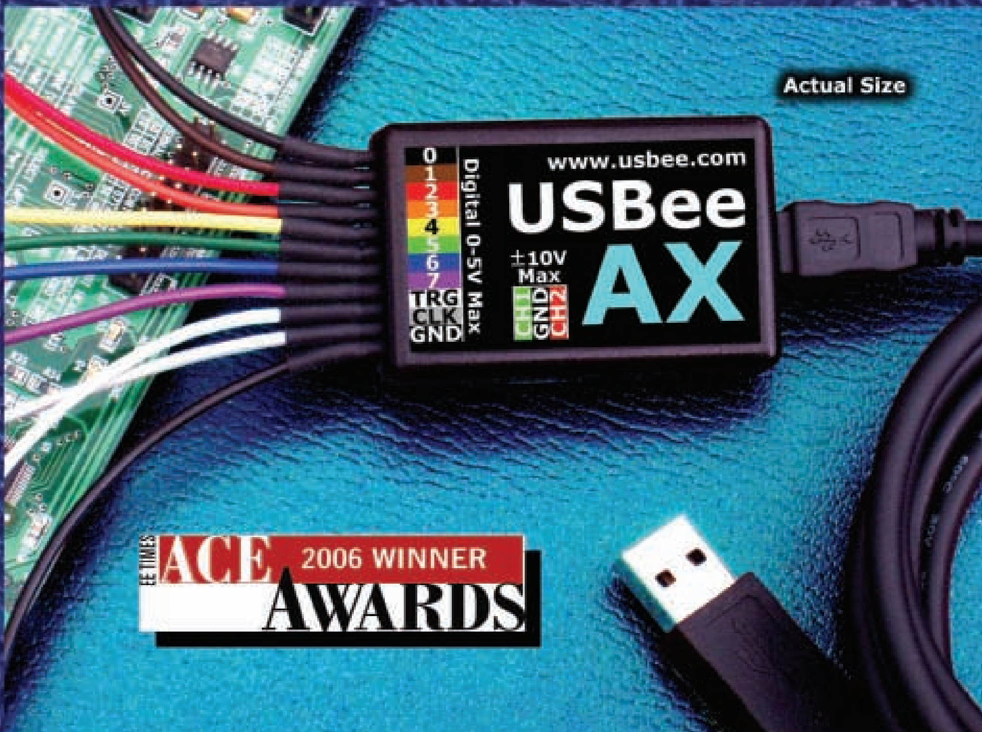
Optional Software Modules
for the USBee AX-Pro

Continuous Real-Time
Embedded Bus Data Streaming

Store Bus Data to Disk or
Send to Your Custom Application

Capture and Process
Entire Test Sequences

Parallel, Serial, I2C, USB, ASYNC,
CAN, SMBus, SPI, I2S, 1-Wire



USBee.com

You need these cutting edge products to quickly solve your toughest design and measurement problems!

New Products

Dualcore	dual ARM IC
Emulin	LINbus/J2602 emulator
WIZnet	TCP/IP IC
Hunt RTG	instant FPGA/DSP
Byteparadigm	USB 2.0 I/O
Systembase	Quad UART chip
Owasys	embedded wireless

HumidiProbe / PT104

HumidiProbe - Temperature and Humidity Logger for PC. **PT104** - High resolution and accuracy. Platinum sensors. USB powered - no power supply required. Free s/ware.

from $241/759

Touch-sensing ICs

Quantum ICs- World-beating capacitive sensor ICs for switching and control with patented features like Adjacent Key Suppression, Spreadspectrum, adjustable sensitivity. QRG ICs in use worldwide by the world's largest consumer/appliance manufacturers **<$1 (10K)**

Video Modem

VPORT - MPEG4 with Web browser/Ethernet interface and built-in motion detection. 352 x 240 color video streams at 30fps with 24 kbp audio coding.

96047627 $450

USB to 24 x I/O

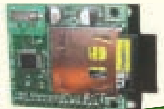
USBI/O24 - 24 latched I/O lines self-powered from your PC's USB port. Ubicom SX52 allows individual pin I/O programming. I/O control via PCs USB port.

USB I/O 24 from $69

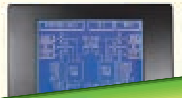
2/4ch 12-bit USB scope

PP298/PP300- High-perf. USB2.0 pwrd scop 100MS/s, +/-100mV t 50s/div C/ VEE drivers

FATfile Storage



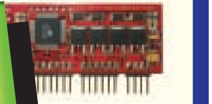
Touchpanel Controller



CAN-USB

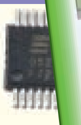
USBwiz

Stepper Motor



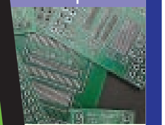
3000 - advanced stepper and control system for per motors up to 35V/3A ideal for robots/industrial ns you into an instant expert! **from $119**

USB



FT232R - NEW! with new "security Each IC has u EEPROM, clockg Smaller/cheaper **FT232RL**

B Adapters



M miniboards prints on either ur ultra fine pitch nts with more spaced holes. uts. **~$10.00**

Easy V

microVGA - graphics allows micros to dis graphics on any V Connect to any controller, embedded PC with a USB or Clever & simple to use.

Logging

cost, self- sed digital ty sensor. o/humidity USB port. **DLP-TH1 from $99.95**

nly $12.99

Cleverscope CS328

PC Oscilloscope, Logic Analyzer, and Signal Generator all-in-one!

Austronics AEEMA '2005 Best New Product' award!

Cleverscope CS328 - NEW 100MHz USB-connected mixed-signal 2-ch PC scope with 8 digital inputs and sig.gen. Huge 4MSample buffer for deep data drilling and features exceeding most expensive stand-alone scopes (sophisticated analog/digital signature/ pulsewidth triggering, glitch capture, whole buffer filestore, "copy/paste" results, etc.). Find hard-to-isolate waveforms! Blows other scope adapters away! Display math-modified and digital signals together. Join multiple units for more channels!

"CS328 is absolutely wonderful, I'm thrilled to bits with it. The software is light-years in front of that (other PC scope). The Math function makes me almost believe I'm using a LxCxxx." Roger – EFE Ltd

CS328 - usually $1045 - only $999 !

Cleverscope software resources include:

- Separate, freely moveable and resizeable windows to display the signal, a zoomed signal view, and the frequency spectrum of the signal, and control panel.
- Spectrum analysis with conditioning windows - log or linear.
- Signal averaging (exponential, block and peak hold) and LPF.
- Math calc including + - / * sqrt, integral, differential, and filtering.
- Signal msmt, including Pk-Pk, RMS, DC, pulse width, T and F.
- Copy and Paste graphic or data to other applications.
- Save and Open from disk.
- User defined units, signal names and scaling (offset and gain).
- Text annotation of each graph.
- Web server for remote viewing of LAN connected unit.

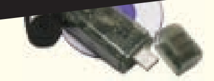
- 2 x 10-bit, self-calibrating A/D channels - 100 MSa/s simultaneously!
- 8 digital inputs - 100 MSa/s simultaneously!
- Huge 4M Samples storage per channel – drill deep!
- Simple "Copy and Paste" into other applications such as Word, Excel
- Highly flexible mixed-signal triggering, scope and spectrum display.
- View small ac signals on large ac or dc offset
- Built-in math capability - multiple display: A-B, invert, diff/integrals, etc.
- Includes all probes cables power supply, s/w, manuals.

- Optional Signal/sweep generator CS700 - +$199.00 - Sine, Square: 0.2Hz–10MHz; Triangle: 0.2Hz – 1MHz - Frequency Shift, Phase Shift diff., phase/gain - great for filter, amplifier testing!

Easy USB

microUSB - Tiny USB ready-made USB Interface to 1 Mbps 0.7" x 0.5" based on SiLabs CP2101. Gnd/RX/TX Suspend, Reset, & 3.3V 100mA source –40°C +85°C.

microUSB only $25



TrackStick™ - gumpack-sized logger tracks position data for shipments, vehicles, personnel, etc. Records date, time, location, speed, heading and altitude at pre-set intervals - shows route/speed on PC! **$249**

ANT8 Matchbox-sized USB-powered logic analyzer. Sample 8 channels at up to 500 million samples-per-sec. View traces on PC. Print or save for later review! (16-ch. ver. too!) **$222**

Tents

EMI shielding tents made from Ni/Ag fabric. -86dB 20MHz to 18GHz. Portable - tabletop to room/lab size for on-site, on-demand pre-compliance testing. External PVC or aluminum frame. **from $3500**

USB Bus Analyzers

Tracker 110/Explorer 200 high-perf. economical USB 1.1/2.0 protocol analyzers. **$999/$2999** **USBEX300** for debugging USB wireless Wimedia **<$29,500**

New - Bluetooth Sniffer <$10K

USB-Serial Adapters

Serial port for laptops without one! **USB-COM** instantly updates older RS232 products to USB! PC thinks it's using COMport up to 8 channel.

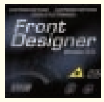
only $30 (1) or $25 (100)

I2C for PCs

PCI93LV: industry-standard I2C card for PCs. **WINI2C/PCI** software gives windows-interface to develop and debug I2C bus systems. **UCA93LV** is new USB version. ***NOW!*** - transparently monitor at 400kHz!

only $499

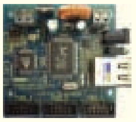
FrontDesign



FrontDesign - software for designing professional-looking front panels - with a host of drawing functions. Print, mill or engrave, or export final at 600 dpi.

FrontDesign only $79.95

Ether-IO



Ether-IO - UDP/IP-controlled 24 X digital I/O board. 3 x 8-bit TTL ports each independently programmable. Connects to any TCP/IP Ethernet network.

Ether-IO 24 from $99

Signal Wizard

Signal Wizard - easy-use real-time DSP-based filter board for audio bandwidth signals. Design filters in *seconds* without any DSP knowledge!

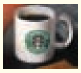
Signal Wizard II only $399

USB to I2C

USB I2C/IO - provides a simple "drop-in" solution for connecting your PC to 90kHz I2C + 20 I/O lines. Free software. Use multiple boards for more I2C/IO.

USB I2C/IO only $79

FREE COFFEE



Call 1-888-7SAELIG

Mention Offer# SBW

to get a free STARBUCKS Card with your order!

While supplies last - not available with any other offers

***Saelig Co.** searches the world to bring you unique, easy-to-use control and instrumentation products from overseas. Customers include: Intel, Philips, NEC, Kodak, Nokia, US Military, Microsoft, Dell, Xerox, Universities, T.I., Harris, Sony, J&J, Thomson, Sandisk, General Dynamics, H-P/Compaq, etc.*

Saelig .com
unique electronics
888-7SAELIG info@saelig.com

Also: Wireless Temp Loggers, Stepper Motor Controller, PC for harsh environments, Automotive Diagnostic Kit, PC for Healthcare, Wireless controller, Oscilloscope, JTAG testing, Indicator modules, Video Motion Detectors, Industrial IO modules, Software for soundcard - front panels - logic simulation and hardware control - CAD

If you don't see what you need maybe we can find it for you? - Ask for sales!

signature-sensing resistor and a 100-nF capacitor to provide an AC signature load. There is also a 60-V Zener diode to provide some sort of overall protection. (An extra fuse connected between this and the input line would not come amiss.) In this simplified circuit, the classification phase is also managed with the same 24.5-kΩ resistor, classifying the unit as Class 0.

Figure 5c is a simple gated switch that turns the load on when the input voltage reaches about 30 V. Figure 5d denotes that the load has to sink at least 10 mA. Figure 5e represents a 36- to 42-V converter, which must be floating (e.g., transformer isolated). Modules in the RECOM International Econoline series are typical examples. They are small potted modules (e.g., an RS4805 that takes 36 to 72 DC input and 5 V at 200-mA output all in a small SIL footprint).

TYPICAL PSE

Figure 3 shows a basic PSE design. Of course, you'll usually use a premade PSE rather than make your own.

The supply consists of a decent 48-VDC power supply and a series regulator controlled by the D/A output from a microprocessor (possibly via a PWM output). The series resistor emulates the current limit and a place to take a sample of the current drawn. One such controller is needed for each Ethernet line or RJ-45 outlet.

The design is pretty straightforward because accuracy is not primordial. One tricky part of the design is the wide-ranging metering of the output current, which needs to cover a range of 100 µA to more than 300 mA. This necessitates either a high-resolution (14-bit) ADC or a means of switching in different shunt resistors for the different ranges.

Note that the series pass transistor won't need much heat sinking. It will normally be operating either fully on (when delivering full power) or at limited current during the initialization phases. The software consists of a simple timed loop to cover the detection, classification, and power delivery phases one at a time. The IEEE 802.af document describes procedures for implementing a version of this flowchart if you have the time and inclination to decipher the gripping notation and methodologies used.

INTEGRATING PoE

Of course, you may not be interested in making your own circuits. There are plenty of ready-made chip and module solutions available out there to make it all easier. But understanding the principles involved will ensure that you won't get caught in many gotchas!

The MAX5940/1 was one of the first kids in the block. These chips provide all of the 802.3af interface detection, classification, and switching facilities. One of the chips is normally used in conjunction with a separate Maxim 48-V switching down regulator (MAX5014) to provide a complete power supply function.

National Semiconductor's LM5070, LM5071, and LM5072 are typical of the all-in-one-chip solutions. They integrate a current-mode DC-to-DC controller, user-programmable under-voltage threshold, a fault current control loop, and many other functions. The LM5071 and LM5072 can accept power from an external AC/DC adapter (a wall wart).

The Texas Instruments TPS2370, TPS23750, and TPS23770 are also big contenders. They combine the functionality of the older TPS2375 controllers and need a minimum number of external components. Similar devices are also available from Linear Technology (LTC4257) and Supertex (HV110).

Chip solutions are also available for the PSE end. Some of these have multiple controllers, which allow four, eight, or even 12 power supply controllers from one chip. Current devices are the Maxim MAX5945, the Texas Instruments TPS2383, the Linear Technology LTC4258, and the PowerDsine PD640*xx* series. For instant satisfaction, check out the PowerDsine 3001 (a single port midspan supply) and the corresponding D-Link DWL-P50 end load adapter, which are considered complete modules. The latter comprises a floating supply that can generate either 5 or 12 VDC at the flick of a switch (see Photo 1, p. 64). This pair can provide a relatively inexpensive solution for small PoE needs. Similar products are also available from suppliers such as Hyperlink Technologies.

You're sure to see many more PoE solutions in VoIP phones, CCTV cameras, and industrial Ethernet applications. Integrating a PoE supply into a module will be commonplace. ◼

Eddie Insam (edinsam@eix.co.uk) lives next to the Thames in southern England. He has been designing specialist signal processing and telecom systems for more than 20 years.

RESOURCES

J. Bachiochi, "Power Over Ethernet Primer," *Circuit Cellar* 187, February 2006.

T. Cantrell, "Powered Points," *Circuit Cellar* 165, April 2004.

IEEE, *IEEE Standards Interpretation for IEEE Std 802.3af-2003*, IEEE, New York, NY, 2005.

Maxim Integrated Products, "MAX5941A/MAX5941B: IEEE 802.3af-Compliant Power-Over-Ethernet Interface/PWM Controller for Power Devices," 19-3069, rev. 4, 2006, http://pdfserv.maxim-ic.com/en/ds/MAX5941A-MAX5941B.pdf.

M. McCormack, "DTE Power Problem Set and Methodology," 1999, www.ieee802.org/3/power_study/public/nov99/mccormack_1_1199.pdf.

SOURCES

DWL-P50 PoE Adapter
D-Link Corp.
www.dlink.com

LTC4257 PoE Interface controller
Linear Technology Corp.
www.linear.com

LM5070/71 PoE PD Interface and PWM controller
National Semiconductors
www.national.com/pf/LM/LM5070.html

Power modules
Recom International Power
www.recom-international.com

HV110 PD Controller IC
Supertex, Inc.
www.supertex.com

TPS2370 Power interface switch
Texas Instruments, Inc.
www.ti.com

APPLIED PCs

by Fred Eady

Advanced USB Protocol Sniffer

If you're serious about USB development, you need the right tools for designing and debugging. One such tool is the Ellisys USB Explorer 200, which is a high-speed USB 2.0 protocol analyzer that Fred recently put to work.

I love my job. Via e-mail and sometimes live by telephone, I get to converse with some very interesting people from all over the world, most of whom are avid *Circuit Cellar* readers. My job with the magazine also allows me to tinker in an unlimited fashion. If I let the magic smoke out of a gadget, I get to tell you about it in an effort to help you avoid going down the same smoky hallway. One of the greatest thrills associated with my job is opening up new equipment without a clue of how it works (or even if it will work at all) and then writing about my experiences.

If you're a regular *Circuit Cellar* reader, you know that Ethernet and communications protocols are near and dear to my heart. You also know that when it comes to the subject of USB, I'm one of Monty Python's horseless cavalry Grail seekers. Well, this month, I'll describe the box full of USB gadgets that I recently explored. So, mount up ye merry ladies and gentlemen. Fair warning: I'm currently under the care of a USB doctor named Bob, and I may need to make an office visit before this is all over.

WHAT'S IN THE BOX?

Imagine how boring life would be if you knew everything. For instance, you wouldn't need *Circuit Cellar* (or any other printed material for that matter) because you'd already know every minute detail about all of the subject matter covered in the magazine. And you definitely wouldn't be thrilled about opening up the box full of USB stuff either. After all, you would already know about how it works right down to the traces on the PCB.

So, if you know everything, you can skip to the last sentence of this column or take a nap. You won't miss a thing either way. But if you don't know everything, you should pull up a stool.

I didn't know everything when I started this project, and I was kind of looking forward to applying power to all of the USB gadgets I had in a box in the Florida room. One box contained the latest version of the Ellisys USB Explorer 200 protocol analyzer. Sniffers, or datastream monitors, are essential tools for engineering communications hardware and firmware. I love sniffers, so I knew that taking a close look at the new Ellisys USB sniffer was going to be fun.

With the USB Explorer 200 in hand, all I needed was an embedded USB device to sniff around. You know, kind of like a dog and a fire hydrant. The fire hydrant I chose for this project was a Microchip Technology PICDEM-FS USB demonstration board. The USB Explorer 200 can do every speed of USB, so I knew that incorporating the full-speed-capable demonstration board into the mix would bring full-speed embedded USB to the Florida room's bench.

You should never go on an extended excursion without your prescribed medications. So, to keep the USB pain from becoming a bit more intense than I could handle, I brought along some of Dr. Bob's USB medicine to be safe. After all, I didn't know what I'd find on the USB firmware end of the stick.

USB EXPLORER 200

The way a piece of test gear is packaged says a lot about its quality. The USB Explorer 200 is built like a tank. The unit is housed in a nice-looking, heavy-duty aluminum case. All of the major USB sniffing is done via a software driver package and the electronic components housed within the aluminum cabinet. So, as you can see in Photo 1, there's no need for a fancy hardware user control panel.

A look at the rear panel of the USB Explorer 200 exposes only a USB analysis data port and a BNC trigger connector. The analysis port connects the USB Explorer 200 to the analysis computer's high-speed or full-speed USB port. The trigger port hardware is bidirectional, and it's only found on the USB Explorer 200 Professional Edition hardware, which I had.

Having access to the USB Explorer 200 hardware trigger

Photo 1—*This box works as well as it looks. The data captured by the hardware contained in this box can be displayed, analyzed, and recorded at your whim.*

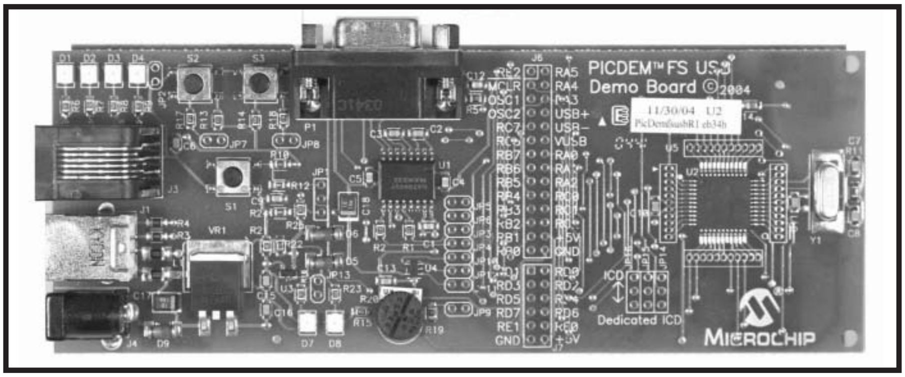


Photo 2—*Here's Microchip's answer to embedded USB. The PIC18F4550 silicon is everything you need to implement an embedded full-speed USB interface.*

allows you to trigger actions based on external events. Trigger pulses can also be selectively generated in reaction to a number of USB datastream events. The front panel trigger LED indicator turns green for an input trigger event. It turns red when an output trigger event is detected.

The USB Explorer 200's analysis USB port is rated for 480 Mbps, which equates to high-speed USB 2.0 operation. The USB Explorer 200's front panel link-under-test connectors use automatic link speed discovery that support low-speed (1.5 Mbps), full-speed (12 Mbps), and high-speed (480 Mbps) operation. Captured USB data is fed directly into the USB Explorer 200's 32 MB of FIFO memory with 16.67-ns resolution. Incoming USB data is shifted through the FIFO and stored in the host PC's system RAM. So, the maximum amount of stored USB data that can be analyzed or recorded and stored is determined by the amount of available RAM in the host PC. The captured USB data can consist of standard data packets and bus states. The USB Explorer 200 also logs low-level errors such as bit stuffing, CR5, and CRC16 errors. Basically, if the situation can happen inside a USB datastream, the USB Explorer 200 can capture it for analysis.

You'd probably expect to pour a wall wart out of a box containing test and development gear. Not so with the USB Explorer 200, which is powered directly from the analysis computer's USB port.

I also really like the method used to upgrade the USB Explorer 200's hardware. All you have to do is download the latest version of the analyzer's code from Ellisys's web site and execute the downloaded file. The USB Explorer 200's decoding engine and firmware are both automatically updated in this way.

PICDEM-FS

The PICDEM-FS USB board's reason for being is to introduce you to Microchip's PIC18F4550 USB-equipped microcontroller. The PIC18F4550 is the big daddy of the PIC18F2455/2550/4455/4550 family of USB-equipped PIC microcontrollers. Each PIC is USB 2.0 compliant. The PIC18F4550 can speed along at 12 Mpbs or idle in low-speed mode at 1.5 Mbps. Everything that a USB node needs is found in the PIC18F4550 silicon. There's even an interface for an off-chip USB transceiver.

Microchip provides reference designs and example applications to help you along. Microchip's USB Firmware Framework is the basis of all the USB firmware stuff included in the PICDEM-FS USB demonstration board's box. The USB Firmware Framework is a set of services that handles things like lower-level USB protocol management, USB interrupt handling, and hardware register control. In addition to the PIC18F4550 firmware, Microchip's general-purpose USB Windows driver is also part of the PICDEM-FS USB package. The Windows driver allows a PC application to communicate directly with a device's endpoints. The MPUSBAPI library, which is also included, can be used to provide programming interfaces to custom PIC18F4550 USB applications.

Honestly, I wasn't really interested in the inner workings of the PIC18F4550. I simply wanted the PICDEM-FS to act as a sniffable object for the USB Explorer 200. Fortunately, I could use the demonstration board's preloaded demonstration firmware in conjunction with the board's personal computer interface application to alter its USB states and functionality. With a little luck, I was able to take the PICDEM-FS through the various USB states of awareness and capture all of the activity with the USB Explorer 200.

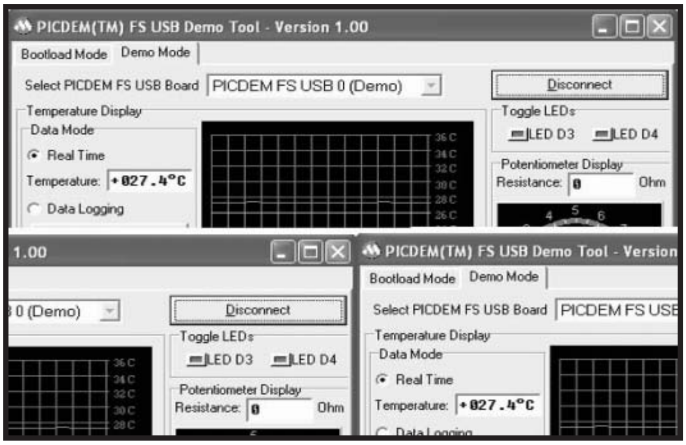
The coolest part of all was that I was able to figure out what was going on without having to direct my eyes to Area 9 of the USB specification. If you haven't seen a noncommercial shot of the PICDEM-FS USB demonstration board, take a look at Photo 2.

THE HOOK UP

The USB Explorer 200 is designed to be used with an analysis computer and a test computer. The former, which runs the USB Explorer 200 application, is responsible for collecting, displaying, and recording the captured USB data.

The analysis and test computers work in conjunction with the USB Explorer 200 to target a third device in the chain, which is the actual device under test. The test computer acts as the host for the device under test, which is the PICDEM-FS USB board in this case. All of the pertinent driver and configuration data for the board is contained in the test computer.

My configuration consists of a standard Pentium 4 desktop in the test computer role. A laptop with a high-speed USB port acts as the analysis computer. Because the PICDEM-FS USB board's fastest USB mode is high speed, the laptop's high-speed USB port works well as the analysis USB portal. The analysis USB port on the USB Explorer 200 mates to a standard type B USB connector with the opposite end of the analysis computer USB cable terminating with a type A USB connector at the laptop USB portal.



Select PICDEM FS USB Board: PICDEM FS USB 0 (Demo)

Temperature: +027.4°C

Resistance: 0 Ohm

Photo 3—*This is a view into the application that communicates with the PICDEM-FS USB demonstration board. Hardware events generated by the board are displayed here. The demonstration application can cause things to happen on the PICDEM-FS USB board.*

This is standard stuff in the USB scheme.

The test computer is connected to the USB Explorer 200's link-under-test type B USB connector in an identical manner. The PICDEM-FS board is connected to the USB Explorer 200's link-under-test type A USB port with a type B USB connector acting as a termination point at the PICDEM-FS USB board. This lashup inserts the USB Explorer 200 as a nonintrusive listening device in the test computer-to-PICDEM-FS USB board datastream.

The PICDEM-FS USB board uses a pair of LEDs to indicate the state of the USB connection. Its indicators flash in differing patterns to signal detached, attached, powered, addressed, configured, and suspended states. I captured these bus states with a USB Explorer 200 trace.

Let's begin by establishing a baseline for the USB trace captures. Photo 3 is a snapshot of the PICDEM-FS board's demonstration application window running on the test computer. The board is equipped with a temperature sensor, two user-accessible LEDs, and a 10-kΩ potentiometer.

I extinguished the user LEDs and zeroed the 10-kΩ potentiometer. A mouse click changed the status of the LEDs during a capture. I flipped the potentiometer around to change the resistance data being sent across the PICDEM-FS USB board-to-test computer USB connection. The temperature hovered around 27°C. I used my body heat to change the temperature reading during a capture. I began the capture with the board detached. I started the USB Explorer 200 capture process before attaching the PICDEM-FS board to the USB Explorer 200's link-under-test USB type A portal.

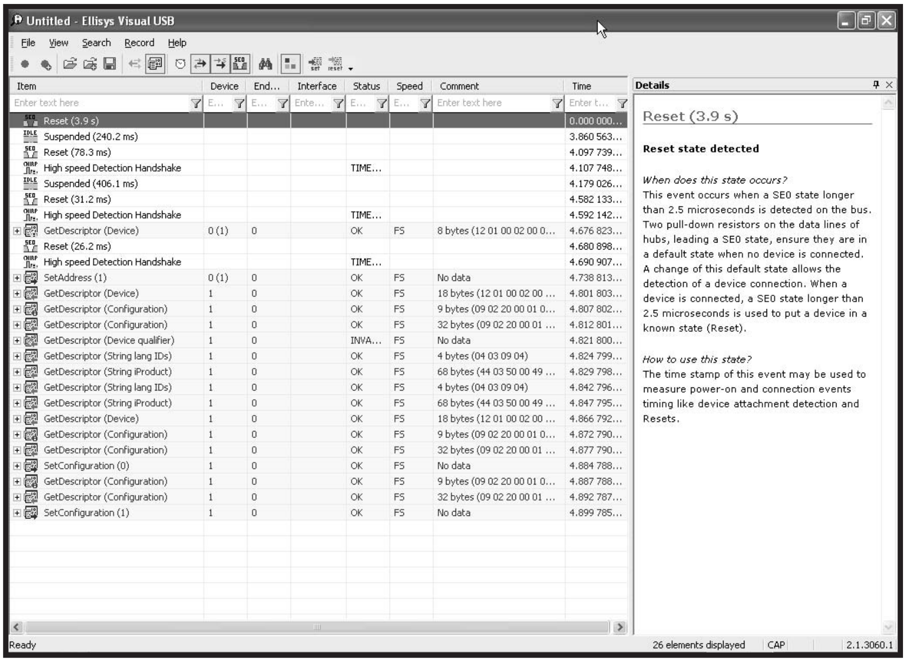
Photo 4 is a trace taken at the PICDEM-FS USB board attachment. The first trace elements describe the USB bus's state. Once attached, the board resets and enters a suspended state. Another reset is then issued and the USB Explorer 200 detects the board's speed. The trace entries that follow reflect actions that you'd expect a well-behaving USB device to initiate and execute. Addressing information is established and various types of USB descriptor information are exchanged. The entire data exchange process ends when the configuration is set.

At this point, I haven't connected the PICDEM-FS USB board to the PC demo application. True to the trace, the board's pair of USB device state status LEDs is flashing. This indicates that the board is in a configured mode.

USE YOUR INTUITION

Let's pretend that you don't have a copy of the USB demonstration C source code and that you want to determine the makeup of the datastream components that control the user LEDs, report the temperature, and indicate the current resistance of the 10-kΩ potentiometer. You can't consult the code for these datastream values, but you can trace the USB link.

Finding the temperature data is low-hanging fruit because the temperature doesn't change rapidly and should produce a repeating datastream that bobbles every now and then. The first test that I performed involved putting my big fat finger on the PICDEM-FS USB board's temperature sensor. But before I warmed up the temperature sensor, I baselined by



Item	Device	End...	Interface	Status	Speed	Comment	Time
Reset (3.9 s)							0.000 000...
Suspended (240.2 ms)							3.860 563...
Reset (78.3 ms)							4.097 739...
High speed Detection Handshake				TIME...			4.107 748...
Suspended (406.1 ms)							4.179 026...
Reset (31.2 ms)							4.582 133...
High speed Detection Handshake				TIME...			4.592 142...
GetDescriptor (Device)	0 (1)	0		OK	FS	8 bytes (12 01 00 02 00 0...	4.676 823...
Reset (26.2 ms)							4.680 898...
High speed Detection Handshake				TIME...			4.690 907...
SetAddress (1)	0 (1)	0		OK	FS	No data	4.738 813...
GetDescriptor (Device)	1	0		OK	FS	18 bytes (12 01 00 02 00 ...	4.801 803...
GetDescriptor (Configuration)	1	0		OK	FS	9 bytes (09 02 20 00 01 0...	4.807 802...
GetDescriptor (Configuration)	1	0		OK	FS	32 bytes (09 02 20 00 01 ...	4.812 801...
GetDescriptor (Device qualifier)	1	0		INVA...	FS	No data	4.821 800...
GetDescriptor (String lang IDs)	1	0		OK	FS	4 bytes (04 03 09 04)	4.824 799...
GetDescriptor (String iProduct)	1	0		OK	FS	68 bytes (44 03 50 00 49 ...	4.829 798...
GetDescriptor (String lang IDs)	1	0		OK	FS	4 bytes (04 03 09 04)	4.842 796...
GetDescriptor (String iProduct)	1	0		OK	FS	68 bytes (44 03 50 00 49 ...	4.847 795...
GetDescriptor (Device)	1	0		OK	FS	18 bytes (12 01 00 02 00 ...	4.866 792...
GetDescriptor (Configuration)	1	0		OK	FS	9 bytes (09 02 20 00 01 0...	4.872 790...
GetDescriptor (Configuration)	1	0		OK	FS	32 bytes (09 02 20 00 01 ...	4.877 790...
SetConfiguration (0)	1	0		OK	FS	No data	4.884 788...
GetDescriptor (Configuration)	1	0		OK	FS	9 bytes (09 02 20 00 01 0...	4.887 788...
GetDescriptor (Configuration)	1	0		OK	FS	32 bytes (09 02 20 00 01 ...	4.892 787...
SetConfiguration (1)	1	0		OK	FS	No data	4.899 785...

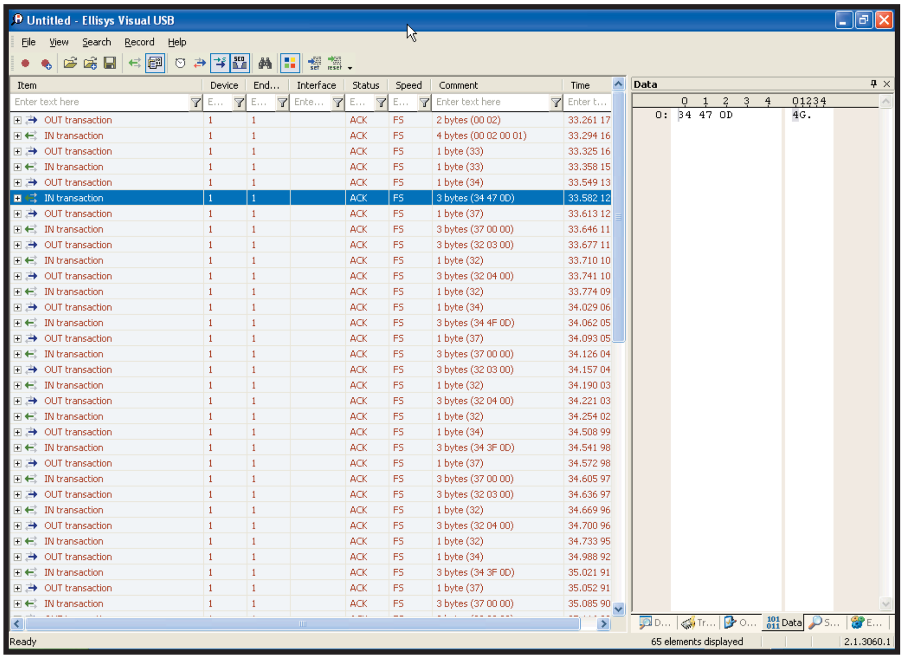
Reset (3.9 s)

Reset state detected

When does this state occurs?
This event occurs when a SE0 state longer than 2.5 microseconds is detected on the bus. Two pull-down resistors on the data lines of hubs, leading a SE0 state, ensure they are in a default state when no device is connected. A change of this default state allows the detection of a device connection. When a device is connected, a SE0 state longer than 2.5 microseconds is used to put a device in a known state (Reset).

How to use this state?
The time stamp of this event may be used to measure power-on and connection events timing like device attachment detection and Resets.

Photo 4—*A sequence of events readied the PICDEM-FS USB board to communicate with the PC demonstration application. Each of the GetDescriptor trace entries can be expanded. The Ellisys USB Explorer 200 decoding engine decodes them.*



Item	Device	End...	Interface	Status	Speed	Comment	Time
Enter text here	E...	E...	Ente...	E...	E...	Enter text here	Enter t...
OUT transaction	1	1		ACK	FS	2 bytes (00 02)	33.261 17
IN transaction	1	1		ACK	FS	4 bytes (00 02 00 01)	33.294 16
OUT transaction	1	1		ACK	FS	1 byte (33)	33.325 16
IN transaction	1	1		ACK	FS	1 byte (33)	33.358 15
OUT transaction	1	1		ACK	FS	1 byte (34)	33.549 13
IN transaction	1	1		ACK	FS	3 bytes (34 47 0D)	33.582 12
OUT transaction	1	1		ACK	FS	1 byte (37)	33.613 12
IN transaction	1	1		ACK	FS	3 bytes (37 00 00)	33.646 11
OUT transaction	1	1		ACK	FS	3 bytes (32 03 00)	33.677 11
IN transaction	1	1		ACK	FS	1 byte (32)	33.710 10
OUT transaction	1	1		ACK	FS	3 bytes (32 04 00)	33.741 10
IN transaction	1	1		ACK	FS	1 byte (32)	33.774 09
OUT transaction	1	1		ACK	FS	1 byte (34)	34.029 06
IN transaction	1	1		ACK	FS	3 bytes (34 4F 0D)	34.062 05
OUT transaction	1	1		ACK	FS	1 byte (37)	34.093 05
IN transaction	1	1		ACK	FS	3 bytes (37 00 00)	34.126 04
OUT transaction	1	1		ACK	FS	3 bytes (32 03 00)	34.157 04
IN transaction	1	1		ACK	FS	1 byte (32)	34.190 03
OUT transaction	1	1		ACK	FS	3 bytes (32 04 00)	34.221 03
IN transaction	1	1		ACK	FS	1 byte (32)	34.254 02
OUT transaction	1	1		ACK	FS	1 byte (34)	34.508 99
IN transaction	1	1		ACK	FS	3 bytes (34 3F 0D)	34.541 98
OUT transaction	1	1		ACK	FS	1 byte (37)	34.572 98
IN transaction	1	1		ACK	FS	3 bytes (37 00 00)	34.605 97
OUT transaction	1	1		ACK	FS	3 bytes (32 03 00)	34.636 97
IN transaction	1	1		ACK	FS	1 byte (32)	34.669 96
OUT transaction	1	1		ACK	FS	3 bytes (32 04 00)	34.700 96
IN transaction	1	1		ACK	FS	1 byte (32)	34.733 95
OUT transaction	1	1		ACK	FS	1 byte (34)	34.988 92
IN transaction	1	1		ACK	FS	3 bytes (34 3F 0D)	35.021 91
OUT transaction	1	1		ACK	FS	1 byte (37)	35.052 91
IN transaction	1	1		ACK	FS	3 bytes (37 00 00)	35.085 90

Data

```
       0  1  2  3  4    01234
   0: 34 47 0D          4G.
```

Ready

65 elements displayed

2.1.3060.1

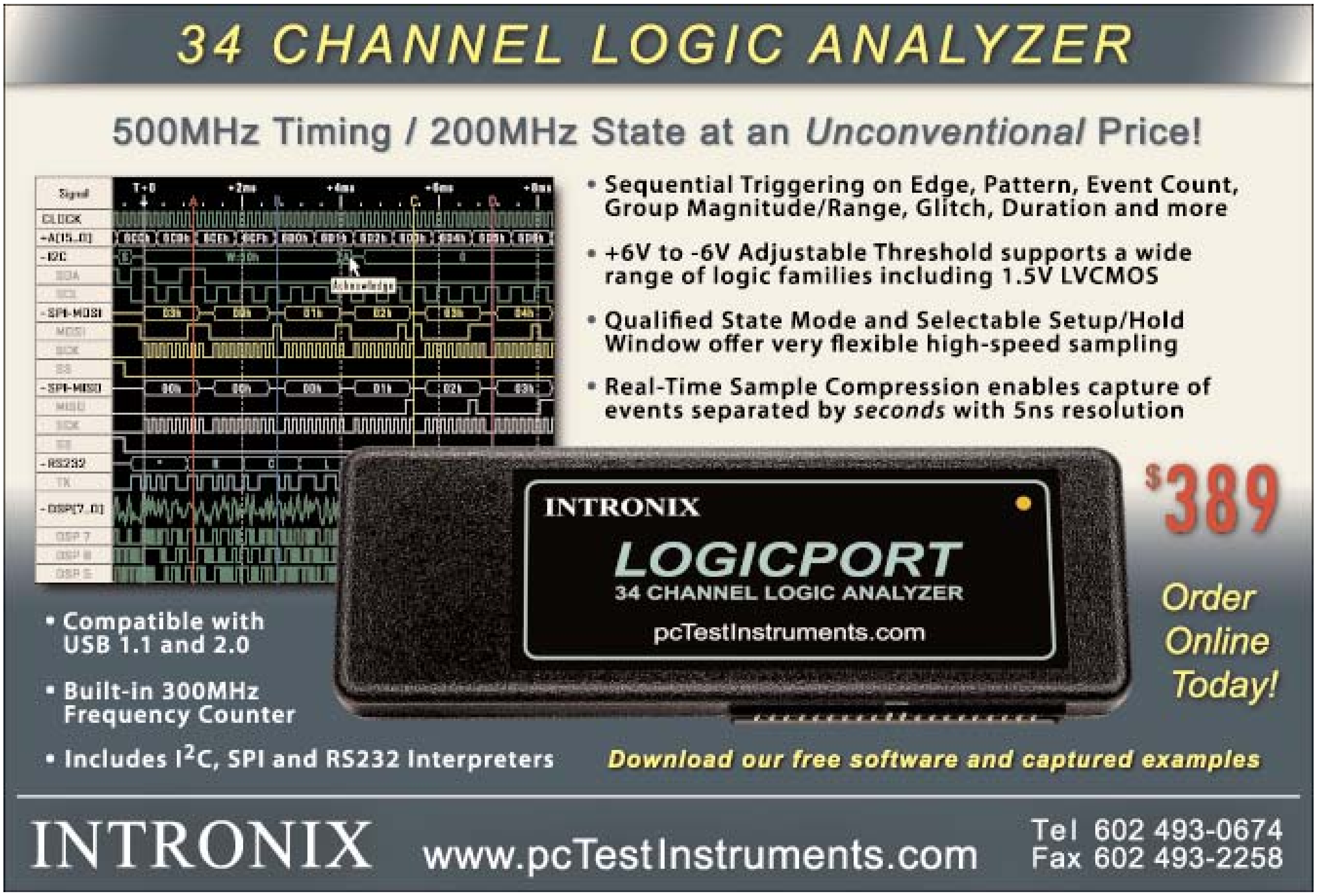
Photo 5—*Every transaction between the PC demonstration application and the PICDEM-FS USB board is captured in this trace view. Obviously, the USB Explorer 200's decode engine can't interpret the custom data, but the USB Explorer 200 is capable of analyzing and decoding many of the standard USB protocols in use by various USB device vendors.*

taking a trace with the temperature variable at equilibrium. (The potentiometer was set for minimum value and the LEDs were in an off status.)

Photo 5 shows a typical trace. I took the liberty of assembling the collected data in a series of runs (see Figure 1, p. 72). All of the values in Figure 1 were selected because they originated from the PICDEM-FS USB board and they formed a repeating pattern in the trace. I culled out the USB protocol stuff to keep things simple. Moving from left to right in Run 2, it seems that the data in the third column represents the current temperature. Note that the hex temperature value changed from 0x0D4F to stabilize at 0x0E87. Thus, you can conclude that all of the data following a 0x34 identifier is temperature data.

Run 3 shows the results of cranking the 10-kΩ potentiometer to its maximum resistance. Judging from the maximum reading in the fourth column, the potentiometer is being monitored by the PIC18F4550's 10-bit ADC.

The results of Run 4 and Run 5 are obvious to even the most casual observer. You have now identified all of the USB datastream variables. USB is documentation intense. If you were to check the C code for the PICDEM-FS board firmware, you'd find defini-


34 CHANNEL LOGIC ANALYZER

500MHz Timing / 200MHz State at an *Unconventional* Price!

- Sequential Triggering on Edge, Pattern, Event Count, Group Magnitude/Range, Glitch, Duration and more
- +6V to -6V Adjustable Threshold supports a wide range of logic families including 1.5V LVCMOS
- Qualified State Mode and Selectable Setup/Hold Window offer very flexible high-speed sampling
- Real-Time Sample Compression enables capture of events separated by *seconds* with 5ns resolution

$389

- Compatible with USB 1.1 and 2.0
- Built-in 300MHz Frequency Counter
- Includes I^{2}C, SPI and RS232 Interpreters

Order Online Today!

Download our free software and captured examples

INTRONIX www.pcTestInstruments.com Tel 602 493-0674 Fax 602 493-2258

```
//Run 1: Baseline
32 03 00  32 04 00  34 4F 0D  37 00 00
32 03 00  32 04 00  34 3F 0D  37 00 00
32 03 00  32 04 00  34 4F 0D  37 00 00
32 03 00  32 04 00  34 4F 0D  37 00 00

// Run 2: Finger on temperature sensor
32 03 00  32 04 00  34 8F 0E  37 00 00
32 03 00  32 04 00  34 7F 0E  37 00 00
32 03 00  32 04 00  34 87 0E  37 00 00
32 03 00  32 04 00  34 87 0E  37 00 00

// Run 3: 10-kΩ Potentiometer at maximum
32 03 00  32 04 00  34 4F 0D  37 FF 03
32 03 00  32 04 00  34 3F 0D  37 FF 03
32 03 00  32 04 00  34 4F 0D  37 FF 03
32 03 00  32 04 00  34 4F 0D  37 FF 03

//Run 4: User LED D3 selected/LED D4 not selected
32 03 01  32 04 00  34 4F 0D  37 FF 00
32 03 01  32 04 00  34 3F 0D  37 FF 00
32 03 01  32 04 00  34 4F 0D  37 FF 00
32 03 01  32 04 00  34 4F 0D  37 FF 00

// Run 5: User LED D4 selected/LED D3 not selected
32 03 00  32 04 01  34 4F 0D  37 FF 00
32 03 00  32 04 01  34 3F 0D  37 FF 00
32 03 00  32 04 01  34 4F 0D  37 FF 00
32 03 00  32 04 01  34 4F 0D  37 FF 00
```

Figure 1—*Putting this set of tables together wouldn't have been possible without the USB Explorer 200.*

tions for all of the recently traced data elements.

WHAT DID THAT PROVE?

First of all, I never had to consult a specification. Secondly, I saw everything that happened on the USB link from a dead start. Lastly, I picked out the data and interpreted it without any idea of how the USB application code was written.

In addition to being able to trace a USB link, the USB Explorer 200 decoding engine breaks down the hex data into human-readable USB context. This is especially useful when you're working with the USB descriptor traffic.

As simple as USB seems, it isn't the easiest protocol to use if you're trying to put it to use in an application from scratch. In my quest for the USB grail thus far, I've come across three useful USB tools: Jan Axelson's USB books, Dr. Bob's HIDMaker software, and the Ellisys USB Explorer 200. By the way, Jan has a new book called *USB Mass Storage: Designing and Programming Devices and Embedded Hosts*.

If you need to get serious with USB development, the aforementioned tools will help you during the design and debugging phases of a project. Jan and Dr. Bob have already put embedded into the USB mix. The Ellisys USB Explorer 200 takes away the complication.

Fred Eady (fred@edtp.com) has more than 20 years of experience as a systems engineer. He has worked with computers and communication systems large and small, simple and complex. His forte is embedded-systems design and communications.

RESOURCE

J. Axelson, *USB Mass Storage: Designing and Programming Devices and Embedded Hosts*, Lakeview Research, Madison, WI, 2006.

SOURCES

USB Explorer 200
Ellisys
www.ellisys.com

PICDEM-FS USB Demonstration board
Microchip Technology, Inc.
www.microchip.com

HIDMaker FS
Trace Systems, Inc.
www.tracesystemsinc.com

4th International System-on-Chip
SoC
Conference & Exhibit

Early Bird Registration Open Now
Register Today & Save!

November 1 & 2, 2006 - Radisson Hotel
Newport Beach, CA

www.SoCconference.com

The Most Informative & Targeted
SoC, ASIC, FPGA, and Foundry
Technology Conference & Exhibit
Event of the Year
Don't Miss Out!

For Information and Questions:
SoC@SavantCompany.com
949-851-1714

www.SavantCompany.com


HomeVision-Pro

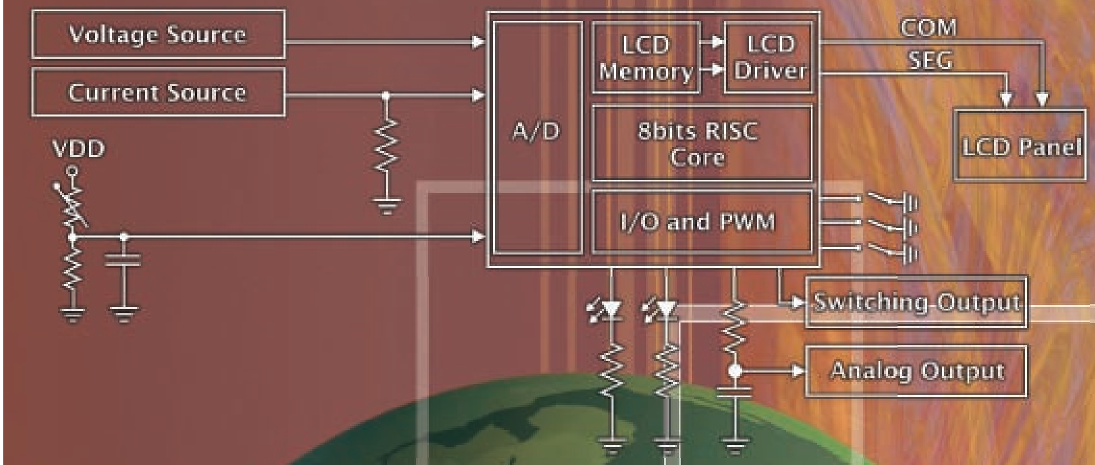
CONTROLS:
◆ Lighting
◆ HVAC
◆ Security
◆ A/V
◆ Much more...

HomeVision-Pro is the newest, most powerful automation controller in the award-winning HomeVision® product line. It's configured from a PC, but can then run standalone. Features:

◆ Two-way X-10 & IR ◆ Video screens on a TV
◆ 8 relays ◆ 16 digital & 4 analog I/O
◆ 64 digital temp sensors ◆ 1 USB & 3 serial ports
◆ Powerful scheduling (>8000 lines, >2000 objects)
◆ Clock ◆ PC web server ◆ Expandable
◆ Software included ◆ 3-year warranty

Custom Solutions, Inc.
800-398-8882, Web: www.csi3.com, E-mail: csi@csi3.com

HOLTEK
www.holtek.com



Superior LCD display MCU solutions

With LCD display applications firmly in mind, Holtek has designed its HT46RX and HT49RX series of MCU devices. Integrated device options, which enable a compatible driver to be selected for specific LCD panels, and with the utilisation of RAM mapping techniques, the task of LCD matching and driving remains a simple one indeed. As the HT46RU66 and the HT49RU80 devices also contain a fully integrated UART function, the ability of these MCUs to manage and display external data is also provided.

The HT46RX series in addition to their LCD driver function, also contain a 9-bit to 12-bit resolution A/D converter, which by interfacing to a range of sensors, allows external signals such as voltage, current and temperature etc. to be detected and displayed. The integral 8-bit PWM function and its associated outputs together with their corresponding internal registers, offer easy Duty Cycle control of the PWM outputs for convenient control of external parameters such as temperature, speed, voltage and current etc.

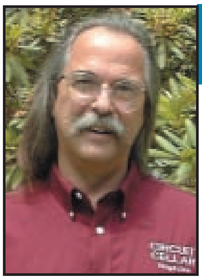
- ◆**Wide operating voltage 2.2V~5.5V**
- ◆**Industrial Specifications -40°C ~ +85°C**
- ◆**High noise immunity and antistatic performance**
- ◆**Real time clock and UART communication functions**
- ◆**Internal LCD driver function**
- ◆**Fully compatible OTP and Mask level devices**

Part No.	Program	Data	I/O	LCD	Timer	A/D	PWM	UART	Package
HT46R62	2Kx14	88x8	20	19x4	8-bitx1	9-bitx6	8-bitx3	--	52QFP, 56SSOP
HT46R63	4Kx15	208x8	32	19x4	16-bitx1	8-bitx8	8-bitx4	--	56SSOP, 100QFP
HT46R64	4Kx15	192x8	24	32x4	8-bitx1, 16-bitx1	10-bitx8	8-bitx4	--	52QFP, 56SSOP, 100QFP
HT46R65	8Kx16	384x8	24	40x4	16-bitx2	10-bitx8	8-bitx4	--	52QFP, 56SSOP, 100QFP
HT46RU66	16Kx16	576x8	32	46x4	8-bitx1, 16-bitx2	12-bitx8	8-bitx4	v	52QFP, 56SSOP, 100QFP

Part No.	Program	Date	I/O	LCD	Timer	UART	Package
HT49R30A-1	2Kx14	96x8	14	18x4	8-bitx1	--	48SSOP
HT49R50A-1	4Kx15	160x8	20	32x4	8-bitx2	--	48SSOP, 100QFP
HT49R70A-1	8Kx16	224x8	24	40x4	8-bitx1, 16-bitx1	--	100QFP
HT49RU80	16Kx16	576x8	31	47x4	8-bitx1, 16-bitx2	v	100QFP

HOLTEK SEMICONDUCTOR INC.

Holmate Semiconductor, Inc.
46729 Fremont Boulevard,Fremont, CA 94538
Tel: (510) 252-9880 Fax: (510) 252-9885
www.holmate.com



FROM THE BENCH by Jeff Bachiochi

Accessing the USB Framework

A Quick Transition from UART to USB

Microchip Technology has integrated USB hardware into some of its microcontrollers. Fortunately, designers can use a USB framework written in C to handle all of the tedious USB routines. Read on to learn how to transition from UART to USB.

Many of you use USB-to-serial dongles (or interface chips) to bridge the gap between products with serial ports and new PCs that have only USB ports. You may even design in these interfaces on new projects using your favorite microcontroller that doesn't have USB capabilities. But sooner or later, you'll want to take the plunge and use a microcontroller that has a built-in USB interface.

Although it is end-user friendly, using USB is much more involved than just setting up a few parameters like the data rate and throwing data into a UART buffer. Because USB is so flexible, it is also complicated. Fortunately, there are some standards that make it tolerable for beginners.

USB-to-serial interfaces don't require additional software. On the PC end, there are standard Windows drivers that allow a USB connection to look like a virtual COM port. Back at the peripheral, the interface has preprogrammed embedded software. Thus, all you need to do is connect it to a microcontroller's existing serial port. To make the jump and eliminate this extra external interface, you must choose a microcontroller that has built-in USB hardware and then write the routines to support the hardware.

Microchip Technology is one of the manufacturers that has integrated USB hardware into some of its microcontrollers. To help jumpstart designers, it published "Migrating Applications to USB from RS-232 UART with Minimal Impact on PC Software," which describes a USB framework written in C language that handles all of the tedious USB routines. The framework initializes the USB hardware and gives you seven functions for accessing the USB data buffers. By incorporating these in the microcontroller's application, you can quickly make the transition from UART to USB as long as the application is in C language.

FIGHTING TOOTH & NAIL

You probably know where I stand with C language. I enjoy writing in assembler. Although I've written just enough C code to get me into trouble, I usually bend over backwards to avoid it. Call me a Neanderthal if you must, but I prefer to spend my time learning about technology rather than trying become proficient in C language. So, in this article, I'll describe the functions and how they can be accessed via an assembly language application program.

I used Microchip's PICDEM FS USB demonstration board as a learning tool. This hardware provides a USB port, a serial port, a few I/O devices (switches, LEDs, a potentiometer, and a temperature sensor), and an ICD2 interface to a PIC18F4550 microcontroller. The included USB framework code allows you to compile working code using MPLAB. The user.c file includes experiments that lead you through a few tutorial tasks using C language. This file does not implement a USB-to-serial application, but the experiments will help you reach that goal.

This revision of Microchip's USB framework handles the basic tasks required for getting a USB interface up and running quickly. The seven functions are all you need to make this connection. As you cans see in Table 1, moving data in and out of the micro-

Function	Description	Syntax
`putrsUSBUSART`	Write a null-terminated string from program memory to the USB.	`void putrsUSBUSART (const rom char *data)`
`putsUSBUSART`	Write a null-terminated string from data memory to the USB.	`void putsUSBUSART (char *data)`
`mUSBUSARTTxRom`	Write a string of a known length from program memory to the USB.	`void mUSBUSARTTxRom (rom byte *pdata, byte len)`
`mUSBUSARTTxRam`	Write a string of a known length from data memory to the USB.	`void mUSBUSARTTxRom (byte *pdata, byte len)`
`mUSBUSARTIsTxTrfReady`	Is the driver ready to accept a new string to write to the USB?	`bool mUSBUSARTIsTxTrfReady (void)`
`getsUSBUSART`	Read a string from the USB.	`byte getsUSBUSART (char *buffer, byte len)`
`mCDCGetRxLength`	Read the length of the last string from the USB.	`byte mCDCGetRxLength (void)`

Table 1—*Various C functions are available when using Microchip's USB framework.*

Syntax	Parameter passed to function		Parameter passed back from function	
`void putrsUSBUSART (const rom char *data)`	Pointer to data	Word	None	—
`void putsUSBUSART (char  *data)`	Pointer to data	Word	None	—
`void mUSBUSARTTxRom (rom byte *pdata, byte len)`	Pointer to data, length	Word, byte	None	—
`void mUSBUSARTTxRom (byte *pdata, byte len)`	Pointer to data, length	Word, byte	None	—
`bool mUSBUSARTIsTxTrfReady (void)`	None	—	0=busy,1=ready	Bit
`byte getsUSBUSART (char *buffer, byte len)`	Pointer to data, length	Word, byte	Actual length of data passed	Byte
`byte mCDCGetRxLength (void)`	None	—	Actual length of data passed	Byte

Table 2—*This syntax is necessary for accessing each of the C language USB functions.*

controller isn't complicated. These functions are similar to other I/O functions available in C language (i.e., those available for a UART).

The framework is written as a polled loop without interrupts, so there aren't any bottlenecks. Any code using the framework should be state machine oriented to avoid any delays in execution. Five of the functions are used for transmitting data. The `mUSBUSARTIsTrfReady` function quickly checks to make sure the transmitter can accept data. If it can, one of the first four functions can be used for transferring data into the USB buffer. Using `putrsUSBUSART` or `putsUSBUSART` requires the data string to be null terminated. These require only a pointer to the beginning of the data. The `mUSBUSARTTxRom` and `mUSBUSARTTxRam` functions require the length of the string (number of characters) and a pointer to the beginning of the data.

Receiving data is simpler. The `getsUSBUSART` function attempts to transfer to you any data in the receive buffer (see Table 2). Following this with the `mCDCGetRxLength` function tells you how many characters were actually transferred.

NOW YOU C IT

The user.c file has already defined I/O output buffers.

```
char input_buffer[64];
char output_buffer[64];
```

Jumping right into the required code is a snap. After perusing the UART functions predefined in the compiler, I found C functions for writing to and reading from the on-board hardware UART. The user.c file already had initialization code for the UART, so all I had to do was interface the USB and the UART functions (see Figure 1).

Using Figure 1 as a guide, I added the necessary C code to the user.c file (see Listing 1, p. 76). After compiling and downloading the resultant HEX file into the microcontroller, the application initialized both the USB and UART ports (initialization code was already included) and transferred data bytes between the two hardware devices.

The PICDEM FS USB demonstration board has USB and DE9 connectors. Two instances of HyperTerminal can be opened with one connected via a COM port to the DE9 serial port and the other via a virtual COM port to the USB port. Data typed in either instance of HyperTerminal is transferred to the other instance.

The demonstration board also has a connection for the ICD2 in-circuit debugger. I use the debugger not only for its obvious debugging capabilities, but also to follow the code execution and understand how the parameters are passed. The functions beginning with the `m` indicate that these are macros.

NOW YOU DON'T

I used some predefined C routines to cobble together a working USB-to-RS232 interface to help me understand how to use the USB framework. Can I use this USB framework without having to write any of my application in C? I really want to substitute a user.asm file in place of the user.c file. The first step involves imitating the working USB-to-RS-232 C application. Four of the seven functions are needed. Let's look at each of these to see what's necessary to make them accessible through assembler.

The two checking functions are straightforward. The `mUSBUSARTIsTxTrfReady` function merely checks the `cdc_trf_state` variable for the present state and returns the state in WREG. Likewise, the `mCDCGetRxLength` function returns the `cdc_rx_len` variable. This is the actual number of data bytes transferred from the USB buffer.

The remaining functions need to pass parameters to the C code. The `mUSBUSARTTxRam` function uses the pointer source (`pCDCSrc`) to hold the pointer to the output buffer. Additionally, the length of the string must be put into the `cdc_tx_len` variable and a constant of 0 (`_RAM`) is placed in the `cdc_mem_type` variable. The actual transferring of the data happens else-

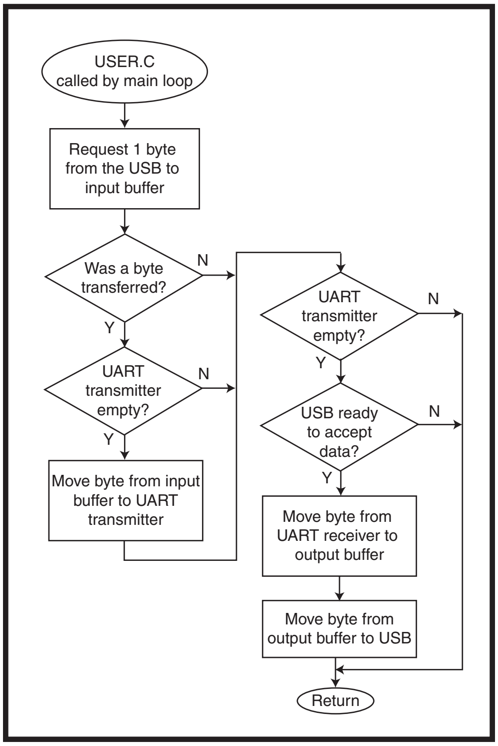


Figure 1—*I added code to the user.c file to interface the hardware USB and UART interfaces on the PIC18F4550. I transferred a single byte of data bidirectionally between the two.*

Listing 1—*I added a few lines to the user.c file to transfer single characters to and from the USB/UART hardware on the PIC18F4550.*

```
getsUSBUSART(input_buffer,1);
if(mCDCGetRxLength())
{
     if(PIR1&0x10)                        // if TXIF=1
     {
          WriteUSART(input_buffer[0]);
     }
}

if(PIR1&0x20)                             // if RCIF=1
{
     mUSBUSARTIsTxTrfReady();
     if(WREG)                             // CDC_TX_READY=1
     {
          output_buffer[0]=ReadUSART();
          mUSBUSARTTxRam((byte*)output_buffer,1);
     }
}
```

where (in another function of the execution loop), but it's triggered by placing a constant of 1 (`CDC_TX_BUSY`) in the `cdc_trf_state` variable.

To get data from USB, the `getsUSBUSART` function requires the requested count and the `input_buffer` address to be passed via a temporary stack (in this case using the indirect pointer FSR1). It looks like a macro handles this before making the actual call to the function. This seems like an odd way to pass parameters, and I gather it has to do with the compiler's attempt to keep things portable. Note: Had I been curious enough to pore over the endless pages of documentation, I would have found an explanation of this in the "Calling Conventions" section of Microchip's "MPLAB C18 User's Guide."

I created small macros to use the last two functions. Besides defining a few of the constants used in the C functions, the macros were all I needed to bridge the gap between those functions and my user.asm file. After the UART functions were replaced by assembly code, I had an assembly module that directly replaced the user.c file. After eliminating the user.c file and adding the user.asm file to the project, I was able to make the project and experience the same results as before using two instances of HyperTerminal.

UNIVERSAL INTERFACE

There have been many times in which I've wanted to make a PC connection a USB one. I've used the external interface chips, which are easy. However, it seems like a waste to add an extra chip to a project when there are so many microcontrollers that now include USB hardware. Perhaps the biggest stumbling block for any hardware engineer is the USB driver necessary for a PC. If you can use a standard driver, you can avoid having to write all that messy software. The standard virtual COM connection is a great interface, but can it be used for other purposes?

I think you can see from what I've presented so far that this will work rather well if you need 8 bits of output and 8 bits of input from a USB dongle. Instead of USB input placed in a UART's transmitter, the data can be presented to an output port just as easily. In addition, an input port can be sampled to provide data for the USB. This is rather simplistic on the PC side because any sent character ends up on the output port, and input port data is reflected back into the PC. Your application must decide how to handle input data. It might be sampled and sent at some fixed rate or sent only when the data actually changes.

What happens when you need more than 1 byte of output and/or 1 byte of input data? At that point, the data you send and receive must be formatted using some kind of protocol. The simplest protocol might be to require data to be passed in multiple byte chunks (say, 4 bytes at a time to cover 32 bits). The problem with this simple protocol is that the data will be wrong if you ever get out of sync.

A better approach might be to add some control bits to each byte being transferred. For instance, data is sent as the lower nibble of each transfer, with the upper nibble consisting of control bits where bit 4 might indicate whether the data (within the lower nibble) is the upper or lower nibble of the actual data. Other control bits might indicate things like whether the transmitted data is requesting or giving information (RD/WR) and if the data is a control function or actual data (CTRL/DATA). If the control function loads a pointer, you can steer the data to different registers (0–255). This approach also requires you to decide when and where to receive input data.

All of this leads up to my design: an SPI dongle. Many new interface chips today have SPI interfaces. In order to experiment with them, they require an application interface. It's a lot of work to develop a special application each time you want to experiment with one of these chips. I wanted a universal interface that I could use to connect to and interrogate a chip with-

	Bit 7 (*WR/RD)	Bit 6 (*PTR/REG)	Bit 5 (Don't care)	Bit 4 (*LSN/MSN)	Bits 3:0 (DATA)
Write pointer	0	0	x	0	Lower 4 bits of data
				1	Upper 4 bits of data
Read pointer	1	0	x	0	Lower 4 bits of data
				1	Upper 4 bits of data
Write register	0	1	x	0	Lower 4 bits of data
				1	Upper 4 bits of data
Read register	1	1	x	0	Lower 4 bits of data
				1	Upper 4 bits of data

Table 3—*The PC application uses this data-passing protocol to communicate with the SPI dongle. Data is passed in 2-byte increments (passing the upper byte first). Actions are executed on the dongle after the second byte (lower data nibble) has been received.*

out a bunch of extra programming. The PIC18F4550 microcontroller has a hardware SPI interface and plenty of I/O, so it wasn't too hard to program this application using the PICDEM FS USB demonstration board.

SPI DONGLE

Although the SPI interface is defined as having a master clock SCK, serial data in (SDI), and serial data out (SDO), it often includes a slave select (SS). As a result, multiple devices connected to SCK/SDI/SDO can be individually selected. Some slave devices can also have an interrupt or other output associated with the interface. To cover these bases, the SPI dongle includes digital input and output bits in addition to the three-wire SPI interface. The need to control multiple registers falls squarely into the aforementioned protocol.

Data passed from a PC application to the SPI dongle uses a split byte protocol. Each byte passes control information via the upper nibble and half of the actual data in the lower nibble (see Table 3). Only three registers are needed for this application: SPI, digital input, and digital output. (Up to 256 registers could be implemented using this protocol.) Other registers might include the configuration of the SPI port (i.e., clock polarity or data edge latching). Digital output bits can be used for an SPI SS enable, LEDs, or other signaling. Digital inputs can be used for monitoring SPI interrupt outputs, external switches, or other devices.

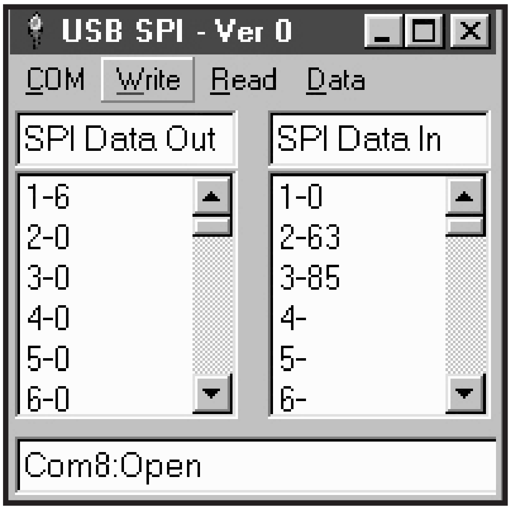
Once initialized, the SPI dongle's application uses a state machine within the user.asm module, which is called as part of the main application loop. Each time it is called, it checks to see if any data has been received. It exits if there isn't any. If data is found, program execution will branch based on the control bits in the upper nibble of the received byte.

There are four basic routines: write or read the pointer and write or read the register. The pointer is used to select which register is accessed. In this application, it has only three possibilities: 0, 1, or 2. When the pointer=0 register data interacts with the SPI device, pointer=1 is the input register and pointer=2 is the output register.

When an upper nibble is received, the data is temporarily stored while waiting for the lower nibble. When the lower nibble is received, the actual data can be reassembled and applied to either the pointer or the register being pointed to. In actuality, reading the pointer or register doesn't require data. As such, that data is discarded into the bit bucket. Some functions may not be necessary, but they're included for completeness (i.e., reading the pointer or output register and writing the input register).

PC APPLICATION

A standard driver included in Windows (98SE and later versions) simplifies PC communication via USB. This driver installs a virtual COM port that is identical in function to the legacy hardware serial ports. Any application that presently selects a COM port can alternately choose a virtual COM port. When it becomes necessary to use a special protocol for passing data (like with the SPI dongle project), the complimentary PC application must know how to format the data sent and received via the virtual COM port. You can use HyperTerminal to com-



SPI Data Out

1-6
2-0
3-0
4-0
5-0
6-0

SPI Data In

1-0
2-63
3-85
4-
5-
6-

Com8:Open

Photo 1—*I used Liberty BASIC to quickly build an application, which can be used to transfer data between the PC and the USB dongle using a special protocol. The under-the-covers protocol breaks down each data byte into two nibbles. Each nibble is sent separately within the lower half of 2 bytes. The upper nibble of each byte holds control bits. This allows the data to be reassembled at the other end and directed to the proper location.*

SMT Prototyping Adapters
Snap-Apart™ PCB's with .100" pin strips
Dozens of assorted adapters on each PCB
A variety of PCB's - patterns on both sides
Eliminate the cost and delay of PCB layout / fab!
SOIC PLCC SSOP QFP DPAK
TSSOP SOT23 MSOP QSOP
SC90 D2PAK SOT89 SOT143
TSOP SC70 SC88 SC326
and many more.
BELLIN DYNAMIC SYSTEMS, INC.
www.beldynsys.com (714) 630-8024
- Rapid Development Solutions for the Technical Professional -

bob
basic overlay board
Introducing:
Vector graphics
Custom Font Capability
SPI or RS-232 Interface
Meet bob-4
Video display generator
www.decadenet.com
DECADE ENGINEERING
503-743-3194 Turner, OR, USA

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

municate, but formatting and receiving formatted data is difficult.

I've used Liberty BASIC in the past to whip up simple application programs. It's an inexpensive development language for Windows programming. Photo 1 (p. 77) shows the application that controls the SPI dongle from the PC. After plugging in the SPI dongle, Windows finds the USB device and installs the appropriate drivers for an additional virtual COM port. When the PC application starts, it searches COM ports for active devices. The COM pull-down menu allows you to choose between active COM devices.

Once connected, you can begin communicating by choosing from the other pull-down menus: data (for setting up data bytes), read, and write (for reading from or writing to the pointer or a register). The application has an output and an input array of 255 bytes each to hold data. These can be viewed in the SPI data out and SPI data in list boxes. The Data pull-down menu offers fill and change functions to the SPI data out array.

In this case, I sent 3 bytes (6, 0, and 0). I used the fill function to fill the array with zeros and then the change function to enter 6 in the first array position. When the Write pull-down menu was clicked, I could choose to write to the pointer or to the register.

Having already written a zero to the pointer (pointing it at the SPI register), I could send data by clicking on Register in the pull-down menu. The application asked how many bytes I wanted to write. I answered three. Note that 6, 0, and 0 were then formatted into 6 bytes and sent to the USB dongle's SPI register. The dongle reconstructed the 3 data bytes and clocked them out of the SDO. Three bytes were returned from the SPI (SDI). These were formatted into 6 bytes and returned via USB to the PC application. The application reconstructed the data and displayed it in the SPI data in the list box as 0, 63, 85.

This sequence of data requests key touch information from a Quantum QT60168 QProx 16-key touch pad. The returned data indicates that no key is presently being touched. This device requires an *SS (low chip select) for each transmitted data byte. To make the application more user friendly whenever there is a request to write data to the SPI register (the pointer=0), the application automatically sends formatted data to change the pointer to the output register and writes a zero to bit 0 (clearing the output bit used as *SS). It then changes the pointer back to the SPI register and sends a byte of data (two formatted bytes) to the SPI register. Finally, it changes the pointer back to the output register and writes a one to bit 0 (setting the output bit used as *SS). This is a lot easier than manually using the output register to clear and set an output bit around each data byte written to the SPI register. This is handled in the PC application and not the USB dongle.

GOTCHAS

I ran into two stumbling blocks while debugging this project. First, I assumed that each time `getsUSBUSART` was used to read from the USB buffer I would be getting the next available data (a FIFO type buffer). The length parameter in this function allows you to limit the passed data, but the buffer is apparently flushed by this action or new data coming in and any remaining (unread) data is gone. This means you should request the maximum data available (as configured in the USBCFH.H file as 8, 16, 32, or 64 bytes). Grabbing a byte at a time won't work. My original program worked well because characters typed on a console are slow enough to get their own USB packet! This was fixed by creating an intermediate buffer where I could hold up to 64 bytes (the maximum endpoint buffer size).

I also assumed that if the USB buffer wasn't emptied new data wouldn't be accepted. Wrong again. New data continually overwrites previous data, so you must be ready to empty the buffer if it has data. If you aren't finished with the old data in the intermediate buffer when new data is available, it will be lost. By making the intermediate buffer into a ring buffer, you can continue pulling from the head of the buffer while new data is added to the tail end. A ring buffer of at least two times the endpoint buffer size should be implemented so new data can be added to the buffer without having to worry about an overflow. However, throughput won't be the problem, just the handling of multiple USB packets.

I understand that this is the first implementation of Microchip's USB framework. Things may change in the future, but I think a bit more documentation on how the functions work would be beneficial to anyone attempting to use the framework.

The application note about migrating an application from RS-232 to USB offered enough information to jumpstart this project. My user.asm module can now be used as a template for other projects requiring a USB interface. And I can continue writing in assembler, delaying again the necessity of learning C.

Jeff Bachiochi (pronounced BAH-key-AH-key) has been writing for Circuit Cellar *since 1988. His background includes product design and manufacturing. He may be reached at jeff.bachiochi@circuitcellar.com.*

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2006/195.

RESOURCES

Microchip Technology, Inc., "MPLAB C18 User's Guide," 2005.

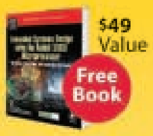
R. Rojvanit, "Migrating Applications to USB from RS-232 UART with Minimal Impact on PC Software," AN956, Microchip Technology, 2004, ww1.microchip.com/downloads/en/AppNotes/00956b.pdf.

SOURCES

Liberty BASIC
www.libertybasic.com

PIC18F4550 Microcontroller and PICDEM FS USB board
Microchip Technology, Inc.
www.microchip.com

QT60168 Touch pad
Quantum Research Group
www.qprox.com



RabbitFLEX

A New Way To Customize

- Click-to-ship in 5 days!
- Pay only for what you need
- Revision friendly
- Perfect for prototype and production

FLEX
Ready

RabbitFLEX™ is an unique build system that gives you the power to develop custom boards without the hassle and the cost. The RabbitFLEX simple-to-use web interface allows you to choose from numerous options such as digital I/O, analog I/O, serial ports, and Ethernet connections on your custom board. Just configure and buy online and our patent pending manufacturing process will deliver your solution in a matter of days. With RabbitFLEX you will reduce design risk, manufacturing cost, and development time.

Start developing now by ordering the RabbitFLEX Tool Kit and your own custom RabbitFLEX board. Take your solution to the next level.

Configure and Buy Online
www.rabbitFLEX.com

Quick-Turn Boards Range From	RabbitFLEX Tool Kit
$149-$279	$199

Test Drive RabbitFLEX

Build your custom RabbitFLEX board online. Add a tool kit to your order for a complete development system including Dynamic C®.

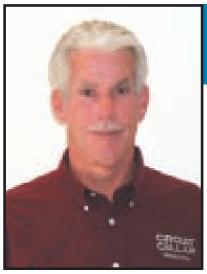
www.rabbitFLEX.com

$49 Value
Free Book

For a limited time
with kit purchase.

RABBIT®
Semiconductor

2900 Spafford Street, Davis, CA 95616 Tel **530.757.8400**
Solutions That Work



SILICON UPDATE

by Tom Cantrell

ARM Twister

Are you ready for a new chapter in the ARM story? It's all about Atmel's AVR32 "RISC MCU." Tom takes you between the lines.

ARM chips have been grabbing all the headlines lately, perhaps leaving the impression that competitors may as well hang up their transistors and go home.

I don't think so. Yes, ARM chips (whether the "originals" like ARM7, 9, and 11 or the new Cortex series) are going to fill a lot of sockets, but that doesn't mean historic competitors are going quietly into that dark night.

In fact, far from the story being over, a new chapter is being written this month. The author is Atmel and the title is AVR32.

TRUTH IN ADVERTISING

According to the press releases, the AVR32 is a new 32-bit "RISC MCU." As far as the "MCU" moniker goes, my opinion is that it should be reserved for chips that can operate alone as single chips with built-in flash memory for instructions and RAM for data. But, the Atmel AT32AP7000, the initial chip to use the new architecture and the subject of this month's column, has only the latter (i.e., 32 KB of RAM), which makes it an "MPU" in my book. However, it wouldn't be hard to make a single-chip version (especially using the expeditious stacked die approach), so I wouldn't be surprised to see a true "MCU" AVR32 emerge in the future.

As for "RISC," it wasn't long after the term first appeared in the early '80s that it began to be abused, soon devolving to mean anything that wasn't an 'x86 or a 68K. The bottom line is that the "Reduced Instruction Set Computer" concept as originally defined has been surpassed by hybrid architectures that combine aspects of RISC, CISC, and even DSP.

Back in the '80s, the focus was on boosting clock rate and MIPS ratings, both held back by the complexity of earlier "CISC" designs. But today, reliance on these simple metrics is not only misleading, but arguably even misguided.

Who cares about instructions? What matters is work. The RISC pioneers did a good job "Relegating Impossible Stuff to the Compiler" to turn the former into the latter. Yes, it took more RISC instructions to get the work done, but the additional MIPS and megahertz more than made up the difference. But these days, cost and power matter as much, if not more, than MIPS. For battery-powered gadgets like cell phones, PDAs, and MP3 players, the goal is to do only the work that must be done while minimizing power consumption and system cost.

Don't get me wrong. The original RISC concept was a wonderful thing, a virtual renaissance that inspired a new way of thinking. RISC reminds me of the Turing machine, something that truly qualified as a paradigm shift by laying the foundation for the next level. Both have their place in history, but not in your design.

PIPE DREAM

You don't have to get far into the 935-page (!) datasheet to discover that AVR32 is a high-performance processor designed to serve mini-me "computer" applications like the aforementioned PDAs and cell phones. A number of factors make it clear that AVR32 is positioned well beyond the simple blue-collar embedded designs typically served by an "MCU."

Get deep enough under the hood and you'll find a rather high-end, seven-stage pipeline with sophisticated features like parallel sub-pipes (ALU, multiply, and load/store), out-of-order execution, result forwarding, and dynamic branch prediction (see Figure 1). It wasn't that long ago that such a design would have been considered relatively bleeding edge, and even today it goes well beyond the simpler three- to five-pipe stage designs that characterize most 32-bit "MCUs."

In turn, the longish pipeline enables a triple-digit clock rate of 133 MHz, which is

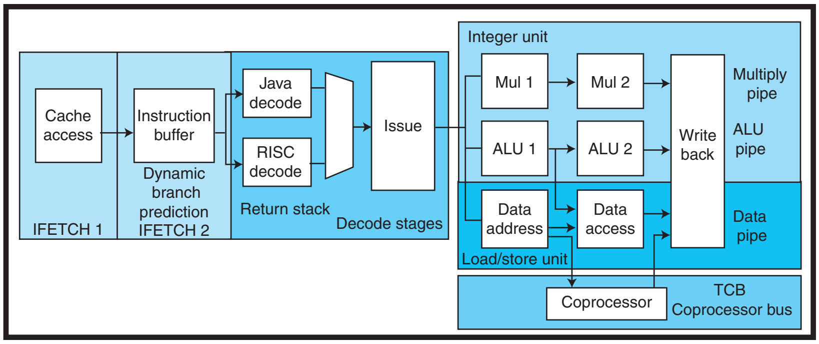
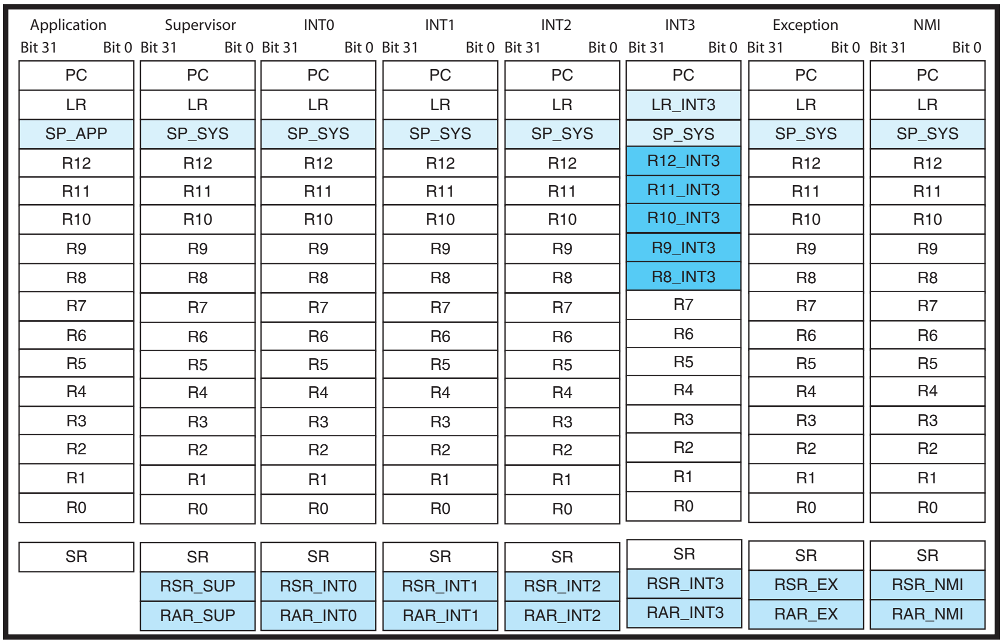


Figure 1—*Under the hood, AVR splits the difference between MPUs and MCUs with a seven-stage pipeline supporting low-triple-digit clock rates. Features such as branch prediction, out of order completion, and Java byte code support were once considered esoteric, but once again Moore's law crams yesterday's high-end features into tomorrow's commodity parts.*

two or three times the clock rate of an entry-level 32-bit MCU. On the other hand, 133 MHz isn't especially fast compared to other "computer" chips. In fact, recalling the earlier megahertz versus work discussion, Atmel specifically touts the ability of AVR32 to keep up with chips that must run at higher clock rates.

The triple-digit clock rate means cache (16 KB for instructions and 16 KB for data) is a must. I'm no big fan of cache for embedded applications, but with clock rates this high, it's the only way to avoid getting bogged down in wait states. Fortunately, for hard real-time applications, the caches are lockable. And for data, there's also predictable (i.e., optionally uncached) high-speed access to the additional 32 KB of on-chip RAM. The RAM is split into two 16-KB blocks that can be accessed in parallel by, for example, both the CPU and DMA controller.

Any doubts about the AVR32's computing pretensions are laid to rest thanks to a full-featured (i.e., virtual memory) MMU. However, I note the MMU is considered an "optional" part of the architecture, which may be replaced by a simpler memory protec-



Application	Supervisor	INT0	INT1	INT2	INT3	Exception	NMI
Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0
PC	PC	PC	PC	PC	PC	PC	PC
LR	LR	LR	LR	LR	LR_INT3	LR	LR
SP_APP	SP_SYS	SP_SYS	SP_SYS	SP_SYS	SP_SYS	SP_SYS	SP_SYS
R12	R12	R12	R12	R12	R12_INT3	R12	R12
R11	R11	R11	R11	R11	R11_INT3	R11	R11
R10	R10	R10	R10	R10	R10_INT3	R10	R10
R9	R9	R9	R9	R9	R9_INT3	R9	R9
R8	R8	R8	R8	R8	R8_INT3	R8	R8
R7	R7	R7	R7	R7	R7	R7	R7
R6	R6	R6	R6	R6	R6	R6	R6
R5	R5	R5	R5	R5	R5	R5	R5
R4	R4	R4	R4	R4	R4	R4	R4
R3	R3	R3	R3	R3	R3	R3	R3
R2	R2	R2	R2	R2	R2	R2	R2
R1	R1	R1	R1	R1	R1	R1	R1
R0	R0	R0	R0	R0	R0	R0	R0
SR	SR	SR	SR	SR	SR	SR	SR
	RSR_SUP	RSR_INT0	RSR_INT1	RSR_INT2	RSR_INT3	RSR_EX	RSR_NMI
	RAR_SUP	RAR_INT0	RAR_INT1	RAR_INT2	RAR_INT3	RAR_EX	RAR_NMI

Figure 2—*To speed context switches and minimize power-consuming memory access, each interrupt and exception level gets its own status and return address registers. In addition, one of the interrupt contexts (INT3) duplicates half the register file, a handy place to keep critical data the handler needs fast and frequent access to.*

VIPER
- 400MHz PXA255 Intel® processor
- Ultra-low power 1.9W (typical)
- TFT/STN graphics controller
- 64Mbytes DRAM / 32Mbytes Flash
- Ethernet, USB and serial ports
- CompactFlash (CF+) and PC/104 bus
- Ready-to-run development kits for:
  - Windows® CE 5.0 - Embedded Linux - VxWorks®

VULCAN
- 533MHz IXP425 Intel communications processor
- x2 100baseTx Ethernet ports
- x4 USB 2.0 and x4 serial ports
- 64Mbytes DRAM / 32Mbytes Flash
- CompactFlash (CF+) and PC/104 bus

ZEUS
- 520MHz PXA270 Intel processor
- 128Mbytes DRAM / 64Mbytes Flash
- x2 100baseTx Ethernet ports
- x2 USB 1.1 and x7 serial ports
- SDIO, CF+ and PC/104 bus

www.arcom.com

Arcom

us-sales@arcom.com
888-941-2224

Think Embedded.
Think Arcom.

tion unit (MPU) in future chip implementations.

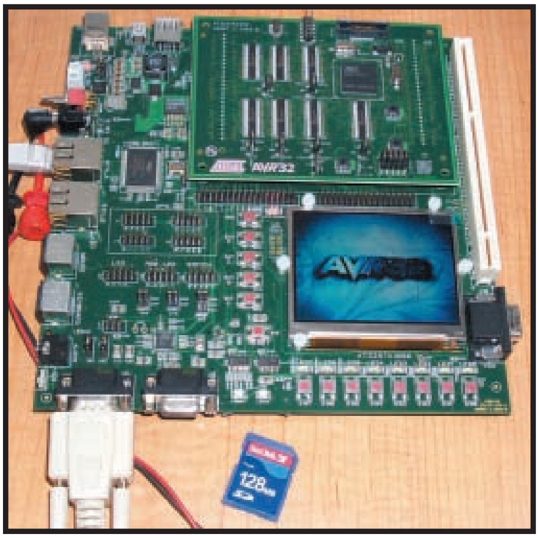


Photo 1—*Remember the old days when you plugged in a floppy disk to boot your computer? The STK1000 comes with the modern equivalent in the form of a 128-MB MMC flash card preloaded with Linux.*

The AVR32 is a pure load/store architecture, perhaps the most obvious holdover from the original RISC concept. Instructions operate only on the arguments stored in the 16 × 32-bit register file. There is provision in the architecture for instructions to set, clear, and toggle bits directly in memory, but those instructions aren't implemented in the AT32AP7000.

Speaking of the register file, the architecture defines distinctly different "non-banked" and "banked" options. Non-banked is the low-cost option that relies on a single register file and uses the stack for saving status and return addresses. The banked option provides dedicated status and return address registers for each interrupt and exception level. Furthermore, the architecture allows duplicating half or all of the general-purpose registers for each interrupt and exception level. As you can see in Figure 2 (p. 81), the AT32AP7000 uses the banked option and duplicates half the register file for the INT3 level.

Like virtually all competitors, the AVR32 abandons the original RISC concept of fixed-length instructions, using a mix of 16- and 32-bit opcodes to optimize performance versus code density. Code density isn't so much about saving space in external memory chips. Indeed, the challenge with PC market-driven SDRAMs, for example, may be more about finding memories that are "small enough" as the price-per-bit sweet spot moves ever upward. However, tighter code does stretch the mileage obtained from the instruction cache, reducing the number of misses and power-consuming spills and fills.

CRISC?

Probably the most dramatic departure from the original RISC concept, one that goes right to the heart of the acronym itself, is the AVR32 instruction set. As the RISC pioneers demonstrated (and Turing before them), a simple machine can be coerced into handling arbitrarily complex applications. However, as I mentioned earlier, the brute force bridging of the semantic gap between simple instructions and complex


PCB-POOL®
Instant online Quotes
Tooling and set-up included!
No Minimum Quantity,
No drill Limitations!
Full DRC on all orders +
Top Quality and on time delivery Guaranteed!
(if its late, get it free!)
Price Example: 16 Sq-Inches, (ds - pth)
2 Days: $ 90
8 Days: $ 22.5
Standard PCBPooL Service
SIMPLY SEND YOUR FILES AND ORDER ONLINE!
WWW.PCBPOOL.COM
DOWNLOAD OUR FREE PCB SOFTWARE
www.free-pcb-software.com
TOLLFREE: 1877 3908541
sales@beta-layout.com

applications demands a high-clock rate and all the baggage (notably power consumption) that goes with it.

By contrast, the AVR32 closes the semantic gap with the instruction set. Atmel analyzed a variety of must-have applications (e.g., the MPEG-4 codec) to identify the hotspots and crafted instructions accordingly. No surprise then to find a lot of DSP-like math instructions, including a measure of vector processing (i.e., SIMD) capability.

The AVR32 goes further than DSP-in-drag though. For example, a "Load with Extracted Index" instruction combines a shift, zero-extend, and addition in a single instruction to accelerate cryptographic block-cipher algorithms. Byte and word loads need not be aligned and feature automatic sign- or zero-extension. There are a full complement of bit- and byte-field operations (extract, extend, reverse, swap, count, etc.) as well as Application Call and System Call instructions that streamline OS (e.g., Linux) interaction. Considering argument (e.g., byte, half-word, word, and long) and extended/compact (i.e., 32-bit or 16-bit length) variants, I tallied about 250 or so distinct opcodes in the AVR32 repertoire, far more than the RISCs of yore.

Beyond the theoretical and theological aspects of the AVR32 architecture, there are pragmatic issues to consider, such as getting your application working. It can be tough figuring out what's going on inside a complicated (i.e., pipelined and cached) processor. When it comes to debug, three AVR32 features stand out.

First is the on-chip debug (OCD) scheme itself, which devotes significant hardware (logic and pins) to the cause. Notably, the chip supports both high-end and low-cost debug regimes. The former exploits a full-featured IEEE/ISTO Nexus 5001 debug implementation, including a dedicated 11-pin, 75-MHz AUX bus for spewing compressed trace data in real time. The low-cost option uses a simple USB/JTAG debugger (the same one used for 8-bit AVRs) but provides a measure of real-time trace capability thanks to a "Nanotrace" feature that uses the on-chip RAM as a trace buffer.

The pitfalls posed for debuggers by a fancy processor are another challenge. Caches that hinder visibility and other performance-oriented features (out-of-order execution, branch prediction, write buffering, etc.) can lead to less obvious gotchas. For example, out-of-order execution can cause a data breakpoint to be associated with the wrong instruction. The AVR32 debug scheme demonstrates a measure of real-world savvy by viewing memory through the caches and allowing the temporary disabling of the aforementioned debug-busting features.

Visibility issues also make it difficult to benchmark and fine-tune your application. The AVR32 sheds some light on the subject with a versatile performance counter scheme that instruments a variety of internal activities. The data so gathered can be massaged to come up with a bunch of useful statistics such as average clocks per instruction (CPI), cache hit/miss ratios, frequency and duration of stalls, branch prediction accuracy, and so on.

PACKING HEAT

I've used the old "kitchen sink" saying a lot recently when referring to the wave of latest-and-greatest, peripheral-packed chips. I guess one fresher way of putting it is that the AT32AP7000, with 256 pins to play with, has everything except the box the kitchen sink came in (see Figure 3). By my count, there are nearly two dozen major peripheral and system functions on board, so let's get started.

First, some housekeeping. The AT32AP7000 requires two power supplies: 3.3 V for the I/Os and 1.8 V for everything else. The datasheet is still preliminary and full of TBDs, but tentative specs put full-bore (133-MHz) power consumption at less than 0.5 W, not bad at all for such a big chip. To further battery-powered applications, the chip features a comprehensive set of low-power modes, including explicit clock control down to the individual module (e.g., bus and peripheral) level.

Speaking of clocks, the AT32AP7000 has three oscillator inputs and two PLLs. One of the

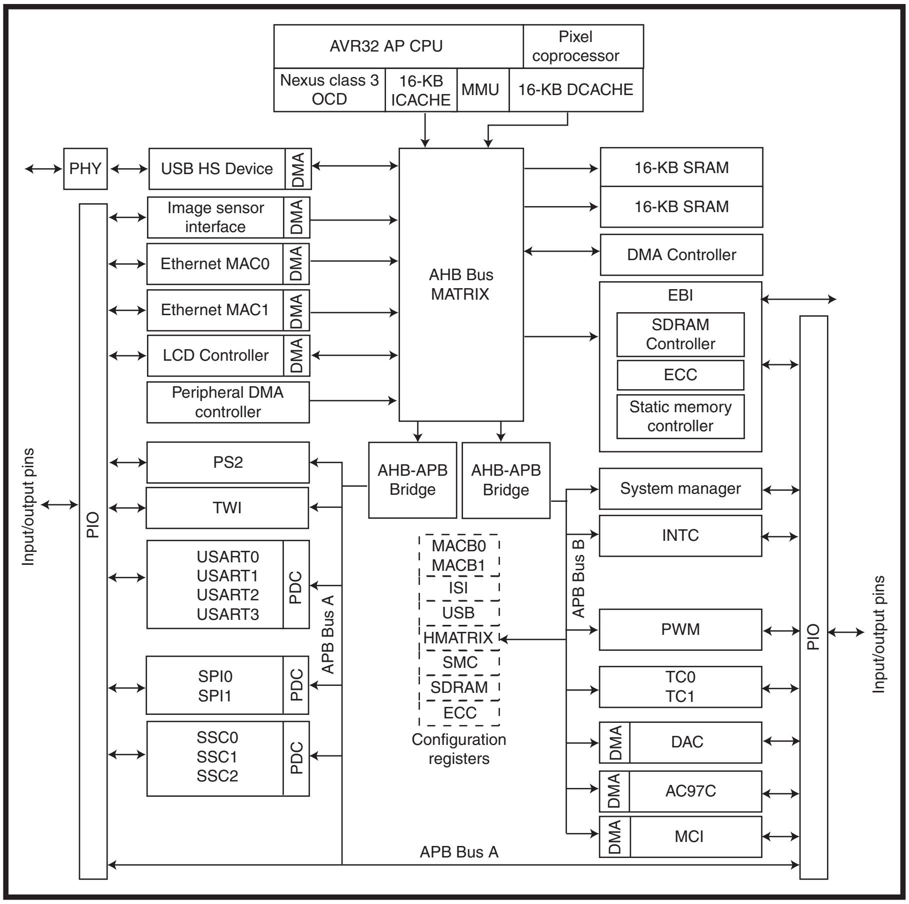


Figure 3—*The "MCU" moniker for the AT32AP7000 is kind of a stretch. Just add memory and Linux and you've got the equivalent of a circa-1980s "UNIX workstation" (à la Sun, MIPS) on a chip.*

inputs is for a 32-kHz real-time clock. The other two are duplicates, each accommodating a 10- to 27-MHz crystal. They drive the PLLs to generate a plethora of synchronous (CPU and on-chip bus) and generic (peripherals and application) clocks. You can get by with a single crystal driving both PLLs if that works for your application.

The bad news is that you need some external memory. The good news is that fully 80-pins are devoted to the cause backed by a six-channel chip select controller that direct connects to up to 64 MB of memory. Virtually all types of memory are supported, including SDRAM, SRAM, flash memory, and even MMC and CompactFlash cards. Note that the data bus supports both 16- and 32-bit modes. The former is a popular solution for cost-sensitive applications, the performance hit mitigated by the on-chip cache and preponderance of 16-bit instructions.

The AT32AP7000 may not be a true MCU (i.e., no flash memory), but it does come with a healthy complement of blue-collar peripherals including PIO, UART, SPI, three 16-bit counter/timers, and a four-channel PWM generator. I note these peripherals are relatively sophisticated and high-performance compared to the afterthought units tacked on to some "computer" chips.

The rest of the peripherals position the chip squarely in the battery-powered gadget fray. Walk the aisles of your local electronics emporium and what do you see? LCD screens, A/V connections, and PC connections. The AT32AP7000 has got 'em all.

The LCD controller is quite capable, supporting a range of resolution and color for both active (TFT) and passive (STN) panels. The AT32AP7000 further ups the ante with a tightly coupled pixel coprocessor. The "Pico" as it's called comprises three vector multiplication units that accelerate video operations such as filtering, scaling, color conversion, and MPEG-4 decoding. For camera applications, there's also an image sensor interface that connects to standard CMOS imagers and generates preview data for direct display via the LCD controller.

For audio, the AT32AP7000 incorporates an AC97 controller and synchronous serial interface for interfacing to standard audio codecs. On the output side, there's also a 16-bit two-channel stereo output sigma-delta DAC.

Making a PC connection is easy with both 10/100 Ethernet and USB to choose from. The Ethernet port includes the MAC and an MII (or reduced pin-count RMII) interface to an external PHY (e.g., wired and wireless). The high-speed (480-Mbps) USB Device Port includes a transceiver, so completing the USB connection is just a matter of adding the connector itself along with three resistors and a capacitor.

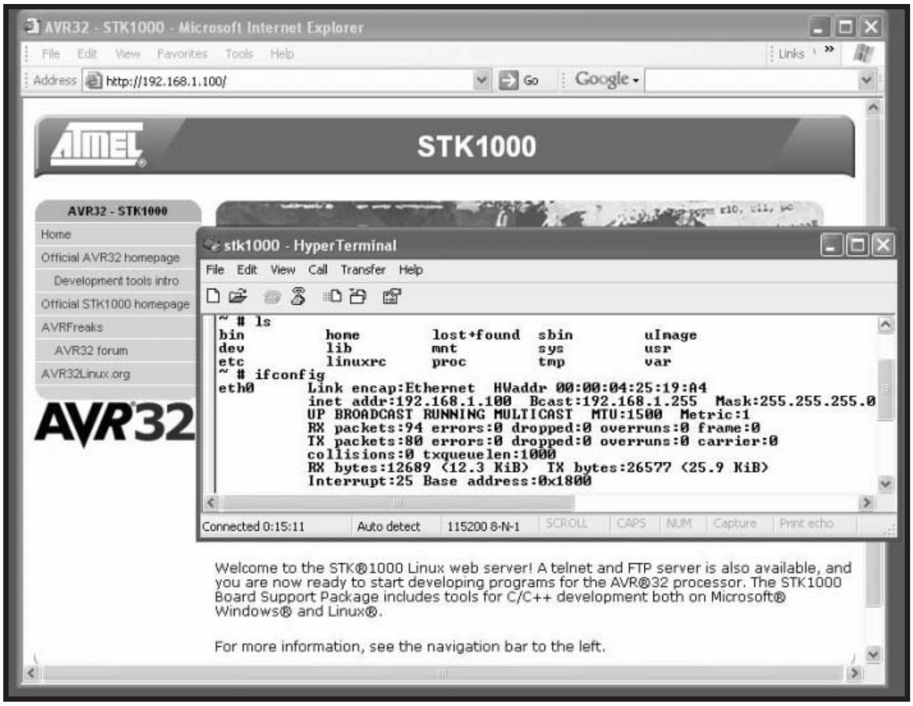
Pulling it all together are a hierarchical matrix of buses, a feature-laden DMA controller (chaining, scatter/gather, and FIFO emulation), and a dedicated peripheral data controller (PDC) that all work together to move data quickly yet unobtrusively (i.e., minimize interference with the CPU).

LITTLE LINUX

I had a chance to fiddle around with the AT32AP7000-based Atmel STK1000 development kit. At $499, it's not an impulse buy, but as you can see in Photo 1 (p. 82), you do get a significant piece of hardware to play with. In addition to 8 MB each of flash memory (×16) and SDRAM (×32), the board includes a color TFT QVGA display and a dozen connectors for the aforementioned on-chip peripherals. Along the right edge of the board, you can also see a PCI connector. But it's not PC compatible; it's just a convenience for prototyping.

With all the connectors, switches, jumpers, and so on, the board looks more complicated than it is. In a real design, the only extra parts that would likely be carried forward (in addition to memory) are the Ethernet PHY and RS-232 transceiver and perhaps the video DAC (if you want to use a CRT instead of an LCD). Indeed, the power subsystem may be the most complicated aspect with a daisy chain of four regulators needed to generate the 5 V (LCD), 3.3 V (AT32AP7000 I/O), 2.5 V (MAC chip), and 1.8 V (AT32AP7000 core) required.

The board has connectors for all three flavors of USB, including Device, Host, and On-the-Go (OTG). The AT32AP7000 includes only a Device port, but the other connectors are routed to the CPU daughterboard, presumably for use by a future Host/OTG capable chip. And while there are PS/2 connectors, note that they use 3.3-V signal levels, which may or may not work with a particular mouse or keyboard (5 V typical).



```
~ # ls
bin          home         lost+found   sbin         uImage
dev          lib          mnt          sys          usr
etc          linuxrc      proc         tmp          var
~ # ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:04:25:19:A4
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:94 errors:0 dropped:0 overruns:0 frame:0
          TX packets:80 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12689 (12.3 KiB)  TX bytes:26577 (25.9 KiB)
          Interrupt:25 Base address:0x1800
```

Welcome to the STK®1000 Linux web server! A telnet and FTP server is also available, and you are now ready to start developing programs for the AVR®32 processor. The STK1000 Board Support Package includes tools for C/C++ development both on Microsoft® Windows® and Linux®.

For more information, see the navigation bar to the left.

Photo 2—*Linux makes a lot of sense for applications that can take advantage of its wealth of standard software, such as TCP/IP networking. Right out of the box, the STK1000 fires up a web server (HTTP), TELNET for remote Linux console access, and FTP to transfer files to and from the MMC card.*

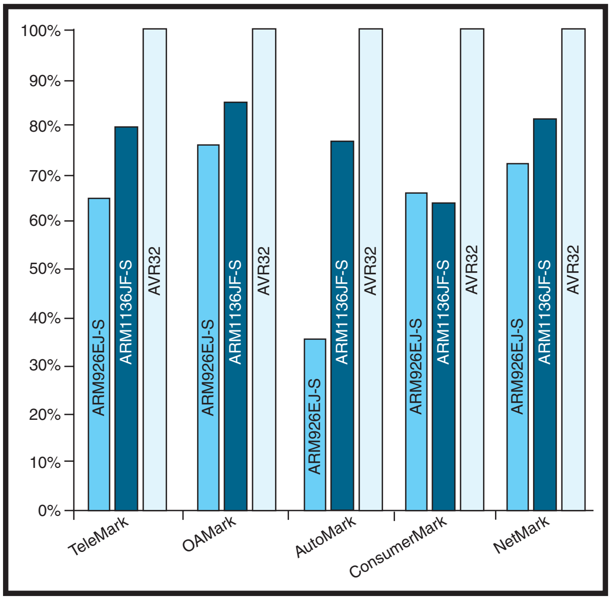


Figure 4—*EEMBC benchmarks give an honest answer (namely, yes) to the question of whether the AVR32 architecture is superior to that of the decades-old ARM design. The question is; Is that the right question?*

The real story of the STK1000, and ultimately the AVR32 itself, unfolds when you plug in the included 256-MB MultiMedia Card (MMC) preloaded with Linux. Atmel may call the AT32AP7000 an MCU, but I had a distinctly good-ol'-days flashback as I QWERTY'd away on the command line. Heck, there's even a WordStar clone editor along with old-time stalwarts VI and EMACs.

The Linux theme is carried forward on the PC side where the GNU software tools that come with the kit run under a UNIX emulator (Cygwin, included on the CD). Atmel has lined up third-party support for alternative tool chains that run directly under Windows, including Ashling, IAR, and their own AVR Studio. There's also mention of the potential to use an Eclipse front-end for the GNU tools. But what came in the box was basically command-line Linux, love it or leave it.

WIZARD OF OS

One conclusion I'm coming to is that terms like MCU, MPU, and DSP are quickly losing their relevance. Atmel can call AVR32 whatever they want. The fact is, it's all three. It's got the peripherals and power consciousness of an MCU, the performance and expandability of an MPU, and the math genius of a DSP. Maybe the real difference anymore is whether a chip is intended to, or at least capable of, running a big-iron OS, namely Windows (XPe or CE), a commercial RTOS (e.g., VRTX), or Linux.

Considering the targeted computer-in-drag applications (e.g., cell-phone, PDA, and MP3 player) of the current AT32AP7000, going with an OS is practically a given. Yes, it's possible to develop stand-alone (i.e., no OS) applications for the chip, but it's tempting to take advantage of all the networking (e.g., TCP/IP) and A/V (e.g., MPEG) software that comes with an OS (see Photo 2).

So is AVR32 an ARM twister? That depends. In its current incarnation, the AT32AP7000 certainly doesn't compete with the true single-chip, single-digit, price-tagged ARM7 and ARM9 MCUs from the likes of Philips, Texas Instruments, Oki, and ST (and, er, Atmel). On the other hand, it's certainly conceivable that Atmel could craft low-cost downsized AVR32s including single-chip versions.

Meanwhile, there's no doubt that the AVR32 architecture is superior to the historic ARM design (see Figure 4). But so is ARM's new Cortex architecture, and arguably those of other competitors. More importantly, how much does that matter? Remember, Atmel isn't claiming that AVR32 has the performance to accomplish things other chips can't. Rather, they're saying they get it done with a slower clock and, thus, lower power.

But in applications like cell phones, PDAs, and MP3 players, the power consumption of the processor chip is only part of the story. For example, the AT32AP7000 itself consumes roughly 20% of the 2.5 W the STK1000 board draws from my benchtop supply. Now, referring to the benchmark results, running an ARM11 at 50% higher clock rate would virtually eliminate the AVR32 performance advantage. Yes, the ARM11 chip power would increase by 50%. Assuming a chip-to-system power ratio of 1:5, that would result in only a 10% hit at the system level. To be fair, the power/performance advantage would become more apparent if Atmel downsizes the AVR32 to target simpler applications where the chip power itself is a more dominant factor.

And what about the argument that the AVR32 is late to a market that's already owned by entrenched heavyweights? That's a serious concern, and it would be easy to consider it a showstopper. Easy, that is, except Atmel themselves did the impossible when they busted into the 8-bit market with their original AVR. One caveat: the technical difference between an AVR and something like a PIC or '51 is like night and day, arguably more compelling than the claimed advantages of AVR32 over its competitors.

My take on the subject is that history shows that no single architecture will dominate the embedded market. That means it is conceivable that the AVR32 could become a significant player. But it's going to take both great parts and great marketing and support, either one alone won't cut it.

Will the AVR32 make it? Can it be late but still great like the original AVR? Or will it end up as little more than a niche player in niche markets? The only thing for sure is that whatever happens, it won't be boring and thank goodness for that.

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.

SOURCES

Development tools for AVR32
Ashling Microsystems
www.ashling.com

AT32AP7000 Microcontroller and AVR32 architecture
Atmel Corp.
www.atmel.com

Embedded Workbench development tools for AVR32
IAR Systems
www.iar.com

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″ × 3″ FORMAT**. Call for current rate and deadline information. Send your disk and digital submission sheet to: IDEA BOX, Circuit Cellar, 4 Park Street, Vernon, CT 06066 or e-mail adcopy@circuitcellar.com. For more information call Shannon Barraclough at (860) 872-3064.

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


PHYTEC

phyCORE® OEMable Single Board Computers

8051
CAN
C166/XC16x/ST10
PowerPC: MPC555/565/5554
x86: Elan SC520
ColdFire: MCF5485
XScale: PXA255
ARM7: AT91M55800/
LPC229x

- sub-miniature SBC subassemblies (as small as 47x55 mm)
- low EMI, multi-layer design plugs like "big chip" into target circuitry
- 32 kB to 128 MB external SRAM and Flash (controller-dependent)
- CAN, Ethernet, RS-232/485; USB, IrDA; ADC, DAC (controller-dependent); free /CS and I/O
- available in Rapid Development Kits w/demos and compilers
- [illegible]: insert-ready COTS subassemblies accompany you from evaluation through prototyping to end production.... accelerating your design cycle and reducing time-to-market

PHYTEC America, LLC ■ 203 Parfitt Way SW, G100 ■ Bainbridge Island, WA 98110 USA
www.phytec.com ■ (800) 278-9913 ■ www.phycore.com

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

DLP Design


Start designing with Freescale's advanced S12X family!

Toll-free: 1-877-963-8996
www.technologicalarts.com


USB-ICP & LCD-Demo Kit

USB-ICP Using a standard USB port, USB-ICP supports In-System Programming (both ISP and ICP mode) for Philips 80C51, LPC9xx, and LPC2xxx ARM7 microcontroller families. In-System Programming uses a two-wire serial interface to program and erase ISP enabled microcontroller devices without removing them from the system. This device is powered over the USB connection, so no external power supply is required! **$69.00.**

LCD-Demo Kit Small footprint "pocket" LCD Demo board with 8-character alphanumeric LCD. Uses a highly integrated Flash based 80C51 device. A single CR2025 3V coin cell battery powers the entire board and the unit is User Re-Programmable via the ICP connector. **$49.00.**

For XA Development Kits and I2C, MDIO and SPI tools, please visit our website.

FDI
www.teamfdi.com
VISA/MC/Amex

Future Designs, Inc.
2702 Triana Blvd
Huntsville, AL 35805
(256) 883-1240
Fax (256) 883-1241


AVOCET SYSTEMS, INC.

Hardware and Software Development Tools

683xx
PowerPC
68HC11
EZ80
Z80
8051
68HC12

- Background Mode Debuggers
- In-Circuit Emulators
- C Compilers
- Assemblers

(207) 596-0080

sales@avocetsystems.com
www.avocetsystems.com


The $69 PLC

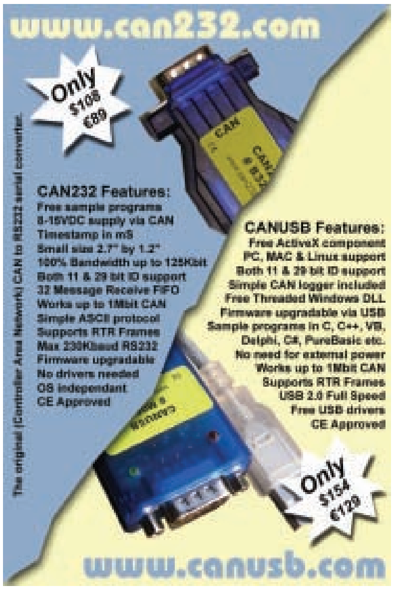
Work as Stand-Alone Ladder Logic PLC.
Or as Smart Remote I/Os of PC/ PLCs.
RS485 allows 256 units to be networked.

Incredibly Easy to Program!
Our software is used by many colleges for teaching PLCs!

Get Free Ladder Logic Simulator:
www.tri-plc.com/cci.htm

Triangle Research

Tel: 1-877-874-7527 - PLC specialist since 1993


www.can232.com

Only $108 €89

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

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

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

Only $154 €129

www.canusb.com


ANSI C software development tools for:
ImageCraft™

NEW!: Global Optimizer

CypressM PSoC
AVR & tinyAVR
HC08,11,12,16
TI MSP430
ARM

Version 7 Tools
Full-featured 45-day demos on our website

hardware kits:
Elektronikladen - HC08 & HCS12 / 12 ref. design & starter kits
CANDIP - AVR/CAN on a chip!
Ethernut - Complete AVR TCP/IP solution

high powered development, garage powered prices!

www.imagecraft.com
info@imagecraft.com
706 Colorado Ave. Palo Alto, CA 94303
(650) 493-9326 • FAX (650) 493-9329

VISA / MC / AMEX
Check / Money Order

...say 'NO' to JURASSIC PRICES!


Get The Whole Picture

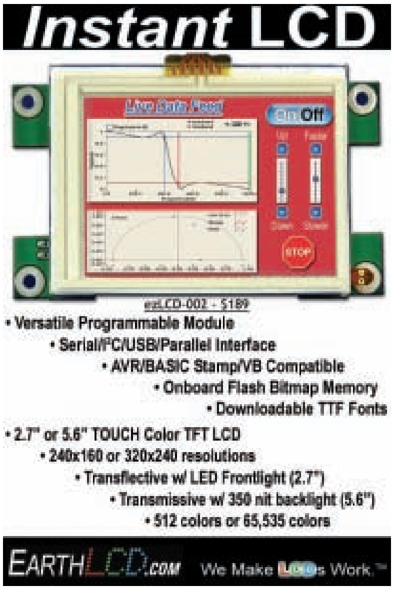
Camera Interface Application Kit for remote monitoring, security and event-capture via the internet.

Kit includes:
- RCM3365
- Serial Camera
- Servos
- Dynamic C®
- Dev. Tools

Buy Now $499

RABBIT Semiconductor
www.rabbitappkits.com

06231


Instant LCD

ezLCD-002 - $189

- Versatile Programmable Module
- Serial/I²C/USB/Parallel Interface
- AVR/BASIC Stamp/VB Compatible
- Onboard Flash Bitmap Memory
- Downloadable TTF Fonts

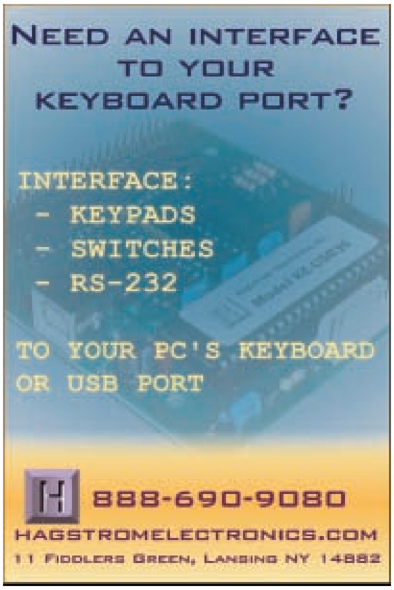
- 2.7" or 5.6" TOUCH Color TFT LCD
- 240x160 or 320x240 resolutions
- Transflective w/ LED Frontlight (2.7")
- Transmissive w/ 350 nit backlight (5.6")
- 512 colors or 65,535 colors

EarthLCD.com We Make LCDs Work.™


JTAG & In-Circuit EMULATORS
IDEs, Compilers, Programmers

80196
AVR®
SENSORY
XEMICS
8051
ARM
PIC®
bq
MAXQ™
MSP430

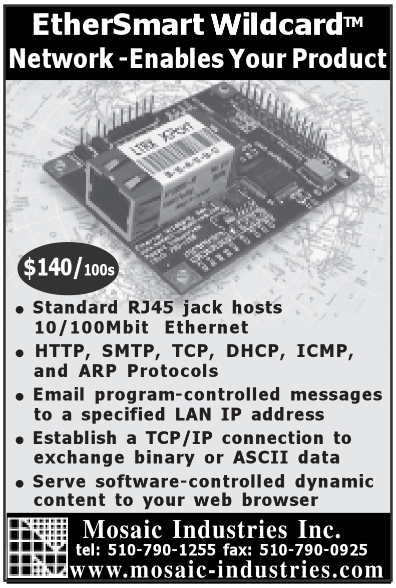
Phyton 718.259.3191
www.phyton.com


NEED AN INTERFACE TO YOUR KEYBOARD PORT?

INTERFACE:
- KEYPADS
- SWITCHES
- RS-232

TO YOUR PC'S KEYBOARD OR USB PORT

888-690-9080
HAGSTROMELECTRONICS.COM
11 FIDDLERS GREEN, LANSING NY 14882


EtherSmart Wildcard™
Network-Enables Your Product

$140/100s

- Standard RJ45 jack hosts 10/100Mbit Ethernet
- HTTP, SMTP, TCP, DHCP, ICMP, and ARP Protocols
- Email program-controlled messages to a specified LAN IP address
- Establish a TCP/IP connection to exchange binary or ASCII data
- Serve software-controlled dynamic content to your web browser

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


I²C™/SMBus

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

MCC
Micro Computer Control

I²C is a trademark of Philips Corporation

www.mcc-us.com

NEW! CAN-4-USB FX

USB to CAN Interface

USB 2.0 Hi-Speed 480Mbps!

Other companies may claim USB 2.0 but if it isn't Hi-Speed it is only 12Mbps.

2006 marks our 10th year of selling CAN interfaces in over 30 countries!

$249 USD.

Zanthic Technologies Inc.
403-526-8318
www.zanthic.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

GoFast™ MATH PACKAGE

- Improve floating point speed without increasing hardware cost
- Replaces compiler math functions
- Intrinsic functions 2-3X faster
- Complex functions 5-10X faster
- In assembly language for speed
- Many processors supported
- Conforms to IEEE 754
- Full source code, no royalties

www.smxrtos.com

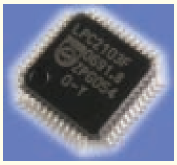
Micro Digital Inc
RTOS Innovators

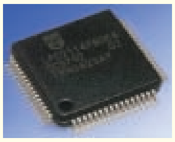
800.366.2491 sales@smxrtos.com

Add USB Peripherals to your system EASILY!

Use the chipset or the complete OEM solutions!

USBwiz™ and uALFAT™ chipsets are USB hosting solutions that contain a complete USB host stack. With simple commands over SPI, UART or I2C, you can access devices such as:



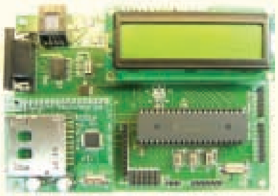


- USB Thumb Drives
- USB Hard Drives
- USB Printers
- USB Joysticks
- USB Mice
- USB Keyboards
- USB Cellphones
- USB serial chipsets such as Silabs, Prolific and FTDI
- Microsoft's XBOX USB controller
- SD and MMC memory cards

USBwiz™ and uALFAT™ also contain full FAT file system to read and write files with full support to folders and subfolders. Simple high level commands such as:

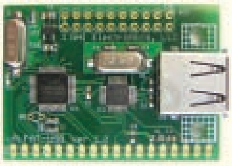
- Open File for Read or Write
- Write to File
- Read from FIle
- Delete File
- Make Folder
- And a lot more commands!

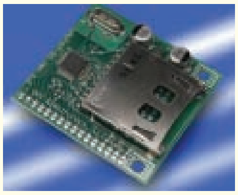
The Development Systems from USBwiz and uALFAT are complete kits with full source code examples and code library.



Use the ready-made USBwiz-OEM™ board for quick results. The OEM board contains everything you need: USBwiz™, Philips' ISP1160, SD card connector and dual USB host connector.

uALFAT-USB™ is the ready-made OEM board solution that incorporates uALFAT™ chipset. Low cost and ease of use makes uALFAT-USB™ the ultimate solution for OEMs and hobbyists.





uALFAT-SD™ is an alternative to uALFAT-USB™ for systems that require an SD or MMC card access. Add millions of bytes with very little effort!

www.ghielectronics.com

www.saelig.com/ghi

Saelig .com
unique electronics
1-888-75AELIG info@saelig.com

MYLYDIA, INC.

Layout Gerber,
Prototype Making

QUICK TURN

PCB & Turnkey
@
The Best Prices

Tel: 800-695-9342

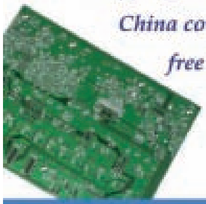
Sales@mylydia.com

WWW.MYLYDIA.COM

CHINA PCB SUPPLIER

- DIRECT SALE FROM CHINA
- PROTOTYPE TO PRODUCTION

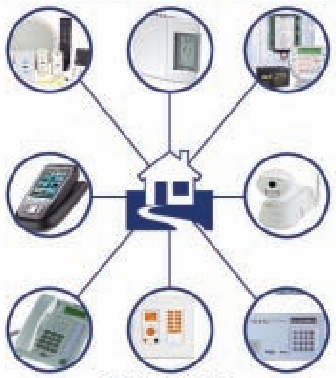
instant online quote
shopping cart ordering system
China competitive prices
free Electrically test



web: http://www.pcbcart.com
email: sales@pcbcart.com
Tel: +86-571-87013819
Add: No. 2 Huixing Road,
Hangzhou, PR China

WWW.PCBCART.COM

SmartHomeUSA



SAVE 10%
ENTER COUPON CODE CELLAR DURING CHECKOUT*

www.SmartHomeUSA.com

DIY Home Automation, Security
& Entertainment Products

Affordable Powerful & Portable

USB Lab Instruments

Oscilloscope $199US

Dual Channel, 20 MS/s
Hardware Preamp w/ 7 Ranges
Pre & Post Trigger Capture
Automatic Measurements
8-bit Vertical Resolution
Variable Trigger Level
Auto, Normal and Single-Shot Trigger

Waveform Generator $139US

Arbitrary DDS Waveform Generation
Four Built-In Waveforms:
Sine, Square, Triangle, Sawtooth
One User Programmable Waveform
0.1Hz to 100kHz
8-Bit Resolution
Amplitude and Offset Adjustments

Open Source Software ■ Windows & Linux

Bonus: Network analyzer software
See our website for details

www.syscompdesign.com
647-839-0325 Toronto, Canada

Low Cost CAN USB Adapter

✓ Simple configuration & use
✓ 10% Discount for first orders Code: GRIDCC10

$139.95
Qty 1

+1 630-245-1445
Naperville, Illinois USA
www.c-a-n.com

gridconnect.

PCBFABEXPRESS
High Quality PCBs @ Low Impact Prices

Get 5 PCBs for $13 each in 5 days

2 layer, 20 sq. in., FR-4, 0.062" thick
FREE Tooling, FREE Soldermask,
FREE Silkscreen

Visit website to see all offers
on 2 to 8 layer PCBs

ORDER ONLINE
408-522-1500
www.PCBFabExpress.com

Also see our fantastic
PCB Assembly prices online

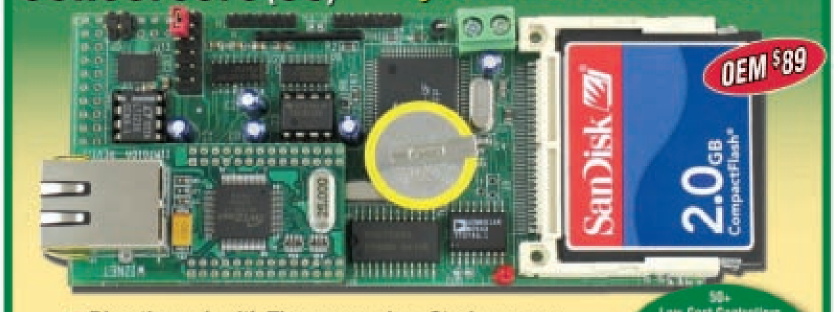
New IP Video Servers

www.ArucaElectronics.com

- Live Video on Web
- User Security
- Web Administration
- Motion Detection
- SDK Available
- FTP and Email
- RJ45, DHCP, DDNS, PPPoE and more

Aruca
electronics

SensorCore(SC)™ *Low Cost Data Acquisition System with 48 24-bit ADCs*



- Directly work with Thermocouples, Strain gauges . . .
- 100 M BaseT, FAT file system and CompactFlash.
- 2.0" x 4.5", C/C++ programmable, 80MHz x86, 48 24-bit ADCs, DAC, RS232, I/Os. Standalone SBC.

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

TERN INC. 1724 Picasso Ave., Suite A, Davis, CA 95616 USA
Tel: 530-758-0180 • Fax: 530-758-0181 www.tern.com • sales@tern.com

ValueCAN

The High Value Tool For Controller Area Network

- USB to CAN
- Simple software analyzer included
- DLL with examples for custom applications
- PC isolated from CAN
- 100% bandwidth at 500Kb

Only $295

ICS RADDAQ

www.intrepidcs.com/vcan

8/16-bit EEPROM | Ser. EEPROM | Flash EPROM | GAL / PALCE | Most MCU's | Low Voltage to 1.3V. | DIL dev. w/o Adapter.

Conitec's last generation Galep-4 employs ASIC universal pin technology for each pin of 40 pin ZIF-socket. **9000+device library** / lifetime free updates. Programs 8/16 bit EPROM'S, EEPROM's, 0-Pwr RAM, FLASH, Serial EEPROM's, GAL, PALCE, microcontrollers such as 87/89xxx, PIC, AVR, ST62, etc. Low voltage devices down to 1.3V. No adapter required for DIL devices. 8 Hrs. operation on battery (AC charger included). Runs **WIN 98,NT,ME, 2000, XP** with Hex/Fuse Editor. Remote control from other apps, (e.g. VisualBasic). Substitutes high priced universal programmers e.g. ALL-11 (HILO) or LAB-TOOL-48 (ADVANTECH) **Provides virtually matching performance at 1/3 to 1/5 the price.** Info, orders, softwr : **619-462-0515**

A PALM-SIZED PROGRAMMER HANDLES OVER 9000 DEVICES ?

ISO 9001:2000 CERTIFIED PROFESSIONAL TOOLS 9001: 2000

9000 DEVICES

SMALL PACKAGE. BIG FEATURES.

SALES, SUPPORT / EMAIL: CONTACT@CONITEC.NET / WWW.CONITEC.NET TEL: 619-462-0515. FAX: 619-462-0519

$189

Tiny Linux controller

16MB fast SDRAM	10/100 Ethernet
64MHz Coldfire MCU	3 serial ports (RS232)
4.5 MB flash memory	1 CAN port
SD card socket (to 2GB)	LCD/KPD port
Eclipse/CDT dev. env.	Free serial debugger

55 I/O pins & capabilities to burn...

www.steroidmicros.com

PROTOTYPE CIRCUIT BOARDS

$85 US for two 5" x 6" 2-layer boards

Shipped NEXT BUSINESS DAY if data is received by 1:00 pm EASTERN

www.apcircuits.com

AP Circuits

(403) 250-3406

VISA MasterCard staff@apcircuits.com

CCS PCWH C Compiler

The easiest way to create optimized code on PIC® MCUs.

- Co-operative RTOS
- Extended Memory Support with Typemod
- User-friendly Integrated Development Environment
- Built-in Functions, Libraries & Examples

Only $425

PIC® is a registered trademark of Microchip Technology Inc. in the U.S. and other countries.

CCS Inc.

Call Sales: 262-522-6500 x35 • Email: sales@ccsinfo.com

www.ccsinfo.com/picc

Low Cost EDA Software
For Windows 98, XP, NT, 2000

- Circuit design package with schematic entry, circuit board layout and simulation (SPICE and digital) for only $599
- Buy modules for as low as $119
- Purchase and download products instantly from our website!
- Libraries include latest micros, memories, analog, and others

Contact **Mental Automation** at
(253) 858-8104
http://www.mentala.com

Front Panels?

Unrivaled in price and quality for small orders

Download the free ›Front Panel Designer‹ to design your front panels in minutes

Order your front panels online and receive them just in time

www.frontpanelexpress.com

Serial • Ethernet
Web Server

Simple, DTE, DCE
No SW Changes

$99.95
Qty 1

+1 630-245-1445
Naperville, Illinois USA
www.gridconnect.com

gridconnect

NAND FLASH
FILE SYSTEM

- Wear Leveling, Garbage Collection
- Power-Fail Safe
- ECC and Bad Block Handling
- File System with Std C Lib API
- FAT File Structure
- Page Caching to Maximize Performance and Minimize RAM
- 20KB Code Size
- RAM Required < 1% Flash Size

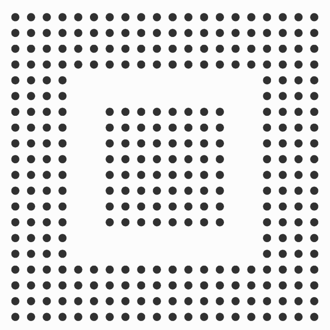
www.smxrtos.com/
ffscc.htm

Micro Digital Inc
RTOS Innovators
800.366.2491 sales@smxinfo.com

Weather
Instruments
for PCs

AAG
electrónica

www.aagelectronica.com

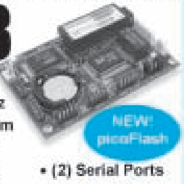


Start at
$20

*BGA Mounting
*BGA Repairing
*Ball Planting

WWW. EzPCB. 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:

$129
Development System

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

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

JK microsystems

NOW YOU CAN HAVE IT ALL!

- Easy to Use
- Low Cost
- Low Power
- Tiny Size
- High Speed
- DOS Environment

with the

From
$99
Q.100

DOS Stamp™

Easy Software Development: Use your C/C++ or Basic compiler to produce DOS EXE. Download EXE to flash disk via serial port. EXE runs on power up.

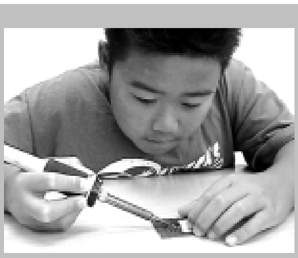
Standard Features: BIOS & DOS-ROM, 128K flash disk, 512K SRAM, 40 MHz CPU, 16 digital I/O (opto rack interface), 2 RS-232, 2 timer/counters, simple bus interface, real-time clock

Options: 8-ch 12-bit ADC, flash disks up to 288 MB, 1 RS-485, TCP/IP

Tiny Size, Low Power: 2"x2.6", 5V @ 200 mA at full speed, 40 mA at lowest speed.

Visit http://www.bagotronix.com for info, prices, and FAQs

BAGOTRONIX
2900-1 Crescent Drive
Tallahassee, FL 32301
850-942-7905 phone & fax



ANYONE
Can Now Easily
Hand Solder Surface-
Mount Components!
Even A 10
Year Old!

www.schmartboard.com

PRINTED CIRCUIT BOARDS

QUALITY PRODUCT
FAST DELIVERY
COMPETITIVE PRICING

PROTOTYPE
through
PRODUCTION

- 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

SPECIAL OFFER:
10 pcs (3days) 1 or 2 layers $249
10 pcs (5days) 4 layers $695
(up to 30sq. in.ea.) includes tooling, artwork, L.P.I. mask & legend

PULSAR, INC

9901 W. Pacific Ave. Franklin Park, IL 60131 • Phone 847.233.0012
Fax 847.233.0013 • www.pulsar-inc.com • sales@pulsar-inc.com

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

DOWNLOAD the HIDmaker FS Test Drive today!

Trace Systems, Inc.
www.TraceSystemsInc.com
301-262-0300

SURFACE MOUNT PROBING ADAPTERS

- BGA, QFP, PLCC, SOIC, & QFN
- SMT adapters solder to target board
- Modular probe boards
- High quality ZIF sockets
- Gold plated test points

In-circuit test your SMT IC's with logic analyzer or scope. Simply solder SMT package emulator and attach probe board.

Also Quick Turn Customs.

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

USB Data Acquisition
ADU208 -USB Relay I/O Interface

Complete SDK Online at: www.ontrak.net

FEATURES
-8, 5-AMP relay outputs
-8, ISOLATED digital inputs
-Port Powered, Aux 5VDC output $189.00 QTY 1

Other Models;
ADU200- 4 Channel Version with RS232 $139.00
ADR218- Solid-State Version 8 Channel $225.00
ADU100- 3 CH, 16-Bit ISOLATED Analog Inputs, PGA, 4 digital I/O, RS232 and 5 AMP Relay Output $199.00

ONTRAK CONTROL SYSTEMS INC.
PH: (705) 671-2652 Fax: (705) 671-6127
www.ontrak.net

JStamp!®

FAST 32-bit native-execution Java
Executes 3 million Java byte codes/sec
compact 1 x 2 inch DIP40 package
Real J2ME CLDC Java (threads, floats)
2048 KB Flash & 512 KB SRAM

www.jstamp.com
SYSTRONIX®
Salt Lake City, Utah, USA
Java is a TM of Sun Microsystems, Inc. JStamp is a TM of Systronix, Inc

NEW
RS232 to TCP/IP

- TCP/com™ v2.0, RS232 to TCP/IP software. Plus TCP/IP to RS232.
- WinWedge™, RS232 or TCP/IP data direct into any Windows app. - Excel, Access, etc.

TAL tech
Free 30 day evals at
www.taltech.com

Order online at:
www.melabs.com

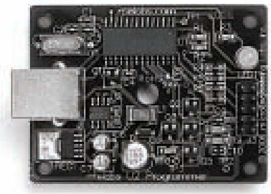
microEngineering Labs, Inc.
Development Tools for PIC® Microcontrollers

Phone: (719) 520-5323
Fax: (719) 520-1867
Box 60039
Colorado Springs, CO 80960

USB Programmer for PIC® MCUs $119.95 (with accessories)

RoHS Compliant

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

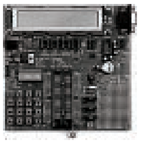


Includes:
Programmer, USB Cable,
ZIF Programming Adapter for 8 to 40-pin DIP,
Software for Windows 98/Me/NT/2K/XP

EPIC™

Parallel Port Programmer starting at $59.95
Serial Port Programmer starting at $79.95

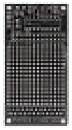
LAB-X Experimenter Boards



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®

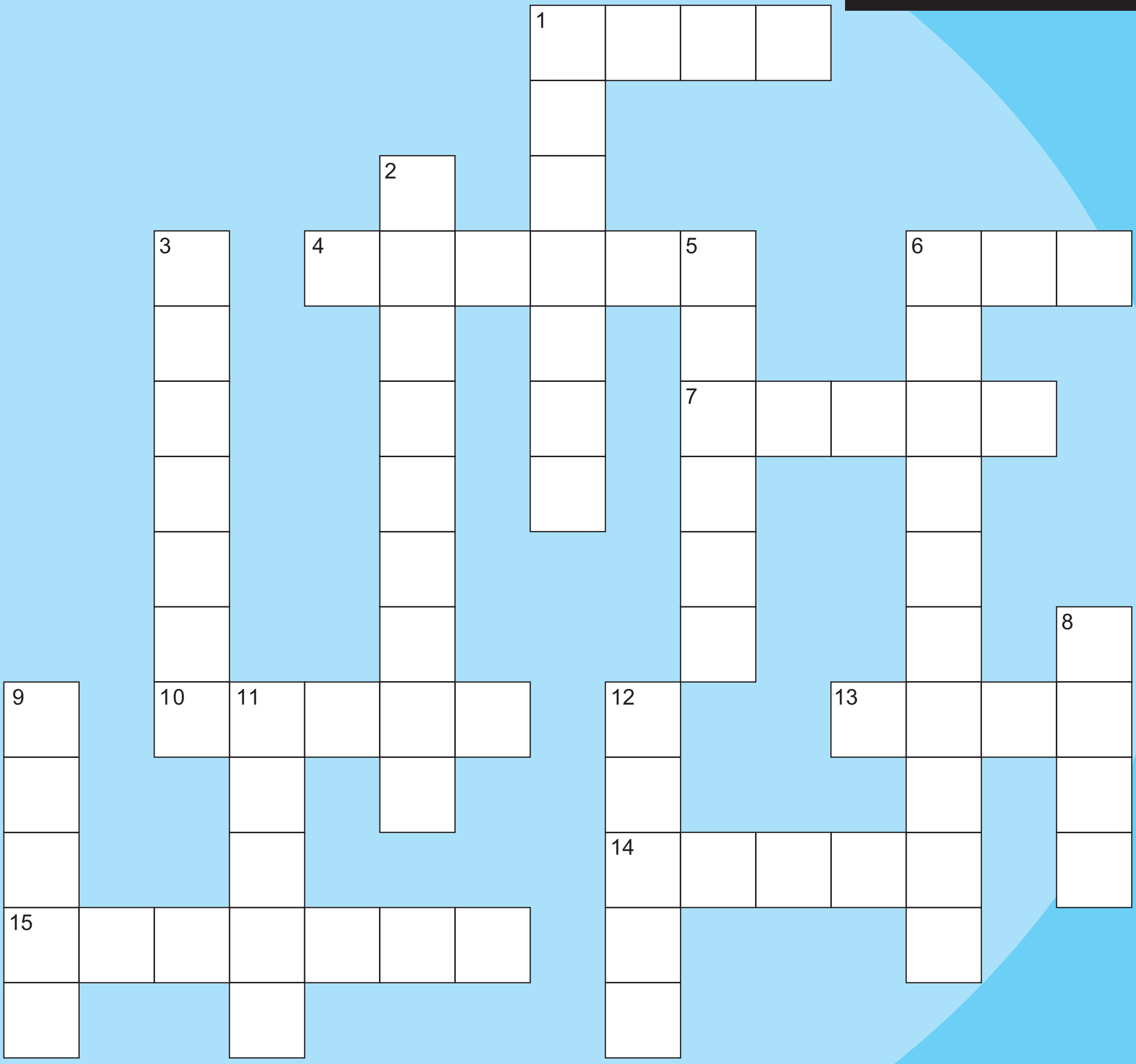
Easy-To-Use BASIC Commands
Windows 9x/Me/2K/XP Interface

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

PICBASIC PRO™ Compiler $249.95
Supports All PICmicro® MCU Families
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

CROSSWORD



Across

1. The actor who played Rick Deckard in *Blade Runner* (1982).
4. Clean up your hard drive.
6. End of line
7. The planet that has a moon named Charon.
10. Abnormal end
13. Keep it simple, stupid.
14. A person who disrupts a chat room or forum.
15. What you see is what you get.

Down

1. The name of the room in which *Circuit Cellar* columnist Fred Eady builds his systems.
2. An antonym for increment.
3. A tendency to remain in state of rest.
5. A soft mineral used in plaster. Hint: "block."
6. The "X" in XHTML.
8. Your left fingers on a keyboard.
9. A blog about legal topics and issues.
11. pH > 7
12. A machine that's used for cutting and shaping metal.

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	
91	AAG Electronica, LLC
88	All Electronics Corp.
90	AP Circuits
81	Arcom Control Systems
39	Arcturus Networks
89	Aruca Technologies
7	Atmel
5	Avnet
86	Avocet, Inc.
91	Bagotronix, Inc.
77	Belllin Dynamic Systems Inc.
42	Bitscope Designs
19	CadSoft Computer, Inc.
64	Circuit Specialists, Inc.
3	Comfile Technology, Inc.
90	Conitec
90	Custom Computer Services, Inc.
72	Custom Solutions, Inc.
65	CWAV
1	Cypress MicroSystems
86	DLP Design
77	Decade Engineering
87	Earth Computer Technologies
28	Efficient Computer Systems
55	Elprotronic
54	EMAC, Inc.
23	ExpressPCB
91	ezPCB
86	FDI-Future Designs, Inc.
26, 27	Freescale Semiconductor, Inc.
90	Front Panel Express
59	Futurelec
89	General Circuits, Inc.
89, 91	Grid Connect
87	Hagstrom Electronics
34	HI-TECH Software, LLC
73	Holmate/Holtek Semiconductor, Inc.
87	IMAGEcraft
63	Infineon
90	Intec Automation, Inc.
90	Intrepid Control Systems
71	Intronix Test Instruments, Inc.
92	Ironwood Electronics
2	Jameco
77	Jeffrey Kerr, LLC
64, 91	JK microsystems, Inc.
50	Keil Software
51	LabJack Corp.
51	Lakeview Research
14	Lantronix
87	Lawicel HB
32	Lemos International
59	Linx Technologies
8	Logic Supply
47	Luminary Micro
41	Matrix Orbital
11	Maxstream
87	MCC
90	Mental Automation
17	Microchip
88, 91	Micro Digital, Inc.
92	microEngineering Labs, Inc.
18	mikroElektronika
87	Mosaic Industries, Inc.
95	Mouser Electronics
89	Mylydia, Inc.
C2	NetBurner
9	NKK Switches
51	Nurve Networks LLC
92	Ontrak Control Systems
C4	Parallax, Inc.
89	PCB Fab Express
82	PCB-Pool
86	Phytec America LLC
87	Phyton, Inc.
54	Pioneer Hill Software
91	Pulsar, Inc.
31	R4 Systems, Inc.
50, 79	Rabbit Semiconductor
87	Rabbit Semiconductor
8	Reach Technology, Inc.
66, 88	Saelig Company
91	Schmartboard
15	Sealevel Sytems
33	SEGGER Microcontroller Sys. LLC
55	Sierra Proto Express
29	Silicon Laboratories, Inc.
89	Smart Home Systems
72	SoC Conerence
21	Spark Fun Electronics
89	Syscomp Electronic Design Ltd.
92	Systronix
92	TAL
C3	Tech Tools
48, 49	Technologic Systems
86	Technological Arts
89	Tern, Inc.
11	Tianma Microelectronics
10	Tibbo Technology, Inc.
92	Trace Systems, Inc.
86	Triangle Research Int'l, Inc.
16	Tri-M Systems, Inc.
88	Zanthic Technologies, Inc.

Preview of November Issue 196
Theme: Embedded Development

USB Flash PIC Programmer

Smart Probing: An R5F2113-Based Soil-Monitoring System

Electronic Gear Control: Add Electronic Gears to a Metal Lathe

Low-Cost Gas Leak Detector

Control Panel Design: Implement a Modular Physical User Interface System

Open-Source AVR Development

TV-Based Oscilloscope

APPLIED PCs **Stand-Alone Ethernet Control**

FROM THE BENCH **Smart Character Display:** MCU-Based Information Messaging Made Simple

SILICON UPDATE **Memory Lane Change**

ATTENTION ADVERTISERS

December Issue 197 Deadlines

Space Close: Oct. 12
Material Close: Oct. 19

Theme:
Analog Techniques

Call Shannon Barraclough
now to reserve your space!
860.872.3064
e-mail: shannon@circuitcellar.com

ST Microelectronics STR7 (ARM®) 32-bit Microcontrollers

Combined ARM7TDMI® 32-bit RISC core (high-performance, very low power, and very dense code), with comprehensive set of peripherals and ST's latest 0.18µ embedded Flash technology. Small, highly integrated microcontrollers expand application capabilities and lower total system cost.

STMicroelectronics mouser.com/st/a

Sharp Microelectronics LH75401 16/32-bit BlueStreak™ Microcontroller with Color LCD Controller

Low-cost microcontroller utilizes ARM7TDMI-S® core with wide function variety. 32KB on-chip SRAM, 3 UARTs, SSP, CAN version 2.0b, three 16-bit counter/timers, analog-to-digital converter with integrated touch screen controller, watchdog timer, and low-voltage detector.

SHARP MICROELECTRONICS OF THE AMERICAS mouser.com/sharpsma/a

Your New Designs Need the Best ARM® Technology!

You need to reduce development time, expand application capabilities, and drive down total system costs. That's why we deliver the application design flexibility you need with the best ARM-based products available today!

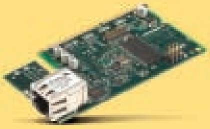
Experience Mouser's time-to-market advantage! Our vast selection of the NEWEST products, NEWEST technologies, no minimums, and same-day shipping on most orders, gets you to market faster. We make it easy to do business with Mouser!

mouser.com **(800) 346-6873**

MOUSER® ELECTRONICS
a tti company

Newest Products for Your New Designs

Digi International ConnectCore™ 9C



Powerful ARM9-based core module enables OEMs to design-in core processing functionality and networking capabilities with a single, high-performance solution. Delivers complete embedded network connectivity; additional bandwidth handles many sophisticated embedded applications.

Digi® mouser.com/digi/a

Keil Software RealView® MDK-ARM Microcontroller Development Kit

Supporting most ARM® derivatives from numerous manufacturers, this kit is perfect for industry-standard compilation tools and sophisticated debugging support.

KEIL™ An ARM® Company mouser.com/keil/a

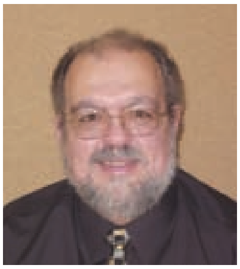
Oki Semiconductor ML67Q4050/Q4060

World's smallest packaged ARM® processors with 33.33-MHz, 32-bit ARM7TDMITM core (64 or 128 KBytes of 32-bit wide zero-wait state FLASH memory); 16 KBytes SRAM; multiple serial interfaces and peripheral functions.

Oki Semiconductor mouser.com/okisemi/a

The NEWEST Semiconductors | Passives | Interconnects | Power | Electromechanical | Test, Tools & Supplies from Mouser Electronics

Mouser and Mouser Electronics are registered trademarks of Mouser Electronics, Inc. Other products, logos, and company names mentioned herein, may be trademarks of their respective owners.



PRIORITY INTERRUPT

by Steve Ciarcia, Founder and Editorial Director

In Your Opinion

Every once in a while we like to make sure we're on the right track. One of the ways we do this is with a reader survey that asks readers specific questions about themselves and us. The knowledge we gain helps us concentrate on publishing the high-quality editorial you like as well as finding advertisers with products you appreciate.

So what did we learn this time? Well, most of you are in your peak professional years. You tend to prefer working in companies with less than 100 employees. You're all overeducated and make a lot of money. Some of you have really weird ideas. But, most of all, you like *Circuit Cellar*. So, I guess that means I still have a job.

Seriously, the survey reinforced my belief that *Circuit Cellar*'s direction should always be evolutionary rather than static. Occasionally, I'm criticized for making *Circuit Cellar* too "high-level." We were all beginners at one time, but now we are professionals. As a magazine, *Circuit Cellar* can either choose to have an editorial level that readers pass through on their career climb or have an editorial focus that tracks evolving technology and stays with readers at the professional level. Overwhelmingly, readers concur with our direction, but they also want us to know that part of that evolution now includes an interest in articles and projects dealing with timely environmental issues like solar panel controllers, fuel/consumption monitors and gauges, and power-saving systems.

The survey is also intended to determine which products should be advertised based on a reader's professional activities and which components, MCUs, software tools, or soft cores they intend to purchase in the future. The conclusion, from a marketing perspective, is that programmable logic is an inexorably growing trend despite challenges from overly complex/pricey tools. And, the budget-priced "deeply embedded" programmable logic market is growing thanks to Moore's law and the emergence of very capable flash-based parts from some key suppliers.

Finally, contests are still a hot draw, and I'm only encouraged more with reader comments like these:

Circuit Cellar-managed contests are an added bonus to the subscription. These contests are a great way to learn about new tools and the technologies that they represent. Reviewing completed contest entries can spark new design ideas. Great work!

A nice opportunity to develop new applications that you've been thinking about but never came to realize. The possibility it might be published even makes you want to put that little vital extra effort into it.

Circuit Cellar readers love participating in our design contests and poring over all the unique embedded systems design projects they provide. Why is this so significant? Consider that our most recently concluded design contest was centered on using higher-functioning ARM MCUs. Not only did we receive an exceptional amount of interest in this contest up front, but our readership survey indicates that the expanded web presentation of these ARM contest projects is perhaps one of the most popular to date among our readers. According to the survey, readers put ARM among their top four selections when it came to a wish list of sponsors for the next *Circuit Cellar* design contest. This is consistent with our conclusion that our readership isn't shying away from more complex, higher-functioning MCU architectures. They're following the evolution in the industry and embracing it. The survey reveals that our readers are still strongly considerate of cost-effective, 8-bit design solutions (65%), but the number of potential ARM users among them will almost double in the immediate future to more than 30% (the highest of any CPU we had listed in the "intend to use in a future design" category).

This documented interest in more complex processors and MCUs corroborates the fact that *Circuit Cellar*'s audience is truly comprised of hands-on design engineers and serious electronic devotees. They are actively engaged in obtaining the precise knowledge and experience necessary to satisfy the most difficult on-the-job design challenges—whether they involve souped-up 8-bit, practical 16-bit, vigorous ARM, or refined soft-core/FPGA. Stay tuned for more.

Steve

steve.ciarcia@circuitcellar.com

DigiView

- Automatic Real-time Hardware Compression can save up to 5 minutes of data @ 10ns.
- Display I2C, Synchronous (SPI), Asynchronous (RS-232), State, Boolean, Bus and Analog Data.
- Edge and Pattern Triggers.
- Pattern Searches with Match, Duration & Animation.
- Specialized Sequential Searches for Serial and State Mode Signals.
- Drag & Snap Markers.
- Dual Waveform View with signal edge snapping.
- Multi-Signal Data Tables with Link Groups.
- Specialized Exports and List Views for Serial Signals.
- Print or Save Images with custom comments.
- USB Powered for Portability.
- USB 1.1 & 2.0 compatible.

100 MHz, 18 Channel, Portable Logic Analyzer

Auto Hardware Compression Captures
from 128K to 30 Billion Samples
@10ns resolution

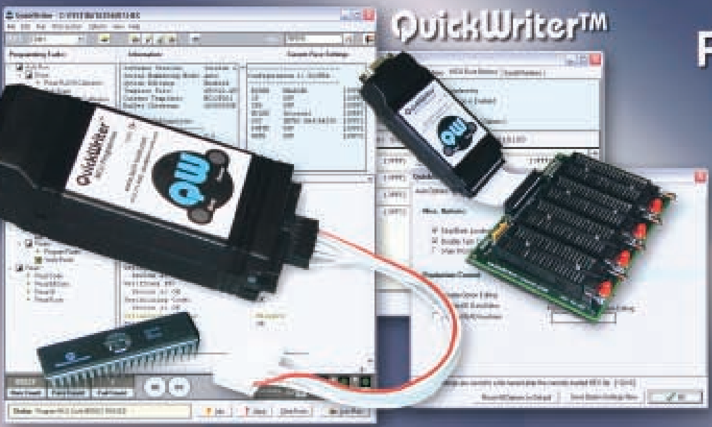
only $499.00

Now with Serial Signal Analysis

Download and try the Software!

Cables & Micro-Clips Included

Professional Capture & Analysis Software Included



PICmicro® MCU Programmer

Multi-Function In-Circuit & Gang Operation

only $199.00

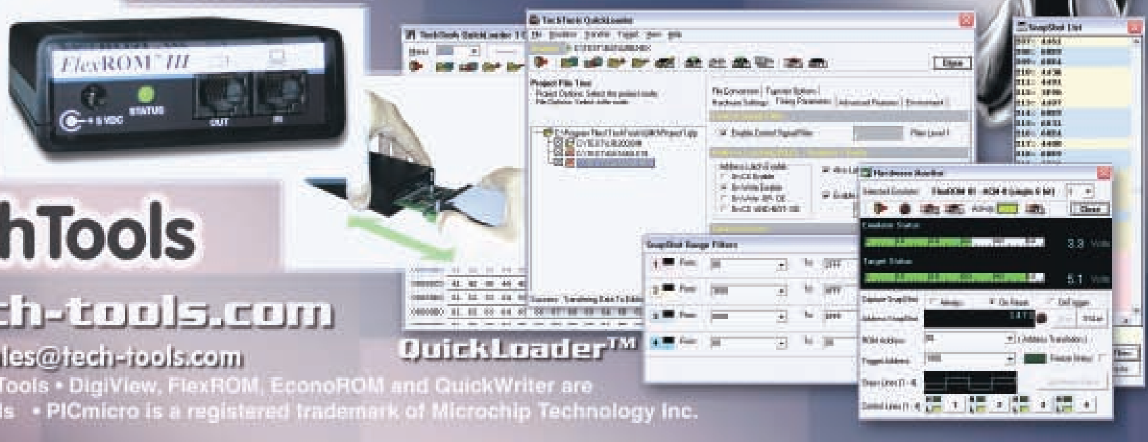
- Supports 10, 12, 14, 16 & 18F Series PICmicro MCUs.
- Edit Code, EE Data, IDs and Configuration Fuses.
- Auto or Manual Serialization in Code or EE Data.
- Firmware auto synchronizes to Software version.
- Control Files protect Option Settings.
- Accepts Command-Line Parameters.
- Tests and Monitors Voltages.
- Control Signal for Self-Powered Circuits.
- Single and 4 Gang Adapters available.
- Includes Cables, Software & U.S. Power Supply.

EconoROM III™

EPROM & FLASH Emulation at its best!

from $179.00

FlexROM III™



TechTools

www.tech-tools.com

(972) 272-9392 • sales@tech-tools.com

Copyright © 2006 TechTools • DigiView, FlexROM, EconoROM and QuickWriter are trademarks of TechTools • PICmicro is a registered trademark of Microchip Technology Inc.

NEW!

SERIAL AND USB COMPATIBLE KITS

from Parallax, Inc.

We've added our new USB to Serial (RS-232) Adapter to our Serial kits for maximum versatility!

The Board of Education® Full Kit is the perfect compliment to our many programming tutorials.

BASIC Stamp® Discovery Kit (#910-27207)
Our most popular starter kit features the "What's a Microcontroller?" (WAM) curriculum and includes all of the BASIC Stamp hardware required to complete over 40 activities and challenges.

Board of Education Full Kit (#910-28103)
A concise package of all the hardware you need to begin building projects with the BASIC Stamp microcontroller. You supply the components, parts, and power supply to build projects.

Our best, most complete way to get started programming the BASIC Stamp microcontroller.

Order these and other Serial and USB compatible specials online at www.parallax.com or call our Sales Department toll-free at 888-512-1024 (Mon-Fri, 7am-5pm, PT).

www.parallax.com PARALLAX INC

BASIC Stamp and Board of Education are registered trademarks of Parallax, Inc.