IMPLEMENTATION: DSP for Lighting Apps
ORIGIN: Italy
PAGE: 38

INSIGHT: Understand & Control Current Consumption
ORIGIN: United States
PAGE: 60

DESIGN TIP: Timekeeping with an RTCC
ORIGIN: United States
PAGE: 66

CIRCUIT CELLAR

THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

MARCH 2011
ISSUE 248

ROBOTICS

MCU-Based Digital Pan Head Controller

Computer-Controlled Machining

Build a Remotely Operated Vehicle

Super-Regenerative Receiver

PLUS

Designer Competency

Project-Essential Concepts

// Scheduling
// State Machines
// Simulation
// Code Validation

0 3>
7 25274 75349 9
$7.50 U.S. ($8.50 Canada)

www.circuitcellar.com

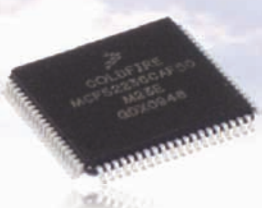
Low-cost Industrial
Serial to Ethernet Solutions

NetBurner
Networking in One Day!

- Instantly network-enable any serial device
- No programming is required for serial to Ethernet application
- Customize to suit any application with a development kit

SBL2e Chip

2-port serial-to-Ethernet server with eight A/D converter inputs and optional SPI, I2C peripheral device support



SBL2e 200

2-port serial-to-Ethernet server with four A/D converter inputs, optional I2C peripheral support and 10-pin header

SBL2e 100

2-port serial-to-Ethernet server with four A/D converter inputs, optional I2C peripheral support and RJ45 connector

SBL2e

Serial-to-Ethernet server with RS-232 support

SBL2e XA

External server with up to four A/D converter inputs, up to eight digital I/O and up to two UARTS (one RS-232)

Hardware Features

Up to three serial ports, 10/100 Mbps Ethernet, up to 10 digital I/O, 12-bit A/D converters, operating temperature -40 to 85 C, 32-bit performance

Software Features

TCP/UDP/Telnet/HTTP modes, DHCP/Static IP support, web-based or AT command configuration

Low Prices

SBL2e Chip .... $10.50 (Qty. 10k, Device P/N: SBL2eCHIP-236IR)
SBL2e 200 .... $19.95 (Qty. 1000, Device P/N: SBL2e-200IR)
SBL2e 100 .... $21.95 (Qty. 1000, Device P/N: SBL2e-100IR)
SBL2e X.......................... $79.00 (Qty. 1000, Device P/N: SBL2eX-100IR)
SBL2e XA........................ $79.00 (Qty. 1000, Device P/N: SBL2eXA-100IR)

Need a custom solution?

NetBurner Serial to Ethernet Development Kits are available to customize any aspect of operation including web pages, data filtering, or custom network applications. All kits include platform hardware, ANSI C/C++ compiler, TCP/IP stack, web server, e-mail protocols, RTOS, and NB Eclipse IDE.

Information and Sales | sales@netburner.com

Web | www.netburner.com **Telephone** | 1-800-695-6828

TASK MANAGER

Control Electronics

Robotics is a term that conjures up a variety of images, from humanoid prototypes to Hollywood animatronic designs to floor-vacuuming consumer appliances. Since 1988, we've published articles about projects pertaining to such robotic electronics applications. In fact, many of our readers both directly and indirectly contributed to the development of those systems, many of which are now commonly used in residential and industrial environments. But this issue doesn't feature typical robotics applications. We're highlighting a topic a bit more specific: control electronics.

MCU-based machine control is one of the most fascinating fields of electronics engineering. And, each year, it's growing in importance as engineers find new ways to incorporate MCU-based sensor systems and motorized designs in the industrial automation, defense, aerospace, and consumer electronics industries (to name only a few). Thus, we're dedicating much of this issue to the ever-evolving field of control electronics.

Turn to page 20 to learn how Richard Lord built a panning control system. The motorized design enables a photographer to rotate a camera to take uniformly overlapping images without distortion.

Next, consider Brian Senese's remotely operated vehicle (ROV) project, which he describes on page 28. You know how a basic RC vehicle system works. But that isn't Brian's focus. Again, what's important is the concept of electronics control. The idea is to plan, build, and then program an efficient system that provides end users with complete control over a mobile vehicle. The possibilities for such applications are endless.

On page 38, Marco Signorini addresses the important topic of light control, which is particularly important in automotive, aerospace, and entertainment applications. He describes how to embed a DSP device and a series of constant-current generators to drive a group of high-power LEDs. Follow his lead to build a lighting control system of your own.

Brian Millier uses control electronics in a more industrial environment: his workspace. In an article titled "CNC Router Design," he presents his computer-controlled CNC router system. A master at instrumentation engineering (just refer to his previous *Circuit Cellar* articles), Brian understands that precision cutting and drilling are the keys to working successfully with metals, woods, and plastics.

We're confident these projects will inspire you to develop your own electronics control systems. However, if after reading this issue you still have a hankering for a classic robotics project like the vision-guided balancing robot Hanno Sander described in *Circuit Cellar* 224 (March 2009), go to www.cc-webshop.com and search away!

C. Abate

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

WEST COAST EDITOR
Tom Cantrell

CONTRIBUTING EDITORS
Jeff Bachiochi
Robert Lacoste
George Martin
Ed Nisley

NEW PRODUCTS EDITOR
John Gorsky

PROJECT EDITORS
Ken Davidson
David Tweed

PUBLISHER
Hugo Van haecke

ASSOCIATE PUBLISHER
Shannon Barraclough

CUSTOMER SERVICE
Debbie Lavoie
Kim Hopkins

CONTROLLER
Jeff Yanco

ART DIRECTOR
KC Prescott

MARKETING ASSISTANT
Daniel Netkin-Colllins

GRAPHIC DESIGNERS
Grace Chen
Carey Penney

STAFF ENGINEER
John Gorsky

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
Valerie Luster
E-mail: val.luster@circuitcellar.com

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

PRINTED IN THE UNITED STATES

CONTACTS

SUBSCRIPTIONS
Information: www.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

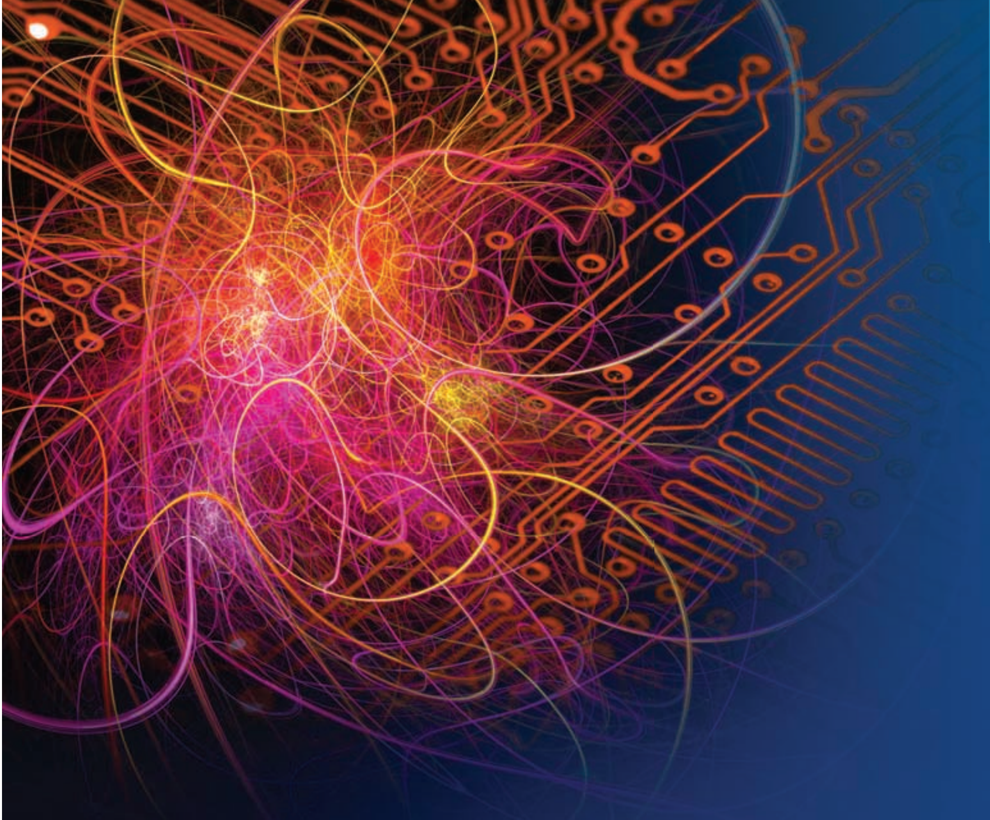
GENERAL INFORMATION
860.875.2199, Fax: 860.871.0411, E-mail: info@circuitcellar.com
Editorial Office: Editor, Circuit Cellar, 4 Park St., Vernon, CT 06066, E-mail: editor@circuitcellar.com
New Products: New Products, Circuit Cellar, 4 Park St., Vernon, CT 06066, E-mail: newproducts@circuitcellar.com

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®, THE MAGAZINE FOR COMPUTER APPLICATIONS (ISSN 1528-0608) is published monthly by Circuit Cellar Incorporated, 4 Park Street, Vernon, CT 06066. Periodical rates paid at Vernon, CT and additional offices. **One-year (12 issues) subscription rate USA and possessions $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.

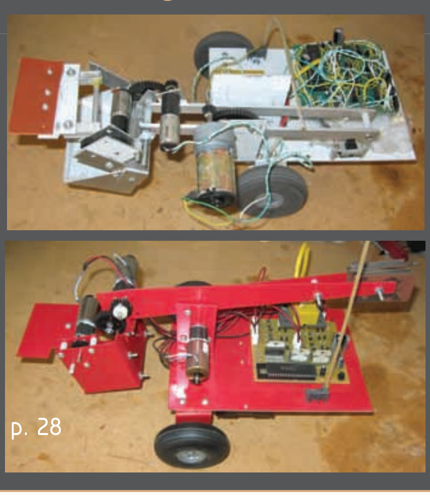


INSIDE ISSUE 248

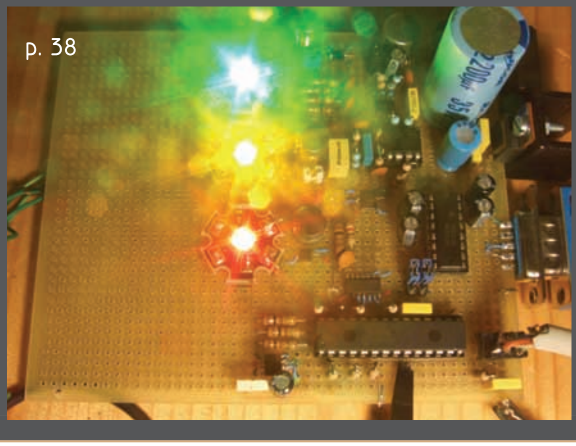
March 2011 • Robotics

Panning Control System

ROV Design and Construction



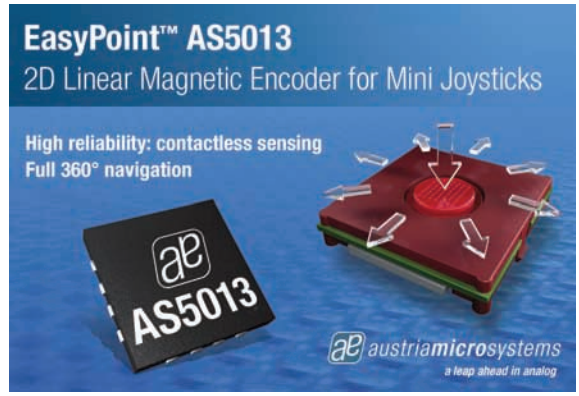
Embedded DSP



HALL SENSOR FOR 360° NAVIGATION IN HMIs

The **AS5013** is a complete Hall sensor IC for human-machine interface (HMI) applications requiring low power. The sensor comprises a contactless magnetic encoder IC that monitors the displacement of a magnet incorporated in a knob relative to its center position, and it provides x and y position information via an I²C interface. The AS5013 is used in the EasyPoint mini-joystick module, which consists of a mechanical stack incorporating a navigation knob, a magnet, and the encoder IC. Its simple construction and contactless sensing technology give the module very high reliability, and it is designed for any kind of 360° navigation input device.

The IC includes user-selectable power-saving modes, five integrated Hall sensing elements for detecting up to ±2-mm lateral displacement, a high-resolution ADC, an XY coordinate and motion detection engine, and a smart power management controller. The XY coordinate registers and magnetic field information for each Hall sensor element are transmitted over the I²C interface to the host processor.



The HMI device also provides two interrupt modes: motion detect, and data ready; and two operating modes: idle mode, with less than 3 µA current consumption, and low-power mode, with selectable readout rate. The low-power AS5013 operates over a power supply range of 2.7 to 3.6 V, and down to a 1.7-V peripheral supply voltage.

The AS5013 contactless magnetic encoder IC is available now and costs **$2.92** in 1,000-piece quantities.

austriamicrosystems
www.austriamicrosystems.com

OPTICAL FINGER NAVIGATION MODULE

The **Optical Finger Navigation (OFN) Module** can add a unique human interface component to BASIC Stamp or Propeller projects. OFN Modules are becoming popular as user input devices in many smart phones as a replacement for trackballs, which are subject to mechanical wear and tear. OFN technology is similar to the technology used in optical mice, and movement across the sensor can be read by any MCU using I²C communication.

The module features a built-in center select button, an on-board red LED that illuminates when finger movement is detected, easy I²C communication to interface with virtually any microcontroller, and an on-board voltage regulator. Resolution is selectable from 500 to 1,000 cpi, and the module offers a breadboard-friendly package with 0.1" pin spacing.

Application ideas include video game input, mouse replacement, or user input for computing devices. The Optical Finger Navigation Module (part number 27903) costs **$19.99**.

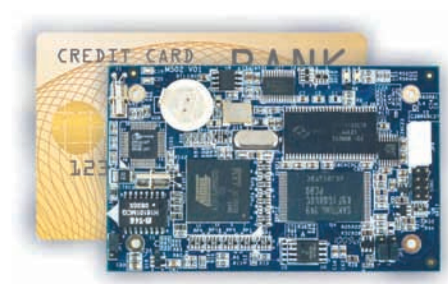
Parallax, Inc.
www.parallax.com

LINUX-BASED ARM9 READY-TO-GO SYSTEM ON MODULE

The M-502 is an ARM9, Linux-based system on module (SoM). The M-502 is powered by a 400-MHz ARM926EJ-S ARM Thumb processor with a memory management unit, and it's equipped with 64-MB SDRAM, 128-MB NAND flash, and 2-MB data flash. The M-502 is also pre-installed with Linux 2.6.29 OS, busybox utility collection, lighttpd webserver, and various hardware device drivers. The module also provides flexibilities in peripheral expansion by integrating one 10/100-Mbps ethernet, two USB 2.0 hosts, four UARTs with hardware/software flow control, and 32 programmable digital I/Os. In addition, the M-502 comes with a secure datacard (SD) interface, serial peripheral interface (SPI), I²C bus, I²S bus, and an 8-bit local bus. The GNU C/C++ cross compiler is included in the development kit, allowing users to develop their application software on PCs without the need to learn and purchase an extra commercial developing tool.

The M-502 SoM combines powerful computing capabilities, large memory space, and high scalability, making it easy for users to develop their application without compromising system performance. Flexible and reliable design leaves users worry free when focusing on their application software development, reducing 80% of the development time and cost normally required.

Contact Artila for pricing.



Artila Electronics Co., Ltd.
www.artila.com

LOW-POWER DIGITAL AMBIENT-LIGHT SENSOR

The **MAX9635** is a digital ambient-light sensor (ALS) with a unique adaptive gain block. The MAX9635 consumes 100x less power than the nearest competitive product, significantly extending battery life.

Replicating the optical response of the human eye with electronic components is difficult. Traditional light sensors measure the amount of light in an environment regardless of wavelength. These designs are unduly influenced by ultraviolet and infrared light, which are not perceptible by the human eye.

Maxim's BiCMOS technology enables the integration of two photodiodes along with an optical filter to reject ultraviolet and infrared light. This allows the MAX9635 to replicate the optical response of the human eye and accurately measure visible light in a variety of environmental settings. Advanced algorithms correct for any spectra variations between light sources, ensuring an extremely accurate lux response.

The MAX9635 is equipped with an adaptive gain block that automatically selects the optimum gain range. The adaptive gain block provides the industry's widest dynamic range (more than 4,000,000 to 1), enabling the MAX9635 to measure light levels from 0.045 lux to 188,000 lux. As an added benefit, the device's low-light sensitivity makes it suitable for applications in which the sensor IC is placed behind black glass, which greatly reduces the amount of ambient light.

Pricing starts at **$1.20** in 1,000-piece quantities.

Maxim Integrated Products
www.maxim-ic.com

NPN

SMALL ISOLATED DC-TO-DC CONVERTER

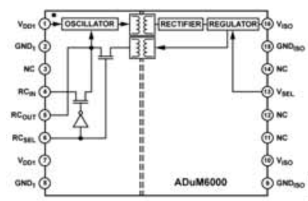
The **ADuM6000** is the industry's smallest isolated DC-to-DC converter. Offering a 5-kV root-mean square (RMS) isolation rating in a 10 mm × 10 mm package, the ADuM6000 is a 0.5-W device that integrates Analog Device's proprietary iCoupler digital isolation technology and isoPower DC-to-DC converter. The new digital isolator enables designers to free up valuable circuit-board real estate while eliminating the time-consuming step of securing medical or other safety approvals (e.g., IEC-60601-1), all at a fraction of the cost of alternative devices, including optocouplers.

The ADuM6000 digital isolator enables designers to reduce the form factor of their system module or to maintain the same form factor even as they add more features and functions. With alternate solutions, such as optocouplers and separate, isolated DC-to-DC converters, it may not be possible to add new functionality to existing system modules.

Also now available are ADuM620x series dual-channel digital isolators that feature isoPower integrated, isolated DC-to-DC converter technology and 5-kV RMS isolation rating. The ADuM620x series includes the ADuM6200, ADuM6201, and ADuM6202 digital isolation products that differ by channel configuration.

ADI's isoPower integrated, isolated DC-to-DC converter uses the same chip-scale transformer technology. But instead of transmitting data, isoPower employs switches, rectifiers, and regulators to generate power that is isolated to the same degree as the data channels.

The ADuM6000 costs **$3.96** in 1,000-piece quantities.



Analog Devices, Inc.
www.analog.com

LOW-COST, SMALL FORM FACTOR GPS RECEIVER

The **M520** is a new GPS receiver with excellent tracking and acquisition capabilities. The M520 GPS module is a highly sensitive, compact single-chip solution for GPS applications. It includes an RF receiver, complete baseband processor, flash memory, and a power control unit. The RF receiver minimizes systems costs by using single-conversion low-IF digital architecture with high-level integration, and leaving a few off-chip matching and decoupling components.

The baseband processor is controlled by adaptive signal processing, and navigation firmware is optimized for execution on a low-power microprocessor. Optimal signal acquisition and tracking strategies are enabled by sophisticated adaptive control algorithms.

The M520 costs **$21** in 10,000-piece quantities.

NavSync Ltd.
www.navsync.com

MICRO-PROXIMITY SWITCH

The **R12575** is a micro-proximity switch that's ideal for use in noncontact position sensing in very small spaces. Built around the proven GR200 ultra-small magnetic reed switch, the R12575 measures just 0.379" in length and 0.098" in diameter, yet provides unparalleled sensitivity and performance. These new switches are ideal for a variety of applications where size reduction is required without sacrificing performance. Typical applications include security systems, safe applications, and industrial control applications where the equipment design leaves little space for the proximity sensor.

The R12575 is well suited for applications where the magnetic field is very low and where space limitations are a major design factor. There is no power draw in the off state, which is an important consideration for battery-driven applications. Lead length and termination can be customized for specific customer specifications.

In addition to a small footprint and tight sensitivity range, with three standard sensitivity ranges available, the R12575 series is responsive—with a typical operating time of 0.2 ms, and a typical release time of 0.1 ms. The R12575 has a power rating of 3 VA maximum and will switch 0.1 Amp. DC (maximum) with a carry current rating of 0.5 Amps. DC. Switching voltage is 50 VDC (maximum). Boasting an operating range of -40° to 125°C, they are well suited for use wherever small size and sensitivity are required for non-contact position sensing applications.

Contact Standex for pricing information.

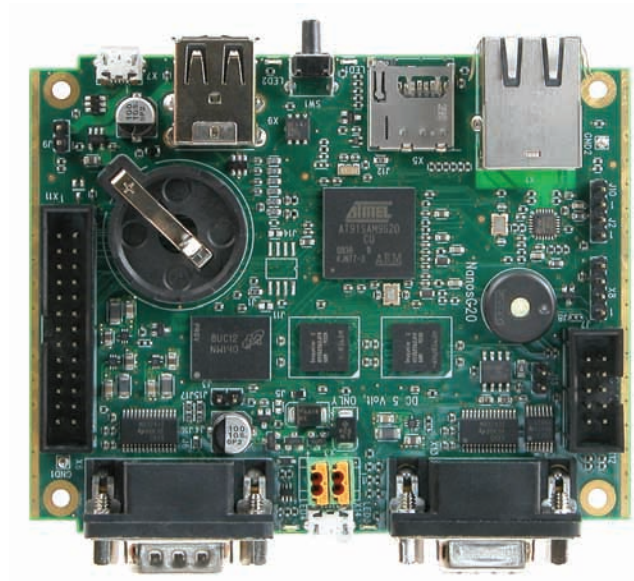
Standex Electronics, Inc.
www.standexelectronics.com

ARM9 SBC WITH LINUX

NanosG20 is a universal Linux computer that consumes just 200 mW during normal use. Its diverse interfaces allow you to connect Nanos quickly and efficiently into its environment and offer a wide variety of usage scenarios.

The full-fledged Debian Linux ensures a comfortable environment and direct access to common Linux software like Samba, Apache, and other powerful network applications. Nanos uses a modern Linux core with support for numerous USB devices like WLAN, Bluetooth, USB mass storage, and much more.

NanosG20 comes with its own compiler for easy program development. There is no need to install additional software on a development computer. Simply use Samba or NFS to unlock a directory on the Nanos, write scripts or programs in your favorite editor, and then use SSH, Telnet, or the serial console to compile and test on the Nanos.



As with any other Linux system, NanosG20 offers the choice of a variety of programming and scripting languages: C, C++, Java, Python, Perl, TCL/TK, and many more. Nanos uses very little electricity as it is, but with its energy-saving modes like power-down or standby, it consumes even less, allowing the board to be battery powered.

The NanosG20 costs around **$135**. A developer bundle containing an enclosure and power supply is also available.

Ledato GmbH
www.ledato.de

NPN

NEW LOW-COST FLOATING-POINT MCUs

Bridging the gap between low-cost Piccolo and high-performance Delfino floating-point microcontrollers are the new low-cost **TMS320F2806x Piccolo** floating-point MCUs. The new Piccolo MCUs offer an enhanced math engine specifically designed to simplify programming and optimize performance in real-time control applications that may require integrated communications. Developers of energy-efficient motor control and renewable-energy applications can now use a single F2806x MCU to cost-effectively execute control loops, as well as power line communications protocols and modulation schemes. Balancing performance with the integration and ease-of-use inherent to MCUs, the new F2806x MCUs also deliver a broader range of connectivity and memory options and are backed by TI's robust tools and free controlSUITE software.

Key features and benefits of F2806x Piccolo floating-point MCUs include a cutting-edge math engine comprising an 80-MHz floating-point C28x core, new Viterbi complex math unit (VCU), and control law accelerator (CLA) options. Increased communications and throughput are provided via USB 2.0 full-speed (host and device), CAN, and direct memory access (DMA). The MCUs also feature IQmath library for code compatibility and scaling throughout the C2000 platform, including the low-cost Piccolo series and high-performance Delfino floating-point series.

The new Piccolo F2806x floating-point MCUs start at **$4.95** in 1,000-piece volumes.

Texas Instruments, Inc.
www.ti.com

NEW STEPPER MOTOR CONTROLLER IC

The **USMC-01** is a unique stepper motor controller IC. The chip can operate as a slave or master. As a slave, it runs under another microcontroller (or PC) or it can operate in auto-run (master) mode, which allows the user to add a few switches and run the stepper motor manually. The controller allows the RPMs of stepper motors to be switch selectable. Gray encoded inputs ensure that only one input needs to be changed at a time to switch between consecutive RPM settings. The RPM calculations are made for 1.8° per step, 200 steps per revolution motors.

The USMC-01 generates control signals that can be used with for both unipolar and bipolar stepper motors with appropriate drivers like the L298 and L293 or even discrete transistors. In addition to its master and slave modes, the controller also features eight RPM selections in free-running mode. The USMC-01 offers half/full-wave step modes and direction control.

The USMC-01 costs **$9.95**.

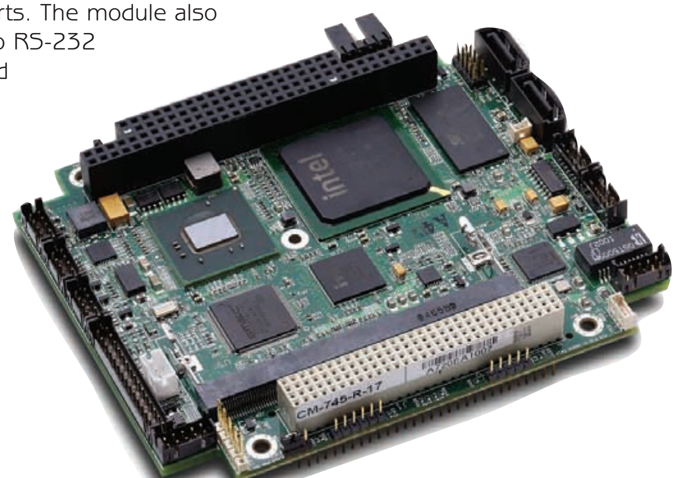
Images Scientific Instruments, Inc.
www.imagesco.com

RUGGED PC/104-PLUS SBC

The **Ampro CoreModule 745** is a PC/104-Plus module that supports a range of Intel Atom processors from the power-efficient N450 running at 1.66 GHz to the performance-oriented dual-core D510. With a thermal design power (TDP) as low as 9 W, the CoreModule 745 simplifies cooling requirements and enables conduction-cooled solutions for small-sealed enclosures in space-constrained applications.

Rugged by design, the module supports a wealth of legacy I/O interfaces, including ISA and PCI buses. Serial ports, one GbE port, and up to 2 GB of DDR2 667-MHz RAM are also incorporated into an expanded PC/104 footprint.

The CoreModule 745 operates at temperatures from –40° to 85°C, and withstands vibration up to 11.95 Grms, and shock up to 50 Grms. With a 4-GB solid-state drive soldered onboard, the module is a complete solution with no moving parts. The module also features a full 16-bit ISA bus, PCI 32-bit bus, one GbE port, two RS-232 serial ports, one RS-232/422/485 port, four USB 2.0 ports, and eight GPIO. In addition, the CoreModule 745 offers excellent graphics performance with an LVDS panel interface and legacy CRT support. For harsh environments, optional conformal coating is offered.



The CoreModule 745 is available now in production quantities with QuickStart Kits, including cables, 2-GB DDR2 RAM, device drivers, and board support packages for many popular operating systems, including VxWorks, Windows CE, Windows XP Embedded, Linux, and QNX.

Contact ADLINK Technology for pricing.

ADLINK Technology, Inc.
www.adlinktech.com

MODULAR CORTEX-M3 TOUCH SCREEN LCD-BASED DEV KIT

The latest addition to Future Designs's touch screen LCD kit family, the **DK-57TVS-LPC1788**, is based on the NXP LPC1788 microprocessor. The system's "brain" is a SOMMDIMM-LPC1788 SoC module, featuring an NXP LPC1788 120-MHz Cortex-M3 microcontroller. The SOMDIMM-LPC1788 features include an internal real-time clock, Ethernet PHY, 8-MB SDRAM, a microSD memory card connector, mini JTAG, 4 KB of internal EEPROM, and a secure external I^2C EEPROM. The DK-57VTS-LPC1788 also features a 5.7" VGA display (640 × 480 resolution) with a four-wire resistive touch-screen interface. Communication protocols include a USB host and device, RS-232/485, CAN, a SPI and I^2C, and peripherals, such as a three-axis digital accelerometer and temperature sensor.

The kit is based on FDI's μEZ (pronounced "Muse") rapid development platform, which includes an extensive library of open-source software, drivers, and processor support, all under a common framework. μEZ development works on the premise of "design once, reuse many times." μEZ enables companies to focus on innovation and on their own value-added applications while minimizing development time and maximizing software reuse.

The kit also includes a J-Link Lite JTAG debugger from Segger and comes complete with all cables and power supplies. All documentation is included on the 2-GB USB flash drive, so everything can easily be copied to the PC.

The DK-57VTS-LPC1788 costs **$460**.

Future Designs, Inc.
www.teamfdi.com

NPN

ECONOMICAL, ECO-FRIENDLY ELECTRONIC LABELING SYSTEM

The new **Green Machine** and **MaxiLabel Pro Ver. 3.0 Labeling Software for Windows** is a technologically advanced and environmentally friendly label printer system for anyone who needs to design and print professional pressure-sensitive labels in house and on demand.

The Green Machine is a rugged label, heat-shrink tube and barcode thermal-transfer printer with advanced features, including a large, high-intensity backlit display for use in all lighting conditions. It also has innovative new hot keys for the instant selection of type styles and point sizes, 12 Euro/Latin language prompts and a hot key, and the exclusive PEELGUARD electronic tape trimmer that rounds label corners on demand, creating more permanent and professional-looking labels. The Green Machine is available in PC or standalone models.

The Green Machine and supplies are designed and built with energy conservation and environmental compliance to the greatest extent possible, including the distinguished European Union Eco-Design Directive (2009/125/EC). Used in the field or on the desktop, the Green Machine runs on standard AA batteries or an international AC adapter (both included with the printer). For longer portable life, special nickel hydride rechargeable batteries can be used. Supply cartridges are recyclable and made from recycled plastic.

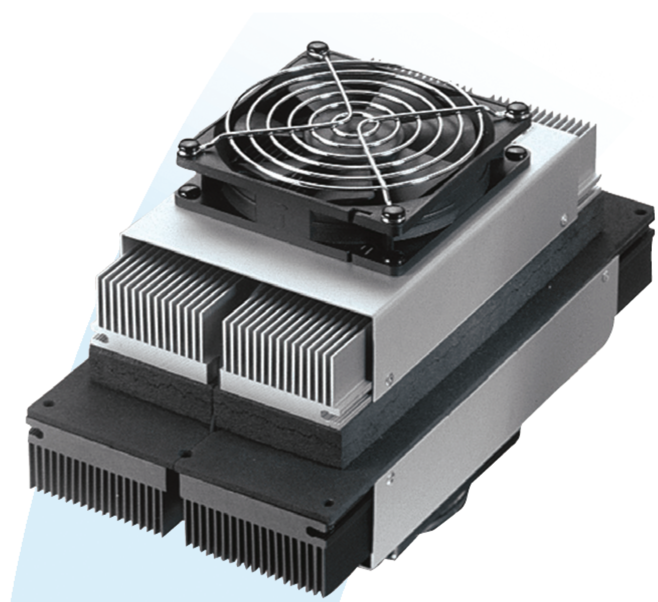
Combined with K-Sun's MaxiLabel Pro Ver 3.0 software, the Green Machine offers maximum power to create hundreds of ANSI safety, industrial, and general symbols, plus the ability to easily import custom images. Add-on symbols libraries include laboratory/medical, electrical, and homeland security.

Pricing for the Green Machine starts at **$339**.

K-Sun Corp.
www.ksun.com

OUTDOOR THERMOELECTRIC COOLER SERIES

The **AA Outdoor Cooler Series** is a ruggedized Air-Air Thermoelectric Assembly (TEA) that uses impingement flow to transfer heat. It offers dependable, compact performance by cooling or heating enclosures via convection. A bipolar thermostatic controller with predetermined set points is integrated inside the TEA to maintain its tight form factor. The unit is available in two popular set point configurations. The LK-81 is programmed to cool when the internal temperature of the enclosure exceeds 25°C and heat when the temperature drops below 10°C. Products in this series are available in four models offering 150 or 200 W capacities and 24 or 48 V operation.



The LE-80 series is programmed to cool when the internal temperature of the enclosure exceeds 35°C and heat when the temperature drops below 5°C. Products in this series are also available in four models that also offer 150 or 200 W capacities and 24 or 48 V operation.

The AA Outdoor Cooler series has been designed to pass rigorous Telcordia test requirements, such as earthquake resistance, salt, fog, wind-driven rain, high temperature exposure, and dust contaminants. This is due to the selection of world-class components, such as brand fans with the highest degree of environmental protection, heavy-duty anodization on the high-density heat sinks, lifetime-guaranteed waterproof connectors, overheat protection, and double environmental seals for the thermoelectric modules (TEMs).

Contact Laird Technologies for pricing.

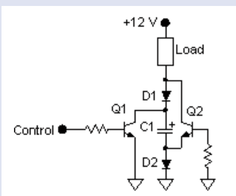
Laird Technologies, Inc.
www.lairdtech.com

CIRCUIT CELLAR

Test Your EQ

Edited by David Tweed

Problem 1—The following circuit is designed to give an extra voltage boost to a load at turn-on. It might be used, for example, to drive a relay or solenoid that requires significantly more voltage and current to "pull in" than it does to simply "hold." The control input is simply a logic-level signal that is either off (GND) or on (VCC). Let's analyze its operation.



When does Q2 turn on?

Problem 2—When do D1 and D2 conduct?

Problem 3—How much of the load current does Q1 need to handle?

Problem 4—If the load is an inductive load, such as a relay or solenoid, should a reverse-biased diode be added across it?

Problem 5—What can you do if you need more than 2x voltage boost at turn-on?

Contributed by David Tweed

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

by George Novacek (Canada)

The Super-Regenerative Receiver

You won't find a super-regenerative circuit in many modern radio receivers. It has frequency stability issues, a limited data rate, and audio quality shortcomings. But when it comes to remote control applications—especially those being developed on a tight budget—a super-regenerative circuit can be a viable option.

In the world of oratory competitions, the term *super-regenerative* is a lightweight compared to a tongue-twister like the 34-letter word *supercalifragilisticexpialidocious*. But in the engineering world, a sensitive, low-cost super-regenerative receiver is hard to beat. In my last column, I mentioned it as the old, reliable, workhorse of wireless control that's second to none in terms of the cost/performance ratio. Many engineers believe it's on the way to becoming obsolete. But while the principle is some 90 years old, I find it's still alive, well, and offered by RF module vendors.

CIRCUIT ANALYSIS

The pedigree of this circuit extends to the early days of radio. You can trace it back to the 1920s and a design by Edwin Armstrong. Then, vacuum tubes were expensive, and the engineers were looking for ways to make sensitive receivers with as few tubes as possible. They knew that when positive feedback was applied across an amplifier, oscillations would ensue at some point; but just before that happened, the gain of the amplifier would be approaching infinity. This principle, called "regeneration," was used to advantage in the radio receivers of the day. Unfortunately, due to the instability caused by a number of factors, it was necessary to continually adjust the feedback by hand, or the radio would oscillate and howl or the signal would fade away. Armstrong developed a circuit that would automatically keep the feedback just

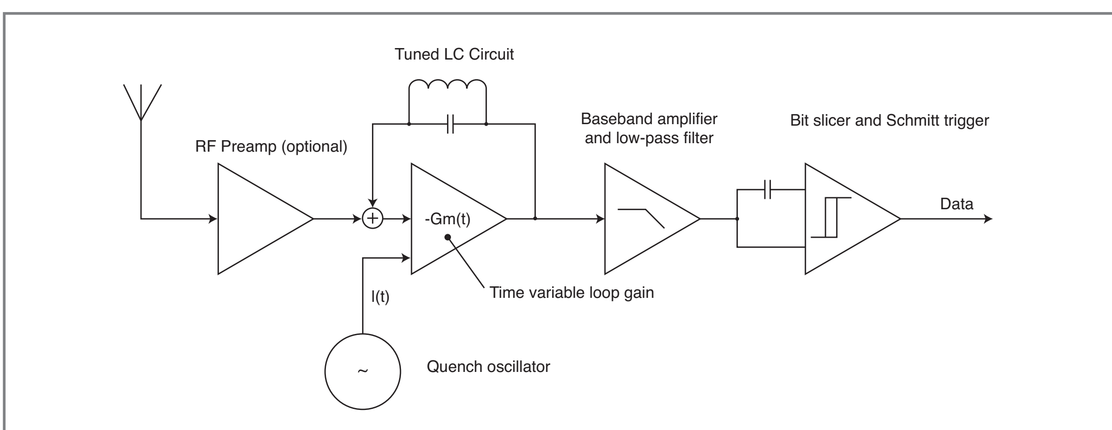


Figure 1—A typical super-regenerative receiver block diagram

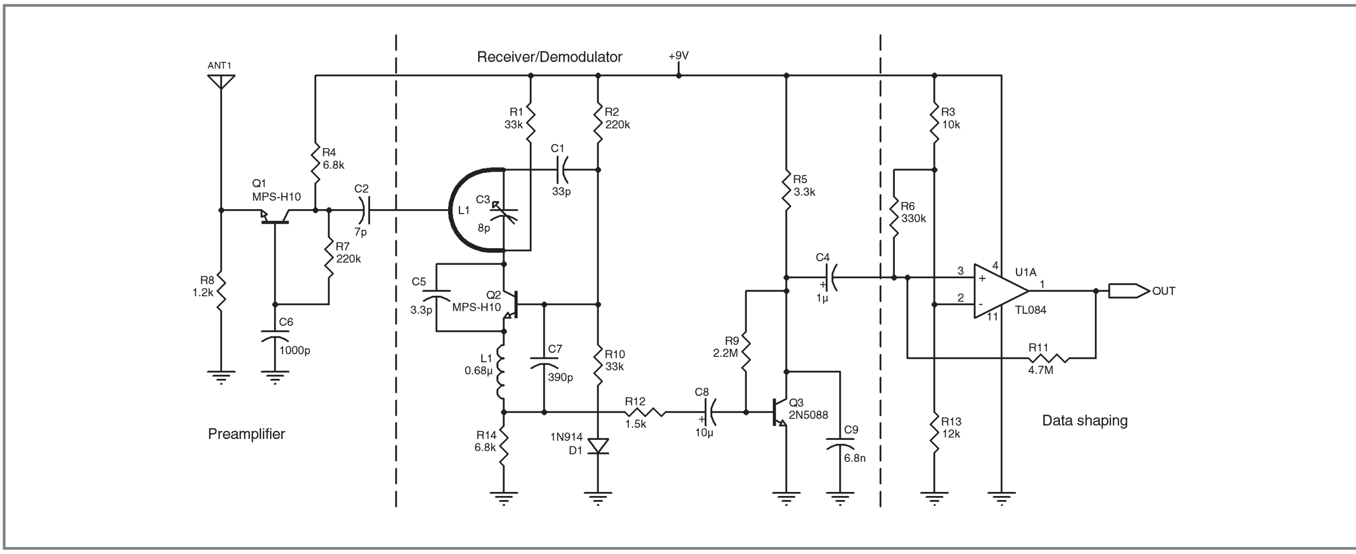


Figure 2—This is a working circuit with a preamplifier, Schmitt trigger, and more.

below the onset of oscillations and called the effect "super-regeneration."

At a repetition rate above the received baseband and below the RF range, the amplifier would automatically increase its positive feedback until oscillations were about to begin. At that point the bias conditions of the amplifier would be changed and the oscillations quenched. In effect, the receiver would sample the incoming RF signal at the quench rate with its maximum sensitivity usually reaching the noise floor. Since the quench signal repetition rate is well above the baseband range, a simple low-pass filter effectively suppresses it. The achievable RF sensitivity of the super-regenerative receiver is in a range of –30 to –100 dBm (decibels referenced to 1 mW).

Paradoxically to its common occurrence, due to its nonlinear operation, circuit analyses of this receiver are scarce. One, if not the best, analysis of the super-regenerative receiver is Eddie Insam's 2002 *Electronics World* article, "Designing Super-Regenerative Receivers." Anyone interested in understanding how this receiver works should study his paper.

INSIDE A RECEIVER

Figure 1 is a block diagram of a typical super-regenerative receiver. It should be noted that the bit slicer and Schmitt trigger circuit would be used for data transmission only. But the super-regenerative receiver can demodulate AM and FM signals, which would then be available on the output of the baseband amplifier.

Refer back to Figure 1. The preamplifier serves to amplify the received signal and to isolate the super-regenerative circuits from the antenna. It prevents back radiation of the super-regenerative oscillations to satisfy FCC rules, provides antenna matching, and stabilizes the load of the resonant tank circuit. The preamplifier is usually aperiodic and unless it has a significantly better noise figure than the super-regenerative detector, it has very little effect on the overall receiver sensitivity.

The time variable loop gain (i.e., the feedback amplifier) causes the resonant tank to oscillate at its resonant frequency. Because of the periodic quenching signal, the super-regenerative circuit operates in an intermittent

When an RF signal around the oscillator resonant frequency arrives at the time when the circuit is about to begin to oscillate, the injected RF signal from the antenna (or the pre-amp) forces the oscillations to start sooner. The difference in the start-up time is detected by the super-regenerative circuit and forms the envelope of the received signal.

oscillatory fashion. The negative conductance –Gm(t) is varied in such a way that the oscillator is driven in and out of oscillations without being allowed to reach a stable oscillatory state.

When an RF signal around the oscillator resonant frequency arrives at the time when the circuit is about to begin to oscillate, the injected RF signal from the antenna (or the pre-amp) forces the oscillations to start sooner. The difference in the start-up time is detected by the super-regenerative circuit and forms the envelope of the received signal.

The beauty of the receiver is that it can be built with a single transistor. It should be noted, however, that all the functional blocks also could be built as separate circuits. For a good example of this, refer to the article listed in the Resources section by Cedric Mélange, et al. This design is very useful for experimentation, but I am not convinced of its practicality. The main advantages of the super-regenerative receiver beside its sensitivity are simplicity, low power, and low cost. If I were to build a receiver with so many parts (as in the Mélange article), I would choose a superhet. Or, even better, I would purchase a monolithic receiver module, such as a Micrel "Quik Radio."

Figure 2 is the actual working circuit I showed last month, with the preamplifier and the data shaping bit slicer and a Schmitt trigger added. Q1 and its associated components form the aperiodic preamplifier, which, as I already pointed out, may be used but doesn't always have to be used. Its improvement of sensitivity, and therefore range, is often insignificant. Q2 forms the super-regenerative receiver, also called super-regenerative detector. The RF signal resonating in the L1/C3 tank circuit is coupled to collector Q2. C5 provides the necessary positive feedback for Q2 to oscillate at the resonant frequency of approximately 315 MHz. L1, C7, and R14 are the quench oscillator. R14 is also the load on which the baseband signal envelope appears. R2, R10, and D1 bias the base of Q2 into active an region for Q2 to start oscillating. D1 improves temperature stability of the circuit.

The quench oscillator should be running at a rate at least twice the maximum baseband frequency. Remember Nyquist? Generally, it is set between 20 and 120 kHz, which is high enough to be removed from the baseband with a simple low-pass filter.

The baseband signal is amplified by Q3. Its low-pass characteristics ensure attenuation of the quench frequency riding on top of the on/off-keyed (OOK) digital data. The following op-amp U1 works as a combined bit slicer and Schmitt trigger. Bit slicing performed by C4/R6 and the op-amp eliminate the DC component of the received signal, which can be especially troublesome when a strange transmitter operates in the same frequency band. Adding a positive feedback by R11 turns U1 into a Schmitt trigger with output clean enough to drive logic circuits.

CONTROL APP ADVANTAGES

Why isn't the super-regenerative circuit used in every radio receiver around us? Because it has poor selectivity and frequency stability, limited data rate, and the audio quality of an AM or FM modulated signal is barely acceptable. But for control applications, especially the cost-sensitive ones, its advantages outweigh the drawbacks. However, there are now monolithic receivers on the market without many of those shortcomings. At low prices, and with low power consumption, these successfully compete with the discrete component designs in all aspects. So, while there may no longer be an incentive to "roll your own," it never hurts to understand how things work. ■

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 of embedded control systems for aerospace applications. George wrote 26 feature articles for Circuit Cellar *between 1999 and 2004.*

RESOURCES

E. Insam, "Designing Super-Regenerative Receivers," *Electronics World*, April 2002.

C. Mélange, J. Bauwelinck, and J. Vandewege, "Low-Power, Super-Regenerative Receiver Targets 433-MHz ISM Band," *EDN*, 2006.

S. Rumley, "Super-Regenerative Receivers for Remote Keyless Access Applications," Valon Technology, www.valontechnology.com/images/REGEN.PDF.

Faster CPU and More Memory?
The CUWIN5000 Now Shipping!

CUWIN

Need more sophisticated features in a touch screen?
Enter the CUWIN? A Revolutionary Touch Controller.

Programmable in C++ or any .Net Language (C#, VB.Net, etc...) for the creation of highly sophisticated applications involving data acquisition, machine control, sensor monitoring, and even web reporting, just to name a few.

Cheaper than a PC, and more features than an HMI touch screen.
The CUWIN? A Revolutionary Touch Controller
Already in use in thousands of applications throughout the world.

▶ CUWIN5200
▶ CUWIN5500
▶ CUWIN6300

- Samsung 2450 (533MHz) ARM Processor
- 128MB SDRAM / 128MB NAND Flash
- Windows® Embedded CE 6.0
- .Net Compact Framework 3.5
- Software Development: C++ or any .Net Language (C#, VB.Net, etc...)
- Ethernet 10/100Base-TX
- RS-232 x 3, RS-485 x 1
- USB 2.0 x 1, USB 1.1 x 1
- Audio (Line Out) / RTC
- Keyboard and Mouse (optional)
- CE, FCC, and KCC certified

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

by Richard Lord (USA)

Panning Control

A Digital Indexing Panoramic Tripod Head

An electronic panning controller enables a photographer to rotate a camera to provide uniformly overlapping images that can be merged successfully without distortion. Here you learn how to build a digital indexing panoramic tripod head from start to finish.

There has been a great interest in wide-view panoramic photography from the very first recorded images of the 1840s. With the invention of flexible film in the 1880s, several manufacturers began offering specialized panoramic cameras that wrapped the film around a curved surface and pivoted the lens during the exposure. The advent of digital photography and image-processing software has ushered in another way to create panoramic images—by stitching multiple overlapping photos together into a single larger image. While many modern digital cameras and software packages offer tools to make stitching somewhat easier, significant geometric distortions result from attempts to merge handheld images.

Several manufacturers now offer mechanical indexing panoramic adapters of varying degrees of sophistication that allow a camera to be accurately rotated to provide uniformly overlapping images that can be merged successfully without distortion. The simplest of these pan heads merely provide a means to lock the axes of rotation at various angles. More sophisticated versions include indexing plates that make it easier to step between fixed angles. However, the angle required to provide the correct overlap of each step varies with the focal length of the lens, so mechanical indexing can be very awkward to use, often requiring calculation and careful setting of each angle to get a good result. I'm someone who loves to find new ways to use inexpensive microcontrollers to solve problems, so I saw this as a natural and obvious application of technology.

Photo 1—The pan head design

I faithfully attend a couple of electronic flea markets every year and am always on the lookout for inexpensive electromechanical gadgetry. As a result, when I was ready to start this project, I had already amassed a useful collection of stepping motors and drivetrains to choose from. Thus, it was inevitable that as an avid photographer I would be destined to make my own digital indexing pan head (see Photo 1). My goal was to make the design extremely simple to use with no angle calculations—just a simple user interface where I set the start position, the step size (determined by moving the pan head while looking through the viewfinder), and the number of steps needed to reach the final position.

The pan head actually moves to these positions as the parameters are entered, so setting up a pan sequence is very intuitive. I also

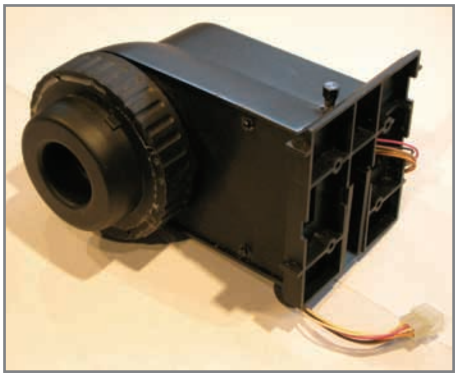


Photo 2—The pan drive motor

added a smooth video panning mode where I set start and end positions and the duration of the pan.

ANOTHER DIMENSION

While rotation about a single pan axis covers many situations, there are obviously times when you might find yourself inside a large interior space where you might want to change the elevation axis and create a view that is several overlapping images high as well as wide. Since I was already developing a controller for the pan axis, it didn't seem to be too big a stretch of the imagination or the electronics to add an elevation axis drive motor as well.

If simple panoramas had been my only interest, I probably would have settled for a simple mechanical pan head. However, I had a bigger goal in mind—much bigger, in fact. As a fairly serious photographer, I find that sometimes I would like to produce poster-sized images. As the size of the printed image increases, the resolution of the camera image must also increase or else the individual image picture elements (pixels) will start to become visible.

Viewing an image from several feet away, the unaided human eye can distinguish about 100 lines per inch of detail in any direction. This means for a high-quality photographic print of a digital image, the printed image must have a minimum of 200 dots per inch (dpi) of resolution to avoid having visible evidence of the individual pixels that make up the image, and more resolution is better. An image captured with a 1-megapixel camera might look fine in a 4″ × 5″ snapshot, but clearly the image quality will suffer when blown up to 16″ × 20″, where the same image is spread over 16 times the area of the 4″ × 5″ print. Based on the 200-dpi minimum, a high-quality printed image takes at least 40,000 pixels per square inch of area. This works out to a maximum of 25 square inches per megapixel, and it means a 12-megapixel camera is not sufficient to produce a quality 16 × 20 print.

Photo 3—A look inside the pan head

To create really sharp 32″ × 40″ and larger photographic prints, you need 50 to 100 megapixels per image. If you have $20,000 to $50,000 of spare change, you can buy large-format digital cameras with this kind of resolution, but for the rest of us, this kind of investment is out of the question. However, using the same image stitching technique that works for forming panoramic photographs from overlapping horizontal images, you can form very high-resolution images from a two-dimensional array of overlapping lower-resolution photographs. By using a telephoto setting and smaller angles for the horizontal and vertical steps, the image that is formed in this way might cover the same field of view as a single lower-resolution photograph, but with a resolution of 100 megapixels instead of the 10 megapixels of the single exposure. Of course, this doesn't work well for action photographs. It may take several minutes to capture all the overlapping images, so the baseball has long ago reached the catcher's mitt and the race car has already completed another lap by the time the digital pan head has gotten halfway through its dance. However, for still life subjects and scenic views, the two-axis digital tripod head provides the opportunity to produce exquisitely detailed large prints with an affordable camera.

SYSTEM OVERVIEW

As you can see from the picture of the full two-axis pan head in Photo 1, this project includes some mechanical construction for the elevation drive mechanism. I am fortunate enough to have recently acquired a small lathe/milling machine setup, so this project has been an opportunity to hone some mechanical skills. However, the basic pan mechanism was built with hand tools and existing materials.

One of the stepping motor mechanisms in my collection of flea-market bargains was a roll-paper spindle drive mechanism, shown in Photo 2, that had once been part of large-format plotter. The drive included a nice bearing assembly for the 0.375″ spindle shaft and a 5:1 gear reduction to drive this shaft from a six-lead 1.8° stepping motor. This was all mounted in an enclosure that was designed to clamp onto the back of the plotter. By orienting it so the spindle shaft was pointing straight up, I had the essential mechanism of the pan drive without any extra effort. I merely had to remove the original clamping mechanism for the plotter and replace it with a plastic box containing the controller electronics. Then I had to figure out how to mount my camera on top of the convenient turntable that had once been the spool for the paper roll when it was

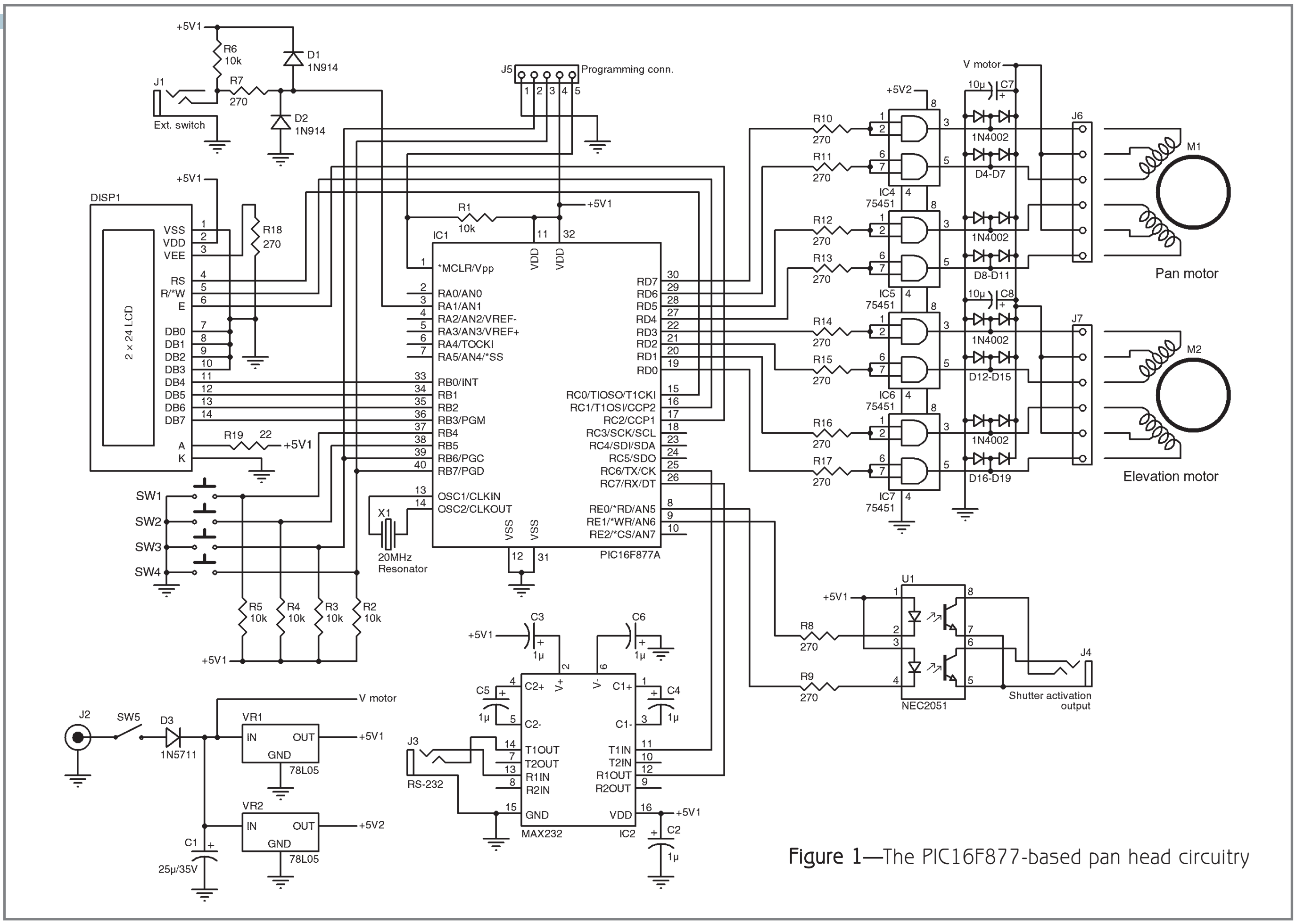


Figure 1—The PIC16F877-based pan head circuitry

part of the plotter. The guts of the pan head and controller can be seen in Photo 3.

For the elevation drive, I came up with a slightly smaller six-lead stepping motor with an attached 4:1 belt-drive reduction driving a 0.5″ diameter 0.20″ pitch sprocket. To complete the vertical drive, I had to buy a timing belt and a 3.25″ sprocket from an online company, Stock Drive Products, for a total of $10. A friend supplied me with a 3″ bearing and some scrap aluminum, which I was able to use to make the rest of the elevation drive. The elevation drive plugs into the controller with a five-pin DIN connector so that the digital pan head can be used with other camera mounts if the elevation feature isn't needed.

CIRCUITRY

Stepping motors come in several flavors of winding configurations. There are many exotic ICs designed specifically for driving stepping motors, but they come in high-density surface-mount packages that are hard to work with for home-brew projects.

Having the unipolar six-lead version for both motors greatly simplified the design of the drive electronics. These motors have two center-tapped coils and can be driven by tying the center tap to the power source and then grounding either end of the winding to provide either polarity of magnetic field. I just happened to have a whole drawer full of 75451 peripheral drivers in my parts bin. These parts have been around for a long time, but Digi-Key still sells them for $0.83 each. This eight-pin DIP provides dual high-voltage open-collector outputs, each capable of sinking 350 mA, making them ideal for driving these smaller unipolar stepping motors.

Each 75451 drives the two ends of a single winding, so only one of the two drivers is active at a time, reducing heat dissipation. As you can see in Figure 1, there are two 75451 drivers for each motor. I added 1N4002 clamping diodes to ground and to the power rail for each output to prevent voltage spikes from feeding back into the circuitry and damaging either the driver or the controller (a lesson learned the hard way in an earlier design that destroyed the microcontroller). The diodes are mounted in close proximity to the outputs and are bussed together to nearby bypass and bulk capacitors so the energy is closely contained and doesn't wander around the circuit board where it could introduce either voltage or current spikes into the rest of the circuitry. To be even more secure, I added series resistors between the driver inputs and the microcontroller and also provided the peripheral drivers with their own 78L05 regulator, separate from the power for the microcontroller.

The rest of the circuitry for this controller is pretty simple. At its heart is a Microchip Technology PIC16F877A with a 20-MHz resonator connected between the clock pins. In addition to the four outputs for each motor, the controller connects to a 2 × 24 character LCD and four push button switches. I added two optically isolated outputs to switch the two-stage shutter input on my camera, an input for an external remote switch for single-stepping the pan head, and an RS-232 port to allow control of the digital pan head from a computer. The controller board is shown in Photo 4.

My method of board construction is to buy the prototyping boards at RadioShack that have holes on 0.10″ centers with pad-per-hole copper on one side. These are relatively cheap and very easy to use. Wherever possible, I try to use integrated circuits in DIP packages inserted into high-quality machined pin sockets. I first wire the grounds by bending the leads from the bypass capacitors to run along a row of pads on the solder side and then beef the wire up by soldering it to each pad and flowing the solder along the wire. This forms a sort of ground plane as can be seen in Photo 5. I then wire in the power, either using the same technique or by using scrap #26 wire from ribbon cable. Finally, I wire the signals by soldering short pieces of #30 wire-wrap wire that are cut to length. This technique lets me adapt and change wiring fairly easily. Wherever possible, I use 0805-size surface-mount resistors which fit nicely between tenth-inch pads on the solder side. This construction method is relatively inexpensive, so it is not a big deal to start all over again if there's a better arrangement of the parts.

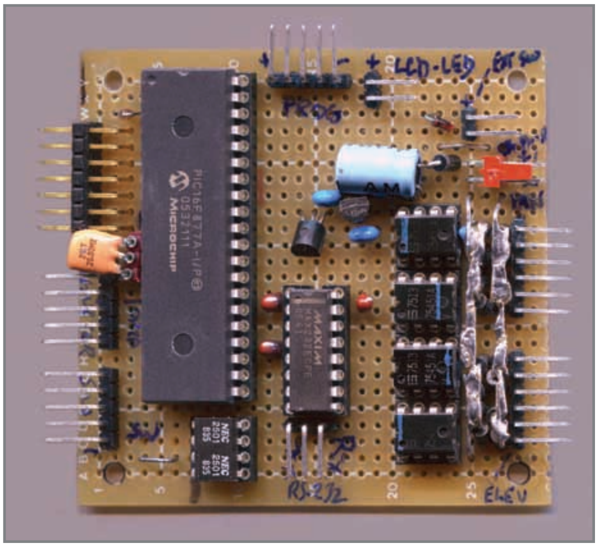


Photo 4—The controller board

Somewhere on the board, I always add a small loop of 18-gauge solid wire connected to ground to clip on the ground lead of an oscilloscope for debugging. When developing the software, I often find it useful to temporarily add code to set or clear an output. For example, to determine the duration of the interrupt handler, I temporarily set the shutter output pin at the beginning and clear it at the end of the interrupt routine.

MAKING IT GO

I love programming PIC controllers in assembly language. I find it an enjoyable challenge to work directly within the constraints of the hardware. Assembly language is an efficient way to squeeze the most performance out of inexpensive 8-bit microcontrollers with limited stack and code space.

For the digital pan head, I wanted to create an intuitive user interface. After playing on paper with several alternatives, I decided that I could provide all the features that I wanted with a 2 × 24 character LCD and four push buttons mounted directly below the display. The function that each button performs is defined using the six characters of the bottom line of the display that are directly above that button. The button labels change as the context changes. The upper line of the display is used to define the current context or "screen." For "screens" that are used to adjust the value of a parameter, the upper line shows the parameter name and value. The two middle buttons are used to decrease or increase the value of the parameter. When adjusting parameter values, the buttons are set up to "auto repeat" so that large changes are easy to program. Each time you press the button, the parameter is initially changed by a single count, but holding the button down causes the change to repeat automatically, slowly at first and then more rapidly if the button continues to be pressed. The rightmost button is used to "Enter" that value and advance to the next menu "screen." The leftmost button enables you to go back to the previous menu screen. For those parameters that affect the position of the pan head, the head actually moves as the value is changed. For the pan axis, the middle buttons are labeled with left and right arrows corresponding to the rotation of the camera toward the left (counterclockwise) or the right (clockwise). For the elevation axis and for non-motion parameters, the labels change to "up" and "down" arrows.

To enter a new pan axis motion "program" into the digital pan head, press the "Prog" button and then select "1-Axis." The upper line then displays "Pan Start Pos=+0000" representing straight ahead. (If you are editing an existing program, the display will show the current start position and the pan drive will move to that position.) The left and right arrow buttons can then be used to move the pan head to the initial position of the pan sequence. Once you press the "Enter" button, you're then prompted to define the "Pan Step Size." The pan head is positioned while looking through the camera viewfinder until the desired image overlap is achieved.

Next, you program the "Pan Step Count," advancing the head to its final position by moving the pan head a step size per count. Signed arithmetic is used to make all the calculations, so the pan head can be programmed to step in either direction from any start position.

Finally, you're prompted to enter the "hold" time that the pan head will wait after pressing the shutter before stepping to the next position (basically, how much time it takes the camera to write the image to the memory card). The position "program" can then be saved in one of 10 12-byte EEPROM blocks for later recall, or it can be run directly.

For a two-axis program the elevation start, step size, and count are programmed first, followed by the pan parameters. Running the program

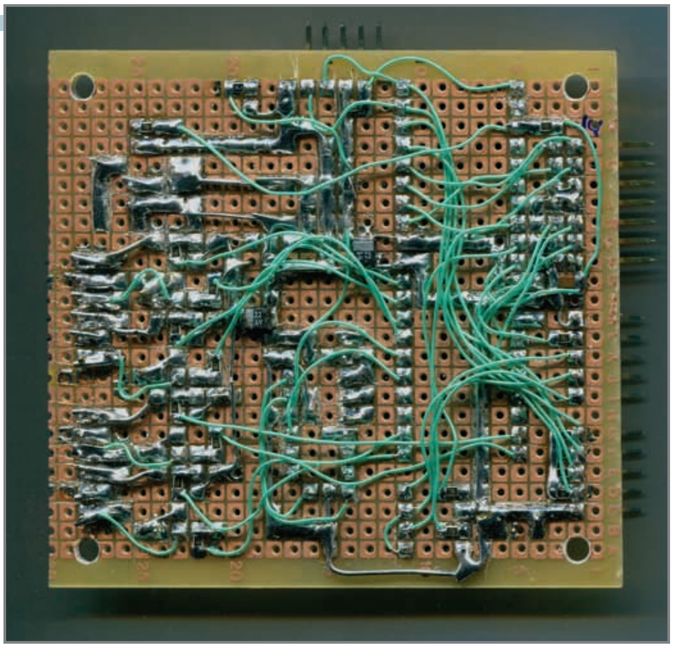


Photo 5—A look under the controller

causes the head to turn to the start position, wait for a settling time, actuate the shutter, wait for the hold time, and then step to the next position. For two-axis programs, the pan sequence is repeated for each elevation position.

Since there is significant mass to the pan head, the motor control software includes the ability to accelerate from a slower speed at the start of the motion and to decelerate as the pan head nears the target position. The settling time, duration of the shutter activation, and acceleration, deceleration, and slew rates of the motor axes are controlled by general system parameters stored in EEPROM that can be programmed from the setup menu that is reached by holding down the leftmost and rightmost buttons during power-up.

The stepping motors are driven by table look up, indexed by the current position counter for each axis. Initially, I chose the common half-step method, where a single winding is energized for the intermediate half step between the full-step positions (where two windings are powered) producing eight half steps per motion cycle or 0.9° of rotation of the motor shaft between entries in the look-up table. When I decided to add a slow continuous pan feature for shooting video, I found that half-stepping produced a jerky motion.

More precise control of stepping motors can be achieved by a technique called "microstepping" in which the amount of power sent to each winding is modulated to achieve intermediate positions. Using sine look-up tables, it is possible to achieve a great many intermediate positions. However, for less precise requirements, it is often sufficient to simply provide several intermediate power values for each winding. For my initial experiments, I decided to use the microcontroller's interrupt timer to pulse-width modulate the power at 0%, 25%, 50%, 75%, and full on, creating 32 steps per stepping cycle. This gave very smooth motion and precise control; but with the microcontroller running at its maximum of 20 MHz and the interrupt handler pared to a minimum amount of code, the best I could achieve for the PWM rate was about 8 kHz, causing the stepping motors to audibly "sing" at the intermediate power positions. I finally decided to eliminate the 25% and 75% steps, and I was able to up the 50% PWM rate to around 18 kHz, which put it above my hearing.

The power output for the motors is controlled in the interrupt handler using two 8-bit "mask" registers, where each bit corresponds to an output to a motor winding. The upper 4 bits correspond to the negative and positive outputs for both windings of the pan axis motor. The lower 4 bits correspond to the outputs for the elevation axis. These bit masks are set by the foreground software from table look up from the stepping table (see Table 1) using the 4 LSBs of the 16-bit motor position register as an index into the table. At each interrupt, the contents of the 100% mask register are loaded into the accumulator. The interrupt handler increments a counter. If the LSB is high, a logical OR is performed with the contents of the 50% mask register. The contents of the accumulator are then written to the motor output port.

If a bit is set in the 100% mask, then that motor output is turned on all the time. If a bit is not set in the 100% mask but is set in the 50% mask, then the corresponding

Four-step	Eight-step	Sixteen-step	Motor windings		Bit mask nibbles	
1.8°	0.9°	0.45°	B-leads	A-leads	100%	50%
					-b+b-a+a	-b+b-a+a
	step 00	step 00	100%	OFF	0100	0000
		step 01	100%	50%	0100	0001
step 00	step 01	step 02	100%	100%	0101	0000
		step 03	50%	100%	0001	0100
	step 02	step 04	OFF	100%	0001	0000
		step 05	-50%	100%	0001	1000
step 01	step 03	step 06	-100%	100%	1001	0000
		step 07	-100%	50%	1000	0001
	step 04	step 08	-100%	OFF	1000	0000
		step 09	-100%	-50%	1000	0010
step 02	step 05	step 0A	-100%	-100%	1010	0000
		step 0B	-50%	-100%	0010	1000
	step 06	step 0C	OFF	-100%	0010	0000
		step 0D	50%	-100%	0010	0100
step 03	step 07	step 0E	100%	-100%	0110	0000
		step 0F	100%	-50%	0100	0010

Table 1—The stepping motor sequencing chart. Note the following: Motor output = 100% mask "OR" (50% mask "AND" INTERRUPT_CTR[0]). Pan motor = upper nibble [7:4]. Elevation motor = lower nibble [3:0].

(9600 Baud, 8 bit, no parity)				
Cmd	**arg**	**Function**	**Response**	**Return format**
v	–	Display current software version	Returns current software version + prompt	[text string]
dp	–	Display current program	Returns current program values + prompt	[decimal table]
ge	–	Get elevation position	Returns current Elevation position + prompt	[4-digit signed decimal]
gp	–	Get pan position	Returns current Pan position + prompt	[4-digit signed decimal]
sh	–	Activate shutter	Returns prompt after completion	
gs	–	Get current system state	Returns current display page + prompt	[2-digit decimal]
sw	0–4	Press key 1–4 (sw0 releases key)	Returns prompt	
db	–	Get all system variables (debug)	Returns 192 system variables + prompt	[hexadecimal table]
kl	–	Lock (disable) front panel keypad	Returns prompt	
ku	–	Unlock front panel keypad	Returns prompt	
The following motion commands require that the keypad be locked/disabled. They return the command prompt once the motion is completed.				
ea	–9999 to 9999	Move to absolute elevation position		
er	–9999 to 9999	Relative move of elevation position		
pa	–9999 to 9999	Set absolute pan position		
pr	–9999 to 9999	Relative move of pan position		

Table 2—Commands that can be sent to the digital pan head through the RS-232 port

winding is powered on alternate interrupts, thus being energized with 50% of the power. Each time the motor position register is incremented or decremented for either axis, a new pair of bit masks is generated.

The control scheme yields 0.45° motor steps with 16 steps per motion cycle. With the 5:1 gear reduction on the pan axis and 24:1 reduction on the elevation axis, this is sufficient to produce satisfactory results without the accompanying howling of the motors that was caused when I had attempted to use the 32-step table.

The interrupt for the motor PWM also provides the timing for the button repeat function, slew rates for the stepping motors, and millisecond counts for the various time settings. This timing is accomplished by decrementing a counter programmed with the number of interrupts per millisecond. Once this counter reaches zero it is reloaded and a separate "millisecond delay" counter is tested. If it is nonzero, this counter is decremented once for each millisecond. When the main software needs to create a delay, it sets this delay counter with the desired number of milliseconds and then tests its value and loops until the value is zero. A similar delay counter decrements in tenths of a second for the longer delays.

I also provided an RS-232 port that can activate the shutter, read the current positions of the two stepping motors, read the current "screen," and remotely press any of the buttons. I also included the ability to specify the absolute or relative position of either motor. With this latter capability, a computer can be used to drive the pan head in complex two-axis motion (see Table 2).

DESIGN EXPLORATION

I designed the digital pan head to work with a specific digital camera (a Canon Digital Rebel) and centered both axes of rotation on the plane of the image sensor. I have subsequently discovered that this is the wrong location to eliminate parallax errors. The rotation needs to occur at the plane of the convergence of light rays in the lens, which varies with each lens or zoom setting. As a consequence, I need to modify the camera mount with a slide that will allow the camera to be moved forward or backward and then clamped in the position that will properly place the center of rotation. I also plan to add adjustments to compensate for different lens heights above the camera base and different locations of the tripod socket with respect to the lens, so that the digital pan head can be used with other cameras.

I haven't yet explored the ability to create complex motion using the positioning commands through the RS-232 port, but this can open up the possibility to do some very creative things such as painting geometric shapes with lights during night photography. I'm just starting to explore all the capabilities of the digital pan head, but it's already proving to be a very versatile tool for a very modest investment of less than $100. It has also been a fun project to build.

Richard Lord (rhlord@comcast.net) holds a B.S. in Electrical Engineering and an M.S. in Biomedical Engineering. During his career, he has designed digital electronics for an aerospace company and several telecommunication test equipment manufacturers. Working as a consultant in the 1980s, Richard designed several medical pulmonary test instruments and the electronics for an autonomous underwater robot. His interests include digital electronics, photography, jazz, and river conservation.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/248.

SOURCES

PIC16F877 Microcontroller
Microchip Technology, Inc. | www.microchip.com

Elektor Shop

The world of electronics at your fingertips!

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

See the light on Solid State Lighting

DVD LED Toolbox

This DVD-ROM contains carefully-sorted comprehensive technical documentation about and around LEDs. For standard models, and for a selection of LED modules, this Toolbox gathers together data sheets from all the manufacturers, application notes, design guides, white papers and so on. It offers several hundred drivers for powering and controlling LEDs in different configurations, along with ready-to-use modules (power supply units, DMX controllers, dimmers, etc.). In addition to optical systems, light detectors, hardware, etc., this DVD also addresses the main shortcoming of power LEDs: heating. This DVD contains more than 100 Elektor articles on the subject of LEDs.

ISBN 978-90-5381-245-7 • $46.00

Software Tools & Hardware Tips

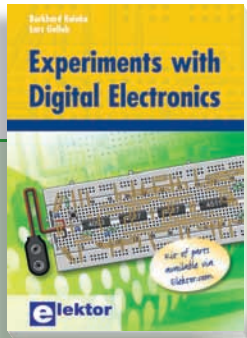
CD Ethernet Toolbox

This CD-ROM contains all essential information regarding Ethernet interfaces. It includes a collection of datasheets for dedicated Ethernet interface ICs from many different manufacturers. What's more, we have compiled a collection of all articles on this topic that have appeared in Elektor magazine and complemented them with additional documentation and links to introductory articles on Ethernet interfaces.

ISBN 978-90-5381-214-3 • $31.50

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

Books



An introduction to digital control electronics

Experiments with Digital Electronics

This book presents fundamental circuits using gates, flip-flops and counters from the CMOS 4000 Series. Learning these fundamentals is best done using practical experiments. Each of the 50 experiments presented in this book has a circuit diagram as well as a detailed illustration of the circuit's construction on solderless breadboard. Building these digital circuits will improve your knowledge and will be fun to boot. Many of the circuits have practical real-life applications.

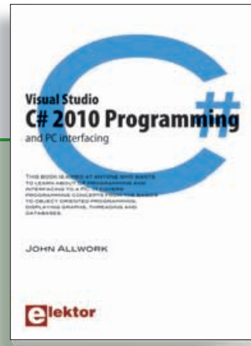
176 pages • ISBN 978-0-905705-97-2 • $42.80

Associated starter kit available

ARM Microcontrollers

This is the perfect book for people who want to learn C and who want to use an mbed ARM microcontroller in an easy and fun way. The mbed NXP LPC1768 uses cloud technology, a revolutionary concept in software development. This means you do not need to in-stall software on your PC in order to programthe mbed! The only thing you need is a brow-ser such as Microsoft Internet Explorer, and a USB port on your PC. No previous expe-rience or knowledge required. You can get access to your project from any PC anywhere in the world and continue working on it. When you are done a few mouse clicks transfer the program to your mbed hardware.

250 pages • ISBN 978-0-905705-94-1 • $47.60



Visual Studio

C# 2010 Programming and PC interfacing

This book is aimed at anyone who wants to learn about C# programming and interfacing to a PC. It covers programming concepts from the basics to object oriented programming, displaying graphs, threading and databases. The book is complete with many full program examples, self assessment exercises and links to supporting videos. All code examples used are available – free of charge – from a special support website. Professional quality software tools are downloadable –also free of charge– from Microsoft.

305 pages • ISBN 978-0-905705-95-8 • $47.60

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

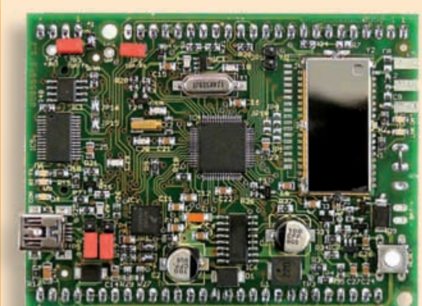
Elektor US
PO Box 180
Vernon, CT 06066
USA
Phone: 860-875-2199
Fax: 860-871-0411
E-mail: sales@elektor.com

NetWorker

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

Module, ready assembled and tested

Art.# 100552-91 • $85.50



Reign with the Sceptre

This open-source & open-hardware project aims to be more than just a little board with a big microcontroller and a few useful peripherals — it seeks to be a 32-bit ARM7 fast prototyping system. To justify this title, in addition to a very useful little board, we also need user-friendly development tools and libraries that allow fast implementation of the board's peripherals. Ambitious? Maybe, but nothing should deter you from becoming Master of Embedded Systems Universe with the help of the Elektor Sceptre.

PCB, populated and tested,
test software loaded

Art.# 090559-91 • $143.60

Kits & Modules

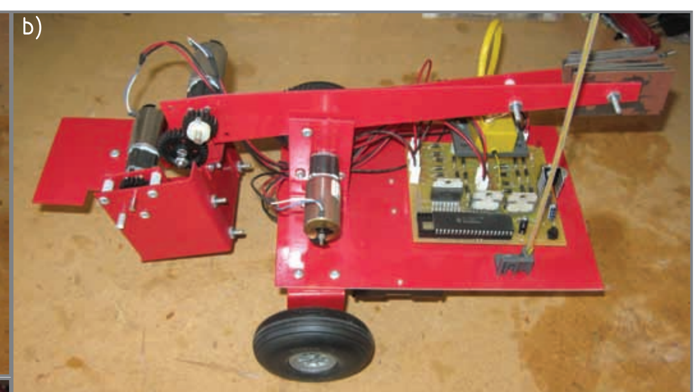
by Brian Senese (USA)

Remotely Operated Vehicle Design

You can control a remotely operated vehicle (ROV) or mobile robot wirelessly with a basic RC transceiver and a little know-how. This design uses FM signals to transmit data. A receiver on the ROV then demodulates the FM signal and enables motor control.

After enrolling two of my children in the "science olympics" at their school, I found that I also had a role to play as a volunteer coach for a "robo-cross" event. Under the direction of the coach, each student is challenged to build a robot for a contest involving picking up items and then placing them into containers for points.

To prepare for the event, I visited the previous year's coach to get some helpful tips. (While I met with him I discovered that he had built "battlebots" in his machine shop. These were serious industrial-strength robots that could fight in an arena.) When he showed me the winning robot from the prior robo-cross event, I knew I was in trouble. To make matters worse, I found that it took a small fortune to buy commercially available robot electronics such



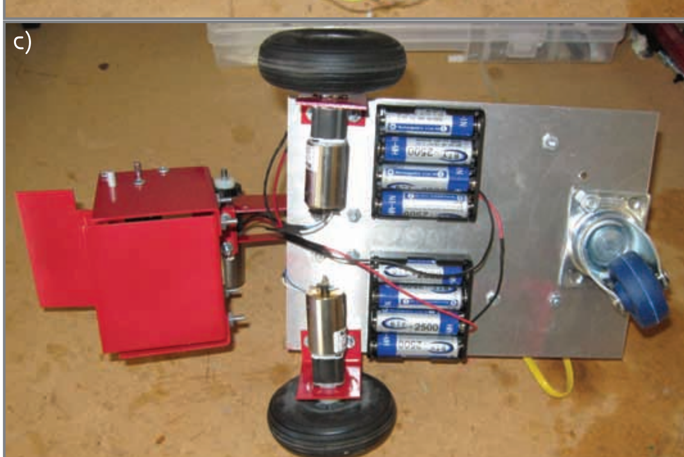


Photo 1a—The prototype ROV competed in a "robo-cross" event and placed sixth overall in a field of more than 40 contenders. b—This second-generation "cousin" design incorporates significant improvements, such as standardizing one motor, improving the gearing mechanisms, and, above all, simplifying the electronics. c—Eight AA cell batteries are held under the ROV.

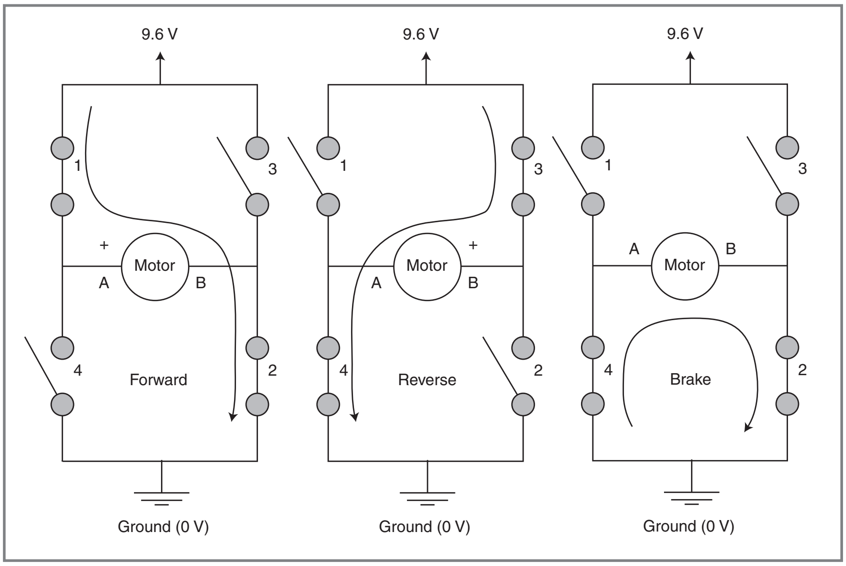


Figure 1—DC motor control is accomplished through an H-bridge circuit where current can be passed through the motor in one of two directions acting to move the motor either in a forward or reverse direction. Power FETs typically act as switches to accommodate the high current requirements. Two control lines determine the direction in which the current is passed through the motor.

as a radio control (RC) transmitter (more than $100), motor controller (five at $30 each), gearhead DC motors (five at $20 each), and metal components. Kids love robotics, but at $400 a pop, such dreams are shattered.

There is a less expensive way to work with robotics, and that is to make a robot from scratch, using surplus electronics and some smart engineering. Much can be learned when creating your own robot and I mean this in the global sense: electronics, software, mechanical and materials engineering to be precise.

In this article I describe the design of a second-generation remotely operated vehicle (ROV) based on a prototype that was conceptualized and built by two students. I'll focus on the underlying electronics and software, which you can easily extend to any mechanical contraption. The beauty of this project is that the entire ROV was about $150 with spare material left over.

PROTOTYPE DESIGN

Photo 1 shows two ROVs, the prototype as well as the second-generation version, which should be easy to identify. The chassis sits atop two drive wheels that are controlled with two independent motors. A third swivel wheel provides maneuverability; it also reduces the amount of force required in turning the robot if the design were to use two statically mounted wheels instead. Eight AA NiCad cells are mounted on the bottom of the chassis providing 9.6 V of power.

Mounted on top of the chassis are the electronics, mechanical arm, bucket, and flap. The bucket is positioned to the side of the ROV so that it can scoop up objects that are in tight corners. And, with only three wheels, balancing the robot, especially when it picks up golf balls, can be a problem. Counterweights on the arm opposite the bucket are necessary to reduce the amount of power required by the motor in lifting any load.

You can tilt the bucket to hold objects inside once they are loaded. The flap is used to coax smaller objects into the bucket (e.g., small coins) as well as hold items in place while the robot is moving. All of the mechanical parts (except wheels) are attached to separate motors using gears. You'll note that the gearing is different for each of the three moving elements. Gearing determines both the speed at which the part moves and the amount of force that can be applied in moving that piece.

An inexpensive, commercially available VEX transmitter/receiver—which is generally marketed as a surplus item on the internet—provides radio control. The trick in using this hardware is in deciphering the receiver output and then using this information to control the five motors. This is accomplished with a Microchip Technology PIC18F4620. Additional circuitry is required to handle the current drawn in driving each motor as well as protect the controlling processor from noise spikes created when motors are turned on and off.

The ROV electronics are responsible for receiving a radio signal, demodulating and decoding it, and then controlling the five motors onboard.

The ROV electronics are responsible for receiving a radio signal, demodulating and decoding it, and then controlling the five motors onboard. This design is atypical since it does not follow the conventional motor control methods used in commercial devices. First of all, the hardware and software is designed to directly control DC motors that have a maximum current draw of 2 A. Additionally, the signal scheme used by the VEX transmitter/receiver pair is nonstandard and it cannot be used with anything other than the VEX robot controller, which limits its use.

The project files (software, a schematic, and a mechanical template) are available on the *Circuit Cellar* FTP site. You can use the files for a design capable of picking up objects ranging from small coins to golf balls and moving across obstacles as high as 0.25″ in height.

MOTOR SELECTION

Motors are the most critical component in an ROV or robot. Speed and torque govern ROV operation and are inversely related. In general, the faster a motor runs, the lower its output torque.

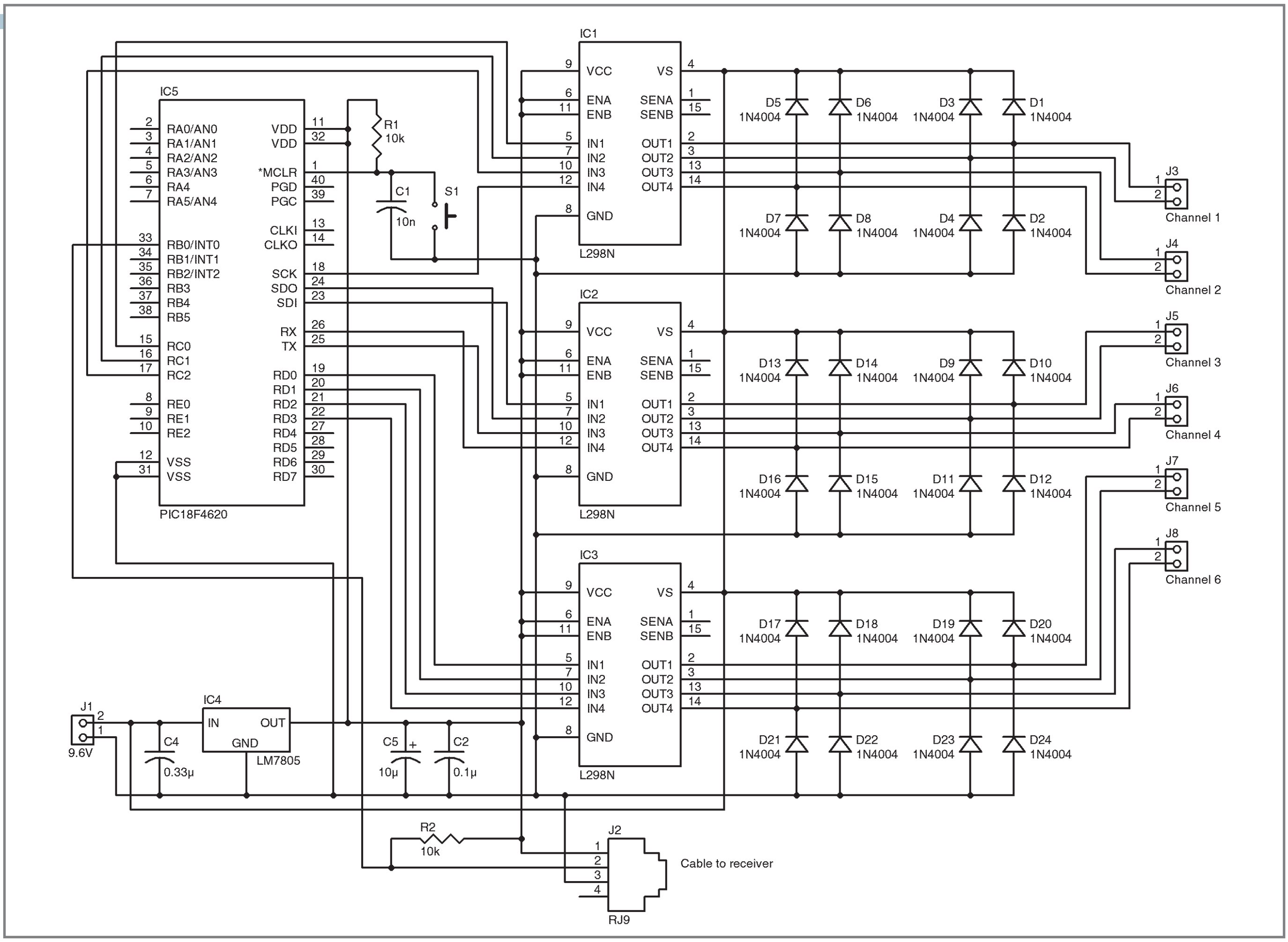


Figure 2—The complete control system. Control signals originate from the transmitter, are demodulated by the receiver, processed by the microcontroller, and eventually used to control five motors.

Another consideration is the nominal voltage that is required by the motor, which in turn determines the batteries required. For common DC motors, a gear head is usually attached to reduce motor RPM and increase output torque to a point making the motor useful. Let's look at how a motor is selected. Fortunately, it requires only simple math to arrive at an approximation. Taking a more rigorous approach would include the transient effects of motor inductance, yet this is not necessary for our purposes.

In this ROV, two Namiki DC motors with RPM of 70 at 6 V (50 mA) were used to drive each 3″ wheel. RPM for different voltage inputs can be approximated by using 70 RPM per 6 V, or 11.7 RPM per volt. Power is supplied by 8 AA batteries, providing a maximum of 9.6 V. If maximum power is applied to the drive motors, they will turn at 112 RPM (i.e., 9.6 V × 11.7 RPM/V). With a 3″ wheel attached to the motor spindle, the ROV has a maximum traveling speed of:

$$\left(3' \times \pi\right) \times 112 \text{ RPM} \times \frac{1 \text{ min}}{60 \text{ seconds}} = \frac{17''}{\text{second}}$$

This is sufficient to maintain control within the confines of tight spaces.

The torque available from both motors will determine if they are strong enough to move a fully loaded metal ROV across a floor, over small objects, or up an incline. Nominal torque first must be calculated at a specific motor speed using the manufacturer's specifications and an internal motor resistance measurement. For this project, it came out to be 6 Ω in using a standard ohmmeter.

$$\text{Torque} = \frac{\text{Electrical power - Internal power loss}}{\text{Rotational speed (rad/s)}} =$$

$$\frac{\left(0.05 \text{ A} \times 6 \text{ V}\right) - \left(0.05 \text{ A} \times 0.05 \text{ A} \times 6\ \Omega\right)}{\frac{2\pi \times 70 \text{ RPM}}{60 \text{ s/min}}} = 0.03888 \text{ N-m}$$

There are two problems with this calculation. First, the motor starts at 0 RPM, not 70 RPM as is specified for this level of torque. Secondly, to achieve maximum power, 9.6 V must be applied to the motor. To get an accurate torque estimate, you must use a variable called a "motor constant" (Km), which relates motor torque to input current. Note that Km = Torque (N-m)/input current (A) = 0.038 N-m/0.05 A = 0.76. You can use Km to give a torque reading when the motor is at a stall (0 RPM) and when you apply maximum

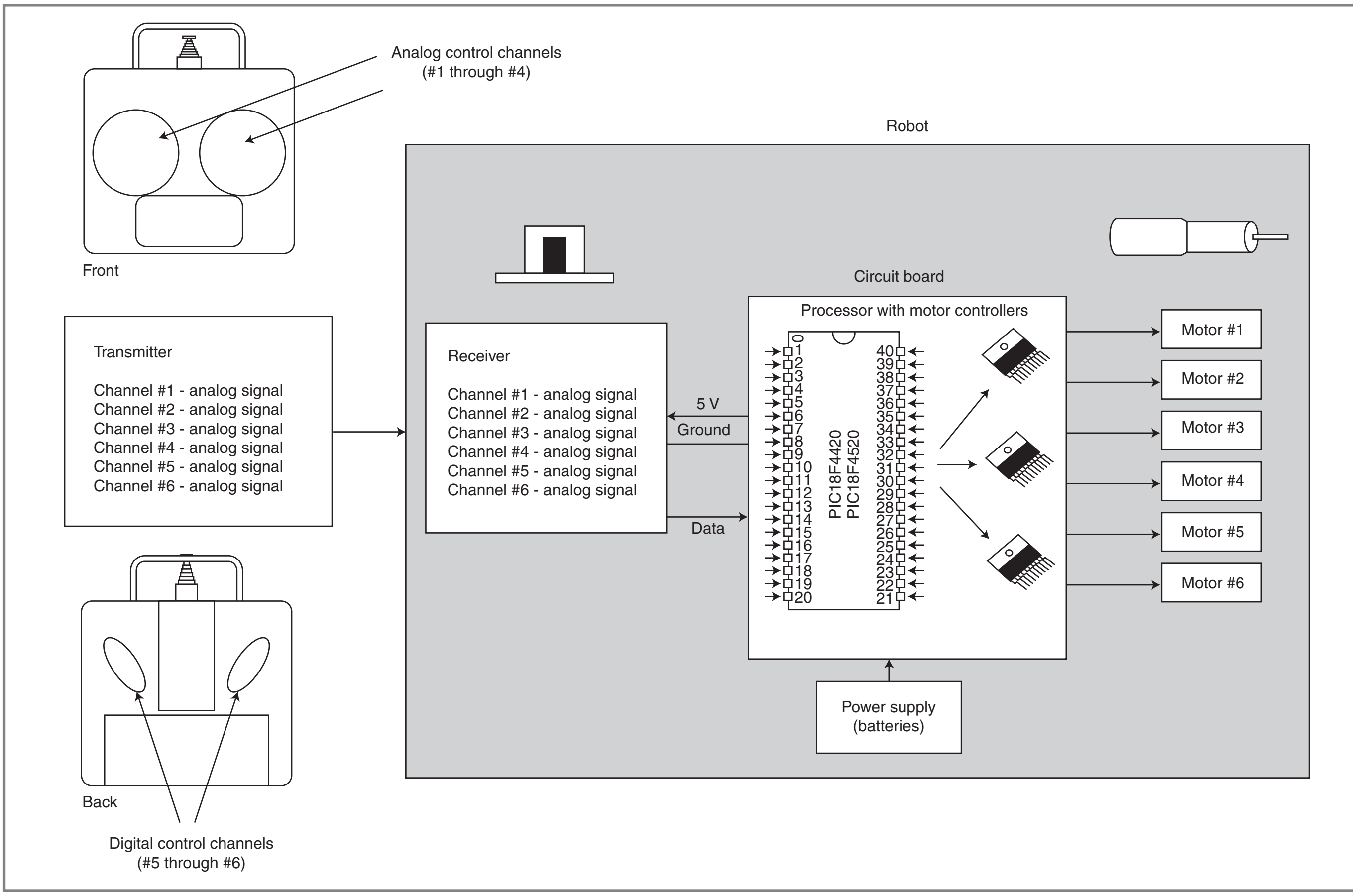


Figure 3—The design features a PIC18F4620, which is used to process the incoming datastream originating from the open-collector receiver. An interrupt-driven input pin deciphers the serial PPM stream and responds by manipulating two output control signals (per channel) which are fed to an H-bridge circuit. Motors are connected directly to the H-bridge and diodes act to protect the circuitry from transient spikes generated by each motor.

voltage. You first estimate the amount of current provided to the motor when it is at rest: 1 A (i.e., 9.6 V/6 Ω). Using Km, you can calculate stall torque for the motor as 0.76 N-m (i.e., 0.76/1 A).

Now add the effects of the 3″ wheel, which will convert the torque into a forward driving force. This force can be determined as:

$$\frac{\text{Torque}}{\text{Distance from motor spindle}} = \frac{0.76\text{ N-m}}{1.5'' \times \frac{0.0254\text{ m}}{1''}} = 20\text{ Newtons}$$

With two motors working in tandem, 40 N of force can be brought to bear in moving the ROV.

Resistance to moving across a flat surface originates from the drive wheels. This force can be found to be 0.6 N per wheel, which is negligible. For an ROV with a mass of 3.5 kg, 40 N of force will yield an acceleration of 11 meters per second with the ROV reaching maximum speed in 34 ms. Needless to say, the motors selected are more than adequate in terms of moving the robot quickly and easily.

Refer to Photo 1. Note that opposite from the drive wheels a single swivel wheel is used to allow the robot to turn easily and provide greater maneuverability. The drive wheels are powered independently and can be controlled to move in forward or reverse.

There are three other motors of the same type onboard to lift the arm, tilt the bucket, and close the flap. The use of gears makes mechanical control easier. In the case of the arm, the gear ratio is 10 (motor pinion) to 50 (gear attached to arm). This reduces the rotational speed of the motor by a factor of five and increases torque by a factor of five as well. Gearing is an essential design consideration since it offers a way through which fine tuning can be applied to mechanical control. Without gearing, a direct connection between the arm and motor would result in an arm that rotates at 360 degrees in 1 second (using 70 RPM at 6 V).

MOTOR CONTROL

Regarding motor control, the direction of spindle rotation is determined by the polarity of the voltage applied to the two input leads. Speed is managed by varying input voltage. This is best achieved using PWM and varying duty cycle between 0% and 100%. Not only is this easier to implement, but it is also offers greater power efficiency since it eliminates the need for a power absorbing variable resistor. When placed in series with a motor, such a resistor would act to control the voltage applied to the motor. Current through the resistance results in power loss.

DC motors are often controlled with an H-bridge (see Figure 1). As I mentioned, a motor can draw 1 A when in a stalled state. To accommodate this amount of current, power FETs are generally used as electronic switches in the H-bridge with the added benefit that there is a negligible voltage drop between drain and source when the device is in an "on" state. Almost all available battery voltage is applied to the motor in such a configuration. The first-generation ROV made use of discrete FETs in forming an H-bridge for each individual motor used. You can see in Photo 1 what a rat's nest that creates. The second-generation ROV was designed to use a dual full-bridge driver (L298N), an older inexpensive IC capable of handling up to 2 A. A drawback to using this solution is that there is a small voltage drop across the internal switches that reduces available power to the motor. This trade-off was acceptable as it reduced the number of components as well as cost.

Two control lines are required per single motor. Setting one of these lines high while holding the other line low allows current to flow through the motor in one direction.

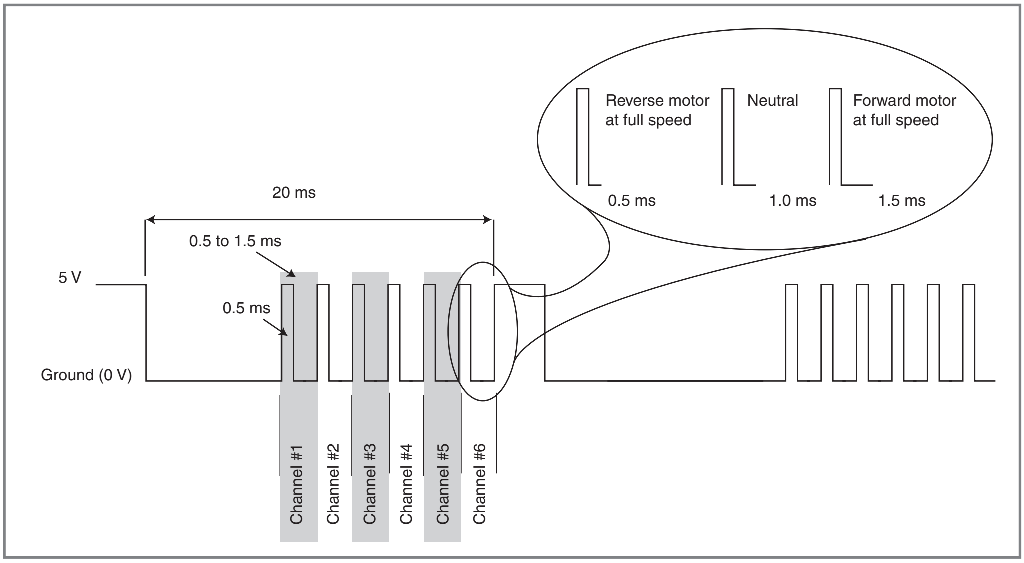


Figure 4—A VEX receiver demodulates information from the transmitter which is in pulse position modulation format. A LO sync pulse that is 9 ms in duration signals the beginning of channel data. Six channels are encoded, four of which are analog and two are digital. A pulse trough of 0.5 ms indicates full speed reverse where 1.5 ms indicates full speed ahead. An idle channel has a pulse trough of 1 ms.

By reversing the control line input, current flows through the motor in the opposite direction. Introduce to this PWM and you have complete motor control and can vary speed as well as direction. The PIC18F4620 allocates two I/O lines per motor. Figure 2 illustrates the interface between the processor, an H-bridge IC (L298N), and each of the five motors. Figure 3 shows the relationship between the circuit board, transmitter, receiver, and power supply. You also see the channels and motors.

Note the extensive use of diodes placed around each motor. This is a protective measure to prevent the L298N from being fried. It also serves to limit the amount of noise generated as a result of the motors being turned on and off. DC motors act like large inductors. When suddenly turning off a motor, current continues to flow through the internal windings. When met with large resistance (controlling circuit being high impedance in an off state), a large voltage spike is created. Diodes limit this spike, yet in doing so introduces a different problem. Diodes are tied to the power supply lines. When a spike is being dissipated, the instantaneous current draw affects the power line voltage.

With a noisy power supply comes unpredictable processor operation and erratic motor operation. Capacitors are added to the power supply lines to reduce such noise spikes to the point where processor instability is eliminated.

WIRELESS CONTROL

RC vehicles use readily available commercial short-range transmitters. The VEX transmitter/receiver is an example. The controller provides two joysticks that manipulate four analog channels (forward/reverse with speed control) and two sets of push buttons that provide two digital channels (foward/reverse with full speed only). This unit was meant to be used with existing VEX robot hardware and therefore uses a funky encoding scheme for all six channels and then transmits this signal. The receiver provides a signal that has to be decoded and processed, providing the control signals to five of the ROV's motors. One channel (digital) remains unoccupied. This solution is a fraction of the cost of others that are commercially available.

Let's now consider decoding this stream, which makes use of the PIC18F4620. All six control channels originating from the transmitter and demodulated by the receiver are first separated and then further processed to create a set of control signals that are eventually provided for each individual motor. The receiver has an open-collector output, a good feature allowing it to operate with different power supply voltages and making it easy to interface to our processor. This output is formatted as a pulse position modulation (PPM) datastream format (see Figure 4), which is fed into an I/O pin on the PIC18F4620 that supports interrupts. Initially, this pin is polled in order to find the low-going synchronization pulse that lasts approximately 9 ms and is used to identify channel positioning. Each of the four analog control channels (channels 1 through 4) is represented by an 0.5-ms high pulse followed by a variable low-going trough that can vary anywhere from 0.5 to 1.5 ms in length, depending on the position of the analog controller at the transmitter. The two digital control channels (channels 5 and 6) create a trough that is either 0.5 or 1.5 ms in duration. Trough duration determines both speed and direction of the motor associated with that channel.

For a motor in an idle state, the trough duration is 1 ms. For full-speed forward, the trough duration is 1.5 ms. Full speed reverse creates a trough that is 0.5 ms long.

The PIC18F4620 is programmed to identify the initial synch pulse by polling the incoming signal while in a continuous loop. When it reads a low signal on I/O pin #33 (RB0) for 5 ms, the channel counter is reset (set to 1), indicating that the first channel soon will be available for decoding. The interrupt on this input is then enabled and programmed to be triggered on a rising edge. When the pulse goes high, the first channel has arrived and is ready to be decoded. Within the interrupt, an internal timer is reset to 0 (marking the start of the first channel) and the interrupt is then re-enabled to be triggered on the

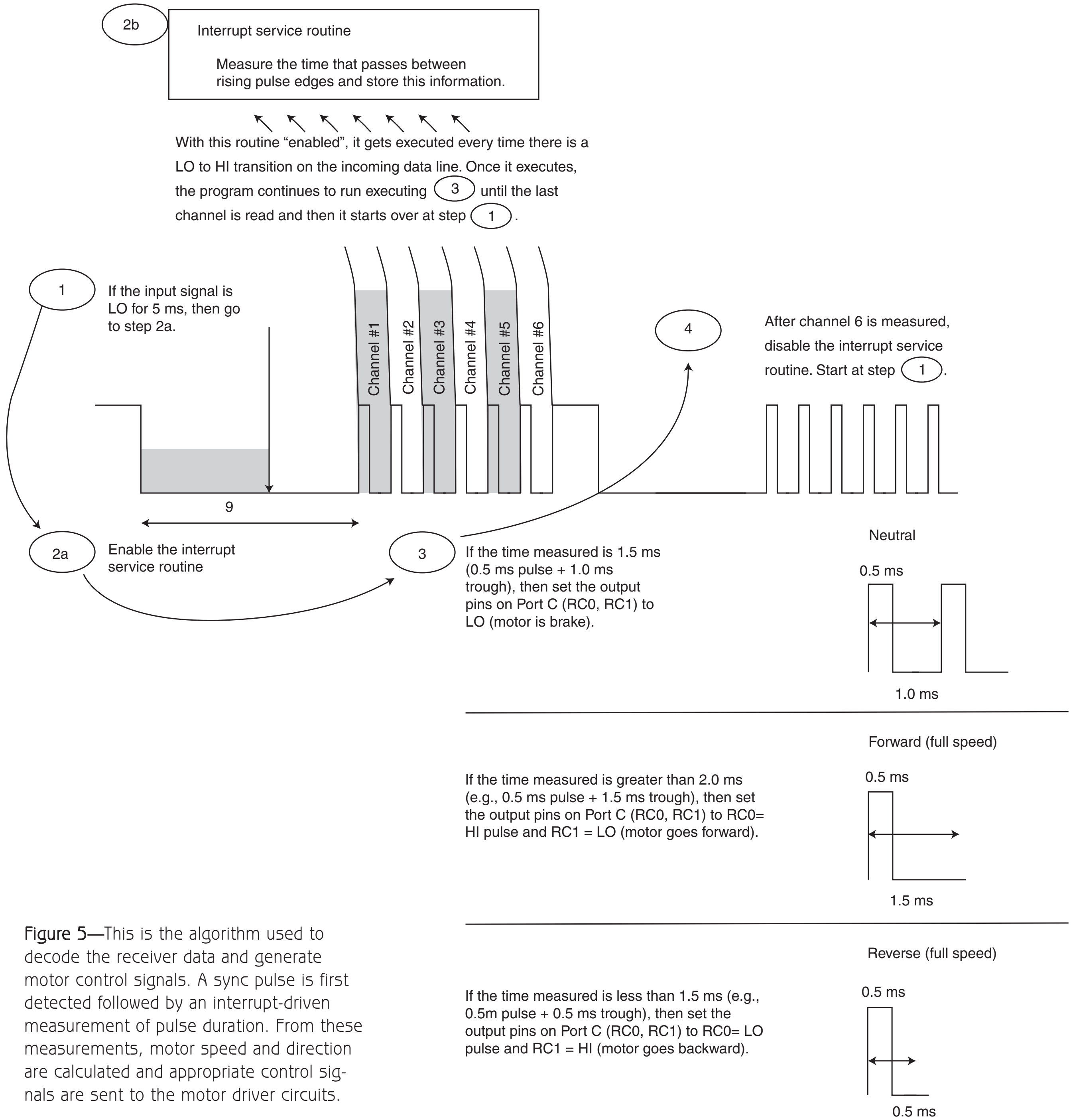


Figure 5—This is the algorithm used to decode the receiver data and generate motor control signals. A sync pulse is first detected followed by an interrupt-driven measurement of pulse duration. From these measurements, motor speed and direction are calculated and appropriate control signals are sent to the motor driver circuits.

next rising edge.

When the interrupt is triggered again, the timer is reread and the value is stored in memory (an array that is associated with the channel number) for later reference. The idea is to measure the time between rising pulse edges and then estimate the width of the trough, which provides information about both the direction and speed of the motor connected to each channel.

After the first channel is measured, the channel counter is incremented (set to 2) and the process starts again with the timer being zeroed when the rising pulse triggers the interrupt again. Processing activity within the interrupt is limited to either resetting or reading a timer value and storing it to memory for later analysis.

The entire program, as captured in Figure 5, measures pulse width duration to calculate motor speed and direction through a set of nested loops. The outer loop is used to calculate and set the duty cycle, which determines motor speed and motor direction for each individual channel.

The inner loop uses the duty cycle/direction information and sets two I/O output pins for each channel that are fed into an H-bridge circuit used in driving the motors.

METAL TIPS

Building an ROV or robot requires a fair amount of mechanical engineering as well. Aluminum sheet metal is useful because it is inexpensive, easy to cut, strong, and light. Unfortunately, it is difficult (not impossible) to bend because it is a brittle material. Using a vise to hold the metal piece and a hammer to bend the metal can work if you are careful.

I started to use cold rolled steel sheet for parts that required bending. This type of sheet metal can be bent without fracture and is readily available from metal shops. Even though it has been stressed while cold, it will bend (perhaps not as easily as hot rolled steel, which is harder to get at metal yards in small quantities). Cutting and drilling through steel is definitely harder as well; it should be since it is far stronger than aluminum. Bending steel is best accomplished by using a metal "brake." I can't imagine using a hammer and vise to make a fold.

DIY ROV DEVELOPMENT

A robot template is available on the *Circuit Cellar* FTP site. You can print the document and glue (rubber cement) it to aluminum sheet metal. With the pattern in place, you can use a metal shear or hacksaw to cut out the parts. You then can drill holes as specified on the template, make bends (again, as indicated on the template), and assemble the design. Attaching gears and wheels requires special attention. I find epoxy works well.

The days when kids built their own go-karts or made crystal radios from wire and a diode seem to be gone. Unfortunately, ready-made kits that take minutes to build and short circuit the learning curve seem to have taken over. It's my hope that kids will want to try and build their own ROV or robot from this information and then take it to the next level by designing an entirely new contraption from imagination.

The electronics can be reused over and over again, unlike those of an FM tube radio I built from plans when I was 14. I nearly electrocuted myself and used the hammer for something other than shaping metal. But that's a story for another time.

Brian Senese (brian.senese@sybase.com) holds an MS in Electrical Engineering. He is a senior product manager at iAnywhere/Sybase in San Diego, CA. Brian's interests include offroading, jetskiing, sailing, and playing with embedded Linux.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/248.

SOURCES

VEX Transmitter/receiver
Innovation First International | www.vexrobotics.com

PIC18F4620 Microcontroller
Microchip Technology | www.microchip.com

DC Motors
Namiki Precision Jewel Co. | www.namiki.net

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:

—

Robot Navigation and Control
by Guido Ottaviani
***Circuit Cellar* 244, 2009**
Guido built a navigation control subsystem for an autonomous differential steering explorer robot. Here he describes a robotic platform and a communication system for remote management. Topics: Robot Navigation, H-Bridge, Motor, Telemetry

—

Inertial Rolling Robot
by Jeff Bingham & Lee Magnusson
***Circuit Cellar* 200, 2007**
This H8/3664-based rolling robot is capable of inertial movement. A DC electric motor is attached to a pendulum and suspended inside an inflated ball, which provides the driving force. Topics: Rolling Robot, Motor, H-Bridge, PWM, Servo

—

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

ICbank
Gateway to Electronics

3 PORT INTERFACE
RS-485 to Ethernet Converter

Only
$170

RS-485 to Ethernet Converter

Powerful feature

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

Specification

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

ICbank
Gateway to Electronics

www.icbank.com

by Marco Signorini (Italy)

Embedded DSP for Lighting

It's possible to embed one DSP device and a series of constant-current generators to drive a set of high-power LEDs in an elaborate lighting system. This project uses digital signal processing (DSP) technology to create a modern version of a 1970s-era light system.

This project uses digital signal processing (DSP) technology to create a modern version of a psychedelic light system from the 1970s. I replaced old-style lights with high-power LEDs, and I created filter banks, implemented with analog components in the original systems, with DSP techniques. A Microchip Technology dsPIC30F2020 is at the center of the system (see Photo 1).

HIGH-POWER LED LIGHTING

LED lighting has become popular. The benefits of LEDs are their efficiency, lifespan, robustness, and small size. Plus, LEDs are available in many colors and their switching speeds make them excellent alternatives to traditional light sources in traffic signals, automobiles, and entertainment systems.

High-power LEDs should be current driven. The 1-W devices in my lighting project require 350 mA to properly operate. In order to maintain high efficiency, it is not a good idea to use a series resistor as a current limiter. A switching, current-limited power supply is required. Even better, in order to build a complete psychedelic light system, a three-channel, current-limited power supply is required: one for the red channel, one for the yellow channel, and one for the green channel. This could be an expensive project in terms of the number of components required when implemented with analog technologies. But it could be really efficient if implemented in the digital domain with pulse-width modulation (PWM) techniques.

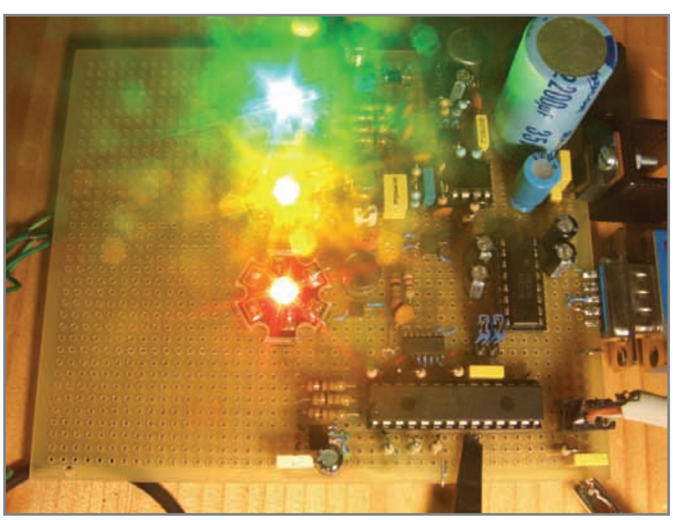


Photo 1—The LED lighting system on a prototype board

The same hypotheses are valid for the filter banks required to "split" the music frequencies into three different channels. In the 1970s, this was done with a set of operational amplifiers that were also used to rectify and integrate the resulting filtered signal. In the digital era, you can replace them with a unique DSP device.

And what happens if the DSP core is packaged with four PWM channels, a ready-to-use ADC, and some memory? That's what I wondered when I started developing this project.

SYSTEM OVERVIEW

Before going into the hardware details, I'll delineate some logical blocks. A simple block diagram is represented in Figure 1. Following the audio signal flow, you can start to analyze the system from the electret microphone. The signal captured is amplified and filtered by a fifth-order, low-pass filter. The resulting signal is applied to a digital-controlled, variable-gain amplifier and then goes onto the ADC present in the DSP. From this point, all the following stages are implemented in the digital domain. Here we can find first a high-pass filter, useful to remove a possible DC

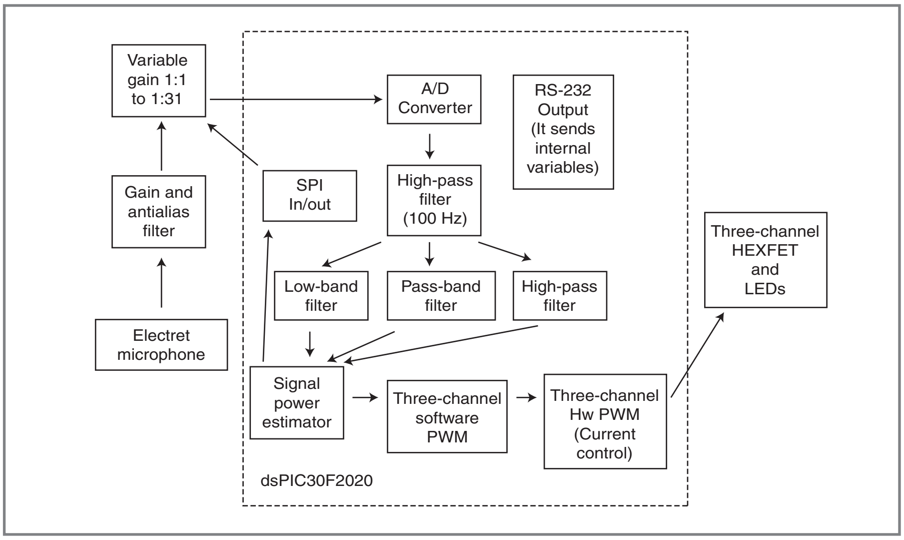


Figure 1—The complete project. The dashed line contains blocks that are implemented inside the DSP device (either in hardware or by a software routine set).

residual from the incoming signal, followed by a filter bank able to split the converted signal into three components: a low band, a middle band, and a high band.

Filtered and unfiltered signals are evaluated by a signal power estimation block that calculates four proportional values. One value is used to control the external programmable-gain amplifier (PGA). The remaining values are fed into the three-channel, software-based PWM generator. This is able to generate three independent 125-Hz waveforms with a 16-step variable duty cycle used for dimming purposes. These waveforms modulate, in software, the three hardware-based PWM channels that generate a 150-kHz square wave suitable to drive the external buck switching circuit MOSFETs.

HARDWARE

Figure 2 depicts the analog front-end subsystem. Figure 3 shows the power supply, the DSP, and the digitally controlled current generators.

Following the audio signal flow, the signal received by the electret microphone, MIC1, is amplified by the first Microchip Technology MCP608 op amp (IC2). In conjunction with IC3 (a Microchip Technology MCP6022), it implements the fifth-order antialiasing filter and is able to amplify the input signal by a factor of 10. In addition, the MCP608 (IC1) provides a VDD/2 voltage that's used as a "virtual ground level" for the analog signals. With a signal fluctuating around VDD/2 instead of ground, you don't need to have a bipolar power supply that includes negative voltages. The voltage shift is achieved through the 100-kΩ R11 resistor.

The filtered signal is fed into IC4, a Microchip Technology MCP6S26 PGA. The PGA output is connected to the ADC inside the dsPIC30F2020. The DSP is also connected to the PGA through a SPI. Unfortunately, the SPI serial I/O pins are shared with the dsPIC in-circuit debugger (ICD2) capabilities. This explains the presence of JP4 and JP5, which need to be shorted to GND when debugging or programming the device. (This makes it impossible to change the external gain when debugging, but it is not really important.)

High-power LEDs are driven by three independent buck-configuration, constant-current generators implemented by three P-channel MOSFETs and related feedback networks. The PWM waveforms, generated by the dsPIC30F2020, are properly shifted by two TC4427A low-side MOSFET drivers. The resistors R1, R6, and R11 are inserted to increase the MOSFETs' switching time in order to reduce the emitted switching noise (EMI). The current that flows through inductors L1, L2, and L3 and the LEDs produces a feedback voltage through the 1-Ω resistors R3, R8, and R13. The feedback signals, filtered by R4, R9, R14, C5, C15, and C17 to remove unwanted glitches are sent to the dsPIC analog inputs. Resistors R2, R5, and R7 are needed to eliminate spurious pulses when the dsPIC is not initialized.

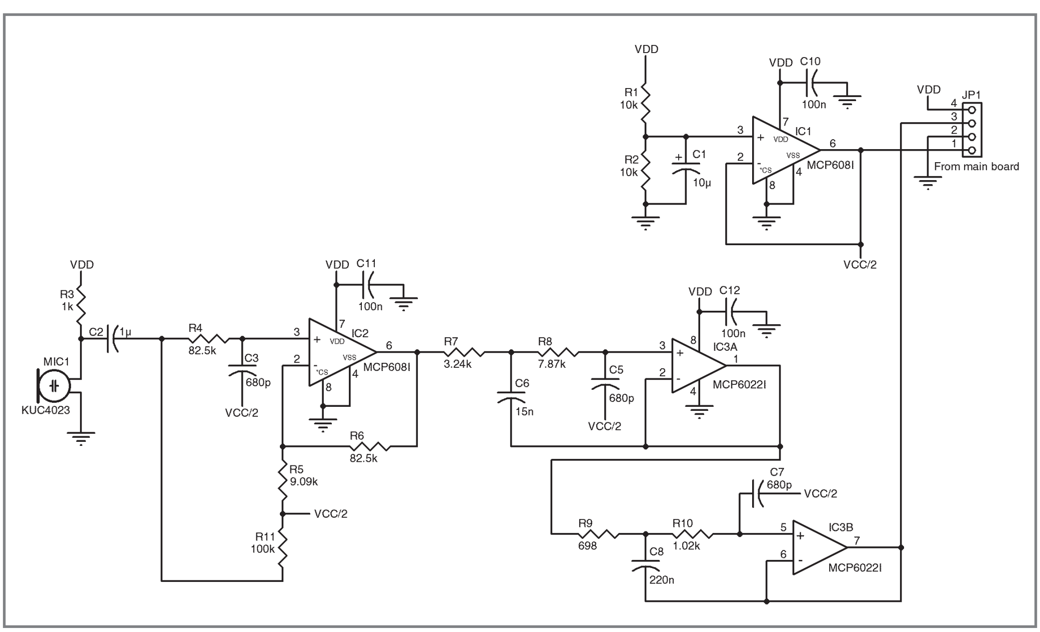


Figure 2—The analog input and antialias filter subsystem

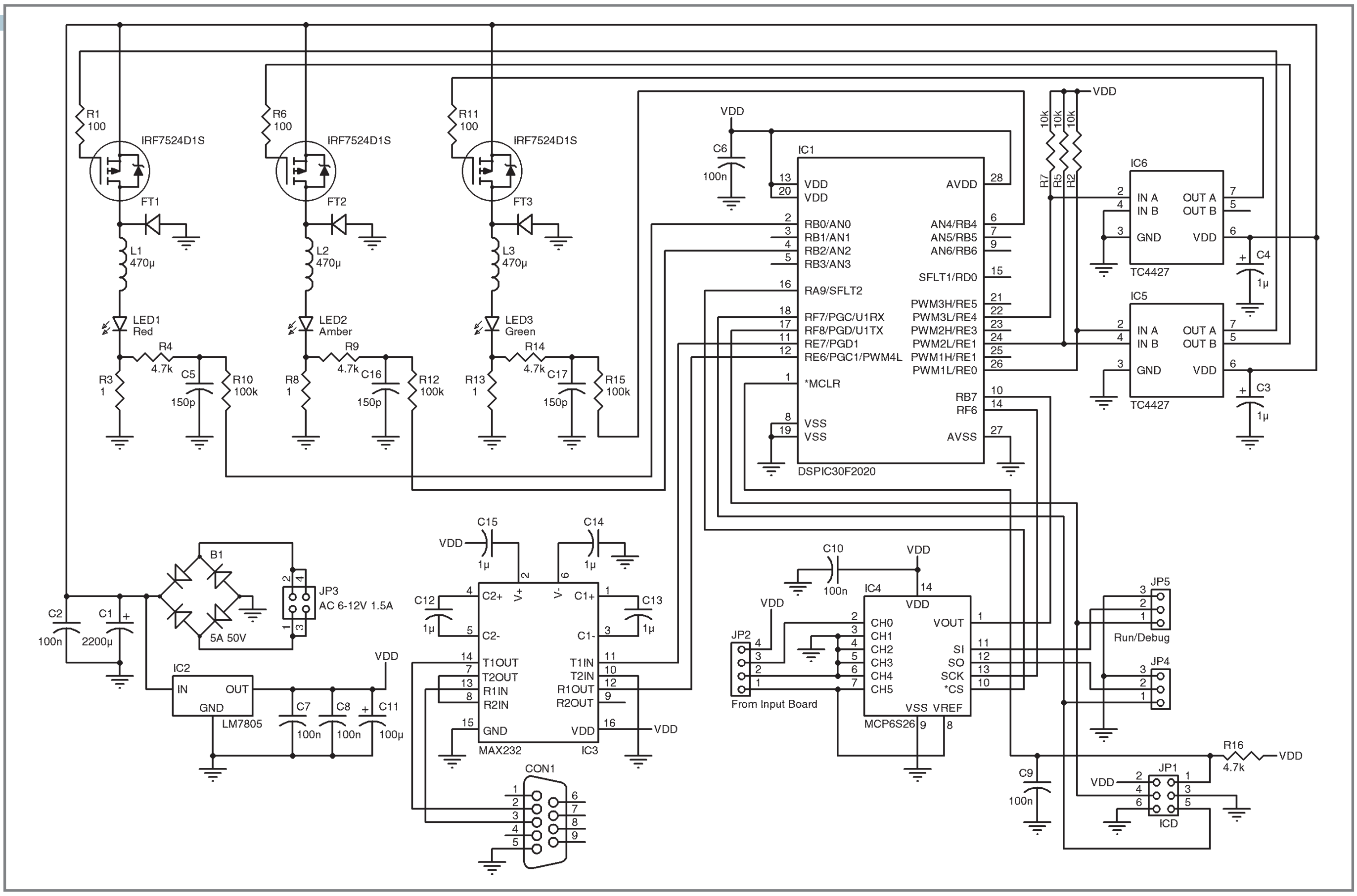


Figure 3—The power supply, variable-gain amplifier, DSP, and buck switching circuits

The dsPIC30F020 contains all the logic needed to implement a fully compliant serial line that is useful for debugging purposes. In order to connect an external PC, the TTL levels produced and received by the dsPIC have to be translated following the RS-232-C physical specifications. This is easily done by the MAX232 chip (IC3) and the related passive components.

A 5-V linear voltage regulator (IC2) provides power for the analog and digital components. The unregulated source is used to power the LEDs.

FILTER DESIGN

As previously noted, even if the filters that split the incoming signal into three bands are implemented in digital format, a front-end analog antialiasing filter is required. When studying digital signal techniques, one of the most important phenomena is related to the so-called Nyquist theorem. It defines the frequency limits that an analog signal, fed into an ADC working at fixed rate F_S, has to satisfy to be properly mapped in the digital domain. In detail, the Nyquist theorem affirms that one ADC, working at a fixed rate F_S, will be able to handle properly all signals with frequencies less than $F_S/2$. All portions of input signal residing in the spectral portion located over $F_S/2$ will be "folded back" below $F_S/2$ with the same amplitude (Lit1). This effect, called aliasing, can be eliminated by filtering out the frequency portion above $F_S/2$ before the digital conversion.

To ensure that amplitudes of aliased frequencies found in the digital domain have minimal impact on real signal amplitudes, the antialiasing filter needs to be able to attenuate signals over $F_S/2$ by a factor that's less than the ADC's signal-to-noise ratio (SNR). The dsPIC30F020 provides a 9-bit ADC block, resulting in an SNR near 55 dB. The audio signal spectrum I want to convert, on the other hand, is limited to 20 kHz. This forces me to select an A/D sampling rate F_S over 40 kHz. We would like to have

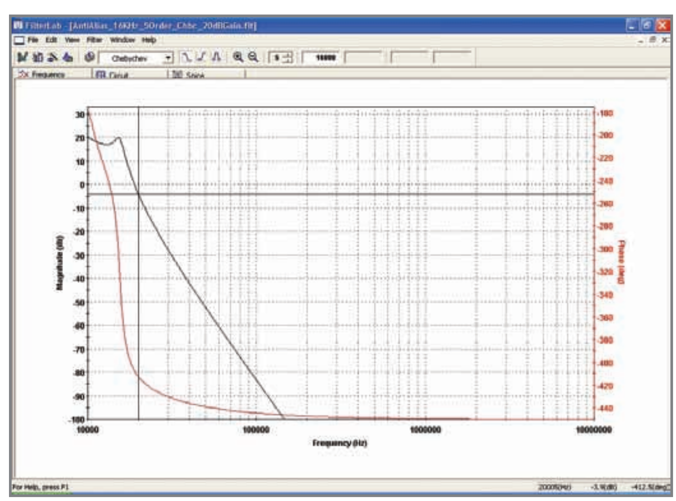


Photo 2—An antialias filter graph generated by FilterLab

	LP Filter	PB Filter	HP Filter
Cut-off frequency	1 kHz	2 kHz and 9 kHz	11 kHz
F stop	2 kHz	1 kHz and 11 kHz	9 kHz

Table 1—LP, PB, and HP filters

a microphone as an input source for the audio signal, so we need to develop an antialiasing filter with gain greater than 1.

Microchip Technology's FilterLab is an interesting tool for speeding up the development of analog filters. By following the "wizard," I can quickly define an initial filter implementation that best matches all my requirements. The resulting filter, unfortunately, was not within my target. However, iterating several times through the wizard and relaxing some initial requirements, the tool was able to provide a good solution. The selected microphone offers a flat response up to 15 kHz, so I chose a Chebyshev design for the filter with a 16-kHz cutoff frequency. With these characteristics, the filter is able to attenuate by more than 24 dB all frequencies over 20 kHz. Note that it's not the required -55 dB, but the attenuation provided by the microphone provides a good compromise. As you can see in Photo 2, the filter has an overall 20-dB gain.

The diagram shows that the digitized audio signal is filtered by a high-pass filter, used to remove any offset or DC residuals, and then feeds into a three-way filter bank (low-pass, band-pass, and high-pass filters). These four filters are implemented in the digital domain via a set of multiplication and sum operations performed by the DSP. As you might think, implementing all in a single dsPIC30F020 was very exciting.

Speaking of the filter bank (just forget for a moment the filter used for the DC removal), I decided to obtain at least 30 dB of rejection in the stop band, and 1-dB maximum ripple in pass-band frequencies, with stop frequencies partially overlapping. This offers good channel separation with the most flat response. These requirements are listed in Table 1.

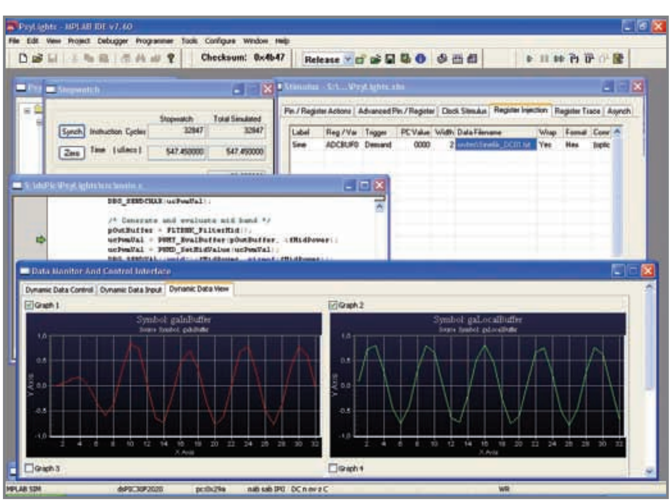


Photo 4—Data monitor and control interface in MPLAB IDE showing a simulated input signal and the corresponding filtered output buffer

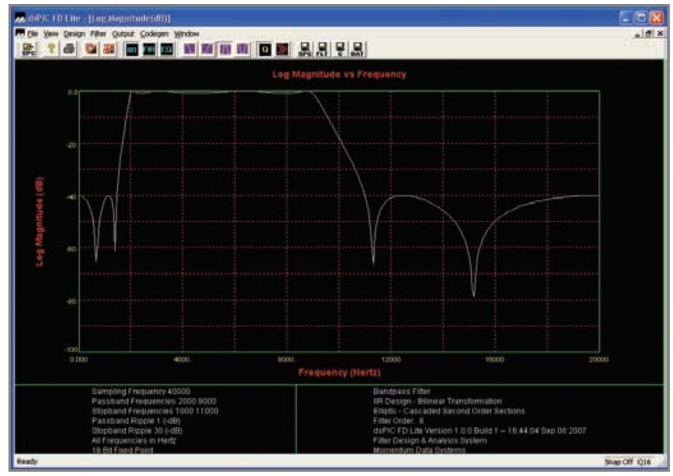


Photo 3—An IIR band-pass filter frequency graph, as reported by dsPIC FD Lite

Momentum Data Systems's dsPIC FD Lite is a tool that helps design digital filters. It was really useful because I could test different filter implementations, simulating and comparing the resulting frequency charts.

The memory constraints imposed by the dsPIC30F020—specifically related to the X_Memory space used by the DSP core to perform fast computations—make it impossible to implement the filters using finite impulse response (FIR) techniques. Simulating the best-matching FIR filters shows that 318 bytes of memory would be needed for the filter coefficients alone (using Kaiser window: 63 taps for low- and pass-band filter, 33 taps for high-pass filter, for a total of 159 coefficients).

Infinite impulse response (IIR) techniques are less expensive in terms of memory requirements, but at the expense of ripple magnitude in the passband. The simulations, made through the tools, revealed interesting results using a fourth-order elliptic filter for low-pass and high-pass filters and an eight-order elliptic filter for the band-pass filter stage. The memory required for the IIR filters' 16 coefficients is just 32 bytes. Photo 3 is an example of a frequency graph generated by the tool. Note that the filter can provide more than 40 dB of rejection in stop band.

Due to the good performance characterizing IIR filters, DC removal was implemented in the same way. I used a fourth-order elliptical high-pass filter with a frequency cut-off equal to 100 Hz and a 50-Hz frequency stop.

GENERATORS

The dsPIC30F020 is a DSP suitable to be used in switching power supplies. It contains a four-channel, high-frequency PWM generator with hardware current-fault control. For this reason, it can be efficiently used to implement a three-channel, constant-current generator required to drive three high-power LEDs in a buck configuration.

The powerful PWM generator stage is so flexible that it is possible to directly drive three external MOSFETs and evaluate the generated impulse feedback voltage without any software intervention. This is perfect to replicate the

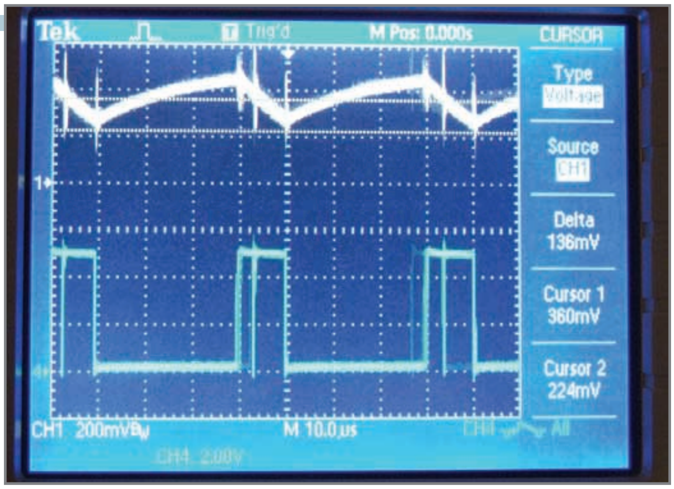


Photo 5—This is a hardware-generated PWM signal in channel 4 and relative feedback voltage in channel 1.

buck topology as explained in Microchip's AN874, reducing the required component count.

The software initializes the PWM registers in order to generate a 150-kHz square wave with a 90% duty cycle. This working frequency was selected because it best matches the requirements to reduce the size of the inductors (an inductor's dimensions are directly proportional to the current flowing through it and inversely proportional to the frequency), and it's easily tolerated by the selected MOSFET and drivers. The current that lights the LED, when flowing in the feedback resistor, generates a signal voltage. The 1-Ω resistor produces 370 mV at the LED's rated current. The feedback voltage is fed directly to the dsPIC analog input, where it is compared by an internal analog comparator with a voltage reference provided by an internal DAC stage properly initialized by the software. The comparison result is used to blank the PWM channel output for the required PWM cycle time. All is done without any software intervention (except for peripheral initialization). The DSP's time is freed and available to perform calculations for filtering purposes and to generate a low-frequency (125 Hz) PWM signal useful to modulate the amplitude of each LED.

FIRMWARE

I developed the firmware in C language. The files are posted on the *Circuit Cellar* FTP site.

For the filtering stages, it uses Microchip's DSP libraries included in the C30 compiler. The main loop is responsible for initializing the micro and the related peripherals. It then periodically calls the proper DSP processing routines sequence at a rate specified by the interrupt routine through some shared flags.

The main loop is also responsible for handling the information related to the estimated signal power and to call some routines, if required, to modify the external PGA gain. Last but not least, it's in charge of sending all debug information through the serial port when compiled with the proper debug option.

The adc.c file implements the routines needed to initialize the ADC module and the related ISR. The ADC produces 40,000 samples each second. These samples are stored in a double buffer implemented by the inbuffer.c file. A double buffer is needed to prevent overlaps between samples coming from the ADC and the already-collected samples processed by the filter banks. With this solution, the ADC result is placed in an area of RAM different from the DSP working area, and the two areas are swapped at the end of any filter calculation. A new DSP computation phase is started by the main loop as soon as the ADC completes its working area.

In the filterbank.c file are the routines used to initialize the filters and to filter the input buffer for each band. They use the filter coefficients generated by the dsPIC FD Lite tool, linked as assembler code to the project.

The buffers containing the filtered signal have to be evaluated in order to decide if and how much the corresponding LED has to be lit or if the PGA gain has to be modified. This is implemented in the pwmeter.c file. The power estimation value is calculated as an average of squared sample amplitudes. The result is averaged with a previously calculated fractional value in order to maintain the signal history integrity. To perform fast computations, the DSP's 17 × 17-bit multiplier and 40-bit accumulator supported by the compiler's built-in functions were used. This makes it possible to fetch data, multiply, accumulate, and shift the result in a single instruction.

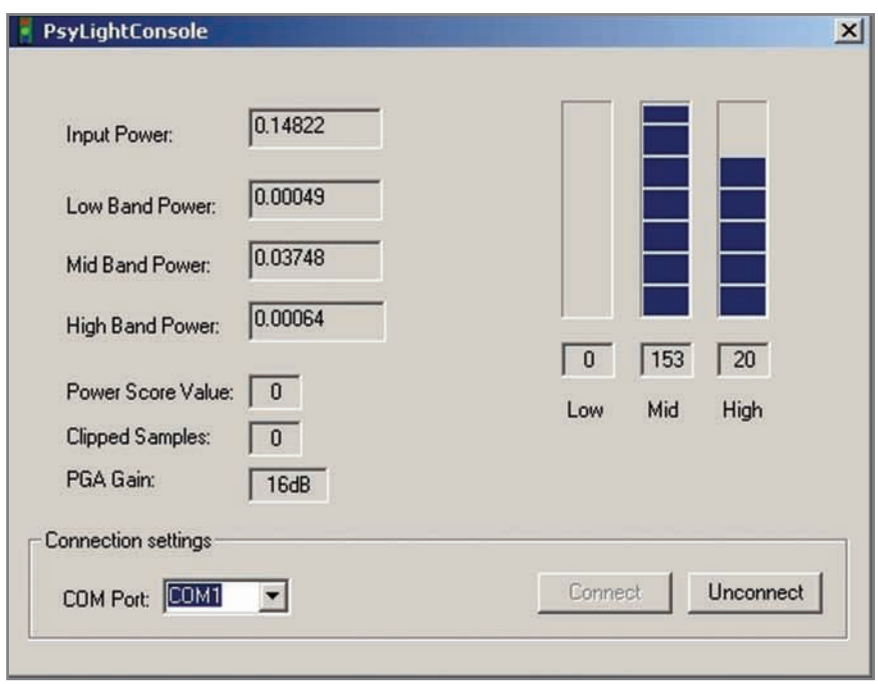


Photo 6—The PsyLight debugging console. An application is running.

ESC
Silicon Valley • May 2-5, 2011

Join the industry's leading embedded systems event
(Translation: Can't miss event)

ESC brings together the largest community of designers, technologists, business leaders and suppliers in one place.

Keynote Speaker

Steve Wozniak

Co-Founder, Apple Computer, Inc.
and electronics industry visionary delivers the opening keynote speech on Tuesday, May 3rd at ESC Silicon Valley!

Categories and Tracks that address the most relevant issues facing engineers and the industry. Take a moment to review this content, so that you can customize your educational experience.

Applications
- HMI and Multimedia
- Systems Architecture
- Reliability, Security, and Perforance
- Remote Monitoring and Wireless Networking

Embedded Software
- Linux/Android/Open Source
- Programming for Storage, I/O, and Networking
- Programming Languages and Techniques
- RTOS and Real Time Software
- Software Processes and Tools
- Windows for Embedded
- Quality Design and Untellectual Property
- Safety Design

Hardware for Embedded Systems
- Challenges and Solutions in Embedded Designs
- Connectivity and Security
- Memory in Embedded Designs
- Microcontrollers in Embedded Designs
- Powering Embedded Designs
- Programmable Logic in Embedded Designs

Tools and Best Practices
- Best Practices
- Debugging and Optimizing
- Design and Test
- Managing and Process
- Tools

Topics in Embedded-System Design
- Architecture Design
- DSP, Communication, and Control Design
- HW and Platform Design
- Quality Design and Intellectual Property
- Safety Design

Did you know that Steve Wozniak is an ESC Alumni?
"It will be good to reconnect with the engineering community that still drives so much electronics innovation."
Steve Wozniak
Co-Founder, Apple Computer, Inc. and Chief Scientist for Fusion-10

Register today and save 10% OFF any package when you enter the promo code: sp
www.embedded.com/cc

Learn today. Design tomorrow.
ESC
Silicon Valley • May 2-5, 2011

UBM
Electronics

The power estimator is also responsible for calculating a proper gain to be applied to the input signal through the external PGA. This is accomplished by some algorithms that take care of parameters such as the number of consecutive samples with maximum dynamics found in the input buffer. The number of consecutive "clipped" samples is counted by a specific function. If this number is greater than a quarter of the input buffer length, I assume the incoming signal has been "squared" by a high external gain and should be attenuated.

This project demonstrates it's possible to embed in a single inexpensive DSP device a series of constant-current generators that are able to drive a set of 1-W, high-power LEDs.

The value calculated by the power estimator is dynamically compressed by the pwmdriver.c routines through one of the experimentally defined curves, selected at compile time. These routines also implement the state machine required to generate the software-based PWMs that modulate the LEDs.

TESTING & DEBUGGING

I developed and tested the firmware mainly in a simulated dsPIC30F020 environment via Microchip's MPLAB IDE. When the prototype board was assembled, the in-circuit debugger was useful for debugging the hardware and tuning some firmware parameters.

The simulated environment enables me to apply an "analog signal" to the virtual ADC to provide stimulus for the test code and, interestingly, to the simulated dsPIC registers. This was mainly done using text-based, comma-separated files containing test waveforms generated using open-source audio editors (e.g., Audacity) and a simple C binary-to-text conversion tool written ad hoc. Displaying input, output, and filtered buffers in a graphical way through the "data monitor and control interface" panel was an invaluable aid when developing and debugging the filter banks (see Photo 4).

The circuit was implemented through a prototype board. I performed tests principally with a digital oscilloscope connected to the feedback resistors. The LEDs were replaced with 10-Ω, 5-W resistors in series with 1N4007 diodes to avoid damaging the expensive high-power LEDs. Photo 5 is an example of a generated PWM square wave found at the dsPIC output pin (in the oscilloscope channel 4) and the resulting feedback signal in channel 1.

I experimentally tuned some firmware parameters—like the coefficients associated with the external gain change or the compression curve needed to have a better and more fluid light display—with an ad hoc Windows console application. It presented, in a graphical fashion, the values sent by the firmware through the serial interface. Photo 6 is a live, running application. You can see various firmware internal parameters: the fractional values for input, low-band, mid-band, and high-band power estimation; a graphical representation of low-band, mid-band, and high-band compressed amplitudes; the current score value calculated by the automatic gain-control engine; the number of clipped samples found in the input buffer; and the selected external PGA gain factor.

IMPROVEMENTS TO COME

This project demonstrates it's possible to embed in a single inexpensive DSP device a series of constant-current generators that are able to drive a set of 1-W, high-power LEDs. It shows how the same device could be used to perform some real-time audio processing, modulating the perceived LEDs' light similarly to old psychedelic light display systems.

The design process was really interesting thanks to some highly productive tools that enable you to simulate and experiment with a lot of technical solutions before writing the C code. However, I have some improvements in mind. The most important is related to the DC removal. Currently implemented by a high-pass filter, it could be replaced by an ad hoc moving-average mean estimator that evaluates the mean offset and subtracts it from the incoming signal.

Another really interesting improvement would be to replace the transformer-based power source with a more efficient, autoranging switching power supply, using the fourth PWM channel available in the dsPIC30F020. And last but not least, spending some time to select a set of higher-voltage MOSFETs would enable you to drive chains of high-power LEDs for each channel, increasing the overall light generated by the system. ◼

Marco Signorini (marco.signorini@libero.it) has a degree in Telecommunications Engineering from the Politecnico di Milano, Italy. After working as a researcher at STMicroelectronics and Whirlpool Europe, he now co-owns INGEGNI Tech S.r.l., which is a private system-integration company focused on open-source technology and VoIP deployment. Marco is an amateur radio operator and enjoys programming and developing embedded systems that can operate in network environments.

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/248.

RESOURCES

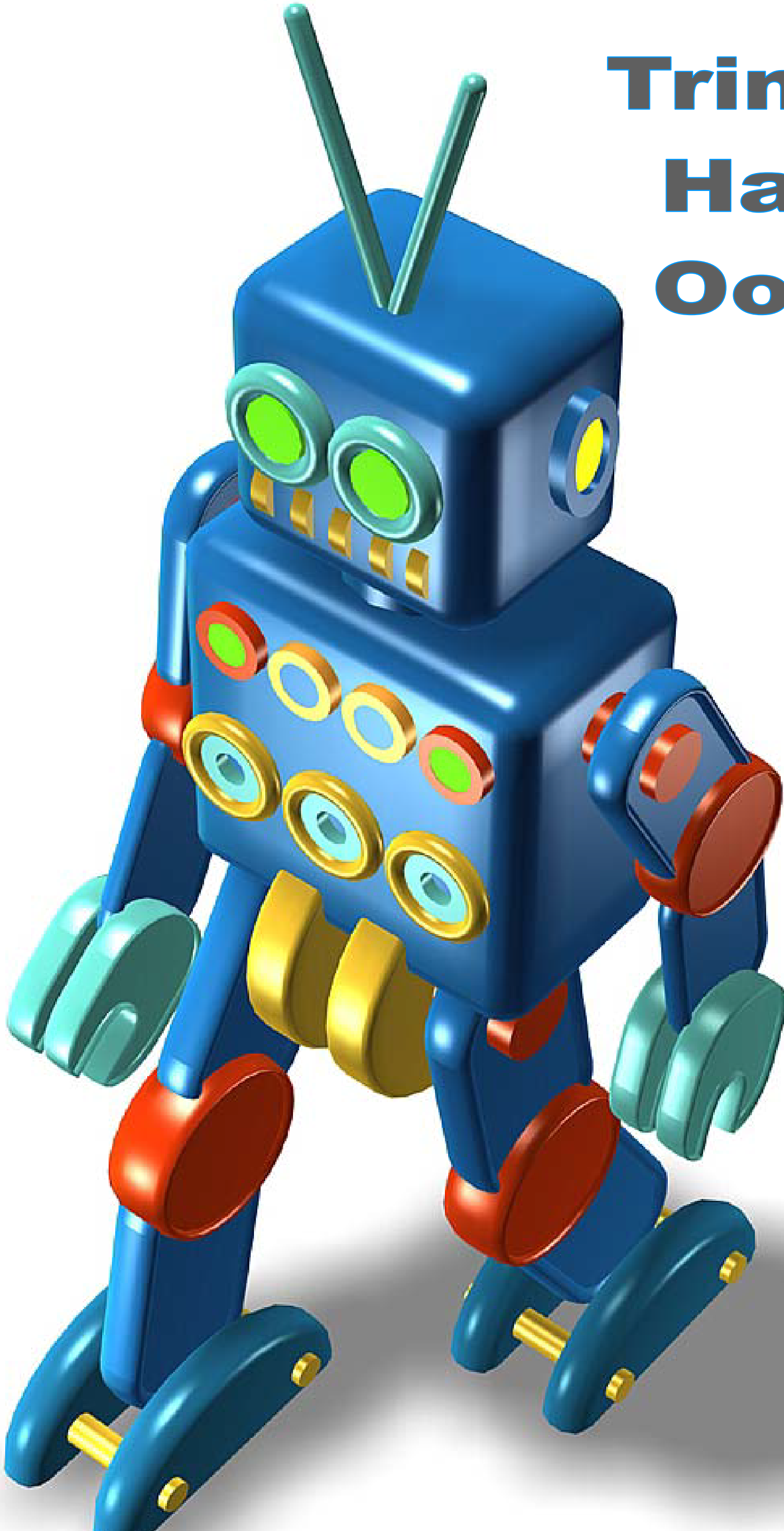
B.C. Baker, "Anti-Aliasing, Analog Filters for Data Acquisition Systems," AN699, Microchip Technology, 1999.

K. Curtis, "Buck Configuration High-Power LED Driver," AN874, Microchip Technology, 2006.

SOURCES

dsPIC30F2020 DSC and MCP6S26 PGA
Microchip Technology | www.microchip.com

dsPIC FD Lite
Momentum Data Systems, Inc. | www.mds.com



Trinity College
Hartford, CT
Oosting Gym

Saturday, April 9
9:00 a.m.—8:30 p.m.
Robot Practice

10:00 a.m.—11:30 a.m.
Robotics Workshops

12:30 p.m.—1:45 p.m.
Robotics Keynote Speakers

2:00 p.m.—4:30 p.m.
Connecticut Council on Developmental Disabilities presents
RoboWaiter Competition

Sunday, April 10
11:00 a.m.—4:00 p.m.
Versa Valves, Inc. presents
Firefighting Home Robot Contest

Learn More. Volunteer. Register.

www.trincoll.edu/events/robot

Connecticut
Council On
Developmental
Disabilities

by Brian Millier (Canada)

CNC Router Design

A Look at Computer-Controlled Machinery

Self-sufficiency is a goal toward which many electronics engineers aspire. Reaching that goal involves more than fine-tuning your design and programming skills. It also requires you to have an accessible set of reliable tools for bringing projects to completion. Here you learn how to build one such instrument—a CNC router.

I am basically a one-man shop when it comes to most of my projects. I design and build custom electronic devices, and write any firmware and application software that is required. I usually do the necessary mechanical and cabinet work. In the past, I'd have to occasionally depend on a machinist friend for assistance when a project required tools such as a milling machine or lathe. But not any more. I recently developed my own computer-controlled milling machine and router.

"THE LOAN"

As luck would have it, back when I was working at Dalhousie University in Halifax, Canada, I knew a faculty member who had a small, unused Taig milling machine, complete with a third-party computer control (CNC) option. He let me have at it, and within a day or so, it was assembled and I had scrounged up a spare computer to run the control software.

My first attempts at cutting patterns in aluminum panels were unsatisfactory. But, after talking to the machinist, I realized that I had to significantly reduce the turning speed of the mill bit, and he also gave me a high-quality mill bit to work with. Question: Why do companies bundle the cheapest, poorest quality bits or blades with their tools, thus ensuring that your first experience will be disappointing? However, after that disappointing start, I was soon able to produce decent-looking cutouts in panels and cabinets.

While Taig calls its product a "MicroMill," it is somewhat

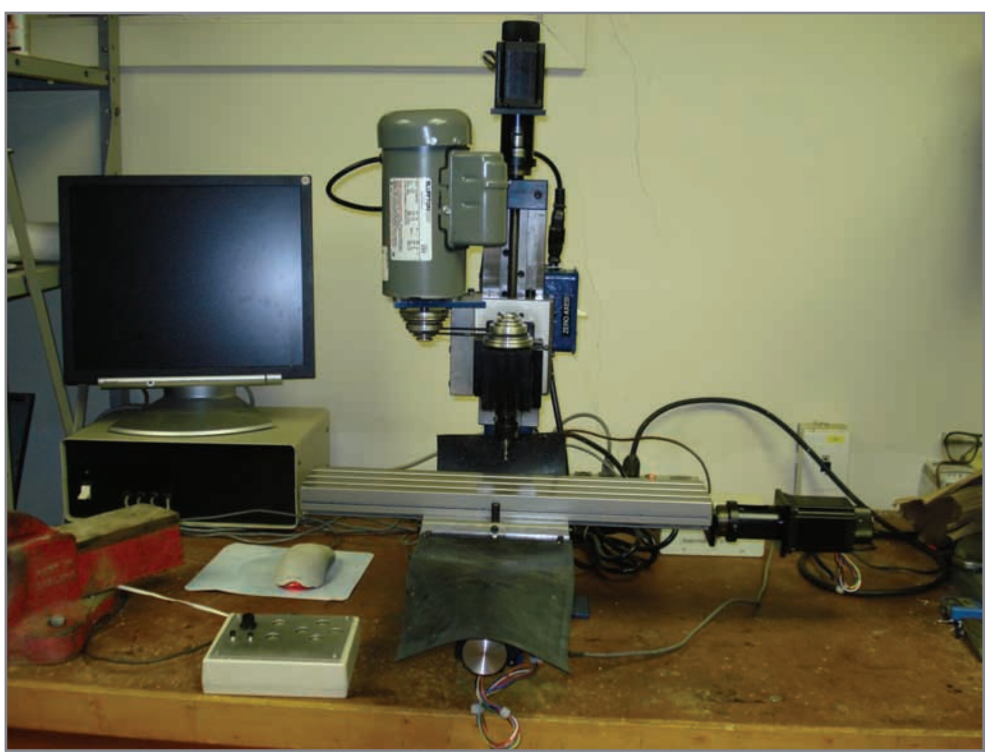


Photo 1—This is the "on-loan" Taig setup I had in my shop at Dalhousie University.

bigger and sturdier than other small units, such as Sherline devices that are basically designed for small work, such as jewelry. The Taig is built like a miniature version of an industrial-grade vertical milling machine, such as the Bridgeport unit that we had in our machine shop. Photo 1 shows the "CNC-ready" version of the Taig MicroMill, which is the model that I had on loan. I have mounted three Keling stepper motors to drive the x, y, and z axes. The stepper motor controller is housed in the enclosure to the left. The blue motor on the left turns the cutting tool via a stepped pulley/belt arrangement. The square black spindle assembly in the center contains bearings and a holder to take one of the six collets to hold mill bits of various shank diameters up to a maximum of 0.375″.

Performing cuts with a vertical mill involves mounting your work to the cross-slide table, which moves side to side (x-axis) and front to back (y-axis) under control of the x/y-axis motors. The spindle, to which the rotating cutting tool is mounted, moves up and down (z-axis) under control of a third axis drive motor.

The Taig mill I have is the "extended model," which has a cross-slide table that's 3.5″ wide × 18.5″ long, but the standard model is a bit smaller. The table travel for this larger model is 12.5″ in the x-axis and 5.5″ for the y-axis. The tower, upon which the spindle/motor is mounted, is not that far back from the cross-slide table itself. That limits the size of your work piece in the y direction. In practice, you can't mill a hole in your work piece more than about 5.5″ away from the top (or bottom) of the piece itself, without having the piece hit the tower itself. Anyone using a drill press has likely run into this "throat size" limitation.

The Taig mill is adequate for some of my projects, but I often run into travel limitations when working on some of my larger cabinets/panels, such as 19″ rack panels. The exception is 1U/2U panels, which can be held down to the table using clamps on the top and bottom.

An important consideration in a machine like this is how well the cross-slide table performs in terms of rigidity. If there's flexing or "play" in either the x-axis or y-axis, the cut will be rough and inaccurate, and the tool will vibrate or "chatter" during milling. You don't want that. The Taig mill is well designed in this respect, with hefty steel ways in which the table can travel, as well as adjustable brass gibs (to compensate for wear over time). Also, a very important criterion in a CNC mill is that the drive train for each axis must be free of backlash. On a manually controlled mill, some backlash in an axis drive can be tolerated or compensated for by the operator; but, in a CNC machine, the computer is generally unable to correct for this shortcoming. The Taig mill uses 0.5″ diameter Acme lead screws and antibacklash lead nuts for each axis drive train, so backlash is not a problem.

The Taig's spindle drive assembly is quite sturdy and accurately machined. There really isn't run-out ("wobble") in the cutting tool as it spins. This is important as it affects the accuracy of a cut. The 1,000-to-10,000-RPM spindle speed range is well suited for general-purpose metal work, but it would be too slow for some tasks, such as engraving with very small bits.

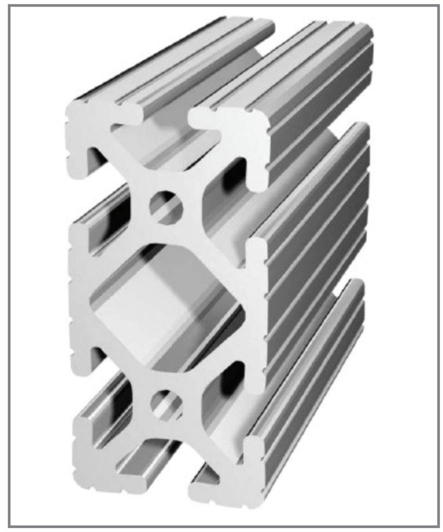


Figure 1—A 1 × 2 section of the 8020 aluminum extrusion showing the slots for carriage bolt fasteners

Generally, customers outfit the Taig MicroMill with NEMA 23 stepper motors for all three axis drives. Using the standard 200-step/turn stepper motors, and given the Taig's 1/2″ 20-TPI lead screws, the step resolution is 0.000125″ when using only a simple half-step controller. Step resolution of five times greater than this is possible when using a 10-microstep controller, such as the Geckodrive controllers, which are very popular with small CNC machines. Note that I am referring to the electrical resolution of the stepper drive alone; in practice, the overall mechanical resolution is less than this due to the tolerances of the lead screw/lead nut, and other parts. However, Taig claims an overall working accuracy of 0.0005″, which is excellent.

Modern NEMA 23 hybrid stepper motors, such as those manufactured by Keling, are available with high torque ratings (200 to 300 oz-in) for

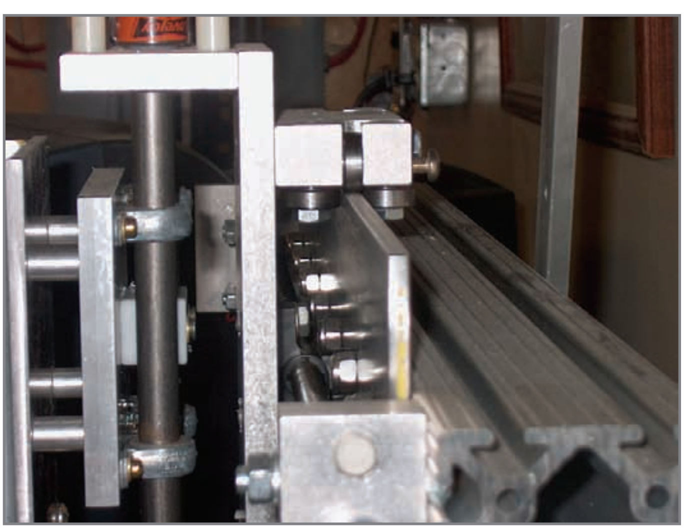


Photo 2—A closeup of the linear carriage assembly, running on the 0.25″ steel rails

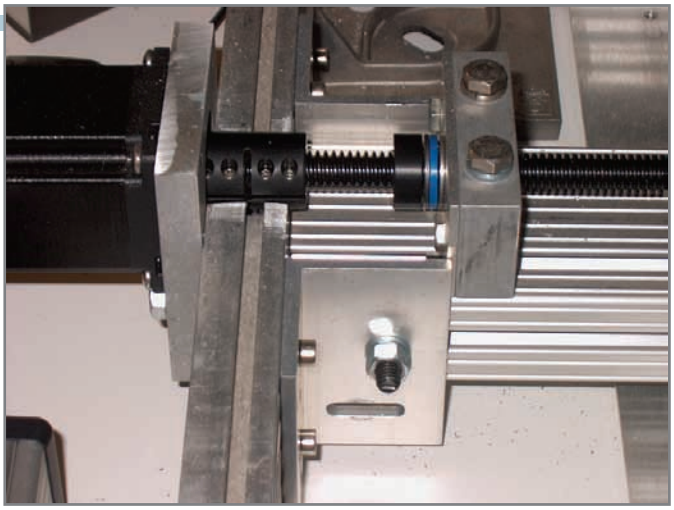


Photo 3—A closeup of the y-axis drive train assembly

around $40. Looking at some older, larger NEMA34 stepper motor "pulls" in my surplus motor drawer, it's obvious that the new stepper motors are much more powerful—and that's at a fraction of the price and half the size! Given the affordability of stepper motors and matching controllers, it's unlikely that one would opt to outfit a Taig MicroMill with a much more expensive servomotor drive, although it's possible. The big advantage of a servomotor drive is in increased travel speed. However, for a small mill like the Taig, it's unlikely this is an important consideration.

In a nutshell, I was very pleased with the Taig's operation for my aluminum panel/cabinet work. However, I was running into travel limitations on some larger workpieces. Thus, I started thinking about buying or building a larger unit for myself.

DECISIONS, DECISIONS

I ruled out buying a larger CNC vertical milling machine mainly due to the price. It was much more expensive than the roughly $2,200 Taig CNC machine. I briefly looked at some decent-sized, manually operated vertical milling machines. They all were made in Asia and were not well suited for CNC retrofitting for several reasons, not the least of which were backlash issues.

I then changed my focus to CNC routers, which are designed somewhat differently and are more commonly used for wood and plastics. Since I really only needed to mill thin aluminum panels (and also do carpentry as a hobby), the CNC router seemed to be a good compromise. On the Internet (YouTube is a good resource), I found many examples of homemade DIY routers, as well as some commercial units and kits. At the low end were commercial/DIY units, the frames of which were either particle board (MDF) or plastic. These are really only good for working with plastics and foam, so I ruled them out. There were also many examples of rather sturdy units, framed with steel or aluminum, which had table travels of 2′ × 4′ or greater. In this case, they were too heavy and expensive to ship, and larger and more costly than what I really needed.

It was at this point that I had a "Eureka" moment. I came across a website describing CNC router kits and parts based around the 8020 extrusion system. Building a CNC router using this method enables a person who does not have a lot of machining tools (such as a milling machine) to build a CNC router accurately from scratch.

The best way to explain the "8020" extrusion system is to take a look at a piece of it. As you can see in Figure 1, this particular profile is made up of two "X" shaped sections. Looking closely at the open areas in the profile, you can see it is shaped to accept common carriage-head bolts, which can slide anywhere along the length of the profile. While 8020, Inc. produces many different profiles (both Imperial and Metric), the 15 Series is ideal for a CNC router. There are numerous 15-Series profiles available (1 × 1, 1 × 2, 2 × 2, etc.) and the slot-to-slot spacing is 1.5″ throughout the series. The slots accept 5/16″ × 18 carriage bolts. For places where you need a different type of fastener, 8020, Inc. makes hundreds of specialty brackets and fasteners for that purpose. Think of the 8020 system as a Lego set for industrial purposes.

Just as important as the flexibility of this system is the fact that 8020 can supply any of these extrusions cut precisely to length for a few dollars per cut. This is critical because many designers don't have a way of performing such accurate cuts themselves. The 8020 website has a number of manuals that include examples and methods of using their components.

Another piece of the puzzle was finding a company that makes and sells linear carriages, bearing blocks, and NEMA 23 (or 34) stepper motor mounts, all sized to mount perfectly on the 15-Series extrusion. (Refer to the Sources section at the end of this article.) Photo 2 shows a smaller carriage running on a 0.25″ thick steel rail, which is used for my router's x-axis.

The final piece to the mechanical puzzle is the axis drive mechanism itself. Photo 3 shows the y-axis drive

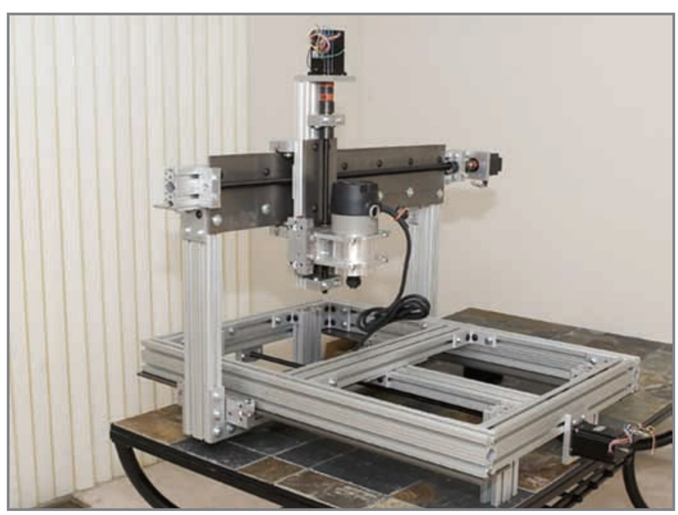


Photo 4—A moving gantry router, built using an 8020 aluminum extrusion. (Source: www.glacialwanderer.com/hobbyrobotics/?p=17, Maurice Ribble)

mechanism used in my router. (The x-axis is similar.) The Keling stepper motor is mounted to the motor mount plate, which comes predrilled to mount onto the end of an 8020 extrusion. To the right is the bearing block, also designed to mount directly to the 8020 extrusion using 5/16″ × 18 bolts and T-nuts placed in the extrusion's slots. The blue item is part of a thrust bearing assembly.

While you would generally need a true flexible coupling to connect the stepper motor to the Acme lead screw, all of the previously mentioned parts are very accurately machined, so the best way to couple the motor to the Acme screw is to use an antibacklash coupler designed to connect the motor's 0.25″ shaft to a 1/2″-10 Acme screw.

The Acme lead screw and lead nut convert the rotation of the stepper motor into linear motion. Acme lead screws come in many sizes and specs, but the 0.5″ size is well suited for a machine of this size. I chose an Acme screw with 10 threads per inch and what is called "single start." These will move 0.1″ per revolution of the motor and provide higher linear torque. Many larger router kits use "five-start" Acme screws, which move 0.5″ per motor revolution. These move the carriage five times as fast for a given motor speed, but with much less linear torque. I'm not interested in high-speed axis motion, and preferred the increased torque provided by the single-start Acme screws. I was surprised to find that Acme lead screws were not readily available in Canada, at least not for someone looking for a few feet of them. My only source of Acme screws was McMaster-Carr in the United States.

I used DumpsterCNC antibacklash lead nuts recommended to me by people who had previously built 8020-based routers. These lead nuts have to match the Acme screw threads you choose, of course.

GANTRY VS. TABLE

Having settled on a basic construction method, I then had to decide on the style of router I wished to build. There are two distinct types: moving table and moving gantry. The most common option is the moving gantry type, as shown in Photo 4. As you can see, the frame is fixed to the table, but it's mounted on short legs to provide for clearance below. There is a large gantry, which extends both above and below the frame, and which moves from one end of the frame to the other, providing for x-axis travel (or y, depending on your point of reference). At the top of the gantry is the y-axis assembly. Fastened to that is the z-axis assembly, used to move the cutting tool (a router) up and down.

The workpiece is mounted to the frame using some form of mounting jig, or, alternatively, a sacrificial wood/plastic board is fastened to the frame to form a full top. This particular machine is an 8020-based design. It uses the same carriage assemblies, motor mounts, bearing blocks, and so forth as described in the previous section, as well as 0.5″ cold-rolled steel rails. Maurice Ribble built and authored a great online-illustrated "build guide" for this particular machine. Refer to Ribble's blog listed in the Resources section at the end of this article.

The advantage of the moving gantry router is that it maximizes the amount of x/y travel for a given machine footprint. The disadvantage is that the gantry is a rather large, heavy assembly that must move smoothly and accurately on the steel rails mounted to the bottom of the frame. To accurately cut metal, this large gantry has to be very precise and sturdy. The thought of that was a bit daunting.

With that concern in mind, I instead decided to build a "moving table" router. I came across a website run by Ilya Dontsov in Russia. He describes such a unit, based on the 8020 extrusion system (see Resources). My unit differs from his in several ways, but the concept is the same. Ilya was a great resource. We had a great e-mail exchange during my router "build."

Photo 5 shows my finished router located in my carpentry shop. You can see that the frame rests securely on a home-built aluminum base unit. Just above the frame is the moving table, which is about 19″ square and constructed entirely from precut 8020 extrusions. The table's left and right extrusions fasten to the three table cross members using 8020 Inc.'s 3278 end fasteners, which make for very accurate, strong joints. This is important as the table assembly must be perfectly square and tight to properly mate with the y-axis carriage assemblies and move smoothly along the steel rails.

In this design, the gantry, which contains the x-axis and z-axis linear motion assemblies, is fix-mounted to the frame. This makes it much lighter and, at the same time, it's easy to make it plenty rigid.

What I have yet to mention, which also biased my choice of designs, was yet another "Eureka" moment. I had been immediately impressed by the 8020 extrusion system, but I was somewhat concerned about shipping all that heavy extrusion into Canada from the United States, since shipping would be expensive. Fortunately, I

Photo 5—My moving table CNC router, shown in my home carpentry shop

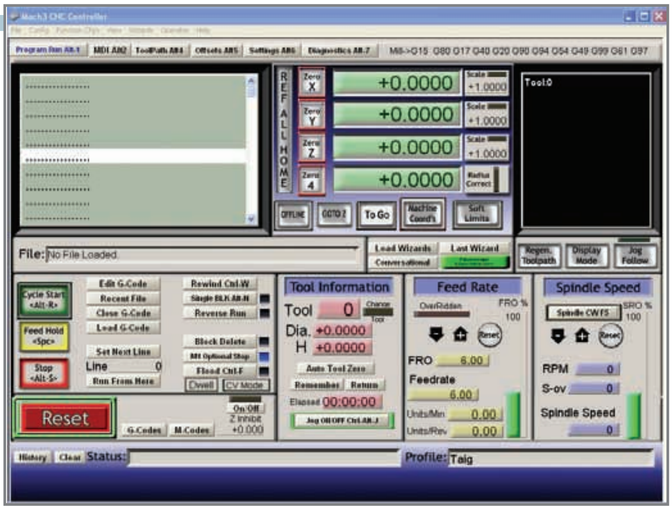


Photo 6—The Mach 3 CNC control software

was approached by a colleague to dispose of a German-made rack cabinet, which had housed a now-defunct scientific instrument. Examining it, I discovered that it was built entirely from aluminum extrusion, along with some cast aluminum corner assemblies. While all of the cabinet's extrusions were sized in metric, and not as versatile a profile as the 8020 extrusions, it was nevertheless possible to build the basic frame unit, as well as the base unit (which contains the controller and computer) completely from aluminum parts recovered from this cabinet. In the end, I found a Canadian supplier of the 8020 extrusion and fittings, but it was nice to be able to recycle all of this surplus aluminum (which is extremely energy-intensive to manufacture).

CNC STEPPER CONTROLLER

I originally tried a three-axis, 2.5-A stepper controller that I found in an online auction, but it did not work properly, so I returned it. I then did what I should have originally done. I ordered a Geckodrive G540, which is a well-respected American-made four-axis controller, handling up to 3.5-A (per phase) stepper motors, and power supplies up to 50 VDC. It has an advanced 10-microstep drive algorithm with mid-band resonance compensation, and operates well without a ventilation fan in my shop.

I purchased Keling KL23H276-30-8B stepper motors for the three axes. These motors provide 282 oz-inch of torque and are rated at 2.1 A per phase, which matched the controller I had initially ordered. The Geckodrive G540 controller handles these motors nicely, but since it is capable of providing 3.5A, it would have been better to have ordered the Keling KL23H284-35-4B, which are a bit more expensive, but provide 387 oz-inch of torque. Since I had chosen single-start Acme screws, the somewhat lower torque was not an issue in my case.

I powered the G540 using a surplus power transformer I had on hand, driving a full-wave rectifier and a large filter capacitor. Here again, this 30-VDC power supply provided the ideal voltage for the eBay controller, but I could have gotten better performance in my system had I used a power transformer that provided closer to the 50 VDC that the G540 is capable of handling.

I used limit switches at each end of all three axes to provide for overtravel protection. The G540 has numerous inputs for such limit switches, and I also incorporated an emergency stop switch, which shuts off all the motors in case something goes wrong. The G540 also has several outputs, one of which I used to drive a solid-state relay module which turns the spindle router on and off, under computer control.

MACH 3 SOFTWARE

All CNC machines require an intelligent controller of some sort. In the past, CNC was only used in professional machine shops and was implemented using dedicated control electronics, in much the same way that factory machinery is run by dedicated PLC circuitry. Today, with lower-cost CNC-ready tools available to smaller machine shops, it is much more common to see the dedicated controller replaced by a standard PC computer running CNC application software.

My "loaner" Taig mill came bundled with a licensed copy of the Artsoft Mach 3 CNC control software. The CD-ROM also contained a variety of allied programs, such as those that convert the output of various mechanical drawing programs into the "G-Code" used by Mach 3 and all other CNC controllers.

Photo 6 shows the Mach 3 program as it appears on my screen, although the layout can also be customized by the user. For a beginner like me, this software is much more than adequate to serve my needs, so it would be pointless for me to try and compare it to any other commercial and freeware programs that are available. I would say, though, that Mach 3 comes with far and away the best setup/operator's manual (in PDF format) that I have ever encountered. Setting up and configuring a CNC machining tool is not trivial, so read this manual thoroughly, early on!

While the purpose of a CNC controller is to run the

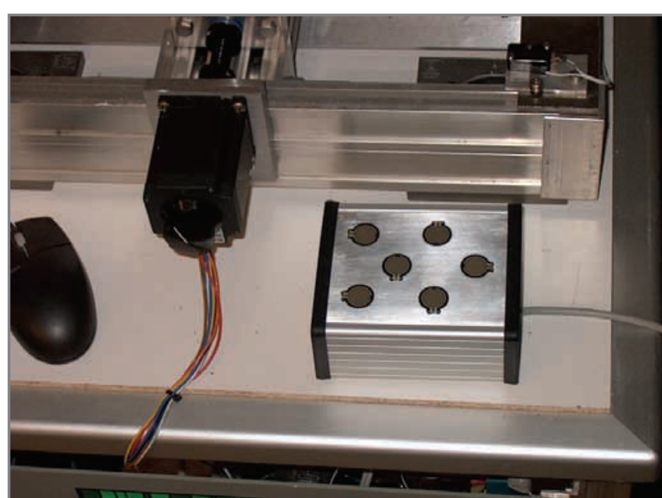


Photo 7—My PS/2 remote control pod, using six force-sensing resistors for the jog switches

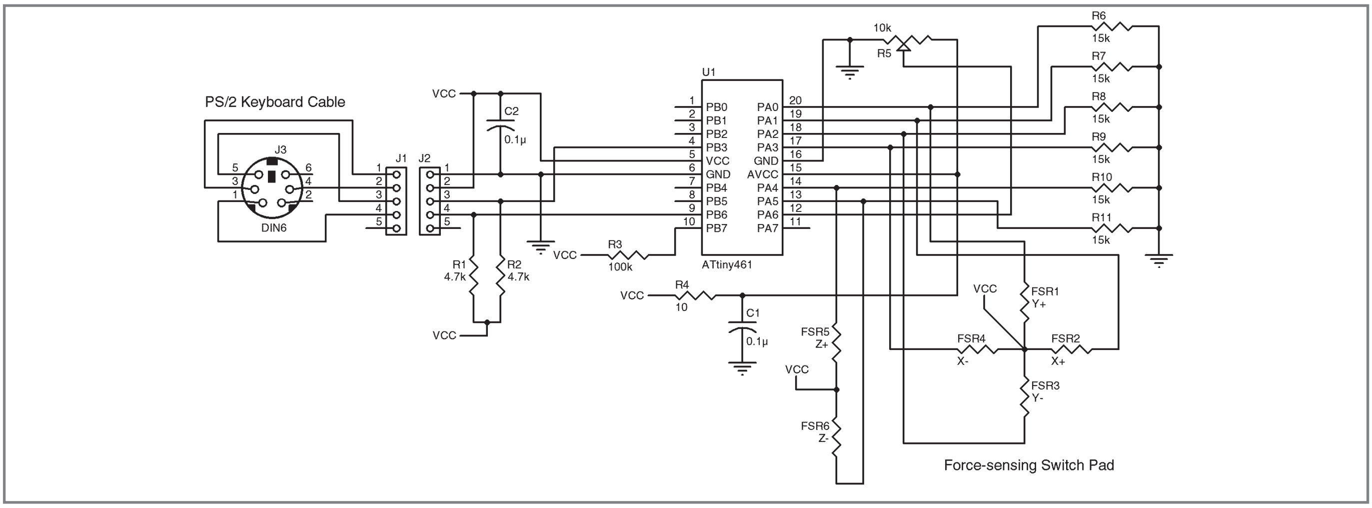


Figure 2—The remote pod is almost completely made up of the FSRs and the MCU.

tool under computer control, there still exists the need to manually position the workpiece on the machine initially, and to manually set spindle height. This manual control is called "jogging" in the trade. To accomplish this, Mach 3 implements a "keyboard and screen" implementation of a remote control pod. This is perfectly functional, but you have to place your computer keyboard/mouse close to the tool itself, making them susceptible to metal shavings.

I decided it would be more convenient if I had a small remote control pod to perform such manual positioning. While these are commercially available, I decided to design and build one of my own. Finally, some electronics!

EXTERNAL CONTROL

Many of the control "buttons" on the Mach 3 screen can be activated by keyboard hot keys, in addition to the mouse itself. The jogging "hot keys" I wished to implement in the remote pod are defined by default as follows: x motion (left and right arrow keys), y motion (up and down arrow keys), and z motion (page up and page down keys). You can redefine these (under Config, System Hotkeys menu), but these defaults make sense, so I stuck with them.

To implement the remote pod, it is mainly a matter of designing what is commonly called a "keyboard wedge." A "wedge" is a device that plugs into a PC's keyboard port and emulates a keyboard by sending out specific key scan codes in response to the user's actions on whatever input device the wedge uses. A true keyboard wedge allows for a "real" PC keyboard to be plugged into it, and merges both the "real" keyboard's data and the wedge's own data into one stream (hence, the "wedge" moniker). In the case of my remote pod design, I don't allow the "real" keyboard to be plugged into the pod, choosing instead to merely plug a standard USB keyboard into a free USB port.

What type of keyboard did I decide to emulate? The old PS/2 style or the newer USB style? Actually, the answer is both. I initially built a USB-style pod for the Taig mill. After I built my own CNC router, I needed another pod for it, and this time I chose the PS/2 interface, as it turns out to be much simpler and less expensive to design the pod this way.

The Mach 3 software is designed to use the standard PC parallel printer port for its machine-control interface. Basically, any PC that has a parallel printer port will also have a PS/2 port, although I grant that such computers are becoming less available over time.

SPECIAL SWITCHES

I put considerable thought into the type of switches I was going to use for "jogging" the three axes. If you're like me, you've been reading a lot about touch switches, touch screens, and the like. There was quite a personal temptation to incorporate such technology, since I am very impressed with my Apple iPod Touch and iPad and their excellent capacitive touch-screen interface. However, for "jogging" purposes, there were a few considerations which led me to favor another method. It would be advantageous if the jog speed were somehow proportional to how hard one pushed the switch—that is, if you "leaned" on the switch, the axis would rapidly move to a general target location, whereas a light touch could be used to "home in" on the specific target location. When using Mach 3's screen-based remote pod, pressing the Shift key prior to hitting any of the jog motion buttons switches to a "rapid" speed. This feature can easily be incorporated into the remote pod, allowing for slow/fast axis motion.

Since there was a distinct possibility that metal shavings from the cutter could end up on the remote pod, I felt that there would be a good chance of false triggering if using touch-based technologies. You definitely don't want false triggering on a CNC machine!

What I settled on was a sensor known as a force-sensing resistor (FSR), manufactured by Interlink Electronics. These FSRs come in various shapes and sizes, but the model I chose is a circular pad, about 0.75" in diameter, as seen in Photo 7. These are rugged devices, with a sticky backing that adheres nicely to a metal panel. The sensor connections are made via a standard flex circuit, which can be easily bent at right angles to the pad, and made to exit via a small slot in the panel.

With no force applied, these sensors have a resistance greater than 1 MΩ. When pressed with one's finger, this resistance decreases, reaching down to about 2 kΩ with a firm finger press. Beyond very low forces of about 20 g, the device responds in an inverse power-law relationship.

These sensors make ideal motion-control switches for the CNC remote pod. They are immune to contamination from metal shavings or cutting fluid. They don't suffer from false triggering problems, as long as one sets the pressure threshold sensitivity properly. In small quantities, they cost about $7, which is about the same as any good-quality mechanical switch (which would not have the touch sensitivity).

In this application, I wired each sensor up as the upper resistor in a voltage divider, and fed the divider output directly to one of the eleven 10-bit ADC channels contained in an Atmel ATtiny461 MCU. With no pressure applied to the FSR, the ADC will see virtually 0 V. A light finger press will result in about 0.5 V going to the ADC. Finally, with a heavy finger press, the ADC voltage will exceed 3.4 V, which is the threshold, fixed in firmware, that switches over to high-speed axis motion.

I use one of the remaining ADC channels to sample the wiper of a pot that is connected across the VCC supply. You adjust this pot to represent the minimum FSR pressure needed to activate the "low-speed" axis motion, which is a personal preference.

THE CIRCUIT & FIRMWARE

Figure 2 is a schematic of the PS/2-based remote pod, which is pretty straightforward. Apart from the six FSR sensor switches, wired as voltage dividers to six of the MCU's ADC channels, there isn't a whole lot more to the circuit. The pod is connected to the PC's PS/2 keyboard port using a cable that I salvaged from a surplus PS/2 keyboard. The four-wire PS/2 interface consists of +5 V and ground wires, as well as a clock and a bidirectional data line. The remote pod uses very little power and is easily powered from the 5-V keyboard power supplied by the PC. Both the data and clock lines are pulled up to VCC by 4.7-kΩ resistors. The Atmel ATtiny461 MCU contains an internal 8-MHz RC clock, which is accurate enough for this application, so no external crystal/resonator is required.

The firmware is written using the Bascom-AVR BASIC compiler. It consists of two main sections: the FSR sensing loop and the PS/2 keyboard emulation routines. I used an add-on PS/2 keyboard emulation library, available for Bascom AVR, rather than trying to roll my own.

The FSR sensing loop is fairly straightforward. To start with, you read ADC channel 6, which is connected to the pressure threshold pot. Whatever value you see here is what you use to compare the six FSR readings with to determine if any of them has been pressed hard enough to be considered a valid slow-speed trigger event. Should a particular FSR provide a voltage to its respective ADC channel that exceeds this threshold, a SELECT-CASE structure is executed, which sends out the proper key scan code to the PC (preceded by the "make" scan code).

Apart from the six FSR sensor switches, wired as voltage dividers to six of the MCU's ADC channels, there isn't a whole lot more to the circuit. The pod is connected to the PC's PS/2 keyboard port using a cable that I salvaged from a surplus PS/2 keyboard. The four-wire PS/2 interface consists of +5 V and ground wires, as well as a clock and a bidirectional data line. The remote pod uses very little power and is easily powered from the 5-V keyboard power supplied by the PC. Both the data and clock lines are pulled up to VCC by 4.7-kΩ resistors.

If a scan of all six ADC channels results in no readings above the pressure threshold, then no buttons are pressed. In this case, you send out a "break" scan code followed by the scan code value of the last key that was sent out. The preceding routine handles the slow-speed axes' motion.

In the case of a firmer press on any of the FSRs, its respective ADC channel will return a much higher value. I empirically determined that an ADC value of 700 corresponds to a pressure that is considerably greater than the slow-speed threshold, but it is not so firm as to make it uncomfortable to achieve. Therefore, in the aforementioned loop, when the six ADC channels are being polled, the value is first checked to see if it exceeds 700. If it does, then we have to send out whatever keystrokes are necessary to invoke high-speed motion on the proper axis. Here it gets a bit more complicated. In Mach 3, to jog at high speed, you must first release any jog key that you may have depressed, next hold the shift key down, and then, with the shift key still held down, press the desired jog key.

To mimic this in my program, whenever I sense that an FSR has been pressed hard, I first check to see whether the last time this FSR was checked, if it was pressed hard, or just gently. If it turns out that the last time it was just pressed gently (usually the case), then I first send out the proper scan codes to indicate that the applicable key has been released. Then I follow up with the scan code for the shift key. Lastly, I send out the scan codes to indicate that the same key (shifted) has been

struck again. The firmware uses just 1,410 bytes of flash memory, so it would be possible to substitute the lesser ATtiny261 MCU.

Photo 7 shows the PS/2 remote pod mounted in a Hammond 1455N1201 extruded aluminum enclosure. These work particularly well because the top panel slides in and out, facilitating assembly. The top panel and the circuit board have to be "sandwiched together" due to the way the FSR flex connectors are mounted to the PC board.

THE FINISH LINE

This turned out to be my most involved DIY project in recent memory. It was unlike any of my usual electronic projects. When building custom, small-quantity electronics projects, I often find it practical to "design it as I build it." I regularly use hand-wired Vectorboards and chip sockets. And with much of the functionality embedded in the MCU firmware (which is crafted last), the method generally works out well.

In the case of a precise mechanical tool such as the CNC router, it's imperative that you do a whole lot of planning well in advance of even ordering the necessary parts. Knowing exact dimensions are critical, as most of the metal is ordered cut to size. Details like thread size and component strength are essential. Also important are sticky details such as whether your design will enable you to place tools in the necessary spots to tighten the bolts sufficiently. Undoubtedly, I spent more time researching this endeavor, before even ordering a single part, than I routinely spend building an entire electronics project.

This project also made me much more aware of how pampered the modern electronics enthusiast is when it comes to parts supply. A typical electronics project BOM can be fulfilled from one or two huge distributors, drawing from their 1-million-part inventory. They are happy to sell you one or 1,000 of a specific component, and you can expect delivery in a day or two for a very low courier fee.

In this project, I had to ferret out numerous suppliers, many of which were small web-based businesses that had much longer delivery times and accepted only PayPal. I also learned to pay attention to the invaluable advice available from the CNCZone forum, which is home to many professional machinists with expertise in techniques, as well as practical experience regarding many of these smaller vendors. In the two cases where I struck out on my own and ignored advice on the forum, I was either disappointed or just plain ripped off. So, *caveat emptor*! Overall, though, working on this project was an extremely enjoyable experience.

Author's note: Go to the Circuit Cellar FTP site to download a complete list of companies that manufacture and sell the motor controllers, mechanical parts, and specialized fasteners I used for this project.

Brian Millier (bmillier1@gmail.com) runs Computer Interface Consultants. He was an instrumentation engineer in the Department of Chemistry at Dalhousie University (Halifax, Canada) for 29 years.

PROJECT FILES

To download a manufacturer list, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/248.

RESOURCES

"DIY 80/20 Aluminum Extrusion CNC Machine," www.8020CNC.com.

DIY CNC Machinist Forum, www.cnczone.com.

M. Ribble, "My CNC Engraver (Part 1)," Hobby Robotics Blog, August 3, 2008, www.glacialwanderer.com/hobbyrobotics/?p=17.

SOURCES

Mach 3 CNC Control software
Artsoft | www.machsupport.com

ATtiny461 MCU
Atmel Corp. | www.atmel.com

Force-sensing resistor
Interlink Electronics | www.interlinkelec.com

by George Martin (USA)

The Project-Ready Designer

A Refresher on Project-Essential Concepts

Project management should involve more than creating a bill of materials and then soldering and testing until the project deadline. You should also frequently review design essential topics, reconsider all tool options (from hardware to languages), and even attend conferences to stay informed on new technology and design techniques.

I often jot down and then store my thoughts on essential design-related topics until I have enough to write an entire column. Project preparedness is one such topic. Well, the time to focus on this subject is now.

SCHEDULING & SIMULATION

For a long time, I'd been capably scheduling both simple and complex projects, doing either manually or with project management software. But what I'd been missing was the ability to work with that schedule to absorb the real-world issues that always arise. I now have a solution.

As a *Circuit Cellar* columnist, I'm frequently offered books to review. Most are very detailed and specific on a topic and, while good books, they don't necessarily cover things I can talk about in this column. But recently, I was given Claudia Baca's book, *Project Management for Mere Mortals*, which is about project management and, more specifically, project scheduling. The book covers project scheduling, handling problems, and keeping bosses and customers satisfied. It's a good book even if you don't schedule projects for a living. It will give you some valuable insight on how to run projects.

I use simulation quite often. It's a must for complicated projects, and it even helps to document what's going on with simple projects. On the messaging system I'm working on, the simulation shows the gain and bandwidth for each of the system's functional blocks. It's usually more time consuming to enter the schematic than it is to run the simple simulation and record the results. But, it helps me to talk to the customer and work through difficult problems. For larger, more complicated systems, it's great to see where marginal operation might be lurking. I use a Linear Technology simulator. It's available for free, as in free beer. Other vendors offer similar products.

My son-in-law is a manager for a construction project. It's a five-story lab building at Stanford University. The drawings for the building are designed and available only in CAD format (paperless). The construction schedule contains between 2,000 and 3,000 line items. They have a program that lets you select several views of the project (even sections through the building) and then it runs the schedule against the CAD drawings. The output is a time lapse picture (model) of the building as it's built per the schedule. On the first run, they found some items that appeared in the output too soon. For example, a light fixture was hung before the conduit (or even the wall) was in place. Looking back into the schedule, they found that these items did not have the proper linkage to predecessor tasks. After a simple change on paper, the output looked great. Imagine running a project with hundreds of workers, and someone trying to hang a lamp before there was a wall to support it. It makes many of our problems sound trivial.

STATE MACHINES

I recently wrote a column on state machines. The editors at *Circuit Cellar* marked it up and reported that I wasn't using the term state machine properly. What I was doing was really a command processor. They were correct, so I changed my article to reflect their input. Well, after the article was published I got an e-mail from the fellow who actually did the closing ceremony for the recent Olympic Games in Canada. Of course, he liked the article and said that was exactly how he did state machines except for a slight difference in how he implemented the timers. He went on to explain his work represented the largest CAN network ever implemented. I had quite a chuckle.

I suspect that state machines were invented and implemented in hardware with rather rigid formal definitions. Now that us software folks have gotten our hands on the topic we've bent the rules (probably actually broken most of them) to suit our purposes.

I use simulation quite often. It's a must for complicated projects, and it even helps to document what's going on with simple projects. On the messaging system I'm working on, the simulation shows the gain and bandwidth for each of the system's functional blocks.

CODE VALIDATION

What is an embedded system? A good (useful) definition I recently heard was that an embedded system is a system where the original programming never changes. So the system has one purpose and, setting upgrades aside, it was never repurposed. Seems like a useful definition.

I use Source Publisher and Understand from Scientific Tools. I use Source Publisher more because it seems I can generate code (both typing and cut and paste) faster that I can keep track of that code. So, the nicely structured printout and highlighted nature of the output helps me keep up.

Understand, is quite a different product. If you've got a large legacy system written in C, you can give that to the program and you will get a dictionary of variables and procedures. Also, for a given routine you'll get who and where it's called and which routines are called by that routine.

Another feature of Understand is it can generate flow charts directly from your C code. This became most useful on a recent medical project. The task was to replace an obsolete micro (National Semiconductor COP8) with a more current device (Renesas M16C28). I did that, and moving C code from CPU to CPU was not all that difficult. It was the assembly code that caused all the gray hairs. It's not easily ported.

For that same medical project, the code needed to be validated. And that meant testing the modules against a flowchart. Previously, the company drew the charts by hand. Well, with Understand, I could just press a button and I had a PDF representing the flowchart for the procedure. This feature alone saved several man-days of work right at the end of the project when everyone was pushing for results.

DEVCON 2010

I attended DevCon hosted by Renesas last Fall. This is a technology get-together between Renesas engineers, companies that support Renesas devices (as in RTOS vendors), and end users like me and you (*Circuit Cellar* readers). No sales personnel were allowed. This four-day conference involved lectures and classes during the days, and meals and keynote speakers in the evenings. I believe it is held every two years, and if you have a chance to attend one, it's well worth the time.

I sat in on the TCP/IP, USB, capacitive touch, and RX610 evaluation board presentations. One of the sessions was about software design. Marketing studies report C is the predominate language for designing embedded systems. Some say about 25% to 30% of us are using C and the percentage is growing. C++ comes in less popular, and its usage is dropping over time. The other fact is that the systems are getting more complicated and time to market is shrinking. That shouldn't surprise you. So, I asked all the experts on the panel just how we were going to accomplish that one. Was it to just work longer hours? They didn't have a ready answer, but they did suggest the use of an RTOS with all of its support and automatic code generation from UML and other sources. IAR was on the panel, and they offer such products. Check out their state machine tools to lessen the burden of code generation.

UML

This brings me to the Unified Modeling Language (UML). I use UML as a drawing tool. It helps me think through problems and present information to others. I have talked about that before and suggested you look into getting started using UML. I use No Magic's MagicDraw package. It has a basic package for a single-user personal version, as well as an enterprise version. Several readers have asked me if they could upgrade from the personal version later for the difference. I was sure they could and asked the factory, which said "yes," within limits (like the same version, in a reasonable amount of time).

My assignment for 2011 is to upgrade and try the automatic code generation feature. Since I'm using the diagramming portion of the product, why not go all the way? I'll keep you posted.

C CODE

Several years ago, I talked *Circuit Cellar*'s founder, Steve Ciarcia, into letting me write a column on C language. Since we are friends who have worked together for several years, he trusted that I had something to offer. Well, solely as a result of my efforts (major joke here), check out the quality of the code and flowcharts in the latest issues. In all seriousness, C has become the primary language for coding, the flow charts show improved organization, and structure and the scope of projects has grown. Sophisticated libraries are now matter-of-factly referenced in articles. Good job, readers!

WHAT'S NEXT?

That brings me to the end of this month's article. I think that it will just

be a waste of paper and ink for me to continue to focus on C code in my articles. By now I've covered all you need know to get a project up and running in C. I have not talked about pointers to arrays of functions, but you have the foundation to do that on your own. If you've followed these articles closely, you might not have your black belt in C, but you should be well on your way.

Next, I will start presenting complete projects. I'll start simply, with a specification and interaction with the user. I'll then move on to system design, hardware/software design, prototyping, and testing. I hope some of the projects will interest you.

I've threatened to design a spring tester because I need one. But we just recently sent a show dog off to be bred and I thought about recording the temperature, pressure, acceleration, and such of her crate as she travels on the airplane. And then, with all the fuss about body scanners at the airport, how about a radiation monitor? I also picked up a Texas Instruments MSP430 watch development platform. That has potential.

I compete in the shooting sports, so perhaps there's a project hidden there. Or, how about a radio that functions like a VCR recording your favorite programs?

As you can see, our plate is filling up fast. If you have anything that piques your interest or curiosity, drop me a line and I'll add it to the list. Until then, keep all your `#defines IN_CAPS`, make all your tasks the same priority, develop a style for your C code, and acquire and release your shared resources in the same order across all your tasks (alphabetical order is a good one). If that didn't make sense, stayed tuned. ■

George Martin (gmm50@att.net) began his career in the aerospace industry in 1969. After five years at a real job, he set out on his own and co-founded a design and manufacturing firm (www.embedded-designer.com). His designs typically include servo-motion control, graphical input and output, data acquisition, and remote control systems. George is a charter member of the Ciarcia Design Works Team. He is currently working on a mobile communications system that announces highway information. He is also a nationally ranked revolver shooter.

RESOURCES

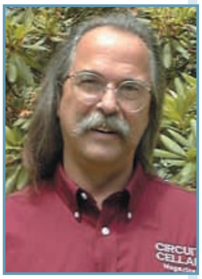
C. Baca, *Project Management for Mere Mortals*, Addison-Wesley Professional, 2007.

SOURCES

Magic Draw
No Magic, Inc. | www.nomagic.com/dispatcher.php

Understand software
Scientific Toolworks, Inc. | www.scitools.com

by Jeff Bachiochi (USA)



Direct Line Connection

When You Don't Require Line Isolation

If you're using a low-current microcontroller and your circuit's total power consumption is minimal, it doesn't make sense to implement a power supply that might consume more current than the circuit requires. Here you learn how to use a linear shunt regulator for low-current applications.

With today's low-current micros able to chug out the MIPS at a handful of milliamps, total circuit consumption can often remain low. It seems like a waste to use a power supply that can eat up more current while idling than your circuit requires. You can steal small amounts of current directly from the AC line, and you can do so without bulky transformers or having to worry about heat dissipation from those series resistors normally required to drop the voltage down to a usable level. In this article, I'll explain how Supertex has improved on the idea of using a linear shunt regulator like a Zener diode for low-current applications.

The circuit in Figure 1 has been used for years to regulate small currents. Once the Zener diode has sufficient reverse (leakage) current flowing, its voltage drop will remain constant until too much current destroys it. The Zener diode's wattage determines the maximum current through it, while the leakage current is the minimum current to sustain regulation. As long as the Zener diode current remains within these limits, the voltage across the Zener diode will remain regulated to the manufactured specification. Zener voltages are available from 1.2 to higher than 200 V! Therefore, a circuit placed across the Zener will be held to that Zener's voltage as long as the aforementioned rule is not broken. To use this type of circuit, you need to know the additional circuitry's current requirements. A fixed current requirement is the easiest to design for. When placed in parallel with the Zener, the series resistor is sized for the combination of the Zener current and the additional circuit's current. However, if the additional circuit current will vary, the current through the

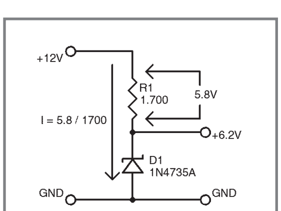


Figure 1—Providing a Zener diode with a minimum of reverse current produces a constant voltage drop equal to its Zener voltage. The remainder of the source voltage must be dropped across its series resistor and the resistor's value sets the current flowing through the Zener. Any load placed in parallel with the Zener will have the Zener voltage as its source and will share the series resistor's current with the Zener.

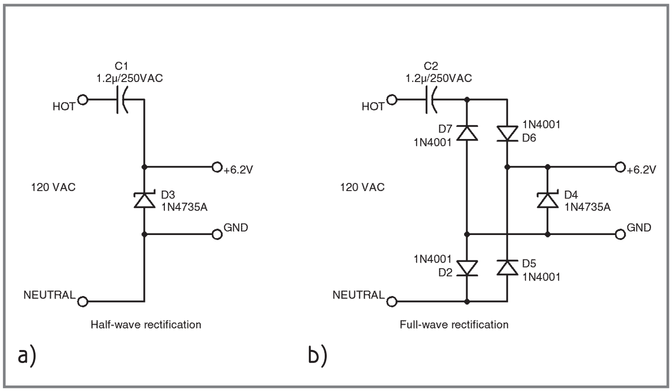


Figure 2a—By substituting an appropriate (AC-rated) capacitor for the series resistor, the Zener current is regulated by the capacitive reactance and the Zener diode will regulate it during the positive half (wave) cycles of the AC source. **b**—The addition of a full-wave bridge allows the Zener diode to regulate during both half cycles of the AC source.

Zener must vary in a complementary way because the series resistor current remains constant. When the current through the additional circuitry is minimal, make sure the maximum Zener current isn't exceeded or the Zener will burn up. And when the additional circuit current is maximum, make sure there is still a minimum current flowing through the Zener or it won't regulate properly.

The voltage across the series resistor will be the difference between the supply and the Zener voltage. The current through it is the total of the Zener current and the additional circuitry's current. Refer again to Figure 1. The 1N4735A is a 6.2-V Zener diode. When connected to a 12-V supply, about 6 V will be dropped across the series resistor. At a current of 50 mA, 0.3 W of heat must be dissipated (i.e., 6 V × 0.05 A = 0.3 W). When connected to a 120-V supply, 114 V will be dropped across the series resistor, and at a current of 50 mA, that's 5.7 W of heat that will need to be dissipated (114 V × 0.05 A = 5.7 W). That's about as hot as a Christmas tree-sized incandescent night light! Using a 10-W resistor to dissipate this heat is bulky, taking up valuable real estate, and will potentially require you to handle ventilation issues.

This discussion is about connection to the AC line voltage. Since the source voltage is varying (AC), we have a trick we can use to cut down the wasted energy we drop across the series resistor. In the aforementioned case, the resistor is 2,200 Ω (i.e., 114 V/0.05 A = approximately 2.2 kΩ). We can replace the series resistor with a series capacitor where capacitive reactance is as follows: (Xc) = 2.2 kΩ. Since the Xc is imaginary, there is no dissipation. By rearranging this familiar formula for finding the capacitive reactance of a capacitor (i.e., $Xc = 1/[2\pi fc]$), we get $C = 1/(2\pi f \times Xc)$, and we can determine what capacitor will have a capacitive reactance of 2,200 Ω (i.e., $1/[2\pi \times 60 \times 2{,}200] = 2\ \mu F$). But let's not get ahead of ourselves here. Note that we need to add a few more components to the basic circuit.

AC/DC

We need to rectify the AC line voltage to change the alternating voltage of the source into rectified DC. Figure 2 shows the substitution of a capacitor for the series resistor in the original figure (half wave) and the addition of 1N4001 diodes for full-wave rectification. Previously, we found that the Zener and the load share the circuit current. When the load is not using any current, the total design current flows through the Zener diode. In this case, this is wasted power. Adding a switch to the input is one way to eliminate this waste. When the load requires no current, the switch shorts the rectified output. When the load requires current, the switch opens and allows current to flow into the load.

In Figure 3 this switch is realized using an FET and a diode. The diode prevents current stored in the circuit's output capacitor from discharging through the shunt FET when shorted to circuit ground. All that's necessary now is some device to control the FET so the load remains at a specified voltage. Enter the Supertex SR10.

The SR10 contains an internal FET switch and the control circuitry to maintain a regulated voltage output (V_{OUT}) to the load. An internal voltage divider provides three preset taps for 6 V/12 V/24 V regulation. Connecting one of these taps to the feedback pin provides an input to the SR10's comparator which controls the FET's state. The user may substitute an external voltage divider to obtain a V_{OUT} other than one of the available internal presets.

The SR10 uses synchronous switching—that is, the FET only turns on when the source voltage is low (less than V_{SYNC}, in sync with the zero crossing). This means that the current may not get diverted from the load for some time after the V_{OUT} has exceeded its regulation point. This produces a bit of overshoot to the regulated output. You can minimize overshoot by sizing the output capacitor correctly. This is analogous to minimizing output ripple.

Note that the series AC capacitor will remain charged when the source voltage has been removed. This situation exposes the user to a shock hazard at the plug. A large value resistor placed across the AC capacitor will safely

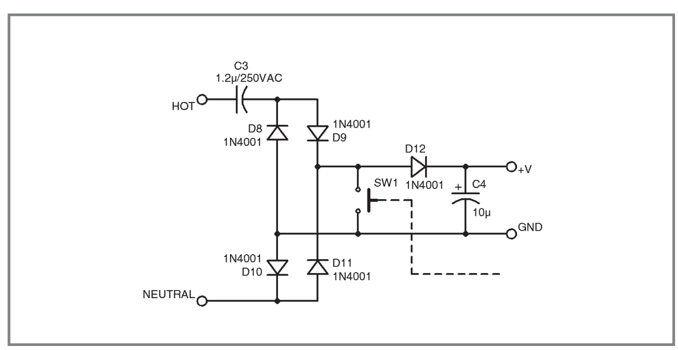


Figure 3—If you add a switch, diode, and output capacitor, surplus input current can be eliminated by shorting out the source. The diode prevents the output capacitor from discharging during this time.

Photo 1—Here is a typical scale-model lighthouse used for decorative landscaping. These are becoming popular as are scale-model windmills.

bleed off this charge without adding any significant power loss. While this AC capacitor self limits short circuit current, the input can be fused should you require catastrophic failure protection.

HOT CHASSIS

While the neutral side of a two-wire AC main is supposed to be at ground potential, this can actually be at some potential other than ground and presents a safety hazard, so one shouldn't depend on it being at ground. Never remove or circumvent the third wire in a three-wire power cable. Many old nonpolarized two-wire appliances used a metal chassis connected directly to one side of the line and, depending on the rotation of the plug, might be at a hot or neutral potential. If the nonpolarized plug is plugged in so the hot wire is connected to the appliance chassis, any exposed chassis screw creates a safety hazard. Thus, the reason for making two wire plugs (and outlets) polarized by widening the neutral prong. While this is a good attempt at preventing disorientation, the addition of a third prong assures a completely separate earth ground.

If you refer to the half wave rectifier in Figure 2a, you'll notice that the load is connected to neutral. When using a full wave rectifier as in Figure 2b, the load is not at a neutral potential. It is a diode drop above the neutral wire (ground). It is dangerous enough working with line voltages, so take extra precaution and prevent damage to your test equipment by paying attention to what is and isn't ground. I strongly suggest using an isolation transformer between the AC line and your circuitry if you are planning to probe around with a grounded scope.

PROJECT POWER

I was looking for a way to power a small circuit I designed to mimic the rotating beacon of a model lighthouse (see Photo 1). The circuit uses a ring of high-power white LEDs to simulate rotation without actually having any mechanical movement. To give a smooth rotational appearance, each LED is driven by its own PWM.

I started this design by deciding what the maximum size of the ring of LEDs could be. For a 4′ model of a lighthouse, somewhere around 2″ is a reasonable ring size. I can use LEDs with body diameters of from 3 to 10 mm (approximately 0.1 to 0.4″). The circumference of a 2″ diameter circle is approximately 6.3″. If I place 10-mm-wide LEDs around the perimeter of a 2″ disk, I could almost fit 16 LEDs (i.e., 6.3″/0.4″ = 15.75″). Using these would increase the outside diameter of the assembly by two times their height, 13 mm, for an increase of around 1″ in diameter. If I wanted to use T1-3/4 LEDs (5.6 mm), the circumference would be reduced to 4″ (16 LEDs × 0.25″ = 4″) or a diameter of about 1.3″. With this smaller (diameter) disk, I could always mount the 10-mm LEDs by spacing them away from the disk's edge.

If I kept the design fairly modular, adding or subtracting a few LEDs would be easy. I picked a SOIC-28 pin Microchip Technology PIC18F2431. While the SSOP-28 is almost twice as small, that also means the pin pitch would be 0.67 mm. This is really tough to hand solder. So, since I wasn't forced into the smallest size by lack of real estate, I went with the easiest to assemble. As you can see in Figure 4, I used the familiar five-pin ICD header to allow using the debugger/programmer. It is mounted on the underside of the PCB below the PIC and offers a convenient way to power the circuit from the SR10 circuitry, which will interconnect through this header (more on this in a bit).

To keep with this modular idea, I needed a way of assigning each PWM to a particular bit number on a particular port. This way the same code can be used for each PWM. To do this, I made extensive use of indirect pointers. Each LED therefore has its own set of registers, a PWM value, a PORT address, a PORT pin bit mask, and a position mask for the FLAG register. The FLAGS register holds a direction indicator (bit) for each

Heat dissipation is a concern for any product. In the case of stealing small currents, you may have to get rid of 5 to 10 W of wasted power if you don't take some measures to eliminate it. The SR10 incorporates everything you need to produce a compact line operated small current source.

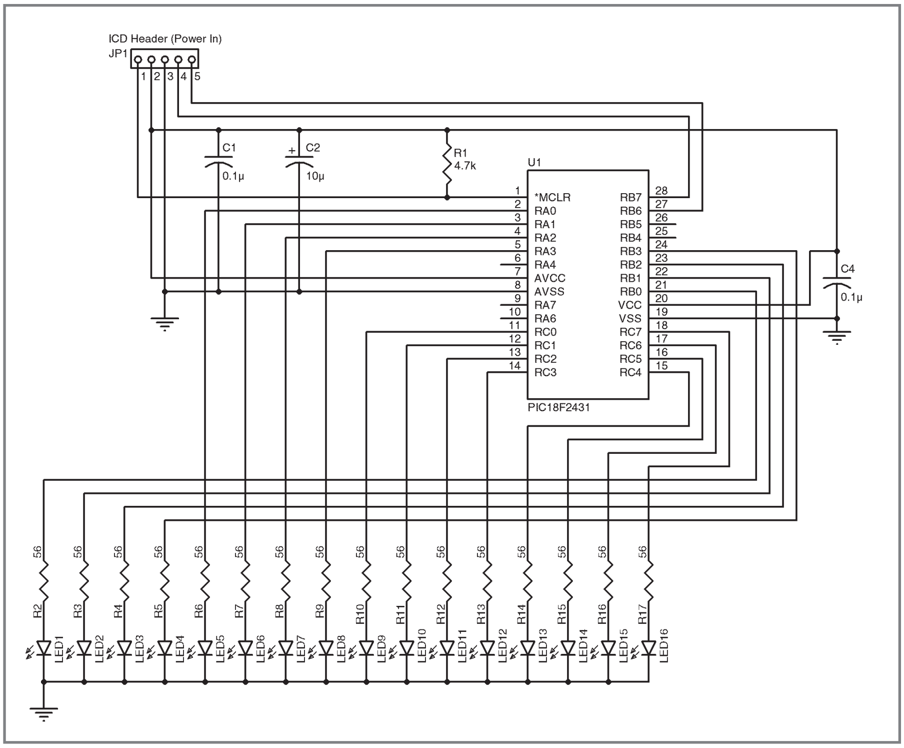


Figure 4—The rotating LED beacon contains 16 LEDs driven off of the PORT bits of a small SMT microcontroller. The total operating current for this circuit is 15 mA.

LED. With 16 LEDs, two byte registers are required with direction bits for LEDs 1 through 8 in the first byte and for LEDs 9 through 16 in the second byte (see Figure 5). These FLAGS are the keys to how this works. If a bit "x" in the pattern generator is 0, then the Ramp routine will reduce the PWMx value for LEDx by one until it reaches zero, in which case it will remain at zero (value of 0 = 0% PWM). If a FLAG bit is 1, then the Ramp routine will increase the PWMx value for LEDx by one until it reaches 255, in which case is will remain at 255 (value of 255 = 100% PWM).

Using 256 values means that PWM resolution is 8-bit or 256 ticks to be able to present all of the states between 0% (OFF) and 100% (ON). The time for each tick is based on the amount of time it takes to execute the DoPWM routine. The DoPWM routine compares each of the PWMx values to the present tick counter PHASE. LEDx is turned OFF unless PWMx is greater than PHASE, and then LEDx is turned ON. To go through this once for all 16 LEDs requires 185 µs. This means it will take ~48 ms (256 × 185 µs = 47,360 µs) for a single loop of the PWM PHASE cycle.

With a resolution of 256 bits, a ramp up (or down) will require approximately 12 s (i.e., 48 ms × 256 = 12,288 ms). Yup, that's right over

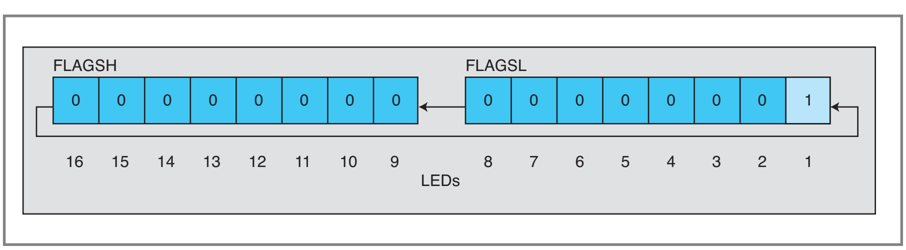


Figure 5—Two 8-bit registers create a pattern generator. Each LED will illuminate when its associated bit is a 1. A 0 will dim the LED. By periodically shifting all of the bits, every LED will have its chance to turn on.

Version	HEX (file size)	DoPWM (time µs)
Indirect pointers with modular code	2K	185
Straight-line code	3K	59

Table 1—While these results show the typical reduction in code size and the increase in execution time by making use of loops and modular code, it might not be the best direction to take in every application.

12 seconds. Since we want all 16 LEDs to go through this cycle, that brings the rotation to over 3 minutes (12 s × 16 = 192 s). No lighthouse I know of has a rotation cycle of minutes. A more suitable time would be no more than a few seconds.

As it turns out, humans require about an 8% change to the relative brightness to perceive a change has occurred. At 1-bit resolution (two states, one change) the change in brightness is 100% (i.e., 100%/[states-1] = 100%). For 2 bits of resolution (four states, three changes) that would be 33% change per bit. For 3 bits, it's 14% (i.e., 100%/7) and 6.6% for 4 bits. Assuming linear changes, 4 bits is all that's necessary to fool the human eye into squashing together the incremental change over time and have it perceived as continuous.

If we take the preceding timings and recalculate for 4-bit PWM resolution, we get some new numbers. The PWM PHASE cycle now takes 3 ms. The ramp time now requires 48 ms. This produces a rotational cycle of 768 ms. Less than once per second is too fast. Instead of using the execution of the program to determine the ultimate loop time, we can insert a timer into the loop. Ordinarily, you might use this approach to assure precise control of a loop. In this case precision isn't the goal. By increasing the timer reload value, we can increase the loop time and slow down the total rotational cycle time to the timing we're trying to emulate. The minimum time here is approximately once per second. This can easily be set to anything between 1 and 10 seconds by using larger reload values.

Indirect pointers can be useful. While they allow one to use data in a clever and flexible way, they add a level of complexity to code that can make it difficult to understand and debug. I challenged my ability creating this modular code. Thankfully, the application was simple enough that it didn't involve too many hours of cleverness. When finished, I had to sit back and ask: Was this complication worth the trouble? To help me answer this question, I decided to recode the application using neither modular code nor indirect pointers. I used just plain straight-line code. Table 1 is a comparison of the two versions of code. The straight-line code was quicker to write, needed less debugging, and was far easier to understand than the modular code. This is a perfect example of keeping things simple.

CURRENT SOURCE

Designing with the SR10 is simple. While there are some decisions you need to make about which additional components you want to include to complement your application, component selection is common for all circuit varieties, except in the selection of the series AC capacitor's value (see Figure 6). You can avoid math if you take advantage of the C_S selection chart found in the SR10's datasheet. Film capacitors are often used in high-voltage AC circuits because they have a lower ESR, which minimizes internal heating. The large physical size of these capacitors makes them an unsuitable candidate for SMT.

I used the same PCB form factor for this power supply circuit as previously used for the LED disk. It uses a similar diameter round PCB with alignment mounting holes and a mating connector. Most scale model lighthouses built for outdoor use have a standard (Edison) E26 candelabra base socket for an incandescent bulb. I removed the (burnt out) guts of a compact florescent bulb. Is it my imagination or do these last fewer hours than an equivalent incandescent? The base of this bulb will become the base of my rotating beacon module. I want to enclose all the circuitry into a sealed unit for two reasons. First, this will insulate all dangerous (line level) connections from the user. And, since it will be used outside, it needs to have some protection from the elements.

BULB HOUSING

You might recall my August 2009 article in which I described how to build a threat-level display ("Threat-Level Indication System," *Circuit Cellar* 229). I demonstrated how to use an FTDI FT245R USB parallel port as a PC-connected output device. To build the stacked color-coded display I shopped the grocery aisles for a suitable container. This same container made a perfect protective case for this project. The candelabra base is firmly attached to container's cover. I drilled a 1″ hole in the cover, stuck the base through from the inside, and used quickset epoxy glue to hold the base onto the cover.

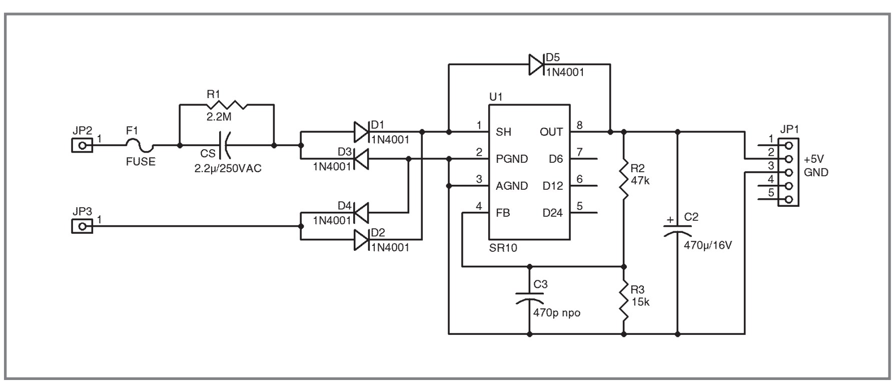


Figure 6—The SR10 circuit requires few external components. Most components including the optional fuse and bleeder resistor can be SMT parts. The large physical size of C_S may require using a through-hole part.

Photo 2—This project can replace the standard light bulb used in a scale model lighthouse, as shown in Photo 1. This gives these works of art a more realistic source of light for guiding lost wanderers home to port. Can you say, "Fog horn?"

The SR10 power supply PCB connects to the AC leads coming up from the candelabra base. It fits in the candelabra's plastic shell and provides 5-V power to the LED PCB through two pins of its solder-side ICD connector. Photo 2 is the finished project. At first, I tried a translucent cover by sanding the plastic surfaces of the container to reduce the point source nature of the LEDs. But in the end, I determined the clear cover offered higher visibility, especially from a distance. For the best results, use an LED that has a viewing angle of at least 360°/LED count, or in this case 22.5°.

HEAT AVOIDANCE

Heat dissipation is a concern for any product. In the case of stealing small currents, you may have to get rid of 5 to 10 W of wasted power if you don't take some measures to eliminate it. The SR10 incorporates everything you need to produce a compact line operated small current source.

Supertex offers the SR10DB1 (demo board) for those who wish to play with a PCB of the SR10 circuitry. The series parts are socketed, so you can easily substitute components of various values. There are jumper selections for full/half wave as well as output voltage.

WHO KNOWS?

I'd like to thank all the readers who've responded with suggested manufacturers for the old electro-optical device I mentioned in my November 2010 article, "Recharging Portable Devices: A DIY Power Adapter Design" (*Circuit Cellar* 244). It seems there are a lot of readers out there who are experts in minutiae. So, I thought I'd make this a continuing exercise for those of you who might be privy to the answers of some curiosities. I'll call it: "Who knows?"

Let's start off with a question that popped up during the background work phase of this project. Who knows where the "T" designation (i.e., T – 1¾) came from when used to describe the size of LEDs and small incandescents? ■

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@imaginethat now.com or at www.imaginethatnow.com.*

PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2011/248.

SOURCES

PIC18F2431 Microcontroller
Microchip Technology | www.microchip.com

SR10 CCSS Regulator
Supertex, Inc. | www.supertex.com

by Tom Cantrell (USA)

Time Traveler

Embedded Timekeeping and More

Imagine if electronic gadgets couldn't keep track of time. Wow, that would put a crimp in our style. Fortunately, with a little help from a quartz crystal (or, these days, perhaps a silicon MEMS equivalent), there are chips that can keep on ticking while they take a licking.

The datasheet for the Microchip Technology MCP794xx real-time clock/calendar (RTCC) is just 20 pages long. Nevertheless, there's more going on under the hood than the datasheet's brevity might imply. Yes, there's a clock, and, yes, there's a calendar, but the '794 goes beyond simple timekeeping with the addition of key features. Let's take a closer look.

With just eight pins to deal with, '794 design-in is blessedly easy (see Figure 1). Pins X1 and X2 are the connections for the ubiquitous 32.768-kHz "watch crystal" or digital equivalent. VSS and VCC are the main supply, which can be anything between 1.8 and 5.5 V. VBAT is the battery back-up supply, typically a 3-V coin cell. As its name implies, the multi-function pin (MFP) can serve as alarm, clock waveform, or general-purpose output. SCL and SDA are the I²C bus connections to the host. Do note that the maximum speed of the I²C interface depends on the operating voltage—100 kHz or 400 kHz for VCC less than or greater than 2.5 V, respectively.

Those of you who've worked with RTCs lately will recognize that the '794 pinout is kind of a de facto standard that's been adopted by a number of suppliers. It's really the features under the hood that make the '794 unique. Depending on the model (see Figure 2), the '794 extras include 64 bytes of SRAM, 128 bytes of EEPROM, and a factory or field-programmed ID. Other RTCs have some of these

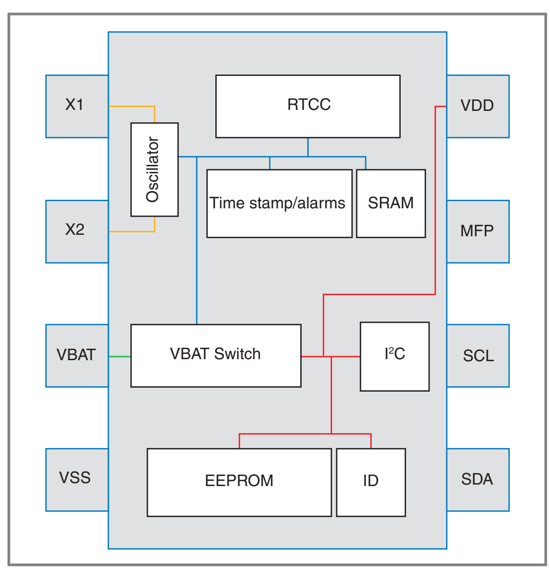


Figure 1—In addition to keeping track of the time and date, the '794 RTCC integrates other handy features including 64 bytes of EEPROM, 64 bytes of RAM, and a factory- and field-programmable 8-byte ID.

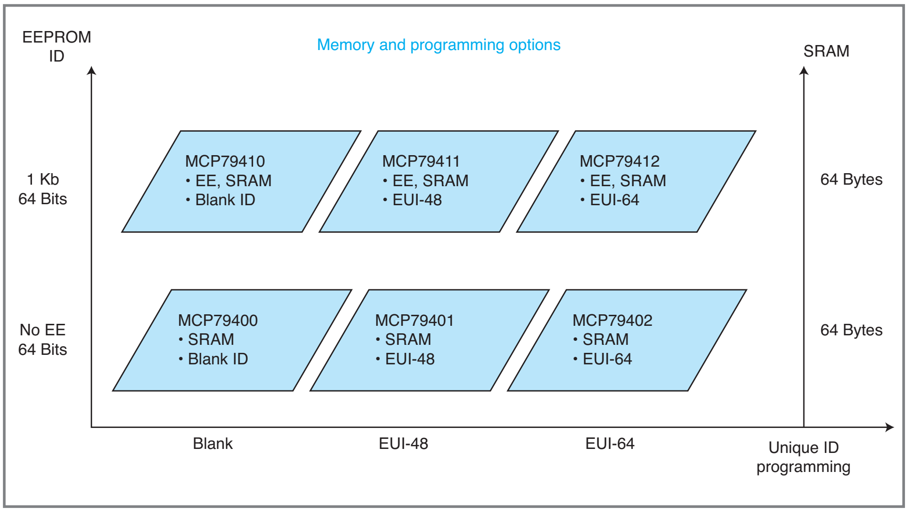


Figure 2—The '794 lineup encompasses variants with and without EEPROM and with or without factory-programmed ID ("Extended Universal Identifier" in IEEE-speak).

features, but none of the others I'm aware of have all three.

There are some other less obvious embellishments that may not seem like a big deal, until you find out you need them. For instance, when the main supply (i.e., VCC) fails, the '794 automatically switches to the battery back-up supply (VBAT). At the same time, it stores the time and date when VCC failed in battery-backed registers. Similarly, when VCC is restored, that time and date is stored in a separate set of registers. It's not hard to imagine applications (e.g., refrigeration, security) that would be well served by knowing when a power failure occurred and how long it lasted.

Calibration is another feature you may not need, but would be sorely missed if you do. All the more so to the degree your application is subject to temperature extremes and requires long term accuracy. The '794 has an 8-bit calibration register you can program to add or subtract up to 127 clock cycles each minute, which works out to a range of ±11 seconds per day (±67 minutes per year).

Besides accuracy, the other reality-check for an RTC is the battery's power. According to the '794 datasheet, the typical power consumption in battery back-up mode is 700 nA. On the supply side, a popular 3-V lithium coin cell, the CR2032, specs capacity as 240 mAh (to 2 V). Plug the numbers into the calculator and you come up with a back-up time of some 342,857 hours, or nearly 40 years! In a real design, there will be losses (battery self discharge) and environmental considerations (higher voltage and higher temperature equal higher current) that need to be factored in, but the point remains that a typical battery will probably last quite a while (e.g., 10 years).

The long-life expectations had me asking about the other shoe that might drop—namely, EEPROM data retention. There's no spec in the datasheet, so I sent off a query to Microchip and was informed they claim the bits will last 200 years! Write cycle endurance is a healthy 1 million cycles, so chances are you won't have to worry about the EEPROM giving out before the battery does.

In an otherwise mains-powered application, the 700 nA (i.e., VBAT) spec is the only one that matters and concern about the active power consumption is surely much ado about nothing. But in a fully battery-powered application, active power consumption could be a consideration. EEPROM writes stand out, with power consumption an order of magnitude higher than other memory accesses (3 mA versus 300 to 400 µA for EEPROM read and SRAM read/write). And the higher current is required for a relatively longer time since it takes 5 ms (typical) to write the EEPROM versus tens to hundreds of microseconds (dictated by the speed of the I²C bus) for other memory accesses. When the chip is powered by VCC, but otherwise idle, power consumption drops to a few microamps.

If indeed EEPROM write power consumption happens to be a concern (energy-harvesting apps come to mind), it's worth noting that according to the '794 datasheet it takes the same time (i.e., 5 ms) to write a single byte as it does an 8-byte page. Furthermore, page writes are more I²C efficient (i.e., less time and power) since one 8-byte transfer requires far fewer clocks than eight separate single-byte transfers.

Better yet, ask yourself if you really need the EEPROM at all. Perhaps you can just get by with the SRAM (i.e., '0x instead of '1x parts) since backing the SRAM up is essentially "free" from a power perspective (i.e., it's included in the 700-nA spec). But a look at the chip pricing reveals the EEPROM costs only a nickel or so (e.g., '79400 is $0.68 at 1,000 units versus $0.74 for the '79401), so feel free to go ahead and splurge.

One other EEPROM write issue to keep in mind is the special "write-busy polling" technique used to

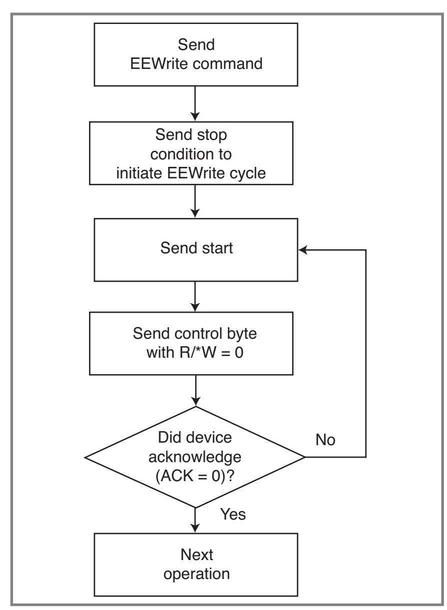


Figure 3—Instead of a status register or pin, the '794 tweaks the I²C protocol to use "ACK polling" to check if an EEPROM write operation is complete.

Address	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE	RESET STATE
00h	ST	10 Seconds			Seconds				Seconds	00 - 59	00h
01h		10 Minutes			Minutes				Minutes	00 - 59	00h
02h		12/$\overline{24}$	10 Hours AM/PM	10 Hours	Hours				Hours	1-12 + AM/PM 00 - 23	00h
03h			OSCON	V_{BAT}	VBATEN	Day			Day	1 - 7	01h
04h			10 Dates		Date				Date	01 - 31	01h
05h			LP	10 Months	Month				Month	01 - 12	01h
06h	10 Years				Year				Year	00 - 99	01h
07h	OUT	SQWE	ALM1	ALM0	EXTOSC	RS2	RS1	RS0	Control Reg.		80h
08h	CALIBRATION								Calibration		00h
09h	UNIQUE UNLOCK ID SEQUENCE								Unlock ID		00h
0Ah		10 Seconds			Seconds				Seconds	00 - 59	00h
0Bh		10 Minutes			Minutes				Minutes	00 - 59	00h
0Ch		12/$\overline{24}$	10 Hour AM/PM	10 Hours	Hours				Hours	1-12 + AM/PM 00 - 23	00h
0Dh	ALM0POL	ALM0C2	ALM0C1	ALM0C0	ALM0IF	Day			Day	1 - 7	01h
0Eh			10 Dates		Date				Date	01 - 31	01h
0Fh				10 Months	Month				Month	01 - 12	01h
10h	Reserved - Do not use								Reserved		01h
11h		10 Seconds			Seconds				Seconds	00 - 59	00h
12h		10 Minutes			Minutes				Minutes	00 - 59	00h
13h		12/$\overline{24}$	10 Hours AM/PM	10 Hours	Hours				Hours	1-12 + AM/PM 00 - 23	00h
14h	ALM1POL	ALM1C2	ALM1C1	ALM1C0	ALM1IF	Day			Day	1 - 7	01h
15h			10 Dates		Date				Date	01 - 31	01h
16h				10 Months	Month				Month	01 - 12	01h
17h	Reserved - Do not use								Reserved		01h
18h		10 Minutes			Minutes						00h
19h		12/$\overline{24}$	10 Hours AM/PM	10 Hours	Hour						00h
1Ah			10 Dates		Day						00h
1Bh	Day			10 Months	Date						00h
1Ch		10 Minutes			Month						00h
1Dh		12/$\overline{24}$	10 Hours AM/PM	10 Hours	Month						00h
1Eh			10 Dates		Date						00h
1Fh	Day			10 Months	Month						00h

Figure 4—The RTCC registers include time, data, and control registers (0x00–0x09), two alarms (0x0A–0x0F, 0x11–0x16), and power-fail and power-restore timestamps (0x18–0x1B, 0x1C–0x1F).

detect write completion (see Figure 3). I'm wondering if this might be a bit "too clever" for typical I²C interfaces or firmware. It may not be a problem, but you'll have to confirm that in your own application.

TIME HAS COME TODAY

Since the '794 is such a simple chip, the easiest way to explain all the features and functions is to go through every single register bit by bit (see Figure 4). The first registers (address 0x00–0x02) are where the action is as they store the current time (i.e., hours, minutes, and seconds). At the top of register 0x00, you'll find the ST control bit, which is super important as it serves to ST(art) and ST(op) the 32.768-kHz oscillator. Needless to say, once you start the clock running you'll want to be very careful about stopping it intentionally or inadvertently since, absent reinitialization, the "real-time" will start getting "unreal."

As an aside, this is an opportune moment to lodge my generic gripe against packing various status, control, and data bits, some read-only, some write-only, some read/write, together.

Address 0x03 combines the day of the week (one to seven) with three important status and control bits. As its name implies, VBAT Enable (VBATEN) determines whether or not the automatic switchover to VBAT occurs when VCC fails (i.e., it falls below 1.5 V). As with the aforementioned oscillator control bit (ST), once you enable battery backup, you'll want to be very careful it stays that way. The next bit, VBAT, is set by hardware when a power failure and VBAT switchover occurs. It needs to be cleared in software, more on that in a moment. Finally, the OSC On (OSCON) bit is set and cleared by hardware to indicate whether the oscillator is running. Along with the ST bit, it gives you a hook for self test and diagnosis. If you turn on the oscillator with ST but it remains off as far as OSCON is concerned, it's likely the crystal is broken. Just don't be fooled by the fact that there's a bit of delay after setting ST before the appearance of the clock is noted in OSCON. Patience, grasshopper.

Next up are the day, month, and year registers (0x04–0x06). Notice there's no provision for keeping

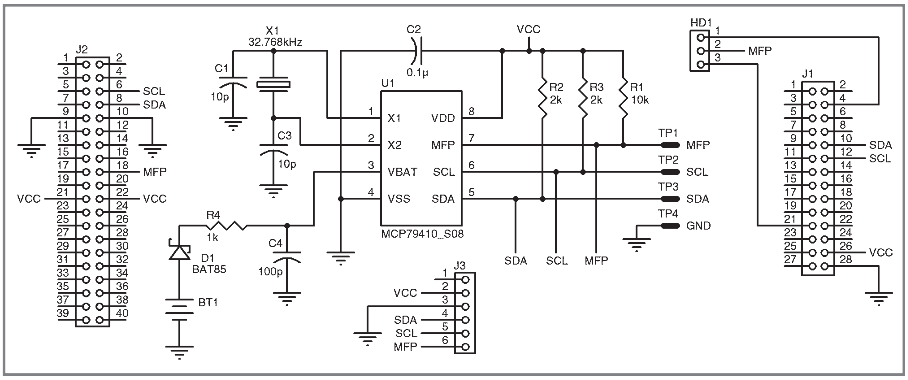


Figure 5—The Microchip RTCC evaluation board provides a recipe for your own design. Just add a crystal and battery and season to taste with a few discretes.

track of the century—so, yes, there will be a "Y2.1K" problem. There's also no automatic adjustment for daylight savings time since that varies around the world (something Apple recently learned the hard way when iPhones used as alarm clocks woke up users in some regions an hour early and others an hour late). However, the chip does keep track of leap years for you with the LP bit, which is set and cleared by hardware.

Register 0x07 contains a variety of control bits, all read/write via software. EXTOSC allows you to use a digital, rather than crystal, 32.768-kHz clock source. OUT sets the level on the MFP pin if you choose to use it as a general purpose output. Alternatively, you can output a clock on MFP by setting the Square Wave Enable (SQWE) bit. The frequency of that clock (e.g., 32.768, 8.192, and 4.096 kHz, once per second) is selected with the RS0-RS2 bits. Note that the general-purpose and square wave output options for MFP are only in effect when the chip is powered by VCC. When running on the battery, MFP reverts to its role as an alarm output, with the ALM0 and ALM1 bits enabling either, neither, or both of the two alarms available.

Register 0x08 contains the calibration factor described earlier—it's a signed 8-bit number that adds (i.e., slows the clock) or subtracts (i.e., speeds the clock) the corresponding number of clock cycles once per minute.

Register 0x09 guards write access to the 8-byte unique chip ID. Writing a 2-byte sequence (0x55, 0xAA) to this register unlocks the unique ID

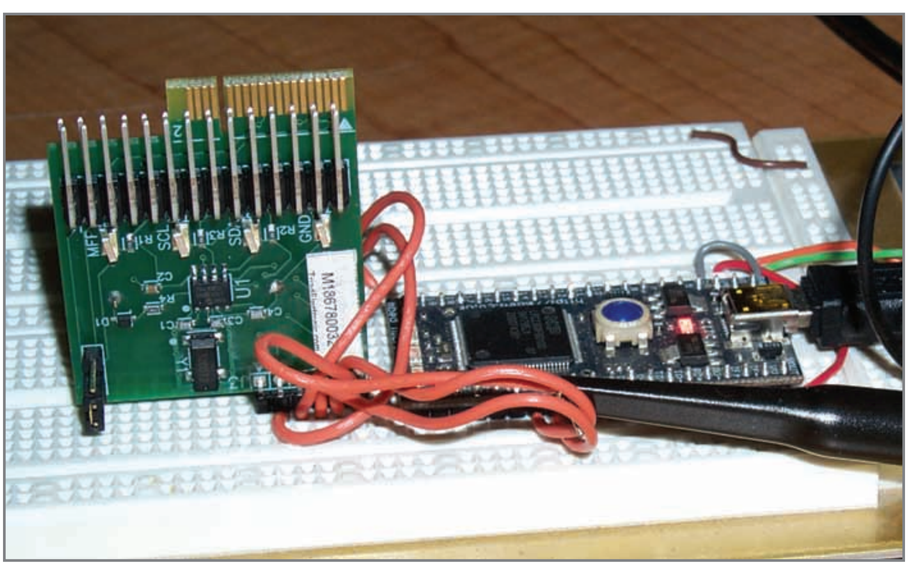


Photo 1—An ARM mbed module and Microchip evaluation board make it easy to check out the '794. A few jumpers, a few lines of code, and a few minutes are all it takes.

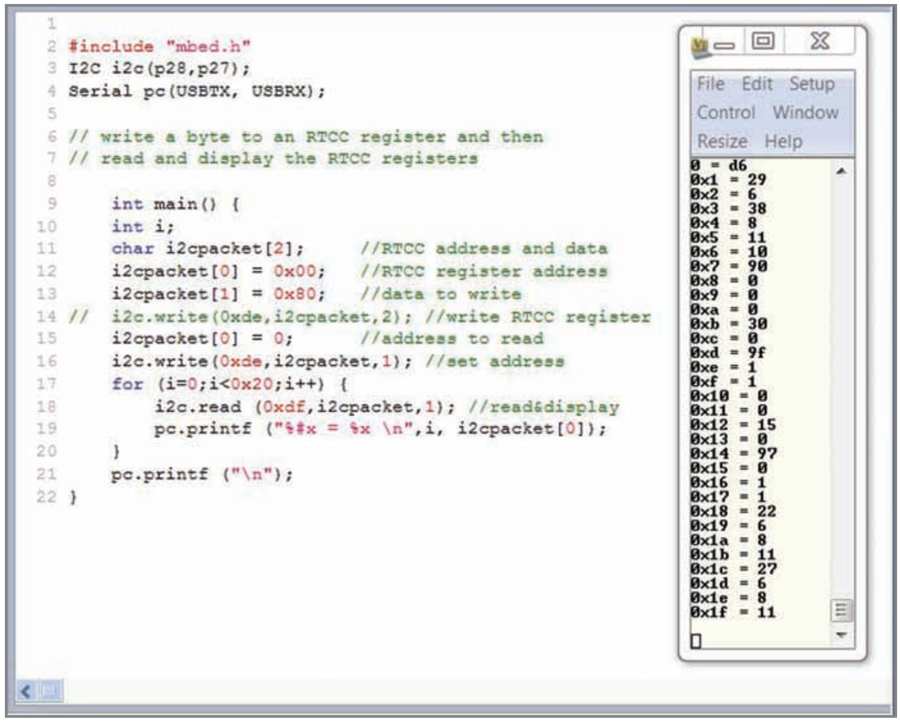


Photo 2—You don't need fancy software just to kick the tires. Decipher the '794 register dump and you'll see it's 6:29:56 AM on November 8, 2010 and there was a 5-minute power failure from 6:22 to 6:27 AM.

for writing using standard EEPROM byte- or page-write commands.

Register blocks 0x0A–0x0F and 0x11–0x16 comprise the date and time settings for the pair of alarms. Notice that there's no setting for years so you can't set an alarm for 12/31/99 to warn of the imminent "Y2.1K" problem. Joking aside, even the most Rip Van Winkle of applications would be wise to wake up at least once a year to celebrate a birthday and make sure everything still works. Each alarm has a polarity bit that determines whether a 1 or 0 is driven on the MFP pin to signify an alarm. The polarity has to be the same for both alarms, so hardware automatically sets ALM2POL to equal ALM1POL. The ALMxIF bits are set by hardware to signal that an alarm has occurred until subsequently cleared by software. Finally, three alarm control bits determine the exact alarm timing. Options include second, minute, hour, or day matching which is an easy way to generate periodic alarms. Or you can specify an explicit date and time for an alarm to occur when all fields (i.e., seconds to months) match.

If you don't need to use one or both alarms, feel free to use the alarm registers as an extra bit of SRAM in a pinch. The fact the values you write to the registers aren't limit checked by the hardware works to your advantage. So, for example, you could set every bit in register 0x0A, even though the result (0x7F) is not a valid value for seconds (i.e., not between 0 and 59 using BCD coding).

The final two blocks of registers are for the power-fail (0x18–0x1B) and power-restore (0x1C–0x1F) "timestamp," which tracks the occurrence of power (VCC) outages. After a power fail and restoration, the timestamps remain latched until you rearm by clearing the VBAT bit (which was set by the power failure) in register 0x03. So, if the timestamps matter, the first thing your application should do after powering up is deal with them (e.g., log the event in the SRAM) and then rearm. Notice that timestamp resolution is minute-by-minute since there are no seconds fields.

TIME CARD

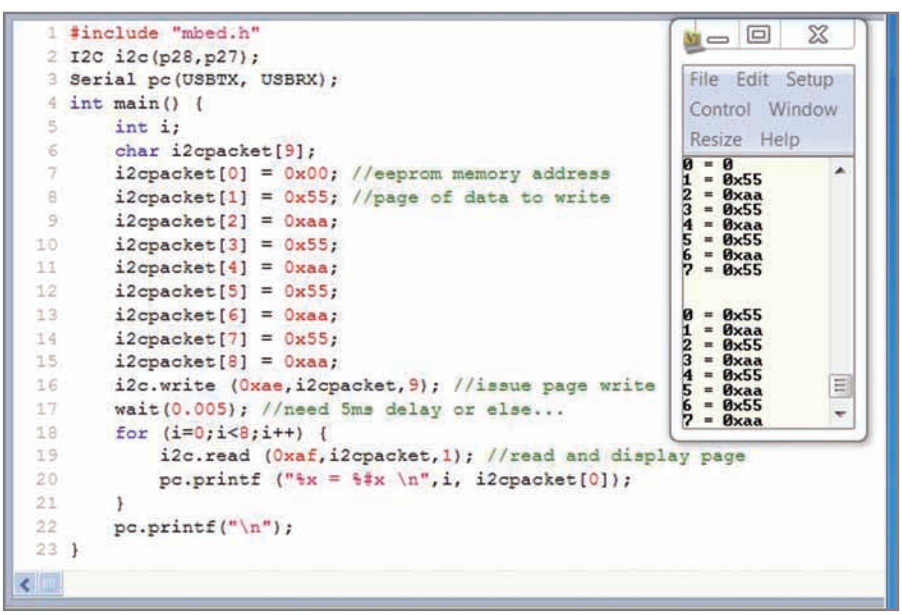
Microchip offers an RTCC evaluation board that makes it easy (and inexpensive at just $45) to see if their chip can pass the test of time. Along one edge the board has connectors that are compatible with their MCU evaluation gear. For those who prefer to hook the RTC to something else, there's a prototyping-friendly six-pin header on the other edge. As you can see in Figure 5, the board provides access to the key signals and includes the crystal and battery (there's a coin-cell holder on the back of the PCB). As well, the schematic shows the cast of supporting characters: capacitors for the crystal and power-supply bypass, a current limit resistor and blocking diode for the battery, and pull-up resistors for the open-collector I²C lines (i.e., SDA and SCL).

I happened to have one of the ARM mbed gadgets sitting on my desk and figured I'd use it to give the RTCC a test drive (see Photo 1). The mbed C library includes routines for I²C bus access so no bit-banging required. It was a simple matter of plugging in some jumpers and hacking a few lines of code to get the RTCC up and running. (For more information, refer to my article, "Easy (E)mbed: An Alternative Approach to Embedded Programming," *Circuit Cellar* 227, 2009.)

The quick and easy mbed development environment encourages a casual programming style that's great for experimenting. In the old days, I might have taken more care to come up with an elaborate and full-featured test suite. But it turns out I needed nothing more than some simple code snippets to read and write the RTCC registers and memory to get answers to my questions. It's easy to just punch the key parameters (e.g., register address and write data) directly into the code and then "blow and go."

The first program was simple. I just wrote an RTCC register and then read and displayed the contents of all of them. I edited and ran the program a few times to initialize the date and time fields. In the real world, you'd take advantage of the ability to load the entire block of registers with a single multi-byte transfer, but I wanted to be able to jump around. Finally, I wrote register 0x00 with 0x80 to set the ST bit and start the oscillator. Commenting out the register write statement and re-running the program, I was happy to see the clock ticking away.

Hooking up a scope, I played around wiggling the MFP pin using the control bits in register 0x07 (e.g., OUT, SQWE, etc.). Next, after installing a battery in the coin-cell



```
#include "mbed.h"
I2C i2c(p28,p27);
Serial pc(USBTX, USBRX);
int main() {
    int i;
    char i2cpacket[9];
    i2cpacket[0] = 0x00; //eeprom memory address
    i2cpacket[1] = 0x55; //page of data to write
    i2cpacket[2] = 0xaa;
    i2cpacket[3] = 0x55;
    i2cpacket[4] = 0xaa;
    i2cpacket[5] = 0x55;
    i2cpacket[6] = 0xaa;
    i2cpacket[7] = 0x55;
    i2cpacket[8] = 0xaa;
    i2c.write (0xae,i2cpacket,9); //issue page write
    wait(0.005); //need 5ms delay or else...
    for (i=0;i<8;i++) {
        i2c.read (0xaf,i2cpacket,1); //read and display page
        pc.printf ("%x = %#x \n",i, i2cpacket[0]);
    }
    pc.printf("\n");
}
```


Photo 3—Whether using the Microchip I2C polling scheme or a software delay, make sure you accommodate the EEPROM write cycle timing. Without the 5-ms wait statement after the EEPROM page write, a subsequent EEPROM read goes sour. Put in the wait statement and all goes well.

holder, I set the VBATEN bit (register 0x03) to enable automatic battery backup. Now, cycling the RTCC power (i.e., yanking the VCC jumper) I could see that time was maintained across power outages. I was also able to see the power-fail and power-restore timestamp feature in action (see Photo 2), remembering to clear the VBAT bit (to rearm the timestamp) after each power cycle.

Testing the SRAM is easy enough. Just change the I²C addresses to point to the SRAM and the program writes a byte and then reads it back for display so you can make sure it got safely stashed away. To confirm battery backup works, pull the VCC jumper, and while you're waiting for the electrons to dissipate, take a moment to comment out the I²C (i.e., SRAM) write statement in the program and reload the mbed. Now plug in the VCC jumper and run the program again to display the SRAM data and make sure it survived the power cycle.

It was all pretty simple and there weren't any real surprises. The only real question I had left was about the special "write busy" I²C polling scheme described earlier. I'd feared the RTCC or mbed might get confused or hung up, but fortunately, their I²C routines played well together. Indeed, accessing the EEPROM was pretty much as easy as SRAM since you don't even have to worry whether a location is erased (i.e., 0xFF) or not so just go ahead and blast away.

Now I was wondering if the "write busy" polling might actually be working as intended out of the box (i.e., Is the mbed I²C write routine actually waiting around for the '794 to signal EEPROM write completion?) I was pretty sure that would not be the case, but there have been plenty of times I've been surprised by stuff working when I had my doubts.

Anyway, no harm in trying, so I hacked the code a bit to exercise the byte and page-write functions. My intuition was confirmed when I tried to write one EEPROM location and then another immediately with no intervening delay to discover bits were lost (see Photo 3).

Let's see. Should I hook up a logic analyzer to the I²C lines, dig under the hood of the mbed I²C routines, and write/test/debug bit-banging code to manhandle the RTCC "write busy" protocol? Or should I just add a 5-ms delay in software after each EEPROM write command? Next question.

The software delay worked fine. However, this exercise does serve as a reminder that, unless you're in a real hurry, it's always a good idea to verify write success in software (i.e., read back and compare) just in case.

CLOCK JOCK

Given that many modern MCUs include the functionality, I was a bit surprised there's still seemingly a decent market for stand-alone real-time clock chips. It must be the case though, since RTC suppliers include the very same outfits (e.g., Microchip Technology, STMicroelectronics, and NXP) making the advanced MCUs.

Beyond the headline clock functionality, at well under a buck ($0.68 to $0.81 in 1,000-piece quantities depending on options), the extras (e.g., EEPROM, SRAM, unique ID) can justify the cost. And there are some other subtle, but nevertheless real, potential cost savings. For example, having Microchip personalize each chip with a unique ID simplifies your factory MCU flash programming flow. Or perhaps you're just making a few hundred gadgets that need a globally unique ID (e.g., MAC address). The versions of the '794 with a preprogrammed ID (i.e., 'x1 and 'x2) are only a nickel more than the blank ID parts, so in low volume, it's cheaper to buy some chips than pay IEEE a license fee.

But wait, what about the "Y2.1K" problem? Well, let's just worry about it then.

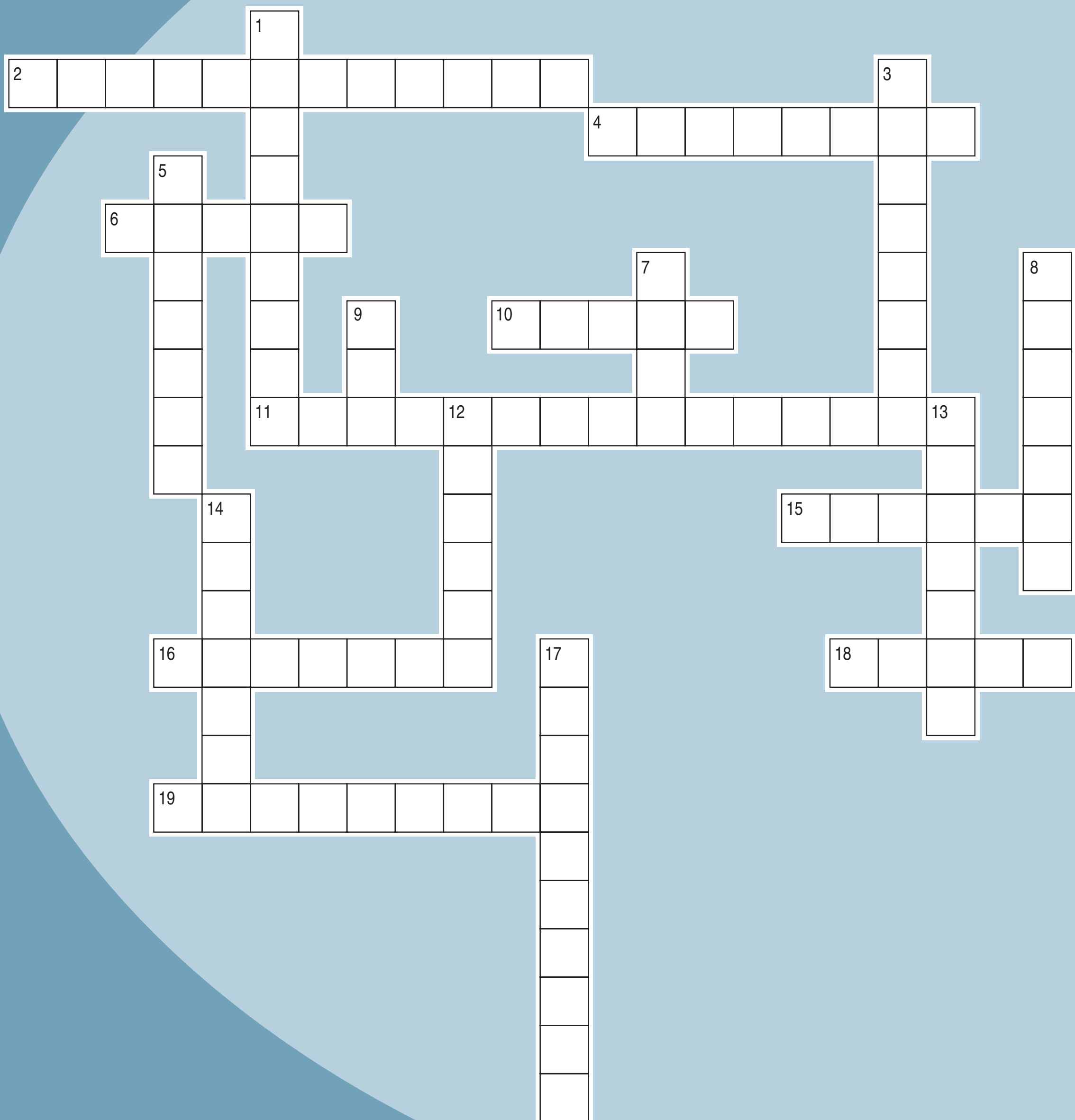
Tom Cantrell has been working on chip, board, and systems design and marketing for several years. You may reach him by e-mail at tom.cantrell@circuitcellar.com or directly at microfuture@att.net

SOURCES

mbed Microcontroller
ARM | www.mbed.org

MCP7941*x* RTCC
Microchip Technology | www.microchip.com

CROSSWORD



Down

1. Tcl, Lisp, Fortran
3. Minimum power delivered
5. CR2032 is a standard
7. Empty set size
8. Schmitt
9. 0.001″
12. Element in a “ground island”
13. 1,024 Petabytes
14. Speaker for high-frequency sounds
17. X

Across

2. 1,000 Watt-hours [two words]
4. Where electricity enters/exits a device
6. Balanced/unbalanced
10. OOK is on/off what?
11. SiO [two words]
15. 0 to 9, Counter
16. Measurement, Calibration, used with main scale
18. Quick fix
19. Rust [two words]

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

IDEA BOX

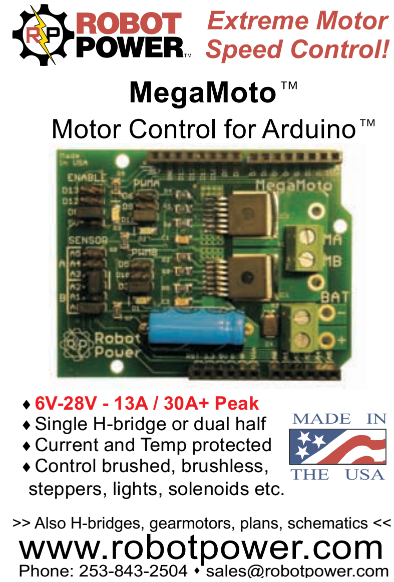
THE DIRECTORY OF PRODUCTS AND SERVICES

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

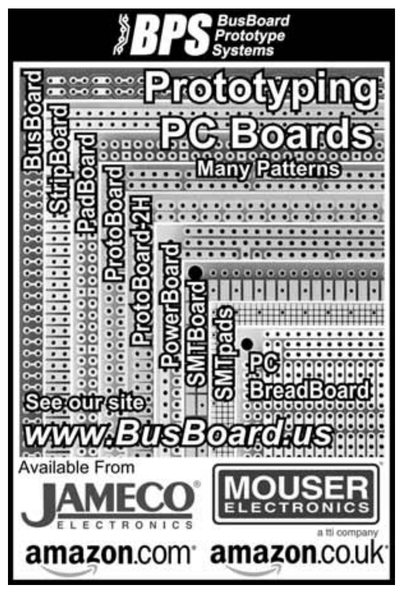
The Vendor Directory at www.circuitcellar.com/vendor/
is your guide to a variety of engineering products and services.


MaxSonar
Perfect for OEMs & Engineers
XL-MaxSonar-EZ
• High acoustic power
• Real-time calibration
• Tiny size
• $39.95USD qty. 1
MaxSonar-WR (IP67)
• Industrial packaging
• Weather resistant
• Standard ¾" fitting
• $99.95USD qty. 1
LV-MaxSonar-EZ
• Power-up calibration
• Smallest MaxSonar
• Low power, 2.5V-5.5V
• $29.95USD qty. 1
MaxSonar-WRC (IP67)
• Compact packaging
• Weather resistant
• Quality narrow beam
• $99.95USD qty. 1
www.maxbotix.com


Tag-Connect
ICD 3
Got UGH?
Eliminate Ugly Great Headers!!
JTAG Cable Plugs Directly To PCB!
Save Space & Cost on Every Board!!
www.PlugOfNails.com
Tag-Connector footprints as small as 0.02 sq. inch (0.13 sq cm)

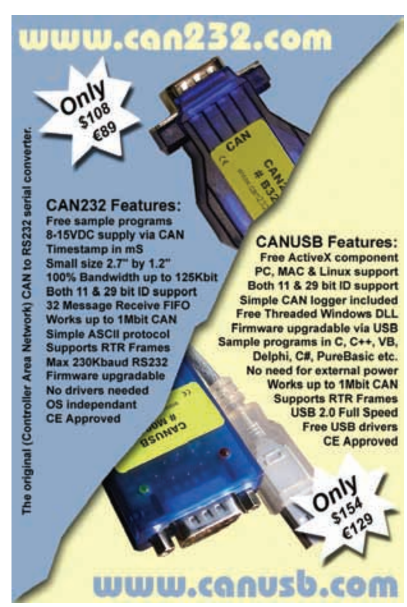

ROBOT POWER
Extreme Motor Speed Control!
MegaMoto™
Motor Control for Arduino™
♦ 6V-28V - 13A / 30A+ Peak
♦ Single H-bridge or dual half
♦ Current and Temp protected
♦ Control brushed, brushless, steppers, lights, solenoids etc.
MADE IN THE USA
>> Also H-bridges, gearmotors, plans, schematics <<
www.robotpower.com
Phone: 253-843-2504 • sales@robotpower.com


Exactly why is USB hard?
Part# QUSB2 - $149 qty 1
* 25MB/sec data bus, bit-bang, RS-232, I2C Master, SPI Master, FPGA port
* Software library and example code
* New, high-performance kernel drivers for Windows, Linux and MacOS
QuickUSB®
www.quickusb.com


BPS BusBoard Prototype Systems
Prototyping PC Boards
Many Patterns
BusBoard
StripBoard
PadBoard
ProtoBoard
ProtoBoard-2H
PowerBoard
SMTBoard
SMTpads
PC BreadBoard
See our site
www.BusBoard.us
Available From
JAMECO ELECTRONICS
MOUSER ELECTRONICS
a tti company
amazon.com amazon.co.uk


I²C™/SMBus
• Bus Monitors
• Protocol Analyzers
• Host Adapters
• Multiplexers
• Battery Applications
• Software Tools
MCC
Micro Computer Control
I²C is a trademark of Philips Corporation
www.mcc-us.com


RC™
Industrial Analog Signal Acquisition and Control
Starting at $99 OEM board
• 5.0 x 3.7", 5-24V DC power
• C/C++ programmable
• 16 ±10V analog inputs, 8 ±5V analog outputs
• 15+ TTL, 7 Solenoid drivers, Temperature Sensor
• CompactFlash, Ethernet, USB
• LCD, RS-232/ RS485/wireless
• 10+ years life time, Power-on-run within 1 second
100+ Low Cost Controllers with ADC, DAC, UARTs, 300 I/Os, solenoid, relays, CompactFlash, LCD, Ethernet, USB, motion control. Custom board design. Save time and money.
TERN INC.
1950 5th Street, Davis, CA 95616 USA
Tel: 530-758-0180 • Fax: 530-758-0181
www.tern.com • sales@tern.com


www.can232.com
Only $108 €89
The original (Controller Area Network) CAN to RS232 serial converter.
CAN232 Features:
Free sample programs
8-15VDC supply via CAN
Timestamp in mS
Small size 2.7" by 1.2"
100% Bandwidth up to 125Kbit
Both 11 & 29 bit ID support
32 Message Receive FIFO
Works up to 1Mbit CAN
Simple ASCII protocol
Supports RTR Frames
Max 230Kbaud RS232
Firmware upgradable
No drivers needed
OS independant
CE Approved
CANUSB Features:
Free ActiveX component
PC, MAC & Linux support
Both 11 & 29 bit ID support
Simple CAN logger included
Free Threaded Windows DLL
Firmware upgradable via USB
Sample programs in C, C++, VB, Delphi, C#, PureBasic etc.
No need for external power
Works up to 1Mbit CAN
Supports RTR Frames
USB 2.0 Full Speed
Free USB drivers
CE Approved
Only $154 €129
www.canusb.com

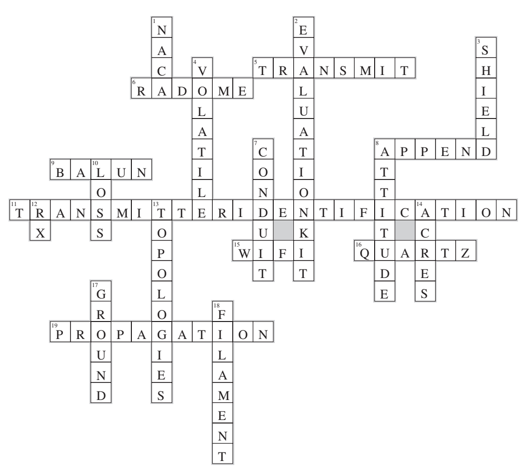
CROSSWORD ANSWERS from Issue 247

Down

1. **NACA**—Before NASA
2. **EVALUATIONKIT**—EVKIT [two words]
3. **SHIELD**—A cover that protects users from high voltages
4. **VOLATILE**—Memory requiring power
7. **CONDUIT**—Piping for wires
8. **ATTITUDE**—Orientation with respect to motion
10. **LOSS**—Drop in dB
12. **RX**—Receive
13. **TOPOLOGIES**—Bus, Star, Ring, P2P
14. **ACRES**—1 square mile = 640 ____
17. **GROUND**—Earth wire
18. **FILAMENT**—Tungsten wire through which electricity passes

Across

5. **TRANSMIT**—Tx
6. **RADOME**—Radar dome; can have shapes such as planar or spherical
8. **APPEND**—Add to end
9. **BALUN**—What you get when you combine balance and unbalance
11. **TRANSMITTERIDENTIFICATION**—TX ID [two words]
15. **WIFI**—IEEE 802.11
16. **QUARTZ**—Timing XTAL
19. **PROPAGATION**—Wave transmission



INDEX OF ADVERTISERS

PREVIEW of April Issue 249

Theme: Embedded Programming

Microprogramming a CPU (Part 1): The Architecture, Programmer Model, and SMP

Reprogrammable UAV Autopilot System (Part 1): System Hardware and Software

DSP-Based Color Organ: Use the Convolution Technique to Create High-Performance Filters

Testing Platform: Construct a Custom Scope and Analyzer

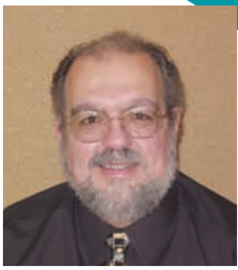
THE CONSUMMATE ENGINEER **Reliable Programming:** Work Toward Fault-Free Software

THE DARKER SIDE **Noise Figures 101**

ABOVE THE GROUND PLANE **Thermal Performance**

FROM THE BENCH **Boot Up:** Bootloading Essentials

PRIORITY INTERRUPT



by Steve Ciarcia, Founder and Editorial Director

Forks in the Road

Not that it is relevant for any other reason than to let you in on how the conversation started, but the primary local pastime down here at the cottage is "happy hour." There are a bunch of home town bars and eateries along the beach where all the "regulars" meet and greet each other as well as mingle with the "touristas." The conversation this particular afternoon was more serious than usual.

Sitting next to me was a 50-ish gentleman who, while quite personable, seemed troubled. Invariably, the conversation ended up discussing the economy and he confided (as much as you can at a bar anyway) that he had been laid off from his engineering job downstate and was searching for prospects elsewhere. I hesitated to tell him that upstate here at the cottage, most people still think "engineer" has something to do with trains, but then again, he wasn't in the bar looking for a job. I wanted him to say that it was indeed an economic issue and the layoff was temporary.

Unfortunately, that was not the case. Apparently the changing needs of the company required a new skill set, and he was no longer part of the solution. The processors and development packages he felt comfortable using were becoming obsolete, as well as the design techniques he employed. He was an engineer of a different age who had become too complacent. He wasn't keeping up and he was replaced by a younger engineer with an up-to-date skill set.

I might be the last person you want to listen to for career advice, especially when all of my advice is usually about starting your own business so you lessen the occasions and consequences of bureaucratic nonsense controlling your destiny. But, bitter and warped opinion aside, I do understand that most engineers have to deal with the establishment, and surviving it takes both wit and wisdom. Here's my take on it.

Engineering jobs in the private sector do not offer career-long job security and you can't expect to just drift into retirement collecting a salary. Unless your engineering performance contributes value to the business, you will be replaced. Be it design or management, the secret to longevity is about understanding what skill set is required at the different stages in your career.

Virtually all of us start as design engineers, and many choose to stay there for their whole careers. I applaud those with the stamina to do it forever because maintaining an up-to-date skill set as a designer is not inconsequential. First, use the Internet because it is the greatest knowledge repository there is. Download and try new software and tools. Keep familiar with the latest processors and development techniques. Second, you need to network, network, network. Engage other engineers and share ideas. Go to trade shows, conferences, or take a class and learn from others. Third, don't get trapped by narrow job descriptions. Seek other assignments and out-of-the box responsibilities, especially ones with high visibility and exposure. Finally, publish, publish, publish. Take the time to write an article about something you know. Believe me, knowing that thousands of smart people may read something you write keeps you on your toes.

The other big career fork for engineers is whether to move into project management. Of course, unless you do all the above to maintain your design skills, you better be good at it, because you aren't going back (not easily anyway). The secret to success here is your engineering training along with expertly applied BS (seriously).Corporate project teams consist of a bunch of people with a variety of talents. You'll have to deal with bean counters, lawyers, manufacturing guys, and design engineers. You can learn what they know, but they can't easily learn what you know. You will have the ability to speak to the engineers as peers and gain their respect. With a little knowledge and a bit of information you'll earn the confidence of the entire team. It's a lot easier for you to learn the basics of finance, contracts, and manufacturing techniques than it is for these people to become EEs. Your management role benefits from the comprehension and ingenuity you have being an engineer but leaves the nitty gritty of implementing it to the team.

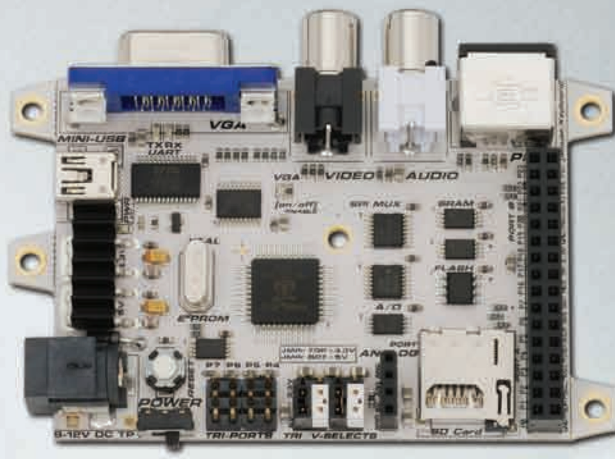
I guess if there is a lesson here it is don't become complacent. Don't wait until a technical deficit appears before expanding your skill set (albeit learning a new compiler or becoming a junior bean counter for the next finance meeting). The demands of being an electrical engineer are evolutionary. Your skill set needs to follow that evolution or you could be sitting at the bar telling a stranger that you are looking for a job, too.

steve.ciarcia@circuitcellar.com

Steve

The C3 is a true "Credit-Card–sized Computer" that augments the multicore Propeller chip with several onboard peripherals to facilitate large scale and complex applications. Ready to support a wide array of peripherals, this system is ideal for industrial control, monitoring systems, high-level languages, robotics, gaming, and more.

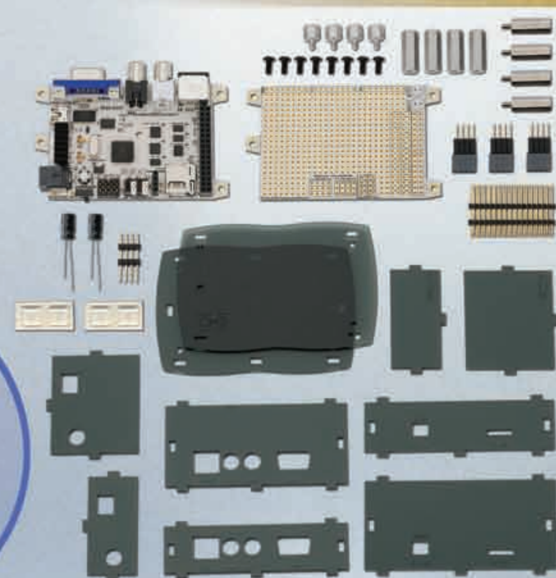
The C3 was designed to be compatible with other Propeller-based development boards, so porting applications is a snap. Additionally, the C3 uses a novel SPI bus design that allows all the onboard peripherals to be accessed with a single SPI bus (3 signals) along with chip-select logic (2 signals) that supports up to 8 devices in total. Thus, the IO impact of all the onboard peripherals is minimal.



C3 PROP

Features include:

- Multicore Propeller P8X32A-Q44 with 5 MHz socketed crystal to upgrade for over-clocking experiments
- 64 KB Propeller boot EEPROM holds boot image plus 32 KB extra for data
- 64K of external SPI SRAM
- Built in 2-Channel 12-bit A/D converter
- 1 MB SPI FLASH memory
- MicroSD card w/SPI interface



The C3 board is available singly (#32209; $79.99) or as part of a "Full Kit" which includes a C3 board, prototyping board, and a laser-cut acrylic enclosure (#32205; $129.99). *Assembly required.*

Order the Propeller C3 (#32209; $79.99) at www.parallax.com or call our Sales Department toll-free at 888-512-1024 (Mon-Fri, 7a.m. - 5p.m., PT).

Friendly microcontrollers, legendary resources.™

PARALLAX INC

www.parallax.com

"ParallaxInc" on Twitter, Facebook, and YouTube