Build A Simple Automatic GARAGE DOOR CLOSER

- The Radio Whisperer
  Become one with WSPR
- Design Cycle
  MPLAB X
- Electronics Q&A
  Flickering Lamp Circuits
  Water Softeners
  USB Tester
The easiest finished product you will ever create

Start with the PK70 EX

Design your own, or use a NetBurner Personality Blade

Add your label to get a finished product

Personality Blades

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

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

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

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

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

Quad UART Blade: RS-232 Version
Network enable up to 4 RS-232 serial devices

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

Quad UART Blade: RS-232/422/485 Version
(P/N NBPKBU-MM5CR) ..................................... $245 ea.
Network enable up to 4 RS-232/422/485 serial devices

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

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

EasyPIC v7

Best selling PIC development board in the world enters its 7th generation of development. It is state of the art in design, functionality and quality. With 4 connectors for each port EasyPIC v7 has amazing connectivity. Ports are logically grouped with their corresponding LEDs and Buttons. Powerful on-board mikroProg In-Circuit Debugger and programmer supports over 250, both 3.3V and 5V devices. Three types of displays, Serial EEPROM, two temperature sensors, Piezo Buzzer, USB connector, RS-232 and FTDI. Oscilloscope GND pins, as well as mikroBus support make this board an irreplaceable PIC development station.

GET IT NOW
www.mikroe.com
BitScope

Digital + Analog

100 MHz Digital Oscilloscope
- Dual Channel Digital Storage Oscilloscope with up to 12 bit analog sample resolution and high speed real-time waveform display.

40 MSPS x 8 Channel Logic Analyzer
- Captures eight logic/timing signals together with sophisticated cross-triggers for precise multi-channel mixed signal measurements.

Serial Logic and Protocol Analyzer
- Capture and analyze SPI, CAN, I2C, UART & logic timing concurrently with analog. Solve complex system control problems with ease.

Real-Time Spectrum Analyzer
- Display analog waveforms and their spectra simultaneously in real-time. Baseband or RF signals with variable bandwidth control.

Waveform and Logic Generators
- Generate an arbitrary waveform and capture analog & digital signals concurrently or create programmable logic and/or protocol patterns.

Multi-Channel Chart Recorder
- Record to disk anything BitScope can capture. Allows off-line replay and waveform analysis. Export captured waveforms and logic signals.

Everything in one tiny package!

BitScope Pocket Analyzer is the world's first Mixed Signal Scope to include a powerful Logic Protocol Analyzer, Waveform & Pattern Generator, Spectrum Analyzer and Chart Recorder in one ultra-portable USB powered package.

Pocket Analyzer is fast, displaying up to 100 frames per second with up to 12kS per frame. Alternatively stream data direct to disk for replay and analysis.

Compatible with major operating systems including Windows, Linux & Mac OS X. Pocket Analyzer is your ideal test and measurement companion.

bitscope.com/nv
Finding the right parts for your robot can be difficult, but you also don’t want to spend all your time reinventing the wheel (or motor controller). That’s where we come in: Pololu has the unique products - from actuators to wireless modules - that can help you take your robot from idea to reality.

Find these products and more at [www.pololu.com](http://www.pololu.com)
Projects & Features

28 Build a Stereo Gain Controller
Tired of being blasted off your seat by excessive audio volume when you switch television channels or when the commercials come on? To mitigate such unnerving changes in program audio, a popular automatic volume controller from the vacuum tube era was updated to today’s technology. This circuit is also useful in home recording.
■ By Dev Gualtieri

34 Build the Garage Door Closer
Leave your garage door open? No problem! This handy device will close it automatically for you. (Plus, you don’t even have to remember to replace batteries in this unit.)
■ By Brian Beard

38 Build a USB Volume and Mute Controller for Your PC
Now you can do things like listen to music on your laptop with the lid closed because volume control is now accessible externally.
■ By Richard Dzioba

44 The Radio Whisperer
Learn about the methodology behind WSPR (weak signal propagation reporter) software.
■ By George Steber

50 Build a Variable Timer

Columns

10 TechKnowledgey 2012
Events, Advances, and News
Find out how to upgrade your iPad 2 with a keyboard case, take 4-D photos, and if Bing is losing big (it is), along with other pretty interesting stuff.

14 Near Space
Approaching the Final Frontier
Global Positioning System Simulator, Ver 2.0.

22 Q & A
Reader Questions Answered Here
Questions on flickering lamp circuits, water softeners, and USB testers are just some of the things talked about this time.

52 Smiley’s Workshop
Programming • Hardware • Projects
Breadboard LCD Navigator Part 1: the LCD.

58 The Spin Zone
Adventures in Propeller Programming
Prop Talk.

68 The Design Cycle
Advanced Techniques for Design Engineers
MPLAB’s X Factor.

Departments

08 DEVELOPING
09 PERSPECTIVES
26 NEW PRODUCTS
51 SHOWCASE
64 NV WEBSTORE
67 ELECTRO-NET
76 CLASSIFIEDS
78 TECH FORUM
81 AD INDEX

Nuts & Volts (ISSN 1528-9885/CDN Pub Agree #40702530) is published monthly for $26.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL MAILING OFFICES. POSTMASTER: Send address changes to Nuts & Volts, P.O. Box 15277, North Hollywood, CA 91615 or Station A, P.O. Box 54, Windsor ON N9A 6J5, subscriptions@nutsvolts.com.
The **DL-020 Sequential Logic Trainer** introduces concepts of sequential logic design, which is the final basic elements to understanding microprocessor and microcontroller logic.

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GSPA SOP-8B</strong></td>
<td>8-Pin SOP to Through - Hole Prototyping Adapter</td>
</tr>
<tr>
<td><strong>GSPA TSOP-48S</strong></td>
<td>48-Pin TSOP to Through - Hole Prototyping Adapter</td>
</tr>
<tr>
<td><strong>GSPA PLCC-44</strong></td>
<td>44-Pin PLCC to Through - Hole Prototyping Adapter</td>
</tr>
<tr>
<td><strong>GSPA-K1</strong></td>
<td>5-pin DIN connector housed on a PC board</td>
</tr>
</tbody>
</table>

**Offer Selection of Over 18 DIY Educational Kits**

**Courseware**
Our courseware offers many options to aid in electronics training and lab instruction.

**Surface Mount Device Soldering & Re-Work Tool Kit**
The set includes 11 tools commonly used for working with surface mount components

**Basic Fixed Test Leads**
**model# CT3737**
Provides users with the basic connection leads to begin using one’s digital multimeter (DMM)

**RDB-10 Resistance Decade Box**
A compact, convenient tool for aiding in engineering design and testing as well as calibration of test equipment.

**PB-503 Analog & Digital Design Workstation**
A robust electronics trainer suitable for all levels of electronics instruction and design.

**PRO-S Lab**
Breadboard with External Power & Jumper Wires

January 2012
How Old is Old Enough?

One of the most often asked questions in my inbox is ‘how old is old enough to begin working with electronics?’ The answer, of course, is that it depends on the individual, what you mean by ‘working with,’ and the amount of supervision provided by an experienced or at least alert adult.

I began disassembling radios and electric toys as soon as I could handle a screwdriver. When I was 11, I built a ham radio transmitter with parts recycled from a discarded TV — working alone in my father’s workshop.

Today, allowing an unsupervised 11 year old to disassemble a potentially lethal high voltage circuit probably constitutes child abuse in some states. Fortunately, the CRT has been supplanted by the lower voltage LCD display and modern electronics, in general, operate at relatively harmless voltages.

If you’re looking to introduce electronics to an eight or nine year old, I’m a big fan of the Snap Circuits series which is sort of like LEGOs with embedded components. The entry-level model is around $30 from Amazon. With color-coded LEGO-like components, you and your youngster can build over a hundred different circuits — from a water alarm to a timer.

The problem with the series is that the user manual is written for someone who can read at a high-school level. So, the kit is great for pre-teens, as long as someone who can read the manual and explain the circuits supervises them.

Although the Snap Circuit and similar products can be used by teens, the solderless electronic breadboard approach is more fitting for capable hands. It takes a good deal of dexterity to plug in ICs without bending the leads. A breadboard and battery or simple power supply and a handful of components are not only inexpensive, but relatively limitless in terms of circuit possibilities.

However, unlike the Snap Circuit, it’s easy to cross connections and burn out components — that is, unless there’s someone watching over the builder’s shoulders. There’s also something to be said for the experience of burning up a few inexpensive components — the lesson can carry over to more expensive circuits later.

If you’re a regular reader of this column, you know that I’m a big fan of teardowns, especially when they’re supervised. Nothing can beat a knowledgeable person tearing down a smoke alarm, compact fluorescent bulb, or other device with a step-by-step description of components and circuit theory. One of my early mentors used to walk me through a teardown and then hand me the tools and let me reassemble everything. I’d often end up with a few extra screws here and there, but the lessons stuck.

The real challenge today is making the introduction to electronics memorable and exciting. After all, electronics as a hobby has evolved considerably since the days when computers were fabricated with discrete components. For example, if your project culminates in a blinking LED, it’s probably not going to compete well with your smart phone or tablet.

I’ve had good success using robotics as a vehicle for circuit design and testing. It’s no Angry Birds app, but a walking robot — regardless of how simple — is bound to capture the interest of youngsters of any age.
BEGINNER'S GUIDE COMPLETE COMBO!

Combo Price $229.95

For complete details, visit our webstore @ www.nutsvolts.com.

SABERTooth 2x60 Dual 60A Continuous Brushed Motor Controller

Specifications:
- 60A Continuous, 120A peak current
- 6V - 33.6V input voltage range (8s lipo max)
- Analog, RC (PWM), and Serial interface
- Overcurrent, overvoltage, undervoltage, and overheat protection built in
- Fan cooling keeps the controller cool
- Heavy duty terminals take up to 2 gauge wire
- 1 year warranty on every Sabertooth
- DEScribe Software Suite Compatible

ONLY $189.99

-EVT accruable-

- Dimensions: 53.5mm x 53.5mm x 60mm
- Weight: 160g
- Output: 12V
- Input: 6V - 33.6V

BaneBots
BANEBOOTS.COM
970-861-8660

PRINTED IN THE USA ON SFI & FSC STOCK.
TRACTOR BEAMS BECOMING A REALITY?

No, we're not talking about the headlights on a John Deere. As anyone who has ever watched Star Trek knows, any decent starship will be fitted with a device — known as a tractor beam — that produces a gravity-like attraction field capable of dragging in pretty much any object, regardless of its mass and inertia. Although the concept appears seriously impractical for any type of large objects, it has been known for some time that certain types of lasers can move small, solid items. For example, last year, scientists at the Australian National University moved some tiny particles 1.5 m through the air using a Laguerre-Gaussian beam. Now, NASA's Office of the Chief Technologist is getting into the act and has awarded $100,000 to a group of laser experts at the Goddard Space Flight Center (www.nasa.gov/centers/goddard/) to study three different approaches for creating the laser equivalent of a DustBuster®.

The first technique involves the "optical tweezers" concept in which two counterpropagating light beams generate a ring-like geometry that traps particles in the dark core of the overlapping beams. Varying one of the beams' intensities heats the air around the trapped particle and thereby moves it along the center of the ring. The drawback here is that the system requires an atmosphere, which isn't exactly omnipresent in space.

The second employs "optical solenoid beams" which were introduced in a 2010 issue of Optics Express and described as "diffractionless solutions of the Helmholtz equation whose diffraction-limited in-plane intensity peak spirals around the optical axis, and whose wavefronts carry an independent helical pitch." Here, the particles are drawn back along the entire light beam, and no atmosphere is required.

The remaining technique has never been demonstrated, but it involves the use of a Bessel beam in which rings of light surround a central point. In theory, the beam could induce electric and magnetic fields in an object's path and use the spray of forward-scattered light to pull it backward, against the beam itself. Whichever method — if any — proves practical, it is expected to be useful not only for extraterrestrial particulate matter but also for single molecules, viruses, RNA, and even functioning cells on Earth.

COMPUTERS AND NETWORKING

NEW ULTRABOOK UNDER $1,000

The Ultrabook category of computing has been defined by Intel as basically a notebook computer that fills the gap between tablets and lightweight laptops — sort of like a netbook with everything upgraded. They are configured with low voltage, high powered processors, fast power-up, extended battery life, and fast storage, typically in the form of SSDs. Late last year, Acer introduced its first Ultrabook: the Aspire S3 series. Going with the trend toward lean and mean, it's only 0.51 to 0.68 in thick and weighs in at just under 3 lb. The operating system is stored in a 20 GB SSD, allowing it to resume from sleep mode in as little as 2 sec. A battery charge is said to last up to 50 days if unused, and you get about 6 hr of continuous computing time.

Other features include a second-generation Core i5 processor, a 320 GB hard drive, and 4 GB of memory. The S3-951 includes Dolby® Home Theater v4. The display is a 13.3 in HD backlit LED with 1366 by 788 maximum resolution. Prices start at $899. For details, visit us.acer.com.
ANOTHER WEARABLE COMPUTER

Every now and then, someone gets the bright idea of producing a wearable computer, hoping that it will be commercially successful. So far, hope has never triumphed, but it appears to be NEC's turn to take another whack at it. What could make a difference here is that the NEC TeleScouter is designed for industrial settings where its price tag (reported at about $5,200 for the unit plus a hefty $25,000 for the software) may not be prohibitive. Plus, the company is aiming to sell only 1,000 systems in the next three years. Perhaps the machine's most interesting feature is the head-mounted AIRScouter display which projects images into the wearer's eyes. It incorporates clear glasses that allow the operator to view the real world at the same time. The system can also be provided with camera-equipped glasses, microphones, and earphones, making it more flexible for remote field operations. The unit is also Bluetooth enabled. The innards include an ARM 500 MHz processor and 256 MB of memory running Windows Embedded CE 6.0. The system weighs about 430 g (15 oz) and will operate better than 4 hr on a charge. At present, the machine appears to be available only in Japan, but NEC's issuance of an English-language press release would indicate that TeleScouter will eventually find its way to our shores.

UPGRADE YOUR IPAD 2 WITH A KEYBOARD CASE

One of the nice things about an iPad is that it eliminates the bulky keyboard. However, one of the drawbacks to the iPad — if you do any amount of real typing — is that it eliminates the bulky keypad. Hence, meet the Keyboard Folio product from the folks at Belkin (www.belkin.com). What you get is an integrated Bluetooth keyboard and protective folio case designed specifically for the iPad 2. It features TruType™ keys that are designed to offer a computer-style typing experience, as opposed to flimsy rubber keyboards. The tri-fold design hides the keys and protects the screen when the unit is not in use. You can pick one up for an MSRP of $99.99.

NEW 2-CHANNEL, 16-BIT PC OSCILLOSCOPE

There's nothing wrong with a standard benchtop 'scope, but if you already have a Windows-based PC at hand, you might want to hook it up to one of Pico Technology's units and save some space. The newest is the PicoScope 4262 which is a two-channel, 16-bit very high resolution oscilloscope (VHRO) with a built-in low distortion signal generator. With a 5 MHz bandwidth, it can be used to analyze audio, ultrasonic, and vibration signals; characterize noise in switched mode power supplies; measure distortion; and perform a range of precision measurement tasks. It includes a function generator and arbitrary waveform generator that has a sweep function to enable frequency response analysis. According to the company, it also offers mask limit testing, math and reference channels, advanced triggering, serial decoding, automatic measurements, and color persistence display. It also provides a menu of 11 automatic frequency-domain measurements in spectrum analyzer mode. You can get further details at www.picotech.com and compare the 4262 to other models in the lineup. This one has a street price of a little more than $1,100, including two probes and a carrying case.
CIRCUITS AND DEVICES CONTINUED

TAKE 4-D PHOTOS

O
nce in a while, something comes along that is actually new and amazing, and it appears that Lytro (www.lytro.com) has come up with one. It seems that Lytro CEO Ren Ng wrote an award winning Ph.D. thesis on light field theory back in 2006 while at Stanford University. He then founded a company to move his research from theory to product, and the result is the Lytro light field camera. Like your garden variety Canon or Nikon, it is a digital still camera. However, a light field camera — also known as a plenoptic camera — uses an array of microlenses to capture all the rays of light in a scene; in this case, 11 million rays. It uses "a light field sensor that collects the color, intensity, and the direction of every light ray flowing into the camera, capturing a scene in four dimensions." As a result, one can focus and refocus the image even after shooting the photo. If you share an image online, Lytro's "light field engine" travels with it so that the picture remains interactive on such devices as browsers, tablets, and mobile phones.

Other features include an 8x optical zoom lens with a constant f/2 aperture, instant turn-on and an instant shutter (with no need for autofocus; there is no shutter delay), high sensitivity for use in low light situations without a flash, and storage of 350 or 750 images, respectively, with the 8 GB or 16 GB models. Because all Lytro photos inherently include 3-D data, the company will be releasing software later this year that will allow 3-D viewing and scene perspective shifting. Lytro offers a free software download that enables users to import pictures from camera to computer and share them via Facebook, blogs, email attachments, and so forth. At this moment, it is Mac OS only, but a Windows version may be available by the time you read this. You can pick one up for $399 or $499, depending on the memory capacity. For an onscreen demonstration, visit www.lytro.com/living-pictures.

INDUSTRY AND THE PROFESSION

BING LOSING BIG

If the headline above makes you think more of "White Christmas" than search engines, you're not alone. Nobody else is using Microsoft's www.bing.com, either. Reportedly, Microsoft has lost about $5.5 billion (you read that right) since launching the service in 2009 and continues to sink cash into it at a quarterly rate of almost $1 billion. It doesn't appear to be a major concern, though, as the company has some $53 billion in cash on hand and even boosted its quarterly dividend last year — up to $0.20 per share. Well, it's only money.

REDEFINING BASIC MEASUREMENT UNITS

L
ast November, the International General Conference on Weights and Measures (CGPM) approved a plan to redefine four of the seven base units of the International System of Units (SI), which seems to be creating a frenzy (or at least a lack of a universal yawn) in the field of metrology. The units in question are the kilogram, the ampere, the kelvin, and the mole, all of which "will be redefined in terms of invariants of nature; the new definitions will be based on fixed numerical values of the Planck constant, the elementary charge, the Boltzmann constant, and the Avogadro constant, respectively."

There isn't much here to interfere with your sleep patterns, however. For one thing, the changes won't take place until a range of technical requirements for agreement and some unspecified uncertainties are met, and the CGPM won't even meet again until 2014. For another, the changes will be so infinitesimal that almost no one will
notice the difference. The definition of a kilogram, for example, will change by only a few parts in 100,000,000, so a kilo of bananas will cost pretty much the same. But we thought you’d like to know. Updates will be available at the National Institute of Standards and Technology (NIST) website (www.nist.gov).
Here are four significant improvements in this new version of the GPS simulator. First, it displays the altitude on a PC while producing its GPS output for the tracker under test. This lets you monitor the progress of the balloon flight in real time. Second, five additional GPS sentences are included in the output. The output includes the $GPGGA, $GPGGL, $GPGSA, $GPGSV, $GPRMC, and $GPVTG sentences. The only one not included is the $GPZDA sentence (in order to keep the output updating properly at once per second). However, if one is willing to forgo the simultaneous reporting of the balloon’s flight on a PC, then the simulator can also reproduce this last remaining sentence.

Third, not only does the new version of the simulator produce output like a GPS, it also replicates a GPS attaining and losing its position lock. A pushbutton switch toggles the lock status of the simulator (between lock and no-lock, and back again) and a bi-color LED indicates the current lock status. The fourth and final improvement permits the simulation of a neutrally buoyant weather balloon. A tap on the second pushbutton switch launches the balloon simulation and software controls the ascent, buoyancy, burst, descent, and landing of the balloon.

A second bi-color LED displays the balloon’s flight status (on the ground, rising, floating, descending, or landed). All these features fit inside of 1,617 bytes of the PICAXE-18M2 memory so there’s room to grow the simulation. The simulator runs off a nine volt battery and terminates in a female DB-9 connector that directly plugs into any GPS tracker’s serial GPS port.

The simulator’s electrical design is pretty simple because software does most of the simulation work. The LP2950 is a low drop-out voltage regulator and permits the circuit to continue operating down to about 5.3 volts (a pretty dead nine volt battery). The voltage regulator uses the 22 µF electrolytic capacitor to reduce voltage ripples. You can replace this capacitor with a larger value capacitor if that’s what you have handy. The resistors act as either pull-ups or current limiters.

The pushbutton switches signal the GPS simulator when to toggle the position lock and when to launch the balloon. At startup, the GPS simulator has no lock and the balloon is on the ground. The Lock LED is red to indicate no position lock and the GPS output is set to a string of sentence headers and zeros. Once the Lock button is pressed, the simulator begins producing a valid series of

As some readers may recall, I wrote about a simple GPS simulator back in July ’09. That version used a PICAXE-08M and transmitted only the $GPGGA sentence. It was a fine simulator for near space flight computers that only monitored this single GPS sentence. However, it wasn’t satisfactory for a GPS tracking system monitoring additional GPS sentences. Preventing the expansion of the GPS simulator was the limited memory of the PICAXE-08M. I can now report on a new and improved version of the GPS simulator that expands the original simulator into one suitable for testing most GPS tracking systems.
GPS sentences once every second and the Lock LED turns green. If the GPS lock is switched off after the balloon is already in flight, the simulator keeps track of its virtual position while the GPS simulator reports no position lock.

The balloon is virtually launched by pressing the Launch button. Until then, the GPS position doesn't change and the Status LED is off. The Status LED is green during ascent, toggles between red and green during float, is red during descent, and is off on the ground. The simulator's programming port is a female DB-9 soldered to the PCB, and it does double duty by sending position reports to a PC. The GPS output is through a second female DB-9 attached to the end of a two-wire cable. Attaching it to a cable makes it easier to plug the GPS simulator into a GPS tracker.

**MAKING THE GPS SIMULATOR VERSION 2.0**

The **Parts List** shows the components needed to make the GPS simulator, version 2.0. Many of them you'll already find in your parts bin.

All holes in the PCB (printed circuit board) are 0.032", except for the four holes used for strain relief. These holes (circled in Figure 3) are drilled larger at around 0.050" in diameter. After drilling the PCB, begin by soldering the resistors. Follow with the capacitor and then the IC socket. Do not insert the PICAXE-18M2 at this point. Next, solder the LP2950 voltage regulator and the two miniature momentary pushbutton switches. The Lock and Launch LEDs are bi-color LEDs, and the LED near the voltage regulator is a standard single color LED. Solder all nine of the female DB-9 connector pins to the PCB. Afterwards,
place the spacers between the wings of the DB-9 and the PCB, and bolt the DB-9 to the PCB with the 4-40 hardware. When soldering the electrolytic capacitor, voltage regulator, and LEDs, watch their polarities.

All that remains to solder are the wires for the battery snap and the GPS cable. The battery snap solders to the two solder pads at the top of the PCB diagram in Figure 3. The diagram shows the battery snap leads in red and black to match those in the battery snap. The other two wires (one black and one white) for the GPS cable appear at the bottom of the placement diagram. The other ends of these wires terminate in pins 2 and 5 of the DB-9 connector with the solder cups.

The wires in the battery snap and the GPS cable are strain relieved to the PCB so they will not break off from everyday use. Solder the nine volt battery snap first. Strip 1/4” of insulation from the ends of the wires (if they are not already stripped) and pass them through the large diameter strain-relief holes. Then, bend them over and insert the bare ends into their correct solder pads as shown in Figure 4.

I find leaving a small loop in the wires helps press the bare ends into the solder pad. After soldering both leads of the battery snap, trim the wires and pull the wires tight. There is no need for a loop after soldering since the large diameter hole in the PCB blocks most of the strain on the battery snap from reaching the soldered connection. Now, repeat this for the two wires of the GPS cable. These wires are about a foot long and have 1/4” of insulation stripped from both ends. Start by soldering one end of each wire to the PCB with the same method used for the battery snap. The other end of each wire solders to a solder cup in the DB-9. In Figure 4, placement diagram above, each wire is labeled with its correct solder cup – 2 and 5. Now, the GPS simulator is ready for its first test.

**TESTING THE GPS SIMULATOR**

The first test locates any electrical shorts in power. Use a DDM set for continuity and check between the positive and negative terminals in the battery snap. The meter should not ring. If it does, check the underside of the PCB for any solder shorting traces. Next, test that the proper voltage appears on the PCB. Set the DMM for voltage and plug a nine volt battery into the snap. Measure between pins 5 and 14 on the IC socket and verify that relative to pin 5, there is positive five volts (give or take 1/2 volt) on pin 14. Remove the battery snap and install the PICAXE-18M2. The last test is a test of communications. Start up the PICAXE Editor and set it for the PICAXE-18M2. Type the following program into the editor:

```
PAUSE 1000
DEBUG
```

Plug the nine volt battery back into the battery snap and download this program. Verify that the editor programmed the PICAXE and that the Debug window has one update. At this point, you’re ready to download the GPS simulator code, which you’ll find at the article link. After downloading this code into the PICAXE, start up the terminal program in the editor by clicking the F8 key. Note that the Lock LED is red (no position lock) and the Launch LED is off (balloon on the ground). Unplug the programming cable from the DB-9 on the GPS simulator PCB and plug it into the DB-9 on the end of the GPS cable. The PICAXE terminal will now show data as in Figure 5.

If every test has been successful, then complete the last step in the construction of the GPS simulator. Squirt hot glue around the solder cups in the DB-9 of the GPS cable. Then, fill one half of the plastic DB-9 hood with hot glue and lay the female DB-9 into the hood. Top off any low spots with hot glue and then fill the other plastic DB-9 hood with hot glue. Close the second hood and bolt the entire assembly together. Finish by back-filling the DB-9 hood with additional hot glue. When back-filling the DB-9 hood, squirt glue into the opening in the back of the hood until the hot glue is about to run out.

**FIGURE 3.** Solder the components to the PCB as indicated in this placement diagram.

**FIGURE 4.** This image shows the insulated wire passing through the larger strain relief hole on the right. It then bends over, and the stripped end of the wire passes through the smaller solder hole to be soldered.
Now, let the glue cool.

**SOFTWARE SETTINGS**

The GPS simulator code runs a series of five nested loops to create the GPS time. The nested loops update the hour, minute, second, position, altitude, and wind speed. The code updates and sends position reports once per second. The parameters used to define the flight profile are located at the beginning of the code, so you won’t have to go digging around for them. The parameters are as follows.

- **StartHour:** The starting hour of the simulation in UTC (any value from 00 to 23).
- **Day:** The day of the month in the simulation (any value from 1 to 31).
- **Month:** The month in the simulation, in two digits (any value from 01 to 12).
- **Year:** The last two digits of the year in the simulation (any value from 01 to 12).
- **LaunchAltitude:** The elevation of the launch site in meters (any realistic value).
- **AscentRate:** The ascent rate of the balloon in feet per minute (any value from 300 to 1,200).
- **FloatAltitude:** The altitude float is to occur (any value lower than burst altitude).
- **BurstAltitude:** The altitude balloon burst is to occur (any value higher than LaunchAltitude).
- **DescentRate:** The descent rate of the parachute in feet per minute at sea level (any realistic value around 1,000).
- **RecoveryAltitude:** The elevation of the recovery site in meters (any value less than BurstAltitude).

The important thing to notice is that if the burst altitude is lower than the float altitude, the balloon will never become neutrally buoyant in the simulation.

**USING THE GPS SIMULATOR**

Disconnect the battery and plug the GPS cable into the GPS port of the tracker under test. Then, power-up both the GPS tracker and the GPS simulator. The GPS simulator immediately begins producing output like a GPS without a position lock.

When you push the Lock button, the GPS begins reporting a stationary position and the Lock LED turns green. There is some built-in jitter into the reports, so the reported position jumps around slightly like a real GPS. Once you push the Launch button, the balloon begins rising at the ascent rate set in the software. The Status LED is green indicating the balloon is ascending. Again, there’s a small jitter in the position reports just like a real GPS.

The balloon continues to rise at a constant rate until it reaches 40,000 feet. At this point, the ascent rate drops to 80% like a real balloon (the actual altitude this occurs at is variable, however, 40,000 feet is a good value). The balloon continues to rise until it reaches float or burst altitude. At float, the Status LED flashes alternately red and green. At burst, the Status LED turns red indicating descent.

The descent continues until the simulator altitude descends below the recovery elevation. The Status LED turns off and the GPS reports a fixed position with some jitter. The flight path and heading remain fixed; there is no accounting for changing wind directions in the simulation. However, the wind speed does change and there’s a mild jet stream at 40,000 feet.

A PC plugged into the simulator’s programming and running the PICAXE terminal program will display the current altitude of the simulation. There’s still room in memory for upgrades to the simulator. However, if you make significant changes, you may need to change the speed of the PICAXE or shorten the code to keep position reports appearing once per second. I plan to experiment with software upgrades to permit the testing of cutdown devices and of GPS-based amateur rocket altimeters. I’ll soon make a kit available on my website ([NearSys.com/catalog](http://NearSys.com/catalog)) for anyone who doesn’t want to go through the trouble of making the PCB or locating parts.

Onwards and Upwards, Your near space guide **NV**

**Neutrally Buoyant**

The neutrally buoyant condition occurs when the lift of the balloon is too close to the weight of its payload. Initially, the balloon slowly rises, perhaps on the order of 300 feet per minute. However, at some point during the ascent, the balloon’s lift is too weak compared to the balloon skin’s tension against expansion. At that point, the balloon stops rising and it just drifts downwind. Eventually, the batteries in the GPS tracker discharge and die. In enough time, solar ultraviolet will degrade the latex in the balloon and pop the balloon. However, by then, the balloon could be hundreds of miles away. The chances are now poor that you will ever recover the payload. To guard against this, always fill the balloon with enough gas to lift two or three pounds more than the payload weight. Otherwise, adding a cutdown device to the balloon load line is in order. After activation of the cutdown, the payload separates from the balloon and begins its descent. Without the weight of the payload, the balloon shoots rapidly upward in altitude and bursts. The GPS simulator version 2.0 described here permits you to test the operation of a cutdown device while it remains on the ground.
* Quality Quick Turn Sheet Metal Products & Precision Machined Parts

* No Minimum Quantity Restrictions

* Quote in 24 Hours or Less!

* Several Metallic & Non-Metallic Materials in Stock

Visit www.iitmetalfab.com today for your free, no obligation quote!

---

RS485/422/232/TTL

ASC24T
RS232 - 422 & 485 Converters

I-BS485HV
3 Port Isolated RS485 Repeater
$349.00

$45.00 board only

Converters / Repeaters
Multi-Repeaters / Hubs
Fiber Optics / Isolators
Extended Distance Units
Serial to Digital I/O
Large Multi-Drop Networks
Custom Units & Smart Units
Industrial, 3.0 KV Isolation
Call the RS485 Wizards
513-874-4796

RES  R.E. Smith Inc.
www.rs485.com

---

AP CIRCUITS
PCB Fabrication Since 1984

As low as...

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com

---

RS485/422/232/TTL
www.rs485.com  
Converters / Repeaters
Multi-Repeaters / Hubs
Fiber Optics / Isolators
Extended Distance Units
Serial to Digital I/O
Large Multi-Drop Networks
Custom Units & Smart Units
Industrial, 3.0 KV Isolation
Call the RS485 Wizards
513-874-4796

RES  R.E. Smith Inc.
www.rs485.com

---

Integrated Ideas & Technologies, Inc.

Metal Fabrication: www.iitmetalfab.com  SMT Stencils: www.integratedideas.com

Defense
Aerospace
Medical
Aviation
Communications
Maritime
Electronics

Serving All Industries
Satellites provide global coverage of clouds, water vapor, dust, smoke and the ozone layer. The colorful images provided by the data from these satellites looks very impressive. But satellite instruments don’t always stay calibrated and problems can occur when satellite orbits drift. Amateur scientist, Joe Novice learned about this when he heard a satellite scientist say that the global aerosol cloud formed by the eruption of a giant volcano had dissipated much sooner than expected. Joe suspected the satellite was simply wrong, but he was not a satellite scientist. How did he use some everyday items and several electronic components to prove he was right?

What’s your solution? See if you are correct at www.Jameco.com/search13 where you will find all three of Joe’s solutions.

Order Your Free Jameco Catalog!
Jameco.com/catalog
Synthesized FM Stereo Transmitter

- PLL synthesized for drift free operation
- Front panel digital control and display of all setings and parameters!
- Professional metal case for noise-free operation
- EMI filtering on audio and power inputs
- Super audio quality, rivals commercial broadcasts!
- Available in plastic kit or factory assembled export versions

For nearly a decade we've been the leader in hobbyist FM radio transmitters. We told our engineers we wanted a new technology transmitter that would provide FM100 quality series without the advanced mixer features. They took it as a challenge and designed not one, but TWO transmitters!

The FM30 is designed using through-hole technology and components and is available only as a do-it-yourself kit with a 25mW output very similar to our FM25 series. Then the engineers redesigned their brand-new design using surface mount technology (SMT) for a very special factory assembled and tested FM35WVT version with 1W output for our export only market!

All settings can be changed without taking the cover off! Enter the setup mode from the front panel and step through the menu to make all of your adjustments. A two line LCD display shows you all the settings! In addition to the LCD display, a front panel LED indicates PLL lock so you know you are transmitting. Besides frequency selection, front panel control and display gives you 256 steps of audio volume (left and right combined) as well as RF output power.

A separate balance setting compensates for left/right differences in audio level. In addition to settings, the LCD display shows you "Quality of Signal" to help you set your levels for optimum sound quality. And, of course, all settings are stored in non-volatile memory for future use.

The design includes EMI filters on line level audio and power inputs and a state of the art microstrip PCB inductor to reduce microphonics for the ultimate in alignment free operation. RF output is brought out to a professional, well shielded rear panel BNC connector for a perfect antenna match. Standard RCA connectors are used for left and right line level audio inputs. Both the FM30B and FM35WVT operate on 13.8 to 16VDC and include a 15VDC plug in power supply.

(Note: After assembly of this do-it-yourself hobby kit, the user is responsible for complying with all FCC rules & regulations and obtaining the proper permits for operation.

11.8 to 16VDC and include a 15VDC plug in power supply.

The design includes EMI filters on line level audio and power inputs and a state of the art microstrip PCB inductor to reduce microphonics for the ultimate in alignment free operation. RF output is brought out to a professional, well shielded rear panel BNC connector for a perfect antenna match. Standard RCA connectors are used for left and right line level audio inputs. Both the FM30B and FM35WVT operate on 13.8 to 16VDC and include a 15VDC plug in power supply.

HN Plasma Generator

Generate 2" sparks to a handheld screwdriver! Light fluorescent tubes without wires! This plasma generator creates up to 25kV at 20kHz from a solid state circuit! Build plasma bulbs from regular bulbs and more! Runs on 16VAC or 5-24VDC.

PG13 HV Plasma Generator Kit $64.95

Speedy Speed Radar Gun

Our famous Speedy radar gun teaches you doppler effect the fun way! Digital readout displays plays in MPH, KPH, or FPS. You supply two coffee cans! Runs on 12VDC or our AC125 supply.

S7G Speed Radar Gun Kit $69.95

Broadband RF Preamp

Need to "pick-up" your counter or other equipment to read weak signals? This preamp has low noise and yet provides 25dB gain from 1MHz to well over 1GHz. Output can reach 100MW! Runs on 12 volts AC or DC or the included 110VAC F5. Assemb.

PR2 Broadband RF Preamp $69.95

3-in-1 Multifunction Lab

The handiest item for your bench! Includes a KOH! compliant temp controlled soldering station, digital multimeter, and a regulated lab power supply! All in one small unit for your bench! It can’t be beat! 

LAB1U 3-in-1 Multifunction Soldering Kit $134.95

Follow Us and SAVE $$

Follow us on your favorite network site and look for a lot of super deals posted frequently exclusively for our followers!
Four Mode Vehicular Keyless Entry Test Set

Ah-ha, the conveniences of today's technology in our modern world! Voice recognition, LED's instead of incandescent bulbs, on-board computers, on-board hard drives, automatic parallel parking, wireless remote controls! They make it so simple, just have the "key" (called a key fob) somewhere in your pocket or purse, get near the vehicle, it knows that you are there! Touch the door handle and the vehicle unlocks. Get in and touch the door switch and the vehicle starts. And don't forget all the wireless controls for your house lights, building access and entertainment systems. They're so great... until they don't work!

Testing your system is easy. To test the complete key fob-to-vehicle and vehicle-to-key fob communications path just stand close to the vehicle with the WCT3 and your key fob in hand. Press the test button and the WCT3 will detect the presence of the vehicle.

The WCT3 test set is housed in a compact 2.25" x 4.6" x 9" case and is powered by a standard 9VDC battery (not included).

WCT3 Four Mode Keyless Entry Test Set Kit $95.99

Passive Aircraft Monitor

The hit of the decade! Our patented receiver hears the entire aircraft band without any tuning! Passive design has no LO, therefore can be used on board aircraft! Perfect for air-shows, hears the active traffic as it happens! Adjust for tuned output and has 4 buttons! Patent Pending.

ABM1 Passive Aircraft Receiver Kit $89.95

Voice Activated Switch

Voice activated (VOX) provides a switched output when it hears a sound. Great for a hands free PTT switch or turn on a receiver/circuit! Directly switches relays or low voltage loads up to 10mA. Run your system with the VOX function on the key fob. The same functionality testing can be done with IR key fobs. The modified IR signal is detected and will illuminate the IR test LED on the test set. If you know a few "secrets" you can also see if the tire pressure sensors/transmitters are generating signals or the built-in garage door opener/remote or your garage mirror is transmitting a signal! Bond WC. The majority of building wireless access systems also utilize 125 kHz. Just hold the test set near the building access sensor and the WCT3 will detect the 125 kHz signal. That will help you troubleshoot door access systems. If your WCT3 is not testing any other 315 MHz, 433 MHz, 125kHz, 20kHz and IR wireless control system to verify generation of a signal. The WCT3 test set is housed in a compact 2.25" x 4.6" x 9" case and is powered by a standard 9VDC battery (not included).

VS1 Voice Switch Kit $9.95

RF Preamp

The famous RF preamp that's been written up in the radio & electronics magazines! This super broadband preamp covers 100 kHz to 1000 MHz! Unconditionally stable gain is greater than 16dB while noise is less than 4dB! Covers 100 KHz to 1000 MHz! Unconditionally stable gain is greater than 16dB while noise is less than 4dB!

SA7 RF Preamp Kit $19.95

Mad Blaster Water Alarm

If you need to simply get attention, the "Mad Blaster" is the answer, producing a LOUD ear shattering racket! Super for car and home alarms as well. Drives any speaker. Runs on 9-12VDC.

MB1 Mad Blaster Water Alarm Kit $9.95

Water Sensor Alarm

This little $7 kit can really "bail you out"! Simply mount the alarm where you want to detect water level problems (sump pump)! When the water touches the contacts the alarm goes off! Sensor can even be remotely located. Runs on a standard 9V battery.

MK108 Water Sensor Alarm Kit $6.95

Air Blasting Ion Generator

Generates negative ions along with a hefty blast of fresh air, all without any nozzles or DC voltages. Generates 75V AC negative at 400uA, and that's lots of ions! Includes 7 wind tubes for max air! Runs on 12-15VDC.

IG7 Ion Generator Kit $64.95

Tri-Field Meter Kit

"See" electrical, magnetic, and RF fields as a graphical display on the front panel! Use it to detect these fields in your house, find RF sources, you name it. Featured on CBS's Ghost Whisperer to detect the presence of ghosts! Reel's 4 AAA batteries.

TFM3C Tri-Field Meter Kit $74.95

Electrol Condenser Mic

This extremely sensitive 3/8" mic, has a built-in FET preamplifier! It's a great replace them with dynamic or a perfect answer to add a mic to your project. Powered by 3-5VDC, and we even include coupling cap and a current limiting resistor! Extremely popular!

MC1 Mini Electrol Condenser Mic Kit $3.95

RF1 Sniff-It RF Detector Probe

Measure RF with your standard DMX or VOM! This extremely sensitive RF detector probe connects to any voltmeter and allows you to measure RF from 50kHz to over 1GHz! So sensitive it can be used as a RF field strength meter!

RF1 Sniff-It RF Detector Probe Kit $22.95

VS1 Voice Switch Kit $9.95

OBDDI CarChip Pro

The incredible Onboard diagnostics that has everyone talking! Once plugged into your vehicle it monitors up to 300 hours of trip data, speed, braking, acceleration, RPM and a whole lot more. Reads and resets your check engine light, and more!

B226 CarChip Pro OBDDI Monitor-Addr $79.00

RF1 Sniff-It RF Detector Probe Kit $22.95

Get the brand new 2012 Ramsey Hobby Catalog today! Filled with some of the newest gadgets around! Order your new 2012 catalog or download the PDF at www.ramseykits.com/catalog!

800.446.2295
www.ramseykits.com
FLICKERING LAMP CIRCUIT

Q
I use several oil lanterns that I modified and wired to use 12 volt low voltage landscape sockets and lamps running off a transformer for my Halloween display. I would like to get the bulb to flicker like a flame to be more natural. Is this possible and can I do it cheaply?

— Bill Bartsch

A
In December ’05, the late T.J. Byers published a neat circuit for an LED; I have modified it with a power transistor to drive a lamp (see Figure 1). The transistor is rated at 1/2 amps and may need a heatsink. You can tweak the values to get the effect that you like. The four oscillators are running at different frequencies and are not synchronized, so there is a variation in the light intensity. If your lamp requires more than 500 mA, a TIP41 transistor will work but you will need to tweak the resistors 5-8 for the best effect.

FM RADIO PROBLEM

Q
I purchased and built an FM radio in kit form from one of your advertisers (RamseyElectronics.com) and, although it works just fine, I encounter a problem when I increase the volume. It ‘skips’ to the next station. Can you please point out where the problem may be? I’m using a 12 VDC @ 100 mA power adapter for it, instead of 9 VDC as indicated on the schematic that came with the kit.

— Michael Williams

A
I have reproduced the suspect part of the circuit in Figure 2. The DC from the discriminator is fed back to the tuning diode to hold the station on frequency. If the coupling capacitor, C9, is leaking or in backwards, changing the volume will drag the DC voltage down and cause the frequency to change. Check C9 and if it is not backwards, replace it with a tantalum 10 µF, 16V. The circuit should work okay on 12 volts.

WATER SOFTENER QUESTION

Q
I recently received an ad for an “electronic water softener.” It appears to consist of a coil around a copper pipe with a controller attached. My questions are: Does this have a scientific basis? Wouldn’t the copper pipe shield the water from magnetic and electrostatic fields? Do you know of anyone that has used one?

— Kenneth Keck

A
I have never heard of electronic water softening. You are correct; the copper pipe will shield the water from AC fields. I am sure the ad did not explain how it works because it doesn’t. I found out that this device is marketed by Eden Pure — the company that markets the electric heater.
LED CIRCUIT WANTED

Q I want to ask a few questions on powering red, white, and blue LEDs in parallel. I want a circuit that will power 13 red, 20 white, and 12 blue T-1 LEDs connected in parallel. I didn’t use current drop resistors and they powered up nicely with six AA batteries. They are all medium brightness LEDs, but I had to hook each color up separately to its own supply (two AA batteries for each color). I was finding that when hooked up in parallel, the power took the path of least resistance and lit the red LEDs only. My question is how could I use the lightest weight battery and smallest battery possible to power all three different color LEDs with the least amount of parts, having the T-1 LEDs wired in a string of parallel 1206 or 0805 surface-mount parts okay also and with one battery even a coin cell type or similar but at the same time wanting to use T-1 style LEDs and a circuit for surface mount LEDs also would be appreciated as well as a circuit for T-1 LEDs so that I have the freedom to use either type LEDs. Keeping in mind the smallest circuit board and lowest weight batteries and least amount of parts possible.

— Scott Gates

A The different color LEDs operate at different voltages. Red is the lowest at two volts, green is 2.2 volts, and blue is 3.5 volts. Since white LEDs are combinations of red, green, and blue, white LEDs operate at the highest voltage of 3.5 volts. Operating a red LED at three volts without a series current-limiting resistor will overheat it and it will not last long. The most efficient way to operate LEDs is in series, but you will need a boost power supply. If you run them all in series, you will need 138 VDC and the power required at 20 mA is 2.76 watts. The input power will be about three watts, and the current from two AA cells will be one amp. An alkaline cell is rated 2.5 amp-hours, so such a system will run for about two hours. A coin cell doesn’t have the current capability to run this circuit. My design (Figure 3) uses two switching regulators because high voltage MOSFETS won’t turn on with three volts. The first switcher boosts three volts to 15 VDC, then the second boosts 15V to 140 VDC. A current sense resistor (R9) provides feedback to regulate the current, not the voltage. If the load is open, the voltage will go to the max which may be more than the output capacitors can stand. I used National Semi’s Web Bench to design the switchers; the LM2735Y is a low voltage type with an internal switch. The LM3478 is a current mode switcher; R6 is the current sense which shuts off Q1 at the current peak. The drive voltage output of the LM3478 is 7.5 volts; so that has to be sufficient to turn on Q1. The parts are all surface-mount; see the Parts List in Figure 4.

NEED USB TESTER

Q I have a USB port built into the radio with Bluetooth; went to try it out and almost set my car on fire. I was charging a cell
The USB port supplies five volts DC to power accessories and charge batteries. You can measure the voltage with a voltmeter; if it is more than 5.5 volts, 2.3 amps, and a message came up “cablenot recommended for this device.” I got a new charger and the phone works like the day I bought it. I tried plugging my cell phone into the USB port in the car was damaged so I went to the dealer to have it tested and they told me the only way to test the USB port is to pull the dash apart. I was wondering if there is a device I can plug into the USB port to tell me if it works properly because I really don’t want to have the dash pulled apart unless I have to. I also wanted to know if there is a difference in USB cables because I was thinking that maybe it was only made to charge a cell phone and not made for data. I would like all the help you can give me on this.

— Robert Schwartz

Q: I am having trouble finding a replacement inductor for repairing an audio output for a TV. The TV is using a TI part #TPA3123D2 class D amplifier. My problem is that I found two bad inductors, 22 uH. I went on TI’s website and found a substitute inductor: TOKO part #A7503AY-220M. The problem is that I can’t purchase this part separately. I went on the Coilcraft website and found a substitute for that inductor. My question is: How important is it to use the TI recommended part? Do you have any suggestions as to where I can purchase the inductors? What parameters are important besides the inductance value, current rating, and minimum resistance? Here are the two subs I found: Coilcraft #DROI10-223L and TDK #TSL112RA-220K2R9-PF.

A: The class D amplifier operates at 250 kHz, so the self-resonant frequency of the 22 uH inductor has to be higher than that. Not a problem for any 22 uH coil, I am sure. If the amp is operating from 12 VDC, full bridge, and with an eight ohm speaker, the peak current will be at least 1.1 amps so the rating of the inductor should be higher; say two amps. The resistance should be as low as possible, certainly not over one ohm. The other criteria is that it has to fit in the space provided. For a through hole part, Mouser part number 434-07HCP-220K, 22 uH, .055 ohms, 2.3 amps, should do it. If the inductor is surface-mount, you will have to search either Mouser or Digi-Key to find a compatible form. Or, you can solder the leads of the through hole part to the surface-mount pads and use RTV to support the inductor. If the amp is operating from 24 volts, half bridge, the current will be the same and the Mouser part and the two subs that you found will work.

# replacement inductor
ARDUINO DEVELOPMENT KITS

ARDUINO - Simple to Advanced Projects

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs (including Jaycar stepper motors). Arduino projects can be standalone, or they can be communicated with software running on your computer. These Arduino development kits are 100% Arduino compatible. Designed in Australia and supported with tutorials and guides. See website for complete Arduino range.

“Eleven” Arduino-compatible development board
XC-4210 $29.00 plus postage & packing

An incredibly versatile programmable board for creating projects. Easily programmed using the free Arduino IDE development environment, and can be connected to your project using a variety of analog and digital inputs and outputs. Accepts expansion shields and can be interfaced with our wide range of sensor, actuator, light, and sound modules.
- ATmega232PV MCU running at 16MHz
- 14 digital I/O lines (6 with PWM support)
- 8 analog inputs

EtherMega, Mega sized Arduino compatible with Ethernet
XC-4256 $89.75 plus postage & packing

The ultimate network-connected Arduino-compatible board: combining an ATmega2560 MCU, onboard Ethernet, a USB-serial converter, a microSD card slot for storing gigabytes of web server content or data, Power-over-Ethernet support, and even an onboard switchmode voltage regulator so it can run up to 28VDC without overheating.
- ATmega2560 MCU running at 16MHz, large Flash memory
- 10/100base-T Ethernet built in
- 54 digital I/O lines
- 16 analog inputs
- MicroSD memory card slot
- ProtoShield area
- Switchmode power supply

USBड्रैड, Arduino-compatible with USB-host support
XC-4222 $50.50 plus postage & packing

This special Arduino-compatible board supports the Android Open Accessory Development Kit, which is Google’s official platform for designing Android accessories. Plugs straight into your Android device and communicates with it via USB. Includes a built-in phone charger.
- ATmega232PV MCU running at 16MHz
- USB host controller chip
- Phone charging circuit built in
- 14 digital I/O lines (6 with PWM support)
- 8 analog inputs
- MicroSD memory card slot

EtherTen, Arduino-compatible with Ethernet
XC-4216 $50.50 plus postage & packing

This Arduino-compatible development board includes onboard Ethernet, a USB-serial converter, a microSD card slot for storing gigabytes of web server content or data, and even Power-over-Ethernet support.
- ATmega232PV MCU running at 16MHz
- 10/100base-T Ethernet built in
- Used as a web server, remote monitoring and control, home automation projects
- 14 digital I/O lines (6 with PWM support)
- 8 analog inputs

ProtoShield Basic
XC-4214 $3.25 plus postage & packing

A prototyping shield for the Eleven (XC-4210) and USBড्रैड (XC-4222) both featured above. Provides plenty of space to add parts to suit any project, keeping everything neat and self-contained. Includes dedicated space to fit a power LED and supply decoupling capacitor.
- Goldplated surface

Large Dot Matrix Display Panel
XC-4250 $29.00 plus postage & packing

A huge dot matrix LED panel to connect to Eleven, EtherTen and more! This bright 612 LED matrix panel has on-board controller circuitry designed to make it easy to use straight from your board. Clocks, status displays, graphics readouts and all kinds of impressive display projects are ready to create with this display’s features.
- 32(L) x 16(W)mm high brightness Red LEDs (512 LEDs total) on a 10mm pitch
- 5V operation
- Viewable over 12 metres away
- Handy 16-character by 2-line display ready to plug straight in to your Arduino, with a software-controllable backlight and 5 buttons for user input.
- 2 rows of 16 characters
- Supported by a driver library
- Software-controlled backlight
- Reset button
- Dimensions: 85(W) x 54(H) x 12(D)mm (24mm including header pins)
3-IN-1 DIGITAL METER ADAPTER FOR GEIGER COUNTERS

The new digital meter adapter (DMAD) from Images Co. is now a universal expansion module. It enhances the capabilities of Geiger counters. Using the TTL digital output pulse from any Geiger counter, it is a:

1) Digital radiation meter, counting radioactive particle detections (pulses) and outputting the equivalent radiation level.
2) RS-232 interface adapter for computer graphing programs.
3) True random number generator.

The DMAD functionality includes a countable pulse resolution and range of one count per minute (CPM) for 9,360 counts per second (CPS). Radiation resolution and range is 1.0 uR/hr, or 655 mR/hr (metric: .01 uSv/hr or 6.5 mSv/hr).

An adapter plugs into the digital out of an analog Geiger counter and provides a random number each time a radioactive particle is detected. The Geiger counter is set up to read background radiation. The random numbers generated are truly random since they are based on naturally-occurring radioactivity.

The random number generator is compatible with analog meter Geiger counters that use either an internal tube or a wand with a shielded cable. Typically — depending on location — the generator will produce 20-40 random numbers per minute. TTL serial output random numbers are sent out via a TTL serial port that may be interfaced to a microcontroller or PC.

The digital adapter also provides an RS-232 computer output for use with Images’ Geiger counter graphing program. The Windows Geiger counter program is free to download for non-commercial use only. (Commercial applications must be licensed.)

The module may be purchased as a kit or assembled and tested.

For more information, contact: Images, Co.
Web: www.imagesco.com/kits/dmad.html

SUB-1 GHZ WIRELESS CONNECTIVITY MODULE AND BOOSTERPACK

Anaren, Inc., announces a new sub-1 GHz radio frequency (RF) module, based on the CC110L sub-1 GHz value line transceiver from Texas Instruments. Additionally, Anaren has developed TI’s RF BoosterPack plug-in board, containing the CC110L-based module which is compatible with TI’s MSP430™ microcontroller (MCU) Value Line LaunchPad development kit.

The new A110LR09A module-based 430BOOST-CC110L RF BoosterPack, provides electronic devices and equipment designers with a speedy and facile way to develop and test wireless solutions using the popular MSP430 LaunchPad environment. The A110LR09A module is a high performance, dual-band FCC-certified and ETSI-compliant radio module that incorporates TI’s CC110L low cost transceiver chip in the industry’s smallest package (9 x 16 x 2.5 mm). Operating in the ISM bands at 868/915 MHz, the A110LR09A is well-suited for applications such as sensor networks, industrial monitoring and controls, home and building automation solutions, and remote control toys, among many others.

Parallel to this initiative, Anaren will also be offering its own line of AIR-equipped BoosterPacks, featuring many of the company’s other AIR modules (including A1101R08A, A2500R24A, and A1101R09A).

As with all AIR modules, the A110LR09A AIR module is designed to help OEMs challenged with adding wireless capability to new or existing devices. Leveraging Anaren’s 40+ years in microwave/RF technology and TI’s renowned semiconductor technology, the AIR module product family offers:

- Easy integration and implementation on existing or newly-designed two-layer PCBs.
- Tiny common footprints across the product line.
- Pre-certified to applicable standards to save time and
SMD SOLDERING AND RE-WORK TOOL KIT

Global Specialties announces the addition to its product line of the new Model GST-SMT-1. The GST-SMT-1 is a high quality tool kit used to solder and rework surface-mount devices and circuitry. The set includes 11 tools commonly used for working with surface-mount components, such as anti-static tweezers with replaceable tips, suction pen with three different sized tips, soldering iron, and magnifier. The kit comes complete in a convenient molded plastic case. The GST-SMT-1 is perfect for educational experiments, engineering and development, and repair applications. The GST-SMT-1 also can be used to attach surface-mount components to Global’s own GSPA-K1 and GSPA-K2 surface-mount adapter kits. The suggested retail price is 79.75.

For more information, contact: Global Specialties 22820 Savi Ranch Parkway Yorba Linda, CA 92887 Web: www.globalspecialties.com

SINGLE-CABLE MULTI-RADIO MOBILE SEPARATION KIT

E4S, LLC has unveiled SwapMyRigs which standardizes single-cable installations of mobile radios with remotable control.

So you can swap radios without reinstalling cables. By routing all connections through a common cable, any radio with industry standard RJ jacks can be installed or replaced without using manufacturers’ proprietary multi-cable separation kits.

The photo illustrates the separation of a Kenwood TM-D710 transceiver from its remote controls. SwapMyRigs consists of two small field-configurable powdercoated steel boxes called SMRs; one is at the transceiver, the other is at the remote location, connected by a standard single-cable.

For more information, contact: Anaren
Web: www.Anaren.com

Continued on page 80...
The video reception on cable was terrific — since I bought an HD television at the same time — but the audio was an enigma. There was a tremendous variation in volume going from one channel to another, and the commercials seemed too loud. We never had such problems with off-the-air signals. That's because broadcasters can't risk transmitter over-modulation. At the same time, they want to keep their signals as high as possible to cover a wide audience. These competing demands force them to keep firm control of their audio signals. That doesn't seem to be the case for cable.

What does an electronic hobbyist do when faced with a technical challenge like this? He heads for the parts drawer and a soldering iron! Fortunately, I had a short career as a broadcast engineer so I knew a lot about the technology of automatic volume control. I also knew that I didn't want to build a compressor or a limiter. I really wanted a platform gain controller.

Compressors and Limiters

Circuits to regulate the amplitude of audio signals have been around since the vacuum tube days. They were patented as early as 1932 (H. A. Wheeler, "Volume Control," U.S. Patent No. 1,879,863, September 27, 1932). The most common types of audio gain controllers are compressors and limiters. A compressor modifies the dynamic range of an audio signal to make quiet passages louder and loud passages softer. As a side effect, it makes the audio signal sound louder overall. A limiter functions somewhat like a compressor, but it's designed to pass signals unchanged up to a threshold and then completely block signals from rising in amplitude above the threshold.

Figure 1 is a block diagram of the circuitry involved for both compressors and limiters. The differences between these is the threshold at which the device starts to control and the gain of the amplifier that drives the control signal.
for the voltage-controlled amplifier. There is also a
difference in the attack time constant (RaC) and release
time constant (RrC) of the RC filter in the control loop.

The attack time for both compressors and limiters is
fast (< 1 msec), but a compressor has a longer release time
than a limiter. The release time in a compressor is usually of
the order of a few hundred milliseconds, but for a limiter it
can be shorter than 10 milliseconds.

The operation of a compressor is shown in Figure 2. At
signal levels below a threshold, the input and output levels
track each other and the audio is unchanged. Above the
threshold, increasing the input level causes a smaller
change in output level. The ratio of input signal change to
output signal change is called the compression ratio. A 5:1
ratio means that a five dB increase in signal amplitude gives
just a one dB increase in output. In this way, loud signals
are suppressed.

A large compression ratio will effectively suppress loud
signals, but there are problems with too high a ratio. First,
you'll have no dynamic range for your audio, and
everything will sound too loud. There's also too much
amplification of low level signals that leads to what's
poetically described as "whoosh." Whoosh is the sudden
insult of system noise you'll hear during pauses in the audio
signal; for example, between sentences in speech or
between songs. Aside from the whoosh problem, when the
input signal reappears the attack time might be slow
enough to let a short over-amplitude burst pass through.

A high compression ratio also leads to waveform
distortion at low frequencies. What happens is that the
compressor tries to level the peaks and valleys of the audio
waveform itself. For such reasons, smaller compression
ratios are used. Since this isn't wholly effective in preventing
over-modulation of transmitters, the compressor is followed
by a limiter that absolutely prevents any signals from going
higher than a predetermined level.

The way a limiter processes audio is shown in Figure 3.
The action is just like a compressor below the threshold —
there's no change in the signal. Above the threshold, a
limiter acts like a compressor with a very high compression
ratio, essentially infinity. A limiter is designed to absolutely
block any signals above the threshold. Since limiters follow
compressors in an audio chain, the compressor does most
of the work and the limiter acts as a safety valve. In proper
operation, a limiter rarely goes above its threshold, and it
doesn't contribute to whoosh.

Platform Gain Principle

A compressor followed by a limiter was the usual signal
processing chain for most broadcast stations until the
1960s. At that time, a new type of audio controller hit the
market and started to dominate. This was the platform gain
controller, conceived by CBS Labs. CBS — a major
television and radio network with many corporate-owned
stations — had a vested interest in advancing broadcast
technology. CBS Labs was started in 1936, and it operated
in one form or another for 50 years. A CBS Labs employee,
Dennis Gabor, even received a Nobel Prize in Physics,
although it was for work in holography not related to CBS
Labs activities.

In 1959, CBS Labs introduced the first platform gain
controller — the Audimax Audio Gain Controller — that was
designed to replace compressors. CBS Labs also introduced
a Volumax line of audio limiters. These were still the
vacuum tube days, and the patents on the platform gain
principle — now long expired — show that (B.B. Bauer and
Arthur Kaiser, Gain Control Apparatus Providing Constant
Arthur Kaiser and Emil Torick, "Compensated Platform Gain
Emil Torick and Arthur Kaiser, Control Circuit for Restricting
3,398,381, August 20, 1968).

The platform gain principle is shown in Figure 4 which
has some similarity to the limiter transfer function of Figure
3, with a strange twist. Just as for a limiter, signals below a
threshold are passed without modification, and any signals
above a threshold are blocked. That's when the signal is
increasing, as from point B to point C. When the signal
pulls back from its peak at point C, the output changes in a 1:1 correspondence with the input, as it did before the threshold was reached.

It's only when the input signal is reduced quite a bit (at point D) that the gain control kicks in to maintain the output at a higher signal level. Finally, if the input signal is reduced quite a bit, at point E the output tracks the input. If somewhere in between — from point A to point B — the input level again increases, it takes a measured increase in the input signal before the gain control again kicks in to prevent over-modulation.

The regions B-C and D-E are the gain platforms, and the signal level difference between these two platforms sets the dynamic range for the controller. What all this means is that the dynamic range is preserved, while at the same time peaks are prohibited and average volume is kept high. The circuit effectively acts like a careful engineer who turns up the volume when the signal is too low, and turns it down when it gets too high.

**Circuit**

Some magic tricks are done with mirrors. The
platform gain trick is done with diodes.

We replace the simple resistor-capacitor time constant circuitry in a limiter with the network shown in Figure 5. As you can see in the figure, there are two capacitors; one of which is driven with a reduced voltage from a voltage divider composed of resistors R1 and R2. When resistor R1 is equal in value to resistor R2, the voltage on capacitor \( C_r \) is twice that of \( C_p \). That means that on any peak in the audio, \( C_r \) will always be at a higher voltage than \( C_p \).

Since the only connections to \( C_p \) are to the diodes and the high impedance input of the VCA, this capacitor has no discharge path, so it will hold the peak voltage. \( C_p \) will discharge only when the voltage on \( C_r \) goes lower than the voltage on \( C_p \). That's the only time when both capacitors can discharge through \( R_r \). It's the voltage difference between the two capacitors — set by the ratio of R1 and R2 — that fixes the gain platform. With equal values for R1 and R2, the platform is set for 50%, or 3 dB. Figure 6 and Figure 7 show the circuit diagram for the platform gain controller.

**Printed Circuit Board**

A printed circuit board layout that uses just a single copper plane is available with the article downloads; a parts placement diagram on the component side is shown in Figure 8. Figure 9 is a photograph of the populated circuit board, and Figure 10 shows the external connections to the circuit board, including one possible way to implement the power supply.

The current requirements for the positive and negative voltages are quite low — less than 25 mA each. Probably the best way to power the controller (as shown in the figure) is with a wall wart power supply feeding a DC-DC converter. One possible converter to use is a Murata NMH0515SC two watt converter that converts a five volt supply to ±15 volt supplies of 67 mA each. Use a regulated five volt wall wart, use the recommended bypass capacitors, and keep the converter away from the low level audio stages. Such converters do generate a little EMI.

The input pots should be of the "audio taper" variety, meaning you get more control at higher attenuation than with a linear pot. A ganged pot is preferred over two separate pots, but if this is in a set-and-forget application, that's not too essential. The meter (which is optional), is driven by the control voltage signal, so it always has a little bias voltage on it. I removed this with a zener diode (as shown in Figure 7), but many analog meters have a mechanical zero that will do just as well. My controller — which has an LED bar graph display instead — is shown in Figure 11. This was built into the box of a surplus oscilloscope plug-in module. Waste not, want not!

Figure 12 shows the meter calibration for my unit using the series zener diode. Figure 13 is artwork for the meter scale used in one of my controllers. Normal signal levels put the needle in the blue band. The scale is in dB gain reduction, which means that negative values — towards the top of the scale — indicate that the gain has been increased. Positive values — at the bottom of the scale — show that the gain has been reduced.
Circuit Operation

Referring to the circuit diagrams in Figure 6, the gain control element (IC3) is an LM13600 dual operational transconductance amplifier. This chip functions as a voltage-controlled amplifier with a current source-sink output. Fortunately, the chip has an internal current-voltage converter that makes it suitable as a stand-alone VCA for most applications. The simple current-voltage converter that’s used in this chip is not exactly suitable for professional audio applications, so a revised chip with better circuitry — the LM13700 — was introduced.

There are, however, still many sources for the older LM13600 chip (including my parts drawers), so the controller circuit adds its own current-voltage converters that allow the circuit to operate with either chip. These converters — which are just operational amplifiers (IC4-5) — also double as convenient output buffer stages. Other operational amplifiers (IC1-2) buffer the inputs. The circuit is socket compatible with the LM13700.

This is a stereo gain controller which is not exactly the same as two monaural gain controllers in the same box. In stereo, it’s important to maintain the stereo image, which is the virtual location of sound sources in the audio. In classical music, especially, you don’t want to have the violins or horns wandering all over the sonic space. For this reason, the control voltage for the processor is derived from the sum of both stereo channels. IC6a is the summer. The gain is reduced or enhanced in both channels simultaneously.

Rectification of the audio to obtain a DC control signal is done by a full wave rectifier built around IC6b-IC6c. This signal is fed to the platform circuit, and the peak holding capacitor Cp is buffered by a high impedance amplifier, IC7. A level-shifter derives a control voltage that’s suitable for the VCA and an optional meter.

You’ll notice that the platform circuitry is slightly

---

**FIGURE 10.** External connections to the circuit board, including one possible way to implement the power supply.

**FIGURE 11.** The controller — which has an LED bar graph display — was built into the box of a surplus oscilloscope plug-in module.

---

**PARTS LIST**

<table>
<thead>
<tr>
<th>ITEM</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1-R4</td>
<td>47K</td>
</tr>
<tr>
<td>R5, R6</td>
<td>220K</td>
</tr>
<tr>
<td>R7, R8</td>
<td>4.7K</td>
</tr>
<tr>
<td>R9, R10</td>
<td>1K, 20 turn Trim Pot</td>
</tr>
<tr>
<td>R11, R12</td>
<td>15K (Note 1)</td>
</tr>
<tr>
<td>R13-R17</td>
<td>47K</td>
</tr>
<tr>
<td>R18, R19</td>
<td>2.2K</td>
</tr>
<tr>
<td>R20, R21</td>
<td>10K</td>
</tr>
<tr>
<td>R22, R23</td>
<td>220K</td>
</tr>
<tr>
<td>R24</td>
<td>4.7K</td>
</tr>
<tr>
<td>R25-R27</td>
<td>10K</td>
</tr>
<tr>
<td>R28, R29</td>
<td>20K (Note 2)</td>
</tr>
<tr>
<td>R30</td>
<td>4.7K</td>
</tr>
<tr>
<td>R31</td>
<td>1 M</td>
</tr>
<tr>
<td>R32, R33</td>
<td>10K</td>
</tr>
<tr>
<td>R34</td>
<td>100K</td>
</tr>
<tr>
<td>R35, R36</td>
<td>470K</td>
</tr>
<tr>
<td>R37</td>
<td>22K</td>
</tr>
<tr>
<td>R38</td>
<td>25K (Note 3)</td>
</tr>
<tr>
<td>C1, C2</td>
<td>0.47 F; 35V</td>
</tr>
<tr>
<td>C3, C4</td>
<td>47 F; 25V</td>
</tr>
<tr>
<td>C5, C6</td>
<td>0.47 F; 35V</td>
</tr>
<tr>
<td>C7</td>
<td>2.0 F; 35V (Note 4)</td>
</tr>
<tr>
<td>C8</td>
<td>0.47 F; 35V (Note 4)</td>
</tr>
<tr>
<td>D1-D9</td>
<td>1N4148</td>
</tr>
<tr>
<td>IC1, IC2</td>
<td>LF351N</td>
</tr>
<tr>
<td>IC3</td>
<td>LM13600N or LM13700N</td>
</tr>
<tr>
<td>IC4, IC5</td>
<td>LF351N</td>
</tr>
</tbody>
</table>

---

**Notes:**

1) Can be a parallel combination of 22K and 47K.
2) Can be a series combination of two 10K resistors.
3) Can be a parallel combination of 33K and 100K.
4) Non-electrolytic capacitor. Polypropylene, polystyrene, or Teflon preferred.
5) Best power supply is either AC Motor with 2 watt 5V±15V DC/DC converter.
more complicated than what we discussed earlier in the article. There are a few more diodes and a transistor. The diode connection — four diodes (D3-D6) in series — may seem strange. The reason for this is that we're taking advantage of the fact that diodes are not ideal rectifiers. Silicon diodes have a threshold voltage of about 0.7 volts, below which they don't conduct well. Four diodes in series will bump that threshold up to nearly three volts. The reason for the diodes and the transistor that they drive is to add a "gate" function to the platform gain controller. As we know from our discussion of compressors, when the audio input signal is removed (as for the natural pauses in speech) the noise whoosh kicks in. The gate prevents the release capacitor from discharging through R31 during such pauses. The controller holds its last gain state.

Tuning and Performance

Once the circuit is assembled and passes the smoke test, it's time to adjust the trim pots R9 and R10. These pots balance the DC input currents of the operational transconductance amplifier. Since the outputs are DC coupled, this is done by putting a voltmeter at the left channel output, adjusting R9 for very close to zero volts, and then doing the same for the right channel and R10. If this isn't done correctly, you might force the processor into a very low gain state when you first apply an audio input. That's because the DC "thump" will be processed as a very large change in the input signal.

Figure 14 shows the input-output transfer function of the processor when the input signal is always increasing. You can see how a very high output level is maintained over a very large change in input level. If we decrease the input voltage, the signal would need to drop by 3 dB before leveling off, because of the gain platform.

Figure 15 gives you an idea of the audio quality of the platform gain controller. The spectra shown in the figure are taken using Audacity — an open source audio editing and processing program that works on both Linux and Windows. I use the Linux version. Voltage-controlled amplifiers are notorious for introducing distortion and noise, but the LM13600 seems well behaved.

The dark trace is the oscillator input to the controller, set at 440 Hz and a voltage to reach the controller operating midpoint. The light trace is the controller output. The controller adds just a little more noise and distortion than is already present in the laboratory-quality audio oscillator, or added by the PC sound card used as an input device. The second harmonic of 440 Hz is 880 Hz. Some of the extraneous peaks are from power line harmonics on the signal leads — the bane of any audio engineer.

Well, that wraps things up. I hope you've 'gained' some insight into stereo gain controllers. NV
BUILD THE GARAGE DOOR CLOSER

By Brian Beard

Discuss this article in the Nuts & Volts forums at http://forum.nutsvolts.com.

Automatic garage door openers can be a real blessing. I know, I feel that way every time I drive home in the rain. As convenient as they are, they are also a potential danger. Everyone — at one time or another — has gotten distracted and failed to close the garage door. An open garage door is an invitation for thieves and vandals. And don’t forget the mischief birds, cats, raccoons, and other animals can cause in your garage. So, I decided it was time to build a circuit that would close my garage door when it was unintentionally left open.

Investigating My Automatic Garage Door Operator

My goal was to design an automatic closer circuit that was simple and reliable. Because it is a safety device intended to compensate for my forgetfulness, I didn’t want to power it with batteries that I would have to remember to change in the first place. If the closer could be powered by the garage door opener itself, that would be ideal. I have a Genie radio-controlled automatic garage door operator. Like every automatic garage door operator, it also has a local normally-open pushbutton to open and close the door.

The Genie manual instructs the installer to connect the local pushbutton to the two screw terminals labeled PUSH BUTTON and COMMON. Using a voltmeter, I measured 19 VDC between COMMON and PUSH BUTTON. Using an oscilloscope, I observed this voltage to have approximately 2V of ripple (18V-20V) with peaks about every 8.3 milliseconds, suggesting it is the output of a full

<figure>

FIGURE 1. Voltage and current available on the PUSH BUTTON output of my Genie garage door operator as a function of load resistance.

</figure>

Beard - Garage Door Closer.qxd  12/7/2011  9:10 AM  Page 34
wave rectifier. Placing a variable load between PUSH BUTTON and COMMON, I recorded the voltage and current as a function of the load resistance (Figure 1). The garage door opener didn’t trigger until the PUSH BUTTON voltage was 0.3V or less. The maximum current of 10 mA available from PUSH BUTTON told me a low power closer circuit could be powered from that line.

**Design and Build**

I designed the closer circuit using a CMOS 4060, 14-stage binary counter and oscillator, shown as U2 in Figure 2. The datasheet gives the formula for the oscillator frequency as $F_{osc} = 1/(2.3*Rt*Ct)$. The values shown in the schematic approximate a frequency of 43 Hz. A 14-stage binary counter can count to 16K which means the most significant bit’s output (Q14) will go high at count 8192. When Q14 goes high, it turns on transistor Q1 which grounds the PUSH BUTTON input to the automatic garage door opener, causing the garage door to close.

How long does that take? At 43 Hz, the period (one count) is 23.25 milliseconds; 8192 counts $= 190.5$ seconds $= 3$ minutes and 10.5 seconds. Note that when Q1 turns on, it effectively shorts out the power supply for the closer circuit. The Vdd supply for U2 must persist long enough for the automatic garage door operator to react to the short on PUSH BUTTON. C1 stores enough charge to keep Q1 turned on for more than 100 milliseconds. D1 prevents reverse current flowing from C1 through U1.

Speaking of Vdd, the 4XXX series of CMOS logic is designed to operate from three to 15 volts. I included the 78L15 (U1) low power linear regulator to insure that Vdd...
did not exceed the maximum operating voltage. When I want to keep the garage door open for an extended period — like when I am mowing the lawn or washing the car — I can flip switch SW1 to the disable position. This removes power from the closer circuit and lights a red LED warning me the closer is disabled. Even when the closer circuit is disabled, the garage door can still be opened or closed by pressing pushbutton PB1. When you think about it, the closer circuit is nothing more than a fancy way of pressing the local pushbutton (PB1).

One thing I haven’t mentioned yet is how the closer circuit detects when the garage door is open. This is done via switch SW2 shown in Figure 3. I am using a normally-open rotary switch connected to a piece of music wire. When the garage door is closed, SW2 is open and the reset pin of the 4060 is pulled high by R1. When the garage door reaches the top of its travel, it pushes against the left end of the music wire which rotates the shaft of SW2 by 90 degrees, causing it to close. When SW2 closes, it grounds the reset pin of the 4060 allowing the count to start from zero.

I built the closer circuit on a small perf board (RadioShack 276-148) shown in Figure 4. The perf board and the manual controls were installed in a small (2.5 x 2 x 1.5 inch) aluminum project box that I mounted on my garage wall (Figure 5). This was a fun project to build. It’s actually useful and it doesn’t even use a microcontroller!

### Building Your Own

If you are thinking of building your own garage door closer circuit, here are some things you should consider:

- Check the voltage and current available at your automatic garage door operator’s local pushbutton. Depending on the voltage available, you may want to use a 78L12 or 78L05 instead of the 78L15.
- Determine the voltage and duration of the pulse needed to trigger your automatic garage door operator; mine was 0.3V or less for 100 milliseconds.
- Note that a CMOS 4060 can operate from three to 15 volts. A 74HC4060, on the other hand, can operate from two to six volts.
- The closing time for my circuit varies from three minutes and 15 seconds in the summer to three minutes and 45 seconds in the winter. I did not use temperature stable components. If you want a more consistent closing time, use temperature stable precision components for Ct and Rt.
- Shorter closing delays can be achieved by using one of the 4060’s less significant outputs — like Q12 or Q10 — or by increasing the frequency of the oscillator.
- Longer closing delays can be obtained by connecting the Q14 output of the 4060 to the clock input of another counter — like a 4040, 12-stage binary counter. If you do this, be sure to connect the reset pin of the 4040 to the reset pin of the 4060 so both chips start counting from zero.
- I used a rotary microswitch to detect the garage door was open because I happened to have it available. A magnetic switch — like those used as door and window sensors in alarm systems — would probably be more reliable in the long run.

Hope you find this circuit as handy as I do.
Cost effective prototypes and production runs

Powder-coated finish and panel thickness up to 10 mm now available

Choose from aluminum, acrylic or customer provided material

1, 3 and 5-day lead times available

Custom Front Panels & Enclosures

FREE Software

Designed by you using our FREE software, Front Panel Designer

- Cost effective prototypes and production runs
- Powder-coated finish and panel thickness up to 10 mm now available
- Choose from aluminum, acrylic or customer provided material
- 1, 3 and 5-day lead times available

Sample price $57.32 + S&H

Come see us February 14-16 at the Pacific Design & Manufacturing Expo
Anaheim Convention Center, Booth 3860

January 2012 NUTS & VOLTS 37
This project started out from a need to add externally adjustable volume and mute capability to my laptop computer. I listen to a lot of music and found that the existing touch-sensitive volume slider control on my laptop did not make it easy to do fine volume control adjustments. Plus, if I wanted to listen to music with the laptop lid closed, the volume controls were inaccessible and there was no way to hit the mute button. I needed an external control that was compact, powered from the USB port, and would not require any special drivers to be installed.

You are probably thinking “why not just get an MP3 player and your problems are solved?” Yes, that’s true. However, but if you are an avid electronics hobbyist, you know that isn’t how it’s done. Simple solutions are not as much fun as a project! Here was a good opportunity to learn about USB and end up with something of practical use. See my solution in Figure 1.

**DESIGN**

I chose the AT90USB162 microcontroller from Atmel’s AVR eight-bit line for this project. It has both a general-purpose microcontroller and a USB function controller on one chip. It can be powered from the USB port, so no additional power supply is needed. As an added bonus,
the AT90USB162 comes with a Device Firmware Update (DFU) bootloader that allows you to program the device using the USB port. All you need to do is download, install, and run Atmel's free DFU programmer software to load the firmware supplied with this article; no external programmer is required.

Once programmed, when the device is connected to a USB port it will tell Windows to automatically install the correct driver and your volume/mute controller will be up and running. Pretty neat!

The AT90USB162 is really a USB “toolkit.” It contains a USB Serial Interface Engine (SIE) that manages all of the basic USB communications functions:

- Error detection and correction.
- Data encoding and decoding.
- Bit stuffing and unstuffing.
- USB packet generation.

This takes away a lot of complexity from the software, leaving the implementation of the USB protocol and managing the encoder and pushbutton used for the volume control and mute. Only a few external components are needed to complete the design. Refer to the schematic in Figure 2.

A rotary encoder is used for data entry. The encoder controls the volume and the part specified contains a pushbutton switch that is activated by pressing down on the encoder shaft for the mute function. This keeps the overall design very compact.

**HUMAN INTERFACE DEVICE**

I planned the project around the USB Human Interface Device (HID) class driver built into the Windows operating system — no additional drivers are required. The HID class specification describes all sorts of devices including mice, keyboards, and joysticks, and includes the media controls needed for this project.

All of the information needed to identify our device as a HID is stored in the device's programmable memory in a data structure called a descriptor. Within the descriptor, there is an additional structure called a report that describes the features in the host's operating system that are to be controlled. The report for our device brings together some useful media controls and is one byte in...
length. Each bit — called a usage in USB language — defines a particular media control. The layout of the report is shown in Figure 3. I only needed to control the volume and mute functions for this project but if you want to experiment with the software provided, you can change the source code to control the other usages.

The media usages function likes push on/push off buttons. For example, to toggle mute on, the device’s firmware needs to send a “1” in bit location 0 to the host. The device must then send a “0” in that bit position to reset the toggle because the usages are triggered on the 0 to 1 transition.

When the device is plugged into a USB port, it receives power, resets, and runs through its initialization routine. The device then connects an internal resistor (1.5 KW) from the D+ line to the internal 3.3V regulator, creating a voltage change on the D+ line. This lets the host know that a full speed device has attached and causes a sequence of events to occur called enumeration. During enumeration, the host reads the contents of the descriptor. Once enumeration is complete, the device is configured and ready to send data to the host. To move data, the USB standards specify that HID devices use the Interrupt Transfer protocol.

**INTERRUPT TRANSFERS**

All USB devices operate on the same basic principle: The host makes a request for data and the device responds. So, how does the device signal the host when it has data available? The type of data transfer used by HID devices is called an interrupt transfer, but this is misleading. There is no mechanism for the device to call the host when it has data ready. To receive reports from the device, the host makes periodic requests for data — in our device, every 10 milliseconds — called a “polling interval.” When the device has data ready to send to the host, the device loads its data into a special memory location called an endpoint. During each polling interval, the host asks the device to send the data in the endpoint. If the device has no data to send, it responds with a Zero Length Packet (ZLP) instead.

To update a media control, the device firmware must read the encoder/pushbutton, set the correct bit in the report, and wait for the host to poll that information. Before the main program can reliably use the pushbutton and encoder values, the inputs need to be cleaned up.

**CONTACT BOUNCE**

Mechanical switches require contact debouncing to prevent false signals. Figure 4 shows an example oscilloscope capture of what the microcontroller sees when the mute button is pressed. The encoder switches have a similar contact bounce characteristic. The strategy is to wait until the switch has stabilized after the first closure has been detected, and I decided to implement debouncing in software. A hardware solution was possible but would have meant more components.

Debouncing is handled using one of the AT90USB162’s built-in timers to sample the switch input pins periodically. The encoder and the mute pushbutton require separate debouncing algorithms due to their different characteristics. Debouncing the pushbutton is simple. A software routine uses an eight-bit shift register to accumulate a sample from the

---

**FIGURE 3.** The device implements just volume and mute but if you are adventurous, you can hack the supplied source code to implement the other media controls shown.

**FIGURE 4.**
switch input every millisecond. A steady run of seven zeroes following the last bounce reliably signals the button has been clicked.

The encoder is more complex because it has two pins to work with at the same time, and either the on or off states may be valid. Every millisecond, a counter is incremented for a stable sample (i.e., the current value equals the previous value). Otherwise, it is decremented. This acts like a capacitor filtering out any contact bouncing. Once a number of stable samples have been achieved, the direction of rotation can be determined. I found the best compromise between responsiveness of the encoder and reliable sampling by experimenting.

ENCODER DIRECTION DECODING

The type of encoder specified in the Parts List is an incremental encoder and requires some logic to figure out if you are turning it clockwise or counterclockwise. Direction is determined by “exclusive or-ing” the encoder’s previous most significant bit with the current least significant bit. Figure 5 shows the sequence generated by the encoder in either direction.

To ensure the encoder logic reports the correct direction after initial device reset, the previous encoder value is initialized to the current port value during device initialization.

SOFTWARE

The software provided includes all of the functions needed to manage the HID protocol and switch interfacing. The module UsbHid.c exposes all of the functions needed to manage the USB interface and are summarized in Figure 6. The source code supplied with this article shows how to use the library.

Once the USB module has been initialized, the main loop waits for the device to be configured, at which point the host begins regularly polling the device’s endpoint. When the Windows host goes into stand-by, a USB signal is sent to the device to tell it to go into low power suspend mode. When the device detects this state, the device turns off the configured LED and executes the sleep instruction to halt the CPU. Power consumption in this state is less than 0.5 milliamps. When the host resumes, the device automatically wakes up and becomes operational.

CONSTRUCTION

The AT90USB162 does not come in a standard...
experimenter-friendly Dual In-Line (DIP) package. It is only available as a surface-mount device (SMD). SMD components are surprisingly easy to work with, and this package form should not deter you from building this project. An excellent tutorial on SMD soldering can be found at www.eevblog.com, episode #186. (Warning: Try not to get too distracted by all the other interesting electronics videos on Dave’s site until you have finished this project.)

I was able to make a single-sided PCB using the Press ‘n Peel Toner Transfer system for the prototype. Due to its simplicity, the device could easily be constructed using perf-board techniques. Figure 7 and Figure 8 show how I used the box described in the Parts List to build the prototype.

Begin assembly by soldering the AT90USB162 chip (make sure pin 1 is correctly oriented before soldering). Next, solder the SMD resistors and capacitors, crystal, encoder, reset pushbutton, and status LED.

Finally, feed the USB cable through a hole in the side of the box and then through the hole in the PCB that acts as a cable strain relief. Solder the cable wires to the board making sure that the color coded wires attach to the correct pads. Soldering mistakes or shorts could damage your USB port. I used a cable salvaged from an old USB mouse, but you can purchase an open ended cable for this purpose (see Parts List). Don’t close up the box just yet. You will need to access the reset pushbutton to program the device.

PROGRAMMING

Download the project files from the article link. Save the files on your hard drive in a folder called UsbMediaControl. Next, install the software to program the device. To do this, download and install the FLIP software from Atmel’s website (www.atmel.com). I recommend that you install the version that includes the Java runtime to avoid any compatibility issues.

Once FLIP is successfully installed, attach the device’s USB cable to a USB port on your computer and place the device in DFU mode by performing this sequence:

1. Press and hold the reset button.
2. Press and hold the mute button.
(i.e., press down on the encoder shaft).
3. Release the reset button.
4. Release the mute button.

If everything is going according to plan, your computer will load the DFU driver. You can confirm this by looking at Device Manager — you should see LibUSB-Win32 Devices entry in the list (see Figure 9).

Start FLIP and load the object file from the project files you downloaded earlier. Here are the steps:

1. Click on File -> Load Hex File. Navigate to the Release folder in the UsbMediaControl directory where you saved the project files and double-click on UsbMediaControl.hex.
2. Click on Device -> Select -> AT90USB162 to select the correct device.
3. Click on the USB connector icon on the menu, then select USB from the pop-up and click on Open when prompted.

The FLIP control panel will look like Figure 10 at this point. To load the firmware into the device, complete the following steps:

1. Click on Run. The firmware will load and the dots next to Erase, Program, and Verify should all turn green if everything is okay.
2. Click on Start Application. The blue configured LED on the device should light indicating the device is programmed, configured, and operational.

Try turning the encoder and operating the mute pushbutton (press down on the encoder dial) to confirm the device is working properly.

**FINAL NOTES**

The source code has been included so that you can experiment with different configurations. You can download Atmel’s (free) Studio 5 development environment that includes a C compiler and code editor. One thing to note is that you will have to change the Solution Configuration from “Debug” to “Release” to ensure the compiler is operating in optimized mode. This object code won’t run properly in the (default) non-optimized mode. NV
If you are interested in learning more about this technology and how you can participate by building your own low cost WSPR receiver, keep reading. We will cover all of that and more. Before we get to that, let’s talk about a special application. If you are an educator or know someone involved in teaching science, please take note. WSPR is not only for electronics experimenters, but can be used to help teachers motivate their students in science while presenting relevant technical methodology. Motivation will likely occur because WSPR encompasses a wide range of subjects such as electronics, communications, Internet, software, physics, geography, and more.

Philosophically speaking, modern scientific methodology is something we learn, not something we are born knowing. We have an inherent, untapped capacity to explore and comprehend the world around us. Curiosity is at the root of science. The question, “How does that work?” is key to that curiosity. Indeed this path has been followed from Ptolemy to Einstein to Hubble.

In the classroom, a WSPR receiver is simply connected to a wire antenna and a PC or MAC. As it runs, stations from around the globe (on certain bands) may be received. It is different than hearing commercial shortwave broadcasting because you are receiving data of a scientific nature. This is an exciting, motivating phase for the kids. They will want to know how it works. So, don’t be surprised if some of them ask to borrow the receiver to take home to use on their own.

A particularly nice feature of...
WSPR is that “receptions” (called spots) can be automatically downloaded to a central database (WSPRnet.org) with results shown on a large map of the world (see Figure 1). A flag will appear on the map to show your location and which stations you have received. Hence, exact transmitter locations can be determined and distances, headings, and other information reported. The world map can be zoomed to learn more about the geography and radio path found. Particularly thrilling for younger children is this reporting phase as they feel a sense of participation.

More details about this project (including how to obtain ready-to-go WSPR receivers, lesson plans for teachers, and more) are available at http://stellarwspr.com. WSPR signals are in the amateur radio part of the spectrum and transmitters will require a license in most jurisdictions. Happily, many WSPR transmitters are now operating around the globe to provide interesting “catches” for the kids.

Why Do We Need WSPR?

We know from our high school physics class that when a transmitter signal moves down a wire to an antenna that part of it is launched into space as an electromagnetic (EM) wave. This energy propagates through space in various directions dependent on many factors, like the antenna size and orientation. All kinds of things can happen to the signal after it is launched. It can be refracted and polarized, for example. For communication, another antenna some distance away from the transmitter can intercept part of the signal and receive a message if one is contained in the signal. For example, Morse code is sent by turning the signal on and off (modulating it). Of course, for modern radio communications more advanced modulation methods are used. In an ideal universe, the EM wave would propagate forever and still be able to be demodulated perfectly at almost any distance with just the tiniest amount of signal available. However, in our real world there is another factor. In addition to the things mentioned above, we need to consider the ramifications of electrical noise.

As the EM wave propagates, it spreads out in space and therefore diminishes in strength in any given direction. Simply stated, it gets weaker. To add to the problem, the ionosphere — which carries much of the high frequency signals — also is continuously changing, causing propagation variations like fading. Noise, on the other hand, is all around our real universe being generated from all kinds of things like appliances, radar, and natural phenomenon like lightning. The noise level continuously fluctuates and causes interference to our signal.

To partially deal with the noise and also to provide additional features, various modulation techniques have been developed. Two classic ones you are probably familiar with are AM (amplitude modulation) and FM (frequency modulation). Some modulation methods are better than others at dealing with noise, but they all get into trouble when the signal strength approaches the noise level.

What makes WSPR different? It’s the protocol. WSPR software utilizes an extremely narrow frequency band with specially-coded forward error correcting (FEC) and frequency shift keying (FSK). FSK is like very narrow band FM. This technique reduces errors and improves the possibility of copying the message in noise. The signal fits into a tiny 200 Hz segment. Within each segment, the signal bandwidth is only 6 Hz. This allows several tens of stations to coexist in a segment with minimal interference. There are 12 segments presently (denoted here as the WSPR bands), located within the radio frequency spectrum (see Figure 2).

The WSPR protocol is extremely effective at signal-to-noise ratios as low as –28 dB in a 2,500 Hz bandwidth. This is over 10 dB below the threshold of audibility. In other words, you can sometimes copy signals that you cannot hear. It is because of this capability that even low power WSPR signals can be decoded in the farthest reaches of the globe.

WSPR Acknowledgement and Usage

WSPR was conceived and put into operation by Joe Taylor (K1JT), an amateur radio operator (Professor of Physics at Princeton) and fellow amateur Bruce Walker.

![Figure 3. Typical sound card hookup.](image-url)
The open source software runs on Windows, Linux, and other systems. It is available from www.physics.princeton.edu/pulsar/K1JT as a free download.

WSPR utilizes the internal or external sound card of your PC/MAC (see Figure 3). These cards have good specifications. Foremost are the data converters providing a large 90 dB dynamic range which is used to the fullest. WSPR is programmed to send and receive messages through the card. Here’s more about WSPR protocol.

WSPR uses time-synchronized communications based on Coordinated Universal Time (UTC). Transmissions last somewhat less than two minutes each, nominally starting one second into an even UTC minute. Hence, it is important for your computer to be accurate to within a second or so of UTC. So, set your computer clock to an accurate source such as www.usno.navy.mil.

Since most computer clocks drift, you will need to update your clock periodically.

WSPR is designed to do just one thing: find a communication path. It communicates via specially formatted messages aimed at determining if a propagation path is open on a given transmitting frequency. Formatting contains a name, four character grid locator, and power level in dBm (decibels relative to one milliwatt). This information is compressed into 50 binary digits and encoded using a convolutional code of length 32 and rate 1/2. The resulting 162 bits are transmitted using four tone FSK at 1.46 baud. The least significant bit is defined by a pseudo-random sequence known by the software at both transmitter and receiver. It is used to establish accurate time and frequency synchronization. Long convolutional codes are advantageous since undetected decoding errors are rare. Normally, a Viterbi algorithm is used for decoding but due to complexity, the WSPR decoder uses a special sequential algorithm.

When a station is decoded, other information such as “receive location,” name, S/N ratio, and DT (time difference) is routinely logged. This information can be automatically downloaded to WSPRnet.org using the “spots” option. Your name will then be shown on a flag on a world map with others. Options on the website can be used to find the distance and direction of the station received.

**FIGURE 4.** Block diagram for the WSPR receiver.

![Block diagram for the WSPR receiver](image)

**FIGURE 5.** WSPR receiver schematic.
**WSPR Receiver**

What do we need to receive WSPR signals? First, realize that WSPR is transmitted as a single-side band (SSB) signal so it cannot be received with an ordinary AM receiver. Some shortwave radios have a BFO (beat frequency oscillator) and could be used if they possessed sufficient accuracy and stability. Also since the signal is in a very narrow band and almost inaudible, manual tuning would be difficult. Ham band receivers are another option, but more costly. So, it appears that finding a low cost WSPR receiver can be problematical. Is there another approach? Yes! Let’s design our own unique WSPR receiver from the ground up. On the hardware side, we need to build a WSPR receiver with high accuracy, stability, and sensitivity at low cost. In addition, it should not require complicated alignment or adjustment procedures involving special equipment. Quite a task.

Recall that the function of the receiver is to convert the WSPR radio signals to audio for the computer sound card. A good candidate for these requirements would be some type of fixed, direct conversion (DC) radio. DC receivers convert radio signals to audio directly without any intermediate stages. A block diagram of the proposed radio is shown in Figure 4. Here is how it works.

RF signals from the antenna are bandpass filtered at the WSPR receive frequency to prevent overloading the mixer from strong out-of-band signals. A local oscillator (LO) tuned to the WSPR transmit frequency is mixed with the RF. Since the two frequencies are separated by 1,500 Hz, audio centered at 1,500 Hz is produced at the output of the mixer. Finally, this audio is fed to the computer sound card. WSPR software has a 200 Hz bandpass centered on 1,500 Hz with narrow digital filters. So, if we get the audio to the sound card there is a good chance it will be decoded. Implementing the block diagram requires some unique thinking. First, we do not want an adjustable LO as this would require special equipment. Also, we do not want special coils that need to be tuned because measuring these would be difficult for many hobbyists. Figure 5 shows the circuit conceived. It functions like the block diagram.

Two changes need more explanation. First, the LO is replaced with a programmed oscillator. This removes the need to adjust an oscillator. Programmable oscillators are essentially custom programmed crystal oscillators. Secondly, the input bandpass filter is now a crystal tuned to the WSPR frequency. Crystals provide a narrow bandpass function without using conventional tuned circuits. However, special crystal frequencies are required. Crystals for the 20 and 30 meter WSPR bands will be available as noted later. Here is a description of the circuit in Figure 5. From the antenna, the signal is passed through the crystal filter to a FET amplifier. Next, it is fed to one input (pin 1) of U2 — the SA612 mixer. The SA612 functions well in this application since it has gain (unlike passive mixers) and is protected from out-of-band signals by the crystal filter. Programmable oscillator U1 — programmed to a specific WSPR transmit frequency — feeds pin 6 of the mixer. The result is a differential audio signal on pins 4 and 5 of the mixer. This signal is further processed by U3 and presented to the sound card via JP1 and J2. Select the “line” or “microphone” jumper for your computer sound card.

The crystal filter may need peaking via the 20 pF variable capacitor CV. Since this is a sensitive place in the circuit, use care to avoid false readings. Adjust for maximum audio at 1,500 Hz. One way is to look at the audio spectrum with a free program like “Spectrum Lab,” from www.qsl.net/dl4yhf. A better method of alignment is to receive a nearby signal generator set to the WSPR receive frequency. Adjust CV for maximum while viewing the 1,500 Hz audio spectrum peak.

The circuit is powered by five volts through an AC to DC wall mount converter. Such converters can be troublesome. Check its voltage under load before using. Exceeding 5.5 volts may damage some parts. Using a computer USB port may work. We had good experience powering from the USB ports of some older laptops, but experience with other computers has produced strong interference from the USB port. In these cases, use the five volt wall mount converter.

For demonstration purposes, a breadboard of the 20 m receiver was built on a circuit strip shown in Figure 6. This technique, generally speaking, is not a good method for RF circuits but it worked here. A PCB is a better option and will be available also noted later.

Figure 7 gives the Parts List for the project. Most
parts for the receiver are readily available. Programmable oscillators are available from Digi-Key and others as a custom order. You need to specify the package (usually eight-pin DIP) and frequency. Some special crystals are available on the Internet. A PCB and most of the critical parts such as crystals and programmable oscillators for the 20 and 30 meter bands will be available on eBay, including ready-to-go receivers. Search for WSPR. Also check http://stellarwspr.com for the latest information on this project.

### WSPR Software and Receiver Operation

Using the WSPR software is easy. To get started, download and run the software from www.physics.princeton.edu/pulsar/K1JT/. Two windows will appear on your screen. The black one is for error messages. Figure 8 shows the main window. To check operation, you can play back an audio sample included with the program. If you want to check out actual band activity, use WSPRnet.org for current spots.

Assuming you have a receiver, connect it to a good antenna and the audio output to your sound card. A long antenna high in the air working against a good ground works best, but even a random length wire works. If possible, try to make your antenna one-quarter wave long as it will better match the input. For example, on the 30 meter band make the wire 7.5 meters long (24.6 feet). It cannot be over stressed that a good ground helps enormously. Even a long ground wire on the floor helps. Keep the antenna away from the computer. Using a short

---

**FIGURE 7.** Receiver Parts List.
coax cable may help.  
Now, set your computer clock to UTC within one second or so.  Correct UTC can be found on the Web. We use the Naval observatory at www.usno.navy.mil. Now comes the anxious part. You need to wait for an even numbered minute (0, 2, 4, etc.) for the software to begin receiving. Watch your computer clock. At this point, the message in the lower right corner will change from “waiting” to “receiving.” The lower left corner will specify the level of the input audio in dB. This can be adjusted with your sound card recording level control.  
A level of 12 to 19 dB works well, even though the software says otherwise and produces a red window. There are a few other things you can do, like putting in your grid location and call (or name) if you want to report your spots. But that will come later. For now, sit back and watch for your first catch.  

Final Comments  
WSPR is amazing technology. The simplicity of the WSPR receiver described here makes it attractive as a low cost radio propagation tool. With a bit of luck, it will motivate experimenters, teachers, and students to begin exploring the many facets of science and engineering touched upon here.

As stated, the question, “How does that work?” is indeed the key to curiosity. Hopefully, it will help the young stay interested in science.  
This receiver is only the first step. Other projects involving more sophisticated receivers and transmitters will surely follow. There was recently an article “Chips In Space” by Mason Peck (IEEE Spectrum August ’11) which described how satellites the size of small integrated circuits could revolutionize the way we explore space. Project Sprite involves designing and later on launching many thousands of low mass chips (under 50 milligrams) for the single purpose of monitoring space with simple sensors. Each chip will have a weak radio transmitter and signal barely discernable from noise. Sophisticated signal processing software will be needed to pull the signals out of the noise.  
Perhaps a youngster intrigued by WSPR will be the future engineer to design that software.  

FIGURE 8. WSPR operating window.
This is an adjustable timer circuit. The time that the output pulse is HIGH on pin 3 of the 555 timer IC is controlled by the potentiometer setting.

**Build the Circuit.** Using the schematic along with the pictorial diagram, place the components on a solderless breadboard as shown. Verify that your wiring is correct.

**Do the Experiment. Theory:** This is a 555 timer IC set up as a monostable multivibrator, or timer. Monostable refers to the idea that the output stays at LOW until triggered to go HIGH. Once triggered, it remains HIGH for a period of time and then returns to LOW. The duration of the HIGH on pin 3 depends on the values of R2, R4, and C1. The potentiometer needs to be set in the full clockwise position to start. When you press pushbutton S1, you put a LOW (a negative) on pin 2. This 'triggers' the timer. It now puts out a HIGH on pin 3 for a period of time, depending on how you have adjusted the potentiometer. Capacitor C1 can be replaced with larger values in order to get longer periods of time.

**Procedure:** Connect a nine volt battery to the battery snap. Press and release S1 and observe that the red LED is lit for a period of time and then shuts off by itself.

These experiments are provided by GSSTechEd at www.gssteched.com

You can order parts for this experiment from their website as follows:

**Resistors, 1/2 watt 5%**
- GK01049 220 ohms
- GK01065 1,000 ohms
- GK01085 6,800 ohms

- GK33008W 100K Potentiometer w/wires
- GK05005 100 mfd Electrolytic Capacitor
- GK06001 Jumbo Red LED
- GK14004 555 Timer IC
- GK25006 N/O Pushbutton Switch w/wires
- GK35002 Nine volt Battery Snap
- GK45011 Seven Wires 4” long

**Wires**
- W1 = 30c and 28h
- W2 = 19a and 30b
- W3 = 19h and 30h
- W4 = 10i and 30j
- W5 = 7j and 10j
- W6 = 11d and 17d
- W7 = 16g and 19d

**Battery**
- Black = 36g
- Red = 36e
SHOWCASE

WEIRD STUFF WAREHOUSE
Software, Computers, Electronics, Equipment, Dickies
WE BUY/SELL EXCESS & OBSOLETE INVENTORIES
FREE COMPUTER AND ELECTRONICS RECYCLING
GRANT 10,000 SQ. FT. A-15 SECTION
384 W. Caribbean Dr. Sunnyvale, CA 94089
Mon-Sat 9:30-6:00 Sun 11:00-5:00
(408) 743-5650 Store x324
WWW.WEIRDSTUFF.COM

MATCO SECURITY SOLUTIONS
4 Ch Mobile DVR Card
- 4 Ch Audio & Video
- MPEG4 Compression
- 3/4 ATA Hard Drive Input
- Up to 8GB SD Card
- 9PS Functionality
- 60 FPS Record Frame Rate
- 720x480 Record Resolution

 chaired

$599

30A PWM DC Motor
Speed Control
Control the speed of
your electric motors or
the brightness of your
bright bulbs.

QKITS.COM
Power Supply 12V/24V Load Voltage

$24.95

Battery rebuilding service
Dead Batteries? Don’t toss them.
Send them to us – our rebuilds are
two standard specifications.

www.primecell.com

PowerBotix
Robotic Power System
A powerful set of components designed to
calculate your mobile project.

USB connection to your on
board processor allowing your processor:
full control of the platform’s power system.

www.powerbotix.com

CHIPINO
The Arduino Style Module for the Microchip PIC User
$19.95

www.chipaxe.com

Extreme Motor
Speed Control!
MegaMoto - Motor Control for Arduino™

• 6V-28V - 15A / 30A peak
• Single H-bridge or dual half
• Current and Temp protected
• Jumper select Enable and
PWM pins

$250

www.robotpower.com

Digital Oscilloscope
50MHz - Up to 100ns/div
30 Mpoints - 16M readings
800x400 - 40k V/div

$599

SDPSI
800-819-8900
FREE CATALOG

RF SPECIALISTS
ZigBee Solutions ProBee Series
High Performance and Robust Data Transfer Capability
ZigBee Solutions

LEMOSS
www.lemoss.com
(516) 239-1467
sales@lemoss.com

SHOWCASE Jan12.qxd  12/7/2011  11:34 AM  Page 51
Recap

Over the past several months, we looked at Digital I/O (DIO). In Part 1 (October ‘11), we looked at the software side of DIO as it is done in the Arduino using sequentially numbered pins on the Arduino board. We wrote a library of functions similar to the Arduino DIO library functions but used regular C concepts and tools (AVRStudio, WinAVR, avrdude, avrlibc, etc.).

In Part 2 (November ‘11), we saw that the Arduino pins are simple abstractions of the deeper AVR microcontrollers’ concept of ports that are eight-bit arrays of pins. We wrote a library that specifically handles ports and their pins as they are used by raw AVRs.

Last month, we dropped the abstractions and looked at how DIO is really done in AVRs using the tools available in C, without having to write special libraries to manipulate the ports and pins for DIO. We also went deep into bitwise operators — a painful but necessary bit of learning that is required if you really want to know what C programming is all about (and microcontrollers in general, for that matter). We applied all that to a simple chaser light application that you can find at: http://code.google.com/p/avrtoolbox/. You can test it in hardware with the BreadboardArduino Projects kit available from the Nuts & Volts webstore.

We are now going to have some fun and apply what we’ve learned about DIO by building something really useful: an open source project that has an LCD and five buttons used to navigate through menus shown on the LCD. You can think of this as a tiny computer terminal and keyboard; a very minimalist (read cheap) one for the AVR. Like other Smiley’s Workshop projects, you can get the parts kit from the Nuts & Volts webstore, as well.

Theory Section: Digital I/O in C

The C programming language knows nothing about AVR DIO. C is a hardware independent programming language and runs on any computer. The AVR is a specific computer that does the same sorts of things that are done by other microcontrollers (such as the 8051 or PIC), but it does those things using different hardware. C abstracts the sorts of things you can do with a computer into a higher-level concept that mimics a generic computer. C leaves it up to the compiler to convert the C code into the actual
assembly language instructions that a particular computer uses.

C does such a good job of abstracting the way computers work that it is often called a generic assembly language. In early October ‘11, Wired Magazine posted a eulogy to Dennis Ritchie, the father of C:

Ritchie’s running joke was that C had “the power of assembly language and the convenience of... assembly language.” In other words, he acknowledged that C was a less-than-gorgeous creation that still ran very close to the hardware. Today, it’s considered a low-level language, not high. But Ritchie’s joke didn’t quite do justice to the new language. In offering true data structures, it operated at a level that was just high enough.

Our job is to take the generic things that C can do and use them with the specific things that our AVR can do. Herein lies the rub. AVRs can do a LOT, and as I’ve said before, one of the biggest problems with these things is that they can do so many different things in so many different ways, that it is a chore to figure out which of these many is ‘best’ for a given application. The datasheets are the ultimate resource. Unfortunately, they tell you how to do everything, but don’t give much of a hint as to which things you should choose as the best way to do the task at hand. If you haven’t already done so, I recommend that you get the datasheet for the ATmega328 from www.Atmel.com and skim through it—especially the Digital I/O section. What you’ll notice is that the datasheet offers so much bounty that it is nigh impossible to sort out what we really need to do to get these pins inputting and outputting, much less use them with the specific things that our AVR can do. Herein lies the rub. AVRs can do a LOT, and as I’ve said before, one of the biggest problems with these things is that they can do so many different things in so many different ways, that it is a chore to figure out which of these many is ‘best’ for a given application. The datasheets are the ultimate resource. Unfortunately, they tell you how to do everything, but don’t give much of a hint as to which things you should choose as the best way to do the task at hand. If you haven’t already done so, I recommend that you get the datasheet for the ATmega328 from www.Atmel.com and skim through it—especially the Digital I/O section. What you’ll notice is that the datasheet offers so much bounty that it is nigh impossible to sort out what we really need to do to get these pins inputting and outputting.

Let’s take the bitwise stuff we learned last month and see what you’ll notice is that the datasheet offers so much bounty that it is nigh impossible to sort out what we really need to do to get these pins inputting and outputting. Let’s take the bitwise stuff we learned last month and see how that is used to set up the DIO registers.

**AVR Registers**

The AVR uses eight-bit memory locations called registers to set the functionality of each of its peripheral devices. We’ve sort of hit at AVR registers in other Workshops, but let’s take them on again; this time from the perspective of how we will use them with C and bitwise operators.

Register space is located near the beginning of the AVR memory space. There are a total of 86 registers listed for the ATmega328. They sport names like SPCR, TCNT0, EEARH, EIFR, TIFR2, and so forth. Many of these registers have bits that you set or clear to activate or deactivate a specific AVR function. These bits also have names like TWAM6, COM2B1, ADIE, PCINT22, SP2, ACBG, and so forth. The point I’m making here is that there are hundreds of acronyms used to control the various AVR functions, and neither you nor I have a chance in heck of remembering even a small fraction of them. Thus, it is a painful necessity to get intimate with the datasheet. This pain can be somewhat alleviated by using the Atmel Application notes, forums like www.avrfreaks.net, and tutorials like this one, but ultimately if you are going to get proficient with microcontrollers you have to get real friendly with datasheets.

**AVR I/O-Port Registers**

If you refer to the I/O-Ports section of the datasheet and look at the Register Description, you’ll see 10 registers listed. We are interested in nine of these registers that are used to control Ports B, C, and D. Each port has a PORTx data register, a DDRx data direction register, and a PINx port input pin address register. The bits of each of these registers maps directly to the external I/O pin on the device as shown in Figure 2. The PORTx register is used primarily to write data to these pins, the PINx register to read the external state of the pin, and the DDRx to set the direction of the pin. [We discussed the electrical aspects of I/O pins in Part 1 of the DIO series].

**Using AVR Registers in C**

You might look at this and think something like, ‘Okay, I want to write a 1 to pin 0 of Port B,’ so I write:

```
// WRONG!
PORTB0 = 1;
```

But (you complain), the datasheet says the name of that pin is PORTB0, so how come this doesn’t work? The answer is that because in the avrlibc I/O header, the #define for PORTB0 is 0. So, all you’ve written is: 0 = 1; which it isn’t. All the PORTB0 define does is provide an alias for the pin number which is 0. So, how do you set PORTB0 to 1?

January 2012 NUTSVOLTS 53
Oh bother! That requires those darn bitwise operators. If you think setting the bit to 1 looks weird, how about clearing it to 0:

```c
// Clear bit to 0
PORTB &= ~(1<<PORTB0);
```

Yeah, looks sort of like what we learned last month with the bit_set(p,m) and bit_clear(p,m) functions – only now we’ve taken the armor off. This is real C folks. Tighten your seat belts.

### Setting the Pins for Input or Output

Since we want to set the pins to output to control our LCD display, we read in the datasheet:

The DDxn bit in the DDRx Register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

Set the DDRx bit for the pin of interest to 0 and it is an input. Set it to 1 and it is an output. Let’s replace the x with B and see how it is done in our very own port_pin_mode function [int8_t port_pin_mode(uint8_t portx, uint8_t pin, uint8_t mode)] from the elementary digitalio library we discussed in the November and December Workshops:

```c
if ( mode == INPUT ) // set DDRB bit to 0
{    DDRB &= ~(1<<pin);
} else // ( mode == OUTPUT ) // set DDRB bit to 1
{    DDRB |= (1<<pin);
}
```

This would be a good time for you to open up port_pin_mode.c [it is in the avrtoolbox repository] in Programmers Notepad to see the actual implementation. So, memorize this: In 0 out 1. Or IzeroOone. Or Izzy struck out once. Or just keep the datasheet handy which is what I do because I can never remember which does what.

Despite the datasheet, we see that it really is simple to set up and use digital input and output, so why even bother with the elementary digitalio library functions? Yeah, my point exactly, but since the Arduino has it and one of my goals with the elementary library – as I say elsewhere – is to provide a C transition for Ardu-refugees, I’ve included it in the avrtoolbox. Use it. Look at the source. Then move on.

### Moving On

For those who want to use C as is was intended (okay, that’s an opinion), we initialize the LCD control pins by providing aliases for the relevant port, pin, and data direction registers. Then, we use bitwise operators to set them as shown next. To use the LCD, we must set up four data pins and two control pins which we alias to the definitions in the avrlibc output header file for the ATmega328 (iom328p.h located in your WinAVR directory ..\avr\include\avr\).

```c
// Define the specific ports and pins used for the LCD
#define LCD_D4_PORT PORTD
#define LCD_D4_DDR DDRD
#define LCD_D4_PIN PD5
#define LCD_D5_PORT PORTD
#define LCD_D5_DDR DDRD
#define LCD_D5_PIN PD4
#define LCD_D6_PORT PORTD
#define LCD_D6_DDR DDRD
#define LCD_D6_PIN PD3
#define LCD_D7_PORT PORTD
#define LCD_D7_DDR DDRD
#define LCD_D7_PIN PD2
#define LCD_E_PORT PORTB
#define LCD_E_DDR DDRB
#define LCD_E_PIN PB3
#define LCD_RS_PORT PORTB
#define LCD_RS_DDR DDRB
#define LCD_RS_PIN PB4
```

To initialize these pins to outputs, we use:
// set LCD DDR pins to 1 for output
LCD_D4_DDR |= (1<<LCD_D4_PIN);
LCD_D5_DDR |= (1<<LCD_D5_PIN);
LCD_D6_DDR |= (1<<LCD_D6_PIN);
LCD_D7_DDR |= LCD_D7_PIN);
LCD_E_DDR |= (1<<LCD_E_PIN);
LCD_RS_DDR |= (1<<LCD_RS_PIN);

Now the registers are set up so that we can control
the LCD. Of course, controlling an LCD of the HD44780
variety is moderately complex, but we'll get to that later.

**Lab Section: The LCD Navigator**

**Assemble the LCD Navigator Projects Kit**

The LCD Navigator Projects Kit (shown in Figure 4)
is available from the Nuts & Volts webstore.

**Figure 4. LCD Navigator parts kit.**

**Figure 5. LCDNAV schematic.**
Instructions on assembling the board are available on www.smileymicros.com under the LCDNAV menu. The schematics for this board are shown in Figure 5.

**LCD Hardware: The HD44780 LCD**

I read a book (I think it was David Brin’s Practice Effect) where some primitive people found a digital watch with an LCD display. They were amazed that whoever made the thing was able to train all the little black bugs to run around and align themselves in such peculiar patterns. That’s the extent of the detail I’ll give on the underlying technology of LCDs. We’ll concentrate instead on using C to train the little black bugs to do our tricks.

We are lucky since Hitachi developed a simple way to control the LCD that has now become an industry standard for low cost character LCDs: the HD44780 driver/controller chip that you’ll find built into our display. They provide a parallel control interface that can send data in either eight-bit or four-bit chunks, and control the communication with enable and read strobe lines. Since we like to save pins in our AVR designs, we will use the four-bit mode. Of course, all that brain fatiguing stuff we learned about bitwise operators is going to come in handy.

**Wiring LCDNAV to the Arduino**

Well, after all that preaching to get folks to drill down through the simpler library functions and use the underlying C, we are going to do our first demonstration of the LCD Navigator with the Arduino! Not really. We will only be using the board and do the code in C using AVRStudio, WinAVR, and avrdude. The Arduino is an easy to use development platform and you don’t have to use the Arduino IDE or libraries — you can use it with plain old C. You can see how to wire this up in Figure 6 and Figure 7.

**LCDNAV wiring to the Arduino**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>DB7</td>
</tr>
<tr>
<td>3</td>
<td>DB6</td>
</tr>
<tr>
<td>4</td>
<td>DB5</td>
</tr>
<tr>
<td>5</td>
<td>DB4</td>
</tr>
<tr>
<td>11</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>RW</td>
</tr>
<tr>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>12</td>
<td>RS</td>
</tr>
<tr>
<td>6</td>
<td>LFT</td>
</tr>
<tr>
<td>7</td>
<td>UP</td>
</tr>
<tr>
<td>8</td>
<td>CTR</td>
</tr>
<tr>
<td>10</td>
<td>DWN</td>
</tr>
<tr>
<td>9</td>
<td>RGT</td>
</tr>
</tbody>
</table>

**LCD Software**

**Using the LCD**

We will use our avrtoolbox LCD elementary library which somewhat duplicates the function in the Arduino LiquidCrystal library (but in a more generalized fashion) to use with the regular AVR C tools: AVRStudio, WinAVR, and avrdude. The source code is located in avrtoolbox\libavr\testers\source \lcd_hd44780. I owe a debt to Peter Dannenger for his LCD tutorial on AVRFreaks. His code provided a good starting point for porting the Arduino LiquidCrystal functions [www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&p=828978]. You can find the source code for this library at http://code.google.com/p/avrtoolbox/libavr/source/driver/external_hardware/lcd-hd44780. The library has the following functions:

```c
lcd_init()
lcd_clear()
```
As shown in Figure 8, you can access the avrttoolbox documentation at www.smileymicros.com/avrttoolbox.html.

**Yes, but can you use it with the Arduino?**

I’m really not trying to make folks lives more complex than necessary, but you can use the LCD with the Arduino LiquidCrystal library. It is wired up for it, anyway. The only caveat is that their code is for a 16x2 (16 characters, two lines) LCD while our LCDNAV board uses a 8x2 LCD (it is a lot cheaper). You need to change one line in each example: from lcd.begin(16, 2); to lcd.begin(8, 2);. Note that this doesn’t make a difference in some of the examples since they are hardwired to 16 characters. I suggest sticking with the avrttoolbox code for now.

Well, as usual, we stop in the middle of things. For now, you can wire this up as shown and use it with several applications at http://code.google.com/p/avrttoolbox/. These include tester programs for the LCD and Nav button libraries and an LCDNAV_demo program in the avr_application directory. Have fun playing with it and we’ll get more of the details in later Workshops. Next month, we’ll look at the Navigator buttons and a menu application for the LCD Navigator project.

Questions? Nuts & Volts is hosting forums for its writers and you can find mine at http://forum.servomagazine.com. If you want a really quick response — especially to a question not directly related to an article — you can put on your biohazard suit and start a thread at www.avrfreaks.net. (First, read my blog entry that will tell you why you need the biohazard suit at http://smileymicros.com/blog/2011/01/24/using-an-internet-forum.) If you just can’t wait and want to get a leg up on all this serial stuff and real C programming for the AVR (while helping support your favorite magazine and technical writer), then buy my C Programming book and Butterfly projects kit and the Virtual Serial Port Cookbook at the Nuts & Volts webshop. NV
PROP TALK

From time to time, I just want to tell my Propeller project what to do — and it seems others do, as well. In recent months, I’ve worked with many who have asked about passing information between a PC and the Propeller, or even Propeller to Propeller. My focus will be on the former, that is, using a terminal to exchange information with the Propeller. What I needed was a protocol, and there are certainly any number of those available. Cutting right to the chase, I created my own called HFCP: for Human Friendly Control Protocol. Yep, cheesy name — but very easy to use. Now, this isn’t going to be appropriate for everything you develop, but for small projects that want to “talk” to you via your PC, this works really well.

So, what makes HFCP human friendly? Well, it’s text based so that we can use a simple terminal to exchange information. Secondly, it uses (by default) decimal values so we don’t have to decode anything. If you’ve ever looked at a MODBUS ASCII stream, you know what I’m talking about. Of course, I’d love to have you think that I’m some wildly clever guy. The fact is, I’m a regular guy who knows a good idea when he sees one. The idea for my protocol structure came from GPS.

GPS strings (NEMA 0183) are easy to read: plain ASCII text with fields separated by commas and terminated with a carriage return. The values are decimal — easy. Having worked on a GPS parser last year, I knew that if I adopted GPS string properties I could adapt some existing code. As you know, the devil is in the details and it did take a bit of time to get everything together — especially with the features I decided to add — but it’s done now, ready to use, and it works very nicely.

HFCP BASICS

A message from the system master (usually a PC) will start with the ‘>’ character. The header is followed by two or more fields, and the fields are separated with commas. Spaces and a few special characters in the message are tolerated. The structure is as follows:

>`target, command, parameters, value1, … , value8<CR>

In my stock version, I can support a message with up to eight parameters. If your applications tend to require more values, you can easily change the P_MAX constant in the object. What I should note is that — other than the terminating CR — all fields are numeric. Thus we have our first, “Shoot, how do I get around that?” moment when actually working through a project.

If we want to transmit text in a message, there is a special message type: The parameters value is specified as the “@” character and the text is placed in the fourth field. Later, we’ll see how we can determine which kind of message was received (numeric versus string).

Some messages will not require any
data, so the shortest possible message contains only the target address and the command value.

**HFCP IN ACTION**

In order to keep this very lightweight, the HFCP object is simply a set of methods that we will use with our favorite serial driver. Using a PC, for example, that will usually be `FullDuplexSerial` or one of its many derivatives. As you saw in the previous discussion, the message contains the target address so it is intended to operate on 'networked' systems. I do a fair bit of work with RS-485 and have written a half-duplex driver (see `jm_hd485.spin`) that handles the hardware handshaking of the RS-485 circuitry. You'll find that file in the downloads, and Figure 1 is my favorite circuit for connecting the Propeller to a half-duplex RS-485 buss.

In order to use HFCP, we have to do a little bit of work — it's not simply "plug and play," and neither is it difficult. Here's a simple shell example that we might use:

```spin
pub main | t, c

serial.start(RX1, TX1, %0000, 115_200)
hfcp.start

  t := cnt
  repeat
    c := serial.rxcheck
    if (c => 0)
      hfcp.enqueue(c)
    if (hfcp.has_msg)
      process_msg
    waitcnt(t += clkfreq >> 5)
```

In this example, we're starting the serial object and connecting it to the Propeller's programming port at 115.2K baud. The HFCP object has a `start` method that doesn't require any parameters; this method simply resets all its internal variables.

Receiving characters is handled in a synchronized loop that runs 32 times per second. This isn't particularly zippy, but it's much faster than you or I can type command strings into a terminal. Note that we are using the `rxcheck` method — this is very important. We may need to do other things in the loop and don't want to get hung up at this point. By using the `rxcheck` method, we'll get a return value of -1 if there is nothing in waiting in the serial RX buffer.

Let's assume that a character has arrived. It gets moved to the HFCP object using the `enqueue` method. Enqueue is fancy-schmancy for "put it into a buffer." The next step is to check for the presence of a complete message; when that is true, the program will handle it.

Obviously, there's more than meets the eye to the `enqueue` method, so let's have a look:

```spin
pub enqueue(c) | fc

if ((qidx == 0) and (c <> REQ_HDR))
  return
else
  if ((c <> EOM) and (c <> 0))
    queue[qidx++] := c
  else
    reset
    fc := field_count
    if (fc => 2)
      msgtype := parse_fields
    bytefill(@queue, 0, Q_LEN+1)
```

In the very beginning, I told you that a master-to-node message must start with the '>' character (this is defined in the `REQ_HDR` constant). If we are at the beginning of the message queue (`qidx` is zero), the program will wait for the proper header before moving on. Once things get going, we'll watch for the end-of-message character which I've defined as carriage return (13). As you can see, we simply add characters after getting the header and stop when we get an appropriate EOM.

You may be wondering why `qidx` is not defined locally. The reason is that we need to preserve this value between calls to the method; hence, it's in the object's global variables definitions.

When the end-of-message is detected, all internal message type and parameter variables are reset and the number of fields in the message is counted; if the count is two or more, the fields of the message are parsed/converted and made available. Counting fields is simply a matter of counting the number of separator characters (comma or EOM) in the message. When the EOM character is detected, the loop is aborted and the number of fields in the message is returned:

```spin
pub field_count | fc, idx

fc := idx := 0
repeat
  case queue[idx++]
    FLD_SEP:
    ""
```

### BILL OF MATERIALS

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Mouser Part #</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1-8</td>
<td>220 ohms</td>
<td>299-220-RC</td>
</tr>
<tr>
<td>C1</td>
<td>0.1 µF</td>
<td>594-K104K15X7RF5TH5</td>
</tr>
<tr>
<td>J1</td>
<td>Terminal</td>
<td>571-2828363</td>
</tr>
<tr>
<td>JP1a,JP2a</td>
<td>0.1 header</td>
<td>517-611TG</td>
</tr>
<tr>
<td>JP1b,JP2b</td>
<td>Shunt</td>
<td>517-950-00</td>
</tr>
<tr>
<td>R1,R2</td>
<td>10K</td>
<td>291-10K-RC</td>
</tr>
<tr>
<td>R3</td>
<td>3.3K</td>
<td>291-33K-RC</td>
</tr>
<tr>
<td>R4</td>
<td>120</td>
<td>293-120-RC</td>
</tr>
<tr>
<td>U1</td>
<td>ST485B</td>
<td>511-ST485BN</td>
</tr>
</tbody>
</table>
Parsing the fields is a little more involved; the first thing we need to know is where a particular field is located in the message string. For this, the `field_pos` method is used. This takes a zero-indexed field number and returns its position. If the field number passed is not valid for the message, then -1 is returned:

```c
pri field_pos(fn) | idx
    if (fn < 0)
        return -1
    idx := 0
    repeat while (fn > 0)
        case queue[idx++]
        FLD_SEP:
            --fn
        EOM, 0:
            idx := -1
            quit
    return idx
```

The position of the desired field is located by iterating through the characters in the message, decrementing the field number value when a separator is encountered. When the field number hits zero, we're sitting on the position of the desired field in the message and that is returned to the caller.

The final step is to extract and convert the various fields and make them available to the calling application; this is handled with the `parse_fields` method:

```c
pri parse_fields | pos, fc, idx
    reset
    taddr := strings.asc2val(queue[1])
    pos := field_pos(1)
    cmd := strings.asc2val(queue+pos)
    pos := field_pos(2)
    if (strings.instr(queue, string(STR_MSG)) => 0)
        pos := field_pos(3+idx)
        param[idx] := strings.asc2val(queue+pos)
    return MSG_NUM
```

At the top, we're going to reset all the internal variables; this sets all values returned to the user to zero since we don't know how many parameters are included in the next message. We use zero because the parser allows for positive and negative values (signed, 31 bits).

The first thing extracted is the target address for the message. This is removed from the message using the `asc2val` method from the strings object. Over the last year or so, I've been collecting and translating useful string functions and I finally stuck them all into a handy object. The target address is in field 0 of the message, so we don't have to use `field_pos` for this one. We do have to skip over the header, though, as the parser only allows for spaces, identifiers, and numeric characters.

The next field extracted is the command. We do need to use `field_pos` here because we don't know how long the target address field is. That's the nice thing about this simple little protocol: The fields can be variable width, so entering values in a terminal program is quite easy.

The final mandatory extraction is the parameters count. You'll remember that this field has a special case: When we transmit a string message, the parameters count is set to '@'. We can look for this using the `instr` method and if it's found, the string in field three is copied to an internal byte array called `tfield`. The `parse_fields` method exits with a return value of `MSG_STR` (2).

Simple command messages will have a parameter count of zero, so we can bypass the parameters extraction. When a message does have parameters, a loop finds and extracts them from the message. Numeric messages return a value of `MSG_NUM` (1) from `parse_fields`.

Okay, time to put this stuff to work. Here's an update to the demo loop I presented earlier. As you can see, I'm running the loop using a fixed timing value and using that to drive a simple timer/rtc that our project can use:

```c
pub main | t, c
    setup
        serial.tx(CLS)
        cnt := 0
        repeat
            c := serial.rxcheck
```
if (c => 0)
  serial.tx(c)
  hfcp.enqueue(c)
if (hfcp.has_msg)
  qualify

if (++timer[T_TIX] => (1_000 / LOOP_MS))
  timer[T_TIX] := 0
if (++timer[T_SECS] => 60)
  timer[T_SECS] := 0
if (++timer[T_MINS] => 60)
  timer[T_MINS] := 0
if (++timer[T_HRS] => 24)
  timer[T_HRS] := 0

waitcnt(t += constant(LOOP_MS * MS_001))

Messages are qualified and dispatched with the qualify method:

pub qualify

if (hfcp.target == MY_ADDR)
  case hfcp.has_msg
    hfcp#MSG_NUM:
      process_num
      hfcp#MSG_NUM:
      process_str
    hfcp.clear_msg

The important step here is to qualify the target address of the message; if it is not intended for this node, then there's nothing to do except clear the message flag. If the last message was intended for this node, then it is dispatched to the particular handlers based on the message type (numeric or string).

Most of the messages for this node are the numeric type; some (like read clock, #90) take no parameters, while setting the clock (#91) requires three. Here's the numeric message handler:

pub process_num | cmd, n

  cmd := hfcp.command
  n := hfcp.p_count

  case cmd
    0:
      if (n == 1)
        leds.set_all(hfcp.p_read(1))
    1..8:
      if (n == 1)
        leds.set(cmd-1, hfcp.p_read(1))
    9:
      if (n == 1)
        leds.dig_set(hfcp.p_read(1))
    11..18:
      if (n == 0)
        hfcp.write(1, leds.read(cmd-11))
        hfcp.build_msg(MY_ADDR, cmd, 1)
        serial.str(hfcp.response)
      if (n == 3)
        hfcp.write(3, @rtc[T_SECS])
        hfcp.build_msg(MY_ADDR, cmd, 3)
        serial.str(hfcp.response)
    90:
      if (n == 0)
        hfcp.write_block(3, @rtc[T_SECS])
        hfcp.build_msg(MY_ADDR, cmd, 3)
        serial.str(hfcp.response)
    91:
      if (n == 3)
        hfcp.write_block(3, @rtc[T_SECS])
        hfcp.build_msg(MY_ADDR, cmd, 3)
        serial.str(hfcp.response)

Let me jump up on a soap box for just a second. It's easy for all of us to ask, "Where's the object for that?" This behavior is not limited to Propeller programmers; it's human nature. A good friend in the special effects business says the running joke in that industry is new animators asking, "Where's the dinosaur plug-in?"

Remember that there was a time when we didn't have eight processors at our disposal and we did seem to get along, albeit with more work. Still, there are times when it is not necessary to launch another cog for a simple process.

The bare-bones timer/rtc here is a perfect example. It's just a few variables and a few lines of code. My favorite aspect of Spin is that we can run "foreground" loops with precise timing (remember the old days of padding loops to make the timing work out?) and with that, we have tremendous power. Okay. I think you get my point. There won't always be a ready-made object for everything, and sometimes a few lines of code is enough.

The top part of the loop checks for, displays, and enqueues characters. When a message is detected, the qualify method is called to deal with it. When that returns, the timer/rtc is updated. Easy, right? Absolutely!

Okay, then. Let's try some messages. Let me explain the goals here first. It's clear to you that follow my column I love lighting control, so I dropped the LED modulator from last time into the program and now I can control LEDs from a terminal. If I connect this to an RS-485 system, I can control any number of LEDs from long distance.

The demo commands are simple:

0     Set all channels to same level
1..8  Set specific channel to level
9     Set all channels using digital value
11..18 Return channel level
20    Display string message
90    Read timer/RTC
91    Set timer/RTC (for use as clock)
92    Clear timer/RTC (for use as timer)
we alert the compiler by using special symbols. What this means is that the command above can be changed to:

```
>1,9,1,%1000_0000
```

And, yes, the parser also allows the use of the underscore character to separate groups of digits in a number. As with the editors we use, $ is the designation for hexadecimal values.

Now have a look at command #90; this requires no parameters but does give something back (the current values in our timer/rtc). The write_block method is used to quickly transfer the working values of the timer/rtc to the parameters list in the HFCP object. Since that list uses longs, we use longs for the timer/rtc values to accommodate this transfer. The next step is to build the output message which — like the input — will be a string with comma-separated fields. The difference is that the response string will use ‘<‘ as its header. This allows us to monitor a system and distinguish between command and response messages.

The build_msg method is used and we will pass the response address, the command we're responding to, and the number of parameters to have in the response message. Now, you understand why we loaded the parameters first. Finally, we can send the message by using the str method of our serial object.

So, there you have it. An easy protocol handler that will simplify talking to your Propeller projects. Have a look at Figure 2 for an exchange with the demo program.

Of course, there are features we don't have space to cover, like the ability to set the numeric style (decimal, binary, hex) of parameters in the response message, etc. I know that those of you who have experience in data communications are asking, "Where's the checksum?" You're right. In this version, that doesn't exist. It's an easy add, though, and down the road I will take another page from the GPS strings and use the '*' symbol to indicate that there is a checksum appended to the message. This will be optional, of course, and will only apply when the messages are being generated by a computer — be it another Propeller or a PC on the network.

Okay, then. Load up the demo and start talking. This "talking" stuff is really a lot of fun and we will continue next time. If you missed Fred Eady's excellent article on CAN networking (November '11), you may want to go read it as that's where we're headed next. We'll build a CAN module for the Propeller Platform and I'll show you how to put it to use. It's really neat stuff.

Until next time, Happy New Year and keep spinning and winning with the Propeller! **NV**

---

**JON “JONNYMAC” WILLIAMS**

jwilliams@efx-tek.com

PARALLAX, INC.

Propeller chips and programming tools

www.parallax.com

---

**FIGURE 2.**

We start by retrieving the command and parameters count from the message, then drop into a case structure that handles the messages known to this node. The parameters count helps further qualify each message, especially in the case where one or more parameters are required.

As you can see, it's very straightforward stuff. Remember that we are calling this from the middle of our main loop, so we dare not dally. Let me point out a little 'Easter egg' that I neglected to mention earlier.

If you look at command #9, you'll see that it takes a single parameter and then calls the dig_set method (new to the LED modulator) to set the LEDs. This method treats the outputs like a standard parallel port; we turn on an LED with a "1" bit in the parameter and a "0" bit will extinguish the LED for that position. So, if we wanted to turn on the MSB LED and extinguish the others, we could do something like this:

```
>1,9,1,128
```

Most of us know that 128 is the MSB bit of a byte, but what happens when we want to set three or four random bits, and it's not an easy to remember value? No worries! The asc2val method in the strings library can deal with binary values — so long as they're indicated.

I took a page from the BASIC Stamp and Propeller editors: They allow us to use binary and hex values in our code;
12 KEY MEMBRANE KEYPAD
12 key, matrix-encoded membrane keypad. 2.74” x 3.0”.
Ultra thin with self adhesive backing. 3” 7-conductor ribbon leads with SIP socket connector, 0.1” centers.
CAT# PHS-20
$9.20 each

20W 8 OHM HORN SPEAKER
Universal# SP-20B.
Horn-type communication speaker. 20 Watt 8 Ohms. 5.25” dia. x 6” long. 3-hole adjustable mounting bracket. 12” leads.
CAT# PHS-20
$9.95 each

20W 8 OHM HORN SPEAKER
Universal# SP-20B.
Horn-type communication speaker. 20 Watt 8 Ohms. 5.25” dia. x 6” long. 3-hole adjustable mounting bracket. 12” leads.
CAT# PHS-20
$9.95 each

BATTERY HOLDER, 4 AA CELLS
Holder for 4 AA size batteries, in a 2x2 stacked format. 2.25” long x 1.20” x 1.10” high. Unbreakable black plastic. 6” wire leads.
CAT# BH-342
$11.00 each

DOOR LOCK ACTUATOR
12 Vdc motor-drive actuator for automotive door locks. Nylon plunger moves 0.75”. Push or pull, depending on polarity. Includes mounting hardware and a 10” metal extension rod which fits through the hole in the end of the plunger. Overall dimensions of actuator assembly (retracted): 8.42” x 2.35” x 1.15”. Rubber boot protects against moisture and dust. Pigtail leads.
CAT# DLA-1
$5.50 each

2.4 GHZ DESKTOP ANTENNA
2.4GHz desktop antenna for WLAN terminal applications. 4” high. 3” diameter weighted rubber base. SMA-R connector on 6’ cable.
CAT# ANT-37
$4.50 each

PC MOUNT BEEPER
TDK # SDC161425-01. 5 Vdc (operates on 4-8Vdc) electromagnetic beeper. 16mm diameter x 14mm high. 1900Hz. 85dB @13cm min. Average current, 30mA. PC pin terminals on 5mm centers.
CAT# SBZ-582
$1.45 each

8 CHARACTER X 2 LINE LCD
Crystralontz # CFAH0802A-NYG-JT. 8x2 character LCD. Yellow/Green LED Backlight. STN Positive, Transflective. 8-bit or 4-bit parallel interface. Industry-standard HD44780 compatible controller. -20°C - +70°C. Module size: 38.0 x 16.0mm. Character size: 2.96 x 5.56mm. RoHS Compliant.
CAT# LCD-812
$5.50 each

EMI SUPPRESSION BEAD
Fair Rite # 2643054201. 43 shield bead for 20-300MHz. 14.5mm long x 9.5mm o.d. x 6mm i.d. RoHs.
CAT# FB-94
2 for $1.00

21MM PIEZO ELEMENT
50mm wire leads.
CAT# PE-56
$1.00 each

12 VDC 5.83A SWITCHING POWER SUPPLY
Pihong Model PSC80U-120. Input: 100-240V 50-60Hz 2.0A. Output: 12.1V 5.83A. Table-top model. 5.93” x 2.78” x 1.54”. 46” output cord two ferrite bead EMI suppressors. 4-pin mini-DIN connector. Detachable grounded IEC power cord. cULus. CE, TUV.
CAT# PS-12583
10 for $17.00 each

RGB COLOR-CHANGING LED
5mm (T1-3/4) led cycles through, red - green - blue and combinations of those colors, yielding white, purple and other hues. Changes color every 3 or 4 seconds. Water-clear in off-state. Great for attention getting displays, costumes or jewelry.
CAT# LED-158
10 for 70¢ ea. • 100 for 50¢ ea.

18 GAUGE TEFLON TUBING, NATURAL
PTFE 18 TW NATURAL. 18 Gauge, thin-wall Teflon tubing. Natural (white). TFE-TW-300. Type VW-1. Rated 200°C Degrees, 300V rated. Nominal I.D. 0.042”. Wall Thickness: 0.012”. CSA. 5’ min.
CAT# PTFE-18
5 ft. for $1.50

MINI-PANCAKE VIBRATING MOTOR
LG Innotek Co. # TA09-T3001459AA / Spec MVMF-A346T3. 10mm dia. x 3mm thick. Operates on 1.5 - 4 Vdc. Two 12mm wire leads. Prepped with adhesive patch on one side.
CAT# DCM-707
50 for 80¢ each • 100 for 75¢ each

Shop ON-LINE www.allelectronics.com
ORDER TOLL FREE 1-800-826-5432

MAIL ORDERS TO: ALL ELECTRONICS CORP.
14928 OXNARD ST., VAN NUYS, CA 91411-2610

FAX (818) 781-2653 • INFO (818) 904-0524
E-MAIL allcorp@allcorp.com

NO MINIMUM ORDER • All Orders Can Be Charged to Visa, Mastercard, American Express or Discover • Checks and Money Orders Accepted by Mail • Orders Delivered in the State of California must include California State Sales Tax • NO C.O.D. • Shipping and Handling $7.00 for the 49 Continental United States - ALL OTHERS including Alaska, Hawaii, P.R. and Canada Must Pay Full Shipping • Quantities Limited • Prices Subject to change without notice.

MANUFACTURERS - We Purchase EXCESS INVENTORIES... Call, Write, E-MAIL or Fax YOUR LIST.

QUALITY Parts
FAST Shipping
DISCOUNT Pricing
CALL, WRITE, FAX or E-MAIL
for a FREE 96 page catalog.
Outside the U.S.A. send $3.00 postage.

12/6/2011 12:28 PM Page 63
You get started right away. Arduino ADK programming tutorial helps you learn how to use them for calculations. Chuck Hellebuyck introduces the software side of Arduino and explains how to program microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software interface, and explains the ADK and USB Open Application Interface, and explains the digital gadgetry to cherished analog products of yesteryear. About the Author: Michael Jay Geier began operating a neighborhood electronics repair service at age eight that was profiled in The Miami News.

$24.95

How to Diagnose and Fix Everything Electronic

Master the Art of Electronics Repair!

In this hands-on guide, a lifelong electronics repair guru shares his tested techniques and invaluable insights. How to Diagnose and Fix Everything Electronic shows you how to repair and extend the life of all kinds of solid-state devices, from modern digital gadgetry to cherished analog products of yesteryear. About the Author: Michael Jay Geier began operating a neighborhood electronics repair service at age eight that was profiled in The Miami News.

$24.95

Programming Arduino
Getting Started with Sketches

Program Arduino with ease! Using clear, easy-to-follow examples, Programming Arduino: Getting Started with Sketches reveals the software side of Arduino and explains how to write well-crafted sketches using the modified C language of Arduino. No prior programming experience is required! The downloadable sample programs featured in the book can be used as-is or modified to suit your purposes.

$14.95

Electronics Explained
The New Systems Approach to Learning Electronics

Don’t spend time reading about theory, components, and old ham radios — that’s history! Industry veteran, Louis Frenzel, gives you the real scoop on electronic product fundamentals as they are today. Rather than tearing electronics apart and looking at every little piece, the author takes a systems-level view. For example, you will not learn how to make a circuit but how a signal flows from one integrated circuit (IC) to the next, and so on to the ultimate goal.

$29.95

Arduino + Android Projects for the Evil Genius

By Simon Monk

TEAM ARDUINO UP WITH ANDROID FOR SOME MISCHIEVOUS FUN! This wickedly inventive guide covers the Android Open Application Development Kit (ADK) and USB interface, and explains how to use them with the basic Arduino platform. Methods of communication between Android and Arduino that don’t require the ADK — including sound, Bluetooth, and WiFi/Ethernet — are also discussed. An Arduino ADK, programming tutorial helps you get started right away.

$24.95

Electronics An Introduction

by Jim Stewart

This book is designed as an in-depth introduction to important concepts in electronics. While electronics can be highly mathematical, the text is not about calculations. It is about how electronic equipment is able to extract, process, and present information held in electrical signals. If you are in — or studying to be in — a profession that requires the use of electronic equipment, then this book will provide the insight necessary to use such equipment effectively.

$29.95

Build Your Own Electronics Workshop

by Thomas Petruzzellis

BUILD YOUR OWN DREAM ELECTRONICS LAB! This value-packed resource provides everything needed to put together a fully functioning home electronics workshop! From finding space to stocking it with components to putting the shop into action — building, testing, and troubleshooting systems. This great book has it all! And the best part is, it shows you how to build many pieces of equipment yourself and save money, big time!

Reg Price $29.95
Sale Price $22.95

HTML: A Beginner’s Guide

by Wendy Willard

Create highly functional, impressive websites in no time. Fully updated and revised, HTML: A Beginner’s Guide, Fourth Edition explains how to structure a page, place images, format text, create links, add color, work with multimedia, and use forms. You’ll also go beyond the basics and learn how to save your own web graphics, use Cascading Style Sheets (CSS), create dynamic web content with basic JavaScript, and upload your site to the web. By the end of the book, you’ll be able to build custom websites using the latest HTML techniques.

$29.95

Programming PICs in Basic

by Chuck Hellebuyck

If you wanted to learn how to program microcontrollers, then you’ve found the right book! Microchip PIC microcontrollers are being designed into electronics throughout the world and none is more popular than the eight-pin version. Now the home hobbyist can create projects with these little microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software language. Chuck Hellebuyck introduces how to use this development setup to build useful projects with an eight-pin PIC12F683 microcontroller.

$14.95

PICAXE Microcontroller Projects for the Evil Genius

by Ron Hackett

This wickedly inventive guide shows you how to program, build, and debug a variety of PICAXE microcontroller projects. PICAXE Microcontroller Projects for the Evil Genius gets you started with programming and I/O interfacing right away, and then shows you how to develop a master processor circuit.

$24.95
There are 1,000 projects on this CD, not to mention the additional technical info and bonus features. It doesn’t matter if you’re a complete novice looking to do their first science fair project or a super tech-head gadget freak; there are enough projects on the single CD-ROM to keep you and 50 of your friends busy for a lifetime!
During the 2010 MASTERS Conference, I saw it on the “X”... MPLAB X, that is. MPLAB X has come a long way since its inception. In addition to the Microchip line of C compilers, MPLAB X now supports the CCS C compiler and microEngineering Labs new PICBASIC PRO compiler 3.0. If you want to use your MAC to create PIC firmware with MPLAB X, now you can. MPLAB X is also Linux friendly.

MPLAB X VERSUS MPLAB IDE

The MPLAB 8 IDE is a workspace-oriented development tool. That is, the project’s source code, hardware tool (programmer/debugger) information, and target microcontroller (device) information are all contained within the project workspace. Conversely, the MPLAB X development tool is project oriented. Everything associated with the project (hardware tool, compiler, device, etc.) is contained within the scope of the project. Since each MPLAB X project is a self-contained entity, multiple projects using differing programmers/debugging devices, PIC devices, and compilers can be debugged simultaneously within the MPLAB X environment.

The root of any MPLAB X project is the source code. Multiple configurations can be built around a central piece of source code that is common to the project. This MPLAB X design point allows you to run differing hardware tools and PIC devices against the project’s common source code root within the same project.

Device memory is presented in precast memory windows within the MPLAB 8 IDE. The latest version of MPLAB X also allows the use of precast memory views. However, MPLAB X takes it one step further. Each memory view can be changed to an alternate view on the fly. The format selection of the memory views in the MPLAB X

SCREENSHOT 1. This is a composite capture of the steps taken to create a new MPLAB X project. The PIC18F465J13 does not have an associated hardware header, so the Select Header step is ignored.
environment has also been simplified. Instead of accessing a drop-down menu to select a data format view—which is the way of MPLAB 8 IDE—the format of an MPLAB X memory window’s data view can be changed within the memory window.

While the MPLAB X differences are fresh in our minds, let’s create an MPLAB X project called NUTSANDVOLTS and examine the areas of MPLAB X we just discussed. Take a look at Screenshot 1. Step 1 in creating a new MPLAB X project requires that we select a project type. In that we want to spawn a totally new project, a Microchip Embedded Standalone Project will do nicely.

Moving to Step 2, we are prompted to select a target microcontroller (device). Today, I happen to be intrigued by the PIC18F46J13 and I happen to have that PIC mounted on an EDTP general-purpose development board.

It is possible that the selected device has an optional hardware header. There is no hardware header for the PIC18F46J13. So, Step 3 is skipped.

The tool is selected in Step 4. Note that the PICkit2 does not support the PIC18F46J13. However, the PICkit3 is highlighted by a couple of green lights. So, the PICkit3 is our tool in this configuration.

The compiler of choice is the CCS C compiler. Every compiler loaded on my laptop is listed as available in Step 5. Yes. I have them all.

Screenshot 2 is the final step in the MPLAB X new project creation process. As you can see, our new project NUTSANDVOLTS resides in the NUTSANDVOLTS.X folder, which lies inside of the nv-mplabx folder. We have no source code, so Screenshot 3 is a skeleton of our new project. In that there’s no gas in the tank, the Dashboard view is a null presentation. Note that the PICkit3 is dormant. That’s because MPLAB X does not use the PICkit3 when not debugging or programming.

This seeming snub has its advantages. Since we know that we can run multiple debugging sessions, it makes sense that the tool should not be hogged by a single project. If we’re forced to share a single tool among multiple MPLAB X projects, only the project that is running a debug or programming sessionseizes control of the PICkit3 tool. When that project finishes with the tool, it releases it for use by other projects.

Looking at 0xFF and 0x00 characters in memory views doesn’t do a thing for me. So, let’s crank up the CCS C compiler and generate some header code for the...
PIC18F46J13. Once that’s done, we’ll put a bit of CCS C source to use it.

**ADDING FILES TO OUR MPLAB X PROJECT**

Yes. We could create all of the necessary

 PIC18F46J13 fuse information manually. However, the CCS compiler creates fuse, clocking, and RS-232 configuration values without breaking a sweat. So, I ran its PIC Wizard to create the nutsandvolts.h file contents. I could just copy the CCS C-generated file to my MPLAB X project file, but that’s just dull. Let’s exploit yet another cool feature of MPLAB X. We’ll create the MPLAB X-approved nutsandvolts.h file using the new file feature that is part of MPLAB X. In Step 1 of Screenshot 4, I chose to create a CCS compiler header file. The only typing I had to do was to give the new header file a name. Everything else was filled in by MPLAB X. All that’s left to do is cut and paste the CCS C-generated file filler into our new nutsandvolts.h file.

For those of you that are not familiar with the configuration header file created by the CCS PIC Wizard, here is the final nutsandvolts.h file contents we will use with our C source code:

```c
/*
* File:   nutsandvolts.h
* Author: Fred
*
* Created on November 1, 2011, 2:53 PM
*/
#include <18F46J13.h>
#define ICD=TRUE
#define NOWDT //No Watch Dog Timer
#define WDT128 //Watch Dog Timer uses 1:128 Postscale
#define PLL1 //No PLL PreScaler
#define NOPLLEN //4X HW PLL disabled, 4X PLL enabled in software
#define STVREN //Stack full/underflow will cause reset
#define NOXINST //Extended set extension
//and Indexed Addressing
//mode disabled (Legacy
//mode)
#define DEBUG //Debug mode for use with ICD
#define NOPROTECT //Code not protected from reading
#define HS //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
#define SOSC_HIGH //High-power SOSC circuit is selected
```

SCREENSHOT 4. Here again, the creation of a file can be performed manually. Why do it by hand when you can call upon MPLAB X to do the job and format the file for the CCS C compiler automagically? As you can see, I’ve asked MPLAB X to create a properly formatted CCS C header file called nutsandvolts.h.

**SOURCES**

Custom Computer Services
CCS C Compiler
[www.ccsinfo.com](http://www.ccsinfo.com)

Microchip
MPLAB X
PIC18F46J13
[www.microchip.com](http://www.microchip.com)

microEngineering Labs
PICBASIC PRO Compiler 3.0
[www.pb3.com](http://www.pb3.com)
```
#include <nutsandvolts.h>
void main(void) {
    while (TRUE);
}
```

Now, I’ve got two problems. MPLAB X can’t find `#include <18F46J13.h>` and it can’t find the `nutsandvolts.h` file that contains `#include <18F46J13.h>`. Okay, Now, I’m a bit excited. Hopefully, registering the `nutsandvolts.h` and `nutsandvolts.c` files with MPLAB X will solve the problems.

The `nutsandvolts.h` and `nutsandvolts.c` files we created reside in the project directory (`NUTSANDVOLTS.X`). So, I right-clicked on Source Files in the Projects window and chose “Add Existing Item” which resulted in Screenshot 6. Since the project is king, MPLAB X automatically looked into the project directory for the existing files. It is also possible to search other folders for files you need to include in the project. Once `nutsandvolts.c` was officially a member of the project, all of the “can’t find” errors

At this point, MPLAB X is not happy. MPLAB X is telling me that it can’t find `#include <18F46J13.h>` since things aren’t quite “done” yet, I’m not getting too excited about MPLAB X’s lack of vision. Let’s see what happens if we create a CCS C compiler `.c` main file.

**Screenshot 5** is a `.c` version of **Screenshot 4**. This time around, I chose to go with the MPLAB X’s `.main.c` template to create the file `nutsandvolts.c`. Here’s what MPLAB X gave me:

```
/*
 * File: nutsandvolts.c
 * Author: Fred
 * Created on November 1, 2011, 3:09 PM
 */
```
disappeared. At this point, I felt that it was okay to flesh out the `nutsandvolts.c` file with the C code that the CCS compiler generated, mixed with my own special sauce:

```c
void init(void) {
    setup_psp(PMP_DISABLED);
    setup_spi(SPI_MASTER | SPI_SCK_IDLE_LOW | SPI_CLK_DIV_64);
    setup_spi2(SPI_SS_DISABLED);
    setup_wdt(WDT_OFF);
    setup_timer_0(RTCC_OFF);
    setup_timer_1(T1_ENABLE_SOSC);
    setup_timer_2(T2_DISABLED, 0, 1);
    setup_timer_4(T4_DISABLED, 0, 1);
    setup_comparator(NC_NC_NC_NC);
    setup rtc(RTC_ENABLE);
    setup_low_volt_detect(LVD_TRIGGER_BELOW | LVD_30); // 3.0 volts
    set_tris_a(0b11111111);
    set_tris_b(0b11101111);
    output_b(0b11011111);
    set_tris_c(0b10010011);
    output_c(0b11111111);
    set_tris_d(0b10011100);
    output_d(0b10011101);
    set_tris_e(0b00000000);
    output_e(0b00001111);
    nutsandvolts_time.tm_year = 11;
    nutsandvolts_time.tm_mon = 9;
    nutsandvolts_time.tm_mday = 29;
    nutsandvolts_time.tm_wday = 6;
    nutsandvolts_time.tm_hour = 14;
    nutsandvolts_time.tm_min = 34;
    nutsandvolts_time.tm_sec = 0;
    rtc_write(&nutsandvolts_time);
    lastsec = nutsandvolts_time.tm_sec;
    enable interrupts(INT_HLVD); // LOW VOLTAGE INTERRUPT
    enable interrupts(GLOBAL);
}
```

I’ve prepared the PIC18F46J13 for interrupt on low voltage detect and activation of the PIC’s RTCC. The RTCC routines are part of the CCS C PIC18F46J13 device driver. The plan is to have the PIC keep time and rely on battery backup to keep the time when main power drops below 3.0 volts. Let’s see if this puppy will compile.

## COMPILING AND RUNNING OUR MPLAB X PROJECT

MPLAB X allows the human programmer to build the project; build the project with a clean; build the project into debug; and make and program the project. I kinda like the automated make and program option.

If things go as planned, the time beginning at 14:34 will be sent every second via the PIC18F46J13’s serial port to an AccessPort terminal emulator session. What do you think about **Screenshot 7**?

## BACK TO BASICS

Let’s use what we’ve learned about creating a CCS C project using MPLAB X and apply it to creating an MPLAB X project for the PICBASIC PRO compiler. The project creation steps are identical, except we must replace any references to the CCS compiler with PICBASIC PRO compiler directives.

We begin by creating an MPLAB X embedded standalone project. We’ll call our PICBASIC PRO project `PBP3_MPLABX`. Naturally, as you can see in **Screenshot 8**, we’ll select the PICBASIC PRO compiler as the project compiler. When the project creation steps are complete, we will have created a project folder called `PBP3_MPLABX`.

`PBP3_MPLABX` is lacking a PICBASIC PRO source code file at this point. So, we’ll right-click on Source
Files and add a new file formatted for the PICBASIC PRO compiler. Note that a PICBASIC file selection is offered in Screenshot 9. Our new PICBASIC PRO file is called PBP3CODE.pbp. Here’s what the MPLAB X PICBASIC file template placed in the newly created file:

```
****************************************
*  Name    : PBP3CODE
*  Author  : Fred
*  Notice  : Copyright (c) 2010
*          : All Rights Reserved
*  Date    : Nov 3, 2011
*  Version : 1.0
*  Notes   :
*          :
****************************************
```

The first order of business is to specify the PIC18F46J13's configuration fuse settings. To make the job easy, reference the PIC18F46J13.PBPINC file which is a part of the PICBASIC PRO compiler install package. All of the valid configuration fuse settings are contained within the PIC18F46J13.PBPINC file. Here’s how we should configure the PIC18F46J13:

```
#CONFIG
CONFIG WDTEN = OFF ;Disabled - Controlled by SWDTEN bit
CONFIG PLLDIV = 1 ;No prescale (4 MHz oscillator input drives PLL directly)
CONFIG CFGPLLLEN = OFF ;PLL Disabled
CONFIG STVREN = ON ;Enabled
CONFIG XINST = OFF ;Disabled
CONFIG CP0 = OFF ;Program memory is not code-protected
CONFIG OSC = HS ;HS
CONFIG OSCCSSEL = HIGH ;High Power TiOSC/8OSC circuit selected
CONFIG CLKOE = OFF ;CLKO output disabled on the RA6 pin
CONFIG FCMEN = OFF ;Disabled
CONFIG IESO = OFF ;Disabled
CONFIG RTCCSC = T1OSCRE ;RTCC uses TiOSC/T1CKI
CONFIG DBOREN = OFF ;Disabled
CONFIG DSWDTEN = OFF ;Disabled
CONFIG IOL1WAY = ON ;The IOLOCK bit ;(PPSCKCON<0>) can be set once
CONFIG ADCSEL = BIT10 ;10 - Bit ADC ;Enabled
CONFIG MSSP7B_EN = MSK7 ;7 Bit address ;masking mode ;Configuration Words ;page not erase/write-protected
CONFIG WPDIS = OFF ;WPFP(6:0), WPEND, ;and WPCFG bits ignored
#ENDCONFIG
```

Once the configuration fuses are specified, we need to tell the PIC how to spit out the time values to a serial port. Basically, we inform the PICBASIC PRO compiler as to how fast the PIC is being clocked and enable the PIC’s hardware UART:

```
define OSC 12
define HSER_RCSTA 90h
define HSER_TXSTA 20h
define HSER_BAUD 9600
```

Once the compiler knows that we’re running our PIC18F46J13 at 12 MHz, it “automagically” calculates the SPBRG values for our specified baud rate of 9600 bps.

## A PICBASIC PRO CODE

### MPLAB X EXAMPLE

What follows in the PICBASIC PRO code is a driver for the PIC’s internal RTCC. The driver consists of subroutines that set and read the RTCC clock registers. You can use the RTCC routines to perform events under the control of the time generated by the RTCC. However, that wouldn’t do here because you couldn’t see the results of the RTCC’s work. So, I funneled the time out of the RTCC into a terminal emulator. Here’s the code I’ve written to set the RTCC registers:

```
SET_TOD:
  EECR0 = $55 ;unlock sequence
  EECR2 = $AA
  RTCCFG = $5
  ;write enable RTCC register
```
RTCCFG = %00100011
RTCVAL = currentyear
RTCCFG = %00100010
RTCVAL = curreniday
RTCVALH = currentmonth
RTCVAL = currenthour
RTCVALH = currentweekday
RTCVAL = currentsec
RTCVALH = currentmin
RTCCFG.7 = 1 ; start the clock
EECON2 = $55 ; unlock sequence
RTCCFG.5 = 0 ; write disable RTCC register
RETURN

Note the unlock sequences. They are required to allow and prevent access to the write enable and clock enable bits in the RTCCFG register. Once the RTCC has been loaded with the current time and date values, the clock is started. To be able to use the RTCC, we've got to be able to read its registers on the fly. Yep, I wrote some code to do that:

SCREENSHOT 9. Once MPLAB X makes the new PICBASIC PRO file available to us, we'll flesh it out with some PICBASIC PRO source that will drive the PIC18F46J13's RTCC.

---

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

Multi-Seat Licensing for Educational Institutions
Upgrade from PICBASIC™ Compiler (PBC)

Download a FREE trial version now.

www.PBP3.com

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

PICBASIC and PICBASIC PRO are trademarks of Microchip Technology Inc. in the USA and other countries. PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.
SCRENSHOT 10. I once had a very competent programmer tell me “It doesn’t matter what programming language you use, as long as you get the desired results.”

GET_TOD:
    RTCCFG = %10000011
    currentyear = RTCVALL
    RTCCFG = %10000010
    currentday = RTCVALL
    currentmonth = RTCVALH
    currenthour = RTCVALL
    currentweekday = RTCVALH
    currentsec = RTCVALL
    currentmin = RTCVALH
RETURN

As you can see, reading is just a reversal of roles between the time/date variables and the RTCC registers.

As you would imagine, the main example algorithm simply reads the clock continuously and checks for a rollover of the seconds register. Every time the seconds register is incremented, the current time is transmitted to the terminal emulator. The results can be seen in Screenshot 10.

THE X FACTOR

MPLAB X contains mountains of features we didn’t touch on in this discussion. The more you use MPLAB X, the more you will learn about it. MPLAB X allowed me to use a PICkit3 for programming and debugging both the C and BASIC projects we hammered out. As you uncover MPLAB X features, you’ll ask yourself why you haven’t already added this incredible IDE to your design cycle. NV
SURPLUS

SURPLUS ELECTRONIC PARTS & ACCESSORIES

Over 16,000 items in stock.

Required |

Connectors | Displays | Fan | Hardware | LEDs | Motors | Potentiometers | Speakers | Switches | Test Equipment | Tools |

No Minimum Order.

Credit Cards and PAYPAL Accepted. Flat $4.95 per order USA Shipping.

Surplus Material Components

SMC ELECTRONICS

www.smcelectronics.com

COMPONENTS

RF PARTS” CO.

Tubes, Transistors, Power Components

Email: rfpar@rfparts.com • Web: www.rfparts.com

800-737-2787 Fax 888-744-1943

New

uM-FPU64

64-bit Floating Point Coprocessor

64-bit and 32-bit IEEE 754 compatible

SPI or I2C interface, 3.3V Supply, 5V Tolerant I/O

GPS input, local device support, Flash functions, IDE

Robotics Navigation Sensor Modules Additional products...

DIP-28, SOIC-28, TQFP-44

www.micromegacorp.com

KITS/PLANS

QKITS.COM

50 AMP PWM

1-888-GO 4 KITS

High Quality Electronic Kits & Modules

$59.95

UK1133

50 AMP DIGITAL PWM (OPTIONAL DISPLAY)

FM Transmitters Timers Audio Amplifiers Motor Controllers PIC Programmers / Arduino

www.canakit.com

www.nutsvolts.com

CLASSIFIEDS

ROBOTICS

MaxSonar

Ultrasonic Ranging is EZ

LV-MaxSonar-EZ

• Power-up calibration

• Smallest MaxSonar

• Low power, 2.5V-5.5V

• $29.95 qty. 1

MaxSonar-WR (IP67)

• Industrial packaging

• Standard ¾” fitting

• $99.95 qty. 1

XL-MaxSonar-EZ

• Beam pattern choice

• Real-time calibration

• High acoustic power

• $99.95 qty. 1

MaxSonar-WRC (IP67)

• Compact packaging

• Weather resistant

• Quality narrow beam

• $99.95 qty. 1

www.maxbotix.com

HARDWARE

WANTED

DEC EQUIPMENT WANTED!!!

Digital Equipment Corp. and compatibles.

Buy - Sell - Trade

CALL KEYWAYS 937-847-2300

or email buyer@keyways.com

www.keyways.com

DEVICES & COMPONENTS

Digital Equipment Corp.

and compatibles.

Buy - Sell - Trade

CALL KEYWAYS 937-847-2300

or email buyer@keyways.com

www.keyways.com

DESIGN/ENG SERVICES

The Smartest Indicator

You Will Ever Buy

The LS1-55

Load Status Indicator

www.brandino.com

ENERGY SOLUTIONS

MaxSonar-EZ

Ultrasonic Ranging is EZ

MaxSonar-WR (IP67)

• Industrial packaging

• Standard ¾” fitting

• $99.95 qty. 1

XL-MaxSonar-EZ

• Beam pattern choice

• Real-time calibration

• High acoustic power

• $99.95 qty. 1

MaxSonar-WRC (IP67)

• Compact packaging

• Weather resistant

• Quality narrow beam

• $99.95 qty. 1

www.maxbotix.com

INTERNET

www.brandino.com

www.earthled.com

LIGHTING

www.SunnySolutions-shops.com

800-315-0309

www.sunnySolutions-shops.com

www.earthled.com

Earn Money On Your Electric Bill

www.brandino.com

www.earthled.com

www.sunnySolutions-shops.com

800-315-0309

www.earthled.com

www.earthled.com
Often copied - never equaled:

Online PCB price calculation
- ✔
- ✔
- ✔
- ✔
- ✔

FREE STENCIL
- ✔
- ✔
- ✔
- ✔
- ✔

Manual DRC included
- ✔
- ✔
- ✔
- ✔
- ✔

Collision detection
- ✔
- ✔
- ✔
- ✔
- ✔

Accepted file formats
- ✔
- ✔
- ✔
- ✔
- ✔

Face to Face Order Tracking
- ✔
- ✔
- ✔
- ✔
- ✔

Discount for feedback
- ✔
- ✔
- ✔
- ✔
- ✔

Etest included for Multilayer
- ✔
- ✔
- ✔
- ✔
- ✔

Direct order from EAGLE
- ✔
- ✔
- ✔
- ✔
- ✔

Watch"ur"PCB
- ✔
- ✔
- ✔
- ✔
- ✔

On Time or Free
- ✔
- ✔
- ✔
- ✔
- ✔

Data restoration service
- ✔
- ✔
- ✔
- ✔
- ✔

Sunstone
- ✔
- ✔
- ✔
- ✔
- ✔

Advanced Circuits
- ✔
- ✔
- ✔
- ✔
- ✔

Batch PCB
- ✔
- ✔
- ✔
- ✔
- ✔

Express PCB
- ✔
- ✔
- ✔
- ✔
- ✔

16

6

2

2

1

The Original since 1994!

www.pcb-pool.com

Call Tyler: 707 447 7744
765 Eubanks Drive
Vacaville CA 95688

Toll FREE USA: 1 888 977 7443
Email: sales@pcb-pool.us
## Reader-to-Reader Tech Forum

### Questions

#### Autoranging DVM Using 7101
I want a small circuit using a microcontroller to convert an analog DVM to an autoranging DVM using a 7107 IC. Or, just explain how to convert the input of a 7107 DVM to an autoranging input.

---

#### Power Supply For Fluke 8050A and 8600A
I have a couple of older Fluke 8050A and 8600A multimeters that were outfitted with battery options. The batteries are long gone and I never really used it anyway. I would like to use them as bench only meters and reconfigure them to a non-battery type supply. Does anyone have an internal power supply plan that I could use in these meters without drilling holes in the cases and cutting up traces on the circuit boards? Something using capacitors, diodes, etc., that would fit into the battery area and plug directly into the battery connections.

---

#### Multi Output Power Supply
I want to know the simplest solution for this power supply need. My input is 36-75 DC.

- I need these outputs:
  - +65V @ 70 ma
  - +5V @ 200 ma
  - +15V @ 150 ma

---

#### Self-Powered Current Switch
I would like to know how to build a self-powered current switch to monitor a lamp pole light. I need it to output an open or closed contact, so I can sense if there is current or not.

- The light pole voltage is 120V at a range of 1A-10A. Anything above 200-

---

### Answers

#### Yagi Antenna Question
I live where cellular reception is poor, and I depend on a cellular modem, amplifier, and Yagi antenna (Wilson Electronics 301111) mounted on our chimney for an Internet connection. In the summer it works pretty well, but in the rainy winter, the signal deteriorates significantly and our connection speed drops. If we’re using the fireplace, and the weather dries out for a few days, the signal improves. However, if it’s rainy but we’re not using our fireplace for a few days, the signal also improves.

My best guess as to the reason is that the combination of smoke from our fireplace and moisture from rain or dew are coating the antenna with a film that is conductive enough to create a partial short across elements of the antenna, causing some attenuation of the signal. The antenna is mounted about six or seven feet above the chimney cap, and it’s not really practical to mount it higher without going to a much more complicated mast setup with guy wires.

I’d like to know if I can spray some insulating coating on the Yagi antenna that will prevent the film contacting the conductive parts of the antenna, but that also will not in itself attenuate the signal. Something like WD-40 might work for a while — causing water to just run off — but I want something more permanent, like maybe liquid tape that hardens into a rubbery coating. I just don’t know if that type of coating might in itself attenuate the signal.

Does anybody have any advice on how I could seal my antenna without attenuating the signal by an appreciable amount? I’m also open to other ideas, except to go to a different connection technology (DSL isn’t available, T1 is too expensive, satellite is worse than what we’ve got now).

---

#### Vintage Transformers
I have some industrial equipment inherited from a deceased customer. Two transformers, vintage WWII, 10 and 5 KVA and one transformer probably, 15 KVA with SCRs but no control board. This equipment was originally intended for wire heat treat service, 28 VAC secondaries. I have had them 20 years or more. I’ve been trying to cook up a stable circuit for firing SCRs for all the transformers. I have worked with such circuits all my professional life but [now 76 yrs] stability is eluding me. Industrial electrical service is always noisy and welding especially so. I know I need zero crossing sync and I need 0.008 sec to zero sec adjustability for the triggers. It must also be jitter free. This is just a hobby project for me. Any help out there?

---

#### Troubleshooting
How does one replace a spider IC?

---

#### Vintage Transformers
I have some industrial equipment inherited from a deceased customer. Two transformers, vintage WWII, 10 and 5 KVA and one transformer probably, 15 KVA with SCRs but no control board. This equipment was originally intended for wire heat treat service, 28 VAC secondaries. I have had them 20 years or more. I’ve been trying to cook up a stable circuit for firing SCRs for all the transformers. I have worked with such circuits all my professional life but [now 76 yrs] stability is eluding me. Industrial electrical service is always noisy and welding especially so. I know I need zero crossing sync and I need 0.008 sec to zero sec adjustability for the triggers. It must also be jitter free. This is just a hobby project for me. Any help out there?

---

All questions AND answers are submitted by Nut & Volts readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. Questions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted by readers and NO GUARANTEES WHATSOEVER are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals.

Always use common sense and good judgment!
>>> YOUR ELECTRONICS QUESTIONS ANSWERED HERE BY N&V READERS

Send all questions and answers by email to forum@nutsvolts.com.
Check at www.nutsvolts.com for tips and info on submitting to the forum.

because nobody buys antique transformers except for scrap, which might be my best business decision considering the times. I do have the transformers and the SCRs. My heirs can scrap the units out after my reward. I have lots of scrapables to play with and nothing else but time.

#1127 Robert Gibson
Aurora, IL

LCD Display Help
In the recent article on the web-based thermostat (Sep-Oct’11) an EA DOG-M LCD display was used. I have two of these displays and haven’t been able to get them working. I use PIC BASIC PRO and it seems there are a lot of people that are trying to use this LCD with some trouble. There isn’t much in the way of information on the forums. Could someone please show how to initialize the display with PIC BASIC PRO in both SPI and four-bit mode?

#1128 Craig
Bettendorf, IA

Extra Bright Picture/Exit Bulb
I’m looking for help locating or building a super bright exit/picture/display style bulb with an E26 base for my mom. My 95 year old mother has macular degeneration in both eyes. She still enjoys reading but finds it harder by the day. Her main complaint is that she can’t get enough light on the printed page.

I am asking for help to locate and/or build a bulb with what I know as ultra-bright LEDs. It will throw the equivalent of 200 [fairly cold] incandescent watts of light down onto her page. I’m no engineer, but am handy with a soldering iron and multimeter. Can anyone help me?

#1129 Rod Reynolds
Bowie, MD

Half Cycle Magnetizer
I’m looking for a half cycle magnetizer schematic for 60 cycle, 120 volts. I used one at work before I retired, but it was built for a higher industrial voltage. I would like to build one from scratch using the standard 120 volt line. I know that it must be contained in a protective box, takes many turns of transformer windings, and uses a high speed thyristor.

#1120 Bob Macias
Fernandina Beach, FL

>>> ANSWERS

[#11115 - November 2011] Power Conversion
I would like to make a special power supply/converter with a difference. It can be supplied from:
- A nominal 12 VDC (nine to 15 actual) from a vehicle supply.
- Or by a nominal 24/28 VDC supply (20 to 32) from a vehicle or aircraft supply.
- Or, by international mains (90 to 250 VAC).
- All three inputs need to be isolated and capable of being accidentally or deliberately connected at the same time.
- Have all three inputs tolerant to transients, e.g., mains derived from generator.

The unit needs to have four outputs, all of which are DC and each of which is configurable internally by a trim pot to deliver four output voltages between 10 and 24 VDC (e.g., 12V, 15V, 18V, and 22V).

- Output to be unaffected by change of input source.
- Once set, each output voltage needs to maintain a tolerance of ±0.5 VDC, regardless of input changes.
- Each output to be capable of delivering 120 watts.
- Robust, portable unit.

I understand that these parameters are often mutually exclusive, but the following considerations are also desirable:
- Small size.
- Low heat dissipation.
- Low noise.
- Low interference.
- High quality.

My goodness! Four 120-watt outputs? I think you are pretty well stuck with switching power supplies here! I’m not a switching power supply guru — to put it mildly! — so I won’t try to design that, but I will suggest an ‘architecture.’

Clearly, you need four switching power supplies for the four different output voltages (or at least, that seems the cleanest to me).

I would suggest having three input switching power supplies to convert the three differing input voltages to some handy intermediate voltage that the four output switchers take as their input. Set the output of each switcher such that the mains gives a few volts more than the other two, and then pick which one of the other’s you want to supply the power if both are hooked up. Isolate these three input supplies from each other with diodes. That is, each of these three has a diode between it and the intermediate supply rail. With the mains switcher having the higher output voltage, it will take (most of) the load even if the other two are active. The diodes will keep the three input switchers from feeding each other.

Rusty Carruth
Tempe, AZ

[#11119 - November 2011] Loudspeaker/CB Combination
I was wondering if it would be possible to cheaply build a system in which I speak into a CB type radio and...

Phil Karras, KE3FL

have the signal sent to a loudspeaker about 1,000 feet away or slightly more. The current system I have is a bullhorn, and I have to stand a good distance back from the crowd to accomplish my task. It kills my voice. I was thinking if I could build a loudspeaker (what wattage?) and a set of CBs that I should be able to modify the other CB to feed into the amplified loudspeaker so I can simply talk into it.

While this is definitely possible, CB wasn't really made for this; the quality may not be good enough but if you don't want to buy an FM mic/receiver or build an FM mic kit to FM radio receiver, then this will work. It will also all work on 12V which may be needed.

First, at your location test out the CB with a good mobile whip antenna and see what it receives. Remember, for receive only it really doesn't need to be properly balanced. The SWR doesn't really matter and, in fact, an antenna with worse reception due to poor SWR may end up being better for this application. DO NOT transmit from the "receiving" radio into a poor antenna. One reason a poor receiving antenna may be better right now is that skip is starting to come in as I write this in November 2011, so you may have to contend with competition. Remember there is NO stronger signal capture for CB AM signals as there is for FM receivers. Also, if you're close to a major highway — unless you can find a clear channel — this may not work at all. Keep in mind that there may be traffic on whatever channel you pick. The receiving radio should have the squelch set to its highest setting to keep out as much other traffic and noise as possible but NOT during your tests. The test should be run with either squelch open or at the lowest setting possible for white noise only block-out.

You might need to build a poor receiving antenna to restrict what the receiver picks up. If you need suggestions on that, check out http://cs.yrex.com/ke3fl (my website). I wrote an article for building CB HT "rubber duck" type antennas. The article was published in Popular Communications a few years ago. You will probably want to limit your output power to as low as needed for the distance you're covering, which I'm assuming is line of sight. A 100 mW transmitter may be enough for 1000'. Recommendation: Either find or modify the handle-talkie. You want one that runs on AA cells — not AAA or 9V batteries since the AAs are far more cost-effective and have a much better energy density. They will last longer — MUCH longer — than a single 9V battery and more than 3X longer than AAA batteries. The higher the voltage the better, so an HT using 12V is better than one using 9V, which is better than one using 6V as far as current draw for the same power output. This means the 12V will last longer than the 9V (which will last longer than the 6V assuming they all use the same kind of batteries).

If the CB receiver you are using doesn't have enough audio power to drive the external speaker you want, then you'll need to put together — pick up at a yard sale or hamfest or buy and build — a small audio amplifier to take the output from the SPK (external speaker) jack of the CB radio to the input of the amplifier that then drives the speaker.

I have seen kits from RAMSEY Electronics, Inc. (www.ramseykits.com) for audio amplifiers (you only need a mono-amp, not stereo), as well as small FM broadcast transmitters. I'm sure there are others as well (Jaycar, www.jaycar.com and Cana Kit Corp. www.canakit.com — both of which may have audio amps and FM transmitters). Unless you have a very well stocked junk drawer, I'd buy a kit. At Ramsey, there is a 10W audio amp, (#CK003, doesn't say stereo) for ~$13. If you're using a CB receiver, you do NOT need a mic pre-amp. Cana Kits has the same audio amp for the same price as Ramsey, but they also have some stereo amps at 25W and 35W per channel, ~$44 and $70, respectively. I was unable to check jaycar.com.

The transceiver’s microphone, control, and speaker jcks are connected to corresponding jcks in the co-located SMR which translates the radio’s conductors to those of a VGA cable. At the remote location, an identical SMR maps the VGA conductors back to the transceiver’s connections. The microphone, control, and speaker plug into the SMR’s jcks as if they were connected directly to the radio. By changing the SMR’s jumpers, different radios can be used in the same car, or one transceiver can be used in different cars by running VGA cables in both of them. Replacement radios are installed simply by changing SMR jumper settings.

The patented SMRs are sold in pairs for $79.95, preset and with radio interface cables for any RJ-compatible radio. High quality five meter VGA cables are also available. All prices include free shipping in the continental US and a moneyback guarantee.

Pololu announces the release of the dual VNH5019 motor driver shield for Arduino which provides an easy way to control up to two high power DC motors with an Arduino

continued from page 22
NEW PICO LINEAR SERVO

Now available from Solarbotics is the VS-19 pico linear servo which is a tiny and affordable linear actuator with 2 cm of travel. A linear actuator is an actuator that creates linear motion, as opposed to rotary motion, i.e., the spinning of a motor. Many linear actuators (such as this one), are actually driven by a non-linear motion. The VS-19 uses a geared pager motor to spin a worm gear section with a floating nub.

The VS-19 can be used to turn light switches on and off, raise and lower legs, trigger a latch, and/or bring up a periscope. Because of its size, it’s perfect for stealthy remote control applications. This actuator accepts a servo pulse from 800-2,200 microseconds (µs) with a neutral position of 1,500 µs.

The traveling nub takes about a second to move from one side of the channel to the other (2 cm) at 3.7V. This linear servo can run at voltages as low as 2.5V, but at the expense of speed. Price is $9.85.

For more information, contact: Solarbotics, Ltd.
Web: www.solarbotics.com

ADCvertexer INDEX

Find your favorite advertisers here!
Programmable DC Electronic Load

- 0~80V / 0~40A

The 3721A Programmable DC Electronic Load provides excellent performance with sophisticated features found on much more expensive units. This 400 watt, 40 Amp, 0~40 volt Programmable DC Electric Load can be used to test all sorts of DC power sources including power supplies and is especially helpful to battery manufacturing processes. This DC load features constant voltage, constant resistance, constant current and constant power settings. The end user can design programs that control precisely all of the load values and time durations for each step of a test sequence. Up to nine 10 step programs can be internally stored in the 3721A Programmable DC Load.

- 4 basic functions: CC, CV, CR & CP
- 8 basic test modes: CCL, CCH, CV, CRL, CRM, CRH, CPV & CPC
- Minimum operating voltage is less than 0.6v at the load's full rated current.
- High-speed sequence, high-speed transient, short circuit, Programable current slew rate.
- Multiple groups of parameters and lists can be saved & recalled.
- Supports SCPI and LabView with included software.
- Current Rating: 0~40A
- Voltage Rating: 0~80V
- Power Rating: 400W at 40°C

<table>
<thead>
<tr>
<th>Item#</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS1721A</td>
<td>Programmable DC Electronic Load</td>
<td>$720.00</td>
</tr>
</tbody>
</table>

60 Watt Digital Soldering Station

- For use with traditional or Lead Free Soldering

This is a terrific soldering station featuring a microprocessor controlled temperature set up system with great versatility. The unit includes 3 preset temperature settings that are user definable so you can turn on the system & push one button to go to the temperature range you desire. Specific system temps can also be set with an easy to use push button up/down button AND when you turn off this station, the unit keeps the last used temperature in memory & automatically returns to that setting the next time you turn on the system. The temperature display can be set to display in Celsius or Fahrenheit scale.

The CSI-Station-3DLF is a powerful 60 watt soldering system. The fast heat recovery provided by a 60 watt system like this allows the user to solder both traditional solder and lead free solder. This system features a grounded tip to protect delicate circuits from static charge. Also included is a separate iron holder. Circuit Specialists stocks a large supply of tips for this station.

<table>
<thead>
<tr>
<th>Item#</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS1711A</td>
<td>60 Watt Digital Soldering Station</td>
<td>$499.00</td>
</tr>
</tbody>
</table>

Circuit Specialists carries a wide selection of Traditional & Lead-free soldering equipment, including soldering irons, soldering stations, hot-air rework stations, focused infrared systems, reflow ovens, pre-heating systems, regular and lead-free solder, solder flux, and soldering aids for both thru-hole and surface-mount soldering.

See our complete line of soldering equipment at: www.CircuitSpecialists.com/soldering-equipment

Circuit Specialists offers a wide range of nozzle designs and many have built-in holders for efficient placement of heat/airflow over the desired component. Our many rework solutions offer a wide range of nozzle designs and many have built-in holders for efficient placement of heat/airflow over the desired component.
Best Value, Low Cost Station

CSI-Station1A

Includes
1 Year USA Warranty

You get both a 200 MHz Oscilloscope and a multi function digital multimeter, all in one convenient lightweight rechargeable battery powered package. This power packed package comes complete with a convenient carry case, test leads, two scope probes, charger, PC software, USB cable and a convenient nylon carrying case.

Features:
- 200MHz Handheld Digital Scopemeter with Integrated Digital Multimeter Support
- 200MHz Bandwidth with 2 Channels
- 50MHz’s Max. Sampling Rate
- 50Ω Equivalent Time Sampling Rate
- 6,000-Count DMM resolution with AC/DC
- Large 5.7 inch TFT Color LCD Display
- USB Host/Device 2.0 full-speed interface connectivity
- Multi Language Support
- Battery Power Operation (Installed)

Item #

CSI-Station1A

www.circuitspecialists.com/csi-station1a

$29.95

Hantek 5000B Series Digital Storage Oscilloscopes

Introducing the Hantek 5000B Series Digital Storage Oscilloscopes. Available in 60MHz, 100MHz and 200MHz bandwidths. Each one provides real-time sample rates up to 1GSa/s and equivalent sample rates up to 25GSa/s. In addition, they have a maximum 1M memory depth for better observation of waveform details. The 7 inch color TFT LCD Display with Windows-style interface and menus provide easy operation.

Abundant menu information and easy-to-use buttons give you plenty of measurement information.

The multifunctional knobs and the powerful shortcut keys help save time during operation.

The Autoset function lets you detect sine and square waves automatically.

The Probe Check Wizard guides you to adjust the probe compensation and set the Probe attenuation factor.

Three help methods (context-sensitive, hyperlinks, and an index).

You can quickly master all features to greatly improve your efficiency in production and development.

Features:
- 60, 100 or 200 MHz bandwidth
- 1GSa/s Real-time sample rate
- 25GSa/s Equivalent sample rate
- Large (7.0-inch) color display
- 60MHz Hand Held Scopemeter, test leeds, two
- You get both a 200 MHz Scopemeter, test leeds, two
- 60MHz Hand Held Scopemeter
- 25GSa/s Equivalent sample rate
- Large (7.0-inch) color display
- USB host and device connectivity, standard
- Multi Language Support
- Battery Power Operation (Installed)

Item #

Hantek 5000B Series Digital Storage Oscilloscopes

www.circuitspecialists.com/hantek-5000b-series-digital-storage-oscilloscopes

$385.00

Programmable DC Power Supplies

www.circuitspecialists.com/programmable-dc-power-supplies

$379.00

www.circuitspecialists.com/programmable-dc-power-supplies

$379.00

www.circuitspecialists.com/programmable-dc-power-supplies

$379.00
Parallax sensors, microcontrollers, robots, kits, and accessories

...AT A STORE NEAR YOU!

Most Parallax sensors and accessories are compatible with a wide variety of 3.3 V and 5 V microcontrollers. Now they're available in electronics hobby stores near you! Get started on your latest invention today. Look for Parallax wherever fine microcontrollers are sold. Prices may vary.

Look for Parallax product at select Radio Shack, Fry's Electronics, Micro Center, All Electronics, and Tanner Electronics stores.

See the full retail line at www.parallax.com/rt
Order online or call us toll-free at 888-512-1024 (M-F, 8 AM-5PM, PT).

Friendly microcontrollers, legendary resources.”

Prices subject to change without notice. Parallax and the Parallax logo are trademarks of Parallax inc.