#145 August 2002

www.circuitcellar.com

CIRCUIT CELLAR®

THE MAGAZINE FOR COMPUTER APPLICATIONS

ANALOG TECHNIQUES

Trouble-Free Audio System

Are You Grounded?

Embedded Fixes

Open-Source HCS



08>
7 25274 75349 9
$4.95 U.S. ($5.95 Canada)

Are Stamps in your Class?

Stocking your classroom.101



The **Stamps in Class** program was created to provide educators, students and engineers the material they need to learn microcontroller programming and interfacing. We've published six **curriculum** for you to choose from. Each curriculum has its own strengths in introductory PBASIC programming, sensor calibration and interfacing, data logging, motor control and PID control (proportional - integral - derivative). See the descriptions on our Stamps in Class web site (www.stampsinclass.com) or download the books for a quick review. Each text has an accompanying parts kit which you can purchase from Parallax or build your own.

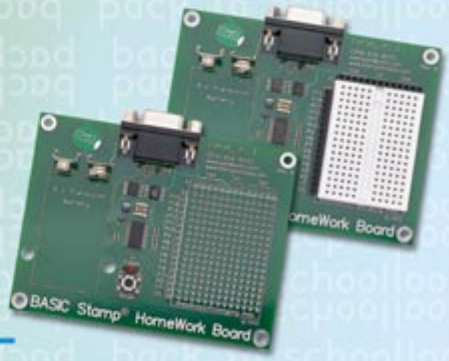


Educators stock their classes with the **Board of Education Full Kit** (28102). The Board of Education is the project board for BASIC Stamp microcontroller projects, including the Stamps in Class curriculum. The board was designed in coordination with our educational customers to teach microcontroller interfacing and programming. Even if you aren't using the Parallax published curricula, the Board of Education is still an ideal set of hardware for instructor-authored lessons. The Board of Education is available in a Full Kit that includes a BS2-IC module, power supply, and serial cable, or may be purchased individually (#28150).



Getting the class started with our **Robotics! curriculum** is easy with Parallax products. The **Boe-Bot** (28132) is a wheeled robot with a well-developed educational tutorial demonstrating the essentials of motor control, sensor interfacing and program structuring. The Boe-Bot has a number of add-ons for line following, sonar and speech generation.

Once you've learned the Boe-Bot, try moving on to the **Advanced Robotics curriculum** featuring the **Toddler** (#27310, #27311). This bipedal robot has 34 unique movements, and is capable of almost everything a Boe-Bot can do but requires much more emphasis on programming and software/hardware interaction.



Educators can send students home with the new **BASIC Stamp HomeWork Board**. These boards were designed in response to requests from educators who desired a low-cost BASIC Stamp 2 that students could either take home or dedicate towards a permanent project. Less expensive than a Board of Education Full Kit. Available in 10 packs with either a breadboard (#28157) or through-hole (#28158) prototype area.

PARALLAX

BASIC Stamp is a registered trademark of Parallax, Inc.

For more information visit www.stampsinclass.com or call toll-free in the US at 888-512-1024

got LCD?

12.1 TFT LCD Monitor Kit $299



P/N SK-2005R

1,000,000 LCD's in Stock

- LCD Character Modules
- LCD Serial Displays
- LCD Video Monitors
- LCD Replacements
- LCD TFT Displays
- LCD Controllers
- LCD Graphic Displays
- LCD Touch Monitors
- LCD PC Monitors
- LCD Vehicle PC
- LCD Panel PC
- LCD Buyers

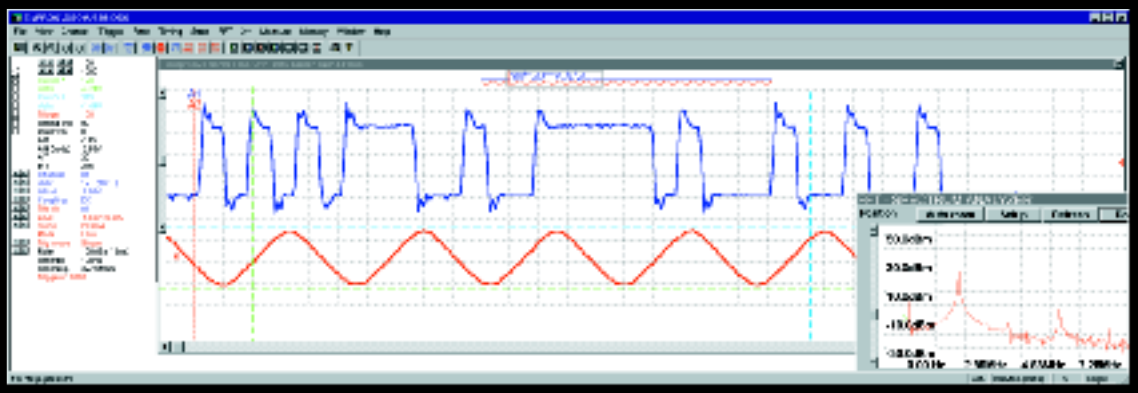
EARTHLCD.com

"The World Wide LCD Source"

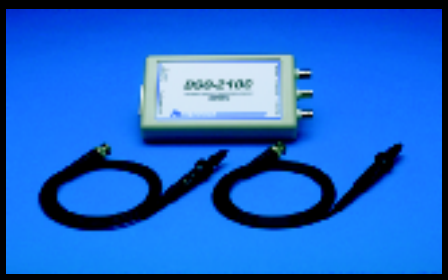
32701 Calle Perfecto - San Juan Capistrano, California 92675

Ph:(949) 248-2333 Fax: (949) 248-2392 lcdking@earthlcd.com

Digital Oscilloscopes

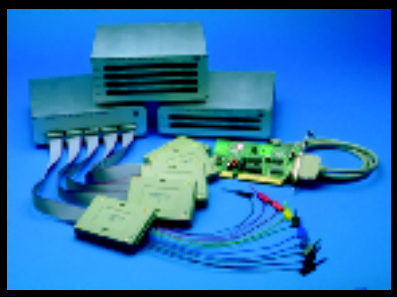


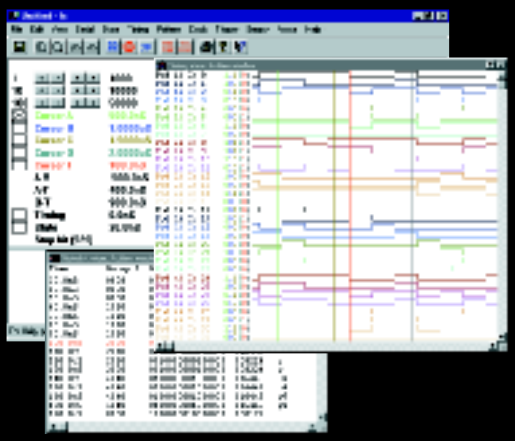
- 2 Channel Digital Oscilloscope
- **100 MSa/s** max single shot rate
- 32K samples per channel
- Advanced Triggering
- Only 9 oz and 6.3" x 3.75" x 1.25"
- Small, Lightweight, and Portable
- **Parallel Port** interface to PC
- Advanced Math options
- FFT Spectrum Analyzer options

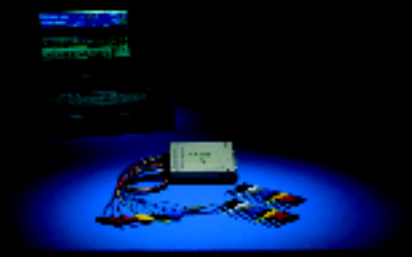


DSO-2102S $525
DSO-2102M $650
Each includes **Oscilloscope**, Probes, Interface Cable, Power Adapter, and software for Win95/98, WinNT, Win2000 and DOS.

Logic Analyzers







- 40 to 160 channels
- up to 500 MSa/s
- Variable Threshold
- 8 External Clocks
- 16 Level Triggering
- up to 512K samples/ch
- Optional Parallel Interface
- Optional 100 MSa/s Pattern Generator

LA4240-32K (200MHz, 40CH)	**$1350**
LA4280-32K (200MHz, 80CH)	**$2000**
LA4540-128K (500MHz, 40CH)	**$1900**
LA4580-128K (500MHz, 80CH)	**$2800**
LA45160-128K (500MHz, 160CH)	**$7000**

All prices include Pods and Software

- 24 Channel Logic Analyzer
- 100MSa/S max sample rate
- Variable Threshold Voltage
- Large 128k Buffer
- Small, Lightweight and Portable
- Only 4 oz and 4.75" x 2.75" x 1"
- Parallel Port Interface to PC
- Trigger Out
- Windows 95/98 Software

LA2124-128K (100MSa/s, 24CH) Clips, Wires, Interface Cable, AC Adapter and Software **$800**

www.LinkIns4.com

Link Instruments • 369 Passaic Ave • Suite 100 • Fairfield, NJ 07004 • (973) 808-8990 • Fax (973) 808-8786

Seetron Serial LCDs

...almost too easy!

LCDs have a well-deserved reputation for being a pain in the neck to interface. Their non-standard parallel busses and weird timing are enough to turn your hair gray—if you don't pull it out first. Our LCDs have an easy-to-program serial interface (up to 9600 bps) that makes them a snap to use, particularly with friendly micros like BASIC Stamps®.

• *Easy hookup and mounting.* We don't stop at simplifying display interfacing and programming. For easy, quick connections, we offer a variety of ready-made wiring harnesses. Need mounting hardware? We've got that, too.

• *In stock right now.* Hate to wait? Our products are in stock and ready to ship the day you place your order.

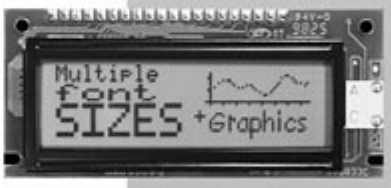
• *Online satisfaction.* Our customers love our web site—it's fast, uncluttered, and jammed with tech data on our products. The online store runs weekly sales with savings up to 40%. Checkout is a breeze with a simple one-page form.



BPI-216N
2x16 serial LCD
ultra-low power
45.00



ILM-216L
2x16 serial LCD
w/ switch inputs,
config EEPROM
49.00



SGX-120L
120x32-pixel
serial graphics LCD
99.00



TRM-425L
4x20 serial LCD
w/ keypad input,
95-screen memory
119.00

www.seetron.com

Many more models available at seetron.com

Scott Edwards Electronics, Inc.
1939 S. Frontage Rd. #F, Sierra Vista, AZ 85635
phone 520-459-4802 • fax 520-459-0623
web www.seetron.com • email sales@seetron.com

TASK MANAGER

The More Things Change…

In the past five to 10 years, we've witnessed momentous events. We've just seen the end of the twentieth century and crossed the threshold into the new millennium. And in the world of engineering, we've seen remarkable achievement, leading to an age of digital everything. However, for those of you for whom change sends a chill down your spine, rest assured that some things remain the same. Regardless of how far we've come in the digital revolution, you still find analog parts on the shelves.

Yes, traditional methods still apply in today's world. For instance, speakers are still analog devices. For insight into the world of audio systems, look no further than page 30 for Ed Nisley's article. You probably think building a trouble-free audio system is a tremendous ordeal. Everything has to be high quality, right? There are the PC, software, as well as digital data and high-resolution converters to worry about. Well, maybe not. You can't learn everything from the datasheets, says Ed.

Longtime writer George Novacek also has an interesting topic to talk about. Following up his coverage of EMC, ESD, and transient protection, he now focuses on grounding and bonding (page 12). George provides the basics of this expansive issue. Armed with some no-nonsense solutions, you'll be ready to experiment on your own.

In your experiments, do you often find yourself battling the effects of nonlinearity in analog devices? You're not alone. Tom Napier has spent a good deal of time dedicated to combating this obstacle. As with most things in life, a solid education in the principals you're dealing with is the key to success. Interestingly, the solutions evolve out of not just a solid understanding of analog techniques, but also of the digital domain. When you know which components work best as analog or digital, you're a step closer to fixing the flaws.

This month, you also get a special treat from Jeff Bachiochi. Following the Design Logic 2001 contest sponsored by Atmel, Jeff came out from behind the bench to take a look at the projects entered. Impressed by the professional-quality work accomplished by a group of Cornell University seniors, he decided to find out more. "RISCy Business" (page 44) is the result of interviews with the students and their professor who encouraged them to enter. If you're interested in finding out what the future holds for engineering, you'll want to take a look at what college kids are building these days.

What will happen in the next five to 10 years? During the early twentieth century, consequential inventions including the airplane, light bulb, and analog cellular phone were created. In 1945, John Mauchly and John Eckert along with their colleagues introduced the Electronic Numerical Integrator and Computer (ENIAC), which played a pivotal role in the progression to modern electronic computers. With the proven capabilities of analog techniques coupled with the growth of the digital age, I wonder what inventors will discover in the early twenty-first century.



jennifer.huber@circuitcellar.com

CIRCUIT CELLAR®

THE MAGAZINE FOR COMPUTER APPLICATIONS

EDITORIAL DIRECTOR/PUBLISHER
Steve Ciarcia

WEB GROUP PUBLISHER
Jack Shandle

MANAGING EDITOR
Jennifer Huber

TECHNICAL EDITOR
C.J. Abate

WEST COAST EDITOR
Tom Cantrell

CONTRIBUTING EDITORS
Ingo Cyliax
Fred Eady
George Martin
George Novacek

NEW PRODUCTS EDITOR
John Gorsky

PROJECT EDITORS
Steve Bedford
David Tweed

CHIEF FINANCIAL OFFICER
Jeannette Ciarcia

CUSTOMER SERVICE
Elaine Johnston

ART DIRECTOR
KC Prescott

GRAPHIC DESIGNER
Mary Turek

STAFF ENGINEERS
Jeff Bachiochi
John Gorsky

QUIZ COORDINATOR
David Tweed

EDITORIAL ADVISORY BOARD
Ingo Cyliax
Norman Jackson
David Prutchi

Cover photograph Ron Meadows—Meadows Marketing
PRINTED IN THE UNITED STATES

ADVERTISING

ADVERTISING SALES MANAGER
Sean Donnelly — Fax: (860) 871-0411
(860) 872-3064 — E-mail: sean@circuitcellar.com
Cell phone: (860) 930-4326

ADVERTISING COORDINATOR
Valerie Luster — Fax: (860) 871-0411
(860) 875-2199 — E-mail: val.luster@circuitcellar.com

ADVERTISING CLERK
Deborah Lavoie — Fax: (860) 871-0411
(860) 875-2199 — E-mail: debbie.lavoie@circuitcellar.com

CONTACTING CIRCUIT CELLAR

SUBSCRIPTIONS:
INFORMATION: www.circuitcellar.com or subscribe@circuitcellar.com
To Subscribe: (800) 269-6301, www.circuitcellar.com/subscribe.htm, or subscribe@circuitcellar.com
PROBLEMS: subscribe@circuitcellar.com

GENERAL INFORMATION:
TELEPHONE: (860) 875-2199 Fax: (860) 871-0411
INTERNET: info@circuitcellar.com, editor@circuitcellar.com, or www.circuitcellar.com
EDITORIAL OFFICES: Editor, Circuit Cellar, 4 Park St., Vernon, CT 06066
NEW PRODUCTS: New Products, Circuit Cellar, 4 Park St., Vernon, CT 06066
newproducts@circuitcellar.com

AUTHOR CONTACT:
E-MAIL: Author addresses (when available) included at the end of each article.

For information on authorized reprints of articles, contact Jeannette Ciarcia (860) 875-2199 or e-mail jciarcia@circuitcellar.com.

CIRCUIT CELLAR®, THE MAGAZINE FOR COMPUTER APPLICATIONS (ISSN 1528-0608) and Circuit Cellar Online are published monthly by Circuit Cellar Incorporated, 4 Park Street, Suite 20, Vernon, CT 06066 (860) 875-2751. Periodical rates paid at Vernon, CT and additional offices. **One-year (12 issues) subscription rate USA and possessions $21.95, Canada/Mexico $31.95, all other countries $49.95. Two-year (24 issues) subscription rate USA and possessions $39.95, Canada/Mexico $55, 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 © 2001 by Circuit Cellar Incorporated. All rights reserved. Circuit Cellar and Circuit Cellar INK are registered trademarks of Circuit Cellar Inc. Reproduction of this publication in whole or in part without written consent from Circuit Cellar Inc. is prohibited.

check out what's new at www.saelig.com!

DrDAQ

Educational Datalogger

with built-in sensors!

www.drdaq.com



DrDAQ plugs into a PC for useful datalogging at school, college, industry. Built-in sensors for light, sound, temp. or add pH sensor and run one of the many suggested science experiments! ***- only $99!***

PC-based Instruments

ADC-10 8-bit **$85** through **ADC-216** 16-bit **$799**—display scope, spectrum and meter **simultaneously. Connect to PC parallel port and start gathering/displaying data immediately!**



- **EnviroMon** temperature logging/alarm system standalone or with PC.
- **TH-03** thermistor-to-PC converter
- **TC-08** 8x thermocouples

NEW!!

12-bit 100Ms/S dual-ch scope adapter

ADC-212/100

$1015!!

RS232 to RS422/485 self-pwrd converters



- Plug directly into PC — **self powered!**
- Drive any RS422 or RS485 devices.
- Send control and data 100s of feet!

K422/K485, 25pin > 9pin . . . $69

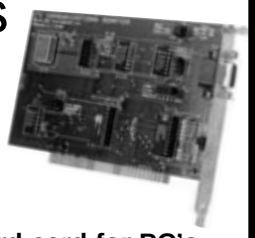
K2 9pin > 9pin $69

Isolate RS232/422/485 signals

K4xx-ISOL 25pin self-powered $139

NEW! 9p-9p K3-ISOL - $129!

Make PCs talk I²C easily!



Industry-standard card for PC's
ISA/P-port/PCI versions
2-6V I2C bus versions

- Master, Slave or Bus monitor
- Control or program I²C devices

ICA90/93LV - PICA90/93LV PCI90/93LV - from $299!

"How to I²C" www.saelig.com

Remote control & data acquisition

BIT*link*®

power & data on two-wire control network



NEW! Euroquartz filters/crystals!

2-year self-contained DATALOGGERS for volts, switch-closures, events, flow, pressure, etc. See www.abidata.be for details

CANbus Starter Packs

PCI/ISA/PCMCIA/PC104/ VME/cPCI format boards. Software drivers for most OS's. CAN/Ethernet bridges, etc.

by Janz for all computers

SMD PCB adapters for prototyping

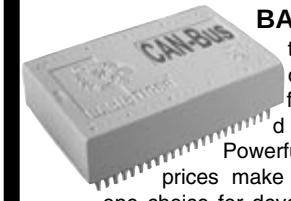
USB ic's

USB <> RS232 easily!!

US232: USB <> RS232 cable $35

WILKE TIGER MODULES

multitasking powerful BASIC building blocks



BASIC Tigers are tiny multitasking computer systems for quick project development. Powerful features and low prices make Tigers a number one choice for developers: super-fast development cycle, high reliability products, >100,000 instructions/s, up to 38 I/O lines, A/D, D/A, I2C, SPI, , text/graphic LCD interface, up to 50,000 lines of BASIC, RTC/watch-dog timer. Easy connection and software for **Expansion Modules** for CANbus, TCP/IP for net-access, opto-I/O, 64 analog inputs, 64 digital outputs, high-current outputs, etc.

TIGER Starter Kits start from $159!

NOW! GPS Logging

PCMCIA Datalogger TDS2020 lowpower PCcard logging



- Store analog/digital /GPS or CANbus data on FlashATA cards - read on your own PC!
- > 100 customizable software modules—finish REALLY quickly.
- 8ch 10bit A/D, 33 I/Os, I²C, 2 x RS232, interrupts, sleepmode, pre-emptive multitasking, easy to attach LCD or keypad.
- CANbus adapter—recompile or log data over huge network!

Saelig Company *brings you unique, easy-to-use control and instrumentation products to USA, mostly from Europe, but now from worldwide. (Need USA sales help - overseas companies?)*

Our customers comment on our unrivalled FREE after-sales support.



"Hi - I'm Alan - you can email me at saelig@aol.com for free advice for your control or measurement problem."

Saelig Co. Inc.

585-425-3753 • Fax: -3835

www.saelig.com • saelig@aol.com

Customer list inc: Intel, Compaq, Philips, NEC, Kodak, Nokia, US Military, Microsoft, Dell, Xerox, Universities, T.I., Dalsa, Harris, Litton, Sony, J&J, Thomson, H-P, Agilent, etc.

ISSUE 145

FEATURES

12 **Are You Grounded?**
George Novacek

20 **Embedded Smarts Fix Analog Flaws**
Tom Napier

26 **Driving the NKK Smartswitch**
Part 2: Graphics and Text
Aubrey Kagan

36 **Digital Ignition System—**Building Without a Distributor
Frantisek Bachleda

44 **RISCy Business**
Part 1: RISC Projects by Cornell Students
Jeff Bachiochi

52 **An Open-Source HCS Project**
Neil Cherry

62 **The AT89C51/52 Flash Memory Programmers**
Noel Rios

COLUMNS

30 **ABOVE THE GROUND PLANE**
PC Audio Bits
Ed Nisley

56 **ROBOTICS CORNER**
The OO CarolBot
Chris Cantrell

70 **APPLIED PCs**
Building a Modular Programming Platform
Part 2: Building the PCB
Fred Eady

76 **FROM THE BENCH**
SmartMedia File Storage
Part 3: Reading a File
Jeff Bachiochi

80 **SILICON UPDATE**
FPGA News Flash
Tom Cantrell

4 **Task Manager**
Jennifer Huber
The More Things Change...

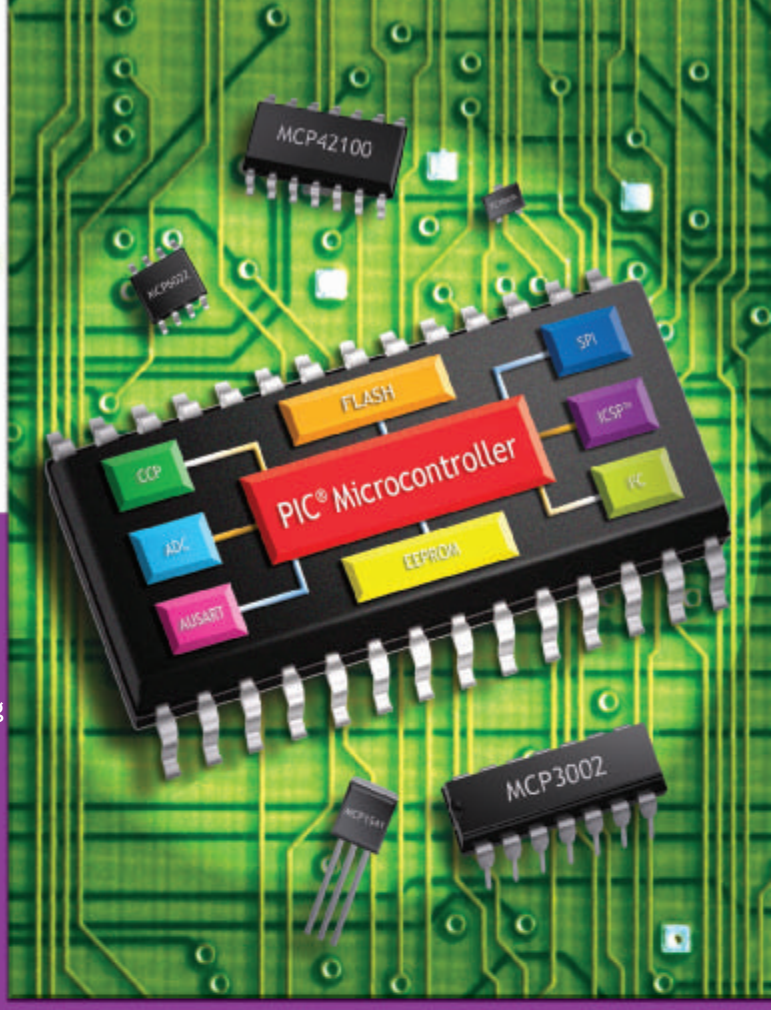
8 **New Product News**
edited by *John Gorsky*

11 **Test Your EQ**

94 **Advertiser's Index**
September Preview

96 **Priority Interrupt**
Steve Ciarcia
Upgrade or Die

Mad Dash for Flash Cash



Does your flash design idea have what it takes to win? Use Microchip's flexible PIC® FLASH microcontrollers to meet all your design needs:

- Programming Flexibility
- Re-programmability
- Remote Self Programming
- In-Circuit Serial Programming™ (ICSP™)
- Socket & Software compatibility for migration
- Flexible peripherals both on and off chip

Microchip 2002 Design Contest

Be one of nine winning entries and you'll be eligible for great prizes like MPLAB® certified development tools and FREE registration to Microchip's Annual Summer Technical Exchange Review (MASTERs) in 2003. At MASTERs, you'll learn how to solve your greatest embedded control challenges with in-depth training on Microchip products. This rewarding 4-day conference will give you the opportunity to dialog between your peers, consultants/3rd parties, distributor FAEs, design houses and Microchip engineers. So, make a mad dash to the web site below to request your contest registration materials today and compete for flash cash. Hurry! The deadline for submissions is December 16, 2002.

Grand	MPLAB ICE2000 with Processor Module and Free Registration to MASTERs*
1st Place	MPLAB C compiler and Free Registration to MASTERs*
2nd Place	PICSTART® Plus and Free Registration to MASTERs*
3rd Place	Free Registration to MASTERs*
5 Honorable Mentions	MPLAB In-Circuit Debugger 2 (ICD2)

MPLAB CERTIFIED

*Please visit the website for alternative cash prizes

www.circuitcellar.com/flash2002

CIRCUIT CELLAR®
THE MAGAZINE FOR COMPUTER APPLICATIONS

MICROCHIP

The Microchip name and logo, PIC, MPLAB and PICSTART are registered logos and In-Circuit Serial Programming and ICSP are trademarks of Microchip Technology Inc. in the USA and other countries. Circuit Cellar is a registered trademark of Circuit Cellar Inc. ©2002 Microchip Technology Inc. All rights reserved.

NEW PRODUCT NEWS

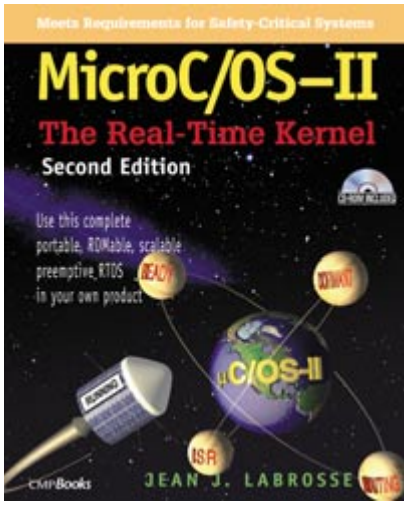
Edited by John Gorsky

HIGHLY PORTABLE RTOS

The **MicroC/OS-II** is a highly portable, scalable, preemptive real-time, multitasking kernel (RTOS) for microprocessors and microcontrollers. Its performance is comparable (in some cases exceeding) to that of many commercially available kernels.

Written in ANSI C for maximum portability, the MicroC/OS-II has been ported to more than 40 different processor architectures ranging from 8- to 64-bit CPUs. Certifiable for use in safety-critical systems, this RTOS has proven to be robust, reliable, and safe enough to use in your own applications.

The second edition of Jean J. Labrosse's book *MicroC/OS-II, The Real-Time Kernel* has been released. This book describes the design and implementation of MicroC/OS-II and teaches you the fundamentals of a multitasking real-time system. You'll learn about scheduling, context switching, task and time management, semaphores, mutual exclusion semaphores (mutexes), event flags, message passing, and dynamic memory allocation. The description of the code has been rewritten to make it easier to follow.



The companion CD-ROM contains all of the source code for MicroC/OS-II and ports for the Intel 80x86 processor running in Real mode and for the large model. Within just a few minutes, you'll be able to test some of the features of the MicroC/OS-II on your Windows-based PC.

Micrium, Inc.
(954) 217-2036
www.uCOS-II.com

ICEPIC DAUGHTER BOARD

The **DBF77** is a plug-in daughter board for the popular ICEPIC and ICEPIC2 in-circuit emulators. The board reduces the time required to develop and test systems based on the PIC16F66/67/76//77 microcontrollers.



The daughter board allows real-time emulation at speeds of up to 20 MHz. You can set an unlimited number of hardware trigger break points. Source-level debugging in either assembler or C language is possible. By executing code in both Single Step and Procedure Step modes, difficult software bugs can be quickly identified and corrected.

The ICEPIC system is connected to a host PC via an 115-kbps RS-232 interface. The software is able to run under Windows 2000/98/95/NT. The configuration of key parameters and the recording and analysis of key test results is possible through an intuitive user interface. The user interface is also used to display and modify the program and data registers. The ICEPIC is also fully compatible with Microchip's software support tools and will operate within the MPLAB IDE.

The DBF77 daughter board costs **$305**.

OKW Electronics
(412) 220-9244
www.okwelectronics.com

NEW PRODUCT NEWS

56K EMBEDDED SOCKET MODEM

With low operating power requirements and small form factor, the **56K90 Socket Modem** is ideal for many applications such as embedded control systems, remote diagnostics, data collection, remote maintenance and web-enabled devices. The 56K90 Socket Modem provides a complete solution for an OEM to integrate into your equipment. It features a solid-state DAA designed to support international operation with compliance to all applicable telephony standards.



The 56K90 is pin-compatible and interchangeable with Conexant's Socket Modem, allowing it to be interchanged in existing designs where power, cost, and availability are areas of concern. The serial socket modem is designed to operate using V.90 and V.34 modulation protocols with fall back to lower speed V.series and Bell protocols. Error correction is provided with support for LAP-M/MNP 2-4. Data compression is supported by V.42bis/MNP-5.

The 56K90 Socket Modem features include a small footprint (1″ × 2.5″), choice of TTL or RS-232, 3.3 or 5 V, fax rates up to 14.4 Kbps, hardware or software handshaking, and caller ID. It uses the AT command set Application Protocol Interface. An evaluation board is also available for use with a TTL or RS-232 modem

Pricing for the 56K90 is **$65** each in quantities of 100 and **$49** each in quantities of 10,000.

Micromint, Inc.
800-635-3355
www.micromint.com

Fastest Path from Prototype to Production



"Easy GUI saved me months of hardware design and programming effort..." – *Mark Doble, President, MD Designs, Tampa, FL*

"Easy GUI may be the answer to your prayers..." — *Tom Cantrell, Circuit Cellar Magazine*

Find out why people are talking about our technology... Order a Starter Kit Today!

©2002 Easy GUI is a Trademark of Amulet Technologies. U.S. and Foreign Patents Pending.

Three Graphical User Interface (GUI) hardware solutions available, LCD Modules, Controller Boards, and GUI Chips

Regardless of which hardware solution you choose, here's a partial list of what our technology can do for your product:

- ***Considerably shorter GUI development time***
- ***Reduced microprocessor load***
- ***FAST user-interface execution and navigation***
- ***FAST and easy updates and changes***
- ***Field updateable GUI***
- ***Animated graphics a piece-of-cake!***

Now Available... 3.8″ and 5.7″ backlit modules with touchscreens!

Amulet Technologies
GUI Engines For Embedded Systems
Visit us at www.GUIchip.com or call (408) 244-0363 today!

Now Distributed by MOUSER ELECTRONICS

NEW PRODUCT NEWS

DIGITAL SOUND SYSTEM

The **QuikWave EM3018B** is a digital sound system that uses CompactFlash memory for sound storage. Housed in a rugged metal case measuring 12 × 15 × 4 cm, the EM3018B is a stand-alone sound repeater capable of reproducing more than 100 different stero/monaural CD-quality sounds.

The EM3018B is directly compatible with industry standard, uncompressed Windows .wav files, and offers superior sound qualtiy over systems that use compression schemes such as MP3. System programming is as easy as copying sound files onto a CompactFlash card via a reader attached to a PC. The card is then inserted into the EM3018B.



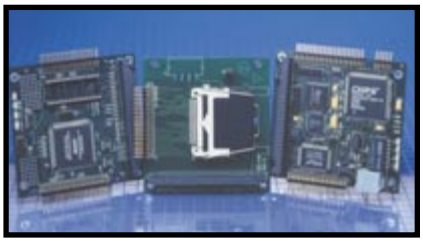
Sound playback can be activated by push buttons, motion sensors, and PLCs. Standard interfaces include direct , binary, and sequential. Keypad, RS-232, and other custom interfaces are available by special request. Playback modes such as Normal, Interrupt, and Hold are also supported. The built-in power amplifier (40-W per channel) eliminates the need for an external amplifier in most cases. Should further amplification be necessary, a line-level output is also available. The system can be powered from a 10-to 32-VDC supply.

The EM3018B is priced at **$289**.

Eletech Electronics
(626) 333-6394
www.eletech.com

TRIO OF PC/104 I/O CARDS

The **MPC920**, **MPC425**, and **MPC205** are three new PC/104 I/O cards that offer a high degree of flexibility to embedded system designers. The MPC920 provides a 256-macrocell, JTAG-programmable CPLD. Full PC/104 interface logic is provided in both VHDL and schematic formats. This board can be programmed for various digital functions such as I/O, timers, PWM, state machines, and shift registers. The MPC920 may be plugged into a PC/104 bus or it can run as a stand-alone board.



The second card in the trio is the MPC425. The MPC425 uses a CompactFlash card to emulate an IDE hard drive. This provides mass storage without the inherent drawbacks of rotating media devices.

The final member of the group is the MPC205. This board contains all of the I/O needed to turn a headless PC/104 single board computer into a full-blown computer system. The MPC205 provides interfaces to a VGA display, floppy drive, IDE drive, PS2/AT keyboard, PS2 mouse, and 10BaseT Ethernet.

Single quantity pricing is **$275** for the MPC920, **$85** for the MPC425, and **$295** for the MPC205.

Micro/sys, Inc.
(818) 244-4600
www.embeddedsys.com

EXTENDED TEMPERATURE RANGE SBC

The **VSBC-8e** is an extended-temperature-range single board computer designed for use in high-end embedded applications. The board is based on a 350-MHz Celeron CPU and operates over the –40°C to 85°C range.

This socket 370 SBC features PC/104-Plus expansion, AGP video with flat panel support, 10/100BaseT Ethernet, sound support, and up to 256-MB SDRAM. The VSBC-8e includes a full complement of on-board industrial features, including an Opto 22 compatible digital I/O port, four serial ports (two RS-232/422/485), and three extra timer/counters.



The board also features a high-reliability design and construction, including a CPU temperature sensor, latching I/O connectors, and a latching high-reliability memory socket. The board is backed by a two-year warranty and is compatible with a wide range of operating systems.

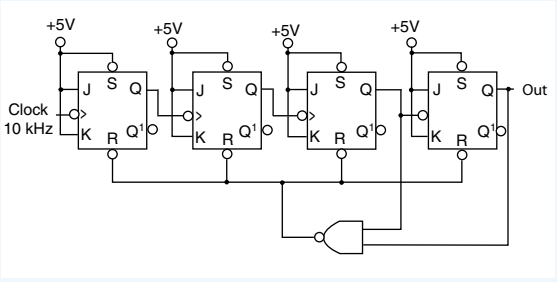
The VSBC-8e is priced at **$878** in low OEM quantities.

VersaLogic Corp.
(541) 485-8575
www.versalogic.com

CIRCUIT CELLAR

Test Your EQ

Problem 1—In the figure below, the J and K inputs of all the flip-flops are held at logic high. What is the frequency of the output signal if the clock is 10 kHz?

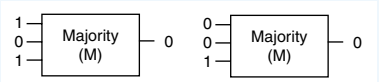


Contributed by David Tweed

Problem 2—What key piece of information is missing from the CP/M file system?

Contributed by Dave Tweed

Problem 3—In nanotechnology, the basic gate is a three-input majority gate whose output is the majority of all the inputs (as shown below). How would you realize AND and OR Boolean functions using this gate?



A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Contributed by Naveen PN

Problem 4—The original single-sided, single-density 8″ floppy disk used by CP/M uses what format. What is its storage capacity?

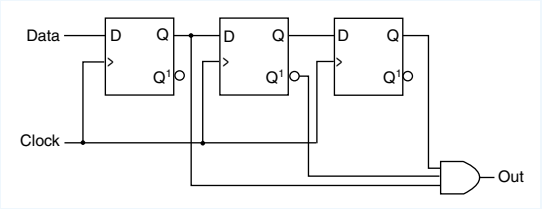
Contributed by Dave Tweed

Problem 5—Would the following code work? If yes, what would be the output?

```
#include <stdio.h>
main() {
     int n=5, f=10;
     printf ("\nn=%*d", f, n);
}
```

Contributed by Naveen PN

Problem 6—What does the circuit below do?



Contributed by Naveen PN

Problem 7—A circular disc can rotate clockwise and back. Use minimum hardware to build a circuit to indicate the direction of rotation.

Contributed by Naveen PN

Problem 8—Suppose you have some 3-bit data, say, grayscale values for which 000 = black and 111 = white. You have a display device that takes 8-bit data, and you want to extend the bit width of your data to match. If you just pad the data with zeros, you get the value 11100000 for white, which is not full white for the 8-bit display. Full white would be 11111111. What can you do?

Contributed by Olli Niemitalo

What's your EQ?—The answers and 4 additional questions and answers are posted at www.circuitcellar.com/eq.htm You may contact the quizmasters at eq@circuitcellar.com

FEATURE ARTICLE

George Novacek

Are You Grounded?

In previous issues, George helped us develop sound EMC, ESD, and transient control designs. This month, he wants to make sure our designs are grounded. In this article, he talks about several pragmatic and nontheoretical grounding and bonding methods.

Good ground is a fundamental requirement not just for electromagnetic interference (EMI), electrostatic discharge (ESD), and transient control, but for the very operation of every electronic circuit, as anyone who has ever smelled molten solder would attest.

Without good grounding and bonding, the best EMC and transient control schemes are destined to fail. In addition, your circuits will be unstable and noisy, and your amplifiers will oscillate or not work at all.

What do we understand by grounding and bonding? One definition states that grounding is the process of connecting circuits' return paths by a conductive medium to a reference point, such as a PCB ground plane or equipment enclosure. In this process, care must be taken not to create ground loops.

Electrical bonding, on the other hand, is the joining of two or more conductive surfaces in order to obtain an electrically conductive, low-resistance, 1- to 3-mΩ path to the reference ground. Because this path serves to divert energy, often with high-frequency content, it is important that the inductive component of its impedance is extremely low as well.

Ground in electronic equipment is not necessarily synonymous with earth, although this is where the expression originated and where, ultimately, many systems are grounded for safety in case of a power fault. But from the functional requirements perspective, ground is also the common reference point in the circuit and often the negative power return. It can be a ground plane, a trace on a printed circuit board, a bus bar, a chassis of an automobile or aircraft, or a metal cabinet rack.

Unlike the system reference, a true system ground must satisfy several criteria. The first and most common (as it is equally needed from your internal ground) is for the ground to provide an equipotential point or plane that serves as a reference potential for a circuit or system. Second, to make immunity to interference signals possible, ground must provide a low-impedance path for interference currents to return to their source. Third, to be effective in suppressing ESD, system ground must present itself as a mass large enough to be capable of supplying or accepting large electrical charges.

It is generally recognized that a single reference ground is the only acceptable scheme for electronic systems and circuits design. This seems like an obvious concept, but its seeming simplicity is deceptive. Grounding is generally not given to straightforward definitions. Consequently, mathematical modeling and analysis are extremely difficult because many uncontrolled or unknown factors affect performance.

Engineering experience plays an important role in achieving an effective design. Unfortunately, this experience is not always fully transferable to other systems. In the end, grounding is one area of electrical engineering design where pure science is powerless without a healthy dose of art, experience, and creative imagination.

I had to accept that writing an all-encompassing, universal cookbook on bulletproof grounding methods is impossible. So, the next best thing is to review the fundamental grounding concepts and discuss the reasons behind them. While doing so, I will

touch on some practical solutions to the problem. The rest (i.e., getting hands-on experience) will be up to you.

SERIAL AND PARALLEL

There are two basic approaches to grounding: serial and parallel, or star grounding (see Figure 1). You should prefer parallel grounding for every application, because it is much more predictable. However, you need to come to terms with the realities of life and recognize that on the circuit design level, you will almost always end up with a combination of the two. Pure serial or parallel ground is usually achievable only at higher, well-defined modular or system architecture levels.

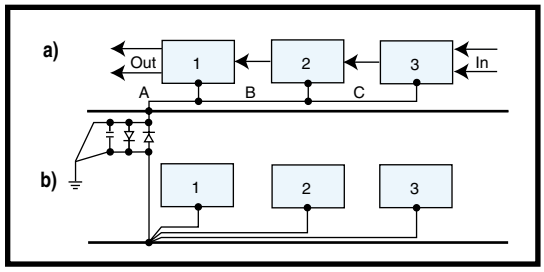


Figure 1—*It's important that all grounds lead to a single point. This can be done in serial (a) or parallel (b) fashion. Each approach has its advantages and disadvantages, but often our choice is limited by reality. On PCBs, the grounds should be joined in a single point with a capacitor and two opposite parallel diodes.*

Because the majority of electronic circuits today are constructed on printed circuit boards (PCBs), you have to assume that the most prevalent grounding scheme will be serial through the PCB ground trace or plane. It could be argued that because of its low impedance and subsequent voltage drop to significantly lower than when only a ground trace is used, the PCB ground plane is as good as a single point ground. This would provide a parallel grounding scheme model with all of its advantages because of the absence of common return impedance. You might accept such an argument for some layouts under certain conditions; however, you must be aware of the existence of some common return impedance and deal with the potential traps it represents.

Looking at Figure 1a, you can see that functional blocks 1, 2, and 3 share the return path segment A, which would ordinarily serve as a power return. In the real world, segment A would have a nonzero impedance. If you suppose that block 1 drives a relatively heavy load (e.g., a speaker), then current flowing through this load would return through segment A.

Because of segment A's nonzero impedance, an AC voltage proportional to the load current and impedance of the segment would develop. In a properly designed system (see Figure 1b), the load return current does not share a path with the source return current in block 3. Consequently, the AC voltage seen by block 3 (because of the common impedance coupling through segment A) could be effectively filtered out by the power supply bypass capacitors. Should the blocks be reversed, with the load connected to block 3, the common impedance voltage would add to the sensor return in a differential mode and be next to impossible to eliminate. Depending on the phase, it could cause oscillations, distortion, or gain change.

Common engineering wisdom tells you that power to a circuit should enter at the output stage. This forces the output signal current to flow directly into the low-impedance power supply. If the output current is allowed to flow through impedance associated with the input, as would be the case with segment C in Figure 1a, then the result would be current feedback. Its effect could be the coupling of output noise into the input, or it could be a source of system instability and gain error.

In other words, it could introduce negative feedback with a beneficial effect on the system. But, I would much rather introduce feedback through a properly designed, well-defined interface than through an obvious design flaw. [1] This grounding approach may appear controversial because other sources suggest the exact opposite; namely, that the most sensitive circuit is closest to the power supply. [2]

Figure 2a explains the coupling phenomenon in terms of impedances and signals. In Figure 2b you can see the chassis ground tied to the input stage. Oren Hartal argues that the power supply has a low impedance, so the voltage developed across its terminals because of the noise generated by the output stage is below the sensitivity of the input. The unit farthest away from the power supply is the noisiest. This one example illustrates why a universally valid cookbook on grounding cannot be written and why experience plays such a major role in design projects.

I can imagine situations where one or the other grounding topology would possibly work better. In the end, you need to understand the source of interference, power requirements, and the types of loads. For the majority of applications, however, Figure 1a should be the safest bet.

PCB LAYOUT

Obviously, parallel grounding solves the problem of shared current returns, but it's difficult to achieve on a PCB. On a purely logic, relatively slow board with a 5-V power supply and CMOS circuits (with their inherently high noise immunity), the PCB layout is fairly forgiving and a ground plane could be considered to provide parallel grounding ability (see Figure 1b). But this situation changes drastically as soon as you start using high-speed, low-voltage logic or analog circuits.

The best approach to eliminate noise coupling is to provide separate grounds for digital and analog circuits and to treat the analog ground scheme as serial. That means placing the output circuits closest to the power supply entry and the most sensitive circuits farthest from the power, which should also be the common grounding point. The analog and the digital grounds should be joined together in a single point (see Figure 1).

Although, strictly speaking, this is not a grounding issue, I could not resist adding the two diodes and the capacitor where the grounds are joined in Figure 1. This is a good practice that serves not only to limit any potential differences that may exist between the separate grounds, but also to make sure that in case the ground strap between the two planes fails, the potential difference between the two grounds does not exceed one diode

drop and cause permanent damage to the circuits. The diodes should be fast-switching and the capacitor should be around 0.01 µF.

There are quite a few PCB layout rules, but I won't dwell on this subject. PCB layout is an art in its own right; it cannot be learned just from reading books. Good PCB designers are worth their weight in gold, but this is not always recognized. If you were to believe all of the commercials, then you would think autoplacement and autorouting tools are so good today that anyone taking a short course could become a PCB layout expert. To be sure, those tools are efficient in the hands of an expert, but on their own they possess just a little more intelligence than a hammer.

PCB layout is like playing a musical instrument. Mastering the sophisticated layout software is only the beginning. After that you need practice, practice, and more practice. A good layout technician needs to understand the principles behind the circuits he is going to place and route. In addition, you must appreciate their functions and manufacturability, and have imagination, aesthetic feel, patience, god-given talent, and plenty of experience.

Although there are many books on the subject, you should save yourself time, money, and frustration by hiring an outside contractor (unless you plan to design a relatively simple board or want to master the art and become an expert yourself). Present-day PCBs are a far cry from the boards of only a few years ago, when many circuit designers could lay out their own as a kind of relaxation activity.

SYSTEM GROUNDS

Even when it comes to the most complex systems, the grounding principles are exactly the same as those on the circuit level. But once on the modular, subsystem, or black box level, the parallel grounding scheme is the one that's most commonly found.

Figure 3 shows the architecture of such a system. You can imagine the system being an aircraft with its fuselage (i.e., chassis) representing the system ground. Numerous subsystems (i.e., flight controls, cockpit, and propulsion) comprise black boxes for individual functions. The black boxes or panel racks are shown with broken lines. Internally, they often consist of modules and functional blocks, such as plug-in cards, each following the same grounding rules. And here comes the problem.

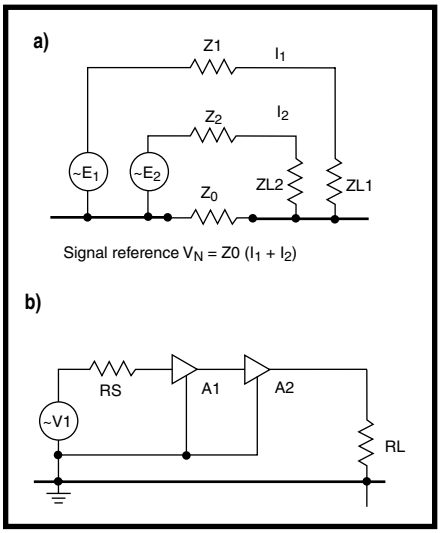


Figure 2—*Sharing ground return impedance Z_0 leads to the coupling of signals E_1 and E_2 through voltage V_N, which could also be noise, developed across the common impedance Z_0 by the combined signal current returns I_1 and I_2. In the series grounding scheme we must decide whether it is the DC power or signal immunity that is the major issue.*

To minimize interference, you must eliminate common return impedances. The parallel grounding scheme does that, but imagine that the individual cabinets (black boxes) are grounded to the same common point by a wire. This will work well at low frequencies, but after the length of the grounding wire has reached the quarter wavelength of an interfering signal, the single ground becomes ineffective and the electronics connected through the wire will be ungrounded because of the 90° phase shift.

At 1 GHz, a quarter wavelength is a mere 75 mm, or about 3". Therefore, you must use a multipoint grounding system, which is also mechanically advantageous because black boxes are tied to the chassis or mounted on a rack.

On aircraft, you will see each black box individually bonded to the chassis ground, as is shown in Figure 3. I will cover how this is done when I explain bonding. But unfortunately, the multiple grounding points will cause another problem, ground loops.

GROUND LOOPS

Figure 4 shows the simplified schematic of one subsystem. The system battery is usually bonded to the chassis ground by a grounding stud, and wire pairs distribute the power in a parallel fashion to each subsystem. Therefore, each subsystem has its own power return.

Essentially, there is no common impedance (except for the grounding stud) shared by the subsystems. Each subsystem is in its own metal cabinet that is bonded to the chassis. All of the interfaces between the inside and outside of the subsystem go through low-pass filters to provide electromagnetic compatibility (EMC). The low-pass filters are LC pi filters.

Now, let's look at Figure 5a, which makes the problem quite clear. Capacitors C1, C2, and C3 are parts of the pi filters and allow the internal signals to couple so that the internal reference ground, typically the negative supply voltage, is on a potential different from the cabinet. The result of this unintentional, parasitic feedback will vary with components' values and circuit types, but you should expect increased noise, instability, outright oscillations, or gain shift.

Even though you might not see the effects because of the low sensitivity of the circuits or signal threshold (i.e., when high noise margin digital circuits are involved), this situation is definitely unhealthy and must be corrected. The risk always exists with component tolerances, aging, or temperature variations that some unpredictable, embarrassing problem could occur later.

Even if you do not have to satisfy strict EMI requirements, removing the pi filters does not solve the problem. C1, C2, and C3 will still exist as parasitic capacitances, although their values will be less. The only viable solution to this problem is to connect the internal reference ground to the cabinet, as shown in Figure 5b, and make V3 = 0. But having solved this problem, you created

$99 SPECIAL OFFER
Shareware Version
Download it Today

PROTEUS VSM

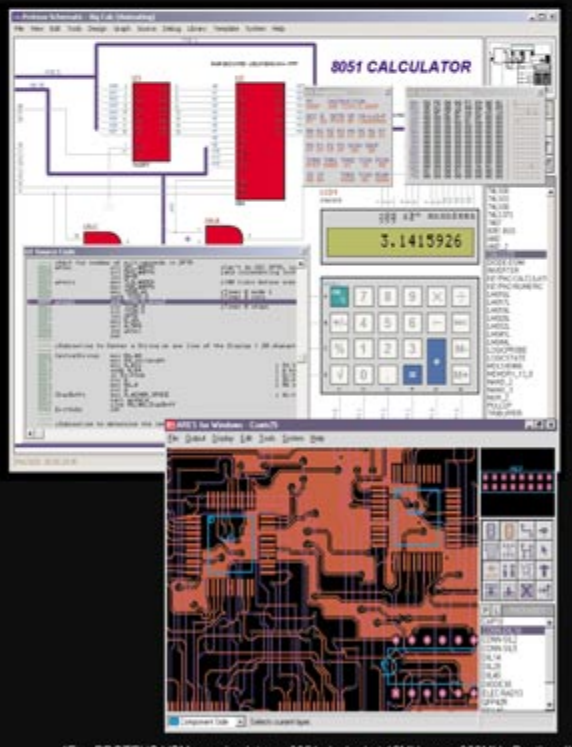
Virtual System Modeling

Build It In Cyberspace

Web: www.labcenter-electronics.com

Develop and test complete micro-controller designs without building a physical prototype. PROTEUS VSM simulates the CPU and any additional electronics used in your designs. And it does so in real time.*

- CPU models for 8051/8052, Mc68hc11, PIC and Atmel AVR series micro-controllers available now. More CPU models under development. See website for latest info.
- Interactive device models include LCD displays, RS232 terminal, universal keypad plus a range of switches, buttons, pots, LEDs, 7 segment displays and much more.
- Extensive debugging facilities including register and memory contents, breakpoints and single step modes.
- Source level debugging supported for selected development tools.
- Integrated 'make' utility - compile and simulate with one keystroke.
- Over 4000 standard SPICE models included. Fully compatible with manufacturers' SPICE models.
- DLL interfaces provided for application specific models.
- Based on SPICE3F5 mixed mode circuit simulator.
- Integrates with PROTEUS PCB design to form a complete electronics design system.
- VSM integrates with LabCenter's Schematic Capture tool ***ISIS***, the Mixed Mode Circuit Simulator ***PROSPICE*** and the PCB design tool ***ARES***.



*E.g. PROTEUS VSM can simulate an 8051 clocked at 12MHz on a 300MHz Pentium II.

LabCenter products distributed in the USA, Canada and Mexico by R4 Systems Inc.

Labcenter Electronics

Write, phone or fax for your free demo CD - or email info@r4systems.com.
Tel: 905.898.0665, Fax: 905.898.0683 Web: www.labcenter-electronics.com.

Real Fast SMALL Java™



32-bit native execution MCU
J2ME CLDC w/options
512 KB to 2 MB Flash
512 KB SRAM
1x2 inch DIP 40
3.3 or 5-14 VDC
www.jstamp.com

JStamp!™

Experience all the benefits of Java technology with *FAST* 32-bit native execution and deterministic real-time capability. JStamp's JVM and RTOS are in silicon. *Native execution* means that *Java is now your "machine language"!*

Find out why native execution Java is the biggest news in embedded systems since C, and how it can slash your product development time and lifecycle cost by 50% or more. Get the facts at www.jstamp.com/reality.htm

JStamps are under $100 (10+). Development kits are about $299. All are available now.

INTER-PRETER

JRealTime™

See the entire JRealTime family at www.jrealtime.com

SYSTRONIX®

Salt Lake City, Utah, USA
Tel:+1-801-534-1017

Java is a TradeMark of Sun Microsystems, Inc. JRealTime, JStamp, SaJe, JStik are TradeMarks of Systronix, Inc

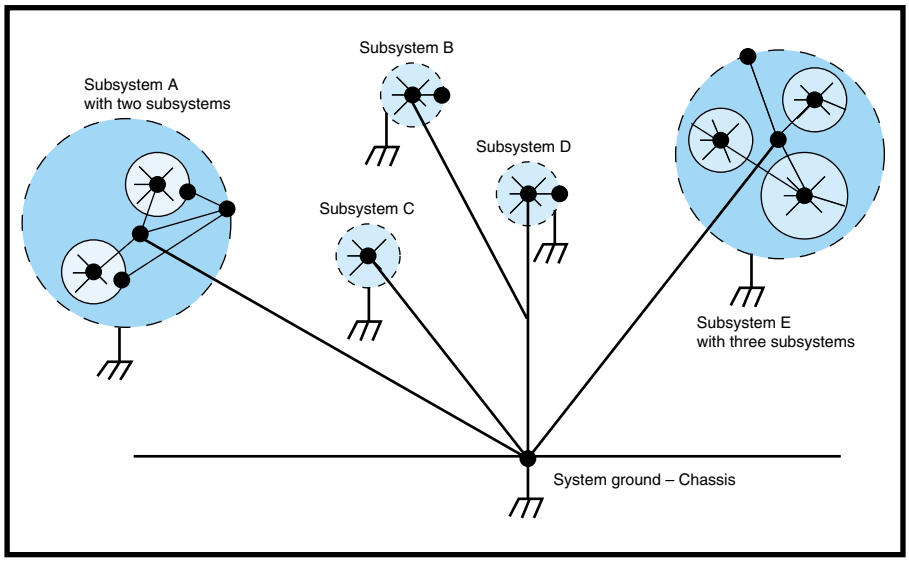


Figure 3—*The ground topology of a complicated system with many subsystems will typically look like this. Both parallel and serial ground connections exist, but they must always lead to a single point.*

two others, which are no less serious: common return impedance and the feared ground loop.

The inset Rx-Cx in Figure 4 stands for the grounding strap that you would add to bring V3 to zero. For the moment, let's assume Rx = 0 Ω. In other words, it's a jumper, a dead short, so the Cx value doesn't matter. But now at least a portion of the power return current will flow through the chassis. The power return lead, usually a 22 AWG wire, probably has higher resistance than the chassis path at DC. At AC, the low-pass filter and wire inductance will cause almost all of the return current to go through the chassis.

With several subsystems like this, there will be shared chassis return impedance to provide coupling for interference and crosstalk between the subsystems. What's more, the power and/or signal lines and their chassis ground return will form large ground loops. Even a partial current return through ground makes the twisting of signal wire pairs ineffective by introducing imbalance into these small loops. The large loops will pick up extraneous magnetic fields or radiate them. At their worst, the ground loops will destroy most of the shielding effectiveness and defeat HIRF as well as transient protection.

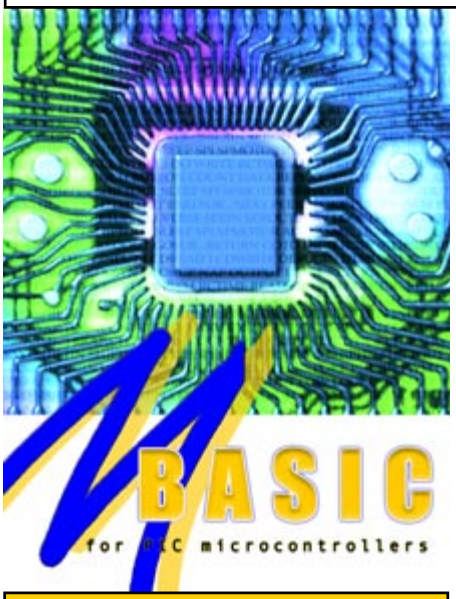
One solution to this dilemma, provided the rest of the design does not need power or signal return through ground, is to use an RC combination instead of a wire jumper (see Figure 4). In avionic and military systems, Rx > 10 kΩ and Cx < 10 µF is an acceptable approach that provides the necessary ground potential equalization while minimizing ground loop current. Unfortunately, the implementation is not always possible, especially when driving external loads with their own return or return through chassis. Many such loads still exist in automobiles. Several examples and their wiring are addressed in my previous articles (*Circuit Cellar* 117 and 118).

The best solution in this case is electrical isolation, which can completely eliminate return current through the chassis. Today, most power supplies are switching regulators, so full input/output isolation is not a problem. There are also transformers and many optocouplers, some of which are designed for linear operation, as well as isolation amplifiers, to make this task simple.

Just remember that DC isolation does not guarantee isolation at high frequencies. Quite often, the simplicity and robustness of an isolated I/O will outweigh its seemingly higher initial cost after you take into account all of the EMC and transient protection requirements of a standard circuit.

As a rule, no signal should be referenced to ground. Each input or output line should have its own return line and, of course, balanced

PROGRAM PIC®'S IN BASIC



Learn to Program PIC Microcontrollers in easy to use BASIC

DOWNLOAD MBASIC LITE FREE !

Try before you buy !

MBASIC SOFTWARE

Whether your just learning or a professional, Programming PICmicro® MCUs has never been easier ! MBasic is much simpler than C or Assembly. MBasic creates a one click solution that allows you to experiment and test code changes on-the-fly! Bring your projects to life quicker and easier with MBasic for PICmicro® MCUs !

ICD - IN CIRCUIT DEBUGGER

Stop wasting time strategically planting debug statements throughout your entire program. MBasic includes a built-in ICD (In Circuit Debugger) FREE. Watch variables, SFRs and RAM values as each line executes. MBasic's ICD is so easy to use, even the first time user can have it up and running in minutes !

MBASIC AND ASSEMBLY LANGUAGE

Seasoned Assembly programmer or just learning ? MBasic allows an easy mix of BASIC and Assembly. Instead of spending weeks writing your program, have it done in days !

SYNTAX

A complete set of easy to use commands ! Serin, Serout, If..Then..Elseif..Else..Endif, Do..While, While..Wend, OWin, OWout, ADin, Pulsin, Pulsout, PWM, Xin, Xout and more! Plus easy acces to all PICmicro built-in hardware peripherals.

MATH

MBasic is the only BASIC compiler that supports 32 bit floating point and integer math. This includes 32 x 32 bit divides and multiplies.

MBASIC PRO ONLY $229.95

Includes Printed Manual and CD-ROM with MBasic.

PICMICRO TOOLS



OPTIONAL ISP-PRO ITEMS AVAILABLE
- Plastic Enclosure
- 40 Pin Zif Adapter
- Universal Adapter





ISP-PRO PROGRAMMER

- Uses PC Serial port (USB w/ adapter)
- Very Simple to use
- Free Software updates
- Complete with Windows IDE!
- Easy in-circuit programming
- Supports PIC, Scenix, I2C and more!
- Firmware upgrades Free!
- RJ-11 / 10 Pin Header
- Includes RJ-11 Cable

ONLY $59.95

SOLDERLESS DEVELOPMENT

- Completely Assembled Board
- In Circuit Programmable (ISP)
- Solderless Bread Board
- Built in RS232 (w/ Max232)
- Built in Power Connector
- Removable Oscillator
- Documentation
- Auto Disconnect for RB3, RB6, & RB7
- Available in several models

STARTING AT $59.95

DEVELOPMENT KIT

Includes:
- MBasic Pro Compiler
- 2840 Development Board
- ISP-PRO Programmer
- PIC16F876-20
- 10Mhz Resonator
- Serial Cable
- Power Supply
- CD-ROM and Manual

STARTING AT $159.95

JOIN OUR ON-LINE PIC FORUMS, INFORMATION AND HELP FREE!!!

BASIC MICRO INC
Microcontrollers Made Easy™

TO ORDER VISIT **WWW.BASICMICRO.COM**
OR CALL 734-425-1744
M-F 10 AM TO 6 PM EST.

MBASIC is a registered trademark of Basic Micro Inc.
PICmicro is a registered trademark of Microchip Inc.

PICMICRO MCU® TOOLS

Tigers: powerful CPU modules from Wilke & JED Microprocessors

The BASIC, Tiny and Economy Tigers are sold worldwide, with W98/NT software and six JED, Australian-made boards.

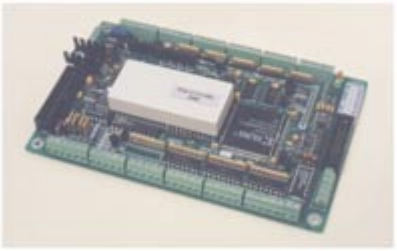


Tigers are modules running true compiled multi-tasking BASIC in a 16/32 bit core, with typically 512K bytes of FLASH (program and data) memory and 32/128/512 K bytes of SRAM. All Tigers use the same low cost W98 or NT compile, debug and download software.

The **Tiny Tiger** has four, 10 bit analog, lots of digital I/O, two UARTs, SPI, I²C, 1-wire, RTC. The module is 60mm by 28mm. **From $US 89.**

The **BASIC TIGER** is a larger unit, with more I/O pins, up to 4Mbyte of FLASH and 2Mbyte of SRAM. **A CAN version is also available.**

The latest is the 28-pin **Economy** or **E-Tiger,** which is only 40mm long, and is an economical way to apply the power of Tigers to cost-sensitive applications. **From $US 49.**



TIG505 Single Board Computer

The TIG505 is one of six Australian Tiger boards, and uses the TCN4/4 Tiger processor with 512K FLASH & SRAM. With 50 I/O lines, two RS232/485 ports, SPI, RTC, LCD, 4x10 ADC, 4 (opt.) DAC, and FLASH mem. Various Xilinx FPGAs add special I/O, eg 3x 32bit quad shaft encoder, X10 or counter/timer functions.

Standard TIG505 with TCN4/4: $US 210

JED Microprocessors Pty Ltd

173 Boronia Rd, Boronia, Victoria, 3155, Australia. (Visa card accepted).
Ph. (+61) 3 9762 3588, Fax (+61) 3 9762 5499
(Freight and local taxes not included in prices)

www.jedmicro.com.au

inputs and outputs should be chosen for their superior performance whenever practical.

MULTIPOINT AND SHIELD

As you now know, larger systems are generally multipoint grounded by virtue of their individual cabinets bonded to the chassis ground. But there are systems where the ground needs to be provided by wire (e.g., a system with a nonmetallic chassis). In those situations, if

$$S < \frac{15}{f}$$

where *S* is the length of the grounding wire in meters and *f* is the highest operational frequency in megahertz, it is safe to use a single point ground. Otherwise, multiple ground, which is achieved by either bonding to a chassis or providing some ground plane in a nonmetallic structure, is required.

Wire grounding could be practical for audio equipment, but if your system is a controller that must be immune to frequencies up to 20 GHz, then the maximum grounding wire length you could use is 0.75 mm, or 0.030″.

Designers often run into conflicting requirements, so experience is important. Suppose you're working with a low-frequency, low-level signal that is carried by a shielded wire, exposed to a high-frequency, high-intensity radiated field (HIRF). The shielded conductor carries a low-frequency signal; therefore, the shield should be grounded at one point only to avoid common impedance ground noise coupling.

The shield, on the other hand, also carries high-frequency current induced by the HIRF, so it should be multiple point grounded. The solution in this case is to bond the shield to the chassis at one end and provide high-frequency grounding at the other end, or even at regular intervals along its length, by small 0.01-µF capacitors.

The common mode voltage induced into the wires within the shield is the result of the current flowing through the shield multiplied by the cable's transfer impedance. Therefore, the principal requirement in the design of the shield grounding is the reduction of the noise current induced into the shield by an external field.

Obviously, if you were to ground the shield at only one end, there would be no current flowing through it and, presumably, no interference reaching the shielded wire. This method is effective at low frequencies, up to about 100 kHz, or when the shield is shorter than 0.05 λ of the common mode interference frequency (for 1 MHz this is about 15 m, or 49′).

With the shield grounded at only one end and at high frequencies, the coupling of the internal wire to the shield occurs through the cable capacitance. The higher the frequency, the more efficient the coupling.

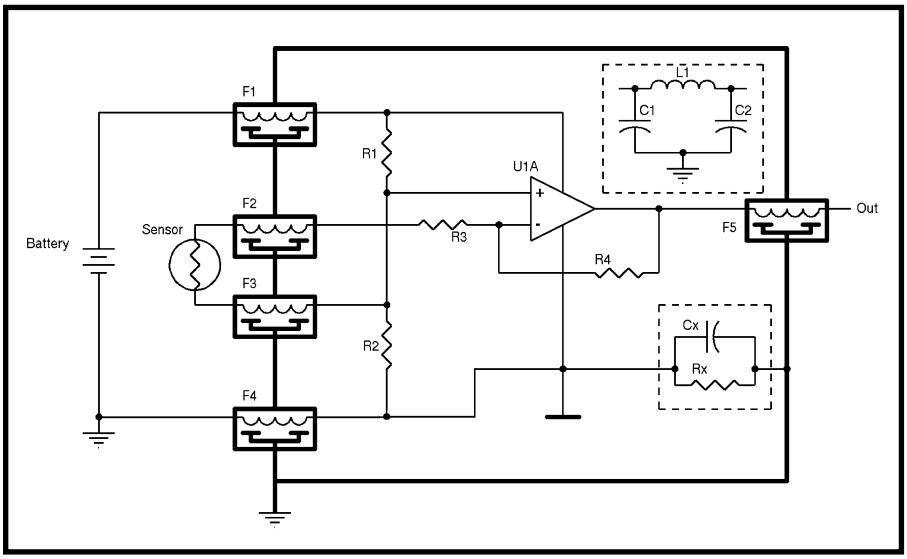


Figure 4—*It's not an easy task to prevent ground loops and also maintain proper grounding. Study this simplified circuit and notice how difficult it can be to satisfy all of the grounding requirements.*

If only one side of the shield is bonded to ground, it should be done wherever the equipment is grounded. For example, with a remote sensor connected to a controller and the controller grounded, that's where the shield should be grounded too. The sensor housing should be connected to only the shield. If the cable shield is used to prevent the radiation of a noisy signal carried by the cable, it's best grounded at the source of the signal. Cables longer than the 0.05 λ, which means essentially all the equipment requiring immunity up to gigahertz frequencies, need to have their shields grounded at both ends or at multiple points.

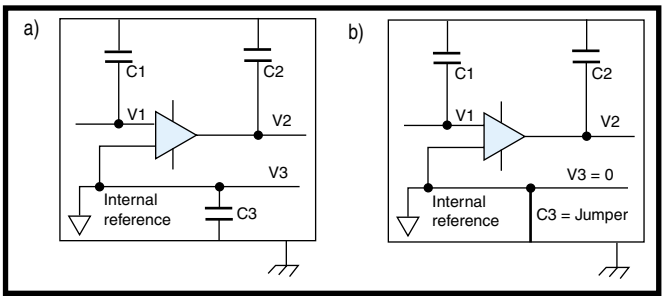


Figure 5—*In (a) you can see the problem capacitances between the signal leads and the grounded cabinet. The only way to eliminate the interference manifested as V_3 is to replace the C_3 with a jumper to the cabinet, but that creates a ground loop.*

BONDING

Bond is the point where the ground is connected to the chassis. Deterioration of EMC performance is usually linked to the deterioration of bonding characteristics because of corrosion, mechanical failure, or other environmental reasons. Bonding is defined by its impedance, which is frequency dependent. Mechanical strength, corrosion resistance, current carrying capacity, and ease of inspection and maintenance also define it.

A stud that is welded, brazed, or otherwise electrically and mechanically attached to the chassis provides a commonly used bond. Soldering is acceptable only if the stud is mechanically secured in some other fashion. Grounding, power, and signal return wires in line with the principles shown in Figure 3 are brought to the stud, connected to it through lugs, and secured by a nut with a star or lock washer. Even though such an arrangement has very low impedance, proper bonding is so critical that some system designers specify the order in which individual returns are placed on the stud.

Equipment in a metal cabinet is usually bonded through its mounting feet. Bare metal on bare metal is the best, so care must be taken not to paint the contact surface. To avoid corrosion, the contact areas must often be plated or chemically treated. Make sure you do not degrade the metal surface bonding characteristics. Bonding through mounting feet is not always practical, particularly when antivibration (AV) mounts are used. Then a threaded stud is welded or brazed to the cabinet and a strap installed between it and a similar stud on the chassis. The impedance must be as low as possible, usually 1 to 3 mΩ is acceptable, so the strap must be short and low inductance.

At high frequencies the strap inductance becomes the dominant factor. This is purely a function of the strap (or wire) dimensions. As a result, when the bonding conductor gets longer, there is little difference between a heavy wire, thin wire, or woven strap.

The conclusion is that bonding must be kept short to be effective, or at high frequencies its impedance will be too high to allow for effective bonding. The rule of thumb is that the strap's length should not exceed four times its width.

Finally, bonding the cable shield to the connector is no less important than the grounding and bonding of the electronic circuits. It used to be a common practice to ground shielding through a pigtail. In such cases, a length of the shield was stripped, the wires pulled out, and the shield twisted tight to make a tail that was then soldered to a ground lug. While simple, this practice is problematic because the pigtail is in effect an inductor connected between the ground and shield. As you now know, the inductance is purely a function of the dimensions, and anything with a length that exceeds four times the width spells trouble.

Properly, shielding must be terminated along its circumference, and then connected to the connector backshell. The cable shielding and the connector become an integral extension of the cabinet. Connector manufacturers sell a multitude of rings and lugs for the proper termination of shielding, so don't degrade the performance of your masterpiece by trying to save a few pennies on shielding termination.

YOU'RE GROUNDED

This concludes my series on EMC, ESD, transient protection, and the grounding and bonding of electronic equipment. There is abundant literature on the theory of these disciplines. I tried to address them on a practical level that was closer to the heart of the designer, who has neither the time nor desire to study the phenomena, but needs to know how to handle them in the real world.

If you want to study these disciplines in more depth, the references in this article are a good place to start. They will lead to more theoretical works on the subject.

George Novacek has 30 years of experience in circuit design and embedded controllers. He is currently the vice president and general manager of Hispano-Suiza Canada, a division of Snecma, a world leader in aerospace engine and landing gear systems. You may reach him at gnovacek@nexicom.net.

REFERENCES

[1] R. Morrison, *Grounding And Shielding Techniques*, 4th ed., John Wiley & Sons, Inc., New York, NY, 1998.

[2] O. Hartal, *Electromagnetic Compatibility By Design*, R&B Enterprises, West Conshohocken, PA, 1996.

RESOURCE

RTCA, Inc., "Environmental Conditions and Test Procedures for Airborne Equipment," DO-160D, www.rtca.org.

FEATURE ARTICLE

Tom Napier

Embedded Smarts Fix Analog Flaws

Broad-minded engineers tend to think outside the box when they're trying to fix flaws in their designs. Tom draws on his understanding of analog and digital design, numerical processing, and embedded firmware to cope with the nonlinearity of analog components.

In two recent projects (*Circuit Cellar* 134 and 138), I took advantage of the nonlinear properties of transistors, but usually nonlinearity is something to be avoided. Outputs are no longer proportional to inputs; inputs interact with each other, and harmonics are generated. While high gain and linear feedback can beat most nonlinearities, this won't work for transducers, which are devices that convert a physical quantity into an electrical signal or vice versa.

Some useful measurement devices are inherently nonlinear. Thermistors are the classic example. Ingenious circuits have been developed to give thermistors a roughly linear temperature-to-voltage characteristic, but there is another approach. In my student days it was not uncommon for instruments to come with a calibration chart. To get a rough measurement you took the meter reading at face value, but for real precision you looked up the chart, found the correction factor to be applied, and added it to find the true value.

These days, you can let an embedded processor do the work. It would not be difficult to make a linear thermometer using a thermistor with a known nonlinear characteristic (e.g., Radio Shack part number 271-110A). It would use an A/D converter, a microcontroller, and an LCD. The thermistor's calibration chart would be programmed into the controller's memory. A look-up and interpolation routine would convert the measured voltage into a temperature reading.

OUTSIDE THE DIGITAL BOX

If my articles have a message, it is that to be an effective engineer you shouldn't specialize in analog, digital, or software techniques. Some of my most profitable designs have been those where I asked, "Which parts of this equipment must be analog, which must be numerical or digital, and where should the dividing line come?"

When I designed a phase detector to lock a 420-MHz clock to a noisy 210-Mbps NRZ datastream, the answer turned out to be an integrating analog filter, a flash ADC, some ECL number crunching, and a DAC. This gave an output current proportional to the clock's phase error. [1] It may sound complicated, but it turned out to be cost-effective.

This approach has an advantage over others because the same device samples the phase and the data. This eliminates the propagation delays that plague standard phase detectors. It has since been successfully applied at lower bit rates. [2]

A good design engineer needs at least an awareness of the analog behavior of digital chips, of how digital functions can be implemented in firmware or, as I am about to explain, how a microcontroller can solve a thorny analog problem.

LET'S MAKE JITTER

A board I developed some years ago had to generate a clock with a frequency that could be set anywhere within a 10-Hz to 50-MHz range. Sinusoidal frequency modulation was then applied. The object was to test data decoding equipment to see how it handled jitter in the input data rate.

What was unusual about this board was that both the amplitude and frequency of the modulation were abnormally high. The clock frequency could

be modulated as much as 20% at rates up to 10% of the center frequency. An extreme, but legitimate, clock output would have been a square wave with a frequency that varied sinusoidally from 4 to 6 MHz and back 500,000 times per second (Let's see the Bessel function gurus plot that spectrum!).

In practice, it would be unusual for both the modulating frequency and amplitude to be at simultaneous extremes. Telemetry engineers use jitter to describe two quite different phenomena. One is the slow variation in the input data rate caused by the velocity of a satellite relative to the ground station. This Doppler shift can displace clock edges that are many thousands of bits away from their expected positions, but it doesn't greatly change the length of any one bit. In audio parlance this would be "wow," a high-amplitude, low-frequency pitch variation.

The other cause of jitter is noise in the signal. This makes the length of a data bit vary about the expected value, but on average data transitions remain near their proper positions. Audio buffs call this flutter, which is a low-amplitude, high-frequency pitch variation. My test equipment had to generate both.

ROCK AROUND THE CLOCK

Because the clock output could be as high as 50 MHz, this was obviously a job for a voltage-controlled oscillator (VCO). You can derive a nice 25- to 50-MHz square wave by running a VCO between 50 and 100 MHz and using a binary divider. Dividing down the VCO frequency generates lower rates. I used a variable ECL prescaler and buried the lower frequency dividers in the FPLA, which also contained a frequency counter.

To allow for modulation beyond either end of a 2:1 range, the basic VCO had to have a 2.5:1 tuning range. Prepackaged VCOs are widely used in frequency synthesizers and as local oscillators in radios and TVs. Most have two features that reflect how they're usually used, inside a phase-locked loop generating a well-defined output frequency. They tend to have a limited tuning range and an audio-frequency modulation bandwidth. I had to build my own VCO using an MC1648 oscillator chip.

The core of a VCO is a tuned circuit that has a fixed inductor and a variable-capacitance (varactor) diode. The higher the reverse bias applied to the diode, the wider its depletion layer gets and the lower its capacitance becomes.

An LC oscillator's frequency has an inherent inverse square root law of frequency against capacitance. The varactor diode's voltage/capacitance characteristic is also nonlinear. The diode capacitance must vary about 7:1 because the minimum tuning capacitance is set by the circuit strays.

Luckily, some varactor diodes (e.g., Motorola MVAM*xxx* series) have been designed with all of this in mind. They can vary over a 9:1 range and their voltage characteristic largely cancels out the oscillator's nonlinearity. This allows you to build a VCO that has a linear voltage/frequency characteristic.

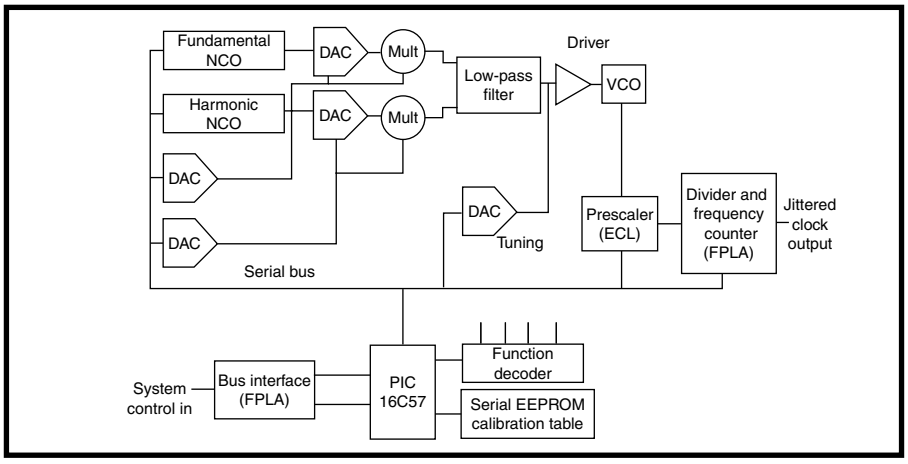


Figure 2—*The PIC-controlled jitter generator uses both analog and digital frequency generators to create the required output.*

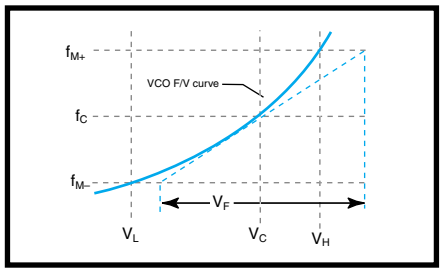


Figure 1—*To modulate the output frequency equally about a center frequency, f_C, you need a nonlinear voltage swinging between V_L and V_H. This is made from a fundamental input, V_F, from which is subtracted a V^2 (second harmonic) term to bring the resulting waveform peaks to V_L and V_H.*

PIC ALL THE WORK

Two problems remained to be solved: how to set a precise center frequency, and how to achieve linear frequency modulation. The former lends itself to a frequency-locked loop, where, in essence, you connect a frequency counter to the output and use the difference between the output and the desired frequency to correct the voltage applied to the VCO. With the proper connections, frequency modulation still can be applied without upsetting the loop.

For this application I used a different approach. The frequency counter is still there, but it is read by a PIC16C57 microcontroller. The PIC also drives a 16-bit serial-input DAC, which supplies the tuning voltage to the VCO. Whenever the equipment is turned on, the PIC tells the DAC to generate a series of evenly spaced voltages. At each voltage it reads the output of the frequency counter. This generates a calibration table with 128 entries that is stored in a serial EEPROM. The whole process takes about 30 seconds.

The PIC then inverts the table, that is, it uses interpolation to generate a second table in which each entry represents the voltage required to generate one of 128 evenly spaced frequencies. Every time you dial up a specific center frequency, the PIC interpolates into the second table and finds what voltage will cause the VCO to run at that frequency.

NOT AN NCO AGAIN!

That was easy, but now what about the modulation? Because it can have any frequency from 10 Hz to 5 MHz, an NCO was the obvious way to go.

ALL ELECTRONICS
CORPORATION
SHOP OUR ONLINE STORE
www.allelectronics.com
Jumbo Pushbuttons
Durable plastic S.P.D.T. momentary pushbutton switch frequently found in arcade games. Useful in any application requiring a large, rugged pushbutton switch. Button operates a 10 Amp snap-action switch. Plunger is 0.86" diameter. Bezel is 1.33" diameter. Threaded bushing is 1.1" diameter and will mount in panels up to 0.75" thick. Includes large plastic mounting nut. Available in two colors:
RED SWITCH CAT# PB-211 $3.75 each
BLACK SWITCH CAT# PB-216 $3.75 each
3.6 Volt 1200 mA Lithium-Ion Cell
Sony # 18650. Rechargeable lithium-ion cell with solder-tabs. Ideal for high energy battery packs. 0.72" dia. x 2.56" long.
CAT # BTE-1200
50 for $3.00 each
$3.50 each
12 Volt 35 Watt Halogen Lamp
General Electric Q35MR11/ NSP(FTE) MR-11 quartz halogen lamp. 12 Volt, 35 Watt. 1.38Ó diameter Multi-Mirror® lamp. 0.15Ó pin spacing.
CAT # HLP-350
10 for $30.00
$3.50 each
Flashing LED
SPECIAL - RED T-1 3/4 water clear
CAT# LED-84
3 for $1.00 each
100 for 25¢ each
1000 for 18¢ each
ORDER TOLL FREE
1-800-826-5432
CHARGE ORDERS to Visa, Mastercard, American Express or Discover
TERMS: NO MINIMUM ORDER. Shipping and handling for the 48 continental U.S.A. $6.00 per order. All others including AK, HI, PR or Canada must pay full shipping. All orders delivered in CALIFORNIA must include local state sales tax. Quantities Limited. NO COD. Prices subject to change without notice.
CALL, WRITE FAX or E-MAIL for our FREE 96 Page CATALOG
Outside the U.S.A. send $3.00 postage.
MAIL ORDERS TO:
ALL ELECTRONICS CORPORATION
P.O. Box 567
Van Nuys, CA 91408
FAX (818)781-2653
e-mail allcorp@allcorp.com

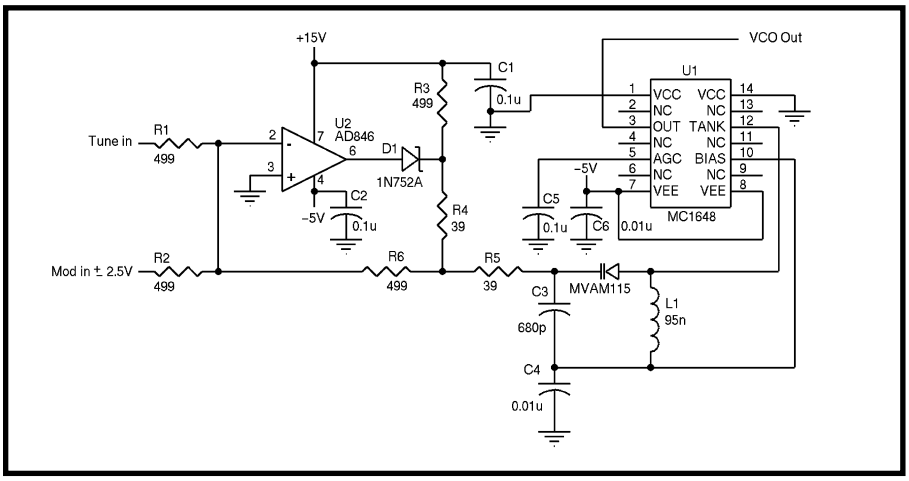


Figure 3—*Wide-band frequency modulation without phase distortion takes a drive amplifier and a custom VCO.*

To vary the modulation amplitude, the NCO drives a DAC, which has a variable reference voltage. The output current of the DAC drives a current multiplier chip controlled by the same variable voltage. This allows the output amplitude to be set over a 100:1 range with a control voltage that varies over a 10:1 range.

Controlling both the reference and multiplier makes for a more accurate and less noisy output than driving either one with a 100:1 voltage ratio. Besides, it let me get away with a 12-bit DAC. Because the PIC is doing the work, the square law relationship between the control signal and the desired output amplitude is no big deal.

PREDISTORTION SAVES THE DAY

So, we've set the center frequency and generated a sinusoidal modulation voltage that has adjustable frequency and amplitude. What's missing?

Well, the VCO characteristic is fairly linear, but it does have some nasty curves, particularly toward the ends of the tuning range. A sine input would not swing the frequency equally in both directions (i.e., the frequency modulation would have second harmonic distortion). It's time for more PIC smarts (and some engineer smarts, but I'm downplaying that part).

The system bus tells the PIC what center frequency and modulation percentage are required. The PIC calculates both extreme frequencies and then looks them up and the center frequency in its calibration table. That tells the PIC what center and peak voltages must be generated. It then calculates a quadratic fit to these three points.

That gives what mix of linear and V-squared signals must be applied to generate clean sinusoidal modulation over the desired range. Figure 1 shows what I mean. As all good engineers know, a V-squared term in a response generates a second harmonic. Thus, you can generate linear frequency modulation of the VCO by adding a controlled level of second harmonic to the drive signal.

This second harmonic comes from a second NCO. It runs synchronously with the first, but is programmed with twice the frequency. Its output goes through a similar variable gain DAC and multiplier chain. The result is summed, via an attenuator, with the fundamental modulating voltage. Flipping the phase of the second NCO gives negative second harmonic terms as needed.

The summed currents go through a low-pass filter to create a smooth signal from the DAC outputs. This filter has a constant delay to over 10 MHz to keep the second harmonic in phase with the fundamental. The result is summed with the main tuning voltage and fed to the varactor drive circuit to give pure sinusoidal frequency modulation. Figure 2 shows an overall picture of the system.

AMPLIFIERS ABHOR CAPACITORS

One problem remained: how to drive the varactor. The driver amplifier applies a DC bias and modulation to the junction of the varactor diode

and fixed capacitor, which, along with an inductor, make up the oscillator's tank circuit (see Figure 3).

The amplifier is looking into the 680-pF capacitor and a variable capacitance diode. It has to supply a DC level in the range of 0 to 12 V to tune the diode to the center frequency and then apply several volts peak-to-peak modulation at up to 10 MHz. And did I mention that the group delay through the amplifier has to be constant to 10 MHz?

Considering that most fast amplifiers oscillate when you connect a 680-pF load, this was a tough job. Part of the answer was to use a current feedback amplifier. I used the AD846.

The real secret was where to connect the feedback. To stabilize an amplifier with a capacitive load, the standard solution is to connect a resistor between the amplifier and load. If you take feedback from the amplifier output, the amplifier will work just fine but the resistor/capacitor combination will play havoc with the high frequency output. Take the feedback from the load and the extra pole in the loop will drive the amplifier into oscillation.

The answer, once more, is predistortion. Use two series resistors and take the feedback from their junction. At high frequencies this looks like a times-two attenuator; the amplifier output peaks at 6 dB but is otherwise stable. Surprise, surprise, peaking is just what is needed to compensate for the roll-off caused by the load and second series resistor. The result is a stable amplifier with a flat gain and constant delay to 8 or 10 MHz. The only downside is that the output swing is limited at high frequencies, because the amplifier peak AC output is twice the load voltage. Because you need a large DC range with a limited AC signal superimposed, no problem arises.

A BUSY PIC

The microcontroller has a lot of work to do. It reads your requirements, which come over the system bus from the main controller board. An FPLA with a serial output handles that interface. The PIC sets up the center frequency, computes both terms of the jittering signal, and sets up the division ratio of the ECL chips, which convert the 50- to 100-MHz output of the VCO into the desired clock rate.

The on-board serial bus uses a 3-bit parallel address to set up the latches that control the board's functions. This bus also drives the DACs, which generate the control voltages. Solid-state switches shift the system from Calibrate to Operate mode. And then, of course, the PIC has the autocalibration to do and the look-up table to prepare every time the power turns on.

BE BROAD MINDED

So there it is, a product that uses all four areas of electronic expertise: analog design, digital design, numerical processing, and embedded control firmware. Part of the logic is embedded in CMOS FPLAs, but the programmable clock divider also uses discrete ECL chips. This implementation needs not only expertise in analog and fast digital circuits, but also the services of an expert in designing with field programmable logic arrays and a good PC layout technician.

It pays to learn something about all aspects of electronics. Not only will your designs be more cost-effective, but also you will be able to understand what the other people on the team are talking about.

Tom Napier has been a space scientist, health physicist, and designer of space data receivers. Now, he provides electronics consulting services and writes about his projects.

REFERENCES

[1] T. Napier, "Recover data from noise with a flash ADC," *EDN*, September 1, 1998.

[2] ———, "Sifting Signals from Noise," *Electronics Now*, April 1999.

SOURCES

PIC16C57 Microcontroller
Microchip Technology
(480) 792-7200
www.microchip.com

MVAMxxx Varactor diodes
Motorola
(847) 576-5000
www.motorola.com









TS-5300

$199 100 pc.
$259 1 pc.

- PC compatible: BIOS, DOS, RTC, I/O & memory map
- 10 base-T Ethernet - TCP/IP stack pre-installed
- IDE Compact Flash socket
- 16 MB SDRAM - upgradable to 32 or 64 MB
- PC/104 bus - 4.0" x 5.4"
- 2 PC serial ports - RS-485 option
- DiskOnChip socket - 32 KB SRAM option
- 2 MB Flash Drive
- User programmable PLD for 40 digital I/O
- ROM-DOS or Linux pre-installed

586 Single Board Computer
Ethernet - Compact Flash - PC/104 - 133 MHz



The TS-200 is a metallic enclosure and PC/104 daughter board that houses any of the TS Embedded PC's. The TS-200 provides two additional COM ports, an IDE interface, a modem option and power input options.

The front panel bi-color LED and user configurable switch allow OEM designers flexibility for special applications.

Custom mechanical and electrical solutions with reasonable design costs can be achieved with orders as low as 100 pcs.

TS-200 Enclosure for SBC

Standard Features	20-30 VDC input 4 RS-232 serial ports 40 pin IDE header Isolated RS485 - COM3
Options	33.6K or 56K Modem - COM4 Isolated DC/DC power Mini hard drive

Enclosure
Power Supply - IDE interface - Modem option



TS-200 with TS-5300 $455 1 pc.
TS-200 with TS-3300 $395 1 pc.

33.6K Modem add $100

www.embeddedx86.com

(480)-837-5200
16610 E. Laser Drive #10
Fountain Hills, AZ, 85268 USA

Technologic SYSTEMS

TS-3100 - $99 100pc
- Low cost DOS machine
- 8 bit PC/104 - 25 MHz
- 512 KB SDRAM
- 512 KB flash drive



TS-3300 - $169 100pc
- Spread Spectrum EMI reduction
- 10 base-T Ethernet - TCP/IP stack
- 33 MHz - 8 MB RAM
- IDE Compact Flash socket
- PC/104 bus - 4.0" x 5.4"
- Analog converter 8 ch. 12-bit
- User programmable PLD for 40 digital I/O





TS-3200 - $119 100pc
- Low cost DOS machine
- 25 MHz - 8 MB SDRAM
- 37 DIO - 4.0" x 4.25"

TS-3400 -$129 100pc
- Spread spectrum clock
- 33 MHz - 8 MB SDRAM
- 37 DIO - 4.0" x 3.8"



TS-3000 series
- PC compatible: BIOS, DOS, RTC, I/O & memory
- 8 MB SDRAM - upgradable to 16 or 32 MB
- 2 PC serial ports - RS-485 option
- DiskOnChip socket - 32 KB SRAM option
- 1 MB Flash Drive
- ROM-DOS or Linux pre-installed

386 Single Board Computer
PC/104

All of our single board computer products are PC compatible based on the Intel 386 or the AMD Elan integrated processors. Using the x86 architecture allows developers to utilize the vast array of operating systems, sample code, development tools and plug-in hardware available. Choosing the PC architecture also allows the development process to take place on a desktop system where many tools and resources are available, before moving the application to the embedded system for final hardware testing and debug.
Technologic Systems also provides design services for special applications, we can combine our PC compatible core with your custom electronics to provide you with an x86 compatible design that has the lower cost and higher reliability of a single board solution.

Technologic SYSTEMS

The TS-9500 video/keyboard adapter also includes an IDE Compact Flash interface. The combination of VGA, PC/AT keyboard and Compact Flash as a hard drive allows compiling/tracing/debugging to take place directly on the target hardware.

TS-9500 $240 1 pc.
- PC compatible VGA adapter
- Standard PC/AT keyboard interface
- IDE Compact Flash socket
- TS-BIOS and rescue tools installed

VGA keyboard adapter
Compact Flash - PC/104



TS-Linux v3.0

Introducing TS-Linux version 3.0 Our latest distribution is built from open-source featuring a 2.4.18 kernel, Apache web server, FTP client/server, bash, etc. With a footprint of 10 MB or less this is a great place to start your application.

- The TS development environment compiles application code using the libraries and headers installed in TS-Linux. Applications developed on your desktop computer copy directly.
- Our User Mode Linux patched kernel runs TS-Linux 3.0 kernel and file system as an application on your desktop computer during development.

PSoC Design Challenge 2002 Cypress MicroSystems Contest

CONTEST ENTRY

FEATURE ARTICLE

Aubrey Kagan

Driving the NKK Smartswitch

Part 2: Graphics and Text

Whether your message is one of workplace safety or world peace, the long nights of brooding over ways to tell the world are over. Now that you're comfortable with the interface to drive the Smartswitch, Aubrey will show you how to display your ideas as text and graphics.

In Part 1, I described the basic interface to drive the Smartswitch. In this article, I will add the bells and whistles to allow both text and messages to be placed anywhere on the screen.

GENERATING AN IMAGE

The software I described last month merely converts the bits stored in the memory array cMatrix[][] to pixels on the LCD. Now, I will show you how to generate an image on the display that enables a level of abstraction.

I've tried to create an interface that enables the mixing and matching of graphics and text. As I mentioned before, the LCD output is maintained in RAM, so writing an object to RAM affects only specific pixels. Everything else is left untouched.

Interface to these functions is through up to 12 bytes of RAM (cCMDs[12]). The application software that uses this interface writes these desired bytes and then waits for a completed signal. Bit 0 of cCMDs[0], when set by the application software, indicates that there is a valid set of command and data bytes in this array.

The Display Interface Handler processes this array and when it's complete it clears bit 0 of cCMDs[0]. The application software polls the RAM location cCMDs[0] and waits for bit 0 to go to zero. After it does, the software updates registers cCMDs[1] to cCMDs[11] as needed and then sets bit 0 of cCMDs[0] to one.

The command set implements all the functions necessary to control the display. The commands include clearing and turning on a pixel block, writing text, placing graphics, reverse video, and changing the backlight. You can download the command set from the *Circuit Cellar* web site.

CHARACTER GENERATION

In order to represent a character in dot matrix format, the symbol must be reduced to a series of pixel patterns. These patterns are held in a look-up table where the character code (normally ASCII) and row number access the specific pattern. This is called a character generator.

Five columns by seven rows is a common format for small displays, but I felt a second, slightly larger format, 7 × 9, also would be beneficial because the Smartswitch is a small device. A 5 × 7 format accommodates a maximum of six characters per line, while a 7 × 9 only allows for four characters per line. I implemented only two character sets in order to economize on program memory space.

Creating a character set can be tedious, so I was delighted when I discovered that I could use Excel to expedite the process. This brilliant idea from Alberto Bitti was published as a design idea in another magazine in

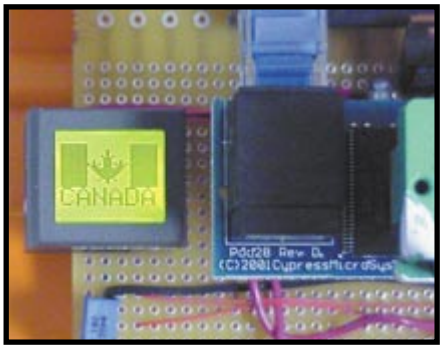


Photo 1—*This flag is actually created in four steps. First, a large block is written across the screen. Second, a gap is opened up in the middle of the block, creating the effect of the sidebars. Third, the maple leaf is inserted into the gap. Finally, the text is written in the bottom line.*

September 2001. [1] I was motivated enough to create some macros to speed up the generation process. This implementation was published in April 2002. [2] You may download the associated Excel file from the *Circuit Cellar* web site.

In summary, you have to enter the desired matrix pattern for each character and the Excel worksheet generates the bit pattern as a number for every row. This is then saved to a file, which can be edited and read into the application (see the cCharSet1 and cCharSet2 arrays in the C module LCDmem.c).

In order to ensure that previous entities are cleared, and to allow spacing between characters, I have added a column of blank pixels to the right of each character and a row of blank pixels underneath each character. If the most significant bit of a character in RAM is set, that character is underlined in the row underneath the character. A standard ASCII character set goes from 0x20 to 0x7F. I have allowed for an additional 32 characters, from 0 to 0x1F, to permit customized characters.

GRAPHIC GENERATION

It's possible to create a graphic image to cover the full 24 × 36 pixels of the display. As with characters, the image is broken into a bit pattern for every row of the graphic image.

Rather than use large blocks of program memory, I restricted the graphics size to a maximum of 16 × 16. This requires 32 bytes per image. I also tacked on 2 bytes to describe the width and height of the image (in pixels), which would work out to 34 bytes. In addition, I limited this to five images because I am not great at graphics (so far I have only managed a fair representation of a maple leaf). The image is included in LCDmem.c as image zero. The generation of images one to four is left to your "image-ination."

The image should be left and top justified within the 16 × 16 matrix so that the first bit of the image is at pictures[x], iBitPattern[0] bit 15. Creation of an image is a lot of work, so once again I resorted to Excel. I used an extended version of the Excel workbook described earlier. You can also download this file from the *Circuit Cellar* web site.

It's possible to expand the size of these images by modifying the program in a few areas. The images are contained in a structure called pictures, defined in LCDmem.c and repeated as a header in LCDmem.h. iBitPattern would have to be redeclared as an unsigned long type to cater to the increased number of horizontal pixels and the number of rows would have to be incremented from 16 in the iBitPattern[16] declaration.

In addition, the procedure LoadGraphic in the Command.c module would need to have the iMask modified to a long unsigned type and its initialization (iMask = 0x8000;) changed to suit the larger image. You can also increase the number of images by changing the array declaration pictures[5] in LCDmem.c and LCDmem.h, and by adding in the values for the graphics in the constant array.

THE USER INTERFACE

Before I bore you to tears with a description of the software operation, I want to demonstrate how the user interface works. You can download the command sequences and photos showing the results of the commands from the *Circuit Cellar* web site.

As you can see, Photo 1 is a mixture of graphics symbols, bit manipulation, and text entry. For all of the command sequences presented, I've ignored stating that the interface must poll cCMDs[0] and wait until bit 0 is zero. I have also omitted stating that setting bit 0 of cCMDs[0] to one should happen after all of the other bytes have been set up in order to initiate the action.

Listing 1—*GenPixel identifies the byte and bit associated with a particular pixel through a composite pointer using cPixRow, cPixColByte, and cPixColMask.*

```
void GenPixel (unsigned int iPixPnt)
{
//Generate a mask and a byte pointer
//First find row. There are 80 bits in a row
      cPixRow=(unsigned char)(iPixPnt/N_COLS);
      cPixColByte=(unsigned char)(iPixPnt%N_COLS);
      cPixColMask=cPixColByte%8;
      cPixColByte=cPixColByte/8;

      switch (cPixColMask)
      {
              case 0:
                     cPixColMask=0x01;
                     break;
              case 1:
                     cPixColMask=0x02;
                     break;
              case 2:
                     cPixColMask=0x04;
                     break;
              case 3:
                     cPixColMask=0x08;
                     break;
              case 4:
                     cPixColMask=0x10;
                     break;
              case 5:
                     cPixColMask=0x20;
                     break;
              case 6:
                     cPixColMask=0x40;
                     break;
              default:
                     cPixColMask=0x80;
                     break;
      }

}
```

SOFTWARE OPERATION

The code for the previously described functions is contained in the module Command.c. The decoding for the commands occurs in the procedure ProcessCommand(), which is a simple C language switch construction.

The heart of all these functions is locating the bit in RAM where the function will start being implemented. This procedure is GenPixel, which takes the pixel number as an integer parameter and generates a value for three variables: cPixRow, cPixColByte, and cPixColMask. These are then used to address a byte in RAM as cMatrix [cPixRow][cPixColByte] and the particular bit is contained on cPixColMask (see Listing 1).

Rather than deal with each item in a given row, the items are handled individually, row by row. For instance, a letter will be copied from the character generator table to RAM before the next character is analyzed.

Another interesting procedure is IncPixPnt, which will increment the pixel being dealt with along a line, changing the cPixColMask and cPixColByte when necessary (see Listing 2). Each time a new row is started, the pixel pointer is set 40 pixels ahead of the pointer to the beginning of the current line. No attempt has been made to check for pixels beyond the limits of the RAM matrix. Remember that the leftmost pixel has a numeric value greater than the pixels to the right of it on the same line.

Listing 2—*IncPixPnt increments the compound pointer to point to the bit associated with the next pixel on the right.*

```
void IncPixPnt (void)
//Bumps the pixel on one place to the right, but not onto the
     next row
     if (cPixColMask==0x01)
     {
            cPixColByte--;
            cPixColMask=0x80;
     }
     else {
            cPixColMask=cPixColMask>>1;
     }
     if (cPixColByte>=5)
     {
            cPixColByte=0;
//Prevent overrun
     }
}
```

SWITCH INPUT

All of the above ignores the fact that the Smartswitch has a single, normally open contact. There isn't code to configure the microcomputer input (which could use the on-board pull-up resis-





tors), debounce the signal, or present it to an application through a RAM bit. But that should not be a problem.

DOCUMENTATION

When you use the PSoC Designer to construct a new project, the program wisely creates a new folder with all the necessary code and setups. Therefore, your modifications will be restricted to this project. One disadvantage to this approach is that when you transfer projects you have to use source files, and then the entire folder has to be zipped.

You can download the zipped file along with the Excel file for graphics creation. I have also included the PSoC datasheet created specifically for this project.

A SMART FINISH

The NKK Smartswitch is relatively expensive, so it would probably be in your best interest to minimize the number of switches per system. You can achieve this by using the backlight and timeouts. For instance, to invoke a setup menu, you could hold the button in until the software (after a period) changes the backlight from on to flashing. Answers to yes/no could be achieved through one or two activations of the switch or by a single push for yes and a 5-s default for no. The projects I design don't require frequent setup, so my user interface doesn't have to be quick. I hope to implement it using only one switch.

Note that in the time it has taken to publish this article, Cypress has released several new versions of the PSoC Designer. The latest version, 3.10, has implemented a lot of the changes that I've mentioned in these two articles.

Aubrey Kagan (P.E.) has a BSEE from the Technion, Israel Institute of Technology and an MBA from the University of the Witwatersrand. He has designed projects that have operated two miles underground in a mine to 600 miles above the earth in the International Space Station. He is currently a senior design engineer for Weidmuller Canada. You may reach him at akagan@weidmuller.ca.

SOFTWARE

To download the code and other files, go to ftp.circuitcellar.com/pub/Circuit_Cellar/2002/145/.

REFERENCES

[1] A. Bitti, "Excel Offers Painless LCD Initialization," *EDN*, www.ednmag.com, September 2001.

[2] A. Kagan, "Tricks to Improve on Excel Initialization," *EDN*, www.ednmag.com, April 2002.

SOURCES

PSoC microcontroller, PSoC Designer software
Cypress Microsystems
(425) 939-1000
(877) 751-6100
www.cypressmicro.com

NKK Smartswitch
NKK Switches Of America, Inc.
(480) 991-0942
www.nkkswitches.com



RS232/RS422/RS485 Converters



CMC's low cost converters adapt any RS232 port for RS422 or RS485 operation. These converters provide your RS232 device with all the advantages of RS422 or RS485 including reliable high speed operation (up to 200 kbaud) and data transmission distances up to 5000 feet. Two AD422s can be used to extend any RS232 link up to 5000 feet. Completely transparent to the system; no software changes of any type are necessary.

RS232 TO RS422
- Converts bi-directionally between RS232 and RS422
- Use as a short haul modem
- Plug in and go. No software changes required

AD422 (Requires 9VDC) $79.00
AD422-1 for 110VAC 89.00
AD422L signal powered 84.00

RS232 TO RS485 2 wire
- Makes your RS232 port an RS485 port
- Supports up to 40 RS485 devices
- Automatically determines data direction.
- Signal powered version available

ADA485 (requires 9VDC) $79.00
ADA485-1 for 110VAC 89.00
ADA485L signal powered 84.00

RS232 TO RS485 4 wire
- Converts an RS232 port for use with RS422 or RS485 devices
- Supports up to 40 RS485 or RS422 multidrop devices
- Adds multidrop capability to RS232 devices
- Automatically determines data direction.

ADA425 (requires 9VDC) $89.00
ADA425-1 for 110VAC 99.00

Mention this ad when you order and deduct 5%
Use Visa, Mastercard or company purchase order
code CC83

CMC Connecticut microComputer, Inc.
PO BOX 186, Brookfield,CT 06804 (203)740-9890
WWW.2CMC.COM Fax:(203)775-4595

ABOVE THE GROUND PLANE

Ed Nisley

PC Audio Bits

What's so hard about building a trouble-free audio system? All you need is a stock PC, good software, and decent digital data and high-resolution converters, right? Not necessarily. Ed shows you what the datasheets cannot as he notes what happens when bits meet hardware.

Remember the enthusiasm about the first multimedia features on PCs? I always wondered what was "multi" about sound in addition to light, but somehow "bimedia" didn't sound catchy enough. Well, decades later, we still don't have touch (the IFeel mice from Logitech mice aside), taste, or smell (fortunately).

One side effect of the multimedia revolution is that sound input and output jacks appear on all PCs these days, sporting decent resolution and frequency response. Although most people are content to just listen to their PCs and occasionally talk to them, geeks tend to think about what they might do with a fast CPU between pairs of ADCs and DACs.

A stock PC can become a decent audio-frequency oscilloscope, spectrum analyzer, function generator, or even an arbitrary waveform generator. It's just, as Steve is wont to say, "a simple matter of software." After all, with perfect digital data and high-resolution converters, what can possibly go wrong?

Well, quite a lot, as it turns out. Let's take a look at some fundamental things you don't see in the datasheets and the nastiness that can happen when bits meet hardware. I'll use a Wintel laptop for my examples in this article, but you can follow along with whatever hardware and OS you have available.

PURE ANALOG

At its heart, a sound card has audio inputs routed to an analog-to-digital converter to create bits from sound and a digital-to-analog converter routed to an audio output to create sound from bits. Most cards also include a MIDI synthesizer and tone controls, direct digital inputs and outputs, and so forth. For our purposes here, however, I'll use only the basic analog audio parts of the chain.

Sound cards include two audio mixers, which can provide a rich source of confusion. The Recording Control determines which audio sources reach the input of the ADC and sets their relative volumes. The Volume (or Playback) control selects audio sources for the output amplifier that produces the output voltage going to the speakers.

How you display these two mixers depends, of course, on which OS you're using. Photo 1 shows what they look like under Windows XP on my Dell Inspiron 8100. In order to display both controls at the same time, you'll probably have to start two Volume Controls, then use

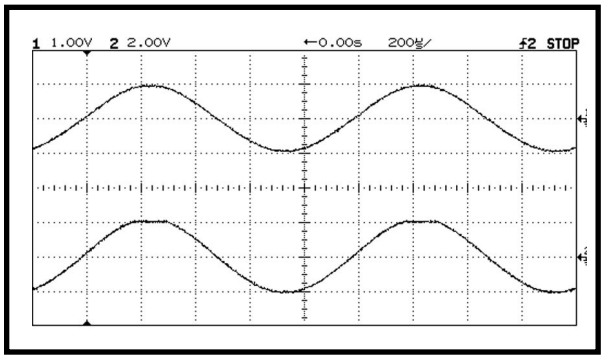


Figure 1—*The line-in jack can accept 1 V_P and the earphone jack will produce nearly 2 V_P into an open circuit without clipping. The digital input and output values are different.*

Options|Properties to convert one to a Recording Control.

The ESS Maestro sound circuitry in my 8100 allows only one input to the ADC at a time, despite being able to display a Mixer control slider. Fancier cards can perform on-the-fly audio mixing from various sources into the ADC input. As always, you (may) get what you pay for!

Not unexpectedly, dissimilar sound cards and drivers handle audio signals differently. The first thing you should determine is the maximum input and output voltages that do not cause clipping, along with the calibration of the volume control sliders.

You have to connect a function generator producing a 1-kHz, 0-V sine wave to the Line-In jack and connect the Line-Out jack to your oscilloscope. My 8100 has only a Earphone-Out jack and your PC may have only a Speaker Output, so measure what you have.

The first slider on the left in the Volume Control window, also confusingly labeled Volume Control, sets the overall audio output level. Move it about one-fourth of the way from the bottom, then set the Line-In slider all the way up. Don't use the sliders in the Recording Control window because you are not digitizing any audio yet.

Crank up the signal generator until the output waveform starts to clip, then back off just a little bit. The Maestro of the 8100 began clipping the sine wave's peaks at about 1 V_p. This determines the maximum input signal level.

Next, increase the master Volume Control slider until the output again begins to clip, which will give you the maximum possible output voltage. Figure 1 shows that I can expect just under 2 V_p from the Earphone jack with the slider one click down from the maximum setting.

The sliders have a roughly logarithmic response with the actual values determined by the audio hardware, drivers, and software. The ESS Maestro volume control sliders have roughly 20 values spaced 1.5 dB apart, even though the arrow keys move them in much finer steps. In this case, what you see on the screen is definitely not what you get at the output.

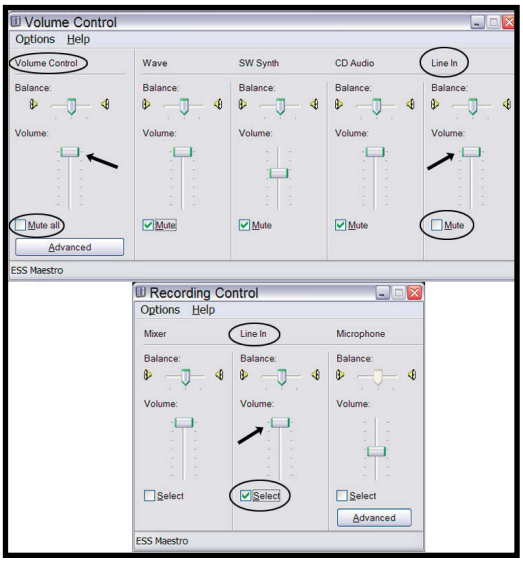


Photo 1—*Sliders on the two mixer controls set the amplification and attenuation for each signal source. Determining what each slider actually controls poses the first challenge to using a PC as a signal processor.*

Remember that decibels are calculated based on power, not voltage, so a 1.5-dB step corresponds to a voltage ratio of 0.84 and 3 dB to a 0.71 factor. As a result, the slider's half-scale setting of –7.5 dB produces about 840-mV_p output.

Because the oscilloscope presents a high impedance to the audio signal, you have just measured the open-circuit response. It's also instructive to find the circuit's output impedance and the maximum voltage for a load of that magnitude.

I have a military surplus audio output level meter with a staggeringly complex variable impedance control, so I can simply tune for maximum power (a whopping 15 mW) and read off the corresponding impedance (10 Ω) at the point when it begins clipping. You can get similar results by terminating the audio output with ordinary resistors and computing the delivered power for each. The resistance that absorbs the most power roughly equals the audio output's source impedance.

For what it's worth, 15 mW going into an ordinary earbud produces a deafeningly loud squeal. Don't do these tests with the earbud in place!

VOLTAGE IN

Although the line-in jack can accept a 1-V_p signal without clipping, the ADC that chops it into bits has a different limit. To find that, you must look at the digital values produced by the converter.

The Sound Recorder program included with Windows can suffice for a quick look. Use File|Properties to select a 16-bit stereo recording format with 44,100 or 48,000 samples per second, which gives the prettiest display. Make sure the Line-In slider in the Recording Control window is selected and shoved up all the way, and then click the Record button. Adjust the function generator's output until the pseudo-oscilloscope display just has rounded peaks, as shown in Photo 2.

It's worth noting that the waveform displayed in the Sound Recorder window has little to do with the input signal. Flat tops indicate clipping, but don't mistake it for an oscilloscope!

Next, measure the actual input voltage at the jack, which will have a peak value equal to the ADC's maximum input. On my system, that corresponds to only 325 mV, not the 1-V analog clipping level.

If you're using a DMM to measure the input voltage, remember that it

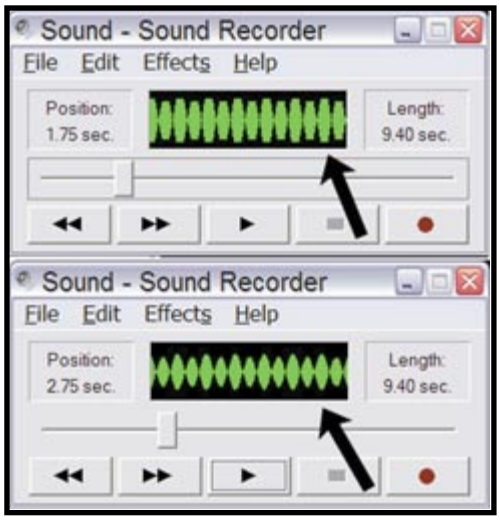


Photo 2—*Flat tops on a sine wave input give a quick indication of voltages above the ADC's maximum input level.*

displays the RMS value of the sine wave. Multiply the RMS reading by 1.4 to get the peak.

You can verify that you've found the correct level by playing the recorded sound back and monitoring it on your oscilloscope. In the Volume Control window, mute the Line-In slider (so you don't combine the analog input with the digital output), increase the volume using the wave slider, shove it up all the way, and then click the Play button.

Because the output Volume Control is still set to the maximum value that didn't cause clipping from the analog line-in source, the output voltage should be the maximum you can expect from the wave DAC source. Verify that by bumping the Volume Control slider up a notch and watching for clipping.

The ESS Maestro, like most sound cards, can digitize input signals with 16-bit resolution, which slices the entire 700-mV_{PP} range into 65,536 chunks. In theory, therefore, the least significant bit of the ADC represents just under 11 µV.

Let's look at the output side before we start celebrating.

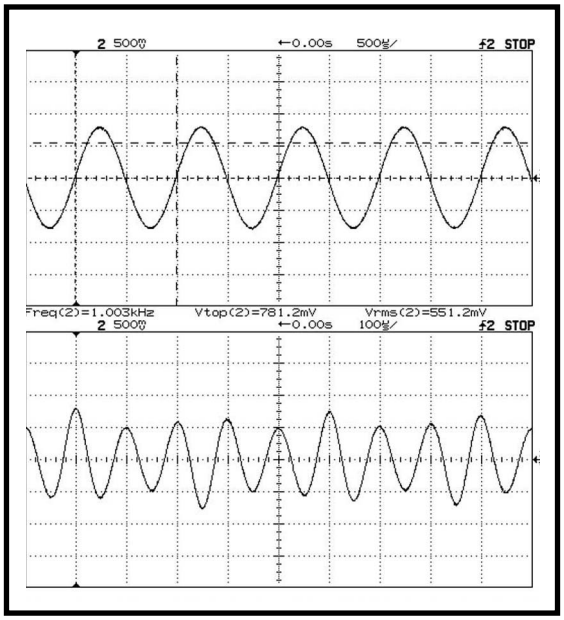


Figure 2—*A 1-kHz full-scale sine wave looks good. At 10 kHz, you can see a distinct interaction with the 48-kHz sampling rate.*

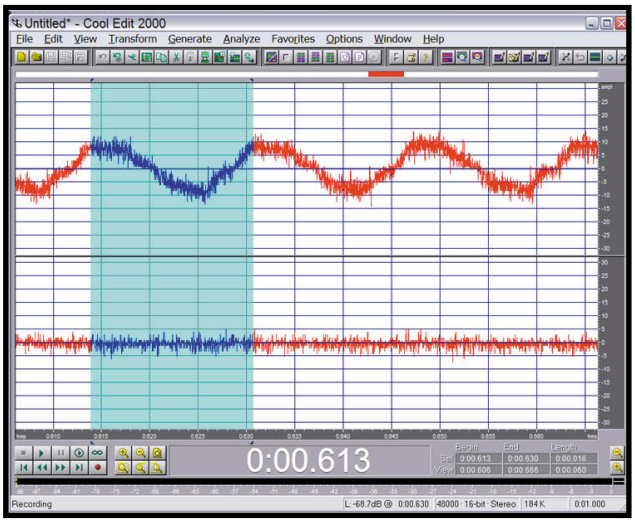


Photo 3—*Input noise can soak up a few LSBs all by itself. An open input produced the upper 60-Hz hum, a shorted one gave a flatter line.*

BITS OUT

Figure 2 demonstrates an interesting phenomenon. The 1-kHz full-scale sine wave in the top screen looks fine, but simply changing the frequency to 10 kHz produces the lumpy trace in the bottom screen. What happened?

Both waveforms use a 48-kHz sampling rate that produces one output voltage every 20.833 µs. One cycle of a 1-kHz wave lasts 1000 µs and contains exactly 48 output samples.

At 10 kHz, however, a single 100-µs cycle contains 4.8 samples, which doesn't work out evenly. The samples occur at 75° increments, so the two voltages bracketing what should be the full-scale peak at 100 µs actually have values of 0.5 and 0.97. The modulated 10-kHz waveform repeats every five cycles and 24 samples. You can hear the amplitude modulation imposed on the output even through cheap speakers.

It turns out that you are seeing the interaction between (at least) the output sampling rate, signal frequency, and the output reconstruction filter of the card. A search on the Internet will turn up more information on all of those; at least now that you know there's something to look for.

BRING THE NOISE! (AGAIN)

The output's ideal dynamic range spans all 65,536 levels of the 16-bit digital values. Figure 2 shows that full-scale output from the Earphone jack is about 800 mV into a 10-Ω load, so the least significant bit corresponds to about 25 µV (800/32k).

Figure 3 shows a 1-kHz sine wave that is 48 dB below full scale and should be about 3 mV_p. As you can easily see, the signal pretty much vanishes in the output noise.

When you run the numbers, 48 dB down from a peak of 32,767 works out to 130. Simply put, that means the low-order seven bits do not make any difference. The output may have an ideal resolution of 16 bits, but a useful dynamic range of only 9 or 10 bits.

Things may not be as bad as they seem, however, if you plan to use the output strictly as an audio signal. The noise frequencies tend to be significantly higher than the audible limit of about 20 kHz, so the output sounds better than you'd expect. In particular, I can easily hear a –80-dBFS, 1-kHz tone that is only 2 bits tall. At least with external amplified speakers cranked up all the way, it's still audible against a faint background hiss.

However, if you plan to use the output of a sound card as input to a device that isn't band-limited, you must include a filter to get rid of the hash. Pay careful attention to the low frequencies, too, because the DC offset shown in Figure 3 disappears when I unplug the lithium ion battery charger of the 8100.

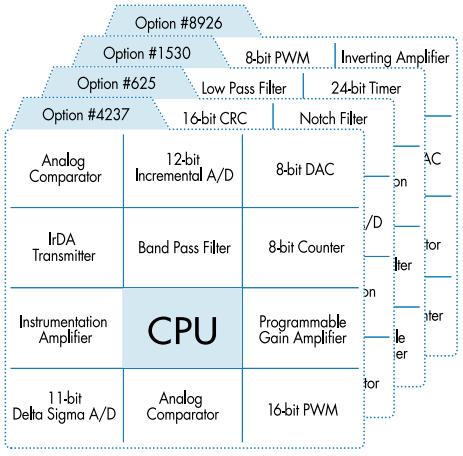
On the input side, I plugged a short 3.5-mm jumper into the Line-In jack and shorted one channel to ground. Photo 3 shows the results of digitizing two signals at 48 kHz: the expected 60-Hz hum on one channel and little on the other.



MCU later.

Customized MCU Design In 20 Minutes or Less.

Build your custom PSoC™ microcontroller with programmable analog and digital functions from our extensive mixed-signal library.



There are many more blocks to work with—and thousands of MCU configurations. To learn more about our innovative PSoC solutions and to enter a drawing to win a free PSoC Development Kit (a $1495 value), visit: **www.cypress.com/ad/mcu**.

Use the Cypress PSoC™ instead of an MCU for more flexibility, fewer parts and lower cost.

The versatile PSoC™ Programmable System-on-Chip™ microcontroller, winner of *EDN* magazine's Innovation of the Year Award in the 8- and 16-bit microcontroller category, is the world's first MCU that lets you custom configure the exact part you need.

- Graphically select, place, and interconnect the peripherals you want and adapt the architecture with PSoC Designer™ software.
- Dynamically reconfigure a single PSoC chip multiple times—changing functionality on the fly in any application.
- Reduce BOM cost by reducing the number of external components.

Cypress PSoC™ Microcontroller Family

Part	Flash Program Memory	SRAM (Bytes)	Pin Count	Packages
CY8C25122	4KB	256	8	PDIP
CY8C26233	8KB	256	20	PDIP, SOIC, SSOP
CY8C26443	16KB	256	28	PDIP, SOIC, SSOP
CY8C26643	16KB	256	44/48	PDIP, TQFP, SSOP

CYPRESS MICROSYSTEMS
A Subsidiary of Cypress Semiconductor

Cypress, PSoC, Programmable-System-on-Chip, and PSoC Designer are trademarks of Cypress Semiconductor Corporation. ©2002 Cypress Semiconductor Corporation.

STAMP VIDEO !!

LOW COST SINGLE CHIP VIDEO CONTROLLER



$25 EVAL (1)

$7.10 OEM (1K)

- Works with Stamp, z80, AVR, 8051, x86, PIC, etc.
- 8 col x 6 row alpha-numeric or 640 pixel graphic
- Includes high speed serial and parallel interfaces
- Drive CRT, TV, LCD with NTSC or VGA signal
- Source code examples in C, BASIC, Assembly

SINGLE CHIP COMPUTER $7

eval kit oem (1k) $1.99

-ZERO external components
-Built-in BASIC / Assembly
-RS232 program download
-1K flash,64ee,3irq,2timers
-15 I/O bits, A/D comparator
-20mips,faster than pic/8051
-20 pin DIP part #MV1200



NEW! 8K SUPER CHIP
40x the BASIC program space!
-32 i/o, 12 irq, 3 timers, Bus
-8K flash, 512ee, 512nvram
-Watchdog with internal osc.
-40 pin DIP part #MV8515
oem(1k) $7.10 eval(1) $25.00

CREDIT CARD COMPUTER 2



New PLUG-N-GO, no cables/power supply to buy!
Lo-power RISC cpu 10x faster than PIC, Z80, 8051
4meg NV mem,ser,par,RTC,4ch 12bit ADC,ISA bus
Built-in BASIC, Assembly, C compiler included
Friendly instruction set, unlike PIC or 8051
CCC2 eval(1)$75/oem(1k)$21 CCC1 $50/$14



LO COST MINI-PC

Includes DOS,NV mem,ADC,RAM,clock,ISA bus
Ports for serial, parallel, LCD, keyboard
Program in Turbo C, BASIC, MASM, etc.
Complete, no costly development kits required
Lowest power lowest cost PC compatible available
XT: oem $27 eval $95 AT: oem $55 eval $195

SERIAL MINI-TERMINAL

RS232 terminal for Stamp, PC, Z80, AVR etc.
-super low-current, powers from serial line
-LED backlit LCD, visible in all conditions
-115.2kbps, DB9 conn, simple commands
-specify 20 customizable or 16 tactile keys
eval(1) $75,oem(1k) $21.30,w/BASIC cpu $27



640x480 VGA LCD $27

Controller for most single/dual scan LCDs
Works with lo-res (160x120, 320x240,etc.)
Use with PC or SBC, standard VGA BIOS
Source code demo shows VGA initialization
Adaptable for other CPUs (i.e. Z80, HC11)
oem(1k) $27 evalkit(1) $95 w/10"LCD $195



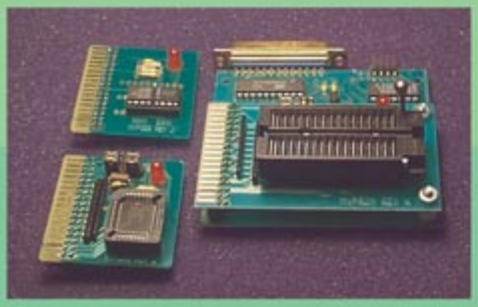
STAMP DRIVE



Read / Write PC compatible hard disk, PCMCIA, & Compact Flash on stamp, 8051, pic, avr, z80, x86
-up to 4 gigabyte capacity
-low power, simple commands
-RS232 @ 9600, 115.2k, 667k
-$95eval $27oem, IDE version$14

LIMITED TIME OFFER!
- FREE SERIAL EE
- FREE PLCC ADAPT.
- FREE ZIF SOCKET

$95 PRINTER PORT PROGRAMMER



LOWEST COST, FASTEST, EASIEST TO USE!

Does FLASH, EE, NVRAM, EPROM to 8m, (27080). Adapters for micros.
Super small size! Perfect for notebooks or drive bay of desktop computer.
WINDOWS and DOS software with script file ability to simplify operation.
Higher yield and quicker programming than others. "Gang Bang" option.

WWW.STAR.NET/PEOPLE/~MVS

MVS Box 803
Nash.,NH 03060
(508) 792 9507

.MVS

5yr Limited Warranty
Free Shipping
Mon-Fri 10-6 EST

SERVING THE EMBEDDED COMMUNITY SINCE 1979!

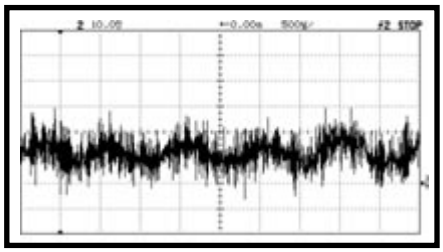


Figure 3—*A 3-mV$_p$ sine wave 48 dB below full scale is almost indistinguishable from the noise. So much for a 16-bit dynamic range.*

In short, you cannot expect to get the full dynamic range from a sound card without additional filtering. Obviously, the narrower the bandwidth you can use, the tighter your filter can be and the less effect the noise will have.

It's also vital to use the highest input signal levels you can generate. Expecting the sound card to amplify low-level inputs won't pay off, so use your own signal processing ahead of the inputs to put those bits to work!

CONTACT RELEASE

I've looked at only the most obvious characteristics of a single sound card and ignored a number of key parameters that you may find vital for your application. I highly recommend investing some time working with your function generator and oscilloscope to find out what your sound card can actually do. The results may surprise you, but it's better to be surprised in advance than to depend on the glowing specs you read on the box.

It should be obvious that not everything you read about sound card performance can be true all at once. That mythical 90-some-odd-dB dynamic range requires far more analog circuit performance than commodity hardware will provide, noise may limit applications other than PC-grade speakers, and the controls can hamper precise input and output settings.

But what you can do is truly amazing, particularly when you consider that this stuff comes free with every PC. Go forth and check out the analog part of digital sound!

Ed Nisley, PE, is an electrical engineer and a ham radio geek (call sign KE4ZNU). You may contact him at ed.nisley@ieee.or.

RESOURCES

Sound card interfacing
www.arrl.org/news/features/1999/0701/2/

Sound card test results
www.pcavtech.com.

SOURCES

SoundArb arbitrary function generator
David Sherman Engineering
(425) 258-4083
www.wavebuilder.com

Inspiron 8100
Dell Computer
(800) 999-3355
www.dell.com

Cool Edit 2000 software
Syntrillium Software
(480) 559-7778
www.syntrillium.com

TrueRTA oscilloscope, spectrum analyzer, function generator
True Audio
(865) 494-3388
www.trueaudio.com

On Time RTOS-32

On Time's royalty-free embedded RTOS for 32-bit x86 implements a Windows NT subset kernel in only 16k.

On Time RTOS-32 has a scalable component architecture. Only the components actually needed by the application are loaded on the target.



On Time RTOS-32 Features:

- Source and binary compatibility with Windows 95/98/NT/2000/XP
- Supports DLLs
- Unmatched real-time performance
- Context switch 0.73µs on P120
- Supports (but does not require) PC compatible hardware
- Boots from disk, BIOS extension, ROM, or DOS
- Windows 95 look-and-feel GUI
- Free tech support
- Full source code available
- Supports Microsoft, Borland, Delphi Compilers
- **Fully integrates with MS Visual Studio 6/.NET**
- **Free Evaluation Kit** available at http://www.on-time.com
- **No run-time royalties**

On Time RTOS-32 Components:

RTTarget-32
Development System
Boot Code, Locator, Cross Debugging, Win32 Emulation Library
Price: $1500

RTKernel-32
Real-Time Multitasking Scheduler
Hard real-time, unlimited number of tasks, 64 priorities, 1µs time resolution
Price: $1500

RTFiles-32
Embedded FAT File System
FAT-12, FAT-16, FAT-32
For floppy disks, IDE hard and flash disks, M-Systems DiskOnChip, linear flash
Price: $1500

RTIP-32
Embedded TCP/IP Stack
BOOTP, RARP, ARP, ICMP, UDP, TCP, DNS
Add-Ons: PPP, FTP, TFTP, Web Server, SMTP/POP3, TELNET, SNMP, DHCP
Price: $3000

RTPEG-32
Embedded GUI
Windows 95 look-and-feel Graphical User Interface
Price: $2500

North America: On Time • 15 Spur Lane • Centereach, NY 11720 • USA
Phone (888) 667-8200 • Fax (631) 471-8850 • email info@on-time.com

International: On Time • Hofweg 49 • 22085 Hamburg • Germany
Phone +49-40-2279405 • Fax +49-40-2279263 • email info@on-time.de

http://www.on-time.com

On Time
REAL-TIME AND SYSTEM SOFTWARE

Design Logic CONTEST 2001 ENTRY

FEATURE ARTICLE

Frantisek Bachleda

Digital Ignition System

Building Without a Distributor

Tinkering with cars can be an expensive hobby. But upgrades don't always have to cost a fortune. In this article, Frantisek shows you a cost-effective way to build a digital distributorless ignition system that will boost your car's overall performance and power.

Does it make sense to build an ignition system when it's possible to buy a complete engine control system (with fuel injection) from a commercial manufacturer? I'll answer that question in this article. I'll discuss a digital ignition system without a distributor for two-cylinder internal combustion engines.

There are many old vehicles whose owners would like to equip them with new technology to improve performance, fuel consumption, engine control at wide range of temperatures, or the precision of the spark timing.

If you're a performance seeker with a limited budget (and a cheap old vehicle), you know there aren't many affordable high-performance engine controllers and calibration tools, especially if the engine has only two cylinders (I have a Fiat 500, which uses the same engine as a Fiat 126). I decided to build an electronic fuel injection and ignition control system that can easily be set for optimum power and performance. I built the ignition system first because it could be developed separately and is an easier problem than fuel injection.

The system satisfies all of my requirements. It's inexpensive; the most expensive parts are the two transistors that charge the coils. The system is able to send all operating information to a PC. In addition, you can reprogram the ignition curve via UART from the PC in real time (you can modify it and hear the effects on the engine). And lastly, I've implemented temporary incremental curve correction for use with the knock sensor.

In most vehicles equipped with electronic control systems, an inductive pickup senses the rotation speed and position of the crankshaft. I decided on another solution. I got a spare distributor assembly and sensed the position and speed of the camshaft using two optoelectronic sensors that are offset by 180°. A rotating disk that has one hole and is secured to the original distributor shaft interrupts the beam. Pulse timing is set so that the pulses come 45° before top dead center (BTDC). Take a look at the timing diagram in Figure 1.

In order for the engine to work optimally, the spark must come at the right point in the rotation of the crankshaft. This depends on the revolutions per minute and the octane number of the gas, among other things. The right point is specified in terms of the crankshaft angle in degrees BTDC. The microcontroller calculates the revolutions per minute from two incoming pulses and then waits an appropriate amount of time to generate the spark to the correct cylinder.

STANDARD IGNITION SYSTEM

The ignition system of a gasoline engine operates at high voltages. An ignition coil produces the high voltage. The point at which the spark occurs during the compression stroke is called the ignition timing. Making the spark occur earlier is called advancing the timing; conversely, allowing the spark to occur later than normal is a called retarding the timing.

There are several general principles regarding ignition timing that you should keep in mind. First, it's difficult to move ignition timing during a race. Second, the more power an engine can develop, the more you can advance the timing. Third, advancing the timing causes the spark to occur earlier and makes for a bigger bang

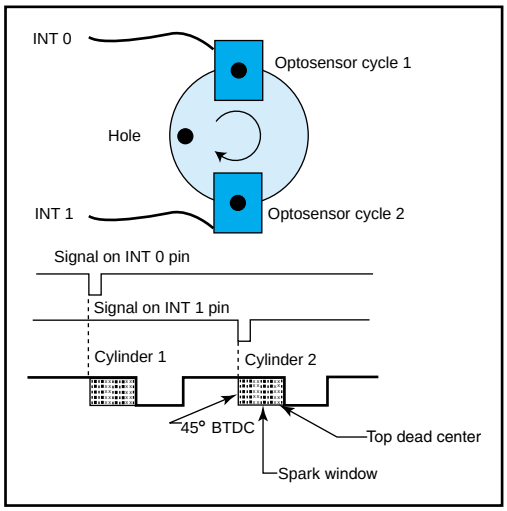


Figure 1—*My solution includes two optoelectronic sensors and a rotating disk that is fastened to the distributor shaft. You can also see the signals on the microcontroller interrupt inputs related to the pistons at top dead center (TDC).*

during combustion (bigger bang equals more power). Fourth, too much advance causes an underpowered engine to struggle and overheat (digitally controlled engines have a knock sensor to prevent this situation, which could happen as a result of different working conditions). Finally, too much retardation causes a loss of power and overheating.

A knock sensor is installed in electronically controlled engines, especially those with fuel injection. It listens to the explosion sound because the engine can be damaged if the spark comes too soon. If this happens, the sensor hears it and the processor retards the ignition timing. This also reduces the power from the engine, but the processor compensates for this by changing the air/fuel ratio.

The spark must come at the correct point before top dead center. Ignition timing depends on the revolutions per minute (i.e., if the piston moves faster, the ignition must start earlier for complete combustion) and gasoline octane number. Higher-octane fuel burns slower, so the ignition needs to occur earlier for complete combustion.

The standard distributor on the Fiat 500 consists of a contact breaker with its capacitor, the ignition cam, the actual distributor, and an automatic timing control device that determines the optimum ignition timing relative to the operating conditions of the engine.

The lower end of the distributor's shaft is connected to a gear in the engine block; its upper end carries the ignition cam. This shaft rotates at the same speed as the camshaft of the engine. The ignition circuit consists of two sub-circuits, the primary (low voltage) and secondary (high voltage). As the shaft rotates, the ignition cam breaks the primary current flow (by pushing the contact breaker), which causes high voltage to pass from the coil through a rotor in the upper end of distributor shaft to the spark plugs. The rotor position and moment when the contact-beaker opens are so arranged that the rotor is always at a contact just when the contact breaker interrupts the circuit of the primary winding of the ignition coil. This creates a spark in the appropriate cylinder.

The standard distributor also contains the timing control. This device automatically adjusts the ignition timing to its optimum value. A centrifugal control device provides basic adjustment. The centrifugal force (dependent on the revolutions per minute) changes the relationship between the ignition cam and contact breaker. So, when the shaft rotates faster and the centrifugal force increases, it causes a larger deflection and earlier ignition.

The negative pressure in the intake manifold of the engine produces an additional adjustment. This partial vacuum, which develops behind the throttle valve in the manifold, is used to adjust the ignition curve.

What I have described is a standard ignition system with a mechanical distributor. Many manufacturers are moving to digital ignition systems without distributors, because they're advantageous in several ways. Ignition energy is not wasted sparking across the contact breaker in the distributor, and there is no need to design vacuum lines and centrifugal mechanisms because these parameters are calculated by a microprocessor based on the revolutions per minute. If there is a coil for each cylinder, there is no mechanical rotor. This provides a better spark at high revolutions per minute because each coil has enough time to charge.

Microcontroller-controlled ignition systems are easy to maintain, calibrate, and upgrade. The microprocessor can be programmed to deliver maximum power under all conditions (e.g., different grades of gasoline). In addition, the system has no mechanical parts to wear out, and the lack of breaker points reduces the electromagnetic interference that causes errant signals in other electrical equipment.

The ignition system consists of several key components (see Figure 2). The mechanical pieces support the assembly and couple the mechanical motion of the engine to the electronics. Power-conditioning circuitry provides clean power to the electronics.

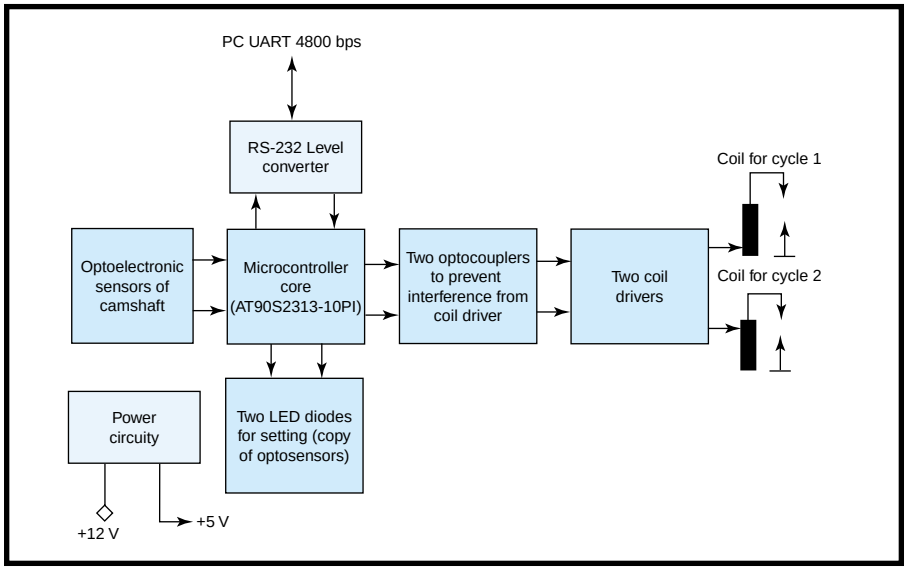


Figure 2—*I chose the AT90S2313-10PI to run my ignition system.*

Two optoelectronic sensors (consisting of infrared LEDs and phototransistors of the type currently used in remote controls for TVs and other electronics) sense the camshaft position.

A microcontroller (AT90S2313-10PI) manages the process. Two photocouplers are used to protect the microcontroller from possible interference coming from the coil switch transistors, and two indicator LEDs indicate the status of the input sensors in order to easily set the pulses at exactly 45° BTDC. A pair of coil drivers interfaces to the ignition coils, and an external RS-232 level converter provides an interface to an external PC.

MECHANICAL PARTS

One standard solution to detect the position and speed of the crankshaft in electronic ignitions is an inductive sensor placed near a gear on the camshaft. As I mentioned before, I have decided on another solution—two optoelectronic sensors.

Everything (sensors and electronics) is built into a spare distributor. As a result, I have made a compact device that is as easy to attach to the engine as a standard distributor.

I disassembled the distributor and took out the parts, keeping only the shaft and basic body of the distributor. The distributor shaft is in two parts for the centrifugal control, which must be secured together. I drilled a hole through these two parts and riveted them together to form a single shaft.

To make things easier, I had a machine shop turn three new parts. Two of the parts are for the stator, where the optosensors are connected. The third part is the disk (stuck on with silicone rubber and secured by a bolt) on the upper end of the distributor shaft. This disk has a hole to interrupt the optosensor beams. The stator parts are secured with bolts to the distributor body. The electronic boards are located above the mechanical parts and everything is covered. You can see the overall mechanical arrangement in Figure 3.

POWER SUPPLY

If you want to use a microprocessor in a car, especially to control the engine, it's necessary to provide it with a clean power supply.

The power circuitry consists of a filter and voltage regulator. The filter includes a toroidal coil (with as much inductance as possible) and additional capacitors connected between the supply wire and ground, both before and after the coil, to prevent the propagation of fast transients that can appear on the car's electrical system.

The filtered voltage goes into a regulator (78M05) that provides the voltage for the microprocessor and other circuits. A pair of capacitors (electrolytic and ceramic) is connected between the regulator input and ground, and also between the regulator output and ground.

The regulator and its capacitors are located on the microprocessor board, which is built into the new distributor assembly. The filter (i.e., toroidal coil and its bypass capacitors) is in a separate box secured near the ignition coils.

Despite all my effort to provide a clean supply for the microcontroller, it might not be enough. You can provide an absolutely clean voltage for the MCU by using totally separate grounds, isolating the MCU and logic ground from the car ground. The power supply is a DC/DC converter with isolation transformers (the cheap way is to use an appropriate Maxim or Motorola IC).

Such a solution would provide an absolutely clean voltage for the microcontroller. I am working on a solution now, but I decided to use a complete DC/DC converter module made by TRACO Electronic. It is more expensive, but it's easier to implement. You can see the modifications in Figure 4.

OPTOELECTRONIC SENSORS

For the optical sensors, I used phototransistors and infrared LED emitters (3 mm in diameter) like those used in consumer electronic remote controls, because they are cheap and easy to find.

In systems with an inductive sensor (with a gear on the crankshaft), the input signal conditioning is possibly the most crucial circuit in the ignition system. Sensor output signals must be modified in some way to resemble a signal that a microprocessor can read. Because only one inductive sensor is used, there is information missing about the position of the camshaft. Cars still have only one ignition coil and a high-voltage distributor is used to direct the spark to the correct cylinder. Only more expensive cars use a coil for each cylinder, and the electronic control system in them has the sensor on the camshaft.

Optosensors have numerous advantages. First, they are easy to connect to the microprocessor (no additional circuits are required to provide a sig-

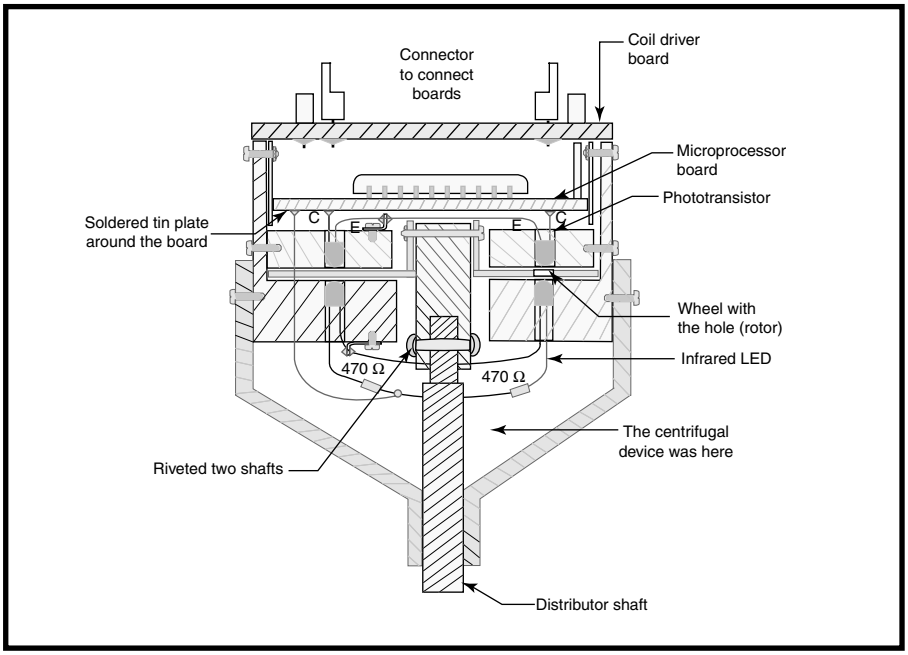


Figure 3—*With a little effort, you can assemble all of the mechanical parts with the electronic board. Modifications to the classic distributor allow you build in optosensors and an electric board.*

nal that the processor can read). Second, the sensing of the camshaft position includes information about which cylinder needs the spark. Two ignition drivers and two ignition coils are used. Third, the system uses two sensors (one for each cylinder), which allows you to use one ignition coil for each cylinder (no mechanical distributor eliminates sparking there). Coil energy is used to make a spark in the cylinder, so electromagnetic interference is decreased and each coil has more time to charge. Finally, optosensors are simple mechanical solutions.

Despite these perks, there are some drawbacks. Dust and oil could contaminate the optosensors. Such contaminants can block the light and put the system out of operation. Revolutions per minute are calculated only once per spin, which is a problem for the first spark because the microprocessor can calculate the revolutions per minute only after the second pulse occurs. Finally, timing inaccuracies could appear because the sensors are on the camshaft and not the crankshaft. The wearing of the chain that connects the camshaft and crankshaft (in the ratio 1:2) can cause this.

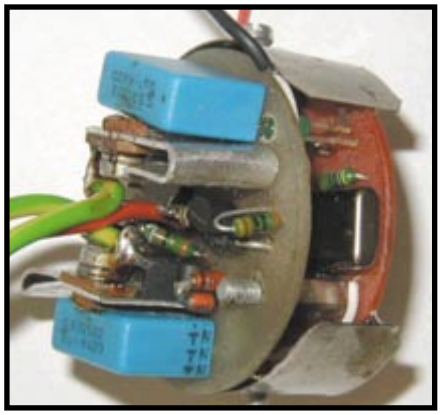


Photo 1—*The coil driver board is stacked on top of the microprocessor board. The boards are 42 mm (1.65″) in diameter.*

Each LED emitter has a series resistor to adjust its current to approximately 5 mA. The emitters of the phototransistors are connected to ground and the collectors are connected to the microprocessor input pins (INT0 and INT1). External pull-up resistors are also connected between each collector and 5 V. This generates low-going pulses on these pins when the shaft rotates and the hole in the disk activates the photodetector.

MICROPROCESSOR CORE

There is nothing special here. The microprocessor is connected in a standard way. There are two green LED diodes on PB6 and PB7 that display the status of the input sensors to simplify calibration at 45° BTDC. I'll come back to this later. A 10-MHz crystal is used as a clock source.

OPTOCOUPLERS

Because the switch coil driver transistors could be a possible source of voltage interference to the microprocessor circuitry, I decided to insert two optocouplers by which the microprocessor can drive output transistors. These couplers are in SMD packages soldered to microprocessor board.

COIL DRIVERS

The most expensive parts are the coil driver transistors. These transistors must withstand high collector-



emitter voltages and high peak collector currents. I used a BU931P transistor in TO220 package, which is targeted for ignition applications.

The transistors are connected according to the manufacturer's recommended circuit. Each is driven by two small transistors (a BC640 and one that is part of the optocoupler), which provide the gain required to drive the output transistors from the microprocessor. Because the output transistors work in switching mode, they need only minimal heatsinking (i.e., a small piece of aluminum plate), as shown in Photo 1.

Before I move on, I would like to make a few important points regarding automotive applications. First, you should remember that all electronic components and all other materials used must work in the wide range of temperatures encountered in the engine compartment (e.g., the microprocessor is the PI version). Secondly, because of the vibration, it is also necessary to secure all electronic components with silicone rubber. Supporting them only via their soldered leads is not durable enough.

SOFTWARE ARCHITECTURE

The microprocessor software is organized primarily around four key interrupts: external 0, external 1, timer compare, and serial UART receive. The AVR source was compiled with the AVR macro assembler. A C program to generate EEPROM data for the ignition curve was compiled as a WIN32 console application in Visual Studio 6.0. The watchdog is enabled.

All spark handling is done in the interrupt routines. The main loop program handles UART communications with the PC. The external 0 interrupt is handled by the same routine as the external 1 interrupt. The only difference is that a flag is used to direct the spark to the correct cylinder.

The timer interrupt is set to occur every 50 µs. There is a 3-byte free-running counter in software that is incremented by the interrupt handling routine. The value of this counter is read in the external 0/1 interrupt handler, and the actual revolutions per minute is calculated by subtracting the current value of the counter from the previous value.

The handler then sets a 2-byte variable that determines the time to wait to create the spark, based on the revolutions per minute value. After this variable is set, the timer interrupt routine starts to decrement this value. When it becomes zero, the spark is generated (the output transistor is turned on for 1.8 ms) in the correct cylinder. The delay value is set in one of three ways, depending on the revolutions per minute value (see Figure 5).

When you start the engine, the revolutions per minute value is less than 400. The delay value is calculated by dividing the revolutions per minute value by eight, which establishes a fixed delay of 45°, causing the spark to occur at 0° BTDC. It probably

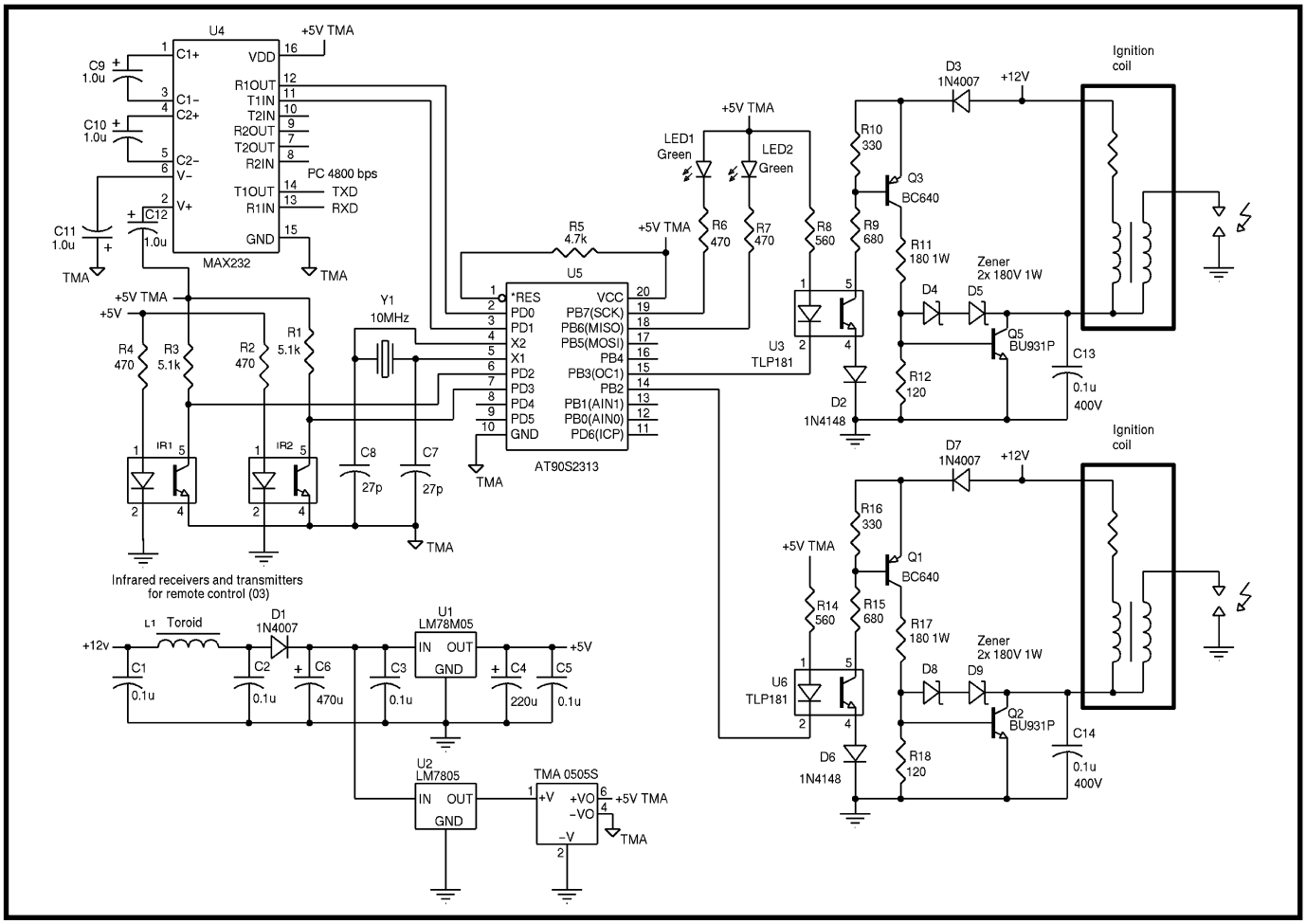


Figure 4—*Take a look at the modifications I've made. They are the electronic solutions to the problems of a digital ignition system without a distributor.*

should not be zero exactly, but rather about 2° to 3° BTDC. I used zero because it provides a quick implementation (division by eight is equivalent to a 3-bit right shift).

The working range of revolutions per minute is 400 to 6451. The data for this range (124 values) is stored in an EEPROM data table. This range is changeable via the UART in real time. Normally, the revolutions per minute does not go higher than 6451. Above approximately 3500 rpm, the spark advance is constant at about 28° BTDC. This is calculated by multiplying the revolutions per minute value by 1.5/32, which produces a fixed delay of 17°.

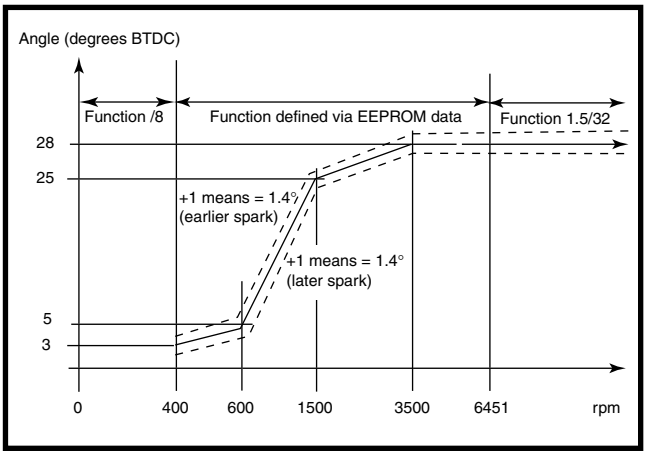


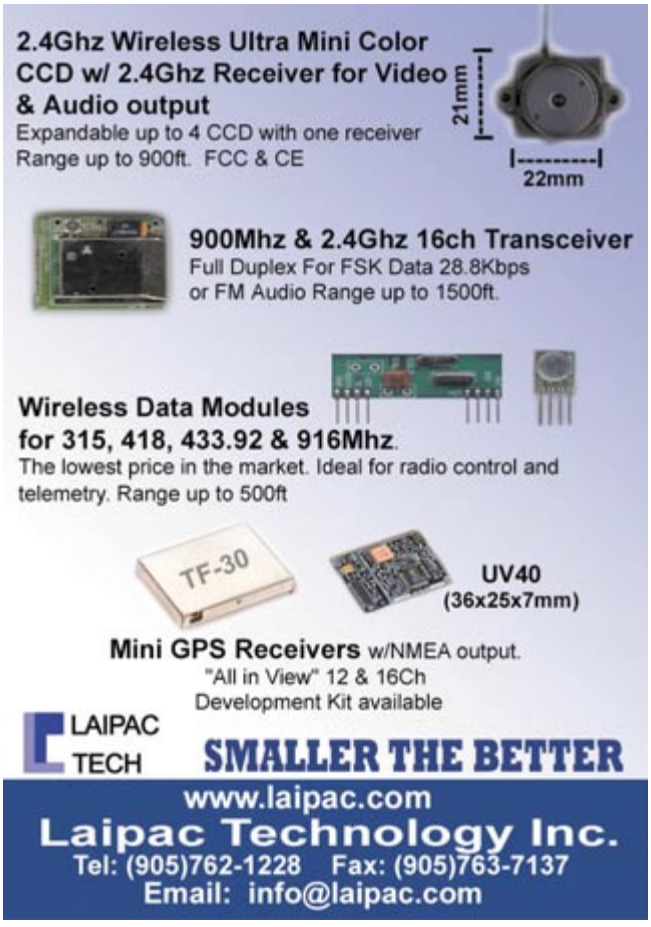
Figure 5—*Spark timing is dependent on the revolutions per minute as it is defined via software.*

There is a problem with the very first spark. When the first pulse occurs, the microprocessor does not have a previous timer value that it can use to calculate the revolutions per minute. There are two bytes reserved in EEPROM for this situation. After a second impulse occurs, the microprocessor calculates the revolutions per minute, divides it by eight, and stores this value in EEPROM. The value is then used as the delay value for the first spark and is used immediately after reset.

The microprocessor is able to communicate with an external PC via its UART. The communication is in ASCII characters. Any terminal program can be used on the PC. Set it for 4800 bps, 8 data bits, 1 start bit, 1 stop bit, and no parity.

The serial protocol consists of simple 6-byte command packets sent to the microprocessor and the corresponding responses. Eight commands are available. The first byte of each packet is the command number (1 through 8), followed by five bytes whose meaning depends on the specific command. Each byte is ASCII. So, the packet 10324x (ASCII codes 0x31, 0x30, 0x33, 0x32, 0x34, and 0x78) means write the value 0x24 to address 0x03 in the EEPROM. The details of each command are given in Table 1.

When the processor is ready to receive a command, it transmits the word "Ready." If you enter an invalid command (other than 1 through 8), the command is ignored, but the packet must still contain 6 bytes. This means that you must enter throw-





away characters until the processor responds with the "Ready" prompt.

CALIBRATION AND INSTALLATION

There is a mark on the crankshaft pulley and another on the engine block. When they are lined up, the crankshaft is at top dead center (TDC). It's a good idea to make a new mark at the position 45° before TDC because this makes it quick to set up the new distributor assembly each time you're ready to try it. Of course, it is also necessary to install a second ignition coil into the car and connect it to 12 V.

Installation proceeds as follows. First, set the crankshaft to 45° BTDC, and then insert the new distributor assembly (with the cover off) into the engine. Next, you need to connect 12 V to the device and rotate the main body of the distributor to the position where one of the green LEDs turns on (the impulses must occur at 45° BTDC). Secure the distributor with its clamping bolt and put the cover on. Then, connect the wires to the ignition coils and the high voltage cables from the coils to the spark plugs.

At this point, you should try to start the engine. If it doesn't start, try swapping the high-voltage cables on the spark plugs. Murphy's Law says that the first attempt won't be successful. Because there are only two cylinders, the trial-and-error method works well.

Try to start the engine again. If it still doesn't start, then there's an error in the design. In this case, it's a good idea to disassemble the distributor and spark plugs, switch on the voltage source, rotate the distributor shaft by hand, and look to see if the distributor generates sparks. Don't forget to ground the disassembled spark plugs and distributor.

When you finally have the engine running, the last task is to tune the ignition to the optimal curve for the engine. You can see the preliminary curve in Figure 5. This curve is stored in EEPROM and program memory. The latter is provided in case a mistake is made during the tuning process. You can restore the EEPROM data to the original curve using the 8 command. Restart tuning again from the beginning without reprogramming the EEPROM in the device programmer.

For curve tuning, it's good to have the knock sensor connected. If you do not have one, you must be careful when adjusting the ignition timing. You should listen to the engine's sound after each adjustment step.

Connect the PC via the serial interface module to the distributor. Execute a terminal program by setting the UART parameters to 4800 bps, 1 stop bit, and no parity. When everything is connected and the engine is started, the distributor will send a "Ready" message to the terminal and you can start to tune.

The point-by-point tuning method goes as follows. First, ask the distributor for the revolutions per minute, EEPROM address, and value (4 command). Second, vary the ignition tim-

Command Packet	Description	Response
"1", addr_h, addr_l, data_h, data_l, "x"	Update byte in EEPROM; when adjusting ignition curve, you need to update values in EEPROM	"OK" if successful, "Fail" otherwise, and then "Ready"
"2x", "+" or "–", deg_h, deg_l, "x"	Correction of spark timing; move the whole timing curve in multiples of 1.4° (eg., +1 means +1.4° and +2 means +2.8°). The value is written into RAM and correction stays in effect until the next reset.	"OK" if successful, "Fail" otherwise, and then "Ready"
"3xxxxx"	Delete correction—set the timing curve to zero	"OK" and then "Ready"
"4xxxxx"	Send revolutions per minute data, EEPROM address, and wait for a constant	TTTTTaAA = DDDD, where TTTTT = number of 50-µs ticks for one crankshaft revolution, AA = EEPROM address and DDDD = delay value, which is also in 50-µs ticks. Then, "Ready."
"5xxxxx"	Send spark timing correction value; show actual timing correction value accumulated by the 2 commands above	±DD (units of 1.4°), then "Ready"
"6xxxxx"	Send all EEPROM data	AA = DD, AA = DD, AA = DD… The DD values are in units of 100 µs
"7", adr_h, adr_l, "xxx"	Send 1 byte from EEPROM; get one value from EEPROM data memory at the specified address	DD, where DD is the requested byte, followed by "OK" and then "Ready"
"8xxxxx"	Update EEPROM ignition curve from program memory. In the program memory is stored data that represents the first ignition curve stored in EEPROM. Because you can change the EEPROM data, it can be set back to the initial curve without reprogramming the device in an external programmer.	"OK" when finished, and then "Ready"

Table 1—*The 2 command is helpful during fine-tuning, but it is primarily intended for the support of a future fuel-injection controller. Note that the data units for the 4 command are 50-µs ticks, while data units for the 6 and 7 commands are 100-µs ticks.*

ing using the 2 command and listen to the engine for optimal operation for that number of revolutions per minute. If you hear detonation, retard the timing. Third, reset the curve offset to zero (using the 3 command) and write the corrected value for this number of revolutions per minute to the EEPROM (using the 1 command) at the address reported in the first step. Finally, increase the revolutions per minute (accelerate the engine) for the next EEPROM address/value and repeat the tuning procedure for the entire range of revolutions per minute.

A word of caution: Advancing the spark too much can be dangerous to the engine. Start with a conservative curve and gradually advance it. If your engine has a knock sensor, monitor it. If it doesn't, then be gentle. Advancing the spark until you can hear a knock is not a good idea because knocking is potentially damaging. Have you ever seen a hole burned clear through a piston? It isn't a pretty sight.

After the curve is optimized, it's a good idea to read out all of the EEPROM data (using the 6 command) and save this data to a file on the PC. You can also insert the new curve into the program memory, recompile, and reprogram the AT90S2313 device for the next development cycle. Be sure to have a lot of fun!

FINAL NOTES

This system also could be used in a four-cylinder (or more) engine if it provides one impulse per crankshaft rotation (e.g., a Hall effect sensor on the crankshaft) at the position 45° BTDC. These impulses could be connected to one of the external interrupt pins (INT0 or INT1). The unused input pin should be connected to logic 1, or this interrupt should be disabled in the software. The system will have only one coil driver (on an appropriate output pin) and the standard high-voltage distributor will direct the spark to the correct cylinder.

The serial UART port can be used as a communications channel to integrate the ignition system with other microprocessor systems, such as a fuel-injection controller or a dashboard computer system that displays the actual revolutions per minute.

WAR STORIES

This project was the result of many attempts, some exploring, a lot of tests and mistakes, and the incremental process of development. As every developer knows, you encounter many interesting (and funny) events during the development process, especially if you do tests with something as simple as a car engine. Let me tell you about two events that occurred while I was developing this ignition system.

In the first version of the device, I made all of the mechanical parts at home. The stator was cast from epoxide and the rotor was made from plastic, but it wasn't very precise.

Approximately once a month, I travel with my wife and young daughter to my parents' home, which is about 120 km from my home across some hills. One time we traveled on a Friday afternoon. While driving up the first hill, I heard (and felt) the engine lose power and start jumping. But after a few minutes, the engine started running well again. At the top of the hill, I was not able to decide whether or not we should continue.

We decided to go, but after a few more kilometers we could see that it was not a good decision. So we turned around with the hope that the car could make it back to town. That was a terrible few kilometers. At times it seemed like it would have been better to stop some car and have our car towed back to town. There were four kilometers uphill that I had to drive in first gear while the engine screamed terribly.

Later, at home, I disassembled the ignition system and saw that my imprecise mechanical parts were polluting the optical interrupters. The rotor and stator sometimes touched each other and this produced small filings that blocked the infrared beam. We ended up making that trip by train.

Another time, we traveled on a holiday to a cottage about 300 km from our town. We traveled during hot weather (about 30°C, or 86°F). Of course, I had already added the second ignition coil to my car. The original coil did not have a series resistor and worked without problems. I bought the same type for the second coil, but the new coil had a series resistor. So, I decided to remove the resistor in order to have a bigger spark.

When we were about 150 km into the trip, I heard a big explosion in the engine space. The engine immediately stopped. When I opened the engine compartment, I saw that the new ignition coil had overheated and burst its top cover. There was asphalt everywhere from inside the ignition coil. I realized that removing the series resistor was a terribly bad idea.

Fortunately, by happy coincidence, our friends were passing through the village where our car died. They towed our car to their home and we spent the night. In the morning, I bought a new coil, connected the series resistor, and we continued our journey.

These stories show the darker side of the development process.

Frantisek Bachleda is a software developer for Ekosoft. He earned an MS in the Theory of Automatic Control from The Technical University of Kosice, Slovakia. You may reach him at bachleda@ekosoft.sk.

SOURCES

AT90S2313-10PI microcontroller, AVR macro assembler
Atmel Corp.
(408) 494-3388
www.atmel.com

Visual Studio 6.0
Microsoft Corp.
(425) 882-8080
www.microsoft.com

BC640 transistor
Philips Semiconductors
+31 31 40 279 1111
www.semiconductors.philips.com

BU931P ignition coil driver
SGS-Thompson Microelectronics, Inc.
(602) 485-6201
www.st.com

DC/DC converter module
TRACO Electronic AG
+41 1 284 29 11
www.tracopower.com

FEATURE ARTICLE

Design Logic CONTEST 2001 ENTRY

Jeff Bachiochi

RISCy Business

Part 1: RISC Projects by Cornell Students

Jeff is heading back to college. In this article, he walks you through several projects that Cornell University students entered in the Atmel Design 2001 contest. So, grab a pen and pad and get ready for a lecture on the RISCy business of digital system design.

I've heard it said, "If you remember the '60s, you probably weren't there." I was there (I think). College was a blur of all-nighters, girls, frats, slide rules, heavy books, and something called laundry. Sparsely scattered throughout this blur were hours of class lectures, often at some unforgiving hour of the morning. I certainly didn't appreciate what I had access to then, but I'm ready for school now. At least my mind is ready. It's my body that couldn't take the abuse, as I remember it.

I'm reflecting on the good old days because of Atmel's Design Logic 2001 contest. I'm sure this doesn't make any sense to you, so let me explain. In looking over the list of entrants to Atmel's contest, I noticed an interesting fact that would have been undetectable in a normal list of names. However, in this list, I noticed numerous e-mail addresses with the same domain: cornell.edu. Ten entrants came from Cornell University. What was going on?

The Cornell web site proudly displays the headline: "Cornell freshman Travis Mayer, who majors in food science, won the Olympic silver medal in men's freestyle skiing." I guess I'm not surprised, because Ithaca, NY measures snowfall by the foot instead of the inch. The web site also shows that Cornell's College of Engineering offers instruction in a full range of disciplines in 10 engineering fields. I picked a random entrant and e-mailed him with a few questions about the group (I just assumed these entries weren't isolated efforts). The reply affirmed my suspicions. The entrants were all from a class on microcontroller design (EE 476).

Bruce Land (br14@cornell.edu), the course professor and administrator of independent student microcontroller projects, encouraged his students to enter their projects (along with one of his own) in the contest. Professor Land's senior course enables students to carry out sophisticated designs of modern digital systems, which now appear in products such as automobiles, appliances, and industrial tools. He explained to his students that the basis of such systems is the microcontroller, a microcomputer optimized for single-chip system design by possessing many peripheral devices geared to real-time applications. The students used Atmel AT90-series RISC microcontrollers for their projects.

Professor Land's course was a design course. The students were expected to demonstrate considerable creativity, flexibility, and motivation. They had to utilize the 'Net and spend time in the library. In addition, the students had to implement material from many of their other courses and find solutions on their own from incomplete specifications. Essentially, Professor Land's course resembled the real world.

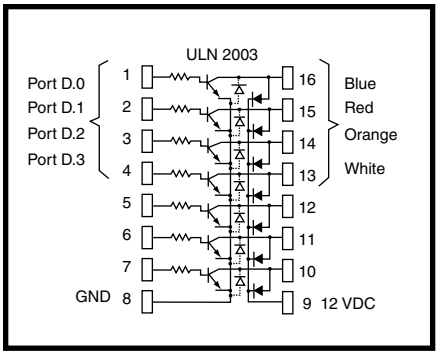


Figure 1—*The ULN2003 provides seven coil drivers in a single IC package (including protection diodes).*

AS A RESULT

This conglomeration of contest information seemed too good to pass up. Professor Land was kind enough to act as director of communication for me. I asked each of the contest teams to help me put together a feature about their contest design entries. Although you won't find any of their projects in the official winner's circle, I want them to know (and all of you who enter any of *Circuit Cellar*'s contests) that they don't have to take home a prize to be appreciated. And so, I present to you the students of Cornell University. In Part 1, I'll cover the first six projects, starting with the Vertical Plotter.

VERTICAL PLOTTER

Gone is the chalk dust. White boards have practically put an end to the vile screech of fingernails dragging across a blackboard. Victor Aprea (vaa4@cornell.edu) and Paul Grzymkowski (pjg19@cornell.edu) make use of this new technology by implementing an Atmel AT90S8515 microcontroller to drive a stepper motor interface.

Dual steppers are mounted in the upper corners of a white board. Each stepper has a spool of string attached to the shaft. The free ends of both spools are fastened to a dry marker mount, which hangs suspended from the two steppers. Each stepper can independently pull in and pay out string. The hanging marker can be positioned anywhere around the white board. If the motor movements are choreographed precisely, the marker will draw specific shapes or patterns by leaving a trail of erasable color behind.

The 12-V stepper motors have 48 steps per revolution. Using approximately 1.25″ diameter spools, the motors can pull in or pay out roughly 4″ of string per revolution for a resolution of around 0.08″ per step.

The controller also handles an LCD and keypad interface. The single-line LCD gives you feedback on data entry. The keypad allows you to calibrate and initialize the position of the marker. In addition, after entering the list of x, y coordinates, the plotter will go off calculating and initiating motor movements to proceed from one coordinate to the next.

After experimenting with an unknown stepper motor, Victor and Paul found that the five-leaded stepper is a unipolar motor. A unipolar stepper requires a single supply. It has a common lead and an additional wire for each of the four coils. Motor direction is determined by the sequence in which the coils are energized.

Victor and Paul designed discrete circuitry to drive each of the coils. Later they found an integrated driver for seven loads, the ULN2003 (see Figure 1). Using two of these drivers (one for each of the two motors) proved to simplify the prototyping.

After attaching the motors to the white board, disappointment struck in the form of insufficient torque. They altered the stepper motor coil timing to allow overlapping coil enables, which provided two powered coils at all times, increasing the available torque (see Figure 2).

The most difficult part of the project was determining how the two motors worked together to get from point A to point B. The movement choreography establishes the shape of the line drawn. According to the two students, if they had more time, they would have added a PenUp/Dn function, using a solenoid.

Victor and Paul were highly motivated to work on this project because they were able to choose their own task. As a result, they were eager to learn about all of the project's intricacies and really wanted to see it succeed.

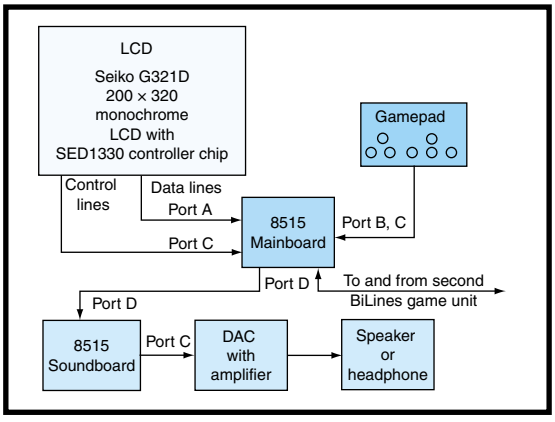


Figure 3—*Take a look at the BiLines hand-held gaming unit.*

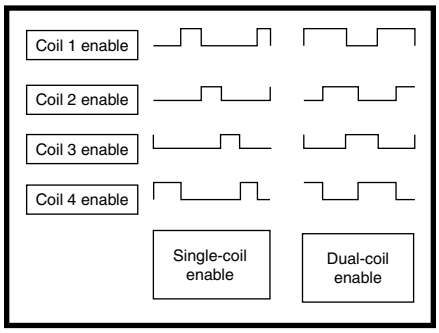


Figure 2—*Increasing the number of coils energized at the same time increases the available torque.*

BiLines

You can always tell a real gamer, they don't think of much else. Where else could you design a game and get class credit for it? BiLines is a Tetris-type game for one or two players.

Michael Jordan (Mmj5@cornell.edu) and Crystal Soja (cas53@cornell.edu) used an Atmel AT90S8515 microcontroller in their BiLines design because it has enough I/O for a gamepad, an LCD, a soundboard interface, as well as a serial connection for competition play. In fact, a second '8515 is used to create the soundboard. The soundboard's micro gets its instructions from the main processor and plays background music to help keep your mind on the game (see Figure 3).

The LCD is a Seiko G321D (200 × 320 monochrome graphics display with a built-in controller). With this LCD, 25 lines of 40-character text can coexist with graphics. The gamepad is made up of a set of seven push buttons. Two-player action synchronizes game start for a head-to-head battle with the identified winner or loser (see Photo 1).

As one of seven different shaped game pieces drops from the top of the playing field, the player must direct the piece by horizontal movement and rotation controls. When the game piece has settled at the bottom of the playing field, another piece begins to drop. If a player can position the falling pieces to fill in two complete rows, they are eliminated. Otherwise, the game pieces will rapidly stack up, resulting in a quick finish to the game.

The soundboard accepts commands to play game sounds for start/pause, cleared lines, and

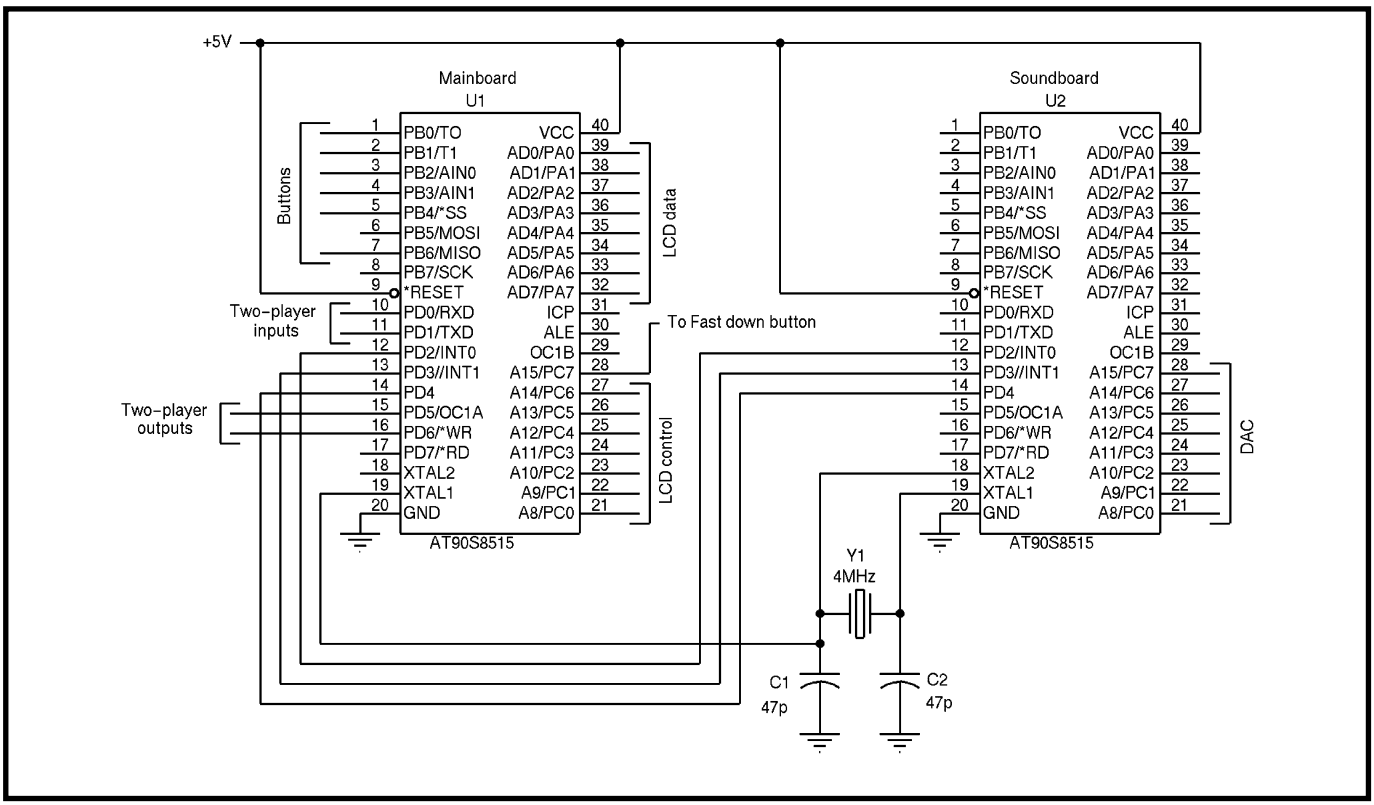


Figure 4—*Both processors run on the same 4-MHz crystal. The soundboard adds audio as a slave device to the main board's game application.*

game over. In addition, two music tunes, "Canon in D" by Johann Pachelbel and "Für Elise" by Ludwig van Beethoven can be selected for continuous play. An R2R ladder on an output port creates a hardware DAC for the sound output. A table of values sent to the DAC reproduce a sine wave output. The interrupt-controlled duration between samples determines the pitch of the note while a second timer determines note's duration (see Figure 4).

When I first demonstrated Telstar Pong for my parents in the '70s, they couldn't have cared less. Why would anyone choose to watch a white dot bounce back and forth across the screen when you could have your choice of watching any of the three channels our black-and-white TV received? Little did I realize how interacting with the screen would affect my future. This project proves that gaming is not a fad. Crystal and Michael created something more intriguing and advanced than Pong.

Timing was crucial for their game, so having two timers was essential. C was used to speed up the development time, but the two designers said they could have used more memory space.

Michael designed all of the playing pieces based on a 4 × 4 matrix. Copies of the pieces along with their possible rotations are stored in a look-up table. This took far less processing power than to do rotational calculations on the falling piece.

Crystal worked on the LCD code combining both text and graphics modes. In addition, she designed the soundboard to generate music accompaniment for the game.

Crystal and Michael relied heavily on their friends for beta testing. They found it difficult to interpret comments about piece behavior without actually experiencing (seeing) the problem.

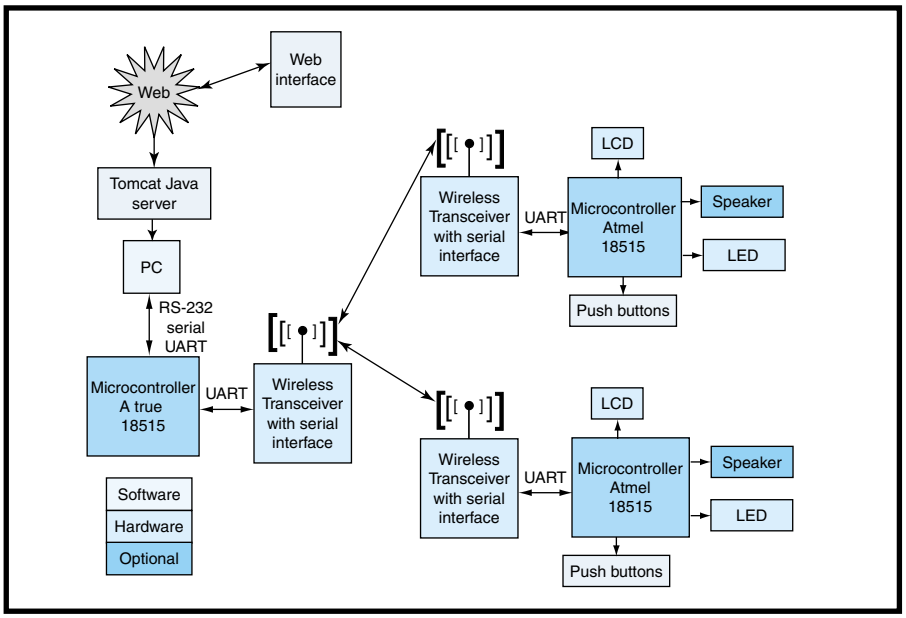


Figure 5—*This block diagram illustrates how the components are used to construct the Wireless Pager system.*

Cornell tends to focus a lot of attention on theory. Crystal and Michael found their BiLines project to be a great way to apply the theories they had been studying since freshman year.

WIRELESS INTERNET PAGER

In the past, when a beeper broke the silence (often at a seemingly inappropriate time) you could rest assured that a doctor or nurse was following up on an emergency. Today's annoyance comes from a wide selection of cell phone ringers. In most cases, no emergency exists. Pagers still play an important role in situations when personal contact isn't required. In this project, Kevin Ferguson (kpf1@cornell.edu) and Anish Jain (aj32@cornell.edu) used multiple Atmel AT90S8515 microcontrollers to create a server/receiver paging system (see Figure 5).

Messages created through a browser are processed by a JAVA servlet and sent to the Atmel server connected to the PC's serial port. The micro handles message traffic and time of day (TOD). The hardware UART supports the PC communications while a software UART is used to communicate with the RF transceiver. Radiometrix BiM2 433-MHz transceivers are used for wireless data communications between the server and the pagers.



Photo 1—*It isn't exactly portable, but this prototype version of BiLines can be played. Note the two separate processor boards.*

The RF pager units contain both a BiM2 and an '8515 microcontroller. The addressable controller recognizes only messages containing its address. The processor handles incoming messages as well as supports an LCD and push buttons. A 4 × 20 LCD displays pager messages and user menu prompts. Push button inputs allow you to switch between display modes. TOD is synchronized from the server to assure an accurate TOD display. Along with each text message, the sender can initiate two optional methods of message indication. The pager LED can be enabled as a visual message indicator and a piezo buzzer can be enabled for audible message indication.

Kevin and Anish based their project on Atmel's application note "AVR305: Half Duplex Compact Software UART." [1] This application note describes how to implement a polled software UART capable of handling speeds up to 614,400 bps on an AT90S1200.

The tiny Radiometrix modules are effective to about 50 m (inside) and 200 m (outside). Kevin and Anish agree that the carrier detect output of the transceiver is helpful in identifying spurious data (noise) during non-transmitting periods.

Using a browser to submit pager data via a form allows messages to be transmitted from anywhere. The local



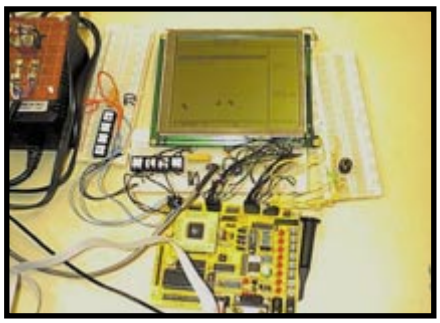


Photo 2—*The Cooking Coach uses an Atmel ATmega103 microcontroller and a large 200 × 320 pixel LCD to display plenty of data.*

web server handles the short haul RF communications with the pager units.

The Jakarta Tomcat Servlet Engine allows the form's JavaScript to do some text checking. The Sun Java comm.api provides access to the PC's serial port.

COOKING COACH

For many, eating is one of the pleasures of life, even if it's fast food. Some people find great pleasure in the preparation of culinary treats. The Cooking Coach, which was developed by Paras Shelawala (phs11@cornell.edu) and Rodolfo Cuevas (rac32@cornell.edu), archives recipes, generates shopping lists, and prompts self-proclaimed chefs with preparation instructions.

An Atmel ATmega103 microcontroller uses an external 32-KB SRAM to hold recipes while interfacing to a Seiko 321D 200 × 320 pixel LCD for menu prompts and preparation instructions. Seven push buttons provide user input. While three buttons have fixed functions, four buttons are considered soft keys and receive their function via the LCD screen (see Photo 2).

Appliance (range/oven) control is simulated through an output port. LEDs connected to the output port indicate the state of appliances (i.e., burner/oven/buzzer). Even though the Cooking Coach initiates automatic appliance control, appliance operation prompts are presented on the LCD as well.

Recipes are loaded via a serial port. Each line of a recipe begins with an command, which will instruct the application on how to handle it. Only six simple commands are necessary: B, E, I, S, A, and T1/2.

The start of a new recipe is indicated by B, which includes the recipe's name. E signifies the end of a recipe. The I command contains an ingredient and S indicates that the following is a step in the preparation of the recipe. A indicates an action to be taken (e.g., appliance/object output on/off). Finally, T1 or T2 indicates a timing period for one of the two timers. This can be used in creating a delayed action, like turning off an appliance after so many minutes. Battery operation allows the Cooking Coach to travel to the grocery store and remind the chef's helper which items need to be purchased for the preparation of selected recipes.

Hungry for knowledge? Although this project satisfied the minimum daily requirements for education, it gave Paras and Rodolfo little time for eating. They had to change processors mid-design because of a lack of I/O. Because some of the '103's pins have a restricted purpose, Paras and Rodolfo had to go back several times and reassign the purpose of each port's pins.

The two students agreed that there are additional features they would have liked to incorporate in their design. A touch screen would have eliminated all of the user push buttons, and a network adapter would have reduced the reliance on a PC for recipe downloads, which might come from various web sites or cooking shows. Oh yeah, I like that. Watch your favorite cooking show and give a command that downloads the recipe, gathers and prepares the ingredients, and cooks the meal without getting off the couch.

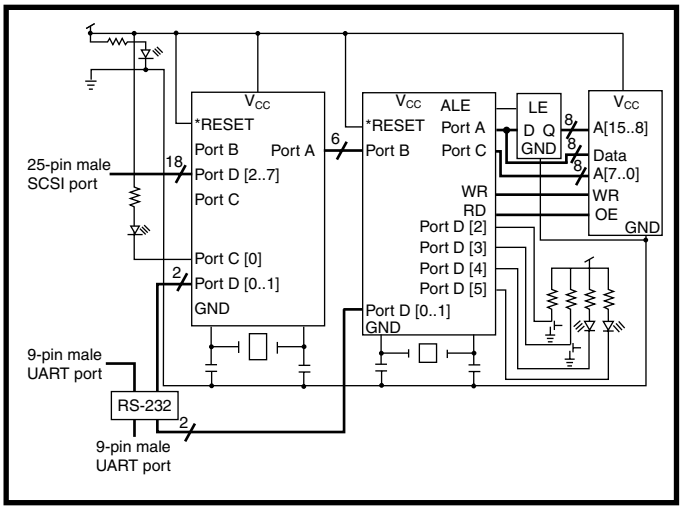


Figure 6—*The UART of the first processor handles the digital camera interface while the second UART offers some debugging help for the zip disk interface.*

BARBIE'S ZIP DRIVE

Doesn't Barbie have just every accessory imaginable? You'd think so after all these years. As you'll see in this project, Barbie rides the crest of technology. Here, Barbie's camera (digital) is used as input for a storage system, which includes a SCSI interface to store her pics on a zip drive.

Chris Bartholomew (cab52@cornell.edu) and Sean Keller (sjk26@cornell.edu) used Twin Atmel AT90S8515 microcontrollers to control the camera operation and SCSI interface (see Figure 6). The camera can be placed in single-shot or stop-motion video mode by the first micro when sent serial commands at 57,600 bps. The '8515 can receive and store one picture (20,680 bytes) in its external SRAM. The data then can be transferred to the second micro for storage.

The second '8515 handles two interfaces, an interprocessor interface and a SCSI interface. The SCSI interface is a modified SCSI DB-25 connection supported by zip drives. Because of I/O limitations after the SCSI interface definition, the interprocessor interface is limited to nibble data transfer with handshake lines. Data transfer with the SCSI device is based on a 512-byte block/page. Thus, the picture data must be buffered (micro one) so transfer can be paused.

Chris and Sean found that working with the digital camera was a snap because of the hardware UART support in the '8515. They had difficulty with the zip drive interface because of a lack of documentation, so they used the ANSI/ISO SCSI interface documentation to probe the drive to determine what protocols it supported. The 25-pin zip drive interface isn't an SCSI standard, but is used in the MAC. A lack of I/O prevented the guys

Use Your Imagination.

New Communications Modules... Visit Our Website!

Call about Micromint's new LOW cost 56k & 2400 baud embedded modems. TCP/IP also available!





Delivering Embedded Controllers for Over 20 Years. 20 YEARS

Embedded controllers with an attitude.

With a bit of imagination and a relentless pursuit of innovation, the divers that film Shark Week™ for the Discovery Channel figured out how to extend their dive times from only 30 minutes to eight hours by creating a re-breather that uses PicStic 4™ technology.

With over 250,000 controllers in the marketplace, Micromint has been providing innovative, turn-key solutions to the OEM market for over twenty years– from design through production, as well as packaging and shipping the final product. Our broad line of embedded controllers and turn-key solutions can turn your imagination into reality.

Micromint

Visit our website @ www.micromint.com to see our complete line of OEM Solutions.

902 Waterway Place | Longwood, FL 32750 | 800·635·3355 | 407·262·0066 | Fax 407·262·0069

from supporting the 8-bit, 50-pin SCSI A-type cable.

Chris and Sean spent over 40 man-hours soldering and wiring the hardware together. Getting the prototype wired up is an important part of any project because none of the software can be tested in the real world without it. Although simulators are fine for debugging those software routines, when it comes to I/O, you need the actual hardware.

Chris and Sean wrote test routines for the transferring of data between two MCUs, which allowed them to have confidence in their interprocessor communication. Debugging small sections of code at a time helps to localize the search for problems when your application misbehaves. Hey Engineer Barbie, are you taking notes?

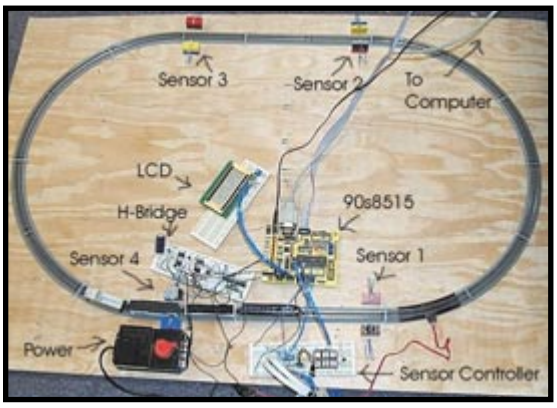


Photo 3—*The N-gauge layout includes sensors going into and out of each turn. The sensors going in allow automatic speed control and prevent any train from entering a turn at potentially high speeds.*

SUPER TRAIN CONTROLLER

Trains, locomotives in particular, have always fascinated me. It might be their massive power, or the promise of unobtainable freedom known only to the hobo. Almost everyone I've known has had a set of electric trains. In this project, Chad Potocky (cap34@cornell.edu) and Ballard Smith (bjs30@cornell.edu) use this hobby to teach us something about computers and control.

An Atmel AT90S8515 microcontroller has plenty of steam for motor control, track sensors, a stand-alone user interface, and PC control via a serial port. The processor's PWM output is used along with a direction output bit to derive H-Bridge control signals for an N-scale locomotive (see Photo 3).

Although a single-chip H-Bridge might be a simple solution, making a discrete circuit goes a long way in teaching how things work (see Figure 7). Four TIP32/31 power transistors driven by four 2N3904/6 transistors make a simple H-Bridge. 4N35 optocouplers isolate each of the four control signals and keep the logic supply separated from the motor's noisy power supply. The PWM and direction output signals are combined using discrete logic to produce either forward PWM control or reverse PWM control signals for the H-Bridge.

Four infrared LEDs and phototransistor pairs are used to detect the train's position at four places along the track layout. As the train moves between an IR transmitter and receiver, the invisible beam is interrupted



and the receiver signals the microprocessor through four inputs.

Five push-button inputs provide user input for manual control of forward, reverse, slow down, speed up, and stop. The 4 × 20 character LCD displays feedback on the current mode of the train. Dynamic speed control is programmed into the system to prevent the train from derailing or from switching direction at full speed and to simulate the acceleration and deceleration characteristics of the real world. By placing the sensors before and after corners the speed can be automatically adjusted to prevent running full speed on dangerous curves. The PC can feed the Super Train Controller with simple commands to keep the model running on a preprogrammed schedule of service.

Initially, train operators are similar to slot car drivers. They want to know how fast their baby can go. After learning that the sport is keeping the cars and engine on the track, things go a lot smoother. Actually, moving slowly is part of the authenticity of model railroading.

The scale speed of many amateur model trains is more like that of the bullet train, than the actual model being represented. To make the train more realistic, Chad and Ballard implemented speed control optimizations. The inertia is measured by the slow ramp-up/down to the desired speed. This is based on the weight of the load being towed.

The LCD gives the engineer feedback, indicating the status of the engine. This is a real-time guide to the direction, speed, and progress of any preprogrammed script. Chad and Ballard needed to add one-shot filtering to the optical sensors because each car would again trigger the sensor as the space between them provided an unobstructed view to the sensor.

This project provides motor control to the track itself, which means that additional engines on the same track will not have individual control. According to the two students, if they had additional time they would have investigated how they could have controlled the engine through an addressable on-board controller.

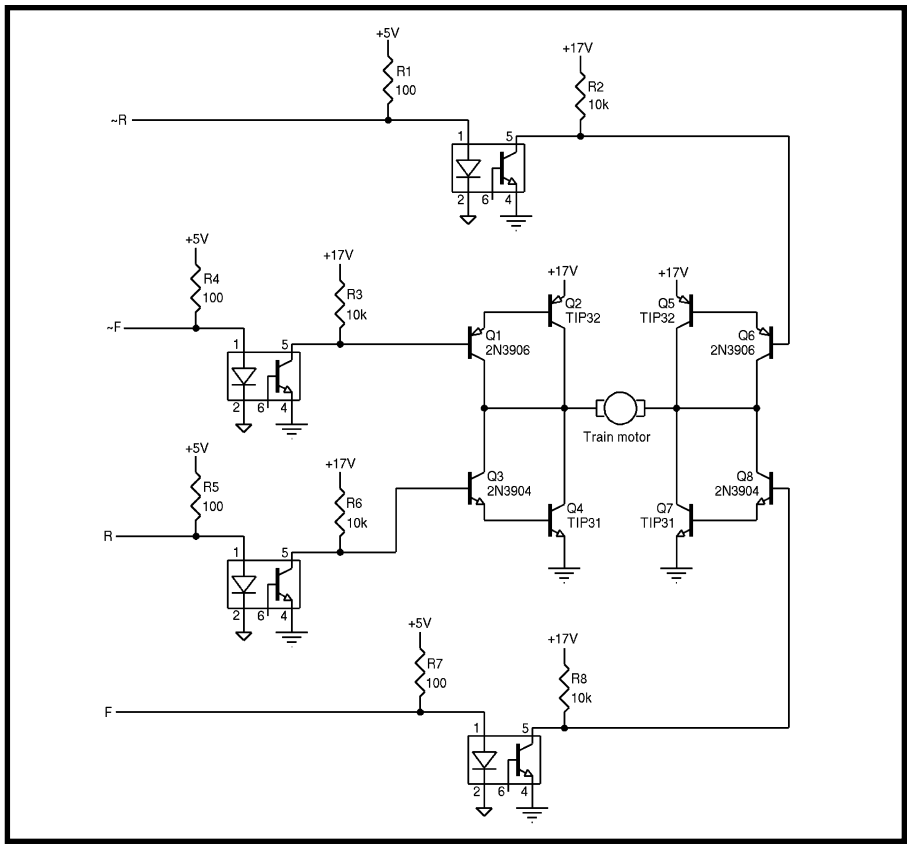


Figure 7—*This optocoupled discrete H-Bridge will keep the logic control well isolated from the power track power.*

However, it's difficult to experiment with on-board control using N-gage because it is so small.

Digital control is rapidly becoming an integral part of model railroading. Like upgrading any device, compatibility issues often add complexity to the system just to provide a smooth transition to those already with large investments. All aboard the digital train.

NEXT TIME

Next month I'll introduce you to the rest of Professor Land's class and their Atmel projects, which range from home audio control to a wireless messaging device.

Jeff Bachiochi (pronounced BAH-key-AH-key) is an electrical engineer on CircuitCellar*'s engineering staff. His background includes product design and manufacturing. He may be reached at jeff.bachiochi@circuitcellar.com.*

REFERENCE

[1] Atmel Corp., "AVR 305: Half Duplex Compact Software UART," 0952A-A, September 1997.

SOURCES

AT90 RISC microcontrollers
Atmel Corp.
(408) 494-3388
www.atmel.com

HT-640 Encoder, HT-648L decoder
Holtek Semiconductor, Inc.
886 2 2782 9635
www.holtek.com

BiM2 Transceiver
Radiometrix, Ltd.
+44 020 8428 1220
www.radiometrix.co.uk

TWS434 Transmitter, RWS434 receiver
Reynolds Electronics
(719) 269-3469
www.rentron.com

321D and G321D LCDs
Seiko Corp.
+81 3 3563 2111
www.seiko-corp.co.jp

FEATURE ARTICLE

Neil Cherry

The Open-Source HCS Project

It's been a while, so now it's time to set the record straight. Despite the rumors, the HCS II project is not dead. In fact, HCS has been licensed and is now an open-source project. In this article, Neil brings us up to speed on the HCS II project's past, present, and exciting future.

The inmates have taken over the asylum! The HCS users have gotten together and asked Steve Ciarcia to let us turn the HCS project into an open-source project. Steve graciously licensed the HCS II project to us under the FSF's GPL. *Circuit Cellar* has no affiliation with the current project other than its history and the fact that many of the project's participants are subscribers.

Up-to-date details of the project can be found at our Source Forge web site, and we're keeping the discussions open and in the *Circuit Cellar* HCS newsgroup. This makes perfect sense because the HCS users have had a group there since the HCS II inception in 1992. Now, many of the HCS users may think the project has been put away and forgotten, but we're here to let you know that it's up, running, and has a bright future.

So, here's what happened. During the early months of 2001, a few users asked about the status of the HCS II. On April 11, 2001, Charles Byrnes posted the fateful question, "Is HCS dead?" A flurry of responses then went back and forth, and on April 23, 2001, Steve posted, "Is the HCS II Dead? Not if you help." Since then, more than one-third of the messages posted were on subjects related to the resurgence of HCS. I would say that there is still significant interest in the HCS II.

Over the next few months, a large number of the online HCS community chipped in to perform all sorts of duties to pull together the Open-Source HCS project. We still have much to do and plenty of work to spread around. Also, please remember that this effort is totally voluntary. We are not getting paid to design hardware or write software. We do it out of curiosity and because of our interest in automation.

SO WHAT'S AN HCS?

The HCS II is an expandable, standalone, network-based (RS-485), intelligent-node, industrial-oriented supervisory control (SC) system intended for demanding home control applications. The HCS incorporates direct and remote digital inputs and outputs, direct and remote analog inputs and outputs, real time or Boolean decision event triggering, X10 transmission and reception, infrared remote control transmission and reception, remote LCDs, and a master console. Its program is compiled on a PC with the XPRESS compiler and then downloaded to the SC where it runs independently of the PC.

The HCS was first introduced by Steve and Ken Davidson in *Circuit Cellar* 25 (February and March 1992). Since then, there have been many articles that documented the HCS, such as the RTOS, COMM-Links, and network protocol. All of this information can be downloaded from *Circuit Cellar's* HCS online library (www.circuitcellar.com/HCS/PDFindex.htm).

WHO IS IT GEARED TOWARD?

The project is geared toward the type of people who read *Circuit Cellar* magazine, especially those of you who are interested in home automation and have a do-it-yourself (DIY) attitude (Don't you just love obvious answers?).

Most likely, we will not have plug-'n-chug boards, so some soldering, programming, and assembly will be

required. It is doubtful that anyone would be willing to put out the monumental effort required to pull together the stock and assembly for the boards, not to mention the outlay of cash required to put such a venture together. Let's just call it a lesson learned the hard way by more than a few of us already involved in the project.

The people currently involved have backgrounds in various fields of electronics and software. For those of you with little knowledge of electronics, we provide links and other resources so you can get involved without going to school for an electronics degree. We also provide similar resources for people who wish to program but have little knowledge of programming languages.

For this project, the group will attempt to use off-the-shelf components and prototype boards to make it easier to find the parts. The user community that congregates in the *Circuit Cellar* newsgroups is active. The community's input is important to the project. New users are welcome and questions are encouraged.

GOALS

The initial goal is to provide support for the existing user base. So far, we've managed to get all of the software for the HCS II SC (ROM images, host, and compile). The group is now working on getting the ROM images for the original COMM-Links. We have the source code for a few of them and need to compile them. Mike Baptiste (www.cc-concepts.com) was kind enough to turn his entire library of PIC-based COMM-Links into open source. We're also working on posting the hex files to Mike's COMM-Links and creating new PIC-based modules to use his code.

One of the major stumbling blocks is the host software. It was originally written to run under DOS. This works well for Linux using a DOS emulator (and 16,550 UART serial ports) and Windows (up to '98). But for later releases of Windows, the host software fails because the OS will not give it direct access to the serial ports. I've got bits and pieces of Perl/TK code, which I'm hoping to put together into a program called tkhost. This program should be portable between Windows and Linux, which would allow a larger base of users to get involved.

In addition to the current support, we have plans for the future. Currently, there are plans for another Zilog-based controller that we're calling the HCS II.V. It's more advanced, faster, and much more expandable than the current HCS II. A few of the developers have started working on the HCS III, which was an idea that Mike Baptiste put forth last year. Mike's idea is based on Rabbit Semiconductor's development kits.

Both of these projects will support the current Express code and COMM-Link modules. The group is also putting together plans for an Ethernet and a second generation of faster COMM-Links.

CHOICES

I'm sure you have a lot of questions as to why we did what we did. Without going into minute detail (join the Local.cci.hcs2 newsgroup for the exact details), here are some of our decisions and the reasons why.

The first thing we needed was a license that would allow sharing of the code without abusing the code. We read through about 20 different licenses, and after some discussion (and watery eyes) concluded that the GPL was the appropriate license. Other licenses were too specific to their projects, too loose in their control, or just did not apply to what we were doing. We decided right away that writing our own license was not an option. After all, we're engineers not lawyers.

We then proceeded to easier things such as deciding on future enhancements and a possible HCS SC module. The future SC processor has been brought up several times since then and I cringe each time, because each of us has our favorite processor or OS.

This is a discussion that verges on breaking into a religious war every time it's brought up. Basically, there are two camps on this. One side wishes to work with a microcontroller-based SC, and the other wants a PC solution. You have to understand that we have limited resources (i.e., people, time, and money), so we must select work to be done in a frugal manner.

We chose a new Zilog chip because it allowed us to easily port over the RTOS of the SC with little work. It also has low power requirements, is quiet (no need for a fan), low-maintenance (no moving parts), and standalone. But when it was announced that Zilog was reorganizing, we decided to try to port the tasks over to the C language.

The decision made sense because several members of the group already had decided to use the Rabbit processor and had started to port the code over to that base. Instead of allowing that to splinter the group, I thought it wise to take advantage of their work. The side effect is that the code can then be ported over to other OSs, such as those available for PCs (and Macs) or another processor (e.g., ARM controller). The one minor drawback is that we'll only be able to use the COMM-Links and not the stacked I/O boards.

The next issue to address is where we get the boards. We've been able to take advantage of the fact that Micromint still sells many of the boards that the original HCS was designed from. That means that we have support for the SC module (i.e., the RTC180 can be used).

The COMM-Link modules are another matter. Our first solution is to use the prototype boards available from ME Labs. These boards use the Microchip family of processors, which are easy to obtain. They require soldering and adding of parts to the prototyping area of the boards. Considering that our main customers are DIY types, this isn't too much of a problem. But for those of you who are more software types, we've provided links on various electronics topics on our web site. We're also looking at additional boards, so if your favorite processor isn't listed, you should post a suggestion. As I said earlier, comments and suggestions are always welcome.

After the hardware, the question of software comes up. There are several

solutions for the software. Most are commercial products, but we also have open-source solutions.

On the commercial side, we have the C compiler for PICs from CCS. At $99, the price isn't bad. For those of you who don't like C, we also have BASIC. ME Labs has two BASIC compilers, PIC BASIC ($100) and PIC BASIC Pro ($250).

On the open-source side, we have the SDCC C compiler. It supports the 8051, Z80, 14-bit Microchip PICs (e.g., the '16F84, '16F87*x*, and '12CE6*xx* chips), and AVR. I've only had a chance to play with the 8051 and Z80, but all of them look promising. I'll spend some time porting the PL-Link over to SDCC and see what I get. For the AVR, you can also use the GNU C compiler. In addition, there are a large number of freely available assemblers for almost every processor, so I won't try and list them here. But I will talk about the ASL assembler we used to assemble the HCS II RTOS code.

Before Mike Baptiste provided his source code, we disassembled the 4.01 RTOS code (the SC ROM code). We had the 3.60 source but weren't sure what assembler it was for. We did know, however, that it used macros and that those macros made the code extremely easy to read, so we had to save them. After a whole lot of searching, someone found ASL. It supports the macros and the 180 chip. A few tweaks to the assembly code were necessary to assemble the RTOS, and now we have one file for both the 3.6 and 4.0 RTOS.

LESSONS LEARNED

I knew this project would not be a simple one. First, we had to get an open-source license that everyone could agree on. We agreed on the GPL. Then, we needed to get the source code for everything. Now, we have most of the C source and all of the assembly source. Next, we needed to get not just the executables for the host and compile programs, but their source code as well. After that, we had to make sure we could compile and assemble everything into a project that works (we're still working on the old COMM-Link modules).

It was at this stage when we discovered that the host program has dependencies on proprietary libraries that work only under DOS and with the Borland C compiler. A few of us have the libraries but cannot share them. We are, however, able to share the executables. This has been a troublesome issue because we have no replacement other than WinHCS. So, we need a portable host program; my hope is that someone can create something that will fit the bill.

Finally, there's the documentation, which is not up to date yet. And then we need to introduce everything and get the word out that HCS II is not dead. I'm doing my part to spread the news with this article.

NEW ARTICLES

I posed the question to the HCS newsgroup regulars to see if a few of them were interested in writing articles about some of the work they're doing or have done. A few have said they will try. One of the first things I'm hoping to see is the COMM-Links module based on C or BASIC libraries. I also want to see a readily available prototype board and an introduction on how to get a module to communicate with the SC.

In addition, I'm hoping to either provide links to or new articles on some of the basic electronics needed to interface the HCS to the real world. Many of these articles have already been written. A few may need to be revisited so they can be explained in a different way.

STILL LEFT TO DO

We have a long list of projects that need help. The RTOS tasks have to be ported from assembler to C. COMM-Link libraries need to be written and tested. "You're favorite processor goes here" must be tested as a COMM-Link. The SDCC header files need to be written. Documentation needs to be updated.

It's amazing how little spare time we have and how much work has to be done. Don't be surprised when you log onto the newsgroup and no new messages have been posted for days at a time. Sometimes there is nothing new to post. Other times, especially around holidays, we're busy with other things.

THE FUTURE

The future is up to the user base. If we work as a team, we will be able to add new features and maintain present compatibility.

The best news is that no matter what happens, the source code to everything that has been done is available, so it should be possible to create patches and present new features even if the leaders disappear into obscurity.

Neil Cherry is a service development engineer with AT&T Labs. He has been active in computers since 1978. He has an AAS in Electrical Engineering Technologies, has maintained the Linux Home Automation pages since 1995, has been active in home automation with Unix since 1988, and has used the HCS II since 1994. You may reach him at ncherry@comcast.net.

RESOURCES

***Circuit Cellar* newsgroups**
www.circuitcellar.com/products/newsgroups.asp

Free Software Foundation
www.fsf.org

Mike Baptiste's COMM-Links
www.cc-concepts.com/opensource

Open-Source HCS project
hcs.sourceforge.net

SOURCES

C compilers
CCS, Inc.
(262) 797-0455
www.ccsinfo.com/picc.shtml

PIC BASIC, PIC BASIC Pro, prototype boards
MicroEngineering Labs, Inc.
(719) 520-5323
www.melabs.com

RTC180 board
Micromint, Inc.
(407) 262-0066
www.micromint.com

RABBIT™ Semiconductor

NEW

RCM3000 RabbitCore Series

Microprocessor core modules

Ethernet
RCM3010 RabbitCore
$43 qty.1000
$49 qty.100



Jumpstart your design with our compact, powerful core modules as the heart of your custom board. Pin-compatible models let you swap out Ethernet/non-Ethernet versions and change memory capacities. Low-EMI features help eliminate the EMI problems that often derail development schedules.

Features:

- *Fast Rabbit 3000™ processor*
- *Low EMI (typically <10 dB µV/m @ 3 m)*
- *3.3 V (with 5 V tolerant I/O)*
- *6 serial ports*
- *52–54 parallel I/O*
- *128K–512K SRAM*
- *256K–512K Flash*
- *Battery backable real-time clock*
- *Remote cold boot*

RCM3110 RabbitCore
without Ethernet
$29 qty.1000
$35 qty.100



Low-Cost Development Kits provide a complete embedded design system:

RabbitCore module, Dynamic C® SE software (not a trial version) and complete documentation on CD-ROM, prototyping board, serial cable for programming and debugging. Ethernet versions include royalty-free TCP/IP stack with source.



2932 Spafford Street, Davis, CA 95616
Tel 530.757.8400 • Fax 530.757.8402

www.rabbitsemiconductor.com/rcm3/

ROBOTICS CORNER

Chris Cantrell

The OO CarolBot

Don't learn from your mistakes; avoid making them instead. Chris is an advocate of foresight rather than hindsight. In this article, he will show you how to use object-oriented programming techniques to build modules that can be reused in future robot designs.

Most projects begin with a nearsighted list of requirements. There is an immediate need, and you design the hardware and software to fit that need. Object-oriented (OO) methodology adds forethought on top of the present requirements. Yes, there is an immediate need that must be filled, but with a little planning you can design modules that are easy to reuse in future endeavors.

Take the CarolBot, for instance. I had an immediate and desperate need for a robot that roams the halls playing Christmas carols. But with forethought, I created the basic framework for many years of robotics experimentation.

In this article, I'll show you how to use the OO mindset in hardware and software to build pluggable, reusable modules for robotics. I will also discuss the pros and cons of using an OO approach, especially in the embedded world. In addition, you'll learn how to use your Palm Pilot or cell phone as a robot brain to control your robot with Java code.

THE OO MINDSET

Despite what you may have heard, OO is not an alternative to procedural programming; it's a layer of abstraction above it. In procedural programming, you start with low-level functions and build up complexity in layers of functions that call the lower layers. In OO programming, the meat of your code is still procedural functions, but you organize the functions and data into objects. Objects protect your data from direct memory access by hiding the data behind methods. An object's virtual function pointer table allows you to specify a function's signature at compile time and change the implementation of the function at runtime.

The extra abstraction makes it easier to maintain a large body of code, but it also requires additional training and forethought at design time. If your code base remains small and rarely evolves, the OO learning curve might be wasted. The virtual function pointer table makes it easy to add features to a product in the future, but the extra indirection imposes a performance penalty and additional memory overhead. If your product is performance-sensitive, you may not have the extra overhead to apply OO effectively.

OO modules are designed for reuse, so they often contain features (or support for features) that are not immediately practical. If I need a linked list of command objects in my robot code, I might start by creating a general-purpose linked list for reuse in other parts of the code and later products. I would spend a little extra time adding generic list functions that aren't in use. These extra functions are easier to create while I have the innards of the linked list in mind.

Designing for reuse is a gamble. If I never reuse the linked list, then I've wasted valuable development effort on it and the list may not be as fast as it would have been if I had custom fit it to my immediate needs. But if my modules are generic, I can quickly reuse them and recombine them to add new features down the road.

OO is a mindset, not a language. You could write lean and mean procedural code in Java, and you could write flexible, reusable, and object-filled code in C. But OO languages offer automatic support for the additional abstraction layer, so they will save you time if you need it.

OO HARDWARE DESIGN

Discussions concerning OO design usually become aloof philosophical arguments filled with strange terminology. Fortunately, I can demonstrate sound OO concepts in a simple physical design.

The CarolBot has various TTL hardware pieces that require a regulated 5-V power supply. My first thought was to mount a 9-V RC car battery and 7805 voltage regulator directly to the robot base I scavenged from a Wal-Mart toy. There is a cranny in the base that seems made for a battery, and I thought the frame would easily dissipate the heat generated by the regulator.

This myopic design is sleek and efficient. If I were selling a product at the lowest possible cost and size, the design would be optimal. But as an experimenter, I am more interested in creating a generic power supply that could be reused in future robots.

My OO design of the power supply is built into a stand-alone rectangular frame made from balsa strips (see Photo 1). I added a power switch to the frame along with a jack for an external 9-V wall transformer to bypass the batteries. This jack seems wasted on the mobile robot, but it is a good general-purpose power supply feature that will be useful in future stationary projects.



Photo 2—*The CarolBot modules can be swapped in and out with new modules to create new robotics applications. This incarnation of the robot features a faster motor base and the TINI brain.*

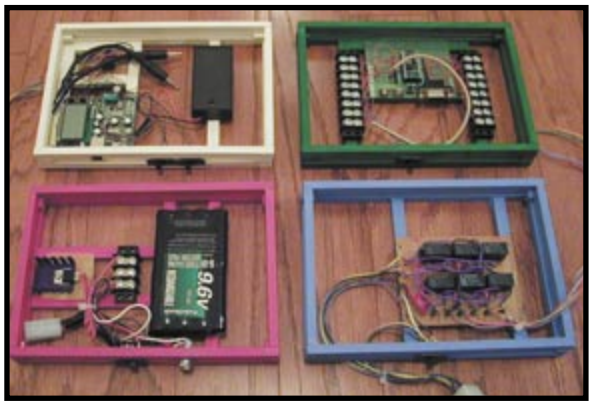


Photo 1—*The CarolBot is built from pluggable modules. The MP3 player module is in the upper left corner; the I/O coprocessor module is in the upper right corner. The power module is in the lower left corner; the motor relay board is in the lower right corner.*

The rectangular frame contains three screw terminals for the other modules to tap into. I am providing access to the 5 V and ground that I immediately need, in addition to an extra screw terminal for the unregulated 9 V. Again, I don't currently need a 9-V supply, but it's easier to add the extra screw terminal now while I'm building the supply than to try to add it later when I need it.

All of these added features of the generic supply seem wasteful in my current application. The extra power jack and screw terminal consume space and add weight. I have to stick in a bulky heatsink for the regulator instead of using the robot base. If I never reuse the power supply in a future project, then the extra time and space is just wasted. On the other hand, if I ever add a module to the robot that needs 9 V, then my power supply needs no additional work.

By providing a standard interface (three screws), I make it easy to swap out the current power supply with another supply that provides the same three-screw interface. The "clients" of the power module depend only on the three-screw interface and not the guts of the power module. Encapsulation and interface are key concepts in the OO methodology.

The CarolBot is a physical incarnation of OO design principles. The robot is a collection of rectangular modules all built to the same size for easy stacking (see Photo 2). The robot is symmetrical and almost too tall with much wasted space in the core. Some modules contain few components but are placed in a uniformly sized container. Without the balsa frame, the robot could be much lighter. Therefore, OO systems (i.e., hardware and software) are fatter and fluffier than they need to be with all of the generic features I hope will prove useful in the future.

I am an experimenter, so flexibility is the most important feature of the robot. Because I'm not tied to any particular robot base, I could easily swap bases to the new tank treads I pulled from a Tonka Toy (see Photo 3). I could swap out the sensor module for better technology tomorrow or replace the MP3 player module with a spitwad shooter module to give the robot a new personality. I could remove the rolling base completely and use the servo module to make a Internet-controlled fish feeder for my aquarium.

HARDWARE MODULES

We have already discussed the CarolBot's power supply module. Photo 1 shows the innards of the power module and three other generic robot modules.

I built a simple relay board to control the motors in the robot platform (see Figure 1). Each motor has two control lines: one for direction (i.e., forward or backwards) and one for on/off. DC motors need a lot of power,



Photo 3—*The CarolBot modules stack neatly on top of one another. This incarnation of the robot features the Tonka tank base and the Palm Pilot brain.*

and the relay board depends on external power from the robot platform. The platforms I used have built-in battery compartments for four C or D cells.

The sensor module uses three Sharp IRRDS sensors to sense the distance of objects to the left, right, and front of the robot. Each sensor returns an 8-bit distance value to the controller through a serial line. The IRRDS sensor costs around $20, but it comes with sample BASIC control code and schematics to interface to a microprocessor (e.g., BASIC Stamp).

For the MP3 module, I gutted an inexpensive MP3 player I bought from Wal-Mart. I carefully soldered two wires to each side of the three buttons I needed. Then, I used a 4066 quad bilateral switch to press the buttons from microprocessor I/O lines. I used a 9-V battery-powered mini-amp from Radio Shack to amplify the music.

I added a Crystalfontz LCD to the top of the robot for season's greetings. The LCD uses a single control line from the I/O processor to accept serial display commands.

CAROLBOT'S BRAIN

When we hear the phrase "embedded programming," we think of optimizing every byte of ROM and RAM for a PIC or some other resource-limited CPU. Moore's Law, however, insists that CPU speeds and memory double every 18 months. Yesterday's workstations are today's embedded processors, and we will find the capabilities of today's PCs in tomorrow's toasters!

Given the choice, I would shrink my 1.6-GHz PC (hard drive and all) down to a tiny chip and use it for my embedded projects. Current physical limitations, however, force me to choose a processing environment less capable. But remember Moore's Law: the same processor tomorrow will be half the size that it is today. You can already see many general-purpose computers making their way into the embedded space.

The TINI board from Dallas Semiconductor is a full-featured Java computer on a tiny 72-pin SIMM card. The board features a built-in Ethernet chip, two serial ports, a file system, and a UNIX-like operating system. You transfer your compiled Java code to the board across the network with FTP and then log in remotely with TELNET to run it from the command prompt. You

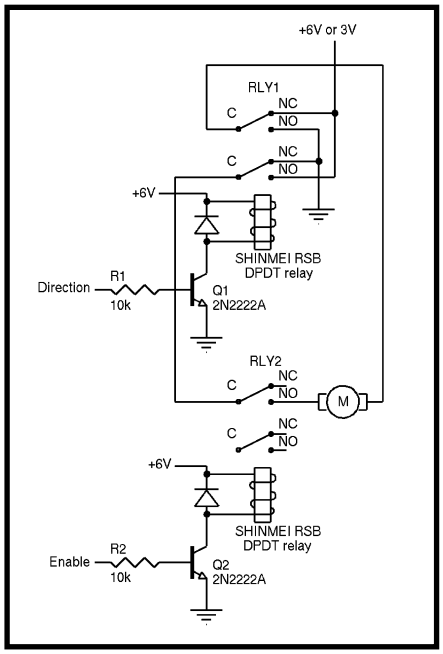
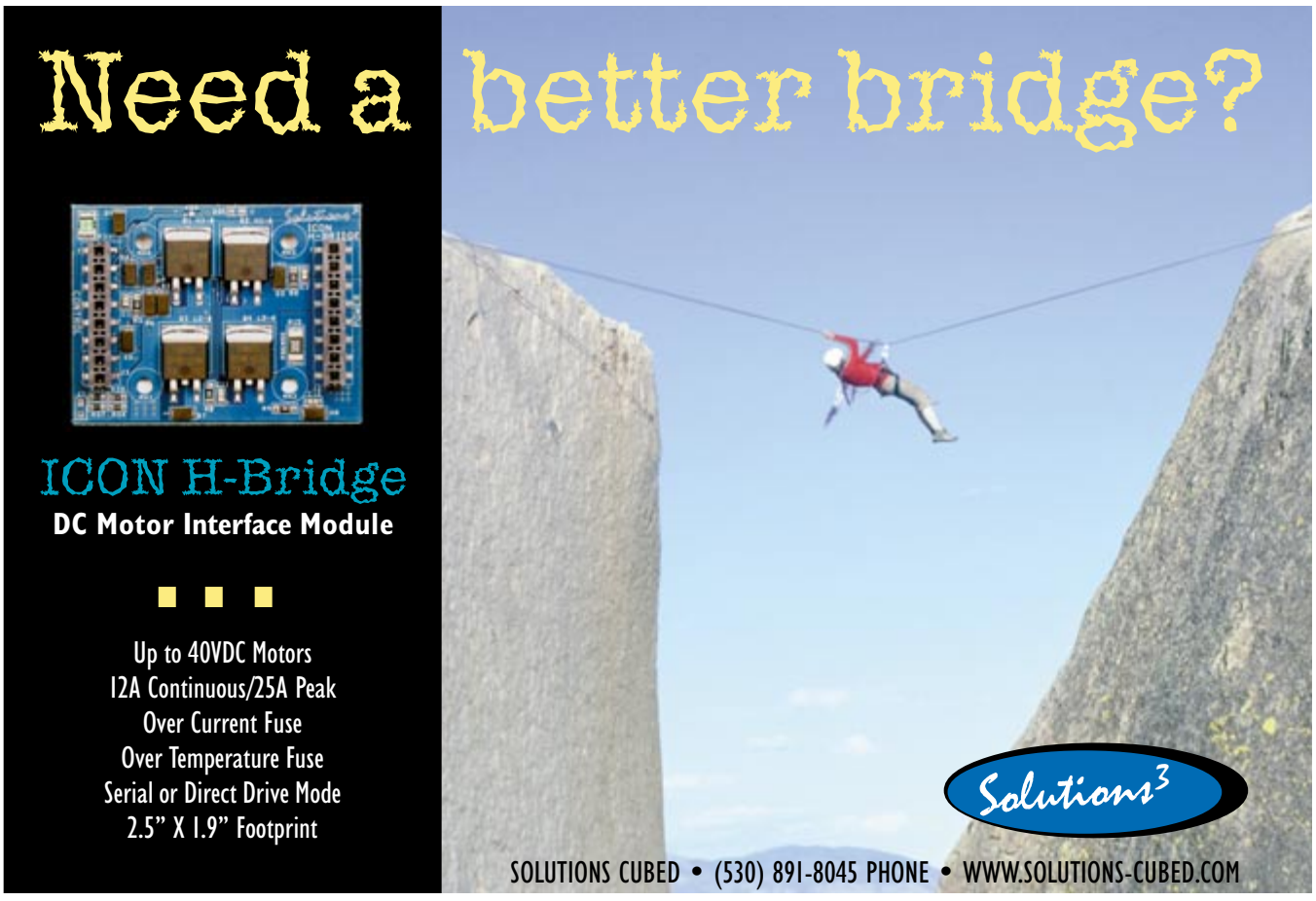


Figure 1—*The robot base controller uses a couple of DPDT relays for each motor providing forward, backward, on, and off control. A Molex connector allows different bases to be plugged into the relay board.*



can even create a web site and surf to the TINI board with your web browser. You get all of this for about $50.

You probably already have a general-purpose embedded brain for your robot right there on your belt; look no further than your Palm Pilot. PDAs continue to grow in capability and shrink in price. Although a general-purpose PDA may not be practical to redistribute within a commercial product, it's perfect for a robot enthusiast. You already use the portable computer to track your appointments and phone numbers, so why not use it to run your robot too?

Cell phones are merging with PDAs. How about using your cell phone as a general-purpose computer to control embedded experiments? With wireless Ethernet on your phone, you could put your mobile robot on the Internet. All you have to do is connect your phone, Palm Pilot, or other PDA to your robot hardware. And how do you do that?

General-purpose computers usually have wonderful high-level features like operating systems and IP stacks, but they have little general-purpose hardware control capability. On the other hand, PICs and other traditional embedded solutions have lots of I/O lines but generally lack the high-level computing features. You can merge the best of both worlds by using both solutions at the same time.

The CarolBot uses a general-purpose I/O processor module that connects to the main computer with a serial cable. Most general-purpose computers have a serial port. I used a BASIC Stamp as my coprocessor simply because I had one on hand. Many PICs cost much less and offer the same kind of I/O support. Still, the BASIC Stamp has built-in features like direct servo control. I'm not taking advantage of these features in the CarolBot, but who knows what the future will bring?

The CarolBot's I/O processor module provides 16 I/O connections through two eight-screw terminals (see Photo 1). I mounted my Stamp development board inside the frame and connected all of the I/O pins to the screw terminals. The brain module sits on the top of the robot and connects to the DB9 serial connector of the BASIC Stamp.

When the control software running on my Palm wants to turn on the robot's wheels, it sends a sequence of text characters through the serial port to the coprocessor. When my robot algorithm wants to read the sensors, it sends a command to the BASIC Stamp

Listing 1—*Every robot behavior is coded in an object that implements the executable interface. Sequences of behavior objects are stored in lists managed by the ExecutableList object. The ExecutableList also implements the executable interface, allowing sub-lists to be stored as single list entries.*

```
import java.util.*;

public interface Executable {
      public void execute() throws Exception;
}
public class ExecutableList implements Executable  {
      private List commandList;

      public ExecutableList() {
              commandList = new LinkedList();
      }
      public List getCommandList() {return commandList;}

      public void setCommandList(List commandList) {
              this.commandList = commandList;
      }
      public void execute() throws Exception {
              Iterator i = commandList.iterator();
              while(i.hasNext()) {
                      Executable e = (Executable)i.next();
                      e.execute();
              }
      }
}
```


Intuitive Circuits, LLC
(248) 524-1918
http://www.icircuits.com



OSD-232 is an on-screen display character overlay board. From any RS-232 serial source like a PC, PIC, or Basic Stamp, display 28 columns by 11 rows of information (308 characters total) directly onto any NTSC or optional PAL baseband (video in) television or VCR. OSD-232 can overlay monochrome text onto an incoming video source or display colored text on a self-generated colored background screen.

OSD-232 $99.00

PIC-SERVO
Motion Control
Low-Cost Motion Controllers for Brush, Brushless and Stepper Motors

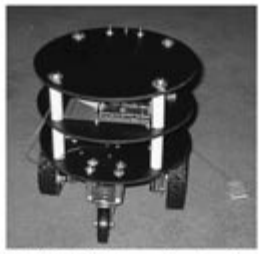


- On-board power amplifiers
- RS232 / RS485 Interface - Up to 32 axes per port
- Linear and circular interpolation
- Software support for Windows, embedded controllers, and other platforms
- Excellent on-line customer support
- Used in Electronics, Biotechnology, Material Handling, CNC and Robotics applications around the world

Our ***PIC-SERVO***, ***PIC-STEP***, and ***PIC-I/O*** chip sets are also available for use in custom designs

J R KERR AUTOMATION ENGINEERING
www.jrkerr.com

Mobile Robotics
Used world wide for research!



Mobile Robots Controllers
Artificial Intelligence
Sonar Vision Systems Optics
Zagros Robotics
PO Box 460342
St. Louis, MO 63146-7342
Phone (314)768-1328 Fax(314)576-5568
http://www.zagrosrobotics.com

and then reads the serial result back. You can use a terminal program like HyperTerminal to talk to the coprocessor manually (great for debugging).

You can develop your robot code on the PC using its serial port to run the robot. Then you can compile your code for your Palm Pilot and use the hotsync cradle's serial cable to run the robot. If you're writing Java code, you can run the (nearly) same piece of code on any general-purpose computing platform. That's the ultimate in code reuse.

Speaking of programming languages, there are several factors to consider when choosing a language for your embedded project. If performance is crucial, you might need raw assembler. If future enhancement is crucial (as to a robot enthusiast), you might consider the built-in reuse features of an OO language like C++ or Java.

Java is particularly easy to learn and tinker with. You don't have to worry about complex compiler options or make files. That leaves you more time to spend on the business logic at hand. But the best language to use is the one you are most comfortable with. You can write flexible, reusable code in any language, but you'll have more fun and be more successful in home projects if you are speaking your native tongue.

I chose Java for the CarolBot primarily because I love it. So, I used embedded CPUs that support Java. The TINI board has a JVM built into the operating system. For my Palm V, I downloaded the free WABA Java package that runs on a variety of embedded platforms. Note that Sun's Java Micro Edition lacks support for the serial port.

CAROLBOT'S SOFTWARE

When I first sat down to write the CarolBot control software, I created several global methods to perform various hardware functions. The setMotors() function accepts two integers that represent the left and right wheels with values for each: off, backwards, or forwards. The pressButton() function accepts a single integer representing a button to press on the MP3 player: On/Play, Off/Stop, and Next. The getSensor() function accepts a sensor number and returns the 8-bit distance value read from the specified Sharp IRRD sensor. Each of these functions sends the appropriate serial data to the coprocessor and reads any return values.

Next, I organized the methods into separate objects. All of the robot-base methods went in the Robot object, the sensor methods went in the Sensor object, and the MP3 methods went in an object named MP3. Why the extra organization? It seems wasted for such a simple design, but it adds flexibility.

If I swap out physical MP3 modules in the future, I'll need to change the code in the MP3 functions. If I keep the MP3 functions together and alone in a single class file, I can simply create a new MP3 class and use it in my future code. I don't have to worry about recompiling the other code (like the sensor class). And I can create a new SuperMP3 class that extends the older MP3 class. If the new MP3 class provides the same methods as the old class, then I can just swap out the MP3 class without changing the other code that uses the methods. That's a powerful design goal in OO programming.

Next, I needed higher-level algorithms to track the walls of my office building. The walls are nice and straight, which made the algorithms fairly easy. I started with an algorithm that tracks a wall on either side of the robot by turning the motors on and off to keep the wall between a minimum and maximum sensor reading.

Then, I created other control algorithms, such as methods to track around an open corner (where the wall vanishes) or a closed corner (where the robot finds a wall in front of it). These algorithms are ultimately what robot programming is about, and here is where I wanted the most flexibility. I need to be able to add additional routines in the future with minimal changes to the rest of the code.

I chose an OO design pattern called Command Pattern to abstract the algorithms from the controller code that uses them (see Listing 1). I created sep-

Listing 2—*The TrackDoorToWall command object moves the robot across a doorway to pick up tracking the wall on the other side. The algorithm constants are accessed from template methods making it easy to override the constants in a new derived class.*

```
public class TrackDoorToWallCommand implements Executable {

      private SensorDriverInterface sensors;
      private MotorDriverInterface motors;
      private int wallID;
      private int wallSensor;

      protected int getDriftTime() {return 10;}
      protected int getWallThreshold() {return 80;}

      public TrackDoorToWallCommand(int insideMotorID,
             MotorDriverInterface motors,
             SensorDriverInterface sensors){
             this.wallID = insideMotorID;
             this.motors = motors;
             this.sensors = sensors;
             if(wallID == MotorDriverInterface.LEFT) {
                    wallSensor = SensorDriverInterface.LEFT;
             } else {
                    wallSensor = SensorDriverInterface.RIGHT;
             }
      }

      public void execute() throws Exception {
             motors.forward();
             while(true) {
                      Thread.sleep(getDriftTime()); int s =
                       sensors.getSensorValue(wallSensor);
                    if(s>getWallThreshold()) return;
             }
      }
}
```

arate classes for each algorithm, giving each class a single method named execute(). The execute method of each class performs all of the steps for a single robot function (e.g., tracking a wall down to a doorway). Listing 2 shows a simple robot command function.

To create a sequence of commands for my robot, I generate a list of command objects in the order I want them executed. A typical sequence for my office might contain a list of nine objects like this: TrackWallTo-Door, PlaySong, TrackWallToDoor, PlaySong, TrackWallToDoor, PlaySong, TurnLeft, TrackWallToDoor, PlaySong. This sequence would traverse three doorways, stopping briefly to play music at each door, make a left turn around a corner, and go over to another door.

The central controller simply reads a list of command objects, takes each object in turn, and calls the execute() method. When the control method returns (when that robot function is complete), the next object is executed. Listing 3 shows the code for a simple command list. New and exotic control objects can be added without changing the central controller.

In fact, I created a small GUI to select command objects from a pull-down list. I can choose from my favorite sequences that match my hall, or I can build new sequences for other halls on the fly. Suddenly, that GUI screen on the PDA is useful! At first I had thought it to be wasted space, but more often than not in OO designs, if you build it they will come.

My favorite command object for the CarolBot is the RemoteControl object, which displays a GUI touch screen on my Palm. Using a long serial cable, I can walk behind the robot and drive it with the GUI remote control.

TOMORROW'S PLATFORM

The CarolBot was just the beginning for these reusable hardware and software objects. Before I went on vacation, I created a new hardware module, the Servo Control module, which controls a couple of external hobby servos (remember, the BASIC Stamp interfaces directly to hobby servos). I hooked up the power module (with external wall power), servo module, coprocessor module, and TINI board brain. Then, I plugged the TINI brain into my Ethernet hub and presto! I can feed my fish from the Internet.

My immediate project was a mobile MP3 player that tracks walls. I could have packed all of the hardware onto the toy robot base, used a super-tiny PIC as the embedded processor, and gotten by with a few control functions written efficiently in assembler. But, my pride and joy CarolBot would now be a dusty trophy sitting on the bookcase in my office. By applying just a little forethought and OO design in the hardware and software, I created a flexible, reusable platform for years of experiments to come.

Chris Cantrell is a software engineer at ADTRAN in Alabama. He received his MSEE from the University of Alabama. Java is his passion, and he teaches Java courses for the Continuing Education department at the University of Alabama, Huntsville. You may reach him at ccantrell@knology.net.

Listing 3—*The start-up code organizes robot behavior objects into lists and sub-lists. Varying the types and order of behavior objects in the master list can easily change robot behavior.*

```
//Object to track wall on left to next door
Executable toOffice = new TrackWallToDoorCommand(
      MotorDriverInterface.LEFT, motors,sensors);

//Object to play next song for 60 seconds
Executable doPlay = new PlaySongCommand(0,60,true,mp3);

//Object to track door on left to next wall
Executable toWall = new TrackDoorToWallCommand(
      MotorDriverInterface.LEFT, motors, sensors);

//Build the sub-list of things to do for one office
List list = new ArrayList();
list.add(toOffice);
list.add(doPlay);
list.add(toWall);
ExecutableList subList = new ExecutableList();
subList.setCommandList(list);

//Build a list that executes the sub-list 10 times for
   ten separate offices along a left wall.
list = new ArrayList();
for(int x=0;x<10;++x) {
      list.add(subList);
}
ExecutableList masterList = new ExecutableList();
masterList.setCommandList(list);

//Run the master list
masterList.execute();
```

SOFTWARE

To download the code, go to ftp.circuitcellar.com/pub/Circuit_Cellar/2002/145/.

SOURCES

LCD
Crystalfontz America, Inc.
(509) 291-3514
www.crystalfontz.com

Java TINI board
Dallas Semiconductor Corp.
(972) 371-4000
www.dalsemi.com

BASIC Stamp
Parallax, Inc.
(916) 624-8003
www.parallax.com

J2ME
Sun Microsystems
(800) 555-9786
java.sun.com/j2me/index.html

The Waba Java environment
Wabasoft Corp.
www.wabasoft.com

FEATURE ARTICLE

Noel Rios

The AT89C51/52 Flash Memory Programmers

When faced with a plethora of applications to design, it's essential to have a versatile microcontroller in hand. Noel knows just what you're looking for: the AT89C51/52 microcontrollers. To get you started, he'll describe his inexpensive microcontroller programmer.

The AT89C51 and AT89C52 flash memory microcontrollers are versatile devices that can be used in a variety of applications. Unlike standard 87C51 UV EPROM microcontrollers in expensive windowed ceramic packages, they do not need to be erased with ultraviolet light before reuse. They can tolerate at least 1000 write/erase cycles, so you don't need to throw the device away if you want to modify the code.

How do you program an AT89C51/AT89C52 microcontroller? With a device programmer, of course. However, commercial device programmers are a bit expensive and out of reach for many people. The microcontroller programmer presented here addresses this problem; you don't have to spend a fortune to build this device. To program a flash memory microcontroller, all you need is this unit and a computer with a standard RS-232 C serial port and a terminal program such as Procomm, Telix, or Windows HyperTerminal.

The programmer features a menu that is similar to some commercial device programmers. It has the capability to check for a blank device, erase a loaded microcontroller, program the device, verify the device against the buffer, and load the buffer from Intel hex object code. It also determines which of the two devices, the AT89C51 or AT89C52, is going to be programmed by calculating the size of the file you loaded into the device programmer.

To program this family of chips, you need to drive the V_{PP} pin to a voltage higher than the standard supply voltage. The programmer features a DC/DC converter that provides this super voltage, so the only external power needed is 6 V. It also features an efficient voltage switch so the voltage of the DC/DC converter is fixed and there is no need for calibration. Finally, the microcontroller used to control the programmer itself is in-system programmable, so you don't need access to another device programmer to build this one.

WHY USE A DS5000?

A DS5000 controls the flash memory microcontroller programmer (see Figure 1). It controls V_{PP}, the timing, and provides the necessary signals to load the hex file. Some other device programmers are controlled by a PC, which makes their timing dependent on the speed of the PC. In fact, some of these programmers won't work when used with a Pentium-class PC. The DS5000 also replaces several ICs used in some flash memory microcontroller programmers.

The DS5000 belongs to the secure line of microcontrollers from Dallas Semiconductor. The module contains an 8051-compatible microcontroller, SRAM chip (8 or 32 KB), and a lithium power source. The Dallas microcontroller uses nonvolatile RAM for program and data storage instead of ROM. Because it uses NV RAM instead of flash memory or ROM, it can be reprogrammed many times with no need for erase cycles.

Features of the DS5000 include 100% code compatible with the 8051, nonvolatile memory control circuitry, and a 10-year data retention in the absence of power. The DS5000 directly accesses 64 KB of program and 64 KB of external data memory. In addition, the microcontroller features in-system reprogrammabilty via the serial port, 128 bytes of fast access scratchpad RAM, two general-purpose 16-bit timers/counters, one UART, and five interrupts

Partition (MCON.7-4)	Range (MCON.3)	Program memory	Data memory
0000	0 (8 KB)	0 KB	8 KB (0–1FFF)
0001	0 (8 KB)	4 KB (0–0FFF)	4 KB (1000–1FFF)
0010	0 (8 KB)	8 KB (0–1FFF)	0 KB
0000	1 (32 KB)	0 KB	32 KB (0–7FFF)
0001	1 (32 KB)	4 KB (1–0FFF)	28 KB (1000–7FFF)
0010	1 (32 KB)	8 KB (1–1FFF)	24 KB (2000–7FFF)
0011	1 (32 KB)	12 KB (0–2FFF)	20 KB (3000–7FFF)
0100	1 (32 KB)	16 KB (0–3FFF)	16 KB (4000–7FFF)
0101	1 (32 KB)	20 KB (0–4FFF)	12 KB (5000–7FFF)
0110	1 (32 KB)	24 KB (0–5FFF)	8 KB (6000–7FFF)
0111	1 (32 KB)	28 KB (0–6FFF)	4 KB (7000–7FFF)
0111	1 (32 KB)	32 KB (0–7FFF)	0 KB

Table 1—*The MCON register controls the mapping of the DS5000 NV SRAM to the program and external data address spaces of the CPU.*

with two external sources. Other features include a dedicated memory bus to preserve the four I/O ports, power-fail reset, early warning power failure interrupt, and a watchdog timer.

The on-board SRAM is connected to the CPU via an internal non-multiplexed byte-wide bus that does not interfere with the four standard I/O ports. This bus is separate from the usual external bus formed using ports P0 and P2. The 8051 architecture includes separate memory spaces for code and external data, and the SRAM of the DS5000 can be partitioned so that it can be used in either or both spaces. A special nonvolatile register, MCON, controls this partitioning, and must be set to the correct range and partition addresses before the program is loaded. Table 1 shows the valid combinations of values. The range is the size of the SRAM connected to the byte-wide bus. The partition is the division between program memory and data memory space, which is user-defined. Care must be taken that the partition is not greater than the range. If the MCON register is configured incorrectly, the device may operate incorrectly or cease to function.

The chip select of the internal SRAM is decoded automatically based on the range and partition, which is configured by you. The areas used for program memory space are write-protected and treated as ROM. Data areas are also protected when the supplies are out of tolerance. Depending on the ambient temperature, the internal SRAM is protected for at least 10 years.

The DS5000 is in-system programmable, so you don't need a device programmer to program it. You can upload new code at any time using the serial port, as long as the RESET pin is pulled high, the PSEN pin is pulled low, and P2.6 and P2.7 are floating or pulled high. You must also have a crystal of the proper frequency connected so the built-in ROM-based program downloader can configure the device for the proper data rate. Using this feature, you can program the DS5000 in the field whenever the firmware needs to change.

The DS50000 also boasts features for high reliability operation without supervision, designed for remote applications. It goes into reset when the power supply is out of tolerance. The early warning power fail interrupt enables you to save critical data. And the watchdog helps if the DS5000 goes out of control. You can also save the operating state in nonvolatile memory so you can resume the task when power is restored.

CIRCUIT DESCRIPTION

A single DS5000 microcontroller (U1) controls the AT89C51/52 flash memory microcontroller programmer (see Figure 2). It controls the timing of the pulses to the target microcontroller, manages the loading of the hex file to the SRAM buffer, configures the control signals for the various programming modes, and controls the high voltage supply. It can also check whether or not the flash memory microcontroller is blank, erase the flash memory, program

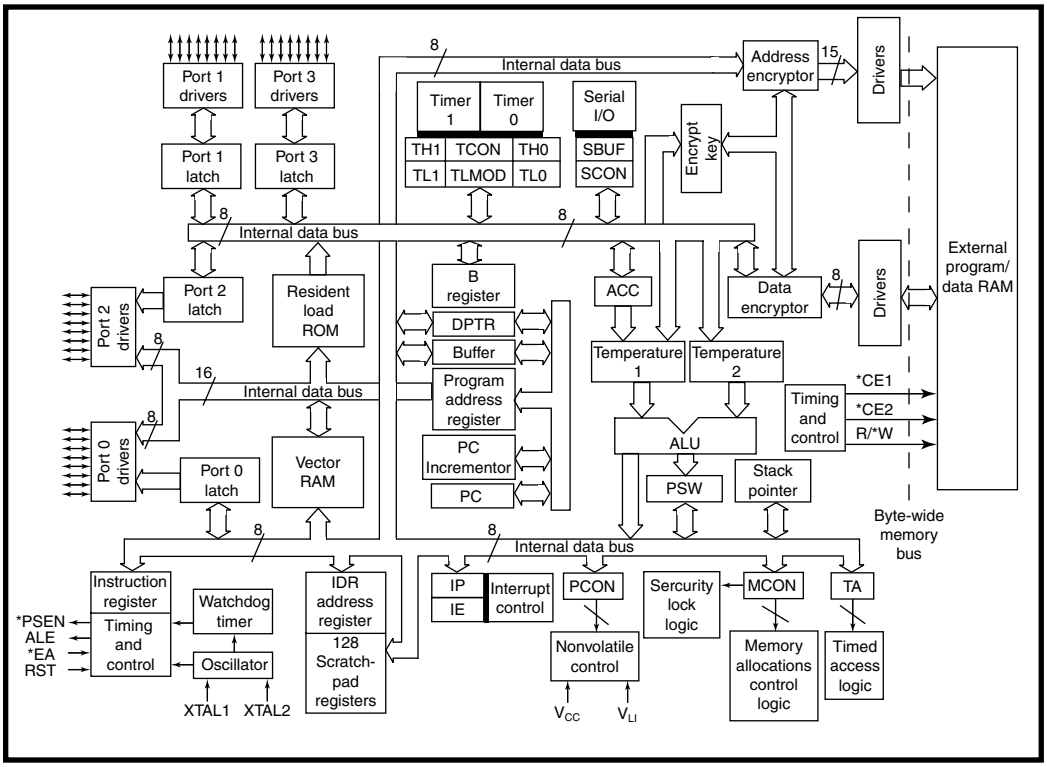


Figure 1—*The DS5000 is a versatile chip that provides most of the functions of many 8051-based single board computers.*

the microcontroller, and, finally, verify that the programming was successful. Note that the DS5000 does not use the expanded Bus mode of ports P0 and P2; a pseudo address bus and data bus are created using the I/O ports instead.

To enter the proper programming mode, four I/O lines are used to control the target's mode control lines. Table 2 shows the proper levels of these signal lines for programming the flash memory microcontroller. You can see that the RESET and PSEN pins are the same for all three programming modes; they are tied directly to V_{CC} and ground, respectively. The flash memory microcontroller code memory array is programmed byte by byte. It is normally shipped with the on-chip flash memory array in the erased state (i.e., the contents = 0FFh, ready to be programmed). To program any non-blank byte in the on-chip flash memory, the entire memory must be erased using Chip Erase mode.

Programming the flash memory microcontroller requires the following five steps. First, you have to output

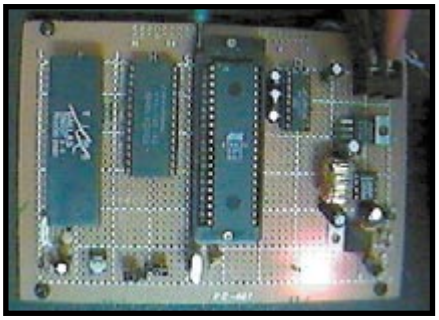


Photo 1—*I constructed the programmer using point-to-point wiring on a standard prototype board. I used a ZIF socket for U3 for easy insertion and extraction.*

the desired memory location on the address lines. Second, output the appropriate data byte on the data lines. Third, activate the correct combination of the control signals. Fourth, raise *EA/V_{PP} to 12 V. And fifth, pulse ALE/PROG once to program a byte in the flash memory array.

The byte write cycle is self-timed and typically takes no more than 1.5 ms. Steps one through five are repeated, changing the address and data for the entire array or until the end of the object file is reached. The RDY/*BSY output signal monitors the progress of the byte programming for its success. P3.4 is pulled low after ALE goes high during programming to indicate that it's busy. P3.4 is pulled high during programming to indicate that it's ready. The entire flash memory array is erased electrically by using the proper combination of control signals and holding ALE/*PROG low for 10 ms.

U2 is an 8K × 8 SRAM that stores the program code. Notice the absence of a transparent latch or 74HC373. This is because the extended Bus mode of P0 and P2 isn't used to access the SRAM. Because of this, the software controls writing and reading data to the SRAM. I/O lines of the DS5000 control the output enable and the write strobe of the SRAM. Note that the SRAM is always chip-selected because of the lack of an I/O line to control it. If you can get a DS5000 with 32 KB of on-board SRAM, this external memory is not required, but you'll need to modify the firmware to work with the on-board memory.

U4 (MAX232) is a level shifter that converts TTL signals to RS-232 signals and vice versa. It has a built-in

Tools to Move Data

RS-232 <> Ethernet <> Web Server

TCP/IP Solutions



µFlashTCP-EP*
10Base-T Ethernet
2 Serial Ports
7-34 VDC
Optional I/O Interface
Rugged Enclosure w/
Industry Standard
Connectors
Starts at $229

LogicFlex*
10Base-T Ethernet
2 Serial Ports
46 Digital I/O's
Programmable Xilinx CPLD
Hardware Clock/Calendar
Expansion Bus for
Peripheral Boards
Starts at $189

Dual-E*
2 Ethernet Ports
2 Serial Ports
3 Counter/Timers
5 Digital I/O's
Onboard Connectors
USNET TCP/IP
Software in Kits
Starts at $199

LogicFlex-EPX*
10Base-T Ethernet
2 Serial Ports
16 Opto-Isolated inputs
16 Relay Outputs @ 500mA ea.
Quick-Disconnect Connectors
Rack Mount Enclosure
LCD & 6 Pushbuttons
Starts at $499

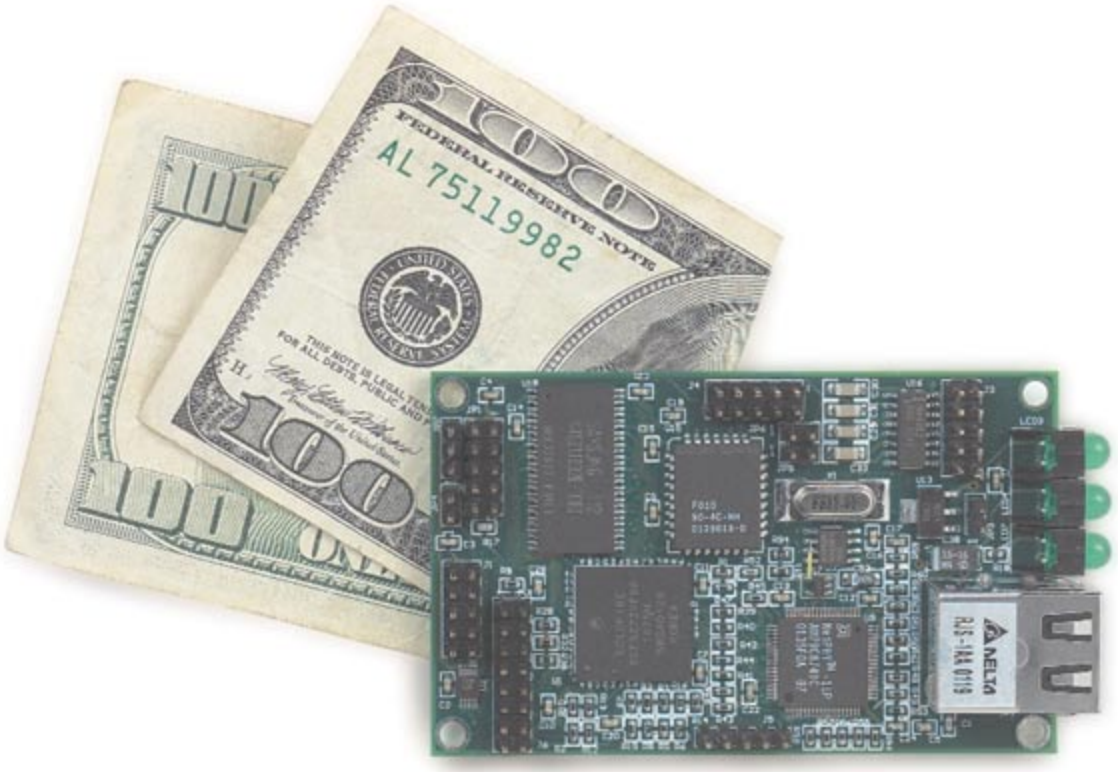
***All products based on the Intel 386Ex with DOS and NE2000 compliant, 10Base-T Ethernet. Standard memory includes 512K RAM & 512K Flash plus DIP socket to accept an M-Systems DiskOnChip. Development systems contain necessary hardware and software tools for fast development.**

JK microsystems connects you with embedded control solutions. Our cost-effective DOS based controllers are ideal for data acquisition, networking or industrial technology applications.

JK microsystems, Inc.
1403 Fifth St., Suite D
Davis, CA 95616 USA
Visit us on the web www.jkmicro.com
Call 530-297-6073 Fax 530-297-6074

Network Enable Any Serial Device

10/100 Ethernet for $99*



www.netburner.com

NetBurner

Networking in 1 Day!

THE FASTEST WAY TO ENABLE YOUR NETWORK DEVICE

*Quantity 100

voltage doubler and inverter so that there is no need for external 12- and –12-V power supplies. U4 interfaces the device programmer to the PC so that you can transfer the hex file to the device programmer.

U6 (MAX632) together with L1, C1, and C2 form a DC/DC converter that provides the super voltage needed for programming the flash memory microcontroller. A boost converter converts the 5-V supply to 12 V. It's a fly-back converter, which means that U4 periodically shorts L1 to ground so that energy is stored in its magnetic field. Then, U4 opens the switch according to its internal pulse width modulator so that the fly-back voltage plus V_{CC} can charge the output capacitor (C1) to 12 V.

C1 filters the voltage coming out of U6. Be sure that C1 is connected to U6 when power is applied, because U6 will be damaged if C1 is left unconnected. C4 (100 µF) is an input supply decoupling capacitor. This is needed because short spikes of current are drawn when U6 charges L1 from V_{CC}.

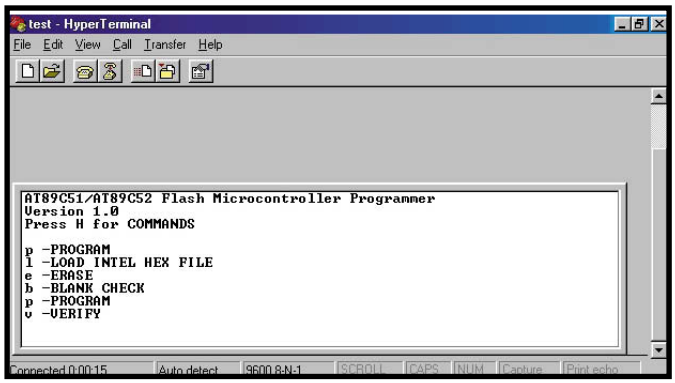
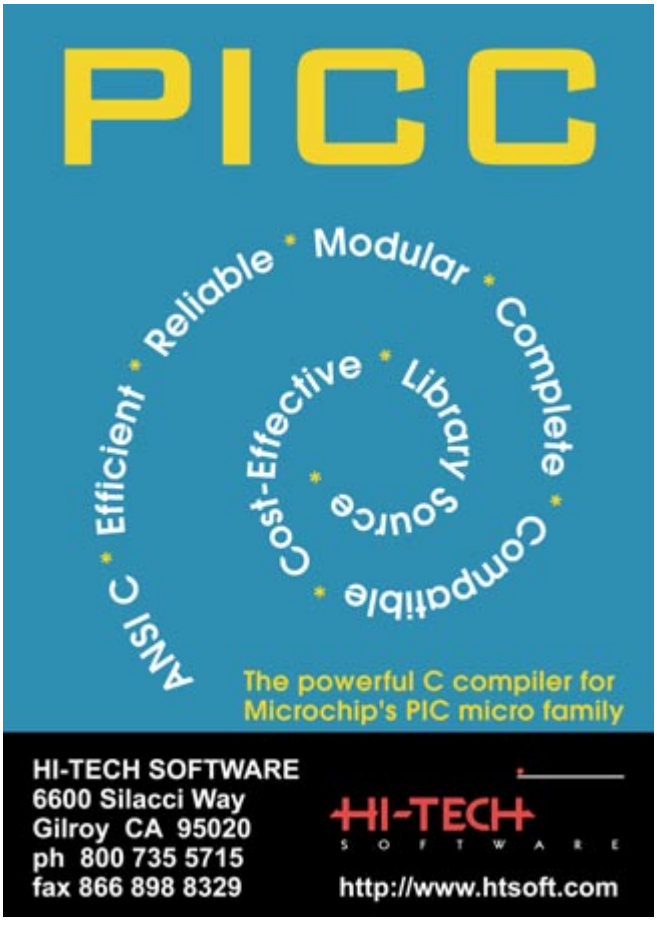


Photo 2—*The Help command of the flash memory microcontroller programmer shows all of the available commands.*

Q3 (IRF9530), Q2, Q4, R2, R8, and R9 form a voltage switch to connect the 12 V to the flash memory microcontroller at the right time. Two transistors are used to control Q3, a P-channel MOSFET, because the I/O pin that controls the switch is high following a reset of the DS5000. The transistors form a double inverter so that the voltage switch will be off when the pin is high. A P-channel MOSFET is used with a low RDS on so that the voltage drop across the switch is negligible. C3 absorbs transient voltage entering from the switch. Without it, the transient voltage may damage the flash memory microcontroller. D2 is a silicon rectifier diode that provides 5 V to the EA/V_{PP} line to create the logic high needed for verifying the flash memory microcontroller.

U7 is a low-dropout linear voltage regulator. Instead of using a 7805, I used an LDO regulator for greater efficiency in terms of power consumption. C5 is the filter capacitor needed by U7 to function properly. C11, D3, and R5 form the reset circuit needed by U1. D3 discharges C11 when the power is disconnected so that U1 resets properly when power is connected again immediately. A SIP resistor pack is connected to the data bus as indicated in the datasheet for programming the flash memory microcontrollers.





A 16-MHz crystal, Y2, clocks the target processor. I chose a high crystal frequency for Y2 because the data and address setup times depend on the crystal frequency. The clock must be running for the flash memory microcontroller, because the device is executing internal address and program data transfers during the programming.

A 7.3728-MHz crystal is used with U1 so that 9600-bps communication can be established by configuring the built-in UART inside U1. C12 is a load capacitor, which provides a small phase shift combined with the output resistance of the internal oscillator's amplifier. It also provides some attenuation of overtones. Capacitor C13 is also a load capacitor. When combined with the crystal resistance, C13 provides additional phase shift.

D4 is an LED, and R7 limits the current flowing through D4. D4 is not only a power-on indicator, it also indicates whether or not the flash memory microcontroller inserted in the socket is severely defective (because a defective flash memory microcontroller usually draws a large amount of current). If the microcontroller pulls a large amount of current, V_{CC} will be pulled down and the LED will dim or not light at all.

FIRMWARE OVERVIEW

You can download the firmware from the *Circuit Cellar* ftp site. First, the hex is stripped of other characters to obtain the object code. I found out that it's necessary to fill the SRAM with 0FFh so that it will match the original object file located in the computer or PC. This is because after the hex file is stripped of other characters, it won't fill the SRAM sequentially but will depend on the address located inside the hex file.

Note that the communication has no hardware handshaking. This is OK because there's a checksum embedded inside the hex file, so you can know if the transmission of data is successful from the PC to the device programmer.

CONSTRUCT AND ASSEMBLE

The prototype was constructed using point-to-point wiring, as shown in Photo 1. A prototyping PCB was used with wire wrap wire. Use different colors of wire for different functions so you can trace each connection from one point to another. I used ordinary IC sockets (with solder connections), but if you prefer, you can use wire wrap sockets. Arrange the sockets so that the ICs that share many

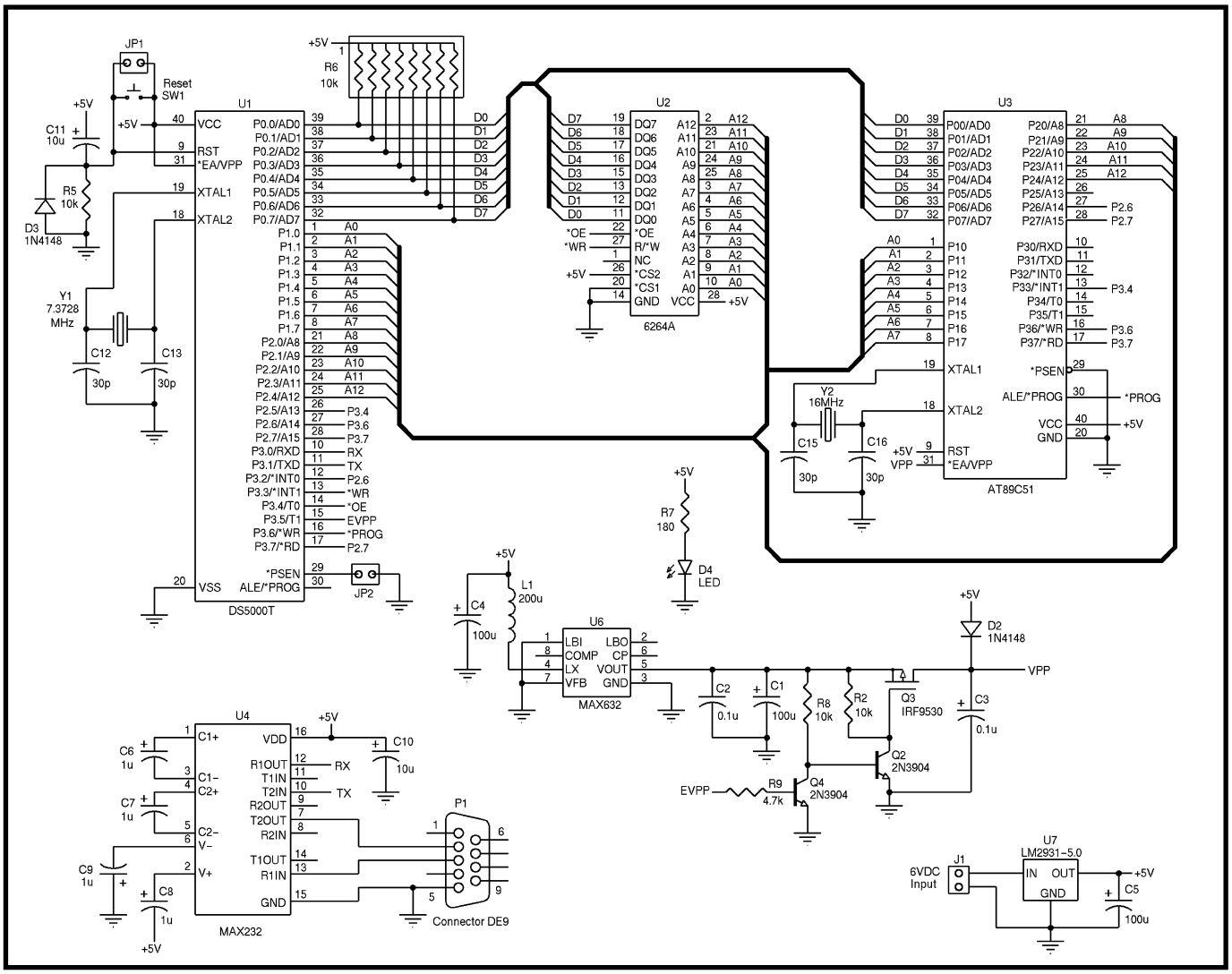


Figure 2—*The programmer incorporates advanced features such as a DC/DC converter to provide the high voltage needed for programming.*

Mode	RST	*PSEN	ALE/PROG	*EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code	H	L	Pulse	12 V	L	H	H	H
Read Code	H	L	H	H	L	L	H	H
Chip Erase	H	L	Pulse	12 V	H	L	L	L

Table 2—*A relatively small set of signals controls the programming modes of the Atmel flash memory microcontrollers.*

connections are near each other. For example, place U2 (6264) close to U1 (DS5000). Place the crystals and load the capacitors (e.g., C12 and C13) close to their corresponding microcontrollers. I recommend using a ZIF socket for U3, the target microcontroller.

After wiring the circuit, check the connections between the different ICs according to the schematic (see Figure 2) using a multimeter. This will save you a lot of time finding a wrong connection through troubleshooting. Before inserting the ICs, check for the presence of 5 V on the V_{CC} pin of each IC. Also verify that the GND pin of each IC is connected to ground on the power supply.

You must also load the provided firmware into the microcontroller U1. To get the firmware into the DS5000, install the two shorting jumpers (JP1 and JP2) and then connect 6 V to J1. Configure your terminal program for 9600 bps, 8 bits, 1 stop bit, and no parity. Press Enter on your keyboard, and the Dallas Semiconductor downloader banner will appear on the screen.

Next, press the letter "L" on your keyboard, followed by Enter. Then, use your terminal program to send the ASCII text file PRGM1.HEX. You will see no error messages if the loading is successful. After that, disconnect the 6 V and remove the jumpers. Reconnect the 6 V and press the Reset button. While still connected to the terminal program, you should now see the banner from the programmer firmware telling you the device is working.

CHECK OUT AND USE

Remove the DS5000, and check for the presence of the super voltage, which is 12 V on pin 31 of the U3 socket. This voltage is important in order to program the microcontroller. Then, disconnect the power and plug in the DS5000 and 6264. After the DS5000 is plugged in, check for the presence of 5 V on pin 31 of the U3 socket. This voltage is necessary for verifying the microcontroller.

Open your terminal application and configure it to 9600 bps, 8 bit, 1 stop bit, and no parity. Also check the COM port where the device is connected. Connect the power to the device. You will see a banner each time the power is connected or when the Reset switch is pressed. Press "h" to see the menu (see Photo 2). To check if the device is blank, press "b". Press "e" to erase the device, and press "p" to program it. To load the hex file, press "l", and be sure to send the text file in your terminal program as an ASCII text file. You'll also want to verify the flash memory microcontroller by pressing "v". Always remember to erase the flash memory before reprogramming the microcontroller.

TROUBLESHOOTING GUIDE

There is a handful of ways the programmer might fail. For one, the ICs might not have power (i.e., no 5 V across the ICs). Another possible problem is a wrong or missing connection of the control signals between the two micros (P2.6, P2.7, P3.6, and P3.7). Or perhaps you won't get V_{PP} or programming voltage. If this happens, check the wiring around U6, the MAX632.

You don't want the V_{PP} (pin 31) at a high potential all the time. Make sure you check that Q3 is correctly wired. Don't forget to also check Q2 and Q4. Incorrect wiring of the data bus or address bus can cause more problems. Using a logic probe, check if controller U1 is active by monitoring the pins of U1 (ALE, in particular). If it isn't functioning, check the reset circuit (C1 and D3), and check the connections of the crystal (Y1). Also verify that pin 31 of U1 (*EA) is wired to V_{CC}.

If no banner appears in your terminal program, check that RX and TX are connected to the MAX232 and that the connections between the MAX232 and DE9 connector TD and RD are not exchanged. In addition, check for the presence of a signal on pin 10 of U4 (MAX232) after pressing the Reset switch. That covers the troubleshooting you might have to face.

I have tested the programmer with the Atmel AT89C51 and AT89C52, and the results were good. The circuit can be modified to accommodate flash memory microcontrollers from different manufacturers, as well. However, it is important to note that the Atmel devices contain embedded algorithms that somewhat simplify the firmware of the programmer. If you're going to modify the circuit, you'll also need to modify the firmware.

Noel A. Rios is an electronics and communications engineer. He has worked with semiconductor and electronics companies like Microcircuits, IMI, Allegro, and ASTEC. His interests include computers, embedded control, power conversion, testing and measurement, and GPIB control. You may contact him at nar@edsamail.com.ph.

SOFTWARE

To download the firmware, go to ftp.circuitcellar.com/pub/Circuit_Cellar/2002/145/.

RESOURCE

Dallas Semiconductor, Inc., "Secure Microcontroller User's Guide," 051002.

SOURCES

AT89C51/52 Microcontrollers
Atmel Corp.
(408) 436-4270
www.atmel.com

DS5000 Chip
Dallas Semiconductor, Inc.
(972) 371-4000
www.dalsemi.com

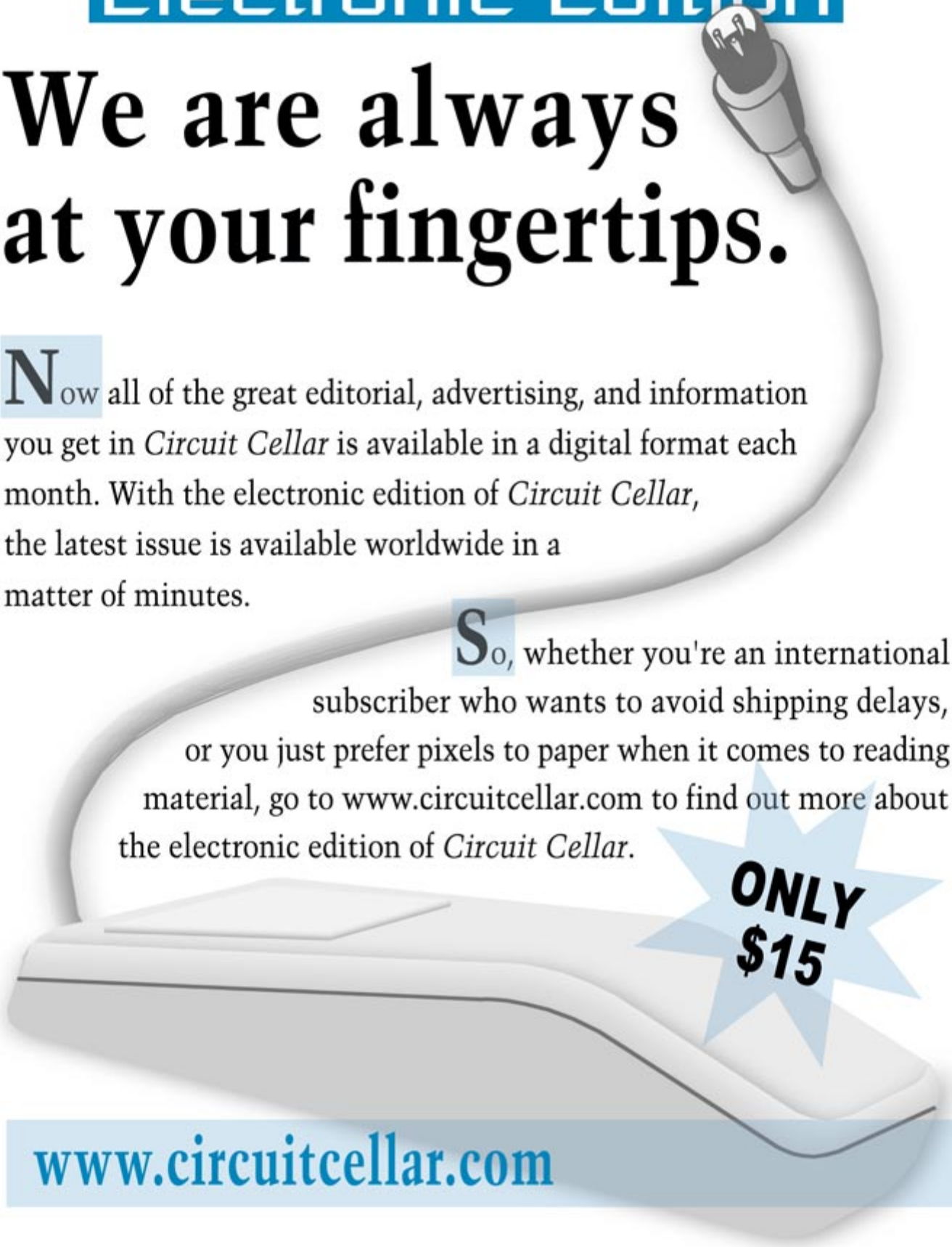
CIRCUIT CELLAR®

Electronic Edition

We are always at your fingertips.

Now all of the great editorial, advertising, and information you get in *Circuit Cellar* is available in a digital format each month. With the electronic edition of *Circuit Cellar*, the latest issue is available worldwide in a matter of minutes.

So, whether you're an international subscriber who wants to avoid shipping delays, or you just prefer pixels to paper when it comes to reading material, go to www.circuitcellar.com to find out more about the electronic edition of *Circuit Cellar*.



www.circuitcellar.com

Fred Eady

Building a Modular Programming Platform

Part 2: Building the PCB

Fred still doesn't have the PIC18F*xxx* parts he's been waiting for, but that doesn't mean he has lost sight of his goal to develop a low-cost '18F*xxx* programmer. This month, he shows you how to design an MPP PCB and challenges you to start your own MPP project.

It seems that PIC18F*xxx* parts are really difficult to come by. I ordered 27 and received only six, with the rest on back order for at least a month. That's OK because I have some other important MPP concepts to cover before you can use this version of the MPP to start putting real code into the vaporware PIC18F*xxx* devices.

I managed to assemble an MPP using point-to-point construction and it turned out to be a furry looking beast. I found that I needed much more space to mount the various MPP power modules than I had planned for. I had panels hanging off that little board that looked like solar panels on a space station. So, I decided to bite the bullet and design an MPP main printed circuit board (PCB) containing ample area for the power modules and, if necessary, breadboard area for those last second design changes. Having the breadboard area also allows time for those "oops" moments.

A raw version 1 MPP printed circuit board is the subject of Photo 1. The design of the MPP printed circuit board is going to change as you determine which components of the subsystems work best for the target application. I'll also divulge the particulars of the final PCB so you can clone your own MPP design.

DOING MY HOMEWORK

Since our last meeting, I've been just as busy producing PCBs as I have been reading datasheets. I settled on a single series of National Semiconductor Simple Switcher power converters for this spin of the MPP power module series. I'll be using the LM2674. The LM2674 is a high-efficiency, 500-mA step-down voltage regulator. Using the LM2674 implies the use of a higher voltage common supply that will be distributed among the MPP Simple Switcher power converters. I will also introduce you to a step-up design that can be used to feed the MPP Simple Switcher power converter network or stand alone as a source of programming high voltage. And, no discussion of switching power supplies would be complete without mentioning the LM78S40 switching voltage regulator.

I've collected a number of high-quality surface-mount Panasonic electrolytic capacitors, Coilcraft inductors, and Simple Switcher power converters. The Florida room inventory also includes a flock of PIC16F877 microcontrollers, NEC uPD71055L-10 parallel interface units, Dallas 128-KB nonvolatile memory modules, and the ever-present Sipex SP233ACP RS-232 communications ICs. Before I put the components on the MPP printed circuit board, let's do the ground training beginning with the National Semiconductor LM2674.

THE LM2674

I chose the LM2674 for a number of reasons. The MPP won't need a tremendous amount of current to operate, so the 500 mA rating of the LM2674 will easily handle the entire complement of MPP electrical components. A TTL-compatible ON/OFF pin allows you to use the LM2674 as a voltage source that you can switch in the MPP programming electronics and as a fixed bulk regulator for the PIC and its support circuitry. The LM2674 comes in fixed and adjustable versions and is just as easy and in some cases easier to implement than its linear cousins, the LM317 and LM340.

Another advantage to using the LM2674 is the absence of finned heatsinks. The LM2674 uses the available copper of the PCB it's mounted on to dissipate heat generated in the voltage regulation process. This is made possible by the high-efficiency, fixed 260-kHz switching frequency of the LM2674. The attributes that reduce the regulator-generated heat also allow the use of smaller filter components. And, the line and load regulation offered by the LM2674 is significantly better than that of the second generation Simple Switcher power converters that rock and roll at 150 kHz. The MPP LM2674s are in standard 8-pin DIP packages.

I want to begin the MPP power module construction process by the book, or better yet, by the datasheet. In reality, the designs will be guided by software that embodies both. So, I'll use the services of the Simple Switcher power converter design software, Switchers Made Simple. A quick look at the National Semiconductor web site indicates that V.6.24 is the latest and greatest, so that's what I'll use for the designs found in this text.

I'll work closely with Sr. Engineering Manager for Power Management Applications Jon Cronk of National Semi to avoid potential problems in applying the LM2674. If you search for the LM2674 on National's web site, you'll find a section called Design Tools. Click on the PDF and you'll bring up the design software for Simple Switcher DC/DC converters. The link to the latest releases leads to some sage words by Jon about using Simple Switcher power converters. I spoke to Jon about some of the things I'll be doing in this text. I'll throw in his comments as I get to the pertinent subject matter. Now, let's design and build some LM2674 MPP power modules.

POWERING UP

Not only did I build a PCB for the MPP prototype, but I built some PCBs for the MPP power module assemblies as well. I waffled for days on the decision to do the small power modules by hand or send out for boards. I really liked the layout of the demo boards in the LM2674 datasheet and eventually that idea won out. [1]

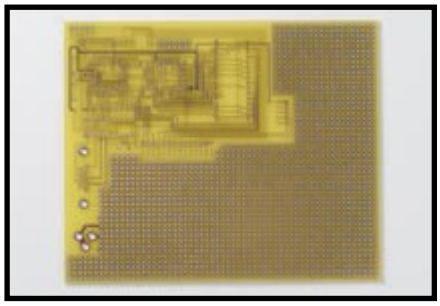


Photo 1—*There isn't much to look at right now, but soon there will be little skyscrapers and parking garages all over this board.*

Photo 2 shows the resultant shiny unpopulated power module PCBs. If you consult the LM2674 datasheet and look closely at Photo 2, you'll see that there are actually two power module printed circuit boards, one for the adjustable switcher and one for the fixed part. I reproduced both of them and left room for a potentiometer on the adjustable regulator's PCB.

The MPP main printed circuit board has plenty of room to piggyback up to six of the little power modules if you and I decide to go crazy with the Simple Switcher power converter on the MPP. Before you start mounting the microcontroller-oriented components of the MPP, let's design and build the fixed 5-VDC power module that will support them.

The first question that the Switchers Made Simple software asks is what the input voltage limits will be. The second question is, what output voltages will be required and how much output current should be allowed for at the selected output voltage?

The highest regulated voltage that will be present on the MPP is 13 VDC. I anticipate a minimal overall MPP current draw that will not tax the 17 VDC at 600 mA unregulated power brick that I'll use in this spin of the MPP design. Why 17 VDC? I happen to have a few of them in the Florida room, and I can't get my hands on a 18-VDC brick. This isn't a government job, so the 17-VDC brick will work fine. After putting the leads of a Wavetek digital VOM on the unloaded output of the 17-VDC brick, I determined that the highest unregulated voltage that would exist on the input of any power module on the MPP would be 22 VDC. The input voltages are well within the input voltage parameter of the LM2674, which, according to the LM2674 datasheet, uses all of the 40 VDC.

To set up, I told tell the Switchers Made Simple software to give me component values for a fixed 5 VDC at 500-mA output. I already know about the features of the LM2674, but I clicked on items in the feature wish list for grins. My LM2674 design selections are shown in Photo 3.

After selecting the LM2674 from the list of Simple Switcher converters that appeared when I clicked on the Start Selection button, the schematic and recommended parts list in Photo 4 appeared. Because I don't have the Vishay-Sprague parts noted in the schematic, I substituted my Panasonic components to closely match the capacitor values in the Switchers Made Simple software design I just generated. A Panasonic 22-µF device at 35 WVDC with an ESR value of 0.034 Ω fills in for the Vishay-Sprague capacitor labeled CIN. A 100-µF device at 35 WVDC with a 0.16-Ω ESR replaces the COUT output filter capacitor. The catch diode, D1, is a 1N5819 on the MPP fixed 5-VDC power module.

As you know, I have a box full of Coilcraft inductors. So, L1's Sumida actually will be a Coilcraft 68-µH DO3308P-683. There's a blurb in the LM2674 datasheet in the Design Procedure area that points out the differences among inductors. It looks like the best inductors when it comes to EMI are Schott EP core inductors and pulse-powdered iron toroid core inductors. Jon also mentioned in our conversation that the pulse toroids are good to use if your design is experiencing startup problems. You won't have to worry about that in your LM2674 designs. The inductors used

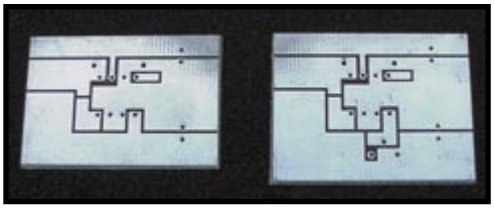


Photo 2—*The continuous copper is there more for providing a substantial heatsink than for suppressing noise. The MPP designs require only 1 square inch of copper surface area.*

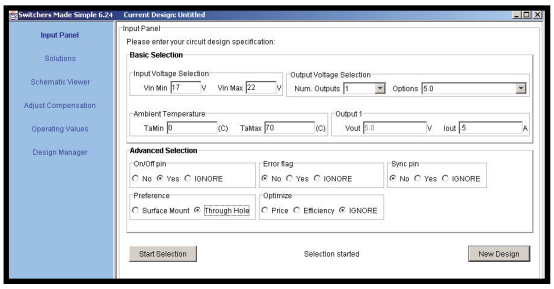


Photo 3—*If you aren't familiar with the National Semiconductor Simple Switcher power converter line, this screen helps you narrow down the choices depending on your application needs.*

are open-core ferrite drum inductors that emit some EMI but not as much as ferrite stick inductors (which are also in the Switchers Made Simple inventory). Your first MPP power module, the fixed 5-VDC design, is shown loaded with the aforementioned parts in Photo 5.

At this point, I returned to the various MPP component datasheets and did a nasty overall maximum current calculation. I came up with about 250 mA for a worst case. You already know what I did next. Photo 6a shows that the 68-µH inductor is now a 100-µH inductor; this is all that changed when I ran the Switchers Made Simple software using 250 mA as the maximum output load current of the fixed 5-VDC module.

Of course, I kept running the maximum current value down toward 100 mA and the L1 inductor values kept rising with each reduction of output current. The reason for this is that in Continuous mode a larger inductor is needed to keep the inductor current flowing continuously at low-output current loads. In Discontinuous mode, the inductor current drops to zero for a period of time during the switching cycle.

Even though the drop to zero statement sounds bad, there's absolutely nothing wrong with running in Discontinuous mode, because the LM2674 (and any other Simple Switcher power converter) can handle it without a problem. The advantages of running in Continuous mode are lower output ripple voltage and lower peak currents in the switching components. The disadvantage in low-output current situations is the need for larger inductors, as the Switchers Made Simple software proved.

The bottom line is that there is a good chance that the power modules riding on the MPP will at some time go to Discontinuous mode even though the design parameters are geared for Continuous mode. If necessary, I'll tweak the power module component values as you learn more about how the MPP is going to behave when power is applied to all of its electronic parts, but I won't concern myself with which mode the MPP power modules assume during normal operation.

Now, let's move on and use Switchers Made Simple to generate a parts list and schematic for the adjustable MPP programming supply. The target output voltage is 13 VDC with a maximum current output of 100 mA. The actual current consumption will be far below the 100-mA level, considering that the MCLR programming pin of the PIC18F*xxx* device draws almost no current at all.

Choosing the LM2674 in the adjustable version results in the schematic shown in Photo 6b. With the exception of the resistors in the feedback loop, this circuit is identical to the fixed version of the schematic. Resistor Rfb1 is fixed and will actually be a 2.4-kΩ 5% resistor instead of the specified 2.49-kΩ 1% resistor. Rfb2 has a value of 24.3 kΩ and will be substituted with a 50-kΩ potentiometer. Notice that the 100-mA output current value results in a recommended inductor value of 220 µH. This time the Switchers Made Simple software asked for a Coilcraft inductor.

The capacitor values go unchanged and I will build up the 13-VDC power module with the same capacitor values found in the fixed 5-VDC MPP power module. I've decided to use an adjustable switching regulator circuit for the target bulk power supply of the PIC, as well. Doing this will allow the programming of target PIC parts that will end up in circuits that aren't powered by a standard +5-VDC supply. The completed 13-VDC MPP adjustable power module is displayed in Photo 7.

I'll add power module status LEDs on the MPP main board for a couple of reasons. It would be beneficial to have a visual indication of the state of each MPP power module's output voltage. The second reason is that the LM2674 requires a minimum current of about 2 mA to overcome a leak that may hamper the operation of the floating gate drive within the LM2674. By using LEDs to load the MPP power modules, I hit two birds with one stone.

That wraps up the MPP power module area as far as step-down regulation is concerned. Everything is in place to power the MPP microcontroller and its friends and provide a source of high voltage and bulk power for the '18F*xxx* programming operation. Although you have a perfectly fine power solution in hand, you should think about alternative ways to perform some of the same power-related tasks.

ALTERNATE POWER

All of the necessary lines required to control the three power modules are grouped on a header strip directly below the PIC, PIU, and RAM. To keep the step-up power options open, I also pulled a line from the PIC that can provide a software-controlled pulse width modulation signal.

The idea is to use the PWM signal of the PIC to bang at the base of a standard small signal switching transistor. [2] The switching transistor is partner to an inductor, catch diode, and output capacitor. At first glance, the component brew smells of a switching regulator.

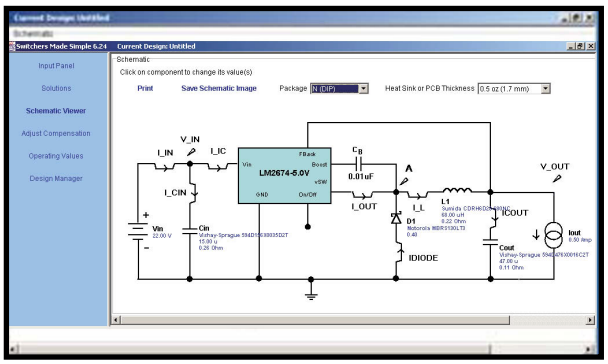


Photo 4—*The conclusions are the same as if you followed the design procedure in the LM2674 datasheet.*

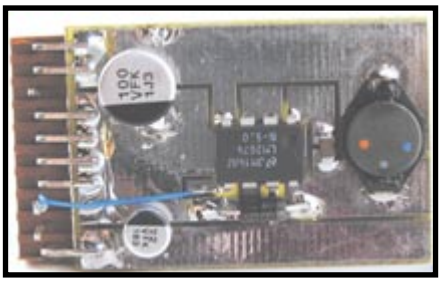


Photo 5—*I removed the plastic mounting tabs from the capacitors and formed the leads. The result is a surface mount capacitor turned radial capacitor with 0.1″ lead centers. The MPP power requirements are small, so I could have used sockets but I opted to do it by the book and solder the LM2674s directly to the printed circuit board.*

In fact, the mix of switching regulator parts forms a sort of step-up switching regulator. The result is not as sophisticated as what you might find in a National Semiconductor step-up converter, but with the addition of a standard low-current LM317LZ linear regulator, it becomes a high-voltage flash programming source (see Figure 1).

The RC2/CCP1 pin of the PIC16F877 has double duty. Although it is a compare/capture pin when not being used as general purpose I/O, it's also the output for one of two PWMs when placed in PWM mode by the firmware. Writing a 0x0C to the PIC's CCP1CON register, which controls the internal capture/compare/PWM hardware of the PIC16F877, puts the PIC pin in PWM mode. When in PWM mode, the RC2/CCP1 pin is capable of producing a 10-bit resolution PWM output.

Timer 2 (TMR2) of the PIC16F877 is integral to the PWM process. In the MPP application, TMR2 is not prescaled. And, with a 20-MHz processor clock, each TMR2 tick is 200-ns long. After experimenting with various values for TMR2 based on waveforms from the LM78S40 driving a similar load, I concluded that generating enough energy to produce a high PIC flash programming voltage requires a TMR2 period of 25.6 µs. That equates to loading the PR2 register of TMR2 with 127 decimal. The PWM frequency is the inverse of the PWM period, equaling 39.063 kHz.

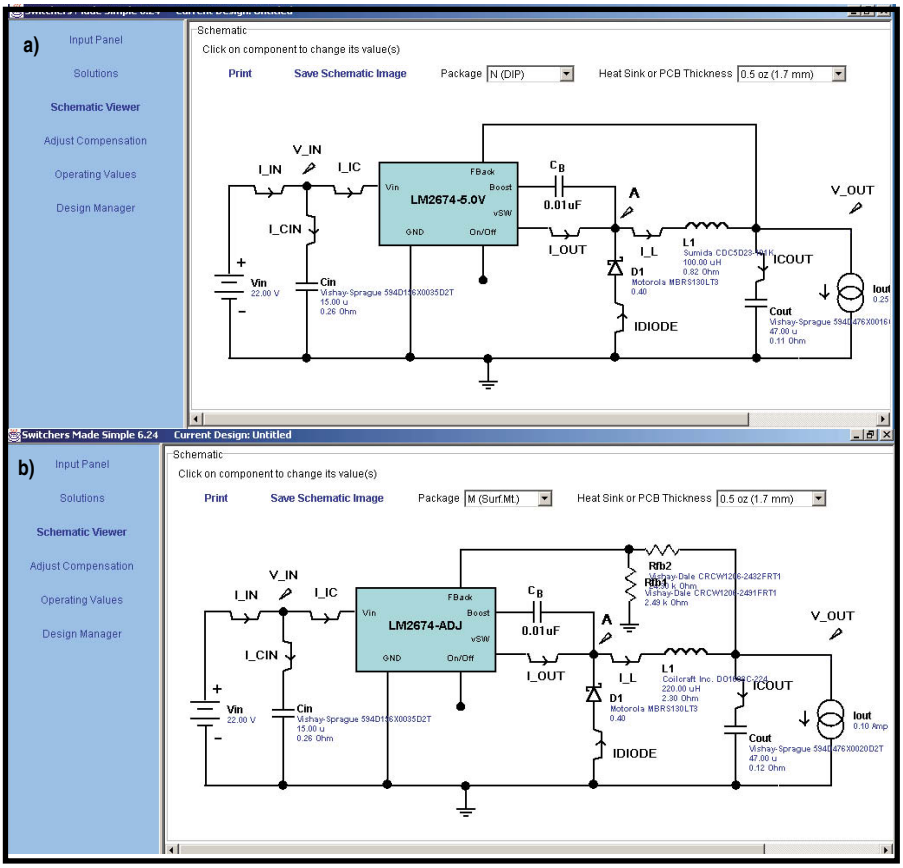


Photo 6a—*The beauty of the Switchers Made Simple software is that what-if design work can be done quickly and easily. And you don't have to go back and verify the results.* **b—***As far as parts count goes, the LM2674 adjustable circuit has only one more component than its linear counterpart. If you want to count heatsinks as components, there is an even number of them.*

All the clues point to B+K Precision®'s new family of Device Programmers



Search no more!

The B+K Precision®'s new model 869 Stand Alone Device Programmer was designed to operate either as a Stand alone Programmer or as a desktop programmer.

Features:

- 48 Pin DIP ZIF socket with universal pin-drivers
- Auto-run mode starts programming automatically upon detectionof chip insertion
- Supports low voltage (1.8V) devices
- Supports Windows2000/NT/9x (XP compatible)

Put an end to your search today!

(714) 921-9095 or
www.bkprecision.com
Ask for your free copy of our new full line catalog



22820 Savi Ranch Parkway,
Yorba Linda, CA 92887
(714) 237-9220 Fax: (714) 237-9214
www.bkprecision.com

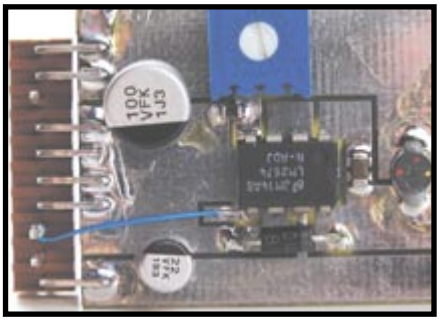


Photo 7—*The MPP module isn't much different than the fixed module. The direct soldered/wire wrap connection is the ON/OFF pin.*

According to the PIC16F877 datasheet, the PWM period is calculated as follows:

PWM period = (PR2 + 1) × 4 × Tosc × TMR2 prescale value

where *PR2* equals 127, *Tosc* equals 50 ns, and the *TMR2* prescale value equals 1. Inverting the 20-MHz clock frequency of the MPP PIC16F877 derives the datasheet value known as *Tosc*. Each PIC instruction cycle consists of four *Tosc* cycles. Plugging the *PR2* and *Tosc* values into the PWM period equation yields a period of 25.6 µs.

With *PR2* loaded with 127, *TMR2* is continually attempting to count from 0 to 127 (the manually set PR2 value) and reset. This discussion concerns generating pulses, so it's logical to assume that *TMR2* can count only as high as PR2 + 1 = 128. And some type of event is generated when the maximum count is reached.

In fact, three events occur when *TMR2* reaches the *PR2* value. First, *TMR2* is cleared. Then, the CCP1 pin is set and the PWM duty cycle value is latched into a holding register, CCPR1H. The setting of the CCP1 pin occurs when *TMR2* and *PR2* are equal. After the duty cycle value is reached, the CCP1 pin is reset and taken low. So, a duty cycle of 100% would theoretically mean that the CCP1 pin would never reset and go low, and a duty cycle of 0% would never allow the CCP1 pin to go high. The time the CCP1 pin spends in a high state within a PWM period is defined as the duty cycle. Again, I experimented with the duty cycle value based on waveforms generated by an LM78S40 driving a similar low-current load. I settled on a duty cycle value of 200 decimal, which closely approximates the duty cycle of the LM78S40.

The duty cycle value is written to the 8-bit duty cycle register CCPR1L and the upper two bits of the CCP1CON register for 10-bit duty cycle resolution. At the completion of a PWM period, the duty cycle value in CCPR1L is passed along to CCPR1H with the two bits of the CCP1CON register. This double buffering of the duty cycle value allows for a glitch-free PWM output. The duty cycle is defined mathematically as:

PWM duty cycle = (CCPR1L:CCP1CON<5:4> × Tosc × TMR2 prescale value)

where *CCPR1L:CCP1CON<5:4>* equals 200, *Tosc* equals 50 ns, and the *TMR2* prescale value equals 1.

The numbers result in a PWM duty cycle of 10 µs. Because the PWM period is 25.6 µs, the CCP1 PWM output pin is high for 10 µs and low for 15.6 µs during a single PWM period time. This equates to a 39% duty cycle and is enough to drive a low-current 22 VDC into the input of the LM317LZ. I've already tested this pseudo step-up switching regulator by using it to successfully program a variety of PIC16F8*xx* parts.

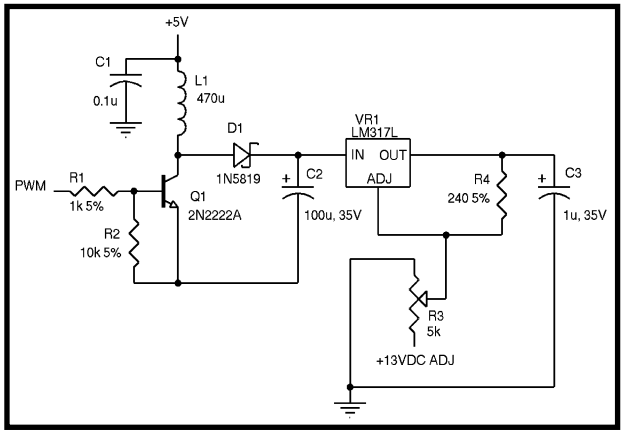


Figure 1—*This is as basic as you can get with a switching regulator circuit. More sophistication could be added by creating a feedback loop that ties in the analog inputs of the PIC to monitor its operation, and thus use the MPU of the PIC to control the PWM duty cycle accordingly.*

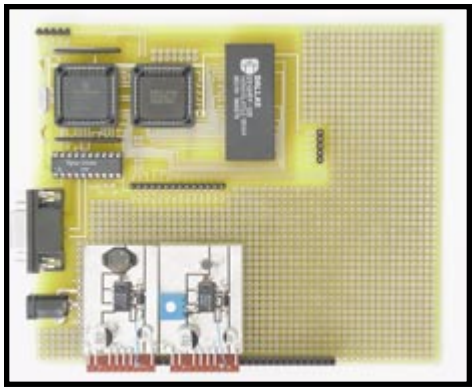


Photo 8—*The current version of the MPP has provisions for ICSP programming as well as modular power sources.*

Speaking of testing, I also recently wrote an article series that featured the LM78S40 switching regulator (*Circuit Cellar* 138–139). The LM78S40 has more spark than the PIC PWM circuit and is capable of providing a considerable amount of output current. If your version of the MPP will be required to host parts that require more than a couple milliamps of current from the high-voltage programming voltage source, the LM78S40 is an inexpensive and easy way to get it. During the course of writing the series, I created an Excel spreadsheet to help with the design. It isn't quite Switchers Made Simple, but it's accurate and reflects the intentions of the LM78S40 datasheet and application notes.

NEXT TIME

Although this discussion centered on MPP power, I placed all of the components on the MPP main board in Photo 8 to give you an idea of what it all will look like when the construction is finished.

Since then, I've added some right-angle header posts to each of the MPP power modules to facilitate easy access, insertion, and removal. Each module is identical in respect to the PCB, so a standard power module pinout came along automatically. The standardization of the MPP power module input and output pins allows me to create a power and control bus that can accommodate any module that adheres to the pinout scheme. All of the design

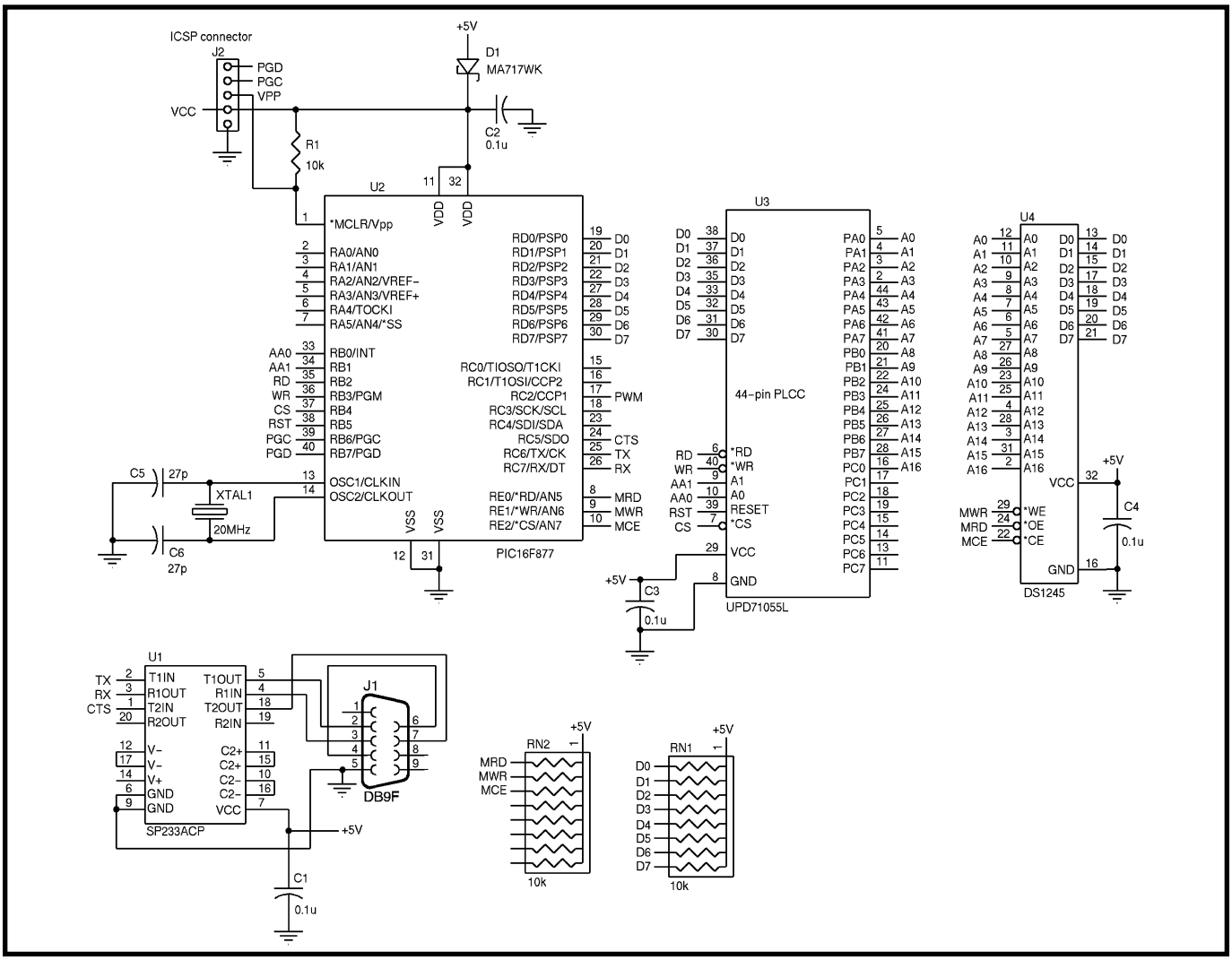


Figure 2—*I decided to take full control of the chip enable pins of the RAM and PIU to avoid any possibility of conflict caused by a hardware design error.*

changes are documented in Figure 2.

Another quick check on the Internet revealed that delivery of the PIC18F*xxx* parts is still pushed to the end of June. That gives me time to finish the rest of the story. Next time I'll talk about finishing the MPP by installing the microprocessor, parallel interface unit, RS-232 converter, and the static RAM. Then, I'll walk you through writing the C-based MPP operating code using the CCS C Compiler. The PIC18F*xxx* microcontrollers have a different programming algorithm so it won't be the same ho-hum code you've seen before. Hopefully, the PIC18F*xxx* parts will be on the market by then and you can grab one and program it.

Fred Eady 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. Fred may be reached at fred@edtp.com.

REFERENCES

[1] National Semiconductor Corp., "LM2674 Simple Switcher Power Converter High Efficiency 500 mA Step-Down Voltage Regulator," DS100041, May 10, 2001.

[2] Microchip Technology, Inc., "PIC16F87X-EEPROM Memory Programming Specification," DS39025F, March 1, 2002.

SOURCES

Coilcraft inductors
Coilcraft, Inc.
(800) 639-1469
(847) 639-6400
www.coilcraft.com

Nonvolatile memory modules
Dallas Semiconductor, Inc.
(972) 371-4000
www.dalsemi.com

PIC16F877 microcontroller
Microchip Technology Inc.
(480) 786-7200
www.microchip.com

LM2674 Simple Switcher
National Semiconductor Corp.
(408) 721-5000
www.national.com

Electrolytic capacitors
Panasonic
www.panasonic.com

SP233ACP Communications ICs
Sipex Corp.
(978) 667-8700
www.sipex.com

FROM THE BENCH

Jeff Bachiochi

SmartMedia File Storage

Part 3: Reading a File



Jeff has explained the basic circuitry and how to navigate within the DOS FAT file structure. Now, he'll demonstrate how to read SmartMedia files and implement ECC routines to verify the integrity of your data.

The stability of a building is based on its cornerstone (or keystone). No matter how well the roof or any single part is constructed, all will fall down if it's built on a weak foundation. Likewise, it is imperative that the basics of any device be understood prior to attempting to use that device. You may wish to use only a small part of its total capability, but understanding how it all fits together can go a long way in making your job easier in the long run. And so it is with the SmartMedia.

So far, I've discussed the circuitry you can use to interface with the SmartMedia, and how it is physically organized and identified (CIS/IDI). To allow the SmartMedia to be accessed, the DOS FAT file system was chosen because it's a standard already used by many other storage devices. Last month, I covered how to locate and use the boot, partition, directory, and FAT sectors in order to find out if any files are stored on the media. This month, I will cover reading files and show you how to make use of sector ECC to determine data validity.

DIRECTORY ENTRY

I ended last month with a display of the directory (or subdirectory) contents. Any file that had been previously stored via a PC with a SmartMedia interface (or digital camera or other peripheral) will show up using the DIR command. Although the directory output displays FILENAME.EXT and file size, you will need more information to determine where the sectors that hold the file's data can be found.

In Part 1, I explained the format of each section of the DOS FAT file system; you can refer to those tables for the scoop. Among the other information stored in each 32-byte directory entry within the directory sectors, the last six bytes contain a 2-byte cluster number and 4-byte file size. The file size not only tells how big the file is, but also how many clusters (blocks) are needed to hold the file's data.

The media's block size is based on media capacity. This is defined as a number of pages per block, where a page is a number of 256-byte units per page. There is terminology confusion with early 1- and 2-MB devices that use one 256-byte unit per page; however, all larger devices are described as having a page equal to 512 bytes (consisting of an even and an odd page). Luckily, the media's block size is consistent with the cluster size of the DOS FAT file system as defined in the partition sector.

Please keep in mind that for the purposes of this discussion, all specific values are based on an 8-MB SmartMedia

```
LBA = 20, PBA = 40 even page
41 54 00 45  00 53 00 54 00 2E 00 0F 00  8F 54 00
58 00 54 00  00 00 FF FFFF FF 00 00 FF FF FF FF
54 45 53 54  20 20 20 20 54 58 54 20 00  AA10  5F
A8 2CA8 2C 00 00 11 5F A82C 02 00 14  00 00 00
54 45 53 54  32 20 20 20 54 58 54 20 00  88 39  5F
A8 2CA8 2C 00 00 3A 5F A82C 03 001A  00 00  00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00  00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00  00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00  00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00  00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00  00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00  00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00  00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00  00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00  00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00  00

LBA = 20, PBA = 40 redundant page
FF FFFF FF FF FF10 04 FF FF FF 10  04 0F F3 CF
```

Figure 1—*This is a dump of LB 2 page 0 (first directory sector). The logic block can be determined from the 2-byte BAF (1004) stored in the redundant page.*

device. You can refer to Part 1 for specific values for other SmartMedia capacities. Although a file has a 2-byte cluster number, the 12-bit FAT table allows only a 12-bit value. Of the 4096 potential values, only 4085 values (2 to 4086) are recognized as legal cluster designations. The 2-byte cluster value tells you where the beginning of the file can be found.

As we discussed last month, a cluster is the minimum space allocated to file storage. If a file's size is less than or equal to a cluster's size (512 bytes/page × 16 pages/block = 8 KB), then it can fit within a single cluster. The first available cluster in the FAT is 2 (0 and 1 are reserved). The partition sector helped determine where the beginning of the storage area is (LB = 3). This means that cluster number 2 really is LB = 3, the difference being a cluster-to-LB offset value (LB = cluster number + cluster-to-LB offset).

A logic block (LB) is not equal to a physical block (PB) on the SmartMedia. Because an LB can be physically anywhere on the SmartMedia, the PB must have within it an indication of which LB it represents. This LB is held within a 16-byte redundant area stored along with each page. Instead of searching through the physical blocks on the SmartMedia for the one logic block of interest, an external I²C EEPROM helps to keep track. A translation table in the EEPROM is indexed by a logic block.

By simply reading an address (LB × 2) in the EEPROM, the returned value is a PB where it will be found on the SmartMedia. Although establishing this table (and keeping it updated) is extra work on the part of the application, it reduces the number of times you need to access the SmartMedia.

TEST.TXT

Let's start off with a small text message file. You entered the DIR command last month and got back a directory listing that shows a TEST.TXT file with 20 bytes in it (see Listing 1). Now, type in the read TEST.TXT command, and the application will begin locating the requested TEST.TXT file name in the directory.

To locate the requested file name, start in the first page of the directory, which is page 0 of LB = 2 or sector LBA = 20 (the first page of the directory). A word is read from the I²C EEPROM beginning at address (LB × 2). This location holds the physical block where logic block 2 can be found (in this case, the word = 0004). Page 0 of LB = 4 is found in sector PBA = 40. Figure 1 shows a dump of that page read from the SmartMedia. At the end of the directory entry for the file TEST.TXT you can see the 2-byte cluster number = 0002 and the 4-byte file size = 00000014. Adding the cluster-to-LB offset = 1 to the 2-byte cluster number = 0002 you get LB = 3.

Now, it's off to the EEPROM once again for a translation of logic block to physical block. This time the physical block is 0005 for LB = 3. A dump of PBA = 50 shows the data in the first 20 bytes of the even page (see Figure 2). Not only is the rest of this page (even and odd) empty, but the next 15 pages are also empty (and wasted, so to speak) because the minimum storage is one cluster (1 block – 16 pages – 8 KB).

Listing 1—*When the DIR command is entered, the following output is displayed. The output is formatted similarly to the DIR command in DOS.*

```
S:
DIR
Volume in drive S is
Volume Serial Number is 0000-0000
Directory of S:

TEST    .TXT 00000014  bytes
TEST2   .TXT 0000001A  bytes

0002  files(s) 000000002D  bytes
0000  dir(s) 00008DFFD3  bytes free
```

```
LBA = 30, PBA = 50 even page
54 68 69 73  20 69 73 20 61 20 74 65 73  74 20 66
69 6C 65 21 00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00
00 00 00 00  00 00 00 00 00 00 00 00 00  00 00 00

LBA = 30, PBA = 50 redundant page
FF FFFF FF FF FF10 07 FF FF FF 10  07 CF FF CF
```

Figure 2—*An even-page dump shows the ASCII characters of the text data in file TEST.TXT. The last three bytes of the redundant page hold the ECC values for the even page.*

The page's data is written into the TX buffer (up to the file size – 20 bytes) for transmission to the host. Longer files might require multiple pages of this PB = 5 to be read and transmitted (PBA = 50 – 5F). Now, I should say something about the PIC18F452's RAM, which includes six 256-byte banks.

COMM BUFFERS

To store a full page of data on the PIC18F452, you use three separate banks—one each for the even page, odd page, and redundant page. To talk with the host using serial communications, separate RX and TX buffers are used in two additional pages of RAM, leaving a page or so for application variables. The RX and TX ring buffers, although large, could be easily overrun with data. To prevent this, hardware flow control is implemented to keep everything in line. Note that the signal levels discussed here are the RS-232 levels not the TTL.

Upon reset, the application raises its CTS line, signaling that it's ready to accept a command from the host. The application monitors the state of the RX ring buffer. When it gets almost full, the application lowers the CTS line to tell the host to pause sending anymore characters.

As soon as the application has sufficiently emptied the RX buffer, the CTS output is raised to notify the host that it may resume sending characters. When a command requires the host to send a file, it raises its RTS output. The application monitors the

```
LBA = 30, PBA = 1200 even page
2E 20 20 20  20 20 20  20 20 20 20 10 00 00 00 00
00 00 00 00  00 00 00  00 21 24 02  00 00 00 00 00
2E 2E 20 20  20 20 20 20 20 20 20  10 00 00 00 00
00 00 00 00  00 00 00  00 21 24 00  00 00 00 00 00
50 31 30 31  30 30 30  31 4A 50 47  20 00 00 00 00
21 24 92 2C 00 00 00  00 21  24 03 00 4D FC 03 00
50 31 30 31  30 30 30  32 4A 50 47  20 00 00 00 00
00 00 6B 2C 00 00 00  00 21 24 23  00 30 E9 03 00
50 31 30 31  30 30 30  33 4A 50 47  20 00 00 00 00
00 00 90 2C 00 00 00  00 21 24  43  00AD 5F 03 00
00 00 00 00  00 00 00  00 00 00 00  00 00  00 00 00
00 00 00 00  00 00 00  00 00 00 00  00 00  00 00 00
00 00 00 00  00 00 00  00 00 00 00  00 00  00 00 00
00 00 00 00  00 00 00  00 00 00 00  00 00  00 00 00
00 00 00 00  00 00 00  00 00 00 00  00 00  00 00 00
00 00 00 00  00 00 00  00 00 00 00  00 00  00 00 00
```

Figure 3—*The subdirectory entry of file I1010002.jpg (pale blue) begins in cluster number 0x0023 (medium blue). The file length (dark blue) is 0x0003E930 bytes long.*

RTS line and knows that the file transfer is finished when it sees that the host has lowered the RTS line.

The hardware handshaking works much the same way for the TX buffer. Command responses are sent to the host only when the RTS line is high. When a file must be sent to the host, the TX ring buffer can be quickly filled by the application. It is only allowed to fill the buffer just so much and must then wait for the host to empty it before adding additional characters. The application uses the CTS output to indicate the status of a file transfer to the host. The CTS output is lowered by the application after the complete file has been sent.

Using this simple handshaking technique communicates the file transfer status without having to send file sizes. This would be unknown for logging applications.

BIGGER FILES

You've already seen how a small file requiring a single cluster is pointed to right from the 32-byte directory entry. The 2-byte cluster value points not only to the cluster holding the file, but also to an entry in the FAT table. If you read the page(s) holding the FAT entry, you should see 0xFFF in the entry for the small TEST.TXT file you just read.

Refer to last month's column to see how the packed 12-bit FAT table entries are stored. The 0xFFF entry verifies that indeed this is the only cluster assigned to the file. A number between two and 0xFF6 would indicate that the file has another cluster assigned to it and the value is the cluster number for the next portion of the file's data. With this chaining scheme a file will consist of multiple clusters ending when a cluster's FAT entry value is 0xFFF. As a check, the number of clusters used should be the file size divided by 8 KB (rounded up to the next higher cluster).

A larger file (i.e., I1010002.JPG entry in Figure 3) differs from the small file in that the first cluster, 0x23, is not marked as the EOF in the FAT table (see Figure 4). Instead, it is chained to the next cluster, 0x24. Altogether, this file uses 32 clusters. 0x3E930 (file size) /8 KB (cluster size).

The application will reduce a file size counter as it transfers page data into the TX buffer for transmission to the host. It continues to access the FAT looking for the next chained cluster after transferring the last odd page within the present cluster. When the file size counter decrements to zero transferring ceases and the CTS output is lowered, indicating an EOF to the host. When the host replies by lowering the RTS, the application sends the ready prompt, S: (and subdirectory path if used), and awaits further commands.

For an ASCII text file, like the first example TEST.TXT, the output can be seen using HyperTerminal. The following shows the text located at the root directory being output from the SmartMedia inserted into the application project:

```
S:
READ TEST.TXT
This is a test file!
S:
```

Because P1010002.jpg is binary, it will cause unpredictable visual output.

```
LBA = 1A, PBA = 12EA even page
F8 FF FF FF  4F 00 05  60 00 07 80 00 09 A0 00 0B
C0 00 0D E0 00  0F 00  01 11 20 01 13 40 01 15 60
01 17  80 01 19  A0 01 1B C0 01 1D E0 01 1F 00 02
21 20  02 FF 4F 02  25 60 02  27 80 02 29 A0 02 2B
C0 02 2D E0 02 2F  00 03 31  20 03 33 40 03  35 60
03 37  80 03 39  A0 03 3B C0 03 3D E0 03 3F 00 04
41 20  04 FF 4F 04  45 60 04  47 80 04  49 A0 04 4B
C0 04 4D E0 04 4F  00 05  51 20 05 53  40 05 55 60
05 57 80  05  59 A0 05 5B C0 05 5D F0 FF 00 00 00
00 00 90  2C 00 00  00 00 21 24  43 00 AD 5F 03 00
00 00 00  00  00 00  00 00 00 00  00 00 00  00 00 00
00 00 00  00  00 00  00 00 00 00  00 00 00  00 00 00
00 00 00  00  00 00  00 00 00 00  00 00 00  00 00 00
00 00 00  00  00 00  00 00 00 00  00 00 00  00 00 00
00 00 00  00  00 00  00 00 00 00  00 00 00  00 00 00
00 00 00  00  00 00  00 00 00 00  00 00 00  00 00 00
```

Figure 4—*The 12-bit packed FAT entry number 0x23 (medium blue) indicates the file is chained to cluster number 0x24. FAT entry 0x24 is also chained to cluster number 0x25 (pale blue) and so forth up to FAT entry number 0x42 (dark blue), which ends with an EOF marker 0xFFF.*

ERROR CORRECTION CODE

I highly recommend that you use the ECC routines to assure data integrity. Although, you may remember that PCs used to have a parity bit and that has been abandoned. In any scheme, the potential exists for the protection to fail as well as the data. By adding protection you increase potential errors. ECC is capable of detecting and correcting single bit errors in the data or an ECC value. Two bit errors are detectable but not repairable. Higher bit errors just play havoc. Here's how it works.

The 6-bit column parity (CP) is based on an 8-bit data byte value. The place value of each of the eight bits is 2^{0-7} or it could thought of as $2^{0b000-0b111}$ (see Figure 5). The 6-bit CP has an odd parity bit (based on the bit values of the data byte) for each of the six possible 4-bit groups, which are: 0bxx0, 0bxx1, 0bx0x, 0bx1x, 0b0xx, 0b1xx (where *x* is don't care).

You will notice that any single bit in the data byte will affect only 3 bits of the 6-bit CP. The CP (6-bit value) is recalculated by XORing the previous CP value with the new calculated CP value for each data byte of a 256-byte data page (either odd or even page, they each have an ECC stored in the page's redundant area). Because any data byte can have only 256 possibilities, a precalculated data look-up table can save computational time, providing you have the space available.

As the bits of each byte are condensed into a 6-bit CP value, the 256 bytes of each page are also condensed into a 16-bit line parity (LP) word. (For more details, read my article from October 1999, "Get SmartMedia—Part 2: Hands-On," (*Circuit Cellar* 111)). When an ECC

CP0 = NOT (2^{0b000} XOR 2^{0b010} XOR 2^{0b100} XOR 2^{0b110})
or
CP0 = NOT (2^0 XOR 2^2 XOR 2^4 XOR 2^6)

CP1 = NOT (2^{0b001} XOR 2^{0b011} XOR 2^{0b101} XOR 2^{0b111})
or
CP1 = NOT (2^1 XOR 2^3 XOR 2^5 XOR 2^7)
CP2 = NOT (2^0 XOR 2^1 XOR 2^4 XOR 2^5)
CP3 = NOT (2^2 XOR 2^3 XOR 2^6 XOR 2^7)
CP4 = NOT (2^0 XOR 2^1 XOR 2^2 XOR 2^3)
CP5 = NOT (2^4 XOR 2^5 XOR 2^6 XOR 2^7)

Figure 5—*Column parity (CP) consists of a unique combination of 4 bits. Six bits (CP0–CP5) define the column parity for a single data byte. Notice that any particular data bit affects exactly three CP bits.*

is calculated on a page of data, the three ECC bytes can be saved in the redundant page when writing a page of data to the SmartMedia, or they can be compared to the redundant area of the retrieved page for a verification of good data. A 1-bit error in the data will show up as an incorrect bit value in one-half of the ECC bits (the odd/even bit pairs). This information points to a specific bit in a specific byte of the page. This bit in the data can be corrected. If there is a 1-bit error in an ECC byte, then there will be an error in the relationship of the odd/even bit pairs. This bit error also can be corrected. The 1-bit error in either a data or ECC value is the only error that can be corrected. All other inconsistencies must be treated as bad data.

A couple of experiments helped demonstrate that ECC is not just a good idea, but should be used. First, I changed the three ECC bytes in the redundant page of the TEST.TXT file (see Figure 2). When I inserted the SmartMedia into my PC's reader, it could not read the file. Changing the ECC bytes back to their original values allowed the PC to again read the file. Even though the data in the even page was not altered, the ECC determined that something was so far out of whack that it would not accept the data. In fact, there was no error message of any kind!

Next, I changed one character of data from an ASCII "a" to an ASC "c" and left the ECC values unchanged. This time when I popped the SmartMedia module into the PC, the changed data byte was corrected and displayed as "a" even though the actual data is physically a "c." The ECC values determined that a bit was wrong and corrected for data error.

GROUND WORK

All of this leads up to the last part of this SmartMedia story. Many of you might only need to write to a file to the SmartMedia module for your project. I believe you'll agree that all of this groundwork is a necessary evil, especially if you ever need to format the media or search for a specific logical or physical sector. Next time we'll get down to it. Writing to the SmartMedia isn't as easy as writing data out to a page. If you've been following closely, you might have a pretty good idea about what's involved with trying to keep everything straight.

Jeff Bachiochi (pronounced BAH-key-AH-key) is an electrical engineer on Circuit Cellar*'s engineering staff. He has worked in product design and manufacturing. You may reach him at jeff.bachiochi@ circuitcellar.com.*

RESOURCE

SmartMedia
SSFDC Forum
www.ssfdc.or.jp/english/index.htm

Frustration? No, thanks.
Fun? Yes, please.

NEW VERSION

Satisfied customers - the key to our success

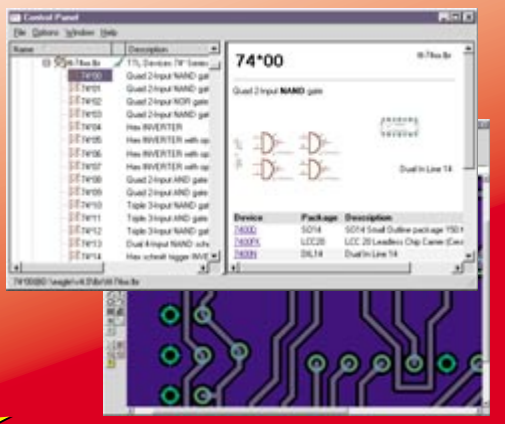
- that´s why every new EAGLE version is based on the feedback from our customers
- that´s why all our customers have access to our highly acclaimed, comprehensive support, free of charge
- that´s why EAGLE has no hidden costs for libraries or modules which prove to be indispensable after purchasing
- that´s why we really want customers to enjoy working with EAGLE
- that´s why EAGLE is one of the top-rated programs for schematic capture and board layout



EAGLE 4.0
Schematic Capture • Board Layout
Autorouter
for Windows®
and Linux

Windows is a registered trademark of Microsoft Corporation
Linux is a registered trademark of Linus Torvalds

Version 4.0 Highlights
- New Library Management with Component Browser
- Technology and Package variants for components
- Design your own commands via User Language
- Unlimited length for component names/values
- Design Rules define pad/via dimensions and shapes
- Net Classes for Autorouter and DRC
- Minimum Autorouter grid: 0.02 mm
- SMD pads can be rounded or round
- Different pad shapes for Top, Bottom, or Inner layers



EAGLE 4.0 Light is Freeware!
FREE
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.

The Standard Version is suitable for boards in Eurocard format with up to 4 signal layers 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

SILICON UPDATE

Tom Cantrell

FPGA News Flash



After two decades at the top of the FPGA world, Xilinx and Altera may soon find customer loyalty shifting in another direction. Tom explains that Actel is on the verge of sparking a flash memory revolution.

Word association test: When I say "FPGA," what do you say? Chances are most folks respond Xilinx or Altera.

That's understandable because these two companies have dominated the biz for a long time (see Figure 1). It's more than a matter of market share stats. Unlike some of the other contenders (Atmel and Cypress come to mind), Altera and Xilinx are single-minded in their pursuit of programmable logic, with no other product lines to diffuse their focus.

But there's another pure-play FPGA provider that gets far less attention: Actel. It's true that based on overall sales Actel might be considered minor league. Revenues are only about a quarter of those for either of the dynamic duo market leaders. Judging by the response to *Circuit Cellar*'s recent reader survey, that ratio extends to design interest as well. According to the survey, 26% of respondents expect to use Xilinx FPGAs and 20% anticipate shopping at Altera, but only 6% are signing up with Actel. It would be easy enough to explain the discrepancy if Actel was the new kid on the block. But the fact is, Actel, Xilinx, and Altera were all founded at about the same time in the mid-'80s.

So what happened? Why is Actel seemingly an also-ran at this point? More importantly, is there anything they can do to catch up, or are they doomed to bring up the rear forever?

ANTIFUSE DOWN

Actel's second-tier status can be traced back to the earliest days, and can be understood by taking a close look at the "P" in FPGA.

While Xilinx and Altera rely on SRAM to configure their chips, Actel placed all of their opening bets on a unique antifuse process technology. As the name implies, rather than using an SRAM bit to control a switch, Actel relied on a special process technology that essentially grows (like a fuse in reverse) the desired on-chip connections (see Figure 2).

In principle, designers couldn't care less about what's going on under an FPGA's hood. Put in a schematic or HDL description and out comes a chip. However, the antifuse FPGA approach has a fundamental weakness compared to Altera and Xilinx parts.

Unlike the latters' SRAM-based chips, Actel's antifuse devices can be programmed only once. Traditionally, OTPs are based on an EPROM (UV-erasable) process in a no-window package. But from the user perspective the result is the same: program the part once and that's it.

The difference might not have been compelling but for one minor problem. As I recall, early on, Actel had trouble getting the novel process recipe right and got a bad rap for antifuses that were prone to being more "anti" than "fuse" (i.e., flaky). That, in turn, led to pro-

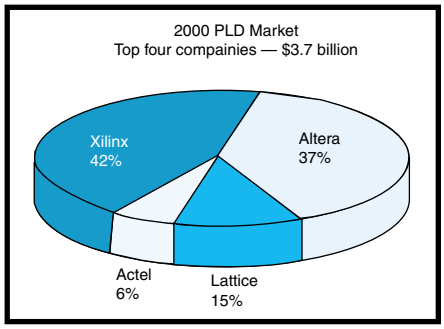


Figure 1—*According to the stats on Xilinx's web site, Actel lags in the programmable logic market.*

duction problems and missed deliveries, casting a pall over the entire concept.

Let's say you design-in and deploy an antifuse part and all seems well. But eventually, your customers report the appearance of minor bugs or intermittent glitches. Is it the FPGA or something else causing the trouble? If it's the FPGA, is it a legitimate design problem or is your FPGA starting to lose it's mind? This could get ugly.

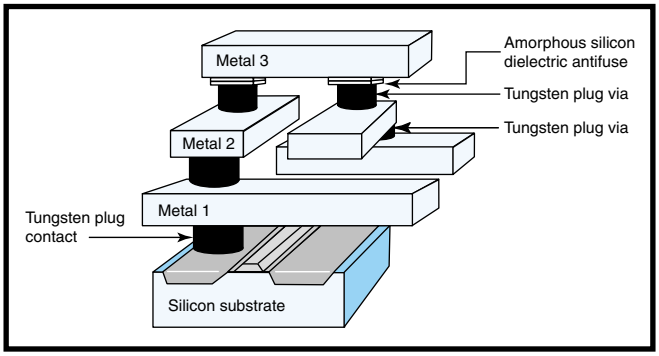


Figure 2—*By now, Actel has got antifuses wired, but it wasn't always so. Reliability concerns and production problems caused problems early on.*

Needless to say, as with all chips, it takes only one flaky connection to bring everything—including design-ins—to a screeching halt. Actel bet everything on antifuse, but Altera and Xilinx anted up SRAM and called their bluff.

Ironies abound. Early on, SRAM-based parts had folks worrying about the issue of volatility. Indeed, Xilinx spent more than a few years working up an antifuse-based product line, only to see the need evaporate as Actel's problems mounted and SRAM reprogrammability came to be viewed by the market as a feature rather than a flaw.

IT'S ABOUT TIME

I presume Actel eventually got the antifuse recipe right because they've been successfully selling parts for many years. However, I think it's safe to say the stumble out of the starting gate set them back.

When it comes to complex logic, when designers commit to an architecture and toolchain, it's not likely they'll switch on a whim. That's certainly true for micros, for example, witnessing the durability of classic parts like the PIC, '51, and '68 holding their own against newer entrants such as the AVR and H8.

The same goes for FPGAs. When designers were hooked on Altera or Xilinx, it's no surprise they'd be unlikely to switch without some compelling reason to do so. Even after they fixed their antifuse problems, the earlier uncertainty and designer inertia conspired to hold Actel back.

I have to confess to a bit of this FPGA-market myopia myself. Way back when, I put Actel into a little antifuse box and shoved it in the corner (out of sight, out of mind). That was a mistake. It turns out that for the last couple of years Actel has been making some moves (notably, the acquisition of a startup called Gatefield) that could well lead to redemption or even revolution in the FPGA marketplace.

If Yogi Berra were to comment, he might say nobody's picked up Gatefield-now-Actel's idea, namely a flash FPGA, because it's so obvious.

FPGA VS. FRGA

Let's take a deep breath and ponder the various implications, starting with the "Field-Programmable" tag.

Actel's original antifuse parts literally meet the definition of field programmable, even if it is only one time. Meanwhile, by all rights SRAM-based parts from Altera and Xilinx should really be called field reprogrammable.

Going further, who wants to go out into the middle of a field to program a part? The fact is, the relevant option is being able to program and/or reprogram parts in the system (i.e., ISP).

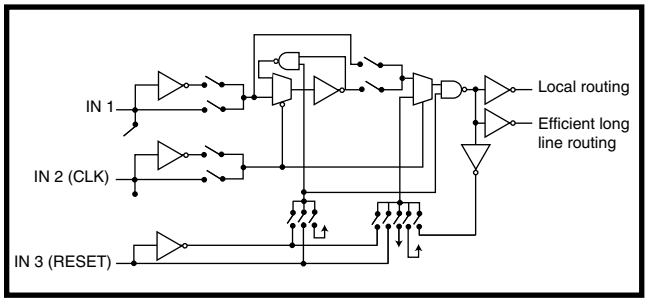


Figure 3—*The tiles in Actel's flash memory FPGAs, handling any combinatorial function of three inputs (except XOR), are simpler than the logic blocks found on SRAM LUT-based FPGAs.*

The main prerequisite for easy ISP is a programming scheme that uses relatively few and unshared pins. It's possible to use brute force to get around a lot of shared pins by adding a bunch of external logic, but it's not a pretty sight. ISP provides obvious benefits on the production line, and is helpful in the lab as well.

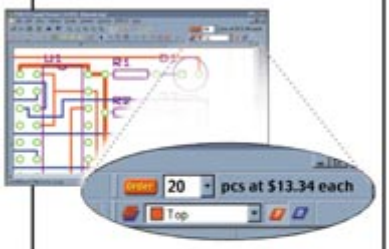
A subtle, but in some cases critical, twist is the ability to program chips "in operation" (call it IOP). Typical ISP schemes mandate the system be reset or otherwise go off-line while the part is programmed. In some highly reliable or "automatic update" applications, having to cease normal operation during programming can be problematic. IOP extends the advantages of ISP from the production line to the installed base, specifically as it relates to post-purchase bug fixes and upgrades.

Partial reconfiguration is an extension of the IOP concept in which the device is, possibly frequently, all or partially updated in response to instantaneous application requirements. If IOP allows a part to be programmed in operation, partial reconfiguration encourages and demands it. Key factors include the speed of updating and the degree to which it's easy to change part of the functionality without affecting the rest. The advantage is less about streamlining production or fixing bugs than potentially achieving superior application price/performance compared a traditional fixed-function chips.

By understanding the embellished definition, it can be seen that antifuse, SRAM, and now flash memory-based parts may all share the FPGA moniker, but in fact are quite different from each other in ways that fundamentally effect the proper choice for a particular design-in.

In particular, Actel's flash memory parts are in-system reprogrammable via their JTAG interface, falling short of SRAM-based parts only in terms of in-operation reprogrammability (16.5 V and

Ordering circuit boards from PCB123...



...begins with free design software that features real-time pricing as you design.

PCB123

Visit us today at www.pcb123.com

The complete solution – design through order – for quick, low-cost circuit boards in small quantities.

USB DAQ



LabJack™ U12

$99

* 12-bit analog inputs
* 10-bit analog outputs
* 20 digital I/O
* 32-bit counter
* Watchdog function
* Built-in screw terminals
* Easy-to-use USB
* Everything included
* Use with C, VB, LabVIEW
* Windows 98SE/ME/2000/XP

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

www.labjack.com

JUMP START YOUR USB DESIGNS



New Edition Includes USB 2.0

"This is the best book on the topic. I recommend it highly." Jon Titus, *Test & Measurement World*

USB Complete

Everything You Need to Develop Custom USB Peripherals Second Edition, by Jan Axelson

Lakeview Research 523 pages
ISBN 0-9650819-5-8 $49.95

www.Lvr.com/usb.htm

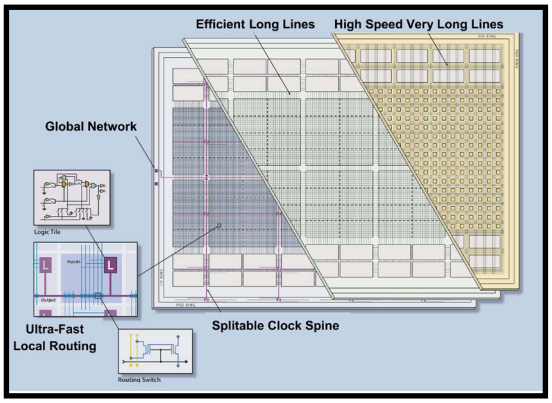


Figure 4—*A four-tier interconnect hierarchy boosts routing capability, which facilitates integration with conventional ASIC toolchains.*

–12 V required for programming, 100 programming cycles max) and the rarely used partial reconfiguration.

GA-GA OVER GATES

We've talked about the "F" and the "P." It turns out that even the "GA" part of the acronym hides significant differences between the parts.

The SRAM FPGAs from Altera and Xilinx are based on complex logic blocks comprised of multi-input look-up tables (LUTs). These are coarse-grained versus the simple gates associated with traditional ASIC gate arrays. By contrast, the Actel Pro-ASIC flash memory parts rely on a finer-grain element known as a tile (see Figure 3). Between the ProASIC parts and the new ProASIC+ series, there are devices with 5000 to 50,000 tiles.

For the most part, it's tempting to write off any cell-versus-tile difference as semantic quibbling along the lines of the old RISC-versus-CISC rhetoric. As mentioned earlier, most designers are just interested in punching in a schematic or some HDL and getting a chip that works without getting bogged down in the details.

However, there may well be some merit to Actel's assertion that their finer-grained design is more amenable to use with traditional ASIC (i.e., true gate array) toolchains. While everybody's parts work with the popular FPGA packages (e.g., Synplicity), Actel offers an ostensibly easy (or at least easier) migration path for people who are moving from or to the ASIC option.

The finer grained logic strategy puts more stress on routing. To that end, the ProASIC parts utilize a four-layer metal process that serves a four-level hierarchy of high-speed local (adjacent tile) nets to global very long lines (see Figure 4). Special attention is paid to clock distribution, especially on the ProASIC+ series, which incorporates two integrated 1.5- to 240-MHz PLLs.

One potential advantage for LUT-based parts is that it's easy to use LUTs as memory because that's in fact what they are. However, that's an expensive memory option, which is why even LUT-based parts now include dedicated block RAM.

Because ProASIC tiles can't realistically perform such double logic or memory duty, the Actel parts include dedicated RAM in 256 × 9 blocks, from six at the low end to 88 at the top of the line. The blocks can be combined to make wider and deeper memories.

Better yet, each block can be configured as either a regular two-port (one read and one write) SRAM or as a FIFO with programmable threshold. Parity protection (read, write, or both) is an option as well.

FLASH FORWARD

Designers who aren't locked in to Xilinx or Altera should really give the Actel Pro-ASIC parts a close look. There are a number of major advantages over the SRAM-based parts.

Most obvious is eliminating the need for the external boot memory required to initialize the SRAM at power-up. Unless a boot mechanism is a built-in byproduct of the design (e.g., a PC plug-in) this is an easily overlooked, but nontrivial, cost of doing business with SRAM FPGAs.

The need for booting externally also represents a security compromise for SRAM-based parts. It's all too easy for a would-be "cloner" to capture the bitstream information and copy your design. To counter, some SRAM FPGAs are moving to encrypted bit streams with an on-chip key requiring external battery back-up, but that approach comes with it's own set of baggage.

By contrast, the Actel flash memory parts eliminate the need for an external boot source, are live at power-up, and fully secure using a straightforward scheme much like those found on flash-memory MCUs.

There's also reason to believe flash memory-based parts can offer better price as the simpler memory cell structure offsets the ever-declining process complexity differential. Actel claims a 7:1 cost advantage for their flash memory cell versus the competitors' SRAM, though it isn't clear whether or not that comparison is entirely fair considering different processes and any overhead circuits.

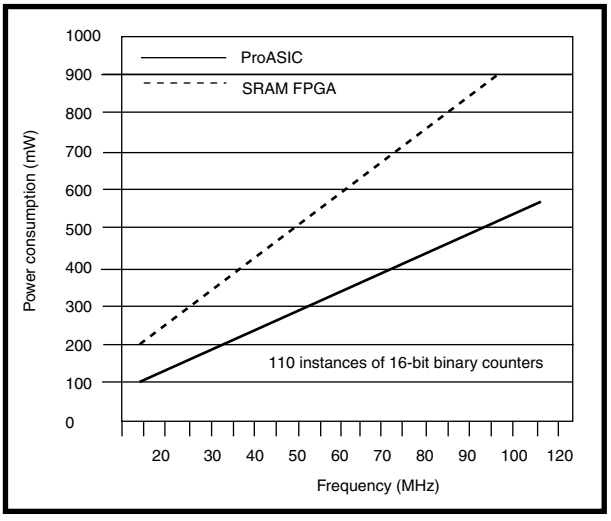


Figure 5—*Power consumption will become an increasing concern as FPGAs move into mainstream designs.*

Anyway, you don't have to fall for their pitch entirely to get the point. Just look at the mainstream memory chip market where flash memory chip pricing is more like DRAM than the more expensive SRAM. Long-term, the trend will likely continue as flash memory permeates more and more chips, both logic and standalone memory, while SRAM seems increasingly relegated to niche applications.

However, keep in mind the street price doesn't always reflect the actual manufacturing cost at any moment in time. Furthermore, tactical marketing machinations can come into play. I suspect Actel may have to do some juggling relative to the pricing of their existing antifuse-based lineup.

Power consumption is another area of pragmatic concern. I'm not sure to what degree it's simply a result of different memory technology, but Actel's flash memory power advantage claims are backed up with some seemingly credible comparisons against SRAM-based parts. Take a look for yourself in Figure 5.

Indeed, the only weakness for the Actel Pro-ASIC lineup relative to Xilinx and Altera at this point is at the entry level. Actel has low-end antifuse parts (ACT-1 and ACT-2), but their smallest flash memory chips have a lot more gates (100K) and pins (208) than the entry-level devices from Xilinx and Altera. This gap is presumably easy enough for Actel to address and I encourage them to do so.

Bottom line? Just as with memory and micros, I have little doubt that flash memory is the wave of the FPGA future. Xilinx and Altera will be able to rest on their SRAM-based laurels and exploit the inertia of their historic leadership for quite some time. But I suggest that if they aren't already cooking up something flash memory-based in the lab, they'd better get cracking.

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@circuitcelllar.com.

SOURCES

Flash memory and antifuse FPGAs
Actel Corp.
(408) 739-1010
www.actel.com

SRAM FPGAs
Altera Corp.
(408) 544-7000
www.altera.com

Xilinx, Inc.
(408) 559-7778
www.xilinx.com

All current articles available on an individual basis at www.circuitcellar.com

NO MORE

Dog-eared, Missing, Ripped, Written On, Highlighted, Coffee Stained Pages!

Don't surrender your issue ever again. Every article in this issue of *Circuit Cellar* is available for download. Any time of day you have instant access to single articles. Take no more excuses, tell them where to get their own copy.
www.circuitcellar.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 kc@circuitcellar.com. For more information call Sean Donnelly at (860) 872-3064.

Suppliers Directory now available online. Check out our web site www.circuitcellar.com to see who has what and how to get it!

PICmicro® BOOK

Serial PIC'n

- Asynchronous serial communications
  - -Bit-bang
  - -On-chip Uart
  - -RS-232 explained
- Synchronous serial communications
  - -I2C (Philips Semiconductors)
  - -SPI (Motorola)
  - -Microwire (National Semiconductor)
- Dallas Semiconductor 1-Wire bus
- Many code examples - Use modules in your own applications

Table Of Contents On Web Site

PIC and PICmicro are trademarks of Microchip Technology, Inc.

SQUARE 1 ELECTRONICS

P.O. Box 501, Kelseyville, CA 95451
Voice (707) 279-8881 FAX (707) 279-8883

http://www.sq-1.com

BUY! SELL!

COMPONENT BAZAAR

A GRASS ROOTS SOLUTION TO SEMICONDUCTOR PART AVAILABILITY FOR PROTOTYPING AND SHORT RUN PRODUCTION

WWW.COMPONENTBAZAAR.COM

ALLIED COMPONENT WORKS

80C51
166 · ARM · DSP
TriCore · 80186
80196 · Z8

in circuit emulators

CAN TCP/IP Flash

... because your project deserves the best!

89C51RD2
C505CA
89C51CC01

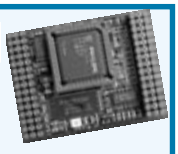
emulators
debuggers
compilers

SIGNUM SYSTEMS

1-800-838-8012 www.signum.com

PHYTEC

phyCORE Modules:
NEW GENERATION
SINGLE BOARD COMPUTERS



- Insert-ready sub-mini SBCs (small as 47x55 mm.) supporting the Philips 8xC591, 89C51Rx2, XACx, XAGx, Infineon C167Cx, Motorola MPC555 & ST Microelectronic ST10F168
- Low EMI design achieved via GND circuitry, 6 to 8 layer PCB, by-pass capacitor grid and short signal traces achieved via small footprint and use of 0402 SMD passive components
- 32 KB to 8 MB external SRAM & Flash (controller-dependent)
- FlashTools enable on-board in-system (ISP) programming
- RS-232, RS-485, I²C & CAN interfaces; ADC; Chip-Select signals
- Controller signals extend to standard (2.54 mm.) or high-density Molex (0.625 mm.) header pins on two sides of the board, allowing the SBC to be plugged like a "big chip" into targets
- Available in Rapid Development Kits including Development Board, AC adapter, serial cable and SPECTRUM CD with eval software tools (Keil, TASKING), FlashTools, electronic documentation and demos

PHYTEC America LLC ■ 255 Ericksen Avenue ■ Bainbridge Island, WA ■ USA 98110
www.phytec.com (800) 278-9913

PRINT to VIDEO!

Tiny BOB-II module superimposes up to 308 characters on NTSC/PAL video or generates video automatically. Fast 2.4~19.2kbps RS-232 serial interface. Simple to control; like a printer. Many powerful applications:

Home Automation - MATV
Video Inspection & Testing
Surveillance - CCTV - ATV
Remotely Piloted Vehicles
Gaming - Racing - Sports
Process/Experiment Monitor
Robotics - Electronic Signs

BOB-II-NTSC only $79.95

Information & Ordering: www.decadenet.com
DECADE ENGINEERING
5504 ValView Dr. SE, Turner, OR 97392
Tel: 503.743.3194 ~ Fax: 503.743.2095

IDE, Compilers, ICD...
Emulators
...Simulators, Programmers

8051
AVR®
PIC®
XEMICS
SENSORY
80196

Hi-End Professional Tools
Integrated Development Environment
Affordable Prices

Phyton 718.259.3191 www.phyton.com

PICMICRO® TOOLS



ISP-PRO 3.0

All-in-one Programmer

- Uses PC Serial port (USB w/ adapter)
- Very Simple to use
- Free Software updates
- Complete with Windows IDE!
- Easy in-circuit programming
- Supports PIC, Scenix, I2C and more!
- Firmware upgradable!
- RJ-11 / 10 Pin Header
- Includes RJ-11 Cable

Optional Items Available:
- Plastic Enclosure Available
- 40 Pin ZIF Adapter All-in-one
- Universal Adapter

All for $59.95 Complete

Solderless Development Boards

Develop your projects the easy way



- Completely Assembled Board
- In Circuit Programmable (ISP)
- Solderless Bread Board
- Built in RS232 (w/ Max232)
- Built in Power Connector
- Removable Oscillator
- Documentation
- Auto Disconnect for RB3, RB6, & RB7
- Available in several models

Starting at $49.95

Join our online PIC forums
tons of information and help *FREE!!!*

VISA • Master Card • American Express
Order Online or Call **1-734-425-1744**
Basic Micro, 34391 Plymouth, Livonia, MI 48150
See more online at http://www.basicmicro.com

PC KEYBOARD EMULATION

Interface Keypads, Switches, or RS-232 to your PC Keyboard Input

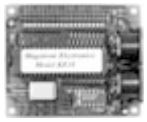
MODEL KE24 ONLY $99.95



• Up to 12 x 12 matrix • Programmable • RS-232 Port • Macro Capability

The KE24 is the ultimate in flexibility. Inputs or serial data can emulate any of the 101 keys from a standard keyboard.

MODEL KE18 ONLY $44.95



• Up to 9 x 9 matrix • 2.5" x 3.0" Size • PC Keyboard Port • PCXT, AT Compatible

The KE18 combines a multitude of features with small size at an economical price. Custom units available.

HAGSTROM ELECTRONICS
11 Fiddlers Green
Lansing, NY 14882

Toll Free: 888-690-9080
Phone: 607-533-4441 Fax: 607-533-4443
www.hagstromelectronics.com

USB FAST!



RomEm®
ROM Emulator

- Downloads 8Mbit in 1.5 Seconds
- GUI or Console Interface
- Dependable and Reliable
- $675.00 8Mbit

details at:
www.RomEm.com

MicroSystems Development, Inc.
1177 Park Ave, San Jose CA 95126
V: (408)280-1226 ext 20 F: (408)280-6868



ATLANTIC QUALITY DESIGN, INC.

*Electronic Product R&D for companies on a **budget** and **schedule***

"Diligent." "Professional." "Expert." "Committed." "Under budget." That's how our customers describe us. See our web site, then call to discuss your project.

Free Initial Consultation
(540) 966-4356
WWW.AQDI.COM

All-In-One Embedded Solution
Every feature required
In a Compact Platform



FB6500 GX-1 300Mhz Embedded 64MB,VGA/LCD,LAN,Audio,USB 4 COM, 1 P all on board.
All For $294 at qty.1
127x127mm Run Linux and WinCE !!

FB2330



3.5" 386 Embedded PC w/ 4MB RAM VGA,LAN, 4S/1P all on board upto 288MB DiskOnchip support
$193.20 at qty. 1



FB2310
PC/104 386 CPU Module w/4MB RAM 2S1P, IDE,DiskOnChip support, Bootable Flash Disk Onboard. *$119* at Qty. 1

FB4617 Compact Flash Adapter
Add bootable embedded Mass Storage to any sytems w/ IDE Bus, Interface to CompactFlash Reliable, Low cost, No special s/w needed. $40 Qty 1
Please visit our site for more details.

Find More Pentium, Socket 370 and 386 boards (Embedded, PC/104,ISA & PCI) in www.abiatech.com

ABIA TECHNOLOGY
714-821-8762 FAX: 714-952-3306
www.abiatech.com Email:info@abiatech.com
**** Custome Design and OEM Welcome ****

I2C, FLASH, CAN, and XA Microcontroller Development Kits



PC-ISP-FLASH $199.00

PC-ISP-FLASH programs 5V ISP/IAP FLASH microcontrollers from Philips and Atmel. Includes 44-pin PLCC programming socket with other packages supported by vendor specific Family Adapter Boards. Also includes target interface cable to ISP program devices on target board. Supports ISP FLASH 80C51, XA, & AVR devices. The PC-ISP-FLASH software runs under Windows 9x/NT4.0/2000. The programmer allows easy user upgrades to the list of supported ISP/IAP FLASH microcontrollers.



PC-I2C-KIT $149.00

PC-I2C-KIT lets your PC "talk" I2C and makes it easy to develop I2C-based systems. The PC-I2C Adapter interfaces to any PC and allows bi-directional communications with I2C-based peripherals. The PC-I2C software runs on Windows 9x/NT4.0/2000 and uses simple menus to select I2C peripherals from the list of supported devices. A "universal mode" also supports any I2C device at any address mapping. PC-I2C also supports on-board programming of I2C EEPROMs.



XTEND Kits

***XTEND*, XA Trainer and Expandable Narrative Design**, provides the user with a Quick Start kit for any Philips XA Microcontroller design. Use the *XTEND* as your actual prototype or just use it for debug until your XA PCB is ready. All XTEND kits include XA Development Board with on-board Monitor and Debugger, Software, Documentation, Schematics, Power Supply, and Serial Cable. Volume Pricing available.
$249.00 for *XTEND-Gxx (G3 & G49)* or *XTEND-Sxx (S3)*
$299.00 for *XTEND-C3* or *XTEND-H4*,
$449.00 for *XTEND-SCC*

FDI
Future Designs, Inc. http://www.teamfdi.com
11521 Gilleland Rd, Suite A, Huntsville, AL 35803
(256) 883-1240 Fax (256) 883-1241 email: sales@teamfdi.com

DEVICE PROGRAMMER, USB ROM Emulator, EPROM ERASER

TopMax **ChipMax**



$995 $345

TopMax : Programs over 7000 devices, 48 Pin ZIF Socket ,Tests TTL/CMOS Logic IC, Low voltage, On-board processor and built-in Universal AC, 9X/NT/2000/XP,CSA, UL, Made in USA, Lifetime free software, 2 year warranty, Free software update for life time

TM-8G - DP	8-gang (32-pin DIP) adapter for E(E)PROM	$325
TM-8G - PL	8-gang (32-pin PLCC) adapter for E(E)PROM	$495

ChipMax : Reliable and low price ,Programs over 4000 devices, EPROM, FLASH Memory, Microcontrollers, PIC, Serial EPROM, EPLD, GAL, PALCE, 40 Pin ZIF Socket, Parallel, 9X/NT/2000/XP, Made in USA ,1 year warranty, Lifetime free software, 110 or 240 AC Adapter. CM-4G: 4-gang adapter 32-DIP $115

EeRom-8U
$495
"USB Interface"



EeRom-8U : 8Mbit-45ns EPROM/Flash Emulator, High speedy USB port interface (Plug & Play)
- Fast download (8Mbit per second), Lower Voltage, 8bit, 16bit, and 32bit data split
- High Speed access time (built-in 8M SRAM 15ns)
- No external power is needed, 9x/XP/NT/2000
- Accepts all standard Hex formats and binary files
- Jumper less Emulation size configuration



LA6T-110
$115



LA6T-110 : UV ERASER for Personal user, Fast and efficient, Molded Plastic Case, Hold 9 of 28-pin or 6 of 40-pin EPROM. Timer(15-40 min),
Programming Adapters(low price) : 8-SOIC($70), 32-PLCC($70), 44-PLCC($115), 48-TSOP($135), 44-QFP($65), 52-PLCC($165), 68-PLCC($165) ,
EPROM Emulator Adapter, 32/28-pin($95) 44-pin($115) DIP-to-PLCC

EE TOOLS
E.E. Tools, Inc.
549 Weddell Drive
Sunnyvale, CA 94089
www.eetools.com
Tel. 408-734-8184
Fax. 408-734-8185
VISA MasterCard

Give your PC a high speed CAN connection
CAN-4-USB
(Full PC source code included)

If we can provide all the HW and SW for a 4 wheel drive, WALKING, hydraulic vehicle



Photo Credit DRES

Imagine what we can (CAN?) build for you!

Zanthic Technologies Inc.
Phone 403-526-8318
email: zanthic@zanthic.com
web: www.zanthic.com

PIC Programmers

Super Value!



from $16.95 +S&H $4.95*

The P16PRO: can program up to 40 pin PICs including the popular **16F84** & **12C508** ● Needs software (extra $20) ● Available assembled or starting from **$16.95** for the kit. See **www.electronics123.com** for more info!

The **PICALL** programmer can also progam **Atmel AVRs** in addition to the PICs it can program ● Free software ● PICALL programmer kit at **$69.95** See **www.electronics123.com** for more info!

Digital Camera module



1/4" CMOS Camera Module, black & white, with digital output. Size: only 1.38"x1.14". Lens: f7.4mm, F2.1 FOV 32 deg x 24 deg. Pixels: 384x288. Scanning: progressive. Video output: digital 8bit. Programmable features. Shutter: 1/50 to 1/5000 sec. Power Supply. 5VDC. Standby Current <10mA. Needs regulated power supply. *S&H to Canada is $7.95

$57.95 +S&H $4.95* Code: CM4881

Toll Free: 1-888-549-3749 (USA & Canada)
Tel: (330) 549-3726. Request a FREE catalog or visit us at: www.electronics123.com for more.
Amazon Electronics, Box 21 Columbiana OH 44408

RS485/422/232/TTL

ASC24T $45



Converters
Repeaters
Fiber Optics
Digital I/O
Multidrop RS232
Custom Units
Auto TX Enable

Extensive Interface Product Line
RS232 "Extension Cords"
Up to 115.2 Kbps, 4000 ft.++
Large Multidrop Networks.
Isolated Units. Smart Units.
Remote Relay "Extension Cords"
Call the RS485 Wizards at:
(513) 874-4796

RES R.E.Smith
www.rs485.com

New!
ActiveWire™ USB Simple USB Interface

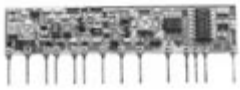


- Internet Browser Script-able
- 24 MHz CPU core with USB
- Firmware downloadable via USB
- 16 bit parallel I/O
- Expandable add-on boards
- New firmware and scripts available from website

$59 plus shipping
ActiveWire, Inc.
www.activewireinc.com
ph(650) 493-8700 fx(650) 493-2200

RF Modules with intelligence

AM-RTD-315 Transceiver Module



Low Cost AM Transceiver Module
315MHz operation
CMOS/TTL compatible i/o
Up to 10Kbps data rates
Range up to 300ft
5V, 10mA (TX), 3mA(RX)
Low power Mode: 1mA
2.5" x 0.6" x 0.15"
0.1" pin spacing

AM-RTD-315.....US$24.09ea.

www.abacom-technologies.com

ABACOM Technologies
Tel: +1(416)236 3858
Fax: +1(416)236 8866
abacom@abacom-tech.com

RF Modules with intelligence

DPC-64-CTL Transceiver Modules



-CTL



-RS232

Crystal Controlled PLL Design
5V, <15mA.....FM Operation
433.92, 868, 914.5MHz versions
Embedded Data Packet Controller
Automatic Data Encoding/Decoding
Automatic Error Checking
9600,8,N,1 protocol compatible
Half Duplex Operation
Up to 700ft range
CMOS/TTL compatible
Automatic RX/TX switching
RSSI output (on - SS version!)
'Test Link' diagnostic mode
Carrier Detect Output
Small Size: 1.3" x 0.9" x 0.5"

DPC-64-CTL.....US$159.00ea.
DPC-64-RS232.....US$178.00ea.

www.abacom-technologies.com

ABACOM Technologies
Tel: +1(416)236 3858
Fax: +1(416)236 8866
abacom@abacom-tech.com

Migrate to an Internet-enabled Modem for just $39.95*

Cermetek's New CH2124



Send or Receive Remote System Data:
- Data, system status files, diagnostics.
- Alarm / Alerts, e-mail messages, OS / Firmware Uploads.
- FTP and dynamic web page transfers.

www.cermetek.com

Complete 2400 bps or 56K / V.90 Modem with built-in TCP/IP Solution:
- Space-saving dual-in-line module.
- FCC Part 68 pre-approved, UL1950 listed.
- Easy to configure/program with iNetWizard® set-up.
- Engineering evaluation kits available.
- Low cost $2.00 / month ISP access / hosting service.

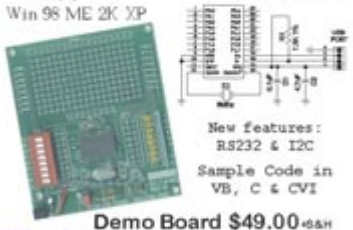
* CH2124 (2400 bps) price is $ 39.95 for 1,000 pce. lots.

Tel: 800-882-6271
Sunnyvale, CA USA
Cermetek microelectronics

CypressM PSoC
HC08,11,12,16
tinyAVR
AVR
IMAGEcraft™
Fast, easy to use, low cost professional ANSI C development tools
Check it out! Version 6 Tools
Full-featured 30-day demos at:
www.imagecraft.com
Standard Toolset = $199
...Why pay $2K-$5K elsewhere?
VISA / MC / AMEX
Check / Money Order
...say 'NO' to JURASSIC PRICES!
info@imagecraft.com
706 Colorado Ave. Palo Alto, CA 94303
(650) 493-9326 • FAX (650) 493-9329

EVBplus2 68HC11

* development system, trainer, and prototype station
* motorola HC11 EVB, EVM and EVS replacement
* symbolic debugging and windows IDE
* all RAM $00-$1ff & user code $8800-$ffff available
* phantom monitor™ preserves user RESET vector
* exit user program with Esc key when it's hung
* 68HC24 PRU included, Dual UARTs
* IR transceiver, RS485, LCD, LED display and more
* robot servo outputs, 12V programming voltage
SUPER LOW COST www.EVBplus.com

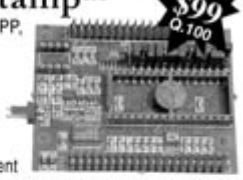
USB for 8 Bucks
Add USB to your next project
16 bit I/O USB Chip
Preprogrammed USB chip for $8ea. ($6 in 100qty)
USB
Supports
Win 98 ME 2K XP
Minimum parts required
1 IC 1 Xtal 1 Res & 2 Caps

New features:
RS232 & I2C
Sample Code in
VB, C & CVI
Demo Board $49.00 +S&H
Delcom Engineering
(914) 967-8840
(914) 967-7962 Fax
Order Online
WWW.DELCOM-ENG.COM

okwelectronics.com
In-Circuit Emulators
ICEPIC JUNIOR 5X
PIC12C508/509
PIC16C52/54/55/56/57/58

$375
• Our lowest cost, full speed real time emulator •
• Source level debug in assembler or 'C' •
• Runs under MPLAB, Win 95/98/NT •
• High speed RS232 interface included •
• Single/Procedure step, Step Out/Over •
• Unlimited number of break points •
More info. @ www.okwelectronics.com
OKW Electronics, Inc.
Tel: (800) 965 9872 Fax: (412) 220 9247
300 Old Pond Road, #203, Bridgeville, PA 15017
Email: sales@okwelectronics.com

okwelectronics.com
PIC Project Boards
Millennium Starter Kit
For engineers and students

$195
Part# MIL-BOARD
Useful project tool for experienced engineers and learning platform for students.
The Millennium Starter Kit contains all the necessary components to start learning the intricacies of Microchip's PIC microcontrollers.
More info. @ www.okwelectronics.com
OKW Electronics, Inc.
Tel: (800) 965 9872 Fax: (412) 220 9247
300 Old Pond Road, #203, Bridgeville, PA 15017
Email: sales@okwelectronics.com

EMBEDDED INTERNET MADE EASY! with the
DOS Stamp™
From $99 Q.100
• Free TCP/IP, PPP, Mini-server
• Easy to Use
• Low Cost
• Low Power
• Tiny Size
• High Speed
• DOS Environment

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 AM188ES CPU, 16 digital I/O (opto rack interface), 2 RS-232, 2 timer/counters, simple bus interface, real-time clock with timed power-up
Options: 8-ch 12-bit ADC, flash disks up to 288 MB, 1 RS-485
Tiny Size, Low Power: 2"x2.6", 5V @ 200 mA at full speed.
Visit http://www.bagotronix.com for info, prices, and FAQs
BAGOTRONIX
2900-1 Crescent Drive
Tallahassee, FL 32308
850-942-7905 phone & fax

PROBLEM SOLVED!
Custom Hardware & Software Design

• Rapid Prototypes
• Cost-effective Design
• Free Consultation and Quotes
MJS CONSULTING INC.
Mark J. Samarin M.S.E.E.
http://www.mjsconsulting.com
916-967-6275

SBC/Operator Interface with Ethernet • 24 I/O
OP6800 MiniCom

Actual Size 3.6" x 4.5"
$249 qty.1
$199 without Ethernet
• 10Base-T Ethernet •
• C-programmable •
• 24 industrialized I/O •
• 122 x 32 backlit LCD •
• 2 RS-232 / 1 RS-485 ports •
• 256K Flash/128K SRAM (standard) •
• Meets NEMA 4 water resistance •
The MiniCom gives OEMs complete design flexibility—a C-programmable operator interface and single-board computer in a compact package with network connectivity! Program with industry-proven Dynamic C (extensive TCP/IP libraries and sample programs included).
ORDER ONLINE @
www.zworld.com
or CALL TOLL-FREE
888.362.3387
Z-WORLD
2900 Spafford Street • Davis, CA 95616
T: 530.757.3737 • F: 530.757.3792 • E: zworld@zworld.com

CALL 1•800•A•Dreamer LOCAL & INTERNATIONAL 1•510•353•1800 VISIT www.DreamTech.com



Starting at

Dream P4
POWERSTATION
1.6GHz
ITEM #3438
$529

- 1.6GHz Intel Pentium 4 Processor
- Intel 845GBVL Main Board
- 256MB PC/2100 DDR Memory
- 40GB 7200RPM Hard Disk Drive
- 3.5" 1.44MB Floppy Drive
- 52X CD-ROM
- Integrated Audio, Video & Lan
- 107 Enhanced Keyboard
- P4 Mid Tower Chassis & ATX PS

Dream P4 845S
P4 with Intel 845WN MB 32MB VGA & sound. OS ready!
1.9GHz
ITEM #3318
$599

POWERFUL PENTIUM 4 ~ WITH INTEL 845WN MAIN BOARD & FAST PC/133 MEMORY! VALUE PRICED WITH 256MB RAM, 20GB HD, MOUSE, SOUND & MODEM!

- 1.9GHz Intel Pentium 4 Processor
- 256MB PC/133 SDRAM Memory
- 20GB 7200RPM Hard Drive
- 3.5" 1.44MB Floppy Drive
- 52X CD-ROM
- 1 Parallel, 1 Serial, 4 USB & PS/2 Port
- ATI Xpert 2000 32MB AGP Video Card
- Sound Card & 120 WATT Speakers
- Logitech PS/2 Mouse
- 107 Enhanced Internet Keyboard
- Netgear 10/100 PCI Ethernet Card
- P4 Mid Tower Chassis & ATX PS

Dream P4 Entertainer
with DVD & WinXP Home
2.0GHz
ITEM #3409
$799

BEST BUY! INTEL 845BG MAIN BOARD WITH INTEGRATED SOUND, 60GB HD & 256MB RAM. AVAILABLE UP TO 2.2GHz!

- 2.0GHz Intel Pentium 4 Processor
- Intel 845BG Main Board
- 256MB PC2100 DDR Memory
- 60GB 7200RPM ATA 100 Hard Drive
- 3.5" 1.44MB Floppy Drive
- 16X DVD CD-ROM
- 1 Parallel, 1 Serial, 4 USB ports
- 64MB AGP GeForce Video Adapter
- P4 Mid Tower Chassis & ATX PS
- Integrated Sound & 120W Speakers
- 107 Enhanced Keyboard
- Lucent 56k v.90 PCI Modem
- Windows XP Home & Optical Mouse

Dream Profile
with 850MV & Windows 2000 Pro
2.2GHz
ITEM #3445
$1079

FAST & RELIABLE WITH RAMBUS MEMORY, WINDOWS 2000 PROFESSIONAL & MORE!

- 2.2GHz Intel Pentium 4 Processor
- Intel 850MV Main Board
- 256MB PC/800 RAMBUS Memory
- 80GB 7200RPM ATA 100 Hard Drive
- 3.5" 1.44MB Floppy Drive
- Plextor 40x/16x/10x CDRW
- P4 Mid Tower Chassis & ATX PS
- 1 Parallel, 1 Serial, 4 USB ports
- 64MB AGP GeForce Video Adapter
- Integrated Sound & 107 Keyboard
- Netgear 10/100 PCI Ethernet Card
- Windows 2000 Pro & Logitech Mouse

DreamTech
QUALITY COMPUTERS SINCE 1983
Email: Sales@DreamTech.com
Tel: 510.353.1800 Fax: 510.353.0990
40950 Encyclopedia Cr. ~ Fremont, CA 94538
© August 2002

SEALEVEL®
INNOVATIVE I/O SOLUTIONS



LOW PROFILE PCI
COMM+850.PCI
- One Port RS-232
- 16850 UART provides 128 byte FIFO
- Data rate up to 460.8K bps
- Universal PCI connector, 3.3V or 5V
- MD1 Low Profile board

LIFETIME WARRANTY
SEALEVEL
864.843.4343
Made in USA
©2002 Sealevel Systems, Incorporated
View our 2002 catalog at www.sealevel.com or email us at info@sealevel.com

SmartLCD-Color™
C/C++ programmable controller with SED1375
Qty 1 $299



- 6.5 x 4.3 inches, 19 ADCs, 6 DACs, RTC
- 60+ TTL I/O, 6 timers, 21 solenoid drivers
- Color, QVGA LCD, touch screen and bezel
- 5 serial ports, CompactFlash, file system, Ethernet.

40+ Low Cost Controllers with ADC, DAC, solenoid drivers, relays, file system with CompactFlash, LCD, DSP motion control, 11 UARTs, 300 I/Os. 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

Voice Extreme™ Toolkit
Includes all software and hardware to create your own voice recognition products!



$129
SENSORY
408.327.9000
www.VoiceActivation.com
1991 Russell Ave. Santa Clara, CA 95054

SMART 'FLEXIBLE I/O' BOARDS
Designed specifically for time critical applications.



- PC-104+ / PCI / USB
- 100+ User I/O
- User Configurable Altera SRAM PLD for Communication and Signal Processing
- External Clock / Sync
- Daughterboard or Ribbon Cable Interface
- Real-Time HW & SW Interface
- Standard and Embedded Operating System Support
- Integration & Daughterboard Design Services Available

SIGNAL PROCESSING FOR THE REAL-TIME WORLD ...
VISIT US AT:
www.rpaelectronics.com
OR CONTACT US AT:
RPA ELECTRONICS DESIGN, LLC
607-771-0393 Phone
607-771-0658 FAX
sales@rpaelectronics.com

Windows® CE
Embedded Controllers
Touch Screen
6.5" VGA Color LCD
640x480 TFT / 64k Colors
Microsoft Visual Basic & C++
Embedded Tools Supported
Give Your Product A New & Better Look
Easy To Program
Easy to Interface
Visit Our Web Site For More Detailed Information & Pricing
StrongARM
* Solid State FLASH Disk
* Serial Ports, Parallel I/O
* PCMCIA Card Slots
* USB, PS/2, Audio
* Analog I/O
* Custom I/O Supported
RLC www.RLC.com
R.L.C. Enterprises, Inc.
Toll Free 1-888-RLC-TECH

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

Rapid PCI Development
PCI Interface Chipset:
• 128 KByte Read & Write FIFO's
• 132 KByte/sec DMA Master
• FIFO Style Interface to FPGA
• 24 Bits Programmable I/O
Choice of FPGA:
• Xilinx XC2S200 or Altera EP1K30.
• Design Your Application with Free HDL Software from Xilinx or Altera.
Driver Software & API Included:
• Access DMA Data with C/C++ Stream Functions
• C/C++ Functions to Control 24 I/O Bits
Complete Prototyping Tools:
• 24 I/O Pins Available from FPGA
• LED's, Switches, Numeric Display
• Removable Wirewrap Area
• Connectors to Attach Custom PCB's
• Documentation & Sample Apps.
Now Only $395
PCI ProtoBoard
TECHNIPRISE INC. (614) 431-5610
www.pciprotoboard.com

PRINTED CIRCUIT BOARDS
QUALITY PRODUCT
FAST DELIVERY
COMPETITIVE PRICING
PROTOTYPE through PRODUCTION
• UL approved
• Single & Double sided
• SMOBC
• 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 • Modem 847.233.0014 • www.flash.net/~yogii • yogii@flash.net

FREE RTOS
Evaluation Kits
• Based on SMX Modular RTOS
• Available for popular boards
• Popular tool chains supported
• ARM, ColdFire, PowerPC, x86 16- & 32-bit, SH3/4
• Hard real-time multitasking
• TCP/IP, web srv, file mgr, GUI
www.smxrtos.com/evalcc.htm
Micro Digital Inc
RTOS Innovators
800.366.2491 sales@smxinfo.com

EMBEDDED INTERNET
Prototyping Board with the CCS C compiler and libraries
Complete hardware kit with 56K modem and examples using CCS C source code on a PICmicro® MCU. Examples include an embedded e-mail program and a web server.
Internet Prototyping Board includes:
PIC16F877 Flash Chip
ICD Interface
Full V.90 Modem
RS232 Port
Push Buttons
Analog Inputs
Schematics
PC Cable
Power Supply
Phone Cable
Example Programs
Our limited time offer also includes:
Debugger Unit (ICD)
CCS PCW Windows IDE Compiler
TCP/IP Lean book, by Jeremy Bentham
ALL FOR ONLY $ 499!
(OVER 20% IN SAVINGS)
Use discount code IPB0702-CC to order... ACT NOW!
PICmicro® MCU
CCS C COMPILER
A high order language with extraordinary optimization specifically designed for the PICmicro® MCU.
Access to all PICmicro hardware features form C
Built-in libraries that work with all chips for RS232 Serial I/O, I²C, discrete I/O and precision delays
Source code drivers and ready to run programs included for LCD modules, keypads, 24xx and 93xx serial E² PROMS, X10, DS1302 & NJU6355 Real Time Clocks, Dallas Touch Memory Devices, DS2223 & PCF 8570 serial SRAM, LTC1298 & PCF8591 A/D converters and more
Integrates with simulators/emulators, such as MPLAB®.
PRICES STARTING AT:
COMMAND-LINE COMPILERS FOR $125
WINDOWS IDE COMPILERS FOR $350
http://www.ccsinfo.com/picc
262.797.0455 x35

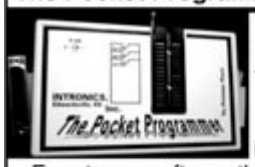
RAD PROTO

Bring your dream to life

PSoC PIC AVR RABBIT MSP430
Rapid Prototyping Service
Prototyping Support Products
Embedded Engineering Services
Wireless Systems Specialists
Embedded Networking Specialists
Programming in C - ASM - FORTH
Data Collection Specialists
Data Logging Products
Quality.Timely.Affordable
Certified PSoC Consultant

WWW.RADProto.COM
Denver, Colorado, 720-940-5002
sales@radproto.com

The Pocket Programmer$149.95



The portable programmer that uses the printer port instead of a internal card.

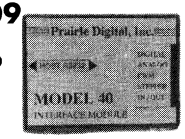
Easy to use software that programs E(E)prom, Flash & Dallas Ram. 25/27/28 & 29 series from 16K to 8 Megabit. Adapters available for 874X, 875X MCU's, Pic, Serial Eprom's, 40-Pin X 16, Proms and More.

Same Name, Address & Phone # for 19 Years... Isn't it Amazing ?

Intronics, Inc.
Box 13723 / 612 Newton St.
Edwardsville, KS 66113 Add $7.00 COD
Tel. (913) 422-2094 Add $6.00 Shipping
WWW.IN-KS.COM Visa / MC/Amex/Disc

CONTROL • MEASURE • INPUT

MODEL 40-$109
- RS-232 interface
- 28 lines digital I/O
- Eight analog inputs
- PWM output
- Three stepper ports



MODEL 100—$279
- 12-bit 100kHz A/D • Four analog outputs
- Three timer counters • 24 digital I/O



PRAIRIE DIGITAL, INC.
855 NINETEENTH ST., INDUSTRIAL PARK
PRAIRIE DU SAC, WI 53578
TEL: (608) 643-8599 • FAX: (608) 643-6754
Send CHECK, MO, VISA, MASTERCARD, AMERICAN EXPRESS or P.O. Please include $8 for S&H

Internet PLC

Program/control your PLC anywhere via the Internet using your web browser! (Password-protected Access)



Ladder + BASIC
0 - 5V Analog i/os
24V Digital i/os
LCD Port
10A PWMs,
Real Time Clock
Stepper pulsers
Interrupts,
Pulse Measures
MODBUS Ports
Any Questions?

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
TRI Triangle Research
Tel: 1-877-689-3245 - PLC specialist since 1993

AVOCET SYSTEMS, INC.
www.avocetsystems.com

HMI-200
In-Circuit Emulator



- ✓ Affordable
- ✓ Expert Support
- ✓ Easy-to-Use
- ✓ Supports all major compilers
- ✓ Be up and running in minutes

8051
68K
Z80
Z180
68HC11
683xx
64180
68020

Call today and ask about our Complete Package for the 8051

(800) 448-8500
sales@avocetsystems.com

Upgrade Kits

Make Your Furby Programmable



- 2 Minutes of Audio Data
- 64K Program Flash
- Free C-Compiler Available
- 8051 CPU
- RS-232 Port
- Infra-red TX/RX
- Tilt Switch
- Motor Control
- Power Management
- Self Contained
- Great Study Kit!

www.appspec.net

IDA Pro Disassembler



The Best tool to Reverse Engineer

From Atmel to Zilog, hundreds of μProcessors and μControllers supported. Older Unlimited Freeware version available for 80x86 and z80. Free Evaluation Version. The PC de-facto standard.

PC Magazine 2001 Technical Excellence Awards Finalist

www.datarescue.com



Microcross
GNU X-TOOLS™
Open Source, Ready-to-Run
Professional Embedded Development Tools

Version 2.0 Includes:
- C/C++, Java, Fortran Cross-Compilers
- Visual Debugger/Simulator
- Assembler Tools
- Binary Utilities
- Board Support
- Visual IDE
- Libraries
- More...
- Hosts:
  - Windows 98/NT/2000/XP
  - Linux

$499.95

Target Support -- 21 CPU families, 200 variants
ARM/Thumb i386 M68k/Coldfire PowerPC MIPS SPARC SHx XScale i960 ... and many more

www.microcross.com

Non-Contact Distance Measurement

Machinery & Industrial Automation



Ultrasonic sensors measure target distances from 2 in. to 37 feet

- Motion Control
- Dimensioning
- Positioning
- High sensitivity
- Easy to install
- Inexpensive
- Many options
- OEM models

CALL now...
800-677-3649 Senix®
e-mail: cci@senix.com www.senix.com

MICROSTEPPING MOTOR DRIVER



• 3,200 µsteps / rev • PCB Mount or Cable
• Micro-Full-Half Step Modes • 1.5 Amps / Ph
• 12-42• VDC • Opto-Isolated Interface
• Bipolar Chopper • Compact Package

Order # CY-21
$108 in qty 1 – Call for OEM pricing

800-328-3938 CyberPak company
www.cyberpakco.com
email: sales@cyberpakco.com

Rabbit Microprocessor C Compiler & Debugger



• Globally optimizing C compiler creates small and fast programs.
• IDE includes project manager, source editor, compiler, assembler, linker, source-debugger, and flash programmer.
• Compatible with all Rabbit developer's kits and the debug serial port.
• Supports 1MB for C code and data with far functions/far data and pointers.

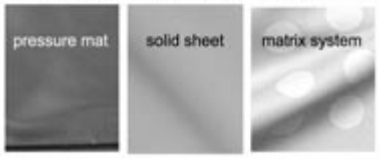
Softools
Quality development tools since 1989.
860-236-4201
www.softools.com • www.rabbittools.com

µLink® Wireless Remote Control



• 2/4/8CH keychain size FCC APPROVED RF transmitters
• 2/4/8CH keychain size IR transmitters
• 2/4/8CH RF or IR receivers
• Each receiver can learn up to four transmitters
• Microcontroller based encoder/decoder system
• NRE customization of firmware available to fit your application

Zoflex™ Conductive Rubber



www.xilor.com
© Xilor inc. Tel: 800-417-6689 or 865-546-9863
Visa/MasterCard/Discover 10am-6pm ET

HVW TECHNOLOGIES
FPGA Prototyping Made Easy...



Introducing the Intro-FPGA™ Development Programmer
for the Altera Max7000S™ Series FPGAs
Fits any Breadboard for Easy Prototyping
Ideal for Development or as a Trainer

BASIC and 'C' Pic Compilers-Programmers-Protoboards
PIC-OOPic-BASIC Stamp- Serial LCD/VFD/Graphic
RS-232 to RS-485 Adapters-IR Ranging Sensors

Satisfied Customers in Over 45 Countries Worldwide

HVW Technologies Inc.
Tel: (403)730-8603 VISA/MC
Fax: (403)730-8903 www.HVWTech.com

CUSTOM MEMBRANE KEYBOARDS / SWITCHES



• 1 TO 2 WEEKS TURNAROUND
• VERY COMPETITIVE PRICING
• Ex.: (5) 4-switch keyboards for $395.00
• PCB backed switches
• Custom metal backplates/assemblies
• Electronic assemblies/graphic overlays
• Electronic file transfer capabilities

Picofab Inc.
4780B Blvd. Henri-Bourassa
Charlesbourg, Quebec, Canada G1H 3A7
Tel: (418) 622-5298 ● Fax: (418) 622-9996
Email: sales@picofab.net

NEW!
Internet Appliance Module

• PIC16F877, 20MHz
• ICSP/ICD support
• CAN 2.0B or RF interface
• I-Button
• RS-232 port
• Cermetek Modem interface
• LCD/Keypad interfaces
• 6-12V unregulated input

Smaller than a credit card!
C, assembly, basic programmable

also
80188, 8051 and Pentium SBC's
HiTech C Compilers and MORE

Copeland Electronics, Inc.
440 Colony Place
Gahanna, Ohio 43230
Phone: 614-475-1690
Email: info@copelandelectronics.com

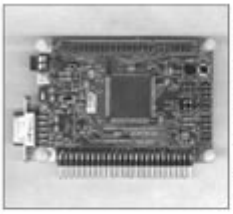
www.copelandelectronics.com

• View Control and Configure your micro over the Internet
• Drag and Drop embedded development tools included
• Build powerful applications with proven Components



microCommander.com
X86 HC16 Z80

Your 68HC12 Source



Adapt912DT60 features Motorola's 112-pin 68HC912D60 running in single-chip mode, with 86 I/O, dual 8-channel 10-bit analog-to-digital converters, 60K on-chip Flash, 2K SRAM, 1K EEPROM, RS232, RS485, I2C, 4-channel PWM, CAN, and lots more- all on a 3.25 in. x 2.25 in. board! Piggy-back RAM board available for rapid development.

Starter Package includes serial cable, prototyping card, solderless breadboard adapter, assembler, Flashload utilities, example code, and documentation. (Requires BDM pod)
• AD912DT60SP $149
C Developer Package has Starter Package + 64K RAM card + BDM pod + ICC12 C Compiler (Windows)
• AD912DT60CDP $499

Adapt912DG128 same as 'DT60, with dual CAN subsystem and more on-chip memory: 128K Flash, 8K RAM, 2K EEPROM.
Starter Package:
• AD912DG128SP $189
C Developer Package:
• AD912DG128CDP $539

Many other 68HC12 and 68HC11 products available!

TECHNOLOGICAL ARTS
toll-free 1-877-963-8996 • fax 416-963-9179
www.technologicalarts.com

R101 Gazelle Microcontroller



Low Cost!

• High speed 8051 uCore – 25MIPS – Flash
• 30 digital I/O, high current outputs
• 8 12-bit ADCs w/programmable gain
• 2 12-bit DACs, voltage or current
• 2 UARTs / JTAG programming port
• Watchdog timer, RTC, 16-bit PWM
• Expansion port using SPI, SMBus or I²C

R2 Controls
Call: 916-797-9252
www.r2-controls.com Fax: 916-313-3477

Servo Drivers from $73



- Standard Step and Direction inputs.
- Incremental encoder TTL input.
- Output up to 200V 24A.
- Analogue output +/- 10 V.
- Brush/brushless DC servomotors.
- Full digital PID filter.
- Step multiplier 1 to 255.

www.rutex.com
Rutex, PO Box 3500, Wamberal 2260, Australia

WWW.CONITEC.NET

$333 COMPLETE KIT

GALEP-4 UNIVERSAL PROGRAMMER



Introducing a pocket programmer with true Universal Output



Latest generation Galep-4 uses ASIC universal pin technology for each pin of 40 pin ZIF-socket. 1300+ device library / lifetime free updates. Programs 8/16-bit EPROM'S, EEPROM's, 0-Power 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) Providing virtually matching performance at only 1/3-1/5 the price. Info, orders, softwr : **619-702-4420**

ISO 9001 CERTIFIED PROFESSIONAL FOUND 9001

info, downloads: www.conitec.net

CONITEC DATASYSTEMS - 1951 4TH AVE, SUITE 301 - SAN DIEGO, CA 92101 - TEL: 619-702-4420 FAX: 619-702-4419



MC68332
512KB Flash,
8KB ram
PLD, JTAG port
$89.00

- BDM emulation port
- RS232 and RS485 interfaces
- Project development board $49.00

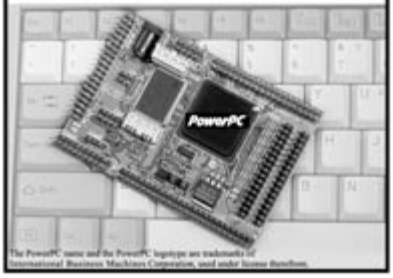
- MC68376 CAN module
- 1MB Flash, 256KB ram
- Isolated CAN bus interface
- RS232 interface
- $299.00

www.spectrumengineering.com
Seattle, Washington
toll-free 877-851-0646

Embedded PowerPC™ based MPC555 Microcontroller

Intec Automation
www.steroidmicros.com

448K Flash — 32ch A/D
512K SRAM — 32ch TPU
10ch Double Action Timer
8 Interrupts — 2 CAN Ports
16 GP I/O — 2 SCI Serial
8ch 16 bit PWM ..and more..



The PowerPC name and the PowerPC logotype are trademarks of International Business Machines Corporation, used under license therefrom.

PROTOTYPE CIRCUIT BOARDS

$85 U S 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
staff@apcircuits.com

PICmicro Tools — EPIC Programmer — PicBasic Compilers — LAB-X Experimenter Boards — PICProto Boards

PicBasic and PicBasic Pro Compiler



Write programs for PICmicro MCUs in BASIC. Can be used in DOS or Windows (includes Windows editor/IDE software).

PicBasic Compiler - $99.95: • BASIC Stamp I compatible instruction set • May use up to 2K of code space • Supports PIC12C67x, 14C000, 16C55x, 6xx, 7xx, 84, 92x, 16F62x, 8x, and 87x

PicBasic Pro Compiler - $249.95: • Built-in LCD support • Direct access to any pin or register • Page boundary handling past 2K • Bit, byte and word arrays • Real If..Then..Else..Endif • Hierarchical expression handling • Interrupts in BASIC and assembler • BASIC Stamp I and II library • Supports PIC12C5xx, 12C67x, 14C000, 16C5x, 55x, 6xx, 7xx, 84, 92x, 16CE62x, 16F62x, 7x, 8x, 87x, 17Cxxx and 18Cxxx

LAB-X1 Experimenter Board



The LAB-X1 is a pre-assembled experimenter's platform. It provides a testbed containing most of the circuitry commonly used with PICmicro microcontrollers (MCUs). Possible applications include calculator, voltmeter, LCD backpack, tone dialer, and RC servo control. The LAB-X1 can run programs written in Assembly, C, PicBasic, or PicBasic Pro. (requires 40-pin MCU)

Bare PCB - **$49.95**, Kit w/parts - **$139.95**, Assembled - **$199.95**

LAB-X2 (not shown) for 28 or 40-pin MCUs - **$69.95** (Assembled)
LAB-X3 (not shown) for 18-pin 16F8x or 16F62x - **$119.95** (Assembled)

EPIC Plus PICmicro Programmer



The pocket-sized EPIC Plus Programmer quickly and easily programs the following PICmicro microcontrollers: PIC12Cxxx, 12CExxx 14C000, 16C505, 16C55x, 6xx, 7xx, 84, 92x, 16CE62x, 16F62x, 7x, 8x, 87x, 17C7xx, and 18Cxxx (some MCUs require an additional adapter). Interactive software included for Windows and DOS. Requires two 9V batteries or AC adapter (not included).

Bare PCB w/software - **$34.95**, Assembled - **$59.95**
Assembled w/AC adapter, 25-pin cable and ZIF adapter - **$99.95**

PICProto Prototyping Boards - $8.95 to $19.95



• High-quality blank prototyping boards for PICmicro MCUs • Large prototyping area • Holds microcontroller, 5-volt regulator, oscillator, capacitors, DB9-25 connector • Various sizes and layouts to support most PICmicro microcontrollers • Double-sided board with plated-through holes, solder mask, and silkscreen • Includes schematic, parts list, and placement information

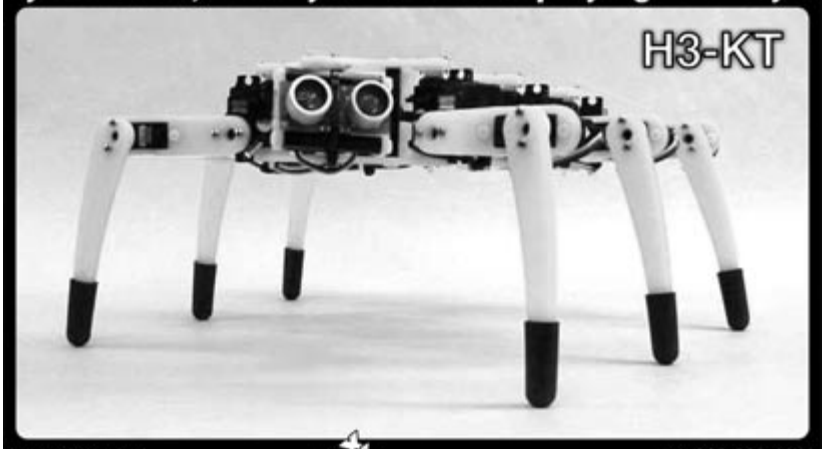
microEngineering Labs, Inc.

Phone: (719) 520-5323 Fax: (719) 520-1867 www.melabs.com
Box 60039, Colorado Springs, CO 80960 Email: info@melabs.com

For more products and information, or to purchase online, visit our website at: www.melabs.com

Lynxmotion, When you're tired of playing with toys!

H3-KT



Lynxmotion, Inc.
PO Box 818
Pekin, IL 61555-0818
www.lynxmotion.com

Lynxmotion
Visit our website or ask for our free catalog!

Tel: 309-382-1816
Fax: 309-382-1254
sales@lynxmotion.com
tech@lynxmotion.com

TX1 / RX1

Wireless Data Link Modules
Transmitter & Receiver Pair



173.225 MHz AND 173.250 MHz

The TX1 & RX1 form a miniature VHF radio transmitter/receiver pair designed for PCB mounting. These are suitable for use in fixed/in-building alarms as well as industrial/commercial telemetry applications. Link ranges of 10 km+ are achievable with suitable choice of data rate and antennas.

www.lemosint.com

LEMOS INTERNATIONAL
Microwave and RF Specialists

1305 Post Road
Fairfield, CT 06430
Tel: (203) 254-1531
Fax: (203) 254-7442
e:Sales@lemosint.com

PCB Production

Double side—12¢ per in²
Four layers—20¢ per in²
Six layers—32¢ per in²
UL approved
With solder mask, Silkscreen
Three weeks

www.mylydia.com

MYLYDIA INC.

Call 1-800-MYLYDIA
E-mail: PCBsales@mylydia.com



Make your web cam move from anywhere in the world!

- Digineck is a build-it yourself pan & tilt robotics kit for your web cam.
- Experiment moving your web cam with a remote control from your personal computer or through the Internet.
- Great for teleconference applications.
- An excellent technology project.

digineck

Call toll free 1-877-425-8163
or visit us at www.digineck.com

High Speed Wireless Data Transceiver
Physical Dimensions: 33 mm x 23 mm x 4 mm

BiM2

900 MHz Available January 2002

- Data rates up to 160 kbps
- SAW controlled FM transmitter
- RX sensitivity -100dBm
- Usable range up to 200m
- Low power requirements
- TX power 10mW @ 5V
- 3V and 5V versions
- Fully screened
- Double conversion Superhet receiver
- Plug in replacement for BiM-433-F

433 MHz



www.lemosint.com

LEMOS INTERNATIONAL
Microwave and RF Specialists

1305 Post Road
Fairfield, CT 06430
Tel: (203) 254-1531
Fax: (203) 254-7442
e:Sales@lemosint.com

Weeder Technologies

www.weedtech.com DATA SHEETS

RS-232 Stackable

1710-B Brighton Cove, Ft Walton Beach, FL 32547 850-863-5723

Digital I/O Module - 14 I/O channels individually configured for input or output. Turn on/off relays. Sense switch transitions and button presses. 4x4 matrix decoding using auto-debounce and typematic repeat. **$59**

Analog Input Module - 8 single-ended or 4 differential inputs. Self-calibrated, 12-bit ADC, reads voltages from 0 to 4095 mV. High and Low alarm trip-points for each input. **$69**

Analog Output Module - 4 outputs that span -10 to +10 volts using 12-bit DAC. Ramp generator, software calibrated, select POR defaults. **$89**



Stepper Motor Driver - Directly drives a unipolar stepper motor rated up to 30VDC @ 2A. Self-generated S-curve accel/decel slope profiles provide smooth start and stop motion. 24-bit absolute motor position counter. **$69**

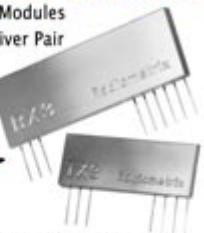
Pulse Counter/Timer - Read frequency from 0.50000 Hz to 1,500,000 Hz using floating decimal point and 5-digit resolution throughout range. Measure period, RPM, duty cycle, pulse length, the velocity of a projectile using a pair of trip wires. 24-bit pulse count accumulator. **$79**

Plug end-to-end
Stack 32 modules on the same RS-232 cable.

Solid State Relay Module - 5 optoisolated relays can be wired directly to existing low-current buttons and switches to provide software control of their operation. Built-in event sequencer. **$69**

TX3A / RX3

Wireless Data Link Modules
Transmitter & Receiver Pair



869 MHz AND 900 MHz

The TX3A & RX3 data link modules are a miniature PCB mounting UHF radio transmitter and receiver pair. They enable the simple implementation of a data link at up to 64 kbits/s at distances up to 75 meters in-building and 300 meters open ground. North American 902-928MHzISM band.

www.lemosint.com

LEMOS INTERNATIONAL
Microwave and RF Specialists

1305 Post Road
Fairfield, CT 06430
Tel: (203) 254-1531
Fax: (203) 254-7442
e:Sales@lemosint.com

ADVERTISER'S INDEX

The Advertiser's Index with links to their web sites is located at www.circuitcellar.com under the current issue.

Page		Page		Page		Page	
86	Abacom Technologies	93	Digineck	7	Microchip Design Contest	88	RPA Electronics Design, LLC
85	Abia Technology	88	Dreamtech Computers	90	Microcross	92	Rutex
66	Accutech	1	Earth Computer Technologies	89	Micro Digital Inc	5	Saelig Company
86	ActiveWire, Inc.	23,82	ECD (Electronic Controls Design)	92	microEngineering Labs, Inc.	3	Scott Edwards Electronics Inc.
22	All Electronics Corp.	86	EE Tools (Electronic Engineering Tools)	49	Micromint Inc.	88	Sealevel Systems Inc.
84	Allied Components	29	EMAC, Inc.	85	MicroSystems Development, Inc.	90	Senix Corp.
86	Amazon Electronics	87	EVB Plus	87	MJS Consulting	88	Sensory, Inc.
9	Amulet Technologies	23	ExpressPCB	39	Mouser Electronics Inc.	84	Signum Systems
92	AP Circuits	85	FDI-Future Designs, Inc.	34	MVS	91	Softools
90	Appspec Computer Tech. Corp.	85	Hagstrom Electronics	93	Mylydia Inc.	47,58	Solutions Cubed
85	Atlantic Quality Design, Inc.	66	HI-TECH Software,LLC	65	NetBurner	92	Spectrum Engineering
90	Avocet Systems, Inc.	91	HVW Technologies Inc.	95	Netmedia, Inc.	84	Square 1 Electronics
73	B & K Precision	87	IMAGEcraft	87	OKW Electronics Inc.	50	SUMBOX Pty Ltd.
87	Bagotronix, inc.	91,92	Intec Automation, Inc.	35	OnTime	16	Systronix
17,85	Basic Micro	90	Intronics, Inc.	28	Ontrak Control Systems	C3	Tech Tools
79	CadSoft Computer, Inc.	59	Intuitive Circuits, LLC	C2	Parallax, Inc.	89	Techniprise Inc.
89	CCS-Custom Computer Services	18	JED Microprocessors Pty Ltd.	84	Phytec America LLC	24,25	Technologic Systems
86	Cermetek Microelectronics Inc.	64	JK microsystems	84	Phyton, Inc.	91	Technological Arts
92	Conitec	59	JR Kerr Automation & Engineering	91	Picofab Inc.	88	Tern Inc.
29	Connecticut mircoComputer Inc.	82	LabJack Corp.	90	Prairie Digital Inc.	90	Triangle Research Int'l Inc.
91	Copeland Electronics Inc.	41	Laipac Technology, Inc.	89	Pulsar Inc.	28	Trilogy Design
91	Cyberpak Co.	82	Lakeview Research	91	R2 Controls	93	Weeder Technologies
33	Cypress MicroSystems	93	Lemos International	15	R4 Systems Inc.	91	Xilor Inc.
C4	Dataman Programmers, Inc.	2	Link Instruments	55	Rabbit Semiconductor	87	Z-World
90	DataRescue	93	Lynxmotion, Inc.	90	Rad Proto	59	Zagros Robotics
84	Decade Engineering	41	MaxStream	86	R.E. Smith	86	Zanthic Technologies Inc.
87	Delcom Engineering	89	MCC (Micro Computer Control)	89	RLC Enterprises, Inc.		

Attention Advertisers:
November Issue Deadlines

Space Close: Sept. 11
Material Due Date: Sept. 18

Theme: Embedded Development

BONUS DISTRIBUTION
Embedded Systems Conference East

Convert Your PC Sound Card—
Make a DC-Coupled Arbitrary Waveform Generator

Build Your Own 8051 Web Server

Office Supervisor—A Control System Based on SMS

RISCy Business—Part 2:Roundup of Cornell Projects

Killing the EMI Demon

The PSoC 5-Cent Modem

Robotics Corner: Put Out the Fire—Detect Flames with the UVtron Sensor

Applied PCs: Internet Enabling Made Easy

From the Bench: SmartMedia—Part 4:Getting Your Data Inside

Silicon Update: Fast Times at the Forum

PREVIEW 146

Internet & Connectivity

www.basicx.com

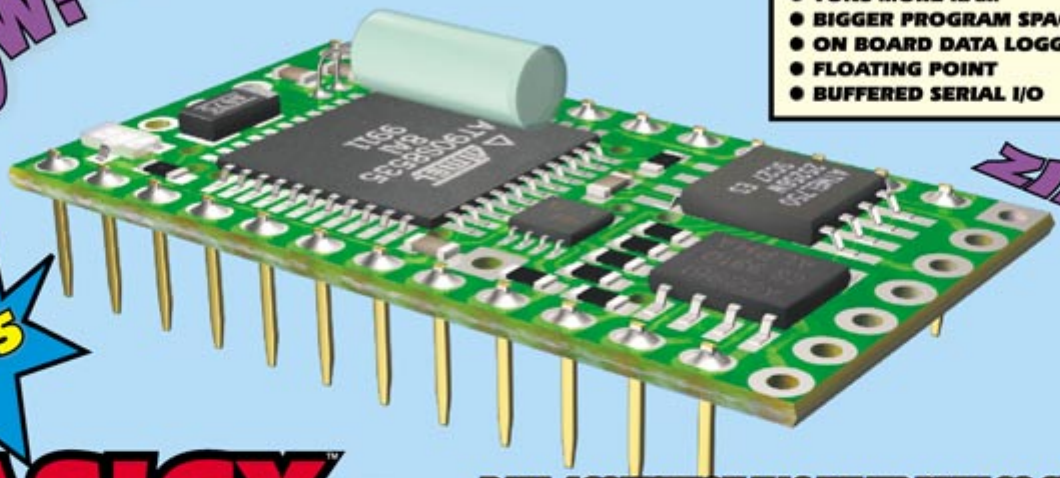
NetMedia's DYNAMIC DUO

BASICX LICKS THE STAMP

POW!

BAM!

- EIGHT 10 BIT A/D'S
- TONS MORE RAM
- BIGGER PROGRAM SPACE
- ON BOARD DATA LOGGING
- FLOATING POINT
- BUFFERED SERIAL I/O



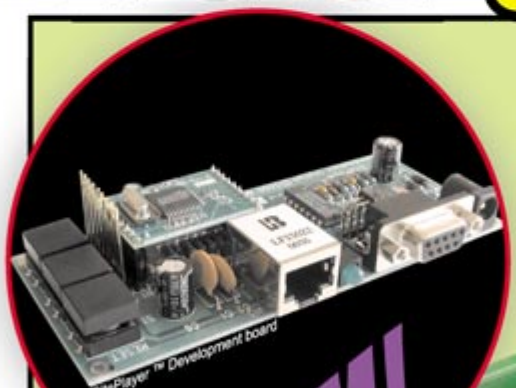
ZING!

$49.95

BASICX™ & SITEPLAYER™

DATA ACQUISITION HAS NEVER BEEN SO SMALL!

WEB SERVER MODULE



ZOOM!

CRUNCH!



- COMPLETE 1 SQ. INCH SERVER
- WEB ENABLE YOUR PROJECTS
- ETHERNET ON MODULE
- UP AND RUNNING IN NO TIME
- DEVELOPMENT KIT JUST $99.95
- SERIAL PORT INTERFACE

$29.95

WORLDWIDE GRAPHICAL USER INTERFACE

NETMEDIA INC. 10940 N STALLARD PL TUCSON AZ 85737

520.544.4567 sales@netmedia.com www.netmedia.com

www.siteplayer.com

PRIORITY INTERRUPT

Upgrade or Die



I don't remember exactly the reason, but recently I needed to check the pinout for an XR2211 PLL chip. The part number rung a bell, and rather than dig through a pile of dusty and obsolete EXAR manuals, I grabbed an early *Ciarcia's Circuit Cellar* book off the shelf. Ah, lets see. Way back then I wrote an article titled, "A Build-It-Yourself Modem for Under $50." It used an XR2211 phase-locked loop chip and the voltage-controlled oscillator section of an NE567 connected to an acoustic coupler. Today, I cringe when I look at the complexity of some of those early brute-force designs, but they worked exceedingly well for their time.

Back then, I think I was still working for Control Data Corporation as one of their consulting engineers. I didn't consult to them. It was the other way around. They had a professional services division where they used engineers at a customer site to consult to the customers. As long as we didn't criticize the sanctity of massive centralized computer control schemes and didn't create more messes than we solved, job security was good. Unfortunately, my after-hours activities put me on a collision course with such absurd technical dogma.

My personal system was a Digital Group Z-80. For its time it was a powerful machine and had everything. The Z-80 had tape drives (later upgraded to floppies), memory-mapped ASCII terminal, 64 KB of memory, an operating system with BASIC, and even a Votrax speech synthesizer board. Best of all, it had a modem.

Back then, 300 bps was king. There was no Internet, no web, nothing. Except for a few military geeks, cyberspace was still science fiction. There were no hubs or routers. One computer simply picked up the phone and called another. If that computer was outfitted with a bunch of telephone lines (or a service like Telenet) and message-logging software, it became a bulletin board system (BBS) resembling something like today's single-subject chat groups. BBSs were typically local because of flat-rate telephone billing. Anything outside the local area was an expensive call. There wasn't any of this penny a minute stuff we have now. Occasionally you'd venture out to other BBSs but you took your chances. The meter started running when the other computer answered, but you never knew how long it would take or even if you would eventually get in.

The good news was that we all spoke the same data rate. Initially, it was 300 bps. There was no spam, trash messages, or viruses, just smooth ASCII characters streaming in at human speed. Everything was text or graphics built from ASCII characters, and it simply scrolled down the screen. In truth, 30 characters per second got old fast and 1200 bps became the standard for a longer time. There were many experts who even thought that 1200 or 2400 bps was the limit for our 100-year-old copper-wired telephone system. Until they broke the barriers, Hayes modems were kings.

These early years were innocent times. Because everything was ASCII text or limited text graphics, BBS users concentrated on communication without crap (often misspelled, mis-punctuated, and sometimes all caps). Yes, there were the commercial interruptions, but those things deemed as interruptions were easily remedied. If someone posted an unwanted commercial message, the BBS moderator would simply remove it. There was none of this sending a thousand individual spam messages to everyone on the BBS.

If it seems that I'm rambling, I agree. I guess that I'm merely lamenting the fact that this early period may have been the last time that serious personal computer users got the benefit of the full bandwidth. As limiting as 30 or 120 characters per second may seem, it was one-on-one and it worked.

Yes, we've had a lot of evolution since then. However, the real change is not the data rate, but how we have changed using it. Low data rates dictated a privately controlled network with limited participation. Higher data rates have allowed a portion of the bandwidth to be utilized in making everything into a public highway and a public communications system. If the message is simply "you need an XR2211," you could hardly tolerate a couple thousand bytes of header and routing information at 1200 bps.

Today, with 500 kbps or greater connection speeds, a few kilobytes of routing information is not only innocuous, but you probably don't even mind that this simple 17-character message comes displayed on 200 KB of custom-graphic virtual stationary. While simpler technology offered emotional security, today we've all gotten used to surfing for practically everything; and there is no going back. Moreover, I have come to expect that what I find on the web will be detailed and plentiful. Any personal computer system able to support those lofty ambitions has a price as well as a penalty. The penalty we pay for an unrestricted cyberspace is that sometimes it's like living in the old Wild West. The price for keeping pace with all these bandwidth-intensive Internet functions is, upgrade or die.

steve.ciarcia@circuitcellar.com

TechTools™ Rapid Embedded Development Tools

BIG performance for a Small Price!

QuickWriter™

NEW!

PICmicro® MCU Programmer



Program 4 Microcontrollers as quickly as programming a single device.

SPECIAL FEATURES:

- Supports '5X, 'XX and 18 series PICmicro® MCUs.
- Creates Control Mode HEX files for production personnel and archiving.
- Automatic Firmware FLASH updates and version checking.
- Manually control Serial Numbering and Incrementing.
- Automatic programming and supply voltage verification.
- Manually select individual sockets on Gang Adapters.
- Manually control Individual programming and verify tasks.
- "Auto RUN" feature for 1 button operation.

Other Features:

- Accepts all Parallax and TechTools Programming adapters.
- Individual HEX/ASCII Editor for CODE, Data and Calibration.
- Tracks total programming cycles and Pass/Fail count.
- Accepts INTEL HEX 8bit merged file format.
- Saves HEX files in Microchip, TechTools or Control Format.

See Our Other Rapid Embedded Development Tools:

- PICmicro MCU In-Circuit Emulator.
- EconoROM™ III EPROM Emulator.
- FlexROM™ III EPROM/FLASH Emulator.
- UniROM™ Advanced Memory Emulator
- CVASM™ PICmicro MCU Assembler.

Get all the details and preview the software at: **www.tech-tools.com**

Copyright © 2002, TechTools ClearView, CVASM, QuickWriter, the "Wizard" symbol and TechTools are trademarks of TechTools, P.O. Box 462101 Garland, Texas 75046. PICmicro is a trademark of Microchip Technology, Inc. All other trademarks are trademarks or registered trademarks of their respective company.

STILL THE WORLD'S MOST POWERFUL PORTABLE PROGRAMMERS?

CE

$795
INC 4MB RAM





NEW MODEL



$1295

SURELY NOT.
SURELY SOMEONE SOMEWHERE HAS DEVELOPED A PORTABLE PROGRAMMER THAT HAS EVEN MORE FEATURES, EVEN GREATER FLEXIBILITY AND IS EVEN BETTER VALUE FOR MONEY.

ACTUALLY, NO. BUT DON'T TAKE OUR WORD FOR IT. USE THE FEATURE SUMMARY BELOW TO SEE HOW OTHER MANUFACTURERS' PRODUCTS COMPARE.

DATAMAN-48LV

- Plugs straight into parallel port of PC or laptop
- Programs and verifies at 2, 2.7, 3.3 & 5V
- True no-adaptor programming up to 48 pin DIL devices
- Free universal 44 pin PLCC adaptor
- Built-in world standard PSU - for go-anywhere programming
- Package adaptors available for TSOP, PSOP, QFP, SOIC and PLCC
- Optional EPROM emulator

DATAMAN S4

- Programs 8 and 16 bit EPROMs, EEPROMs, PEROMs, 5 and 12V FLASH, Boot-Block FLASH, PICs, 8751 microcontrollers and more
- EPROM emulation as standard
- Rechargeable battery power for total portability
- All-in-one price includes emulation leads, AC charger, PC software, spare library ROM, user-friendly manual
- Supplied fully charged and ready to use

S4 GAL MODULE

- Programs wide range of 20 and 24 pin logic devices from the major GAL vendors
- Supports JEDEC files from all popular compilers

SUPPORT

- 3 year parts and labor warranty
- Windows/DOS software included
- Free technical support for life
- Next day delivery - always in stock

Still as unbeatable as ever. Beware of cheap imitations. Beware of false promises. Beware of hidden extras. If you want the best, there's still only one choice - Dataman.

Order via credit card hotline - phone today, use tomorrow.

Alternatively, request more detailed information on these and other market-leading programming solutions.

MONEY-BACK 30 DAY TRIAL

If you do not agree that these truly are the most powerful portable programmers you can buy, simply return your Dataman product within 30 days for a full refund

hotline
800 328-2336

VISA MasterCard AMERICAN EXPRESS

Orders received by 4pm will normally be despatched same day.
Order today, get it tomorrow!

DATAMAN

Dataman Programmers Ltd
215 East Michigan Avenue
Orange City, FL 32763
Telephone (904) 774-7785
Fax (904) 774-7796
Home page: http://www.dataman.com
Email: sales@dataman.com