PROJECT: A Receiver for Digitizing RF Signals
LOCATION: Australia
PAGE: 14

INSIGHT: Power Supplies & Surge Voltage
LOCATION: Canada
PAGE: 44

HOW TO: Use High-Speed ADCs Efficiently
LOCATION: France
PAGE: 52

CIRCUIT CELLAR

THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

FEBRUARY 2012
ISSUE 259

WIRELESS COMMUNICATIONS

Construct a GPS-Based Vehicle Timing System

Bypass Capacitors & EMI Filtering

An OS-Free Platform for CPU Experimentation

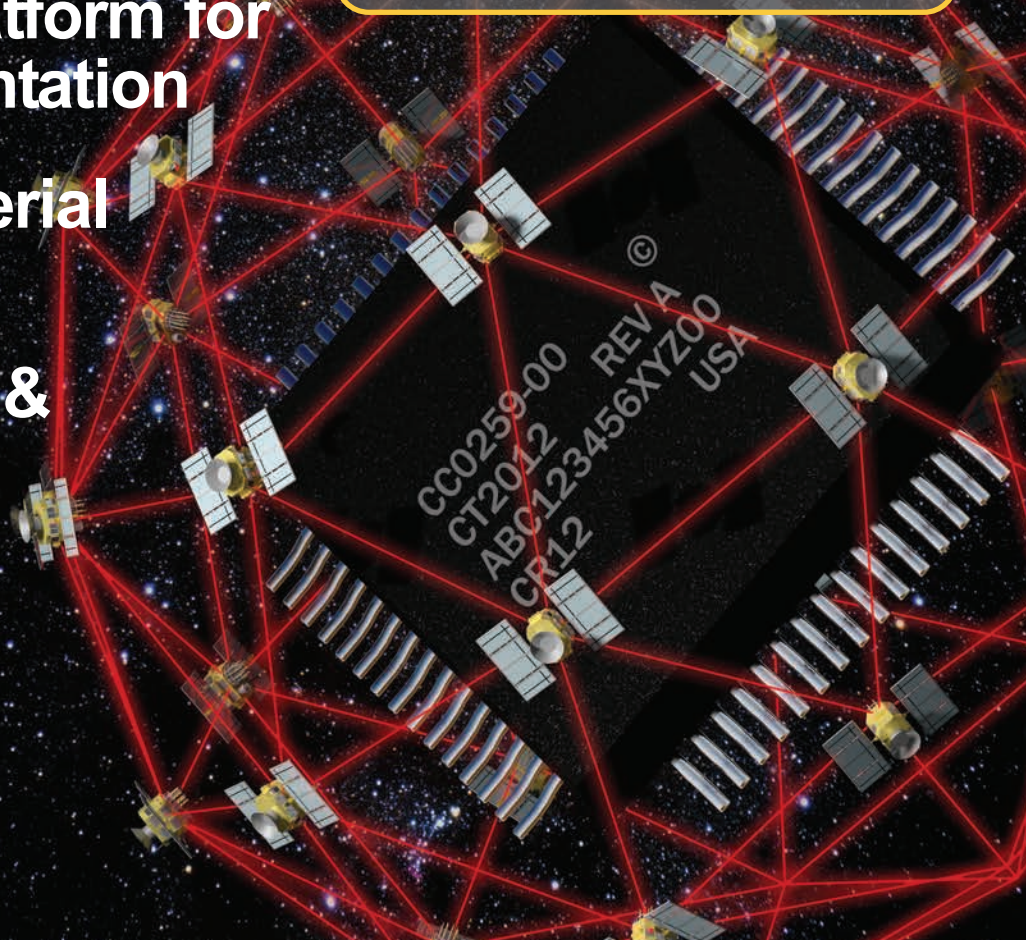
Build a Multi-Serial Protocol Tool

Analog Design & Beyond

PLUS

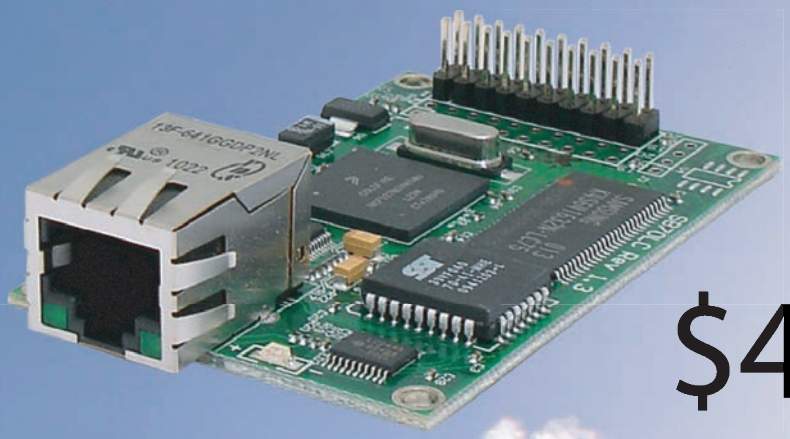
PROGRAMMING ESSENTIALS

Embedded Linux Licensing Issues
// The Four Main Licenses in Detail
// Problems with GLP-Licensed Software
// Keep IP Private While Using Linux
// And More



0 2>
7 25274 75349 9
$7.50 U.S. ($8.50 Canada)
www.circuitcellar.com

The NetBurner SB70LC



$49.00
Qty. 100
SB70LC Board

The complete hardware and software solution

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



$99.00
SB70LC Development Kit

The SB70LC Development Kit

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

NetBurner
Networking in One Day!

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

freescale™
Alliance Member

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

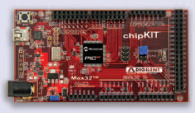
Feature	Core	Performance	Program Memory (KB)	RAM (KB)	Additional Features
chipKIT™ Uno32	32-bit	80 MHz	128	16	PMP/PSP/RTCC
chipKIT Max32	32-bit	80 MHz	512	128	USB, 2x CAN, Ethernet, DMA, RTCC

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)



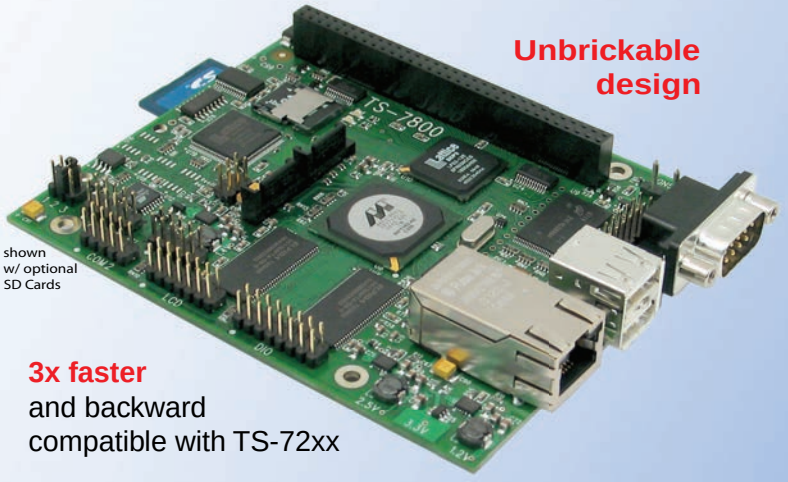
chipKIT Max32 (TDGL003)

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

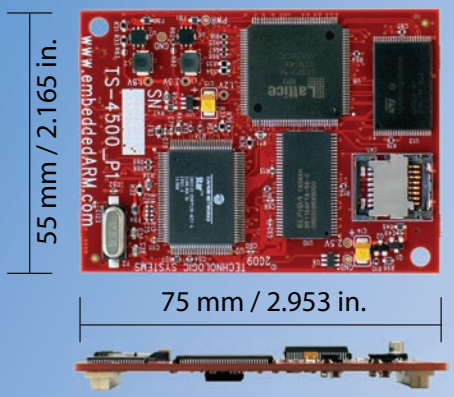


- Low power (3.2 watts), fanless
- Power via 5-12 VDC, USB, PoE (opt.)
- 64MB DDR-RAM
- 256MB ultra-reliable XNAND drive
- Micro-SD Card slot
- RS-232, RS-485, CAN, RTC
- Header with SPI and 11 DIO
- 480Mbit/s USB, Ethernet, PoE option
- Boots Linux 2.6.24 in < 3 seconds
- Un-brickable, boots from SD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



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

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

Technologic Systems

We use our stuff.

Visit our TS-7800 powered website at

www.embeddedARM.com

TASK MANAGER

Innovation in Wireless Technology

The topic of wireless communication is more relevant than ever. The technology is used everywhere from vehicles to hospitals to consumer electronics to military systems. Therefore, electronics engineers around the world are devoting their careers to innovating technologies to meet the ever-growing demand for wireless devices.

Since its inception in 1988, this magazine has been publishing articles on wireless technology. Examples include Ed Nisley's "Digitizing Infrared Signals" (*Circuit Cellar* 2), Maurizio Ferrari's "Secrets of Using the DS1209 in an RF Transponder" (*Circuit Cellar* 41), and Steve Ciarcia's "Wireless Remote Control of the AVMux" (*Circuit Cellar* 46). This month we continue the tradition.

On page 14, Clayton Gumbrell describes his QRSS-Rx design. The mbed-based receiver can digitize a small bandwidth of RF signals and send them to a server for processing.

Turn to page 30 to learn how Steve Lubbers built his "KartTracker," which is a standalone GPS-based timing system used to time racing activities. An LCD operates as a user interface, an integrated accelerometer records G forces, a serial port provides connections to a GPS receiver and a wireless transmitter, and removable flash memory stores data.

Ed Nisley tackles the topics of capacitor self-resonance, GPS technology, RF energy, and data transmission in the first part of his series titled "RFI Bypassing" (p. 38). He presents a radio project featuring SMD capacitors for suppressing RF energy, and he describes test-fixture measurements on isolated capacitors.

As usual, this issue is not devoted solely to its theme. CPUs, power supplies, ADCs, Linux, and protocols are also covered.

On page 24, Alexander Pozhitkov begins a two-part series about the NakedCPU platform. It is designed to provide full access to hardware and a CPU without any OS restrictions while working in the protected mode.

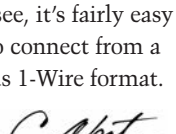
In the first part of his series on building a robust power supply, George Novacek described how to make a power supply operate properly when the primary power distribution bus voltage is lower than nominal. This month he explains what happens when it encounters an abnormal surge voltage (p. 44).

On page 52, Robert Lacoste details how to work with high-speed ADCs. The article is intended to demystify the process of using high-speed converters effectively.

Another topic that needs demystification is embedded Linux, which is why Bob Japenga has been covering the subject in his ongoing series "Getting Started with Embedded Linux." This month he addresses various licensing issues you'll face as you work with Linux (p. 60).

Columnist Jeff Bachiochi wraps up this issue with the article "Multi-Serial Protocol Tool" (p. 64). As you'll see, it's fairly easy to build a multi-serial protocol tool designed to connect from a USB host an I²C, SPI, RS-232/422/485, or Dallas 1-Wire format.

cj@circuitcellar.com



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
Bob Japenga
Robert Lacoste
George Martin
Ed Nisley
George Novacek
Richard Wotiz

PROJECT EDITORS
Ken Davidson
David Tweed

PUBLISHER
Hugo Van haecke

ASSOCIATE PUBLISHER
Shannon Barraclough

ART DIRECTOR
KC Prescott

CONTROLLER
Jeff Yanco

CUSTOMER SERVICE
Debbie Lavoie

ADMINISTRATIVE COORDINATOR
Valerie Luster

MARKETING REPRESENTATIVE
Amanda Anderson

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®.

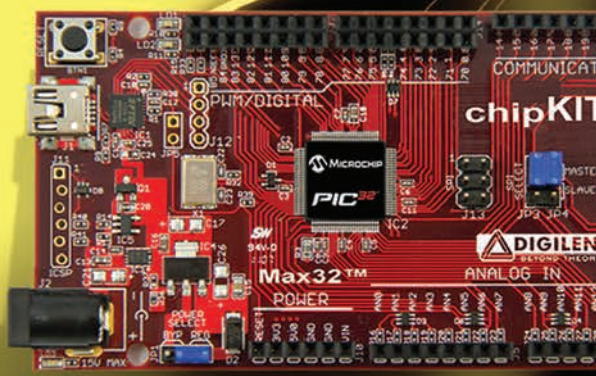
The information provided by Circuit Cellar® is for educational purposes. Circuit Cellar® makes no claims or warrants that readers have a right to build things based upon these ideas under patent or other relevant intellectual property law in their jurisdiction, or that readers have a right to construct 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 © 2012 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.

DESIGNSPARK

Turn a hot idea into a cool solution.

DesignSpark chipKIT™ Challenge



Have you entered the **DesignSpark chipKIT™** Challenge yet? Visit **www.chipkitchallenge.com** today to join the fun!

When you submit a proposal for an energy-efficient design, your project will automatically be considered for a chipKIT™ Community Choice Award.* In February, participants of the chipKIT™ Challenge will have the opportunity to vote on what project they think is the best. If your project receives the most votes, you will win a $100 voucher for RS Components/Allied Electronics and a free digital subscription to *Circuit Cellar* and *Elektor* magazines!

Register your project at **www.chipkitchallenge.com** to participate.

Visit www.chipkitchallenge.com
for complete rules and details.

* Participation in the Community Choice Awards does not increase your chances of winning the Grand Prize with your Final Project(s) submission. The deadline for Final Project submissions is March 27, 2012. See website for more information.

IN ASSOCIATION WITH:

elektor · MICROCHIP · RS · CIRCUIT CELLAR · ALLIED ELECTRONICS · DIGILENT BEYOND THEORY

DesignSpark chipKIT™ Challenge

chipKIT™ is a registered trademark of Microchip Technology Inc. Max32™ is a registered trademark of Digilent, Inc.

INSIDE ISSUE 259

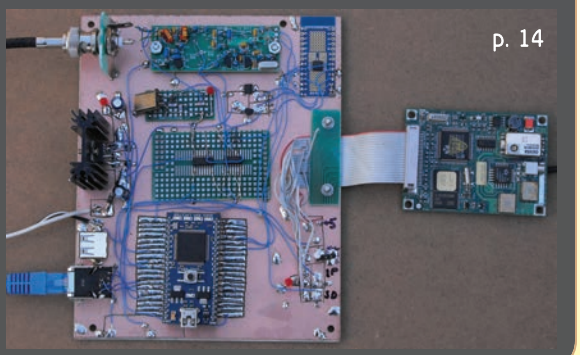
February 2012 • Wireless Communications

14 **QRSS-Rx**
A Network for Radio Communication
Clayton Gumbrell

24 **The NakedCPU (Part 1)**
Hardware Experiments and a Roadmap for Navigating Hardware Documentation
Alexander Pozhitkov

30 **KartTracker**
A GPS-Based Vehicle Timing & Monitoring System
Steve Lubbers

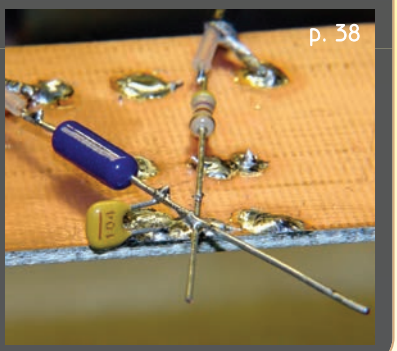
A Receiver for Digitizing RF Signals


p. 14

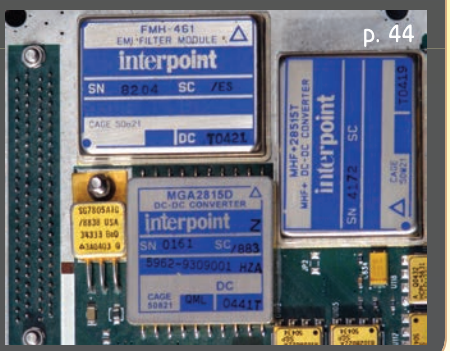
GPS-Based Timing


p. 30

Capacitor Self-Resonance


p. 38

Power & Surge Voltage


p. 44

38 **ABOVE THE GROUND PLANE**
RFI Bypassing (Part 1)
Capacitor Self-Resonance
Ed Nisley

44 **THE CONSUMMATE ENGINEER**
Build a Robust Power Supply (Part 2)
Abnormal Surge Voltage
George Novacek

52 **THE DARKER SIDE**
Playing with High-Speed ADCs
Robert Lacoste

60 **EMBEDDED IN THIN SLICES**
Getting Started with Embedded Linux (Part 3)
Linux Licensing Issues
Bob Japenga

64 **FROM THE BENCH**
Multi-Serial Protocol Tool
Jeff Bachiochi

TASK MANAGER 4
Innovation in Wireless Technology
C. J. Abate

NEW PRODUCT NEWS 8

TEST YOUR EQ SOLUTIONS 13

QUESTIONS & ANSWERS 48
An Analog Life (And a Lot More)
An Interview with Chris Gammell

CROSSWORD 74

INDEX OF ADVERTISERS 79
March Preview

PRIORITY INTERRUPT 80
Business in the Cloud?
Steve Ciarcia

AS9120A
Certified Distributor

Scan Here

mouser.com
Semiconductors and electronic components for design engineers.

Authorized Distributor

We deliver ASAP, PDQ and JIT. So, you're never SOL.

Mouser delivers the components you need, on-time. And with local Technical Support and Customer Service Experts in 19 locations around the world, you'll find the newest components to launch your new design seemlessly.

mouser.com

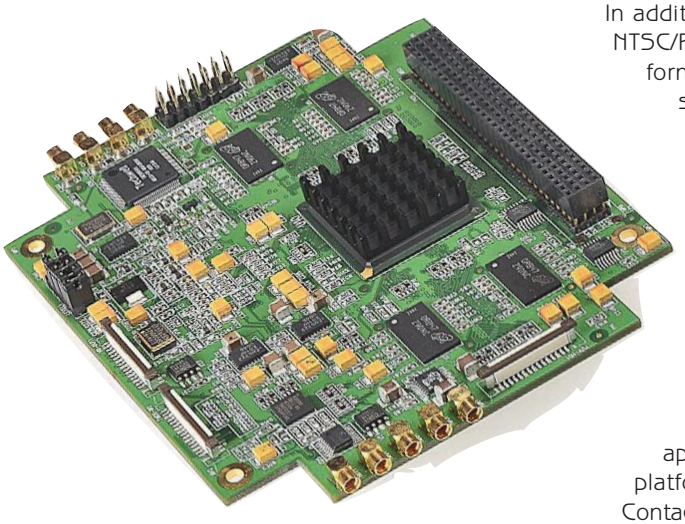
The Newest Products for Your Newest Designs®

MOUSER ELECTRONICS®
a tti company

Mouser and Mouser Electronics are registered trademarks of Mouser Electronics, Inc. The Newest Products for Your Newest Designs is a registered trademark of Mouser Electronics, Inc.

ULTRA-LOW-LATENCY QUAD H.264 ENCODER

The **H264-ULL-SD4** is an ultra-low-latency, quad-channel, H.264 encoder on a single PCI/104 form factor board. The encoder can capture and compress up to four analog video inputs at full size and at full frame rate to the H.264/MPEG-4 AVC (Part 10) standard.



In addition to the ultra-low-latency, full-frame rate encoding of four NTSC/PAL analog video sources, the H264-ULL-SD4 can also perform stream duplication to provide multiple encodings of the same input. This enables streams to be created at different resolutions, compression settings dependent on requirements and available bandwidth. The high-performance H.264 video compression and efficient bus utilization enable up to four of the encoders to be fitted in a PCI/104 system.

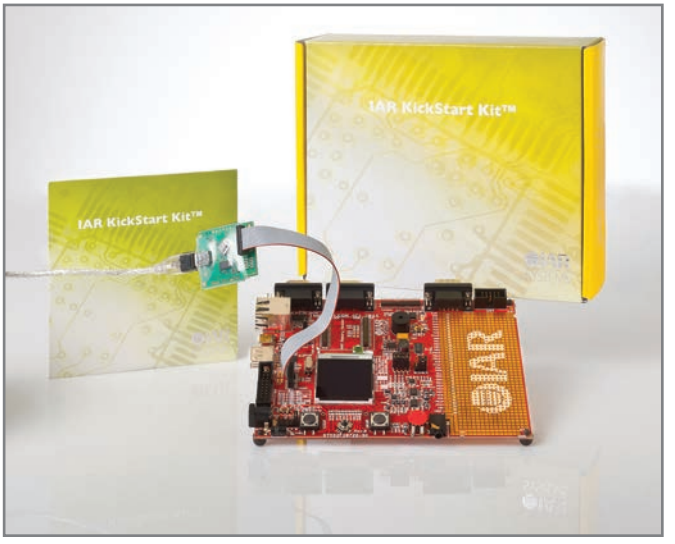
The H264-ULL-SD4 quad-channel encoder is supported by a set of well-documented and established software development kits that minimize development risk and shorten time to market for applications requiring video recording or streaming. These high-definition, ultra-low-latency H.264 encoder cards are well suited for applications including defense, transportation, and remote platforms.

Contact Advanced Micro Peripherals for pricing.

Advanced Micro Peripherals, Inc.
www.ampltd.com

STARTER KIT FOR THE STM32 F-2 SERIES

The **STM32F2*xx*** series is a complete starter kit for STMicroelectronics's ARM Cortex-M3-based high-performance STM32 F-2 series. The starter kit series combines advanced 90-nm process technology with the adaptive real-time memory accelerator (ART accelerator) and the multilayer bus matrix. **IAR KickStart Kit for STM32F207** contains all the necessary hardware and software and enables embedded software applications to be designed, integrated, and tested on hardware. It includes a feature-rich evaluation board, software tools, a debug probe, example projects, and board support packages for several real-time operating systems (RTOSes). It is easy to use for evaluation and prototypeing purposes due to a high level of integration between its hardware, software, and tools.



The evaluation board is fitted with peripherals including an STM32F2072G microcontroller, a color LCD, connectors for USB host, USB OTG, Ethernet and CAN, and a headphone jack. Debug connectors for ETM trace, SWD, and JTAG are also included.

The IAR J-Link Lite debug probe, which is also included, provides JTAG and SWD debug interfaces. Additional debug capabilities can be enabled by separate debug probes. IAR J-Trace enables full instruction trace, and IAR J-Link Ultra enables power debugging features in IAR Embedded Workbench. Both probes are sold separately.

The kit is prepared for use with IAR Embedded Workbench, a toolchain for ARM-based microcontrollers recognized for its efficient code generation, comprehensive debugger, and user-friendly integrated development environment (IDE).

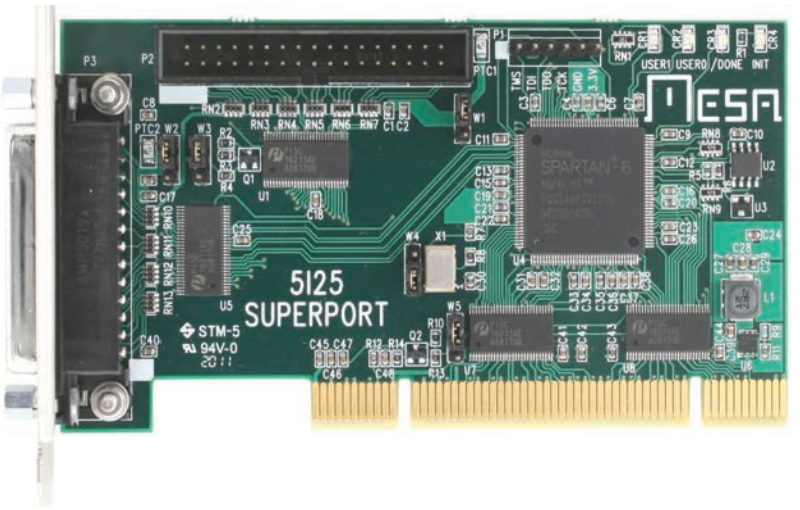
The IAR KickStart Kit for STM32F207 costs **$289**.

IAR Systems
www.iar.com

NEW PRODUCT NEWS

5I25 PCI ANYTHING I/O FPGA CARD

The **MESA 5I25** is a low-cost, low-profile, general-purpose programmable industrial I/O card for the PCI bus. The card uses standard parallel port pinouts and connectors and is compatible with most parallel-port interfaced motion-control/CNC breakout cards/multi-axis step-motor drives, enabling a motion-control performance boost while retaining a reliable real-time PCI interface. Unlike the parallel port the 5I25 replaces, each I/O bit has individually programmable direction and function.



Open-source field-programmable gate array (FPGA) firmware configurations are provided for hardware step/dir generation, pulse-width modulation (PWM) generation, analog servo control, absolute (SSI and BISS) and incremental encoder counting, real-time remote I/O, timing, event counting, and high-speed serial communication. All 5I25 I/O bits are 5-V tolerant. The 5I25 has a 25-pin DB25F backpanel connector and a 26-pin header for the second I/O port. The PCI bus interface is 3.3-V and 5-V compatible. A jumper-selectable PTC-protected power option enables 1 A of 5-V power to be supplied to the external daughtercards. Four-layer construction minimizes radiated electromagnetic interference (EMI) and provides optimum ground and power integrity. A series of daughtercards is available for industrial motion control, CNC retrofit, high-speed real-time I/O, RS-422 interfaces, encoder counting, and other applications.

The 5I25 costs **$64** in 100-piece quantities.

Mesa Electronics
www.mesanet.com

NPN

Ethernut 5
is a safe place for
your data

Ethernut 5:
185 €
(sorry, squirrel not included)



Linux and Nut/OS RTOS support | Extended temperature range | Long term availability | Open source hardware with Eagle CAD files | AT91SAM9XE ARM CPU with protected Flash | 1 GB NOR Flash | 128 MB RAM | PoE IEEE 802.3af | RTC, USB, A/D...

egnite

www.egnite.de

DOS
It's been around a long time...for a reason

Flashlite 186



Preloaded with DOS & Flash File System
33MHz 186 Compatible Processor
512K Flash, 512K DRAM
32 Pin DIP Socket to accept
44 Digital I/O Lines
2 Serial Ports
Console / Debug Port
Watchdog & (2) 16 bit Timers

Read write file system

Boots in less than a second

picoFlash



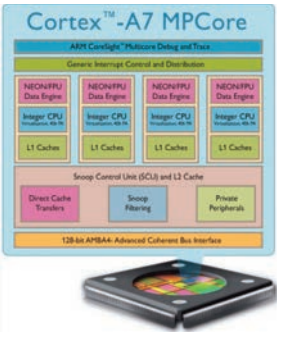
DOS & Web Server
TCP/IP & Flash File System
40MHz 186 compatible processor
512K Flash, 512K DRAM
10Base-T Ethernet
16 Digital I/O Lines
2 Serial Ports,
Console/Debug Port (RS232)
Watchdog & (2) 16-bit Timers
Hardware Clock/Calendar

Made in the USA - Distributed Worldwide

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

CORTEX-A7 MPCore PROCESSOR WITH big.LITTLE PROCESSING

The **ARM Cortex-A7 MPCore processor** now features **big.LITTLE processing**—a flexible approach that redefines the traditional power and performance relationship. The Cortex-A7 processor builds on the low-power leadership established by the Cortex-A8 processor featured in many smartphones. A single Cortex-A7 processor delivers five times the energy efficiency and is one fifth the size of the Cortex-A8 processor, while providing significantly greater performance.



big.LITTLE processing enables devices to seamlessly select the right processor for the right task, based on performance requirements. This selection is transparent to the application software or middleware running on the processors. The efficiency of the ARM architecture is the reason why ARM processors use less power and occupy a smaller footprint. The Cortex-A7 processor occupies less than 0.5 mm^2, using 28-nm process technology, and provides performance in both single and multicore configurations.

big.LITTLE processing combines two different but compatible processors within the same system-on-a-chip (SoC) and enables the power management software to seamlessly select the right processor—or multiple processors—for the right task. The processors appear identical from an application's software perspective.

The "LITTLE" lowest-power processor—in this case, the Cortex-A7—runs the operating system (OS) and applications for basic, always-on, always connected tasks, such as social media and audio playback. The OS and applications can then be seamlessly migrated to the higher-performance processor as demands increase for high-end tasks, such as navigation and gaming. The time for this migration is approximately 20 µs. This flexible approach of choosing the right processor for the right job enables optimized processing which results in significant energy savings for common workloads.



The efficient and seamless switching of workloads between the two processors is supported by advanced ARM system IP, such as AMBA 4 ACE Coherency Extensions. This ensures full cache, I/O, and processor-to-processor coherency between the Cortex-A15 and Cortex-A7, and across the complete system. Software and applications can therefore continue to run unhindered, and unnoticed by the user, as the tasks are rebalanced.

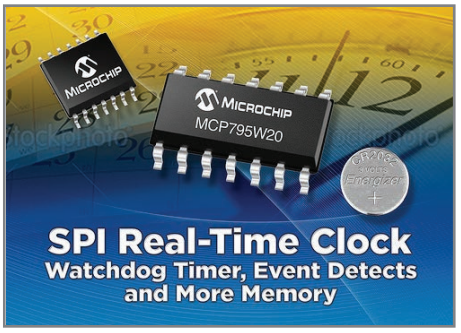
big.LITTLE power management software is developed by ARM ecosystem partners using ARM DS-5 tools and Fast Model virtual prototyping technology. The virtual platform, available now to lead partners, contains the Cortex-A15 and Cortex-A7 processors, as well as cache coherent interconnect system IP, enabling full system software development.

Contact ARM for pricing.

ARM Ltd.
www.arm.com

STANDALONE RTCC DEVICES

The **MCP795WXX/BXX** family of standalone real-time clock/calendar (RTCC) devices feature a 10-MHz serial peripheral interface (SPI), millisecond alarms, and nonvolatile memory. The RTCC devices are designed to reduce component counts and costs for applications in markets including smart energy (e.g., thermostats, power meters, and commercial refrigeration), home appliance (e.g., coffee makers, stoves, and microwave ovens), automotive (e.g., dashboard controls, car radios, and GPS), consumer electronic (e.g., office equipment and video systems), and communications (e.g., radios, cordless phones, and network systems).



The 10-MHz SPI featured on the MCP795WXX/BXX RTCC devices supports fast data access, resulting in longer microcontroller sleep and power-down modes, and less power draw from the battery. Other low-power features include two event-detect inputs that draw minimal current in the RTCC device, but wake the microcontroller when events such as a switch press or system tampering have occurred. The devices also include three types of nonvolatile memory: EEPROM, SRAM, and a unique ID. The memory can be ordered blank or with a preprogrammed MAC address, as a standard product to further reduce component counts and lower costs associated with serialized programming. In addition to the automatic battery switchover for back-up power, a power-fail time stamp logs the time and duration of any power failure and, for accurate timekeeping over a broad temperature range, the devices feature a wide digital-trimming range that compensates for crystal frequency drift over temperature. In designs that require a low-frequency clock source, a boot-clock option in the RTCC can provide a 32-KHz clock output at power-up and eliminate the need for a second crystal.

The MCP795WXX and MCP795BXX RTCC devices are available in 14-pin SOIC and TSSOP packages. Pricing starts at **$1.18** each, in 10,000-unit quantities.

Microchip Technology, Inc.
www.microchip.com

by Ian Bromley

DesignSpark chipKIT™ Challenge Update

Now in its third month, the DesignSpark chipKIT™ design challenge for energy-efficient applications is witnessing the development of some wild and innovative projects.

We've been absolutely delighted by the amount of enthusiasm we've received for the DesignSpark chipKIT™ competition—and also the high quality and imagination of the ideas that have been submitted thus far. As a reminder, or for those hearing about this for the first time, the DesignSpark chipKIT challenge, which was launched in November last year at Elektor Live!, is about encouraging engineers, students, and hobbyists to develop new and innovative energy-efficient solutions, while also maintaining an eco-friendly footprint. And what's more, total cash prizes of $10,000 can be won, including a first prize of $5,000.

I think it's reasonably clear that we need to achieve a much higher level of energy sustainability on both a local and global level, and innovative energy-efficient embedded electronics can make a significant contribution in meeting this goal. As I said in last month's issue, it's not just about developing new technologies at the device level, such as ultra-low-power microcontrollers or other electronic devices, there are many possibilities and opportunities at the system or application level. For example, a more energy-efficient light bulb helps save energy for sure, but perhaps even more valuable in terms of energy saving is a home automation control system that handles lighting around the home, turning lights on and off when we actually need them.

Back at the launch of the competition at Elektor Live!, we certainly enjoyed a couple of highly successful workshops with some excellent cooperation and camaraderie from all the participants. I had superb help from my competition colleagues: Jeroen Hobbelmans from Microchip Technology and Clemens Valens from Elektor, so, my thanks go out to them. The only technical issues we experienced were with USB drivers and connecting the chipKit boards to PCs. So, a couple of tips that could be useful for any new entrants experiencing difficulties in getting their board up and running: first, within the open-source multi-platform integrated development environment (MP IDE) tool library is a wide range of FTDI drivers, which should fix most connectivity issues; second, although it wasn't a widespread issue at the workshops, could be the possibility of some issues with those running Windows 7, however this can easily be fixed by running Windows XP Mode, where available.

With the competition well under way, entrants are now busy developing energy-efficient and environmentally friendly applications based on the chipKIT™ Max32™ development platform from Digilent, which features Microchip's 32-bit PIC32 microcontroller. The chipKIT™ Max32™ development platform is a 32-bit Arduino-compatible solution that enables engineers, students, and hobbyists to easily and inexpensively integrate electronics into their projects. The chipKIT™ hardware is compatible with existing 3.3-V Arduino shields and applications, and can be developed using a modified version of the Arduino IDE and existing Arduino resources, such as code examples, libraries, references, and tutorials.

Some of the submissions we've seen so far include those designed for applications in home automation, energy monitoring, controllers for solar power charging, and smart wind-turbine monitors. A couple of the more exotic ones include a control system for unmanned underwater glider (a type of AUV—autonomous underwater vehicle), and a "miles per gallon" (that's km/L for those who prefer new money) energy-consumption efficiency display for use in old or classic cars to enable more economical driving.

As a reminder, all entries must include an extension card developed using RS's free-of-charge and award-winning DesignSpark PCB software tool with code compiled using Microchip's MPLAB® IDE software. Additionally, during the competition, which finishes at the end of March, entrants are strongly encouraged to engage and interact with other members of the online DesignSpark community, available at www.designspark.com, by posting information on their projects, providing updates on progress, and sharing comments and ideas on their respective designs. Participants will automatically qualify for entry into bonus Community Choice Awards, in addition to admission into spot prize draws for the best collaboration to win vouchers exchangeable for products ordered from RS Components/Allied Electronics.

The competition entries will be judged on the level of energy efficiency and the quality of the extension card's PCB design. Entries are due on March 28, 2012, and the winners will be announced in April 2012.

Further details and registration for the DesignSpark chipKIT™ challenge are available at www.chipkitchallenge.com.

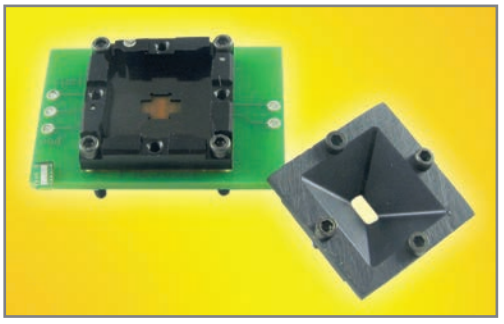
Ian Bromley is a Technical Marketing Engineer at RS Components and the Project Manager for the DesignSpark PCB software tool. Prior to working for RS, Ian worked for many years as a design support consultant with Texas Instruments, in addition to working as a field applications engineer immediately following his graduation in 1994 with an honours degree in Microelectronic Engineering.



To register and enter the DesignSpark chipKIT™ Challenge, go to: www.chipkitchallenge.com

HIGH-PERFORMANCE ELASTOMER SOCKET

The **SG-MLF-7059** is a high-performance elastomer socket for 0.5-mm pitch 10-pin Optical DFN package. The socket is designed for a 3.5 mm × 2.1 mm × 0.65 mm package size and operates at bandwidths up to 10 GHz with less than 1 dB of insertion loss.



The contact resistance is typically 20 MΩ per pin. The socket connects all pins with 10-GHz bandwidth on all connections. It is mounted using supplied hardware on the target printed circuit board (PCB) with no soldering required. Due to its small footprint, inductors, resistors, and decoupling capacitors can be placed very close to the device for impedance tuning. The SG-MLF-7059 also incorporates a simple hardware installation method so ICs can be quickly changed out. The socket lid has a unique window opening to enable much light to be exposed on top of the package for optical functions.

The SG-MLF-7059 sockets are constructed with high-performance and low-inductance elastomer contactor. The temperature range is –35°C to 100°C. The pin self inductance is 0.15 nH with mutual inductance of 0.025 nH. Capacitance to ground is 0.01 pF. Current capacity is 2 A per pin.

The SG-MLF-7059 costs **$252**.

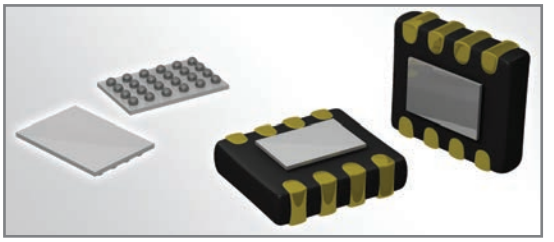
Ironwood Electronics
www.ironwoodelectronics.com

SIMPLE SWITCHER NANO MODULES & REGULATORS

Texas Instruments (TI) introduced four SIMPLE SWITCHER power management integrated circuits for space-constrained, point-of-load designs in industrial, communications, and automotive applications. The National **1-A LMZ10501** and **650-mA LMZ10500** nano modules with on-chip inductor and **2-A LMR24220** and **1-A LMR24210** nano regulators feature high performance and nano packaging as small as 7.5 mm^2. Used in conjunction with the WEBENCH online design tool, the products can help simplify and speed the design process.

In a typical space-constrained design, the nano power modules, which support an input voltage range of 2.7 V to 5.5 V, can be designed in tandem with the nano power regulators to step down from an intermediate voltage rail to the point of load, providing a complete power management solution in space- and height-limited applications.

The Laguna GW2380 Network Platform is a miniature, full-featured, embedded router board with a footprint smaller than 5 in^2 and power consumption less than 3 W. Designed for space-constrained, low-power embedded applications, the GW2380 features TI's new 1-A LMZ10501 nano module along with the 3-A LMZ14203 power module.



The SIMPLE SWITCHER power modules adhere to the CISPR 22 Class B radiated and conducted electromagnetic interference (EMI) standard for communications applications. The new nano modules also meet the CISPR 25, Class 5 radiated noise emissions standard for automotive applications. For portable applications, TI also offers the 600-mA TPS82671 in a fully integrated 6.7-mm^2 MicroSiP package.

At 95% peak efficiency, TI's 1-A LMZ10501 and 650-mA LMZ10500 nano modules provide the efficiency of a synchronous switching regulator with the simplicity of a linear regulator. With their integrated inductor and tiny size, the modules eliminate the complex layout placement challenges typical of switching regulator designs. Featuring a fixed 2-MHz frequency pulse-width modulation (PWM) mode and low output ripple of less than 10 m V_{PP}, the modules deliver low noise in a small solution size.

The SIMPLE SWITCHER nano power regulators also have several key benefits. The 2-A LMR24220 and 1-A LMR24210 nano regulators operate over a 4.5-V to 42-V input voltage range and deliver an output voltage up to 24 V. Each regulator features integrated synchronous MOSFETs for peak efficiency of 94.6% and reduced bill of materials. The regulators's constant on time (COT) with emulated ripple mode (ERM) architecture requires no loop compensation and provides ultra-fast transient response. The unique bump configuration eliminates the need for micro vias within the printed circuit board (PCB), which reduces the cost and simplifies the PCB design.

The nano modules are offered in an eight-pin leadless surface-mount package measuring 2.5 mm × 3 mm × 1.2 mm. The LMZ10501 costs **$1.80** and the LMZ10500 costs **$1.30** in 1,000-unit quantities. The nano regulators are offered in a 28-pin micro SMD package measuring 2.45 mm × 3.64 mm × 0.60 mm. The LMR24220 costs **$2** and the LMR24210 costs **$1.50** in 1,000-unit quantities.

Texas Instruments, Inc.
www.ti.com

Test Your EQ

Answer 1—Q1 and Q2 are wired as a differential amplifier, but in this digital application, it's probably better to think of them as a current switch. Whenever VIN < VTH, the current through R1 is shunted to ground, and whenever VIN > VTH, the current passes instead through Q2 to the base of Q3, turning it on.

As long as Q1 and Q2 are reasonably well matched, VTH is the input voltage at which the switchover occurs, and this is relatively stable with respect to temperature. For example, if VIN is being driven by standard TTL logic, you might set VTH to 1.5 V.

Answer 2—R1, combined with VTH, sets the amount of current flowing through Q2 to drive the base of Q3. This current should be sufficient to drive Q3 into saturation, given the expected load on it (R3 plus the external circuit).

R2 serves to make sure that Q3 isn't turned on by any leakage current through Q2 when it's supposed to be off. For example, a value of 100 kΩ would bypass currents of up to 6 µA or so.

Answer 3—When VIN is high, there is virtually no current flowing through the input terminal—just the leakage current through Q1's base-collector junction.

When VIN is low, the driving circuit must sink the current set by R1 divided by the beta (current gain) of Q1. For example, if R1 is 4,300 Ω, giving a current through Q1 of about 1 mA, and the beta of Q1 is 50, then the driving circuit must sink about 20 µA.

Answer 4—Better: The component count is reduced by one transistor.

Better: VTH is stabilized by the forward drop of a diode, making it less dependent on the exact value of the positive supply.

Worse: The switching voltage is now determined by the combination of D1 and the base-emitter drop of Q2, both of which vary with temperature.

Worse: Now the driving circuit must supply all of the base current for Q3 when it is in the high state.

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





FEATURE ARTICLE

by Clayton Gumbrell (Australia)

QRSS-Rx

A Network for Radio Communication

The QRSS-Rx features an mbed at the core of a QRSS receiver, which is an RF receiver that can digitize a small bandwidth of RF signals and send them to a server for processing into a spectrum image for real-time display on a website. This project is intended to improve the QRSS grabber system.

What is QRSS? QRSS is one of many modes of communication and experimentation used by radio amateurs. It is the transmission and reception of very slow-speed Morse transmissions at very low-power levels—typically less than 1 W, and often a lot less.

The term QRSS comes from the QRS Q-code (a message encoding system, or a brevity code, used by radio amateurs among others) meaning, "please send slower." Due to the extreme slowness of this mode, QRS is adapted to QRSS. Morse speeds are usually specified in words per minute (WPM), but for QRSS speed is instead expressed in the period of a Morse dot. The commonly used speed of QRSS3 means a dot lasting for three seconds. This equates to about 0.4 WPM.

Due to the very slow speed—and using the magic of digital signal processing (DSP)—these transmissions can be extracted from below the noise and picked up from thousands of kilometers away. Reception is performed using "grabbers." Audio from a receiver is fed into a computer sound card where software processes it to produce a spectrum image of a small portion of the radio spectrum, typically 100-Hz wide. This image is made available on a website, providing almost real-time access to how a signal is being received across the globe (see Photo 1).

The hardware involved with a grabber receive system is not insignificant. It requires a frequency-stable radio receiver, a PC (with soundcard), and an Internet connection. The receiver audio is fed into the PC soundcard. Software then converts the audio to an image. Additional software makes these images available on a webpage. There is a variety of different software configurations used, with each grabber having its own look and feel.

Each grabber is hosted on its own webpage—although there are some websites that will aggregate a number of grabbers onto the same page. Finding your signal can therefore require the inspection of a number of websites.

Dealing with such a small RF spectrum requires the grabber receiver to be precisely tuned to the frequency. Typically, a receiver will drift, so keeping a grabber on frequency can be challenging. Because of the challenges and required resources, there are few grabber stations operating, with only about a dozen grabbers commonly available.

ENTER THE QRSS-Rx

This project—the QRSS-Rx—is designed to improve the QRSS grabber system by putting together a "QRSS Network."

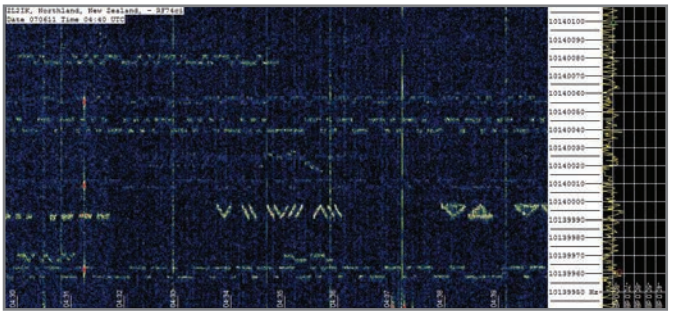


Photo 1—An example of a QRSS grabber image. This image represents 10 minutes of spectrum activity. Note that some signals are sent as slow Morse, but others are graphical where the transmitted signal varies in frequency to create symbols or text in the spectrum image. (Photo courtesy of Peter Mulhare, ZL2IK)

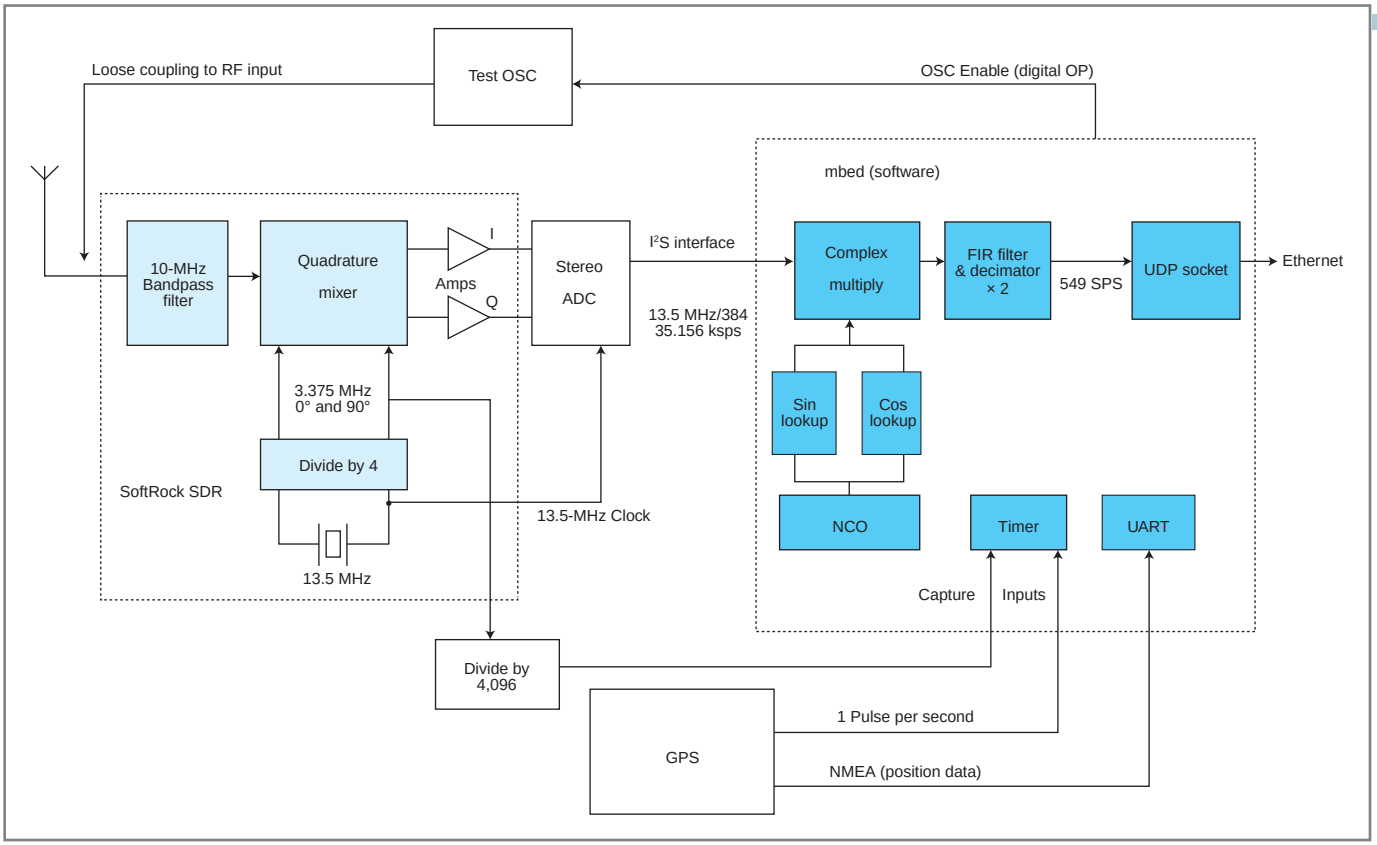


Figure 1—A block diagram of the QRSS-Rx. Signals come in the antenna, through the RF front-end, and are converted to digital. DSP software then processes the data before sending it out over Ethernet. A GPS module provides frequency calibration.

This network has a central server supported by a world-wide network of QRSS-Rx devices. The server manages the QRSS-Rx devices, accepts their data, and makes it available through a website as spectrum images. Spectrum data can therefore be presented in a synchronized and consistent manner, with a single webpage showing world-wide QRSS activity in almost real time.

The QRSS-Rx is the key to the construction of the QRSS network. The QRSS-Rx is small enough and inexpensive enough that it is likely to be more widely installed than the current grabber installation base. All that's needed for a QRSS-Rx receive station is the QRSS-Rx, an Internet connection, an antenna, and power. The data rate from a single QRSS-Rx is relatively low—currently about 35 kbps—but it could be reduced a lot lower than this. It is unlikely to be noticeable in a user's broadband capacity.

Once installed, the QRSS-Rx can operate completely automatically. Calibration, configuration, and even software updates can be managed from the server.

A grabber doesn't need to tie up expensive radio and PC equipment and won't require constant maintenance. This should encourage the grabber density to improve.

THE QRSS-Rx DESIGN

The prototype QRSS-Rx is designed to be an inexpensive, simple, and self-contained unit (see Figure 1). It is based on the mbed NXP Semiconductors LPC1768 prototype printed circuit board (PCB). The NXP LPC1768 has an ARM Cortex-M3 Core running at 96 MHz, 512-KB flash, 64-KB RAM and lots of interfaces including Ethernet, universal serial bus (USB), serial peripheral interface (SPI), integrated interchip sound (I²S), and other I/O. The mbed is supported by an online development system including the compiler and a many libraries.

The RF front-end is a SoftRock receiver. This is a software-defined radio receiver available as a kit from radio amateur Tony Park (callsign KB9YIG). The SoftRock does the initial down convert from RF frequencies (at a center frequency of 10.125 MHz in this case) to audio frequencies. This is a quadrature converter. There are two mixers happening—one with the local oscillator (LO) and the other with a phase-shifted (by 90°) LO. The two outputs are referred to as the I (in-phase) and Q (quadrature phase) outputs. By generating both of these signals further processing can be used to distinguish the two sidebands (i.e., signals above the LO in frequency and signals below the LO in frequency). Without the two outputs, the audio would be a mixture of both sidebands with no way to separate them.

The SoftRock actually performs another trick called undersampling. Normally, the LO should operate at the 10.125 MHz central frequency, which would mean an oscillator frequency of four times this (40.5 MHz) to generate the quadrature clocks. This frequency is getting a bit high for the simple SoftRock hardware. Since we have a band-limited input (the RF input includes a 10-MHz band-pass filter), we

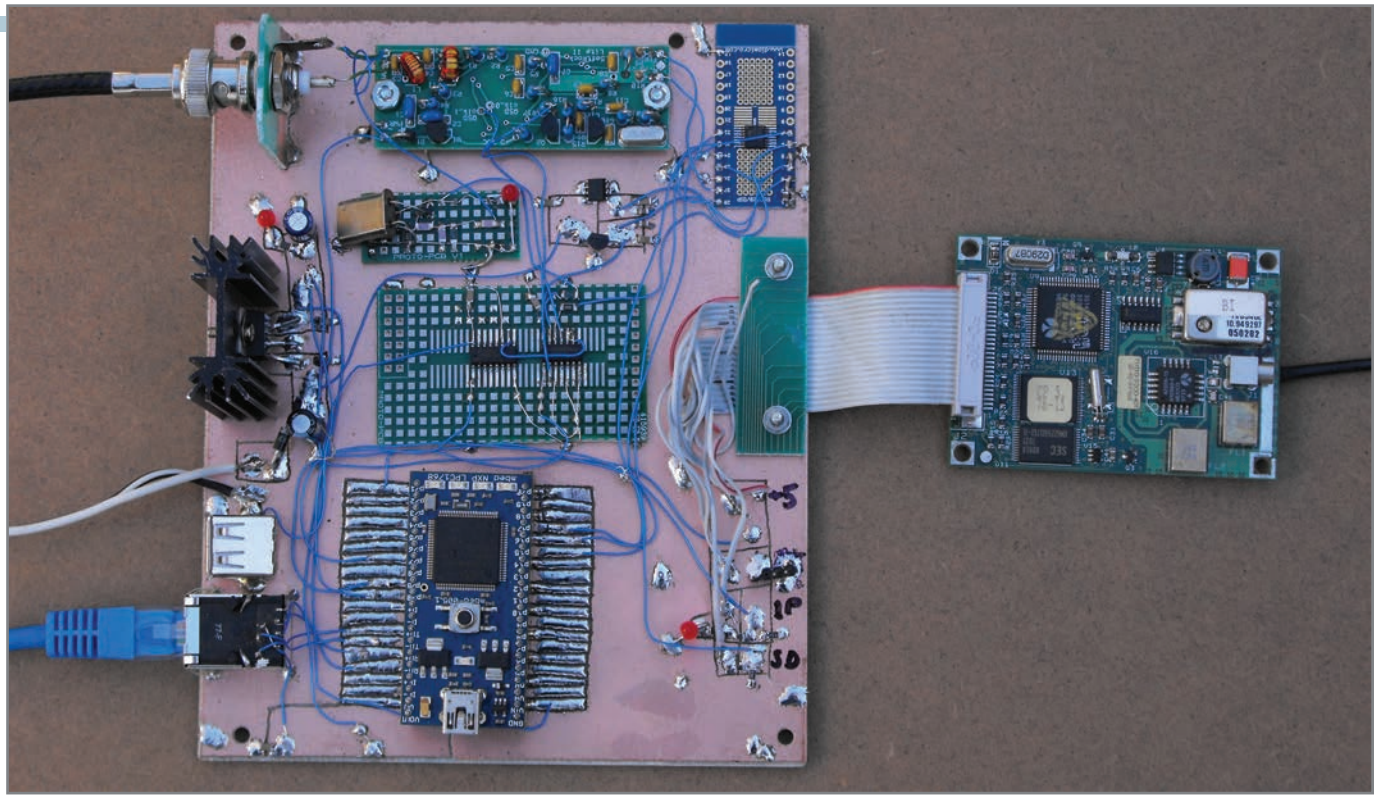


Photo 2—The QRSS-Rx construction makes use of prototype printed circuit boards (PCBs), manually cut PCB tracks, and direct wiring.

can instead use undersampling and operate the LO at one third of what would otherwise be required (3.375 MHz).

Normally, the SoftRock output signals would be fed into a stereo PC soundcard where software implements a normal radio receiver. In the case of the QRSS-Rx, the signals feed directly into an analog-to-digital converter (ADC) device. A 24-bit stereo ADC with an I²S interface is used, compatible with the mbed I²S port. Using 24 bits enables us to support signals over a large dynamic range. To avoid the need for another oscillator, the converter is clocked by the SoftRock's 3.375-MHz clock. The ADC sample rate is therefore 35.156 kilo samples per second (ksps). With this sample rate, the Nyquist sampling theorem tells us that we can capture signals up to 17.578 kHz (i.e., half the sample rate). With the quadrature down converter we can therefore capture signals within 17.578 kHz of the SoftRock central frequency of 10.125 Hz—or from about 10.107 MHz to 10.143 MHz. The primary QRSS band at 10.140 MHz is within this range.

Once the signal is sampled into the digital world, DSP takes over. Our frequency range of interest is from 10,140,000 Hz to 10,140,100 Hz—just 100-Hz wide. This is 15 kHz away from the 10.125-MHz central frequency. A DSP down convert operation is therefore performed. The I/Q data is treated as complex numbers (i.e., as a real and imaginary component). This enables the DSP to perform a frequency shift by doing a complex multiply between the signal samples and the software-generated oscillator (operating at 15 kHz) to bring the frequency band of interest down to 0 Hz. The down-converted output is then filtered with a software low-pass FIR filter having a cut-off frequency of about 200 Hz. Again, the Nyquist sampling theorem helps us. As we no longer have any signals above 200 Hz, only 400 samples per second are required. We can therefore decimate the samples by discarding samples above this Nyquist rate. The QRSS-Rx software actually decimates at a ratio of 1:64—only one sample per 64 is kept. This gives an output sample rate of 549 complex samples per second. The FIR filter uses 511 taps (i.e., it scales and sums up 511 samples); however, since most of the samples are discarded, it only needs to perform this calculation once for every 64 samples. Although this may still seem like a lot of calculations, the ARM processor running at 96 MHz easily performs it. The filter operation consists of 511 multiplies and 511 additions, 549 times per second, for each of the I and Q values. This is only about 1 million operations per second. The processed samples are then sent to the QRSS server over Ethernet using the user datagram protocol (UDP).

The prototype QRSS-Rx also includes a 10.14-MHz test oscillator. The CPU can turn this on and off, under instruction from the server. The oscillator output is loosely coupled to the RF input of the SoftRock by a wire loosely wrapped around the RF input. This oscillator is included to provide a signal to test the system performance.

KEEPING ON FREQUENCY

With such a small band of interest (100 Hz), frequency accuracy is important. The QRSS-Rx therefore includes an interface for a GPS receiver for calibration purposes. An inexpensive GPS module (eBay is your friend) provides GPS time

HUMANDATA®

FPGA / CPLD Boards from JAPAN

SAVING TIME & COST with readily available FPGA boards

- Basic and simple features, single power supply operation
- Quality and reliability is provided by years of sales
- Same board size and connector layout – ACM/XCM series
- All stocked items are ready to be shipped immediately
- Over 100 varieties of FPGA/CPLD boards are available
- Customizing speed grade and/or any features are possible
- Free download technical documents before purchasing
- High quality and highly reliable FPGA/CPLD boards from Japan
- Almost all products are RoHS compliance

ALTERA FPGA Board

Cyclone IV E F484 FPGA board
ACM-023 series

Cyclone IV E | MRAM

EP4CE55F23C8N
EP4CE75F23C8N
EP4CE115F23C8N
Credit card size (86 x 54 mm)

RoHS compliant



CycloneIV GX F484 FPGA board
ACM-024 series

Cyclone IV GX | DDR2 | SIF40

EP4CGX50CF23C8N
EP4CGX75CF23C8N
EP4CGX110CF23C8N
EP4CGX150CF23C8N
Credit card size (86 x 54 mm)

RoHS compliant





Cyclone IV E F484 FPGA board
ACM-107 series

Cyclone IV E | MRAM

EP4CE55F23C8N
EP4CE75F23C8N
EP4CE115F23C8N
Compact size (43 x 54 mm)

RoHS compliant



Cyclone III USB-FPGA board
EDA-004

Cyclone III | USB Config. | USB Comm. | MRAM

EP3C55F780C8N
Credit card size (86 x 54 mm)

RoHS compliant



XILINX FPGA Board

Spartan-6 FGG484 FPGA board
XCM-018/018Z series

Spartan-6 | MRAM | DDR2

XC6SLX45-2FGG484C
XC6SLX75-2FGG484C
XC6SLX100-2FGG484C
XC6SLX150-2FGG484C
Credit card size (86 x 54 mm)

RoHS compliant



Spartan-6 FGG484 FPGA board
XCM-110/110Z series

Spartan-6 | MRAM | DDR2

XC6SLX45-2FGG484C
XC6SLX75-2FGG484C
XC6SLX100-2FGG484C
XC6SLX150-2FGG484C
Compact size (43 x 54 mm)

RoHS compliant



Virtex-5 FFG676 FPGA board
XCM-011 series

Virtex-5 | FRAM | SDRAM

XC5VLX30-1FFG676C
XC5VLX50-1FFG676C
XC5VLX85-1FFG676C
XC5VLX110-1FFG676C
Credit card size (86 x 54 mm)

RoHS compliant



Virtex-5 LXT FFG665 FPGA board
XCM-017 series

Virtex-5 | SDRAM | RocketIO | SIF40

XC5VLX30T-1FFG665C
XC5VLX50T-1FFG665C
Credit card size (86 x 54 mm)

RoHS compliant





See all our products, download cables, A/D D/A conversion board, boards with USB chip from FTDI and accessories at :

www.hdl.co.jp/CC/

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: Core, 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



Spartan-6 | PLCC 68

XC6SLX16-2CSG225C
16Mbit Configuration Device
Two User LEDs
One User Switch(Slide)

RoHS compliant

XP68-02 Spartan-3AN PLCC68 FPGA Module



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



MAX II | PLCC 68

EPM240F100C5N
EPM570F100C5N
External Clock inputs

RoHS compliant

AP68-02 MAX V PLCC68 CPLD Module



MAX V | PLCC 68

5M570ZF256C5N
External Clock inputs
On-board Voltage regulator

RoHS compliant

AP68-03 Cyclone III PLCC68 FPGA Module



Cyclone III | PLCC 68

EP3C10U256C8N
4Mbit Configuration Device
Two User LEDs
One User Switch(Slide)

RoHS compliant

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

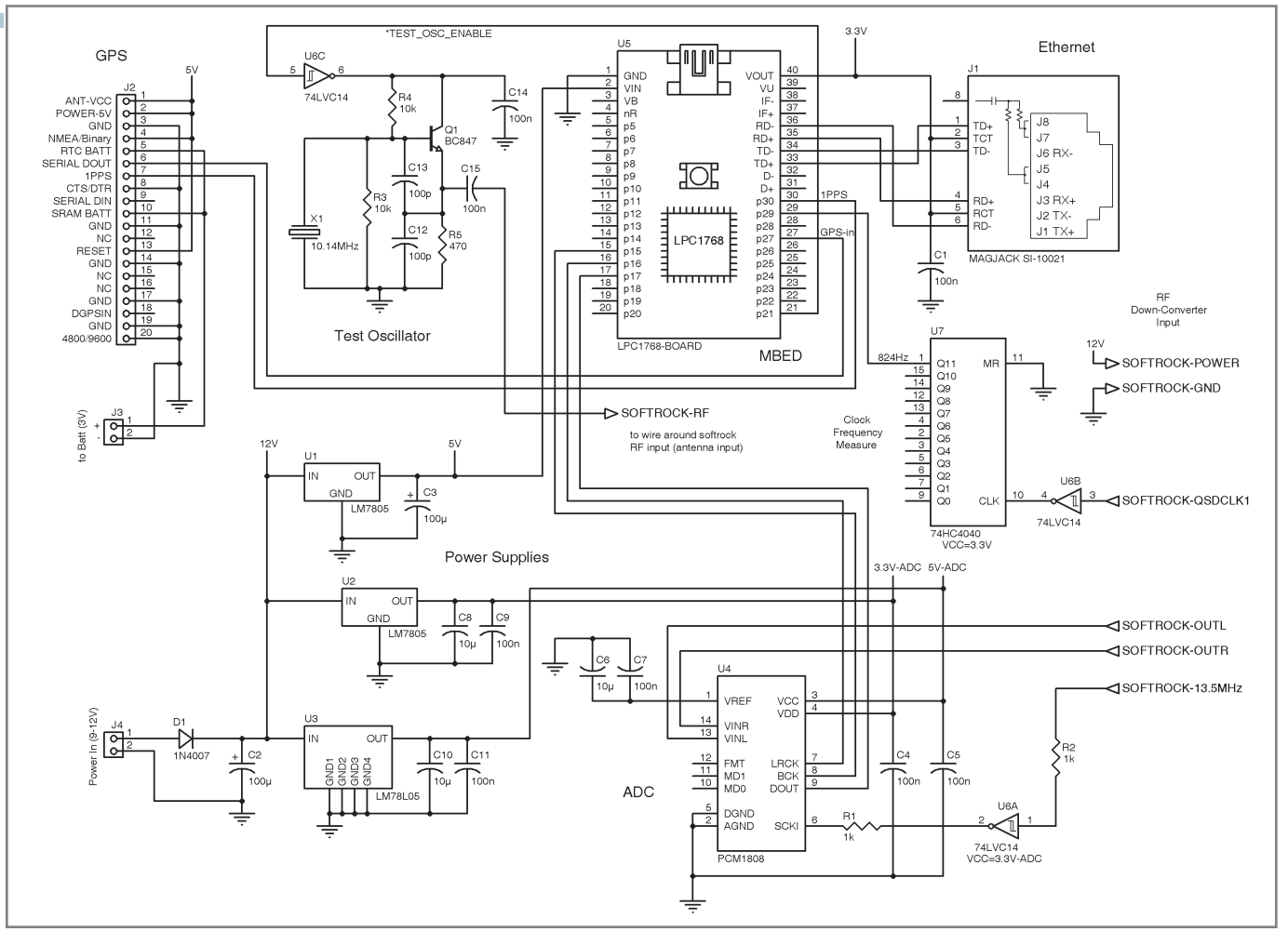


Figure 2—The QRSS-Rx circuit diagram

and location over a serial port and a precise timing pulse as a one pulse per second (1 pps) signal.

The QRSS-Rx software parses the serial port data to extract time and position information. The GPS 1-pps signal is connected to the LPC1768 internal timer capture input to capture the timer value at the 1-pps pulse. With this data, we can calculate the exact operating frequency of the CPU clock (96 MHz). In a similar way, the SoftRock LO frequency (3.375 MHz) is measured. The LO is divided by 16,384 (4 in the SoftRock itself, and 4,096 in a counter chip) to about 824 Hz and then fed to a CPU capture input. The software is interrupted every second by the 1 pps, and stores away the captured timer value for the 1 pps and the last LO capture. These values are passed up to the server where they can be used to determine the CPU clock frequency and thus the SoftRock LO frequency. Even though only every 824 LO captures are recorded (and captured asynchronously, so there may in fact be only 823 cycles), we can make assumptions on the clock accuracy and still get an accurate measure of the SoftRock LO frequency.

The server performs the necessary calculations to determine the CPU frequency of the QRSS-Rx and the true operating frequency of the SoftRock LO. Since the ADC is clocked from the SoftRock LO, the exact sample rate can also be determined. The server can then calculate the exact value for the DSP software oscillator to tune to the desired receive frequency. This setting is passed to the QRSS-Rx in a UDP message.

The GPS module also reports position. This is useful additional information for the QRSS server to position the receiver on the planet.

BUILD IT!

The QRSS-Rx prototype is assembled on a piece of blank double-sided PCB (see Photo 2). The PCB was hand-routed using a rotary tool and a dental bit to cut islands of copper in the board. Islands are cut for the mbed socket—a small pad for each mbed pin. Additional islands are cut for some of the power-supply circuits and the GPS connections. Other components are mounted on various prototype PCBs. All interconnections are then made using thin wire-wrap wire. The SoftRock RF PCB is mounted on the PCB and directly wired as necessary into the circuit.

The power regulator requires a heatsink. This is bolted down to the PCB and the regulator is mounted to it. A separate power supply (5 V and 3.3 V) is implemented for the ADC device and clock buffer to minimize noise into these devices.

The GPS module used has a 20-way

Poll Message—sent every six seconds from the QRSS-Rx to the server		
Offset	**Name**	**Details**
0	Poll message ID	10
4	Message protocol version	1
8	Hardware ID identifying the hardware structure type	1
12	Software version—the version of QRSS-Rx software	1
16	GPS quality identified	0 = offline, 1 = online
20	GPS satellite count	
24	GPS Time	seconds since midnight UTC
28	Latitude	millionths of a degree units, Negative = south
32	Longitude	millionths of a degree units, Negative = west
36	Altitude	meters
40	Nominal CPU reference clock	96,000,000
44	Nominal RF mixer clock	10,125,000
48	Nominal sample clockrate	13,500,000
52	Output sample rate divider (from sample clockrate)	24,576 (i.e., 384 × 64) ADC divider and decimation rate
56	1-pps capture values	The last eight capture timer values newest to oldest
88	Divided RF mixer/sample clocks capture values	The last eight capture timer values newest to oldest
120	QRSS-Rx unique name	10 ASCII characters
Command Message—sent from the server to the QRSS-RX in response to a poll message		
Offset	**Name**	**Details**
0	Command message ID	11
4	Message protocol version	1
8	Command code	0 = stop, 1 = start
12	DSP Oscillator rate	Numerically controlled oscillator (NCO) phase increment value
16 – 19	Test oscillator state	0 = off, 1 = on
Data Message—sent from the QRSS-RX to the receiver when it is running (started)		
Offset	**Name**	**Details**
0	Data message ID	12
4	Timestamp	CPU timer value at first sample
8 – 519	Data – 64 samples	32-bit Q data followed by 32-bit I data, signed (two's compliment), little endian

Table 1—The three messages are used. All fields are 32-bit words sent in network byte order (high byte first). Note the use of a Version field to enable the message protocol to be adapted but for the server to still easily identify the message type.

pin header. The connection to this is made with a length of ribbon cable and a header plug. The ribbon cable wires are split, stripped, and connected to PCB islands which are then connected to the signals required. A GPS antenna connects directly to the GPS module.

The Ethernet connection (which includes the Ethernet magnetics) is secured to the PCB by soldering it down upside down. Its pins are then direct wired to the mbed. A USB connector is also added to the PCB and connected to only five volts. It is there to power an Ethernet-to-Wi-Fi adapter that was used during testing (see Figure 2).

COMMUNICATIONS

QRSS-Rx-to-server communications use UDP datagrams. A very simple custom message protocol was developed.

The QRSS-Rx is hard coded with the server DNS name. A DNS lookup is used to translate from the server name to its IP address. Every 6 s the QRSS-Rx sends a "poll" message to the server. This message identifies the QRSS-Rx device and passes GPS data, captured timing registers (for the calibration of the CPU and LO clocks), and some information about the QRSS-Rx hardware. This additional information is included, ready for when there are other designs of QRSS-Rx devices operating on different frequency bands.

The server may then command the QRSS-Rx to start receiving. In this command, it also includes the configuration for the DSP oscillator, as calculated using the calibration information from the QRSS-Rx. The command also includes an option to enable or disable the test oscillator.

Once commanded to start, the QRSS-Rx sends processed sample data packets. Each packet contains 64 samples and a timestamp. The timestamp is the CPU timer value at the first sample of the packet. Since this timer is the same as is used for the calibration data, it enables the server to precisely synchronize the samples to real time (see Table 1).

THE SOFTWARE

The software is written in C++. Extensive use is made of the libraries from the mbed online compiler. The mbed library provides access to the basic ports (serial ports, digital IO) and runs software timers. The NetServices library handles the Ethernet and UDP communications. A DNSResolver library is used for DNS lookup, to determine where to send the poll message to.

A set of buffer classes was developed to handle the data samples. Each sample consists of a pair of ADC values—the I value and the Q value. The buffer system defines a sample-set

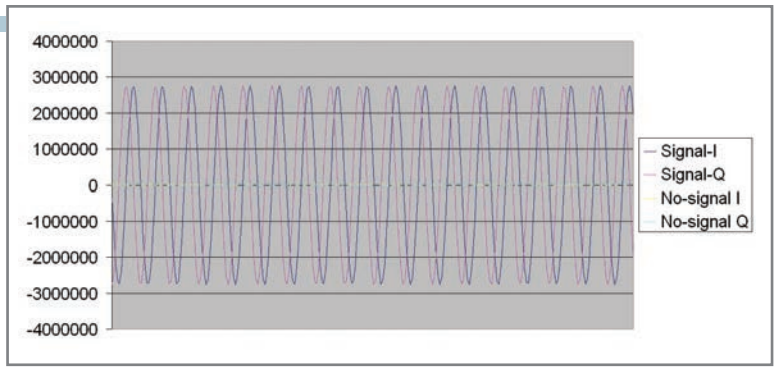


Figure 3—Graphed sample data received by the server showing samples with the test oscillator enabled and disabled

buffer class to hold a set of sample data. It has a buffer-handler class, which can "hold" a sample-set buffer object. Also implemented is a buffer-queue class that can be used to queue up sets of sample data. Finally, a buffer-pool class is defined. This is a queue for unused buffers—new buffers are created from a pool. When a buffer is done with, it is automatically returned to the pool.

The buffer system is set up for a sample-set size of 512 samples. Since each sample is 8 bytes (two 32-bit words) each buffer is 4-KB long. There is only room for four buffers in the system, but this seems to be enough and can cope with the various delays in the system processing.

The ADC interface uses an I²S interface which is supported by the mbed hardware. An I²S driver needed to be written—the I2SReceiver class. It uses DMA transfers to read I²S data into a sample-set buffer object. As each buffer is filled, the DMA interrupt occurs. The buffer is then queued and a new one is obtained from the pool for more data.

A GPS controller class handles the data from the GPS serial port, and performs the 1-pps capture operation. The serial data from the GPS hardware is parsed to extract the GPS status, number of satellites, position (latitude, longitude, and altitude), and time.

The communications class handles all communication over the Ethernet. It manages the Ethernet initialization and the sending of the UDP poll messages to the named server. It accepts commands from this server, along with parameters, to start and stop the receiver. When started, it handles the transfer of sample sets to the server.

Sample processing is performed by the DSP class. This class is inherited from the buffer-handler class. It adds two functions to the handler to perform DSP operations on the sample-set buffer held—a mixer function and a filter function.

The mixer function is to mix a software oscillator with the samples. The oscillator is a numerically controlled oscillator (NCO), which, with each sample, adds a phase value (this determines the NCO frequency) into a phase accumulator. The accumulator is then used to look up a table and obtain a cosine and sine value. This value pair is treated as a complex number, and is multiplied (as a complex multiply operation) with the data sample. This has the effect of frequency shifting the samples. The server sets the frequency of the NCO so that the band of interest is shifted close to zero frequency.

The filter function performs a low-pass filter (LPF) and decimator (reduction in samples) operation. A FIR filter is used. The filter is applied to the I and the Q data separately. It sums up 511 scaled samples (each one scaled based on the filter coefficients) to produce one output. As the samples are then decimated 1:64, this operation only needs to be performed on every 64 samples—that sample being the result of 511 other samples. The coefficients for this filter were obtained from an online calculator.[1] The DSP class output is an array of eight sample sets for each data buffer processed. This size is derived from the buffer size—512 samples with 64:1 decimation, giving eight output samples. The DSP class maintains the interim FIR filter totals between one sample set and the next.

A C "main" routine pulls all the classes together. It runs the main processing loop which handles the transfer of data between the ADC receiver, through the DSP processing, and to the communications module. It also handles the debug console (the serial port), accepting keystrokes and executing the appropriate test routine.

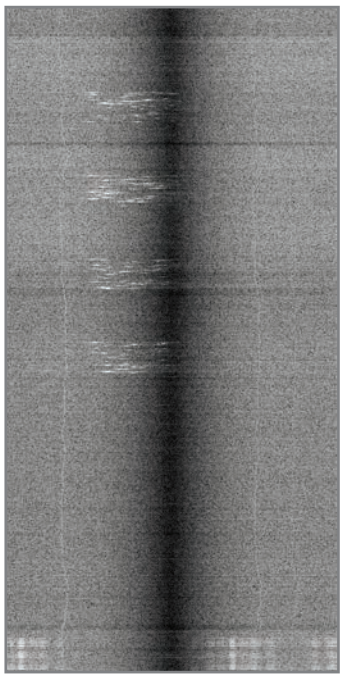


Photo 3—A server-generated spectrum image of an incoming signal

TESTING

This article is about the QRSS-Rx device, and not about the server it connects to. However, for testing purposes, a server was required. A simple implementation of the server was developed in C in a "cygwin" environment. Initial testing involved the server simply receiving and outputting sample data that could then be captured and processed separately. The results in Figure 3 show a graph of captured data with the test oscillator operating which was then graphed in Excel.

For further testing, the wfft library was included in the server application to perform the FFT processing to generate frequency data from the received samples. The resulting data was then used to build up a bitmap image that could then be displayed (see Photo 3).

NEXT STEPS

The mbed environment proved to be an excellent platform for this project,

being a good match in the peripherals and processing power available. The libraries made the task a lot simpler, as effort could be put into the core task rather than needing to code Ethernet stacks and the like. The CPU's processing power proved to be useful, enabling the FIR filters to be operated easily while still performing the necessary communications and GPS handling.

The QRSS-Rx concept is now proven. The next steps are hardware development for easy reproduction and the creation of the QRSS network server. Watch this space!

Originally from New Zealand, Clayton Gumbrell (clayton@gumbrell.net) now lives in Canberra, Australia, where he works as a Senior Design Engineer with Locata Corporation (www.locatacorp.com). His interests include amateur radio (callsign VK1TKA).

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/259.

REFERENCE

[1] T. Fisher, "Interactive Digital Filter Design," www-users.cs.york.ac.uk/~fisher/mkfilter.

RESOURCES

KnightsQRSS Blog, "Getting Started with QRSS," 2010, http://knightsqrss.blogspot.com/2010/01/getting-started-with-qrss.html.

mbed, http://mbed.org.

Tony Parks, KB9YIG, www.kb9yig.com.

SOURCES

Cortex-M3 Processor
ARM Ltd. | www.arm.com

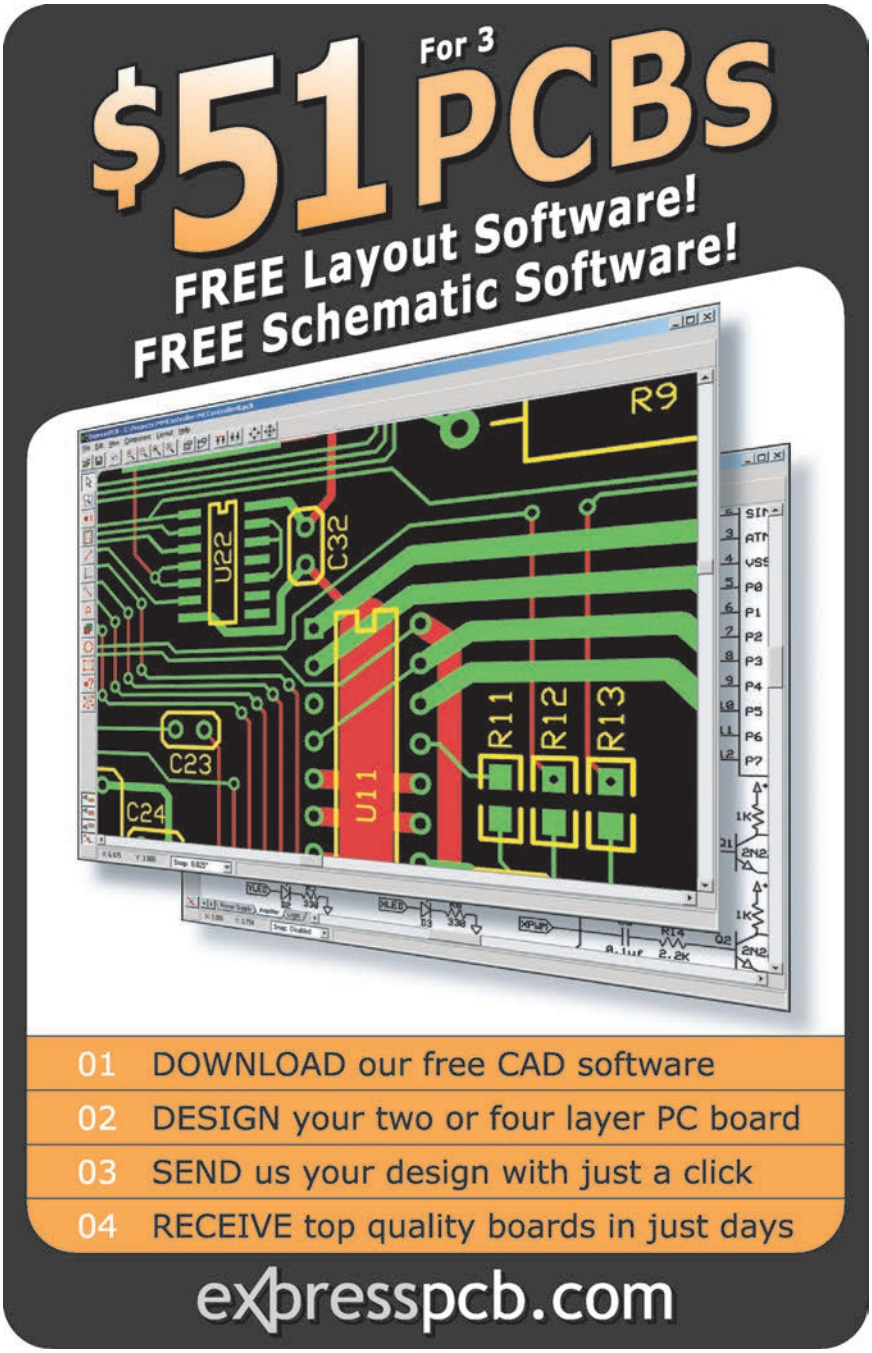
LPC1768 Microcontroller
NXP Semiconductors | www.nxp.com

NEED-TO-KNOW INFO

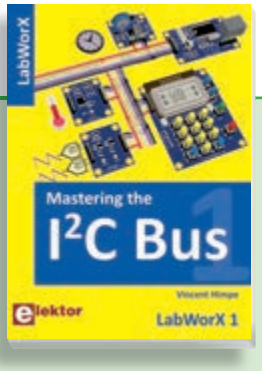
Knowledge is power. In the computer applications industry, informed engineers and programmers don't just survive, they *thrive* and *excel*. For more need-to-know information about some of the topics covered in this article, the *Circuit Cellar* editorial staff recommends the following content:

Signal Generation Solution
Build an Inexpensive RF Signal Generator
by Neal Martini
***Circuit Cellar* 182, 2005**
Tired of going to a local university lab to use a signal generator, Neal designed his own PIC16F877A-based controller and RF module. Topics: RF, Signal Generator, Impedance Matching

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



Books



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

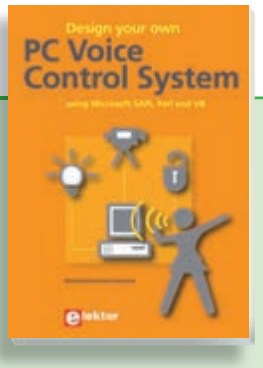


Creative solutions for all areas of electronics

311 Circuits

This book contains circuits, design ideas, tips and tricks from all areas of electronics: audio & video, computers & microcontrollers, radio, hobby & modelling, home & garden, power supplies & batteries, test & measurement, software, not forgetting a section 'miscellaneous' for everything that doesn't fit in one of the other categories. 311 Circuits offers many complete solutions as well as useful starting points for your own projects. Both categories and anything in between represent a veritable fountain of inspiration for cultivating your own ideas and learning about electronics.

420 pages • ISBN 978-1-907920-08-0 • $47.60



Talk with your computer

Design your own PC Voice Control System

This book guides you through practical speech recognition, speech annunciation and control of really useful peripherals. It details a project which will enable you to instruct your computer using your voice and get it to control electrical devices, tell you the time, check your share values, get the weather forecast, etc. and speak it all back to you in a natural human voice. If you are interested in the practical technology of interfacing with machines using voice, then this book is your guide!

216 pages • ISBN 978-1-907920-07-3 • £29.50

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

Kits & Modules

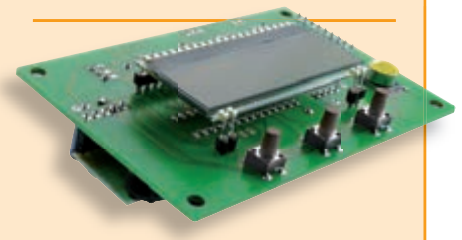


Improved Radiation Meter

This device can be used with different sensors to measure gamma and alpha radiation. It is particularly suitable for long-term measurements and for examining weakly radioactive samples. The photodiode has a smaller sensitive area than a Geiger-Müller tube and so has a lower background count rate, which in turn means that the radiation from a small sample is easier to detect against the background. A further advantage of a semiconductor sensor is that is offers the possibility of measuring the energy of each particle, allowing a more detailed investigation of the characteristics of a sample. The optional PC-based software displays the energy spectrum, permitting a very detailed analysis to be carried out.

Kit of parts incl. display and programmed controller

Art.# 110538-71 • $57.30



USB Long-Term Weather Logger

This stand-alone data logger displays pressure, temperature and humidity readings generated by I²C bus sensors on an LCD panel, and can run for six to eight weeks on three AA batteries. The stored readings can be read out over USB and plotted on a PC using gnuplot. Digital sensor modules keep the hardware simple and no calibration is required.

Kit of parts incl. PCB, controller, humidity sensor and air pressure sensor modules

Art.# 100888-73 • $50.20

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

Elektor Shop

The world of electronics at your fingertips!

Bestseller!



This DVD and more are available at www.elektor.com/shop

CD/DVD-ROMs

Packs 1.2 gigabytes worth of tools and documents

DVD Wireless Toolbox

On this DVD-ROM you'll find a number of technical documents and tools that will enable you to add wireless data exchange to your electronics systems. The choice of equipment depends on the transmission distance: a few centimetres using Near Field Communication (NFC) or Radio Frequency Identification (RFID), tens of metres with the Bluetooth, Wi-Fi or ZigBee systems, or indeed thousands of kilometres using a module for receiving GPS data. In accordance with the principle of our Toolbox series, we've brought together technical documentation (spec. sheets, application notes, user guides, etc.) on various devices according to the frequency and/or protocol used. All of the documents are PDF files (in English). Browsing around the DVD is made easy by an HTML menu. Finally, this Wireless Toolbox DVD contains a collection of articles on this topic (RFID, xBee, DCF77, GPS, infrared, etc.) that have appeared in Elektor magazine.

ISBN 978-90-5381-268-6 • $46.00



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



Circuits, ideas, tips and tricks from Elektor

CD 1001 Circuits

This CD-ROM contains more than 1000 circuits, ideas, tips and tricks from the Summer Circuits issues 2001-2010 of Elektor, supplemented with various other small projects, including all circuit diagrams, descriptions, component lists and full-sized layouts. The articles are grouped alphabetically in nine different sections: audio & video, computer & microcontroller, hobby & 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

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

by Alexander Pozhitkov (USA)

The NakedCPU (Part 1)

Hardware Experiments and a Roadmap for Navigating Documentation

The first part of this two-part series provides an overview of the NakedCPU. This platform is designed to provide full access to hardware and a CPU without any operating system restrictions while working in the protected mode.

Electronic projects involving microcontrollers are popular among design enthusiasts and professionals alike. Many interesting applications have been made with microcontrollers, and programming them can be a lot of fun. At the same time, the powerful central processing unit (CPU) found in the personal computer (PC)—which serves every electronics designer on a daily basis (including microcontroller programming)—is lacking such attention. Most experimentation with a PC is limited to developing high-level code software with the aid of numerous libraries and technologies hiding the hardware beneath layers and layers of code. Unlimited experimentation with PC hardware is rarely possible. However, you have to install drivers, enabling some access to hardware, because the operating system (OS) naturally does not permit us to do any low-level activities. The sad part is that such drivers are mysterious themselves. It is safe to say hardware programming was well known to many computer professionals and enthusiasts in the 1980s. Later, many people forgot about it, while the technology tremendously leapt ahead. In this article, I try to bridge the gap in time and revive interest in hardware programming based on state-of-the-art technologies and concepts. There is a Russian saying, "Everything new is actually well-forgotten old."

WHAT IS THE NakedCPU?

This article is a result of my interest in the Intel CPU, chipset, I/O controller, and other essential PC devices from the perspective of low-level hardware programming unobstructed by an OS and drivers. My motivation was to reach out to people with inquisitive minds who would appreciate the possibility to directly experiment with the CPU, chipset, and other hardware. Here I'll present the NakedCPU, which provides full access to hardware and a CPU without any restrictions imposed by the OS. Importantly, the processor isn't obscured by Linux, DOS, or Windows, and is operating in its most interesting and powerful mode—the protected mode. In this article, the users are referred to as inquirers, because the NakedCPU is made for researchers (i.e., devoted geeks) rather than regular users.

My aim is to provide you inquirers with some help navigating hardware documentation, which is confusing and otherwise difficult to find. I won't restate the documentation because many computer concepts and technologies quickly become obsolete. With this article, however, you'll find it easier to follow the newer technologies and documentation.

PROCESSORS & PLATFORMS

Think for a moment about one of the modern Intel CPU varieties: the Intel Core 2 Duo processor. This impressive

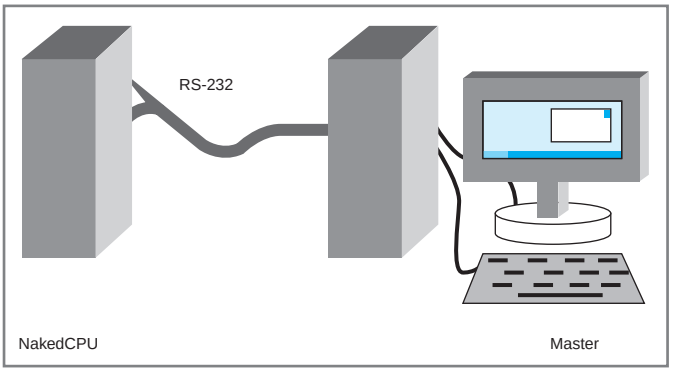


Figure 1—The NakedCPU and master computers are connected via a serial interface. The NakedCPU does not need a keyboard or a monitor.

Description	MBR Location
Determining the current address while the processor is still in real mode after power on. BIOS has loaded the MBR somewhere into the memory and transferred control to our code. The current address is necessary to locate physical address of the pseudo-descriptor, which is in turn defining a physical address and a limit for the Global Descriptor Table (GDT).	0x3E–0x4D
LGDT instruction (Load GDT register) is loading pseudodescriptor, which is pointing to GDT.	0x52
GDT and Interrupt Descriptor Table (IDT) are copied into a new memory location, beginning from linear address 0x0. GDT and IDT are defining memory segments for the processor to operate in protected mode.	0x57–0x64
The MBR contains a very tiny 32-bit protected mode "operating system" named the NakedOS.	0x80–0x186
The NakedOS is copied into a new memory location beginning with linear address 0x800.	0x65–0x71
Switching into protected mode is accomplished by adjusting the machine status word using a LMSW instruction.	0x72–0x78
Transfer control to the NakedOS.	0x7B
Set up 8259 interrupt controller.	0xF5–0x105
Transfer control to the inquirer's executable.	0x106

Table 1—Anatomy of the MBR

processor is capable of consuming up to 75 A of current![1] It is not a simple processor. Its documentation consists of five volumes with the total page count of approximately 3,800 pages.[2] The Intel CPU does not operate alone. It is interfaced to a chipset known as a graphics and memory controller hub (GMCH). The chipset on the other side is connected to an I/O controller hub (ICH). Interestingly, this arrangement is analogous to your nervous system with a brain, a brainstem, and a spinal cord. GMCH and ICH are processors themselves, containing hundreds of configuration and control registers. The documentation on GMCH and ICH spans more than 1,400 pages.[3, 4] It's no wonder OSes hide actual hardware under a thick blanket of intermediate code!

It is not easy to experiment with the Intel CPU given the complexity of surrounding hardware, such as the chipset, network controller, and so forth. Another difficulty is the fact that the documentation is filled with electrical engineering abbreviations and concepts. Also, pervasive layers of OS code interfere with truly free exploration.

The NakedCPU is an experimental platform exposing the hardware internals of a PC. Experimentation with NakedCPU requires two computers (see Figure 1). One is the master computer with Windows and Visual Studio software, whose job is interacting with us and the second computer. The second computer—the NakedCPU—is connected to the master via an RS-232 interface.

The NakedCPU computer is booted up with a small amount of start-up code (available on the *Circuit Cellar* FTP site), which enables it to communicate via RS-232 with the master. Upon start-up, the NakedCPU expects two separate packages of bytes: one is a stream of Intel CPU opcodes to be executed (i.e., the executable) and the other one is the data to be processed. The executable can modify any part of memory, chipset registers, and so forth, and even overwrite the start-up code. In other words, the freedom is yours.

STARTING THE NakedCPU

The NakedCPU won't run without some sort of a start-up code. At start-up, two tasks must be accomplished: switch the CPU into the Protected mode and begin listening on the serial port for two packets of bytes, executable and data. The easiest way to supply the NakedCPU with a start-up code is to prepare a bootable floppy disk with your own code. Certainly, one can also put this code onto a hard drive.

The start-up code (up to 512 bytes) was written in Assembly language and must be stored in sector 0—that is, the master boot record (MBR) of the disk. The Assembly language compilers and linkers, such as MASM, were not used to prepare the MBR due to various difficulties, although they may be suitable as well. There is, however, a binary editor, called HexIt, which among other things enables direct conversion of Assembly commands into binary code.[5] Using this editor, a binary file of the future MBR was created. The file's content is available on the *Circuit Cellar* FTP site. Refer to Table 1 and Table 2 for more details about the content.

A small utility, Firstsectwrite.exe (also available on the *Circuit Cellar* FTP site), was written to transfer this file into sector 0 of the disk. Although the code of this utility is quite simple, it deserves some attention. A Windows API call to `CreateFile(TEXT("\\\\.\\A:")...)` opens raw communication with a disk—a floppy drive A in this case—to enable writes into the sector 0. It is important to note that this call will be only be successful under the administrator account.

I used a Dell Optiplex 760 computer to conduct the experiments. It had a floppy drive attached via a USB. BIOS start-up options enabled me to boot up the computer from such a drive.

THE NakedOS

It may sound contradictive to the spirit of the article to be OS-free; however, the NakedCPU is booted up with a tiny (262 bytes long) 32-bit "operating system," the NakedOS, which enables the NakedCPU to communicate with the outside world via a serial port. In fact, I did not compromise my principle of truly free exploration, because the NakedOS

Structure	MBR Location
Pseudo-descriptor IDT	0x194
Pseudo-descriptor GDT	0x1BA
Null descriptor	0x1C0

Table 2—Critical data structures

Segment	Base	Size	Descriptors, type
Extended memory	0x100000	~128 Mb	0x28, data
Screen, character mode	0x0B8000	4 Kb	0x20, data
Target executable	0x93B	64 Kb	0x30, code 32 0x38, data
NakedOS	0x800	315 bytes	0x10, code 32
Stack	0x400	1024 bytes	0x18, stack 32
System data IDT: 0x3FF–0x200 GDT: 0x1FF–0	0x0	1,024 bytes	0x8, data
Interrupts	**Info**		
INT 0x20	Read a packet from serial port; destination ES:[EDI]; mandatory condition DS=ES. First 4 bytes of the packet indicate in bytes the length of the subsequent string. Upon return, ECX contains the number of bytes received.		
INT 0x21	Send to serial port a string of ECX bytes, located at DS:[ESI].		
IRQ0	Hardware interrupts base vector is 0x28.		

Table 3—Memory segments and interrupts defined by the NakedOS

is absolutely transparent. Its code is available on the *Circuit Cellar* FTP site.

The NakedOS defines several memory segments, which are useful as an initial environment for the inquirer's executable (see Table 3). Intel documentation provides an explanation for protected mode memory segments, global descriptor table (GDT), and interrupt descriptor table (IDT).[2] In addition, the NakedOS defines two software interrupts and a base vector for hardware interrupts. Note that the IDT interrupts have nothing to do with DOS or BIOS; they are solely defined by the code.

Immediately after start-up, the NakedOS expects two transactions: one for the executable code and another for data. Each transaction is a stream of bytes sent via the

Byte index	0	1	2	3	0	1	2	…	N-1
Description	Length, N				Executable code or data				

Table 4—The format of NakedOS transactions. The first 4 bytes indicate the length of the subsequent byte stream.

Get PUBLISHED. Get NOTICED. Get PAID.

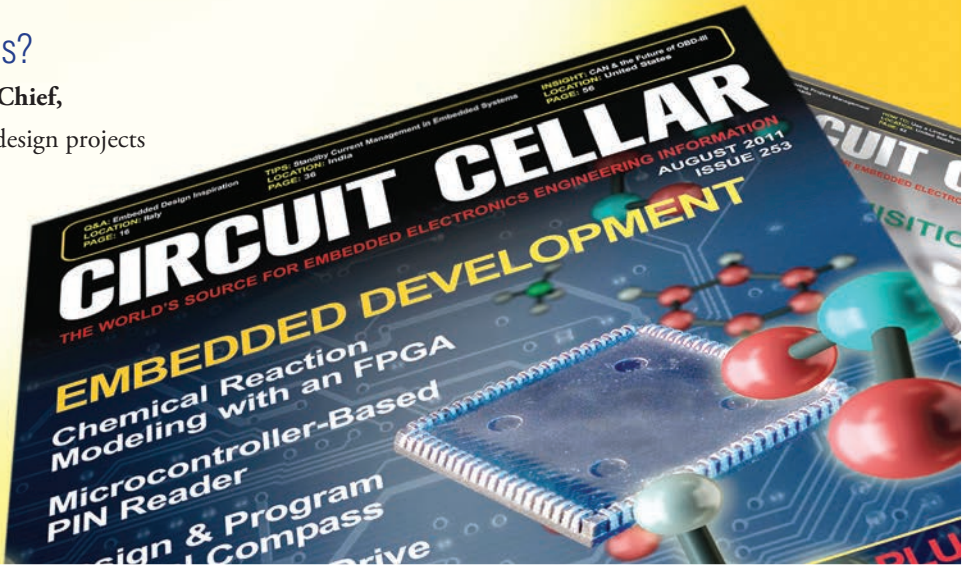
Circuit Cellar feature articles are contributed by professional engineers, academics, and students from around the globe. Each month, the editorial staff reviews dozens of article proposals and submissions. Only the best make it into the pages of this internationally respected magazine.

Do you have what it takes?

Contact **C. J. Abate, Editor-in-Chief,** today to discuss the embedded design projects and programming applications you've been working on and your article could be featured in an upcoming issue of *Circuit Cellar* magazine.

editor@circuitcellar.com

CIRCUIT CELLAR



Listing 1—A fragment of a constructor for a class derived from NakedCPUcode

```
...
DWORD pe, ps;
__asm
{
       mov pe, offset end          // mandatory label end
       mov ps, offset start        // mandatory label start
       jmp end                     // master jumps over the code
       ////////////////////////
       start:      mov ax, 0x28    // loading
                   mov es, ax      // data and
                   mov ds, ax      // stack
                   mov ax, 0x18    // segment registers
                   mov ss, ax      // initializing
                   mov esp, 0x3fe  // stack pointer
                   //...
                   //here goes executable code
                   //...
                   _emit 0xEA  // this is
                   _emit 0x00  // a long jump to
                   _emit 0x00  // finish the executable
                   _emit 0x00  // and transfer control
                   _emit 0x00  // to NakedOS
                   _emit 0x10  // (if it is still
                   _emit 0x00  // in the memory)
       end:        nop
}
...
```

RS-232 (see Table 4). The first transaction is written into the memory segment "target executable," while the second transaction goes into the "extended memory" segment. After the second transaction, the NakedOS transfers control to the executable by a long jump: `jmp 00030:000000000`. From that moment, in principle, any memory occupied by the NakedOS can be overwritten by the activities of your executable.

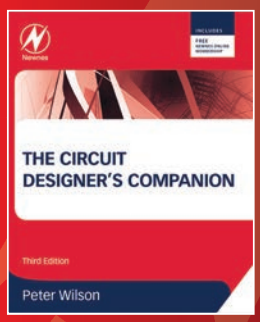
The hardware interrupts are normally masked when the NakedOS is running; however, the 8259 interrupt controller is set up (refer to the *Circuit Cellar* FTP site) to handle them if you decide to unmask them. Detailed instructions on programming the interrupt controller are provided in the documentation for the ICH.[4]

THE NakedCPU EXPLORER

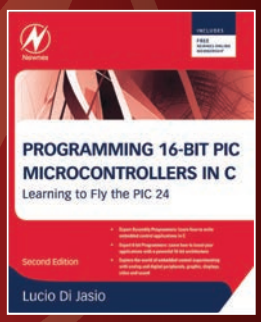
An important issue remains: sending an executable to the NakedCPU to conduct experiments. At the beginning of this article, I said two computers are involved. The master has a Microsoft Visual C++ project, the NakedCPU Explorer, which acts as a "shell" that enables the inspection and modification of chipset registers and memory. The code defines a class having a constructor, which provides `__asm{}` brackets for you to fill with executable code (see Listing 1).

Newnes Press

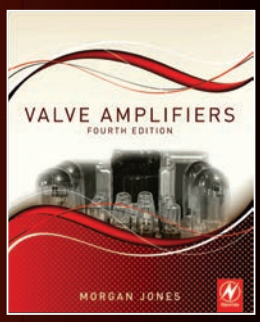
Education Never Ends. Everything you need to know to get started.



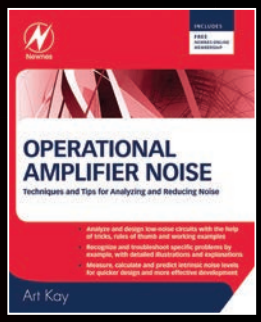
By: Peter Wilson
ISBN: 9780080971384



By: Lucio Di Jasio
ISBN: 9781856178709



By Morgan Jones
ISBN: 9780080966403



By: Art Kay
ISBN: 9780750685252

Save 30% on all titles when you purchase from www.newnespress.com.
Enter promotional code Newnes30 at checkout.

Newnes

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!

"Immediately after start-up, the NakedOS expects two transactions: one for the executable code and another for data. Each transaction is a stream of bytes sent via the RS-232. The first transaction is written into the memory segment 'target executable,' while the second transaction goes into the 'extended memory' segment. After the second transaction, the NakedOS transfers control to the executable by a long jump: `jmp 00030:000000000`."

Since Visual C++ is running on the master PC with an Intel CPU, the compiler translates the Assembly code into appropriate opcodes, which are naturally suitable for the NakedCPU! Specifically, this class is derived from another class, `NakedCPUcode`, which performs preparatory work by extracting the opcodes produced from the code in the `__asm{}` brackets and making them available for sending over to the NakedCPU. Note that the NakedCPU only receives the code between `start` and `end` labels. It is important to understand that the master computer will not execute the code in the `__asm{}` brackets; it simply jumps over it. The strange keyword `_emit` enables the direct placement of opcodes by their hexadecimal values. For some reason, a long jump is not permitted when using the Visual Studio compiler.

The project also defines a class `SerialComm` and a function `SendNakedCPUdataRecvResponse` to send and receive data. It is worthwhile to examine the project's straightforward code to understand the details of communication with the NakedCPU. Besides serving as an example, the NakedCPU Explorer sends an executable to the NakedCPU, which permits the interactive examination and modification of various chipset and I/O controller registers. The NakedCPU Explorer offers eight commands: `write`, `write32`, `read`, `read32`, `pci`, `memread`, `memwrite`, and `quit`. The first four commands will ask for a port address (i.e., an address in the CPU I/O space). With these commands, the NakedCPU will write to and read from a GMCH or ICH register, 1 or 4 bytes. The fifth command will ask for Bus (decimal), Device (decimal), Function (decimal), and Register (hexadecimal) values. Their values will be packed into the port 0xCF8 to open a "window" into the PCI configuration space that's accessible via port 0xCFC. Details on addressing PCI devices are provided in the chipset documentation.[3] The `memread` and `memwrite` commands enable the reading and writing of double words from and to the memory, respectively.

A regular PC, ubiquitous in most homes, is filled with powerful and interesting hardware. Unfortunately, it tends to be difficult to experiment with PCs due to the lack of documentation and overly protective OSes. The first part of this article detailed at the path to the hard-to-find documentation. I also described the NakedCPU, which is my OS-free platform for experimenting with a PC's internals.

NEW EXPERIMENTS

The NakedCPU is controlled from another computer, the master, which provides you with an interface. In the next part of this series, I'll describe how to use the NakedCPU Explorer for experiments with the speaker, parallel port, and LAN adapter. In addition, I'll give you a peek at the BIOS—the power-on code in particular. Undoubtedly, the suggested experiments will be stepping stones to help you begin even more interesting research.

Author's note: The NakedCPU Explorer does not use any hidden "helper" drivers or libraries. The code is entirely transparent for the inquirer's perusal.

Dr. Alexander Pozhitkov (pozhit@uw.edu) has an MS degree in Chemistry and a PhD in Genetics from Albertus Magnus University in Cologne, Germany. He has been working for 12 years on interdisciplinary research involving molecular biology, physical chemistry, software, and electrical engineering. Currently, Dr. Pozhitkov is a researcher at the University of Washington, Seattle. His technical interests include hardware programming, vacuum tubes, and high-voltage electronics.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/259.

REFERENCES

[1] Intel Corp., "Intel Core 2 Duo Processor E8000 and E7000 Series Datasheet," 2009, http://download.intel.com/design/processor/datashts/318732.pdf.

[2] ———, "Intel 64 and IA-32 Architectures Software Developer's Manuals," www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html.

[3] ———, "Intel 4 Series Chipset Family, Datasheet" 2010, www.intel.com/Assets/PDF/datasheet/319970.pdf.

[4] ———, "Intel I/O Controller Hub 10 (ICH10) Family, Datasheet," 2008, www.intel.com/content/www/us/en/io/io-controller-hub-10-family-datasheet.html.

[5] M. Klasson, "HexIt – The Hex Editor," 2011, http://mklasson.com/hexit.php.

RESOURCE

E. Nisley, "Journey to the Protected Land," *Circuit Cellar* 48–65, 1994–1995.

SOURCES

Intel Core 2 Duo Processor
Intel Corp. | www.intel.com

Visual C++ Integrated development environment (IDE)
Microsoft Corp. | www.microsoft.com

ROUTE FASTER !



WITH PROTEUS PCB DESIGN

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

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

PROTEUS DESIGN SUITE

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

Features:

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

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

labcenter Electronics

www.labcenter.com

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

by Steve Lubbers (USA)

KartTracker

A GPS-Based Vehicle Timing & Monitoring System

The KartTracker is a standalone GPS-based vehicle timing system. The LCD on the system's development board operates as the user interface, an integrated accelerometer records G forces, a serial port provides connections to a GPS receiver and a wireless transmitter, and removable flash memory stores data.

In a variety of racing activities, time is of the essence! It is useful to both know and record information such as lap times, top speeds, and best lap times. At a more personal level, the questions are "Did John really win again?" and "Am I racing any better than I was last year?"

In any type of race (i.e., cycling, running, or motor vehicle), you want to know who is fastest. Outside of the race, timing information is important to help determine if training, testing, and adjusting are making you better or worse.

In the fall of 2010, Renesas announced the RX MCU Design Contest. Contestants were asked "What can you do with…?" and were tasked to bring their ideas and designs to reality. This challenge gave me a golden opportunity to showcase my talents and prototype a timing system to use for kart racing (see Photo 1). Next time we visit the track, we will know who is quickest!

THEORY OF OPERATION

I decided on a standalone device to time my racing activities. This would enable operation anytime and anywhere for any type of activity. Since small global positioning system (GPS) receivers have become inexpensive enough to appear in cars, watches, and pet collars, I investigated their use for my project. I figured if Jill Garmin can direct me—"Please drive the highlighted route"—perhaps her third cousin Bubba can help my racing! A GPS-linked device appeared to be the perfect solution to my problem.

Most of the pieces of my KartTracker are already built into the Renesas Electronics RX62N development board (see Figure 1). The liquid crystal display (LCD) on the development board operates as the user interface and shows the driver what is happening as he races. The integrated accelerometer can be used to record the G forces experienced while racing. A serial port provides connections to a GPS receiver and a wireless transmitter. Removable flash memory stores all the race data so you can brag to your friends. You now have all of the pieces of my KartTracker.



Photo 1—Karts provide the neighborhood NASCAR fans with an inexpensive way to experience racing. Always on the lookout for another engineering project, I wondered how I could add a computer system to this adventure.

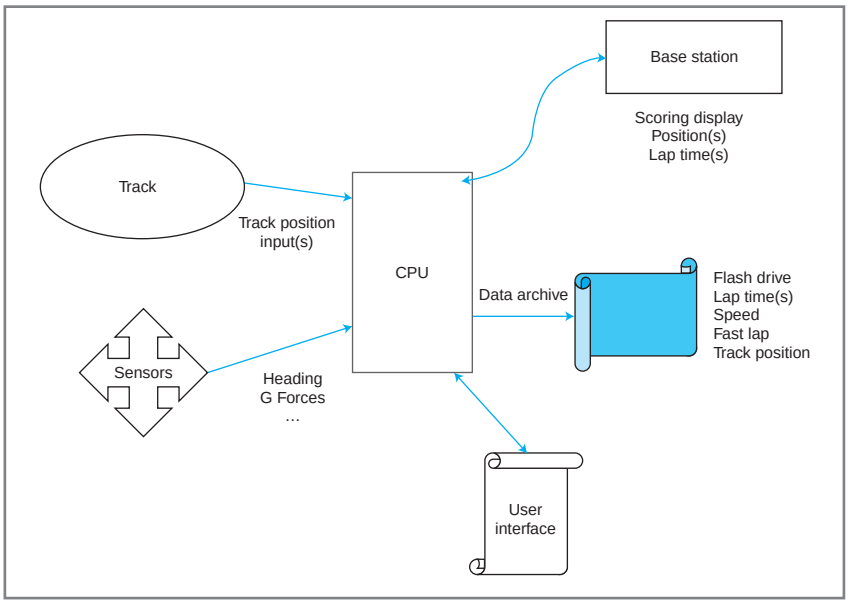


Figure 1—My original concept for a racing timer included a series of sensors connected to a computing system. GPS and acceleration sensors supply the processor with the information to compute timing data. The results are shown to the user, stored for posterity, and broadcast to a base station.

GPS POSITION & PARSING

GPS receivers output a continuous stream of serial data that contains the longitude and latitude position of the receiver. The SiRF "NMEA Reference Manual" describes the operation of the Parallax GPS I used. There are 12 different GPS messages. For my project, I only need one: the recommended minimum specific GNSS data ($RMC) message which contains time, speed, longitude, and latitude. The format of $RMC is shown in Table 1.

Since GPS data looks like standard ASCII text, it is possible to record the data and use it to replay a race. I used an old handheld computer to capture several laps' worth of data. Then, as needed, I could replay the data back into KartTracker. This made it possible to develop the software and test the system without constantly driving around in circles.

G FORCE

The RX62N demonstration kit contains an integrated three-axis accelerometer. I use this sensor to detect the forces acting on the driver during a race. The development kit came with an I²C master software sample that showed exactly how to use this sensor, so I incorporated its sample into my project. Each time a position message ($RMC) is received from the GPS, my software goes out and obtains a corresponding accelerometer reading. Now, for each second of racing, you have the kart's position and the forces acting on it.

Any other sensors could be added to KartTracker in a similar fashion. For example, for an off-road event, you could record the altitude as well as the position. Additional sensors are left as an exercise for the student.

TRACK MAP & POINTS

The KartTracker uses GPS points to determine its position. Those points are compared against an internal reference. The GPS points from a sample track are shown in Figure 2.

Name	Example	Unit	Description
Message ID	$GPRMC		RMC protocol header
UTC time	193405		hhmmss.sss
Status	A		A = data valid or V = data not valid
Latitude	3939.159		ddmm.mmmm
N/S Indicator	N		N = north or S = south
Longitude	8410.3933		dddmm.mmmm
E/W Indicator	W		E = east or W = west
Speed over ground	10.3	knots	
Course over ground	359.91	degrees	TRUE
Date	271110		ddmmyy
Magnetic variation		degrees	E = east or W = west
East/West indicator			E = east
Mode	A		A = Autonomous, D=DGPS, E = DR, N = Output Data Not Valid
Checksum	*4D		
<CR> <LF>			End of message termination

Table 1—The recommended minimum data message from GPS ($RMC) fulfills my timing system's need. This is a 4800-bps ASCII message that contains position and time information.

BEST SCOPE SELECTION
& lowest prices!

25MHz Scope
Remarkable low cost 25MHz 2 channel plus trigger USB bench scope with 8 inch full color LCD display. Spectrum analysis and autoscale functions.
PDS5022S $287

iPhone Scope
5MHz mixed signal oscilloscope adapter for the iPhone, iPad and iPod Touch! A FREE iMSO-104 app is available for download from Apple App Store.
iMSO-104 $297.99

100MHz Scope
High-end 100MHz 1GSa/s 2-channel benchscope with 1MSa memory and USB memory port. Includes a FREE scope carry case! New low price!
DS1102E $399

60MHz Scope
60MHz 2 channel digital scope with a 500 MSa/s sample rate, 10MSa memory & 8" color TFT-LCD screen.
SDS6062 $399

World's Smallest
The world's smallest MSO! This DIP-sized 200kHz 2 channel scope includes a spectrum analyzer and arbitrary waveform generator. It measures only 1 x 1.6 inches in size!
Xprotolab $49

100MHz MSO
2 channel 100MSa/s oscilloscope and 8-ch logic analyzer USB 2.0 and 4M samples storage per channel with sophisticated triggering and math functions.
CS328A $1359

Handheld 20MHz
Fast, accurate handheld 20MHz 1-ch oscilloscope.
- 100 M/S sample rate
- 3.5 in. color TFT-LCD
- 6 hour battery life
Inc. rugged impact-resistant case
HDS1021M $299.95

More selections at:
www.saelig.com
Follow us on:

Saelig .com
unique electronics

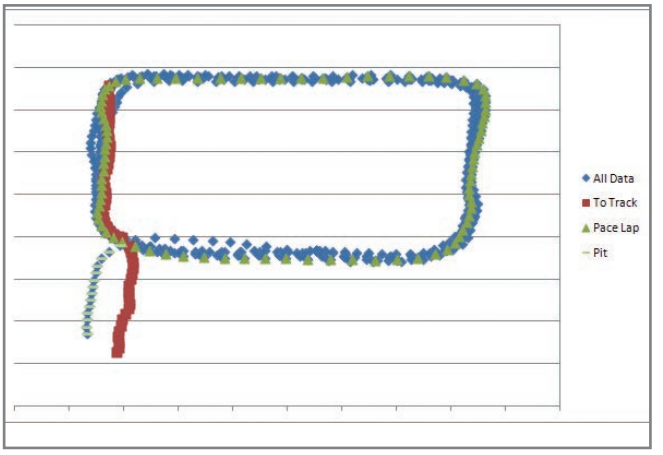


Figure 2—GPS points were recorded during a few laps around the racetrack. The resulting data was plotted using Microsoft Excel. It sure looks like the path I drove!

For most operations, the absolute position of the kart is not important, only its proximity to a stored location. For example, I don't care about the finish line's locations. I just care when I cross it. This abstraction enables the software to use the GPS in its native format of degrees and minutes. Magnitude comparisons tell us when we pass the point (wherever that point is). The actual kart position in degrees/minutes/seconds is not important for calculations.

On the other hand, absolute position can be useful if you want to show a map of the kart's path. You can take the kart's actual positions and shrink them down to the size of a display screen and then produce a map of the track with a cursor showing the vehicle's position. So, I use a combination of relative positions for timing calculations and absolute positions for a graphic display.

DATA SAMPLING

My GPS receiver generates a $RMC message every second and has a stated accuracy of 5 m. With this slow sample rate and a 5-m error, I wondered if my project would succeed. In 1 s, my kart could move 50′, and the position reading could be off by an additional 15′. I drove around a track recording data to see what quality of information I would really get (see Figure 2). The track map showed acceptable repeatability for my purposes.

CIRCLE OF PROXIMITY

The waypoints and finish line are stored as points internal to the KartTracker. Given a track that is 28′ wide and a GPS that returns a sample every 30′ to 60′, the chances of hitting an exact spot on the racetrack lap after lap is quite small. So, in order to detect individual points on the track, a circle is drawn around each point as shown in Figure 3. The software looks for the kart to pass through this circle to determine that it passed by a given point. Detailed

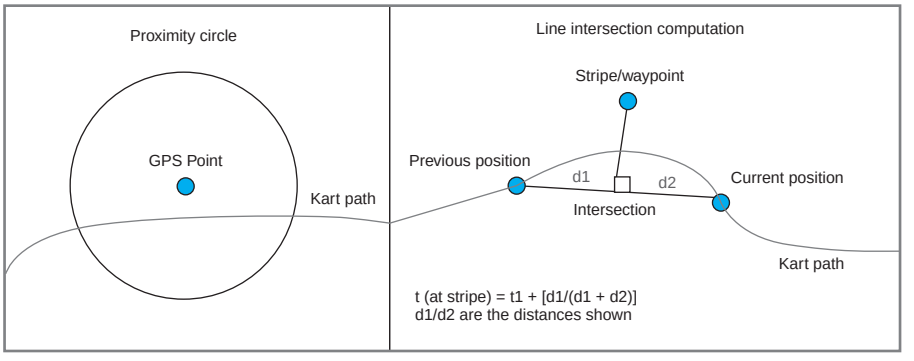


Figure 3—Variances in vehicle position meant I would need a bit of creativity to extract my timing information. A specific point on the track is defined as a circle around a point. The exact time at the "fuzzy" point is determined by interpolating between data samples as we pass by the point.

> "I have created embedded systems with data storage. In all those cases, I wanted to get the data onto a PC for analysis and storage. This suggested that I use a standardized data format and typical PC storage media with the KartTracker ... you can replay GPS data contained on the memory card, and the memory card will hold race statistics."

timing calculations are only performed on the points to which it is "close." Hysteresis ensures that you leave the circle before a new computation will begin.

INTERPOLATION BETWEEN TIMES

Since a "point" on the track turned out to be a circular region, and the kart can travel 50′ between samples, I needed to approximate the kart's position between two points. To do this, the software performs a linear interpolation between the last time/position before a point and the first time/position after the point. The time "at the point" shown in Figure 3 is the linear interpolation of the intersection of the kart path with a point.

The time between samples is the 1-s GPS sampling interval, and the velocity is assumed to be constant. The start/finish line is sometimes called "the stripe." The time at the stripe is determined by drawing a line perpendicular to the kart's path that passes through the starting point. The intersection point with the kart path is the point used in the finish line time calculations.

WAYPOINTS

The collection of waypoints is a trail of GPS breadcrumbs collected during a single closed lap. A point is recorded every 5 s. Forty points enable a 1.1-mile track at 20 MPH. If you take every fifth point from the track map, you will have a good idea of what the waypoint collection looks like.

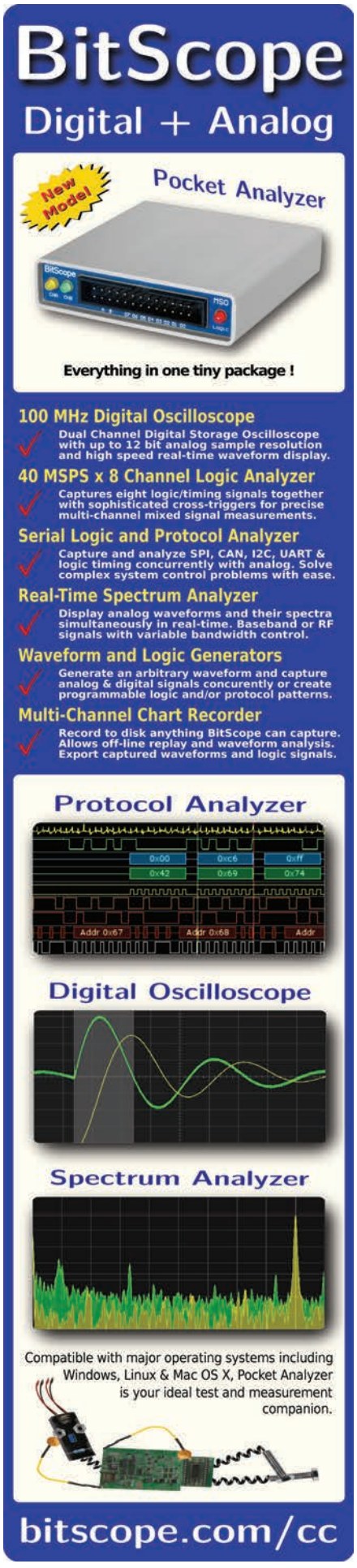
Once the waypoints are collected, the software follows the kart around the track. Similar to the navigation, compare the position to the waypoints. As the kart passes a point, the time at the point is computed. This new time can be compared to the last time at the same position. This gives you an idea of whether you are driving faster or slower than the previous lap.

GREAT CIRCLE DISTANCES

When the software is looking to determine if you have passed a waypoint or the starting line, it was sufficient to just look at relative positions and distances. However, if you want to know how far you traveled in miles, the problem becomes a bit more complicated! Because the earth is a sphere, the translation between longitude and latitude positions doesn't directly convert to distance. Instead, the computation of "great-circle distance" and "great-circle arc" is required to execute the conversions. Since I expect my points to be within a mile of each other, the Haversine formula is used. Trigonometry textbooks carefully explain the mathematics involved. I followed the textbook formula to create my precise calculation routines.

FILE STORAGE OF DATA

I have created embedded systems with data storage. In all those cases, I wanted to get the data onto a PC for analysis and storage. This suggested that I use a standardized data format and typical PC storage media with the KartTracker. The project development board contains a socket for a microSD card. FAT is a standard storage format for PC data. That set the main direction for data storage. Files would be stored as standard text on a microSD card to enable easy use by PC programs, such as Excel, for analysis and Word for examination and printing. You



```
   +-------------+
1  |  -123.45    |    (Time delta, reference previous lap)
2  |             |
3  |Speed 99.99  |    (Track speed in MPH from GPS, smoothed)
4  |             |
5  |Lap #   123  |    (Current lap number)
6  |Time 123.45  |    (Lap time for previous lap)
7  |             |
8  |Nav X  Log X |    (Operating status)
   +-------------+
```

Figure 4—An eight-line LCD is used to show the driver's racing statistics. The most important training aid is the time delta compared to the previous lap.

can replay GPS data contained on the memory card, and the memory card will hold race statistics.

POSITION REPORTING

Now someone using the KartTracker knows how fast he is going and how much better he is doing. What he doesn't know is how the other guy is doing. The spectators don't have access to the race information either. A wireless data network could link multiple trackers to a base station for continuous updates to officials and fans alike. I imagined something like the ham radio automatic packet reporting system (APRS) where position and identification data is continuously broadcast. For KartTracker, the identification would be the car number and the position would be the latest waypoint.

OPERATING INSTRUCTIONS

The GPS KartTracker provides an easy-to-use method to time and track your racing activities! An LCD screen shows real-time data as you race around a track. Overall race statistics are stored for later analysis on a PC. And, just for fun, a moving light-emitting diode (LED), which is on the development board LED ring, shows the real-time position of the kart on the track.

READY? LET'S RACE!

Drive to the start/finish line and press the Start button. As you drive this first pace lap, the KartTracker will build an internal map of your racetrack. As you approach the starting line again, get ready to race! As you cross the stripe, the KartTracker begins counting your laps, displaying your speed, and calculating your speed difference compared to the previous lap.

Race as long as you like. Watch your speed on the KartTracker display (see Figure 4) and see if you are doing better or worse than the previous lap. When you capture the checkered flag at the end of the race, press the Stop button. Your overall race statistics are written to a microSD card. Take a look at the file stats.txt on your PC (see Figure 5). It tells how well you did and the maximum G forces you experienced.

If your friends also have track reports, compare the "end time." The person with the lowest time wins the race. Compare lap times and top speeds. Brag and collect free beverages as dictated by your local racing group.

TESTING & DEMO MODE

I added a feature that enables me to replay stored GPS data. This feature was vital during software development and debugging. As a side effect, this feature can be used to see how the system operates without actually driving around a track.

To initiate this test mode, press and hold SW2 and reset the RX62N board. The KartTracker will begin running, but the GPS data will come from a data file (GPSplay.txt) stored on the microSD card. All other operations will be the same. A compile time variable can change the GPS replay rate so that it processes faster or slower than real time. A faster replay makes for a better demonstration than watching a 5-min. race on the small LCD.

HARDWARE & BLUETOOTH

My KartTracker prototype is built around the Renesas YRDKRX62N demonstration kit for RX62N (see Photo 2). The control switches, an LCD and a ring of LEDs for computed track position compose the user interface for my system. The included microSD slot accommodates secondary storage for recording race information. The only thing I required that wasn't included on the RX62 demonstration board was a GPS receiver. Any commercial GPS with a serial output could be plugged into the circuit board's COM port, as shown in Figure 6. I used a Parallax SiRF GPS receiver.

I performed some experiments to determine how to network the KartTracker with a base station. Ham

```
Car # 18 Driver:     Steve Lubbers
Lap #    Lap Time    AVG MPH    Top Speed        |    Seconds
_____    ________    _______    _________        |  __________
  0          0.00       3.82        11.75        |   89919.22
  1         85.60      12.25        14.27        |   90004.82
  2         86.00      12.27        14.57        |   90090.82
  3         85.85      12.26        14.86        |   90176.67
  4         88.23      11.89        15.48        |   90264.90
  5        207.69       9.87        14.19        |   90472.59
_____________________________________________________________

Top speed     11.75 MPH     Max accel -1.20G     Yaw 0.30G

End time     90472.59     UTC (seconds)
```

Figure 5—At the end of a race, the driver's individual statistics are written to a memory card. He can brag about his top speed and marvel at the G forces he experienced.



Photo 2—The prototype KartTracker is based on the Renesas RX62N development kit. A Parallax GPS receiver is connected to the available serial port. My RF tests were performed using the IOGEAR GBS301 shown in the upper right of the picture.

radio provides one possible solution called APRS, but it wasn't really intended for this update rate, and an FCC license is required to use it. My next thought was to use the Wi-Fi module that was designed for the Renesas board. Wi-Fi would create an actual computer network between vehicles and a base station. It seemed like a good idea until I recalled my neighborhood's experiments with sharing Wi-Fi for computers. The next door neighbor's network worked, but the computer across the street was too far away. Thus, it looked like Wi-Fi was out. Not to be deterred, the next attempt was Class 1 Bluetooth. A pair GBS301 Serial adapters from IOGEAR are advertised to have a range of 330'. This was still not good enough for a final product but effective enough to prove the concept and build a prototype. Note that the minimum data rate for the Bluetooth module is 9600 bps. To use both the GPS and the Bluetooth

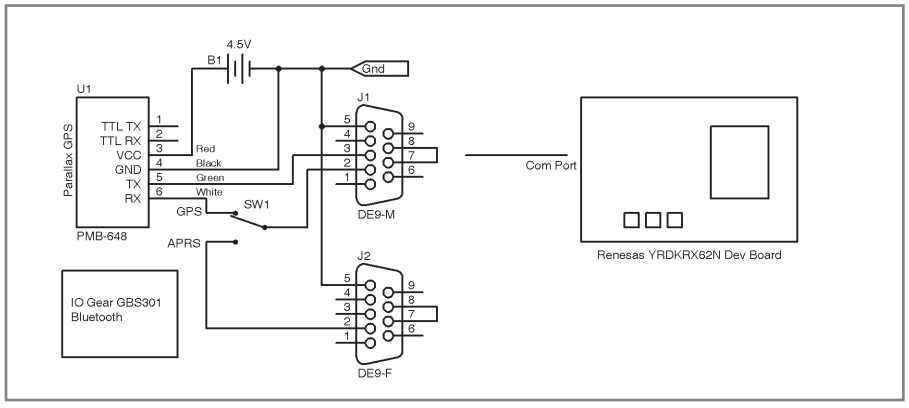


Figure 6—The KartTracker is a Renesas development board with a couple of external serial devices. The GPS always sends data to the CPU and the CPU always sends data to the Bluetooth. In case you want to configure the GPS, a switch enables the CPU to talk to either the GPS or the Bluetooth.

THE NEW PICOSCOPE
2205 MSO
MIXED SIGNAL OSCILLOSCOPE
GREAT VALUE, PORTABLE, HIGH END FEATURES AS STANDARD AND EASY TO USE
Think Logically...

pico Technology

Channels	2 Analog, 16 Digital
Resolution	8 bit
Bandwidth Digital frequency	Analog 25 MHz, Digital 100MHz combined
Sampling rate	200MS/s
Trigger modes	Edge, Window, Pulse width, Window pulse width, Dropout, Window dropout, Interval, Runt pulse, Digital, Logic
Price	$575

www.picotech.com/pco467
1-800-591-2796

Listing 1—This is the pseudocode for the main processing loop for the KartTracker. The software continuously acquires GPS data and calculates race statistics. C module names are shown to the right

```
Initialize Hardware and Software
While Lap 1
          Build Waypoint Trail

While (Racing)
          Get GPS Sentence                              (GPS)
          Parse Longitude/Latitude/Time                 (GPS)
          If Passing the start line                     (Navigation)
                    Calculate Lap Statistics

          If Passing a waypoint                         (Waypoint)
                    Calculate Lap time difference
                    Broadcast track position

          Show User Data                                (Display)

Store Race Statistics                                   (TFAT)

End
```

devices requires reconfiguration of the GPS and the serial port to 9600. Since the wireless link is still a "work in progress," the delivered software was set to the GPS default of 4800 bps.

SOFTWARE

The KartTracker software was built around the UART software sample provided with the RX62N development kit. To provide file system support, the Renesas microSD/Tiny FAT software was added. Finally, my custom GPS KartTracker software was added to the Renesas samples.

My software consists of GPS, navigation, waypoints, and display modules. Support software was added to interface to the UART serial port, the file system, and the user display and control on the RX62N circuit board. The software's normal operating loop is shown in Listing 1.

GPS, NAVIGATION, & WAYPOINTS

The GPS software is responsible for interpreting the NMEA sentences received from the Parallax serial GPS unit. The theory of operation section detailed the interpretation of the required GPS $RMC message. The longitude, latitude, and time for the current and previous positions are retained so that at any given time the KartTracker will know where it is and can compare that to where it was for predicting what is to come. Each time position is recorded, the G forces are also measured and recorded. Live GPS data can be processed as well as file data from a previous run.

The navigation software is responsible for interpreting and processing the GPS data. As shown in the theory of operations, the current and previous positions are compared to the starting line point. To accomplish this, various functions compute line and point intersections in two-dimensional space. In most cases, the software is looking for points that are "close" with a trend of direction. Additional functions can precisely calculate the distance between points, if required.

The waypoints software creates a trail of points around the racetrack. The waypoint software creates this trail. Then, similar to the navigation software, it determines the closest waypoint to the kart. Time and distance calculations are used to determine elapsed time and are compared to a previous lap.

DISPLAY SOFTWARE

The display module software is responsible for showing the user what is happening. A set of 12 LEDs are used to show the relative track position of the kart. The single LED closest to the track position of the kart is lit. The LED moves around the track as the actual vehicle does the same.

The LCD shown in Figure 4 is divided into sections to tell the driver what is happening. The LCD shows the current lap number and speed as well as relative speed compared to the previous lap. The LCD data is updated at the completion of each data-processing loop.

TRACK-MAPPING SOFTWARE

My endeavor to create a moving map display of the kart's path was successful, but disappointing. It seemed interesting to show the track on the 64 × 96 graphic LCD screen. The plan was to show a picture of the track with a car icon moving around it.

The mapping software runs by examining all of the track's waypoints and then scaling them down and onto the display. A rough map is drawn by placing dots at the waypoints. As the kart moves, its GPS location is added to the map, showing a dot moving around the map.

For the fans, the screen turned out to be way too small. For the driver, it presents more of a distraction than a benefit. There wasn't enough resolution to show more than a pixel for each feature. Use of multiple pixels reduced the resolution below where the display was interesting. I found that the circle of LEDs provided the driver the same type of information but was easier to use.

In the end, I left the graphics code in place with a compile time selection in `Display.c\Disp_Loop()\x Display=1;`, where you can choose either the text-based statistics or a graphic display. When I find a display I am happy with, a user interface option will be created to select the display mode during use.

FILE SYSTEM SOFTWARE

Using the Renesas sample project RX62N_RDK_SD_DRIVER software, facilities exist to store race statistics and to save and restore waypoints. The sample FAT software only supports files in the 8.3 format, so keep your names

short. A more sophisticated user interface is required to provide user controlled playback of achieved GPS data.

The file system provides the ability to replay stored GPS data into the KartTracker. This feature was vital during software development and debugging. As a side effect, this feature can be used to see how the system operates without actually driving around the track. During data playback, "live" GPS data is replaced by the contents of a data file read from the memory card.

POSITION REPORTING SOFTWARE

My wireless network demonstration currently uses a Bluetooth serial port. This enables the position information to be created as an ASCII text string that is sent to a serial port. The position reporting software keeps the judges and fans up to date with the ever-changing race. Each time a kart crosses a waypoint on the track, the KartTracker will broadcast the car number, the waypoint number, and the time at the waypoint. The IOGEAR dongle performs the magic of connecting the position data to the matching receiver. In the future, I'll create a base station to collect and display this information.

My Bluetooth experiments showed the ease of creating the position data but also the complications of actually broadcasting the information. The jury is still out for a simple long-range solution.

THE FINISH LINE

I fulfilled my original goals for the creation of my prototype KartTracker and then exceeded my own expectations. In the first version of the prototype, the vehicle was timed, data was displayed to the operator, and statistical data was archived to removable media. Successive iterations have added the sensing of G forces, processing a position map, and broadcasting position and timing information. It's about time for engineering to let the project escape, so that kart useable hardware can be constructed.

Let's go racing, boys!

Steve Lubbers (KE8FP@arrl.net) started his budding electronics career in the days of "Ciarcia's Circuit Cellar" when he designed an 8080-based computer as a high school science fair project. Since then, Steve has achieved an Extra Class Amateur Radio License and a B.S. in Computer Science. He currently builds amateur radio embedded systems for fun and embedded automotive electronics for profit.

PROJECT FILES

To download project files and a link to a project video, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/259.

RESOURCES

SiRF Technology, Inc., "NMEA Reference Manual Revision 2.2," 2008.

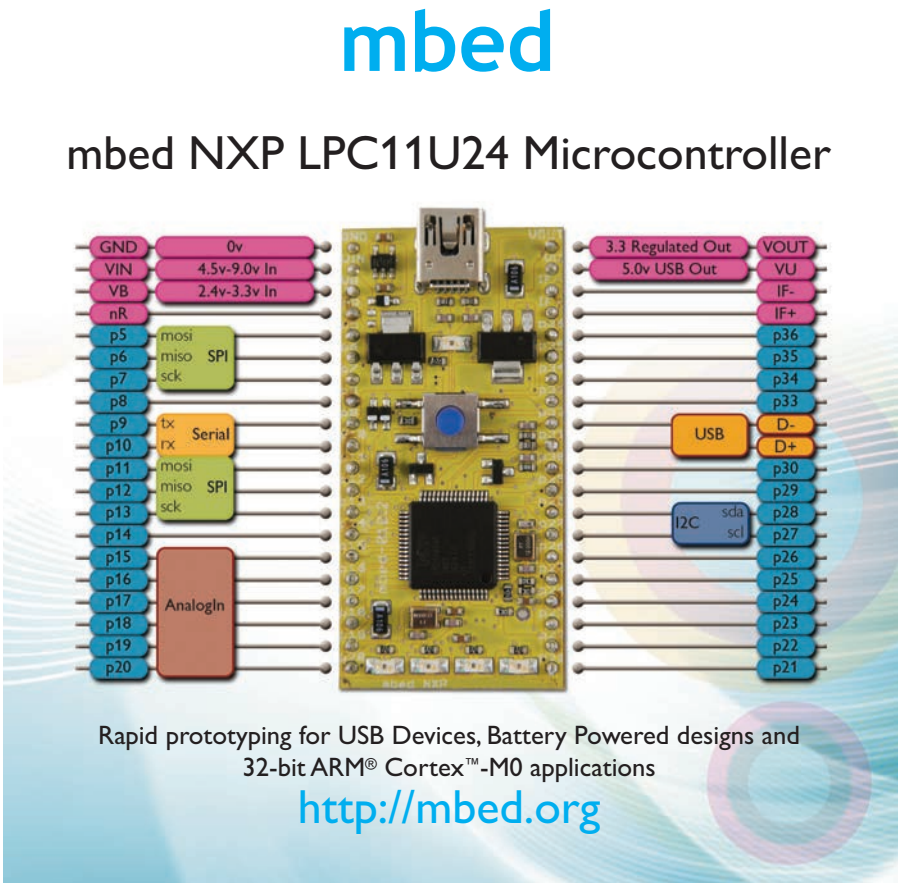
Renesas Electronics Corp., "Application Note R20AN0043EJ0100: How to Use the Sample Codes for TFAT and MMC Drivers on SCI/RSPI," 2010.

SOURCES

GBS301 Serial adapter
IOGEAR | www.iogear.com

PMB-648 GPS module and SiRF GPS receiver
Parallax, Inc. | www.parallax.com

RX62N microcontroller and YRDKRX62N demonstration kit for RX62N
Renesas Electronics Corp. | www.renesas.com



by Ed Nisley (USA)



RFI Bypassing (Part 1)

Capacitor Self-Resonance

With GPS technology and audio radio interfaces on his personal fleet of bikes, Ed's family can communicate with each other while sending GPS location data through an automatic packet reporting system (APRS) network. This article details how to use a KG-UV3D radio interface rigged with SMD capacitors to suppress RF energy. Test-fixture measurements on isolated capacitors are also described.

Riding a recumbent bicycle turns out to be an excellent attention getter, but nearly everyone also comments on the amateur radio antenna mounted on the seat frame of my bike. With a global positioning system (GPS) and audio radio interface, as shown in Photo 1 and described in my October 2008 column ("HT Audio, GPS for APRS, and What Works," *Circuit Cellar* 219) on each of our three bikes, we can talk on our family rides while sending GPS location data through the automatic packet reporting system (APRS) network.

My long-obsolete Icom IC-Z1A radio finally failed last year after perhaps 15,000 miles of mechanical vibration. Its replacement, a Wouxun KG-UV3D radio, has a different electrical interface, but I managed to hack my circuit board into compatibility. The Icom and Wouxun battery packs also have, of course, completely different dimensions, so I settled for strapping the interface box outside the radio mount and avoided riding in the rain.

While designing a new board for the Wouxun radio, I wondered whether the surface-mount device (SMD) capacitors I'd put across the old board's input and output cables were as effective at suppressing RF energy as I had assumed. In this column, I'll describe some test-fixture

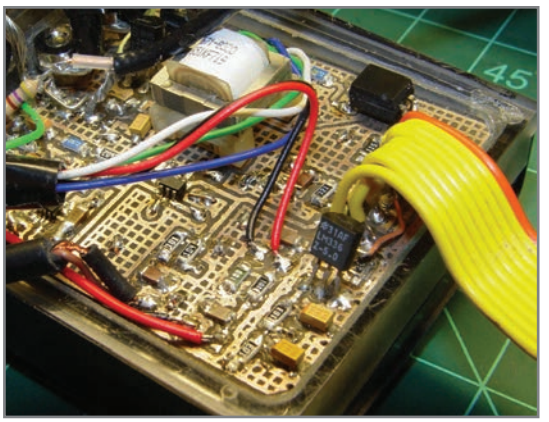


Photo 1—This audio and power interface board replaces the Icom Z1A radio's battery pack. The resistor attached to the RG174 cable entering from the upper right probes the tracking generator's signal at the mic input. The yellow ribbon cable connects to the TinyTrak 3+ GPS encoder.

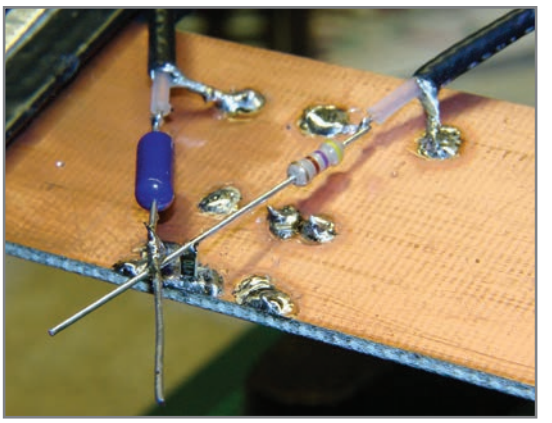


Photo 2—A pair of axial-lead resistors isolate the tracking generator and spectrum analyzer from the components under test. The 47- SMD resistor, standing upright just to the right of the resistor lead junction, forms an almost perfect terminator.

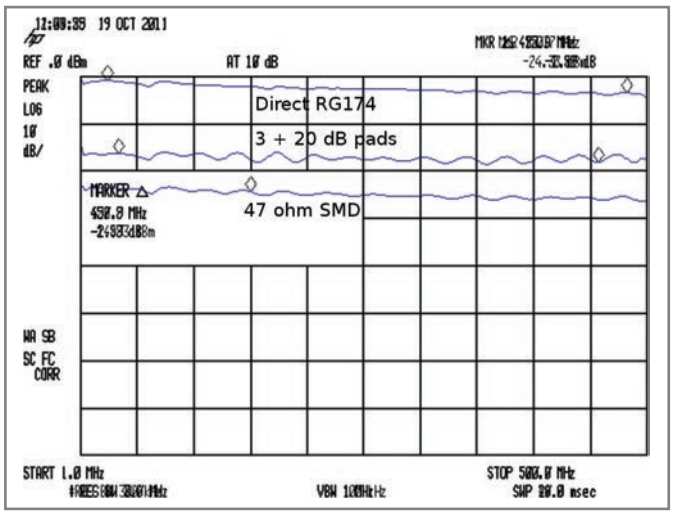


Photo 3—These three superimposed spectrum analyzer traces show that the RG174 cables and attenuators have a reasonably flat frequency response out to 1 GHz. The 47-Ω SMD resistor has no appreciable stray inductance.

measurements on isolated capacitors that show some surprising results. Then, in the next column, I'll extend these results to a circuit and cables.

RF COMPONENT TEST FIXTURE

Repeatable and dependable measurements require a solid test fixture. Although the collection of parts in Photo 2 may look like a kludge, it's an exemplar of the "ugly construction" technique that's actually a good way to build RF circuits. "Some Thoughts on Breadboarding," by Wes Hayward, W7ZOI, gives details and suggestions for constructing RF projects above a solid printed circuit board (PCB) ground plane.

Long-time *Circuit Cellar* readers may recall the capacitor measurements in my February 2005 column ("Capacitors at RF," *Circuit Cellar* 175) with a similar test fixture. I'm now using miniature RG174 coaxial cables for the connections to my HP8591 spectrum analyzer and tracking generator, with single resistors acting as buffers.

The top spectrum analyzer trace in Photo 3 shows that a direct connection between the tracking generator output and spectrum analyzer input through a pair of 2 m RG174 cables has less than 3 dB of loss across the 1 MHz to 500 MHz frequency range. The generator produces 0 dBm = 1 mW at its output connector.

Typical RF circuits, including the spectrum analyzer, maintain a constant 50 Ω resistive impedance, but in this case I was interested in the wildly varying impedance of individual components. The 21 Ω resistor helps to isolate the tracking generator's output from even a direct short to ground and the 470 Ω resistor reduces the loading from the spectrum analyzer's 50 Ω input. The middle trace in Photo 3 shows that their series combination produces about 17 dB of loss, with ripples arising from the poor impedance match to the 50 Ω coaxial cables.

The bottom trace in Photo 3 shows 23 to 27 dB of loss with a 47 Ω resistor soldered to the ground plane from the junction of the two padding resistors, as shown in Photo 2.

What's important here is that the response remains fairly flat across the entire frequency range: the parasitic inductances of the resistors don't contribute appreciable error to the measurement. Visit *Circuit Cellar*'s FTP site to download my Spice model of the circuit to examine the effect of various parasitics.

To check for external parasitics, I replaced the SMD resistor with a 12-pF SMD capacitor (which looks exactly the same!) and got the pronounced self-resonance shown in Photo 4; note the narrower frequency span. The capacitor forms a series resonant circuit with its internal parasitic inductance. The resonant frequency depends on the capacitance and inductance:

$$F_R = \frac{1}{2\pi\sqrt{LC}}$$

Given a 12 pF capacitance and 851 MHz resonance, L works out to 2.9 nH, which is in the right range for SMDs and short wires. At resonance, the reactance of the capacitor and inductor will be numerically equal. Because I can directly measure the capacitance with a meter, its reactance has less error:

$$X_C = \frac{-1}{2\pi f C} = -16\ \Omega$$

The capacitor's parasitic resistance, known as its equivalent series resistance (ESR), determines its Q, a measure of the 3 dB bandwidth of the resonant response. The left-hand cursor in Photo 4 is 3 dB above and 25 MHz from the minimum, so the bandwidth is 50 MHz. Knowing that, the circuit Q factor is:

$$Q = \frac{F_R}{BW} = \frac{851\ \text{MHz}}{50\ \text{MHz}} = 17$$

Given Q, the capacitor's ESR is:

$$ESR = \frac{X_C}{Q} = \frac{16\ \Omega}{17} = 900\ \text{m}\Omega$$

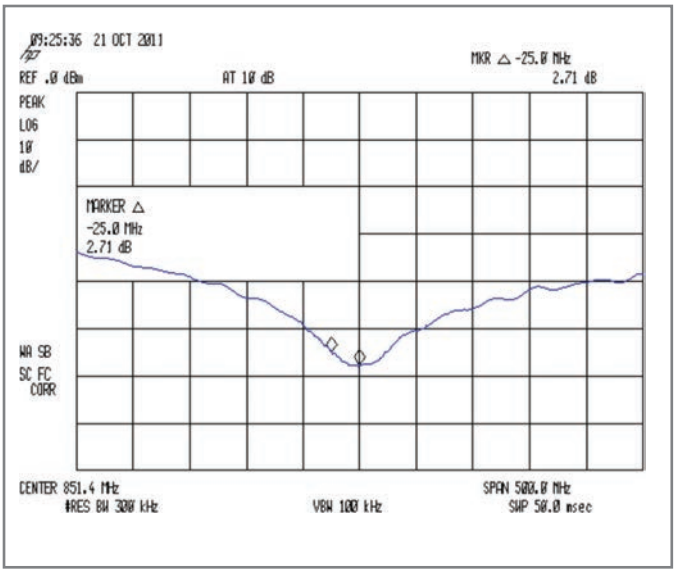


Photo 4—A 12-pF SMD capacitor resonates at 851 MHz with a 50-MHz bandwidth, indicating 2.9 nH and 900-mΩ parasitics.

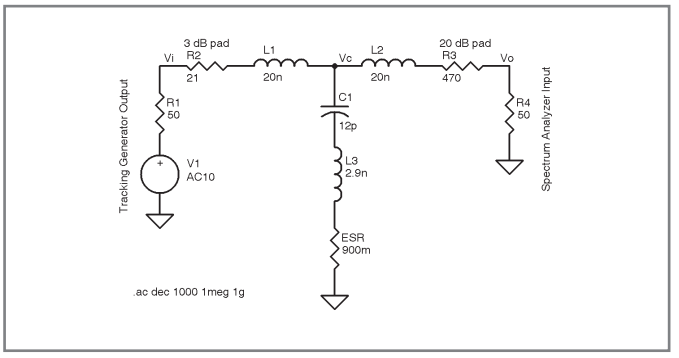


Figure 1—Combining estimated inductances for the test fixture resistors with the calculated SMD capacitor parasitics produces excellent agreement with the measurements.

Plugging those values into the Spice simulation model in Figure 1 produces a result that matches Photo 4 so closely there's no point in printing it: you can download the model and see for yourself.

With a reasonable test fixture in hand, it's time for some measurements!

SURPRISING SELF-RESONANCE

The four 100 nF ceramic capacitors in Photo 5 span nearly half a century of technology, from the hulking red disk found on 1960s-era PCBs to the 1206-size SMD cap that's six times larger than the 0201 pepper-flake caps in mobile electronics. Several engineering generations grew up knowing that nearly any noise problem can be solved with a sufficient dose of 100 nF ceramic capacitors, so PCBs have these caps sprinkled across their surfaces: bypassing digital logic chips, protecting I/O lines, and paralleled across larger electrolytic capacitors.

In each case, conventional wisdom says that capacitors supply local energy storage that reduces electrical noise. Because capacitive reactance is inversely proportional to frequency, bypass capacitors should have little effect at low frequencies, while acting as low-impedance paths to shunt high frequency signals to ground.

The small 100 nF ceramic cap soldered into the test setup in Photo 6 comes from a large stockpile that I've soldered on many of my through-hole PCBs before switching to surface-mount parts. Because it's physically small with short leads on 0.1" centers, I've always assumed it had a relatively low impedance at high frequencies.

However, the spectrum analyzer trace marked Bypass in Photo 7 shows that, even in this simple test circuit, the capacitor's impedance rises from its self-resonant minimum near 7 MHz. In the VHF amateur band around the 144.39 MHz APRS frequency it provides only 20 dB more attenuation than an open circuit.

That's much better than the red ceramic disk capacitor with untrimmed 40-mm leads, which reduces the signal by barely 5 dB at VHF. Cutting the leads to a more practical 5-mm adds another 10 dB of attenuation, but you can see why disk capacitors aren't useful in contemporary circuits. Incidentally, those long component leads came in handy when production circuits used point-to-point wiring between

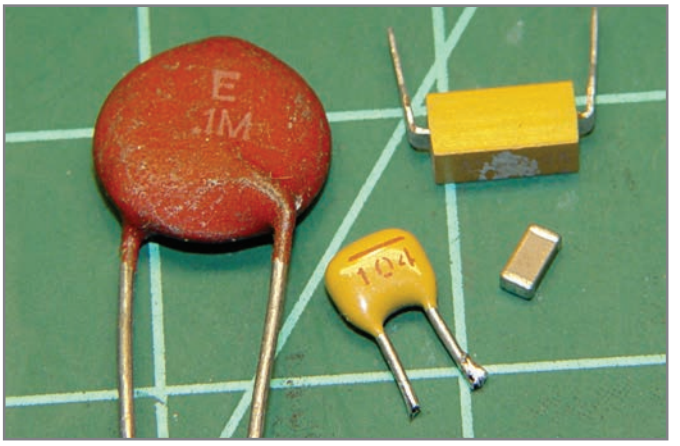


Photo 5—Ceramic capacitors similar to these 100-nF parts often bypass power supplies. That huge red disk would look at home in a vacuum tube circuit!

vacuum tube sockets and multi-lug terminal strips.

The rectangular DIP capacitor in the upper right of Photo 5 dates back to the late 1970s heyday of TTL ICs, where it fit neatly at the pin 1 end of a chip and bypassed the 5-V supply to the underlying ground plane. Surprisingly, the DIP trace in Photo 7 shows it has only marginally better attenuation than the trimmed disk capacitor, but the parallel effect of dozens of the capacitors did help reduce the power supply impedance at TTL's relatively low clock frequencies.

The bottom trace in Photo 7 is actually two nearly identical traces produced by a 100 nF SMD capacitor and a 3 mm wire jumper directly to ground. As you would expect, the wire to ground provides nearly perfect attenuation at DC, but stray capacitance and inductance reduce its effect in RF circuits. Surprisingly, a dead short may not act as a dead short!

The 50 MHz frequency span in Photo 8 zooms in on the resonances to show that the SMD capacitor has a self-resonance at 12 MHz that corresponds to L = 2 nH, Q = 2.2, and ESR = 60 m Ω. Although it's better than the other capacitors,

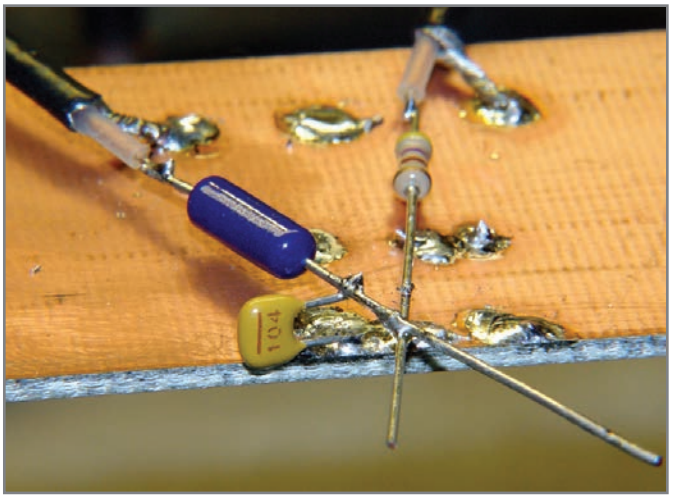


Photo 6—I've soldered many of these 100-nF ceramic capacitors on through-hole PCBs. The resistor leads add much less inductance than those long wires in Photo 1.

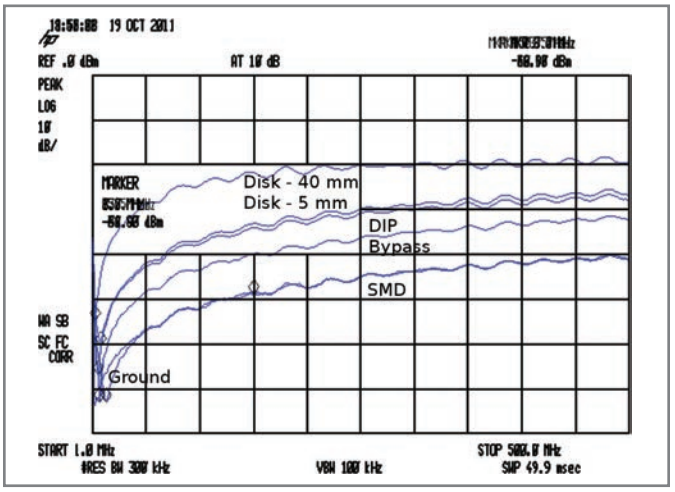


Photo 7—The impedance of all four 100-nF capacitors, as well as the short to ground, rises as the frequency increases due to lead inductance and parasitic coupling. The spectrum analyzer's noise floor is about –75 dBm in this configuration, just below the resonance minima.

it still provides barely 20 dB of attenuation in the VHF band.

Turning off the tracking generator output verified that the spectrum analyzer's noise floor is about –75 dBm for the components, filter bandwidths, scan speeds, and so forth I used for these traces. The poor attenuation really happens, even in a simple test setup, so the painstaking component selection and layout recommendations for RF circuits make a difference.

Although they're used nearly everywhere, including my GPS interface, it should now be obvious that 100 nF capacitors really aren't appropriate, even for frequencies well below the VHF range.

BYPASSING BEYOND VHF

I used 100 pF SMD capacitors for the radio interface's audio cables, where the higher circuit impedance would form an RC filter with 100 nF caps that would roll off the audio response. That might not affect what I hear, but the rest of the family has better ears!

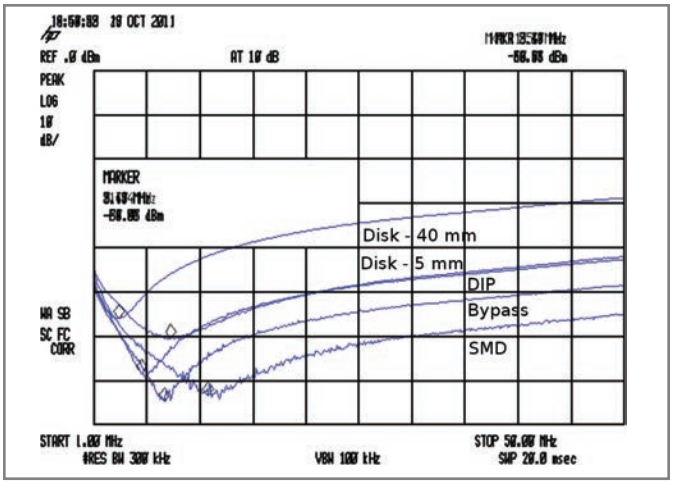
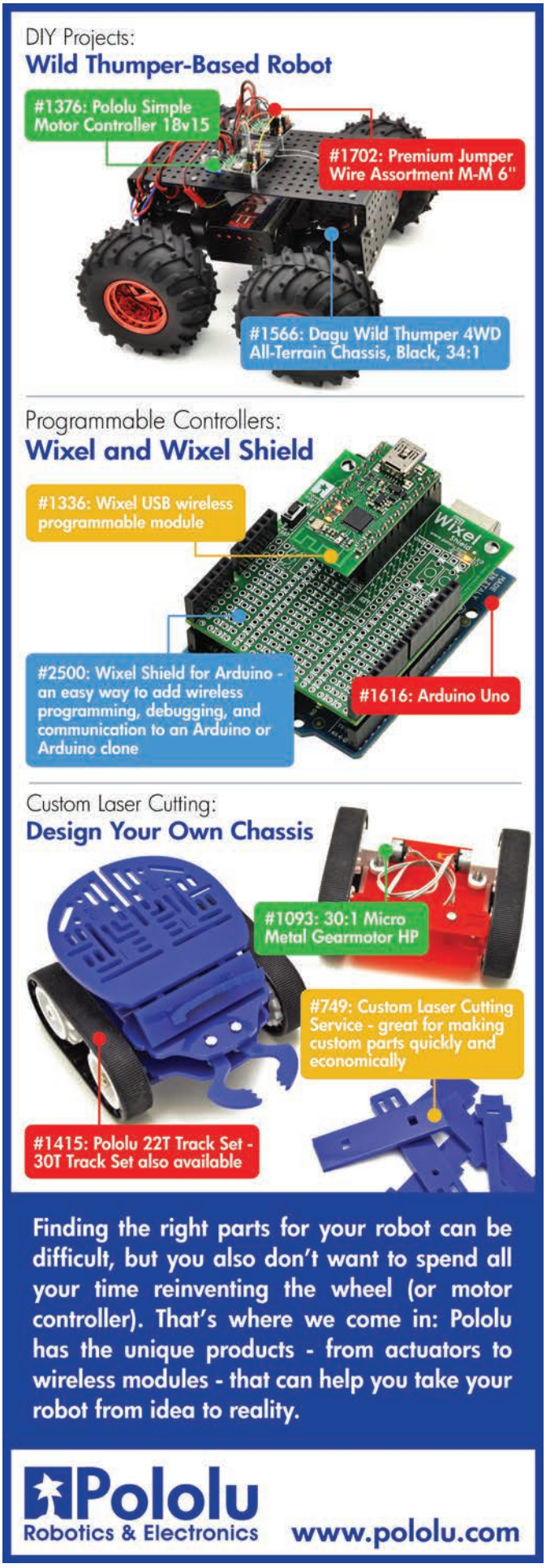
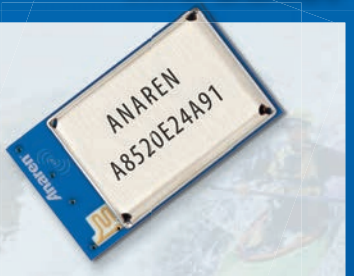


Photo 8—A closer look at the self-resonant peaks shows that the disk capacitor with 40-mm leads is essentially useless above 50 MHz, where the SMD cap is 25 dB more effective.



Low latency. One, tiny wireless audio module.



Need high-quality audio in a miniature package? The A8520 Anaren Integrated Radio (AIR) module offers:

> Uncompressed wireless digital audio/full CD quality (44.1/48KHz/16 bits)

> Robust RF protocol – with virtually no RF engineering experience necessary

> Tiny 11 x 19 x 2.5mm form factor

> Integral antenna

> Pre-certified/compliant: FCC, IC, ETSI (pending)

To learn more, write AIR@anaren.com, visit www.anaren.com/air, or scan the QR code with your smart phone.

TI Developer Network Low Power RF

Anaren®
What'll we think of next?®
800-411-6596
www.anaren.com
In Europe, call +44-2392-232392

Available from:
ARROW ELECTRONICS, INC. AVNET MOUSER ELECTRONICS

The spectrum analyzer trace in Photo 9 shows a 100 pf cap resonates at 303 MHz, where its parasitic inductance L = 2.75 nH and impedance X_C = 5.3 Ω. The 5.6 MHz bandwidth shown in Photo 10 indicates Q = 54 and ESR = 100 mΩ. This is obviously a much better capacitor, with a higher Q, than the 100-nF caps.

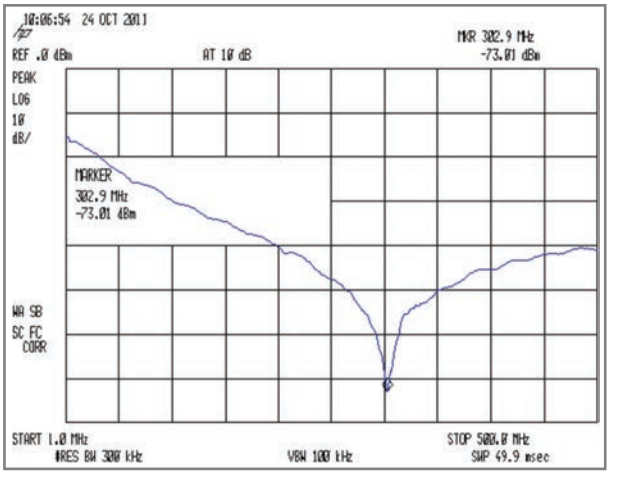


Photo 9—A 100-pF SMD capacitor self-resonates at 303 MHz .

However, the 100 pF cap's attenuation at lower frequencies is correspondingly lower: it's a worse VHF filter than the 100-nF SMD cap and becomes essentially equal to an open circuit below 50 MHz. In effect, its higher Q makes the 100 pF cap a better notch filter, which isn't what you want for broadband RFI suppression.

So the principle of using a single capacitor to suppress wide-band noise, particularly from the VHF band upward, fails on the reality of self-resonance. This isn't unexpected, but I was surprised at both the relatively low self-resonant frequencies for common 100 nF caps and the narrow notches formed by SMD caps.

It's now obvious why RF circuits must have multiple bypass capacitors in parallel, often with seemingly odd values: you choose each capacitor to put its self-resonance at a specific frequency that requires high attenuation. In effect, these bypass capacitors act as prepackaged RF notch filters, rather than ideal capacitors, and you choose each part based on its self-resonant frequency rather than its capacitance.

Ceramic capacitors have notoriously broad tolerances and high temperature coefficients that make them poor choices for RF filters that require specific, stable center frequencies and bandwidths. Those applications demand better components, although self-resonance will certainly affect the design.

CONTACT RELEASE

My next column will examine how parallel capacitors behave in the test fixture, then add long audio-grade cables to the circuit. Although my Icom Z1A interface board worked well enough, I expect the Wouxun version will be better.

You'll certainly measure different results for capacitors in your circuits due to the effects of component variations, plus wiring length and routing. Even if you don't have a spectrum analyzer, though, a few measurements at interesting frequencies should give you an idea of how well your RFI suppression works.

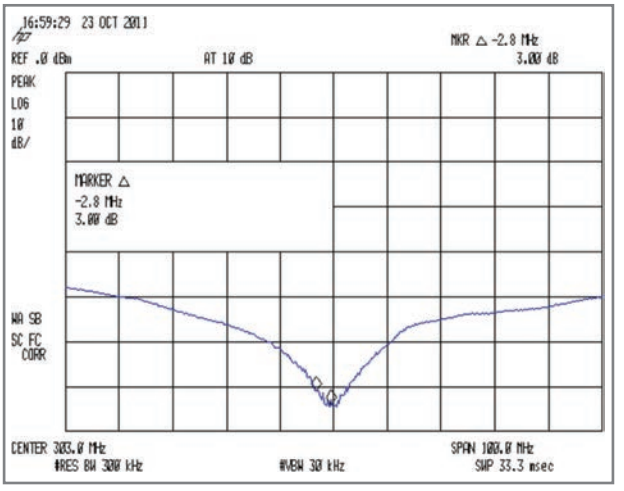


Photo 10—The 100-pF SMD capacitor has a much higher Q and lower ESR than the 100-nF parts.

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 measurement data and script, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/259.

RESOURCES

B. Bruninga, "Automatic Packet Reporting System (APRS)," WB4APR, www.aprs.org.

W. Hayward, W7ZOI, "Some Thoughts on Breadboarding," 2007, http://w7zoi.net/bboard.pdf.

Icom America, Inc., icomamerica.com/en/amateur.

E. Nisley, "Capacitors at RF," *Circuit Cellar* 175, 2005.

———, "HT Audio, GPS for APRS, and What Works," *Circuit Cellar* 219, 2008.

M. Robinson, Fun With Tubes Discussion Group, "The Best Breadboard for Vacuum Tube Circuits," 2002, www.angelfire.com/electronic/funwithtubes/New_Tube_BB.html.

SOURCES

TinyTrak 3+ GPS encoder
Byonics | www.byonics.com/tinytrak

LTSpice IV
Linear Technology Corp. | www.linear.com/designtools/software

Wouxun KG-UV3D handheld amateur radio
Powerwerx | www.powerwerx.com/wouxun-radios

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:

—

Accurate Capacitance Meter
by Alexander Popov, Jordan Popov, & Peter Popov
***Circuit Cellar* 179, 2005**
This comparable capacitance meter is built around a CY8C27443 MCU. Topics: Capacitance Meter, PSoC, LCD, DC, Current Source

—

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

System on Module
Internet Appliance Engine

- Atmel ARM9 400Mhz CPU
- 6 Serial Ports & 2 SPIs
- Up to 60 Digital GPIOs
- 10/100 BaseT Ethernet
- SODIMM Bus Expansion
- SSC/I2S Audio Interface
- Linux with Eclipse IDE
- Real Time Clock Calendar
- SD/MMC Flash Card Interface
- Up to 512 MB Flash & 64 MB RAM
- 2 USB 2.0 Host Ports & 1 Device Port
- 4 10-Bit A/Ds & 6 16-Bit Timer/Counters



2.6 KERNEL
Microsoft Windows CE

The SoM-9G20 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: Flash, Memory, Serial Ports, Ethernet, I2S Audio Interface, PWMs, Timer/Counters, A/D, digital I/O lines, Clock/Calendar, and more. Like other modules in EMAC's SoM product line, the SoM-9G20 is designed to plug into a custom or off-the-shelf Carrier board containing all the connectors and any additional I/O components that may be required. The SoM approach provides the flexibility of a fully customized product at a greatly reduced cost. Single unit pricing starts at $150.

http://www.emacinc.com/som/som9G20.htm

Since 1985 OVER 24 YEARS OF SINGLE BOARD SOLUTIONS

EMAC, inc.
EQUIPMENT MONITOR AND CONTROL

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

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

by George Novacek (Canada)



Build a Robust Power Supply (Part 2)

Abnormal Surge Voltage

In the first part of this two-part article series, you learned how to make a power supply operate properly when the primary power distribution bus voltage is lower than nominal. This article details what happens when it encounters an abnormal surge voltage.

Last month we examined how to make a power supply operate correctly when the primary power distribution bus voltage is lower than nominal. Now let's see what happens when it encounters a condition called abnormal surge voltage. This is not unusual on battery-backed power buses as found in automobiles, for example, and is the result of events such as clearing of system faults, inductive load switching, etc. There are several categories of the power supply hardness. On a 12-V bus, the surge can be as high as 30 V and can last for seconds. What can you do about it?

SURGE & PROTECTION

Look at Figure 1, which is the same as in Part 1 of this series. Depending on the characteristics of the surge, which depends on the system characteristics and its environment, choke L1, transzorb D5, and C1 may be sufficient to suppress it. The transzorb must clamp the input voltage below the maximum allowable input to the regulator module. However, if it is also required to clamp voltage spikes and transients caused by lightning, the transzorb alone may not be capable of dissipating all that power. In this case, you must use a

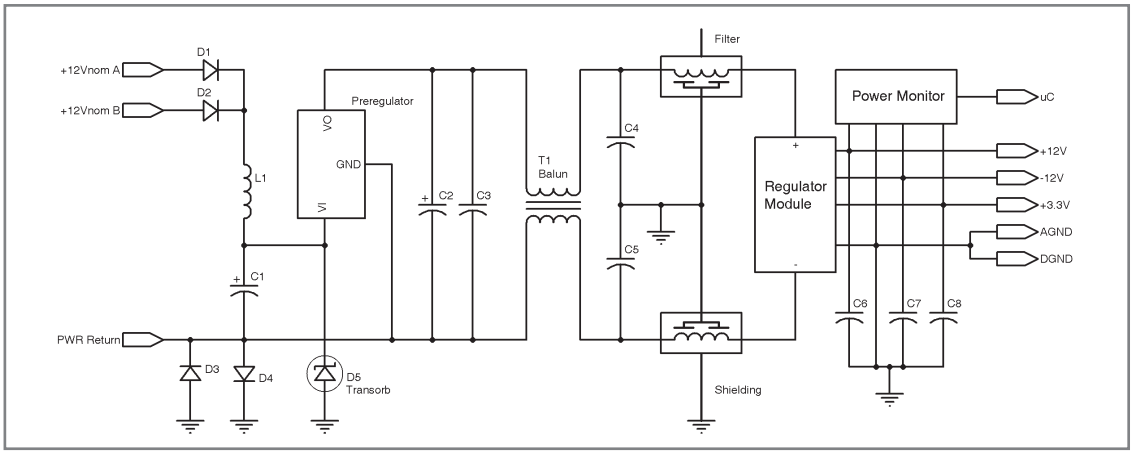


Figure 1—A full-featured power supply

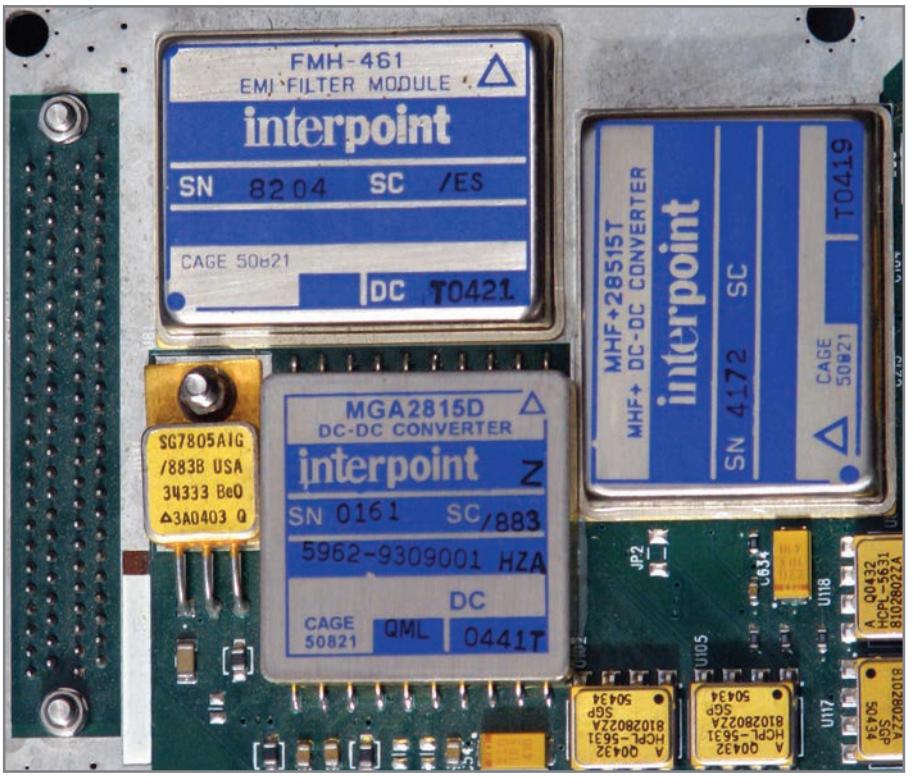


Photo 1—Two switching regulators and an EMI filter are the business end of the power supply. The external shield has been removed.

two-stage protection. Typically, a pre-regulator will maintain the regulator module's input voltage at a safe level. The transzorb with the choke and C1 are acting as the first stage protection, clamping the input voltage at a level safe for the pre-regulator (e.g., 80 V for Linear Technology's LT4356 surge stopper). For a higher voltage, you may have to design your own with discrete components.

While the system grounds—including the negative terminal of the battery—are connected to the chassis, it is a bad idea for modern electronic subsystems to use the chassis for power return (G. Novacek, "My Analog World: The Significance of Grounding," *Circuit Cellar* 244, 2010). The chassis should be used for transients and electromagnetic interference (EMI) returns only. Diodes D3 and D4 ensure that during heavy transients the voltage difference between the power return and the chassis doesn't exceed one diode drop, because two points on a chassis are rarely on the same potential. Consequently, a heavy surge current may be flowing through the diodes, which need to be rated accordingly. Using two transzorbs in place of the diodes is sometimes preferred for the transzorbs's speed and ability to dissipate high-pulse energy in a small package.

Switching regulators are notorious for generating a lot of EMI which propagate not only by radiation but also by conduction through the power leads. The module needs to be shielded, while balun (also called a common-mode choke) T1 together with capacitors C3, C4, and C5 filter the common mode-switching noise. Two feed-through π filters attenuate the differential EMI (see Photo 1).

Off-the-shelf regulator modules may include some RF filtering, but in my experience it is usually insufficient. Therefore, external filters need to be added. The same applies to the output lines, which need additional bypass capacitors, such as C6, C7, and C8 as shown.

EMI STANDARDS

There are numerous standards guiding the EMI levels the equipment may generate and levels it must be immune to. Although the standards are similar, many industries—including medical, automotive, aerospace, commercial, and military—have their own versions, which, because of different test methods, are not always comparable. Devices for attenuation of the EMI are shown in Figure 1. The capacitors, chokes, common-mode choke, and filters do a good job. But sometimes even metal-enclosed switchers—such as those shown in Photo 1—need additional shielding to reduce unwanted emissions.

Power supplies are also exposed to voltage spikes arriving on the input power leads. They can cause permanent damage, insulation breakdown, or degradation of the equipment. A typical spike is depicted in Figure 2.

POWER SUPPLIES

Note that 200-V spikes are not uncommon and can cause serious damage. Fortunately, the spikes are narrow and, therefore, of limited energy. To effectively suppress them, C1 has to be a low (equivalent series resistance) ESR type. It is not unusual for C1 to be formed by a combination of several capacitors, from large electrolytics for the energy storage down to small devices to effectively bypass high frequencies.

In the first part of this series, I mentioned the input voltage ripple, which is due to the rectification of the alternator voltage in the 12-V automotive system. In addition to this ripple, switching transients and even audio frequencies may

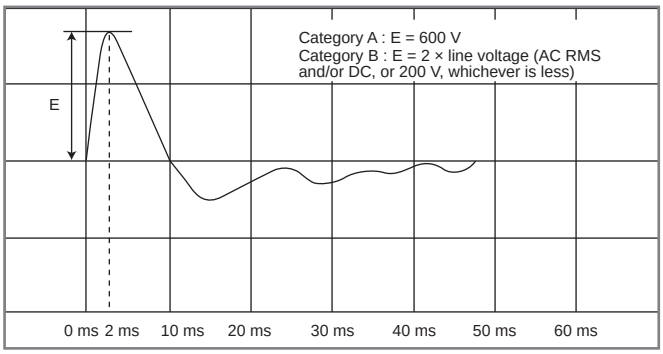


Figure 2—Voltage spikes often experienced by power supplies



▶ 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

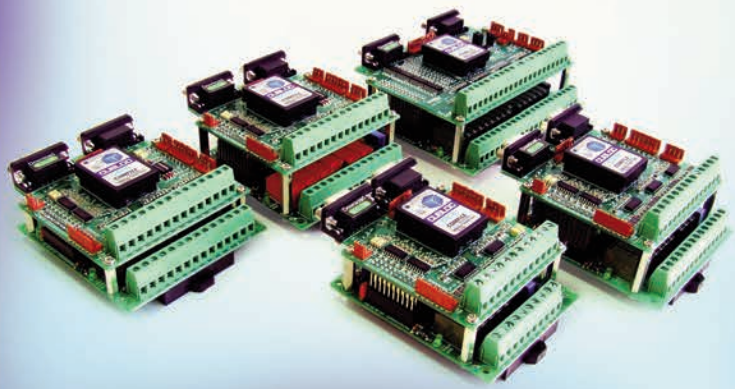
▶ CUPC-P220

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

be induced onto the input wires by external magnetic fields. Up to 50 Hz or 60 Hz from close-by transformers are common, but also higher frequencies from sources such as alternators, speakers, and others—many of them with high harmonic content. Measurement of susceptibility to such interference is performed by injecting test frequencies up to 15 kHz into the power leads through a transformer. Their effective suppression depends mainly on the quality of L1, C1, and grounding.

Not all embedded controller power supplies work with battery-backed sources. Many power supplies use a transformer followed by a rectifier. Nevertheless, the principles of the design verification to ensure the power supply is robust and not the cause of unexpected behavior remain the same. Some components in Figure 1 may be eliminated, others may be added, and the input voltage range and power interrupt may be recalculated (all based on the primary power source). Input transients or electromagnetic compatibility (EMC) may be more or less of a concern. The designer must carefully analyze all the characteristics of the primary power source and then take the steps to satisfy the application.

THD

Few embedded control engineers need to worry about primary AC power phase displacement, voltage and frequency modulation, and other issues seen in specialized AC supplies that are beyond the scope of this column. One issue, however, has become very important for power supplies working from an AC distribution net. It is the total harmonic distortion (THD) of the input power waveform. It can be calculated as:

$$THD = 100 \times \frac{\sqrt{\sum_{n=2}^{\infty} X_n^2}}{X_1}$$

where X represents current or voltage, X_1 is the value at the fundamental frequency, and X_n are harmonics. The main cause of the THD is the rectifier charging the storage capacitor, causing distortion of the input AC current. The result is a low power factor and its repercussions are basically I^2R losses. Some standards and specifications call for maximum allowed THD; others call for the minimum power factor. Both mean the same thing. Meeting those requirements often requires a power factor correction (PFC) circuit.

POWER SUPPLY DESIGN

I hope you found this article series useful. My goal was to show you that a seemingly mundane job—such as power supply design—often evolves into a fairly complicated issue and must not be taken lightly.

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.*

RESOURCES

G. Novacek, "Building a Robust Power Supply (Part 1): Power Dissipation," *Circuit Cellar* 258, 2012.

———, "My Analog World: The Significance of Grounding," *Circuit Cellar* 244, 2010.

Department of Defense, Interface Standard, "MIL-STD-461 Requirements for the Control of Electromagnetic Interference Characteristics of Subsystems and Equipment," 1999, www.tscm.com/MIL-STD-461E.PDF.

"MIL-STD-704, Aircraft Electric Power Characteristics," 2008, https://assist.daps.dla.mil/quicksearch/basic_profile.cfm?ident_number=35901.

Radio Technical Commission for Aeronautics (RTCA), "New Revision G on Environmental Conditions and Test Procedures for Airborne Equipment," DO 160, 2010, http://standardsforum.com/?p=464.

SOURCE

LT4356-1 and LT4356-2 Surge stoppers
Linear Technology, Inc. | www.linear.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:

—

Portable Power
A Power Supply for Embedded Applications
by Jason Wu, Kiran Kanukurthy, and David Andersen
***Circuit Cellar* 193, 2006**

This team of designers built an inductively charged power supply for embedded applications. The portable system provides 100-mA, 3.3-V continuous power. Topics: Power Supply, LTC1325, Wireless, Inductive Charging, ATmega8, LM2621, Coil

—

Quad Bench Power Supply
by Brian Millier
***Circuit Cellar* 149, 2002**

How can you deliver an adequate power supply while keeping workbench clutter to a minimum? Brian decided to tackle this classic engineering conundrum with a small, low-capacity quad bench power supply. Topics: Power Supply, Floating, Metering, LCD, IRL520, TL082, MCP42010, MCP3202, ZXCT1009, Isolation, GMR, IL717, AT90S8535, SIMM100

—

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

QUESTIONS & ANSWERS

An Analog Life (And a Lot More)



An Interview with Chris Gammell

Chris Gammell wears several hats (and safety goggles, when necessary). He currently runs a consulting business, maintains his own website, contributes to and administrates an engineering blog, and co-hosts *The Amp Hour* podcast with Dave Jones. In December, we discussed his many interests from FPGA-based projects to hardware design to the benefits of writing about what you know.—*Nan Price, Associate Editor*

NAN: Where do you live and work?

CHRIS: I live and work in the suburbs of Cleveland, OH. Different suburbs though.

NAN: When did you start working with electronics? Was it something that piqued your interest when you were young?

CHRIS: No. Unfortunately, I was a late bloomer, as are many in my generation. I had limited exposure to electronics until halfway through high school. But once I learned it, I loved it! I wrote about my entrance to electronics on Engineer Blogs (http://engineerblogs.org/2011/02/fits-and-starts).

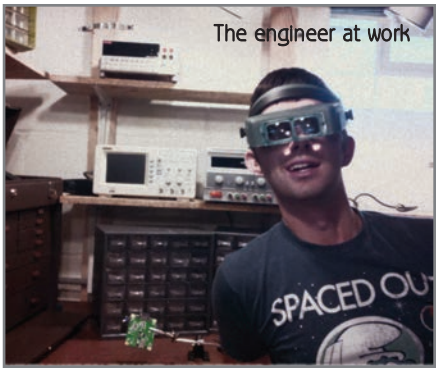
NAN: What types of microcontroller-based systems do you like to design?

CHRIS: I'm not as much a microcontroller person so much as an FPGA-with-the-occasional-embedded-micro kind of person. I love parallel logic and designing hardware, even if it's on-chip hardware. I have done some little projects using Texas Instruments's MSP430 microcontrollers and Arduinos, but nothing to speak of. I'm trying out a BeagleBone right now though, and I'm very excited about it!

NAN: Give us some background on your education.

CHRIS: I went to Case Western Reserve University (CWRU) right here in Cleveland, OH. I think I have a penchant for gloomy weather or something. (I grew up in Buffalo to top it off!) However, I haven't been here the whole time. I moved to Austin, TX, for a few years to try out some different types of work and more sunshine than I had seen in the previous 20 years. My degree in school was a BSEE, and I focused on signals and circuits. I enjoyed CWRU a lot, for the people and for the opportunities afforded me.

NAN: Can you tell us about some of the electronics projects you worked on in college?



CHRIS: My final project was actually another FPGA-based design. Me and a couple buddies worked on an audio-to-MIDI converter. The project was completed successfully but sat in a drawer henceforth, so it never quite went anywhere. Like a lot of college students, we had dreams of taking it to mass market before we realized it was a crude idea and a product that no one wanted. Ah, the tough realities of marketing and product design. Engineers have to learn them some time, I suppose.

NAN: You are busy. You have a website and podcast. You are present on Twitter. Let's start with your company, Analog Life, LLC. What services do you offer?

CHRIS: Analog Life, LLC is my part-time design services consulting business. I still work full time as an engineer all day. I've only had a few clients so far, but it's been quite rewarding. It really allowed me to branch out to different industries and work with some great people. I tell people that I consult on analog projects mostly, but that's a bit of a sham. My last project was almost exclusively an FPGA-based project—and I loved it!

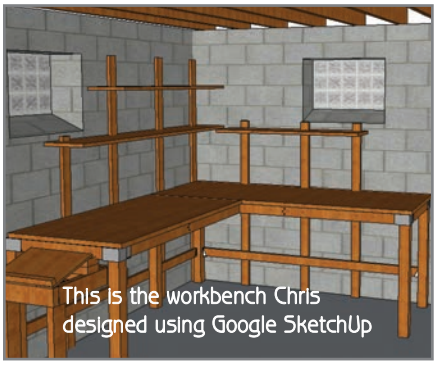
NAN: Tell us about some of the work-related projects you've done.

CHRIS: My first consulting project involved low-level signal processing for a light-detection system. That was really a fun project, and I was working with another great engineer who helped me along in my fledgling consulting career. He provided the digital support and the kind responses to my terrible billing practices; I provided op-amp designs and part evaluations. A truly symbiotic relationship.

NAN: Your website—*Chris Gammell's Analog Life: Analog Electronics and Everything Else Between 1 and 0* (www.chrisgammell.com)—features useful information about electrical engineering. In the past year, you've covered some interesting projects: building an electronic workbench, putting an MP3 player and output stage board into an antique radio enclosure using KiCAD, and more. Tell us a little about those two projects. What was the impetus for each?

CHRIS: The first is pretty simple to explain: I was sick of working on an old tabletop that kept getting cluttered! I decided to design and build a custom workbench for my basement, based upon a design I cobbled together in Google SketchUp. A friend from work who had built similar workbenches in the past stopped by and helped me construct the frame. I did all the finishing and electrical work, an oft overlooked branch to most electrical engineers. I'm always amazed at how I can know the intricacies of a complex analog and digital circuit...and a light switch can still manage to trip me up occasionally!

The second is a work in progress. I wanted to have a way to pipe our podcast through a vintage radio, just for the look of it all. And while I was designing a new circuit, why not try out KiCAD, a program I had never previously used? I also thought I could get a couple of videos out of it. It turns out my ambition turned into a much longer project than I thought. So, like any engineer, I continue to push out the deadline and thank my lucky stars I don't have a manager on this project pushing me to get it done!


This is the workbench Chris designed using Google SketchUp

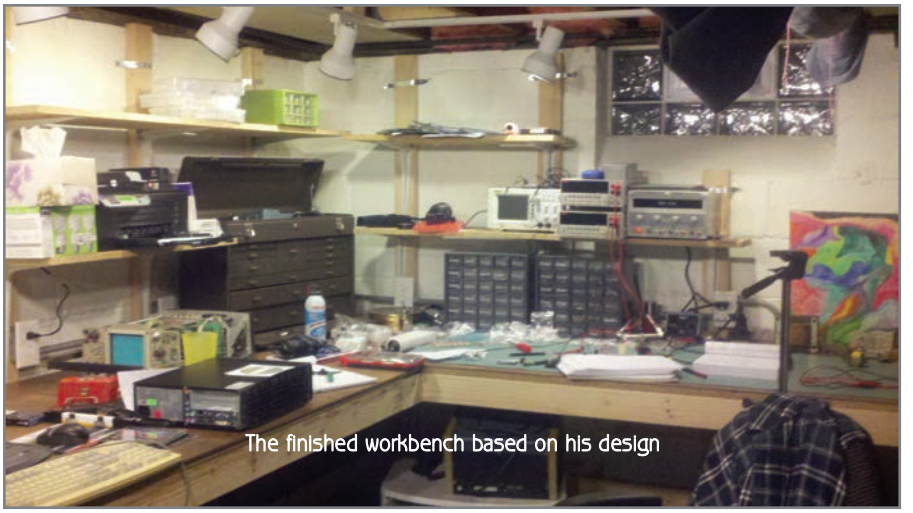
NAN: What inspired you to start your own site? Do you receive a lot of reader feedback? What types of questions/comments do you typically receive?

CHRIS: I started ChrisGammell.com back in 2007 for all the wrong reasons: I thought I was going to make money from it. To date, I still haven't made any money from my websites, namely because I don't like advertising being on the site. Also because I really don't get that much traffic. The most-viewed article on my personal site is actually about how op-amps work, so I get a lot of beginner questions, especially if they have found me through Google. While I'm glad my articles help people out, I've moved on to other projects and don't post as often on my personal site anymore.

Once I wised up and money no longer was the motivation behind my site, I took to what many people call "personal branding." I'm not such a big fan of the term, and my opinion of it has changed over the years, but the basic idea is that you put out articles and other knowledge you might have so you are easier to find on the web. Or perhaps you become better known. My new thinking about this is that you should put out articles and write about things that interest you because you enjoy it and it keeps your mind sharp. The rest will come with time, especially if you have quality posts. Some of my favorite sites are on my RSS reader simply because they continue to post good articles or videos, week after week.

Yet another advantage—and a lesson I picked up from other consultants—is that writing about what you know and adding insight to the topic is a great way to advertise your services. If people are looking for microcontroller experts and you have just written a great piece on the finer points of programming or troubleshooting a microcontroller, that will help a potential client find you and be confident in your work. This is another reason I always encourage people to post technical and other articles under their real names on the Internet. Your article about JTAG chains today could help you get a client one, two, or 10 years down the line. You just never know.

NAN: Along with some other engineers, you also started EngineerBlogs.org in 2011. Tell us a little about that site.


The finished workbench based on his design

CHRIS: Engineer Blogs is the main place I post these days. I love that other engineers are interested in collaborating on a common site. It's mostly "softer" topics of engineering, namely because we're a common site for *all* types of engineers. While my circuit design experience might not be relevant to a mechanical engineer, I can guarantee the more general design principles (e.g., prototype early and often) are quite relevant.

Similar to my personal site, we don't take advertising and there is no monetary benefit to any of the writers. So it's mostly for personal development and "branding." (There he goes again!) But really, it's nice to have others to bounce ideas off of and have people who provide genuine feedback. The social pressure also keeps me on my toes and consistently posting an article per week. We just started taking guest bloggers, so if people are interested in joining up and trying out their literary skills, please get in touch.

NAN: What was the impetus for starting your radio show, *The Amp Hour* (www.theamphour.com) which you co-host with Dave Jones of the *EEVblog*?

CHRIS: I'm a member of the online community reddit, and actually an administrator of the ECE sub-group (http://reddit.com/r/ECE). Two years ago, there was a request on the site for podcasts and blogs featuring electronics, because the submitter wanted to learn more while in the car and on their bike. I tried a podcast by myself and it was dreadful. But once I posted it, Dave Jones happened to hear it and suggested we give it a shot together. We didn't have much of a plan at first, just to get on the line and shoot the breeze for an hour or so. After the first two episodes, we began soliciting our listeners and Dave's regular viewers for a name for the show. One of our listeners suggested "The Amp Hour" and it's been roughly an hour-long show every week since. It's one of my favorite times of the week and I'm so glad I have been able to continue doing the show, 70-some episodes later. If your readers haven't had a chance to listen, I hope they will at some point in the future. I personally think the best time to listen to the show is on the way to or from work, or while sitting at your bench soldering.

NAN: Some recent *Amp Hour* interviewees included: Jeff Keyzer of the blog MightyOhm.com (December 2011); John Edmond, co-founder and director of Advanced Optoelectronics Technology at Cree (November 2011); and Joe Grand, president of Grand Idea Studio and a former member of the legendary hacker collective L0pht Heavy Industries (September 2011). How do you decide who to feature on the show? Can you tell us who has been your favorite or most memorable guest?

CHRIS: Your embedded programming readers out there undoubtedly know Jack Ganssle, who was a guest on our show. It was a ton of fun trading stories with him and hearing how the computer industry used to be driven by the demands of the... ahem...online adult industry. But really, all of our guests have been amazing. We choose guests who not only have interesting stories and work but also can chat about the weekly news and who enjoy electronics. I'm sure a couple of members of the *Circuit Cellar* audience will be in our sights as potential guests in the future!

NAN: Is there a particular electronics engineer, programmer, or designer who has been the inspiration for the sort of work you do today?

CHRIS: As I often talk about, analog engineering is the kind of stuff I usually specialize in. And, up until a few months ago, I had two shining beacons, not just of analog electronics but also technical writing. Bob Pease and Jim Williams might not be known as much to the microcontroller crowd, but you won't find many analog guys and gals who didn't know their names. Both were prolific writers and great engineers. Jim passed away from a stroke and Bob died tragically in a car accident on the way home from Jim's memorial service. It was a great loss for the industry and I'll miss their wit and insight.

I still have many others I look up to in the industry though, and I hardly have to look outside the workplace. Though not famous, I've had the opportunity to work with some wonderful engineers who have molded me and helped me on my journey, and I'm appreciative of all of them.

NAN: Are you working on or planning any other projects—including websites, podcasts, or blogs—for the near future?

CHRIS: Nothing new for the foreseeable future, only because of my newly started job (which I wrote about on Engineer Blogs http://engineerblogs.org/2011/12/weekend-journal-starting-an-engineering-job). I will be designing large-scale systems and the workload might be significantly higher than my last job, something I'm looking forward to. With great responsibility, comes great hours spent at the office!

NAN: Like some of *Circuit Cellar*'s staffers, you travel to embedded-related conferences to check out the new technologies on the horizon. What embedded technology have you seen or worked with recently that really excited you?

CHRIS: Because I'm such a fan of FPGAs, I'm really interested in how they're once again pushing hard processors back into chips and then surrounding it with programmable logic (Xilinx Zynq or the Altera Arria V). I think there are a lot of benefits to having customizable logic around a microcontroller and I hope that normally micro-only manufacturers begin integrating programmable logic as well. The benefits of parallel processes and offloading recurring tasks are often under emphasized, in my opinion.



INTERNATIONAL
CTIA WIRELESS 2012®
A Division of **CTIA**–The Wireless Association®

All things mobile period

CTIA is All Things Mobile. Now with new dates in May, **International CTIA WIRELESS 2012** is even better positioned to meet the needs of the global wireless marketplace. This bold move is more than a change of date. It is the *best* opportunity to create a more personalized, powerful experience at the one place for carrier services, enterprise, communications, apps, content and social networking.

New Dates!

Pre-Conference Events May 7, 2012

Convention & Exposition May 8–10, 2012

Ernest N. Morial Convention Center New Orleans, LA

Get Connected!



WWW.CTIASHOWS.COM

by Robert Lacoste (France)



Playing with High-Speed ADCs

Analog-to-digital converters (ADCs) have come a long way since their inception back in 1954. But they aren't always easy to use. Here you learn to use high-speed converters effectively.

Welcome back to the Darker Side! According to Walt Kester in his great data converter handbook, *The Data Conversion Handbook* (Analog Devices, Inc., 2004), the first commercial analog-to-digital converter (ADC) was introduced in 1954 and was designed by Bernard M. Gordon from EPSCO. The so-called "DATRAC" was a full vacuum tube 50-ksps 11-bit ADC, weighting 70 kg and requiring no less than 500 W.[1] The ADC was sold for $8,500 at that time, which translates into about $70,000 in 2011. Impressive, isn't it?

Today, you can buy a 125-Msps 16-bit ADC, such as the AD9265-125 from Analog Devices, for $65 in medium volumes. I made the calculation for you, this translates into an 86-million times improvement in terms of performance to cost ratio. Does this mean high-performance ADCs are now easy-to-use commodity devices? You bet not. The ADC is just one part of the system, as illustrated in Figure 1. Do you want to avoid having your boss ask you why the product you've designed with a costly 16-bit ADC is not significantly better than the competitor's 8-bit based device? This month, I will do my best to give you some advice on how to efficiently use high-speed ADCs.

DECODING ADC DATASHEETS

Let's start with the ADC itself. You must read the datasheet carefully, but you may have difficulties understanding what it's all really about. As an example, Figure 2 gives you the key performance characteristics of the AD9265-125 I just mentioned. Of course, this chip is only an example, but it is a typical high-end ADC from one of the leading ADC vendors. What do all these figures mean? Let's review them one by one, starting with the DC specifications.

First, no missing code is guaranteed. This means that all $2^{16} = 65{,}536$ possible output values

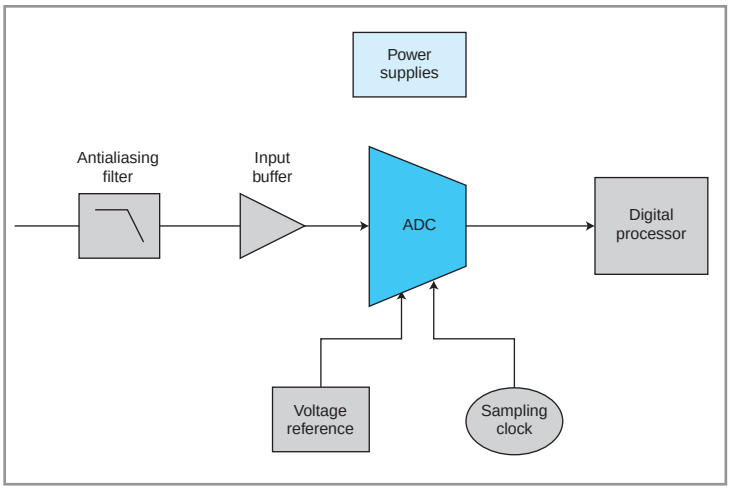
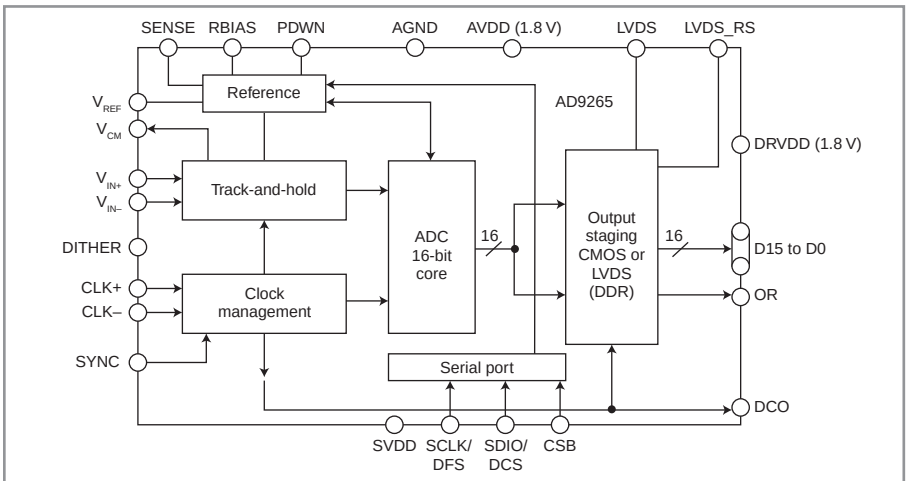


Figure 1—A high-speed ADC is just a part of a high-performance digitizing system. All the other building blocks should be designed carefully if you don't want to jeopardize the performances.



AD9265-125 from Analog Devices
16-bit, 125-MIPS, 1.8-V analog-to-digital converter

Some DC specifications (typical at 25°C):
No missing code: Guaranteed
Offset error: +/– 0.05% FSR
Gain error: +0.4% FSR
Gain temperature drift: +/–15 ppm °C
DNL +/– 0.7 LSB
INL +/– 3 LSB
Input referred noise: 2.17 LSB ms
Input impedance: 6 kΩ + 8 pF

Some AC specifications (typical at 25°C, 70 MHz, 125 Msps):
Analog input bandwidth: 650 MHz
SNR = 79.0 dBFS
SINAD = 78.7 dBFS
ENOB = 12.8
SFDR = 93 dBc

Figure 2—I selected the AD9265-125 high-end ADC from Analog Devices. Here are some of its key characteristics, as extracted from the datasheet.

will at a given time be present on the outputs when the input voltage is increased slowly enough. Great.

The DC gain and offset error are more interesting. The typical gain error is, for example, 0.4% of the full scale. That may seem low, but this figure is huge compared to the resolution of such an ADC which is 0.0015% (i.e., 1/65,536). In practical terms, this ADC—much like all high-end fast ADCs—provides a very good resolution (16 bits) but a poor accuracy of 0.4%. Think twice, this accuracy is worse than even the ±1-least-significant bit (LSB) gain error of the 10-bit ADC built into low-cost Microchip Technology PIC microcontrollers. Why? Simply because the input frequency range is orders of magnitude higher. Just try to develop a DC-to-100-MHz amplifier with a gain error lower than some percent.

Next, the differential nonlinearity (DNL) is ±0.7 LSB. That means that the input voltage difference required to increase the output value by one step is guaranteed to be 1 LSB ± 0.7 LSB, or between 0.3 to 1.7 LSB (see Figure 3). The important point is that this DNL is lower than 1 LSB, which implies that the converter is monotonic: the output value is always higher for a higher input voltage. Be careful, this is not always the case in particular for flash-type ADCs. The integral nonlinearity (INL) is a related specification. Measure and plot the relationship between the input voltage and output code and compare it with a theoretical straight line. The maximum difference is the INL.

Noise is also an important topic. Our example ADC has an input referred noise of 2.17-LSB root mean square (RMS). That means that even if a stable DC voltage is applied on the input, the measured value will show random variations of 2.17-LSB RMS around the mean value. Be careful as this value is an RMS value. A good estimation of the peak-to-peak variation is 6.6 times the RMS value, which is 14.3 LSB, a value close to 16 LSB, or 4 bits. This means that even if the converter is 16 bits, its four LSBs will randomly change, only the highest 12 bits will have a chance to be stable! That doesn't mean that such a 16-bit ADC is not better than a 12-bit variant. The average output value will provide more precise information on the signal thanks to the extra bits, even if they are noisy. When using a

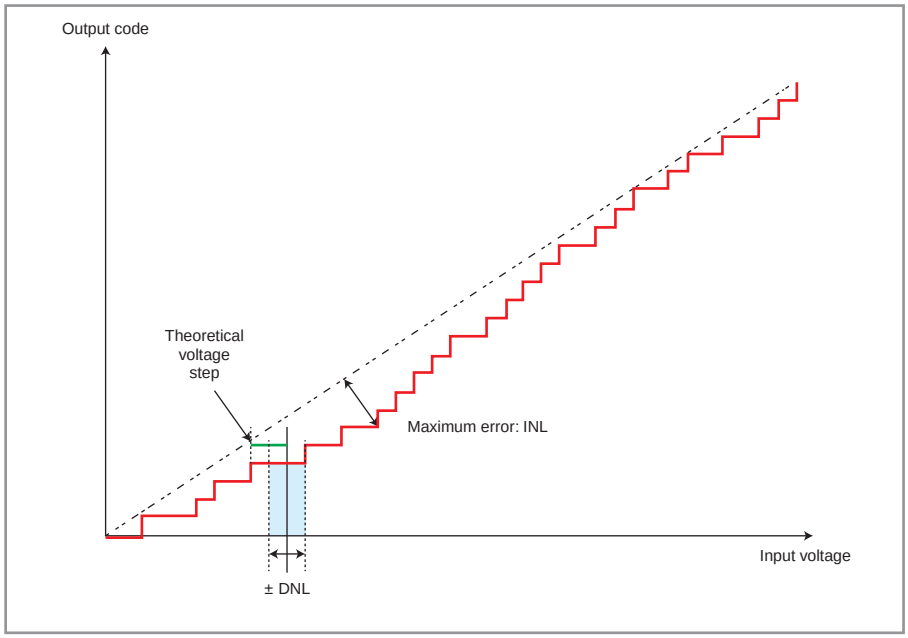


Figure 3—Differential nonlinearity (DNL) and integral nonlinearity (INL) are some key specifications of an ADC DC behavior.

frequency analysis (Fourier transform), these extra bits will also provide a far better dynamic range on the measurement. It should just be used with care.

The last key DC specification is the input impedance of the converter. In our example, it is 6 kΩ = + 8 pF. This parasitic capacitance of 8 pF must not be neglected: With an input frequency of 50 MHz it brings the input impedance down from 6 kΩ to 400 Ω (i.e., 1/2π fC). This has to be anticipated by the designer!

AC SPECIFICATIONS

Okay, this was the easy part; let's move on to the AC specifications. The AD9265-125 has an analog input bandwidth of 650 MHz. Just wait, how can a 125-Msps ADC have such a high input frequency? Shouldn't the input frequency be limited to the Nyquist frequency, which is half the sampling rate? No, aliasing will occur if a higher frequency is sent to the ADC, but this can be really useful. For example, a 125-Msps ADC can't differentiate between a 10-MHz signal and a 115-MHz (i.e., 125 – 10) signal as shown in Figure 4a. This is not a problem if you are sure that it can't be 10-MHz through a proper analog filtering. Basically, with this chip, you can digitize any signal from DC to 650 MHz as long as you are sure that it stays in a 62.5-MHz (i.e., 125/2) window, called Nyquist zones (see Figure 4b). The use of an ADC in a Nyquist zone other than the first one is called undersampling, and is a common use of fast ADCs.

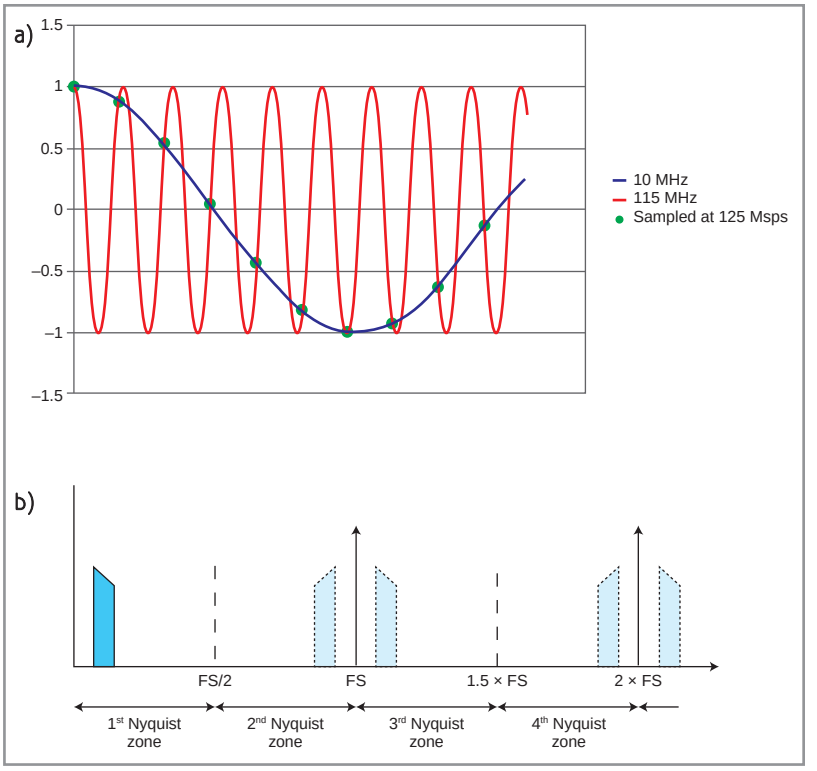


Figure 4a—This plot shows you a 10-MHz sine wave (blue) and a 115-MHz sine wave (red), sampled at 125 Msps. The sampled values (ticks) are exactly the same. This is aliasing. b—This aliasing occurs for signals at corresponding frequencies in different Nyquist zones. The first Nyquist zone extends from DC up to half the sampling frequency, but the other ones can be used too.

The next specification says SNR = 79 dBFS. Roughly speaking, it means that the RMS noise level on the output data is 79 dB lower than the level of a full-scale signal at the indicated frequency, here 70 MHz. 79 dB converts into a power ratio of $10^{79/10}$, or a voltage ratio of 8,900 (i.e., $10^{79/20}$). This has to be compared to the SNR of an ideal ADC measured from DC to the first Nyquist limit, which can be calculated as 6.02 N + 1.76 dB, where N is the number of resolution bits. For a 16-bit ADC, this gives a theoretical maximum of 98 dB. This dynamic SNR, measured with an actual high-frequency signal at the input, is always worse than the DC evaluated "input referred RMS noise," which was, if you recall, only two 2-LSB RMS.

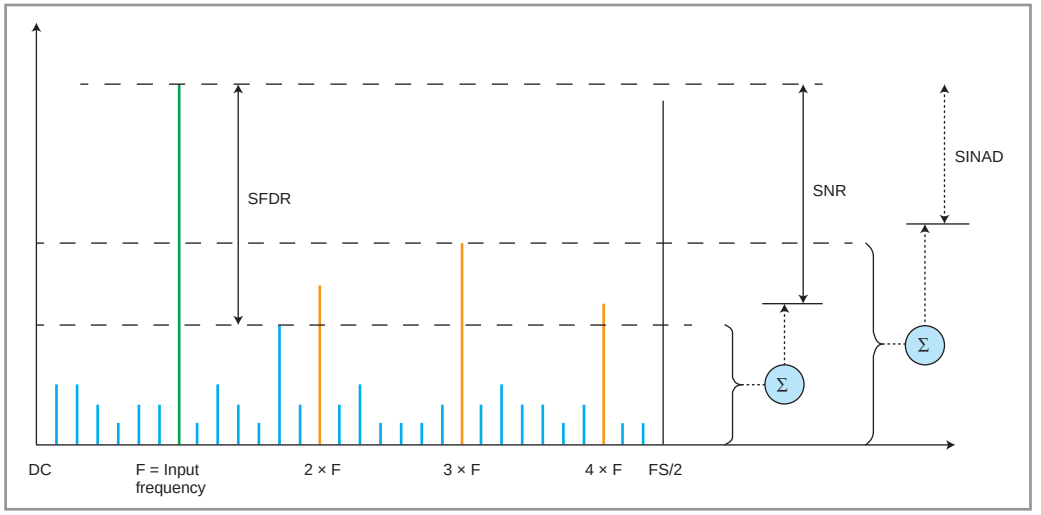


Figure 5—The difference between the three key AC specifications of a fast ADC: spurious free dynamic range (SFDR), signal-to-noise ratio (SNR), and signal-to-noise-and-distortion ratio (SINAD).

Then the specifications state SINAD = 78.7 dBFS. SINAD means signal-to-noise-and-distortion ratio. It is similar to SNR, but includes not only noise but also the unwanted harmonics of the input signal, which are usually quite high. Therefore, SINAD is always worse than SNR. In our example, the difference is very small (78.7 versus 78 dBFS) which means the analog design of this ADC is very linear. The SINAD specification can also be expressed in terms of effective number of bits

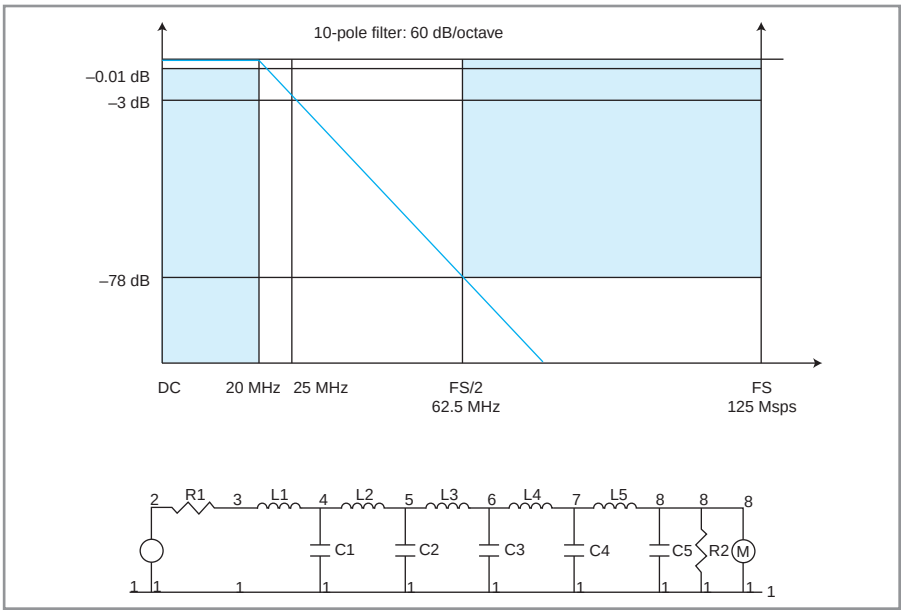


Figure 6—Designing a good antialiasing filter for a high-performance ADC is a nightmare. In this example we need to ensure at least a 78-dB attenuation above the first Nyquist limit. Even when using a complex 10-pole LC filter, its cutoff frequency can't be higher than 25 MHz.

(ENOB), through the following formula:

$$\text{ENOB} = \frac{(\text{SINAD} - 1.76\ \text{dB})}{6.02}$$

Lastly, the datasheet provides the spurious free dynamic range (SFDR) of the ADC, which in our example is 93 dBc. What is SFDR? Imagine you calculate the frequency spectrum of the ADC output data through a Fourier transform or similar. The SFDR is the difference between the input signal level and the highest unwanted spurious signal visible on the output spectrum. So SFDR is better (meaning higher) than SNR and SINAD, as only one frequency component is used to calculate the difference. This figure gives an optimistic view of the usable dynamic range of the chip, when wide-band noise is not a concern. Figure 5 shows you a comparison of these three AC specifications, SNR, SINAD, and SFDR.

Be careful, as all these dynamic specifications are dependent on the input signal frequency as well as on the input signal level. The datasheet usually provides tables or graphs giving SNR, SINAD, and SFDR from DC up to the maximum usable frequency of the chip and for different power levels.

ANTIALIASING FILTER

Now you should be able to read and understand the datasheet of your





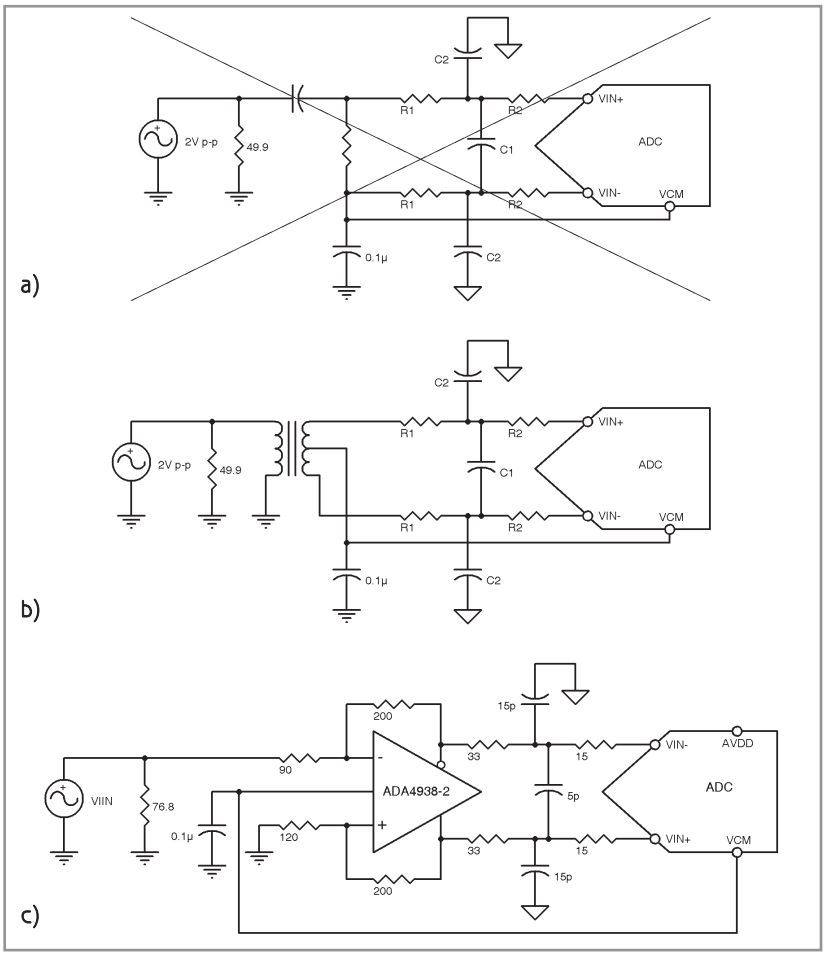


Figure 7a—Using only one of the inputs of a differential input fast ADC is not recommended as it will significantly degrade its performances. b—The best method is to use a coupling transformer as a unipolar to differential converter. c—If you need to go down to DC, then the best solution is to use dedicated differential drivers. (Source: 7b and 7c are based on figures in Analog Devices's AD9265-125 datasheet)

selected ADC and to check that it is compliant with your system requirements. Assuming that it is okay, you now have to design the remaining parts of the schematic. Let's start with the antialiasing filter. As explained in Figure 4, an ADC is a sampling device and can't distinguish between two signals that are equivalent frequencies in different Nyquist zones. With our 125-Msps ADC, a 10-MHz signal will provide the same output datastream as signals at frequencies of 115 MHz (i.e., 125 – 10), 135 MHz (i.e., 125 + 10), 240 MHz (i.e., 2 × 125 – 10), 260 MHz (i.e., 2 × 125 + 10), etc. Therefore, you must ensure that there are significant signals in one and only one Nyquist zone. You might not need an antialiasing filter at all if you are 100% sure of your input signal, but usually you will need to implement either a low-pass filter—selecting the first Nyquist zone, from DC to 62.5 MHz (i.e., 125/2)—or a band-pass filter if you want to use the ADC in undersampling mode. I will then just consider the first case for simplicity.

The design of this anti-aliasing filter is far from obvious if you want the best performance. Just assume that you have no idea at all about the input signal frequency content and you want to use 100% of the ADC performances. As discussed, its SINAD is 78.7 dBFS. Therefore, you need to have a filter providing the same level of performance. It must cut anything above 62.5 MHz with an attenuation close to 78 dB. If not, a spurious signal—say at 62.5 + 10 MHz and at the same power level as your 10-MHz input signal—could jeopardize the impressive ADC SINAD figure you paid for. Similarly, if you don't want to degrade the ADC gain error of 0.4% FSR then the anti-aliasing filter must have a very flat response in its pass-band, better than 0.4% FSR, which translate into about ±0.01 dB. This means that you can't come close to its 3-dB cutoff frequency (see Figure 6).

The problem is that such a filter is difficult to build. Remember that a one-pole filter provides an attenuation of 6 dB per octave. Even a complex 10-pole filter will only provide 60 dB (i.e., 6 × 10) per octave, so the 3-dB cutoff frequency should be 1.3 (i.e., 78.7/60) octaves lower than the 62.5-Mhz band-stop requirement. This means you should use a 10-pole filter with a 25.3-MHz (i.e., $62.5/2^{1.3}$) cut-off frequency. Moreover, the usable frequency range should be lower than this 3-dB cutoff in order to satisfy the ±0.01 dB requirement, say around 20 MHz maximum.

Let's summarize: if we want to use our example 125-Msps/16-bit ADC up to its full performance in the first Nyquist zone, then we need to include a 10-pole 25-MHz low-pass filter and restrict its use to signals below 20 MHz. Quite far from the 62.5 MHz (i.e., 125/2) you were expecting, right? Of course, in actual designs, the specifications could be relaxed as you may know what your input signal is, but I think this example speaks for itself. Moreover, a 10-pole filter is a nightmare to build as it needs precise and high-quality components. That's why it is often interesting to use a high sampling rate ADC even for relatively low-frequency signals: You can then drastically relax the requirements on the antialiasing filter. This is called over-sampling and is the basis of sigma-delta converters. I will talk about in my next column!

INPUT BUFFER

The output of the anti-aliasing filter goes to the ADC input pin. This is another critical section of the schematic. If you are using a high-speed ADC, don't expect to directly connect your input, or your passive filter, to the ADC input. The input impedance of such chips is often quite low. More importantly, it is frequency-dependent, so an external buffer amplifier is required. But once again, take care. High-speed ADCs require very good external amplifiers if you don't want to compromise their performances. You may wonder why manufacturers don't integrate this amplifier on-chip? Well, some do it,

but it seems that due to technological foundry issues it is difficult to embed a good amplifier on the same chip as a CMOS-fast ADC.

You may encounter another difficulty: high-speed ADCs usually have differential inputs, and unfortunately real signals are often unipolar. The easy but dirty solution is to use only one of the ADC inputs and connect the other one to a ground or virtual ground (see Figure 7a), but this drastically reduce the dynamic performances of the ADC. The best solution, often used on the evaluation boards of fast ADCs, is to use a miniature transformer for this unipolar to bipolar conversion, like one of those sold by Minicircuits (see Figure 7b). Unfortunately, this method can't be used if you really need to go down to DC. In that case, you will need to use operational amplifiers, but your life will be far easier if you select dedicated differential ADC driver chips, which are optimized for that purpose (see Figure 7c).

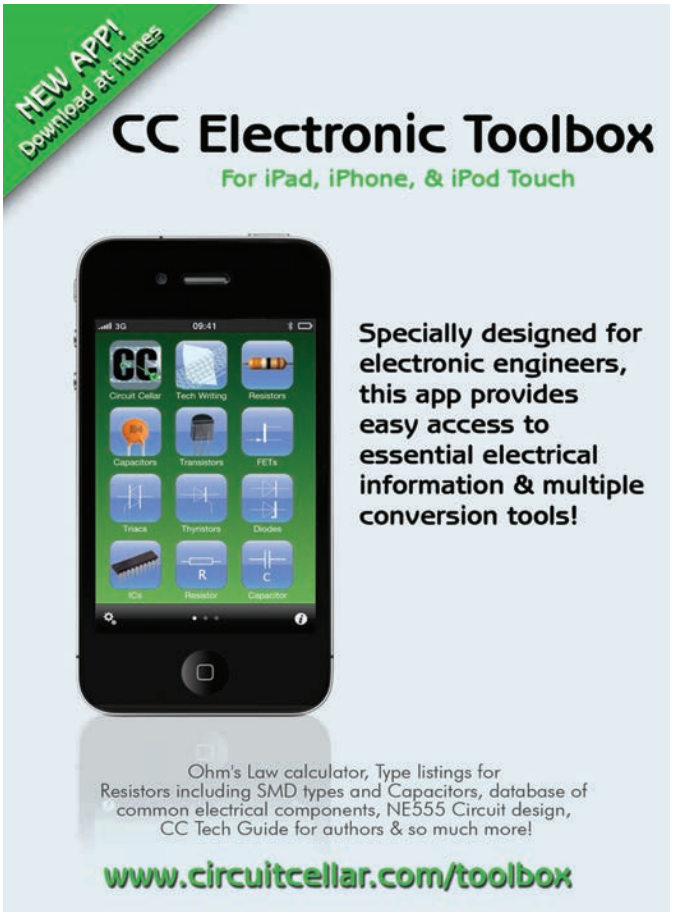
> "You might not need an antialiasing filter at all if you are 100% sure of your input signal, but usually you will need to implement either a low-pass filter—selecting the first Nyquist zone, from DC to 62.5 MHz (i.e., 125/2)—or a band-pass filter if you want to use the ADC in under-sampling mode."

In your low-frequency designs you may be used to rail-to-rail amplifiers. These chips are more and more common and are easy to use, but they can't provide the same performance as actual dual-voltage amplifiers simply because designing a rail-to-rail amplifier implies some concessions on their linearity. So, in a nutshell, if you target high performances then be prepared to spend some design hours—and some dollars—on the input amplifier schematic and associated power supplies.

CLOCK GENERATION

Let's move on to the clock. The goal of an ADC is to sample the input signal at precise time steps, so it is not unreasonable to discover that the clock driving the ADC is also one of the critical design sections, is it? Imagine that the input signal is a perfectly linear voltage ramp, but that your reference clock has some jitter, which means that the ADC sampling times are moving a little around their theoretical position (see Figure 8a). What will happen? This time jitter will give the same result as noise on the ADC input itself. The output values will be randomly above or below what you were expecting.

Similarly, if the input signal is a pure sine wave, then a jitter on the sampling clock provides extra noise on the ADC output samples, which means a reduced signal-to-noise ratio.





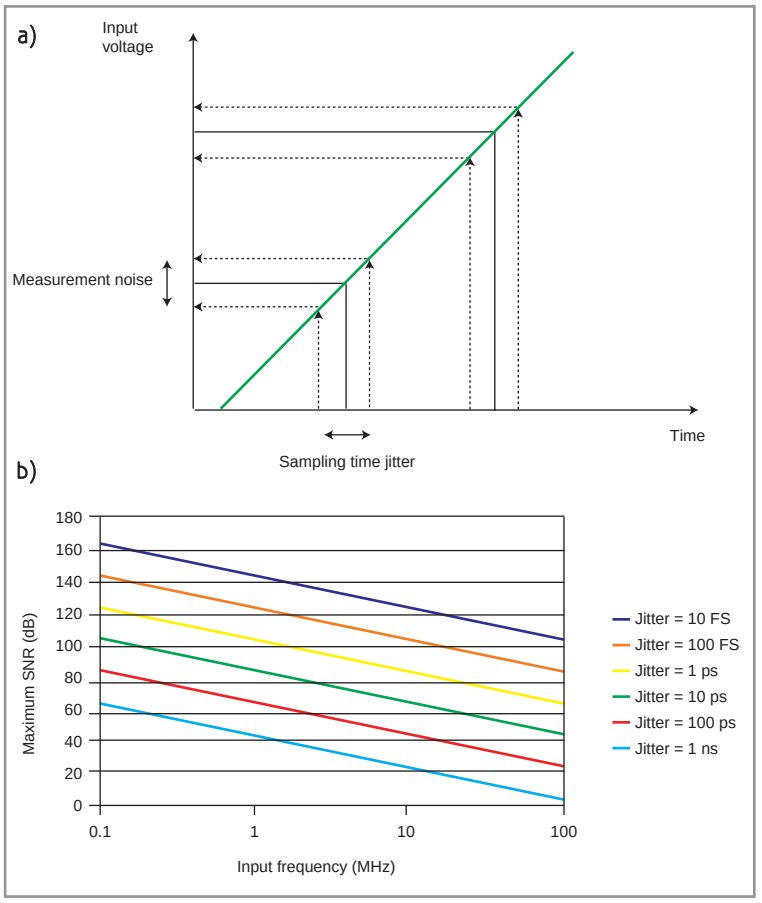


Figure 8a—A time jitter of the reference sampling clock of an ADC is equivalent to extra noise on the measured voltages. **b**—Such a clock jitter can drastically limit the signal to noise performance of the converter, as shown on this graph.

In fact, it can be shown that the maximum SNR you can expect could be calculated as:

$$SNR_{MAX} = 20 \log_{10} \left(\frac{1}{2\pi \, ft_{JITTER}} \right)$$

In this formula, f is the frequency in Hz of the input signal (not the sampling frequency), and t_{JITTER} is the RMS jitter of the reference clock, in seconds. Let's do a numerical calculation. The AD9265-125 I used as an example has a SNR of 79 dB. What should the maximum RMS jitter of the reference clock be if you don't want to compromise this SNR, assuming that the maximum input frequency is 20 MHz? Use your calculator, and you will find that the maximum RMS jitter you can tolerate is about 0.9 ps. Yes, below 1 ps. Figure 8b shows you this relationship graphically.

Is a time jitter of 0.9 ps really a problem? You bet! The RMS jitter of a single 74-HCT00 gate is already around 2.2 ps, so standard logic gates should be avoided in the clock section of high-speed ADCs. Opt for low-jitter sine-wave clock sources such as TCXOs and OCXOs, and if you need to manipulate the clock signal, then use only transformers or non-saturated discrete transistors to limit the extra jitter. And remember that low currents usually mean high noise, so choose high-bias currents whenever jitter is a concern.

VOLTAGE REFERENCE

What else? An ADC compares an unknown input voltage to a reference voltage, so of course the accuracy of the measurement is also directly linked to the accuracy of the reference. Even if it is usually not a concern for high-speed ADCs, let's spend a minute on this topic. Imagine that you want to use a not-so-high-end 16-bit ADC driven by a 2-V voltage reference. If you want full 16-bit accuracy, then your reference must have a precision better than $2.0/2^{16}$ V which gives ±15 µV. Unfortunately, you will probably not find such a reference chip anywhere! Maybe the best reference you can find will be the MAX6126 ultra-low-noise, high-precision, low-dropout voltage reference from Maxim Integrated Products. It provides an astonishing ±0.02% + 3 ppm/°C accuracy. Assuming that you live in a perfectly temperature-controlled environment, this translates into ±400 µV for a 2-V reference. Very good, but still far from the ±15 µV required by our 16-bit ADC.

To make this story short: If you restrict yourself to off-the-shelf components, you can't expect an accuracy higher than 12 or 13 bits, even if you buy a 16- or 24-bit ADC. With a custom calibration, this could be improved maybe up to 16-bit, but unfortunately not more. This proves a high-end instrument like Agilent's 3458-A 8-1/2 digit multimeter and its 8-ppm guaranteed one-year accuracy is impressive.

WORKING WITH ADCs

At this point of my column, you may have already jumped off, or you may think fast ADCs are not actually cool to work with. I'm afraid that is true. And I have not discussed other pleasant subjects such as thermal problems (as these chips are dissipating watts), power supply headaches (as ultra-clean clean power supplies are required for best performances), prototyping nightmares (as manufacturers like to package them in ultra-small 0.5-mm pitch packages), and so on. Moreover, the digital side of the ADC is also a pleasure, as you will have to design something capable of working with hundreds of megabits per second of useful data probably sent on low-voltage differential signalling (LVDS) lines. You will also find plenty of subtleties. For example, it is not unusual to find an ADC that provides incorrect values for the first conversions after power-up or register change due to its internal dataflow management. I've met one, and of course it was not so obvious when reading the datasheet.

Anyway, as an engineer, you have always two solutions in such a difficult case. You can try to stay as far as possible from the problems, using ready-made boards as often as you can or changing the project's requirements to accept lower-end ADCs. Or you can say: "Great! Another super-interesting project. I'm sure I'll learn plenty of new things when designing it! Let's switch on the soldering iron and the oscilloscope!"

Don't get me wrong. The first solution is by far the better one if you are working on a risky project with a constrained schedule and budget, but I hope you will at least consider the second solution if you have some time! In both cases, I hope this column helped to bring fast ADCs out of the dark.

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.

REFERENCE

[1] W. Kester, *The Data Conversion Handbook*, Analog Devices, Inc., 2004, www.analog.com/library/analogDialogue/archives/39-06/data_conversion_handbook.html.

RESOURCES

Analog Devices, Inc., "AD9642 Datasheet: 14-Bit, 170 MSPS/210 MSPS/250 MSPS, 1.8 V Analog-to-Digital Converter (ADC)," 2011, www.analog.com/static/imported-files/data_sheets/AD9642.pdf.

D. Kress, "Analog Devices Technical Article MS-2124: Understanding AC Behaviors of High Speed ADCs," 2011, www.analog.com/static/imported-files/tech_articles/MS-2124.pdf.

Maxim Integrated Products, Inc., "MAX6126 Datasheet: Ultra-High-Precision, Ultra-Low-Noise, Series Voltage Reference," 2010, http://datasheets.maxim-ic.com/en/ds/MAX6126.pdf.

D. Redmayne, E. Trelewicz, and A. Steer, "Linear Technology Design Note 1013: Understanding the Effect of Clock Jitter on High Speed ADCs," Linear Technology, 2006, http://cds.linear.com/docs/Design%20Note/dn1013f.pdf.

SOURCES

3458A Digital multimeter
Agilent Technologies, Inc. | www.agilent.com

AD9265-125 and AD9642 Analog-to-digital converters
Analog Devices, Inc. | www.analog.com

MAX6126 Voltage reference
Maxim Integrated Products, Inc. | www.maxim-ic.com

NEED-TO-KNOW INFO

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

—

ADCs and Signal Filtering
by George Martin
***Circuit Cellar* 258, 2012**
You understand the process of creating a new product from start to finish. Now it's time to dig into the specifics, starting with analog-to-digital converters and their supporting hardware and software. Many of the techniques in this article can be applied to other design dilemmas. Topics: ADC, Analog, Digital, Converters, Interface

—

Build a Simple USB DAC
by Yoshiyasu Takefuji
***Circuit Cellar* 213, 2008**
Yoshiyasu describes the step-by-step construction of a simple USB digital-analog-converter (DAC) around an ATtiny45 and a MAX517. You can use the system as a programmable power supply. Topics: USB, Digital, Analog, DAC, ATtiny45, MAX517, Cygwin, WinAVR, AVR-USB, GNU C, Protocol Stack, NRZI, Device Driver

—

PSoC 101
Applied PCs
by Fred Eady
***Circuit Cellar* 169, 2004**
Analog design doesn't have to be complicated, especially if you use a PSoC. In this column, Fred describes his PSoC development system, which is geared toward the CY8C27443. Topics: Analog, Digital, PSoC, CY8C27443, Function Blocks, Integrated, Development, Environment, IDE, Training

—

Single-Pin Analog-to-Digital Conversion Techniques
by Ingo Cyliax
***Circuit Cellar* 162, 2004**
Picture this: you need to add an analog input to a design, but you have only one digital I/O pin remaining. Ingo presents an intelligent, cost-effective solution. Topics: Analog, Digital, Converter ADC, RC, RL, Temperature, Thermistor, NTC, Proximity

—

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

by Bob Japenga (USA)



Getting Started with Embedded Linux (Part 3)

Linux Licensing Issues

The purpose of this article series is to get you started with embedded Linux. This month you learn about the various licensing issues an embedded systems designer faces when using Linux.

The other day I switched my wife's cell phone provider of 12 years to a downgraded pre-paid plan. She doesn't use her phone often and was using an eight-year-old hand-me-down phone from one of our grandsons. I estimated that the change was going to save me about 50% compared to what we were currently paying. When I notified our carrier that we were switching, the customer service representative warned me that it was going to cost me more in the end. As I hung up the phone, I wondered where they hire people to say those things.

I also remembered getting a similar warning from one of the premier commercial off-the-shelf (COTS) embedded operating system (OS) suppliers a few years back. We developed a software system in conjunction with our customer's proprietary hardware that used this supplier's highly rated OS. The tools were superb. The support was mediocre (and expensive). Sales languished on the product. Eventually we abandoned the OS and ported the system to Linux. In six months, our customer was selling the new product on three different hardware platforms and sales were taking off. As I spoke to the representative of this proprietary OS about porting to Linux, he told me to make sure I told our customer that he would have to release all of their proprietary software to the open-source community. Again I wondered how much they pay these folks to say these things.

This month, I want to look at some specific licensing questions concerning embedded Linux. It is my desire to dispel some of the fears that are fostered throughout the development community. This article is not intended to be legal advice, but I hope to frame the question, point you to the resources available, tell you how we handle the license issues, and describe some pitfalls to avoid.

SOME BACKGROUND

If you are going to build a new embedded product using Linux, there are essentially four licenses you need to be concerned about. The Linux kernel is released under the GNU general-purpose license (GPL) Version 2. The kernel enables you to create proprietary application software in user space. You are not required to disclose the source code of that proprietary application code as long as it is not a "derived work" from the kernel. The kernel license is very clear about this, saying,

HELPFUL DEFINITIONS

COTS: Commercial off-the-shelf
Derived work: A legal term defined under the copyright laws of the U.S. as a work based upon one or more pre-existing works
GPL: General-purpose license
LGPL: Lesser general-purpose license

"This copyright does *not* cover user programs that use kernel services by normal system calls—this is merely considered normal use of the kernel, and does *not* fall under the heading of 'derived work.'"

Accordingly, you can issue your proprietary software under any type of restrictive licensing you deem appropriate for your business model.

The latest version of the GNU GPL is Version 3. Some of the common Linux applications are released under this license. These powerful applications can be included in your system without requiring disclosure of any of your proprietary application code unless your proprietary code is in one of these modified Linux applications.

The libraries used by the free toolchain (GNU) are released under the GNU lesser general-purpose license (LGPL). For example, this license enables you to link your application to the C run-time library and distribute your proprietary application in binary form.

Some Linux applications are released under the Apache license. You are free to modify any software released under this license and incorporate it into your embedded system without any subsequent restrictions other than notification issues.

All of these software license variants have been around for many years. Each has its own nuances which you should be familiar with. This is no different than your need to become familiar with the licenses of any commercial software you buy. The problem is that most of us scan to the bottom and click "I agree" and have no idea what we are agreeing to. For the sake of our fiduciary responsibilities (exposing our companies to lawsuits), and for our own integrity, we need to understand our rights and our restrictions when we incorporate open-source software into our products.

Using and incorporating these into your embedded Linux product is consistent with the spirit of the open-source community.

ISSUES WITH GPL-LICENSED SOFTWARE

A number of issues arise when you modify and distribute the source code of open-source software. If you are using the software for your company's own private use, you can modify it and are under no obligation to disclose your modifications. Should you modify it and distribute it, you are obligated to distribute your modified source code under the same license agreement as the GPL.

Example 1: You don't like the kernel's out of memory (OOM) logic (which is a major problem for robust embedded systems). You invest an immense amount of time and effort and create the coolest and most bullet-proof OOM logic for embedded systems ever devised. You put this into your product. You are obligated to publish this whiz-bang solution by distributing the source code to your customers under the terms of the GPL (which means they are free to modify and distribute it to whomever they like).

Example 2: Your hardware platform has some proprietary devices attached to it. The devices are a generation ahead of your competition. The device drivers you previously used on your COTS OS, if made public, would give your competition a four-year jump start enabling them to catch up to you in short order. If you build this device driver into the Linux kernel and deliver it as a product, you are obligated to release this driver code to your customer. And your customer could give it to your competitor.

> "For an individual embedded software developer, there are always needs for new device drivers for Linux. Individuals can choose certain pieces of hardware and create great drivers for these devices which then, once approved, can be included in the Linux kernel distribution. Your name would be forever linked to the driver and you could become the world's expert on these devices for Linux."

Example 3: You are developing a system that uses a high-speed serial analog-to-digital converter (ADC) for which there is no Linux driver. You develop the new driver and incorporate it into the kernel. As with Example 2, you are obligated to release this source code to your customers.

Example 4: You are developing some application code that interfaces to a device that uses a CRC-64 algorithm. Rather than developing this from scratch, you see that it is embedded in the kernel with no external interface. If you lift this source code and place it into your application, you are obligated to disclose the source code of your entire application to your customers.

Example 5: You are using an LGPL library that uses a lot of dynamic memory allocations. You modify the library to incorporate Dmalloc into it to help you detect memory leaks. Under the LGPL, you are obligated to release the object modules to your customers (not just your binaries) so your customer could relink your modules with a different library!

HOW OUR COMPANY HANDLES THESE ISSUES

First and foremost, we have one person who is knowledgeable about the issues and enjoys keeping up with these licensing issues. He is the go-to guy when license issues come up.

Let's look at how we handle each of the examples.

OOM example: Although we have not created a whiz-bang and robust OOM, should we do that, we would distribute that source code with our binaries under the GPL as we have other kernel modifications we have made.

Proprietary hardware example: Kernel drivers can be either embedded into the kernel or loaded from application space. It is our understanding of the GPL that a loadable driver that uses normal kernel calls that is written from scratch is not a derived work and thus can retain its

proprietary nature and does not need to be disclosed. If however, you modified the kernel and used some proprietary calls in this loadable module, it would fall under the GPL.

New ADC example: As I mentioned in my December 2011 column, we try to use peripherals that already have Linux drivers. That is our first priority and is most cost effective. In other cases, we have chosen to release the source code as part of the distribution.

CRC-64 example: We needed to incorporate a number of general-purpose check algorithms into one of our products. Although we found these algorithms in GPL code, we basically looked and found the same algorithms implemented under Apache-style licenses. In another case, when we couldn't find something comparable, we bit the bullet and implemented the algorithm from scratch.

LGPL example: We modified a library licensed under the LGPL for test purposes. However, we chose not to release the modified library and kept the modified library only for in-house use because of the complications of releasing the object code of our proprietary application.

Finally, how does our company distribute the modified GPL software to our customers (who probably couldn't care less)? We provide a link to the source code in the embedded webpage that is part of the configuration tools we provide.

WORKABLE BUSINESS MODELS WITH GPLs

At the end of the day, if you are in business you need to make money. There are a number of ways individuals and companies can create a viable business model using Linux with its GPL. Here are just two.

For an individual embedded software developer, there are always needs for new device drivers for Linux. Individuals can choose certain pieces of hardware and create great drivers for these devices which then, once approved, can be included in the Linux kernel distribution. Your name would be forever linked to the driver (or perhaps that genre of drivers) and you could become the world's expert on these devices for Linux. Consider it advertising. There are a number of consultants who do just that.

For companies developing hardware systems, the Linux GPL model will somewhat alter how you develop your software. But with simple planning, you can keep your intellectual property private while at the same time utilizing all of the power of Linux. That is a good business deal.

PITFALLS TO AVOID

As system developers, we face many pressures. One of those pressures is to "pretend not to know" when it comes to software licenses. Avoid this if at all possible. Take the time to read the licenses. It is not rocket science. You will have the peace of mind that you did it right.

Another pressure is to incorporate just a little bit of copyrighted code into your application thinking, "No one will know. We are a small company. I am sure I could find it somewhere else if I just took the time." *You* will know and if that is not enough incentive, then you don't understand the power of the conscience.

FINDING BALANCE

In the spirit of "Embedded in Thin Slices," I hope I have shared a little bit of what we have learned about licensing issues when using Linux in embedded systems. There are many possible scenarios that go beyond the scope of this column. I hope you can see that you can protect your intellectual property and still be faithful to the licensing requirements of Linux and its associated applications.

Hopefully I have dispelled the notion espoused by one who should know better: "The way the [general-purpose] license is written, if you use any open-source software, you have to make the rest of your software open source. … Linux is a cancer that attaches itself in an intellectual property sense to everything it touches. That's the way that the license works."

Where do they hire people to say such things?

The next column will discuss toolchains that can be used when developing an embedded Linux system.

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.

RESOURCES

The Apache Software Foundation, "Apache License, Version 2.0," 2004, http://www.apache.org/licenses/LICENSE-2.0.html.

Dmalloc, Debug Malloc Library, http://dmalloc.com.

GNU Operating System, "GNU General Public License, Version 2," 1991, www.gnu.org/licenses/gpl-2.0.html.

———, "GNU General Public License, Version 3," 2007, www.gnu.org/licenses/gpl-3.0.html.

———, "GNU Lesser General Public License, Version 3," 2007, www.gnu.org/licenses/lgpl.html.

———, "Frequently Asked Questions about the GNU Licenses," 2011, www.gnu.org/licenses/gpl-faq.html.

B. Japenga, "Getting Started with Embedded Linux (Part 2): Choosing a Platform for Your Embedded Linux System," *Circuit Cellar* 257, 2011.

K. Yaghmour, *Building Embedded Linux Systems*, O'Reilly Media, 2008. Appendix C, Linus Torvald's Interpretation of the GPL for Linux.

CIRCUIT CELLAR

Microprocessor Design Using Verilog HDL

With the right tools, such as this **new book, designing a microprocessor can be easy.** Okay, maybe not easy, but certainly less complicated. Monte Dalrymple has taken his years of experience designing embedded architecture and microprocessors and compiled his knowledge into one **comprehensive guide to processor design in the real world.**

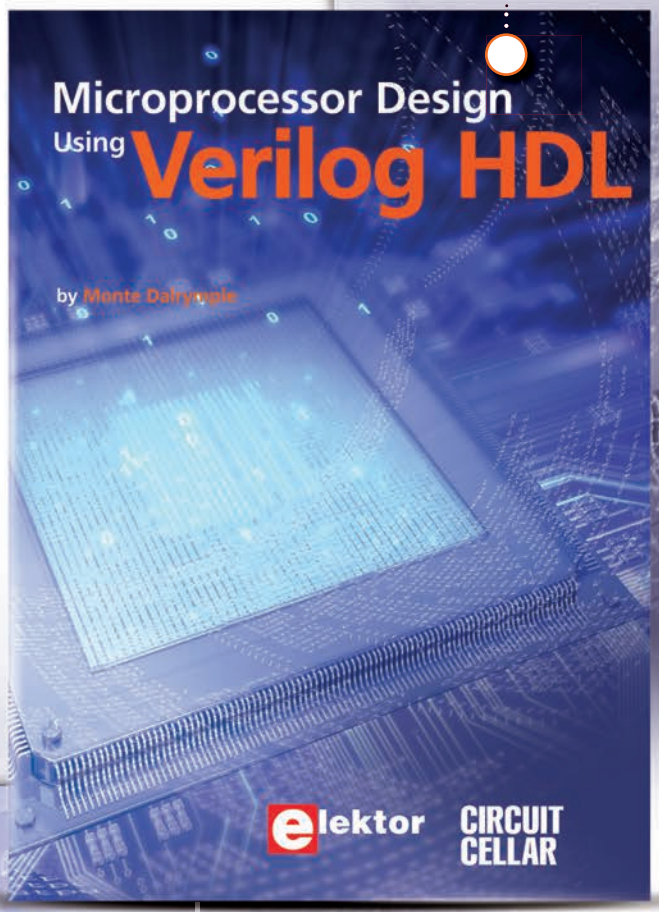
Buy it before March 15, 2012 and **SAVE 15%** off the cover price.

Yours for just **$38.25!**

Monte demonstrates how Verilog hardware description language (HDL) enables you to **depict, simulate,** and **synthesize an electronic design** so you can **reduce your workload and increase productivity.**

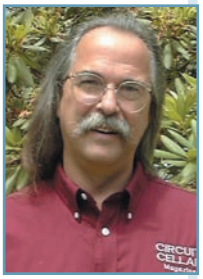
Microprocessor Design Using Verilog HDL will provide you with information about:

- Verilog HDL Review
- Verilog Coding Style
- Design Work
- Microarchitecture
- Writing in Verilog
- Debugging, Verification, and Testing
- Post Simulation and more!



www.cc-webshop.com

by Jeff Bachiochi (USA)



Multi-Serial Protocol Tool

Armed with a simple serial terminal application on your PC, you can build a multi-serial protocol tool designed to connect from a USB host an I²C, SPI, RS-232/422/485, or Dallas 1-Wire format. You just need to know the format of the data for each device with which you'd like to communicate.

It seems like every Thanksgiving Day at least one radio station plays "Alice's Restaurant Massacree." It's a rather lengthy musical monologue by Woody Guthrie's son, Arlo, who is quite the storyteller, just like his Dad. After meandering through visiting with friends, doing good deeds, dumping garbage, getting arrested, and a big court battle, we finally get to hear about the draft. For you younger readers, the draft was the U.S.'s method of choosing those who would serve in the military. Each year those (boys) turning 18 would be entered into a lottery drawing. The winners would spend their next years in service of our country.

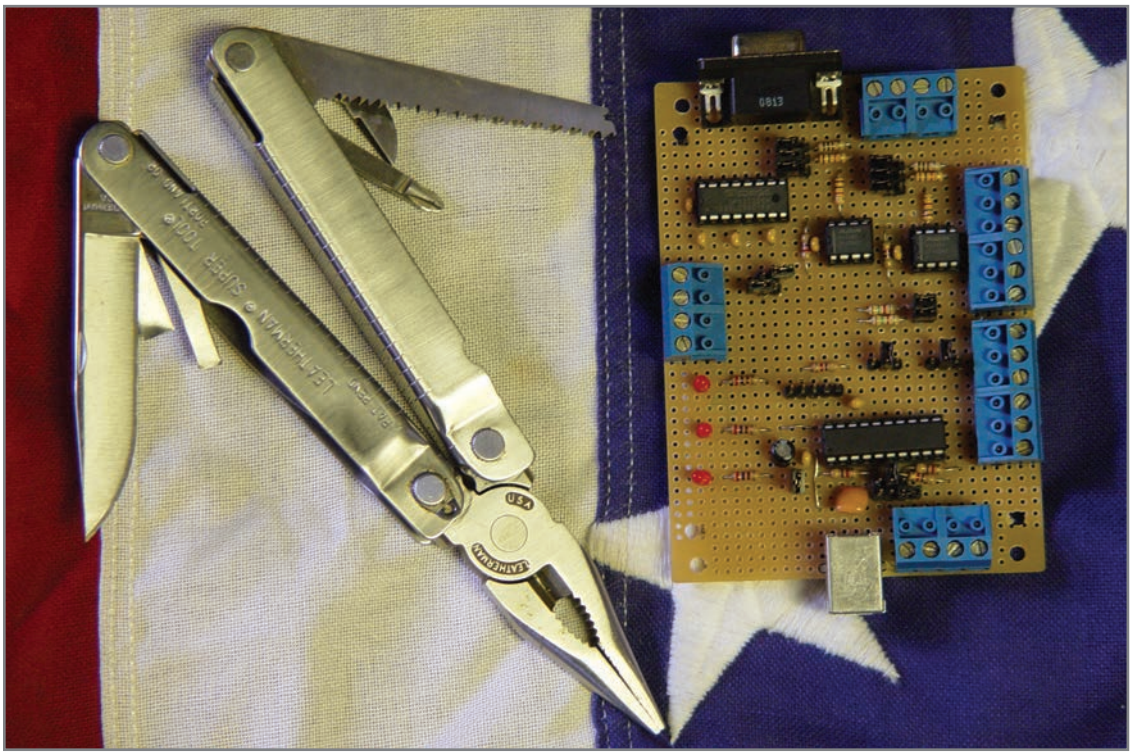


Photo 1—Multitools made in the U.S. I sized the printed circuit board (PCB) of this project to screw to the cover of a plastic enclosure. This keeps the underside insulated when I place it in my crowded work area, which is usually cluttered with parts, wires, and other short-producing material.

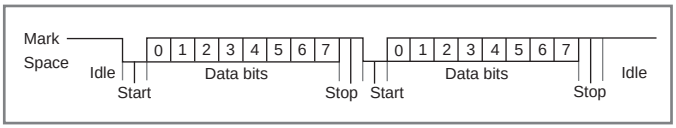


Figure 1—Asynchronous serial transmissions begin with a logic-low start bit (12 V on RS-232). Eight data bits follow with an additional logic-high stop bit (–12 V on RS-232). Since no clock is actually sent, each device must agree on the bit time used (bitrate).

Arlo's 10-minute pre-story about being arrested for littering turns out to be a crucial part of the second 10 minutes of the song, which are about the draft. The whole idea here is that sometimes you need a little context or things just don't make sense. Recently, while working on a precipitation measurement system ("Quantifying Precipitation: A Device to Measure Rainfall Volume," *Circuit Cellar* 258, 2012), I realized how useful the right tools were for the design process. In that case, I wanted to use I²C or SPI as the medium for communicating with the smart sensor I developed to measure water. This is a "chicken or the egg" situation, as I didn't have the application that would ultimately be using the sensor, so I had no way of developing the sensor's communication. In the past, I'd built an RS-232-to-I²C dongle for speaking I²C with my PC. Lately, I've found the lack of RS-232 ports problematic.

After beginning a design for a USB-to-I²C dongle, I realized that a multiprotocol tool would make a great project. So, the project I'll describe in this article gives you a tool to connect from a USB host to any of the following formats: I²C, SPI, RS-232/422/485, and Dallas (now Maxim IC) 1-Wire. A simple serial terminal application on your PC is all that's necessary. You need to know the format of the data for each device to which you wish to talk. However, you are prompted with the general formats for I²C, SPI, and Dallas 1-Wire data, while RS-232/422/485 are strictly passed through.

TOO BIG FOR A DONGLE

Supporting all of these communication buses means lots of real estate. Most of that comes from connectors. I chose screw terminals as you can quickly connect up wires without any special headers or crimping pins (see Photo 1). You might think to limit your build to only that bus of interest, but I highly recommend that you go all the way and not limit the potential for future investigations. Let's take a quick look at each of these buses, starting with the RS-232/422/485.

RS-232/422/485

All these converters use same data format, an asynchronous data byte sent least-significant bit (LSB) first surrounded by a start and stop bit used for maintaining synchronization (see Figure 1). All devices on the bus must agree on the data clocking speed and data format (i.e., parity). RS-232 began as an interface between electromechanical teletypewriters and modems. The single-ended (ground return) ±12-V swing includes a –12-V or mark/idle state (logic 1) and a 12-V or space state (logic 0). The apparent confusion in state levels comes from the transmitter and receiver that invert the TTL levels to establish the bipolar swing. Start bits are therefore 12 V, and stop bits are –12 V. Twin unidirectional buses enable only two devices, as three (or more) devices would cause interference between multiple driving sources. The combination of a single-ended bus and data rate tends to limit the maximum useable bus length.

RS-422 uses a differential bus in an attempt to increase its immunity to interference and provide for much longer distances (up to 1,200 m) between devices. These transmitters/receivers operate on 5 V and eliminate the need for ±12-V supplies. Like RS-232, RS-422 uses twin unidirectional buses for data traffic. However, with RS-422 came multi-drop capabilities. This enables a master device to talk with multiple slave devices because transmitters can disable their drivers to prevent interfering with one another's responses.

RS-485 is similar to RS-422 in that it uses the same differential bus, but the bus is reduced to a single differential pair. This means that data flow is now bidirectional and all transceivers must share the same bus. Only one device can talk on the bus at any time. Everyone must keep their drivers disabled until the bus is free (has been in an idle state for at least 1-byte time.) When used over large distances, such as between buildings, it is common to use an isolation circuit at one end to prevent differences in building grounds from exceeding acceptable limits.

The differential bus should be terminated with 120 Ω at each end of its length. If there is a possibility that all drivers are disabled, you should also ensure that the bus remains stable and in the idle state without a driver by pulling up the 5-V node and pulling down the 0-V node.

In most cases, you will only have to set the data rate for communication. Whatever data rate you choose at the PC application will be used on the microcontroller's UART. The UART is preset for 8 data bits and no parity. The driver for RS-485 is disabled whenever data isn't being sent and will only be enabled if the receiver has been in an idle state for 10 bit times to prevent bus collisions.

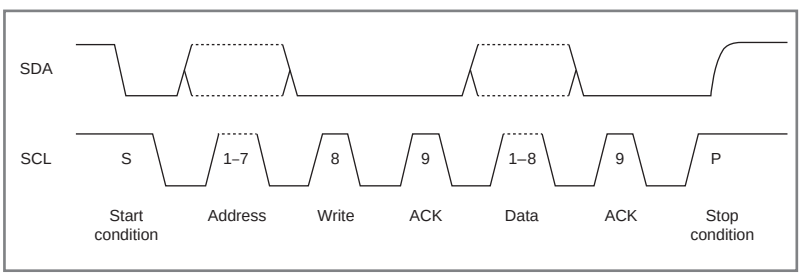


Figure 2—The I²C bus uses two I/O bits, SCL (clock) and SDA (data). This format requires nine clock cycles, eight for information and one for acknowledgement for each byte passed.

I²C

When Philips needed a way of interfacing peripherals to a microcontroller without using its address and data buses it created the inter-integrated circuit bus, or I²C as we

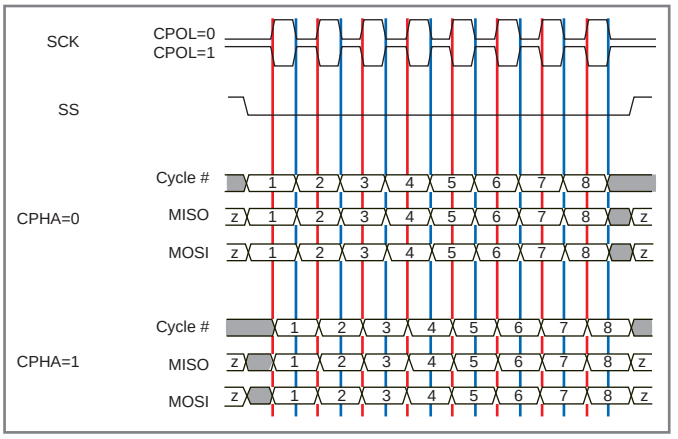


Figure 3—The SPI bus uses four I/O bits, SCK (clock), MISO (data), and MOSI (data), along a with slave select (SS) for each slave peripheral connected. This format requires only eight clock cycles for each two bytes passed, one byte in each direction.

know it today. The I²C bus consists of two bidirectional logic-level lines—SCK (clock) and SDA (data)—that have 1.5-KΩ nominal pull-ups. Drivers for each line are open collector (can only drive the bus low). This synchronous serial bus uses a special message format that includes an address and data byte(s) as shown in Figure 2. The idea for this short-distance intraboard connection system was that each peripheral would be manufactured with its own address to which it would respond. This enables the master (microcontroller) to speak with a specific slave (peripheral) without interference from other devices, all connected to the same two-wire bus. The master's first byte contains an address and read/write request. Peripherals look for their personal address and process the data that follows based on the read/write request. Other peripherals remain silent until they see a request that includes their address.

So, how does a slave know when a first byte is sent since the value of an address can look just like data? There is a special relationship between the SCK and SDA lines that make up the two-wire bus. During address/data transfer, SDA will only change state while SCK is low. Should a peripheral ever see SDA change while SCK is high, it stops all activity and gets ready for an address transmission. Logically, there are two possible cases where SDA changes while SCK is high. If SDA falls, this indicates the start of a message. If SDA rises, this indicates the stop or end of a message. The item that seems to cause the most confusion is the difference between the 7-bit address and the first byte's address value. This value consists of the 7-bit address shifted left one bit (bits 7:1). The LSB (bit 0) is set for a read and cleared for a write command. So, for an address of 0x7F, the actual value sent would be 0xFF for a read command and 0xFE for a write command.

After the eighth bit sent by the master, the slave must acknowledge receipt of the byte by forcing SDA low during the ninth bit time. If the command was a read and the slave is supplying the data on SDA, then it is the master's responsibility to supply an acknowledge to the slave. This also tells the slave to continue sending additional data. When an acknowledgement does not occur, transmission will cease. Should the slave need more time to process a request before it can send (or receive) data, it can signal this to the master by forcing SCK low. The master must pay attention to the SCK line's actual state. It is in control of SCK. But if it finds that after releasing the SCK line (enabling it to go high via the pull-up on the bus) it is not high, it knows the slave is asking for more time and it will wait for the SCK to go high before continuing. Note: This is a dangerous act for a slave to perform as it can permanently disable further communication if it hangs in this state. To prevent this, use a watchdog timer (WDT) to perform a reset on the slave if execution is caught in an endless loop.

In general, there are two kinds of I²C peripheral devices, those that require their register pointer to be set and those that don't. When a device requires its pointer to be set, it will set the data pointer using the first data byte of any write transfer. Otherwise, it will be initialized to zero at the start of each transfer. Data bytes will be written to or read from the indirect address of this auto incrementing pointer.

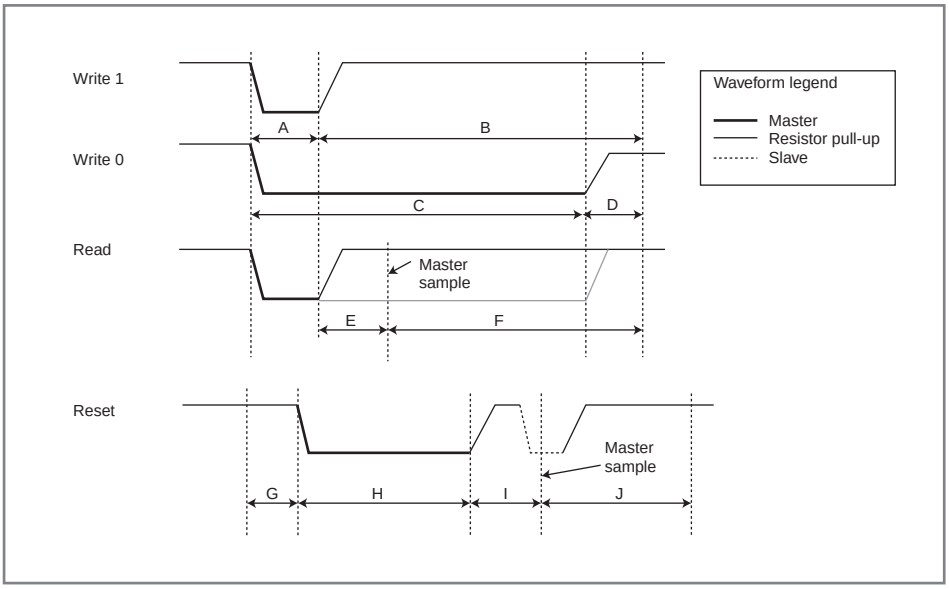


Figure 4—The 1-Wire bus uses a single I/O bit to supply power to the peripherals (while the bus is high) and for bidirectional data transfers. Data is passed using timeslots. A timeslot begins when the idle high bus is pulled low for 1 µs. Any timeslot that remains low for a minimum of "E" and maximum of "C" is considered a "0."

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

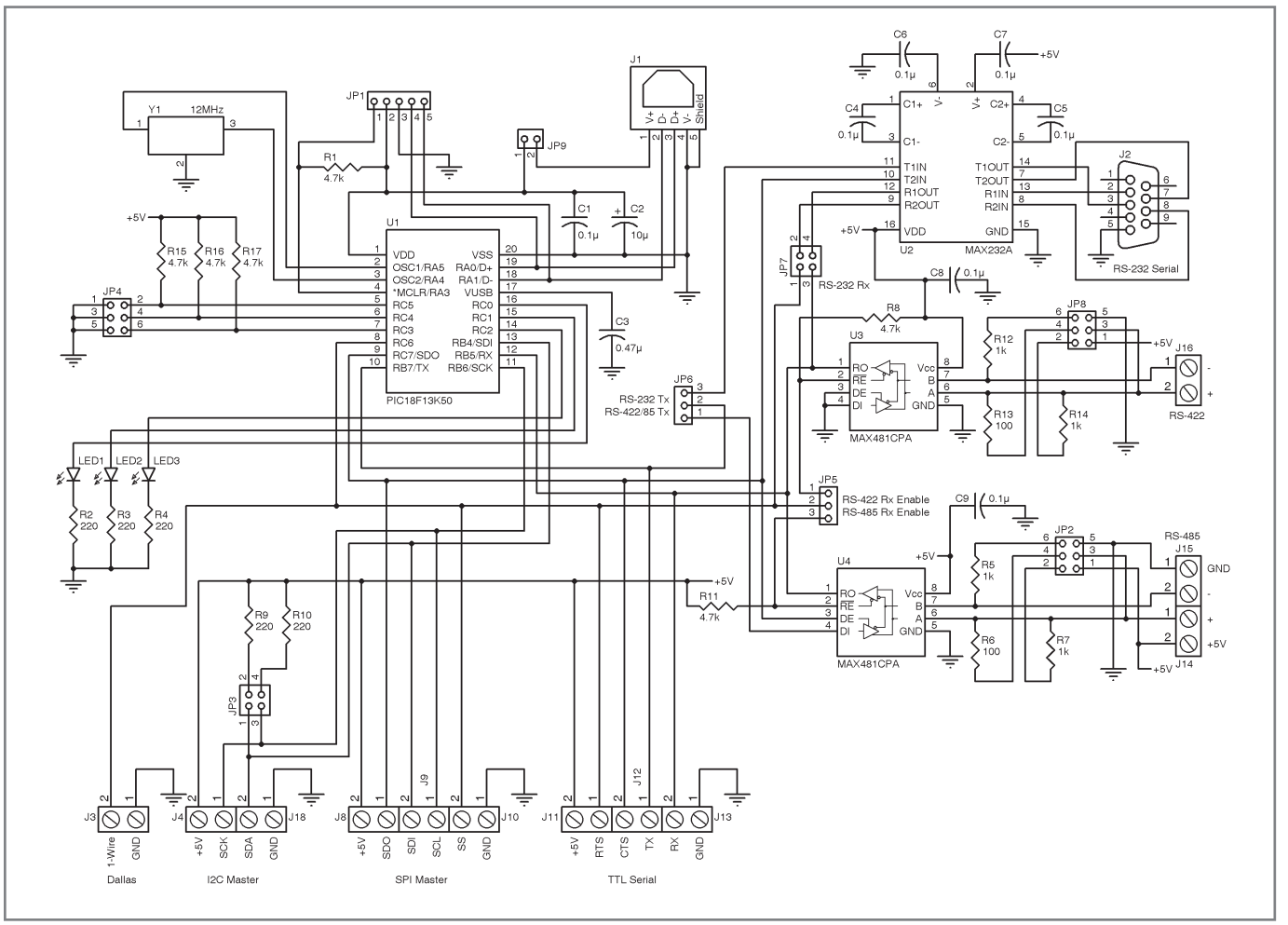


Figure 5—This project's schematic has a number of user-installed jumpers used to configure the operating mode and provide flexible termination and option selections. All power is supplied by the USB connection. Screw terminals are provided for each bus.

SPI

Motorola chose a full-duplex scheme using four logic-level connections (see Figure 3). Dubbed the serial peripheral interface (SPI), it is based on a master that supplies a synchronous clock (SCLK), two data lines—one out (MOSI) and one in (MISO), and a slave select (SS). When multiple devices are attached to a SPI bus, one SS is required for each slave peripheral. While this eliminates the overhead of having to communicate an address, it requires more I/O lines in systems that have a large number of peripherals connected. The data format is simple, the master provides eight data clocks and one data byte is transferred from the master into the slave and at the same time one data byte is transferred into the master from the slave.

You may see these signals labeled as SCK, SDI, SDO, or CS (or other morphed conventions). These alternatives are slightly less informative. It's common to get inputs and outputs confused between a master and its peripherals. The MISO and MOSI designations leave no doubt in the signal's direction. What seems simple on the surface does require some agreement on signaling format. There are actually four possible modes of operation. These have to do with clock polarity and data capture phase. The clock polarity (CPOL) can be high idle (=1) or low idle (=0). The data capture phase (CPHA) can be the first edge of the clock (=0) or the second edge of the clock (=1).

SPI uses no acknowledgement by a slave peripheral, so the master must assume all is well. A SPI slave can't signal for more time, so it must be ready when the master begins. How does a peripheral know what to place in its output buffer before being asked? In some cases, it might have just a single response. If so, the peripheral might refresh its output with the latest data and it can be ready to transfer as soon as it sees the SS line go low. More likely, it will need to receive one data byte (or more) to understand what data it needs to provide. In any case, you can see that for each byte clocked out by the master, the data received from the peripheral may not contain any relevant information. And once the slave figures out what it needs to return, additional clocks are required by the master to get that data from the slave. During these clocks, the data coming from the master might be irrelevant.

What this all means is that you must have intimate knowledge of the slave device and its requirements for

Listing 1—The sign-on of each mode of operation. I will use this to jog my memory on the formatting and commands available the next time I try to use it.

```
USB to SPI Mode

To write data to a SPI register use the format...
 <command><sp><8-bit data>[<sp><8-bit data>]<CR>
You will receive 1 byte for each byte sent.
Dummy bytes may be necessary to retrieve additional bytes.
You must know the data format for the device connected.

M? to query SPI M(ode)
M=0 to use 0
M=1 to use 1
M=2 to use 2
M=3 to use 3
Presently M=0

K? to query I2C/SPI K(bps)
K=0 to use 100 K(bps)
K=1 to use 400 K(bps)
K=2 to use 1000 K(bps)
Presently K=0

H? to query H(ex) Data Format
H=0 to use (3 Decimal Digits 000-255)
H=1 to use (2 Hexadecimal Digits 00-FF)
Presently H=0
?
------------------------------------------------------------
USB to I2C Mode

to write data to an I2C address use the format...
W <address><sp><data>[<sp><data>]
remember... if the I2C device uses registers,
 the first data byte updates the register pointer.
to read data from an I2C address use the format...
R <address><sp><dummy>[<sp><dummy>]
 use dummy data for each byte you want to read.
remember... if the I2C device uses registers,
 update the register pointer with a write before reading.
remember... to use the proper data format 2/3-digits

S? to query I2C Bus status
Presently I2C bus is Idle

I? to query I(2C) address length
I=0 to use 1 byte (7-bits)
I=1 to use 2 bytes (10-bits)
Presently I=0

K? to query I2C/SPI K(bps)
K=0 to use 100 K(bps)
K=1 to use 400 K(bps)
K=2 to use 1000 K(bps)
Presently K=0

H? to query H(ex) Data Format
H=0 to use (3 Decimal Digits 000-255)
H=1 to use (2 Hexadecimal Digits 00-FF)
Presently H=0
?
------------------------------------------------------------
USB to RS232/TTL/422/485 Mode

All characters go between the USB and the UART.
The UART data rate is set by the USB application.
3 escape characters '~~~' received from USB triggers
```

Continued on p. 71

communication. There is no "one size fits all" for SPI communication. The data transfer is simple, but the format of the message is specialized for each peripheral.

1-WIRE

If SPI with its four-wire bus is simple, and I²C with its two-wire bus is minimalistic, then what would Dallas's 1-Wire bus be known as? You might call this bus "vampirous." It must be noted that SPI and I²C both require power and ground connections in addition to their noted number of signal (bus) lines. The 1-Wire bus includes only an additional ground connection, as device power is drawn from the one bidirectional bus line. Like I²C, this bus is pulled up to V_{CC} and driven low with open-collector drivers. In its idle pulled-up state, the bus supplies attached devices with parasitic power.

The communication's format is based on time slots (see Figure 4). To begin communication, the master must get attached devices listening by producing a reset pulse consisting of a logic-low level on the bus for a minimum of 480 µs. All devices will respond to the release of the reset pulse within 60 µs with a logic low for at least 60 µs. The master monitors for this "presence" signal to determine if there is anybody out there to talk to. Any device with an "alarm" may hold a logic-low state for at least 960 µs. No "good" device should hold the bus low for longer than 3,640 µs. This would indicate a short on the 1-Wire bus and would require user intervention. The absence of "presence" or "alarm" would indicate no devices are connected. Most 1-Wire commands are 1 byte in length and data is in 8-byte groups. While we haven't discussed 1-Wire slave devices, at this point, you can imagine that if a bus contains multiple devices they would (and they will) interfere with one another.

The master can send the command byte 0x33 to retrieve a device's 8-byte ROM identity code. It does this by sending the command byte LSB first. Each bit time is a minimum of 480 µs. To send a "1," the bit time begins with a 1-µs logic-low pulse. To send a "0," the pulse must be a minimum of 60 µs. Slave devices sample the logic-low times to determine the bit values. If the slave recognizes the command, it will respond with (in this case) its 8-byte ROM value. The master must

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

Listing 1—Continued from page 71

```
     an escape to 'command' mode, where, after
     processing one command, you will exit command mode.

N? to query UART output polarity
N=0 to use inverted
N=1 to use normal
Presently N=0
?
-------------------------------------------------------------
USB to Dallas 1-Wire Mode

All 1-Wire communications begins with a Command followed by...
 a <CR> (carriage return) or one or more groups of 8-bytes,
 each byte is separated by a <sp> (space), ending with a <CR>
A typical group of 8 bytes would be a 'ROM code' made up of...
 1-byte family code, 6-byte serial number, and 1-byte <CRC>.
If more than 1 device is connected on the bus,
 you will need to know each device's 8-byte ROM code, or
 use 'SearchROM' to identify all the devices.
ReadROM='<Command><CR>', where Command='33'.
SkipROM='<Command><CR>', where Command='CC'.
MatchROM='<Command><8-byte ROM code><CR>', where...
 Command='55' and all bytes are separated by spaces.
SearchROM='<Command><CR>', where Command='F0'.
ReadMemory='<Command><16-byte address><CR>', where...
 Command='F0' and all bytes are separated by spaces.

H? to query H(ex) Data Format
H=0 to use (3 Decimal Digits 000-255)
H=1 to use (2 Hexadecimal Digits 00-FF)
Presently H=0
?
```

supply 64 "1" bits (8 bytes of 0xFF); the slave uses each 1-µs logic-low pulse to synchronize its response. It must hold down the bus for at least 45 µs if it wants to send a "0" and do nothing if it wants to send a "1." The master samples the bus after the start of each bit time looking for the state of the bus "0" or "1." After 64 bit times all 8 bytes of the ROM identity code have been transferred.

The nature of an open-collector bus is that it only takes one connected device to pull the bus low. So a "0" will always be received even if all the other connected devices are trying to send a "1." With multiple devices on the bus, if the master was to send the 0x33 command, the 8 bytes returned by all devices at once will have conflicting data (with one or more bit conflicts being received as "0." In this case, the CRC byte received will not match the other 7 bytes. So how can we talk to a specific device if they are all going to respond?

The simple answer is to use the Match ROM command. This command



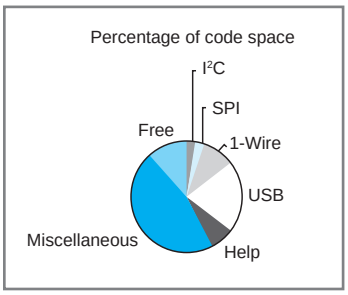


Figure 6—This is a breakdown of the relative code size required to support each of the application's functions.

(0x55) requires an additional 8 bytes of the device you are interested in. All devices compare these 8 bytes to their own ROM identity code. All devices that don't match this code will go silent and remain offline until a reset pulse is again issued by the master. But, if you don't know the code to start with, then what? The `SearchROM` command is actually a software routine that uses a tree search, systematically trying `MatchROM` commands for every possible family and serial number. A quick search time is accomplished by limiting a branch search to only those that have conflicts. (Remember that bad CRCs can indicate multiple device interactions.) You can learn more about this in Maxim Integrated Products's "Application Note 187, 1-Wire Search Algorithm."

The ROM identity code may be all you need (e.g., if your door lock application only needs to recognize when an i-button has been touched to the bus). However, many 1-Wire devices have other functions as well. To explore these, there is a second set of commands used once you have made contact with one specific device.

"What started out as a way to get my serial-to-I²C testing dongle connected to my serial portless laptop turned into a USB-to-multiprotocol project. I took the USB-to-serial circuit/code I used to help explain the inner workings of USB in my article series 'Application Communication with USB' (*Circuit Cellar* 239, 240, and 241, 2010) and adapted it to include RS-422/485, I²C, SPI, and 1-Wire. This tool is all you need to connect your PC's USB port to a test circuit that requires one of these protocols."

MULTITOOL

What started out as a way to get my serial-to-I²C testing dongle connected to my serial portless laptop turned into a USB-to-multiprotocol project. I took the USB-to-serial circuit/code I used to help explain the inner workings of USB in my article series "Application Communication with USB" (*Circuit Cellar* 239, 240, and 241, 2010) and adapted it to include RS-422/485, I²C, SPI, and 1-Wire. This tool is all you need to connect your PC's USB port to a test circuit that requires one of these protocols. The circuit in Figure 5 uses screw terminals to enable separate bus connections for each format without special crimping tools. You could cut the number of connectors in half if you put all the available signals in one row. And, as long as your labeling is clear, you would just connect to the signals associated with the protocol of interest.

For the simple pass through of serial data, you have a choice of connections for TTL and RS-232 serial (Maxim Integrated Products MAX232 5-V powered, multichannel RS-232 driver/receiver), RS-485 (bidirectional differential pair using a Maxim MAX481 low-power transceiver for RS-485 and RS-422 communication), and RS-422 (twin unidirectional differential pairs using two MAX481s). The RS-422/485 differential buses have termination and pull-up/down resistors that are jumper-enabled. Since the RS-485 bus carries bidirectional data, its transmitter is only enabled when the bus is idle. All UART functions are set up through the USB interface. That is, when you open a USB connection at some data rate, the UART in the circuit's microcontroller is initialized to the same data rate.

If you are familiar with modem operation, you know that since all data is just passed through, there is normally no way to enter any commands to the circuitry. So, a particular series of characters (e.g., ~~~) was chosen that would halt the pass-through process and enable any characters that followed to be interpreted as a command, until it found, say, a "carriage return" <CR>. While the circuit's UART is set up via the terminal application program on the PC, there really are not any commands necessary. However, I instituted this protocol anyway to enable the user to invert the data out of the UART. This is sometimes used to invert the TTL data so it can be connected directly as a subset of RS-232C. Refer to the sign-on/help captures in Listing 1 and Figure 5 to see which commands are available in each of the four modes.

Unlike the pass-through mode of RS-232/422/485, all the other protocols are all totally command based. The microcontroller has an I²C peripheral that eases the implementation. No additional ICs are required as this is a logic-level bus. Note that 1.5-kΩ pull-up resistors can be jumper-enabled for the clock (SCK) and data (SDA) bus lines.

This microcontroller also has a SPI that eases this implementation. No additional ICs are required as this is a logic-level bus. All four lines of the bus are bipolar so pull-up resistors aren't required.

The 1-Wire mode is not supported by any of the microcontroller's peripherals. Although it is easily implemented using a Dallas Semiconductor serial-to-1-Wire interface DS2480B chip for less than $6, where's the fun in that? It only takes one I/O bit and a bunch of supporting code to do it all in software. I'd used both methods in

past projects and forgot the amount of code necessary. Some of the parts can be rather complicated and many commands are special to a particular part. I implemented only the basic functions needed to get access to a part when multiple parts were on the same bus. Of these, I included the ReadROM, SkipROM, SearchROM, MatchROM, and ReadMemory.

Families of 1-Wire devices are grouped and identified by the first byte in the ROM code. A unique 6-byte serial number follows the family code with a 1-byte cyclic redundancy check (CRC) of the first 7 bytes appended as a quality check on the total 8-byte packet. The 8-bit CRC uses the $X^8 + X^5 + X^4 + 1$ polynomial. Refer to my 2011 article "Error Checking" to learn more about CRCs (*Circuit Cellar* 250).

TROUBLE IN PARADISE

One of the things that constantly gets me in trouble is documentation. It seems as though users have an aversion to reading manuals. While I tend to pile all owners' manuals in one place, I can never seem to find the one I need. Or, in the case of a project I built, there isn't one (except for maybe the column itself). When I pull out an old project to assist with a new one, it takes me a while to remember (figure out) how to use it again. This time, I put in plenty of text help to aid me as I hedge against potential memory issues.

Presently, two of the three configuration jumpers are being used. This gives four different modes of operation, I²C, SPI, RS-232, and 1-Wire. When the circuit is powered up, it waits to receive a character (any key press) over the USB connection. This way, the sign-on message isn't lost in transmission before a connection is made. The message identifies the mode of operation and displays a bit about the configured mode and the commands pertinent to it. There are five commands that give the user some flexibility, such as the two-digit hexadecimal or three-digit decimal data display format. These choices are stored in EEPROM and are nonvolatile. When you make a choice, the new value is compared to what is presently stored and the EEPROM is updated, if necessary. So, if you choose to work in the two-digit hexadecimal format, your preference is permanently noted and preset the next time the board is used.

The "?"—which is used as a prompt for user input—seems to be a reasonable choice for a help command. Entering a "?" will redisplay the sign-on's help screen. In this application, I used roughly 88% of a Microchip Technology PIC18F14K50 microcontroller's code space. Figure 6 shows a general breakdown of the code space used by the various support functions. In this family (PIC18F*xx*K50), the microcontroller with 20 pins has the most (8,000 words/instructions) available. If I want (or need) more than another 12% for additional functions, I'll go to a 28-pin package as in the PIC18F46J50 family. This would have the advantage of up to an eight-time increase in code space with a slightly higher cost (approximately $3 versus $4). While the package size does increase a bit with a larger part, you don't need a special adapter for debugging and you might make good use of the extra I/O.

I admit that I can think of a number of improvements that would increase this project's flexibility and ease of use. Do you have a favorite interface that isn't covered here and think would add to this project's versatility? Your constructive comments can help make projects like these more useful for all of us.

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/2012/259.

RESOURCES

J. Bachiochi, "Quantifying Precipitation: A Device to Measure Rainfall Volume," *Circuit Cellar* 258, 2012.

———, "Error Checking," *Circuit Cellar* 250, 2011.

———, "Application Communication with USB (Part 3): Assembly Code Finale," *Circuit Cellar* 241, 2010.

———, "Application Communication with USB (Part 2): The Importance of Descriptors," *Circuit Cellar* 240, 2010.

———, "Application Communication with USB (Part 1): The Enumeration Process Explained," *Circuit Cellar* 239, 2010.

I²C Bus, www.i2c-bus.org/addressing.

Maxim Integrated Products, "Application Note 126, 1-Wire Communication through Software," 2002, www.maxim-ic.com/app-notes/index.mvp/id/126.

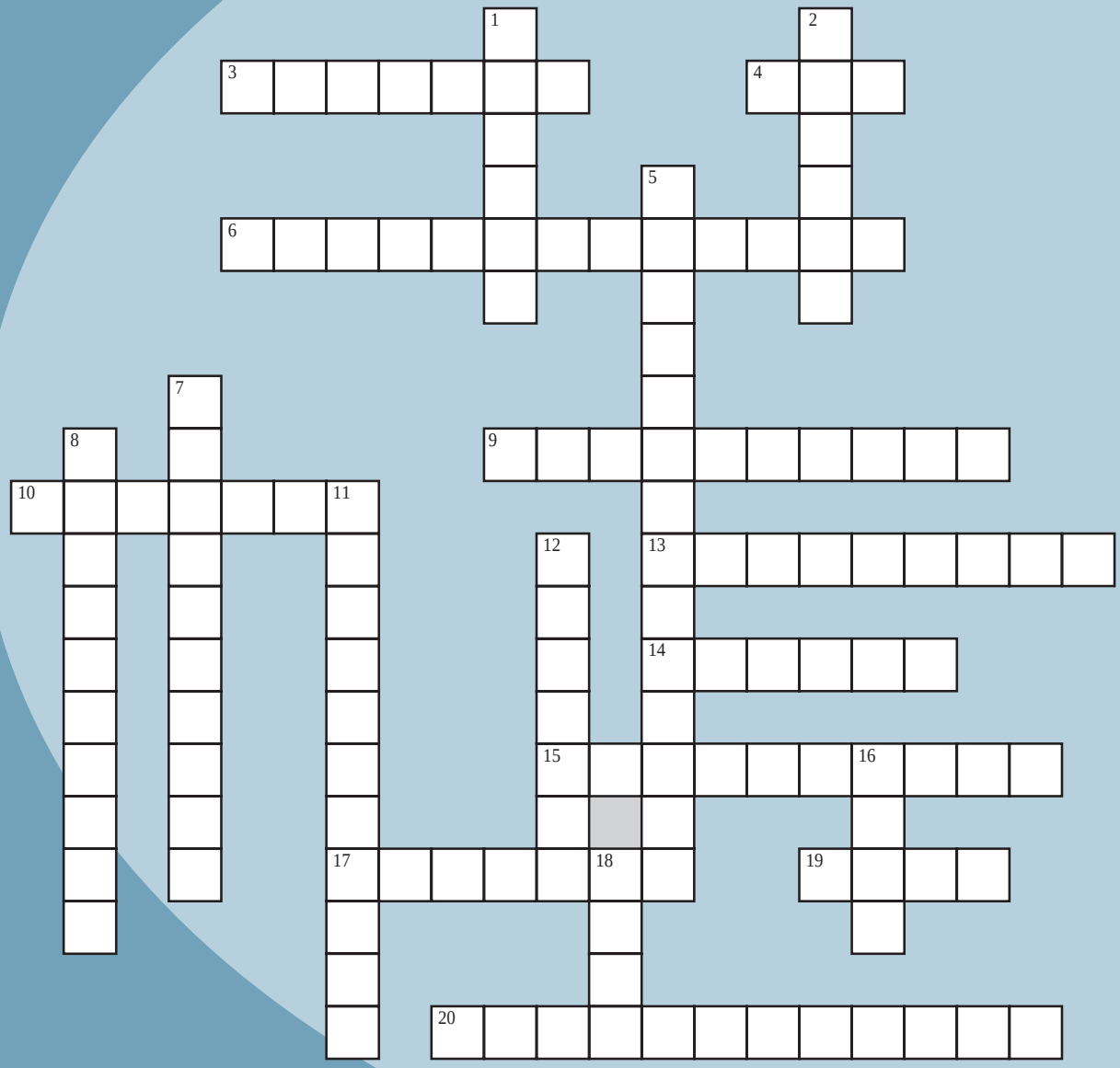
———, "Application Note 187, 1-Wire Search Algorithm," 2002, www.maxim-ic.com/app-notes/index.mvp/id/187.

SOURCES

DS2480B Serial-to-1-Wire line driver, MAX232 drivers/receivers, and MAX481 transceivers
Maxim Integrated Products, Inc. | www.maxim-ic.com

PIC18F14K50 Microcontroller and PIC18F46J50 family of microcontrollers
Microchip Technology, Inc. | www.microchip.com

CROSSWORD



Across

3. Having the same turning point
4. Used to compress and archive data
6. Moves groups of data [two words]
9. Unreadable without a key
10. A short length of wire used to connect components
13. Uses an I^2C bus as its electrical hardware interface [two words]
14. A group of numbers or symbols found in a rectangle
15. A type of signal that represents a binary digit of 1 [two words]
17. Has the power to accept requests from and grant control of buses
19. Can be a character, digit, signal, or symbol
20. The relationship between the signal level at the output of a device to its input under specific conditions [two words]

Down

1. A method of checking accuracy
2. Fluctuate, vibrate
5. Operates from single-phase AC or DC power [two words]
7. Audible at more than 20,000 Hz
8. An audio signal that isn't quite right
11. Radiates acoustic energy
12. Used to balance current flow
16. Stop command
18. Sound wave bounce

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

Audio amateur

Build the speaker of dreams for your:

Home, Car, or Home Theater!

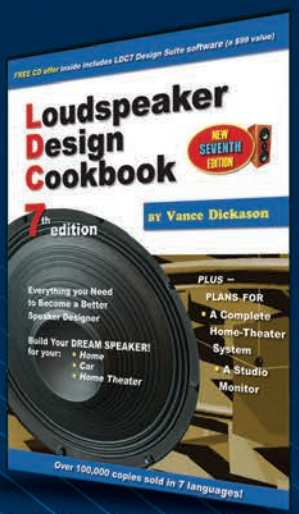


Back and better than ever, this 7th edition provided everything you need to become a better speaker designer. This newly revised edition now includes Klippel analysis of drivers, a chapter on loudspeaker voicing, advice on testing and crossover changes, and so much more! Ships complete with bonus CD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



IDEA BOX

THE DIRECTORY OF PRODUCTS AND SERVICES

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



$795
introductory price

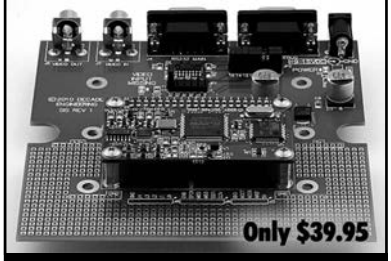
- ¼ " **Ultrasonic Microphone** & Amplifier
- 500 Hz – 100KHz +/- 3 dB BSL
- Sensitivity: 1 V/P

KarlZurn@gmail.com
321-613-3862

FLORIDA RESEARCH INSTRUMENTS, INC.
www.FloridaResearchInstruments.com

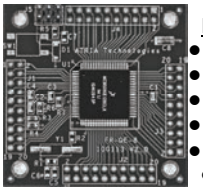
NEW **SIS Board**

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



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

i2cStick™



Discover I²C
Read, Write, Connect...

MCC
Micro Computer Control

www.mcc-us.com

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

The industry-standard BASIC compiler for Microchip PIC® microcontrollers.

PICBASIC PRO™ Compiler 3.0



Experimenter - $49.95
Silver Edition - $119.95
Gold Edition - $269.95

Academic Licensing Available

Download a FREE trial version now.

www.PBP3.com

PICBASIC and PICBASIC PRO are trademarks of Microchip Technology Inc. in the USA and other countries.
PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.

ALL ELECTRONICS CORPORATION
Electronic and Electro-mechanical Devices, Parts and Supplies. Wall Transformers, Alarms, Fuses, Relays, Opto Electronics, Knobs, Video Accessories, Sirens, Solder Accessories, Motors, Heat Sinks, Terminal Strips, L.E.D.S., Displays, Fans, Solar Cells, Buzzers, Batteries, Magnets, Cameras, Panel Meters, Switches, Speakers, Peltier Devices, and much more....
www.allelectronics.com
Free 96 page catalog
1-800-826-5432

PIC-SERVO MOTION CONTROL
MOTION CONTROLLERS FOR BRUSH, BRUSHLESS AND STEPPER MOTORS.
• controller chips
• controller boards
www.picservo.com
JEFFREY KERR, LLC

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

Revolutionary new expandIO-USB chip
USB
expandIO-USB
PC
USB Status
A/D-I/O-SPI-I2C
• Ideal for adding USB to sensors & peripherals
• No drivers needed for Windows, Mac, Linux
• No microcontroller programming required
• Also check out our USB-232 USB to UART
www.hexwax.com - Buy from Mouser & Farnell

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

EZ Web Lynx WiFi
Develop Embedded WiFi
Using Just HTML
EZ Web Lynx WiFi Development Kit w/ IDE and Tutorial only $179
WiFi module only $67
Qty 100+ $47
• Includes docking station and EZ Web Lynx WiFi module
• Interfaces to sensors, switches, relay drivers, motor drivers and more
• Use HTML IDE to develop custom dynamic web pages and send alarm/status emails
www.ezweblynx.com/ccwifi
sales@ccsinfo.com
262-522-6500 x 35

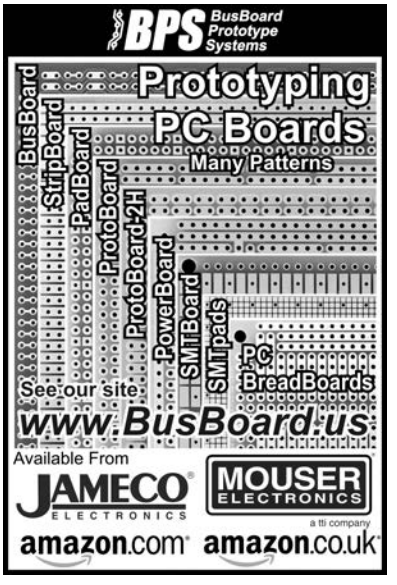
INTEGRATED
PLCs with Integrated Ethernet
Integrated Features :
- ETHERNET / Modbus TCP/IP
- 16 or 32 digital I/Os
- 10 analog I/Os
- RS232 and RS485
- LCD Display Port
- I/O Expansion Port
- Ladder + BASIC Programming
$229 and $295
before OEM Qty Discount
tel : 1 877 TRI-PLCS
web : www.tri-plc.com/cci.htm
TRIANGLE RESEARCH INTERNATIONAL

JazzOS - Java for Makers
Get started with the JazzOS Explorer
Program in Java
32 Bit Native Code Performance
28 Multipurpose I/O Channels
8 Megabytes RAM
16 Megabytes FLASH
Plug-and-Play Development
$89.00*
* Quantity 1 ~ Volume discounts available.
[PERNETICS] Practical Innovation
PERNETICS.COM
(800) 651-8698











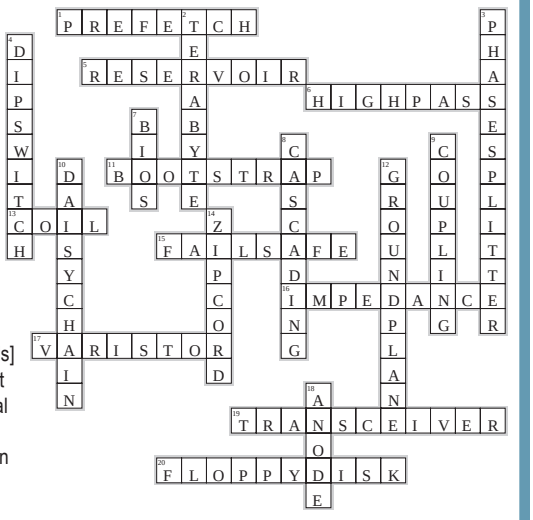
CROSSWORD ANSWERS from Issue 258

Across

1. **PREFETCH**—Saves it for later
5. **RESERVOIR**—A capacitor that holds and releases a DC charge
6. **HIGHPASS**—Says yes to high frequencies and no to low frequencies [two words]
11. **BOOTSTRAP**—A circuit that does things on its own
13. **COIL**—An inductor
15. **FAILSAFE**—Defaults in the event of a power loss [two words]
16. **IMPEDANCE**—Measures electric current
17. **VARISTOR**—A resistor whose value decreases when its voltage gets too high
19. **TRANSCEIVER**—Combines a transmitter and a receiver
20. **FLOPPYDISK**—Created in the early '70s and replaced by CD-ROMs in the '90s [two words]

Down

2. **TERABYTE**—1,099,511,627,776 bytes
3. **PHASESPLITTER**—Separated signals [two words]
4. **DIPSWITCH**—Used on PCBs instead of jumper blocks [two words]
7. **BIOS**—Windows-based computers rely on it to get going
8. **CASCADING**—Circuits feeding one another
9. **COUPLING**—Transfers energy from circuit to circuit
10. **DAISYCHAIN**—A ring of wires [two words]
12. **GROUNDPLANE**—A layer of copper that helps reduce noise and increase potential [two words]
14. **ZIPCORD**—Electrical conductors that can be easily pulled apart [two words]
18. **ANODE**—Works the opposite way of a cathode



CLIENT INDEX

Page		Page		Page		Page	
77	All Electronics Corp.	5, 11	DesignSpark chipKIT™ Challenge	78	Ironwood Electronics	C4	Oscium
42	Anaren	55	Earth Computer Technologies	77	Jeffrey Kerr, LLC	77	Pernetics
59	AP Circuits	9	egnite GmbH	9, 78	JK microsystems, Inc.	35	Pico Technology Ltd.
37	ARM	22, 23	Elektor	29	Labcenter Electronics	41	Pololu Corp.
76	Atria Technologies, Inc.	69	Elektor	13, 78	Lemos International Co.	32	Saelig Co., Inc.
33	Bitscope Designs	27	Elsevier, Inc.	77	Maxbotix, Inc.	13	SiliconRay
78	Busboard Prototype Systems	43	EMAC, Inc.	76	MCC, Micro Computer Control	2, 3	Technologic Systems
67	CC Gold	21	Express PCB	1	Microchip Technology, Inc.	78	Tern, Inc.
47	Comfile Technology, Inc.	77	FlexiPanel Ltd.	76	microEngineering Labs, Inc.	77	Triangle
51	CTIA Wireless	76	Florida Research Instruments, Inc.	43	Midwest Patent Law		
77	Custom Computer Services, Inc.	55	Grid Connect, Inc.	7	Mouser Electronics, Inc.		
76	Decade Engineering	17	Humandata Ltd.	C2	NetBurner		

PREVIEW of March Issue 260

Theme: Robotics

Robotic Navigation: Positioning and Routing with Acoustic Delay Triangulation

The Rainbow Color Reader (Part 1): A Talking Color Identifier for the Visually Impaired

NakedCPU (Part 2): Experimentation and Communication

All-in-One Joystick: Build a Propeller-Based TV Gaming Platform

---

THE CONSUMMATE ENGINEER **Microelectronic Component Obsolescence:** How to Prepare for an Inevitable Scarcity of Parts

FROM THE BENCH **Wheel-Free Mobile Robotics**

EMBEDDED UNVEILED **Infrared Thermal Detectors**

LESSONS FROM THE TRENCHES **Printer Port Connections**

ATTENTION CLIENTS

April Issue 261 Deadlines

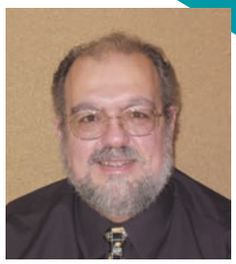
Space Close: Feb. 13
Materials Close: Feb. 20

Theme

Embedded Programming

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

Business in the Cloud?

There's a lot of talk these days about freeing the user from the confines of a particular device via "cloud computing." A typical example is Apple's "i" family of devices where you can buy a song on iTunes with your iPhone, and then also play it on your iPad or home computer. Similarly, you can take a picture on any of these devices and share it seamlessly on all the others, or share it with your friends.

As a concept, this is a great thing, but it does have some strings. The song or picture, along with information about your ownership details, is stored out there "in the cloud," or more specifically, somewhere on Apple's servers. We don't get excited over this because sharing photos, music, etc. among casual users isn't considered "real data." It's merely an extension of the whole concept of social networking that is already relatively open.

The next level of cloud services from Google and others includes complete suites of web-hosted office applications (word processing, spreadsheet, database, etc.) and a lot more "real data" vulnerability. Along with services such as YouSendIt (file transmission) and Carbonite (file backup), the intention is for you to run your whole business on cloud servers and have instant ubiquitous access.

The question is: Are these services really up to the task of handling the data of a serious business?

Let's talk about reliability first. Can you rely on important data being accessible the instant you need it? Well, if you take a look at the "Terms of Service" on any of the web services, they read pretty much like the end user license agreements (EULAs) we've all been reading for the past 30 years: this service is not guaranteed to do anything useful, this service is not guaranteed to be available, we are not responsible for any direct or consequential harm caused by this service, blah, blah, blah. This isn't the sort of thing that should inspire confidence in anyone trying to start or run a business in today's shaky economy.

At the very least, it means that for any data you put into a web application, you'd better make a private back-up copy and a back-up plan. You have to have access to that back-up copy via a means that does *not* involve the same application's cloud server, an alternate ISP if the primary one fails (I have both cable and DSL for that reason), and a way to run the application elsewhere if the cloud business running the web app goes south.

But the bigger concern in my mind is the question of security. Who has access to the data you put into a web application, both now and in the future? Again, looking at those applications, I have yet to find a Terms of Service that promises any level of privacy for your data. Indeed, Google is particularly egregious in this area. Their Terms of Service (https://accounts.google.com/TOS) includes item #11, "Content License from You," which states, "By submitting, posting, or displaying the content you give Google a perpetual, irrevocable, worldwide, royalty-free, and non-exclusive license to reproduce, adapt, modify, translate, publish, publicly perform, publicly display, and distribute any Content which you submit, post, or display on or through, the Services." This is about as far from a promise of privacy as you can get!

You really need to think defensively about your data. Surely, Google harbors no intent to harm any business using its services, but with tens of thousands of employees, it's hard keep track of the one among them with internal access to your data who is less scrupulous. Without concrete liability for misappropriating or deleting client data, can you trust placing your confidence and livelihood in an illusion of credibility and a mere promise of service? Yes, this is a harsh statement, but show me where these guys take responsibility even if they screw up everything.

Be proactive with sensitive business files! When using file transmission or storage services, either encrypt (and/or digitally sign) the data and/or deal only with companies that understand the real issues involved. One good example is Intuit, a company that provides accounting services to small businesses. (Carbonite and YouSendIt also fall into this category.) They devote an entire section of their website (http://security.intuit.com) to the issues of client privacy and data security.

The bottom line is that existence in the "cloud" is an up-and-coming trend, both personally and commercially. Always read the terms of service and privacy/security documentation of any web service you're considering using and carefully evaluate which kinds of data you're going to put out there. But most of all, in this wild and wooly age of botnets and malicious intent, make sure you have an independent copy and a back-up plan for any data you consign to the ether.

steve.ciarcia@circuitcellar.com



CIRCUIT CELLAR

Start the New Year off right
with Shlomo Engelberg's

ADuC841 Microcontroller Design Manual:
From Microcontroller Theory to Design Projects



$35 USD

Shlomo Engelberg provides a comprehensive guide to designing and programming with the Analog Devices, Inc. ADuC841 microcontroller as well as other microcontrollers in the 8051 family. This book provides valuable introductory and advanced labs which demonstrate the many ways you can use a microcontroller to solve practical problems. Perfect for university classrooms and self study!

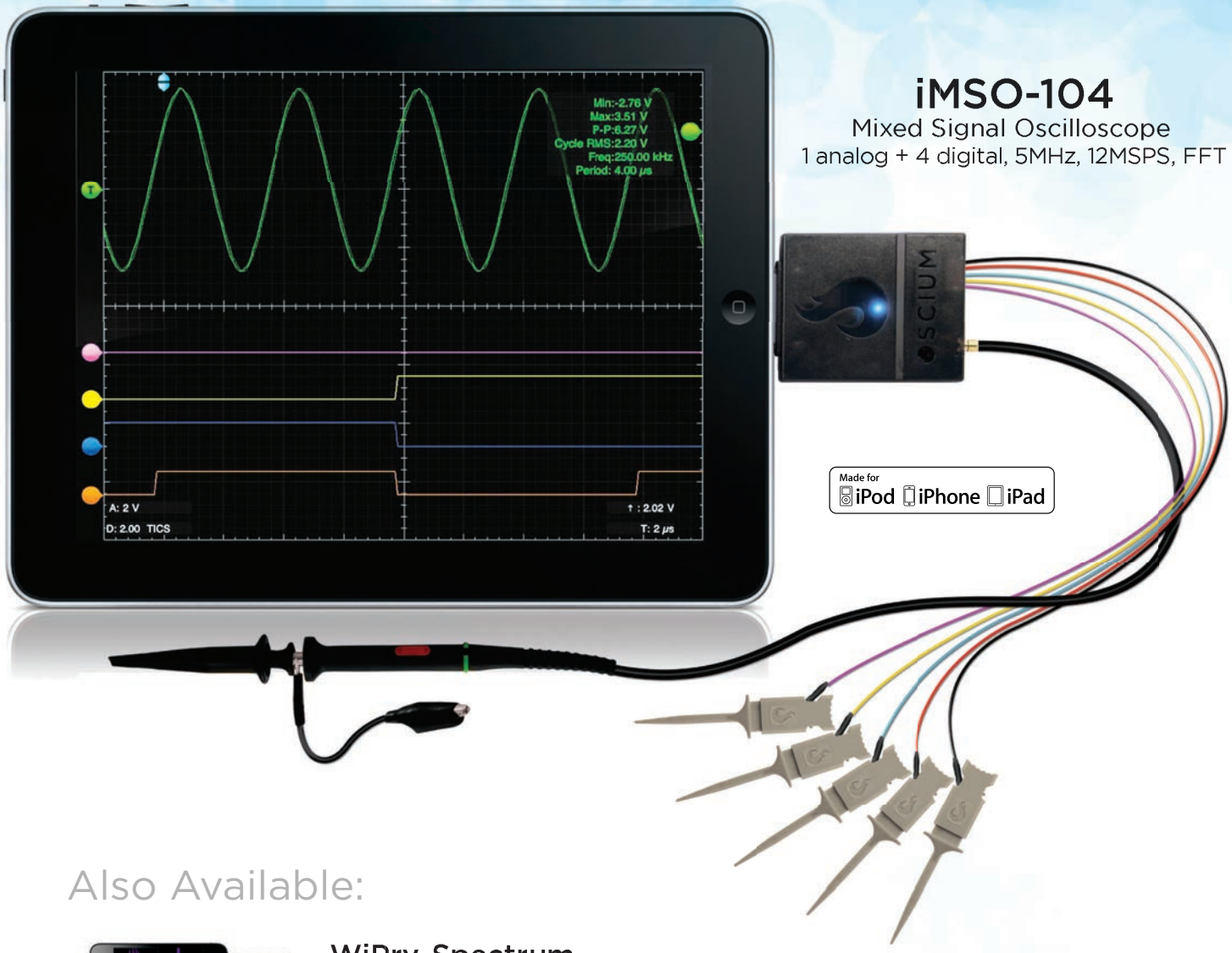
To get you started, Analog Devices and Circuit Cellar will select 12 lucky readers to receive a complimentary ADuC841 QuickStart Development Kit. Buy the book today and you could be a winner!



Visit **www.cc-webshop.com** and enter **coupon code KITDRAW2012** at checkout.*

*By entering coupon code KITDRAW2012 at checkout you are indicating that you have read and agree to the terms and conditions posted at www.circuitcellar.com/books. Promotion is open to U.S. residents only and ends February 29, 2012.

Tired of **Buttons** & **Knobs**?

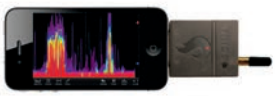


iMSO-104

Mixed Signal Oscilloscope

1 analog + 4 digital, 5MHz, 12MSPS, FFT

Also Available:



WiPry-Spectrum

2.400-2.495GHz, -40dBm to +20dBm
Resolution 2.0dBm, Sweep Time 200ms



WiPry-Power

100MHz to 2.7GHz, -45dBm to +20dBm
Resolution 0.2dBm, 2 µs/div to 1 s/div



WiPry-Combo

WiPry-Spectrum & WiPry-Power Combined!

OSCIUM

www.oscium.com