Q&A: Programming, Testing, & Success
LOCATION: United States
PAGE: 40

INSIGHT: Environmental Stress Screening
LOCATION: Canada
PAGE: 52

HOW TO: Choose Linux for an Embedded App
LOCATION: United States
PAGE: 56

CIRCUIT CELLAR

THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

OCTOBER 2011
ISSUE 255

SIGNAL PROCESSING

Line-Coding Techniques for Wireless & Wired Protocols

An Accurate Programmable Voltage Source

A Versatile Platform for Ethernet-Enabled Systems

Microstepping Explained

An API for Controlling a Wi-Fi Thermostat



PLUS

DIY Heating Plate

Build an MCU-Based Cup Warmer

// Component Selection
// Chip Programming
// Program Operation
// PID Basics
// And More

10>
7 25274 75349 9
$7.50 U.S. ($8.50 Canada)
www.circuitcellar.com

The easiest finished product you will ever create

Start with the PK70 EX



Design your own, or use a NetBurner Personality Blade



Add your label to get a finished product



Personality Blades

Prototype Blade *(P/N NBPKBP-100CR)* ..... **$25 ea.**
Create your own Personality Blade

CPLD Digital I/O Blade *(P/N NBPKD-100CR)* ..... **$59 ea.**
Programmable Xilinx CPLD
32 digital I/O

Multi-I/O Blade *(P/N NBPKM-100CR)* ..... **$99 ea.**
Eight 12-bit analog-to-digital inputs
Two 16-bit digital-to-analog outputs
16 digital I/O

FPGA Digital I/O Blade *(P/N NBPKX500-100CR)* ..... **$169 ea.**
Programmable Xilinx Spartan 3E FPGA
High-density 62-pin connector

NTP Server with GPS Blade *(P/N NBPKX500-100CR)* ..... **$99 ea.**
Network Time Protocol (NTP) server
Precision GPS time reference

Quad UART Blade: RS-232 Version
(P/N NBPKBU-232CR) ..... **$99 ea.**
Network enable up to 4 RS-232 serial devices

Quad UART Blade: RS-422/485 Version
(P/N NBPKBU-485CR) ..... **$119 ea.**
Network enable up to 4 RS-422/485 serial devices

Quad UART Blade: RS-232/422/485 Version
(P/N NBPKBU-MMSCR) ..... **$245 ea.**
Network enable up to 4 RS-232/422/485 serial devices

PK70 EX & Development Kit

PK70 EX *(P/N PK70EX-100IR)* ..... **$199 ea.**
32-Bit Freescale ColdFire 147 MHz CPU
8MB SDRAM • 4MB Flash Memory
SD/MMC Flash Card Support • I2C • SPI • Address and Data Bus
10/100 Ethernet • 2 UARTs

PK70 EX Development Kit
(P/N NNDK-PK70EX-KIT) ..... **$299 ea.**
The Development Kit features NetBurner's Eclipse, an enterprise level professional IDE that provides editing, downloading and debugging in one environment.

NetBurner
Networking in One Day!

Information and Sales | sales@netburner.com **Web** | www.netburner.com **Telephone** | 1-800-695-6828

NEW chipKIT™ Development Platforms from Microchip

The first 32-bit MCU platforms compatible with Arduino™ HW and SW



The chipKIT™ Uno32 and Max32 development boards are the first 32-bit microcontroller-based platforms that are compatible with existing Arduino™ hardware and software.

The chipKIT platform allows hobbyists and academics from many disciplines, such as mechanical engineering, computer science and artists to develop original embedded applications easily and quickly including: motor control, LCD display, wired and wireless communications, LED matrix control and sensor networks

Key Features:

- Application development using an environment based on the original Arduino IDE, modified to support PIC32 devices while still supporting the original Arduino line. Leverages existing code examples, tutorials and resources.
- Pin-out compatibility with many existing Arduino shields
- Higher performance at a lower price-point than existing solutions
- Advanced capabilities including:
  - Integrated USB (Device/Host, OTG)
  - Integrated Ethernet
  - CAN

Development Board Comparison

Feature	Core	Performance	Program Memory (KB)	RAM (KB)	Additional Features
chipKIT™ Uno32	32-bit	80 MHz	128	16	PMP/PSP/RTCC
chipKIT Max32	32-bit	80 MHz	512	128	USB, 2x CAN, Ethernet, DMA, RTCC

IT'S EASY TO GET STARTED!

- Visit **www.microchip.com/chipkit**
- Purchase a chipKIT development board
- Download the free software
- Order free PIC32 samples



chipKIT Uno32 (TDGL002)



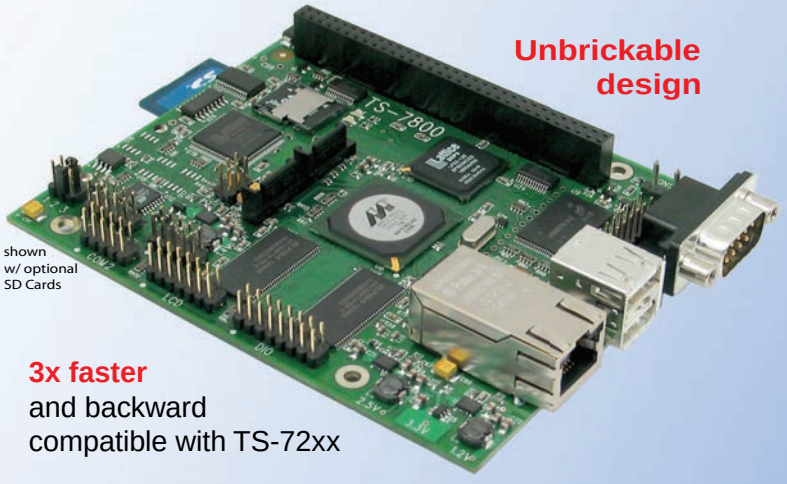
chipKIT Max32 (TDGL003)

The Microchip name and logo, the Microchip logo and PIC are registered trademarks and chipKIT is a trademark of Microchip Technology Incorporated in the USA and other countries. All other trademarks mentioned herein are the property of their respective trademark holders. These trademark holders are not affiliated with Microchip Technology Incorporated, and do not support, sponsor or endorse chipKIT products or solutions. © 2011 Microchip Technology Incorporated. All rights reserved. 5/11

MICROCHIP

Embedded Systems

High-End Performance with Embedded Ruggedness



shown w/ optional SD Cards

Unbrickable design

3x faster and backward compatible with TS-72xx

TS-7800
500 MHz ARM9

- Low power - 4W@5V — $229 qty 100
- 128MB DDR RAM
- 512MB high-speed (17MB/sec) onboard Flash — $269 qty 1
- 12K LUT customizable FPGA
- Internal PCI Bus, PC/104 connector
- 2 host USB 2.0 480 Mbps
- Gigabit ethernet
- 2 SD sockets
- 10 serial ports
- 110 GPIO
- 5 ADC (10-bit)
- 2 SATA ports
- Sleep mode uses 200 microamps
- Boots Linux 2.6 in 0.7 seconds
- Linux 2.6 and Debian by default

TS-TPC-7390
7" Color Touch Panel Computer

- Low Power, Industrial Quality Design
- Mountable aluminum frame
- 200 MHz ARM9
- 64MB SDRAM (128MB opt)
- 512MB Flash w/ Debian Linux
- 800 x 480 video core
- Dedicated framebuffer- 8MB RAM
- Audio codec with speaker
- Boots Linux 2.6 in less than 2 seconds
- Unbrickable, boots from SD or NAND
- Runs X Windows GUI applications
- Runs Eclipse IDE out-of-the-box



$449 qty 1

More Touch Panel Computers on our website

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

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

New Products

TS-WIFIBOX-2

A Complete Solution for 802.11g WiFi Applications

NEW!

$185 qty 1

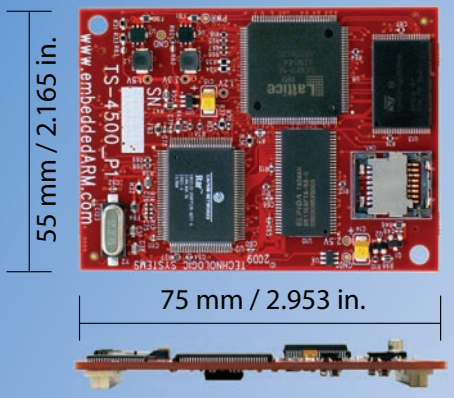


- Low power (3.2 watts), fanless
- Power via 5-12 VDC, USB, PoE (opt.)
- 64MB DDR-RAM
- 256MB ultra-reliable XNAND drive
- Micro-SD Card slot
- RS-232, RS-485, CAN, RTC
- Header with SPI and 11 DIO
- 480Mbit/s USB, Ethernet, PoE option
- Boots Linux 2.6.24 in < 3 seconds
- Un-brickable, boots from SD or flash
- Customizable FPGA - 5K LUT
- Optional DIN mountable enclosure

Ideal for gateway or firewall, protocol converter, web server, WiFi audio, and unattended remote applications

NEW!

series starts at $92 qty 100



TS-SOCKET Macrocontrollers

Jump Start Your Embedded System Design

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

Current projects include:

- TS-4200: Atmel ARM9 with super low power
- TS-4300: Cavium ARM11 with dual 600 MHz and FPU
- TS-4500: Cavium ARM9 at very low cost
- TS-4700: Marvell PXA168 with video and 1.2 GHz CPU
- TS-4800: Freescale iMX515 with video and 800 MHz CPU
- Several COTS baseboards for evaluation & development

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

Technologic Systems

We use our stuff.

Visit our TS-7800 powered website at

www.embeddedARM.com

TASK MANAGER

Get Grounded

Last month I introduced Richard Wotiz's new column, titled "Embedded Unveiled," which will appear every other month in even-numbered issues (i.e., *Circuit Cellar* 254, etc.). Now I'd like to announce another new *Circuit Cellar* columnist, Bob Japenga, whose articles will appear every other month in odd-numbered issues. Bob's column, "Embedded in Thin Slices," is geared toward helping readers to hone their embedded programming kills and apply them to real-world designs.

Why is our editorial team interested in these topics? The idea is to "ground" our readers in the basics of embedded design and programming, and particularly embedded Linux. Ironically, the seed for this column was sown nowhere near the ground; it was planted in the air during a work-related conversation that took place at Le Jules Verne restaurant in the Eiffel Tower. Let me explain.

Back in January 2011, I was sitting above Paris and talking about engineering with *Elektor*'s Editor-in-Chief Wisse Hettinga, *Elektor*'s France-based Editor Clemens Valens, and *Circuit Cellar*'s France-based columnist Robert Lacoste. Among the many topics covered, we discussed was *Circuit Cellar*'s past, present, and future content. Which engineering subjects did we cover well and which needed more attention? It was then that the topic of embedded Linux came up. It was clear our staffers and readers had been interested in embedded Linux for some time. But how could we start covering the topic if many of our long-time authors were unfamiliar with subject? In fact, some of those authors were the same people with a hankering to learn about it.

At the conclusion of our meeting, the task for the *Circuit Cellar* editorial department was clear: we had to find a way to present embedded Linux—and embedded programming in general—in an accessible, useful way on a consistent basis (either monthly or bimonthly).

Fast-forward several months. By chance, my research brought me to a Connecticut-based engineering and programming company whose co-founder—Bob Japenga—happened to have a strong background in embedded design and programming. It turned out he was familiar with *Circuit Cellar* and was interested in writing. That was great news for all of us.

Turn to page 56 for the first article in Bob's three-part series "Getting Started with Embedded Linux." And for more on programming, read the interview with Larry Cicchinelli on page 40. Larry describes why Assembly language still matters. To brush up on Assembly, check out Larry's book, *Assembly Language Essentials*, which is available at www.cc-webshop.com.

Need more grounding in electrical engineering essentials? This issue has you covered. Topics include: a control platform for Ethernet-enabled systems (p. 18); an MCU-based heat control system for mugs (p. 26); microstepping and stepper motor drives (p. 46); USNAP Wi-Fi (p. 60); and line-coding techniques (p. 66).

That's a lot of content. It's time to start reading. The November issue will be on its way soon enough!

cj@circuitcellar.com

C. Abate

CIRCUIT CELLAR®

THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

FOUNDER/EDITORIAL DIRECTOR
Steve Ciarcia

EDITOR-IN-CHIEF
C. J. Abate

ASSOCIATE EDITOR
Nan Price

CONTRIBUTING EDITORS
Jeff Bachiochi
Robert Lacoste
George Martin
Ed Nisley

PROJECT EDITORS
Ken Davidson
David Tweed

PUBLISHER
Hugo Van haecke

ASSOCIATE PUBLISHER
Shannon Barraclough

CUSTOMER SERVICE
Debbie Lavoie

CONTROLLER
Jeff Yanco

ADMINISTRATIVE COORDINATOR
Valerie Luster

ART DIRECTOR
KC Prescott

GRAPHIC DESIGNER
Grace Chen

Circuit Cellar's mission is to collect, select, and disseminate need-to-know information around the world in the fields of embedded hardware, embedded software, and computer applications. Circuit Cellar uses an assortment of print and electronic content-delivery platforms to reach a diverse international readership of professionals, academics, and electronics specialists who work with embedded, MCU-related technologies on a regular basis. Our aim is to help each reader become a well-rounded, multidisciplinary practitioner who can confidently bring innovative, cutting-edge engineering ideas to bear on any number of relevant tasks, problems, and technologies.

ADVERTISING
800.454.3741 • 978.281.7708 • www.circuitcellar.com/advertise

ADVERTISING REPRESENTATIVE
Peter Wostrel
Strategic Media Marketing, Inc.
1187 Washington St., Gloucester, MA 01930 USA
800.454.3741 • 978.281.7708
peter@smmarketing.us • www.smmarketing.us
Fax: 978.281.7706

ADVERTISING COORDINATOR
Kim Hopkins
E-mail: khopkins@circuitcellar.com

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

PRINTED IN THE UNITED STATES

CONTACTS

SUBSCRIPTIONS
Information: www.cc-access.com, E-mail: subscribe@circuitcellar.com
Subscribe: 800.269.6301, www.cc-access.com, Circuit Cellar Subscriptions, P.O. Box 5650, Hanover, NH 03755-5650
Address Changes/Problems: E-mail: subscribe@circuitcellar.com

GENERAL INFORMATION
860.875.2199, Fax: 860.871.0411, E-mail: info@circuitcellar.com
Editorial Office: Editor, Circuit Cellar, 4 Park St., Vernon, CT 06066, E-mail: editor@circuitcellar.com
New Products: New Products, Circuit Cellar, 4 Park St., Vernon, CT 06066, E-mail: newproducts@circuitcellar.com

AUTHORIZED REPRINTS INFORMATION
860.875.2199, E-mail: reprints@circuitcellar.com

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

CIRCUIT CELLAR® (ISSN 1528-0608) is published monthly by Circuit Cellar Incorporated, 4 Park Street, Vernon, CT 06066. Periodical rates paid at Vernon, CT and additional offices. **One-year (12 issues) subscription rate USA and possessions $45, Canada/Mexico $60, all other countries $63. Two-year (24 issues) subscription rate USA and possessions $80, Canada/Mexico $110, all other countries $116.** 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 © 2011 by Circuit Cellar, Incorporated. All rights reserved. Circuit Cellar is a registered trademark of Circuit Cellar, Inc. Reproduction of this publication in whole or in part without written consent from Circuit Cellar Inc. is prohibited.

Making Your Devices Smart & Simple!

Internet Connectivity Wizard

WIZnet's uniquely patented Hardware TCP/IP technology has proven effective in millions of network-enabled devices around the world, allowing users to easily add internet to the devices with greater stability to ensure faster product releases out to the market.

Hardware TCP/IP Offload Technology

- Offloading Protocol Processing from a System MCU
- Stable Hardwired TCP/IP Logic
- Silicon-proven IP for SoC and ASSP
- High Performance & Low Cost Platform
- No Booting Time
- Simple System Composition
- Best-fits OS-less Device

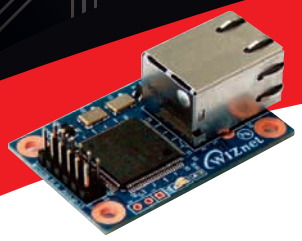
Applications

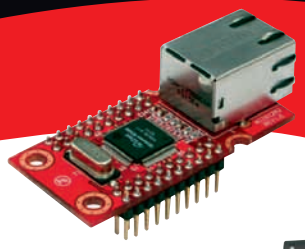
- Smart Grid and Smart Meter
- Medical Device Management
- Industrial Control
- Robot Control
- Security System



Hardware TCP/IP Chip


Wireless LAN Module


Serial-to-Ethernet Module


Drop-in Network Module



External Device Server

Get more information at www.wiznettechnology.com

3003 North First Street, San Jose, CA 408-232-5415
sales@wiznettechnology.com

WIZnet TECHNOLOGY

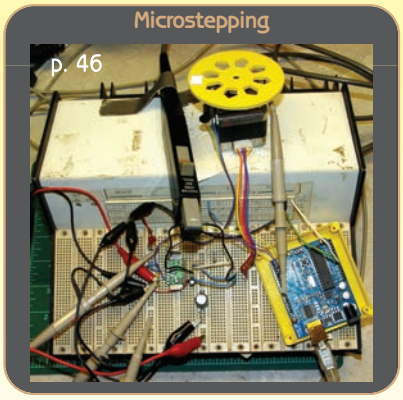
INSIDE ISSUE 255

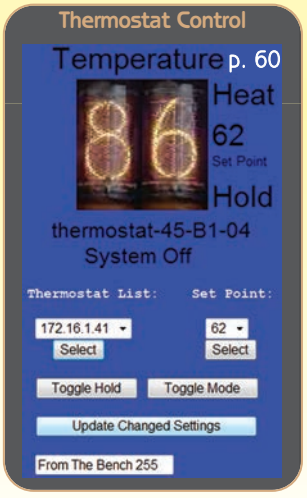
October 2011 • Signal Processing

18 **Embedded System Communication**
A Control Platform for Ethernet-Enabled Systems
Francis Fernandes, Rajendra Gad, Vinaya Gad, Shane Lobo, Gourish Naik, and Jivan Parab

26 **The Ultimate Cup Warmer**
Carl Stoez

36 **High-Accuracy Voltage Reference Using PWM (Part 1)**
Pulse-Width Modulation Theory
David Ludington







46 **ABOVE THE GROUND PLANE**
Stepper Motor Drive (Part 2)
Microstepping
Ed Nisley

52 **THE CONSUMMATE ENGINEER**
Environmental Stress Screening
George Novacek

56 **EMBEDDED IN THIN SLICES**
Getting Started with Embedded Linux (Part 1)
When to Choose Linux for an Embedded Design
Bob Japenga

60 **FROM THE BENCH**
Get Your Head Out of the Cloud
An API for Controlling a Wi-Fi Thermostat
Jeff Bachiochi

66 **THE DARKER SIDE**
Line-Coding Techniques
Robert Lacoste

TASK MANAGER 4
Get Grounded
C. J. Abate

TEST YOUR EQ SOLUTIONS 17

NEW PRODUCT NEWS 8

QUESTIONS & ANSWERS 40
Testing Devices & Writing in Assembly
An Interview with Larry Cicchinelli

CROSSWORD 74

INDEX OF ADVERTISERS 79
November Preview

PRIORITY INTERRUPT 80
What Goes Around
Steve Ciarcia

Scan Here

mouser.com
Semiconductors and electronic components for design engineers.

The **most** advanced, multilingual, multicurrency mobile site for design engineers.



mousermobile™

Compatible with more than 25 mobile platforms, no one supports more phones and tablets than Mouser. Get What's Next now at **m.mouser.com**

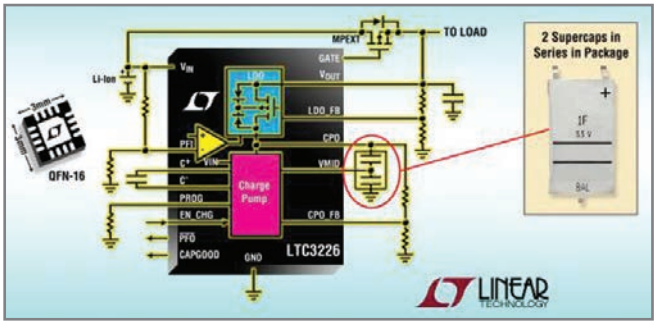
mouser.com | The Newest Products For Your Newest Designs®

MOUSER ELECTRONICS®
a tti company

Mouser and Mouser Electronics are registered trademarks of Mouser Electronics, Inc. Other products, logos, and company names mentioned herein, may be trademarks of their respective owners.

SUPERCAPACITOR CHARGER WITH POWERPATH CONTROL & AUTO CELL BALANCING

The **LTC3226** is an inductorless supercapacitor charger with a back-up PowerPath controller for li-ion or other low-voltage system-rails in applications requiring short-term back-up power. The device employs a low-noise, dual-mode (1x/2x), charge-pump architecture with constant input current to charge two supercapacitors in series from a 2.5-V to 5.5-V input supply to a programmable capacitor charge voltage between 2.5 V and 5.3 V. The charge current is resistor-programmable up to 150 mA. The device's automatic cell-balancing and voltage-clamping features maintain equal voltages on both cells without the need for balancing resistors. This protects each supercapacitor from overvoltage damage that could otherwise be caused by mismatches in cell capacitance or leakage and minimizes the current drain on the capacitors.



The LTC3226 has two modes of operation: Normal and Back-up. The operating mode is determined by a programmable power fail (PFI) comparator. In Normal mode (PFI high), power flows from V_{IN} to V_{OUT} through a low-loss external FET ideal diode and the charge pump stays on to top off the supercapacitor stack. In Back-up mode (PFI low), the charge pump is turned off and the internal LDO is turned on to supply the V_{OUT} load current from the stored supercapacitor charge while the external ideal diode prevents reverse current flow into V_{IN}. Up to 2 A of back-up current can be provided from the supercapacitor through the internal LDO.

The LTC3226 operates with a low 55-µA quiescent current when the output voltage is in regulation. The basic charging circuit requires few external components and takes up little space. The device's high 900-kHz operating frequency reduces the size of external components. Internal current limit and thermal shutdown circuitry enable the device to survive a continuous short-circuit from the PROG, V_{OUT}, or CPO pins to ground. Other features include CAP PGOOD and V_{IN} power fail outputs (PFO) as well as a V_{OUT} RST\output for system housekeeping. Encased in a compact 16-lead, low-profile (0.75 mm) 3 mm × 3 mm QFN package, the LTC3226 operates from –40°C to 125°C.

Pricing for the E and I grade LTC3226 devices starts at **$2.67** and **$3.07** each, respectively, in 1,000-piece quantities.

Linear Technology Corp.
www.linear.com

LOW-COST, LOW-POWER, TRUE RMS-TO-DC CONVERTER

The **AD8436** is a translinear-precision true RMS-to-DC converter that offers 100 µV to 3 V of usable dynamic range and the ability to measure extremely low signal levels with high accuracy and no drop in settling time. It features AC conversion accuracy to ±0.5%. Uniform settling for a wide range of input values enables consistent conversion rates. Results are displayed promptly for near-zero and high-level AC inputs.



The low-cost converter delivers instant true root mean square (RMS) results without any programming or processing expenses. Its 4 mm × 4 mm package is integrated with a FET input and precision DC buffer amplifiers and consumes only 300 µA of power. The AD8436 is immune to temperature, crest factor, and supply voltage variations making it well-suited for use in remote, portable battery-powered instrumentation and energy and automotive applications where limited board space and outdoor environments are considerations.

RMS is an important measurement of the magnitude of an AC and DC signal. It provides a better measurement of the power in a signal than simple averaging. RMS-to-DC converters are used in applications where precise, low-power measurement of AC/DC signals is required. The AD8436 system-on-a-chip design computes the precise DC equivalent of the RMS value and provides a level of linearity and accuracy across the entire input range that enables engineers to design systems with 90 dB of dynamic range.

The AD8436 extends the crest factor to 10 with minimal loss of accuracy. It is useful for high-transit signals and can be used in designs that currently use ADI's low-power, true RMS-to-DC converters, such as the AD736 and AD737. The converter is laser trimmed, characterized at 300-mV RMS input, and comes with dual- and single-power supplies. The AD8436 can operate in the extended industrial temperature range of –40°C to 125°C. It is also available in a 0°C to 70°C temperature range.

Pricing for the AD8436 is **$2.95** each in 1,000-piece quantities for the 0°C to 70°C temperature range and **$4.95** each in 1,000-piece quantities for the –40°C to 125°C temperature range.

Analog Devices, Inc.
www.analog.com

NEW PRODUCT NEWS

PCB ASSEMBLY PCB ASSEMBLY

1st in the Industry

Full Turnkey Online Quote!

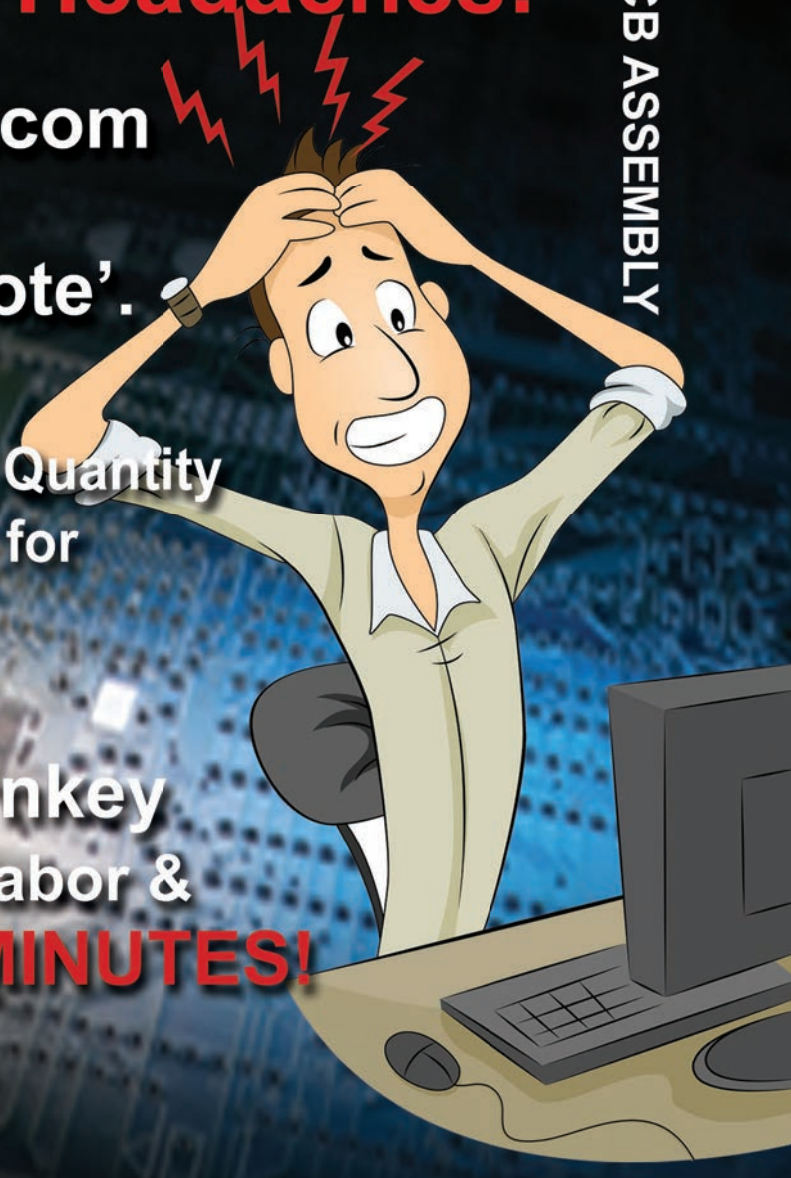
Why suffer from Procurement Headaches!

PCB ASSEMBLY PCB ASSEMBLY

Go to www.PCBnet.com and click on 'Online Turnkey Quote'.

- Just upload your BOM
- Define Part Number and Quantity
- Enter few specifications for Bare Boards

...and you are **DONE**.



You will get Full Turnkey Price (Components, Labor & Bare Boards) within MINUTES!

imagineering inc.
SINCE 1985

www.PCBnet.com
847-806-0003 Fax: 847-806-0004
email: sales@PCBnet.com

10-CHANNEL ELECTROMECHANICAL RELAY INTERFACES

DNA/DNR-DIO-470 10-channel, high-current, electromechanical relay boards are designed for use with UEI's Cube and RACKtangle I/O series. The DIO-470 boards can be used in a variety of high-current switching and digital-control applications. Each channel is configured as a standard Form C (SPDT) relay and switches voltages up to 140 VDC or 150 VAC. Each channel is rated for continuous operation at 4 A at 30 VDC and 5 A at 125 VAC.



All relays default to "NC" on power up/reset. Switching rates up to 125 Hz are supported. Each relay is protected by an on-board 8-A fuse. To support the high current ratings, the DNx-DIO-470 board uses a special version of the 37-pin D connector, which is compatible with the DNA-CBL-470HC series cables and the DNA-STP-470 screw terminal panel.

The Cube architecture is a compact (4" × 4" × 4" or 4" × 4" × 5.8") Ethernet-based I/O platform. It can be deployed in five configurations: Ethernet I/O slaved to a host PC, stand-alone data logger/recorder, Linux-based programmable automation controller (UEIPAC), Simulink/RTW target (UEISIM), or Modbus TCP-based I/O slave (UEI Modbus). Each Cube has a core module (with a processor and a network interface) and three or six open I/O slots. Users select the deployment option and I/O boards—there are more than 30 to choose from—that meet their system requirements. A six-slot Cube provides up to 150 analog inputs (up to 24 ICP channels), 192 analog outputs, 288 digital I/O, 48 counter or quadrature channels, 72 ARINC-429 channels, and/or 24 serial or CAN-bus ports.

The RACKtangle I/O chassis offers a similar capability as the Cube, but in a more conventional, 3U rack form factor with easily accessed front-loading I/O boards. It also offers 12 I/O slots and supports up to 300 A/D (including up to 48 ICP channels), 394 D/A, 576 DIO, 144 ARINC-429, or 96 CAN or serial I/O ports. The Gigabit interface also provides two independent ports, one used for primary system control and the other used for real-time diagnostics. The HalfRACK chassis is the same except with six slots.

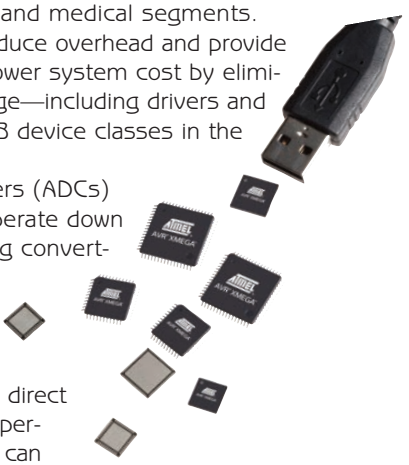
Contact United Electronic Industries for pricing.

United Electronic Industries, Inc.
www.ueidaq.com

ULTRA-LOW-POWER MICROCONTROLLERS WITH USB & HIGH-PRECISION ANALOG

The **AVR XMEGA** family of microcontrollers features a low power consumption of 100 nA with 5-µs wake-up time. The 8/16-bit microcontrollers lower overall system cost through high integration, capacitive touch support, and ultra-low-power consumption. They are designed for applications in the industrial, consumer, metering, and medical segments.

The microcontrollers integrate full-speed USB connectivity with unique functions that reduce overhead and provide high data rates. Designers can use the AVR XMEGA's high-precision internal oscillator to lower system cost by eliminating the crystal oscillator traditionally required for full-speed USB. A free software package—including drivers and communication stacks for AVR microcontrollers—is provided by Atmel for all common USB device classes in the AVR Software Framework.



The AVR XMEGA family of microcontrollers include two 12-bit analog-to-digital converters (ADCs) with programmable gain stages, so there is no need for external amplifiers. The ADCs operate down to 1.6 V and have a combined sample rate up to 4 MSPS. The two 12-bit digital-to-analog converters (DACs) also support systems that require fast and high-precision analog output. The DACs are capable of driving high loads to reduce external driver component costs. Built-in current outputs enable embedded applications to remove external resistors or other constant current sources.

The AVR XMEGA family includes full-speed USB, fast and high-precision analog systems, a direct memory access (DMA) controller, and an innovative event system that maximizes real-time performance and throughput while reducing CPU load. Peripherals and communication modules can use the DMA system to move data giving the AVR XMEGA CPU more idle time to save power or to perform other tasks. The event system enables direct interperipheral signaling for a short and predictable response time without interrupt and CPU usage. Designers can develop a solution with predictable real-time performance and data throughput even under a high system load. Other functions—such as hardware AES and DES encryption and decryption—ensure fast and low-power secure communication. Cryptography protects intellectual software property during remote programming and firmware distribution.

AVR XMEGA microcontrollers can easily realize robust touch sensing interfaces through Atmel's QTouch Library, which enables buttons, sliders, wheels, or proximity for user interfaces. All devices include the Atmel picoPower ultra-low-power technology with true 1.6-V operation, accurate real-time clock (RTC) operation, and full data retention at a current consumption of 500 nA.

Pricing for the AVR XMEGA microcontrollers starts at **$2.00** each in 10,000-piece quantities.

Atmel Corp.
www.atmel.com

PCB Fabrication PCB Fabrication PCB Fabrication

PCB Fabrication PCB Fabrication

5 Competitive Advantages

Quick-Turn Production

CAM USA

Door to Door Delivery

Overseas Manufacturing



1. Overseas Manufacturing

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

2. CAM USA

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

3. Quick-Turn Production

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

Significant Price Saving

Capabilities

- Up to 42 Layers
- Blind Buried Vias
- HDI Micro Vias
- Di-Electric Thickness
- Impedance Control (TDR Tested)
- Plated Edge Holes
- Up to 6oz Copper
- 6 mil Laser Drill
- 2 mil line width/spacing
- Conductive Silver Filled Vias
- Epoxy Filled Vias
- Aluminum Metal Core Boards
- ITAR, ISO 9001 : 2008

4. Shipping Logistics

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

5. Significant Price Saving

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

Over the past 5 years, 90,000 prototypes have been successfully delivered from overseas to over 5000 customers.

imagineering inc.
SINCE 1985

www.PCBnet.com
847-806-0003 Fax: 847-806-0004
email: sales@PCBnet.com

A GOLDEN OPPORTUNITY

The price of gold just went *down* at CIRCUIT CELLAR!

Purchase Circuit Cellar's USB archive, **CC Gold** at $25.00 off the regular price and load it up with select add-ons for only $5.00 each.



Available add-ons:
Previous Contest Projects
Elektor Back Issues
audioXpress Back Issues

Archive includes:
All Circuit Cellar issues in pdf format that are in print through date of purchase. Also includes associated project code.

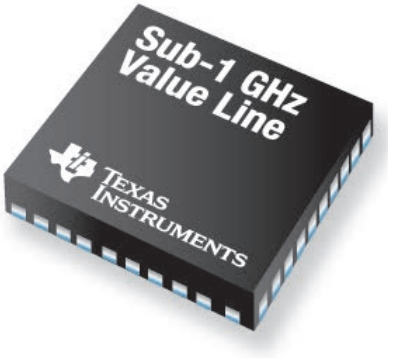
Free upgrade to 16 gigabyte USB during this promotion.

Limited time offer.
For details and to purchase visit:
www.cc-webshop.com

LOW-COST SUB-1 GHz RF PRODUCT FAMILY

The Texas Instruments **Sub-1 GHz RF Value Line** offers developers low-cost connectivity solutions for sub-1 GHz RF applications such as remote controls, toys, home and building automation, and security systems. The product family launched with the CC115L transmitter, CC113L receiver, and CC110L transceiver. In high volume, the line offers a complete, one-way RF link for less than $1. The new devices are based on TI's Sub-1 GHz CC1101 RF technology. They are pin-, register- and code-compatible. They are also backwards compatible with existing sub-1 GHz systems.

The CC11xLDK-868-915 and CC11xLEMK-433 development kits provide complete hardware performance testing and software development platforms for the new product family. The kits also offer low-cost reference designs with a compact printed circuit board (PCB) antenna for 433, 868, and 915 MHz, featuring a low component count and complete design ready for FCC and ETSI certification. The small, compact PCB helix antenna is only a quarter of the size of previous PCB antennas, and is useful for applications requiring high antenna efficiency in a compact area.



Additional features and benefits of the Sub-1 GHz RF Value Line include low-power and a long battery lifetime, including fast startup time with 0.24 ms from power-down to RX or TX and 0.2-µA sleep current. The product line is also easy to switch between one-way and two-way solutions. It features long-range communication (seamless integration with CC1190 range extender providing up to 27-dBm output power and –120 dBm sensitivity, and up to 20-dBm output power without frequency hopping under FCC). It is also compatible with CC1190 range extender, MSP430 Value Line microcontroller, and TPS62730 Step Down Converter with Bypass Mode for Ultra Low Power Wireless Applications.

The Sub-1 GHz RF Value Line is packaged in a ROHS-compliant, 4 mm × 4 mm QFN-32. Pricing starts at **$0.75** in 1,000-unit quantities. The CC11xLDK-868-915 development kit costs **$299** at TI's online store. An additional CC11xLEMK-433 development kit is available for **$99**.

Texas Instruments, Inc.
www.ti.com

NPN





10GBASE-T CONNECTOR MODULE

The **10GBASE-T** (10 GB over twisted-pair) integrated connector module (ICM) is for servers, LAN on motherboard (LOM), adapter cards, and storage devices. This 1 × 1 RJ-45 10GBASE-T platform is halogen-free and RoHS-compliant. The module offers a choice of multiple LED configurations and EMI tab options. It meets PCIe mechanical requirements and supports the latest 10GBASE-T physical layer device (PHY) technology and applications.



10GBASE-T products meet or exceed IEEE 803.an requirements and are tuned to have flat insertion loss curves and excellent return loss performance at high frequencies. The PHY-side return loss up to 400 MHz meets or exceeds return loss requirements. A low insertion loss over the 1 MHz to 400-MHz band enables longer cable reach. This platform is designed to help meet EMC system requirements, such as FCC Part 15 class B.

The 10GBASE-T connector modules are packaged in trays. They are priced at approximately **$8.50** each with volume discounts available.

Pulse Electronics Corp.
www.pulseelectronics.com



MCUs WITH REAL-TIME CONTROL

The **RL78/G14** family of 16-bit microcontrollers is designed for applications such as household appliances, healthcare devices, office equipment, industrial automation systems, motor control, security systems, and mobile devices. The microcontrollers combine the on-chip peripheral functions of Renesas's R8C microcontroller family with the technology of the RL78 family.

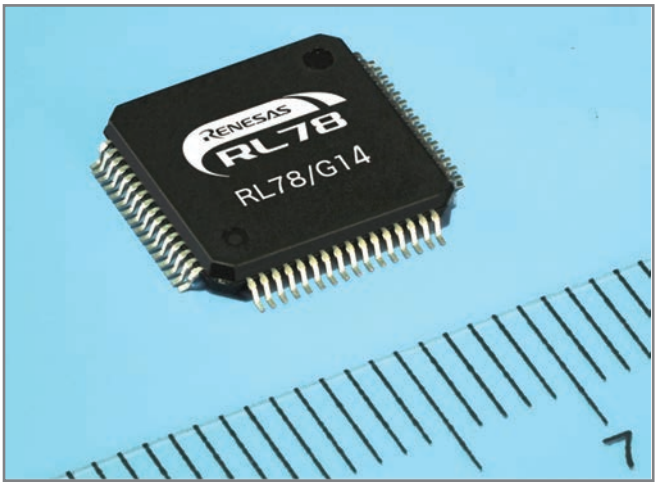
RL78/G14 microcontrollers incorporate advanced timer modules—Timer RD, Timer RG, and Timer RJ—that are compatible with timers in the R8C family. Timer RD is comprised of two 16-bit timers capable of operating at 64 MHz and a pulse-width modulation (PWM) function that outputs six three-phase waveforms with a user-specified amplitude. Timer RG is a 16-bit timer module that automatically measures the counts of a two-phase encoder in phase-counting mode. Timer RJ is a 16-bit timer module that measures the pulse width or cycle of pulse output or external pulse input.

The RL78/G14 microcontrollers integrate a data transfer controller (DTC) and event link controller (ELC), which reduce the CPU load and lower current consumption. The DTC enables memory-to-memory data transfers without the use of a CPU. Compared with the DMA function of the RL78/G13 device, the DTC supports a larger number of transfer channels and activation sources. It also enables data transfers from flash memory. The ELC enables direct connection between peripherals bypassing the CPU and interrupt controller to provide faster and more deterministic real-time control. The ELC can also be used to start peripheral functions when the CPU is stopped, which reduces current consumption.

The microcontrollers add, multiply, divide, and multiply and accumulate instructions to the instruction set of the RL78/G13 CPU core, eliminating the need to handle overflow interrupts when performing arithmetic operations. In addition, the on-chip debugging functions now support trace data for up to 256 branches, which enables efficient system development and evaluation.

Contact Renesas for pricing.

Renesas Electronics Corp.
www.renesas.com



NPN

www.DevtoolsXpress.com

Pick a Tool. Any Tool.

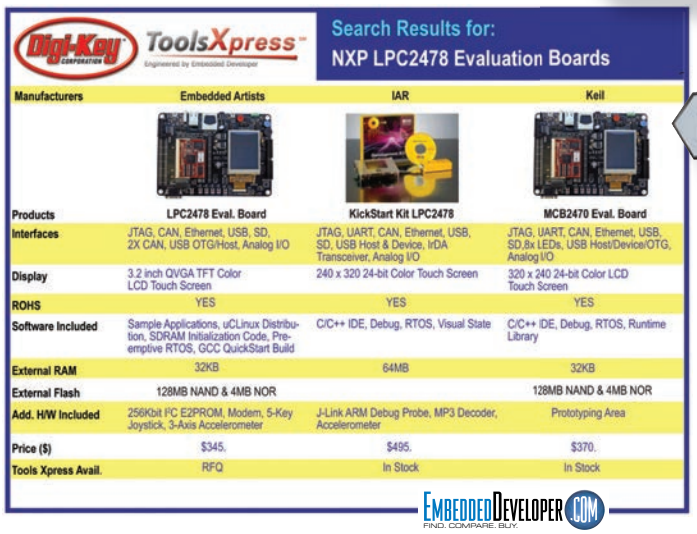
Find the Right Development Tool, Compare it to Other Tools, Evaluate It, and Buy It from Digi-Key Tools Xpress -- Without Leaving Our Site.

The Digi-Key Tools Xpress intuitive research engines are used by engineers worldwide to locate, compare and evaluate hardware or software development tools.



Digi-Key Tools Xpress, engineered by Embedded Developer, is the only site in the industry where engineers can quickly find, compare and buy the leading development tools.

Compare before you buy: tools are listed side-by-side, with features and performance specs, availability, and prices, so you can make an educated decision!


Digi-Key ToolsXpress

Search Results for:
NXP LPC2478 Evaluation Boards

Manufacturers	Embedded Artists	IAR	Keil
Products	LPC2478 Eval. Board	KickStart Kit LPC2478	MCB2470 Eval. Board
Interfaces	JTAG, CAN, Ethernet, USB, SD, 2X CAN, USB OTG/Host, Analog I/O	JTAG, UART, CAN, Ethernet, USB, SD, USB Host & Device, IrDA Transceiver, Analog I/O	JTAG, UART, CAN, Ethernet, USB, SD,8x LEDs, USB Host/Device/OTG, Analog I/O
Display	3.2 inch QVGA TFT Color LCD Touch Screen	240 x 320 24-bit Color Touch Screen	320 x 240 24-bit Color LCD Touch Screen
ROHS	YES	YES	YES
Software Included	Sample Applications, uCLinux Distribution, SDRAM Initialization Code, Preemptive RTOS, GCC QuickStart Build	C/C++ IDE, Debug, RTOS, Visual State	C/C++ IDE, Debug, RTOS, Runtime Library
External RAM	32KB	64MB	32KB
External Flash	128MB NAND & 4MB NOR		128MB NAND & 4MB NOR
Add. H/W Included	256Kbit I²C E2PROM, Modem, 5-Key Joystick, 3-Axis Accelerometer	J-Link ARM Debug Probe, MP3 Decoder, Accelerometer	Prototyping Area
Price ($)	$345.	$495.	$370.
Tools Xpress Avail.	RFQ	In Stock	In Stock

EmbeddedDeveloper.com
FIND. COMPARE. BUY.



Join the thousands of engineers worldwide who use Digi-Key Tools Xpress for their development tool needs.

Digi-Key CORPORATION

ToolsXpress℠
Engineered by Embedded Developer

HUMANDATA

FPGA/CPLD Stamp Module PLCC68 Series

Easy and Quickly Mountable Module

FPGA Module PLCC socket mountable



- 50 I/Os (External clock inputs are available)
- 3.3V single power supply operation (Voltage converters for auxiliary power supply are built-in)
- Separated supply-inputs: power, I/O drivers
- JTAG signal
- All PLCC68 series have common pin assignment
- Very small size (25.3 x 25.3 [mm])
- RoHS compliance
- MADE IN JAPAN



XILINX PLCC68 Series

XP68-01 Spartan-6 PLCC68 FPGA Module



NEW Spartan-6 PLCC 68

XC6SLX16-2CSG225C
16Mbit Configuration SPI-ROM
Two User LEDs
One User Switch(Slide)
RoHS compliant

XP68-02 Spartan-3AN PLCC68 FPGA Module

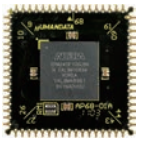


NEW Spartan-3AN PLCC 68

XC3S200AN-4FTG256C
FPGA internal configuration ROM
Two User LEDs
RoHS compliant

ALTERA PLCC68 Series

AP68-01 MAX II PLCC68 CPLD Module



NEW MAX II PLCC 68

EPM240F100C5N
EPM570F100C5N
External Clock inputs
RoHS compliant

AP68-02 MAX V PLCC68 CPLD Module



NEW MAX V PLCC 68

5M570ZF256C5N
External Clock inputs
On-board Voltage regulator
RoHS compliant

AP68-03 Cyclone III PLCC68 FPGA Module



NEW Cyclone III PLCC 68

EP3C5U256C8N
EP3C10U256C8N
4Mbit Configuration ROM
Two User LEDs
One User Switch(Slide)
RoHS compliant

Over 100 varieties of FPGA/CPLD boards are available.



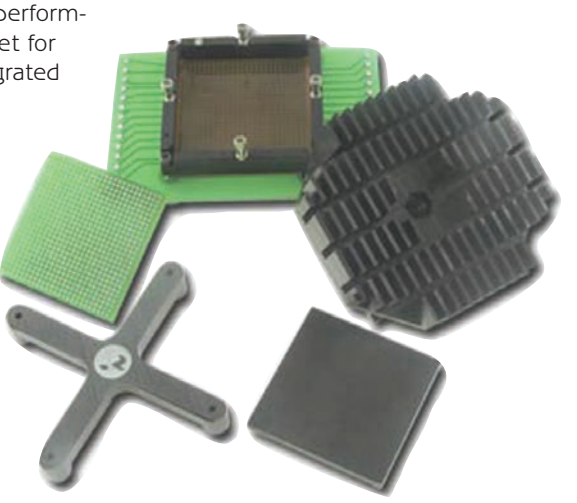
See all our products at:

www.hdl.co.jp/CC/

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

8-GHz BANDWIDTH SOCKET FOR STMicro's 616 FPBGA

The SG-BGA-6346 is a high-performance ball-grid array (BGA) socket for 1-mm pitch, 616-pin BGA integrated circuits (ICs). The socket is designed for 27 mm × 27 mm package size and operates at up to 8-GHz bandwidth with less than 1-dB of insertion loss. The sockets are designed to dissipate 4.5 W with passive heat sinking (natural convection) and can manage up to 40 W with active heat sinking (DC fan on top of heatsink). The contact resistance is typically 20 mΩ per pin. The socket connects all pins with 8-GHz bandwidth on all connections.



The socket is mounted on the target printed circuit board (PCB) without soldering, and has a small 2.5-mm footprint. The socket is constructed with a shoulder screw and a swivel lid that incorporates a quick insertion method so ICs can be quickly changed out. It features high-performance, low-inductance elastomer contactor. The temperature range is –35°C to 100°C. The pin self inductance is 0.15 nH with a mutual inductance of 0.025 nH. Capacitance to ground is 0.01 pF. Current capacity is 2 A per pin. The socket is designed to work with ICs such as STMicro's FPBGA, 27 mm × 27 mm with 26 × 26 array and 1-mm pitch.

Pricing for a single-quantity SG-BGA-6346 socket begins at **$807**.

Ironwood Electronics
www.ironwoodelectronics.com

NPN

SPECIAL ANNOUNCEMENT FROM ELEKTOR

Get ready for the "Elektor Academy Webinar Series in Partnership with element14"

In response to requests from many enthusiastic readers out there, Elektor Academy and element14, the industry's leading online technology portal, have teamed up to bring you a series of five exclusive webinars (web seminars) covering blockbuster projects from recent editions of *Elektor* magazine.

In an Elektor Academy/element14 webinar, those who have registered to attend our online sessions not only get to see and hear leading authors and project designers exploring, discussing, and elaborating on their projects, but also have an opportunity to enter into a debate on design and other technical aspects during the live Question & Answer slot after each presentation. Plus, every attendee receives an exclusive 10% discount off their next purchase with Farnell!

The first Elektor Academy/element14 webinar is titled "Platino—An Ultra-Versatile Platform for AVR Microcontroller Circuits" and will take place on October 13, 2011 at 3:00 EST, live at www.element14.com.

Participation in these webinars is completely free! All you need to do is register via www.elektor.com/webinars to reserve your space. Space is limited, so book soon to make sure you don't miss out.

Future Elektor/element14 Webinar Series topics and times will be announced through *Elektor*'s e-weekly newsletter as well as in our monthly publications and online so watch out for the next exciting episode!

Elektor International Media Publishes Circuit Cellar *magazine*

CIRCUIT CELLAR

Test Your EQ

Edited by David Tweed

ANSWERS for Issue 254

Answer 1—For simplicity, let's consider a D-type flip-flop, with one clock input "C," one data input "D," and one output "Q." The D input is a voltage that is a function of time. For any given instance of this flip-flop, there is a point for each clock edge in a two-dimensional space, defined in terms of voltage and time relative to the clock edge. If the D input value passes through this point, it will result in the flip-flop going into a metastable state for some indefinite amount of time.

The specific position of this point varies with manufacturing tolerances as well as specific operating conditions such as temperature and supply voltage. Therefore, the manufacturer cannot give you a specific location for this point and must instead describe a "window" in which the point is expected (with high probability) to exist. The window is described in the voltage dimension in terms of VIL and VIH and in the time dimension in terms of TSU and TH. As long as all D input transitions are kept out of this window, the flip-flop is guaranteed not to go metastable.

Answer 2—The fundamental principle is called "Buridan's Principle" after the 14th-century French philosopher Jean Buridan, who stated, in essence, that no discrete decision about a continuous variable can be made in a bounded amount of time. The canonical example of this is the donkey placed equidistant from two bales of hay that will starve to death because he can't decide which one to eat first.

The flip-flop is the electrical manifestation of this principle. A discrete decision is called for at each active clock edge, yet the D input can actually take on any of a continuous set of input values (voltages). It's the job of the flip-flop to decide at the clock edge whether the input is the "low" state or the "high" state. But, if the input is changing at the time of the clock edge, it can't always make that decision right away. Another way of looking at it is to ask whether the data changed before or after the clock edge.

Answer 3—No. Each new clock edge represents a fresh decision, so any state from a previous clock edge, including a metastable state, is erased.

The only way to have a given flip-flop in a metastable state for multiple clock periods is to create a new metastable state at each clock edge. In other words, the D input would have to be changing on every clock edge.

It is a fundamental principle in the design of asynchronous interfaces—such as UARTs—that input states must persist for at least two sampling clocks. In fact, most modern UARTs use a 16× clock.

That's not to say that a metastable state can't propagate down a chain of continuously-clocked flip-flops, but the probability of each successive flip-flop in the chain going metastable decreases exponentially.

Answer 4—A plesiochronous system is one in which everything is tied to a master clock so that the same clock frequency is used throughout the system, but because of unknown delays within the system, phase relationships can be completely arbitrary. A classic example is a T1 telephone circuit (1.544 Mbps digital data), where one end is the timing master and the other end uses "loopback" timing to derive its transmit clock from the received signal. By the time the data gets back to the master, it can have arbitrary phase because of the unknown path delays.

Metastability can be a major concern in such systems, but as long as asynchronous design principles are followed, plesiochronous operation does not cause any additional problems.

Contributed by David Tweed

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

You may contact the quizmasters at eq@circuitcellar.com



by Francis Fernandes, Rajendra Gad, Vinaya Gad, Shane Lobo, Gourish Naik, and Jivan Parab (India)

Embedded System Communication

A Control Platform for Ethernet-Enabled Systems

Many of today's industrial and consumer electronic applications have hefty data transmission needs and requirements. This means engineers must incorporate efficient embedded technologies that can communicate effectively with each other. You can customize this SoC control platform for any Ethernet-enabled design on your workbench.

This article presents a system-on-chip (SoC) developed using an Altera Nios II softcore processor, a general embedded system platform for Ethernet-enabled appliance control. The embedded system is implemented on Altera's Cyclone-II FPGA. The designed platform is flexible and versatile and can be customized for any Ethernet-enabled appliance or smart appliance. The client-side frontend interface is .NET-based for a rich and user-friendly design (see Photo 1).

The .NET framework is used by the remote client to connect to the server which uses a Nios II processor. The platform can also be used to implement TCP/IP control for complex control systems. The design uses the lightweight Internet protocol (lwIP) stacks module of Micriμm's μC/OS real-time kernel as a design handshake over the session layer of the OSI model.

THE NEED FOR ETHERNET

The idea of using Ethernet for industrial communications is not a novelty. However, in the 1980s, the idea faced strong reluctance from users because Ethernet was non-deterministic. This randomness caused Ethernet to be rejected in industrial environments because users demanded real-time guarantees, such as a minimum transfer time, a jitter not exceeding some threshold, or some guaranteed bandwidth. The result was the development of a variety of protocols offering deterministic behavior and a pallet of solutions under the name "fieldbus" (e.g., WorldFIP, Profibus, P-Net, Interbus, AS-Interface, SERCOS, LonWorks, MVB, MIL-STD-1553, DeviceNet, SDS, and CAN).

Industrial and automotive networks now carry more information, higher data volumes, images, voice, and even video streams. Hence, existing solutions must be upgraded

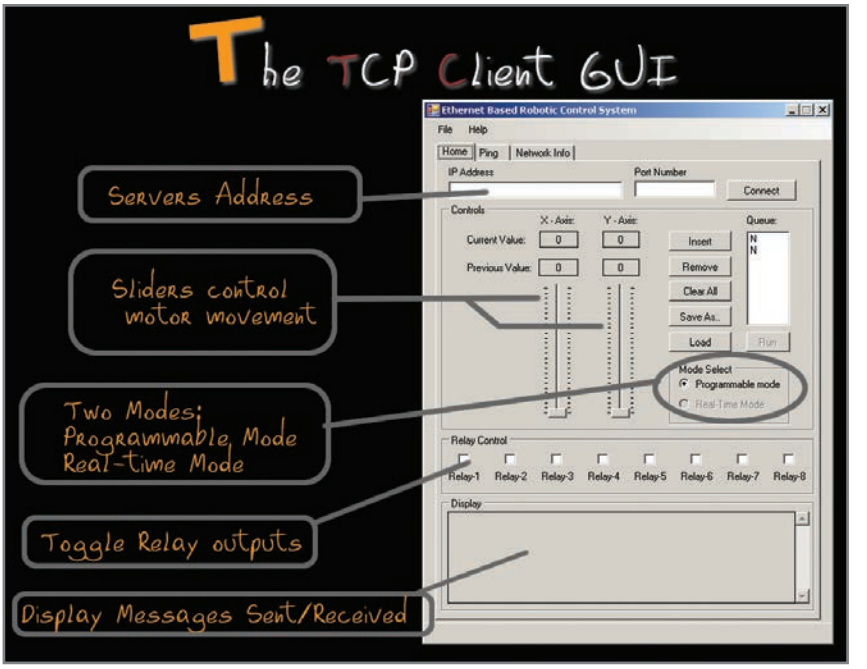


Photo 1—Front-end interface design using .NET on the client side

by increasing the data transmission rate and improving performance.

Ethernet and all the protocols linked to Ethernet—Internet protocol (IP), user datagram protocol (UDP), and transmission control protocol (TCP)—are meeting with such success that network interface costs have dropped sharply. In addition, Ethernet is available at ever-increasing speeds. Gigabit speeds are now commonplace and 10-Gbps interface cards will soon be customary.

With the rapid increase in embedded systems, the need for intercommunication between them grew. Also, some devices needed to be remotely operated or monitored. Communication or networking means needed to be provided. Network communications can make an embedded system more powerful and easier to monitor and control. TCP/IP can be implemented on embedded systems by using a special TCP/IP stack that fits into the memory of the embedded system (e.g., lwIP, smxNS, USNet, and Nichestack). Usually some real-time operating system (RTOS), such as Micriμm's μC/OS-II, is required to implement the TCP/IP stack. Here, we use the lwIP TCP stack and μC/OS-II on a Nios II processor. There are several vendors in the embedded Ethernet solution (e.g., Maxim and Freescale) that also provide these solutions.

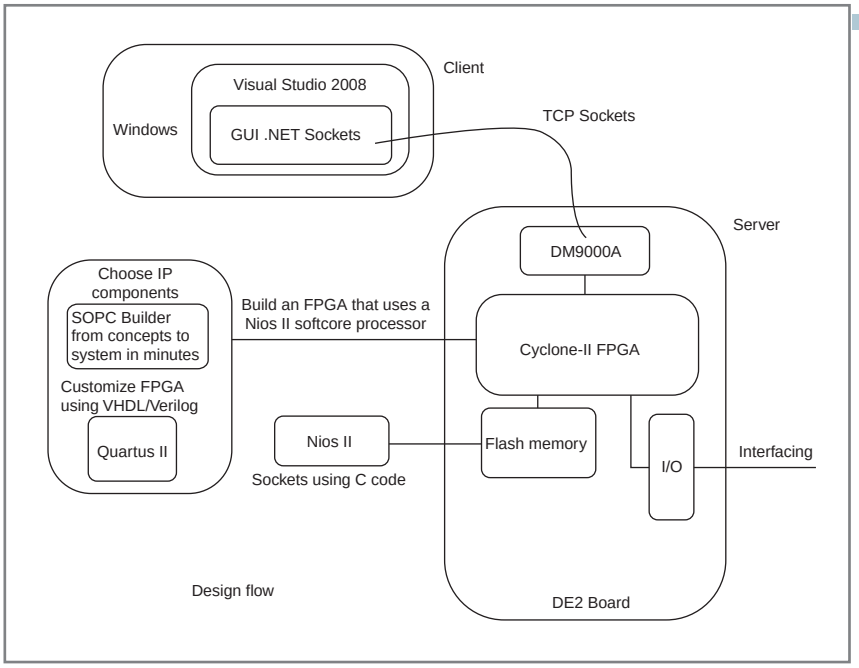


Figure 1—Block diagram of the TCP/IP control platform

EMBEDDED SERVER PLATFORM DESIGN

Figure 1 represents the client-server system configuration over an Ethernet network. The client has a .NET frontend interface for user-friendly purposes and the server is running socket programming at the backend using an lwIP TCP/IP stack over the Nios II softcore processor on a Cyclone-II FPGA. LwIP is a lightweight implementation of the TCP/IP protocol suite. The reason for the TCP/IP stack is that the footprint of such a stack is small compared to a general RTOS. This is because in a real-time system, resource utilization and management are critical for performance as they are limited on the FPGA and the softcore processor. Refer to *Circuit Cellar*'s FTP site for the details of resources used for SoC. The OSI model in Figure 2 shows the respective layers responsible for the establishment of the connections.

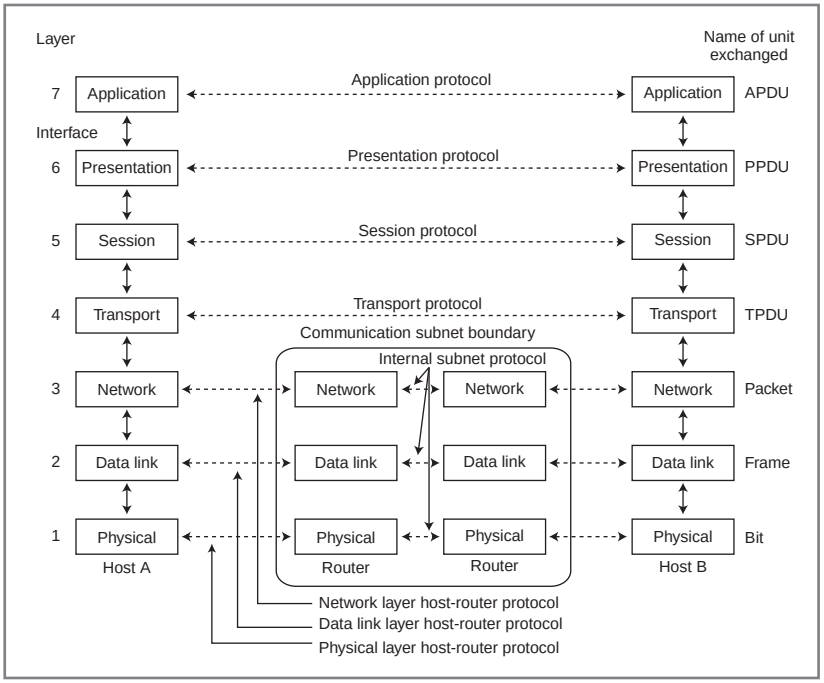


Figure 2—Session layer in OSI model (dashed line)

The physical and data-link layers of the model are programmed on the Ethernet Chip for IPv4 configuration on a Davicom Semiconductor DM9000A 16-bit 10/100-M Ethernet controller chip.

CLIENT-SERVER INTERFACE

Server side, the interface is designed using C/C++-based Nios II IDE supported on Altera's Quartus II design software. Few icons, like switching and reading the status of I/Os and controlling the motor, are provided for testing and troubleshooting the embedded system server platform.

Note the client interface can be HTML, XML, RTOS, .NET framework, or any portable application. If the Ethernet-enabled device is a PDA or mobile phone then portability and compatibility of the client interface is an issue. According to Gartner Inc., the number of installed PCs worldwide has surpassed 1 billion units, of which 10% to 15% are Ethernet enabled. Also 10% of the world

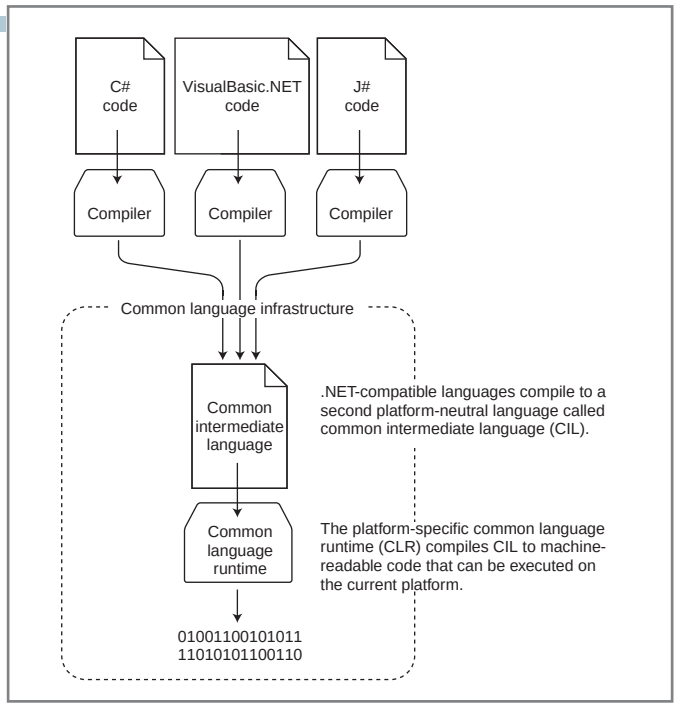


Figure 3—Common language runtime platform

population is mobile and demands home-cum-office attached services. Ethernet-enabled devices and user group are diverse and global respectively demanding a rich and highly user-friendly GUI interface for better performance.

Microsoft provides two .NET frameworks, one for general applications and another for embedded systems, such as .NET Micro Framework (NETMF), which is Microsoft's .NET platform for resource-constrained devices (e.g., ARM processors or Analog Devices's Blackfin processors). The NETMF helps make embedded development faster and easier by enabling access to the tools and technologies used by desktop (PC) application developers. NETMF has a much smaller footprint with class support of one tenth the .NET framework and provides only core functionality for writing application for mobile devices. It claims to bring the managed code advantages of the desktop .NET framework to resource-constrained embedded devices while enabling development with Microsoft's Visual Studio.

Programmers are attracted to the .NET framework for many reasons. The main reasons are its built-in features, including interoperability, common runtime engine, language independence, base class library, simplified deployment, security, and portability. Presently, we have used only the general .NET framework because of the short development cycle. This is due to the availability of a large library of coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework.

Also, Microsoft supports the following languages under .NET: Visual C#.NET, Visual J#.NET, Visual C++.NET, Visual Basic.NET, and JScript for the common language runtime (CLR). Also, many more third-party languages, such as APL, Perl, Python, and S#, are available for programming. In the .NET framework, CLR provides the appearance of an application virtual machine so programmers don't need to consider the capabilities of the specific CPU that will execute the program (see Figure 3). CLR also provides other important services, such as security, memory management, and exception handling.

Microsoft doesn't specifically claim that C# and .NET are intended for real-time systems, but many of the platform's general-purpose features, including type unsafe features, thread synchronization, and overflow-sensitive arithmetic apply to real-time systems. In addition, it offers a number of important traits, such as code portability and cross-language development. It is these features that have attracted our attention into investigating the possibility of using .NET for architecture-neutral real-time systems.

CLIENT INTERFACE USING WINSOCK

To be specific, here .NET is used in Windows XP and socket programming is implemented using the Windows Sockets (WinSock). It enables programmers to create advanced Internet, intranet, and other network-capable applications to transmit application data across the wire, independent of the network protocol used (see Figure 3).

To implement sockets in VB.NET, we used the system.net.sockets namespace, which provides a managed implementation of the WinSock interface for developers who need to tightly control access to the network. The

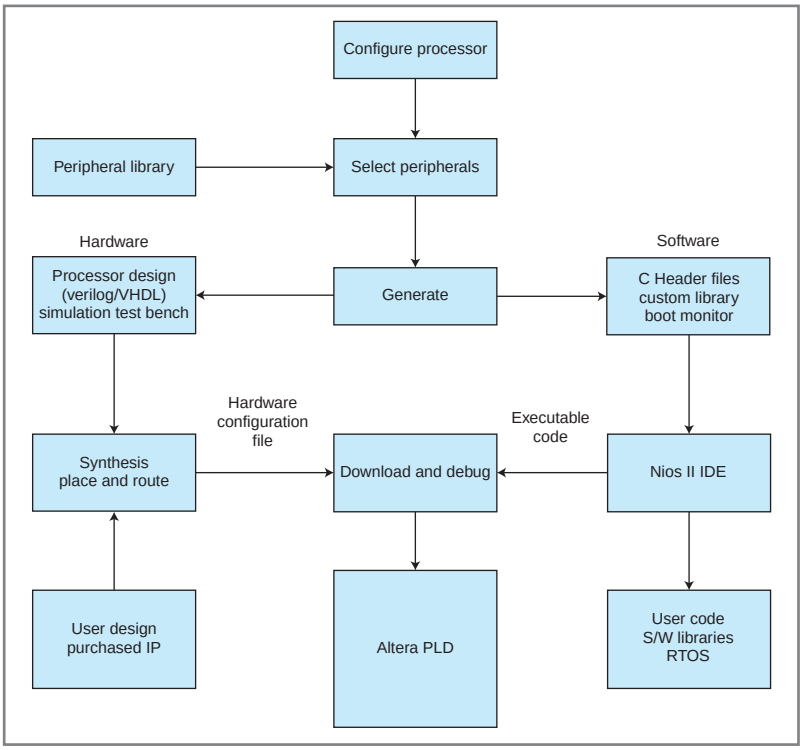


Figure 4—Stages of compilation of SoC on Cyclone-II using a Nios II processor



▶ CUWIN6300
▶ CUWIN5500
▶ CUWIN5200

Windows CE 6.0 Touch Controller

CUWIN

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

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



C-Programmable Modular Industrial Controller

MOACON

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

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

BASIC with LADDER LOGIC CONTROLLER



New

Integrated CUBLOC Controller and I/O Board

CB210

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

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

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

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

CUPC

Industrial Touch Panel PC with ATOM Processor



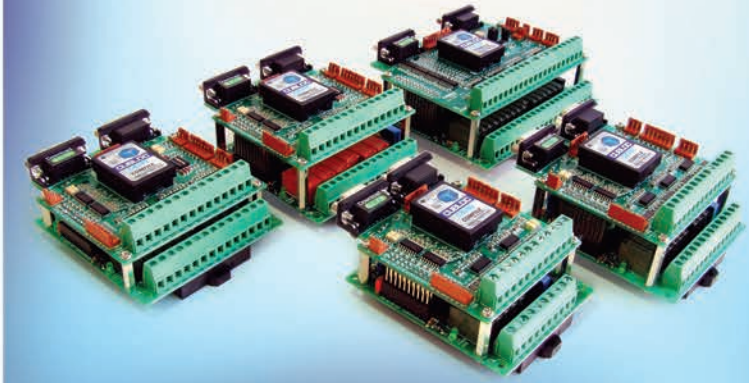
▶ CUPC-P200
▶ CUPC-P220

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

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

CUSB

Integrated Industrial Controllers



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

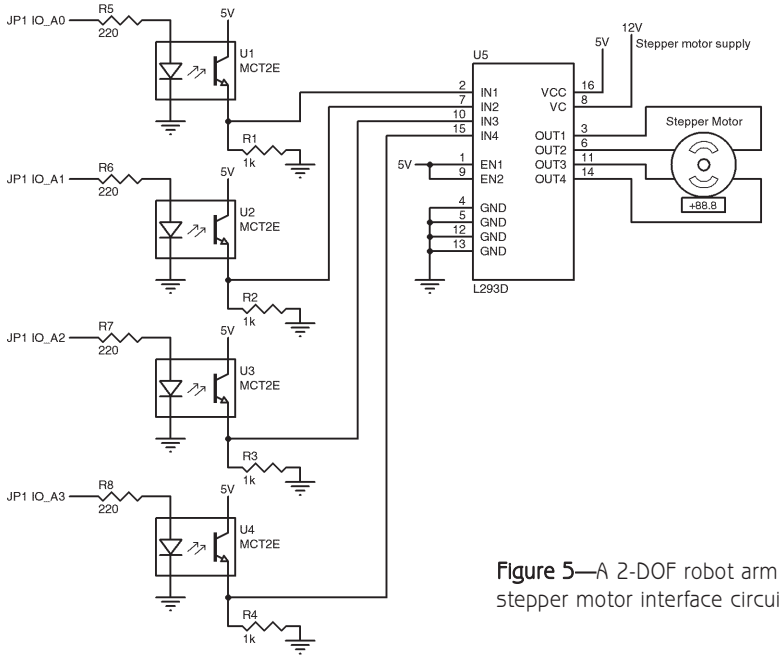


Figure 5—A 2-DOF robot arm stepper motor interface circuit

socket class provides a rich set of methods and properties for network communications. The socket class enables you to perform both synchronous and asynchronous data transfer. To process communications using separate threads during execution, .NET recommends using the following methods which are designed for asynchronous operation mode. While using a connection-oriented protocol such as TCP, use BeginConnect and EndConnect methods to connect with a listening host. Use the BeginSend and EndSend or BeginReceive and EndReceive methods to communicate data asynchronously. Incoming connection requests can be processed using BeginAccept and EndAccept. The detailed frontend interface is described in Photo 1.

SoC DESIGN

The various stages of compilation of the SoC on the Cyclone FPGA are indicated in Figure 4. Initially, the softcore processor is compiled and the required peripherals are selected. Then the system on a programmable chip (SOPC) generates the BDF file of the SoC. The SoC configuration is programmed on the FPGA Cyclone-II using Nios II C/C++ compiler IDE. Also, µC/OS RTOS is installed with the lwIP stack. The session-layer protocol is implemented using the lwIP stack, which works on the .NET application layer. A flowchart showing the interaction over the session layer is available on *Circuit Cellar*'s FTP site. A diagram of the detailed server-end interface implemented using the Nios II IDE is also available on the FTP site.

BENEFITS OF .NET

The client interface is designed to communicate with the server for a two- degrees-of-freedom (2-DOF) robotic arm control (see Figure 5 and Figure 6). Also, 16 I/O lines were tested for the on/off control. Devices that can be networked together have many applications, such as industrial automation, sale terminals, environmental monitoring (distributed sensors), alarm and security systems, remote surveillance, network-enabled peripherals (printers), PC interfacing, web servers, communication devices, robotics, and gaming consoles.

We realize that the .NET framework can be useful for developing applications that enable a device to be interfaced with a computer. .NET provides many solutions for communication using the serial port, USB, and networking. It also has solutions for web applications. The .NET compact framework can be used to write applications for some devices that use a Windows RTOS.

Microsoft Robotics Developer Studio depends a lot on the .NET framework. This makes learning to code using the .NET framework essential. .NET is slowly making its way into the Linux platform through a project called Mono. Thus, applications written using .NET might someday work on both Windows and Linux platforms.

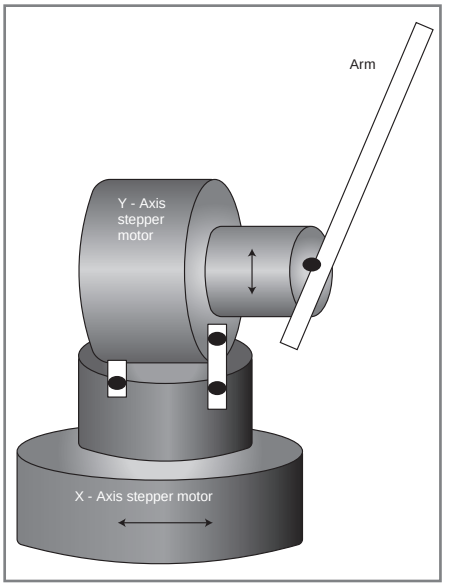


Figure 6—An assembly model of a 2-DOF robotic arm

Francis Fernandes (fernfrancis@gmail.com) is a System Administrator at Air Arabia in Dubai.

Rajendra Gad (rsgad@unigoa.ac.in) is an Assistant Professor of Electronics at Goa University in Goa, India.

Vinaya Gad (vinaya_gad@rediffmail.com) is a research scholar at Goa University in Goa, India.

Shane Lobo (shane.taurean@gmail.com) is a research scholar at the MICD, National Institute of Oceanography, in Goa, India.

Gourish Naik (gmnaik@unigoa.ac.in) is a Professor and the head of Electronics at Goa University in Goa, India.

Jivan Parab (jsparab@unigoa.ac.in) is an Assistant Professor of Electronics at Goa University in Goa, India.

PROJECT FILES

To download graphics and additional resources, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/255.

RESOURCES

Altera Corp., "Nios II, Using Lightweight IP with the Nios II Processor Tutorial," 2004, www.altera.co.jp/literature/tt/tt_nios2_lwip_tutorial.pdf.

———, "Nios II Software Developer's Handbook," 2011, www.altera.com/literature/hb/nios2/n2sw_nii5v2.pdf.

———, "Using MicroC/OS-II RTOS with the Nios II Processor Tutorial," 2011, www.altera.com/literature/tt/tt_nios2_MicroC_OSII_tutorial.pdf.

The Mono Project, http://mono-framework.com.

Swedish Institute of Computer Science, "The lwIP TCP/IP Stack," 2004, www.sics.se/~adam/lwip.

A. Zerzelidis and A. J. Wellings, "Requirements for a Real-Time .NET Framework," ACM SIGPLAN Notices, 2005.

SOURCES

Nios II processor, Nios II IDE, Cyclone-II FPGA, and Quartus II design software
Altera Corp. | www.altera.com

DM9000A Ethernet controller chip
Davicom Semiconductor, Inc. | www.davicom.com

µC/OS and µC/OS-II Real-time kernel
Micriµm | www.micrium.com

.NET Micro Framework, Visual Studio, and Robotics Developer Studio
Microsoft Corp.| www.microsoft.com

NEED-TO-KNOW INFO

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

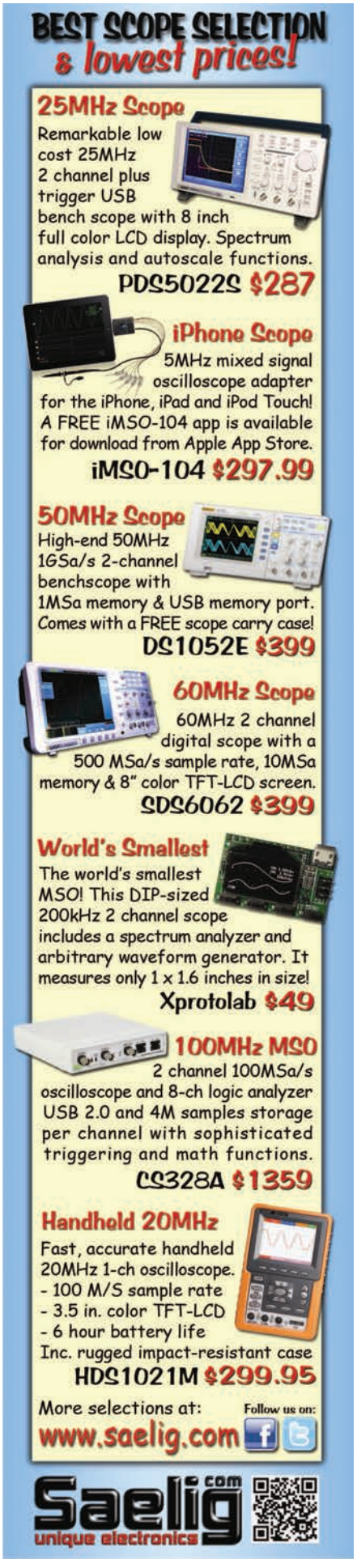
—

Multi-Input Temperature Logger
by Nial Stewart
***Circuit Cellar* 198, 2007**

Nial designed an FPGA-based multiple-input temperature logger with an innovative USB interface. In this article, he describes the FPGA logic to drive the sensors and the flexible one-pin USB interface used to connect it to a PC. Topics: Nios, Logger, Temperature, FPGA, CPLD

—

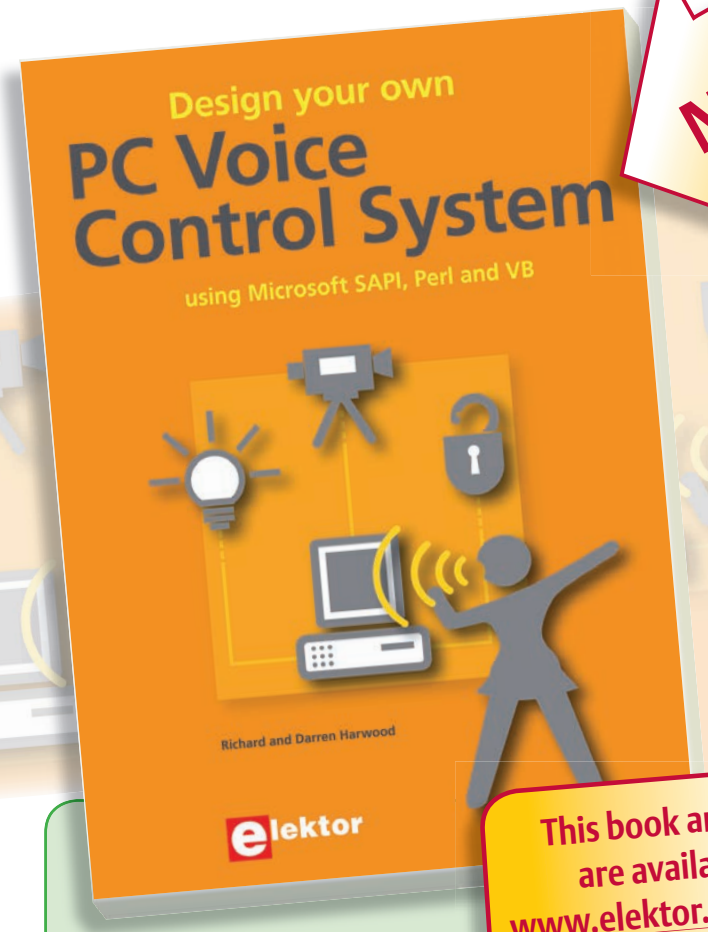
Go to *Circuit Cellar*'s webshop to find these articles and more: www.cc-webshop.com



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

Elektor Shop

The world of electronics at your fingertips!



NEW!

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

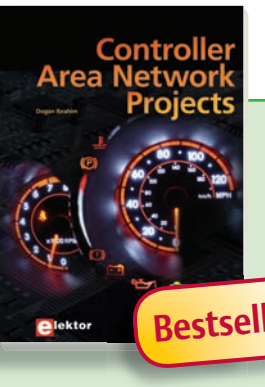
Books

User: "Computer, turn on hall lights" – Computer: "Hall lights on"

Design your own PC Voice Control System

Don't you feel it's just about time we could simply talk with computers and get them to handle some of the everyday house automation tasks - lights, doors, gates, curtains, televisions, music and other domestic appliances? Or even make the information super-highway slicker by talking with you and telling you about the information you requested to be retrieved from the web? Think of all the time and inconvenience you could save - no more need to install yourself at a desk and fumble about with keyboard and mouse. Well now you can! This book guides you through practical speech recognition, speech annunciation and control of really useful peripherals. If you are interested in the practical technology of interfacing with machines using voice, such as talking with your own PC, then this book is your guide.

Approx. 216 pages • ISBN 978-1-907920-07-3 • $47.60



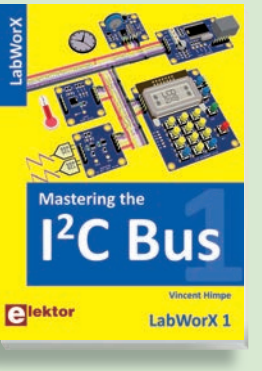
Bestseller!

Free mikroC compiler CD-ROM included

Controller Area Network Projects

The aim of the book is to teach you the basic principles of CAN networks and in addition the development of microcontroller based projects using the CAN bus. You will learn how to design microcontroller based CAN bus nodes, build a CAN bus, develop high-level programs, and then exchange data in real-time over the bus. You will also learn how to build microcontroller hardware and interface it to LEDs, LCDs, and A/D converters. This book comes complete with a free CD-ROM.

260 pages • ISBN 978-1-907920-04-2 • $47.60



LabWorX: Straight from the Lab to your Brain

Mastering the I²C Bus

Mastering the I²C Bus is the first book in the LabWorX collection. It takes you on an exploratory journey of the I²C Bus and its applications. Besides the Bus protocol plenty of attention is given to the practical applications and designing a solid system. The most common I²C compatible chip classes are covered in detail. Two experimentation boards are available that allow for rapid prototype development. These are completed by a USB to I²C probe and a software framework to control I²C devices from your computer.

248 pages • ISBN 978-0-905705-98-9 • $47.60

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

CD/DVD-ROMs

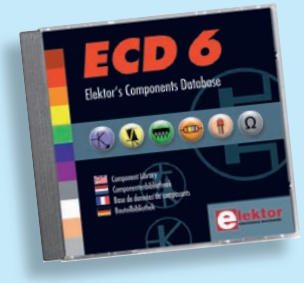


Circuits, ideas, tips and tricks from Elektor

CD 1001 Circuits

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

ISBN 978-1-907920-06-6 • $55.70



Elektor's Components Database

CD ECD 6

This CD-ROM gives you easy access to design data for over 70,000 components! The program package consists of eight databanks covering ICs, transistors, diodes and optocouplers. A further eleven applications cover the calculation of, for example, zener diode series resistors, voltage regulators, voltage dividers and AMV's. A colour band decoder is included for determining resistor and inductor values. All databank applications are fully interactive, allowing the user to add, edit and complete component data. This CD-ROM is a must-have for all Electronics enthusiasts!

ISBN 978-90-5381-258-7 • $40.20



Elektor OSPV

The Elektor Wheelie, which we launched a while ago, has been adopted by schools and private individuals who want to gain a thorough understanding of the technology behind self-balancing vehicles. This new Elektor OSPV is based on the same concept, but with the difference that it's for indoors, it's easy to steer, it's light and foldable and...it's open source. You can configure or modify it to suit your wishes. This self-balancing vehicle is perfect for factory halls, warehouses, hospitals and schools. Check www.elektor.com/ospv for further information!

Kit of parts

Art.# 110320-91 • $1570.00

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

Elektor US
PO Box 180
Vernon, CT 06066
USA
Phone: 860-875-2199
Fax: 860-871-0411
E-mail: order@elektor.com

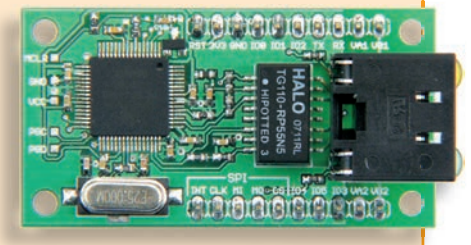


Pico C Meter

Even upmarket digital multimeters boasting a built-in capacitance meter are useless if you want to check out tiny capacitances like 2.7 pF or 5.6 pF. Usually, you're tied to a lowest measuring range of 2000 pF, which is a good laugh to RF designers and radio amateurs. Although at 3.5 digits the DMM's resolution is 1 pF, any measurement below 200 pF or so will produce coarse if not ridiculous results. Pico C does a far better job. Beating many DMMs hands down, this little instrument easily measures capacitances down to fractions of a picofarad.

Kit of parts incl. Elektor Project Case, programmed microcontroller, LCD and PCB

Art.# 100823-71 • $118.40



NetWorker

An Internet connection would be a valuable addition to many projects, but often designers are put off by the complexities involved. The 'NetWorker', which consists of a small printed circuit board, a free software library and a ready-to-use microcontroller-based web server, solves these problems and allows beginners to add Internet connectivity to their projects. More experienced users will benefit from features such as SPI communications, power over Ethernet (PoE) and more.

Module, ready assembled and tested

Art.# 100552-91 • $85.50

Kits & Modules

by Carl Stoez (USA)

The Ultimate Cup Warmer

Built around a PIC16F684 microcontroller and an AD595 thermocouple amplifier, this clever design warms a cup of liquid to a set temperature and automatically cools it down. The design uses a watchdog timer to continually monitor whether the cup has been picked up or a full cup has been placed on the heater plate.

For a coffee cup warmer, this is admittedly over the top. But then again, I usually end up there when it comes to coffee. I always thought the fun part of designing and building your own embedded project was being able to top out a design (see Photo 1). I built this project six years ago. I used it daily at work until we were told to remove all non-UL listed devices from the building.

Recently, I pulled out the design and, low and behold, it still functioned as designed. In fact, I'm using it as I write. All the components except for the power supply are still available, so I thought I'd share the design.

I started out with an old four-cup coffee maker. I disassembled it and removed the heater assembly to use as the heart of the project. I saved the carafe for a pen and pencil holder.

To gain an understanding of how the heater behaved, I breadboarded a simple TRIAC control circuit and programmed the microchip for simple full-power on/off control using a potentiometer as an input to the comparator circuitry of the microcontroller. The results are shown in Figure 1. With the set point at 1.98 V, I found there was a 24.7% overshoot on start up and a 6.2% oscillation above the set point. Once past the initial overshoot, the thermal response from the time the heat shuts off to the time it is back on is almost 50 s. This is not an impressive performance. I was curious to see how a PID control loop would perform.



Photo 1a—My coffee cup warmer greeting me, ready to keep me going. b—The top of the coffee cup warmer was removed to show the components. The load cell is underneath the printed circuit board (PCB) to the right. A separator is used to help keep the heat from the PCB. c—The load cell is shown in the lower middle. It is held in place with a customized paper clip. A bit of rubber on the load cell button gives it a compliant mount and raises it slightly above the rubber pads, one of which is shown in the lower right. The rubber pads help stabilize the project box.

COMPONENTS

Figure 2 is the schematic for this project. The coffee cup warmer features a few

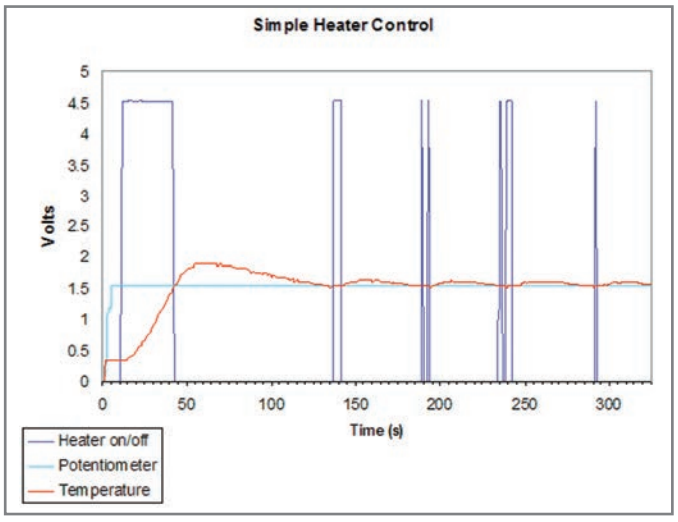


Figure 1—Heater thermal response using on/off temperature control

main components: a Microchip Technology PIC16F684 14-pin, 8-bit microcontroller; an Analog Devices AD595 Type K thermocouple amplifier; a few supporting components, including switches, a display, a TRIAC, and optocouplers; an Analog Microelectronics AM401 voltage amplifier IC; and a Honeywell FSG15N1A force sensor. Yes, you read it right, a force sensor. The load cell weighs the one leg of the coffee cup warmer. I will explain more in a bit.

The PIC16F684 microcontroller was selected because I wanted to try the free HI-TECH Lite C compiler. At the time, the compiler was restricted to specific microchips and would only enable the use of half the memory. I don't believe that is the case now, however. Even with this restriction, I was able to squeeze in everything I wanted.

The PIC16F684 features an internal software-programmable clock from 31 kHz to 8 MHz, low-power sleep, and an enhanced low-current watchdog timer (WDT). Peripheral features include 12 I/O pins, eight of which can be used with the 10-bit A/D converter, two 8-bit timers, and one 16-bit timer. The microcontroller is capable of performing A/D conversions while in Sleep mode when using the internal oscillator. It also includes an enhanced 10-bit pulse-width modulation (PWM) output to one, two, or four channels with programmable dead time and a maximum frequency of 20 kHz. I did not use the PWM. But, why not? I will explain this in a bit, also.

I used the precision internal oscillator software selected at 4 MHz, low-power sleep and the enhanced low-current WDT. Of the 12 I/O pins, I used all 12: three as analog input to the 10-bit A/D converter, and two as a digital input. The LED was driven by six digital outputs. The PWM output to drive the TRIAC through a zero-crossing opticoupler used one additional pin. Of the three timers, I only used Timer0.

The AD595 thermocouple amplifier was required for the PID feedback loop. It made easy work of interfacing with a thermocouple. The AD595 is a low-power (less than 1 mW) IC that provides an output buffer, cold junction temperature compensation, and signal amplification. The output is temperature proportional at 10 mV/°C and takes thermocouple measurements from 0° to 300°C. It is available in two precisions, ±1°C and ±3°C. I used the AD595A with the ±3°C precision. It contains an alarm circuit that senses a broken thermocouple or an open circuit at the device input. I kept knocking the thermocouple leads loose as I worked to install the printed circuit board (PCB), so I implemented this feature to make sure all was well.

The AD595 has a feature that I did not use, the set point controller. It turns out that a microcontroller is not really needed at all if the set point controller feature is used as it can directly drive an appropriate heater controller. How am I to learn about PID if I used this? It can also be used without a thermocouple as a thermometer from –55° to 125°C by taking temperature readings from the compensation voltage.

Prediction of the AD595 output was easily obtained from Table 1 of the specification sheet. For the range of 0° to 300°C I could expect 0 to 3,022 mV. But what I really wanted was an output in degrees Fahrenheit. To obtain this in Excel, I plotted V_{OUT} to Centigrade temperature T_C and did a linear fit to obtain:

$$T_C = 99.025X_1$$

The conversion to Fahrenheit is:

$$X_1 = \frac{9}{5}X_2 + 32$$

However, X_2 is still in volts, so one more conversion is required to work directly from the A/D bits. With 5 V per 1,024 bits, we finally get:

$$X_2 = \frac{5}{1{,}024}Z$$

And when we do all the substitutions, we finally arrive at:

$$T_f = 99.025\ \frac{9}{1{,}024}Z + 32$$

Z is bits output from the A/D converter. I approximated this in the program as:

$$\text{temp} = \left[\left(T_{IN} \times \frac{87}{100}\right) + 32\right]$$

Obviously, I cheated. To be more precise I should have followed up with at least a two-point calibration using ice water and boiling water. The same method of conversion would be used. My PCB includes additional resistors and a trim pot that could be used for trimming the calibration error. In the final design I chose not to populate the board with these components. I could have selected to change the offset and gain of the AD595 to more closely match the 0- to 5-V range of the A/D converter. This sort of accuracy

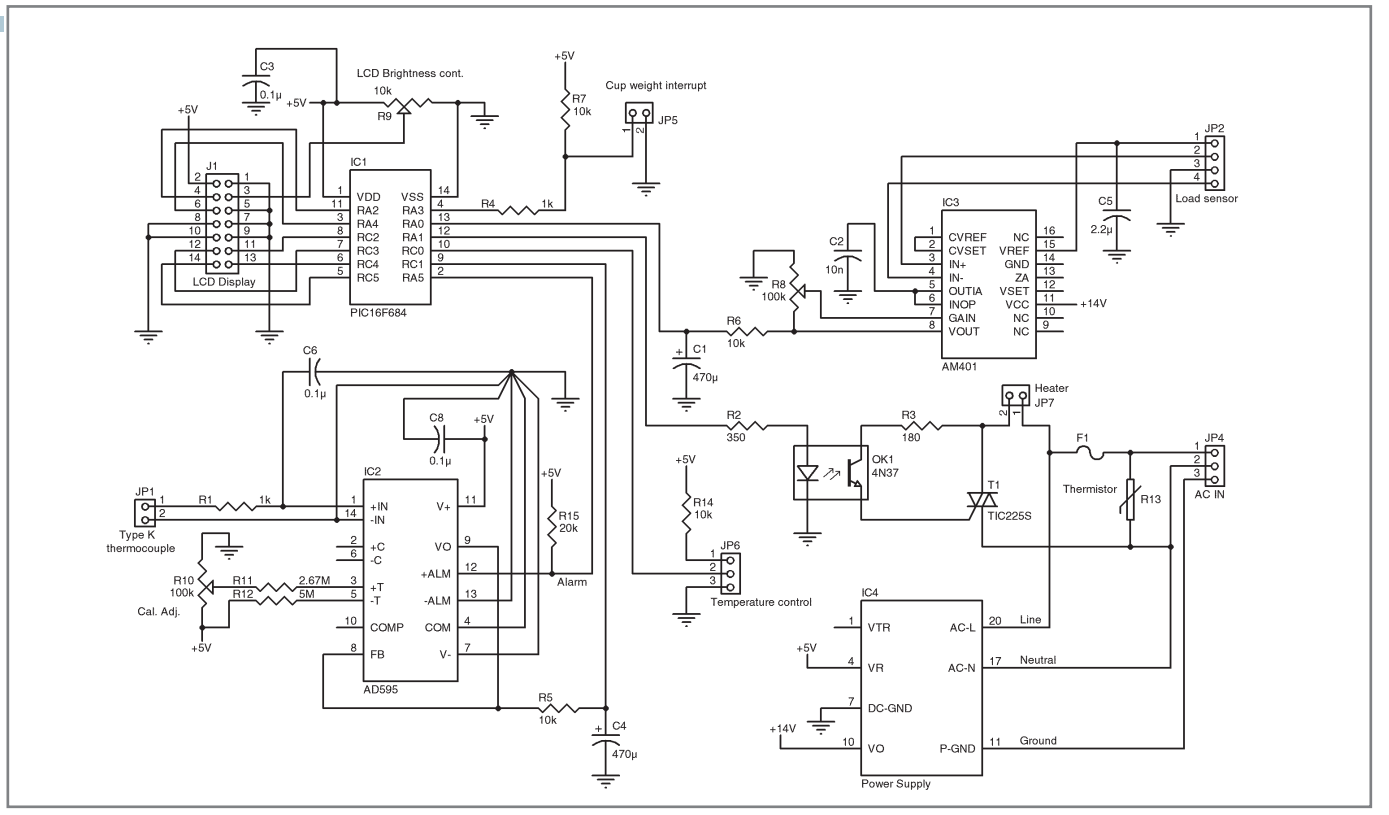


Figure 2—Schematic of the coffee cup warmer

was not required for this project. My tongue is not that finely calibrated.

I used a 10-kΩ potentiometer for the temperature set point adjustment. I can adjust the heater from 32°F to a blistering 490°F. However, the maximum temperature of the heater appears to be 450°F. Looking back on this project from six years ago, there is no telling why I didn't limit the set point temperature to something reasonable. Considering all the plastic in the packaging I certainly recommend it.

One of the goals of this project was to see just how inexpensive it could be made. Up until now, I really had gone cheap. I was using a trial version compiler, Eagle freeware schematic software by CadSoft, and a low-cost Microchip Technology PICkit 1 flash starter kit. I used leftover potentiometers and supporting hardware from other projects. The AD595 and AM401 were samples. The Microchip controller cost all of $3. The Honeywell FSG15N1A load cell is what really put this project over the top at $45. And, since I was there, I went ahead and had some PCBs made. Oh well. It was a good goal, right?

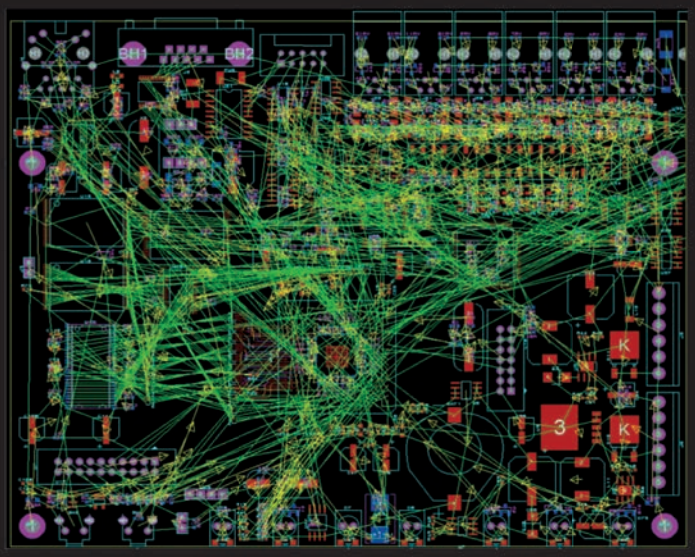
The load cell is a precise

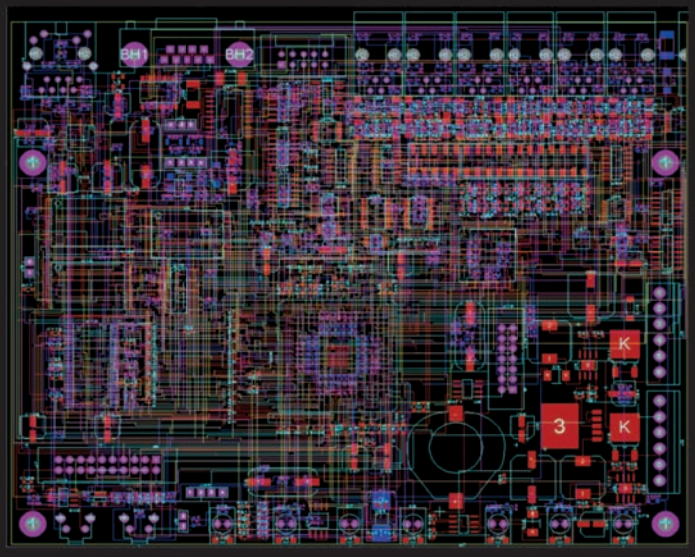
Listing 1—The PID subroutine. Note that error is passed in as "knob-heater" when this is called from State 3. Also, iTerm is now the result of a division, not multiplication, to prevent using a float. Gain coefficients are defined in the header `CupWarmer.h`.

```
// Constants - defined in header
const pGain = 10;                    //  pGain - Proportional Gain
const iGain = 512;                   //  iGain - Integral Gain
const dGain = 2;                     //  dGain - Differential Gain
const iMax = 4096;                   //  iMax - Integral Max
const iMin = -4096;                  //  iMin - Integral Min

int PID(int error, unsigned int T)
{  int  pTerm;          // Everything is in terms of bits per volt
   int  iTerm;
   int  dTerm;
   static int iState;
   static unsigned int TState;
   pTerm =error/ pGain;              // Proportional term
   iState = iState  + error;         // read in last iState value and update
   if(iState > iMax)                 // check for max
      iState = iMax;
   else if(iState < iMin)            //check for min
      iState = iMin;
   iTerm = iState/iGain ;            // Integral term. NOTE divisor to
                                     // avoid using float value
   dTerm = dGain *  (T - TState);    // Differential term, present - last
   TState = T;                       // Temperature update for next iteration
   return (pTerm + iTerm - dTerm);   // return scaled control value
}
```

Our new autorouter will turn this... into this...





in less than 5 minutes!

All levels of the **Proteus Design Suite** now include a world class fully integrated shape based autorouter at no additional cost.

The **Proteus Design Suite** also incorporates:

- Professional schematic capture
- Publication quality schematics
- Fully configurable bill of materials
- Large component libraries for both simulation and PCB layout
- Mixed mode SPICE circuit simulation
- Co-simulation of PIC, AVR, 8051 and ARM7 microcontroller firmware
- Automatic component placement and gateswap optimization
- Highly configurable design rules
- Interactive design rule checking
- Polygonal and split power planes
- RS274X, Excellon and ODB++ database export
- 3D Viewer with 3DS, IDF and DXF export



Prices start from just $249* - visit our website for full details or to download a free demo.

Labcenter Electronics www.labcenter.com

20 Years

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

*exc. tax & delivery

compact unit that utilizes a piezoresistive micromachined wheatstone bridge for the sensing element. The wheatstone bridge output is uncompensated and unamplified. The load cell ratiometric output is specified at 0.024 mV/gram/V.

The PIC16F688 does not have an internal voltage reference comparator. Two options to increase A/D resolution on the analog signal from the load cell are to provide a low, stable, and noise-free reference voltage to the AD converter or to use an external amplifier. I used an Analog Microelectronics AM401 industrial voltage amplifier IC.

The AM401 can power the load cell and provide signal amplification. It has one high-precision instrument amp and two op-amps that are user accessible. The instrument amp is for differential input signals from the transducer. The output of the IA can be feed to an op-amp for reference to ground and final amplification. The other op-amp can be used as a current or voltage source. Included in the IC is a reference voltage source that can be adjusted between 5 and 10 V to power the transducer. It is a versatile chip but it takes some setting up.

I set it up as shown in the datasheet, which is as an industrial bridge amplifier. The AM401 pin V_{REF} can directly provide 5 V to the load cell. To do this, the power supply must be $V_{CC} \geq V_{REF}$ +5 V. Leave pin V_{SET} unconnected to set V_{REF} at 5 V. Connect V_{SET} to GND for 10-V supply out of V_{REF}. I selected a dual-output power supply and provided 14 V to the AM401. By the above V_{CC} constraint I had to limit my load cell supply to 5 V.

To calculate V_{IN} of the AM401 we must start with the load cell output span. Since the load cell is ratiometric, we use the selected supply voltage of 5 V. At 0.024 mV/g/V then with a 5-V supply we calculate sensitivity of the load cell as 0.12 mV/g (i.e., 5 V × 0.024 mV/g/V). At a maximum load of 1,500 g we can expect the load cell output to be 180 mV full span (i.e., 1,500 g × 0.12 mV/g). At a maximum overload of 4,500 g we can expect an output of 0 to 540 mV. I decided to go with the normal full scale of 1,500 g.

With the sensitivity calculated we now finish setting up the AM401 for a 0- to 5-V full span output to the microcontroller for a load cell span of 0 to 1,500 g. Referring back to the datasheet, we find that for the industrial bridge amplifier usage we need to calculate two resistor values, R1 and R2, that form a voltage divider as feedback for OP1 to set gain. To calculate the resistance values they specify:

$$\frac{R_1}{R_2} = \frac{V_{OUT}}{G_{IA} V_{IN}}$$

I followed their example and selected an instrument amp

Listing 2—The main loop state machine. There is no State 6. That fell out somewhere along the journey.

```
for(;;)                 // Main Loop
  {   CLRWDT();         // kick the dog
     if(State==1)
     {   control();
     }
     if(State == 2)
     {   State++;
         read_cup();  // Read cup weight
     }
     if(State == 3)  // Take settings
     {   State++;
         heater = heater/2;
         heater = heater + adc_read(Theater)/2;  // Read A/D, ANS5=RC1
         knob = knob/ 2; // divide by 2 as an averageing filter
         knob = knob + adc_read(Tcontrol)/2;
                              // Read knob T setting ANS4 = RC0
         PIDout = PID(knob-heater, heater);   // knob - heater = error
     }
     if(State == 4)           // Display
     {   State++;             // incrementstate machine
         lcd_clear();
         lcd_puts("S: ");
         convert(knob);       // Convert and display
         lcd_goto(0x40);      // move cursor to next line on display
         lcd_puts("T: ");
         convert(heater);     // Conversion to F
         //lcd_clear();
         //itoa10(String,PIDout);
         //lcd_puts(String);
     }
     if(State == 5)  // Set PWM bounds
     {   State = 1;
         if(PIDout<0)
                 PWM = 0;
         else if(PIDout > 60)
                 PWM = 60;
         else PWM = PIDout;   // Set PWM width
     }
     if(State == 7)           // Alarm.  The thermocouple wire is
                              // disconnected
     {   lcd_clear();
         lcd_puts("Sensor");  // or broken.
         lcd_goto(0x40);      // Move cursor to next line on display
         lcd_puts("Error!");
         PORTA = 0;
         State = 0;           // Disable state
         GIE=0;               // Disable interrupts
     }
     if(State == 8)           // Save Cup weight in eeprom
     {   calibrate_cup();
         PORTA = 0;
         State =1;
     }
  }  // End for
} // End Main
```

gain, G_{IA} = 5. I decided to use a 100-kΩ trim pot for resistors R1 and R2 with the relationship:

$$R_1 + R_2 = 100\ k\Omega$$

We end up with two equations and two unknowns. Using the above information of V_{IN} = 180 mV and G_{IA} = 5, we find that R1 = 82 kΩ and R2 = 18 kΩ. The final signal gain is 27.78. I used an ohmmeter to set R1 on the trim pot before soldering it onto the PCB.

PID BASICS

The PID has been around for a long time and plenty of books and papers have been written about it so I won't go into much detail. The PID consists of three components, the proportional, the integral, and the differential. The components can be used independently or in combination, as required. Each component helps the controller drive to and hold to the desired set point. They are used in numerous applications ranging from motor, position, and thermal controls. Each application has its own requirements and is tuned differently.

My PID loop is shown in Listing 1. It is almost a direct copy-and-paste from Tim Wescott's "PID without a PhD." I didn't use a structure but I did keep the variable names the same. The only change I made was to change the integral and proportional terms from multiplication to division to avoid using a float.

Proportion control is often used as the only control component and is the easiest to implement. In a thermal system the proportional control settles reasonably well but undershoots the control set point. As the proportional component is increased in value the system moves closer to the set point but begins to overshoot and starts ringing.

Proportion is simply the error divided by pGain with the error being the set point minus the reading. In my code, I calculate the error as knob minus heater and pass this value to function PID (int error, unsigned int T). Obviously, to increase the proportional term, pTerm, I decrease pGain.

Integral control adds long-term precision to the process. The integral term iTerm is the sum of all errors. I think of it as a bucket. The iGain determines how much the contents of the bucket influence the system. iTerm can overflow or underflow so limits are placed on it within the code. Intergral control often decreases stability. It should be used with a light touch. In a thermal system integral control can eventually settle the system out and it does so at the correct set point. When combined with a proportional control PI systems are commonly used in thermal systems.

Differential control is the difference in input readings. In my case this is the previous temperature reading – current temperature reading. It gives you an indication as to how fast the system is changing. In a thermal system using PID control the differential term adds stability. It is, however, the most difficult term to implement. It is subjective to signal noise, sampling variances, and high-frequency oscillations.



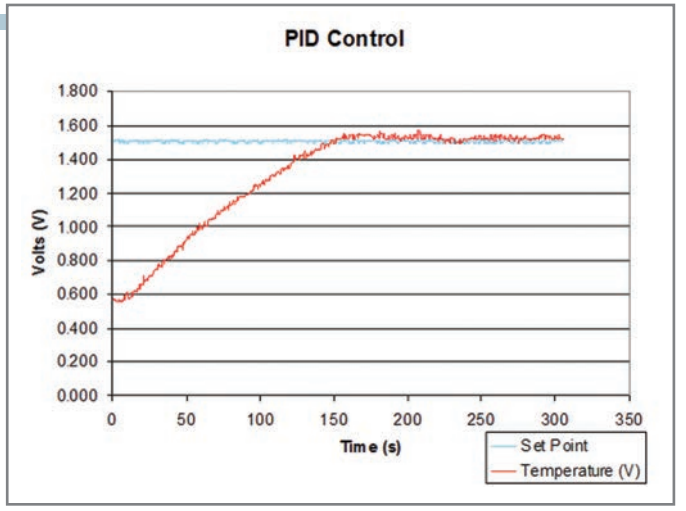


Figure 3—Heater performance after implementation of a tuned PID feedback loop

After implementation the final step is tuning. Wescott recommends setting all gains to zero and starting with the differential gain. Set the proportional gain to obtain a small value and start with an initial differential gain value about 100 times that of the proportional gain. Increase the differential gain until it is on the verge of oscillation and back it off until it is once again stable.

Next, start with the proportional gain. Start with a gain between one and 100. If it oscillates, drop the proportional gain by factors of eight to 10. If there are no oscillations and the system is still slow to respond, increase the proportional gain by factors of eight to 10 until it just oscillates or you see excessive overshoot, then back off.

Finally, you get to play with the integral gain. Start with small factors from 0.0001 to 0.01. Again, look for minimal overshoot and oscillation.

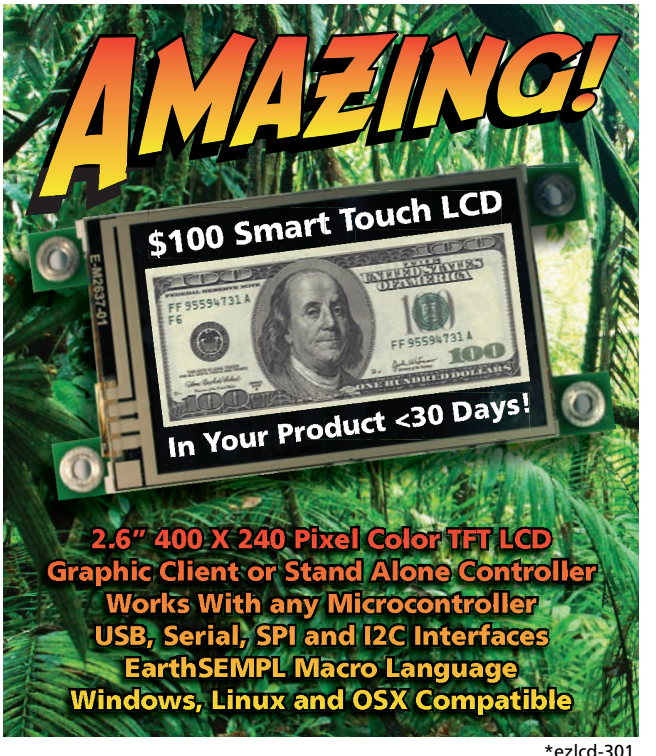
Implementing the PID was pretty simple. Tuning it, on the other hand, was not. It was at this point that I understood why there are so many articles on PID tuning. I learned an important point: Avoid hard code PID gain parameters. Next time I do this, I will use potentiometers to tweak each term. If nothing else, I will use in-circuit serial programming instead of a stand-alone programmer.

PROGRAM OPERATION

When the warmer is plugged in and turned on a welcoming "Hello Carl" display greets me. The display turns off and the system sleeps until I place a cup of coffee on the heater. Sensing a change from the load cell, the warmer wakes up, checks the weight of the cup and the desired temperature set point, and begins to warm up. The set point is controlled with a 10-kΩ potentiometer. The two-line eight-character LCD shows the set point temperature on the upper line and the current temperature on the lower line.

An empty new cup can be tared at any time by placing the cup on the heater and pressing the Momentary Contact button. The user sees the message "Weighing Cup" on the LCD.





*ezlcd-301

EARTHLCD.COM
We Make LCDs Work™
For more information visit http://store.earthlcd.com/cc1011

The tare value is stored in EEPROM. When the cup is empty, the processor goes into Sleep mode and the heater plate is able to cool off.

The load cell is also used to monitor usage. If I have not picked up the cup in 30 minutes, it assumes I am no longer drinking the coffee, puts the processor to sleep, and enables the heater to cool off. I used the WDT to wake up every 0.5 s to check if the cup has been picked up or a full cup has been placed on the heater.

So, here we find the reason I didn't bother using the built-in PWM registers. The maximum period I could set with the oscillator setting of 4 MHz was a little more than 1 s. I could set the minimum PWM frequency to around 600 Hz. The heater has a slow temperature response, almost 50 s. Being a thermal system with no active cooling, I really did not have much control over the settling time. So, I could not see the benefit of having the PWM driving frequency in the near kHz frequency when the thermal response is so slow. Also, manually programming the signal helped me close the proportional-integral-controller (PID) loop for myself.

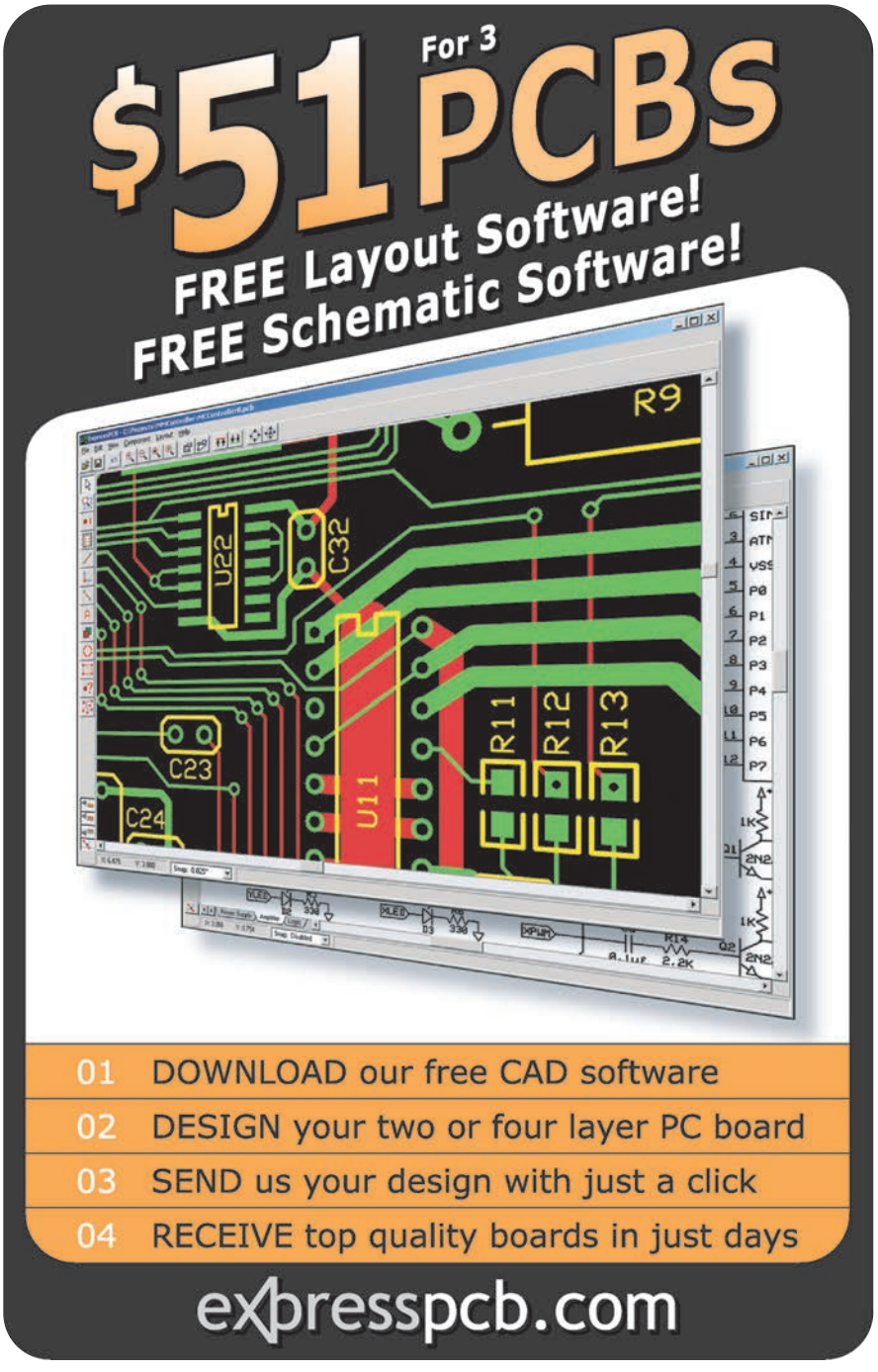
The program is timed with interrupts on Timer0 set at 1/60 s. Set in the interrupt routine are two variables used for long-term timing, Timer3 and Timer4. Timer3 keeps tracks of each 1/60 s. Timer4 keeps tracks of full seconds. The interrupt timing is also the PWM resolution of 1/60 s. The PWM period is 1 s. The data acquisition rate is once a second.

When the program starts, as usual, various I/Os and registers are set and the welcoming message "Hello Carl" is displayed. The program flow is controlled by the variable "State" which is initially set to 0. The program then enters the state machine (see Listing 2). With State set to 0, nothing happens until Timer0 overflows 60 times (or once a second) and sets State to 2 where it reads the weight of the coffee cup and increments State to 3. State 3 is where the temperature of the heater and the heater settings are obtained and the PID controller is calculated. State is incremented to 4 where the readings obtained in State 3 are displayed on the LCD. State is incremented to 5 where the width of the PWM is bounded and set based on the PID output. State is set to 1 where the control signal is sent to the heater, State is set back to 0 and we wait for a minute or until some other interrupt happens.

Two other interrupts are monitored: a button press to tare the cup and a thermocouple error. Both of these events set the appropriate State condition.

PERFORMANCE

We saw earlier how the heater did not respond very well to a simple on/off control. After several evenings spent tweaking and reprogramming, I finally arrived at the graph shown in Figure 3. Here the 24.7% overshoot has been reduced to 2.2%. There is little oscillation around the set point temperature with an error of 0.87%.



With more work I probably could have reduced this more, but I felt that I had achieved my goal of implementing a successful PID controller.

In addition to using pots to set the PID gains, I would also connect up an output to a com port or USB to monitor the PID variables and PWM command variable. That would really help to see how the PID loop is responding, it would also be a tremendous aid in tuning.

After I assembled the entire project, I was finally able to take some true load cell readings. The load cell no cup voltage was 1.1 to 1.2 V. With a randomly selected ceramic cup, I had an empty cup reading of 1.44 to 1.52 V and a full cup reading of 1.68 to 1.87 V. The variance of the readings comes from the location of the cup sitting on the heating element. I was pleased with this result. I had to guess what range of loads the load cell would see. I hadn't built the project box and really didn't know how the load cell would respond when I actually placed a cup on the heater.

I didn't anticipate one thing. When the cup begins to empty, there is some dithering because of the uncertainty of the cup weight. This works out okay because an almost empty cup does not need nearly as much power input to keep it warm.

BREAKING NEW GROUNDS

The end result is shown in Photo 1. Putting together project boxes can be tricky. This one turned out quite well. I was impressed with the PID performance with it improving the precision by factor of 11.2 on overshoot and 7.1 for holding the set point. I found that a set point temperature of 300°F is just right to produce a pleasantly warm and drinkable cup of coffee. I can walk away and leave it, confident that I won't return to a burnt coffee aroma pervading my office. I can also change cups at will as long as I limit my selection to something that can take the heat. Maybe my next design will use an infrared temperature sensor to determine the actual temperature of the coffee. Or maybe it will even monitor my daily intake of caffeine. On second thought, I may not want to know.

Carl Stoez is a mechanical engineer and avid coffee drinker who likes to play in the electrical sandbox from time to time. He is a research and development engineer in the optics group of Baker Hughes, Inc. You can contact Carl at carl.stoesz@bakerhughes.com.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/255.

RESOURCES

Analog Devices, Inc., "Type K Thermocouple Amplifier with Cold Junction Compensation," AD595, 1999, www.analog.com/en/mems-sensors/analog-temperature-sensors/ad595/products/product.html.

Analog Microelectronics, "Voltage Amplifier-IC with Differential Input," AM401, 2006, www.analogmicro.info/english/products/am401.htm.

Microchip Technology, Inc., "PIC16F688 Datasheet," 2004, ww1.microchip.com/downloads/en/DeviceDoc/41203B.pdf.

T. Wescott, "PID Without a PhD," *EE Times*, 2000.

SOURCES

AD595 Type K thermocouple amplifier
Analog Devices, Inc. | www.analog.com

AM401 Voltage amplifier-IC
Analog Microelectronics | www.analogmicro.de/english

FSG15N1A Force sensor
Honeywell International, Inc. | www.honeywell.com

PIC16F688 Microcontroller and PICkit 1 flash starter kit
Microchip Technology, Inc. | www.microchip.com

NEED-TO-KNOW INFO

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

—

PID Control Without Math
by Robert Lacoste
***Circuit Cellar* 221, 2008**

If you need to design an optimized control loop for a hardware control problem, consider trying a proportional integral derivative (PID) controller. In this article, Robert explains that PID regulations are simple to code, wire, and tune to find the results you're looking for. Topics: PID, Math, Temperature, Soldering Iron, Proportional, Integral, Derivative

—

Temperature Calibration System
by Brian Millier
***Circuit Cellar* 202, 2007**

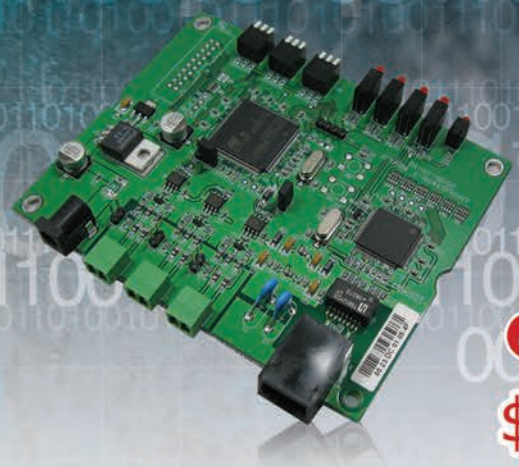
Brian designed a portable temperature meter that contains a PID controller and a user interface for entering a setpoint. The meter can be plugged into a separate calibrator unit, which generates stable temperatures for sensor calibration purposes. Topics: Temperature, PID, Calibrator, RTD, MCP3551, ADT70, ATmega88, Peltier, Relay, Thermistor, Thermocouple, Steinhart-Hart

—

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

ICbank
Gateway to Electronics

3 PORT INTERFACE
RS-485 to Ethernet Converter



Only
$170



RS-485 to Ethernet Converter

Powerful feature

- Protocol converter RS485 between Ethernet
- Offer TCP/IP Communication to Devices with RS485 I/F

Specification

Network	: TCP, UDP, DHCP, ICMP, IPv4, ARP, IGMP, PPPoE, Ethernet, Auto MDI/MDIX , 10/100 Base-TX Auto negotiation (Full/half Duplex)	
Serial	: RS485 3 Ports, 1,200~115,200 bps, Terminal block I/F Type	
Control program	: IP Address & port setting, serial condition configuration, Data transmit Monitoring	
Accessory	: Power adapter 9V 1500mA, LAN cable	
Etc	: - DIP Switch(485 Baud Rate setting)	- LED: Power, Network, 485 Port transmission signal

by David Ludington (USA)

High-Accuracy Voltage Reference Using PWM (Part 1)

Pulse-Width Modulation Theory

You can use pulse-width modulation (PWM) to design a programmable voltage source with high full-scale accuracy. This article details the analog design tradeoffs necessary to achieve the desired accuracy when using this technique.

Pulse-width modulation (PWM) is a technique that is widely used to control components, such as motors, and even as power amplifiers in audio applications. PWM can be thought of as a digital-to-analog converter (DAC) whose analog value is proportional to pulse width and height. It is often used in situations where digital switching provides a highly efficient transfer of power to drive the controlled component.

However, I recently realized that the PWM technique also has the potential to provide precise and accurate analog voltages. This is because the PWM signal is a digital pulse train whose duty cycle is independent of the clock frequency and frequency drift as long as that drift is relatively slow and the clock skew on the positive and negative edges is low.

Two aspects of a PWM waveform that are important for a precise programmable voltage reference are absolute accuracy and resolution. Absolute accuracy is derived from a master reference voltage and transferred to the programmable output voltage by selecting the appropriate pulse-width duty cycle. Resolution (or step size) is determined by the number of minimum pulse-width steps in the total PWM period.

This two-part article series will describe the design approach and tradeoffs used to implement a 0- to 5-V programmable voltage source with full-scale accuracy better than 0.01%. In part one, I'll cover the basics of PWM pertinent to this design along with the analog design tradeoffs to achieve the desired accuracy made possible by the PWM technique. In part two, I'll present a hardware design to evaluate practical PWM circuits and the test setup to evaluate this hardware along with performance test results that show the voltage accuracy achieved.

PWM BASICS

Figure 1 shows a PWM waveform with a minimum pulse width of 1 µs, a PWM period of 5 µs, and a pulse magnitude of 5 V. In this example, the average DC value of the waveform is 1 V (the pulse width divided by the PWM period multiplied by the pulse magnitude). If the pulse width is 2 µs, the average value is 2 V, and so on.

In many control applications, the clock rate and timing are precise but the magnitude of the voltage levels are not held to any great precision. In these closed-loop applications, the control loop adjusts the pulse width to compensate for variations in voltage levels. However, if the pulse magnitude is derived from a precise reference voltage, the average value is also precise. Thus, in Figure 1, the average values of the various pulse widths (1 V, 2 V, etc.) will have the same precision as the PWM reference voltage.

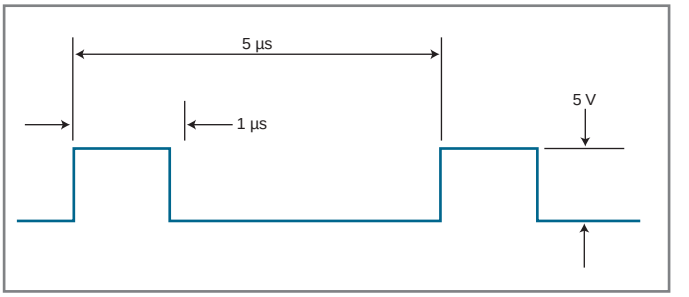


Figure 1—PWM waveform

What about resolution? In this example, the PWM resolution of five steps is rather coarse and equivalent to about 2 bits with a quantization level of 1 V. Although the resolution is low, the output voltage level is accurate if the reference is accurate. If the minimum pulse width is 1 µs and the period is extended to 1 ms, then there are 1,000 resolution steps (equivalent to 10 bits) and the quantization level is equal to 5 mV. In this case, there is both good accuracy and good resolution.

ANALOG DESIGN TRADEOFFS

The discussion so far has been based on an ideal PWM waveform where accuracy and resolution can be chosen with arbitrary precision. However, in the "real world," there are a number of circuit imperfections that must be considered, such as amplifier voltage offset and input bias current and capacitor leakage current, any of which can degrade the final voltage output. In addition, the clock rate may have to be limited to minimize pulse-width errors due to differences in rise and fall times and delays. If these imperfections cannot be satisfactorily controlled, the output voltage won't be as precise as the ideal PWM waveform would suggest.

The goal of my project design is to maintain as much as possible the accuracy inherent in the ideal PWM waveform. The design analysis will be performed using the PWM generation and low-pass filter circuits shown in Figure 2. The output of the PWM generator is a digital signal at typical logic levels (3 to 5 V) with loose voltage tolerance on the logic level. This signal is the input to the PWM driver circuit where the voltage level is more accurately controlled. The output of the PWM driver has the precise timing of the PWM generator and switches between 0 V and a precise reference voltage of 5 V.

The filtering circuits shown in Figure 2 provide averaging of the PWM waveform to give the programmable output voltage. Two filtering circuits are shown (a single-stage filter and a two-stage filter). When there are two or more stages, the stages can be concatenated as shown or buffer amplifiers can be used between the stages. Alternatively, two stages can be designed with one buffer amplifier with capacitors in a feedback arrangement. The single-stage filter circuit will be used to describe the basic filtering action.

Resistor R1 and capacitor C1 smooth the output of the PWM driver and give the average value of the PWM waveform at the input to a unity gain voltage buffer A1. The averaging occurs because the filter time constant ($\tau = R1C1$) is much longer than the duration of the PWM period. The buffer A1 is needed because the juncture of R1 and C1 is a high impedance point that is easily loaded by subsequent circuits thereby lowering the average voltage from its correct level.

The circuits shown in Figure 2 are simple; and for many applications, not much tradeoff analysis is needed. However, in this application, component parameters must be carefully chosen to ensure that the magnitude of the final output voltage is not degraded. In theory, a single stage could provide enough filtering by using large value resistors and capacitors. However, in practice, multiple stages are usually needed because the large component values of a single stage can adversely impact the precision of the output voltage.

The following sections present the component tradeoffs. Many component parameters are interrelated so minimizing one source of error may make another error larger. Thus, the final design is a compromise of component selection with the goal of minimizing the total error of the output voltage.

PWM GENERATOR & DRIVER

The timing parameters of the PWM generator and driver (rise time, fall time, and delays) are not large in modern digital components, but they can still have an effect on the analog output voltage by changing the pulse width from its ideal value. The difference in rise and fall values is important, not their actual value. For example, in Figure 1, with the clock rate at 1 MHz, the minimum pulse is 1 µs. If the delay in both pulse edges is 100 ns, the minimum pulse width is still 1 µs but delayed in time. However, if the delays are different by 10 ns, then the pulse width will be either 1.01 µs or 0.99 µs, depending on which delay comes first. This may not sound like much, but if we are trying for an accuracy of 0.01%, this error can be significant. If the clock rate is decreased, the percentage error for a fixed difference in delays will be less. If the clock rate is increased, the percentage error will be more. More steps in the PWM waveform for the same clock rate will result in a smaller output voltage for the minimum pulse width and thus less full-scale error.

A potential solution to delay differences is a lower clock rate and/or many steps (good resolution). It certainly is no problem generating a lower-frequency PWM waveform and good resolution is desirable. However, problems can arise when turning the PWM waveform into a DC value. To reduce the frequency harmonics of the PWM waveform to well below the PWM step size (for many steps the step size is small) can require the filter to attenuate the PWM

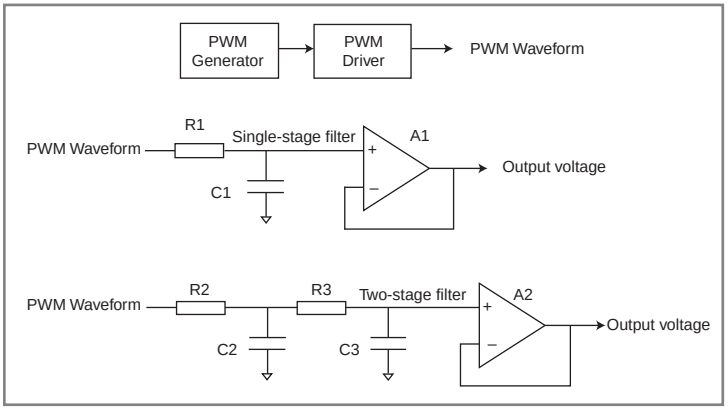


Figure 2—PWM generation and filtering circuits

clock greater than 100 db. A single-stage RC filter has an attenuation of 20 db per decade for frequencies greater than its corner frequency. So, either we need many filter stages or each stage needs to have a very low corner frequency (large-value filter components).

For example, with a 100-kHz clock and 50,000 steps, the PWM period would be 0.5 s and the PWM rate would be 2 Hz. The corner frequency of a single-stage filter would need to be five decades (100,000 times) below the 2-Hz PWM frequency or five filter stages would be needed to provide 100-db attenuation. In the single stage, the resister and capacitor values would be very large and would significantly degrade the output voltage. In the multistage filter, errors per stage are smaller but the corner frequency would still be fairly low, and with more circuit components, errors would accumulate.

Finally, there are potential problems with the resistance and leakage current in the PWM driver output stage. The driver output contains MOSFET transistors that switch the output between 0 and 5 V. There isn't a problem if the lower transistor between 0 V and the output is the same as the upper transistor between 5 V and the output. But, if the transistors are different, the rise and fall time constants for the RC filter are slightly different causing a shift in the average output voltage. Also, if the output transistors have leakage currents flowing through these resistances, there are small voltage drops and the driver output does not quite reach 0 or 5 V, which can result in an error.

RC LOW-PASS FILTER

Resistor R1 and capacitor C1 are related to the filter corner frequency:

$$f_C = \frac{1}{(2 \pi R1C1)}$$

As mentioned, the attenuation of a single RC filter is 20 db per decade. Thus, if the filter corner frequency is a factor of 10 below the PWM rate then the attenuation will be a factor of 10 (20 db). If the values of the resistor and capacitor are large, the filter corner frequency is low providing a large attenuation for the PWM frequencies. However, large values of R1 and C1 can adversely affect the average DC voltage due to their own limitations and their interaction with the following buffer amplifier.

The resistor value should not be too low or it will put too much load on the PWM driver and cause a voltage drop across the driver output. In addition, a low resistor value will require the filter capacitor value to be very large for a low-corner frequency. The resistor value must also not be too high or the input bias current of the buffer amplifier will cause a voltage drop across the resistor and lower the reference voltage output. For this application, the resistor value will probably end up between 100 kΩ and 1 MΩ. In addition, the resistor type should be metal film to minimize the resistor noise.

A small capacitor value is not a problem other than the resistor value (from the preceding equation) becomes large to maintain a low-filter corner frequency. However, if it is too large, leakage current may become an issue. All capacitors have leakage current that can be modeled as a large-value resistor in parallel with the capacitor. This resistance is inversely proportional to the capacitor surface area so that a larger-value capacitor has more leakage current. The leakage current is also dependent on the type of dielectric in the capacitor. Electrolytic and tantalum capacitors have large capacitance but the leakage is too high for this application and they will not be further considered. Ceramic capacitors have less leakage and film capacitors have even less leakage. However, high-value film capacitors can get expensive and physically large.

In addition to leakage, capacitors have dielectric absorption (DA) that can be modeled as small capacitors in series with large resistors all in parallel with the capacitor. Larger values of DA will greatly extend the settling time when the programmable voltage is changed, particularly when settling is to less than 0.01% FS. Film capacitors have the lowest DA.

The values of R1 and C1 don't need to be precise. As long as there is adequate attenuation, the exact value of the corner frequency is not important.

BUFFER AMPLIFIER

The two amplifier parameters that are important for this application are offset voltage and input bias current. Offset voltage is important because it directly adds to or subtracts from the voltage at the amplifier input. Amplifier input bias current is important because it can cause a voltage drop across the filter resistors, which adds to or subtracts from the output voltage depending on the polarity of the bias current. With multiple filter stages, the total series resistance is used to calculate the voltage drop. For example, a bias current of 100 pA through a 1-MΩ resistor causes a voltage drop of 100 µV. Lower bias current amplifiers are available, but they also have the highest voltage offsets. A compromise is needed.

WRAPPING UP

There is potential to design a programmable voltage output using a PWM approach, which is as accurate as the master reference. I also described possible analog filter circuit configurations that can turn the digital waveform into a precise analog voltage. And finally, I performed a tradeoff analysis to describe the sources of error when using these circuits and to determine the magnitude of that error when implementing these designs with off-the-shelf components.

In the next part of this article series, the hardware to test this PWM approach will be designed and output voltage accuracy will be evaluated across the voltage range.

David Ludington (davidludington@hotmail.com) is a retired electrical engineer with experience in low-noise analog design and infrared system design. He earned a BSEE at Michigan State University and an MSEE at Syracuse University.

9th International System-on-Chip (SoC) Conference, Exhibit & Workshops

November 2 & 3, 2011 – Radisson Hotel Newport Beach

www.SoCconference.com

Special 2011 Pricing to Support the Tech Community!

Don't Miss Out!

The Most Informative, State-of-the-Art, and Affordably Priced Chip Design Conference of the Year!

Platinum Sponsors

- Table-Top Exhibit (Free Passes)
- 3-D ICs
- Analog & Mixed-Signal Designs
- Green Chips
- SoC Design & Verification
- MEMS-Based SoCs
- 20nm Design Challenges & EDA
- Customizable SoCs - FPGAs
- IEEE Student Design Contest
- Multicore Software Development
- Many Door Prizes

Microsemi

ALDEC

UCIrvine | Extension

- SOI vs. CMOS
- 4G SoC Innovations
- Multicore SoC Platforms
- CPU/DSP Cores
- Network-on-Chips (NoCs)
- Informative Panels
- Emerging IP Blocks
- IEEE OC Job Fair
- Technical Workshops
- Networking Opportunities
- And Much More. . .

Keynote Speakers



Yale University
Dr. Eugenio Culurciello.



Microsemi
Paul Pickle, Senior Vice President, Integrated Circuit Group.



Johns Hopkins University
Dr. Ralph Etienne-Cummings.

Selected Participating Companies, Universities & Organizations

AMD, FAB, Toshiba, ALDEC, Microsemi, methodics, Cavium Networks, Western Digital, USC University of Southern California, Johns Hopkins University Whiting School of Engineering, UCI University of California, Irvine, UCLA, Berkeley, Yale University, Keio University, UCSD, Target, SOI Industry Consortium, cadence, ARM, Mindspeed, npi, NEC, CRC Press Taylor & Francis Group, K.I.T 金沢工業大学, Ericsson, srs, Movidius, eSilicon, Southern California Edison, MEMS Industry Group, Soitec, elektor, Circuit Cellar, Chip Design, octasic semiconductor, Si, Calit2

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

QUESTIONS & ANSWERS

Testing Devices & Writing in Assembly



An Interview with Larry Cicchinelli

Larry Cicchinelli, author of Assembly Language Essentials, *is a ham radio enthusiast, a* Circuit Cellar *contributing author, and a longtime electronic device designer. In August 2011, Larry and I discussed his background in electronics, the enjoyment he gets from designing and building electronic devices, his affinity for Assembly language, and his future projects.—Nan Price, Associate Editor*

NAN: Tell us about the first microcontroller-based project you designed.

LARRY: My first microcontroller project used an 8080 when I worked for Ford Motor Company. I was on a team that had to design a small, low-cost test set for an automotive electronic product.

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

LARRY: I worked in the test equipment department at Philco-Ford for about 30 years. Our charter was to design and install test equipment for Ford's automotive electronic components. Initially, I was designing semi-automatic test fixtures for early logic devices when Philco was part of Ford. The first testers were for early logic integrated circuits (ICs)—Resistor-Transistor Logic (RTL), Diode-Transistor Logic (DTL), and High-Threshold Logic (HTL). During that time, when transistor-transistor logic (TTL) was being introduced, I also did a little bit with early MOS devices—not CMOS, I believe they were PMOS, but it has been too long.

After testing logic devices, I transferred to a different area of testing—TVs and radios. I was on a team that developed one of the first computer-controlled test sets for TVs. This particular TV was modular—it had plug-in solid-state circuit boards that essentially performed the same functions as individual tubes or sets of tubes. The tester also aligned those circuits that needed alignment, color circuit and IF.

After Ford sold Philco, my department still built test equipment for Ford radios as well as the "new" automotive electronic devices—engine controllers, solid-state ignitions, electronic-voltage regulators, speed controllers, etc. I was part of a small team that designed and built the computer-controlled test set for Ford's first electronic engine controller (EEC). It was during this time that I worked on four of the most interesting projects of my career.

It was during the development of the test set for the EEC that I developed a programming language based on a macro assembler. The source statements were "interpreted" by the assembler based on a large set of definitions I designed in much the same way modern compilers handle `#define` directives in C. This language was in use at the facility where I worked for more than 20 years. I used this language as the basis of my Master's thesis.

Along with this programming language, I also conceived and developed a modular test system based on the LSI-11 from Digital Equipment Corporation. This test equipment used my macro-based language and was used as the basis for the testers for many product lines in the manufacturing plant where I worked.

While working on my Master's degree, I wrote a short paper describing a real-time data collection and reporting system. My boss read it and showed it to our department manager who then asked if I could build a working model. I did and eventually a project was approved for me to develop a system for our plant. I designed all the hardware and software for the system. The data collection hardware was based on the LSI-11 and a multiplexor I designed that enabled up to 64 serial ports.



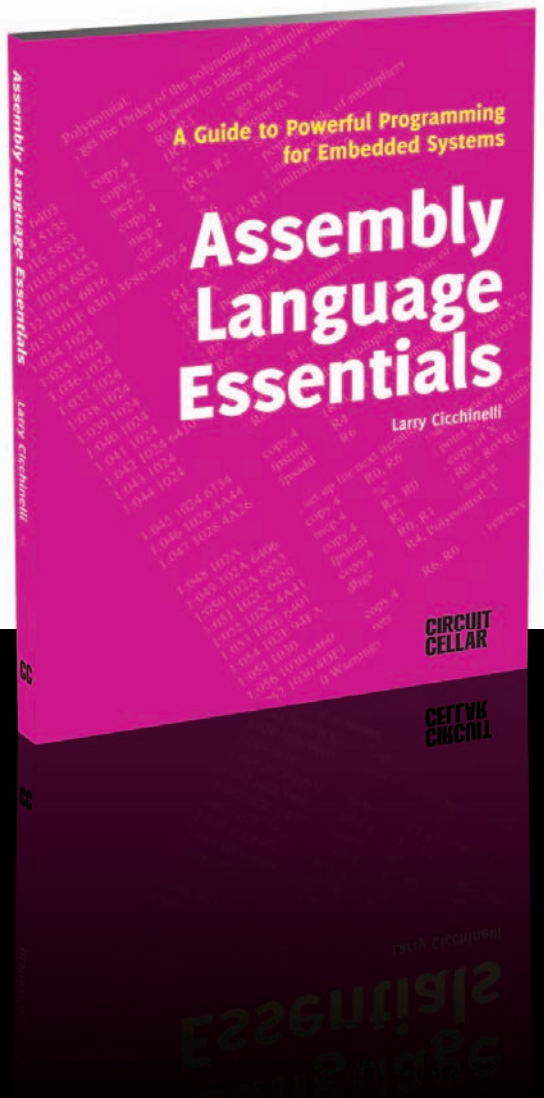
Larry's work/assembly space

The programming was completely in Assembly language. These data collectors were connected to a VAX computer via optically isolated RS-422 and the same muiltiplexor I designed for the data collectors. The programming in the VAX was a combination of Assembly language, FORTRAN, and C.

The last major project I

Assembly Language Essentials

Circuit Cellar's first book, ***Assembly Language Essentials***, is a matter-of-fact guide to Assembly that will introduce you to the most fundamental programming language of a processor.



Author Larry Cicchinelli provides readers with:

- An introduction to Assembly language & its functionality
- Essential terminology pertaining to higher-level programming languages & computer architecture
- Important algorithms that may be built into high-level languages — multiplication, division, and polynomial evaluation
- Overview of Interrupt Service Routines
- Free, downloadable Assembler program …

and more!

On Sale NOW!

$47.50

WWW.CIRCUITCELLAR.COM • WWW.CC-WEBSHOP.COM

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

worked on as the lead engineer was a series of test sets for car instrument clusters. These testers used both a PC and a MAC. The PC was used as the main controller while the MAC was used as a vision processor utilizing 20 multiplexed CCD cameras. Up to four clusters could be tested at the same time. Two units would be loaded by an operator in the front of the system where functional testing would be completed. When done the functional testing, the units would be rotated to the back of the system to face the cameras for visual testing that included calibrating of the meters.

NAN: What's your current occupation?

LARRY: Z-World/Rabbit Semiconductor was purchased by Digi International in 2005. I started working for Z-World in March 2000, about three months after the Rabbit 2000 was introduced. In about May of that year, I became the technical support manager and have held that position since then. My primary responsibilities are to manage my team of five and to help customers successfully use our products. We help customers with both hardware and software issues.

NAN: What types of projects are you working on at Digi International right now?

LARRY: My team is learning some newer products that aren't Rabbit-based. One of them is a programmable ZigBee unit. One of my team members and I have started a project to use one of these units to control an autonomous-roving vehicle. This is relatively simple in that it currently only has one sensor, a sonar unit that is used to detect obstacles and enable the algorithm to find the longest obstacle-free path. We started doing a blog at e-techgarage.com with some videos of the original model which used a Rabbit controller.

NAN: You wrote an online "Ask Larry" column for Digi for a few years. Do you still write it?

LARRY: The column has been suspended for a while, but I am in the process of bringing it back.

NAN: Speaking of writing, in 2004, *Circuit Cellar* published your article "Single-Board Logic Analyzer." What was the impetus for that project? Do you still use it?

LARRY: I have always liked "playing" with test equipment and decided to just see if I could do it. I had published an article in *QEX* in 2000. It was about a logic analyzer designed with discreet logic devices. This time, I wanted to use a PLD I had not previously used. Yes, I still use it to help debug my digital projects.

NAN: In 2010, *Circuit Cellar* published your article "Calibrated Decibel Meter Design." Can you give us some examples of how you've used the calibrated decibel meter? Have you made any upgrades to it?



The calibrated decibel meter with the selectable gain amp and a step attenuator

LARRY: I have used it to measure the output levels of microphones as well as audio preamps and amplifiers. I have not made any upgrades.

NAN: Your MiniSBC design is on your personal website. Tell us about it. What does it do and how does it work?

LARRY: One of the major products Digi International sells that is Rabbit-based is single-board computers. These are programmable devices that can be used in many industrial and communications applications. Another of our products is small, low-cost Rabbit-core modules (RCMs). These are small circuit boards that have a Rabbit microprocessor and all the peripheral circuitry required to make them work. They make the processor pins available on the Input/Output connector. All they need are a single power supply and a program.

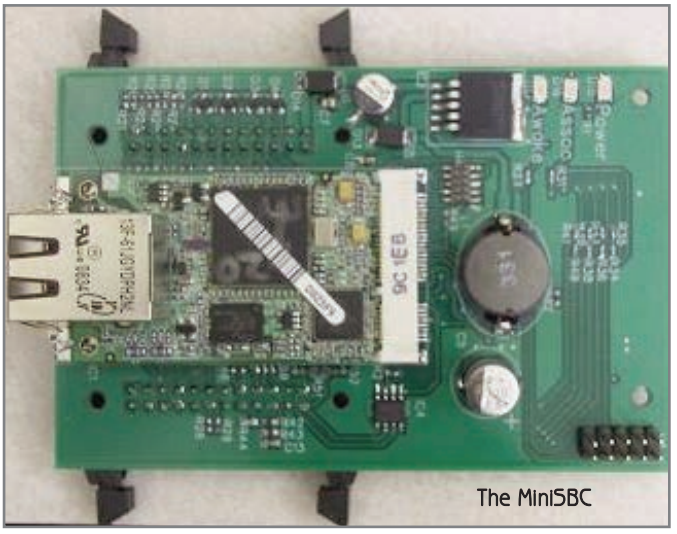
I wanted to see how small a board I could design that used one of these RCMs and still have it be useful in an industrial environment. The MiniSBC on my website (www.e-techgarage.com/Larry) is just a little bit larger than its controlling RCM and has the following specifications: eight open-collector outputs capable of sinking 200 µA each, eight digital inputs protected to ±30 V, two RS232 serial ports, one RS-485 serial port, eight analog inputs (0 to 10 VDC), and two analog outputs (0 to 3.3 VDC). I have also designed and built several more MiniSBCs using newer RCMs.

NAN: This year *Circuit Cellar* published your book titled *Assembly Language Essentials*. Many of our columnists and contributors use Assembly. Do you?

LARRY: Yes, I do use Assembly language. Quite a few



The calibrated meter with just the attenuator


The MiniSBC

Rabbit customers use Assembly language when they want a section of their program to run as fast as possible. In most cases, interrupt service routines (ISRs) are written in Assembly language in order to respond to interrupts and exit as quickly as possible. For faster processors and/or when ISR time is not as important, using a higher-level language may be a better option.

NAN: At the 2011 Embedded Systems Conference in San Jose, CA, several people who stopped by the *Circuit Cellar* booth asked our editors why they should learn Assembly. What's your answer to this question?

LARRY: I have always liked Assembly language—going back to my experience with the 8080 when it was the only option. Obviously compiler writers, at some level, need to know the Assembly language of the processor. I think it also causes the programmer to learn more about the hardware of the processor. In general, there are two main reasons for using Assembly language: When the fastest possible speed is required, and when the memory budget for a project is limited.

As I state in my book, compilers today generally create very efficient code. However, there are times when hand coding is required in order to get the fastest speed and minimum memory usage. Just like any engineering project, every programming task needs to be evaluated to determine the best tools for the job. There are still times when Assembly language is that tool.

NAN: What can readers expect to learn from your book?

LARRY: My hope is that it will be used to enable the reader to gain an appreciation of Assembly language and to not be "afraid" of it. I have heard many times from our customers that they believe it is much too difficult for them to do anything useful with it. I think the reader who understands the contents of my book and uses its samples will be able to use the knowledge gained from it on any other processors the reader has to learn.

NAN: Let's switch the focus a bit and discuss embedded system technology in general. What new technologies truly excite you and why?

LARRY: I think the most exciting is the ongoing effort to make CPUs smaller and faster with lower power and cost. These features make it feasible to connect even very small and low-cost systems to the Internet. The RCM in my function generator has 10/100 Ethernet capability although I will not be using it. Another version of the same RCM has Wi-Fi.

NAN: Which came first? Was it your interest in electronics design or your interest in ham radio?

LARRY: I think it was electronics that led to ham radio as a means to learn electronics.

NAN: So, let's summarize. You are interested in ham radio, Assembly language, and building your own test equipment. What's your true passion?

LARRY: I really enjoy designing and building electronic devices—primarily some type of test device, either measurement or stimulus. As you can see, I have done both analog and digital units. Not all of which used a microprocessor.

I also enjoy my time on ham radio—I have been licensed as K3PTO since 1961—but I don't get as much time as I would like. I usually operate PSK31 on 20 meters.

NAN: Are there any new projects at your workbench?

LARRY: The device I am currently building is based on a Rabbit RCM5700 module. It is a function generator that has multiple synchronized pulse outputs. It will also have a direct digital synthesizer. There is a menu system that uses an LCD, joystick, and quadrature encoder. Most of the programming is in C, but Assembly language is used where I wanted more speed in order to keep the response time as short as possible.


Larry's computer, ham radio (FT450), and his current project, a function generator

ICbank
Gateway to Electronics

MP3P DIY KIT, Do it yourself

(Include Firmware Full source Code, Schematic)

• myPIC



Only
$160 qty 100 **$220** qty 1

Powerful feature

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

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

• myWave (MP3 DIY KIT SD card Interface)



Only
$150 qty 100
$200 qty 1

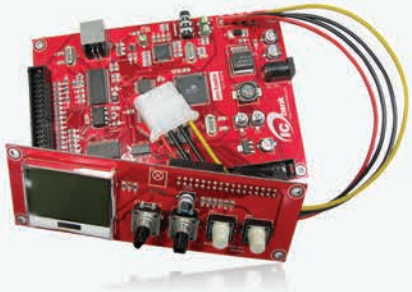
Powerful feature

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

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

• myAudio (MP3 DIY KIT IDE)

Only
$180 qty 100
$220 qty 1



Powerful feature

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

ICbank
Gateway to Electronics

www.icbank.com


The industry's brightest minds
come here to shine.
Every year one place brings together the innovators on technology's cutting edge.
And for those four days, Vegas glows brighter.
2012 International CES
THE GLOBAL STAGE FOR INNOVATION
PRODUCED BY CEA
Tuesday, January 10–Friday, January 13, 2012
Las Vegas, Nevada | CESweb.org
REGISTER NOW

by Ed Nisley (USA)



Stepper Motor Drive (Part 2)

Microstepping

Most new stepper systems depend on microstepping motor drivers with active current regulation. This article covers the fundamentals and advantages of microstepping motor drivers and describes how their chopper current controls function.

Although the classic L/4R full-step drivers you saw in my August 2011 *Circuit Cellar* column may have been acceptable in the early days of stepper motor mechanisms, advances in power semiconductors and logic circuitry rendered them obsolete. Most contemporary stepper systems depend on microstepping motor drivers with active current regulation, often integrated into a single chip. The advantages include greatly reduced power dissipation and smoother motion, with the disadvantages of reduced torque and more complex driver configuration.

In this column, I'll cover the fundamentals of microstepping motor drivers and show how their chopper current controls work. Along the way, you'll discover why steppers sometimes sing simple songs.

TEST CONFIGURATION

The schematic in Figure 1 could represent nearly any stepper motor driver system: a big controller block that converts logic-level Step and Direction inputs into the proper winding currents for a bipolar stepper motor. Additional controller inputs select the microstepping mode, enable the driver, and set other options.

The other blocks represent the oscilloscope sync and current measurement hardware I used for this column. Photo 1 shows the tangle of wires and probes surrounding the green Pololu A4988 Stepper Motor Driver Carrier board on my solderless breadboard, with a NEMA 17 stepper motor atop the case.

I'm using the Pololu board because it's simple, inexpensive, and shows how far stepper driver design has come. When mounted on an adequate heatsink, the board's Allegro A4988 DMOS Microstepping Driver IC can control up to 2 A of winding current from a 35 V supply, even though the tiny Pololu board covers barely 0.5 in^2 and its 16 I/O pins account for much of that area. I made a small aluminum heatsink that fits around the board, but removed it for the picture.

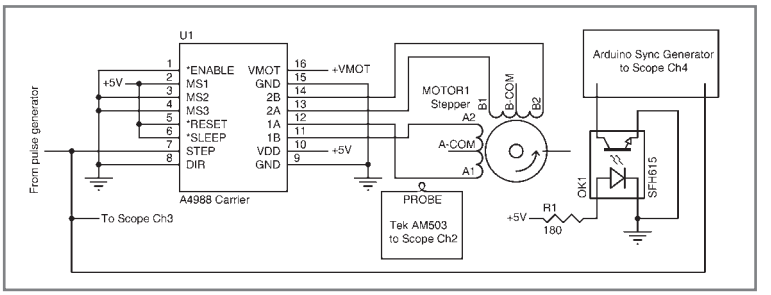


Figure 1—The test setup uses a pulse generator to set the step rate and a current monitor to measure the winding current. The Arduino microcontroller board produces a scope sync pulse aligned with each group of four full steps and locked to the motor rotation with an optical interrupter.

The surplus NEMA 17 motor in this column is a Minebea 17PM-J034, which is evidently a custom motor not included in their catalog. It works better with a microstepping driver than the high-resistance and high-inductance L/R motor you saw in the last column. It has a winding resistance of 2.3 Ω

and an inductance of 2.6 mH, giving an L/R time constant of about 1 ms. While that's about the same as the earlier motor, the much lower inductance enables the much faster current changes required by a microstepping drive.

As you've seen in my earlier columns, you shouldn't use a solderless breadboard for anything other than a quick demonstration circuit. In fact, those breadboard pins aren't rated for more than about an amp and the inadequate power and ground distribution almost guarantees problems with any analog circuit. If your circuit works on such a breadboard, good for you, but when it doesn't, you should use a better breadboarding technique!

MICROSTEPPING FACTORS

Internally, the A4988 resembles the schematic shown in Figure 1 from my April column. Four transistors in an H-bridge configuration connect each motor winding to the power supply, driving current in either direction through the coil. In addition to the two H-bridges, the A4988 includes digital step control circuitry and analog circuits that measure and limit the winding current.

A microstepping controller divides each full step into a fixed number of smaller ("micro") steps and adjusts the current in both windings to interpolate the rotor position accordingly. A stepped sine wave provides smooth motion, although earlier microstepping drives used other interpolations.

Photo 2 shows the actual current waveforms in both motor windings during the course of four full motor steps, with the motor controller in 1/8 microstepping mode. Because I have only one current probe, I stored the first trace and retriggered the scope with the probe on the other winding.

The two positive and two negative peaks of the current waveforms in Photo 2 correspond to the four full-step positions shown in Table 1 of my August column: full current in a single winding aligning the rotor with the armature poles. The rising edges in Trace 3 at the bottom of the screen correspond to the start of the microsteps, with eight pulses between successive peaks.

That sine wave pattern repeats for each group of four full motor steps. The motor I'm using has 200 full steps per revolution, so each revolution has 50 groups and, in 1/8 microstepping mode, 1,600 microsteps per revolution.

Some confusion arises because motor controllers have a *Step* input that accepts pulses at either the full or the microstep rate.

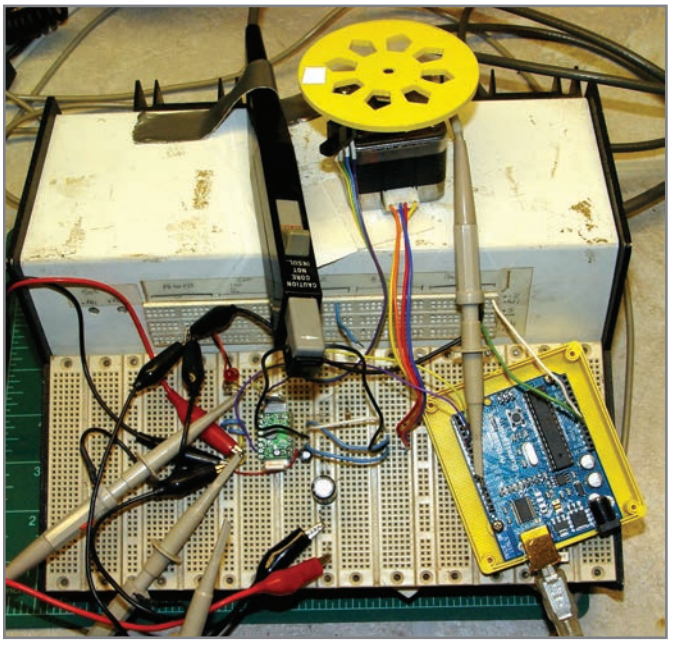


Photo 1—This ungainly collection of parts, wires, and probes produced the data for this column. Don't do this in a production system! The yellow wheel provides a 1/rev pulse to the Arduino board for scope synchronization.

However, datasheets give the motor specifications for the full step rate, so you must adjust their values according to your controller's microstep configuration.

TRIGONOMETRIC CURRENT

As you saw in my August column, the maximum winding current determines the motor's torque and the power dissipation in the windings. The peak value of the microstepping current waveforms must not exceed that specification, because when the motor stops at that microstep what looks like an AC peak value becomes the winding's DC dissipation.

The A4988 limits I_{MAX}, the maximum winding current, in proportion to the voltage at its V_{REF} input pin. That voltage may come from a fixed resistor divider, a trimpot, or a microprocessor's DAC, but is generally regarded as a stable DC input value.

The controller compares V_{REF} with the voltage from a current-sensing resistor that converts the winding current into a small voltage. The Pololu board uses 50 mΩ sense resistors and the A4988 sense amplifier applies a gain of 8, so the

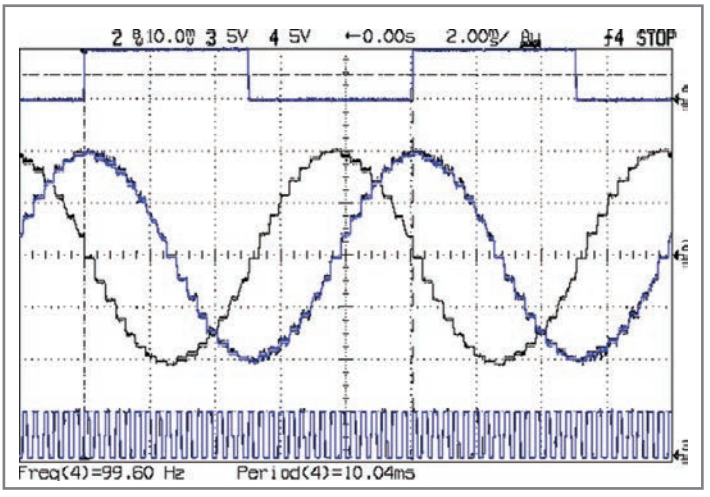


Photo 2—Each cycle of these sinusoidal current waveforms (blue = Winding A, black = Winding B) show the 32 microsteps corresponding to four full steps. The rising edge of each pulse in the lower trace marks the start of a microstep. The vertical scale is 500 mA/div.

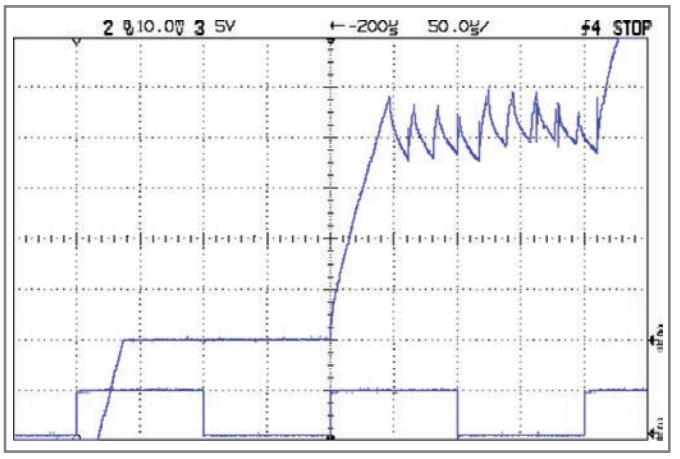


Photo 3—The current changes abruptly at the start of each microstep, with 18 V applied to a 2.3 Ω winding, The A4988 driver limits the current to about 200 mA for the first microstep. The vertical scale is 50 mA/div.

relation between V_{REF} and I_{MAX} will be:

$$I_{MAX} = \frac{V_{REF}}{8 \times R_{SENSE}}$$

Rearranging that equation to solve for V_{REF} shows that a 1 A maximum current requires a trimpot setting of:

$$0.400\ V = 1\ A \times (8 \times 0.050\ \Omega)$$

As you would expect, the winding current for each microstep is a fraction of the peak current determined by a sine function with a period of four full steps. In 1/8 microstepping mode, each full sine wave cycle has 32 microsteps and the fraction of the maximum current for microstep N (in the range 0 through 31) will be:

$$I_{Winding\ A} = I_{MAX} \times \cos\left(\frac{N}{32} \times 360°\right)$$

The phase of the current in Winding B relative to Winding A determines the direction of the motor's rotation. With the Direction input at 0 V, as shown in Photo 2, the phase is:

$$I_{Winding\ B} = -I_{MAX} \times \sin\left(\frac{N}{32} \times 360°\right)$$

Setting the Direction input high flips the phase:

$$I_{Winding\ B} = I_{MAX} \times \sin\left(\frac{N}{32} \times 360°\right)$$

The A4988 uses a look-up table that converts the current microstep number into the corresponding fraction for that microstep: it doesn't do trigonometry on the fly! The MSx microstepping mode pins select the increment used to advance the table index for each Step pulse, while the Direction pin determines whether the increment is added or subtracted from the index.

Incidentally, you can also reverse a bipolar motor's direction with a wiring change, because swapping the two leads of one winding inverts the current in the winding. Although the datasheet gives the motor's winding polarity and you should be able to deduce the Direction signal polarity from first principles, the probability that the motor will rotate in the intended direction when you configure a system seems to be almost exactly 0.50.

CHOPPER CURRENT CONTROL

Because a stepper motor's torque depends on its winding current, an ideal driver would produce an instantaneous current change in the winding and a rectangular current pulse filling each microstep. Alas, that's simply not possible, because wrapping a coil of wire around an iron motor armature makes a good inductor.

Classic L/4R drivers improved the risetime by reducing the time constant with a high resistance, at the cost of greatly increased power dissipation. Microstepping drivers sidestep the winding's inherent L/R time constant with a relatively high voltage and active current limiting.

Both methods depend on the same fundamental equation:

$$V_L = L \frac{\Delta I}{\Delta T}$$

Increasing the voltage across the inductor produces a proportionally faster change in winding current. The problem then becomes how to handle the increased power dissipation from the higher current.

That equation shows why a better microstepping motor must have a low winding inductance: at a given voltage, the rate of current change varies inversely with the inductance. The motor's winding resistance will allow an extremely high current at the applied voltage, but active current limiting holds the current to the much lower value required for each microstep.

Photo 3 gives a detailed view of the winding current at the start of the microstep near the middle of Photo 2. With a 1 A peak, the current for that microstep is:

$$200\ mA = (1\ A) \times \cos\left(\frac{25}{32} \times 360°\right)$$

The initial risetime to 240 mA is about 45 µs, which indicates the voltage across the winding is:

$$13.9\ V = 2.6\ mH \times \frac{240\ mA}{45\ \mu s}$$

I had the bench supply set to 18 V, so 4.1 V vanished in the MOSFETs, winding resistance, back EMF, and other wiring losses. The two MOSFETs in the A4988 add a total of 800 mΩ, so the L/R time constant is:

$$840\ \mu s = \frac{2.6\ mH}{2.3 + 0.8\ \Omega}$$

That is much longer than the 200 µs microstep time, so linear approximations to the actual exponential curves will give reasonably accurate results.

Unlike the smooth exponential rise of the L/4R drive shown in Photo 2 in my April column, the A4988 limits the current by turning its MOSFETs off when the current reaches the upper limit for each microstep, then turning them on after a fixed delay. The Pololu board sets that delay to 12 µs, which corresponds to the duration of the falling current in Photo 3.

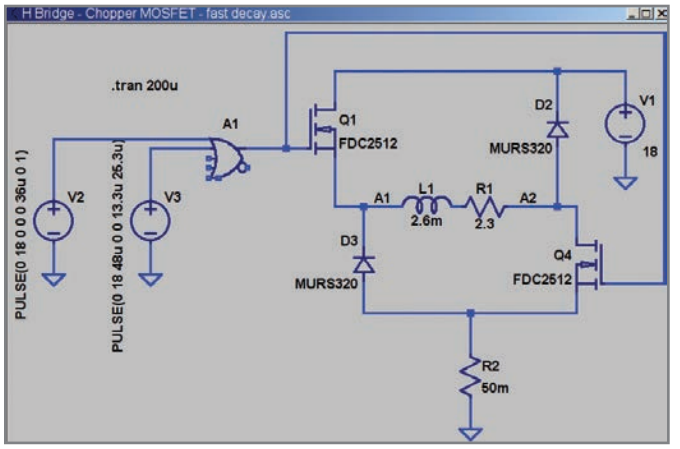


Photo 4—This simulation model shows how the winding current behaves in fast decay mode, with only MOSFETS Q1 and Q4 active. D2 and D3 represent the body diodes of Q2 and Q3, the inactive transistors of the H bridge.

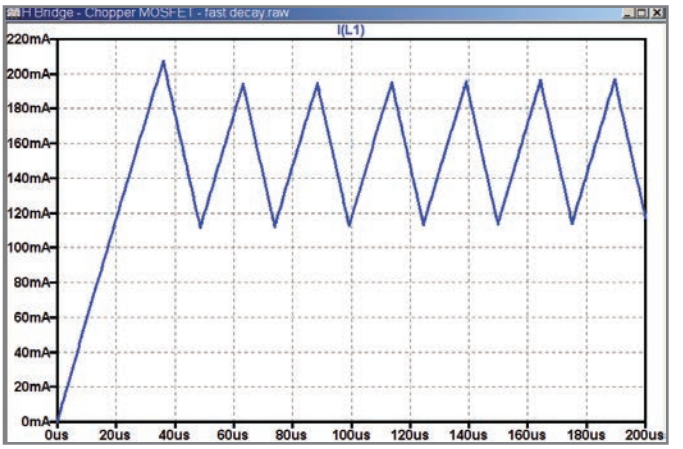


Photo 5—Fast decay mode produces relatively high current ripple, because the MOSFET body diodes allow the full power supply voltage to appear across the winding as the current decreases.

The A4988 datasheet uses the terms *fast decay*, *mixed decay*, and *slow decay* without precisely defining them. It turns out that they refer to the rate of current change while the MOSFETs are turned off and, thus, the voltage applied to the winding. It's easiest to understand those terms by using simulation models of the A4988 circuitry that show how the current behaves.

The LTSpice IV simulation model in Photo 4 includes Q1 and Q4, the two H-bridge MOSFETs that control the winding current for the microstep in Photo 3, plus D2 and D3, the body diodes for Q2 and Q3, the H-bridge MOSFETs that remain off in these conditions. Although I picked discrete MOSFETs and diodes with properties similar to those in the A4988, this model cannot produce results that exactly match the actual chip.

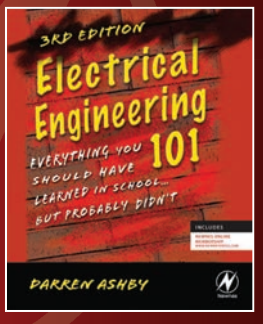
In addition, I simulated the chip's complex analog current sensing and digital control circuitry with a pair of pulsed voltage sources tweaked to produce equivalent

Newnes Press

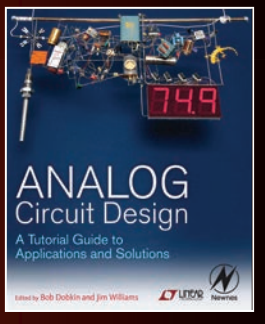
Education Never Ends. Everything you need to know to get started.



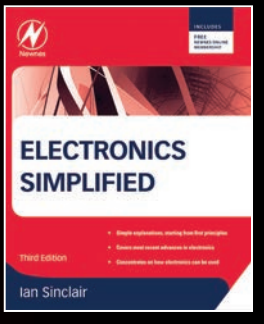
By: Keith Brindley
ISBN: 9780080969923



By: Darren Ashby
ISBN: 9780123860019



By: Bob Dobkin and Jim Williams
ISBN: 9780123851857



By: Ian Sinclair
ISBN: 9780080970639

Newnes

Save 20% on all titles when you purchase from www.newnespress.com.
Enter promotional code Newnes30 at checkout.

Find us on Facebook!

Register for our e-news at newnespress.com
Receive our best discounts · Hear about books before they publish
Access to free sample chapters, video tutorials and more!

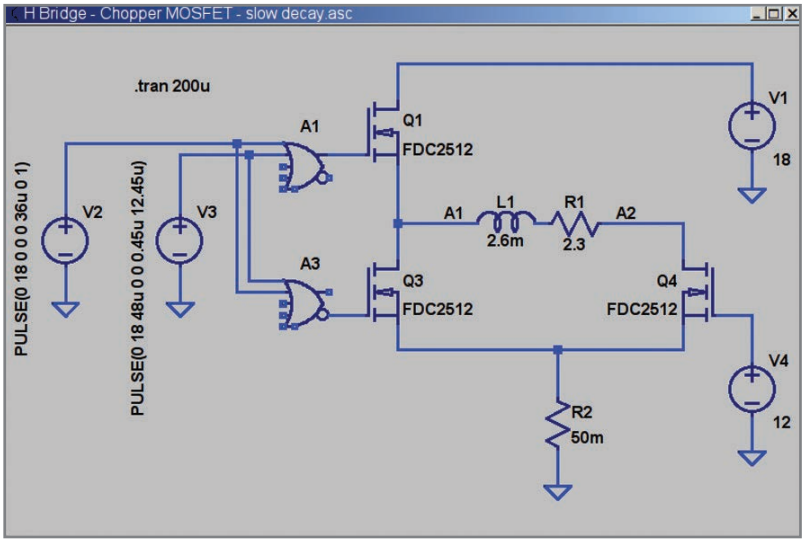


Photo 6—This simulation model shows how the winding current behaves in slow decay mode. Both lower MOSFETs are on when Q1 turns off, trapping the winding current in the lower loop of the H-bridge.

results; a more detailed simulation lies well beyond the scope of this column.

Photo 5 shows that the simulated current resembles the actual current in Photo 3. The current ripple is much larger, however, because the simulation operates in what Allegro calls *fast decay* mode, with the winding current passing through D2 and D3 when Q1 and Q4 turn off, connecting the winding directly across the power supply. The slope of the falling current is thus somewhat higher than that of the rising current:

$$17\ \text{V} = 2.6\ \text{mH} \times \frac{80\ \text{mA}}{12\ \mu\text{s}}$$

Note that the current through R2, the sensing resistor, reverses direction when the winding current declines. If you measure the current with an oscilloscope probe on the sensing resistor, you'll see that voltage go negative during this part of the cycle.

The A4988's *slow decay* mode turns on both lower MOSFETs in the H-bridge when the upper MOSFETs turn off, trapping the winding current in the lower loop of the H-bridge. This takes advantage of three MOSFET characteristics: low on-state resistance, no saturation voltage, and bidirectional conduction.

The simulation model in Photo 6 resembles Photo 4, but now Q3 turns on to conduct current around D3, its body diode, from source to drain when Q1 turns off. A quirk in LTSpice requires separate logic OR models to properly drive the two MOSFET gates.

Photo 7 shows that slow decay mode produces much lower current ripple during the microstep than the fast decay mode in Photo 5; both waveforms have the same axis scales and the ripple has identical off times. Zooming in on the waveform (which you can do in the downloaded simulation model) gives a surprisingly low value for the voltage across the inductor:

$$0.56\ \text{V} = 2.6\ \text{mH} \times \frac{2.6\ \text{mA}}{12\ \mu\text{s}}$$

Reduced ripple current decreases energy loss due to hysteresis in the armature's iron core, which also reduces audible noise from magnetostriction. However, the ripple frequency also drops, which may move the remaining audible noise into the human hearing range. You must consider the interactions between the waveform's harmonics, subharmonics, and their mixing products with the stepping frequency; nearly anything seems possible.

That acoustic energy produces the high-pitched whine coming from a stopped stepper motor. When the motor is in motion, however, the motor may produce audible mechanical vibrations: in 1/8 microstepping mode, a 200 step/rev motor turning at 1 rev/s hums at 200 Hz, 1,600 Hz, and their harmonics. Follow the Resources links to stepper motor motion sequences that play music.

The A4988 generally uses *mixed decay* mode, which switches from fast decay to slow decay mode as the current decreases during the off time. Photo 3 doesn't have quite enough resolution to clearly show this, but you can see a slight bend at about one third of the off time as the controller switches modes. Simulating that logic requires more complexity than my simple models can provide, although another brace of carefully adjusted voltage sources should do the trick.

The shape of the winding current waveform shows why microstepping is also known as *chopper current control*. Because the MOSFETs run either fully on or fully off, they dissipate relatively low power, which enables a very small chip to control surprisingly large motors. The A4988 can control 2 A from a 35 V source, which is enough for many NEMA 23 size steppers.

Of course, you must run the numbers to ensure the chip temperature remains within acceptable limits, but the days of huge power resistor banks in ventilated enclosures above CNC machines are long gone: good riddance!

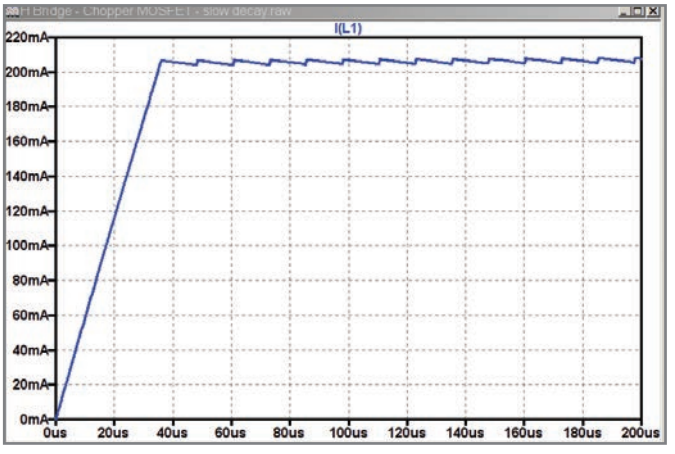


Photo 7—Slow decay mode produces very low current ripple, because the MOSFETs circulate the current with very low voltage drop.

CONTACT RELEASE

This column discussed the rising-current part of the waveform. The other side of the sine wave encounters somewhat different conditions due to the spinning rotor's back EMF. My next column will explore the mechanical end of stepper motors, where you'll see that voltage in action.

My buddy Eks provided an antique Tektronix AM503 current probe amplifier and an A6302 Hall-effect current probe to capture the current waveforms. The contemporary A622 probe should be a more available, albeit still pricey, alternative. The amplifier must have a bandwidth starting at DC to show the proper waveforms, which eliminates clamp-on inductive current probes.

Synchronizing the oscilloscope to the motor rotation and step pulses required a quick-and-dirty Arduino program using the ATmega168's counter hardware to eliminate all the usual software jitter. That was easier than wiring up a bunch of gates and counters, which is the reason why microcontrollers exist!

NEED-TO-KNOW INFO

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

—

Electric Vehicle Inverter Design
Build a System for Powering AC Induction Motors
by Dan Hall, Tristan Kasmer, Doug Krahn, Adam McIntyre, & Dena Ponech
***Circuit Cellar* 217, 2008**

The Electric Vehicle (EV) Inverter project is a starting point for the conversion of a vehicle with a gas engine to one with an electric motor. The inverter was built using a control board and software based on Microchip Technology's MC-1 development board. The finished design can power a wide range of AC induction motors. Topics: Inverter, AC Induction Motor, Vehicle, EV, ACIM, MC-1, IGBT, Gate Driver, BG2B-5015, Space Vector Modulation, PID, CAN

—

Inertial Rolling Robot
by Jeff Bingham & Lee Magnusson
***Circuit Cellar* 200, 2007**

Jeff and Lee's H8/3664-based rolling robot is capable of inertial movement. A DC electric motor is attached to a pendulum and suspended inside an inflated ball, which provides the driving force. Topics: Robot, Rolling, H8/3664, Pendulum, Sphere, Motor, H-bridge, IR2184, PWM, Hall Effect, ACS706, Servo, Pressure Sensor, ADC, MPX5100A

—

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

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

PROJECT FILES

To download the LTspice simulation models and Arduino source code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/255.

RESOURCES

Makerbot Industries, Makerbot Music, http://wiki.makerbot.com/makerbot-music, 2011

E. Nisley, "Stepper Drive (Part 1): Analog," *Circuit Cellar* 169, 2004.

———, "Stepper Drive (Part 2): Digital," *Circuit Cellar* 171, 2004.

———, "Stepper Failure: Death By Disconnection," *Circuit Cellar* 191, 2006.

———, "Stepper Motor Drives (Part 1): The Basics," *Circuit Cellar* 253, 2011.

RepRap.org, Eagle Library, http://reprap.org/wiki/Eagle_Library, 2010.

SOURCES

Allegro A4988 DMOS Microstepping Driver
Allegro MicroSystems, Inc. | www.allegromicro.com/en/Products/Part_Numbers/4988

Microcontroller boards
Arduino | arduino.cc/en

ATmega168 Microcontroller
Atmel Corp. | www.atmel.com

Eagle Schematic and PCB software
CadSoft | www.cadsoftusa.com

Design Simulation and Device Models
Linear Technology Corp. | www.linear.com/designtools/software

17PM-J034 Motor
Minebea Co., Ltd. | www.eminebea.com

A4988 Stepper Motor Driver Carrier
Pololu Corp. | www.pololu.com/catalog/product/1182

AM503 Current probe amplifier and A6302 Hall-effect current probe
Tektronix, Inc. | www.tek.com/products/accessories/current.html

by George Novacek (Canada)



Environmental Stress Screening

Environmental stress screening (ESS) takes time, resources, and equipment. But the advantages—low field returns, design robustness, and component quality tracking—are well worth it.

Your product has passed final inspection and testing with flying colors. Now it's time for it to go to the shipping department and out to the customer. But wait! Not without environmental stress screening (ESS)!

ESS EXPLAINED

ESS is sometimes incorrectly thought to be a burn-in; but, in fact, it is a product-specific, advanced type of accelerated burn-in. It's an economical way to weed out latent and patent defects in your product to assure quality and reliability in the field. No conscientious manufacturer allows its product to be shipped without it. To understand why, let's consider Figure 1. The purple trace (not to scale) is the classic "bathtub" depicting the lifecycle of just about any product. It comprises three separate failure-rate curves: Initial-quality failures, Stress-related failures, and Wear-out failures.

Initial-quality failures, also called Infant Mortality, are the result of latent defects caused by material flaws, workmanship, manufacturing processes, handling, and the assembly process. After some operating time, the latent defects will become overt and the product will fail.

Stress-related failures, shown by the green trace, are the result of the statistical failure rate calculated from the individual component failure rates, based on historical data. This failure rate—a reciprocal of the mean time between failures (MTBF)—remains constant.

Wearout failures, the red trace, signify the end of the life of the product due to wear and tear. This is a concern with moving or otherwise stressed parts, but rarely with well-derated electronic components, as they usually far exceed the life expectancy of the product. Derating means that, by design, during operation the components never exceed a fraction (typically 50% or less) of their rated parameters. So, a 0.5-W resistor will never dissipate more than 0.25 W.

ESS stresses the product before shipping to push it beyond Infant Mortality to significantly reduce costly field returns. High stress levels accelerate the aging process to achieve the crossover into the Stress-related failure mode in typically one to three days.

Even though ESS is product specific, it is often performed based on "something we've done before" or "something the customer wants." If the ESS procedure is merely based on some arbitrary decisions or a rule of thumb, the results will show it. You may be wasting time and money without getting over Infant

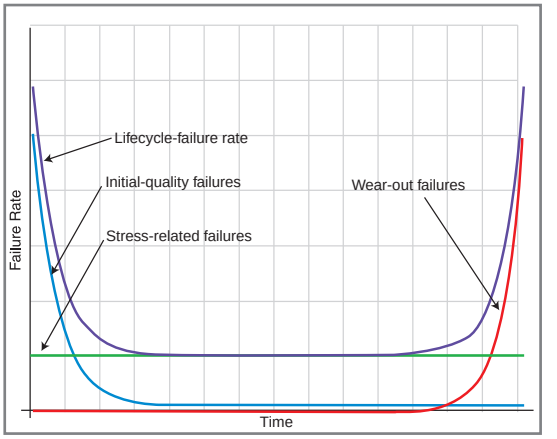


Figure 1—Lifecycle-failure rate "bathtub" diagram

Subscribe to *audioXpress* magazine!

Do your **electronics speak** to you? Are the words **"audio", "vacuum tubes",** and **"speaker technology"** music to your ears?

Then you should be reading *audioXpress!*

Recently acquired by The Elektor Group, *audioXpress* has been providing engineers with incredible audio insight, inspiration and design ideas for over a decade. If you're an audio enthusiast who enjoys speaker building and amp design, or if you're interested in learning about tubes, driver testing, and vintage audio, then *audioXpress* is the magazine for you!

What will you find in *audioXpress*?

- In-depth interviews with audio industry luminaries
- Recurring columns by top experts on speaker building, driver testing, and amp construction
- Accessible engineering articles presenting inventive, real-world audio electronics applications and projects
- Thorough and honest reviews about products that will bring your audio experiences to new levels

Choose from print delivery, digital, or a combination of both for maximum accessibility.



Subscribe to *audioXpress* at
www.cc-webshop.com
today!

Save $5 by entering promo code AUDIO at checkout.

*Discount good thru 10/31/11.

audioXPRESS

Mortality, thus not sufficiently reducing the product returns. Or, you may be overstressing the product, introducing latent defects to come back to haunt you later.

This column does not give me sufficient space to take you through the ESS procedure development step by step. I can merely describe the process in a nutshell. Those interested in the details can find them in MIL-HDBK-344A "Environmental Stress Screening of Electronic Equipment" and other related MIL standards. These can be downloaded free of charge from https://assist.daps.dla.mil/quicksearch. Note that just because the military developed a handbook for the ESS procedure, it is by no means limited to military products. All products should be subjected to it.

TS & VS

Two methods are available for ESS: temperature cycling, which is otherwise known as thermal screen (TS), and vibration screen (VS). Table 1 shows the effectiveness of those screens.

Thermal screen alone is sufficient for most electronic devices, but not effective for discovering loose contacts, debris, loose hardware, chafed or pinched wires, and adjacent parts and boards shorting. For complex electronic equipment, a combined screen, called "Shake 'n Bake" may have to be used. TS alone will be sufficient in situations where the defects not covered by TS can be prevented by design or discovered by inspection.

Sometimes it's difficult to explain to customers or quality assurance (QA) people that ESS is a screen, not a test. Occasional failures are not a cause for panic, they are expected, or ESS is not doing its job. ESS is performed to weed out latent defects, not to verify the design.

Development of the ESS procedure starts by determining defect density (DD):

$$DD = N \times \bar{P} = \sum_{i=1}^{m} n_i \times p_i$$

where:

N = total number of parts in the equipment

$\bar{P}$ = average part fraction defective over all parts

n_i = quantity of each part type i

p_i = detected density of each part type i

m = number of different part types

DD is the average number of latent defects per assembly. It is calculated at baseline stress and adjusted to reflect defect removal through testing during manufacturing. DD is then adjusted for the actual stress expected in field.

In the next step, precipitation efficiency (PE), which is the probability that the screen will precipitate a defect to a detectable stage, is calculated. This is where decisions are made about the use of TS and VS, stress levels, etc. You want to stress the product as much as possible to shorten the ESS, but without overstress causing irreversible damage.

Next, detection efficiency (DE) is established. It is affected by three factors, test type (TT), environmental conditions (EC), and defect removal (DR). These are empirical values listed in MIL-HDBK-344A. TT reflects the type of testing during the ESS, such as functional or parametric (by flying probe). EC is based on the environmental conditions of the test, whether it will be performed at the ambient conditions before and after the ESS only, or continuously during the ESS at various temperatures. Finally, DR predicts the ability to isolate the defect and the probability of successfully removing it without introducing another defect.

ESS effectiveness or screening strength (SS) is the product of PE and DE: SS = PE × DE. Customer reliability requirement in a given environment is often given in hours MTBF. With a safety margin (SM) typically 1.5, the permissible latent failure rate (FR) in field is:

$$FR \leq \frac{1}{MTBF} \times \frac{1}{SM}$$

from which the maximum permissible DD after the ESS, $D_{REMAINING}$, is calculated:

$$D_{REMAINING} = \frac{FR \times t}{SAF \times \left(1 - e^{-k \times t}\right)}$$

where t is the stress duration in field, SAF is the stress adjustment factor and k is an empirically defined field stress constant.

ESS always results in reduction of life expectancy of the unit exposed to it because it accelerates aging through Infant Mortality to quickly reach constant failure rate. This life reduction is calculated as damage index and can usually be kept to 3% to 4% of useful life.

To perform an ESS you need to determine the thermal response of the product to decide on the minimum duration of the thermal cycle. This can be accomplished by testing, exposing the product with a thermocouple in it to thermal steps. With the thermal cycle, for instance, from –40°C to 80°C and vice versa at 10°C per minute, you determine the time it takes to reach thermal equilibriums. During the ESS, the product will dwell at those temperature extremes while the functional/parametric tests are performed.

Defect Type	TS	VS
Defective part	x	x
Broken part	x	x
Improperly installed part	x	x
Solder connection	x	x
PCB Etch, shorts, and opens	x	x
Loose contacts		x
Wire insulation	x	
Loose wire termination	x	x
Improper crimp or mating	x	
Contamination	x	
Debris		x
Loose hardware		x
Chafed, pinched wires		x
Parameter drift	x	
Hermetic seal failures	x	
Adjacent boards or parts shorting		x

Table 1—The effectiveness of the thermal screen (TS) and vibration screen (VS)

ESS RESULTS

ESS results are logged and reviewed regularly because their usefulness goes beyond field-return reduction. A frequent failure of one part may indicate a design weakness or a manufacturing problem. An increased failure of an inductor led us to discover its manufacturer changed the process to reduce cost while failing to inform the industry. The records can also be used to optimize the design by leveling the quality and thus reduce the cost of the entire assembly.

ESS takes time, resources, and equipment, but the benefits of low field returns, design robustness, and component quality tracking are worth the effort.

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

RESOURCE

U.S. Department of Defense, "Military Handbook: "Environmental Stress Screening (ESS) of Electronic Equipment," MIL-HDBK-344A, 1993.

NEED-TO-KNOW INFO

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

—

BITs and BITEs
Tests to Ensure System Safety
by George Novacek
***Circuit Cellar* 251, 2011**

Built-in test equipment is an important component in safety-critical systems. Design engineers must consider what's needed for safe operation and easy maintenance, and then balance safety and over-testing. Topics: Monitoring, Watchdog, Built-in Test

—

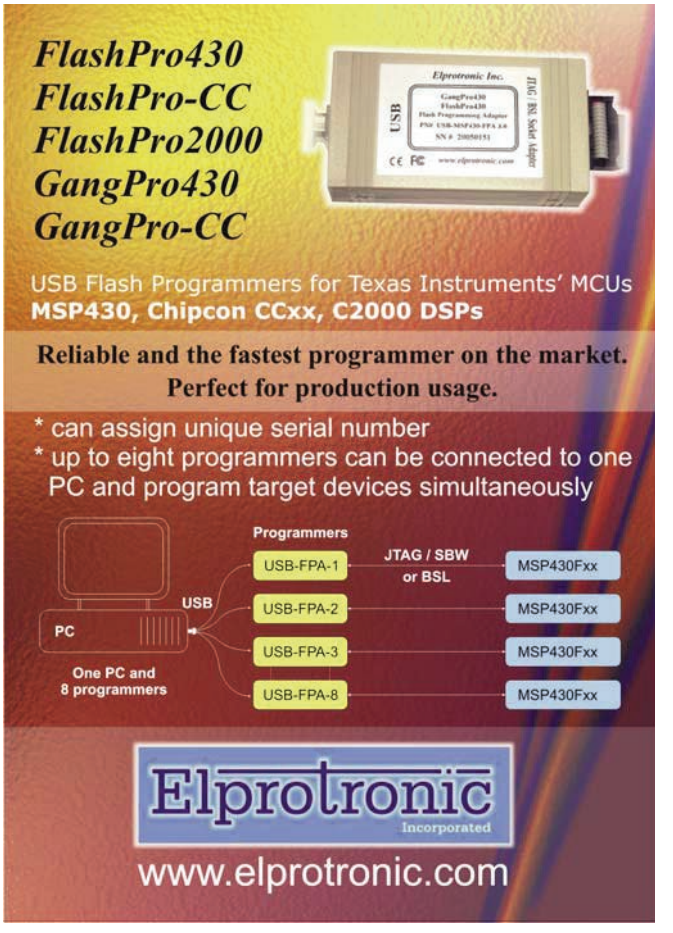
Error Checking
by Jeff Bachiochi
***Circuit Cellar* 250, 2011**

Data security isn't a perk. It's a requirement. This article covers the topics of error checking, checksums, and the cyclic redundancy check. Topics: Error Checking, Checksum, Cyclic Redundancy Check

—

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





EMBEDDED IN THIN SLICES

by Bob Japenga (USA)

"Embedded in Thin Slices" is a new bimonthly column devoted to helping engineers hone their embedded programming and design skills. Topics such as embedded Linux and how to deal with concurrency issues will be covered.



Getting Started with Embedded Linux (Part 1)

When to Choose Linux for Your Embedded Design

This three-part series introduces the topic of embedded Linux. Why should you choose Linux? To begin with, many of the features, devices, and functionality you desire are already built into the operating system. Read on to find out more.

In the 2005 bestseller *Blink*, Malcolm Gladwell talks about the way we make decisions. In particular, he describes a process our brains go through called "thin slicing." Thin slicing describes our ability to find out what is really important with limited data. I have been designing embedded systems for 38 years, yet I frequently feel like a beginner. I always have limited data and a limited vision of what is coming next. The field is so broad, and the applications are so varied, that no one can fully understand even a small portion of it. If you feel the same way, this column is for you. In this and subsequent articles, I'll cover some thin slices of my experience in the field of embedded systems design.

For the next three articles, I'll share my experience with embedded Linux. This month I focus on choosing Linux for embedded applications. In the next two articles will discuss choosing a platform and a tool chain, as well as licensing issues.

RTOS FOR EMBEDDED SYSTEMS

Linux is a powerful tool for embedded systems. However, like any powerful tool, it can skew your perception of a problem. To a man with a hammer, every problem looks like a nail. Likewise, using Linux in embedded systems can be that way. It's so powerful, once you have invested the time to learn how to deploy it (and that investment is not cheap if you port your own version of the kernel—more on that next time) and know how to do it quickly, everything starts looking like a Linux target.

Perhaps a broader question needs to be answered first: When do you need a real-time operating system (RTOS) in your embedded design? The two thinly sliced answers I have found are: when you have more than two or three concurrency requirements, or when you have (or even may have) a number of complex generic devices that require device drivers or when you have one or more complex generic functional requirements.

CONCURRENCY REQUIREMENTS

Basically, concurrency means that you have to do two things at the same time—concurrently. These can creep up on you unawares. Certainly, we can all recognize the apparent ones. For example, a system that has two or more independent control loops is the most obvious one. But anytime you talk to interfaces that have variable response rates with significant delays you have added a concurrency requirement. For example, let's imagine a simple device with a simple TCP/IP stack. The single-thread user interface makes a call to "ping" a device to see if it's there. The stack may take anywhere from 1 ms to 20 s to return. How many of us use inane user interfaces daily that force us to wait the 20 s before the network times out—even to abort the call? That kind of interface puts a concurrency

requirement on your system.

Our company does a lot of projects with only a few concurrency requirements. These are nicely handled with a foreground/background scheme or a single main loop, using a few interrupt handlers for the concurrency. No RTOS is needed. But I have found that once you have a user interface and one or more slow interfaces, some sort of threading/multitasking is essential to keeping the design and user interface clean.

DEVICES & FUNCTIONALITY

One of the other things that drives us to choose an RTOS is the need to interface with a generic complex device. By generic, I am not talking about devices that are unique to our proprietary system. I'm talking about devices that have general utility, such as USB sticks, cameras, cell modems, wear-leveling flash memory, CODECs, PHYs, or Wi-Fi devices, for which device drivers have already been written. We simply cannot take the time to design the software drivers for the vast array of devices that some of our embedded systems need to talk to. Many RTOSes provide a wide assortment of these generic drivers to simplify the task of interfacing to them.

Similarly, sometimes our requirements contain requests for wheels that have already been invented, such as encryption and compression algorithms, mesh networks, remote access, and remote control. We design a lot of systems that often start with a small device and function complement. But user demand forces us to add devices and functionality we never dreamed of long after the design is complete. This is where a full-fledged RTOS like Linux pays for itself over and over again. These already invented devices and functionality are added for low-entry cost and we can concentrate on the problem domain we are trying to address.

Allow me to list a few of the devices and functionality that got added on to some of our designs long after the first release. These were features that we had not given a minute of forethought to in the original design.

The first feature was a full keyboard. A customer wanted to expand his user interface from our keypad to a full QWERTY keyboard. Solution with Linux: No change. Plug a USB keyboard in. No drivers to write. No queues to change.

The second feature was e-mail capability. A customer wanted to alert maintenance people with an e-mail rather than a fax. Solution with Linux: Add an existing open-source mail program and a simple interface to add the e-mail addresses.

> "Linux is a powerful tool for embedded systems. However, like any powerful tool, it can skew your perception of a problem. To a man with a hammer, every problem looks like a nail. Likewise, using Linux in embedded systems can be that way. It's so powerful, once you have invested the time to learn how to deploy it, everything starts looking like a Linux target."

The third feature was remote display and control. A customer wanted to be able to remotely control a custom QVGA display with a custom keypad from his PC. Solution with Linux: An open-source Linux virtual network computing (VNC) library enabled the customer to remotely control and monitor this custom user interface with little effort on our part.

The fourth feature was a cell modem. A customer wanted to provide Internet connectivity in remote sites that did not have Internet service. An open-source Linux PPP daemon was easily configured to talk to a large number of cell modems and networks.

A fifth feature was proxy settings. Networking is a big reason we choose Linux. Any time a system requires flexible, secure, and expandable Internet connectivity, Linux is our choice. With Linux, we get everything we need (and usually more) for safe and secure networking. We had fully deployed a system when, late in the game, our customer discovered that they needed proxy capability for some of their customers. With Linux, the answer to our customer was: "No problem." It's built in. We didn't think of it—but we didn't need to.

Without question, it costs us less time to add these features (as well as many others) to our customers' products because we chose Linux as our RTOS.

If we have these needs, why choose Linux? Certainly the concurrency requirements can be met with a wide range of available RTOSes. One of the hottest items of late is the FreeRTOS project, which offers an RTOS ported for more than 27 architectures. If you require concurrency but not complex generic devices or complex generic functionality, then Linux is probably more complex than you need.

A LINUX RTOS?

Before we leave the topic of concurrency, I need to address the real-time nature of the operating system. I suspect that some of you may be biting your tongue when I describe Linux as a real-time operating system. If by that you mean an OS that must meet hard deadlines in the sub-millisecond range that can never miss a deadline without extreme consequences, then you are correct. Various successful attempts have been made to add real-time extensions to the Linux kernel. But out of the box, Linux does not meet that strict requirement. Basically, the two terms of interest are deterministic scheduling and low and deterministic latency. If there is interest, I can talk further about this in another column.

An example of a simple hard deadline real-time requirement is a generic SPI or I²C slave device (assuming that your processor doesn't have built in support for it). In the latest Linux kernel, you will be hard pressed to find a driver to support this seemingly simple interface. The reason is that Linux can't meet the timing requirements the host would demand of such a device. Although the master can slow the interface down, the device is a

slave to the master's clock. If you don't control the master, your driver will sometimes fail to deliver.

We get around this in our designs by putting simple microcontrollers (less than $0.50) in front of Linux to meet any hard deadlines. It's not worth it to try to bend the OS and to use real-time extensions to make it meet such rigid demands. For the most part, doing it in hardware is inexpensive and, more important, easily verifiable.

That said, we had a requirement for Linux to process interrupts with a latency of less than 10 to 20 µs. Our initial design had a PIC microcontroller in front to handle this. However, since the requirement wasn't "hard" and we could live with a retry, missing one in every 200 of these events wasn't a big deal. So, in production, we eliminated the PIC front end and are handling it just fine in Linux—most of the time. This is why I call Linux a real-time operating system. You can design your application and your drivers with very low latency. Determinism is the problem.

So, in that sense, Linux can handle some pretty rigorous real-time requirements. In addition, the Linux community has done a lot in the last five years to add real-time features into the kernel. But interestingly enough, we have not had a need to utilize most of these features as of yet.

AVAILABLE OPTIONS

If you need concurrency, you've settled the hard deadline issues (either from a specification point of view or by hardware), and your system has one or more complex drivers or generic functions, what are your options? One solution is to roll your own. Certainly this is an option, but I cannot think of one reason to justify it. A second solution is to use a commercial off-the-shelf (COTS) OS. Certainly many of the fine COTS OSes offer a plethora of complex device drivers and complex generic functionality as part of their offering. We have used QNX, Microsoft Windows CE, Wind River VxWorks, Green Hills Integrity, and Mentor Graphics Nucleus on a number of systems, and all offer excellent tools, generic drivers, and much generic functionality. The cost (both recurring and nonrecurring) for some of these is staggering. A third solution is to use Linux. (We'll talk about some of the flavors next time.) The compelling factor for us to choose Linux was not entry cost. There were two driving factors: Availability of complex drivers and complex generic functionality we never dreamed of and availability of the source code when things don't work.

DRIVERS & FUNCTIONALITY

Our customers and their customers are constantly demanding new features. New features require new devices and new devices require new device drivers. It is my thin slice that new Linux drivers for new devices and new functionality are available more quickly than from any other option other than Windows (drivers and functionality for Windows CE is another question). When my customer demands SSH instead of telnet and my RTOS vendor only offers telnet, what are my options? I don't have enough pull to make them change. Do I port SSH to the COTS OS? Do I lose the customer? None of these options are pretty.

Furthermore, if the device requires me to write the driver, Linux again wins hands down in terms of ease of development. Although I would say the documentation is harder to find, there is so much released source code that I can usually pattern my new driver after something similar without completely reinventing the wheel.

> "My thin slice tells me that Linux offers more drivers for more devices and more complex generic functionality than any OS on the planet with the exception of Windows."

Without question, my thin slice tells me that Linux offers more drivers for more devices and more complex generic functionality than any OS on the planet with the exception of Windows. Here is where the issue of entry cost is not the primary concern. How much is it worth to be able to quickly and inexpensively add these features and remain competitive? This is where the real cost savings of Linux works in our environment.

SOURCE CODE AVAILABILITY

Finally, support provided by the COTS is great theoretically. But we have uncovered bugs in four of the five previously mentioned players and gotten stuck. Not one of them was ever fixed by the supplier. These are well-designed OSes, so when we uncover a bug it is usually very obscure and random. How can we get them to duplicate it? They need our hardware and our application. Many times we can't even prove that it's in their code and not ours. Without the source code, we're just plain stuck. We fully understand why, in their business model, they cannot release the source code. But for this reason alone, we choose Linux for the kinds of systems we build.

A THIN SLICE OF LINUX

When our projects require concurrency and flexible and extensible functionality, my thin slice of reality says Linux every time. I would love to hear from you. What has been your experience with embedded Linux or with the other major players?

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

RESOURCE

The FreeRTOS Project, www.FreeRTOS.org.

Unlock the potential of ARM-Powered® designs.

Optimize power consumption, leverage advanced process technology, and gain efficiency in your next embedded design.



200+ speakers, 150+ technical sessions, 100+ ARM IP experts, a single architecture—join the community defining the future.

ARM® TechCon™ 2011 is your "can't miss" opportunity to unlock the proverbial Rubik's cube of design challenges.

- **Immerse** yourself in the latest technologies for chip design utilizing advanced manufacturing technology.
- **Connect** with experts and learn industry-leading strategies and methodologies for optimizing your ARM IP-based design.
- **Learn** proven techniques for accelerating your design with leading operating systems and software.

Advance your knowledge of the best-in-class techniques for designing with the leading embedded microprocessor architecture—ARM. It's simply a smarter way to design everything from sensors to servers. **The design possibilities are endless, but the seats are not. Go to ARMtechcon.com and register today.**

Chip Design Day
October 25

Software & Systems Design Days
October 26-27

The Santa Clara Convention Center
Santa Clara, CA

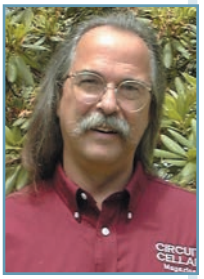
2011 ARM® TechCon™

UBM Electronics

ARMtechcon.com

Join the community defining the future

by Jeff Bachiochi (USA)



Get Your Head Out of the Cloud

An API for Communication with a Wi-Fi Thermostat

The Utility Smart Network Access Port (USNAP) interface, which connects products into home networks, is gaining footing in the marketplace. Here you learn how to build applications to communicate with a Wi-Fi USNAP thermostat using an application programming interface (API).

It's been more than six months since I introduced the USNAP concept and the Wi-Fi-enabled thermostat, the first product to make use of this interface. You might recall that Utility Smart Network Access Port (USNAP) is a SPI-type interface that enables any product to connect into the home automation network (HAN) of today or tomorrow. The use of this interface offers a potential plethora of standard USNAP communication modules. To date Wi-Fi, ZigBee, and Z-Wire have modules available, but just about any medium—wired or wireless—could enter the fold.

The USNAP ball is beginning to roll, but it hasn't taken the world by storm. It has made manufacturers sit up and take notice. We're beginning to see advertisements for smart washing machines that can report their status to a service department. Developed initially for automated metering infrastructure (AMI), the USNAP Alliance's idea offers benefits to all products.

In my initial column, "Smart Network Access Point" (*Circuit Cellar* 246, 2011), I showed how Intwine Energy's mail-order thermostat used a USNAP Wi-Fi module to enable monitoring of the thermostat over the Internet. At the time, Wi-Fi communication was closed, and while monitoring via the web was free, if you wanted to control the thermostat, you were required to pay a $15 annual fee. Today, you can buy a Filtrete Wi-Fi thermostat at Home Depot for $100. Same thermostat, updated Wi-Fi. The new USNAP module uses wireless provisioning (setup) and, as of March 2011, Radio Thermostat Company of America (RTCOA) has opened up monitoring and control via an API for their CT80 large-display touchscreen thermostats, CT30 programmable thermostats, and Filtrete 3M-50 thermostats.

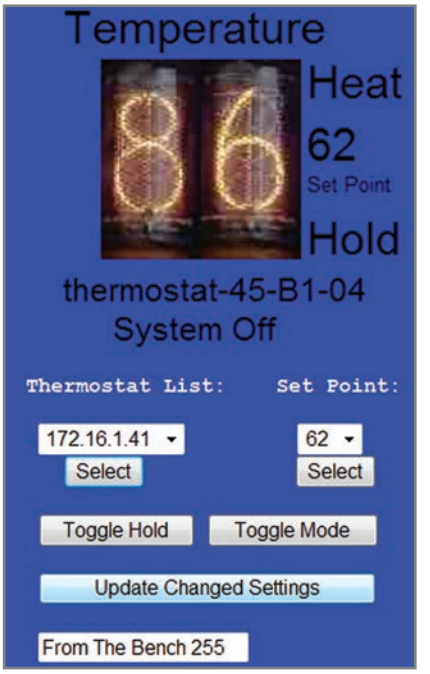


Photo 1—This simplified single-page application is all that is necessary to view the present temperature and change the set-point, if I wish, from anywhere that has Internet access—including my smartphone. Can you say "personal cloud"?

This is an exciting move that opens up the lines of communication for you and me via our existing Wi-Fi network. The API is intended for personal use and has a commercial license available to those who wish to

Data Type	Value	Key/Value Pairs
Number (floating point)	10.7	{"Diameter" : 10.7}
String (double quoted)	"Inches"	{"Diameter Units" : "Inches"}
Boolean (true/false)	TRUE	{"Switch" : true}
Object(s) (curly bracketed key/value pair or comma-separated pairs)	{ "Diameter" : 10.7, "Diameter Units" : "Inches," "Freewheeling" : true }	{"Wheel" : { "Diameter" : 10.7, "Diameter Units" : "Inches", " Freewheeling" : true }}
Array (square bracketed ordered sequence of values, comma separated)	[2248, "Main Street", 47, "Robinson Circle"]	{"Street Addresses" : [2248, "Main Street", 47, "Robinson Circle"]}
Null	Null	{"ID" : null}

Table 1—The list of value types for use in key/value pairs of JavaScript Object Notation (JSON), a human-readable data exchange format common in client/server web applications. Key/value pairs are passed using the HTML protocol methods Get, Put, Post, and Delete.

explore that avenue. The first round of documentation for API V. 1.0 is fairly complete and offers enough information to get you started talking with your thermostat.

USNAP WI-FI

Provisioning is the term used to set the Wi-Fi module's initial connection to your home wireless network. This is done through a webpage served by the thermostat (see Photo 1). When initially powered up, the thermostat will create its own ad hoc network. You can locate and connect to it through "Connect to a Network" (part of the network and sharing center). Once connected, enter the IP address 192.168.10.1 in a browser and you get a webpage served by the thermostat. Select your home network from those available, you will get an advanced webpage where you can enter pertinent information about your wireless router (e.g., password). Next, the webpage will ask for the PIN being displayed on the thermostat. This step assures that someone can't try to hack in without access to the thermostat's screen. After clicking the Set Network button, you're finished. The thermostat now has access to your home network and it will display the last octet value of its IP address on its LCD screen for you until you make contact. If you now enter this value as the last octet of your home network's IP address into the browser, you will receive the initial webpage served by thermostat again. This time, it displays its full IP address and the service set identifier (SSID) of your home network.

Since a USNAP module doesn't have any idea what it may be connected to, we can't expect to have it serve "thermostat" webpages. It can only serve what it knows and that is data about its present Wi-Fi connection. Upon "Connect to the Network" it can present its configuration to your browser and, by gathering up your input, it can remap itself to your network, open to communication via various protocols. One of the most widely used protocols is HTTP. We've already seen its usefulness in displaying its configuration and gathering your input. This user-friendly provisioning is a cool way to collect data that was typed into the fields on the display and transferred to the server using an HTTP Post method to send data from you (the client) to the server. On the server, the data is directed to the appropriate resource and updated with the received data. This new data is sent to the client the next time it is requested using the Get method.

This is all part of a REpresentational State Transfer (REST) architecture that is based around resources. The uniform resource identifier (URI) you type into a browser (i.e., http://www.circuitcellar.com/magazine) has a number of parts: the scheme (HTTP), the host (www.circuitcellar.com), the path (/magazine), and the optional query and fragment. Except for the fragment, the URI contains all the routing necessary to get to a specific resource. A resource may consist of HTML, plain text, an image, etc. This resource can be acted upon in one of four ways: Get, Put, Post, or Delete (a subset of the HTTP protocol). You can think of these as retrieval, create or total replacement, create or partial replacement, and elimination. Post is much more powerful, but I think this definition will do for this column. If you like a dose of humor with your education, visit this site, http://tomayko.com/writings/rest-to-my-wife, for a simple discussion of REST.

So, while a user-friendly webpage makes it easy for a user to enter and transfer data to a server, it is not necessary. URI resources can be replaced or retrieved using the HTML Post and Get methods without ever displaying a webpage. This enables a client to communicate with a server directly, provided the client knows where and what the resource is.

The RTCOA USNAP Wi-Fi module uses these methods for transferring data to and from clients. The module's resources are in the JavaScript Object Notation (JSON) format. This is a text-based open standard for human-readable data exchange using key (string)/value pairs. Table 1 shows the six data types associated with the value: number, string, Boolean, object, array, and null.

The API describes these resources and where they can be found. We can begin with the resources found in the system folder at "http://<IPAddress/sys." To query the system resources, we use this URI with the Get method. If the resource can be found at the URI it will return the "sys" object, in this case a list of objects including the UUID, api-version, fw_version, and wlan_fw_version. In my case, that's:

```
{"uuid":"5cdad445b104","api_version":113,"fw_ve
rsion": "1.04.64","wlan_fw_version":"
v10.105576"}
```

There are many ways you could code this in an application. As a matter of fact, there is already a nice PC

application available for this Wi-Fi thermostat that uses the API: www.setyourthermostat.com. I chose to use Run BASIC for my application because I didn't just want access through my PC, but from anywhere with Internet access. With Run BASIC I can quickly write web server applications I can access via the Internet. Run BASIC supports *httpget* and *httpost* commands that will give me access to any Wi-Fi thermostat I have using the RTCOA API. The format for the commands is:

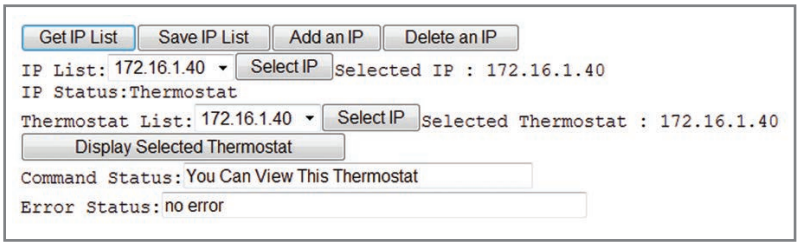


Photo 2—The first application is to help define thermostat IP addresses on your local LAN/WLAN. Once identified, each thermostat can be individually accessed for a number of resources.

```
ReturnString$=httpget$(url$)

ReturnString $=httppost$(url$,object$)
```

where `url$` is a string holding the URI resource we would like to access and `object$` is a string of the object we wish to have access to. A "good" resource will always return something. It might contain an object or just a reply indicating success or failure.

Notice in the reply from the "sys" resource that a lot of string manipulation must be done to pull apart the response and make use of the data received. The first character will always be a "`{`" followed by quotes, since the key in each object pair is always a string. At the end of the key will be a second set of quotes followed by a "`:`". You can use the next character to determine what type of value is associated with the key. If it's a quote, the value is a string. If it's a number, the value is a number. An ASC "`t`" or "`f`" suggests a Boolean value. A "`{`" indicates an object. And a "`[`" indicates an array, with ASC "`n`" interpreted as null. Other than objects or arrays (which are essentially just lists), the value used for objects in API V. 1.0 is either a number or a string.

URI Resource http://IPAdress+	Get Object		Post Object	
/sys	List	uuid, api_version, fw_version, wlan_fw_version		
/sys/services	Array	Service_names		
	List	http_handlers		
/sys/network	List	ssid, bssid, channel, security, ip, ipaddr, pmask. ipgw, ipdns1, ipdns2, rssi	List	ssid, bssid, channel, security, ip, ipaddr, pmask, ipgw, ipdns1, ipdns2, rssi
/sys/updater				
/sys/filesystem				
/sys/firmware				
/sys/fs-image				
/sys/fw-image				
/sys/command			System Command	"reboot"
/sys/mode		mode		mode
/sys/name		name		name
/cloud	List	interval, url, status, enabled, authorization_key	List	interval, url, status, enabled, authorization_key
/tstat	List	temp, tmode, override, hold, t_heat, t_cool, a_heat, a_cool, t_type_post, tstate, fstate, time (time is an array = day, hour, minute)	List	tmode, hold, t_heat, t_cool, a_heat, a_cool, a_mode, time (time is an array = day, hour, minute)
/tstat/program/heat	List		List	
/tstat/program/cool	List		List	
/tstat/model		model		
/tstat/led				energy_led
/tstat/pma	List	line, message	List	line, message
/tstat/uma	List	line, message	List	line, message

Table 2—This table shows the resources for RTCOA Wi-Fi API V. 1.0. The Get method returns active resource Objects. The Post method can include any or all resource objects.

Table 2 shows all the resources available to a user in API V. 1.0. Some resources include a list of objects (i.e., "tstat"). Note that one of tstat objects, "time," is in fact a list of the objects: "day," "hour," and "minute." The resource tstat/program/heat (and cool) is a list of day objects. Each day is an array of temperature settings and minutes since midnight. Each array can have up to seven (four for the CM-30) temperature and time array pairs.

The thermostat has four modes: Off, Heat, Cool, and Auto. The Off mode prevents the thermostat from requesting any heating or cooling. In the Heat mode, the thermostat will request heat from the furnace when the temperature falls below the setpoint (by x degrees, where x = 0.5° to 2°). The request for heat remains until the temperature rises above the setpoint (by x degrees). In the Cool mode, the thermostat requests air conditioning if the temperature rises above the setpoint (by x degrees) and continues until the temperature falls below the setpoint (by x degrees). The Auto mode enables the thermostat to choose between heating and cooling depending on the temperature. Spring and fall days could actually require both, heating for a chilly overnight and cooling for a sunny day. Daily scheduling enables the setpoint to be changed at specific times during the day. For instance, you may want a winter schedule that turns

Audio amateur

Build the speaker of dreams for your:

Home, Car, or Home Theater!

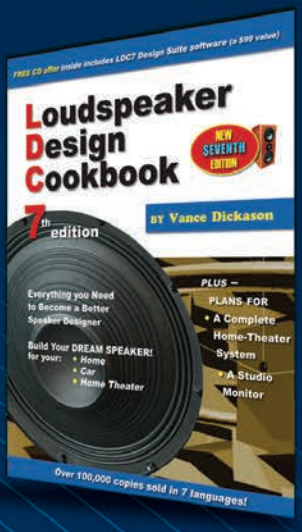


Back and better than ever, this 7th edition provided everything you need to become a better speaker designer. This newly revised edition now includes Klippel analysis of drivers, a chapter on loudspeaker voicing, advice on testing and crossover changes, and so much more! Ships complete with bonus CD containing over 100 additional figures and a full set of loudspeaker design tools.

A $99 value!

Yours today for just $39.95.

Shop for this book, and many other Audio Amateur products, at **www.cc-webshop.com.**



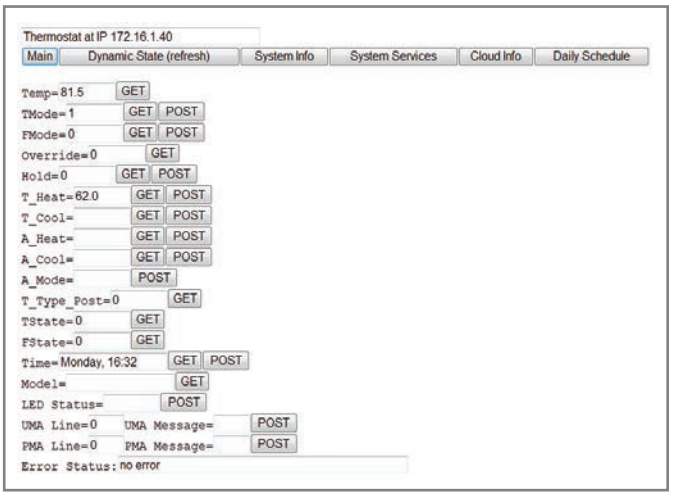


Photo 3—These are the "tstat" resources available for each thermostat. Note: Resources included in a response are shown as having values.

down the heat while you sleep, warms up before you rise, cools after you leave for work, and then rises again before you return home. Not only will a programmable thermostat do that for you, but with Wi-Fi capability, you can override the schedule from your smartphone or office if you plan to return home early or will be away for the weekend.

The present API does not enable you to do some things like change the swing (the x above) for the setpoint, but it will enable more than you may need or want to have control of (refer to Table 2).

I started with an informational application just to play around with the API. If you've gone through the provisioning to put Wi-Fi thermostats on your network, then you already know the IP addresses for those thermostats. It takes a lot of time to search the 255 possible addresses (last octet of your IP). You can use the command line "`ARP -a`" to get a list of all the IPs that were automatically assigned via DHCP. You can use the command line "`PING IPAddress`" to check that a particular IP address responds. To determine if there is a thermostat at an IP address, you can use httpget (`URL$`) from your application and then look for a legal response.

Since I knew my thermostat's IPs, I created a text file that simply listed these. The IPList application enables the user to read the IPList textfile, write an IPList textfile, and enter and delete IP Addresses (see Photo 2). When an IP is selected, a Test IP button appears that will ping the IP to see if it is responding. If it responds, a Test for Thermostat button appears that will test for a thermostat. If it responds with resource data, then the IP is added to an array that holds confirmed thermostat IPs. Once there are verified thermostats, any selected thermostat can be viewed.

The first resource I wanted to display was "tstat" because it provides the temperature and will allow you to adjust the setpoint. The response string is pulled apart using the `instr$()` function to search for objects that are part of this resource. The thermostat returns those objects that are pertinent to the present mode of operation. Photo 3 shows the screen display of all of the attributes returned by the "tstat" resource. Note back in Table 2 that the "tstat" resource folder has a few subfolders that hold other resources. I placed Get and/or Post buttons next to objects that can be individually monitored and/or updated.

At the top of the "tstat" display (and others) is a row of buttons that will take you to the other resources available. These other displays are full of information the API has access to. The reason these thermostats are so popular (and green) is that they can be programmed to adjust the setpoint numerous times throughout the day providing a level of comfort when you are there and saving energy when you're away. For this fact, the Daily Schedule display may be of interest. There is a heating and a cooling schedule for each day of the week. I use a drop-down list of time/temperature setpoints for each day to conserve display space (see Photo 4).

In reality, there are only two functions I am interested in, the present temperature and the ability to change that, if desired. I have three zones in my house, the main floor and two upstairs areas. This gives separate control over the living area, the sleeping area, and the work area. While setting up the daily schedules can be handled remotely, once set, I don't believe these will be altered on a regular basis. Instead, I will be using a remote connection to set temporary overrides on those occasions that I am away unexpectedly or may be returning home early and wish to adjust the temperature appropriately. This application is a bit of overkill for that. Let's tone it down a bit and write one with fewer functions and complexity.

NO FRILLS

Using only the "tstat" resource, provides you with a bunch of useful information. The second application is a simple single webpage that will enable you to select available thermostats (seeded by the imported text file) and

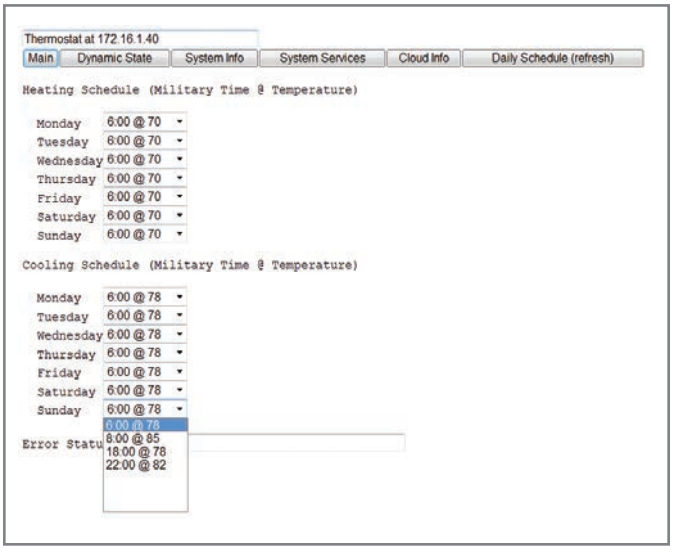


Photo 4—While I believe daily schedules are not something I want to adjust from afar, I've included them in this application, which is basically a test program for using the API.

monitor your temperature, setpoint, and operating mode. You can change modes, adjust setpoint, and apply a hold to any thermostat. Information is presented in a graphic area, while user controls are simple button and pull-down list boxes. I wanted to use something a little different for displaying the temperature, so I created 10 image files. While the image files could be anything that represented the digits 0–9, I chose to use "Nixie" tubes for a slightly nostalgic feeling (see Photo 1).

The total listing for this application is four pages, approximately 250 lines of BASIC text code. The breakdown is 15% initialization, 40% information display, 15% user input, 25% retrieve data, and 5% send data. I think you'll agree that this part of the project was not an intensive one. The learning curve on the API and application to exercise it was much more intense. You can access the thermostat application on my Run BASIC server at jlbachiochi.myvnc.com. Just choose the FTB255 button for this application. For more information on the virtual IP I use, check out my column, "Serving Up BASIC," *Circuit Cellar* 247, 2011.

While what I was hoping for was an API for a generic Wi-Fi USNAP interface, what we have is a Wi-Fi USNAP thermostat interface. It's understandable that since the USNAP Wi-Fi developer is also the thermostat developer, the immediate interest is in boosting the sale of thermostats and not necessarily supporting the complete USNAP specifications. My hope is that this great USNAP Wi-Fi module will eventually support sending and receiving commands as outlined in the USNAP serial interface specification so manufacturers will have a truly universal way of enabling their products to get connected.

Author's note: A special thanks to Dan Goodman and RTCOA for their gift of this API to the world. Their vision of the future is not to look at others as competition, but as cobuilders of a much better world. Without companies like RTCOA, we would all be reinventing the wheel instead of mastering it.

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

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/255.

RESOURCES

J. Bachiochi, "Smart Network Access Point,"*Circuit Cellar* 246, 2011.

———, "Serving Up BASIC" *Circuit Cellar* 247, 2011.

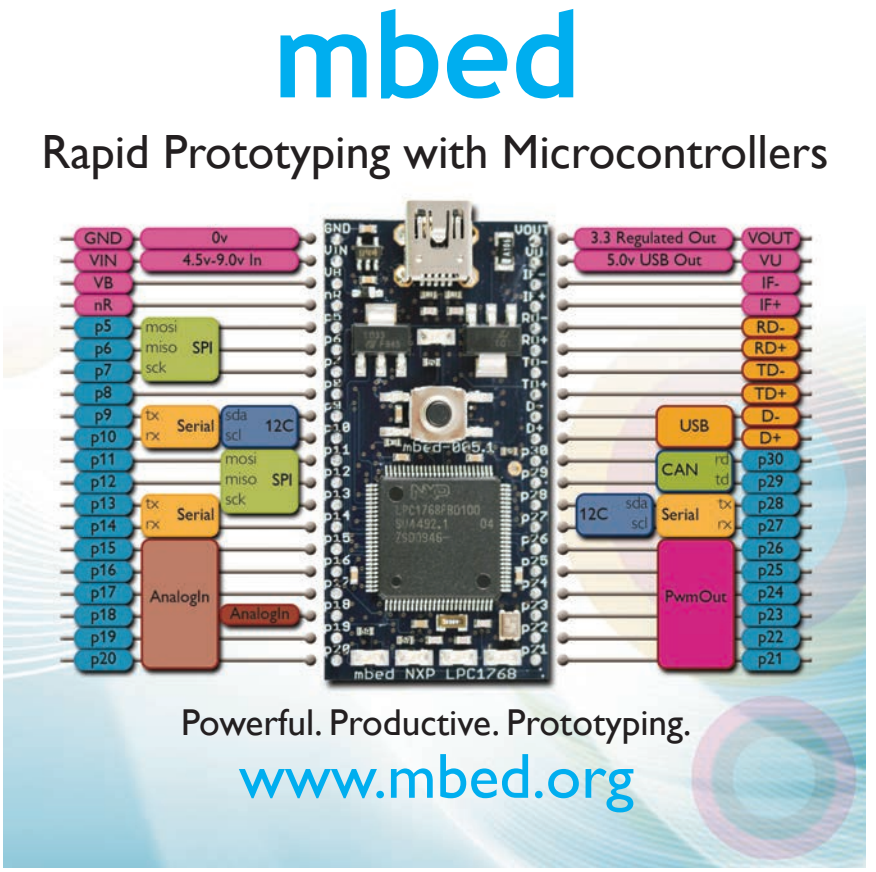
Radio Thermostat Company of America (RTCOA), "Wi-Fi USNAP Module API, Version 1.0," 2011, www.radiothermostat.com/latestnews.html#advanced.

Run BASIC, www.runbasic.com.

The USNAP Alliance, www.usnap.org.

SOURCES

CT80, CT30, and Filtrete 3M-50 Thermostats
Radio Thermostat Company of America (RTCOA) | www.radiothermostat.com



by Robert Lacoste (France)



Line-Coding Techniques

The nonreturn-to-zero (NRZ) method is a binary code in which ones are generally represented with a positive voltage and zeros are generally represented by a negative voltage leaving no null condition. This article details the NRZ method and other line-coding techniques—NRZI, Manchester, Miller, AMO, MLT3, and 4B5B—that can be used in wireless and wired protocols.

Welcome back to the darker side. As you know, data transmission over a single pair of wires goes back to the 1840s, when Morse code was invented. By the way, there is still a controversy about its actual inventor: it is either Samuel F. B. Morse himself or its co-inventor Alfred Vail. We know that these two guys were sitting at both ends of the wire for the first demonstration, but versions vary on who did what. Well, let's avoid the controversy and stay on the technical side. Morse code uses what is now called pulse-length modulation, as the information is encoded using different pulse lengths (dee-dah-dees). This was the easiest solution for human operators, but fixed-length pulses are easier to manage for automated systems.

The first fixed-length code is the 5-bit Baudot system, invented by French engineer Jean-Maurice Emile Baudot in 1874. Baudot invented the first time-division multiplexing telegraph system, with up to five communications sharing the same line. A great improvement at the time! The "baud" unit (one state change per second) still used today is derived from his name. This early coding system evolved up to the first universal asynchronous receiver/transmitter (UART) developed by Gordon Bell

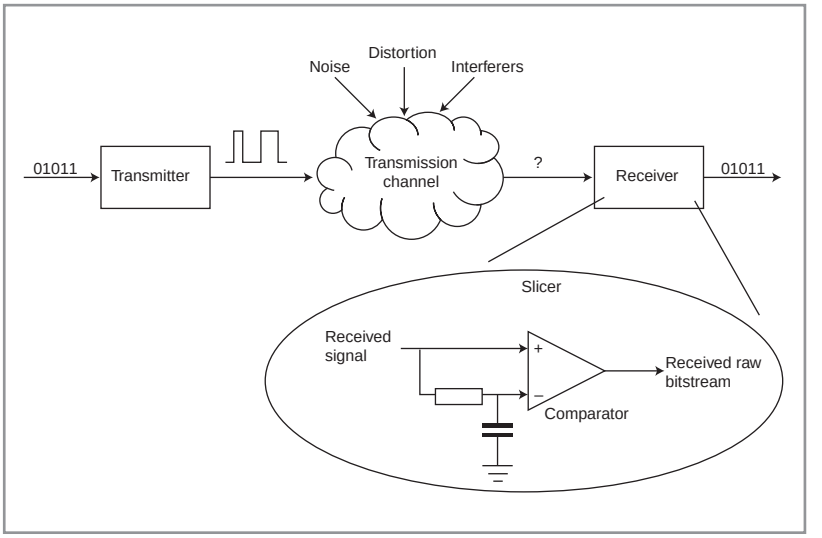


Figure 1—A bit slicer is at the heart of virtually any digital receiver. This module must decide if the incoming bit is a zero or a one, usually comparing a given analog signal with a reference. In order to minimize noise effects, this threshold is defined as the average of the signal over time. This explains why a non-DC-balanced signal may cause a problem.

in the early 1960s for the PDP computer. This was a great idea, as UARTs are still found on almost every microcontroller.

Transferring a binary datastream serially is an efficient solution, as it minimizes the wiring hassle. Let's focus on how to encode zeros and ones on such a wire. Standard UART protocols, like the ubiquitous RS-232, use a straightforward encoding scheme. In RS-232, bits are transmitted serially with a given fixed voltage for zeros and ones. More precisely, for RS-232 any voltage from 3 to 25 V means zero and –3 to –25 V means one. This method is called NRZ, which means "nonreturn-to-zero" as the voltage on the line is almost never null. Simple, isn't it? Well, yes, but this solution must be used with caution. This month I will dig into that subject and explain why plenty of other line-coding methods—such as NRZI, Manchester, Miller, AMO, MLT3, or 4B5B—are needed and effectively used in wireless and wired protocols like Ethernet.

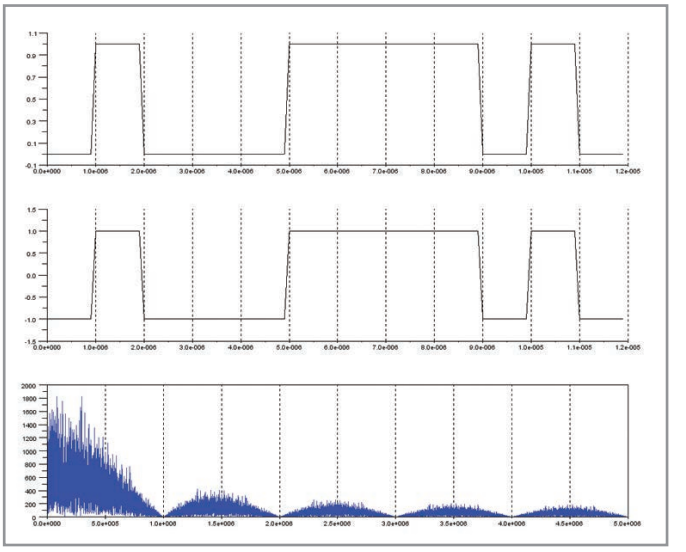


Figure 2—The most straightforward line-coding method is NRZ, which is basically no coding at all. A given voltage corresponds to zeros; the opposite voltage corresponds to ones. Unfortunately, the frequency spectrum of such a signal (bottom graph) has plenty of components from DC up to very high frequencies.

BIT SLICING

Life is boring in a perfect world. When a signal is transmitted on a wire, or on a wireless link, it is affected by noise and all kinds of distortions (see Figure 1). The goal of the receiving equipment is to do its best to recover the transmitted bits out of this garbage. I covered the topic of error-correcting codes in 2010, ("Living with Errors: An Introduction to Forward Error Correction," *Circuit Cellar* 235). But at the physical level, any receiver must, at a given stage, transform the incoming analog signal into a logical one or zero. This step is called bit slicing, and it is far from obvious.

Perhaps the best example is a wireless data link using frequency-shift keying modulation (FSK). In such a system, the transmitted carrier frequency is shifted down or up by a fixed amount depending on the logical level to transmit. However, the carrier frequency is never 100%

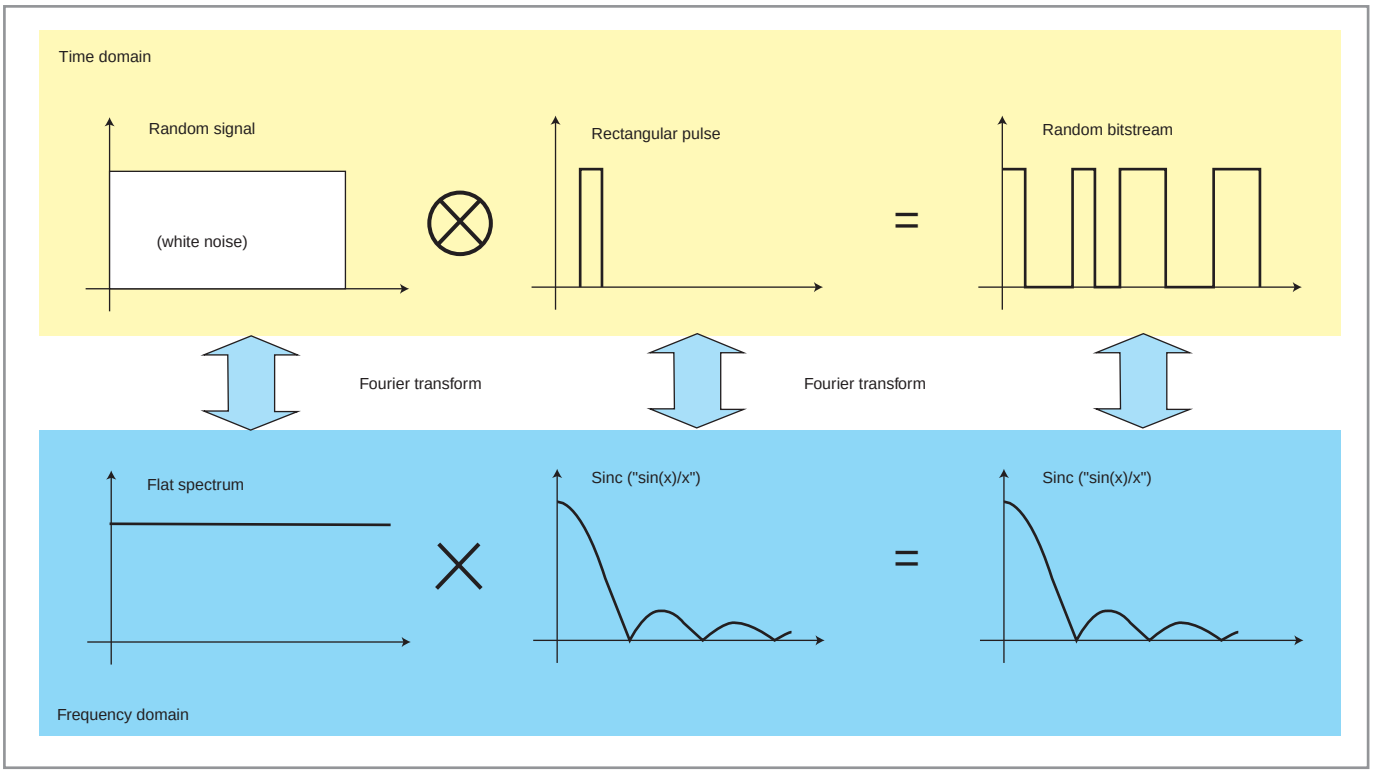


Figure 3—Without going into the mathematical concepts, you can understand the overall shape of the NRZ signal's frequency spectrum when you know that the Fourier transform of a single digital pulse has a sin(x)/x shape. Compare this curve to Figure 2.

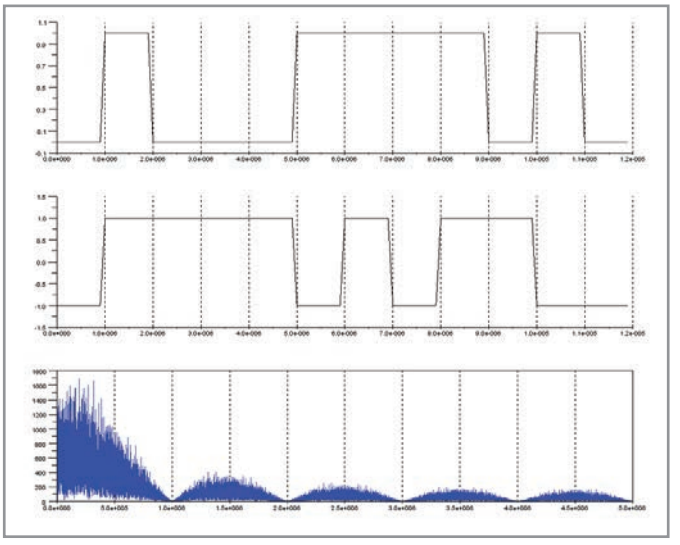


Figure 4—NRZI is the differential version of NRZ. Here the output signal (middle) changes state wherever the input bit (top) is a one and stays constant otherwise. The spectrum is the same as basic NRZ.

accurate. For instance, it could shift if the transmitter's crystal is heated by the sun or if one of the nodes is moving fast thanks to the Doppler shift. So, receivers must decide whether the bit was a zero or one by comparing the received frequency with their best estimation of the central frequency. This estimation is usually an average of the received frequency over time. The bit slicer is then something like a comparator between the input signal and an average of this signal over time, as seen in Figure 1. Such a circuit is, in fact, a high-pass filter. Any DC drift of the signal is nullified as the signal is compared to its average. The bit detection is done based on fast variations of the signal level around its average.

This situation is not specific to wireless systems. For example, all Ethernet links use transformer-based couplers, which block any DC voltage. Bit-level decision for Ethernet interfaces could then only be done based on AC voltages, which means on relative changes of the signal level around its average value. Sounds similar, no?

NRZ ISSUES

What kind of trouble will we run into if we use NRZ coding for our FSK example or for Ethernet? Remember that NRZ uses a given voltage for ones and an opposite voltage for zeros. The average value of the signal will be zero, but only if there are exactly as many ones as zeros in the transmitted bitstream. Unfortunately, this has a low probability: if you take a finite signal—say, 20 random bits—then the probability of having a non-null average is far higher than the probability of a perfect balance. If you are not convinced, toss a coin into the air 20 times and try to get exactly 10 heads and 10 tails. Just send me the coin if you fail, and I will be rich.

There is another way to see this phenomenon: by switching from time domain to frequency domain. I coded a small NRZ simulation for you using the free Scilab software. (The example code is available on *Circuit Cellar*'s FTP site.) Basically, I simply generated 4,000 random bits, converted them to –1 and +1 levels for zeros and ones, respectively, and calculated the frequency spectrum of this signal using a fast Fourier transform (FFT). The result is shown in Figure 2. You will immediately see that this frequency spectrum is not null at 0 Hz. This means the NRZ signal has very low frequency components, including DC, even if the bitstream is quite long and if the bits have equal probability to be zero or one. Why? Because, in any random-bit sequence there are arbitrarily long strings of zeros and long strings of ones. These long sequences translate, in the frequency domain, into a power spectrum going down to DC. The problem is that these low frequencies will be difficult for the bit slicer. Remember that bit slicing is usually done by comparing the signal voltage to its "long-term" average. If there are low frequencies, or even a DC offset in the signal, this long-term average won't be stable, introducing jitter in the output bitstream and finally bit errors. As an extreme example, imagine you have a long sequence of zeros. How would you be sure the bit is still actually a zero if there is a strong DC offset on the line?

I will present some alternative coding methods soon, but take some time to look again at the frequency spectrum of the NRZ signal illustrated in Figure 2. Doesn't its shape look familiar? If you are a regular reader of this column, it may remind you of the shape of the sin(x)/x function in my 2008 column, "Direct Digital Synthesis 101" *(Circuit Cellar* 217). And there is a good reason, as shown in Figure 3. In a nutshell, this sin(x)/x shape is the characteristic of any fixed-duration pulse signal, as it is the frequency spectrum of a single pulse.

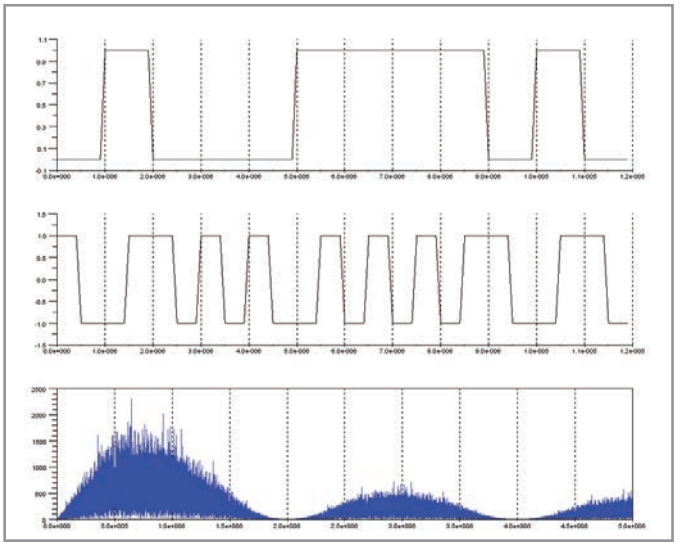


Figure 5—An example of Manchester encoding. Each bit is encoded as two successive bits, either 01 or 10, depending on the input bit. The spectrum shows that there are no longer any DC or low-frequency components; however, higher frequencies are used as the data rate is, in fact, doubled.

Elektor Print

Classical reading: on paper



+

Elektor Digital

Contemporary & innovative reading: on PC, notebook or tablet

=

Elektor PLUS

Comprehensive reading: at home and on the road

Read Elektor with the premium PLUS subscription!

Subscribe now or upgrade: www.elektor.com/usa

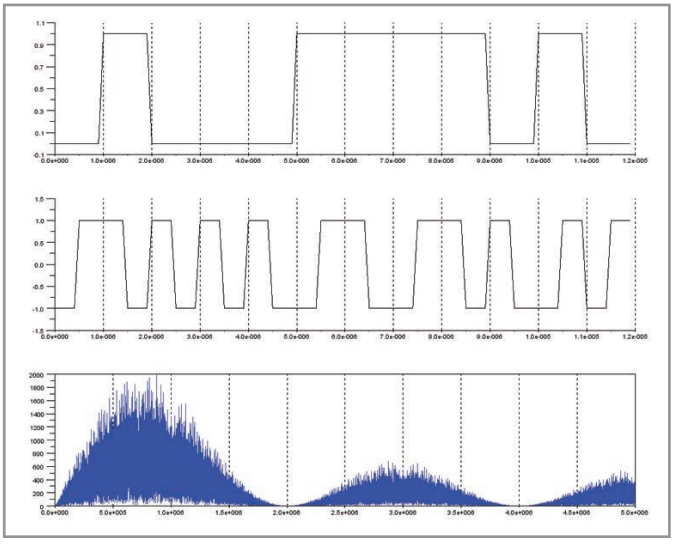


Figure 6—Differential Manchester encoding has an advantage over raw Manchester: independence over signal polarity.

ALTERNATE SIMPLE LINE-CODING METHODS

Let's try to find better coding methods than NRZ. The issues to be solved are first to help the bit slicer work correctly even if there are long sequences of zeros or ones, but also to "more efficiently" use the transmission medium. Have another look at Figure 2. The used frequency spectrum ranges from DC to very high frequencies, this means it would be difficult to use a low bandwidth connection. This also means EMC problems will be a concern. If you can use a coding method that reduces the used frequency spectrum for the same actual bit rate, this will enable you to either increase the bit rate or to use cheaper wires. This method is called spectrum shaping.

The first alternate coding method you may find is NRZI (see Figure 4). The "I" in NRZI stands for "inverted." This is, in fact, exactly the same method as NRZ, but with a subtle change. Here the transmitted level changes (from low to high or from high to low) when the bit to be transmitted is a one, and stays constant if it is a zero.

As shown in Figure 4, the spectrum of an NRZI signal is nearly identical to the basic NRZ, so there is no spectrum shaping here. The NRZI method has two significant advantages over NRZ. First, the coding is not dependent on the signal polarity: NRZI is a differential code. So, if you change the sign of the input (i.e., swap the two wires) then the decoding gives the same bitstream. Only the presence of a transition is important, not the polarity of the signal. Second, it avoids long strings of ones, helping clock recovery for the receiver. But it doesn't help in case of long sequences of zeros. In order to counteract this problem, NRZI is usually used in addition to some kind of bit stuffing or run-length limited code. For example, a variant of NRZI is used for the USB protocol: an additional 0 bit is inserted after six consecutive 1 bits, preventing long strings of zeros and ones.

Another well-known method to avoid any DC imbalance and to drastically improve spectrum shaping is Manchester encoding (see Figure 5). The principle is as follows. Each bit is encoded with a two-state sequence. A one is coded as 01, and a zero is coded as 10. This doubles the data rate for the same actual throughput; but with this simple method, you can be absolutely sure there are as many zeros as ones on the transmitted bitstream. Moreover, there is at least one transition in the middle of each bit, so clock recovery is easy. Take another look at Figure 5, this is also visible on the frequency spectrum as there are no longer any DC or low-frequency components. Manchester code is also used extensively for low-bit-rate wireless links (e.g., remote door openers or RFID) and in 10BASE-T Ethernet.

You may also encounter a variant of Manchester: differential Manchester coding, also called biphase mark code (BMC). Think of it as a mix of NRZI and Manchester. Each input bit is coded on two bits as in Manchester coding, giving two possible transitions per bit. One of the transitions, say, at the middle of the bit time, is used as a clock reference and is always present. The second possible transition, at the start of the bit time, is present only if the input bit is a zero (see Figure 6). The spectrum is the same as pure Manchester. However, as with NRZI, this method is not dependent on the signal polarity. Moreover, the format of the signal, with a clocking transition at the middle of every bit, enables an efficient clock recovery. Differential Manchester is used for example in S/PDIF and IEEE 802.5, also known as Token Ring.

MOVING FORWARD IN PERFORMANCE

Let's move forward. You could get better performances with the same transmission channel by using a more optimal line-coding method. The idea is to transmit the same amount of information but using a smaller spectrum bandwidth. By the way, for those who are familiar

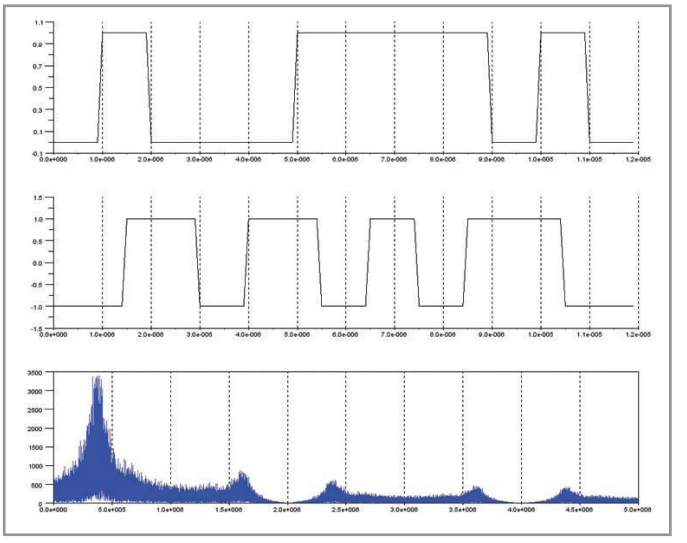


Figure 7—With the Miller coding example shown here, we enter into spectral efficient codes. Look at the frequency spectrum, it is far thinner than the previous examples, isn't it ?

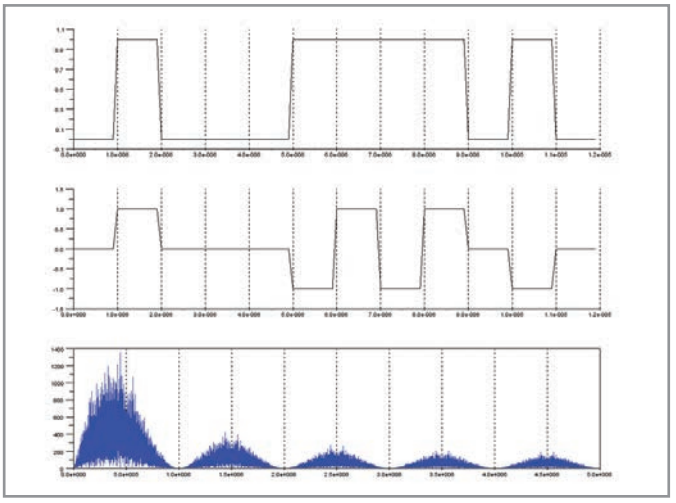


Figure 8—AMI code is an example of ternary code, which uses three different line voltages. Zeros are transmitted as 0 V. Ones are successively transmitted as +1 or –1.

with Shannon's theorem, that will imply a slightly degraded signal-to-noise performance, but an increased capacity. I guess I will need another column to dig into that subject.

One of the solutions is Miller coding, which is a kind of so-called "delay code." It is easy to generate, just start with the Manchester code, and remove one transition every two transitions. This could be simply done by feeding a Manchester signal through a binary divider by two. You will get something like the signal simulated in Figure 7. If you check carefully, you will see the resulting signal obey the following rules: First, there is always a transition in the middle on a "1" bit but never in the middle of a "0" bit. Second, there is a transition at the end of a "0" bit but only if it is followed by another "0." With these rules you will get a signal with a built-in clock, and drastically improved spectral shaping, as shown in Figure 7. Just compare to the original spectrum of the equivalent NRZ signal in Figure 2. The only disadvantage of Miller code, as compared to Manchester, is that there is statistically a small DC offset in the signal. A variant of Miller coding is used in EPC UHF RFID "gen 2" cards.

A very different approach is possible with bipolar Alternate Mark Inversion codes (AMI) which are used in the T1-line telecommunications field. Here the signal is no longer a two-level digital signal, but uses three different line voltages: A positive voltage, a null voltage, and a negative voltage. The idea is that a "0" is always encoded with a null voltage and a "1" is alternatively encoded as a pulse of either positive or negative voltage (see Figure 8). Of course, this allows for a null average voltage, so no DC component, as well as nice pulse shaping. However, the big drawback is a lack of synchronization if the input signal contains only zeros. This was not a problem when T1 lines were transmitting only voice, as there was always

AP CIRCUITS
PCB Fabrication Since 1984

As low as...
$9.95 each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com

VISA MasterCard PayPal IPC MEMBER ASSOCIATION CONNECTING ELECTRONICS INDUSTRIES BBB ACCREDITED BUSINESS

System on Module
Internet Appliance Engine
SoM-9G45
• AT91SAM9G45 ARM9 400Mhz CPU
• 4 Serial Ports & 2 SPIs
• Up to 40 Digital GPIOs
• 2 USB 2.0 Host/Device Ports
• I2S Audio Interface
• 10/100 BaseT Fast Ethernet
• SD/MMC Flash Card Interface
• Up to 1 GB Flash & 256 MB RAM
• Linux with Eclipse IDE & WinCE 6.0
• 8, 10-Bit A/Ds & 4 16-Bit Timer/Counters
• Graphic LCD Interface with 2D Acceleration
• Small, 200 pin SODIMM form factor (2.66 x 2.38")
Windows CE
2.6 KERNEL
The SoM-9G45 uses the same small SODIMM form-factor utilized by other EMAC SoM modules, and is the ideal processor engine for your next design. All of the ARM9 processor core is included on this tiny board including: Touchscreen Interface, Flash, Memory, Serial Ports, Ethernet, I2S Audio Interface, PWMs, Timer/Counters, A/D, Digital I/O lines, and more. Like other modules in EMAC's SoM product line, the SoM-9G45 is designed to plug into a custom or off-the-shelf Carrier board containing all the connectors and any additional I/O components that may be required. The SoM approach provides the flexibility of a fully customized product at a greatly reduced cost. Single unit pricing starts at $190.
http://www.emacinc.com/som/som9g45.htm
Since 1985 OVER 25 YEARS OF SINGLE BOARD SOLUTIONS
EMAC, inc.
EQUIPMENT MONITOR AND CONTROL
Phone: (618) 529-4525 • Fax: (618) 457-0110 • Web: www.emacinc.com

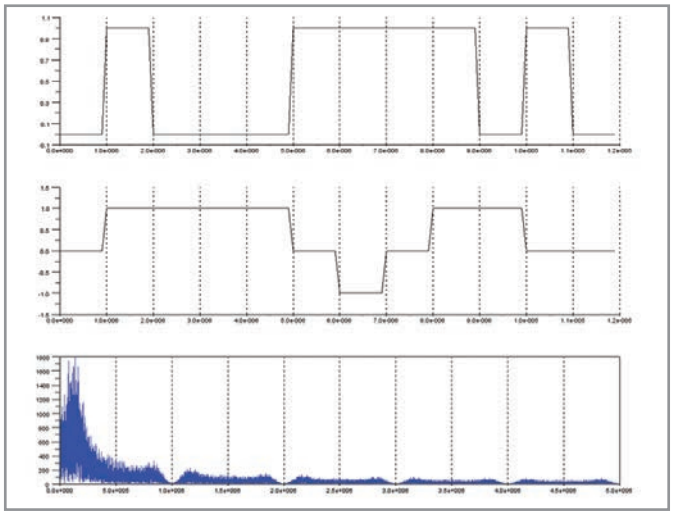


Figure 9—MLT-3 coding provides a drastically reduced spectrum through a clever three-level coding method (e.g., used on 100-Mbps Ethernet).

some noise on the microphone, so non-null bits are the least-significant bits (LSBs). But, for modern digital signals, it is more of a problem. This required forcing bit transitions from time to time to maintain the clocking through the so called, "bipolar violations" (two pulses of the same polarity in a row) and variants like the B8ZS coding scheme.

A variant in the same spirit—but far more efficient on the spectral side—is the so-called MLT-3 code (see Figure 9). MLT means multi-level transmit, as there are three possible voltages on the line (positive, null, and negative), as with AMI. However, the principle is different: If the bit to be transmitted is a zero, then the line state doesn't change. If the bit is a one, the line voltage jumps to the "next state" in a circular fashion: +V, 0, –V, 0, +V, etc. So, the output signal is more or less sine-shaped, which gives very low frequency content, as illustrated in Figure 9. This is due to the very low frequency cycles. As with NRZ, there is, at most, one transition per bit. But the output signal will loop back to the same voltage only after 4 bits at minimum. MLT-3 is used in particular for the 100B-TX Ethernet variant.

Lastly, another way to do spectral shaping is to work not only at the bit level but also at the symbol level, meaning grouping and coding a set of bits together. As an example, you can take 4 input bits and convert them into 5 bits prior to transmission, for example with a simple NRZ or NRZI method. This transcoding, called 4B5B (see Figure 10), ensures that there are transitions in the signal even if the input bits are all zeros or ones. Moreover, thanks to a good choice of the 16 5-bit words (one for each combination of the 4 input bits), the transmitted spectrum can be optimized. For example, 100Base-TX Ethernet uses a 4B5B coding prior to MLT-3 encoding. Longer word variants, like 8B10B, were introduced by IBM in the early 1970s and provide a good DC balance. 8B10B is used everywhere, from Apple II's floppy disks to PCI-Express, Firewire, serial ATA, DVI, HDMI, Gigabit-Ethernet, and USB 3.0, to name a few. I will not offend you by explaining what 64B66B coding is: same idea, longer codes. It is used, in particular, for 10-Gb Ethernet (10GbE).

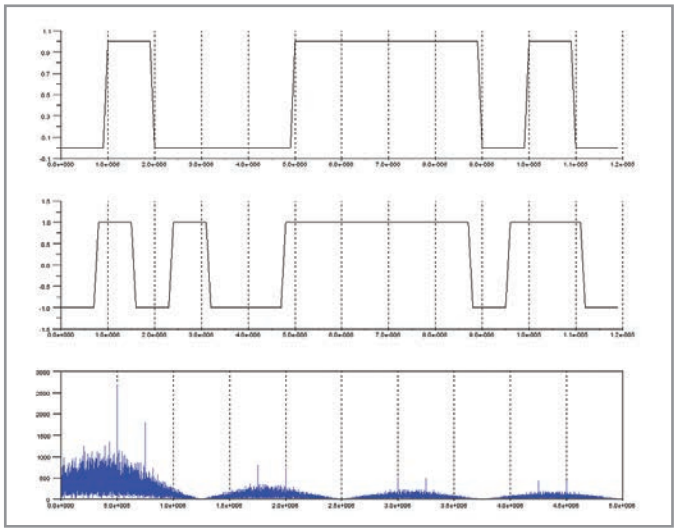


Figure 10—Symbol-coding schemes, like the 4B5B method illustrated here, are usually used in addition to other line-coding methods to improve their efficiency. Four successive bits are encoded as 5 bits through a dictionary. This enables you to shape the spectrum of the signal with great flexibility.

CLEAR TECHNIQUES

I am always amazed to see how many different methods are available to simply send binary ones and zeros over a transmission channel. These techniques must be in your engineering bag of tools, as you will undoubtedly need them at one time or another. I remember I was very disappointed the first time I tried to hook an RF transmitter module to a microcontroller. I thought the pair of transmitter and receiver modules I bought implemented a more or less transparent link, so I simply hooked them to a standard UART on both sides, which meant I used NRZ coding. It worked, but only up to 1 m, whereas the modules were given for hundred times more. In fact, these modules were just low-cost raw FSK transmitters and the receiver end disliked non-DC-balanced transmissions. I drastically increased the RF coverage range as soon as I read the datasheet and implemented a Manchester coder and decoder in my firmware. These techniques were obscure for me at that time, now they are no longer on your own darker side.

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

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/255.

RESOURCES

Computer History Museum, Oral History of Gordon Bell, 2011, www.computerhistory.org/collections/accession/102702036.

E. Gurari, "Digital Data, Digital Signal,"Ohio State University, 1998, www.cse.ohio-state.edu/~gurari/course/cis677/cis677Se12.html.

R. Lacoste, "Direct Digital Synthesis 101," *Circuit Cellar* 217, 2008.

———, "Living with Errors: An Introduction to Forward Error Correction," *Circuit Cellar* 235, 2010.

T. McDermott, *Wireless Digital Communications: Design and Theory*, Tucson Amateur Packet Radio Corp., 1997.

M. Ogorzałek, "EIE 325: Telecommunication Technologies: Line Coding," Hong Kong Polytechnic University, www.eie.polyu.edu.hk/~maciej/pdf/week5.pdf.

Scilab Software Information, The Scilab Consortium, www.scilab.org.

NEED-TO-KNOW INFO

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

—

Digital Modulations Demystified
by Robert Lacoste
***Circuit Cellar* 233, 2009**

Today's blinding data transmission speeds aren't due solely to advances in processor technology. Digital modulation plays an important role, although it can be a difficult topic to understand. This article introduces the subject and demystifies the complex mathematics involved in the theory. Topics: Digital Modulation, NRZ, OOK, ASK, Baseband Filtering, FSK

—

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

Designed a Brand New Device?

Learn to search online databases to see if your device is cutting edge and may be eligible for a patent by using this new instructional DVD!



MIDWEST PATENT SERVICES
info@midwestpatentservices.com
www.midwestpatentservices.com

Gigabit Technology

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

$10.00 each (Qty 100)

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

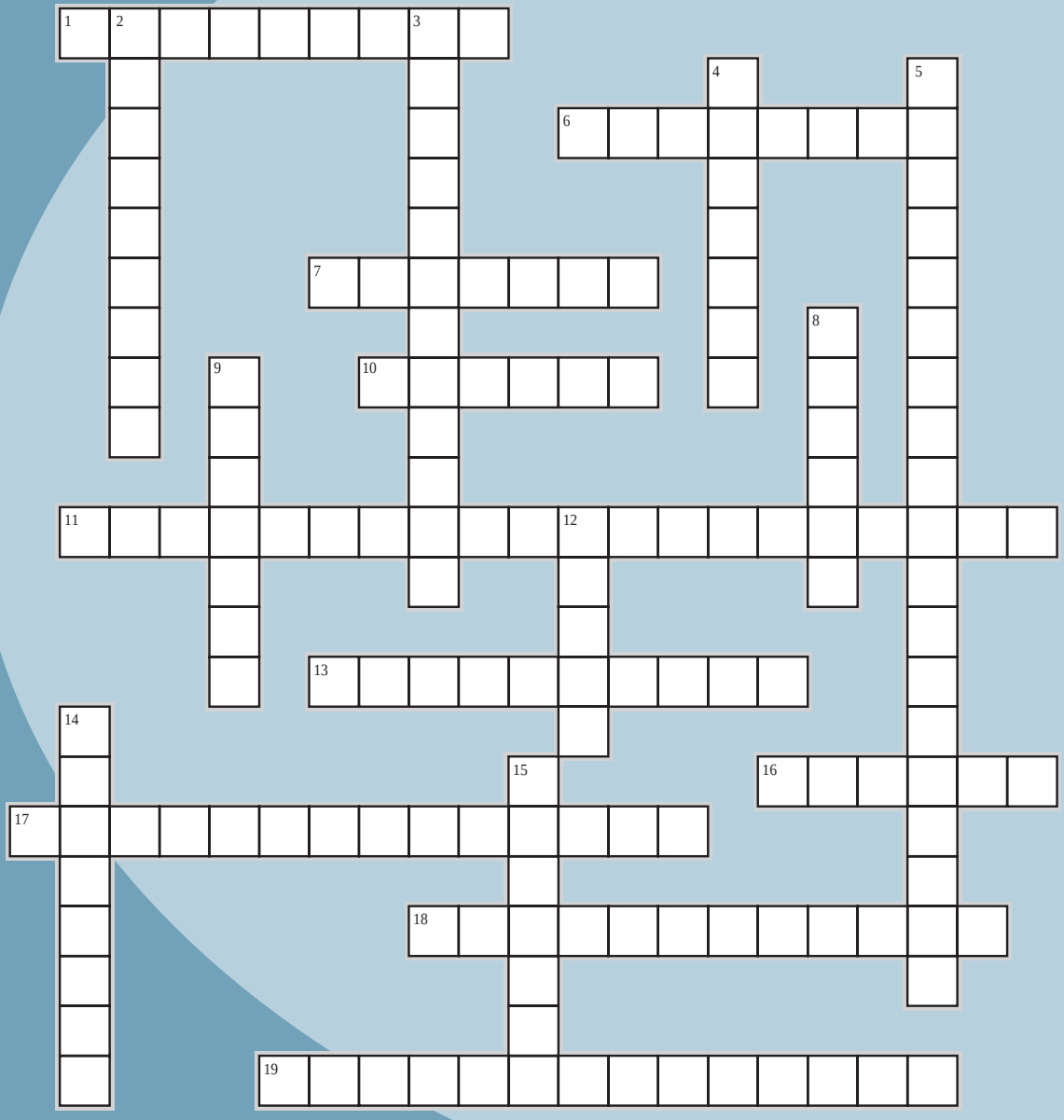
THE NETWORKING EXPERTS

gridconnect®

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

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

CROSSWORD



Across

1. The time recorded by a computer; not real time [two words]
6. Opposite of constant
7. Used to automatically distribute frequently updated information
10. A sequence, instructions
11. PWM [three words]
13. Mechanism for controlling mechanical and electronic devices [two words]
16. A time-varying or spatial-varying quantity
17. Dutch physicist who first quantified the Lorentz force [two words]
18. Merges computer components into one IC chip
19. RMS [three words]

Down

2. Device used to connect a computer to a network
3. µs
4. Electronic, mathematical
5. LSB [three words]
8. Sends local data to a remote system
9. Abbreviated as "V"; usually, there is more than one
12. Take away
14. Family of protocols for real-time distributed control
15. Frequently ends with "com" or "org"

The answers will be available in the next issue and at www.circuitcellar.com/crossword.

IDEA BOX

THE DIRECTORY OF PRODUCTS AND SERVICES

AD FORMAT: Advertisers must furnish digital files that meet our specifications (www.circuitcellar.com/advertise). ALL TEXT AND OTHER ELEMENTS MUST FIT WITHIN A 2" x 3" FORMAT. E-mail adcopy@circuitcellar.com with your file or send it to IDEA BOX, Circuit Cellar, PO Box 180, Vernon, CT 06066. For current rates, deadlines, and more information contact Peter Wostrel at 800.454.3741, 978.281.7708 or peter@smmarketing.us.

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

Control by USB or Web!





TCPmaker for control over the Web

Easy as 1-2-3:
1. **Define Your Data**, with variable names that YOU create.
2. **Lay Out Your Content** (with gorgeous web-ready screen controls that you can ***grab on to***) using TCPmaker's drag & drop Visual Page Designer.
3. **Generate Your Code**, for all Microchip C compilers, to "wire it all together."

NO PC PROGRAMMING AT ALL – just point your web browser at your device!

Simple Upgrade Management System

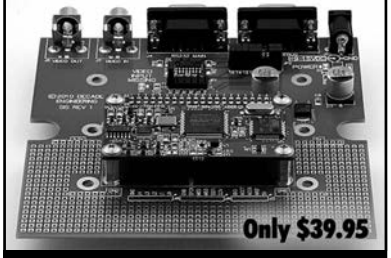
Low cost integrated system manages upgrades across different processors & connectivity types:
> Lets your end users upgrade your PIC firmware safely, simply, and securely.
> **Automatically delivers upgrade by email.**
> Simple for non-technical end users.
> **Encrypted system** protects your firmware from theft or from being programmed into the wrong device.
> **Grows with your product line:** for a new product w/ different processor or connectivity type, just add another SUMS bootloader.

Trace SYSTEMS, Inc.

From the makers of HIDmaker FS:
www.tracesystemsinc.com
888-474-1041

NEW SIS Board

Turbocharges the development of low-cost modular video text and graphics generators using **BOB-4**!



www.decadenet.com

DECADE ENGINEERING
503-743-3194 Turner, OR, USA

BPS BusBoard Prototype Systems

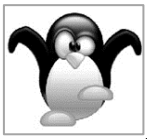
Prototyping PC Boards
Many Patterns

BusBoard StripBoard PadBoard ProtoBoard ProtoBoard-2H PowerBoard SMTBoard SMTpads PC BreadBoards

See our site:
www.BusBoard.us

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

char msg [] =
{78,111,119,32,72,105,114,105,
110,103,0};
If you can read this, we want you!



The Virginia Tech Transportation Institute needs:
Electronic Hardware Developers | Embedded Software Developers | Electrical Engineers | Electrical Technicians | Mechanical Designers

We are expanding and in search of good people.
http://www.vtti.vt.edu/employment-opportunities.php

VirginiaTech TRANSPORTATION INSTITUTE

Disk On Chip
Ready for Delivery



From 16MB to 128MB Available!
Call 530-297-6073 Sales@jkmicro.com
www.jkmicro.com

JK microsystems, Inc.
International Orders Welcome

i2cStick™



Discover I²C

Read, Write, Connect...

MCC
Micro Computer Control

www.mcc-us.com

www.can232.com

Only $108 €89



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

The original (Controller Area Network) CAN to RS232 serial converter.

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



Only $154 €129

www.canusb.com

Embedded Ethernet

ONLY $98



NEW! picoFlash

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

$129 Development System

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

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

JK microsystems

PIC-SERVO
MOTION CONTROL

MOTION CONTROLLERS FOR BRUSH, BRUSHLESS AND STEPPER MOTORS.

- controller chips
- controller boards

www.picservo.com
JEFFREY KERR, LLC

NEW APP! Download at iTunes

CC Electronic Toolbox

For iPad, iPhone, & iPod Touch



Specially designed for electronic engineers, this app provides easy access to essential electrical information & multiple conversion tools!

Ohm's Law calculator, Type listings for Resistors including SMD types and Capacitors, database of common electrical components, NE555 Circuit design, CC Tech Guide for authors & so much more!

www.circuitcellar.com/toolbox

NEW

PIC18F45K22 Development Kit





FREE BOOK written by Fred Eady with the purchase of a PIC18F45K22 Development Kit with C Compiler

Development Kit priced at $600.00
Includes: PCWH C Compiler, ICD-U64 Unit, Prototyping Board, Accessory Cables, and Exercise Book

262.522.6500 x35 • sales@ccsinfo.com
To receive this special offer visit:
www.ccsinfo.com/eadycc

INTEGRATED
PLCs with Integrated Ethernet

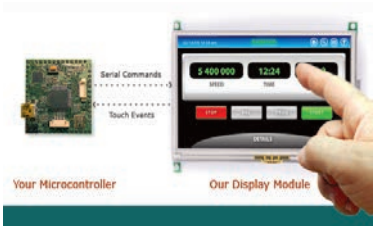


Integrated Features :
- ETHERNET / Modbus TCP/IP
- 16 or 32 digital I/Os
- 10 analog I/Os
- RS232 and RS485
- LCD Display Port
- I/O Expansion Port
- Ladder + BASIC Programming

$229 and $295
before OEM Qty Discount

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

TRI TRIANGLE RESEARCH INTERNATIONAL



Add a Touch Screen to Your Embedded Product

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

Learn more at reachtech.com, call 510-770-1417 or email info@reachtech.com.

REACH TECHNOLOGY INC. 20 1988-2008

Buy vs. Build? Download Handbook at reachtech.com.

MaxSonar
Perfect for OEMs & Engineers



XL-MaxSonar-EZ
- High acoustic power
- Real-time calibration
- Tiny size
- $39.95 USD qty. 1

MaxSonar-WR (IP67)
- Industrial packaging
- Weather resistant
- Standard ¾" fitting
- $99.95 USD qty. 1





LV-MaxSonar-EZ
- Power-up calibration
- Smallest MaxSonar
- Low power, 2.5V-5.5V
- $29.95 USD qty. 1

MaxSonar-WRC (IP67)
- Compact packaging
- Weather resistant
- Quality narrow beam
- $99.95 USD qty. 1



www.maxbotix.com

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

PICBASIC PRO™ Compiler



Bridges the gap between ease of use and professional level results. The simplicity of the language allows both hobbyist and engineer to master it quickly, but the engineer appreciates what's happening under the hood. This is a true compiler that produces fast, optimized machine code that's stable and dependable.

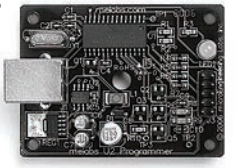
Supports:
More than 300 PIC® Microcontrollers
Direct Access to Internal Registers
In-Line Assembly Language
Interrupts in PICBASIC and Assembly
Built-In USB, I2C, RS-232 and More
Source Level Debugging in MPLAB

PICBASIC PRO™ Compiler $249.95

USB Programmer for PIC® MCUs

$89.95 *(as shown)*

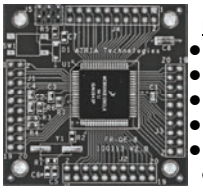
RoHS Compliant



Programs PIC MCUs including low-voltage (3.3V) devices. Includes Software for Windows 98, Me, NT, XP, and Vista.

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

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



Microcontrollers
- MCF51AC256
- MCF51CN128
- MCF51JM128
- MCF51QE128
- MC9S08AC128
- MC9S08JM64
- MC9S08QE128

Modules
- Bluetooth
- RS-232
- USB
- RF
- 2 x 20 LCD
- 4 x 4 Keypad
- Motor Control
- Real Time Clock
- Power Supply

GREAT For:
- Projects
- Breadboards
- Experiments
- Labs
- Prototypes

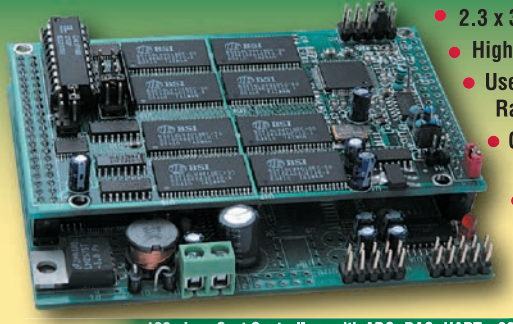
Kits Available

Learn about BASIC ON BOARD

ATRIA *Technologies Inc.*
www.AtriaTechnologies.com

GR16™ Low Cost, High Speed, Analog Signal Grabber — Up to 10MHz SPS, 16-bit ADC, and 8 MB FIFO

Start at $149

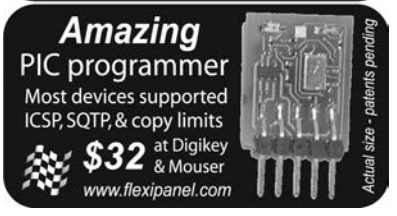


- 2.3 x 3.6 x 0.5 inches
- High speed 16-bit ADC (ADS1610)
- User Programmable Sample Rate, up to 10M SPS
- Configurable FIFO data size, up to 8 MB.
- Stackable multi units sharing Sync, and Trigger
- Driven by 16-bit TERN controllers

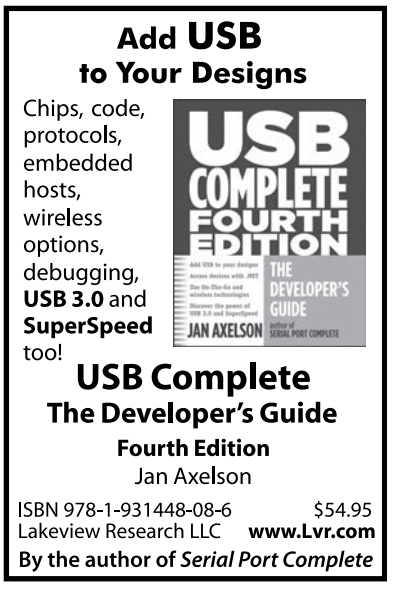
100+ Low Cost Controllers with ADC, DAC, UARTs, 300 I/Os, solenoid, relays, CompactFlash, LCD, Ethernet, USB, motion control. Custom board design. Save time and money.

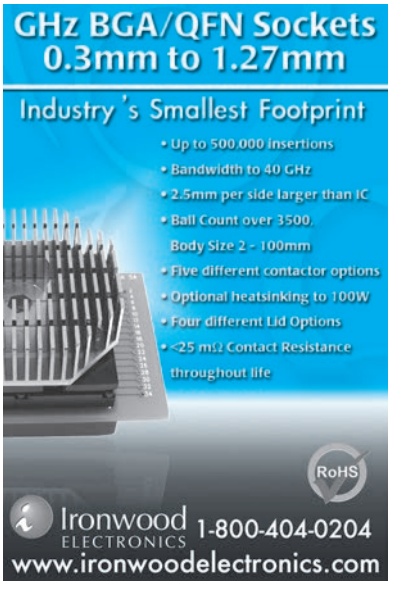
TERN INC. 1950 5th Street, Davis, CA 95616 USA
Tel: 530-758-0180 • Fax: 530-758-0181 www.tern.com • sales@tern.com

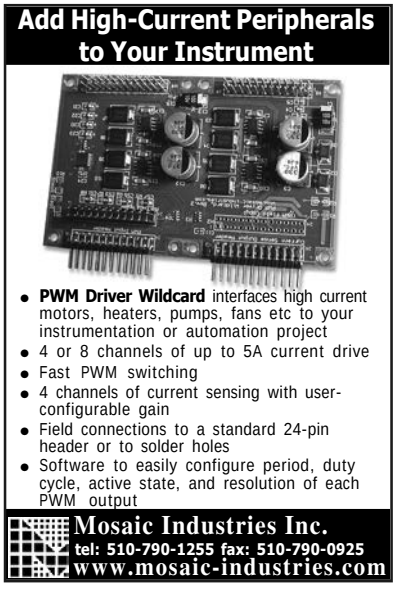














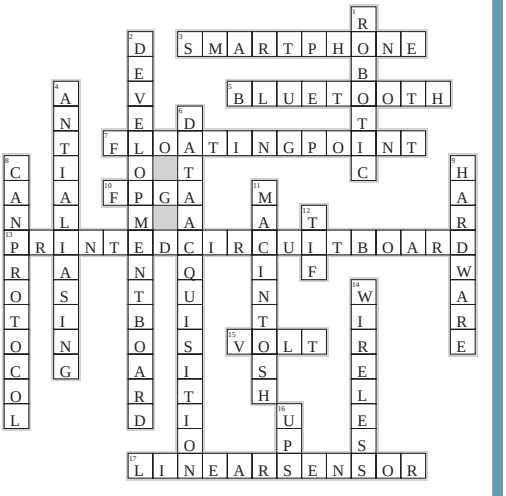
CROSSWORD ANSWERS from Issue 254

Across

3. SMARTPHONE—Portable way to update your Facebook status
5. BLUETOOTH—A wireless communication standard used to transfer data short distances
7. FLOATINGPOINT—A system that represents real numbers in a variety of values [two words]
10. FPGA—Field-programmable gate array
13. PRINTEDCIRCUITBOARD—PCB [three words]
15. VOLT—An SI unit
17. LINEARSENSOR—Reads positioning [two words]

Down

1. ROBOTIC—Automated, computerized
2. DEVELOPMENTBOARD—Helps you learn how to use a new microprocessor [two words]
4. ANTIALIASING—Makes things seem less distorted
6. DATAACQUISITION—Topic featured in this month's *Circuit Cellar* [two words]
8. CANPROTOCOL—A message-based standard for cars [two words]
9. HARDWARE—The physical side of your computer system
11. MACINTOSH—Once characterized by a rainbow-striped apple
12. TIF—Tagged image file format
14. WIRELESS—Disconnected
16. UPS—A power supply that should not be disturbed



INDEX OF ADVERTISERS

Page		Page		Page		Page	
45	2012 International CES	69	Elektor	78	Ironwood Electronics	C2	NetBurner
14	Abacom Technologies, Inc.	55	Elprotronic, Inc.	76	Jeffrey Kerr, LLC	C4	Parallax, Inc.
78	All Electronics Corp.	49	Elsevier, Inc.	13, 32	JK microsystems, Inc.	31	Pololu Corp.
71	AP Circuits	71	EMAC, Inc.	75, 76	JK microsystems, Inc.	77	Reach Technology, Inc.
65	ARM	15	Embedded Developer	29	Labcenter Electronics	23	Saelig Co., Inc.
59	ARM TechCon 2011	33	ExpressPCB	78	Lakeview Research	39	SoC Conference
77	Atria Technologies, Inc.	55	ezPCB	76	Lawicel AB	2, 3	Technologic Systems
53, 63	Audio Amateur	78	FlexiPanel Ltd.	77	Maxbotix, Inc.	77	Tern, Inc.
75	BusBoard Prototype Systems Ltd.	C3	FTDI Chip	76	MCC, Micro Computer Control	75	Trace Systems, Inc.
21	Comfile Technology, Inc.	17	GHI Electronics LLC	1	Microchip Technology Inc.	77	Triangle Research, Inc.
76	Custom Computer Services, Inc.	73	Grid Connect, Inc.	77	microEngineering Labs, Inc.	13	Troniczone
75	Decade Engineering	16	Humandata Ltd.	73	Midwest Patent Law	75	Virginia Tech
32	Earth Computer Technologies	35, 43	IC Bank	78	Mosaic Industries, Inc.	5	WIZnet Co., Inc.
24, 25	Elektor	9, 11	Imagineering, Inc.	7	Mouser Electronics, Inc.		

PREVIEW of November Issue 256

Theme: Analog Techniques

High-Accuracy Voltage Reference Using Pulse-Width Modulation (Part 2): Hardware Design

Auditory Navigator

Sound Tone Detection with PSoC (Part 1): The Hardware

THE CONSUMMATE ENGINEER **Shielding and Transfer Impedance**

EMBEDDED UNVEILED **Ionization Detectors**

LESSONS FROM THE TRENCHES **Design Development (Part 4):** Processors and Interfacing

FROM THE BENCH **Fly-by-Wire Wheelchair (Part 1):** Joystick Alternatives

ATTENTION ADVERTISERS

December Issue 257 Deadlines

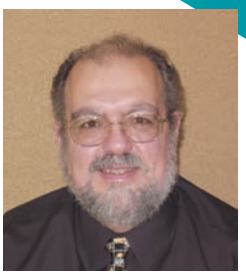
Space Close: Oct. 13
Materials Close: Oct. 20

Theme

Programmable Logic

Call Peter Wostrel
now to reserve your space!
800.454.3741 or 978.281.7708
e-mail: peter@smmarketing.us

PRIORITY INTERRUPT



by Steve Ciarcia, Founder and Editorial Director

What Goes Around

What goes around comes around—or at least something like that, or vice versa. I never could get that straight. What I mean is that I used to do something, stopped, and now I see value in doing it again. What am I talking about? Building your own PC, of course.

Seriously, in pre-PC days, it was the norm for most of us to custom configure, hand construct, or otherwise personally build our computers simply because there was little other choice. It wasn't until desktop PCs became commonplace that we all discarded the personal touch and opted instead for the convenience and economy of mass production. Certainly, it is no small pile of obsolete HP, Sony, Gateway, Samsung, Asus, and Lenovo boxes on the shelves out behind the Circuit Cellar. The common attribute is that all were purchased off-the-shelf.

A dozen obsolete computers, along with my recent laptop disaster (see "Live by the PC, Die by the PC," *Circuit Cellar* 251, 2011), got me asking myself whether I was being smart about computer buying these days. I concluded that since my applications seem to constantly evolve, I should to go back to custom configuring, at least for my main computer. In other words, as great as econo-box prices at Best Buy may seem, at least for me, this kind of off-the-shelf hardware runs out of gas much too soon.

Selecting your main PC is the epitome of "personal computing." And, depending on whether it's for gaming, CAD, video and photo editing, becoming the next chess champion, or just web surfing, the hardware can greatly vary. The primary reason you should consider rolling your own next time is that, with a little forethought, it only takes a bit of component swapping to turn the single-monitor, web-surfing desktop you have into the four-monitor CAD machine you need.

The primary reasons for building your own PC are to get exactly what you want, and do so with as much performance and consideration for expansion as your budget can handle. Custom configuration puts you back in charge of deciding the benefits and tradeoffs. Here are some of those considerations.

DIY is all about extending the useful life of your desktops. For a few dollars more, select a case and power supply that has the capacity to power and cool more than what you need right now. Perhaps even buy a power supply that can handle an additional graphics card should you ever need it (for four monitors). The unbranded power supplies and cooling systems in econo-boxes are notoriously overrated. Beware.

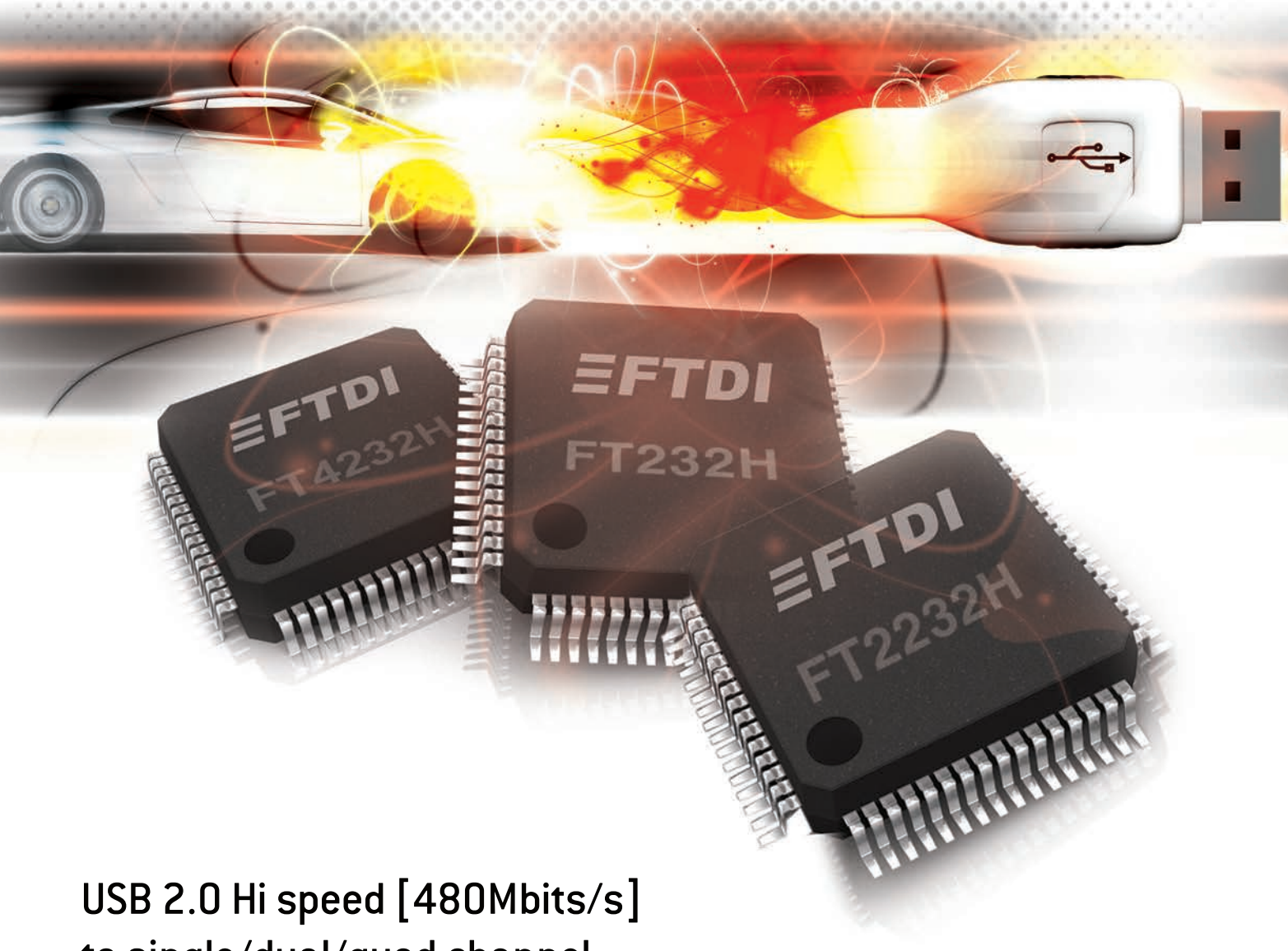
DIY gets rid of the garbage. The profit margins on retail PCs are ridiculously low and the end result is often a box that is high on promise but low on quality and performance. Manufacturers that make no profit selling retail have to make it up someplace. They generally do this by filling the hard drive with bloatware. These are dozens of trial subscriptions and software programs that are an endless pain to avoid and almost impossible to completely remove. Worse yet, an operating system reinstall from the hard drive (no CD-ROMs) puts them all back on the machine.

DIY is also about selecting the best motherboard and processor for your application. Do you even know what motherboard or Intel chipset is used in that HP econo-box on your desk? Or, heaven forbid, is it a regular chipset on a brand-specific motherboard that you can't possibly ever hope to replace or upgrade after the warrantee period? Selecting an industry-standard, brand-name motherboard and the best processor for the bucks gets you documented performance and long-term support. If you choose the hot new Sandy Bridge processor like I did, the motherboard choice these days is between two Intel chipsets, P67 and Z68. Each has its benefits. The P67 doesn't support an on-board GPU, but it does facilitate CPU overclocking. The Z68 permits overclocking, switching between an on-board GPU and discrete graphics cards (for whatever that's worth), and SSD caching. Using one of these, you can readily configure a $300 Sandy Bridge quad-core i7-2600K over-clocked to 4-GHz-plus that gives a $1,000 i7-990X a run for its money.

I'm committed to this new way of thinking because I'm getting tired of junking boxes every time I ask them to do just a little more and find them lacking. I don't want to experience another system meltdown. The good news is that DIY these days doesn't mean what it did 30 years ago. Rather than soldering together your own computer, DIY today simply means picking out all the boards and features you want in your desktop at a custom computer house like iBUYPOWER and having them stuff it all in the box with the proper drivers. They'll even add Windows 7 (with CD-ROMs) with no bloatware. Like the old days, it's time to go back and build it the way we want.

steve.ciarcia@circuitcellar.com

USB IN THE FAST LANE



USB 2.0 Hi speed [480Mbits/s]
to single/dual/quad channel
independent configurable parallel/serial interface
UART/JTAG/SPI/I^2C & parallel FIFO

FTDI Chip

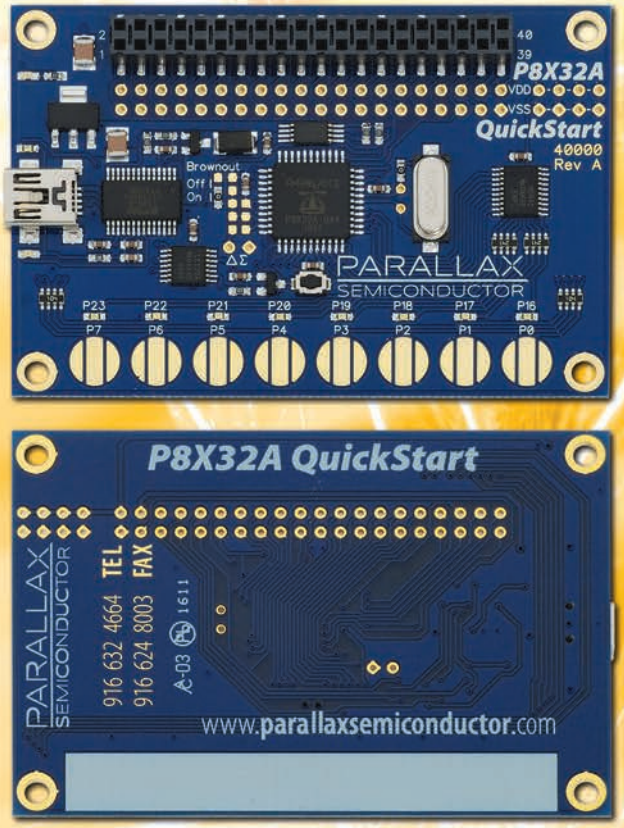
USB MADE EASY

www.ftdichip.com

Multicore Microcontroller Development Board
Propeller P8X32A QuickStart

Simple yet powerful multicore programming on a convenient evaluation board.

Discover multicore microcontroller programming with the Propeller P8X32A on the QuickStart development board. The board includes touch-buttons, LEDs, and access to all I/O pins through an expansion header. Comes pre-loaded withe demonstration code. Power from USB port or use an external power source.



FEATURES

- 32-bit, 8-core P8X32A microcontroller, up to 160 MIPS
- Buffered USB to serial converter with USB bus power
- Eight resistive touch buttons on I/O pins P0-P7
- Eight blue LEDs on I/O pins P16-P23
- 3.3 V regulator supplies up to 500 mA for projects
- 5 MHz on-board crystal, for DC to 80 MHz operation
- 64 KB EEPROM: 32 KB for code, 32 KB general-purpose
- Pads for sigma-delta A/DC circuitry
- Expansion header to access control signals, Vss, Vdd, and all 32 I/O pins
- Open-source hardware PCB design

KEY SPECIFICATIONS

- Power Requirements: 3.3 or 4 to 9 VDC, up to 500 mA
- Interface: USB, 3.3 V serial, I²C, or others as programmed
- Operating temperature: -40 to +185 °F (-40 to +85 °C)
- Dimensions: 2.0 x 3.0 x 0.36 in (5.0 x 7.6 x 0.84 cm)

open source hardware

The P8X32A QuickStart development board is an open-source hardware project. The bill of materials and the DipTrace schematic and PCB layout files are released under the Creative Commons Attribution 3.0 license. The example code is released under the MIT License. All of these files are available from the P8X32A QuickStart product page at **www.parallax.com**.

Order the **P8X32A QuickStart (#40000; $25.00)** at www.parallax.com or call Sales toll-free at 888-512-1024 (M-F, 7AM-5PM, PDT).

Friendly microcontrollers, legendary resources.™

Prices subject to change without notice. Propeller, Parallax and the Parallax logo are trademarks of Parallax Inc.

PARALLAX INC

www.parallax.com

"ParallaxInc" on Twitter, Facebook, and YouTube