

PROJECTS ■ THEORY ■ APPLICATIONS ■ CIRCUITS ■ TECHNOLOGY

NUTS AND VOLTS

NUTS AND VOLTS

www.nutsvolts.com
April 2016

EVERYTHING FOR ELECTRONICS

Build A

MIDI CONTROLLED ANALOG SYNTHESIZER *On A Shoestring*



Wi-Fi better faster stronger!
STANDARDIZING BALLOONSATS
for school STEM projects
DIGITAL ELECTRONICS a beginner's guide
BUILD THE BLACKOUT PROOF LED light system



888-7-SAELIG

www.saelig.com

info@saelig.com



Check out our remarkable, unique products

Test Equipment

Automotive

Cables

Components

Converters

Data Loggers

Displays

Embedded

Industrial PCs

Wireless

cambriox



EtherSync

*Intelligent multiple USB-port
charging/syncing with remote access.*

EMCYD EMC TEST DESIGN, LLC



Smart Fieldmeter® Kit

*RF safety & monitoring,
EMC pre-compliance isotropic
RF field meters and antennas.*



TEKBOX
DIGITAL SOLUTIONS



TBPS01-TBWA2

*High-quality near-field EMC probes
and amplifiers for pre-compliance
testing, LISNs, and TEM cells.*

Triarchy

TELEMAKUS LLC



TSAxGI



TEA4000-7

*Economical USB-powered and
controlled RF building blocks,
spectrum analyzers and signal generators.*



MixNV

*Low cost state-of-the-art RF modules
with better performance than
products costing much more.*

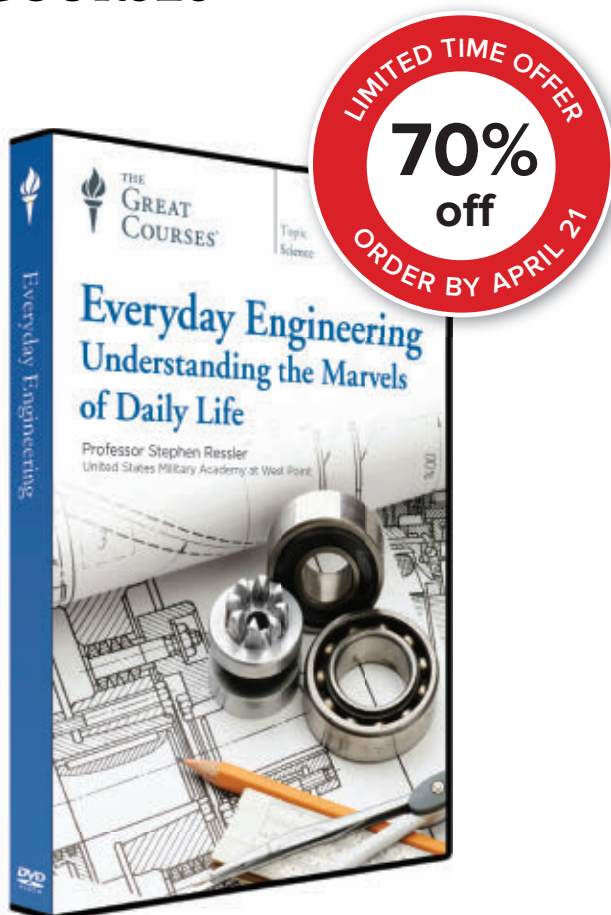


PIE 850

*USA-made NIST-traceable process
calibrators: accurate, reliable, rugged, and
ideal replacements for other leading brands.*

Free Shipping
Online Orders Over \$99
US Domestic Ground Shipping Only
(HI and AK Excluded)

We also provide the widest selection of PC and RF spectrum analyzers, USB and logic analyzers, waveform generators, digital pattern generators, pure RF sources, DMMs, wireless and wired data-loggers, test equipment, SPI/I2C controllers, touch-sense, Ethernet and USB ICs and modules, CANbus and motor control boards, high-reliability industrial and touch-panel PCs, wireless and I/O control, Ethernet-serial/USB-serial/RS232-485, software, EMI tents and enclosures.



Understand the Genius of Everyday Technologies

Great masterpieces of engineering are everywhere. The achievements of everyday engineering are worth admiring for their own sake. They represent some of the most inspired thinking of our civilization, and by looking under the hood to see how these technologies work, you learn about basic scientific principles that have remarkable applications.

Everyday Engineering: Understanding the Marvels of Daily Life is an indispensable guide to the way things work in the world. Conducting this eye-opening tour is Professor Stephen Ressler, an award-winning civil engineer and a nationally honored leader in engineering education. No background in science or engineering is needed to follow this riveting presentation, which gives you deep insight into the complex systems that enhance your life—providing a broad understanding of everyday engineering that you can apply to any technological issue that crosses your path.

Offer expires 04/21/16

THEGREATCOURSES.COM/4NV
1-800-832-2412

Everyday Engineering: Understanding the Marvels of Daily Life

Taught by Professor Stephen Ressler
UNITED STATES MILITARY ACADEMY AT WEST POINT

LECTURE TITLES

1. Engineering and Technology in Your World
2. Your House as an Engineered System
3. Three Structural Systems for Load Bearing
4. Platform-Framed Housing Construction
5. The Building Envelope
6. Site Design and Storm Runoff
7. Dam, Reservoir, and Aqueduct Design
8. Water Treatment and Distribution
9. Wastewater Disposal and Treatment
10. Fossil Fuels: Coal, Oil, and Natural Gas
11. Power Generation from Coal
12. Oil, Gas, and Nuclear Power
13. Renewable Sources of Electricity
14. Electrical Power Transmission: The Grid
15. Electrical Power Distribution
16. Everyday Thermodynamics: Refrigeration
17. Heating, Ventilating, and Air-Conditioning
18. Home Energy Efficiency
19. Passive Solar and Net-Zero-Energy Homes
20. The Plain Old Telephone Service
21. The Global Telecommunications Network
22. Cellular Phone Technology
23. Satellites and Satellite Communications
24. Simple Machines around the House
25. User-Centered Design
26. The Internal Combustion Engine
27. Torque, Power, and Transmission
28. The Drivetrain
29. Suspension, Steering, and Braking
30. Highway Engineering
31. Traffic Engineering
32. Everyday Bridges
33. Tunnel Engineering
34. The Railroad
35. Solid Waste Disposal and Recycling
36. The Future: Engineering for Sustainability

Everyday Engineering:
Understanding the Marvels of Daily Life
Course no. 1116 | 36 lectures (30 minutes/lecture)

SAVE \$270

DVD ~~\$384.95~~ NOW \$114.95
+\$15 Shipping, Processing, and Lifetime Satisfaction Guarantee
Priority Code: 126615

For over 25 years, The Great Courses has brought the world's foremost educators to millions who want to go deeper into the subjects that matter most. No exams. No homework. Just a world of knowledge available anytime, anywhere. Download or stream to your laptop or PC, or use our free mobile apps for iPad, iPhone, or Android. Over 550 courses available at www.TheGreatCourses.com.

April 2016

26 Battery Backed Lighting System

Don't be caught in the dark! Whether you're trying to find your way into the kitchen for a midnight snack or you lose power because of an unexpected outage, you'll be prepared with this automatic on/off and battery backup for an under-the-counter lighting system.

■ By Jim Lacenski

06 PICAXE Primer Sharpening Your Tools of Creativity

Controlling an Electric BBQ Smoker

We will finish up our project with an electric BBQ smoker, and implement a PICAXE 08M2 based smoker control that will enable us to select any target temperature we want (between 100° and 250° Fahrenheit), and effectively maintain the temperature of the smoker within a couple of degrees on either side of the target temperature.

14 Near Space Approaching the Final Frontier

A New Vision of BalloonSats

I've built a lot of BalloonSats in my day; every one of them different and not all of them a work of art. After studying the standardization found in CubeSats, I began wondering if BalloonSats might benefit from some standardization themselves. So, here's my first (and very humble) attempt.

18 Q&A Reader Questions Answered Here

Hand Tools for Servicing Electronic Devices

Thinking about a career (as one reader is) that involves working on electronic equipment and/or are just getting started in electronics and need to know what tools to get? Here's a list for you, plus a few tricks, tips, and safety measures.

50 The Design Cycle Advanced Techniques for Design Engineers

Basic Training for All Programming Languages

When programming, it doesn't matter what language you prefer. What's important is a solid knowledge of the fundamentals of programming that will pull your applications together. So, in this installment, we will test drive the new M.E. Labs Trainer for programming. You won't need to purchase an expensive compiler or programmer/debugger. Everything you need to enhance your coding skills is included with this package. All you need is some free time and a willingness to learn.

32 A MIDI Controlled Analog Synthesizer on a Shoestring

If you've always longed to make electronic music yourself, but have fretted about your pocketbook or the complexity, here's a perfect solution. Learn how to put together a very usable MIDI controlled analog synthesizer for under \$40.

■ By Thomas Henry

38 The Beginner's Guide to Digital Electronics

This article was written specifically for the newcomer to the field of digital electronics. If you want to know how the digital world works, then discover how the numbers "1" and "0" changed the universe of all electronics — digital and analog.

■ By Roger Secura

56 Open Communication The Latest in Networking and Wireless Technologies

So, You Think You Know Wi-Fi

Wi-Fi is just about the most widely used wireless technology in the world. There are access points throughout business and industry, plus, don't forget all those public hot spots in places like convention centers, hotels, and airports. Wi-Fi is convenient as virtually all of our PCs, laptops, tablets, and smartphones have it that we use for email, texting, and Internet access. However, there is a lot more you may not know about Wi-Fi. Here is a primer to bring you up to date.

59 Practical 3D Printing Real World Uses for Electronics Experimenters

MakerBot Customizer for 3D Print Creations

There are a lot of design ideas available on Thingiverse.com. I've often downloaded one and then modified it in Tinkercad. Some of the designs on Thingiverse offer a unique option that lets the Customizer software by MakerBot do the modifications for you. It makes customizing a 3D design so much easier when it's available. Not all designs offer it, and it takes a lot of work by the creator. When done right, however, it can be an incredibly useful tool.

Departments

05	DEVELOPING PERSPECTIVES	25	SHOWCASE
	<i>Indestructibles:</i>	37	ELECTRO-NET
	<i>The Lure of Tube</i>	48	NV WEBSTORE
	<i>Audio Equipment</i>	62	TECH FORUM
24	NEW PRODUCTS	64	CLASSIFIEDS
		64	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; cpcreturns@nutsvolts.com.

Published Monthly By
T & L Publications, Inc.

430 Princeland Ct.
Corona, CA 92879-1300

(951) 371-8497

FAX **(951) 371-3052**

Webstore orders only **1-800-783-4624**

www.nutsvolts.com

Subscription Orders

Toll Free **1-877-525-2539**

Outside US **1-818-487-4545**

P.O. Box 15277

North Hollywood, CA 91615

FOUNDER

Jack Lemieux

PUBLISHER

Larry Lemieux

publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ ADVERTISING SALES

Robin Lemieux

robin@nutsvolts.com

EDITOR

Bryan Bergeron

techedit-nutsvolts@yahoo.com

VP OF OPERATIONS

Vern Graner

vern@nutsvolts.com

CONTRIBUTING EDITORS

Fred Eady

Tim Brown

Paul Verhage

Chuck Hellebuyck

Ron Hackett

Lou Frenzel

Thomas Henry

Jim Lacenski

Roger Secura

CIRCULATION DEPARTMENT

subscribe@nutsvolts.com

SHOW COORDINATOR

Audrey Lemieux

WEBSTORE MARKETING COVER GRAPHICS

Brian Kirkpatrick

sales@nutsvolts.com

WEBSTORE MANAGER

Sean Lemieux

sean@nutsvolts.com

ADMINISTRATIVE STAFF

Re Gandara

Copyright © 2016 by T & L Publications, Inc.

All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: 430 Princeland Court, Corona, CA 92879.

Subscription Information

Nuts & Volts — PO Box 15277, North Hollywood, CA 91615-9218

Call 877-525-2539 or go to **www.nutsvolts.com**

Subscribe • Gift • Renewal • Change of Info

DEVELOPING PERSPECTIVES

by
**Bryan
Bergeron,**
Editor

Indestructibles: The Lure of Tube Audio Equipment

My Sony integrated amp with copper chassis and huge toroidal transformers was a tour de force in my audio setup before the power mains took an indirect lightning hit. Because the microcontroller was fried, I couldn't even get the unit to power up.

Without access to spare parts — including a new microcontroller assembly — I was at the mercy of factory certified technicians. And — because the unit was just out of warranty — I was going to be out \$100 plus shipping in order to get an estimate on the repair.

Well, that amp is still sitting behind my workbench. Someday, I'll find an identical amp on eBay, buy it for parts, and cobble together

something that works. In the meantime, I decided to rebuild a McIntosh 240: a hot, bulky, but virtually indestructible tube amplifier. I spent a weekend replacing the electrolytic capacitors and swapping out the dozen vacuum tubes, one at a time.

The McIntosh 240 — like many other amps from the '60s and '70s — is unimpressive on paper. A mere 40 watts per channel, total harmonic distortion a whopping 0.5%, and a stripped down weight of 56 lbs. Plus, no remote.

Output is via massive potted output transformers through old-fashioned terminal strips. For less than the price of the vacuum tubes, I could have bought an NAD 3020D

Continued on page 47



The advertisement features a Probotix CNC Router in the center, which is a precision machine with a green motor and a silver frame, mounted on a wooden base. Above the machine, the text 'probotix CNC Routers & Control Systems' is displayed in a stylized font. To the right of the machine is a QR code. Below the machine, the text 'When You Are Serious About CNC' is written in a bold, italicized font, followed by the website 'www.probotix.com'.

Controlling an Electric BBQ Smoker

In this month's Primer, we're going to continue working with an electric BBQ smoker, and implement an 08M2 based smoker control project that will enable us to select any "target" temperature we want (between 100° and 250° Fahrenheit) and effectively maintain the temperature of the smoker within a couple of degrees on either side of the target temperature. (Note: For the remainder of this article, you can assume that all temperature values are in degrees Fahrenheit.)

Before we get into the details of our project, I want to briefly discuss the dominant software approach to the digital control of temperature (and other variables). The software that's included in virtually all commercially available digital controllers is based on what's known as a proportional-integral-derivative (PID) algorithm. As its tripartite name suggests, the PID algorithm is based on three different factors, the latter two of which obviously involve calculus, which is what makes a PID algorithm so complex and so precise. For example, I use a commercial sous vide PID controller that maintains the temperature of the sous vide water bath within $\pm 0.2^\circ$ of the target temperature that I set.

Developing a PID algorithm can be a fairly daunting task. In fact, the complexity of the PID algorithm has been a major factor in my reluctance to begin working on a grill/smoker temperature control project. Even if we were to tackle a PID project (which we're definitely not going to do!), the PICAXE limitation of "integer only" mathematics would be a major impediment to the accuracy of any program we might develop.

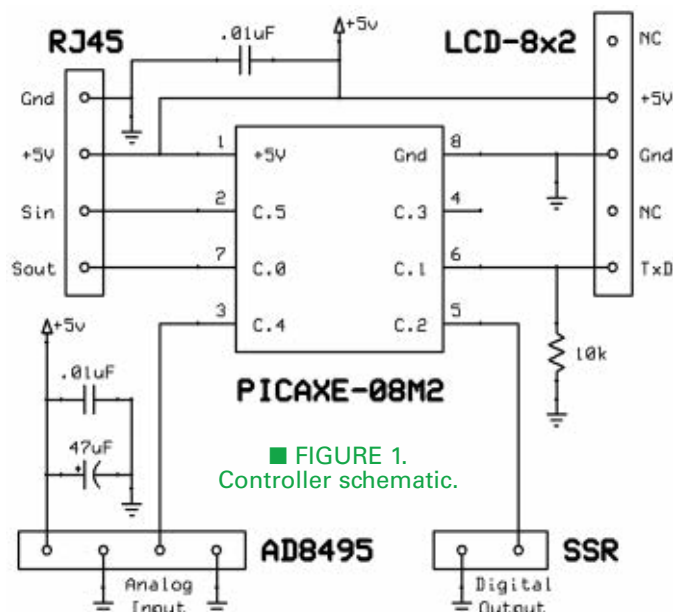
However, I recently read an online article that mentioned that a typical home oven is considered to be "well regulated" if its temperature remains within $\pm 20^\circ$ of its target setting, and that inaccuracy can easily double if the oven's temperature regulation circuitry has become un-calibrated. (Since then, I've read several other online

articles that support this statement.) This discovery made me realize that the precision provided by a PID algorithm may actually be "over-kill" when applied to a simple electric smoker.

Fortunately for us, there's no calculus involved in the "proportional" term of a PID algorithm. Stated in English, it simply means that the amount of heat that's applied to the smoker should be proportional to the difference between the target temperature and the current real time temperature. That approach didn't seem nearly as daunting as a full PID algorithm, and my accuracy requirement for this project wasn't near as demanding as that of my commercial sous vide controller. I decided, if I could develop a purely proportional approach to controlling the smoker temperature and was able to maintain the real time temperature within $\pm 5^\circ$ of the target temperature, that would be more than adequate for my purposes. Happily, that turned out to be surprisingly easy to do.

Hardware

The schematic for the 08M2 circuit that I used for this project is presented in **Figure 1**. As you can see, it's very similar to the circuit we used to monitor the smoker temperature in the previous installment of the Primer. In fact, there are only two additions to our monitoring circuit:



■ **FIGURE 1.**
Controller schematic.

Pin C.2 and ground are brought out to a two-pin header that connects to the SSR (solid-state relay) digital input; and pin C.1, +5V, and ground are brought out to a second header that interfaces with a serialized LCD that displays real time temperature data next to the smoker. I used the two-line eight-character LCD we constructed way back in the June 2009 column, but any two-line (or larger) serial display should work.

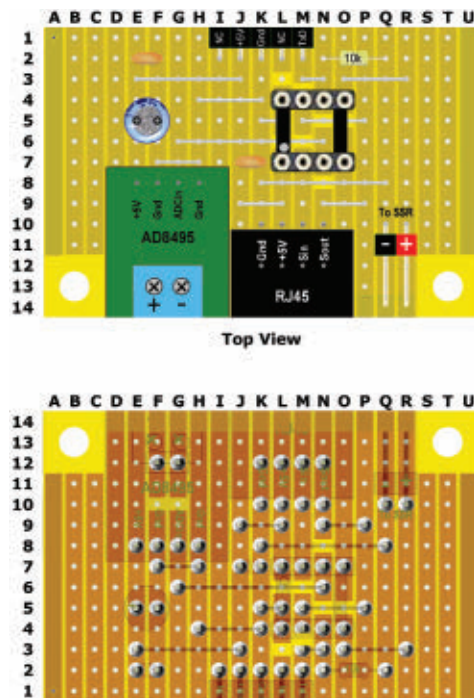
An LCD isn't really necessary because the data will also be sent to a PC for display. If you just want to experiment a bit, you can eliminate the LCD entirely. You don't even need to modify the program we're going to use because it doesn't harm the 08M2 to

send data out on pin C.1 without a receiving device on the other end of the connection. On the other hand, if you happen to be a bit of a BBQ "fanatic" (as I am), you may want to construct a stripboard version of the smoker control circuit.

The stripboard layout that I used is presented in **Figure 2**, and the complete **Parts List** is shown in **Figure 3**. (A full page pdf version of the stripboard layout is included in this month's downloads.) I attached the completed stripboard (along with the 8x2 LCD) to a small piece of plywood and mounted the finished assembly next to my smoker in the furnace room (see **Figure 4**).

The construction of the stripboard is very straightforward, so I'm not including step-by-step instructions. However, there are four points that I do want to mention:

1) The five-pin header at the center of the top edge of the stripboard is a right-angle male header. The layout doesn't show it, but the male header pins extend above the top edge of the board. This header is used for the LCD connection, and its pinout matches that of the 8x2 LCD I used. (If you use a different serial LCD, you may need to modify the stripboard circuit or construct a custom cable to connect to your LCD.) In the December 2011 Primer, we modified the original LCD



■ **FIGURE 2.** Controller stripboard layout.

Smoker Controller Parts List

AD8495 Amplifier Board ([Adafruit.com](http://adafruit.com))
 Capacitor, .01 μ F (two pieces)
 Capacitor, 47 μ F, Electrolytic
 Connector, RJ45 (modified as per text)
 Header, Male, Right Angle, Five Pins
 Header, Male, Right Angle, Two Pins
 Header, Male, Straight, Two Pins
 Resistor, 10K, 1/4 or 1/6 watt
 Socket, IC DIP, eight Pins
 Stripboard, 19 Traces w/14 Holes Each
 (Both Solid Edge; Traces Included)

■ **FIGURE 3.** Controller Parts List.



■ **FIGURE 4.** Controller in action.

driver software to take advantage of the increased speed of the 14M2 processor, and I recently further modified that driver to include a degree sign as one of the custom display characters specifically for our smoker control project. The most recent LCD driver update is included in this month's downloads at the article link.

2) It can't be seen in the top view of the stripboard layout in **Figure 2**, but there's a two-pin straight male header soldered in place underneath the Adafruit AD8495 thermocouple amplifier board. (The two solder points are visible in the bottom view, at points F12 and G12.) This header has no electrical function. The short ends of both pins are snipped off just above the plastic base and filed smooth so the header can provide level support for the AD8495 breakout board. (I stuck a small piece of electrical tape on the bottom of the AD8495 to make sure the breakout board is fully insulated from the stripboard.)

3) I modified the RJ45 connector in the layout (in the

manner that we discussed in the previous installment of the Primer) so that it can be directly soldered to the stripboard. If you prefer to use a standard eight-pin RJ45 breakout board, you will need to either redesign the stripboard layout or use a four-pin cable to make the necessary connections from the stripboard to the RJ45 board.

4) When I originally designed the stripboard circuit, I planned to use a

pwmout signal to control the SSR, which is the reason that I chose pin C.2 for the connection. However, I wasn't able to produce a stable temperature using this approach, so I did some online research to see if there was something I was missing. I discovered that erratic SSR behavior could result when the PWM frequency isn't low enough. I tried using a *pwmout* frequency of 61 Hz (which is the slowest frequency that can be used with an 08M2 running at 4 MHz), but I was still unable to produce a stable temperature in the smoker.

Finally, I decided to use an entirely different approach to control the smoker. Fortunately, the new approach completely solved the problem, so let's move on to discussing the details of it.

Software

In an interesting way, the software we're about to discuss differs significantly from every other program we've ever used in the Primer. In all our previous experiments and projects, it would have been possible for a reader to implement the necessary hardware, download and run the software, and end up with a successfully functioning project. In other words, if the experiment or project worked correctly for me, it would work correctly for everyone. Our smoker control project is definitely an exception to this rule.

If you use the software we're about to discuss, you may get results that are somewhat similar to mine, or your results may differ wildly from mine. If the program we are going to use (*SmokerControl.bas*) works exactly the same for anyone at all, I would be amazed. The reason is that no two smokers are exactly the same. For example, the smoker I used for this experiment is a double-walled unit with about 0.5" of insulation between those walls. Your smoker may have more or less insulation, or no insulation at all. Even if your smoker happens to be the same make and model as mine, I have made a few modifications to my smoker over the years, so your results will still differ somewhat from mine.

As a result, what's most important in this project is not the *SmokerControl.bas* program that you download. The process that I used to develop that program is what really matters because everyone who implements this project will need to go through a similar process in order to end up with a program that functions correctly for his/her specific smoker (or heater, or sous vide controller, etc.). With that in mind, let's take a look at that process.

When I first began thinking about this project, I had two basic assumptions in mind: more frequent temperature measurements and adjustments will result in a greater degree of temperature control; and higher target temperatures will require greater amounts of heat to maintain the temperature at a stable level, because more heat will "escape" from the smoker to its environment.

The first assumption is the reason that I chose to monitor and adjust the level of heat in one second intervals, and the second assumption led me to begin my experiments with an initial goal of determining the precise amount of heat that's needed to maintain several different target temperatures. Because I assumed that the necessary amount of heat varies directly with the target temperature, I needed a variable for this purpose, which I declared as *minHeat*. However, I quickly realized that I couldn't determine accurate values for the *minHeat* variable until I had a working proportional control algorithm, so that's where I began my experiments.

I've been grilling and smoking with charcoal for many years now, so I know that it's best to allow a fire to get established before beginning to control it. For example, if I want to cook at 250°, I keep the grill's air vents fully open until the temperature reaches about 175°, and then begin to close down the vents until I reach a setting that stabilizes the final temperature at about 250°. If the temperature starts to rise significantly above 250°, I close the vents entirely, and re-open them a bit when the temperature starts to drop. That may be a simple procedure, but it certainly can be called an algorithm. So, I decided to translate that algorithm into a series of instructions for the 08M2 to execute.

In order to do that, I knew that I would need a variable to store the target temperature chosen by the user, and another variable for the current real time temperature. In addition — because I was working on a proportional control algorithm — I would also need a variable to keep track of the difference between the target temperature and the current temperature. I declared those three variables as *targetF*, *tempF*, and *error*, respectively.

Finally, because all temperature values are stored in tenths of a degree to maximize the accuracy of the control algorithm, the three variables needed to be defined as word variables. For example, if the target temperature = 250° and the current temperature = 200°, *targetF* would be 2500, *tempF* would be 2000, and *error* would be 2500 – 2000, or 500, so byte variables definitely would not work. As you know, the PICAXE compiler can't handle negative arithmetic, so I made sure that the *error* variable would never be negative by including the following *if/then* statement in the program's main *do/loop*:

```
if targetF > tempF then
    error = targetF - tempF
else
    error = 0
endif
```

Of course, the above code snippet defines *error* to be 0 when it actually is (i.e., whenever *targetF* = *tempF*), but it also defines *error* to be 0 when it isn't (i.e., whenever *targetF* < *tempF*). However, that's not a problem because we want to turn off the heat in both those cases, so we can simply do that whenever *error* = 0.

Implementing the Proportional Algorithm

The charcoal grilling “algorithm” I described above essentially deals with three situations, which makes the *select* case statement ideal for implementing it. After some experimentation and “tweaking,” I settled on the following code, which I placed in the *controlTemp* subroutine, and called it each time through the main *do/loop*:

```
select case error
  case > 475
    high heat
  case 1 to 475
    high heat
    error = 8 * error MIN minHeat
    pause error
    low heat
  case 0
    low heat
endselect
```

As you can see, this *select* case statement chooses among the following three cases:

First Case: If *error* is greater than 475 (47.5°), the heat is turned on for 100% of the time during the current one second iteration of the main *do/loop*. (The program includes a *symbol heat = C.2* statement, so the *high heat* statement turns on the SSR.) Originally, I chose 500 (50°) as the test point for this case, but I hadn’t yet determined specific *minHeat* values. As we’re about to see in the next section, the test point needed to be lowered to 475 (47.5°).

Second Case: This is where the proportional control algorithm is implemented. If *error* is between 1 (0.1°) and 475 (47.5°) inclusive, the *heatOn* time is re-calculated each time through the main *do/loop*. The calculation occurs in the *error = 8 * error MIN minHeat* statement, which definitely requires a brief explanation. Because the program runs at 16 MHz, a *pause 4000* statement produces a one second delay. However, there are many other statements in the main *do/loop* and the subroutines, so I decided that a *pause 3800* statement would produce the longest delay that could be included in the one second main *do/loop*. I chose 475 as the maximum value for *error* that would “trigger” the proportional control algorithm because $8 * 475 = 3800$.

Obviously, the term *MIN* also requires an explanation. *MIN* is a PICAXE mathematical function that limits the results of a calculation to a minimum value. For example, if *error = 475* (the maximum value that would cause this portion of the *select* case statement to execute), $8 * 475 \text{ MIN } \textit{minHeat}$ simplifies to $3800 \text{ MIN } \textit{minHeat}$. Assuming *minHeat* never needs to be greater than 3800 (trust me, it won’t!), we’ll get the following results: As the *error* decreases throughout the range from 475 to 1, the recomputed value of *error* also decreases proportionally, starting at 3800 but never going below the value of

minHeat for the chosen target temperature.

Third Case: If *error* equals 0, the current temperature is at or above the target temperature, so the heat is immediately turned off.

Determining Values for *minHeat*

Now that we have the proportional control algorithm in place, we can return to the issue of how to compute *minHeat* values that will stabilize the smoker temperature at various target temperatures. In the early stages of this project, I decided that I wanted to be able to control the smoker temperature throughout a range of 100° to 250° inclusive. I chose 100° as the lower temperature limit because I frequently dry citrus peels, fresh ginger, and fresh herbs (with or without smoke) for making bitters and tinctures, and 100° is the lowest temperature I use for that purpose. At the other end of my temperature range, I chose 250° as the top limit because all my “low and slow” BBQ cooking (again, with or without smoke) is done at or below 250°.

At this point, I need to briefly outline the process that we’ll carry out to develop the algorithm we’ll use to stabilize the smoker’s temperature at the user’s chosen target temperature. Essentially, we’ll implement a two-phase plan. In phase one, we’ll use a “trial and error” approach to determine the value of *minHeat* that results in a stable smoker temperature at the two “extreme” temperature points in our range: 100° and 250°.

In order to accomplish that goal, we’ll repeat the following steps for each of the two extreme temperatures, until we arrive at the correct value of *minHeat* that stabilizes the smoker’s temperature at the given target temperature:

- Make an “educated guess” for the value of *minHeat* that will accurately maintain the target temperature.

Resources	Product #	Source
K-Type Thermocouple Amplifier Breakout Board (AD8495)	1778	adafruit.com
K-Type Thermocouple Probe for Smoker or Oven (max. 900°F)	TC-K3MM	auberins.com
Thermocouple Connectors, Male and Female	Connectors	auberins.com
Solid-State Relay, AC Load, 25A, Zero Crossing	MGR-1D4825	auberins.com
Heatsink for 25A SSR	HS25ET	auberins.com
RJ45 Breakout Board	BOB-007	sparkfun.com
RJ45 Connector	PRT-00643	sparkfun.com

- Edit the program statement that immediately follows the note that's enclosed by a border of asterisks (a little before the beginning of the main *do/loop* in the SmokerControl.bas program), so that the value of *minHeat* is set to our "educated guess."

- Run the program, and enter the desired target temp.
- After at least 75 minutes, save the data file with an extension of "csv" (comma separated values) so the file can be opened in Microsoft Excel.

- Open the file in Excel, then graph the data by selecting all the data entries (including the "Time" and "Temp" column headings) and then selecting "Charts" >> "Scatter" >> "Smooth Lined Scatter" from Excel's Tab menu.

- Select the portion of the data that is reasonably stable, and compute the average of the stabilized data using the Excel "average" function. In order to be consistent throughout all the tests I ran, I defined the beginning of the stable region as the first low point in an oscillation that occurs at or after the 20 minute mark, and the end of the stable region as the last low point in an oscillation that occurs at or before the 75 minute mark. (If that explanation is a little confusing, it should become clear in the next section.)

- Adjust the value of *minHeat* up or down as necessary, and repeat the above steps until the correct value of *minHeat* is determined for the given target temperature.

In Phase 2 of this process, we'll develop an algorithm that successfully controls the smoker temperature at any chosen target temperature between 100° and 250°.

Determining *minHeat* Values for 250° and 100°

At this point, we don't have any information upon which to base our first educated guess, so we'll just start in

the middle and see what happens. Because the program is running at 16 MHz, setting *minHeat* equal to 2000 results in a 500 ms pause during each pass through the one second (1,000 ms) main *do/loop*. In other words, the heat will be on for 50% of the time, and the stable temperature that results will provide us with helpful information for making our first guess for *minHeat* at a specific target temperature. If you look at the program statement that immediately follows the note that's enclosed by a border of asterisks in the SmokerControl.bas program, you'll see that *minHeat* equals 2000, so we're ready to run the program for the first time.

The Excel chart that resulted from collecting the data for a 75 minute session is shown in **Figure 5**. The overall waveform is very similar to that of a full PID algorithm. After initially overshooting the target temperature by a few degrees, the waveform soon settles into a relatively stable pattern of oscillation. When I analyzed the Excel data, I found that the average magnitude of the oscillations was within $\pm 0.6^\circ$ and the maximum magnitude of the oscillations was within $\pm 1.8^\circ$. A full PID algorithm would certainly control the oscillations more tightly than that, but I was more than pleased with the results produced by my first test.

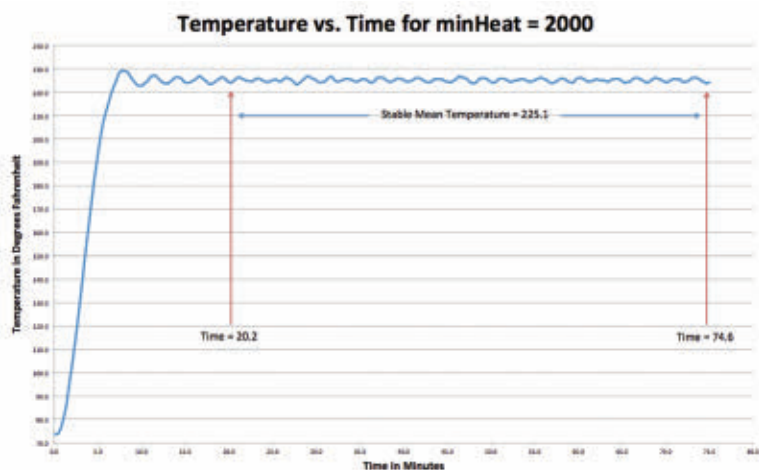
Earlier in this article, I mentioned that I would be satisfied if the temperature controller project was able to maintain the smoker temperature within $\pm 5^\circ$ of the target temperature, and this initial result far exceeded that goal!

In addition to the above encouraging results, finding that a *minHeat* value of 2000 produces a stable smoker temperature of 225.1° provided the information than I needed to make an educated guess. I decided that the next logical step would be to determine the *minHeat* value for the upper limit of my chosen temperature range (250°), so I ran the program again, this time with a *minHeat* value of 2500.

That value turned out to be a little too low, so I increased it to 2600 for the next test, which worked perfectly. The stable temperature was 250.0° to the nearest 10th of a degree, which was fine with me.

Next, I turned my attention to the lower limit of my chosen temperature range (100°), and arbitrarily guessed at a *minHeat* value of 500°, which produced a stable temperature of 125.0°. So, I lowered my next guess to 250° and tried again. This guess also worked perfectly. The stable temperature was 100.0°.

At this point, you may be thinking that I'm either incredibly lucky, or just fudging the data! Many additional trials showed me that there's a wide range of *minHeat* values that produce the same temperature when rounded to the nearest 10th of a degree. For example, somewhere along the line, I also tested a *minHeat* value of 2700, and it produced the same (rounded) value of 250.0° that I



■ **FIGURE 5.** Excel chart for *minHeat* = 2000.

obtained from a value of 2600. The point is that I found myself guessing values that were multiples of 50 or 100, and always found a value that produced the stable temperature I was looking for.

Developing an Algorithm to Determine *minHeat* Values

At this point, I had accurate *minHeat* values for both ends of my temperature range. In the best of worlds, the relationship between *minHeat* and *targetF* would be linear, and the graph of the relationship would be a straight line. If the relationship were linear, the *minHeat* value that's required to maintain the smoker's temperature at 175° (the average of 100° and 250°) would be 1425 (the average of 250 and 2600). To see whether that was the case, I ran the program again using a value of 1425 for *minHeat*. When I examined the Excel data, the stable temperature turned out to be 175.2°, which was a strong indication that we're dealing with a linear relationship. Therefore, the next logical step would be to determine the linear equation, and then use that equation to test its accuracy in maintaining several additional target temperatures in the range of 100° to 250°.

In order to derive the linear equation, a little high school algebra is necessary. You probably remember the standard form for a linear equation: $y = mx + b$ (where m is the slope of the line, and b is the y intercept), and the "slope intercept" method for determining the linear equation given two points on the line. For our smoker algorithm, *targetF* is the x variable, and *minHeat* is the y variable, so the two points that we have so far are (100, 250) and (250, 2600).

In the slope intercept method, the first step is to calculate the slope of the line which equals the "rise" divided by the "run," or the difference in the y values, divided by the difference in the x values. Therefore, for our smoker algorithm:

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2600 - 250}{250 - 100} = \frac{2350}{150} = \frac{47}{3}$$

In the second step, we need to substitute the value we obtained for the slope and the values from either one of the two points back into the standard linear equation, and then solve for the y intercept (b):

$$y = mx + b$$

$$250 = \frac{47}{3} * 100 + b$$

$$250 = \frac{4700}{3} + b$$

$$b = 250 - \frac{4700}{3}$$

$$b = \frac{750}{3} - \frac{4700}{3} = \frac{-3950}{3} \approx -1317$$

Now that we have values for the slope and the y intercept, we can substitute them back into the standard linear equation:

$$y = \frac{47}{3} * x - 1317$$

For our purposes, $x = \text{targetF}$ and $y = \text{minHeat}$, and we will do the multiplication before we do the division, in order to minimize the error when the 08M2 truncates the remainder of the division. Also, we'll add 1 to the result of the multiplication, which is a little trick we have used before to get the PICAXE compiler to correctly round the result of the division. Therefore, our algorithm will be implemented by the following code snippet:

```
minHeat = 47 * targetF + 1
minHeat = minHeat / 3 - 1317
```

Testing the *minHeat* Algorithm

In order to determine whether the above algorithm would produce results that are accurate enough for our purposes, I again edited the program statement that immediately follows the note that's enclosed by a border of asterisks. This time, I removed the statement altogether, and replaced it with the above two-line code snippet.

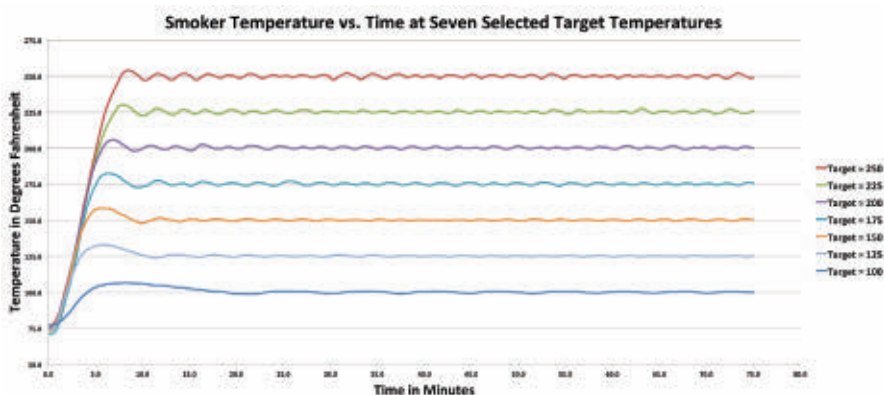
After downloading the program to the 08M2 stripboard circuit, I ran a series of seven 75 minute tests, with a "cool down" period between each test. For the first test, I selected 100° as the target temperature, and for each subsequent test, I increased that value by 25°.

Figure 6 summarizes the numerical data I obtained. As you can see, the largest average temperature error was +0.3° at a target of 200°. Of course, there could easily be slightly larger errors at target temperatures that I didn't test, but I doubt that they would be any greater than ±0.5°. Also, the largest deviation above or below the selected target temperature was ±2.5°.

Again, assuming there may be a few slightly larger deviations at untested target temperatures, I think it's safe to conclude that the algorithm effectively controls the smoker temperature within the range of ±3°.

Smoker's Stable Temperature Oscillations						
		Maximum Under targetF	Average Under targetF	Average Smoker Temp.	Average Over targetF	Maximum Over targetF
targetF	minHeat					
100	250	-1.3	-0.5	100.0	+0.3	+0.5
125	641	-0.6	-0.1	125.1	+0.2	+0.8
150	1033	-0.7	-0.3	150.0	+0.3	+0.8
175	1425	-1.2	-0.5	175.2	+0.7	+2.1
200	1816	-0.8	-0.3	200.3	+0.6	+1.8
225	2208	-1.3	-0.4	225.1	+0.7	+2.5
250	2600	-2.5	-0.7	250.0	+0.6	+2.1

■ **FIGURE 6.** Summary data for selected target temperatures.



■ FIGURE 7. Composite chart for selected target temperatures.



■ FIGURE 8. Ultimate test results.

Figure 7 is a composite chart that visually displays the results of the final seven tests. The chart reveals two interesting relationships: Compared to the higher target temperatures, lower targets result in longer “settling” times, but they also result in oscillations with smaller magnitudes. In addition, the most important conclusion that can be drawn from Figure 7 is that our simple linear algorithm is very effective at controlling the temperature of the smoker throughout our chosen temperature range.

However, as we say, “the proof is in the pudding,” so I felt obligated to conduct one final test: I smoked a rack of baby back ribs for five hours at 225° just to be sure our algorithm is effective. The results are presented in Figure 8, and you can draw your own conclusions. However, as far as I’m concerned, I’m a very happy smoker!

That’s All, Folks!

It’s been more than eight years since we conducted our first PICAXE experiment (“Music, Maestro!”) in the December 2007 issue of *Nuts & Volts*. We’ve come a long way since then, and I’ve had a lot of fun developing and writing about our various experiments and projects. However, it’s time for me to move on.

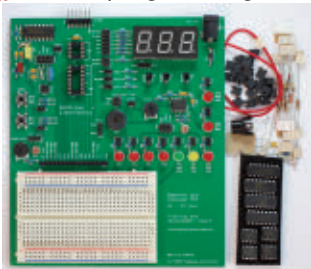
This is my last PICAXE Primer column for *Nuts & Volts*, but I’m not yet disappearing from the face of the earth — at least, I hope not! I plan to continue experimenting with PICAXE (and other) projects, and writing an occasional article for *Nuts & Volts*. I also intend to continue developing printed circuit boards for some of my projects, as well as writing about them and making them available on my website.

I hope you’ve enjoyed reading the PICAXE Primer as much as I’ve enjoyed writing it, and that you’ll continue developing your own PICAXE projects.

Whatever you do, have fun! I know I will! **NV**

Premium tools for makers and creators

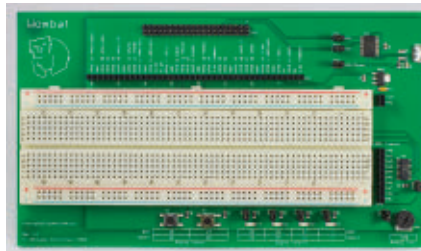
Learn PIC programming



Tutorials and training board

Clear lessons
C and assembler
Hands on examples
Includes enhanced mid-range PICs

Create original projects for Raspberry Pi



The Wombat prototyping board

Provides easy access to all GPIO pins
Adds functionality to your Pi, such as:
analog inputs, LEDs, pushbuttons,
USB serial console port
Includes a set of projects to get you started.

NEW!

See our Joey - a 7-segment LED display for Raspberry Pi!

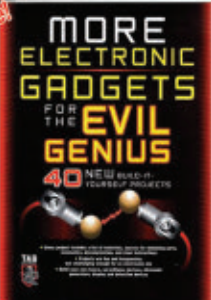
www.gooligum.com

Most popular this! We have 33 amazing titles of PROJECTS FOR THE EVIL GENIUS

Need a project to help you spring into summer?

Check out all the titles on page 49, or visit: store.nutsvolts.com

Only \$19.95 ea



A L L
E L E C T R O N I C S

www.allelectronics.com Order Toll Free 1-800-826-5432

12VDC SPDT 40A AUTOMOTIVE RELAY
12 Vdc, 88 ohm coil. Contacts rated 40A.
CAT# RLY-351 **\$2.55 each**
10 for \$2.25 each




SOCKET FOR AUTOMOTIVE RELAY
5 wire socket
CAT# SRLY-2 **\$2.25 each**
10 for \$1.75 each



10 MINI JUMPERS
1" clips. 2 each of 5 colors.
CAT# MTL-10 **\$2.95 each**



LED MOTION SENSOR LIGHT
Turns on when it senses movement and turns off after 90 seconds. 3.25" x 2.03" diameter. Uses three AAA batteries (not included).
CAT# MSL-04 **\$5.95 each**



9 VOLT BATTERY SNAP
3" wire leads. **4 for \$1.00**
CAT# BST-3 **100 for 15¢ each**



2.1MM COAX POWER JACK, PANEL MOUNT
CAT# DCJ-21 **\$1.75 each**
10 for \$1.60 each



SPST N.O. PUSHBUTTONS
Dome-shaped momentary push-button. 0.70" dia. black plastic bezel. 0.45" button. Extends 0.3" above mounting plane. Threaded bushing mounts in 0.52" dia. hole.

Red Button	CAT# PB-138
Black Button	CAT# PB-139
Green Button	CAT# PB-172G
Blue Button	CAT# PB-172B
Yellow Button	CAT# PB-172Y

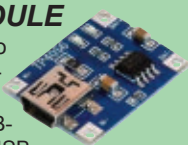
\$1.50 each 100 for \$1.20 each



1" SPEAKER IN ENCLOSURE
Great-sounding 8 Ohm speaker. Sealed 1.25" square x 0.90" enclosure. Mylar cone with rubber surround.
CAT# SK-61 **\$2.00 each**



LITHIUM BATTERY CHARGING MODULE
TP4056 charger for up to 1A Lithium Ion rechargeable battery. Input: 4.5 - 5.5Vdc. Suitable for USB-powered charger. Mini-USB input jack. 1.05" x 0.75" x 0.20".
CAT# BC-4056 **\$3.75 each**



GREAT PRICE! 2 GB MICRO-SD MEMORY
Kingston SD-CO2G 35.
CAT# MSD-2GB **\$2.95 each**
10 for \$2.80 each
100 for \$2.50 each



MAKE YOUR MACHINE **MOVE**

MICRO LINEAR ACTUATORS for RC devices

specifications 10mm-200mm stroke
6v-12v power supply
25kg+ available force
15g-100g net weight

firgelli

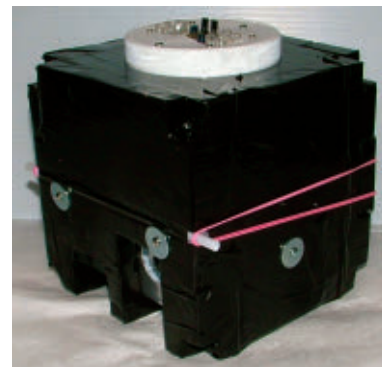
www.firgelli.com





A New Vision of BalloonSats

I've built a lot of BalloonSats in my day; every one of them different and not all of them a work of art. After studying the standardization found in CubeSats, I began wondering if BalloonSats might benefit from some standardization themselves. So, here's my first (and very humble) attempt.



BalloonSats are simplified models of CubeSats, which are four inch cubes packed with electronics (in fact, there can be an overlap in the electronics found in both). Four inch cubes, however, seemed to be a bit small for BalloonSats. That's because its junior high and high school students who don't have a lot of building experience who are making a lot of the BalloonSats. Therefore, I chose five inch cubes for this experiment in standardization.

Past experience has shown that most cameras, video cameras, and experiments can fit inside a cube this size. However, if it turns out that five inch cubes are still a bit too small for many young students, then the standard can easily be raised to a six inch cube.

The Division of Function Inside a BalloonSat

The experiments found in most BalloonSats include some kind of camera, data logger, and set of sensors. Based on this, I split the BalloonSat airframe into three functional compartments: the Imaging Bay, the Avionics Bay, and the Sensor Bay. For protection from the cold of near space, the avionics and camera should remain inside the warmer airframe. The sensors, on the other hand, must be exposed

to the near space environment, and therefore can be located outside the airframe.

Since the data logger is often a smart device — like a programmable flight computer — I felt avionics should be placed in a central location relative to the camera and the sensors. Combining these factors, I

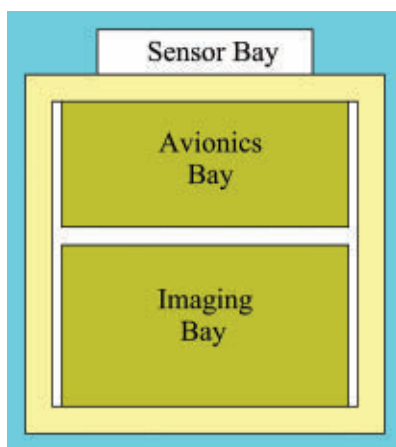
came up with the configuration shown in the figure for a standard BalloonSat.

Avionics Bay

The next question was how to create a simple method for dividing up the interior of a BalloonSat. The answer occurred to me while mailing packages at my local post office. I observed that some of the mail is carried inside of plastic boxes made from Coroplast — a commercially available corrugated plastic. Coroplast is made from polypropylene which is a slick white plastic that is difficult to glue anything to.

So, rather than using glue on Coroplast, I decided to bolt items to bent sheets of Coroplast which is what is done with letter boxes at the post office.

Coroplast has internal channels running its length which make it difficult to bend. However, if you first cut through a wall of one of the channels, it becomes very simple to bend the material along the cut channel. So, to create the divider for the interior of the BalloonSat, I cut through the skin of two channels in a sheet of Coroplast and then folded it into a square horseshoe shape. The sheet of Coroplast was 48 channels wide, and



The three bays of a standardized BalloonSat airframe.



A sheet of Coroplast after two channels are cut open.

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/april2016_Near-Space-BalloonSats.

I cut through the 13th and 36th channels. I cut a second sheet of Coroplast at its 9th and 32nd channels to make a second but shorter square horseshoe.

The two horseshoes were bolted together back to back to create an “H” that holds the flight computer. This Avionics Deck slides inside the airframe and creates the functional divisions inside the airframe.

Sensor Bay

The Sensor Bay is simply a white plastic lid; the same kind found on food containers like peanut butter jars. I selected a white 89 mm diameter plastic lid sold by the Container and Packaging Store in Eagle, ID. I do not work for this company; however, they’re conveniently located and carry a large selection of products.

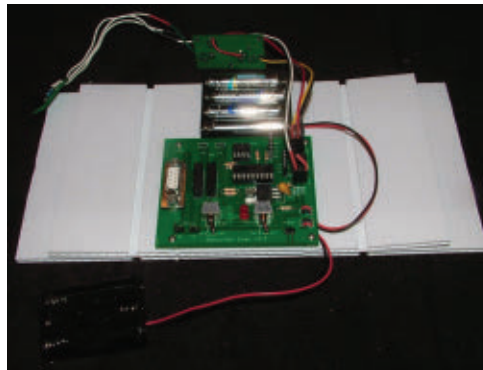
The lid is tall enough to cover most sensors. So, after placing the sensor beneath the lid, I bolted the lid to the top of the airframe with a pair of nuts and bolts. I can then drill a series of ventilation holes around the rim of the lid to permit air – but not sunlight – to reach the sensor beneath.

Plastic lids come in a variety of colors, but I recommend using white to prevent the absorption of sunlight from warming up the sensors and giving false readings.

I did not drill ventilation holes in the BalloonSat for this article since it’s a test for a new LED photometer, plus the LED sensing elements must extend above the lid. In this case, I carefully cut the top of that lid so that only the LEDs were exposed and the temperature sensor remained covered.

Imaging Bay

One of the frustrations I’ve experienced with BalloonSats is the need to build a new one every time I want to change the pointing direction



A BalloonSat Easy flight computer bolted to two sheets of Coroplast.

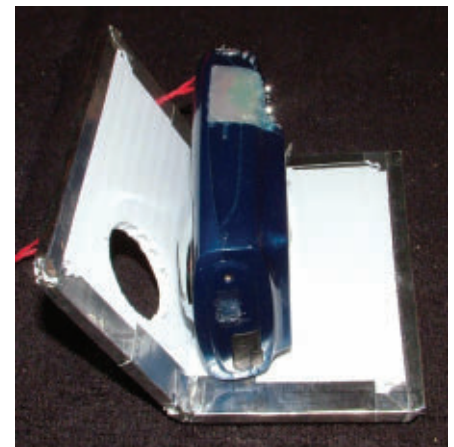


After bolting the flight computer, the “legs” of the Coroplast form an H shaped shelf that slides snugly into the airframe.

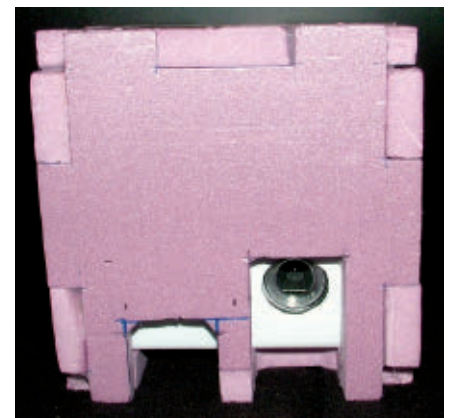
of the camera. For several years, I’ve toyed with the idea of designing some sort of rotating camera mount for a BalloonSat. However, every time I’ve looked into this, it seemed to add too much complexity to the design. This time around, however, I came up with a better idea.

I designed a camera pallet that bolts inside the airframe in one of two orientations: horizontally or vertically. The pallet supports the camera while the airframe supports the pallet in the desired camera orientation.

The pallet was made from



Here’s the camera bolted to its pallet. The lens hole is visible on the left side. I covered the exterior face of the Coroplast with aluminum tape in the hope that it will help retain warmth inside the airframe.



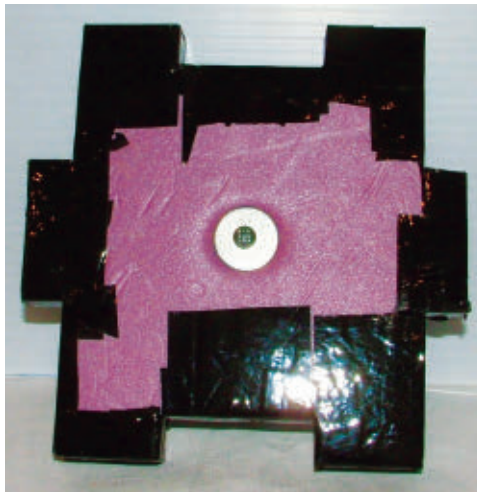
The Camera Pallet is visible inside of this airframe. The two openings are necessary because the camera lens and camera bolt were too far apart for a single smaller opening. If the camera’s mounting bolt was located underneath the camera lens, I would have gotten away with making a single opening in the airframe for the camera.

another sheet of Coroplast with one channel cut so the Coroplast could be folded at a 90 degree angle. I cut two holes in the Coroplast camera pallet: one where the camera is bolted to the pallet; and the second where the camera lens peers through the pallet.

While the camera was still bolted



Before cutting and taping the airframe together, I took this photo to show how well the dovetailed sides of the BalloonSat fit together.



The exterior side of the hatch is covered in black packaging tape. In the middle, you can see the fender washer used to firmly attach the hatch handle to the hatch itself.

to the pallet, I drilled four small holes through the corners of the pallet so it, the pallet, and camera could be

bolted to the inside of the airframe.

After completing the Camera Pallet, I cut two rectangular openings

in the airframe. These openings were for the camera's lens and mounting bolt. The two openings could be combined into a single opening, depending on the location of the camera lens and mounting bolt. However, the camera I used placed the camera's mounting hole too far away from the lens for there to be a single opening in the airframe.

Now, after a successful near space mission, all I need to do to change the pointing direction of the camera is to remove the Camera Pallet, flip it around, and then bolt it back inside the airframe.

Wrapping Up

Prior to starting down this road to standardization, I saw a toy in which children connect six

The Easiest Way to Design Custom Front Panels & Enclosures

Free
Front Panel
Designer



You design it
to your specifications using
our FREE CAD software,
Front Panel Designer

We machine it
and ship to you a
professionally finished product,
no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

**FRONT PANEL
EXPRESS**

FrontPanelExpress.com



AP CIRCUITS
PCB Fabrication Since 1984

As low as...

\$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com



interlocking neoprene foam sheets to form a cube. The neoprene foam sides dovetailed together to create the cube. This got me thinking about doing the same thing with Styrofoam.

In the BalloonSat I created for this month's article, I tested hand cutting dovetails into the Styrofoam squares of the airframe. Even with my less than perfect hand cutting, the pieces snapped together quite well.

I glued five of the six walls of the BalloonSat together. Since the sides were identical, it didn't matter which five I glued. The sixth piece is the hatch which I bolted a handle to. A large fender washer inside the hatch ensured the screw holding the handle to the hatch didn't crush the Styrofoam.

The fit between the dovetails in the hatch and the rest of the glued airframe is tight enough that I believe in some cases, no other method is

needed to keep the hatch closed. This is especially true if the Avionics Deck is bolted inside the airframe as shown in the figure. However, I would still recommend wrapping at least one rubber band around the airframe and hatch for security.

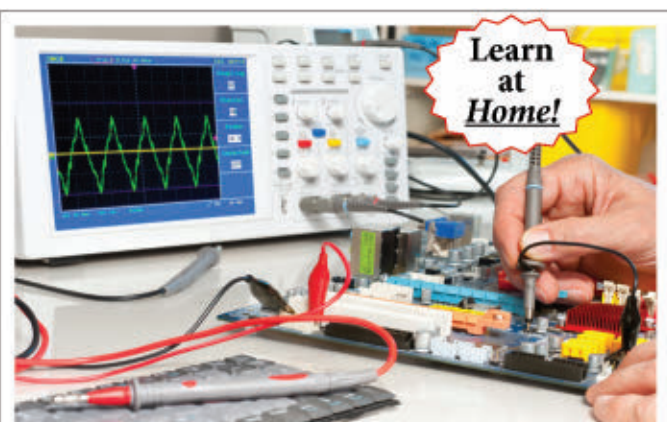
Next up? I think I'll create jigs for cutting Avionics Deck pieces and the dovetailed sides of the BalloonSat airframe. That would make it easier to provide standardized BalloonSat kits that include a flight computer (a miniature science laboratory on a printed circuit board), a variety of sensors, and a camera.

This, I believe, will make BalloonSats an even better STEM machine.

Onwards and Upwards,
Your Near Space Guide **NV**



In this view of the BalloonSat, you can see the white plastic knob I used for the hatch handle, and one of the two bolts holding the Avionics Deck inside the airframe. A single rubber band wraps around the airframe and holds the hatch more securely in place.



Learn
at
Home!

Electronics Courses

Cleveland Institute of Electronics

Train at home to be a professional *electronics* or *computer* technician! Learn with hands-on labs, instructor support, online exams, videos and instructor led chat rooms.

FREE course catalog www.cie-wc.edu or (800) 243-6446

- **NEW!** Robotics Automation Lab
- **NEW!** Computer Security Specialist
- Industrial Electronics with PLC
- Broadcast Engineering
- Electronics with FCC
- PC Troubleshooting
- Electronics Troubleshooting
- Networking

Visit www.cie-wc.edu and start today!

CIE: 1776 E. 17th St., Cleveland, OH 44114 • (800) 243-6446 • Registration 70-11-0002H



EARTHLCD.COM
"The Smart LCD Company"

Pi-RAQ

An open source Raspberry Pi Based 1U Rack Mount Internet Appliance With the World's Only 10" x 1"

With the innovation of the Pi-Raq and existing EarthLCD 10x1 display, 1U Rack mount equipment, such as power controllers, audio/video switchers, network hubs and routers, can now have an easily programmable front panel with a color TFT LCD.



Features at a glance:

- 10" x 1" Color TFT
- 1024 x 100 pixels
- 250 Nits
- 7 Function Tact and Scroll Wheel
- 20V Input Power Supply
- 10x1 LCD Adapter Board
- Raspberry Pi 2
 - 4 USB Ports
 - 100 MHz Ethernet Port
 - 900 MHz Arm Quad Core CPU
 - 1 GB RAM / 4 GB Micro SD Flash

Not exactly what you need? No problem! EarthLCD can customize a solution just for you!

Contact Us:

For more information or to request a quote
949-248-2333 Direct
sales@earthlcd.com
www.Pi-RAQ.com

All logos and trade marks are respective property of their holder. © Copyright 2015, Earth Computer Technologies, Inc.

In this column, Tim answers questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. **Send all questions and comments to: Q&A@nutsvolts.com.**

• Hand Tools for Servicing Electronic Devices

• Mailbag

Hand Tools for Servicing Electronic Devices

Q I am 15 years old and am thinking about a career that involves working on electronic equipment. Could you give me an idea of the hand tools I might need to start buying so that I can work as an electronic technician in the future?

**Sandia Bradford
Steubenville, OH**

A I always like to think of hand tools as little (and sometimes not so little) machines that help us do our jobs better. Hand tools are designed to be an extension of our own hands, and are usually specialized in order to perform a specific set of tasks. For example, screwdrivers are designed for removing threaded fasteners where your hands are not able to. Hammers are used for beating things which would destroy your hands if you tried to use them to perform the same job.

I classify hand tools into:

1. Drivers - screwdrivers, nut drivers, ratchet wrenches, etc.
2. Pliers - needle-nose, chain-nose, diagonal cutters, nippers, etc.

3. Measuring tools - rulers, calipers, gauging tools, templates, etc.; marking tools - pencils, scribes, etc.
4. Electronics testing tools (DMM, VOM, transistor/diode testers, oscilloscopes, signal generator/detector, etc.)
5. Cutting tools - saws, knives, etc.
6. Soldering tools - soldering iron/gun, solder removal tools, soldering assist tools, etc.

Before you start accumulating tools, you first need to decide:

1. What functions you want from your tool collection (electronics only, home repairs, automotive repairs, etc).
2. Where you will use your tools (shop/workbench, home only, mobile locations).
3. How you will store/transport your tools.

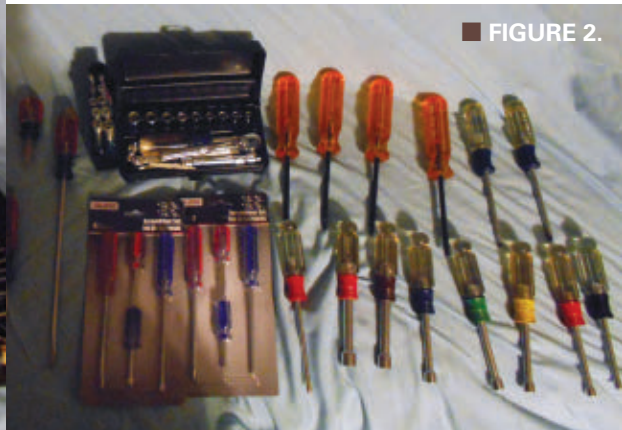
As for my electronics tools, I needed to be able to carry many tools to various locations to conduct repairs, so I opted for a tool case to store my equipment. **Figure 1** shows my tool case with four tool pallets and a space in the bottom for storage. Three pallets fit in the top of the case and one acts as a cover for the storage compartment. Some tool cases have pallets with “wings” that fold out to present more tools. The tools in this case can be used for general electronic and light mechanical repairs.

If at all possible, get a tool case in which the latches will not open if the case is upside down so as to avoid having to put tools back in their correct pouches. A good source for tool cases like mine is TEquipment

FIGURE 1.



FIGURE 2.



QUESTIONS and ANSWERS

Post comments on this article at
www.nutsvolts.com/magazine/article/april2016-QA.



FIGURE 3.



FIGURE 4.



FIGURE 5.



FIGURE 6.

(see **Q&A SIDELINES**). My tool case has a hard cover so I can stack it with other equipment, but you can buy a fabric or leather case if you do not need the ruggedness. If you travel a lot by air, an aluminum case will be more durable than the plastic shell like my case. With tools inside, my tool case weighs 39 pounds. So, if you have to carry the case very far, a wheeled version with an extendable handle will make your life a lot easier.

You can purchase case and tool combinations, but I chose to purchase mine separately to allow me to be more flexible in the equipment I needed. If I need to carry just a few items, I use a soft military surplus bag which is a lot lighter. I have two additional hard shell cases: one is for tools needed for communications repairs (most telephone) and contains various combinations of test leads, RJ-11 jacks/plugs, crimping tools, etc.; the other case is used for mechanical work and contains a mini hacksaw, drill bits, drill motor, nibbling tools, etc.

Figure 2 shows the driver tools I have in the tool case. There are various sizes and lengths of Phillips and slotted screwdrivers. I have Robertson (square) and Torx (star) drivers, plus a Reed Prince driver (has a cross like a Phillips but the grooves are straight instead of curved). I also have various sizes of nut drivers and a small ratchet/socket set in the case. (More drivers to follow.)

Pliers are useful for grasping and holding while you work on electronic devices. **Figure 3** shows a group of pliers I have accumulated over the years to help me with my repair endeavors. I use the standard needle-nose pliers, chain-nose pliers, and diagonal cutters, plus a couple of pairs of bent chain nose pliers. I added a pair of arc joint pliers which have variable sized jaws and are good for removing larger threaded parts (I have used the larger ones to remove difficult soda bottle tops).

I have a pair of wire nippers for cutting larger gauge wire and a pair of duck bill pliers to give extra grip. I included the fuse puller to help in getting fuses out of tight places. The electronic tweezers, vise grips, and hemostat clamps are good for holding components without having to use one of your hands. The arc joint pliers with the white rubber insert are AN-pliers; they are used for removing the knurled AN connectors utilized on some communications devices so as not to mar threaded rings.

Figure 4 shows an assortment of types and sizes of wrenches that are useful in removing bolts and nuts used in electronic assemblies. Always have a good variety of open end, closed end, and adjustable wrenches to fit the fasteners normally found. The closed end wrenches can develop more torque before slipping, but the open end wrenches are easier to use in tight spots. With adjustable wrenches, always turn the fastener in the direction in which the fixed jaw does most of the pulling to avoid slipping. Include both US customary and metric wrenches to be able to remove fasteners from all devices regardless of the country where they were manufactured.

If you are going to align radios or “old timey” TVs, it is good to have a set of non-magnetizable alignment tools. My sets have various shapes of drive heads, and one set even allows you to read the number of turns you have done in case you need to return the setting to its original value.

When working with circuit boards, a good set of picks like those shown in **Figure 6** is a wise investment. These picks (same as used by dentists) can be used to break solder bridges or remove extraneous

copper. The wooden stick is good for probing in “hot” circuits to avoid creating a short circuit which may destroy components other than the ones that were already bad.

As for measuring devices, **Figure 7** shows a sampling of the ones I carry. A small tape measure is good for checking the dimensions of relatively large devices, while the ruler is good for measuring smaller dimensions fairly accurately. I carry several templates for measuring wire size and fastener size (both customary and metric).

Figure 8 shows my soldering tools which include: a soldering iron (several sizes and types of tips are helpful); a vacuum solder removal tool (helps to clean out circuit boards that have already been soldered); braided solder removal tape (works great but hold it with pliers because it gets really hot, really fast); a heatsink to protect active component junctions; an anti-static wrist band for working with CMOS devices; IC removal tools for rapid removal of desoldered integrated circuit devices from circuit boards; a couple of sets of solder aid tools for removing solder bridges, cleaning off soldered joints, or testing solder joints for rigidity; a magnifying glass for checking solder work on small printed circuit boards (PCBs); and a couple of nylon bristle brushes for cleaning PCBs without contaminating them with metal brush whiskers which may create a short circuit (yes, one of the brushes is a repurposed tooth brush).

My cutting tools include: an electrician’s knife for cutting electrical tape; wire strippers for removing the insulation from wire without being cut (as is possible using a knife); automatic wire strippers for consistent lengths of insulation that is removed when you have a lot of wires to strip or small wires (my strippers can strip a range of 30 to 12 gauge wire, plus I have a special set for stripping lamp cord); and a pair of metal snips in case I need to cut metal cases or chassis. **Figure 9** shows my set of cutting tools.

Anyone doing electronic and/or electrical work needs a set of electrical measuring devices. **Figure 10** shows an assortment of these measuring devices: digital multimeter (DMM) for measuring voltage (mine can also measure temperature with its thermocouple probe); current checking diodes/capacitors; a logic probe/logic signal injector for testing digital circuits; a power receptacle tester to make sure you have the correct voltage to your devices in the first place; a voltage tester for “rough” determinations of voltage from 110 to 480 VAC; an inductive probe for checking where there is current without having to breach the insulation; and a set of circuit tracers (plug or wire the sender into a live power circuit and use the tracer to find where the power originates; hopefully the breaker).

Figure 11 shows a group of tools for working on precision (a.k.a., very small) fasteners. I have a set of jeweler’s screwdrivers, precision nut drivers/screwdrivers, a set of precision open end wrenches, and a precision set of hex drivers, slotted/Phillips/Torx/square drivers that can be inserted into a handle.

Figure 12 shows a set of miscellaneous tools: coaxial cable stripper/connector crimper (blue handles); a set of pin punches; a hex wrench set; a vise for splicing tapes (gold); an oiler pen for squeaky parts; two wire wrap tools for working on “ancient” digital technology; and another set of precision drivers for working with security fasteners (good for taking apart “kid’s meal” toys).

Figure 13 shows a miscellaneous set of tools including a small ball peen hammer, a pry bar (so you don’t use/break a slotted screwdriver for prying), and a penlight/inspection mirror for checking in hard-to-reach places.



■ FIGURE 7.



■ FIGURE 8.



■ FIGURE 9.



■ FIGURE 10.

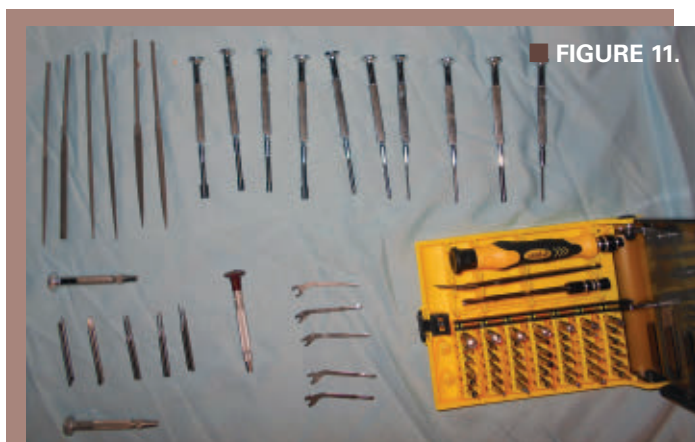


FIGURE 11.



FIGURE 12.

Figure 14 shows a small dual-channel oscilloscope (that can be carried in a tool case) which is powered either using an AC line adapter or internal batteries. This scope is good for use at remote locations but is fairly pricey, so you really have to need the device to justify its purchase cost.

A few hints for improvising if you do not happen to have the right tool with you:

1. A pair of large bent chain-nose pliers makes a good soldering iron holder.
2. Placing a rubber band on the handles of needle-nose and chain-nose pliers makes a good substitute for a component-holding vise in the field.
3. If you do not have a screw holding driver, a bit of chewing gum (or similar material) can hold a screw.
4. For drilling in tight cabinets, a 90 degree drill attachment works well (I have used one to drill holes in an aircraft radio rack without removing the rack from the plane).
5. Put sets of smaller tools in zip lock plastic bags to keep them from getting lost in the bottom of a tool case, yet they are still easily accessible. Plastic bags (compared to bins) are cheaper, more flexible, and easily replaced.

Anyone should feel free to elaborate on these suggestions.

A Few Warnings about Using Tools Safely

If you are using knives, always cut away from your body to avoid stabbing or cutting yourself. Screwdrivers seem fairly benign, but can cause serious punctures and

MAILBAG

Re: Plain Old Telephone Service Blues

Your answer about Plain Old Telephone Service has some things that need to be clarified. The definition of POTS is not limited to only rotary dial phones. Enhanced analog services such as DTMF (Touchtone®), speed-dialing, call forwarding, conference calling, and others are also part of POTS. What is not part of POTS are the digital services provided over the plain old copper wires such as ISDN and DSL.

POTS is also called PSTN (Public Switched Telephone Network) and generally means anything done over the twisted copper pair between a subscriber (your home) and the central office that is analog. While the digital services like ISDN and DSL go over the same wires, their signals are not processed in the central office by the PSTN. In the central office, they use digital computers to handle the analog service to your home.

POTS/PSTN only means the non-digital communications between your home and the central office, and is usually conducted between two or more phone numbers. POTS does not depend on rotary dial phones.

Rick Swenton
Bristol, CT

Thanks for the additional telephone service information. Our readers are a vast source of good intel that helps me as well as other folks. You are quite correct that POTS does include a lot more than the old rotary dial systems. I was trying to limit my answer to the reader's question without giving more information than was asked. One feature of the rotary dial I liked as a kid (back in the dark ages of the late '50s and early '60s) was that you could click the handset off-hook switch the number of times of the number you were dialing instead of using the rotary dial. It sure was a lot more work, but I guess we were hackers even back then trying to use the phone in a way that it was not intended.

lacerations (if I had a dollar for every time I stabbed myself with a slipping screwdriver ...). Soldering irons are notorious for causing burns and collateral damage. I learned the hard way not to reach for the soldering iron on the floor behind you without looking (I found it alright, but not the cool handle end).

We know to avoid the business end of pliers, but you can also get your finger badly pinched in the area behind the pivot. Some pliers have a wire cutter located behind the pivot which can cause a nasty cut too. Ratchet handles can break and cause us to fall, as can slipping wrenches. Whenever possible, pull the handle away from the work piece so if the wrench slips, you do not fall into the work piece. Sometimes we are tempted to put a “cheater” (extend the handle with a pipe or other similar piece) on our wrenches or ratchet handles, but this can damage the tool and possibly cause an accident.



FIGURE 13.



FIGURE 14.

I remember a story from a technician who was aligning microwave antennas in Thailand during the Vietnam Conflict. The military contractors installed the antenna and used a hacksaw to crosscut the threads on mounting bolts to stop enemy insurgents from taking the antenna (I have no idea what they would have done with it). This tech had the largest ratchet handle made by a particular company and broke the bolts loose by standing on a piece of pipe placed

MAILBAG continued

Re: Simple Remote Control

In the January Q&A, pg. 9, Simple Remote Control Answer, I was surprised that your article did not reinforce your advertisers. They do help support the publication. All Electronics — a regular advertiser — has a four-channel on/off 12 VDC remote control for \$38 (look under Misc. at www.allelectronics.com). Linx Technologies (also an occasional advertiser) has a transceiver, antenna, and key FOB available through www.mouser.com (part no. 712-HUM-900-PRO) for less than \$20 ea (for the transceiver) for use requiring higher security transmissions and greater functions. Learn your advertiser's products; it will help you keep them advertising.

Gary
An old subscriber to N&V

It is good to hear from our “old” subscribers. I too am an old subscriber who migrated to Nuts & Volts via Radio Electronics when it ceased to be published back in 2003. I usually try to use our advertisers and I cannot explain why I didn't on the remote control question answer. The N&V advertisers are not only supporting the publication of this magazine, but they are good sources of information and devices to help our readers increase

their competencies in the electronics field. I have actually bought items from both Mouser and All Electronics as an Engineering Technology Department Head at our local community college. I appreciate when our readers point out the errors of my ways so that I can improve Q&A in a way that benefits everyone. I'll try to focus more on our own N&V advertisers in the future.

Re: Scanner Frequency Changes

Regarding your answer to Gary Ross' scanner modification question, I might suggest that he visit a local hamfest where he could pick up a two-meter HT transceiver (for the price of a few custom crystals) to listen to the local repeaters. Who knows, it may provide an incentive for him to obtain his ham license.

Ira Wexler KB3I

Thanks for the additional information on scanner modification. Ham radio is a great field in electronics, and radio is my “first love” of all of the electronics branches. A good source of information on becoming a ham operator and available hamfests in your area is the American Radio Relay League (ARRL) at www.arrl.org. I hope this information helps any other readers who are interested in the radio field.

around the handle and jumping. After a few weeks, the ratchet handle broke (hope they carried a spare). Back in the US, the tech carried the handle in for a warranty replacement and the sales clerk said, "You must have had a mighty big helper."

Another incident involved a two inch pipe that was not going to break loose; we were in a chemical plant where heating the pipe or cutting it with a torch was not an option. Our ingenious pipefitter took a 21 foot piece of pipe, put it on a four foot pipe wrench handle (a.k.a., very thick metal handle), and then four 200 to 300 pound pipefitters stood on the pipe "cheater" to loosen the rusty pipe. The pipe wrench did not break, but it did have a marked curvature. (Incidentally, I was not one of the people standing on the pipe.)

Be sure to check out **Q&A SIDELINES** for sources of electronics hand tools. This answer is not a complete list of tools, but can serve as a starting point to develop the set of equipment which meets your own

requirements and budget.

Readers are welcome — as always — to submit their ideas and preferences for tools they use for servicing electronic and electromechanical devices. **NV**

Q&A SIDELINES

Hand Tools for Servicing Electronic Devices

Tool Cases

www.tequipment.net

Hand Tool Sources

www.allelectronics.com

www.parts-express.com

www.tequipment.net

MAILBAG

Sources of Electronics Components and Devices

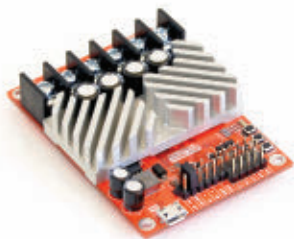
www.allelectronics.com

www.mouser.com

Ham Radio Information

www.arrl.org

Motor Control



- 15 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial

- 45 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial



- 160 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial



www.ionmc.com

Supporting Robotic
Combat for over 10 years!

The Robot MarketPlace

- Combat Robot Supplies
- Motors
- Batteries & Chargers
- Vex Robotics
- Carbon Fiber
- Wheels & Tires
- Speed Controllers
- R/C Systems
- Drive Components
- Hobby Supplies & Toys

(877) ROBOT 99
(877-762-6899)
or (941) 749-6030

Your One-Stop Robot Shop!

RobotMarketPlace.com

NEW PRODUCTS

■ HARDWARE
■ SOFTWARE
■ GADGETS
■ TOOLS

DUAL MOTOR CONTROLLER

Actobotics® dual motor controller provides a compact solution for controlling two brushed DC motors. The small footprint of the controller fits down inside the channel and the mounting holes perfectly align with the hub pattern which perforates the Actobotics building system. The controller has four selectable modes to choose from, which is done by simply moving the jumpers. The four selectable modes are:

(1) Digital Speed Control Mode (RC Mode).

The motor's speed and direction will be controlled by a 1,000-2,000 μ s digital PWM signal.

1,000 μ s = 100% in one direction

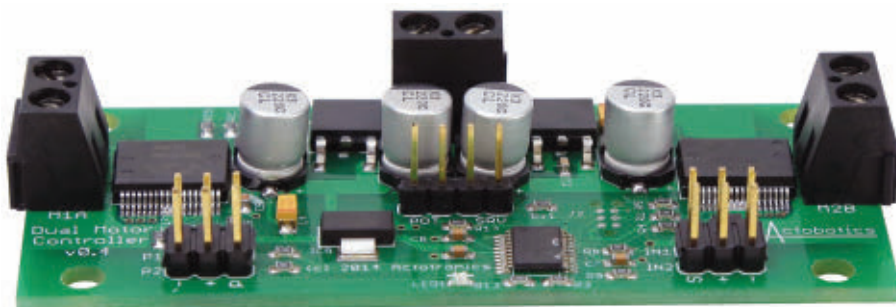
1,500 μ s = No movement

2,000 μ s = 100% in the other direction

(2) Analog Speed Control Mode.

The motor's speed and direction will be controlled by an analog voltage. The voltage is commonly controlled by a potentiometer. Actobotics recommends one with at least 1K of resistance such as their 5K potentiometer board.

0V = 100% in one direction



1.65V = No movement

3.3V = 100% in the other direction

(3) Split Mode.

Channel 1 operates in Position Control Mode while Channel 2 operates under Digital Speed Control Mode.

(4) Position Control Mode (Servo Mode):

In this mode, the motor controller acts like a servo controller. Both channels will utilize a PWM signal as the target position, and an analog voltage (usually via a potentiometer) as the current position. If the current position does not match the target position, the motor will be driven until they match.

The unassembled board is priced at \$89.99; the assembled board is priced at \$99.99.

ECONOMY GEARMOTORS

ServoCity is now offering a new line of 6V-18V economy gearmotors in 10 speeds (19, 25, 34, 43, 56, 98, 124, 170, 485, and 970 RPM). These gearmotors offer extreme reliability at an affordable price. The 4 mm shaft is compatible with many ServoCity gears, hubs, couplers, and wheels. Rated at 12 VDC, users can get nearly 2,700 oz-in of torque from the lowest RPM option. These durable little gearmotors are perfect for robotics and small industrial projects. Pricing is \$14.99 each.



For more information, contact:

ServoCity
www.servocity.com

WEARABLE DEV KIT FOR IoT

MikroElektronika announces the release of Hexiwear: a wearable development kit for IoT (Internet of Things). With a distinctive form factor that makes it look more like a consumer product than an engineering tool, Hexiwear is intended for developers who want to build devices such as wearables, cloud connected remote sensor nodes, or complex controllers for industrial Internet of Things applications.

Developed in partnership with NXP Semiconductors, Hexiwear is powered by a Kinetis K64 microcontroller based on the ARM® Cortex®-M4 core. Another Kinetis MCU (KW40Z) provides Bluetooth Low Energy connectivity. Hexiwear also integrates a wide variety of sensors, as well as a user interface consisting of a 1.1" full color OLED display and six capacitive buttons with haptic feedback. The suggested retail price is \$49.

Full hardware specifications are as follows:

- MCU: NXP Kinetis K64x MCU (ARM Cortex-M4, 120 MHz, 1M Flash, 256K SRAM)
- BLE: NXP Kinetis KW4x (ARM Cortex-M0+, Bluetooth Low Energy, and 802.15.4 Wireless MCU)
- 3D Accelerometer and 3D Magnetometer: NXP FXOS8700CQ
- Three-Axis Digital Gyroscope: NXP FXAS21002
- Absolute Digital Pressure Sensor: NXP MPL3115A2R1
- 600 mA Single-cell Li-Ion/Li-Polymer Battery Charger: NXP MC34671
- Light-to-digital Converter, Digital Humidity and Temperature Sensor, and Heart-rate Sensor
- Additional 8 MB of Flash Memory
- 1.1" Full Color OLED Display

- Haptic Feedback Engine
- 190 mAh 2C Li-Po Battery
- Capacitive Touch Interface
- RGB LED



The functionality of each sensor is showcased through several demo applications that are preloaded into the device. Among them is the Sensor Tag mode that broadcasts the sensor data through BLE.

Compatible iOS and Android apps for monitoring the sensor readings from Hexiwear and logging them to the cloud are available on Google Play and the App Store. Notably, the apps also support Over The Air Programming (OTAP). Both smartphone apps and cloud access are supplied through MikroElektronika's partnership with WolkAbout: a company specializing in tailor-made IoT solutions.

MikroElektronika also announced the Hexiwear docking station. This expansion board with three mikroBUS™ sockets for MikroElektronika click boards™ allows users to enhance the core functionality of Hexiwear through hundreds of additional sensors, actuators, and transceivers.

The docking station also features an onboard programmer for use through USB.

For more information, contact:
MikroElektronika
www.mikroe.com

NATIONAL RF, INC.

TYPE **RF** HFDF
NOISE LOCATION SYSTEM



LOCATE NOISE IN THE HF SPECTRUM
Visit www.NationalRF.com for this
and other Radio Products!
Office: 858-565-1319

Images Scientific Instruments Inc.

PIC Basic Project Board



16F88 PIC Basic Project Board Features

LCD Display - Backlight & contrast control
2 A/D Channels
4 Digital I/O lines
5V and 9V operation
Free Student version of PICBasic Pro & Microcode Studio
Do More Projects:

Frequency Meter	Sensor Reader
Pulse Generator	Elapsed Timer
Radiation Pulse Counter	Volt Meter
Humidity Sensor	Toxic Gas Sensor

www.imagesco.com/microcontroller/index.html

SDP Stock Drive Products
Setting Ideas Into Motion

**One-Stop Shop for
Mechatronic Components**

**EXPLORE
DESIGN
BUY ONLINE**



www.sdp-si.com
no minimum requirement

Designatronics Inc.

ROBOT POWER Extreme Motor
Speed Control!

MegaMoto GT- Mega_Motor Control for Arduino™

- 6V-40V - 35A / 50A+ peak
- Single H-bridge or dual half
- Current and Temp protected
- Jumper select Enable & PWM
- Robust power terminals



The VYPER™

- 8V-42V - 125A / 200A peak!!
- Single H-bridge
- RC and Pot control
- Overload protected
- For your BIG bots

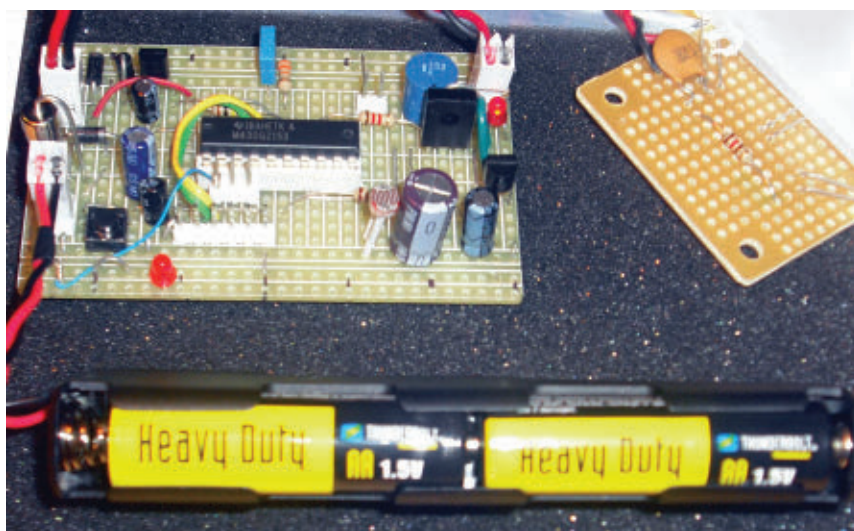
www.robotpower.com
Phone: 360-515-0691 • sales@robotpower.com

Battery Backed LIGHTING SYSTEM

By Jim Lacenski

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?magazine/article/April2016_Battery-Backed-LED-Lighting-System.

Every night, the kitchen under-the-counter LED lighting makes getting that midnight snack a breeze. For years, though, I lamented that there was no on at dark/off at light automatic functionality. In addition, every year in the Pacific Northwest where I live, severe windstorms rip through — windstorms that take down trees and power in their path, plunging the street into darkness. Many times, I power-up the household DC-to-AC inverter ahead of time when I suspect an outage. Unfortunately, sometimes darkness comes faster than I can set flashlights out or power-up. Last November's outage during dinner preparation meant a call for action: Put automatic on/off and battery backup on the under-the-counter lighting system. It was high time for a BBLIS: Battery Backed-Up LED Illumination System.



Design Criteria

The first design criteria were obvious: Have an auto on/off and a battery backup. I knew I wanted to have 12 volts output to the LED set as before, and keep my thrift-store sourced 12 volt adapter. Next, I wanted to be able to use just a few batteries, and have a very power-efficient design. The system would need to fit onto the same credit card sized solderable breadboards that I use for most projects.

Normally, I would reach into my processor bin and grab a Microchip processor. The surprise arrival of some Texas Instruments' (TI) Launchpads onto my doorstep last Christmas Eve provided a "let's learn something new" moment, so I decided to take the plunge and use a TI processor instead.

I contemplated using rechargeable batteries. Due to the very infrequent use and low current drain, designing to inexpensive "heavy duty" or — at most — alkaline batteries was a better fit in this application. With the local "Freight" tool store giving away 24 packs of AA cells at the time, I knew that their heavy duty AA batteries would be my batteries of choice. Four of these would also be easy to hide under the cabinet edge. A simple PWM (pulse width modulation) step-up converter would form the bridge between the four AAs and the 12 volt LED strip.

The Circuit

The schematic is shown in **Figure 1**. The 12 volt AC adapter connects to J1. The six volt battery pack connects to J2, and the LED light strip connects to J3. The top 1N5819 diode provides a low voltage drop path to output when on AC adapter power. The MSP430G2153 senses the adapter input through the 100K/33K resistor divider pair. The photocell (Cd1) provides the light sensing. The boost converter consisting of Q1, L1, and associated components is driven by PWM from a timer when power fails.

On the low side of the load (J3), Q2 is used to switch the output on or off during dark or light. The pushbutton (PB1) allows setting the output voltage higher or lower in steps when on battery power. PB1 also provides manual override during daylight. J4 is used to allow in-circuit programming.

The Code

TI's MSP chip series is designed for extremely low power consumption. The integrated peripherals and core are designed to be powered for weeks — even years — from small power sources such as coin cells. In keeping with this power sipping philosophy, I decided to take advantage of low power features where sensible. The pushbutton uses an interrupt handler instead of polling, and the output from timer TA0 (Timer 0) needs no processor cycles for PWM generation.

It became clear that some polling would still be helpful to the program design. Accordingly, TA1 (Timer 1) is set to be a 250 millisecond clock to wake the main loop. By putting the state handling in the main loop, execution time in the interrupt routines is minimized; this is generally considered good design practice.

The several inputs and outputs are overviewed in ULS_Support.c. The code machine is clearly shown in the main.c file, as part of the main loop. Outputs of the code machine are the PWM output controlled via *PWMControl*, and the output on/off directly controlled by bit 4 of port 1.

While many other MSP family parts can leverage an API library (DriverLib) or even Arduino environment-like code design, I decided that for the simplicity of this project, a combination of bit operations and the *PWMControl*

routine would be most expedient.

After several hit and miss attempts at the overall logic, I moved from the design napkin to lay out the truth table, which is shown in the file ULS_Support.h. With the truth table clearly laid out, laying out the state handling in the main program was a snap. If I were on a code size reduction crusade, more use of bit operators could compact individual variables into a single variable. To be newcomer-friendly, I decided to leave bit crushing for another project.

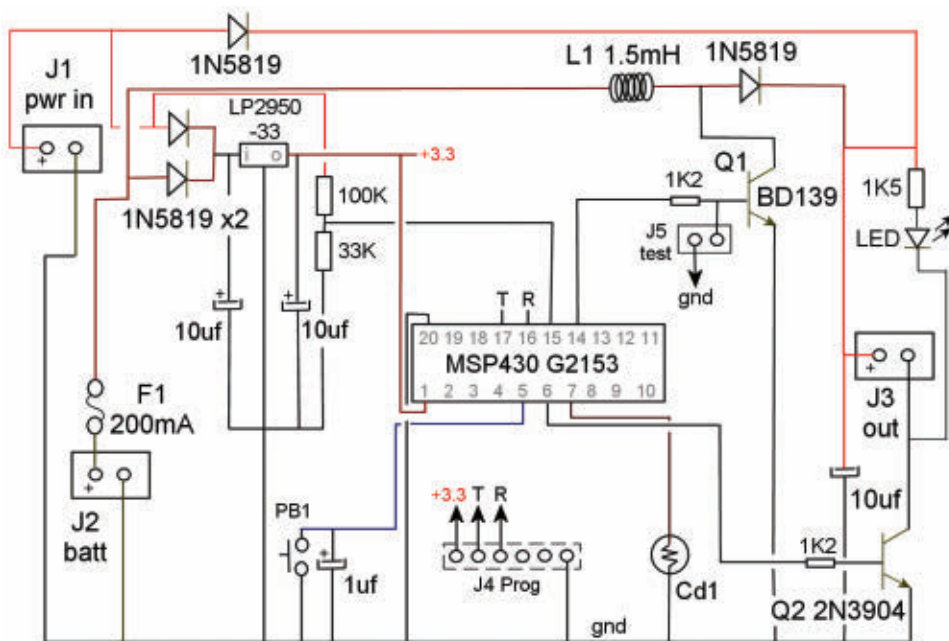
Astute code developers will note that even though the G2153 (or G2553) timers have an output latch which is settable and resettable, I force the timer output to low whenever turn-off is needed. I found that sometimes the timer output could “stick” high. This would cause the output stage to stay on, saturating the boost inductor (L1) and potentially damaging it. Forcing the output low when PWM is off fixes this issue.

Assembly and Programming

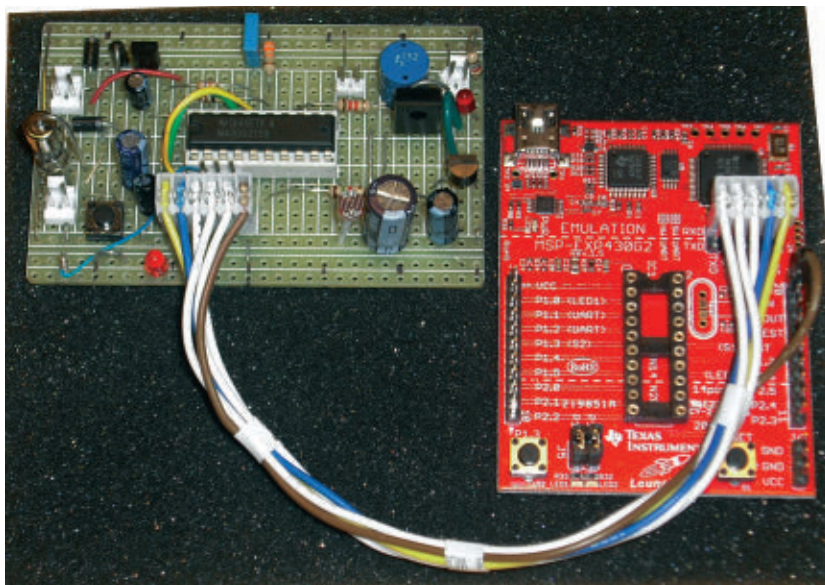
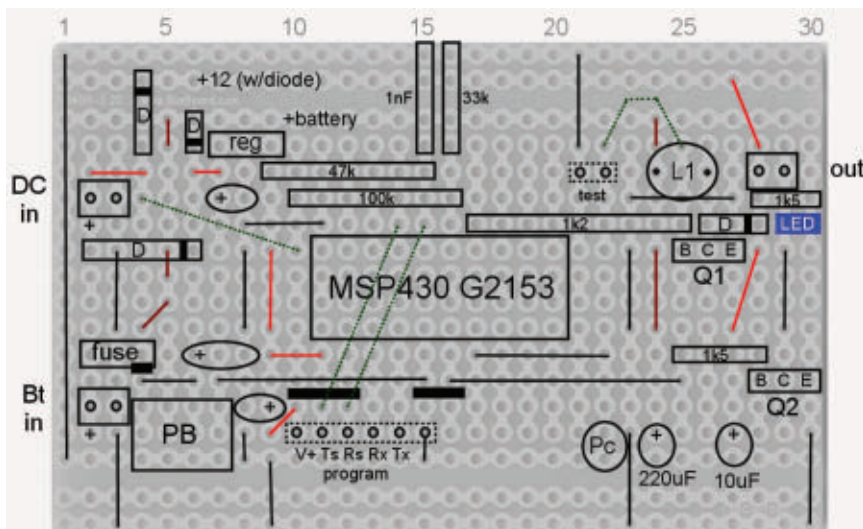
The layout of the prototype is in **Figure 2**. The circuit is assembled on a BPS-400 solderable breadboard. Look carefully for the dark rectangles that indicate which traces need to be cut. I recommend cutting these before installing components. Several jumpers will need to be installed before other components are installed. Closely review the layout before proceeding.

Those of you familiar with the TI G2553 Launchpad will notice how the pushbutton input and main outputs coincide with the key inputs and outputs on the G series

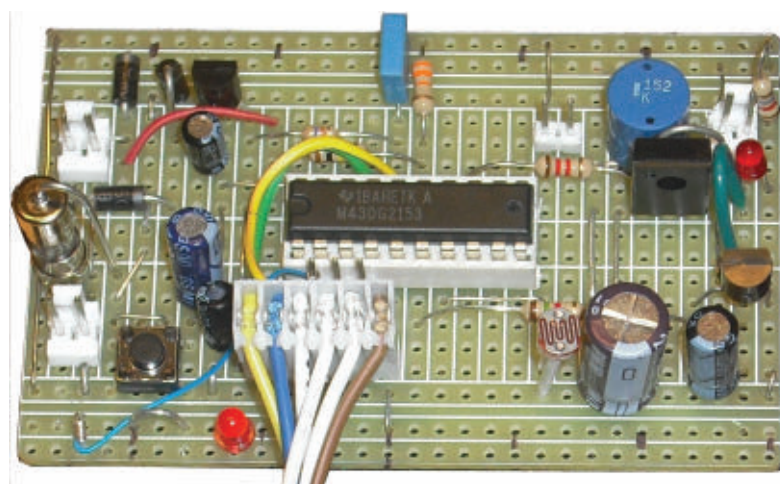
■ **FIGURE 1. Schematic.**



■ **FIGURE 2. Layout master.**



■ **FIGURE 3. BBLIS and programmer.**



■ **FIGURE 4. Board view.**

Launchpad. Much prototyping was done prior to moving the circuit to the solderable breadboard, but there came a point where this design was best finished in-circuit.

Figure 3 shows the simple interconnect cable, the prototype, and Launchpad emulator. Only four lines (Test, Reset, Power, Ground) are actually needed; this is documented in one of the article links shown in **Table 1**. I brought the additional Tx and Rx lines through for future use. A link to more information about creating such a cable is also in the Links table.

Most of the development was done with the G2553 processor that is shipped with the G series Launchpad. Switching to the lower cost G2153 was accomplished with simply changing the destination chip in TI's Code Composer Studio and recompiling.

The completed prototype is shown in **Figure 4**. A five volt LED is visible in the lower right area, which is connected to port P1.0. This is a remnant from the development phase and not needed in the final project, but was very useful for debugging.

A note on parts selection: While many parts have some leeway for different values, note that the output capacitor (lower right of board) should be able to withstand at least 25 mA of ripple current. My surplus house TIP3x transistors had dismal gain, so I opted for a known-first-quality BD139 (Q1) for the up converter. The design uses the pull-up resistor in the MSP430 to pull the photocell line high when dark.

I find that my kitchen needs a fair amount of light for the lights to turn off in the morning, when using the internal pull-up. Using an external resistor for the pull-up would give more control over the "lights off" threshold. Bending the photocell slightly forward can help too.

The Lights

With the advances in LED technology, it was time to replace my old three LED light stick with newer brighter lights. I settled on the CREE white LEDs shown in the **Parts List**. At a rated 28,800

TABLE 1

TI LaunchPad landing page	www.ti.com/launchpad
Adafruit DC-DC up converter page	https://learn.adafruit.com/diy-boost-calc/overview
TI battery charging brief	www.ti.com/lit/an/snva557/snva557.pdf
MSP430 introductory tutorials	www.embeddedrelated.com/showarticle/182.php http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_Tutorials (Many others also available)
LaunchPad workshops	http://processors.wiki.ti.com/index.php/Getting_Started_with_the_MSP430_Launch_Pad_Workshop
Launchpad to project wiring cable	http://43oh.com/2011/11/tutorial-to-use-your-launchpad-as-a-programmer

■ FIGURE 5. LED board.

mcd each, the trio is border line too bright as a night light. My wife has been seen to be loading the dishwasher with none other than the light from this project.

The complete LED board with 120 ohm current-limit resistor is shown in **Figure 5**. In addition to being inexpensive, these LEDs have a very narrow 15 degree dispersion, so a widening lens is used on the center light. This lens is not in the picture, but is included in the **Parts List**.

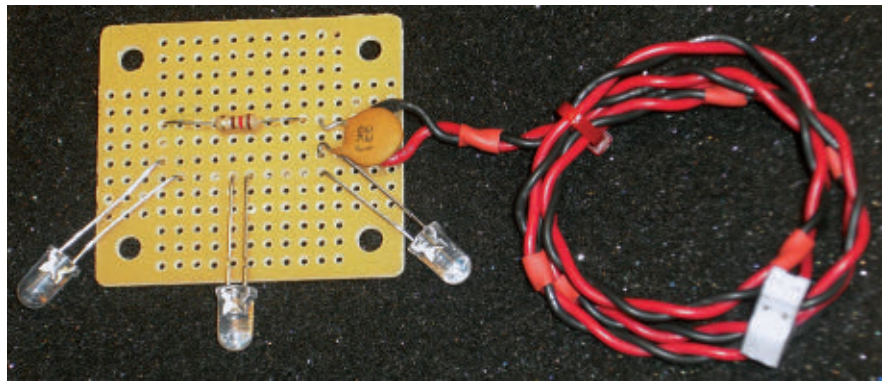
Shining to the back wall of the kitchen counter, the lights cast an oval or round light spot. Not the prettiest, but effective. Light guide design was out of the scope of this project.

Testing

The completed prototype with batteries and LED light board is shown in **Figure 6**. Once all wiring is complete and debugged, code is loaded and power applied, the project is easy to test. The output will briefly turn on at DC input power-up, even if the battery is disconnected. More likely, you will connect the battery prior to powering up from the AC adapter. The PWM up converter should switch on, and power the output immediately.

As soon as the AC adapter power is connected, with normal light levels the output will switch off until dark. When on battery power, pressing the pushbutton will cycle the output and LED brightness slightly higher, then down to a low value with each press before cycling back up.

The setting is remembered for the next power failure. With AC



EARN MORE MONEY Get your dream job!

Be an FCC Licensed Wireless Technician!

**Make up to \$100,000 a year and
more with NO college degree**

**Get your "FCC Commercial License" with our
proven Home-Study Course!**

- No costly school. No classes to attend.
- Learn at home. Low cost!
- No experience needed!
- **MONEY BACK GUARANTEE:** You get your FCC License or money refunded!



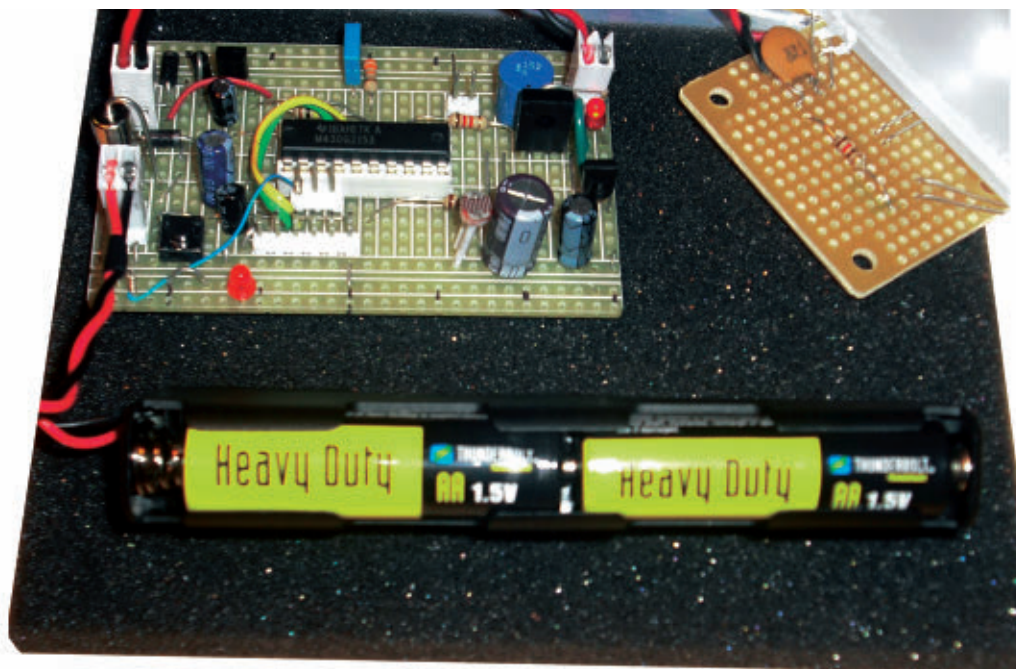
**Your "ticket" to 1,000s of high paying jobs in Radio-TV,
Communications, Avionics, Radar, Maritime & more.**

Call for FREE info: 800-877-8433 ext. 109

or, visit: **www.LicenseTraining.com**

COMMAND PRODUCTIONS • FCC License Training
Industrial Center, 480 Gate Five Rd., PO Box 3000, Sausalito, CA 94966-3000





■ **FIGURE 6. BBLIS complete.**

adapter power applied, pressing the pushbutton will toggle the lights on or off.

How long will the lights stay on? I measured about 35 mA of current consumption to the prototype for the LED assembly shown at normal (near 12 volt) output. This is

simple trickle charge can be accomplished by bridging a 720 ohm, 1/2 watt resistor from the AC adapter input to the battery. Nickel-cadmium — not other battery types — are suitable for this charging method. The TI battery charging application brief in the Links table provides a

with 20 mA to the LED assembly. With those inexpensive “heavy duty” batteries from the local Freight tool store, this means about 34 hours of run time from four AAs. Considerably longer run times are possible with alkaline batteries. I have noticed that commercial LED “uninterruptible” lighting power units have a “by law” run time of 1.5 hours, so set the timeout duration (MaxDuration) accordingly. If you have several interruptions yearly, be sure to change batteries as necessary.

Next Steps

For those who want the lower maintenance of rechargeable batteries, a

DESCRIPTION	ID	QUANTITY	SOURCE, COMMENTS
Two-pin MTA-100 connector	-	3	Digi-Key; A1921-ND, with keying tab
1N5819	-	4	Digi-Key
LP2950-33 voltage regulator	-	1	Digi-Key
10 µf 16 volt capacitor	-	2	Digi-Key; used in voltage regulator stage
200 mA 5 mm fuse	-	1	Digi-Key; 507-1641-ND
100K, 33K, 1.5K resistors	-	1 each	1/4 watt
PB-1	-	1	Digi-Key; P8011S-ND
1 µF capacitor, 16 volt	-	1	Digi-Key
MSP430G2153IN20	-	1	Digi-Key
1.5 mH inductor, 210 mA	L1	1	Digi-Key; TDK TSL series
BD139	Q1	1	Digi-Key; BD137 should be okay
2N3904	Q2	1	Digi-Key
Six-pin MTA-100 connector	-	1	Digi-Key
20-pin IC socket	-	1	Digi-Key, All Electronics
Photocell	Cd1	1	Digi-Key; PDV-P8103-ND
10 µf, 25 volt capacitor	-	1	Low ESR type for output filter, Nichicon UPS series, Digi-Key
1.2K resistors	-	2	Digi-Key
Onboard LEDs	-	1	All Electronics
CREE LEDs (light board)	-	3	Digi-Key; C503 series
120 ohm resistor (light board)	-	1	1/4 watt
SB-400 solder breadboard	-	1	All Electronics
PC-1 perfboard	-	1	All Electronics
Two-pin MTA-100 female connectors	-	3	Digi-Key

(Use of the MTA-100 wire insertion tool when assembling cables is highly recommended.)

PARTS
LIST

wealth of related information. The circuit is designed for up to a 30 mA load, or a single LED series trio.

The Adafruit up converter link in **Table 1** is a place to start if higher current output is needed. With the LEDs chosen, you may find the lights are already on the bright side for night use, though even brighter ones are not much more expensive.

By now, you are wondering the purpose of the test connector near the PWM output stage. I found that during debugging, the PWM output could stick high when I was at a breakpoint in the code. With the line high, Q1 is on and current builds up in the coil. This is a potentially disastrous situation and a big reason for the input fuse on the battery input.

So, when debugging code, I jumper across the test connector which shorts the base of Q1 to ground and disables the output stage. A scope can still be connected at P1.6 to monitor the PWM output line. While jumpering J5 disables the boost converter, it protects the output stage during code tweaks.

We Have Liftoff!

If your destination is the Internet of Things, this

project is much more than just a light controller. My first brush with the MSP product line found me jumping to multiple sites to understand the architecture and master register level peripheral control. (Yes, I am one of *those* guys.) Those who leverage the code and links of this article will have a much shorter liftoff, which brings up a short overview of the LaunchPad family.

Open up your browser to the LaunchPad landing page listed in the Links table. The MSP series is just a start to a wide universe of processing functionality. Next, open up to the LaunchPad workshops page in the Links. Need a platform for a data logger so power-efficient, it can run from a small supercap instead of batteries? Start with the FR5969 Launchpad and tutorial; the out-of-the-box sample code is a simple data logger.

Need an LCD for your next project, a timekeeping clock, and enough I/O for most home automation, all within \$14? The FR4133 `Pad, sample code, and DriverLib API layer has slashed development time for another project. Still want an Arduino-like code development on your way to 500+ family members? Follow the Energia links and information.

Welcome to the Connected Universe. **NV**

SUPERIOR EMBEDDED SOLUTIONS



www.embeddedARM.com

Call (480)837-5200



TS-7250-V2 Single Board Computer

Extensible PC/104 Embedded System with Customizable Features and Industrial Temps

- 800 MHz or 1 GHz Marvell PXA166 ARM CPU
- 512 MB DDR2 RAM
- 2 GB SLC eMMC Flash Storage
- PC/104 Connector
- 8k or 17k LUT FPGA
- Dual Ethernet



Available with
TS-ENC720 enclosure

Starting at
\$165
Qty 100
\$199
Qty 1



PC/104 Peripherals Multi-Function Daughter Cards

Our Line of PC/104 Peripheral Boards Offer Maximum Functionality at Minimum Cost.

- Digital I/O Boards
- Ultra Long Range Wireless
- Software Controlled Relays
- Multi-tech GSM, GPRS, HSPA modem boards
- Intelligent battery back-up
- Much more....

Starting at
\$32
Qty 100
\$30
Qty 1



We've never discontinued a product in 30 years



Embedded systems that are built to endure



Support every step of the way with open source vision



Unique embedded solutions add value for our customers

A MIDI Controlled Analog Synthesizer

By Thomas Henry

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/April2016_MIDI-Analog-Synthesizer.

ON A SHOESTRING



No doubt about it. Synthesizers have become a dominant force in just about all genres of modern music. Early on, they were 100% analog in nature, and hence noted for their warm fat sound. A bit later with the advent of microcontrollers, digital synthesis caught on, offering unparalleled control by means of a MIDI keyboard or sequencer software. It would seem like the best of both worlds is a MIDI controlled analog synthesizer — if only the start-up cost wasn't so intimidating.

Fortunately, we now live in the golden age of DIY! If you've always longed to make electronic music yourself but have fretted about your pocketbook or the complexity, here's the perfect approach. In this article, you'll learn how to arrive at a very usable MIDI controlled analog synthesizer for under \$40. While perhaps not rivaling a professional's Yamaha, Roland, or Casio in all respects, there's no doubt the neophyte can indeed start making serious music with it at once.

The Voice of the Beast

Here's the plan of attack. At the heart of the project is the Gakken SX-150 Mark II (**Figure 1**). Giving the initial appearance of a kid's toy, it really is much more. Indeed, it contains the modules you would expect in a decent analog synthesizer: a voltage-controlled oscillator; voltage-controlled filter; voltage-controlled amplifier; envelope generator; and low frequency oscillator for vibrato and trills. When connected to a good amplifier and speakers, it yields an impressive sound.

When the SX-150 Mark II first came out several years ago, it was fairly costly. However, the price has steadily declined, and I recently picked one up from Amazon for a mere twenty smackers. Incidentally, there was a previous version which also sold like hotcakes, though not nearly as potent sounding. To avoid confusion while shopping, the Mark I is black and the Mark II is red.

While it sounds great, the SX-150's chief defect is its user interface. In particular, the unit was designed to be played by pressing or sweeping a stylus against a resistive strip. That might be okay for weird theremin type effects, but is hardly adequate for handling traditional scaled music. Instead, let's see how to add MIDI control and turn the SX-150 into a legitimate instrument capable of being driven directly by a music keyboard or — even better — put under software sequencer control.

The first design goal, obviously, is to make sure the finished project is musically useful. Then again, the circuit should be reasonably accessible, requiring no exotic components. Most importantly, we'll steer clear of any hardware surgery to the SX-150; the interface needs to be completely non-invasive. All these goals were easily met.

The Voltage Control

That resistive strip mentioned earlier is really little more than an exposed potentiometer strung between two voltage potentials with the stylus acting as the wiper. A bit of investigation revealed the extremes to be at +2.5V and almost +5V. A quick calculation (along with some



■ **FIGURE 1.**

listening) showed that the oscillator responds to a reasonable 0.5V per octave. In other words, if applying +2.5V to the oscillator generates a C note, then applying +3.0V produces the C one octave up from that.

I found that the unit was good for four octaves total — more than reasonable for doing real music. (When I got started with synthesizers in the 1970s, a three-octave keyboard was pretty standard and no one complained).

In addition to providing pitch information, the stylus also directs the envelope generator, telling it when to start and end a sound. Specifically, when the stylus is lifted completely from the strip (representing 0V), any tone goes completely silent. Incidentally, the envelope generator of the SX-150 features variable attack and decay controls on the front panel.

So, what we need to arrange is a circuit to transform digital MIDI messages to reliable analog voltages spanning +2.5V to +4.5V for the tuned notes, and 0V when the note has concluded (the Note Off condition).

Enter MIDI

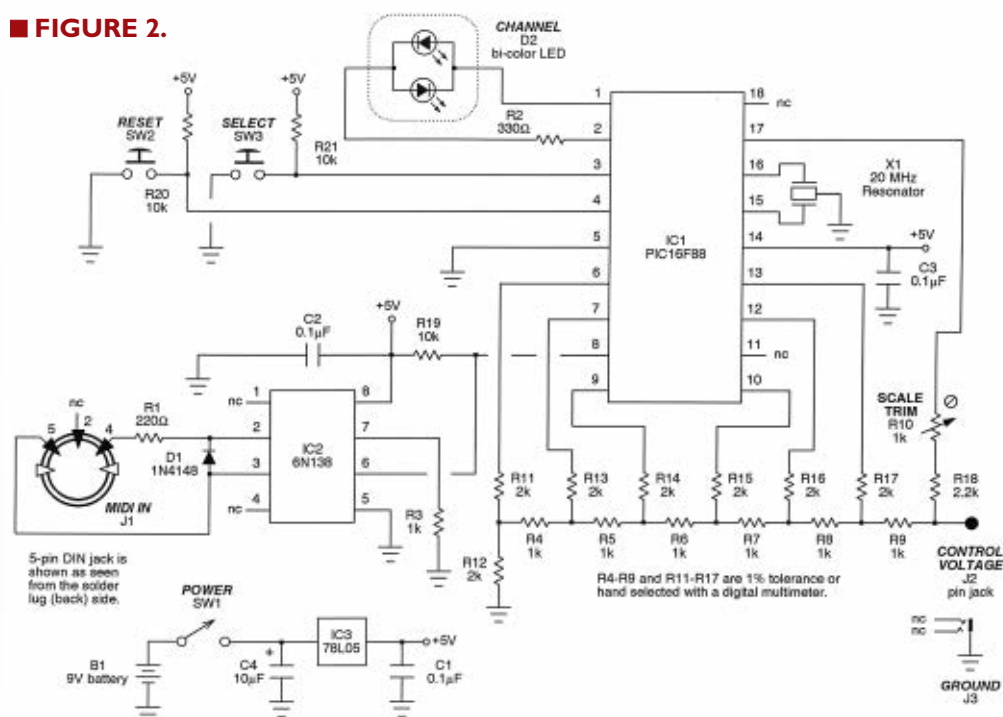
Now that we know what the SX-150 is expecting to see, let's focus on the MIDI requirements. MIDI offers quite a few commands, but we'll only need a handful for a basic functioning synthesizer. In particular, we'll want the unit to respond to:

- MIDI On
- MIDI Off
- All Notes Off
- Omni On
- Omni Off

The first three commands are pretty obvious: Turn a note on, turn it off again, and clear all notes in case something got stuck by accident — sort of a panic button.

Recall that up to 16 MIDI devices can be chained, so

■ FIGURE 2.



All of the software used in the creation of this project is absolutely free of charge — no strings attached. Here are the packages mentioned in the article:

TinyCad (schematic capture):
sourceforge.net/projects/tinycad

FreePCB
 (printed circuit board design):
www.freepcb.com

PIC Micro Pascal (compiler):
www.pmpcomp.fr

The source files for this project are available at the article download link.

each must have a separate channel number to ensure the note data coming down the pike is only acted upon by the desired instrument. Indeed, the interface circuit (to be

sometimes in simple situations you might just want the unit to respond to any note data coming in, completely ignoring the channel business. Turning Omni On gives this. Turning it off again resumes channel-specific behavior.

described in just a moment) lets you customize the channel number of the SX-150 so it will play nicely with whatever else you have in your studio.

On the other hand,

ITEM DESCRIPTION

Resistors

R1	220Ω
R2	330Ω
R3-R9	1K
R10	1K trimmer
R11-R17	2K
R18	2.2K
R19-R21	10K

Capacitors (10V or better)

C1-C3	0.1 μF disc
C4	10 μF electrolytic

Semiconductors

D1	1N4148
D2	Bi-color LED
IC1	PIC16F88 microcontroller
IC2	6N138 opto-isolator
IC3	78L05 voltage regulator
X1	20 MHz ceramic resonator

Switches and Jacks

SW1	SPST toggle switch
SW2, SW3	SPST pushbutton
J1	Five-pin DIN jack
J2	Pin jack
J3	3.5 mm mini phone jack

Miscellaneous: 9V battery and battery snap, battery holder, LED holder, enclosure, wire, solder, rubber feet, etc.

Note: R4-R9 and R11-R17 are 1% tolerance, or can be hand-selected using a digital multimeter.

PARTS LIST

MIDI-to-Analog Made Easy

Let's tuck into the circuit details now. Refer to **Figure 2**, which shows the complete schematic for the SX-150 MIDI interface circuit. Running the show is the ever popular PIC16F88 microcontroller. This is just a great chip: versatile, yet so inexpensive (less than two bucks) and a delight to work with.

The PIC16F88 is clocked by a 20 MHz ceramic resonator which reliably handles the peppy 31.25 kilobaud serial communication required by MIDI. You'll find the resonator on pins 15 and 16. Power to the unit is provided by a 9V battery, stepped down and regulated by the 78L05.

Two pushbutton switches and a bicolor LED handle the channel setting business. While keeping front panel real estate to a minimum, you'll find it all quite easy to use. We'll see how in just a moment.

The MIDI hardware interface is the traditional current loop affair dictated by that protocol. IC2 (the 6N138 opto-isolator) protects the circuit from the outside world, and then drives the serial workings conveniently contained within the PIC16F88.

Finally, we come to the analog side of things. After

trying and rejecting two other approaches for the digital-to-analog conversion (DAC; one too complicated, the other too imprecise), I settled on the trusty old R-2R network and it took off like a champ. Resistors R4 through R9 and R11 through R17 comprise the network.

The resistors just mentioned must be fairly well matched for accurate tuning. I hand selected ordinary 5% values for R11 through R17. After going through several dozen, I found seven that were dead-on with my multimeter. Since I had them handy, I used 1% values for R4 through R9, but still confirmed each with the multimeter as a precaution. Considering that resistors go for two cents apiece, this ends up as a very usable converter weighing in at less than the price of a candy bar!

Pins 6, 7, 9, 10, 12, and 13 of the PIC16F88 drive the DAC. Six bits yield a number from zero to 63. Observe that four octaves mean only 48 levels are actually required (12 notes to an octave multiplied by four), so our six-bit DAC is more than adequate. The unused notes at the low end will become duds indicating a Note Off condition – something below +2.5V. Finally, trimmer R10 lets you tweak the tuning of the interface.

Let's Build It

There is nothing terribly critical about the circuit, so perfboard construction is perfectly acceptable. You should be fine if you are neat, keep the ceramic resonator close to the microcontroller, and don't forget the various decoupling capacitors sprinkled about the circuit.

A printed circuit board (PCB) is my favorite approach. I used the great schematic capture program TinyCad to create the netlist, and then imported that to the equally great FreePCB printed circuit board artwork software. Note that both of these programs are free of charge and readily available on the Web; see the **sidebar** for details.

I concluded this stage by etching my own board in the kitchen using a homemade Tupperware tank and aquarium bubbler.

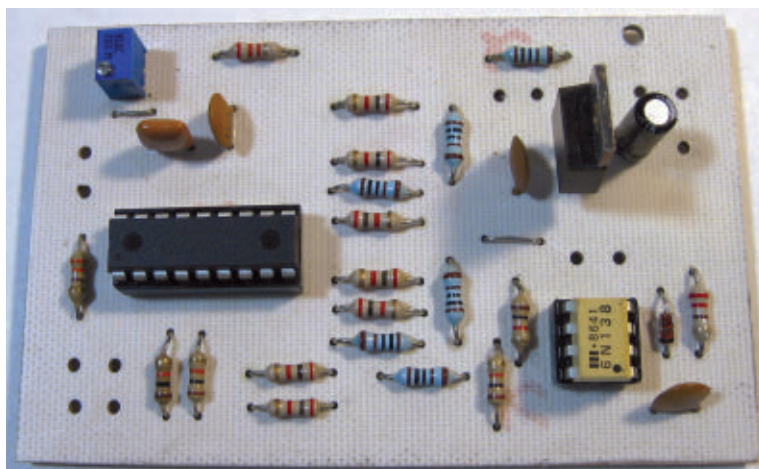
You can see the loaded circuit board in **Figure 3**, while **Figure 4** shows everything about to be sealed up. Eagle eyes will note I used the beefier 7805 since that's all I had on hand at the time. However, the daintier 78L05 specified is more than adequate. **Figure 5** shows the completed unit, ready to be used.

The Firmware

The firmware for the project was generated with PIC Micro Pascal – surely one of the most powerful and easy-to-use compilers around. It too is completely free of charge. Again, refer to the **sidebar**.

The source code is heavily commented should you

■ **FIGURE 3.**



wish to customize anything. In addition, the code provides program stubs for **all** MIDI commands in case you desire to add even more features to your implementation.

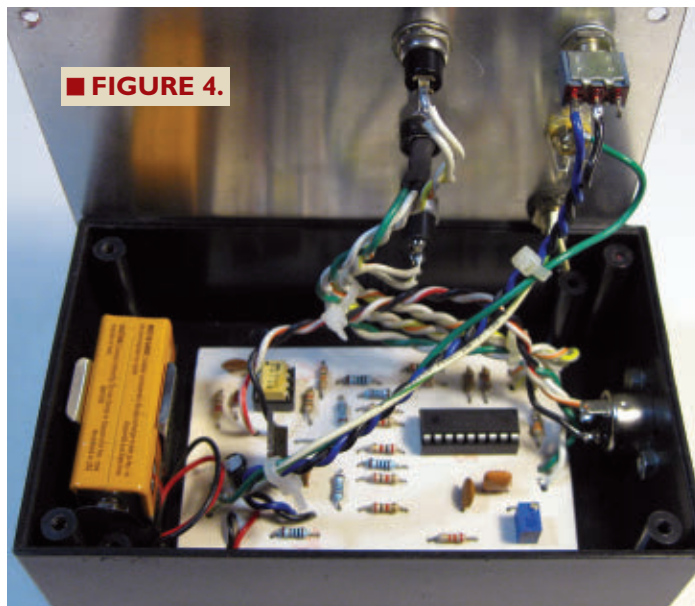
On the other hand, if you simply want to start playing music with no further ado, just send the compiled hex file to your PIC chip and have at it. The download package at the article link contains both source and hex files for your convenience, along with all the PCB files.

From a PIC programming point of view, there are a number of interesting things going on in the firmware. So, indeed, this could also serve as a great educational project. For example, you'll find two types of interrupts being handled, hardware serial communication, a circular queue implemented with an array to buffer the possibly rapid-fire MIDI bytes coming in, and more.

Get Set ...

Do you remember my mentioning that one of the

■ **FIGURE 4.**



■ FIGURE 5.



■ FIGURE 6.

design goals was to leave the Gakken SX-150 Mark II completely unmolested by hardware hacking? This is rather neatly handled. It just so happens that the stylus tip is the same size as a standard pin plug. Thus, a mating pin jack is used to accommodate the stylus. (Remember, that's where the control voltage comes from).

If you check out the SX-150 front panel, you'll note that it sports an "audio in" jack which really is of very little use in most circumstances. Instead, let's employ it to supply ground to the interface. (The tip and ring are just ignored).

Simply insert the stylus into its pin jack, patch in a mini phone cable to supply ground, and away you go.

Figure 6 shows the SX-150 connected and ready to be controlled. Conclude with a MIDI cable, connecting the affair to either a MIDI keyboard or to a MIDI interface on your PC running sequencer software.

As mentioned, you have the freedom to set the SX-150 to respond to any of the 16 MIDI channels. Here's how to do it:

1. Press and hold the Reset button.
2. Press and hold the Select button.
3. Release the Reset button.
4. The bicolor LED now begins blinking in red.

Continue holding the Select button while you count the blinks, and then release it when you reach the desired channel number.

5. Or, after 16 blinks (the highest channel number), the LED turns green indicating Omni mode is on.

As a further nicety, the currently selected channel (or

Omni mode) is stored in the PIC's EEPROM memory. So, when you next fire the unit up, it will resume in the mode you selected last time.

If you should forget what channel you've selected later on, just press the Reset button briefly and the LED will blink the number out for you as confirmation.

... And Play

Like most musical instruments, the MIDI controlled SX-150 needs to be tuned before creating standard music. While playing octaves back and forth, adjust trimmer R10 until you hear perfect intervals. Also, the SX-150 has a trimmer accessible from the underside of its enclosure which lets you set the range — from exceedingly low bass notes on up to ear-splitting treble ones. You will probably find that you need to go around the horn a few times, tweaking both the interface and the SX-150 trimmers until you get a standard tuned scale.

One final thing: The SX-150 does not contain any temperature compensation. So, if your gig takes you from the great outdoors in the summer to an air conditioned room, you might find that you'll have to readjust the tuning a bit.

Join the club! Stringed instrument players are completely accustomed to touching up the tuning even during the course of a performance. It's just part of the music making process.

If you've ever yearned to see what electronic music is all about, the Gakken SX-150 Mark II along with this easy-to-build MIDI interface makes a great first instrument. May your songs always be tuneful! **NV**

ELECTRONET

ALL ELECTRONICS CORPORATION
Electronic Parts and Supplies.
www.allelectronics.com
Free 96 page catalog 1-800-826-5432

For the ElectroNet online, go to
www.nutsvolts.com
click **Electro-Net**

USB Add USB to your next project-- It's easier than you might think!
USB-FIFO • USB-UART • USB/Microcontroller Boards
RFID Readers • Design/Manufacturing Services Available
Absolutely NO driver software development required!
www.dlpdesign.com

INVEST in your BOT!



HITEC

12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecusa.com

PCB, PCBA and More! **Myro** Low cost High Quality
www.myropcb.com
1-888-PCB-MYRO

Ironwood ELECTRONICS GHz Bandwidth Sockets for BGA & QFN
Industry's Smallest Footprint



www.ironwoodelectronics.com

MaxBotix
High Performance Ultrasonic Rangefinders



www.maxbotix.com
info@maxbotix.com Phone: 218-454-0766

AGENT 390
TRACKED ROBOT KIT
\$329.99 @ ServoCity.com



Wanna read even more about electronics?
Check out our book selection at
store.nutsvolts.com

www.servomagazine.com

MOBILE APP NOW AVAILABLE!

SERVO

Download NOW On Your Favorite Mobile Device!

FOR ROBOT BUILDERS

iOS • ANDROID • KINDLE FIRE



Did You Know Preferred Subscribers get access to all the digital back issues of **Nuts & Volts** for free?

Call for details
1-877-525-2539

Did you know that **Nuts & Volts** now has a weekly content newsletter? **Want to get it?**

You have three ways to sign up:

- Visit us on Facebook and click on "Join My List."
- Using your cell phone, send "NVNEWSLETTER" as a text message to 22828.
- Visit the **Nuts & Volts** FAQs to sign up at nutsvolts.com/faqs

The Beginner's Guide to Digital Electronics

LEARN HOW THE NUMBERS '1' AND '0' CHANGED THE WORLD

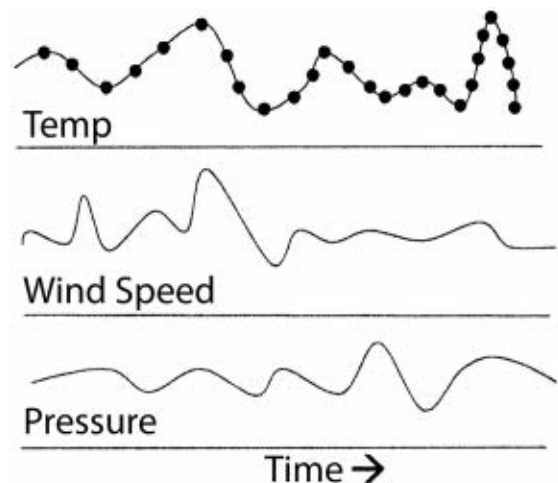
This article was written specifically for the newcomer to the field of digital electronics. If you've always wanted to know how the digital world works, then keep reading. You don't need to know calculus, algebra, or any complex formulas to finish this article. The only requirements are an interest in digital electronics and a desire to learn. Since you're reading this paragraph, obviously you're at least a bit curious about the digital world. Fortunately, curiosity is half the battle to enlightenment.

Analog vs. Digital

The world of electronics is a lot easier to understand if we start by dividing it into two distinct categories: the "analog" world and the "digital" world. The analog world generally refers to any natural phenomenon that varies its own properties over a period of time. Take the outside temperature, for example. We notice that it changes rather slowly throughout the day, and at any instant we can measure how hot or cold it really is by using a simple thermometer.

The same changing properties can be observed, measured, and recorded in other natural phenomenon such as barometric pressure, wind speed, solar radiation, etc. If you were to record and graph each of the above events over a 24 hour period, you would notice one similar characteristic: the physical properties of each phenomenon change over time.

In each case, if you connect all the dots on a graph, the results will always form some type of "analog signal" (wave) like these examples:



It's the job of the **analog** electronics engineer to deal with many of these thermal, magnetic, optical, acoustical, biological, chemical, or electrical "signals" by designing the appropriate analog sensors and control circuitry.

The Digital World

Digital electronics, on the other hand, are a completely different type of animal altogether. How is the digital world different from the analog world? Well, in the digital realm (i.e., digital electronic circuits), there are only two “states” that are important: ON or OFF. For example, when you flip the light switch on in your bathroom, you know there are only two possible positions the light switch can be in (yep, ON or OFF). It’s of no real concern to you that there could be 110, 113, 120, or 125 volts (i.e., fluctuating **analog** “signal”) running through the electrical wiring connected to the light switch.

The important thing to remember is switching the light on not only performed some useful work (light up the bathroom), but it also conveyed some real basic digital information as well – that the light was ON and not OFF. This ON/OFF code is exactly how the world of digital electronics operates. As you will see later, this ON/OFF switching is the same logic used to build the digital electronic circuits inside your laptop computer, GPS device, and smartphone. Take a look at **Figure 1** so the difference between analog and digital worlds is clear in your mind.

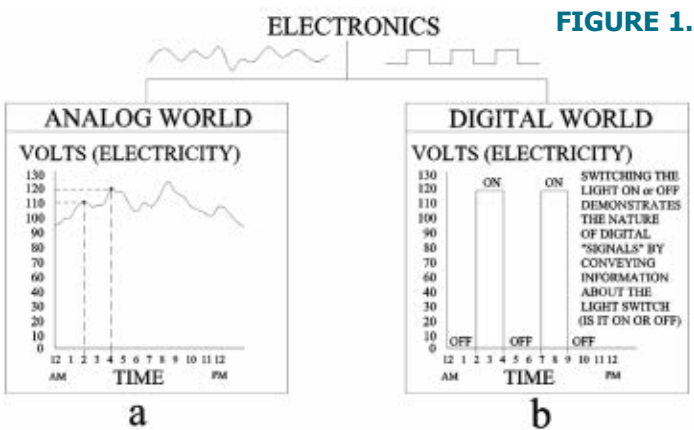
Figure 1a is a graph showing how electricity (voltage level) in your home changes over the course of 12 hours. At one point during the 12 hours, we can see that the voltage level at 2:00 pm was 110 volts. At 4:00 pm, it changed to 120 volts. In **Figure 1b**, on the other hand, we have a light switch that can be turned ON or OFF during the day. Notice that in **Figure 1b**, the light switch was turned ON at 2:00 pm and then turned OFF at 4:00 pm. Then, again it was turned ON at 7:00 pm and then turned OFF at 9:00 pm.

In this example, it doesn’t matter much what voltage level is present throughout the electrical circuit (105V, 110V, 115V, 120V). What is important here is the “information” conveyed in **Figure 1b** – whether or not the switch was ON or OFF. This “instantaneous” ON/OFF electrical switching is commonly referred to as a “digital” signal.

[Author’s note: Not to confuse you, but – and this is very important – digital electronic circuits run on 1.5-15 volts of Direct Current (DC) – not the 110 volts Alternating Current (AC) that comes out of the wall outlet in your home. The AC voltage level in **Figure 1b** was only used to show how a digital signal looks (square wave) when compared to an analog signal (i.e., the fluctuating wave in **Figure 1a**).]

Digital — On or Off

To see how a light switch can transmit digital information, let’s suppose that you told a friend on the phone that if she happens to drive by your house later and



PORCH LIGHT #2	BEDROOM LIGHT #1	MESSAGE
OFF	OFF	PARENTS STILL HOME
OFF	ON	STOP OVER
ON	OFF	MEET AT PIZZA SHOP
ON	ON	MEET AT SUE’S HOUSE

	GARAGE LIGHT #4	LIVINGROOM LIGHT #3	PORCH LIGHT #2	BEDROOM LIGHT #1	MESSAGE
0	OFF	OFF	OFF	OFF	PARENTS STILL HOME
1	OFF	OFF	OFF	ON	STOP OVER
2	OFF	OFF	ON	OFF	MEET AT PIZZA SHOP
3	OFF	OFF	ON	ON	MEET AT SUE’S HOUSE
4	OFF	ON	OFF	OFF	MEET AT JANE’S HOUSE
5	OFF	ON	OFF	ON	MEET AT JOE’S HOUSE
6	OFF	ON	ON	OFF	MEET AT BOB’S HOUSE
7	OFF	ON	ON	ON	ETC.
8	ON	OFF	OFF	OFF	
9	ON	OFF	OFF	ON	
10	ON	OFF	ON	OFF	
11	ON	OFF	ON	ON	
12	ON	ON	OFF	OFF	
13	ON	ON	OFF	ON	
14	ON	ON	ON	OFF	
15	ON	ON	ON	ON	

notices that your porch light is OFF, you’ve got company – don’t come over. However, if the porch light is ON, she should drop in. As you can see, this little “code” allows you to use one light to convey two messages. Now, let’s take this idea a little further and use two lights.

If you tell your friend when she sees the bedroom light OFF and the porch light OFF, it means that your parents are still at home – so, again, don’t come over. However, if the bedroom light is ON and the porch light is OFF – stop over. Also, if she sees that the bedroom light is OFF and the porch light is ON, it means you’ll meet her at the pizza shop.

Then finally, if the bedroom light is ON and the porch light is ON, you’ll meet her later at Sue’s house. Notice that this time, with only ‘two’ lights (switched ON or OFF), you transmitted ‘four’ messages (see **Figure 2**).

The question here is: How many messages can you signal to your friend if you used four lights instead of two?

Check out **Figure 3** for the answer.

Since there are four lights and only two “states” or positions the light switch can be in (ON or OFF), there’s a maximum of 16 messages (2 to the 4th power or $2^4 = 2 \times 2 \times 2 \times 2$) that you can relay to your friend. Obviously, with 16 messages, you and your friend would need a list of instructions to decode each sequence of lights.

Notice in our previous example how information was conveyed (transmitted) by using a simple light switch that was limited to only two positions or “logic states” (ON or OFF). Once again, it didn’t matter how much electricity was flowing in the porch, bedroom, garage, or living room light circuits.

The two most important factors to remember here are whether or not the lights were ON or OFF, and in what sequence or **position** the lights were in.

As you can see, digital electronics are based on a “switching logic” (ON or OFF). On the other hand, analog electronics are more concerned with a *fluctuating* (constantly changing) electrical quantity like voltage and/or current. If necessary, refer to **Figure 1** again so the difference between digital and analog signals is clear in your mind. This is a very important concept to understand, so don’t continue reading until you make the distinction between analog and digital information/signals.

Binary System

In our previous examples, we saw how the digital electronics world operates by limiting itself to two states. The binary (bi = two) numbering system also deals with two states, or numbers: 1 and 0. As you will see, binary numbers are very important and useful in the realm of digital electronics.

Now, suppose we take our previous four light example (**Figure 3**) and instead of using an ON or OFF code, we substitute a ‘1’ for ON and a ‘0’ for OFF. **Figure 4** shows how our message chart looks after this change.

It may appear to you that the ON and OFF arrangement in **Figure 3** and the 1 and 0 layout in **Figure 4** were picked at random. The fact is that the ones and

	GARAGE LIGHT #4	LIVINGROOM LIGHT #3	PORCH LIGHT #2	BEDROOM LIGHT #1	MESSAGE
0	0	0	0	0	PARENTS STILL HOME
1	0	0	0	1	STOP OVER
2	0	0	1	0	MEET AT PIZZA SHOP
3	0	0	1	1	MEET AT SUE'S HOUSE
4	0	1	0	0	MEET AT JANE'S HOUSE
5	0	1	0	1	MEET AT JOE'S HOUSE
6	0	1	1	0	MEET AT BOB'S HOUSE
7	0	1	1	1	ETC.
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	

FIGURE 4.

zeros you see in **Figure 4** just happen to be the way you count in ‘binary.’ (Remember, the binary numbering system is based on the number ‘two:’ $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, etc., and our decimal numbering system is based on the number ‘ten:’ $10^0 = 1$, $10^1 = 10$, $10^2 = 100$, $10^3 = 1000$, $10^4 = 10,000$, etc.)

BINARY NUMBER SYSTEM (BASE 2)

$2^8 \quad 2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$

256 128 64 32 16 8 4 2 1

EQUIVALENT DECIMAL NUMBER

Here’s how the binary number system works:

For example, let’s convert the binary number 1101 from the list in **Figure 4** to its decimal (base 10) equivalent:

HERE’S OUR BINARY NUMBER SYSTEM (BASE 2) → $2^3 \quad 2^2 \quad 2^1 \quad 2^0$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 FOUR-BIT BINARY NUMBER SYSTEM TAKEN FROM FIGURE 4 → $1 \quad 1 \quad 0 \quad 1$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 DECIMAL NUMBER VALUES → $8 \quad 4 \quad 2 \quad 1$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 ONLY ADDING IF THERE IS A ‘1’ IN BINARY 1101 → $8 \quad 4 \quad 0 \quad 1 = 13$

In this example, the binary number 1101 is equal to the decimal number 13.

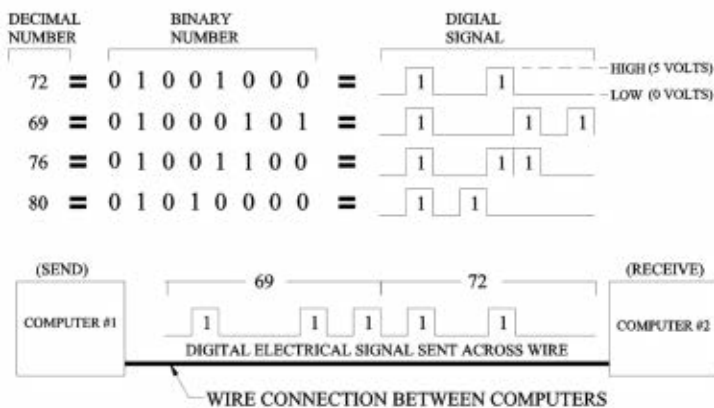
There are four key points to learn from this example:

- (1) Notice how the ‘decimal’ numbers **1, 2, 4, and 8** ‘double’ in value for each increase in power in the binary number system (i.e., Binary $2^0 =$ Decimal **1**; Binary $2^1 =$ Decimal **2**; Binary $2^2 =$ Decimal **4**; Binary $2^3 =$ Decimal **8**, etc.).
- (2) Notice that each binary number 1 or 0 holds a “position” or “weight” in the binary numbering system. In other words, in the digital world, 0 is just as important as 1.
- (3) Notice that each position in the binary numbering system equals an exact decimal number value (1, 2, 4, 8, 16, 32, 64, etc.).
- (4) Notice that you only add the decimal number values together whenever there is a 1 in the binary number.

Let’s convert one more binary number into a decimal number before moving on. Convert **1010110** to a decimal number:

BINARY SYSTEM → $2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 SEVEN-BIT BINARY NUMBER → $1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 DECIMAL VALUES → $64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 $64 \quad 0 \quad 16 \quad 0 \quad 4 \quad 2 \quad 0 = 86$

FIGURE 5.



Once again, notice that although each zero in the binary number 1010110 holds a place and is very important, its decimal value is ignored when we add all the decimal numbers together (total = 86).

These examples should give you an idea as to how the list of binary numbers in **Figure 4** was created. Originally, we used ON and OFF light switches to transmit coded messages (**Figure 3**). Now, instead of messages, we can convert binary numbers into their decimal equivalent.

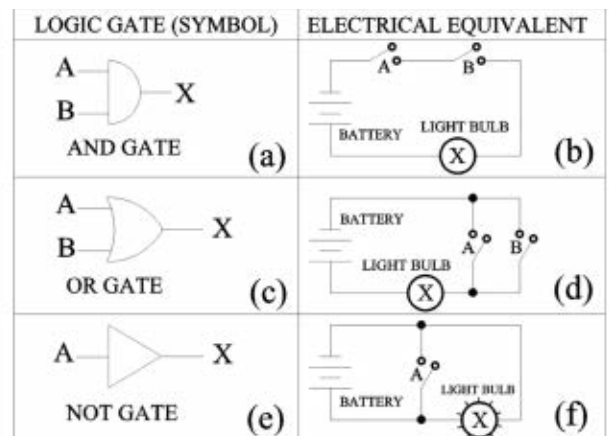
If you're asking yourself what this has got to do with digital electronics, you've asked a very important question. If you remember what we said about digital logic (that it is a switching logic — ON or OFF), you can probably see that transmitting these binary ones and zeros over an electrical wire is just a matter of "switching" an electrical voltage ON and OFF (five volts = 1, zero volts = 0). Remember, you can only do two things with electricity: you can switch it ON or you can switch it OFF.

For example, let's say we wanted to send the decimal numbers 72, 69, 76, and 80 from a computer in one office to a computer in another office. All we need to do is change each decimal number 72, 69, 76, and 80 into a binary number, convert these binary ones and zeros into an electrical signal (five volts = 1, zero volts = 0), and then send this digital code over an electrical wire (see **Figure 5**).

If you study **Figure 5** for a while, the answer to your original question (What has the binary numbering system got to do with digital electronics?) will become apparent. **Figure 5** shows how decimal numbers 72, 69, 76, and 80 and their binary equivalent are transformed by the computer into a digital signal (zero volts and five volts) and then are transmitted over an electrical wire. The digital electronic circuitry inside computer #2 converts the voltage levels (zero volts and five volts) into binary ones and zeros, and then displays (LCD monitor) that information in alphanumeric characters so we can understand the original message.

As you might have suspected, the numbers 72, 69, 76, and 80 in the example above were not picked out of the air. If you look at any American Standard Code for

FIGURE 6.



Information Interchange (ASCII) table, you'll see that the number 72 = H, 69 = E, 76 = L, and 80 = P is a code for the word HELP.

What's important about this example is the fact that we can transform the word HELP into decimal numbers, then into a binary code of ones and zeros, and finally into digital voltage levels that can travel over an electrical wire. Just think about it. This all started by flipping a bathroom light switch ON and OFF!

Now that you know the digital world is controlled by binary numbers (1 or 0), it's time to use this binary system to build some digital logic gates. You'll really see how important binary numbers are to digital circuits after reading the next section.

Digital Logic Gates: The Building Blocks of All Digital Electronic Circuits

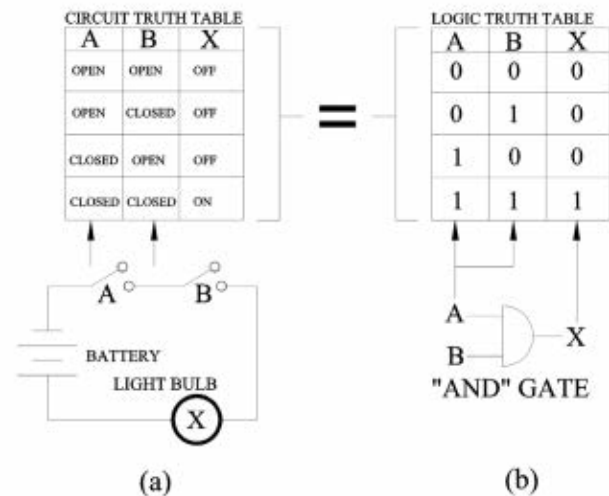
Understanding digital logic gates is a major prerequisite to learning how all digital electronic circuits work. Basically, there are three types of electronic "logic gates:" the AND gate, the OR gate, and the NOT gate. (Actually, there are a few more types of gates, which we will cover later in this article.) Just remember, logic gates are the main building blocks of all digital logic circuits.

The AND Gate

Let's analyze the AND gate in **Figure 6** first. Take a look at the logic symbol for an AND gate in **Figure 6a**.

It has two inputs (A and B) and one output (X). The AND gate operates (logically) in the following way: If there is a binary 1 on input A and a binary 1 on input B, the binary output (X) will equal 1. If input A has a 1 and input B has a 0, output (X) equals a 0. In turn, if input B has a 1 and input A has a 0, then the output (X) will be a 0. Lastly, if input A has a 0 and input B has a 0, then output (X) will also be a 0.

FIGURE 7.



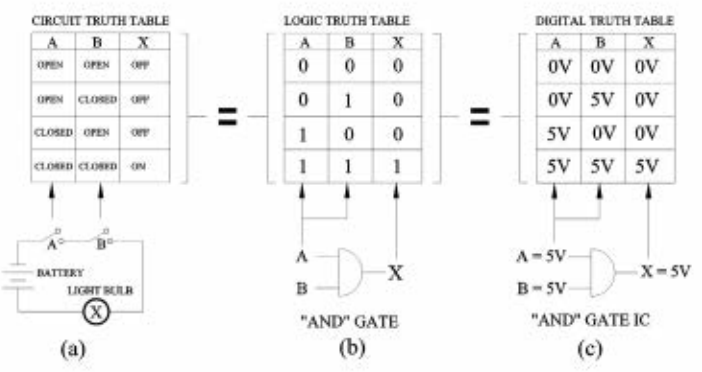
Okay now, let's build an electrical circuit that mimics this AND gate logic. Take a look at the electrical circuit in **Figure 6b**. We have a nine volt battery, a light bulb, and two switches. Notice that if we close switch A, the light will NOT turn "ON" because switch B is open (i.e., the electric light is not connected to the battery).

If switch B is closed and switch A is open, the light remains OFF. Notice that the only time the light will turn ON is when switch A 'AND' switch B are closed.

Now, look at **Figure 7b** and you should see that our old friend the AND logic gate operates in the same way as our electrical light circuit in **Figure 7a**. You can see this very important point by looking at the two logic tables in **Figure 7**.

Notice in **Figure 7b** that the ones (1) in the logic truth table match (same position) the "closed" switches in the circuit truth table of **Figure 7a**.

FIGURE 8.



It's important that you understand how the AND gate logic symbol in **Figure 7b** and the electrical circuit in **Figure 7a** are related. Notice in **Figure 7b** how the binary number system comes into play here. The only time output (X) equals a 1 is when both A and B inputs are also a 1. The only time the light turns on in **Figure 7a** is when both A and B switches are closed.

Okay, the question here is what happens if we change the ones and zeros in **Figure 7** into digital electronic signals (1 = five volts, 0 = zero volts)? Look at **Figure 8** and you'll get the answer to this question.

Looking at **Figure 8c**, you can see that although we changed the binary ones and zeros to five volts and zero volts, the truth tables are all the same. This means that the electrical circuit with two switches A and B (**Figure 8a**) operates like the AND gate logic truth table in **Figure 8b** and **Figure 8c**.

You're probably asking yourself, "Do I need all these truth tables, AND gates, and 1 and 0 logic to understand digital electronics?" The answer to that question is YES! The entire digital world is based on knowing what the input and output values will be for any digital logic circuit.

For example, look at the 7400 integrated circuit (IC) in **Figure 9**. You can purchase this IC on the Internet and verify that it works exactly like the truth tables we've constructed.

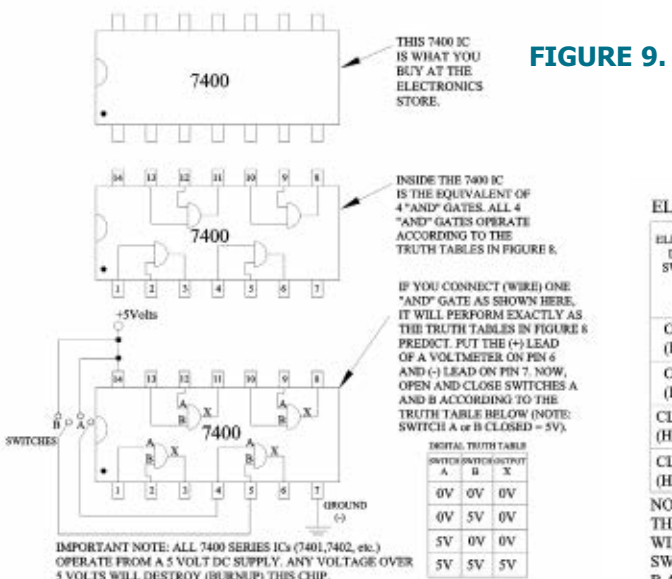


FIGURE 9.

ELEVATOR TRUTH TABLE

ELEVATOR DOOR SWITCH A	ELEVATOR DOOR SWITCH B	AND GATE OUTPUT X
OPEN (LOW)	OPEN (LOW)	STOP MOTOR
OPEN (LOW)	CLOSED (HIGH)	STOP MOTOR
CLOSED (HIGH)	OPEN (LOW)	STOP MOTOR
CLOSED (HIGH)	CLOSED (HIGH)	START MOTOR

NOTE THAT THE ONLY TIME THAT THE ELEVATOR MOTOR WILL TURN ON IS WHEN DOOR SWITCH A AND DOOR SWITCH B ARE BOTH CLOSED.

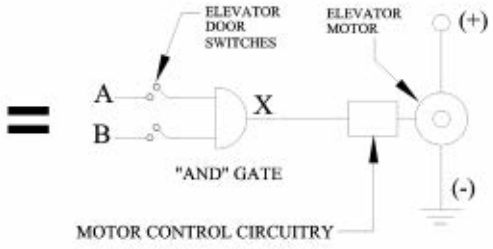


FIGURE 10.

Built into the 7400 IC is the same AND gate we've been talking about in **Figure 7** and **Figure 8**.

If you connect the 7400 IC as shown in **Figure 9**, you can prove to yourself that the five volt and zero volt truth table in **Figure 8c** are correct. More importantly, though, you'll see how digital logic gates and truth tables are used to create digital electronic circuits.

Now that you understand how an AND gate works, let's use this electronic AND gate in a real world application. Let's say we have an elevator with two doors that absolutely must be closed before the elevator starts to move up or down. Each door has an electrical switch (A and B) that is connected to an AND gate.

The output (X) of the AND gate is connected to an electric motor that moves the elevator up or down.

Our job is to make sure that the only time the elevator is allowed to move is when both doors are completely closed. Let's set up a truth table for what we want the elevator to do (see **Figure 10**). Notice how we created the table in **Figure 10**. We determined how the output (X) (motor ON or motor OFF) should respond according to the position of each door (OPENED or CLOSED).

Also notice (**Figure 11**) that the elevator truth table is the same as the AND gate truth table and our logic (binary) truth table.

It should be apparent by now that the AND gate and its truth table can be used to design and build digital electronic circuits that have real world practical applications.

Now, let's look at our next gate: the "OR" gate.

The OR Gate

As mentioned earlier, the AND gate is one of the three basic logic gates (AND, OR, NOT). Let's see how the OR gate differs from the AND gate.

Looking at the electrical circuit in **Figure 12**, you'll notice that two switches (A, B) are connected in parallel. Notice that if switch A is CLOSED, it will complete the electrical circuit and light (X) will turn ON.

Now, if switch A is OPEN and switch B is CLOSED, the light will also turn ON. If both switches are CLOSED, the light is also ON. Notice that as long as one switch (A OR B) is CLOSED, the light will turn ON.

The only time the light turns OFF is when both switches are OPEN. Study **Figure 12** until you see how the OR Gate

FIGURE 11.

ELEVATOR TRUTH TABLE			AND GATE TRUTH TABLE			BINARY TRUTH TABLE		
ELEVATOR SWITCH A	ELEVATOR SWITCH B	ELEVATOR MOTOR X	INPUT A	INPUT B	OUTPUT X	A	B	X
OPEN	OPEN	OFF	0V	0V	0V	0	0	0
OPEN	CLOSED	OFF	0V	5V	0V	0	1	0
CLOSED	OPEN	OFF	5V	0V	0V	1	0	0
CLOSED	CLOSED	ON	5V	5V	5V	1	1	1

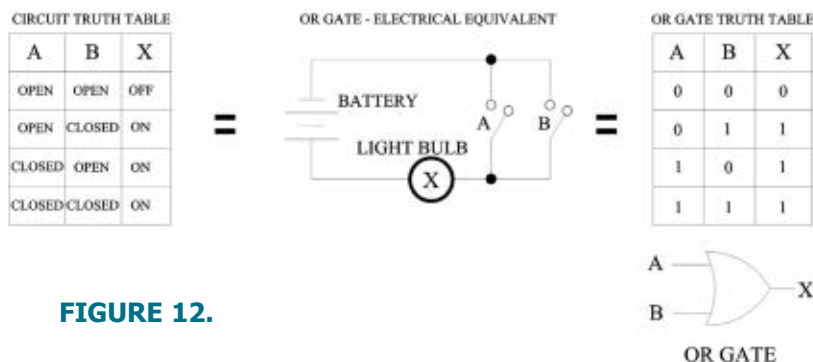
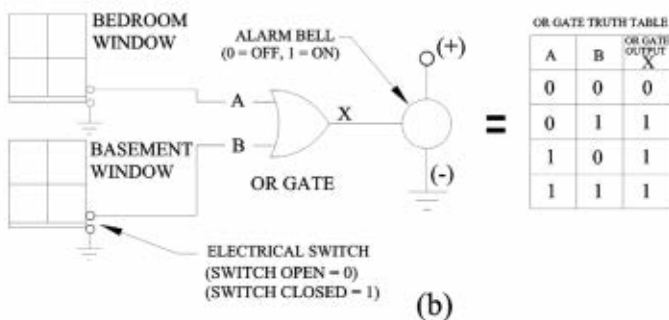
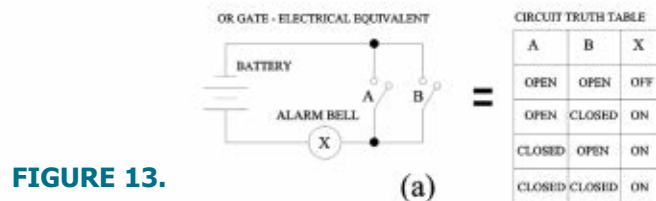


FIGURE 12.



Truth Table matches the Electrical Circuit Truth Table.

Let's take a look at a practical application for the OR gate. An OR gate circuit can be used to protect a house from a burglar. **Figure 13** shows a simple alarm system based on the OR gate.

It's easy to see in **Figure 13b** that if the bedroom switch (A) is CLOSED (window is open) "OR" the basement window switch (B) is closed, the alarm will sound. Notice in the OR gate truth table (**Figure 13b**) that switch A OR B can turn the alarm ON.

As you can see, the OR gate logic truth table can be used to design a burglar alarm system just as the AND gate logic truth table was used to start and stop an

FIGURE 14.

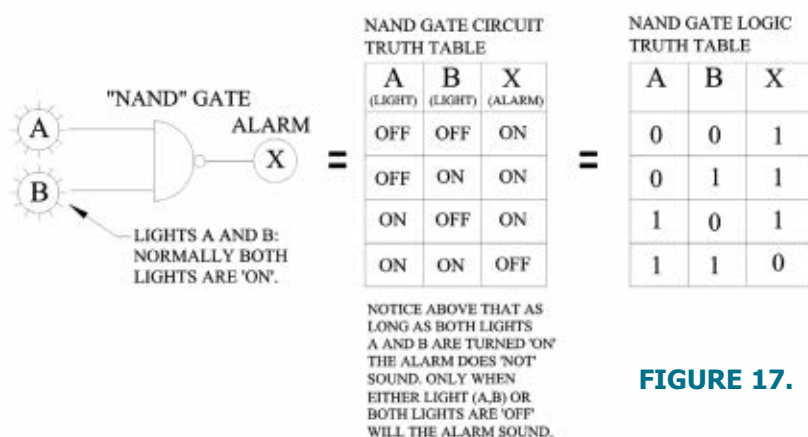
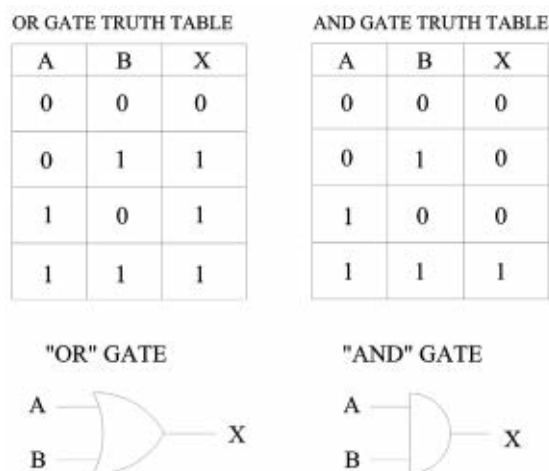


FIGURE 17.

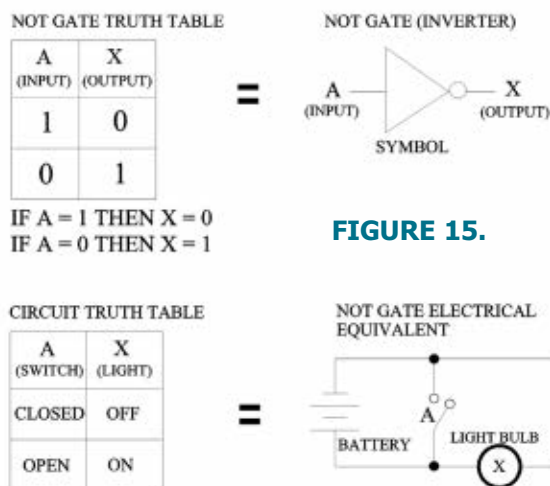


FIGURE 15.

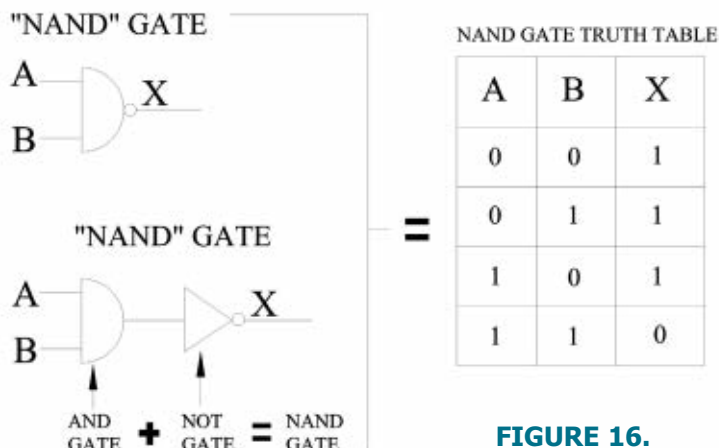


FIGURE 16.

elevator motor.

Take a look now at **Figure 14** so you can see the difference between an AND gate and an OR gate.

Notice the major difference between the OR gate and the AND gate. The OR gate will output a one (1) when either input A OR B is a 1, whereas the AND gate will only output a 1 when both inputs A AND B are a 1.

The NOT Gate (Inverter)

Let's turn our attention to the last of the three basic gates: the NOT gate (see **Figure 15**).

The NOT gate is commonly called an "inverter." It has one input (A) and one output (X). If the input to a NOT gate (A) is a 1, the output will be a 0. Conversely, if the input to a NOT gate (A) is a 0, then the output will be a 1. You can see now why they call the NOT gate an "inverter."

To get a better understanding of how the NOT gate works, look at the electrical circuit in **Figure 15**. Here, if switch A is CLOSED, the electrical current from the

battery will bypass the light; consequently, the light will not have enough current to turn ON.

However, if the switch (A) is OPEN, the light will remain ON because the electrical current bypasses the switch and travels directly to the light. The NOT gate is used extensively in digital logic circuits.

Next, you'll see how it can change the AND gate and the OR gate into NAND and NOR gates.

Inventory

It's time to take inventory of just how much we've learned so far. You might not have realized it, but some very important concepts were tackled. You started with a simple bathroom wall switch that turned a light ON or OFF. You transformed these ON and OFF digital signals into a binary code of 1 and 0. You learned about AND, OR, and NOT logic gates and their associated truth tables.

Finally, you converted all those ones and zeros into electrical circuits with real world applications. Good work!

The NAND, NOR, EX-OR, and EX-NOR Gates

In the previous section, I mentioned how important the NOT gate was in digital electronic circuits. Now, you'll see how the NOT gate is used to create both NAND and NOR gates. In addition, we'll cover two other special gates used throughout the digital electronic world: EX-OR and EX-NOR gates.

The NAND Gate

Let's start with the NAND gate. The NAND (Not AND) gate is basically an AND gate with a NOT gate (inverter) connected to its output (see **Figure 16**).

Notice the circle at the output (X) of the NAND gate in **Figure 16**. This circle is the standard symbol in digital electronics to indicate inversion (NOT = INVERTER). It is equal in logic to the NOT gate with its triangle and circle symbols.

So, from now on, anytime you see a circle on the input or output of an gate, it means that you should invert the signal or logic (1 to 0, 0 to 1, zero volts to five volts, five volts to zero volts, HIGH to LOW, LOW to HIGH).

As you can see in **Figure 16**, the truth table shows that the only time the output of a NAND gate will be 0 (or LOW) is when both inputs A and B are a 1 (HIGH). Also notice in the truth table that any 0 (LOW) on input A or B will make the output (X) a 1. In other words, any 0 (LOW) on the input of a NAND gate will produce a 1 (HIGH) at the output (X).

Like the AND and OR gates discussed previously, let's use the NAND gate in a practical real world application. Suppose we build a NAND gate electrical circuit that monitors two outdoor security lights so that if either light goes out (OFF), an alarm sounds. Once again, the first thing we do is set up a truth table for our light circuit (see **Figure 17**).

Notice again in the truth table, that if 'either' light A or light B turns OFF, the alarm will turn ON. You should also see that the alarm will remain OFF only as long as 'both' lights are ON.

The importance of truth tables in designing electronic circuits cannot be overstated. In any digital electronic circuit that employs logic gates (AND, NOT, OR, NAND, etc.), you must define what you want the circuit to do. Obviously, the best way to see

FIGURE 18.

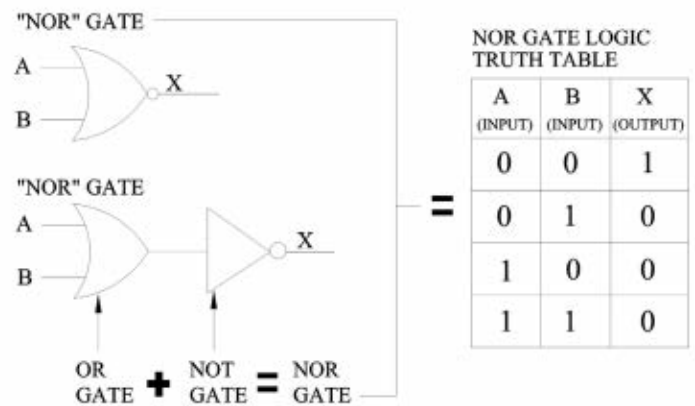


FIGURE 19.

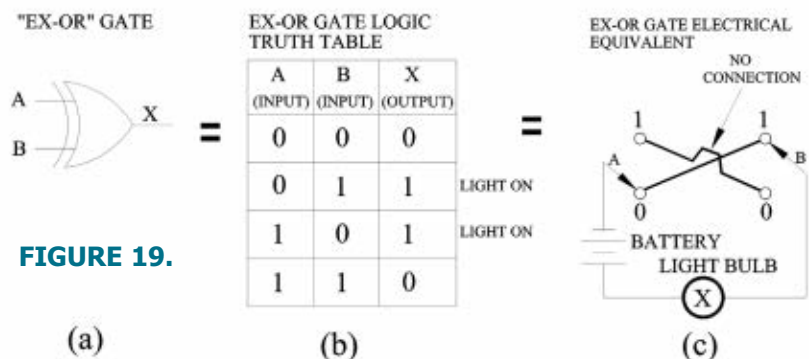


FIGURE 20.

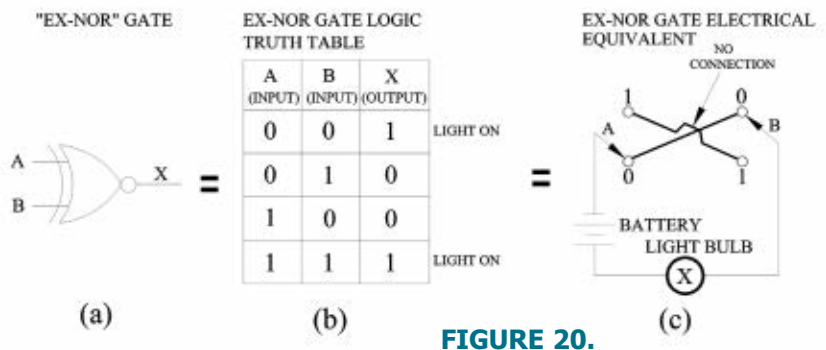
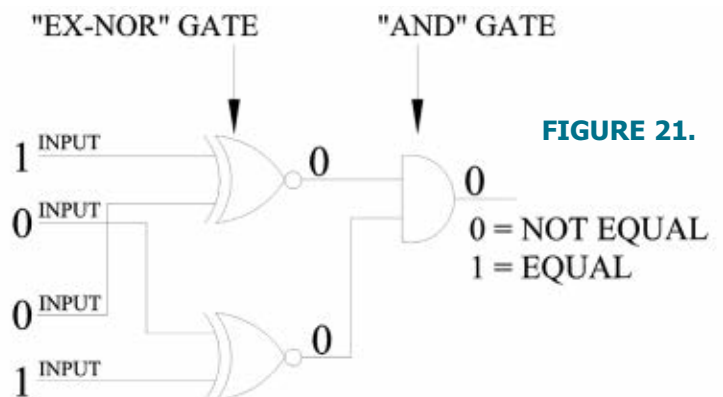


FIGURE 21.



what the input and output logic of your circuit will be is to set up a truth table.

Let's move on to our next gate now: the NOR gate.

The NOR Gate

Once again, the truth table in **Figure 18** tells us exactly how the NOR gate works. The only time the output (X) of a NOR gate will equal a 1 is when 'both'

inputs A and B are 0. Another way to say it is when both inputs to a NOR gate are 0 (LOW), the output of a NOR gate will be 1 (HIGH).

Of course — like our other gates — there is another way to look at the truth table. Notice that if 'either' input A or B to the NOR gate is a 1, the output (X) is a 0. We can also say that any 1 on either input (A or B) of a NOR gate will produce a 0 (LOW) at its output (X).

Let's move on to the next gate: the EX-OR.

The EX-OR Gate

Look at **Figure 19c** and you'll notice that the light will turn ON when switch A is in the 0 position and switch B is in the 1 position. The reverse is also true.

The circuit will be complete (light ON) when switch B is in the 0 position and switch A is in the 1 position. At no time will the light come ON if switch A or switch B are in the same position (1, 1 or 0, 0).

Here's another way to say it: Either 1 or 0, but not both. Let's now discuss EX-NOR gates.

The EX-NOR Gate

As you can see in **Figure 20c**, the light will only turn ON when switch A and switch B are both in either the 1 position or the 0 position.

In other words, you have an output (light ON) when 'both' switch A and switch B are in the same position, but not when A and B are either 1, 0 or 0, 1.

You can say it like this: *both but not either or*.

The EX-OR and the EX-NOR gates are used extensively in digital circuits. One application of the EX-NOR — used quite frequently in communications — is to verify that two binary numbers are equal.

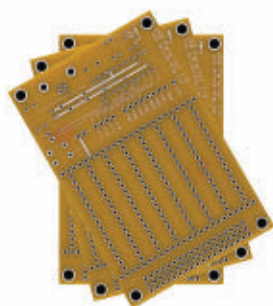
Remember, we can change binary numbers into decimal numbers, and through the use of EX-NOR gates compare two decimal numbers by comparing their binary equivalent.

FREE

PCB Layout Software &
PCB Schematic Software



expresspcb.com

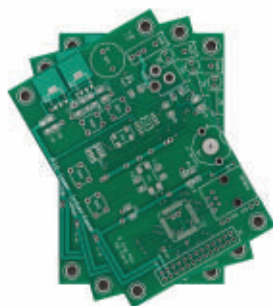


3 PCBs

\$41 + shipping

2-layer
MiniBoard

(Shipped in 1 business day)



3 PCBs

\$61 + shipping

2-layer
MiniBoardPro

(Shipped in 2 business days)



3 PCBs

\$81 + shipping

4-layer
MiniBoardPro

(Shipped in 3 business days)

- 1** **DOWNLOAD** our free CAD software
- 2** **DESIGN** your 2 or 4-layer PCB
- 3** **CHECK** your design with xCHECK DRC
- 4** **SEND** us your design with just a click
- 5** **RECEIVE** top quality boards in just days

Now Offering
**GROUND
SHIPPING**

- Continental US Customers -

LIMITED TIME PROMOTION | www.expresspcb.com

For example, if we needed to compare two binary numbers like 10 and 01 to verify that they are equal, we could use the EX-NOR gate circuit in **Figure 21**.

Binary 1, 0 and 0, 1 presented to the inputs of both EX-NOR gates produces a 0 at the output of the AND gate. Therefore, 1, 0 and 0, 1 are not equal binary numbers.

Now, try 1, 0 and 1, 0 at the inputs of both EX-NOR gates. The output of the AND gate will equal a 1, thereby

signaling that both binary numbers are equal.

Well, that about wraps it up. I hope you enjoyed learning about the digital world. **NV**

DEVELOPING PERSPECTIVES

Continued from page 5

or similar solid-state stereo amplifier with superior specifications, the form factor of a paperback, and the all-important remote.

Although I'll concede on the specifications front, I'll counter that I prefer the warm coloration that vacuum tubes provide. Most of all, I know that I can repair the amp — regardless of what happens. The transformers are a bit scarce, but can be found on eBay and other online sources.

Otherwise, everything is ordinary electronics stock — capacitors, resistors, diodes, and vacuum tubes. Schematics and manuals are freely available on the Web, and there are numerous third parties that cater to vacuum tube amp owners.

Do I miss a remote? I can live without one. Am I concerned that vacuum tubes are about as far from "green" as an electric toaster? Not really, because I use the amp maybe 30 to 40 minutes a day.

Besides, I'm saving one more device from the landfills. And lightning strikes? Bring them on!

In this era of disposable unrepairable electronics, I suspect that there's a growing demand for the simpler but workable electronics of the past.

If you've recently turned to vintage repairable electronics, I'd like to hear about it. **NV**

Get the Parts You Need FAST!

HIGH QUALITY ELECTRONIC COMPONENTS
IN STOCK AND READY TO SHIP



- Consumer, commercial, industrial and MRO applications
- Huge product inventory
- Wide selection and complete specs
- No minimum order
- Cross referencing for thousands of parts
- Same day shipping available

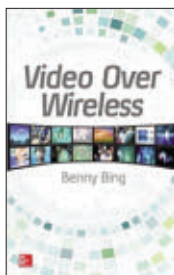
ORDER TODAY!

www.ntepartsdirect.com

GREAT FOR DIYers!

Video Over Wireless by Benny Bing

This practical volume offers expert guidance on delivering high quality video communications over wireless networks.

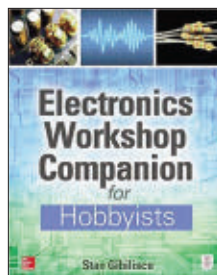


Wireless access has become the dominant medium for network connectivity due to the proliferation of smart-phones and tablets. This guide explains emerging technologies that will help overcome these challenges — all presented in a clear and concise format.

Reg. Price \$120.00 Sale Price \$ 101.95

Electronics Workshop Companion for Hobbyists by Stan Gibilisco

In this practical guide, electronics expert Stan Gibilisco shows you, step by step, how to set up a home workshop so you can invent, design, build, test, and repair electronic circuits and gadgets. Electronics

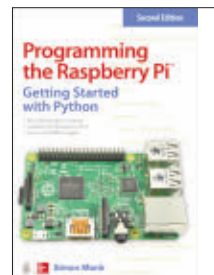


Workshop Companion for Hobbyists provides tips for constructing your workbench and stocking it with the tools, components, and test equipment you'll need. Clear illustrations and interesting do-it-yourself experiments are included throughout this hands-on resource. **\$25.00**

Programming the Raspberry Pi, Second Edition: Getting Started with Python

by Jack Simon

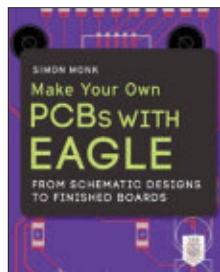
This practical book has been revised to fully cover the new Raspberry Pi 2, including upgrades to the Raspbian operating system. Discover how to configure hardware and software, write Python scripts, create user-friendly GUIs, and control external electronics. DIY projects include a hangman game, RGB LED controller, digital clock, and RasPiRobot complete with an ultrasonic rangefinder



\$15.00

Make Your Own PCBs with EAGLE by Eric Kleinert

Featuring detailed illustrations and step-by-step instructions, *Make Your Own PCBs with EAGLE* leads you through the process of designing a schematic and transforming it into a PCB layout. You'll then move on to fabrication via the generation of standard Gerber files for submission to a PCB manufacturing service. This practical guide offers an accessible, logical way to learn EAGLE and start producing PCBs as quickly as possible.



\$30.00

Programming the Intel Edison: Getting Started with Processing and Python by Donald Norris

Learn To Easily Create Robotic, IoT, and Wearable Electronic Gadgets!

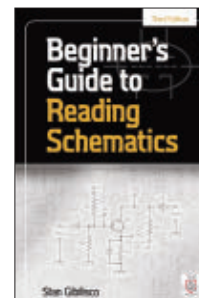
Discover how to set up components, connect your PC or Mac, build Python applications, and use USB, Wi-Fi, and Bluetooth connections. Start-to-finish example projects include a motor controller, home temperature system, robotic car, and wearable hospital alert sensor.



\$20.00

Beginner's Guide to Reading Schematics, 3E by Stan Gibilisco

Navigate the roadmaps of simple electronic circuits and complex systems with help from an experienced engineer. With all-new art and demo circuits you can build, this hands-on, illustrated guide explains how to understand and create high-precision electronics diagrams. Find out how to identify parts and connections, decipher element ratings, and apply diagram-based information in your own projects.



\$25.00

How to Diagnose and Fix Everything Electronic by Michael Jay Geier

Master the Art of Electronics Repair

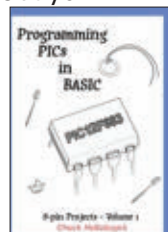
In this hands-on guide, a lifelong electronics repair guru shares his tested techniques and invaluable insights. *How to Diagnose and Fix Everything*



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. **\$24.95**

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**



Programming Arduino Next Steps: Going Further with Sketches by Simon Monk

In this practical guide, electronics guru Simon Monk takes you under the hood of Arduino and reveals professional programming secrets. Also shows you how to use interrupts, manage memory, program for the Internet, maximize serial communications, perform digital signal processing, and much more. All of the 75+ example sketches featured in the book are available for download. **\$20.00**



Order online @ store.nutsvolts.com
Or CALL 1-800-783-4624 today!

EDUCATIONAL

Beginner's Guide Educational Combo!

Price Reduced
Now Only \$219.95

An Arduino Workshop

Are you puzzled about the Arduino but finding it difficult to get all the pieces in one place?

Joe Pardue
SmileyMicros.com

Puzzled by the Arduino?

Based on *Nuts & Volts* Smiley's Workshop, this set gives you all the pieces you need!

Educational Combo \$124.95

We have 33 amazing titles of PROJECTS FOR THE EVIL GENIUS

Need a project to help you spring into summer? Check out the titles below

- 101 Outer Space Projects for the Evil Genius
- 101 Spy Gadgets for the Evil Genius
- 123 PIC Microcontroller Experiments for the Evil Genius
- 123 Robotics Experiments for the Evil Genius
- 126 Physics Projects for the Evil Genius
- 15 Dangerously Mad Projects for the Evil Genius
- 25 Home Automation Projects for the Evil Genius
- 30 Arduino Projects for the Evil Genius, Second Edition
- 30 BeagleBone Black Projects for the Evil Genius
- 48 Science Fair Projects for the Evil Genius
- 50 Awesome Auto Projects for the Evil Genius
- 50 Green Projects for the Evil Genius
- 50 Model Rocket Projects for the Evil Genius
- 51 High-Tech Practical Jokes for the Evil Genius
- Arduino + Android Projects for the Evil Genius
- Bike, Scooter, and Chopper Projects for the Evil Genius
- Bionics for the Evil Genius
- Electronic Circuits for the Evil Genius, Second Edition
- Electronic Gadgets for the Evil Genius 2 Ed
- Electronic Games for the Evil Genius
- Fuel Cell Projects for the Evil Genius
- Holography Projects for the Evil Genius
- Mechatronics for the Evil Genius
- Mind Performance Projects for the Evil Genius
- MORE Electronic Gadgets for the Evil Genius
- PC Mods for the Evil Genius
- PICAXE Microcontroller Projects for the Evil Genius
- Programming Video Games for the Evil Genius
- Raspberry Pi Projects for the Evil Genius
- Recycling Projects for the Evil Genius
- Solar Energy Projects for the Evil Genius
- Telephone Projects for the Evil Genius
- tinyAVR Microcontroller Projects for the Evil Genius

MORE ELECTRONIC GADGETS FOR THE EVIL GENIUS

40 NEW BUILD-IT-YOURSELF PROJECTS

No April Fool's joke! All Evil Genius books on sale for \$19.95 ea for the month of April.

CD-ROM SPECIAL

Nuts & Volts 12 CD-ROMs & Hat Special!

That's 144 issues. Complete with supporting code and media files.

Free Shipping!

Only \$249.95 or \$24.95 each.

The Nuts & Volts Pocket Ref

All the info you need at your fingertips!

This great little book is a concise all-purpose reference featuring hundreds of tables, maps, formulas, constants & conversions. AND it still fits in your shirt pocket!

Only \$12.95

Visit <http://store.nutsvolts.com> or call (800) 783-4624

Basic Training for All Programming Languages

Programming languages are akin to political parties. You believe in and support the political party that you think can get the job done to your liking. Likewise, you use a programming language that is most intuitive to you to manipulate a target microcontroller. In the end, it does not matter if the programming language is C, C++, or BASIC. If you code in BASIC, you can still send a message via a microcontroller's UART just as effectively as a C programmer can.

The common denominator is not which program language you use. A solid knowledge of the fundamentals of programming is the stuff that pulls your application together. With that in mind, the folks at M.E. Labs, Inc. (microEngineering Labs) developed their M.E. Labs Trainer.

M.E. Labs Trainer

You don't need a PICkit3. You don't need MPLAB X. You don't need an expensive compiler. The M.E. Labs

Trainer is a self-contained learning system that is based on the time-tested PICBASIC PRO Compiler. As you can see in **Photo 1**, the Trainer can also be used as a proof of concept platform. Many of the PIC16F1xxx and PIC18FxxKxx microcontrollers packaged in the familiar 40-pin DIP package are pin compatible. So, once you get familiar with the Trainer environment, you can replace its incumbent PIC16F1937 with other compatible PIC microcontrollers. If your favorite PIC isn't packaged in 40-pin DIP, you could also build up a custom jig and adapt your target microcontroller via the header pads at the far right of the M.E. Labs Trainer printed circuit board (PCB).

The Trainer's version of a PICkit3 is shown in **Photo 2**. Programming and debugging the PIC16F1937 is accomplished with this PIC-LED group of components and the MicroCode Studio IDE (integrated development environment), which is also included in the M.E. Labs Trainer package. The PIC18F25K50 based circuitry can also be used as a USB to RS-232 portal. MicroCode Studio has a built-in serial monitor that can be used to visually check out the data passed on this USB portal.

A very bright tri-color LED sits among three pots, eight LEDs, four pushbuttons, five touch pads, a 2x16 LCD, and a temperature sensor. All you have to do is dial up some code to use them. The M.E. Labs Trainer comes with plenty of example code and an excellent tutorial manual.

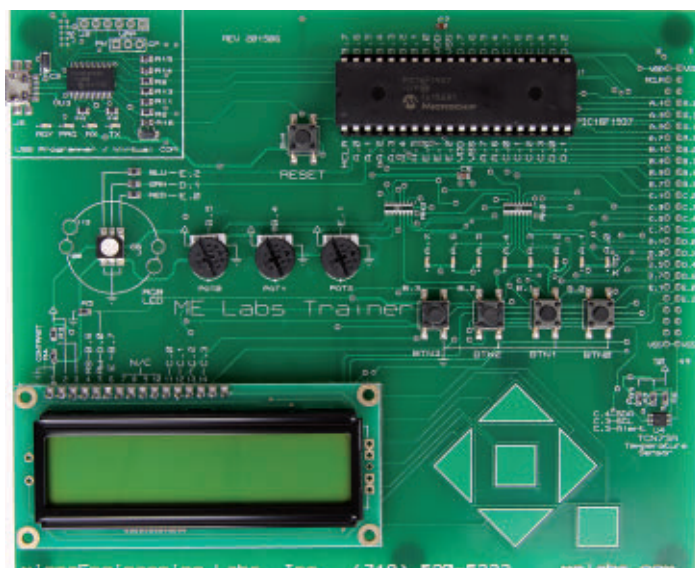
BASIC Glue Code

The idea behind this Trainer is to convey to the user universal coding techniques that transcend language boundaries. In that the Trainer's buttons, LEDs, LCD, pots, and temperature sensor are hardwired, the initial setup of the PIC16F1937 is logically hardwired as well. To that end, the 14-1_Init.pbp file must be included in every M.E. Labs Trainer based project. Let's take a look at the contents of the always-present 14-1_Init.pbp include file:

```

**** Configure and initialize the device ****
DISABLE DEBUG    ' Disable ICD code for this
section
#CONFIG
; set CONFIG1 for internal oscillator
config CONFIG1, FOSC_INTOSC
; set CONFIG2 for int osc pll controlled by
OSCCON register,
; low-voltage programming on
config CONFIG2, PLEN_OFF & LVP_ON
; see file PIC16F1937.INFO (in PBP
installation) for more options
#ENDCONFIG

```



■ **PHOTO 1.** The M.E. Labs Trainer is a complete self-contained embedded development platform. Just about everything you would want to twiddle with via a microcontroller can be found on this board.

The PIC16F1937 is configured to run on its internal oscillator. The low voltage programming option enables the Trainer's onboard programming subsystem. To allow our timing routines to function properly, we must specify and implement a stable CPU clock frequency. In the case of the PIC16F1937, the PLL is not controlled via configuration words, but instead by the OSCCON register:

```
OSCCON = %01111000    \ Set internal osc to 8MHz
                        \ with a 4x PLL enabled
                        \ (4 x 8MHz = 16MHz).
DEFINE OSC 16          \ Tell PBP that the device will
                        \ clock at 16MHz
```

In that the M.E. Labs Trainer is loaded with pushbuttons and potentiometers, we must configure some of the PIC16F1937's GPIO pins as analog and digital inputs accordingly. The Trainer is designed to allow you to identify the GPIO layout without the need of a schematic. Note that all of the pushbuttons, LEDs, and pots are marked as to their GPIO connection to the PIC16F1937. The LCD and temperature sensor connections are also labeled. **Photo 3** is an aerial view of the Trainer's tri-color LED. Here are the PIC16F1937's analog and digital assignments, which are found within the startup include file (14-1_Init.pbp):

```
ANSELA = %00000000    \ PortA all digital
ANSELB = %00110000    \ PortB.4 and B.5 analog
                        \ inputs
ANSELD = %00000000    \ PortD all digital
ANSELE = %00000010    \ PortE.1 analog input
```

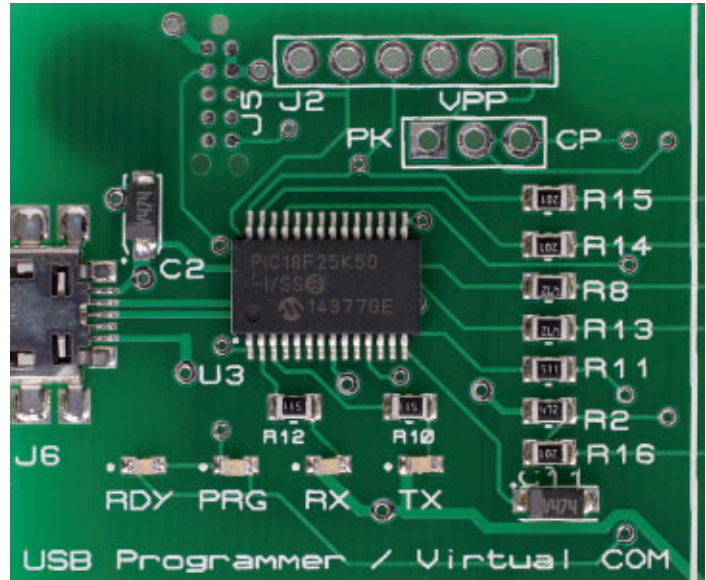
The next thing the 14-1_Init.pbp does is to enable pull-ups on the port pins that service the pushbuttons. Then, the code shuts down all of the LEDs attached to PORTA of the PIC16F1937:

```
WPUB = %00001111      \ Enable pull-ups on pins
                        \ 0-3 of PORTB (RB0-3)
OPTION_REG = %01111111 \ Pull-ups enabled by
                        \ individual WPuX values

PORTA = %00000000      \ Init all PORTA bits to
                        \ low condition
TRISA = %00000000      \ Make all pins on PORTA
                        \ outputs
```

The final act performed by the 14-1_Init.pbp is the definition of aliases for the LCD. Setting up aliases allows the programmer to use human terms for electrical connections to the PIC16F1937's GPIO:

```
\ Set LCD Data port
DEFINE LCD_DREG PORTC
\ Set starting data bit (0 or 4) of 4-bit bus
DEFINE LCD_DBIT 0
\ Set LCD Register Select (RS) port
DEFINE LCD_RSREG PORTB
\ Set LCD Register Select (RS) bit
DEFINE LCD_RSBIT 6
\ Set LCD Enable (E) port
DEFINE LCD_EREG PORTB
\ Set LCD Enable (E) bit
```



■ PHOTO 2. The M.E. Labs Trainer comes equipped with its own programmer/debugger. This subsystem falls under the control of the MicroCode Studio IDE.

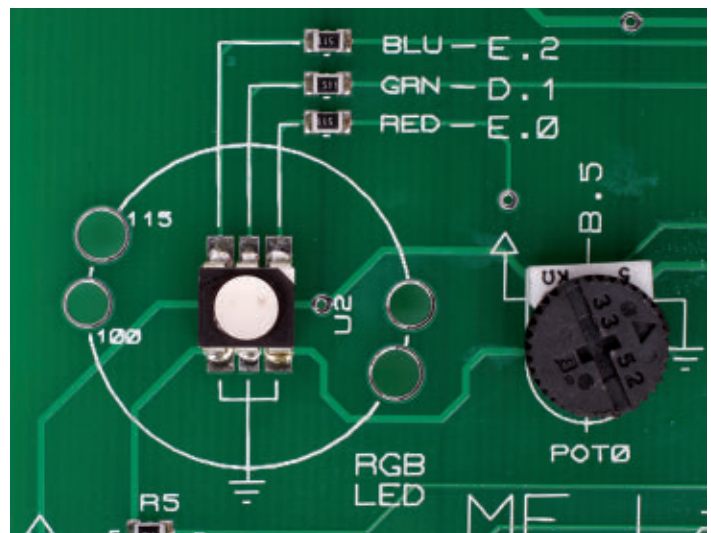
```
DEFINE LCD_EBIT 7
\ Set LCD bus width (4 or 8 bits)
DEFINE LCD_BITS 4
\ Set number of lines on LCD
DEFINE LCD_LINES 2
\ Set command delay time (microseconds)
DEFINE LCD_COMMANDUS 1500
\ Set data delay time (microseconds)
DEFINE LCD_DATAUS 44

PAUSE 200    \ Pause to allow LCD to initialize

ENABLE DEBUG    \ Enable ICD for code that
                \ follows

\*****
```

The PIC16F1937 is ready to rock following the



■ PHOTO 3. I've removed the reflective LED dome that comes installed on the M.E. Labs Trainer. This puppy is BIG and BRIGHT. Note the port notation on the pot.

interpretation and application of the 14-1_Init.pbp code. We are free to make any changes or additions to this code we deem necessary. For instance, if we replaced the PIC16F1937 with another device, odds are we would have to modify the configuration words contained within 14-1_Init.pbp.

Everything can't be done within 14-1_Init.pbp. Application flavor is added with bits of code aimed at particular peripherals and objects. For instance, the Trainer code package includes a factory test application. The factory test code consists of the code contained in 14-1_Init.pbp plus various peripheral setup and exercise routines. Here, we set up the ADC and UART baud rate:

```

DEFINE ADC_BITS 8      \ Inform PBP that we want
                        \ an 8-bit ADC result
ADCON1 = %00110000    \ ADC Left Justified, ADC
                        \ clock Frc, Vss and Vdd
                        \ for Vref

DEFINE HSER_BAUD 9600

```

I Love Interrupts

Most of our PIC applications run in a Super Loop. A Super Loop is a never-ending program cycle that provides round-robin access to the application's tasks. A typical Super Loop goes about its business until it is interrupted. Once an interrupt occurs, the Super Loop execution leaves the round-robin mode and drops everything it is doing to service the interrupt condition.

The M.E. Labs Trainer documentation and example code cover interrupts in the context of PBP3 very well. However, to take full advantage of PBP3's interrupt capability, you must write the interrupt handler in assembler. I have nothing against writing assembler code. I just don't like to do it.

Fortunately, we don't really have to write PBP3 interrupt handlers in assembler anymore thanks to Darrel Taylor's Instant Interrupt system. Not only is the requirement of an assembler based interrupt handler eliminated, the Instant Interrupt system takes most of the register look-up and RTFM pain out of the process. Instant Interrupts can be used instead of the PBP3 *ON INTERRUPT* directive.

To demonstrate how easy it is to apply Instant Interrupts to the M.E. Labs Trainer, let's run through this example:

```

INCLUDE "14-1_Init.pbp"

LED1  VAR  PORTA.0

INCLUDE "DT_INTS-14.bas"  ; Base Interrupt
                          ; System
INCLUDE "ReEnterPBP.bas" ; Include if using
                          ; PBP interrupts

ASM
INT_LIST macro      ; IntSource,      Label,
Type, ResetFlag?
    INT_Handler    TMR1_INT,  _ToggleLED0,
    PBP, yes
endm
INT_CREATE          ; Creates the interrupt
                    ; processor

ENDASM

```

```

    INT_Handler    INT_INT,  _ToggleLED1,
    PBP, yes
endm
INT_CREATE          ; Creates the interrupt
                    ; processor

ENDASM

@  INT_ENABLE    INT_INT    ; enable external
                           ; (INT) interrupts

Main:
    PAUSE 1
GOTO Main

'--[INT - interrupt handler]-----
ToggleLED1:
    TOGGLE LED1
@ INT_RETURN

```

The idea is to toggle the LED attached to the PIC16F1937's RA0 GPIO pin using the pushbutton attached to the PIC16F1937's external interrupt pin RB0. All of the PBP3 magic is contained within the Instant Interrupt include files. As you can see, we have coded in our 14-1_Init.pbp startup file and the necessary Instant Interrupt include files. The interrupt handler is set up within the confines of the ASM/ENDASM directives. The external interrupt is enabled and the program enters an infinite loop at the *Main:* label. Each time the Trainer's BTN0 pushbutton is depressed, an external interrupt is fired and the LED attached to RA0 toggles.

To grasp the real importance of this code, think about "things" happening inside the *Main* loop when the pushbutton is idle. The code only leaves the *Main:* Super Loop to toggle the LED. Otherwise, tasks running inside of the *Main:* loop are executed.

Tick ... Tick ... Tick ...

Let's jettison the pushbutton and force the Super Loop to be interrupted at a constant interval:

```

INCLUDE "14-1_Init.pbp"

LED0  VAR  PORTA.0
LED1  VAR  PORTA.1
LED2  VAR  PORTA.2
LED3  VAR  PORTA.3
LED4  VAR  PORTA.4
LED5  VAR  PORTA.5
LED6  VAR  PORTA.6
LED7  VAR  PORTA.7

INCLUDE "DT_INTS-14.bas"  \ Base Interrupt
                          \ System
INCLUDE "ReEnterPBP.bas"  \ Include if using
                          \ PBP interrupts

ASM
INT_LIST macro      ; IntSource,      Label,
Type, ResetFlag?
    INT_Handler    TMR1_INT,  _ToggleLED0,
    PBP, yes
endm
INT_CREATE          ; Creates the interrupt
                    ; processor

ENDASM

```

```
T1CON = $31 ; Prescaler = 8, TMR1ON
@ INT_ENABLE TMR1_INT ; enable Timer 1
; interrupts
```

```
Main:
TOGGLE LED1
PAUSE 500
TOGGLE LED2
PAUSE 500
TOGGLE LED3
PAUSE 500
TOGGLE LED4
PAUSE 500
TOGGLE LED5
PAUSE 500
TOGGLE LED6
PAUSE 500
TOGGLE LED7
PAUSE 500
GOTO Main
```

```
--[TMR1 - interrupt handler]-----
ToggleLED0:
TOGGLE LED0
@ INT_RETURN
```

With this code, TMR1 overflows will trigger the interrupt and toggle the LED attached to RA0. Using the maximum prescaler value, we've configured TMR1 to run as slowly as possible. Note that the Super Loop is a bit busier here. We are toggling the remainder of the LEDs attached to PORTA. Due to the periodic TMR1 interrupt, LED0 continues to toggle at its steady rate.

What Time is It?

Darrel gave us another goodie that runs right away on the M.E. Labs Trainer platform:

```
INCLUDE "14-1_Init.pbp"
```

```
INCLUDE "DT_INTS-14.bas"
INCLUDE "ReEnterPBP.bas"
INCLUDE "Elapsed_INT.bas" ; Elapsed Timer
; Routines

ASM
INT_LIST macro ; IntSource, Label,
Type, ResetFlag?
INT_Handler TMR1_INT, _ClockCount,
PBP, yes
endm
INT_CREATE ; Creates the interrupt
; processor
ENDASM

@ INT_ENABLE TMR1_INT ; Enable Timer 1
; Interrupts

GOSUB ResetTime ; Reset Time to
; 0d-00:00:00.00
GOSUB StartTimer ; Start the Elapsed Timer

Main:
IF SecondsChanged = 1 THEN
SecondsChanged = 0
LCDOUT $FE,2, DEC Days,"d-",DEC2
Hours,":",DEC2 Minutes,":",DEC2 Seconds
ENDIF
GOTO Main
```

Darrel's elapsed timer code creates an interrupt driven real time elapsed clock that is displayed on the Trainer's LCD. The interrupt creation code and interrupt handler are found within the Elapsed_INT.bas file:

```
; syntax = Handler IntSource,
Label, Type, ResetFlag?
DEFINE Elapsed_Handler TMR1_INT, _ClockCount,
PBP, yes

--[ This is the Interrupt Handler ]-----
ClockCount:
@ RELOAD_TIMER ; Reload
TIMER1
Ticks = Ticks + 1
```



www.melabs.com

Development Tools and Learning Systems

PICmicro® microcontrollers





Reference Manual

PBP3

PICBASIC PRO™ Compiler

- PICBASIC PRO™ Compiler 3.0
- M.E. Labs Trainer
- U2 Programmer
- Prototyping Boards
- Experimenter Boards



Introductory Price of \$129.99

Use Coupon Code NV416 at checkout

Valid April 1- 30, 2016

(Retail price \$189.99 starting April 1, 2016)

Academic and bulk pricing available

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

```

if Ticks = 100 then
  Ticks = Ticks-100
  Seconds = Seconds + 1
  SecondsChanged = 1
  if Seconds = 60 then
    Minutes = Minutes + 1
    MinutesChanged = 1
    Seconds = 0
  endif
  if Minutes = 60 then
    Hours = Hours + 1
    HoursChanged = 1
    Minutes = 0
  endif
  if Hours = 24 then
    Days = Days + 1
    DaysChanged = 1
    Hours = 0
  endif
endif
@ INT_RETURN          ; Restore context and
                      ; return from interrupt

```

If you follow the interrupt handler line by line, you will see there isn't any rocket science involved here. The code is simply accumulating the TMR1 overflow ticks on every timer overflow interrupt.

All Together Now

As you have probably already figured out, we can use Darrel's Instant Interrupt code to fire multiple interrupts from multiple sources:

```

INCLUDE "14-1_Init.pbp"

LED0  VAR  PORTA.0
LED1  VAR  PORTA.1

INCLUDE "DT_INTS-14.bas"  \ Base Interrupt
                          \ System
INCLUDE "ReEnterPBP.bas"  \ Include if using
                          \ PBP interrupts
INCLUDE "Elapsed_INT.bas" \ Elapsed Timer
                          \ Routines

ASM
INT_LIST macro          ; IntSource,      Label,
Type, ResetFlag?
  INT_Handler          INT_INT,    _ToggleLED0,
  PBP, yes
  INT_Handler          TMR0_INT,    _ToggleLED1,
  PBP, yes
  INT_Handler          TMR1_INT,    _ClockCount,
  PBP, yes
endm
INT_CREATE              ; Creates the interrupt
                      ; processor
ENDASM

OPTION REG = OPTION REG & $80 | 1
; Set TMR0 Prescaler to 256, leave RBPU alone
@ INT_ENABLE INT_INT
; enable external (INT) interrupts
@ INT_ENABLE TMR0_INT
; enable Timer 0 interrupts
@ INT_ENABLE TMR1_INT
; Enable Timer 1 Interrupts

GOSUB ResetTime \ Reset Time to 0d-00:00:00.00
GOSUB StartTimer \ Start the Elapsed Timer

```

```

Main:
  IF SecondsChanged = 1 THEN
    SecondsChanged = 0
    LCDOUT $FE,$C0, DEC Days,"d-",DEC2
    Hours,":",DEC2 Minutes,":",DEC2 Seconds
  ENDIF
GOTO Main

\--[INT - interrupt handler]-----
ToggleLED0:
  TOGGLE LED0
@ INT_RETURN

\--[TMR0 - interrupt handler]---(Blinky Light)---
T0Count VAR WORD
ToggleLED1:
  T0Count = T0Count + 1
  IF T0Count = 512 THEN T0Count = 0 : TOGGLE
  LED1
@ INT_RETURN

```

We are using pushbutton BTN0 to fire an interrupt to toggle the LED on RA0. We are also firing a periodic interrupt via TMR0 to toggle the LED attached to the PIC16F1937's RA1 GPIO pin.

During all of this time, we are also ticking off elapsed time per ticks generated by TMR1. The elapsed time is being displayed on the Trainer LCD.

Interrupts, PWM, and Dimming LEDs

Darrel's Software PWM (pulse width modulation) Module works with his Instant Interrupt system to generate multiple software-generated PWM signals on a single PIC:

```

INCLUDE "14-1_Init.pbp"

INCLUDE "DT_INTS-14.bas" ; Base Interrupt System
INCLUDE "SPWM_INT.bas"   ; Software PWM module

DEFINE SPWM_FREQ 40      ; SPWM Frequency
DEFINE SPWM_RES 256      ; SPWM Resolution
DutyVar0 VAR BYTE

ASM
SPWM_LIST macro          ; Define Pin's to use for
                          ; SPWM
  SPWM_PIN PORTA, 0, _DutyVar0 ; and the
  associated DutyCycle variable
endm
SPWM_INIT SPWM_LIST      ; Initialize the Pins
ENDASM

ASM
INT_LIST macro          ; IntSource,      Label,
Type, ResetFlag?
  INT_Handler          TMR1_INT,    SPWMhandler,
  ASM, yes
endm
INT_CREATE              ; Creates the interrupt
                      ; processor
ENDASM

@ INT_ENABLE TMR1_INT    ; enable Timer 1
                      ; interrupts
;

```



```
LoopCount VAR WORD
```

```
Main:
```

```
DutyVar0 = 5  
PAUSE 3000
```

```
FOR LoopCount = 1 TO 4 ; Repeat 4 times  
FOR DutyVar0 = 0 TO 150 ; Fade LED0 UP  
PAUSE 10  
NEXT DutyVar0
```

```
FOR DutyVar0 = 150 TO 0 STEP -1  
; Fade LED1 Down  
PAUSE 10
```

```
NEXT DutyVar0
```

```
NEXT LoopCount
```

```
GOTO Main
```

As you can see, once again all of the interrupt and PWM magic lies inside of the SPWM_Int.bas include file. The PWM signals are made possible by a clock signal that is derived from TMR1 interrupts. The period of the signals produced via TMR1 is varied versus time. The resultant PWM signal is applied to the LED attached to the PIC16F1937's RA0 GPIO pin.

Get One for Yourself

The M.E. Labs Trainer documentation is first rate. Every aspect of logical coding plus multiple tips and tricks are included in the tutorial text. In fact, this Trainer tutorial is so complete, I had to find a subject to write about that it didn't totally cover.

You can get the full skinny on Darrel Taylor's Instant Interrupt system from the M.E. Labs forum, which is part of their website. Otherwise, just about everything you need to know about PBP3 is included in the M.E. Labs Trainer tutorial.

If you simply must write an assembler interrupt handler, the Trainer will show you how. Darrel's Instant Interrupts system supports them too. **NV**

Unfortunately, Darrel Taylor is no longer with us. However, his Instant Interrupts live on. You can get more detailed information and limited support from the microEngineering Labs public forum.

The microEngineering Labs support center does not offer help or support for Instant Interrupts. Therefore, please restrict your questions and comments to the microEngineering Labs community forum.

M.E. Labs Trainer
PBP3
M.E. Labs
www.melabs.com

RF Specialists

"Making your RF ideas into profitable products."

MURS

(Multi-Use Radio Service)



Wireless A40

Portable Wireless Alarm System • Door Alarms • Fence Alarms
Temperature Fire Alarm • Medical Alarm in Hospital/Home Care
Classroom Emergency Alarms • Multi Building Alarms

RF Design Services

Prepared to work with your in-house engineers, or support
your RF project from initial design to implementation.

Industrial • Military • Space • Medical • Smart Grid Metering • SCADA • Lighting Control



LE MOS
INTERNATIONAL

Tel: 1.866.345.3667
orders@lemosint.com
www.lemosint.com

WORLD'S MOST VERSATILE CIRCUIT BOARD HOLDERS



Model
324

Our Circuit Board Holders add
versatility & precision to your
DIY electronics project. Solder,
assemble & organize with ease.

VISIT US ON
Facebook
Twitter
LinkedIn

MONTHLY CONTEST

Visit us on Facebook® to post a
photo of your creative PanaVise
project for a chance to win a
PanaVise prize package.

PANA VISE®
Innovative Holding Solutions



Model
201

7540 Colbert Drive • Reno • Nevada 89511 | (800) 759-7535 | www.PanaVise.com

So, You Think You Know Wi-Fi

Wi-Fi is just about the most widely used wireless technology in the world. There are access points throughout business and industry, institutions, government agencies, and other facilities. Plus, don't forget all those public hot spots in most public places like in convention centers, hotels, and airports. Today, what home does not have a Wi-Fi router? Wi-Fi is convenient as virtually all of our PCs, laptops, tablets, and smartphones have it that we use for email, texting, and Internet access. Isn't the first thing your house guests ask for is your Wi-Fi password? However, there is a lot more you may not know about Wi-Fi. Here is a primer to bring you up to date.

Background

Wi-Fi is a standard developed by the Institute of Electrical and Electronic Engineers (IEEE). It goes by the designation of 802.11 with a letter suffix to indicate the version. **Table 1** shows a list of the major versions of Wi-Fi. The first version emerged in 1997 and used 22 MHz wide channels in the 2.4 GHz Industrial Scientific Medical (ISM) unlicensed band. The modulation was direct sequence spread spectrum (DSSS) or frequency hopping spread spectrum (FHSS) to deliver data speeds of 1 or 2 Mb/s, up to a maximum range of 100 meters.

This initial version was not a big hit, so the IEEE standards development group went back to the drawing board and came up with a revised version 802.11b in 1999. It used the same basic technical specs with DSSS but was able to hit speeds up to 11 Mb/s. It was a major success. Semiconductor companies made ICs and chipsets, and equipment manufacturers made access points, network interface cards (NICs) for PCs, and plug-in modules for laptops. Adoption came quickly.

During 1999, an even newer version emerged: 802.11a. This variation operated in the 5 GHz ISM band. It used OFDM (orthogonal frequency-division multiplexing) and offered speeds to 54 Mb/s under ideal conditions. 802.11a never became popular, but did serve as the basis for later versions. In 2008, 11a was modified into 802.11y for operation in the US licensed spectrum at 3.7 GHz.

The next major development came along in 2003. 802.11g was essentially a version of 802.11a that operated in the 2.4 GHz band. It offered up to a 54 Mb/s data rate using OFDM in a 20 MHz channel. 11g became very popular with most PCs, laptops, and routers adopting it quickly.

It took many more years of development by the IEEE task group to create the next big version. In 2009, 802.11n emerged. It operated in either the 2.4 or 5 GHz band, and could use 40 MHz channels as well as 20 MHz channels. Another feature was the incorporation of multiple input multiple output (MIMO). MIMO is the use of multiple radios and antennas to boost speeds, as well as make connections more reliable despite reflections and other degrading multipath effects of signals at these microwave frequencies. With more bandwidth and MIMO, data rates in excess of 100 Mb/s were possible over a range in excess of 200 feet. As a result, 11n became immediately popular. Today, it is still the most widely used version of Wi-Fi.

The Latest Versions

The most recent iterations of Wi-Fi are 802.11ac and 802.11ad. Also known as WiGig, the 11ad version was ratified in 2012. This version operates in the unlicensed 60 GHz band. It uses OFDM in 2.16 GHz wide channels to deliver speeds up to about 7 Gb/s. Its range is limited inherently by physics, but with the use of high gain beam steering antennas

THE LATEST IN NETWORKING AND WIRELESS TECHNOLOGIES

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/magazine/article/april2016-OpenCommunication-wifi.

a range up to 50 meters is possible without obstacles. 11ad has not been widely adopted as of today, but it has found some applications for short range high resolution video transfer.

The most recent version – 11ac – really seems to be taking off. Many newer laptops and smartphones have already adopted it. This form of Wi-Fi only operates in the 5 GHz band. With bandwidths of 20, 40, 80, or 160 MHz, and up to 8x8 MIMO, 11ac can easily deliver data rates of hundreds of Mb/s with up to 3 Gb/s possible. The range is limited to about 30 meters or so depending on the environment. Look for 11ac to be the de facto version in the near future.

Already, the IEEE working groups are developing 802.11ax and 802.11ay upgraded versions of 11ac and 11ad, respectively. These probably won't be available for a few years, but in the meantime there is plenty of good Wi-Fi to take advantage of.

Something New for Wi-Fi

Some new and different variants of Wi-Fi are soon becoming available. These are 802.11af and 802.11ah. Both address the growing Internet of Things (IoT) and machine-to-machine (M2M) markets. Both are also designed to use frequencies below 1 GHz. At these frequencies, signals travel longer distances and penetrate obstacles better than the microwave frequencies above 1 GHz for a given power level and antenna.

11f is designed for the unused VHF or UHF TV channels (called white space) in the 54 to 790 MHz range. Cognitive radio techniques must be used to avoid interference to local TV stations or nearby wireless microphones. That means accessing an online data base of local spectrum users and implementing a listen-before-transmit process. 11af uses

TV channel bandwidths of 6, 7, or 8 MHz. Modulation is OFDM using BPSK up to 256QAM. In a 6 MHz channel, this could provide a data rate up to about 26 Mb/s. Up to four channels may be bonded to boost rates further. The main benefit of 11af is its long reach. Even with low power (one milliwatt), a range of 1 km or better is possible.

802.11ah is a similar technology but targets the 902-928 MHz ISM band. The Wi-Fi Alliance has named 11ah HaLow. It is a long range, low power technology with a potential range of 1 km or more. It uses 1, 2, 4, 8, or 16 MHz wide channels with OFDM modulation. Its maximum data rate is about 40 Mb/s, although lower speeds like 100 kb/s will probably be more typical.

Some common applications for 11af and 11ah will be smart grid, meter reading, connecting home networks, industrial monitoring and control, and wireless sensor networks.

Neither of these standards is available now, and it is not known how popular they will be. At least they fill a gap on a range that traditional Wi-Fi cannot handle. Both of these new standards will have competition with other low power long range technologies like LoRa from Semtech, France's Sigfox, Weightless, and 5x5 Wireless. Even 2G/3G cellular technology competes in this space. Look for developments related to 11af and 11ah in the years to come.

The Wi-Fi Alliance

One factor that has made Wi-Fi so successful is the Wi-Fi Alliance (WFA). The WFA is a consortium of companies that make Wi-Fi-related chips, equipment, software, and other products. The WFA supports and promotes Wi-Fi. However, the feature of WFA that really makes a difference

PoLabs

For more information or software download please visit 

www.poscope.com/nv

PoKeys Connect, Control



PoKeys is a successful story about versatile and extremely powerful controller.

Save your time with fast connectivity and easy configuration of advanced features.

Enjoy rare moments in life while PoKeys is taking care over your house or garden.

Take a break while PoKeys is controlling stepper motors in your CNC or 3D printer.

Let PoKeys make your imagination come true using our intuitive software and plugins for Mach3/4

Stepper motor drivers Drive

Safest and most efficient force for your motors.



PoScope Mega1+ Measure



Do you need a really powerful and portable oscilloscope that will save your time, money and make your life with measurements easier?

PoScopeMega1+ is lowest power consumption USB oscilloscope, function generator and logic analyzer all-in-one available on the market.

is its certification program.

The WFA tests all Wi-Fi products to ensure that they all not only meet the IEEE standards, but also are fully interoperable. All WFA certified products will reliably talk to one another. The WFA has also created some enhancements to Wi-Fi, as well. These are software additions that extend the usefulness of Wi-Fi. One of these is Wi-Fi Direct. This is a feature that lets Wi-Fi equipped smartphones or laptops or other devices to communicate directly with one another without going through a router or hot spot. Cameras and printers are some examples.

Another feature is Passpoint. This addition eliminates the need for a user to find and authenticate each and every network before using it. This makes it possible for users to roam from one Wi-Fi network to another without having to sign in. Not all devices implement Passpoint or Direct, however, so double-check your device documentation to be sure.

Wi-Fi and the Cellular Carriers

But wait, that's not all! Wi-Fi is also being used by the cellular network providers like AT&T, Sprint, T-Mobile, Verizon, and others. First, some carriers have their own Wi-Fi networks. They use the Wi-Fi networks to offload some data traffic from the cellular network. If access to the network is too great due to high volume video streaming or other traffic, they may offload your connection to their Wi-Fi network to relieve the data burden. This mostly affects data access, but it could be a voice call and you will never know it. In any case, this or offload is not yet widely deployed.

You will also hear of calling that lets you make phone calls using Wi-Fi. Most of the major carriers implement it, and most of the newer Android and Apple smartphones support it. Think of Skype. You may have this and not know it.

Another use of Wi-Fi by the carriers is LTE-U. This is the use of the Wi-Fi 5 GHz unlicensed spectrum for LTE cellular connections. Remember LTE is long term evolution — the fourth generation cellular technology that uses wideband OFDM modulation. Again, if a carrier lacks spectrum, it can “borrow” chunks of the 5 GHz spectrum to cover their high data traffic needs. LTE-U is also called license assisted access (LAA). It is not yet widely implemented. There are efforts to standardize this procedure so that carriers do not interfere with Wi-Fi users.

Wi-Fi is so widely used that it is almost like a utility. We can hardly live without it. Therefore, it is worthwhile to keep up with it. If you want more information on certification or the enhancements to Wi-Fi, go to the WFA website at www.wi-fi.org. Also check out the IEEE working groups for details on all the various modifications and enhancement to the 802.11 standard at www.ieee802.org/11.

NV

IEEE std.	Ratification date	Band	Technology	Modulation	Channel BW	MIMO	Max. Speed
802.11a	1999	5 GHz	OFDM	BPSK, QPSK, up to 256QAM	20 MHz		54 Mb/s
802.11b	1999	2.4 GHz	DSSS	BPSK, CCK	22 MHz		11 Mb/s
802.11g	2003	2.4 GHz	OFDM	BPSK, QPSK, up to 256QAM	20 MHz		54 Mb/s
802.11n	2009	2.4 & 5 GHz	OFDM	Up to 256QAM	20, 40 MHz	Up to 4x4	600 Mb/s
802.11ac	2013	5 GHz	OFDM	Up to 256QAM	40, 80, 160 MHz	Up to 8x8	3 Gb/s+
802.11ad (WiGig)	2012	60 GHz	OFDM	BPSK, QPSK, up to 64QAM	2.16 GHz		7 Gb/s

Table 1: Summary of Major Wi-Fi Standards.

Using 3D printers for practical projects on your workbench.

MakerBot Customizer for 3D Print Creations

I like to create my own 3D print designs, and Tinkercad is one of my favorite programs to use. However, sometimes the near perfect design is available on Thingiverse.com. I've often downloaded a design and then modified it in Tinkercad. Some designs on Thingiverse offer a unique option to let the Customizer software by MakerBot do the modifications for you. It makes customizing a 3D design so much easier when it's available. Not all designs offer it, and it takes a lot of work by the creator. When done right, however, it can be an incredibly useful tool.

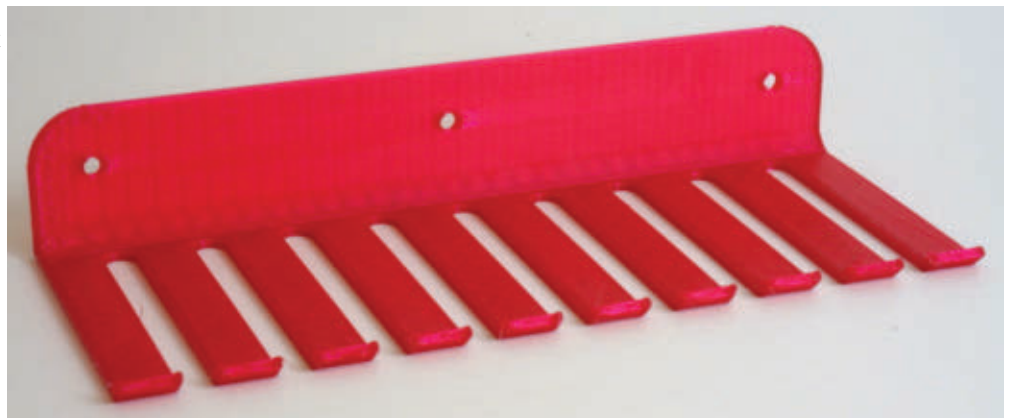
There are two designs I really like that offer the Customizer. The first is a wire rack. I'm sure you've seen these many times and may have some in your electronics lab. They are L-shaped brackets with slots to hold multiple test leads, and are typically made in stamped metal and come in fixed sizes. If you needed a custom size, you'd have to shop around or modify the bracket with a hacksaw and then some work cleaning up the edge.

On Thingiverse, I found this design from user, "thehans." At first, it looks like any standard cable holder with slots for wires, a back bracket with screw holes for mounting, and typical size slots for wires. But then, you notice they designed it for use with Customizer.

Customizer can be launched, and then a whole host of changes to the design are available to you. You can change the number of slots, the thickness of the arms, the size of the back bracket wall, the number and shape of the screw holes, and a lot more. You can do it all while a picture on the right automatically shows you the changes.

It will literally adjust the new design before your eyes so you know what it will look like before you press the "Create Thing" button and have it produce an .STL file. With that file, you can load it into your 3D printer software, slice it, and then print.

On a design like this, you can make different versions to fit at various places on your bench. You can make one for holding smaller wire jumpers. You can make one just for oscilloscope probes. You could probably even make the slots a perfect size to hold a set of small screwdrivers. The ability to change its shape gives you total control of the



■ **FIGURE 1.** Customizable cable organizer.

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/magazine/article/april2016-Practical3DPrinting.



■ FIGURE 2. Abbreviated Customizer options.



■ FIGURE 4. SD/USB stick custom options.

design. All you have to do is enter some parameters. The designer took care of that for you.

Another design I found that uses the Customizer is the SD card/USB memory stick holder shown in **Figure 3** by Thingiverse user, rhmorrison. It was one of the first prints I ever made on my da Vinci 1.0 printer, and I still use it today. I've printed several of them over the past two years, and they are incredibly handy to have on my desk.

The list of parameters to change on this design is shorter than the wire bracket, but it covers what I needed. I can set the number of SD card slots versus the number of USB Flash drive slots. This affects the length.

You can also put a name on the front of it to personalize it or give it a name to help organize what is on the cards. The result is all customizable by you. Clicking on the "Create Thing" button gives you a design that is very easy to print on just about any printer.

Underneath the Customizer is code that creates the design. I have not dug into the code or how to create it. I just find this a very useful option, and I look for designers that know how to use it well. They tend to have some of the most useful designs. I suggest you do your own searching for Customizer designs.



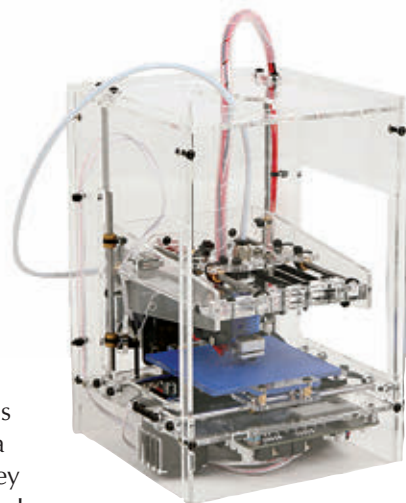
■ FIGURE 3. SD card/USB memory stick holder.

Fabrikator Mini Update

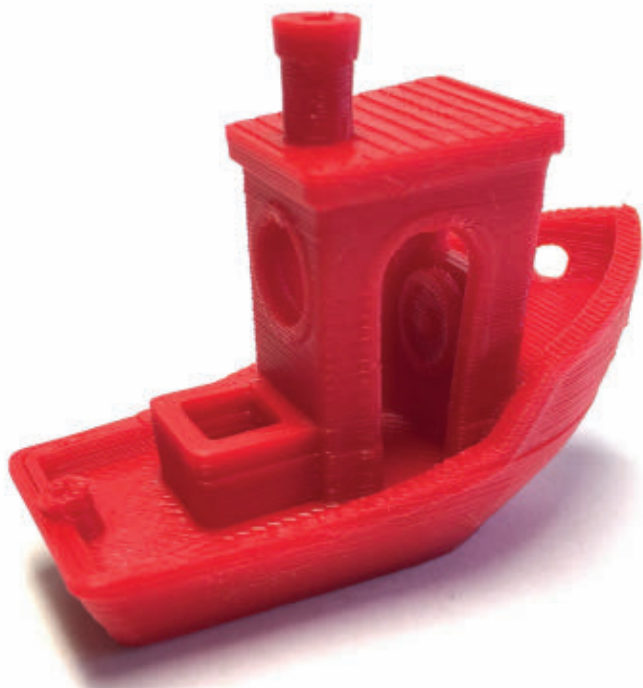
In a previous article, I introduced you to the Fabrikator Mini 3D printer, then right after the article was released, it appeared that **Hobbyking.com** had discontinued manufacturing this tiny (but very portable and handy) printer. I had numerous readers contact me via email asking where they could buy one. Frankly, I was very disappointed that the design was discontinued, as it's one of my favorite 3D printers despite its size. I have printed a lot of knobs and brackets on that little printer with great quality results.

Thankfully, it wasn't discontinued, just taken down for an update. HobbyKing recently released a new version 1.5 with some improvements. It's still the same small size, but now has better heat handling capability for the stepper motor drivers. The design also is put together with a stronger construction method to make it more portable, plus several other smaller updates were added to make it better.

The basic design is very similar to the original, so adding an LCD/SD card controller to it is still a great option to print without being connected to a computer USB port. All the work I did that is posted on my website at www.elproducts.com/fabrikator-mini.html is still applicable and hopefully helpful. I use it with Simplify3D slicing software



■ FIGURE 5. Fabrikator Mini.



■ FIGURE 6. 3DBenchy test print from my Fabrikator Mini.

Resources

Check out my website and blog:
www.elproducts.com

My YouTube channel:
www.youtube.com/elproducts

My 3D designs:
www.thingiverse.com/elproducts/designs

Tinkercad:
www.tinkercad.com

and it produces some of the best print quality I have ever seen. **Figure 6** shows the 3DBenchy test print from 3Dbenchy.com.

So, if you are looking for a small portable (or even a perfect small beginner) printer, this is a great choice.

Using 3D prints in your electronics lab can be very useful beyond making boxes for your electronic designs. It's really just limited by your imagination. Don't forget about the tons of imaginative designs on Thingiverse.com that sometimes can be customized, as well. **NV**

The future is **here.**



XPONENTIAL
 MAY 2-5 | NEW ORLEANS
 AN **AUVSI** EXPERIENCE

XPONENTIAL 2016 is the one event that advances the entire unmanned systems industry. It is the intersection point for commercial and defense applications, and it represents all domains — air, land and sea.

*Experience the latest technology
 you can't find anywhere else.*

xponential.org | [#auvsiXPO](https://twitter.com/auvsiXPO)

>>> QUESTIONS

Data Logger Project Guidance

I want to build two projects.

1. A wireless data logger to monitor temp/humidity at various points in my home to a Windows environment.

2. A wireless data logger to monitor decibels of sound. I live next to a busy street, and want to experiment with various sound suppression options and monitor success rates.

I would appreciate guidance on platforms, hardware, etc. As always, any insights and feedback are appreciated.

#4161

Mark McCurdy
Grapevine, TX

Coils and Magnets and Turns... Oh My!

I have a question about an article I saw where they had a platter of coils and a platter of magnets. They spun the magnets over the coils producing energy. My question is: How do you determine the correct wire size and number of turns, size and the number of coils, and the size and number of magnets to get to a desired voltage and amperage?

Is there a best way to figure out what combination works?

I appreciate any insight or direction.

#4162

Craig MacP
via email

Power Supply Replacement

I have a set of computer speakers which have become separated from their power supply.

Can you help me work out what

power supply I need to replace the missing unit?

It may be helpful (to your other readers as well) if you could suggest a general procedure for this sort of situation that avoids blowing up the equipment.

In my case, I have a head start as the power jack plug is labeled with "DC in" and the polarity (center negative). It does not give the voltage though!

If there were no markings at all I would have to work out AC or DC as well. The connector type should be obvious of course, but I could imagine that even this could be difficult on occasion.

#4163

Mike O'Hagan
Aberdeen, Scotland, UK

>>> ANSWERS

[#2161 - February 2016]

Thirst For Knowledge

I've been an NV subscriber for years. I have an interest, but not a proficiency — yet. Some articles I understand, some I do not, and I really would like to get to the point where I can run with most things presented in each issue, as well as in other magazines. Coming up on retirement, I will have more time to invest in areas of interest. So, I'm looking for guidance in a couple of directions.

1. *I want to get proficient building things using small microprocessors (home automation, data loggers for temp and humidity, robotics, small handheld computers, hacking appliances, other??), but I'm not sure where to start and the best way to learn. I've built a few PCs over the*

years, and while I would have to dust off my memory, I have programmed a little in the past with Visual Basic and SQL.

2. *I want to get more knowledgeable in AC/DC electronics. I've taken a basic fundamentals course at a community college a number of years ago, but would like to refresh that and get on a path to continue building my knowledge base in all aspects, be it AC/DC electronics, computers, communications, etc.*

3. *I have a two year old grandson and I want to get him excited about technology, programming, math, engineering, etc., at age appropriate levels, and see what — if anything — he might want to do with the knowledge as he gets older. Figuring that almost any profession he chooses will benefit from exposure and proficiency in this area, and it can't be accomplished overnight.*

Any insights, thoughts, sources, books, coursework, etc., anyone can share would very much be appreciated, as well as any thoughts on where you see all this going in the next 20 years, so maybe I can guide my grandson to get out in front of things.

#1 *I have worked with various microcontrollers over 30 years and the hard part was always the overhead getting started. The Arduino Uno has less overhead and can do interesting things. As for education, I learned a remarkable amount studying for a Ham Radio License, even if I never needed the license.*

David
via Internet

All questions *AND* answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. All submissions 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!

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

#2 Get a copy of The ARRL Handbook for Radio Communications (www.arrl.org/arrl-handbook-2015). It contains a lot of theory in an understandable format. It also contains full schematics and plans for projects. Go to www.arrl.org/arrl-publication-dealers to find a dealer near you.

Mark Peterson
Plymouth, MN

[#2163 - February 2016]
Cranky Flashlight

I have a three-LED hand-crank flashlight. I've included a copy of the PCB (orange) hoping to find out where the problem may be. The switch (SW1) provides two modes of lighting: one LED only (LED2) and all LEDs (LED1-LED3); in the OFF mode, the hand-crank is to be used in order to recharge the 3.6V rechargeable battery.

The hand-crank flashlight works in either of the two modes ONLY WHEN the hand-crank is being used. When I stop hand-cranking it, that's when lighting stops. I assume the li-on battery is not being charged.

The black/plastic transistor is marked as SC8050. NTE (NTEinc.com) doesn't have a replacement for it. Can someone suggest a replacement and if it's NOT the transistor, suggest where the problem might be?

#1 It looks like part of the PCB track connecting the cathodes of the three LEDs is missing from the diagram. The zener diode cannot be a 1N4148 (a small signal diode); you need a 4.7 or 5.1V zener for the simple voltage regulator to limit the voltage applied to the battery to about 4 to 4.4V. Check the transistor connections. I expect the 2SC8050 collector should go to C1's positive terminal, the base to the junction of the 47 ohm resistor and the zener cathode; the emitter then goes to the

battery positive terminal.

John Swift
Lancashire, UK

#2 Several things seem amiss. Diode 3 should connect to diode 4, probably under R1, R2, and R3. This would complete the negative part of the circuit. Zener diode 1 is not a zener if it is a 1N4148. This is a small signal diode with a reverse breakdown of 75 to 100 volts and with an E-B junction of Q1 that probably protects the generator overvoluting of the circuit. Speaking of Q1, the SC#### is in the form of a Japanese transistor; just needs a 2 first. Search the Internet for a 2SC8050. The picture of the switch does not appear to be an ON-ON-OFF but an ON-OFF-ON.

Now to what's wrong. Since it works when cranked, everything except the battery is working. When not cranking, the circuit fails, then the battery is not charging. A bad solder connection to the board, a bad connection to the battery, or just a bad battery. I would suspect the battery. A voltmeter across the battery would give you the answer.

Allen Baker
via Internet

[#3161 - March 2016]
Don't Burn The Pi

I'm working on a project that requires putting a Raspberry Pi board into a small waterproof box. There is no air circulation for cooling. Is it likely the board will overheat and destroy itself? If so, how can I prevent a meltdown? I'm thinking of using thermal grease and pressing the processor heatsink against the enclosure. Could this work?

I have quite a bit of experience building underwater LED lights for technical diving and well know the challenges of dissipating heat. You

didn't mention what kind of material the enclosure is. Don't use plastic; plastic will just insulate, and heat build-up inside the box will probably destroy the Pi.

The best solution is to use an aluminum box. I'd suggest a rubber adhesive sheet to affix (and insulate electrically) the board to the box with a hole cut out for the processor which is in contact with the aluminum.

The thermal grease (use only a tiny bit) will wick heat to the box and the entire box will act as a heatsink.

Howard Sandroff
Northfield, IL

[#3162 - March 2016]
Snapp, Crackle, POP

Is there a method or modification to prevent the loud pop when removing the headphones from the jack on my tube audio system?

I tried a capacitor across the jack to minimize the pop which worked, but at the expense of all the highs in the music. Anyone have a better solution to try?

#1 The pop is caused by DC reaching the jack. Even if there is a blocking capacitor in the headphone line, you still get the pop before the capacitor charges. Whatever you do, you must reduce or eliminate the DC on the jack. A 470 (?) ohm resistor across the jack will moderate it. If that doesn't suit, get one of Jameco's 1:1 audio transformers. Connect the primary from the set output to ground. Connect the secondary from the jack tip to ground. That should eliminate even the worst case.

Chip Veres
Miami, FL

#2 I would wire a 100 ohm resistor across the socket so there is always a DC path across the amplifier's output.

John Swift
Lancashire, UK

CLASSIFIEDS

LIGHTING



LED lighting for displays, props, scenery, signs, models and more.

www.j2ledlighting.com



Automatic Stair Lighting

www.ReactiveLighting.com

Subscribe today!

Like what you see? Go to www.nutsvolts.com

HARDWARE WANTED

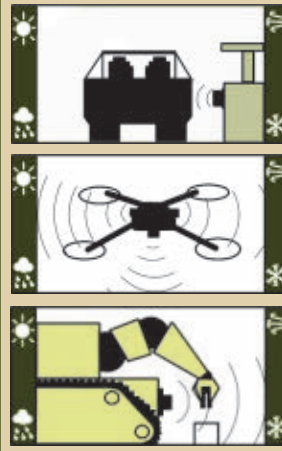
DEC EQUIPMENT WANTED!!!

Digital Equipment Corp.
and compatibles.
Buy - Sell - Trade

CALL KEYWAYS 937-847-2300
or email buyer@keyways.com

ROBOTICS

Need sensors?



www.maxbotix.com

AUDIO/VIDEO

PARTS EXPRESS
YOUR ELECTRONICS CONNECTION

**OVER 18,000
ELECTRONIC PARTS
IN STOCK**



Visit us online to request
a sales flyer and to receive
a special offer!

parts-express.com/nuts
1-800-338-0531

LOOK FOR SEARCH FOR FIND Find your favorite advertisers here!

ADvertiser INDEX

AMATEUR RADIO AND TV

National RF25

BUYING ELECTRONIC SURPLUS

All Electronics Corp.13

CIRCUIT BOARDS

AP Circuits16
ExpressPCB46
Front Panel Express LLC16
Gooligum Electronics12
Saelig Co. Inc.2

COMPILERS

M.E. Labs53

COMPONENTS

All Electronics Corp.13
NTE Parts Direct47
Saelig Co. Inc.2
SDP/SI25

DESIGN/ENGINEERING/REPAIR SERVICES

Cleveland Institute of Electronics..17
ExpressPCB46
Front Panel Express LLC16
National RF25

DEVELOPMENT PLATFORMS/TOOLS

AnarenBack Cover
M.E. Labs53

EDUCATION

Cleveland Institute of Electronics..17

Command Productions29
Gooligum Electronics12
M.E. Labs53
Poscope57
The Teaching Company3

EMBEDDED SYSTEMS

Saelig Co. Inc.2
Technologic Systems31

ENCLOSURES

Front Panel Express LLC16
Probotix5

EVENTS

AUVSI63
Maker Faire65
RoboGames66

LCDs/DISPLAYS

Earth LCD17
Saelig Co. Inc.2

MICROCONTROLLERS / I/O BOARDS

Images25
Ion Motion Control23
M.E. Labs53
Technologic Systems31

MISC./SURPLUS

All Electronics Corp.13

MOTORS / MOTOR CONTROL

Firgelli13
Ion Motion Control23
Robot Power25
SDP/SI25

The Robot MarketPlace23

PROGRAMMERS

M.E. Labs53

RF TRANSMITTERS/RECEIVERS

Lemos International55
National RF25

ROBOTICS

Cleveland Institute of Electronics..17
Firgelli13
Ion Motion Control23
Robot Power25
SDP/SI25
The Robot MarketPlace23

TEST EQUIPMENT

Images25
Poscope57
Saelig Co. Inc.2

TOOLS

PanaVise55
Poscope57
Probotix5

WIRE, CABLE AND CONNECTORS

All Electronics Corp.13

WIRELESS PRODUCTS

AnarenBack Cover
Lemos International55
Technologic Systems31

All Electronics Corp.13
AnarenBack Cover
AP Circuits16
AUVSI61
Cleveland Institute of Electronics17
Command Productions29
Earth LCD17
ExpressPCB46
Firgelli13
Front Panel Express LLC16
Gooligum Electronics12
Images25
Ion Motion Control23
Lemos International55
Maker Faire65
M.E. Labs53
National RF25
NTE Parts Direct47
PanaVise55
Poscope57
Probotix5
RoboGames66
Robot Power25
Saelig Co. Inc.2
SDP/SI25
ServoCity67
Technologic Systems31
The Robot MarketPlace23
The Teaching Company3

PRESENTED BY 

11th

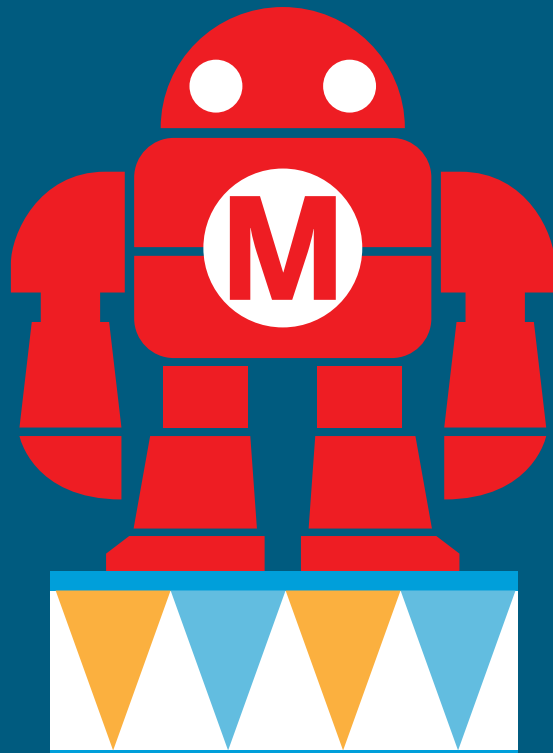
ANNUAL

Maker Faire®

BAY AREA

GREATEST

Show
& TELL
ON
Earth



MAY 20-22 SAN MATEO
EVENT CENTER

7TH ANNUAL WORLD MAKER FAIRE NEW YORK

OCT 1+2 NEW YORK
HALL OF SCIENCE

GET YOUR

TICKETS

TODAY!

makerfaire.com

BROUGHT TO YOU BY
Make: MAGAZINE



ROBOGAMES



Compete at RoboGames!

Last year, over 1000 builders from around the world brought over 800 robots to San Mateo, in the 11th annual international event. This year, we expect even more robots and engineers to compete. Be one! With 80 different events, there's a competition for everyone - combat, androids, sumo, soccer, Lego, art, micromouse, BEAM, or Tetsujin! More than half the events are autonomous. Even if you just come to watch, you'll be overwhelmed with the diversity.

Last year, RoboGames hosted teams with over 800 robots from Argentina, Australia, Austria, Brazil, Canada, China, Colombia, Czech Republic, Denmark, Germany, India, Indonesia, Iran, Japan, Korea, Mexico, Netherlands, Peru, Singapore, Slovenia, Sweden, Taiwan, UK, and the USA.

Be a RoboGames Sponsor!

RoboGames is the world's largest open robot competition - letting people of any age, gender, nationality, or affiliation compete. Sponsoring RoboGames not only helps more people to compete, but also gets your company unrivaled press coverage and visibility. The event has been covered by CNN, ESPN, Fox, CBS, ABC, NBC (live), EBS Korea, NHK Japan, BBC, and countless print and web companies. Your logo can be everywhere the cameras turn!

Rent a Booth!

Booth spaces are at the front of the venue, ensuring lots of traffic. With 3000-5000 people each day, your company will get amazing traffic!

World's Largest Robot Competition
-Guinness Book of Records

North America's Top Ten Geek Fests
-Wired Magazine

SportCenter's Top Ten
-ESPN SportsCenter

"If you are a robot enthusiast, I would definitely encourage you to attend the RoboGames... Take a plane, train, space elevator, but definitely go!"

-Servo Magazine

"Impossible to Imagine, Impossible to Forget!"

-Robot Magazine

Events:

Combat: 340 lbs, 220, 120, 60, 30, 3, & 1 lbs

Androids: Wrestling, Demonstration, Stair Climbing, The Eagle, Door Opening, The Toss, Basketball, Lift and Carry, Marathon, Obstacle Run, Penalty Kick, Dash, 3:3 Soccer, Weight Lifting

Open Events: Fire-Fighting, Robomagellan, Maze/MicroMouse, Walker Challenge, Biped Race, Robot Triathlon, Line Slalom, Ribbon Climber, Vex Open, Lego Challenge, Lego Open, Aibo Performer, Balancer Race, Best of Show, Bot Hockey

Sumo: 3kg - Auto & R/C, 500g, 100g, 25g, Humanoid

Robot Soccer: Biped 3:3 & 5:5, Mirobot 5:5 & 11:11

Junior League: Lego Challenge, Lego Open, Lego Magellan, Woots & Snarks, Handy Board Ball, BotsketBall, 500 g Sumo, 120 lb combat, Best of Show, Vex Open

Tetsujin (ExoSkeleton): Lifting, Walking, Carrying

Art Bots: Static, Kinetic, Bartending, Musical, Drawing

BEAM: Speeder, Photovore, RoboSapien Hacker



April 8-10, 2016

Pleasanton, CA

<http://RoboGames.Net>

GEARMOTORS GALORE



new!
6-12V Precision Planetary Gearmotors w/ Encoders
\$59.99



6-12V Micro Gearmotors w/ Encoders **\$19.99**



3-12V Precision Planetary Gearmotors w/ Encoders
\$19.99



12V Super Duty Actuators (Ball Screw)
\$399.99



12V Heavy Duty Actuators (ACME Screw)
\$129.99



140rpm Right Angle Gearmotor **\$2.49**



6-18V Economy Gearmotors **\$14.99**



6-12V Micro Gearmotors **\$9.99**



3-12V Precision Planetary Gearmotors **\$27.99**



3-12V Standard Gearmotors **\$24.99**

+ ACCESSORIES >



6-12V Precision Gearmotors **\$34.99**



6-12V Heavy Duty Precision Planetary Gearmotors **\$39.99**



Power Input Boards
These boards simply slide over the motor terminals and solder in place. The pre-installed .100" spacing row pins allow plug-and-play connectivity to several different plug styles (JST, Rx battery connector, male servo connector).
\$0.69 - \$0.99



To 6-16V Power Supply or Battery

Digital Manual Speed Controller

A simple and affordable way to precisely control gearmotors and linear actuators. Proportional forward and reverse control from the 5K potentiometer input.

\$79.99



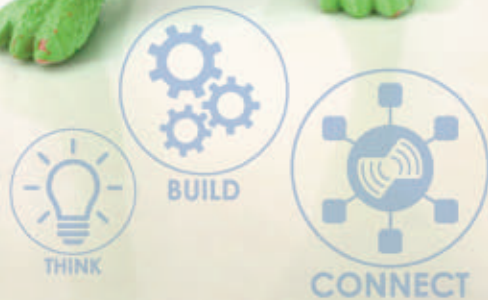
To Motor

new!
Actobotics® Dual Motor Controller
\$89.99-\$99.99



VISIT US ONLINE

"Fetch newspaper, Rex!"



A giant, coffee-carrying IoT dinobot at the command of your index finger? Why not. In fact, wherever your imagination leads you, Anaren empowers you with the industry's easiest-to-use wireless connectivity platform. Scan the code to learn more and register for a chance to win a free Atmosphere MSDK.* Think. Build. Connect.



Anaren®
What'll we think of next?®

* Must be 18 to participate. No purchase necessary. North America only. Void where prohibited. Other restrictions apply.

www.anaren.com/airforwiced/register-win