



circuit cellar

Inspiring the Evolution of Embedded Design

DIVERSE SOLUTIONS ENABLE THE SMART HOME



▶ **Product Focus: Rugged SBCs** ▶ **Application-Specific MCUs** |

Building a Smart Farming Device | **Laser Harp Uses Microchip PIC32** |

Craft a Random Number Generator | **Digital Potentiometers** ▶ **LoRa (Part 1)** |

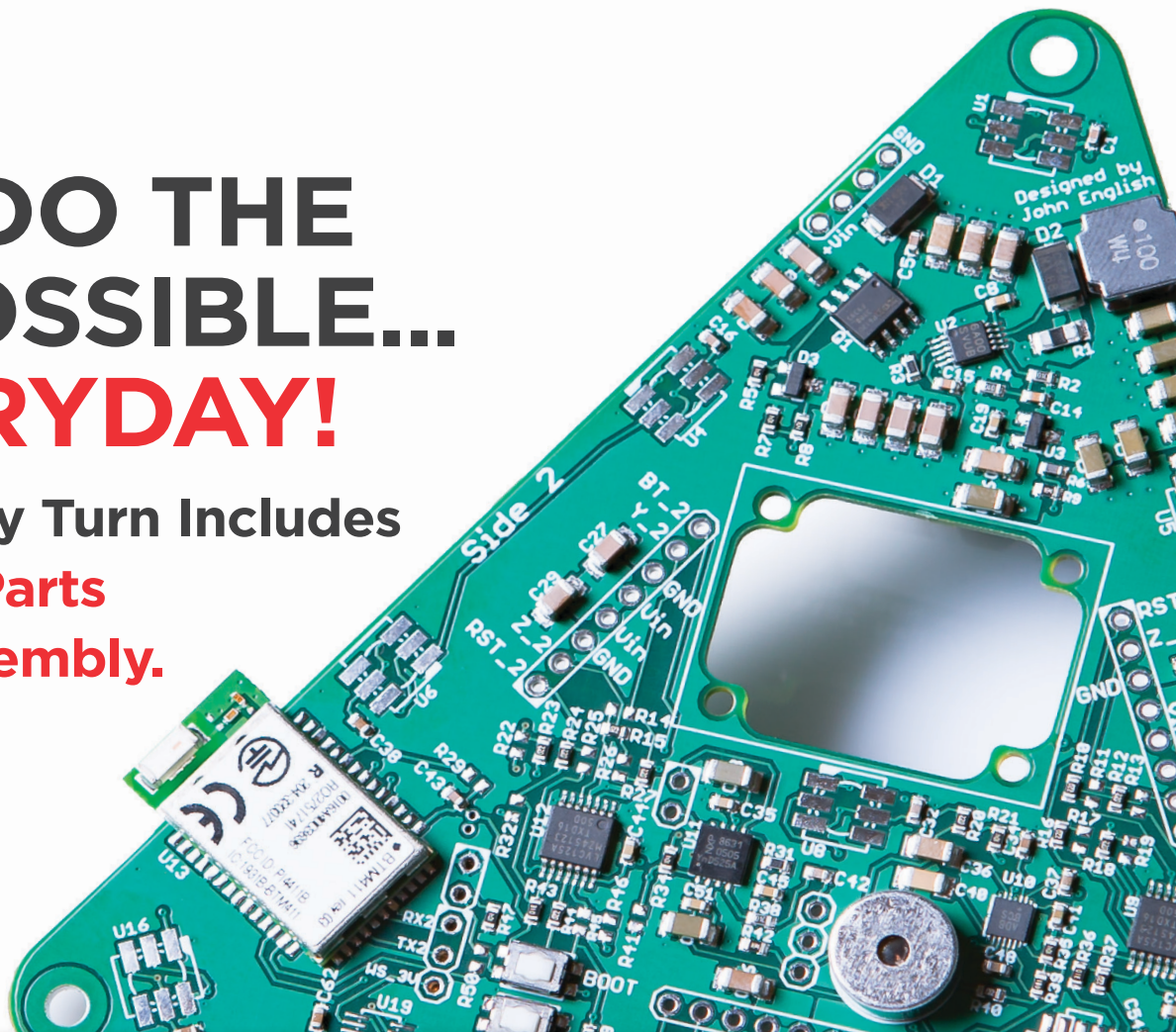
MQ Telemetry Transport (Part 1) | **Semiconductor Basics (Part 2)** |

Intro to Antenna Arrays ▶ **The Future of EDA**



WE DO THE IMPOSSIBLE... EVERYDAY!

Our 5-Day Turn Includes
**Boards, Parts
AND Assembly.**



FREE LABOR

TRY SOMETHING DIFFERENT!

1st time customers receive FREE LABOR, up to \$1,000 on your first turn-key order.

OUR ASSEMBLIES START AT \$250

DOWNLOAD YOUR OFFER CODE HERE: Circuitcellar.com/SlingShot

We want to see **NEW DESIGNS** and **NEW CUSTOMERS!**

No more sacrificing quality for speed or price.

We are your **PCB ASSEMBLY SPECIALISTS!**

SlingShot
ASSEMBLY



Find out why we're different at SlingShotAssembly.com/Different
Call for details: **720.778.2400** or Email: sales@sassembly.com

*Free labor, up to \$1000, for first-time customers on full turn-key assembly orders only.

©2019 COPYRIGHT SLINGSHOT ASSEMBLY

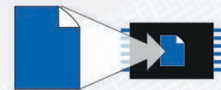


The Embedded Experts



emCompress-ToGo

Compress Data in Real-time on any Embedded System!



More Storage



More Bandwidth



Faster Updates

One Professional Compression Solution for All Applications



Data Loggers



Internet of Things



Space / Avionics



Networking



Medical Devices



Consumer Electronics

- Real-time compression
- Small footprint
- No static RAM required
- Compression of data stream
- High performance
- High compression ratio
- On-target compression & decompression

Worldwide: sales@segger.com

+49 2173 99312 0

U.S. East Coast: us-east@segger.com

+1 978 874 0299

U.S. West Coast: us-west@segger.com

+1 408 767 4068

segger.com

OUR NETWORK



SUPPORTING COMPANIES

Accutrace, Inc.	C3
All Electronics Corp.	77
CCS, Inc.	77
EzPCB	75
Gumstix	33
IAR Systems	37
Productronica 2019	59
Pico Technology	71
Revenue Control Systems	77
SEGGER Microcontroller Systems	1
SlingShot Assembly	C2
SoC Conference 2019	41
Technologic Systems, Inc.	C4, 77
University of Cincinnati Online	23
VersaLogic Corp.	17
WinSystems	31

NOT A SUPPORTING COMPANY YET?

Contact Hugh Heinsohn

(hugh@circuitcellar.com, Phone: 757-525-3677, Fax: 888-980-1303)
to reserve space in the next issue of *Circuit Cellar*.

THE TEAM

PRESIDENT
KC Prescott

EDITOR-IN-CHIEF
Jeff Child

ADVERTISING COORDINATOR
Nathaniel Black

CONTROLLER
Chuck Fellows

SENIOR ASSOCIATE EDITOR
Shannon Becker

ADVERTISING SALES REP.
Hugh Heinsohn

FOUNDER
Steve Ciarcia

TECHNICAL COPY EDITOR
Carol Bower

PROJECT EDITORS
Chris Coulston
Ken Davidson
David Tweed

GRAPHICS
Grace Chen
Heather Rennae

COLUMNISTS

Jeff Bachiochi (From the Bench), Bob Japenga (Embedded in Thin Slices), Robert Lacoste (The Darker Side), Brian Millier (Picking Up Mixed Signals), George Novacek (The Consummate Engineer), and Colin O'Flynn (Embedded Systems Essentials)

Issue 351 October 2019 | ISSN 1528-0608

CIRCUIT CELLAR® (ISSN 1528-0608) is published monthly by:

KCK Media Corp.
PO Box 417, Chase City, VA 23924

Periodical rates paid at Chase City, VA, and additional offices. One-year (12 issues) subscription rate US and possessions \$50, Canada \$65, Foreign/ ROW \$75. All subscription orders payable in US funds only via Visa, MasterCard, international postal money order, or check drawn on US bank.

SUBSCRIPTION MANAGEMENT

Online Account Management: circuitcellar.com/account
Renew | Change Address/E-mail | Check Status

CUSTOMER SERVICE

E-mail: customerservice@circuitcellar.com

Phone: 434.533.0246

Mail: Circuit Cellar, PO Box 417, Chase City, VA 23924

Postmaster: Send address changes to
Circuit Cellar, PO Box 417, Chase City, VA 23924

NEW SUBSCRIPTIONS

circuitcellar.com/subscription

ADVERTISING

Contact: Hugh Heinsohn

Phone: 757-525-3677

Fax: 888-980-1303

E-mail: hheinsohn@circuitcellar.com
Advertising rates and terms available on request.

NEW PRODUCTS

E-mail: editor@circuitcellar.com

HEAD OFFICE

KCK Media Corp.
PO Box 417
Chase City, VA 23924
Phone: 434-533-0246

COPYRIGHT NOTICE

Entire contents copyright © 2019 by KCK Media Corp. All rights reserved. Circuit Cellar is a registered trademark of KCK Media Corp. Reproduction of this publication in whole or in part without written consent from KCK Media Corp. is prohibited.

DISCLAIMER

KCK Media Corp. makes no warranties and assumes no responsibility or liability of any kind for errors in these programs or schematics or for the consequences of any such errors printed in Circuit Cellar®. Furthermore, because of possible variation in the quality and condition of materials and workmanship of reader-assembled projects, KCK Media Corp. disclaims any responsibility for the safe and proper function of reader-assembled projects based upon or from plans, descriptions, or information published in Circuit Cellar®.

The information provided in Circuit Cellar® by KCK Media Corp. is for educational purposes. KCK Media Corp. makes no claims or warrants that readers have a right to build things based upon these ideas under patent or other relevant intellectual property law in their jurisdiction, or that readers have a right to construct or operate any of the devices described herein under the relevant patent or other intellectual property law of the reader's jurisdiction. The reader assumes any risk of infringement liability for constructing or operating such devices.

© KCK Media Corp. 2019 Printed in the United States

INPUT Voltage

If I get into a conversation with anyone in the industry these days—whether by phone or face-to-face—it doesn't take long before I start boasting about *Circuit Cellar* and why it's so great and unique. One of the unique aspects of this magazine is that many of our guest article writers are also readers of the magazine—sometimes very long-time readers. That's probably not so surprising considering that you *Circuit Cellar* readers are yourself quite unique.

Circuit Cellar has a character that separates it from other technology magazines. Yes, there are many long-established publications that cover electronics and whose stated missions are to serve engineers. I've even worked for some of them. But it's fairly obvious by looking at the features and columns in this publication that we don't hold back or curtail our stories when it comes to technical depth. We get right down to the bits and bytes and lines code. Our readers are engineers and academics like you who want to know not only the rich details of a microcontroller's on-board peripherals, but also how other like-minded folks applied that technology to their DIY or commercial project.


Circuit Cellar is also perfectly positioned for a trend in recent years where a rich set of tools, chips and information resources available today allow you to get started on project a very little cost—including free software and inexpensive board-level electronics. These resources put more power into the hands of makers and electronics DIY experts than ever before. On the one hand, there's hardware such as Arduino and Raspberry Pi. On the other hand, there's open source software like Linux and Eclipse that make integrating and developing software easier than ever.

The average *Circuit Cellar* reader like you is a professional engineer developing embedded systems. While those rich resources are of value both to your own "for fun" DIY projects, they're of even more value to you for enhancing your professional engineering projects at work. Part of what makes *Circuit Cellar* so unique is our history of having a foot in both worlds.

The Reader Writer: You

Those new resources aren't replacing commercial software and hardware by any means. When you're developing an embedded system aimed at a professional, commercial application, not everything can be done in DIY mode. Today's embedded systems routinely use millions of lines of code. If you develop that software in-house, you need high quality tools to make sure it's running error free. And if you out-source some of that embedded software, you have to be sure your software vendor is providing a solution you can stake your project's success on.

On the hardware side, the situation is similar. While there's a huge crop of low-cost embedded computer modules available for purchase these days, not all embedded computing modules are created equal. If you're developing a system with a long shelf life, what happens when board's components go end-of-life? Is it your problem? Does the board vendor take on that burden? Have the boards been tested for vibration or temperature specs required for your application?

At *Circuit Cellar*, we're always looking for top-notch technical articles that help readers better understand embedded electronics technology in action. Professional engineers and serious electronics enthusiasts like you are encouraged to submit articles and proposals. Whether it's a project-based article, an article about a technology trend, an analysis of a technical issue or challenge or even a tutorial on a key electronics concept, we're looking for insightful, detailed articles that help your fellow readers do their jobs as embedded system designers. If you have an article or an article proposal, email me at editor@circuitcellar.com. And check out our article submissions page circuitcellar.com/submit for our requirements and guidelines. 



Jeff Child

COLUMNS

PRODUCT FOCUS

46 Rugged SBCs and COMs Tested for Toughness

Tested for Toughness

By Jeff Child

50 The Darker Side An Intro to Antenna Arrays

Elements, Interference and More

By Robert Lacoste

60 Embedded in Thin Slices LoRa (Part 1)

Where Does This Featherweight Fit In?

By Bob Japenga

64 From the Bench MQ Telemetry Transport (Part 1)

Going Aloft with Mosquitto

By Jeff Bachiochi

72 The Consummate Engineer Semiconductor Fundamentals (Part 2)

Diodes in Detail

By George Novacek

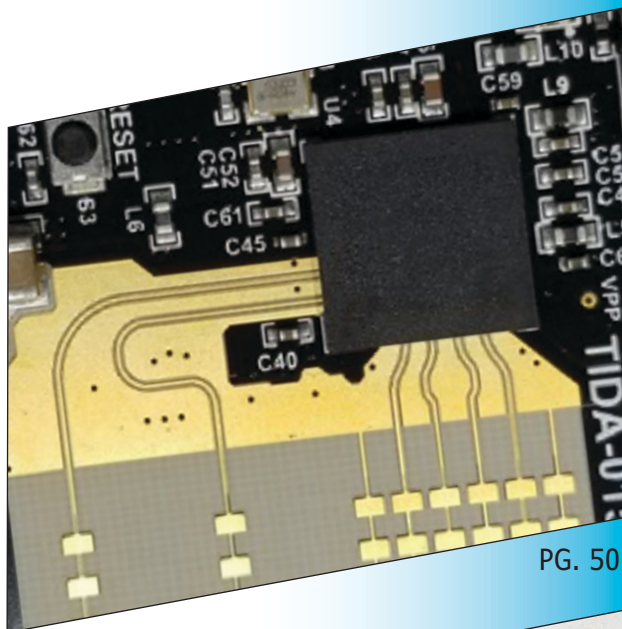
TECH THE FUTURE

79 The Future of Electronic Design Automation Tool Advances Speed Design-to-Manufacture Process

By Keith Lee

76 : PRODUCT NEWS

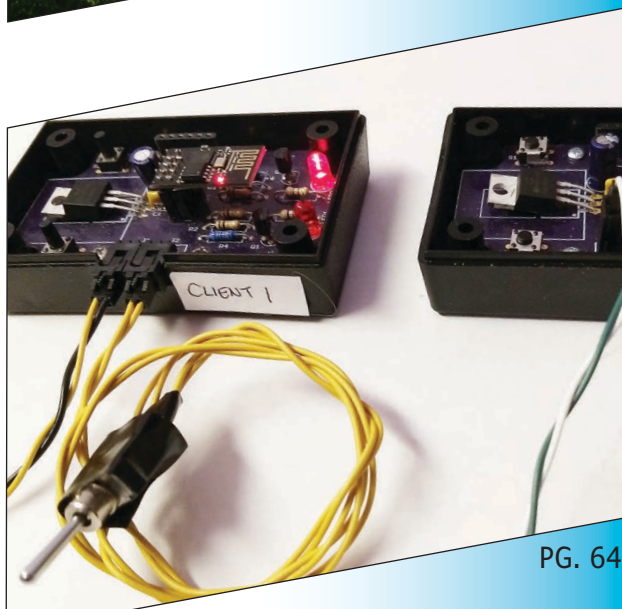
78 : TEST YOUR EQ



PG. 50

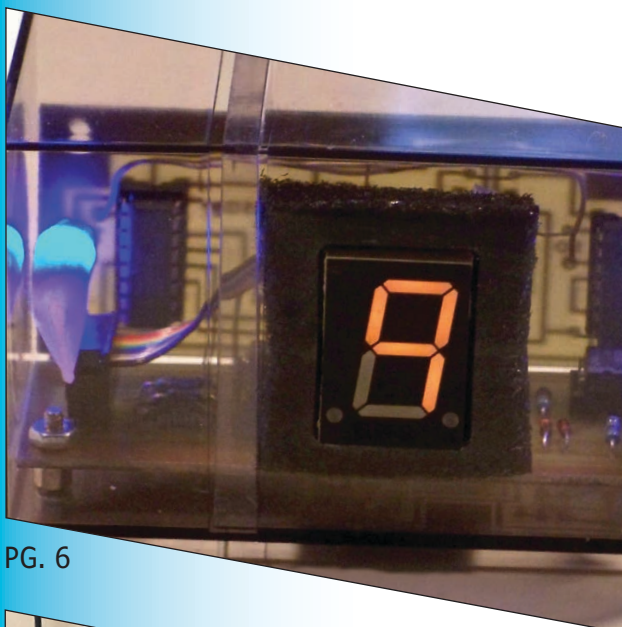


PG. 60



PG. 64

FEATURES



PG. 6

6 A Hardware Random Number Generator

MCU-Based Solution

By Devlin Gualtieri

10 Smart Farming Device Gives Plants a Voice

SmartAgro

By Andrei Florian

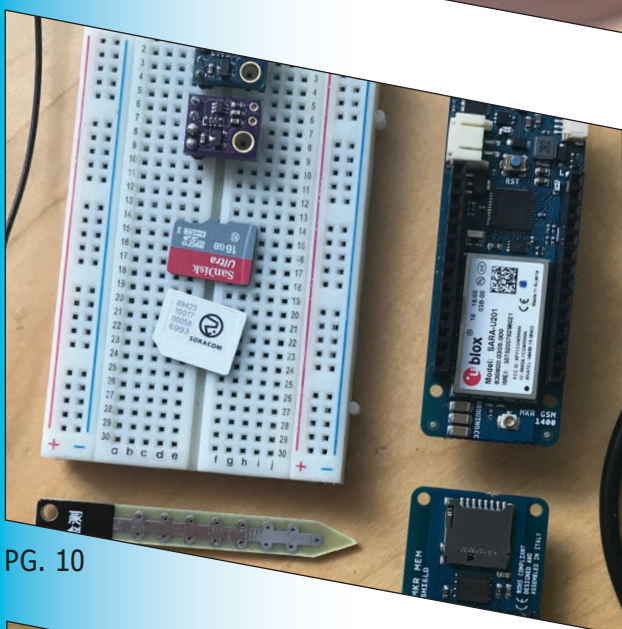
18 Using Digital Potentiometers

Convenient Control

By Stuart Ball

24 The Laser Harp

A Twist on Modern Instruments

By Alex Hatzis

PG. 10

34 Assortment of Tech Solutions Enable the Smart Home

SPECIAL FEATURE

IoT-Leveraged Living Spaces

By Jeff Child

42 Application-Specific MCUs Deliver Rich Functionality

TECHNOLOGY SPOTLIGHT

Purpose-Built Benefits

By Jeff Child

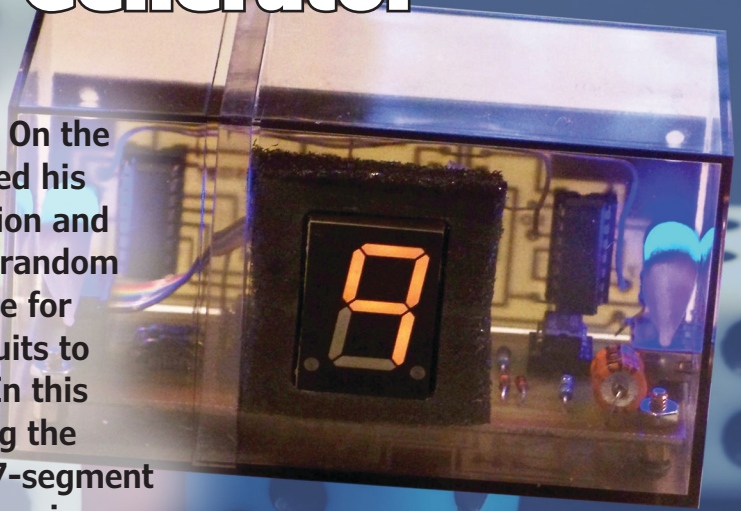
PG. 24

A Hardware Random Number Generator

MCU-Based Solution

FEATURES

Men first walked on the Moon 50 years ago. On the same week as that historic event, Dev divided his time between watching the event on television and building a unique desktop novelty circuit, a random digit generator. This circuit used a Nixie tube for display and a handful of TTL integrated circuits to implement a linear feedback shift register. In this article, Dev updates his original design using the CMOS digital circuits available today and a 7-segment LED display. He also presents an improved version that uses a Microchip PIC MCU.



By
Devlin Gaultieri

I've been interested in random numbers since high school, when I stumbled upon the famous RAND Corporation book of a million random numbers at a public library. This collection of random numbers was based on a study for the US Air Force, and the random numbers were generated using a physical noise source. This book wasn't the first book of random numbers. That milestone goes to "Random Sampling Numbers," created in 1927 by English statistician L.H.C. Tippett, who took random digits from British Census records to create 10,400 four-digit random numbers.

The physical noise source for the RAND study was described as a random frequency pulse generator. This noise source was likely based on Johnson-Nyquist noise—the voltage noise seen across any resistor above absolute zero. The random numbers derived from this noise source were whitened—that is, the data were manipulated to enhance randomness since the circuitry was presumed to be slightly biased against pure randomness, just like a loaded die will show some numbers more than others.

My first attempt at a random number generator used an electronic noise source different from the one used in the RAND study. In my circuit, I used a Zener diode as a noise source, and the amplified white noise

from this source was used to modulate the frequency of an oscillator. I quickly found that the amplitude of low-frequency noise from this source was disappointingly low, so much so that my random numbers were not that random. At that point I decided instead to use a maximal length linear-feedback shift register—something easily built with the TTL logic chips of that era.

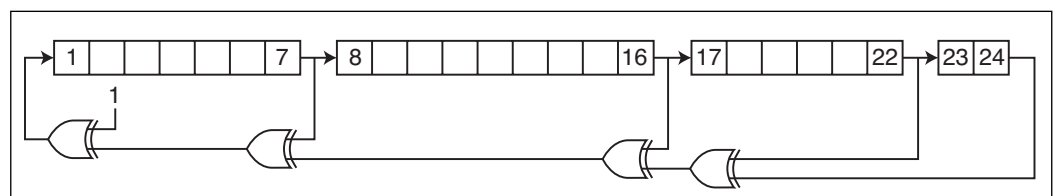
Shift registers operate by shuttling logical "1" or "0" bits at their input along a series of stages, just like students passing notes from desk to desk in a classroom. In a Fibonacci linear-feedback shift register—named after the famous 13th century mathematician, Fibonacci—this input bit is taken from a combination of bits from the stages of the shift register. Exclusive-Or (XOR) logic gates combine bits in a way that maximizes the randomness. For a maximal length 24-bit linear-feedback shift register, the feedback can be taken from cascaded XOR logic gates that tap the outputs of stages 7, 16, 22 and 24, as shown in **Figure 1**. There are other taps that work as well, and taps for both longer and shorter shift registers.

CIRCUITRY

The circuitry to implement a random digit

FIGURE 1

A maximal length 24-bit linear-feedback shift register. Data bits can be extracted from any of the 24 stages, but randomness is only assured when the register is cycled through those bits. The sequence length is more than 16 million, so 4 million random decimal or hexadecimal digits can be extracted.



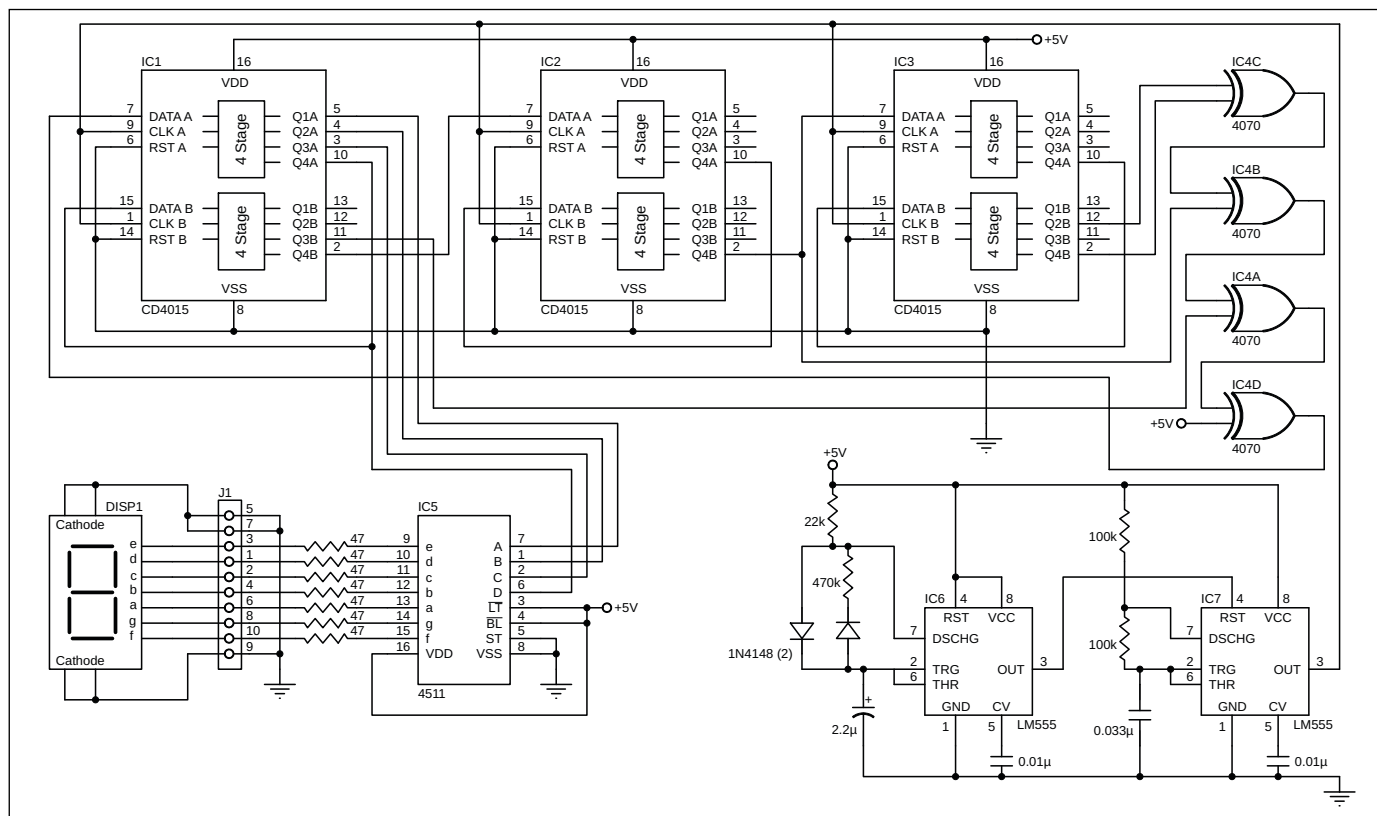


FIGURE 2 Circuitry for the linear-feedback shift register random number generator. The shift register outputs hexadecimal digits, but the 4511-display driver only shows 0-9 and blanks hex digits A-F. The author’s original TTL circuit used a 7447-display driver that outputs arbitrary characters for inputs beyond 9. In that case, some additional logic chips were used to detect these states and blank the display.

generator using this approach, as shown in **Figure 2**, uses seven inexpensive integrated circuits. It has the advantage that no firmware is required. Three of the ICs are 4015 8-bit shift registers that are cascaded to create a 24-bit shift register. There’s a 4070 quad-XOR IC, two 555 timers for cycling the bits and refreshing the display, and a 4511 decoder-driver for a 7-segment LED display. In the circuit shown in Figure 2, IC1-IC3 are the cascaded 8-bit shift registers that comprise the 24-bit linear-feedback shift register. This shift register is tapped at stages 7, 16, 22 and 24 and has its input at IC1-pin7. The XOR chip at IC4 handles the feedback function, and the BCD (binary-coded decimal) digit is taken from IC1.

IC6 and IC7 are 555 timers that cycle the shift register four times each second. The cycle pulses are generated by IC7, and they come in a short burst that’s triggered by IC6. IC6 generates a 50 ms pulse every second, and this signal is sent to the enable pin of IC7 to produce the pulse sequence shown in **Figure 3**. It’s important that there are at least four pulses generated per cycle. If there are fewer than four, successive digits will be correlated and randomness will be lost. There can be more than four, so you can decrease the value of the

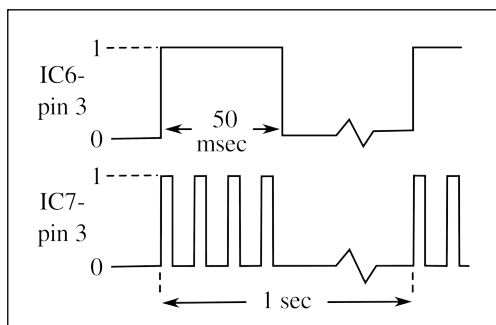


FIGURE 3 Timing pulses generated by the two 555 timers. IC6 gates IC7 to generate four pulses to advance the linear-feedback shift register by a single BCD digit. This pulse train occurs at one second intervals, so there’s a new random number every second.

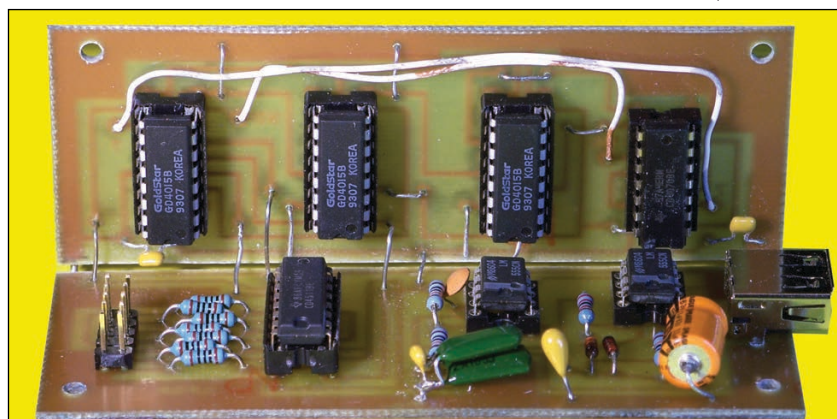


FIGURE 4 The circuit board for the random number generator can be cut in half and folded to produce a more compact design. The USB power connector can be seen on the right, and the connector to the 7-segment display is on the left.

timing capacitor at IC7 to be cautious if you don't have an oscilloscope to count the pulses.

IC5 is a combined decoder-driver that presents the BCD data to the 7-segment LED display. This chip is blind to numbers outside the 0-9 range, so hexadecimal digits A-F are blanked when they occur. This means that random digits will sometimes be more than a second apart. Some additional chips would solve this problem, but I thought that this would unnecessarily complicate the circuit.

There are quite a few ICs in this design, so I designed the PCB so that it can be cut in half and folded into a right angle, as

shown in **Figure 4**. I etch my own circuit boards, so they're always designed with copper conductors on a single side with a few necessary jumpers. Logic circuitry typically needs some longer jumpers for signal connections, in this case from the taps on the shift registers to the XOR chip.

COMPLETED DEVICE

Although this circuit could be used to generate random PIN numbers for your various accounts, it's essentially a desktop novelty—a computer person's bobble head doll. To add to the novelty, I built mine into a semi-transparent plastic case and added two blue LEDs to light the interior. The LEDs needed light diffusers to better spread their light in the inside, and I made these from some plastic tubing and a blob of some translucent silicone adhesive. The completed device can be seen in **Figure 5**.

The device was designed to be powered by a small USB wall outlet transformer, since its current requirement of 135 mA at 5 V is supposedly outside the 100 mA limit of a USB-2.0 computer connector. Since I supposed that computers had internal current-limiting circuitry to prevent damage, I took the chance and plugged into an older desktop computer, and it worked fine. The circuit also worked with the USB-2.0 connection on another home computer. The USB-3.0 specification calls for a 150 mA maximum current draw per connector, so these are presumed safe.

There's a notable Dilbert cartoon, published on October 25, 2001, in which our geek protagonist, Dilbert, is touring the "Land of the Accounting Trolls." He's introduced to their random number generator, a troll who repeatedly says, "nine." Dilbert questions whether this is truly random, and his troll tour guide says, "That's the problem with randomness. You can never be sure." Luckily, there are statistical tests for randomness, the most famous being Diehard, devised by American mathematician and computer scientist George Marsaglia.

While Diehard would be overkill in assessing how well this 24-bit linear-feedback shift register performs, I did check its performance by a computer simulation for a run of 32,000 hexadecimal digits to check the frequency at which each digit appears, as well as the likelihood that particular digits would more likely follow others. The output was reasonably uniform with expected deviations appropriate to the sample size as can be seen in **Figure 6**.

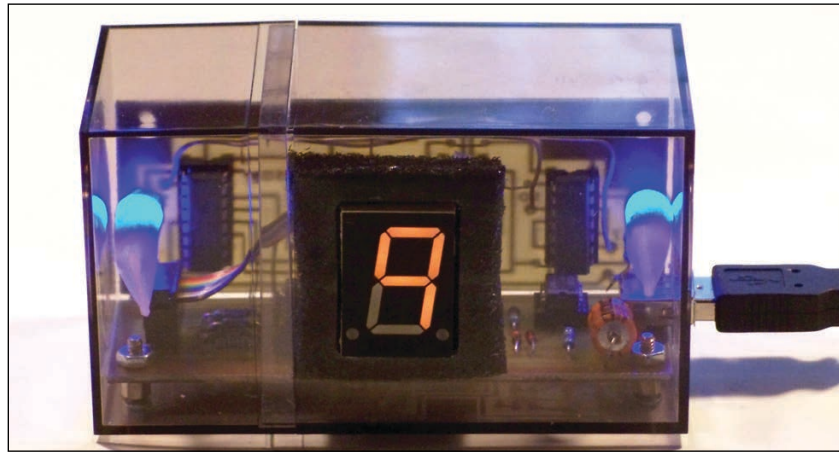


FIGURE 5

The random number generator based on a 24-bit linear-feedback shift register. The device is built into a semi-transparent plastic box that has the internal circuitry lighted by two blue LEDs. The USB power connector is at the right, and the ribbon cable connection to the 7-segment LED display can be seen at the left.

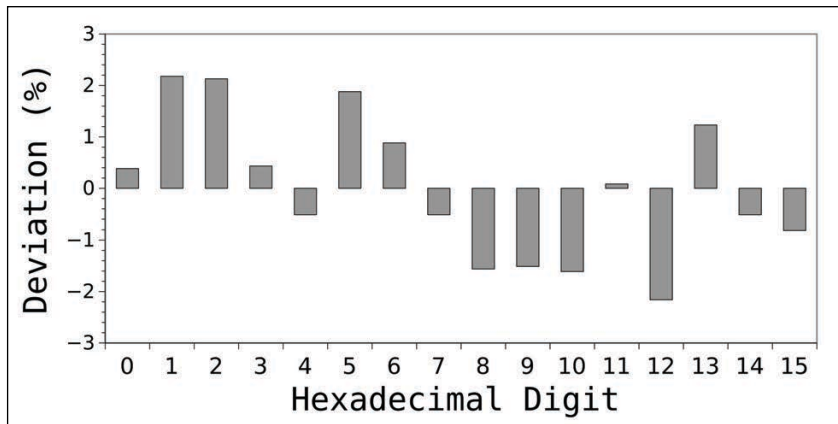


FIGURE 6

Deviation from uniformity for digits from a simulation of the 24-bit linear-feedback shift register. 32,000 digits were analyzed, so the expected frequency of each digit is about 2,000. Statistically, these deviations are what would be expected from the sample size.

For detailed article references and additional resources go to:
www.circuitcellar.com/article-materials

RESOURCES

ME Labs | www.melabs.com

Microchip Technology | www.microchip.com

LIFE IS EASIER WITH MCUs

While the above circuit has the advantage that no firmware is required, I programmed the same linear-feedback shift register into

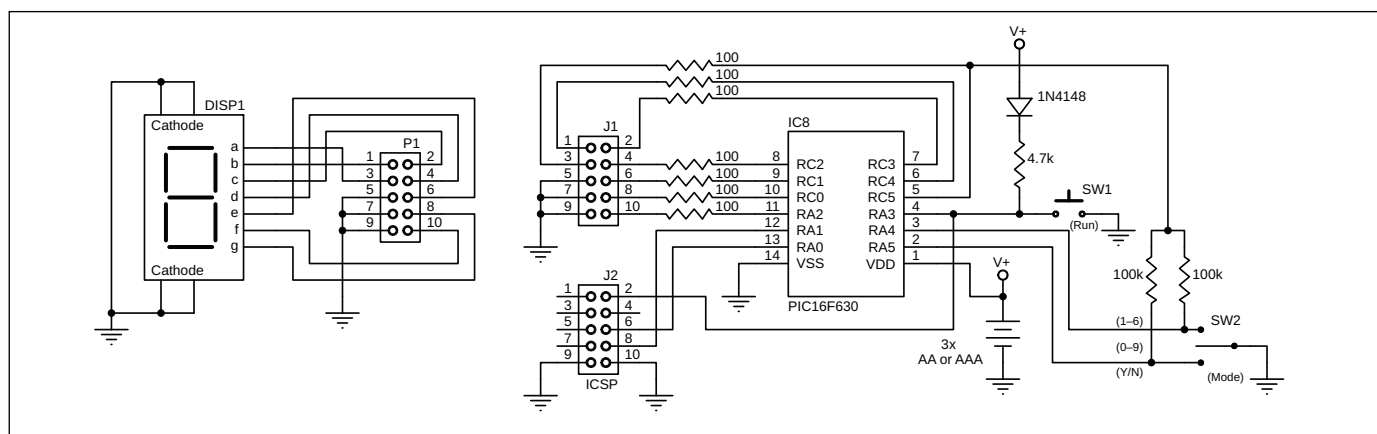


FIGURE 7

Circuitry for the linear-feedback shift register random number generator as simulated by a PIC MCU. Using an MCU allows several modes of operation. The mode switch resistors are powered by the connection at RC5 so they don't draw power when the PIC is in sleep mode.

a Microchip Technology PIC microcontroller (MCU) and gave the device a few options. Since it's possible to program the PIC into a sleep mode that consumes very little current, this second random number generator is battery powered, and it operates by a push button switch. Not only does the device display random digits, but it also simulates dice throws and it will answer a yes/no question.

The simplicity of the circuit is shown in **Figure 7**. The PIC 16F630 MCU has a sufficient number of I/O pins to drive the 7-segment LED display, detect the push button and mode switches, and provide in-circuit serial programming. A center-off toggle switch selects one of three modes: a mode that gives a sequence of six random numbers, another that gives two random numbers from 1-6, and another that randomly gives a "Y" for yes or an "N" for no. The last mode is ideal for making difficult management decisions.

I built the entire circuit—including battery holder, circuit board, switches and 7-segment display—into a container I built from a piece of PVC pipe purchased at a home goods store. I used a PVC end-cap as a removable base to allow access for battery replacement. The current-draw of the device is so small that the batteries will last nearly their shelf life with occasional use. A photo of the device is shown as **Figure 8**.


Unlike the first random number generator, the MCU version requires a firmware program that I wrote using the PICBasic Professional compiler (ME Labs, melabs.com). Because I use Linux exclusively, I run an older version of this compiler and its associated programmer on a very old Windows operating system installed as a virtual machine. On the *Circuit Cellar* code and files download webpage, you can find a compiled hex file to program a PIC MCU without needing a compiler. It would be easy enough to rewrite the source code in C so other compilers can be used, including the free MPLAB C18 compiler from Microchip, manufacturer of the PIC MCUs. 



FIGURE 8

Battery-powered random number generator based on a PIC MCU. A push of the button gives either six random digits, two rolls of six-sided dice, or a yes/no answer, depending on the selected mode. For this version, a simulated 24-bit linear-feedback shift register is used with taps at stages 1, 3, 4 and 24 and the BCD data is taken from stages 21-24. The compiler has a built-in random number generator that can be used instead of the simulated linear-feedback shift register.

ABOUT THE AUTHOR

Dev Gaultieri received his Ph.D. in Solid State Science and Technology from Syracuse University in 1974. He had a 30-year career in research and technology at a major aerospace company and is now retired. Dr. Gaultieri writes a science and technology blog at www.tikaln.com/blog/blog.php. He is the author of three science fiction novels, and books about science and mathematics. See www.tikalnpress.com for details.



Smart Farming Device Gives Plants a Voice

SmartAgro

FEATURES

Smart Farming has many aspects, and among these is the agricultural side. In this project article, Andrei discusses SmartAgro, a device that combines field autonomy with ease of use, allowing farmers to give their plants a “voice.” It lets horticulturists visualize the temperature, soil humidity, UV radiation and more wherever they are, in real time and take action when it is most needed—whether that means turning on an irrigation system or preparing for cultivation.

By *Andrei Florian*

Horticulture has been around for millennia. From the time the first seeds were planted to today, our agriculture has faced many changes. Over time, we humans have learned techniques that allowed us to master farming—from selective breeding of plants that our ancestors performed thousands of years ago, to machines invented to sow better and faster easing the workload of farmers. We have progressed way beyond simply planting a seed. And now the time has come for another agricultural revolution.

In a world enhanced by automation and IoT, farming deserves to get an update too. The application of the Internet of Things (IoT) into farming is known as Smart Farming. Smart Farming offers horticulturists the ability to remotely visualize how their plants are doing in real time—basically giving the plants a voice. That means, when the greenhouse needs watering, it is done automatically. Maybe that sounds too good to be true—to be able to sit down and drink tea while the plants water themselves. Well, there is a catch: the

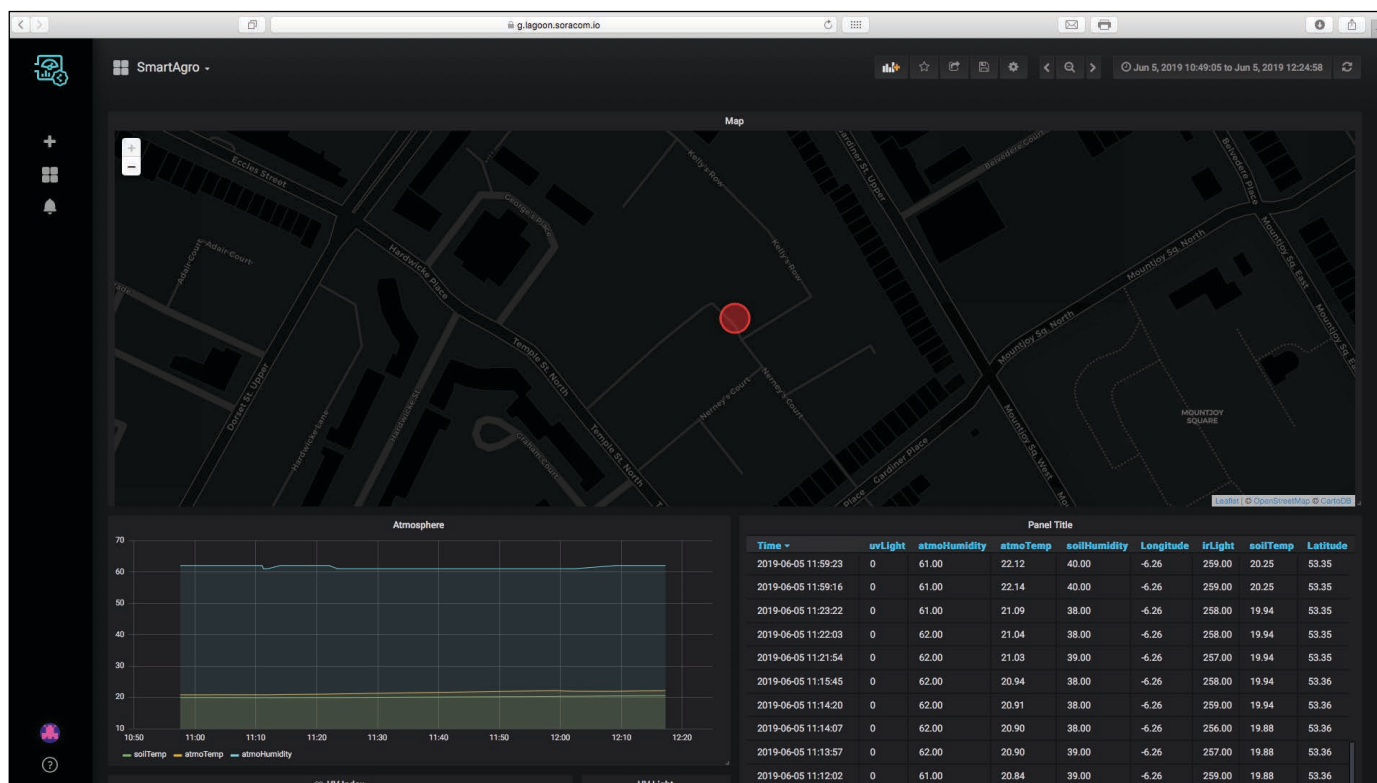
cost. Many Smart Farming solutions are on the expensive side. Recently though, as new tools and technologies have emerged, there are now more options at our disposal allowing us to build applications at a lower cost. This is where SmartAgro comes in.

SmartAgro is a device I created that’s used to monitor multiple parameters on the agricultural field remotely (**Figure 1**). SmartAgro is an inexpensive solution for small and medium horticulture companies to step into the world of Smart Farming. SmartAgro consists of a device that is placed in the field or greenhouse. It can work autonomously in a network with other devices or as a stand-alone platform collecting data from an array of sensors about the soil, atmosphere and light in the area.

The device then reports the data to a back end through GSM where the data is displayed on a serverless dashboard. The end-user (horticulturist or company) will then be able to visualize the data collected by the device or devices on the field with just one click. For the project, I made use of several tools and services from Soracom, including Soracom

FIGURE 1

This is SmartAgro—a device designed to be planted on the field for collecting sensor data. Used as a single unit or in a network, monitoring your horticultural ground can now be done “on the go.”



Lagoon (Figure 2), Soracom Air, Soracom Beam and Soracom Harvest.

SmartAgro can work in a network with an array of similar devices collecting sensor data from the field. The network can include other devices controlling irrigation. Later, in the “Going Further” section, I will explain how this can be done. This article will mostly be based on building one device.

Now, let’s dive into the way this project works—from collecting the sensor data to illustrating it on the dashboard. I have split this discussion into two sections: the front end and the back end. In our case, the front end refers to the device that is placed in the field collecting sensor data and the back end is comprised of the infrastructure that gets the data from the device to the server and the data visualization. The code for this project is available on *Circuit Cellar’s* Article Code and Files webpage.

THE FRONT END

I decided to use the Arduino MKR GSM as the microcontroller (MCU) in this project (Figure 3). I personally like the MKR GSM because it has many great features that make it adequate for operation in the field. The device has a reliable sleep mode that turns off all functions except for the RTC (real-time clock). This significantly increases the device’s battery life, which is crucial to operation on the field. The device is also able to retrieve its geolocation coordinates from the GSM network. All in all, a great device for its price—easy to use and simply runs great.

I have attached an Arduino MKR MEM shield to the device. This is done by simply snapping it on top of the MKR GSM. This shield allows you to control an SD card module by just snapping the shield on the device. Most SD card modules require to be powered with

FIGURE 2

This is a screenshot of the dashboard I created to display the data collected from the sensors. I used a tool offered by Soracom—Soracom Lagoon—to easily visualize the data on maps and graphs. I have also set up alerts that send me emails if the moisture or the temperature of the soil is anomalous.

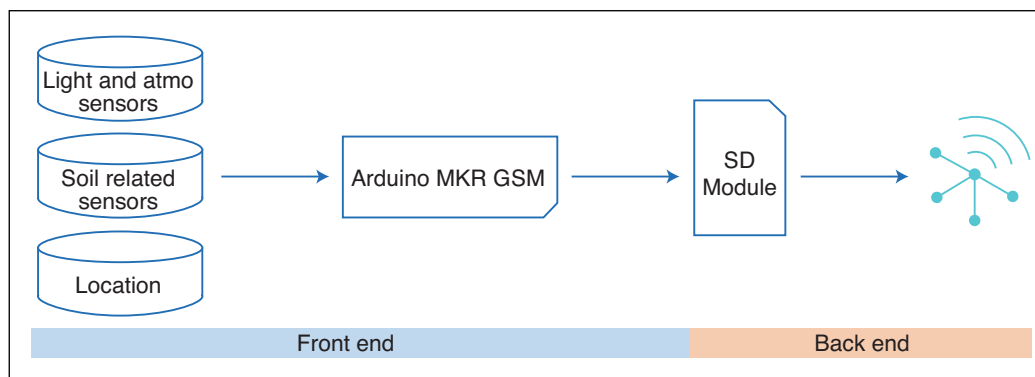


FIGURE 3

This is the functionality overview of the front end. The diagram illustrates the steps taken by the device from taking samples to sending them to the cloud.

Measurement	Sensor Name
[Atmospheric] Temperature	GY21
[Atmospheric] Humidity	GY21
[Soil] Temperature	DS18B20
[Soil] Moisture	Soil Moisture Sensor
[Light] Ultraviolet Light (UV)	GY1145

TABLE 1

This table lists all the sensors used by the device and what they measure. These sensors can be found online in stores like Amazon.

5 V to run. The MKR GSM operates at 3.3 V. This shield enables users to easily operate the module without needing to worry about logic level shifting. An external SD module can be used but that would require stepping up the voltage.

Data is collected from an array of digital and analog sensors attached to the device (**Table 1**). The major worry would be wear and tear of the soil sensors (soil temperature and humidity). I would personally advise the use of a capacitive soil sensor, which has a far longer life span than the traditional one. For the best precision, ensure that the UV sensor is not covered by any material

After the data is collected from the sensors attached to the device, it is compiled by the device. The data has to be saved to an SD card and sent to the cloud. The MKR GSM prepares two buffers. One of the buffers contains all the sensor data arranged in a line of .csv for example:

```
atmoTemp, atmoHum, SoilTemp, SoilHum
23, 53, 22, 45
```

Another buffer is prepared with the data to send to the back end. This buffer is compiled in a JSON buffer and includes other credentials such as the device name (used for identification) and the battery level for example:

```
{
  "Latitude":53.3570404,
  "Longitude":-6.2609935,
```

Data Type	Data
[Identification]	Device Nickname
[Location]	Latitude
[Location]	Longitude
[System]	Battery Level
[Sensors]	[Atmospheric] Temperature
[Sensors]	[Atmospheric] Humidity
[Sensors]	[Soil] Temperature
[Sensors]	[Soil] Moisture
[Sensors]	[Light] Ultraviolet Light (UV)

TABLE 2

This table shows all the data sent by the device to the back end. Parameters such as battery voltage and the device's geolocation coordinates are sent on top of the sensor data.

```
  "soilTemp":20.56,
  "soilHumidity":40,
  "atmoTemp":22.12,
  "atmoHumidity":62,
  "uvLight":0.00,
  "irLight":257,
  "deviceName":"device1"
}
```

The device then burns the buffer formatted to the SD card attached to the shield. I would recommend using a card with 16 GB or less to prevent errors with the shield. I have decided to save the data to the SD card as a backup to the cloud, thereby providing another source with the data other than the dashboard.

The MKR GSM then creates a HTTP client and connects to the Soracom Beam entry point. The device sends the data compiled into the buffer to the cloud. The data is sent through the GSM network and then received by the back end (**Table 2**).

I powered the device with a Li-Po battery (3.7 V). This seems to keep the device running for weeks on the field. To preserve energy, the device goes into a sleep mode between read cycles and wakes up at set intervals of time to collect data from the sensors. When the device wakes up and takes the samples, it will also read the voltage from the battery and send it to the cloud.

THE BACK END

I have mentioned Soracom a couple times earlier, but now let's get into more detail. Soracom is a platform built for IoT. It focuses on GSM and SigFox. The platform allows you to buy a SIM with a pay-as-you-go subscription and connect it to the cloud. Soracom (the company) also offers a bunch of great tools for data processing and analyzing. I used three of its services in this project: Soracom Air, Soracom Beam and Soracom Lagoon. I'll also be touching on Soracom Harvest and Soracom Funnel. The main benefit of using the Soracom platform is its wide variety of services and ease of use.

Soracom Air and Soracom Beam form the entry point for the data you have sent from the device. When the sensor data is sent by our device on the field, it is sent to Soracom's server by Soracom's GSM network: Soracom Air. The data is then input into the back end by Soracom Beam. This service then stores the data in a temporary database and allows other services to query it.

The data is then queried by Lagoon from Beam. Lagoon offers a dashboard that can be accessed by the end user (**Table 3**). The data is queried periodically from Beam and illustrated on maps, graphs and tables (**Figure 4**). Lagoon also allows the end user

to receive notifications via email if data is not within a defined threshold.

PROJECT CONSTRUCTION STEPS

Step 1: Required Materials: I have compiled a list of components needed to construct the project. And **Figure 5** shows of a photo of these components. You can find them in online stores. If you have any questions, feel free to contact me. You will also need a soldering iron to connect the components.

- Arduino MKR GSM 1400
- Arduino MKR MEM Shield
- Soracom Air SIM Card

- LiPo Battery (3.7 V)
- GY21 Temperature and Humidity
- SI1145 UV Light Sensor
- Capacitive Soil Moisture Sensor
- DS18B20 Waterproof Temperature Sensor
- Jumper Wires
- SD Card
- GSM Antenna

Step 2: Connecting the Circuit: I would solder the components together, but you can use a breadboard if it fits the enclosure you want to use for the device. A Fritzing “schematic” of the circuit is shown in **Figure 6**. You should place the MEM shield on top of the

Widget	Description
Map	The map widget illustrates the location of the device with a dot on the map. The location of the device is extracted from the data sent by the device in the payload. Lagoon checks if the soil humidity is below a threshold and will plot a red point if it is and a green point if the humidity is appropriate.
Line Graphs	Line graphs are used to plot the sensor data collected from the sensors over time. The end-user can easily monitor the progress of a current drought and its effects on the crops through the dashboard.
Table	The table lists all the data received from the device with the timestamp.
Alerts	Alerts can be set up by the end-user, this allows the end-user to be notified via email if certain variables are not within a set threshold. For example, send an email if the soil temperature is above 35°C.

TABLE 3

This table provides a short description of the elements on the Soracom Lagoon dashboard and what their roles are.

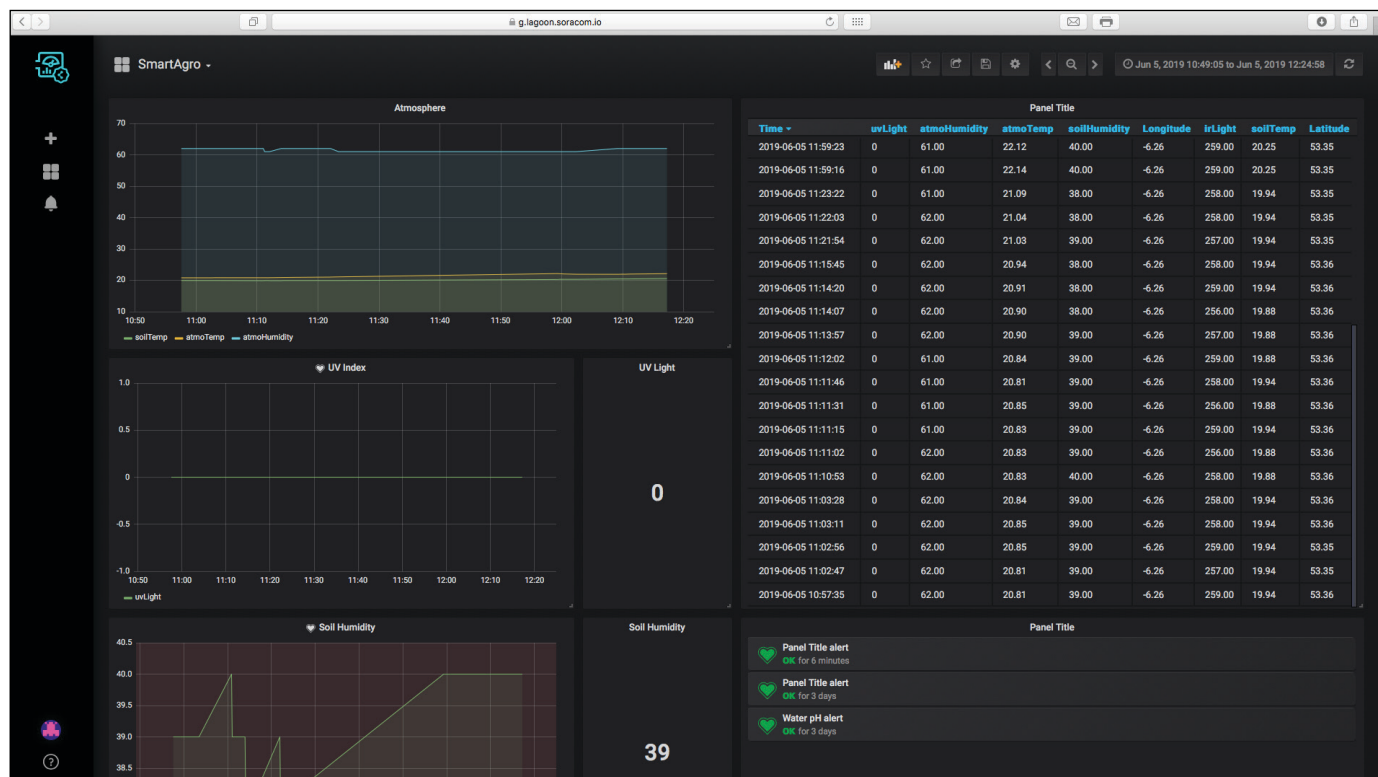


FIGURE 4

Shown here is a screenshot of the Soracom Lagoon dashboard. Multiple graphs, graphs and tables are visible.

MKR GSM and use its pins. The pins on the shield are connected to the equivalent ones on the MKR GSM. Finally, attach your Soracom SIM card and antenna to your device and you are ready to get started.

Step 3: Code: The device that is placed on the field will execute the same code every time

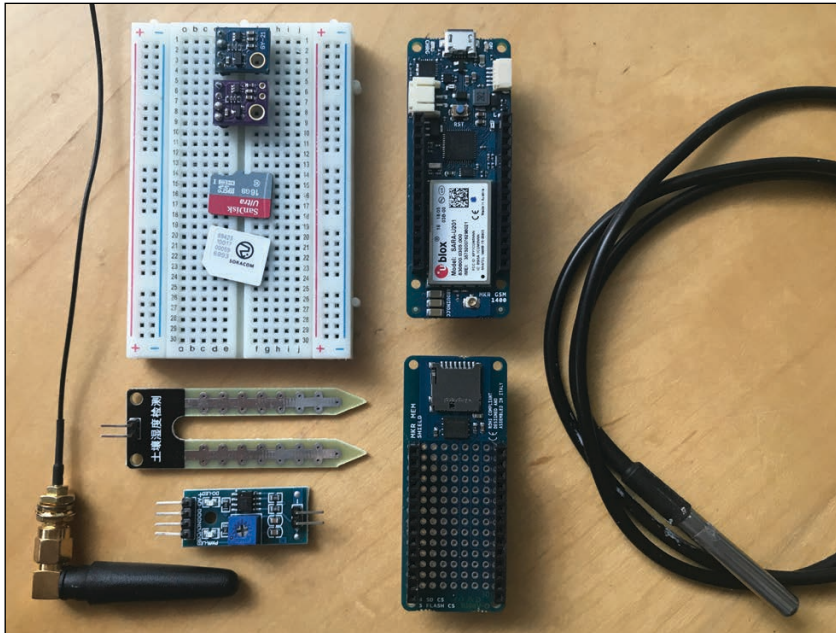


FIGURE 5

This is an image with all the components (excluding the wires) that you will need. Note: I am not using a capacitive soil moisture sensor, but I would advise you to use one.

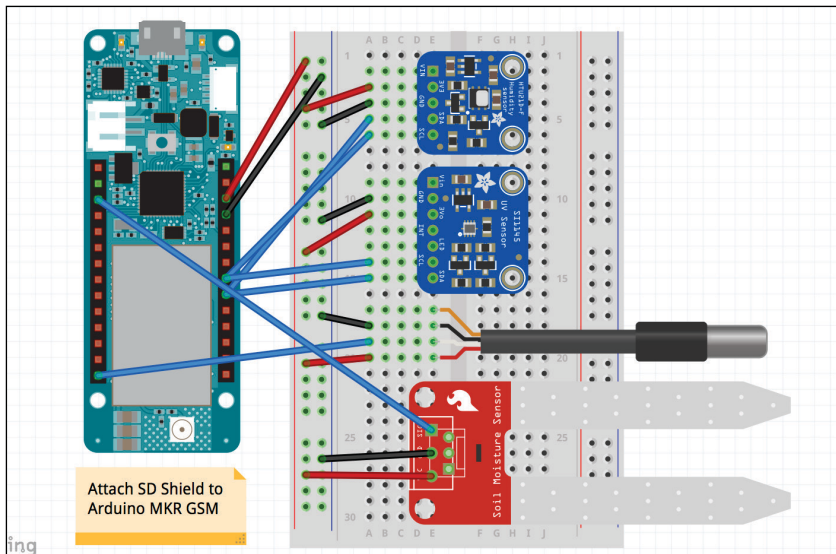
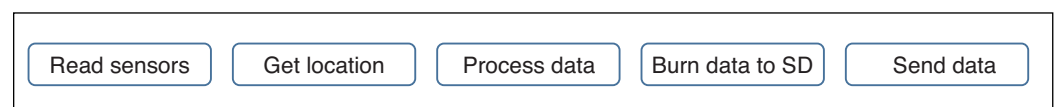


FIGURE 6

Shown here is a “schematic” of the circuit. I prefer showing this breadboard view because it's more colorful and easier to understand.

FIGURE 7

Shown here is the code overview, illustrating the steps taken by the device when taking samples.



it wakes up. **Figure 7** is a diagram illustrating the sections of code on the device and the order in which they are executed. Let's look at each one. *Read Sensors* - the first step taken by the device after it wakes up. The device collects data from the attached sensors with the use of multiple libraries. *Get Location* - the device then gets its geolocation coordinates from the GSM network. The data is verified to ensure that the co-ordinates are accurate. *Process Data* - data is parsed into buffers, one buffer will be saved to the SD card attached while the other is sent to the back end. *Save (burn) Data to SD* - the device then burns the buffer to the SD card. *Send Data* - the data is finally sent to the Soracom server through GSM.

There are multiple variables in the code that have to be changed by users to their preference. These variables are described in **Table 4**. Ensure that you edit these variables before flashing the code.

Step 4: SD Card: As mentioned before, the device will burn the data to an SD card as a backup source. The SD card has to be formatted with MS DOS to allow the MKR GSM to interface it. GoPro provides a great tutorial for doing this on a Mac or Windows. A link to the tutorial is provided on *Circuit Cellar's* article materials webpage.

There are a lot of projects that append data to text files on SD cards. I wanted the data to be easily visualized, so I decided to append the data to a CSV file. The device automatically creates the file on the SD card if there is not one there. If a file is available, the device will append to it.

Step 5: Setting Up Soracom: Now you get to the exciting part: the Soracom platform. You need to go through multiple steps before the back end is up and running. If you do not have an account yet, you can follow a video created by Soracom that will guide you through getting your account ready and registering your SIM. The video is posted on *Circuit Cellar's* article materials webpage.

After you have the SIM registered, you will want to create a new group, which you can then connect to other services. To do this, ensure you are on the main page of your Soracom account and then identify your SIM card from the list and check the checkbox beside it. Then click on the Actions button above the list and select Change Group. Click on the drop-down menu on the pop-up and then select Create and give the group a name. Now click on the menu icon in the top-left of the screen and select

Groups from the menu. You should be able to see your group!

Before you move any further, you will need to flash the code to the device and have it run for a few cycles. This is done to populate the back end with data so you can use it when creating our dashboard. Ensure that the circuit is built and all the sensors are wired accordingly. Set the time to sleep to 2 seconds to send the data quickly to the back end.

Next, you have to ensure that the data gets to the back end. To do this, open Soracom Console again and select Groups from the menu. Click on your group on the next window and then navigate to the Soracom Harvest option and click on it. Ensure it is on.

Now you can return to the SIM Management

page and check the SIM's checkbox. Then press on the Actions button and click on Harvest Data. If you have successfully sent data to the back end, you should see the graph and table populated. If no data is visible, try refreshing the page or continue sending data through the device. Feel free to contact me if you cannot get it working.

SORACOM LAGOON

Please note that you will need to play around with the different options on the page. There is no point in my dictating exactly which graphs to use and where to put them. In this section I will walk you through the graphs I would use and resources that can help you build your dashboard (**Figure 8**). But

Variable	Description
deviceName	This is the custom name given to the device. Any characters can be used to name the device. The name is particularly useful if the device is used in a network with multiple devices, this easily identifies the device the data came from.
sleepTime	This variable dictates the time in milliseconds the device sleeps between reads. I would personally take samples every half an hour or every hour.
proDebug	This is used for debugging. If you are testing the device, you should set it to true. proDebug enables serial prints but requires a serial connection for the device to work. Setting it to false will also allow the device to print to the serial monitor though the connection is not mandatory.

TABLE 4

Listed here are the variables in the code that have to be changed by the users to their preference. These should be edited before flashing the code.



FIGURE 8
The populated graph and table with the JSON data sent by the MKR GSM

FIGURE 9

As seen in this image of SmartAgro, the soil sensors are trailing beneath the device into the soil.




first, to open Lagoon click on the Menu icon in the console and select Data Storage and Visualization -> Soracom Lagoon. You will then be asked to create an account. Do so and add yourself as a user. You can choose any of the three plans presented.

Open the Soracom Lagoon console when you are ready. You will be directed to a menu, create a new dashboard. Your data source will automatically be added to the dashboard. To help you with the dashboard, there's a very useful guide published by Soracom—it's what got me started. The guide explains how to set up any type of graph and is extremely helpful. A link to the guide is provided on *Circuit Cellar's* article materials webpage. I would include maps, line graphs and tables to visualize the data. The table can be published and personal access can be granted to individuals. Everything you need to know is in the link.

Finally, now that the project is working, it would be time to create an enclosure for the device. To be honest this is my weak point. I personally built mine from a piece of piping and some wood. But I'm sure you can do better than that. **Figure 9** shows an image of SmartAgro with the soil sensors are trailing beneath the device into the soil. Just one more quick note: ensure that the UV sensor is not covered by anything so that the data will be precise. Place the soil sensors about 5 cm to 10 cm beneath the soil.

GOING FURTHER

This project can be improved in multiple ways. I personally like the idea of having multiple devices working in a network allowing data collection over a vast area. Soracom provides a service known as Soracom Funnel. This allows you to relay the data received from the device(s) to a third-party database. This would enable you to collect big data from a big number of devices and store it in a database like Amazon's S3. You can then query the data from the database and display it on a dashboard or a mobile application using Amazon's services.

For example, you can display data from the devices on a dashboard using AWS (Amazon Web Services) QuickSight which is a service for data visualization. This would allow the end-user to get more reliable information related to the entire agricultural surface. It is also possible to create an automatic irrigation system by having the device on the field send data to Soracom, which relays it onto AWS. The data is then retrieved using a callback by another device that controls the irrigation system and turns the irrigation on if the moisture is low. 

ABOUT THE AUTHOR

Andrei Florian is a student in Dublin, Ireland. He has been working on tightening the connection between humans and technology by designing applications that will help us in our lives. This includes working on projects that combat pollution and climate change as well as monitoring our natural environment and our cities. He has also been working on personal security and big data. Andrei can be contacted at andreialexanderflorian@gmail.com

For detailed article references and additional resources go to:
www.circuitcellar.com/article-materials

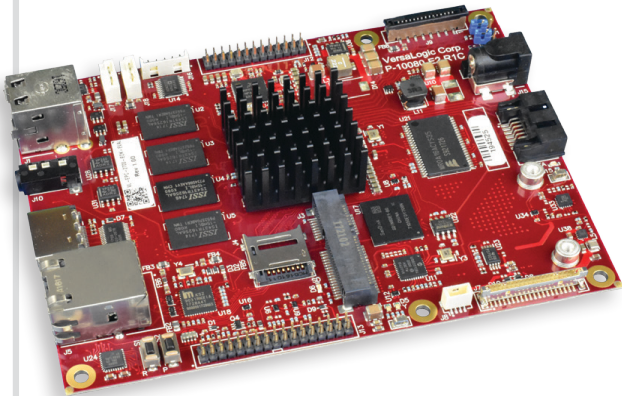
RESOURCES

Arduino | www.arduino.cc

Soracom | www.soracom.io

Enter to Win!

Android Demo/Eval Kit



Visit circuitcellar.com/versalogic

The kit includes:

- 7" 1024x600 HDMI Touch-screen flat panel display
- Tetra Single Board Computer (SBC) with Quad-core i.MX6
- Pre-loaded Android (Oreo 8.0) on MicroSD card
- Wall power adapter
- USB Hub
- Start-up guide
- Required cables

"Set-up was a breeze,
it was up and running
in less than 10 min."

– Electronics Technician



VERSALOGIC
CORPORATION

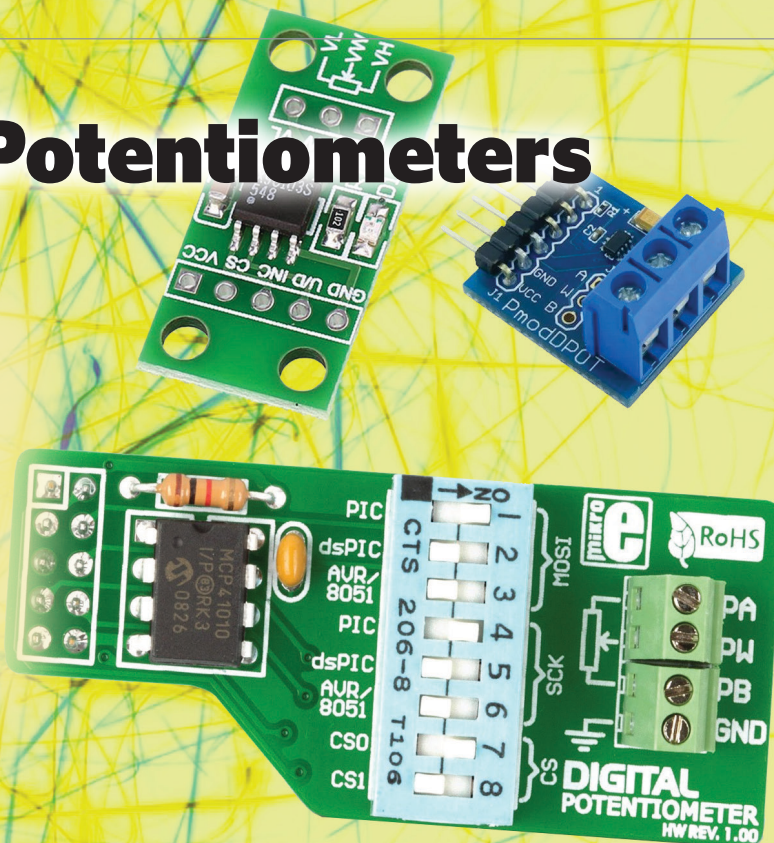
Using Digital Potentiometers

Convenient Control

FEATURES

A digital potentiometer probably can't be considered the most glamorous of electronic components. But it is easy to use and versatile. In this article, Stuart digs into the uses, advantages and disadvantages of digital potentiometers, including how they contrast to mechanical potentiometers.

By
Stuart Ball



One good entry point into the understanding of digital potentiometers is to contrast them alongside mechanical potentiometers. So, let's start there. If you wanted a volume control on a headphone amplifier, you might use a circuit similar to the one shown in **Figure 1**. A mechanical potentiometer connects to the audio input from some signal source, such as a radio or computer. The other end of the potentiometer connects to ground. The wiper connects to the input of the audio amplifier.

The potentiometer has a knob, and as you turn the knob, the wiper moves across a resistive element between the audio input and ground. When the wiper is at the ground end of the resistive element, the amplifier gets almost no signal, and the volume is off. When the wiper is at the other end of the resistive

element, the amplifier gets the full signal amplitude, and the volume is at maximum. Between the two endpoints, the volume is attenuated. A stereo system would typically have two potentiometers on the same shaft, so the volume in both channels is the same.

The mechanical potentiometer has some drawbacks. The most obvious one is the need for one knob for every function to be controlled. If you have volume plus tone controls for high, midrange and bass, then you need three more knobs. If you have volume plus an eight-band stereo graphic equalizer, then you need a total of 17 knobs, one for volume and eight equalizer knobs for each stereo channel.

Another issue with mechanical potentiometers is that the wiper must maintain contact with the resistive element. As the knob is turned, the wiper slowly wears away the material on the resistive element, leaving dead spots where no contact is made. If you've ever heard a scratchy sound as you turn a volume knob, that is usually the result of the potentiometer material wearing away in spots.

The issue of a worn potentiometer element also causes the wiper to be momentarily open or disconnected as the knob is turned. In an audio circuit, this causes undesirable noise. In other circuits, such as those involving motor control, this can cause more serious consequences.

DIGITAL POTENTIOMETER

Figure 2 is a block diagram of a digital potentiometer, or "digipot," connected to a microcontroller (MCU). Like a mechanical

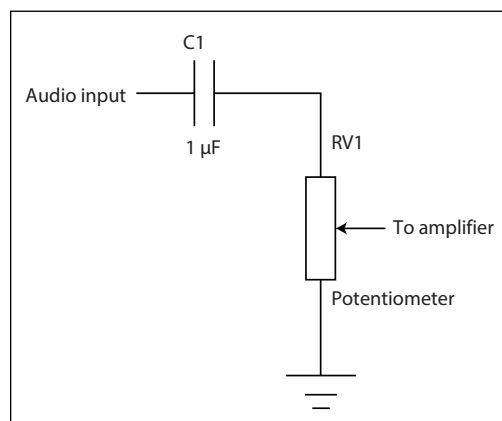


FIGURE 1

A simple mechanical potentiometer used as a volume control

potentiometer, the digital potentiometer has two ends (A1 and B1) and a “wiper” (W1). But the digital potentiometer doesn’t have a single resistance element with a moveable wiper. Instead, it is a series string of resistors with electronic switches that select taps in the string (**Figure 3**). For an 8-bit, 256-position device, there are 256 internal switches and 256 taps in the resistor string. The position of the “wiper” is defined by a digital word, controlling which switch is turned on and which tap is selected. It is controlled electronically by the MCU. In Figure 2 the potentiometer uses the standard SPI interface, though other interfaces such as I²C are available.

Showing an example circuit, **Figure 4** is a schematic of a Texas Instruments (TI) TM4C1233H6 MCU [1] connected to an Analog Devices AD8400 digital potentiometer [2]. In the schematic, the wiper output of the potentiometer drives (through op-amp U4) the base of a transistor that acts as a constant current driver for an LED. The digital potentiometer is an 8-bit device, meaning that the wiper moves between the B1 and A1 values as the control byte ranges from 0 to 255.

The upper terminal of the potentiometer is connected to 3.3 V, and the lower terminal is connected to ground. As a result, the wiper varies between these two voltages as the control value varies from 0 to 255. That voltage drives the transistor base through U4. The current through the LED is approximately equal to the current through the emitter, which is:

$$\frac{\text{base voltage} - 0.7 \text{ V}}{160 \Omega}$$

or about 16 mA when the base is 3.3 V.

U4 is a voltage follower, and provides sufficient drive current to make sure the base of Q1 equals the wiper voltage at W1. Without U4, the wiper of the digital potentiometer could directly connect to the base of the transistor. However, the emitter resistor, R7, when reflected back into the base, would load the digital potentiometer, reducing the base voltage. The circuit would still work, but the LED brightness at any given setting would be lessened.

The combination of U3, U4, and Q1 makes a simple DAC (digital-to-analog converter) that converts the potentiometer setting to an LED current. If you remove Q1, D3, and R7, the output of U4 would be a simple 8-bit voltage DAC with an output range from 0 to 3.3 V. This is not the most efficient way to make a DAC, but it illustrates the principle.

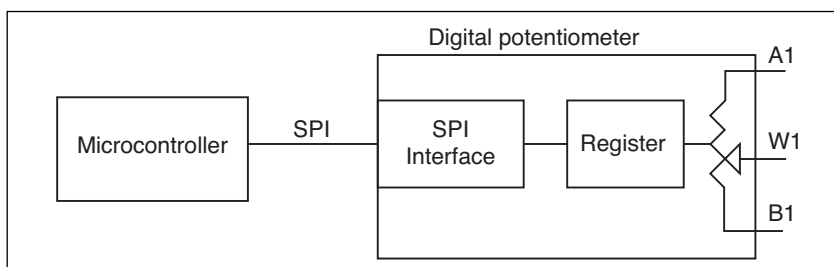


FIGURE 2

Block diagram of a digital potentiometer.

MCU FIRMWARE

The MCU firmware slowly cycles the digital potentiometer between about 0.7 V and 3.3 V, which causes the LED to brighten slowly over the space of about 2 seconds, then turn off and start over. The lower bound of the potentiometer is set at 0.7 V to match the forward bias voltage of the transistor. If you were to connect an oscilloscope to the transistor base, you would see a sawtooth waveform as the voltage slowly ramps up and then suddenly resets to the 0.7 V lower bound.

The example circuit could be implemented with any MCU, but the TM4C SPI connections are well-suited to parts like the AD8400. The AD8400 is one of a family of parts (AD8400/8402/8403) with one, two, or four digital potentiometers in the package. Although the control word to set the potentiometer value is only 8 bits, two address bits are needed to select which potentiometer in the package is being written.

In the case of the single-channel AD8400, the address bits are always zero, but the extra

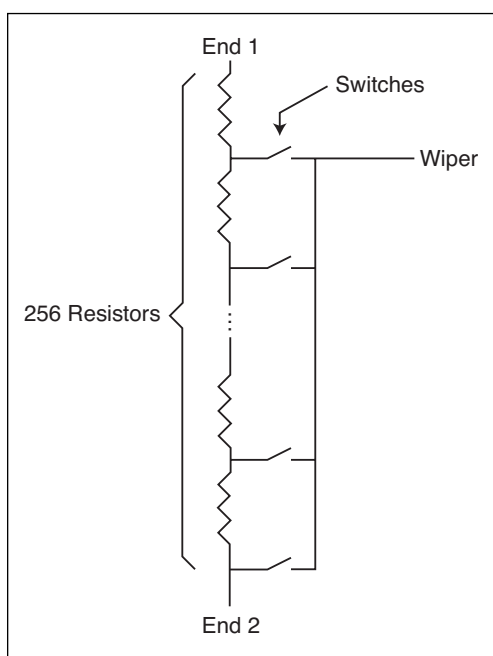


FIGURE 3

Electrical implementation of digital potentiometer

bits make the control word 10 bits wide. The SPI interface on some MCUs only supports 8-bit operation. The SSI (synchronous serial interface) on the TM4C MCU—used to implement SPI—has a programmable word width from 4 to 16 bits. This makes it easy to connect to SPI devices with word widths that are not multiples of 8 bits. For an MCU with an SPI interface that only supports multiples of 8 bits, you could also use a digital potentiometer with an 8-bit SPI interface, such as the AD5160, or implement the SPI interface in the firmware by toggling the interface pins. This circuit could also be implemented by wiring a board with U3, U4, Q1, D3, and R7, and connecting it to a TI TM4C Launchpad board.

J2 is a programming connector and uses an external RS232 converter to connect to the host serial port. This was described in my article “Debugging Embedded Systems with Minimal Resources” (*Circuit Cellar* 312, July 2016). Shorting W1 on start-up enables programming. **Figure 5** is a schematic of the converter.

ADVANTAGES OF DIGITAL POTENTIOMETERS

Although a digital potentiometer requires extra circuitry, including an MCU, there are several advantages to using one.

No mechanical wear: The digital potentiometer doesn't have a mechanical wiper, so there is no mechanical wear on the resistive elements.

Combined functionality: A digital potentiometer can be combined with other parts of a circuit to enhance functionality for the user. For example, if you have a Bluetooth-connected phone in your car, the car's audio system can mute the radio when the phone rings, and unmute it when the call is completed. You don't have to manually turn down the volume to answer the phone. If your car has a Bluetooth phone capability, it probably works exactly this way.

Fewer manual controls: Again, using the car as an example, you can have anything from a three-band tone control for the radio, or even a multiband graphic equalizer, all without extra knobs. The tone control

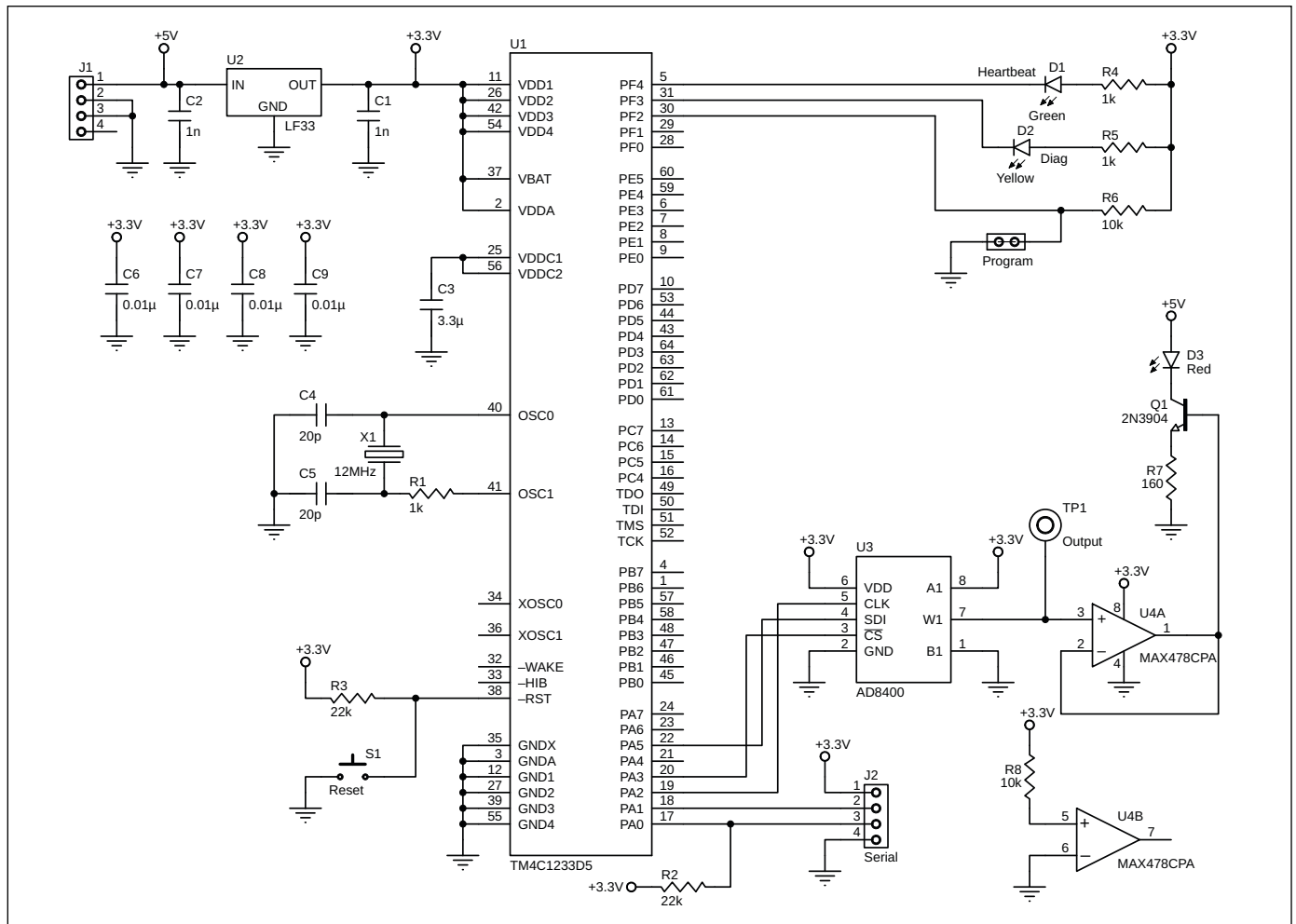


FIGURE 4

Schematic of a microcontroller-operated AD8400 digital potentiometer [2]

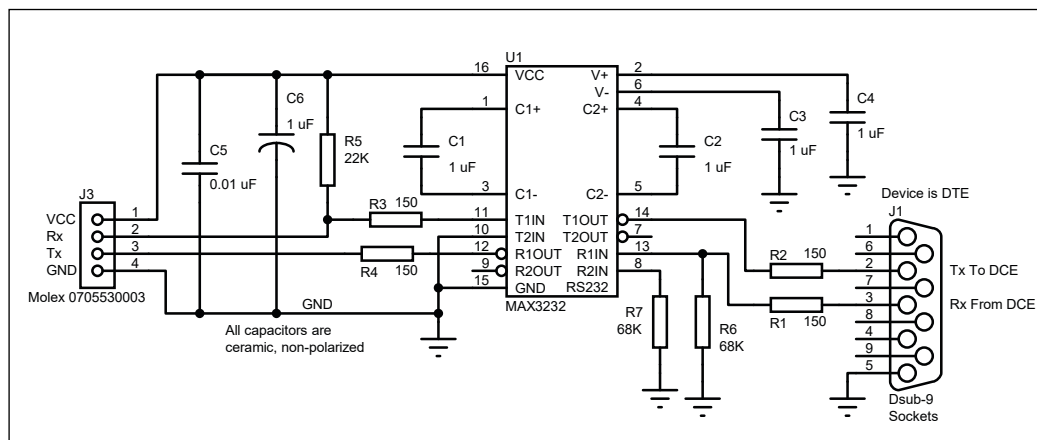


FIGURE 5
Programming adapter converts RS232 to logic levels.

adjustments can be made from the display in the console, with the MCU adjusting a digital potentiometer for each function.

Multi-use controls: Because the volume and tone controls are electronic, you don't need one knob for each function. Instead, a single knob can be used to control multiple functions. For example, when using the phone, the volume knob might control audio volume to the phone, without affecting the radio volume when the call ends and the radio is unmuted. The volume knob can be context-sensitive and can control functions that are completely unrelated to volume. When in the clock-setting mode, the volume knob can be used to adjust hours or minutes. When adjusting those tone control values, the volume knob can be used to adjust the selected tone band. Because the MCU is adjusting the various digital potentiometers and other things, the knob can be used for any number of different functions.

Logarithmic potentiometers: Mechanical potentiometers used in volume control applications are usually logarithmic, matching the response of the human ear. Although the AD8400 used in the example is a linear device, there are digital potentiometers with logarithmic response. It is also possible to drive a linear potentiometer with logarithmic values, though this requires additional calculations or a look-up table in the MCU firmware. See Sidebar "Making a Linear Potentiometer Logarithmic" for details.

DIGITAL POTENTIOMETER LIMITATIONS

As with most design decisions, digital potentiometers also have some drawbacks.

Voltage limitations: Besides the obvious need for an MCU to control the device, the input values are limited. A mechanical potentiometer can be driven with any voltage, as long as the power and voltage limitations aren't exceeded. But for the switches to work, the three resistance terminals of the digital

potentiometer can't go beyond the supply rails. In the example circuit, that means the A1, B1 and W1 pins can't go below ground or above 3.3 V. This can be an issue with audio circuits, which tend to be AC-coupled and, therefore, swing between a positive and negative voltage.

Figure 6 shows how you might solve that problem using a digital potentiometer. Instead of connecting the lower resistor terminal to ground, it is instead connected to a 1.5 V reference diode. A 10 μ F capacitor bypasses the reference, so that the lower terminal is effectively at ground for AC signals. The net result is that the upper terminal is at 1.5 V with no signal applied.

The circuit in Figure 6 fixes the issue of the AC signal driving below ground, but we're not out of the woods yet. The amplitude of the input signal has to be below 3 V peak-to-peak (PP). If the input is driven with a 4 V_{pp} signal, the peaks will drive the upper terminal beyond the supply rails. The moral of the story is to be sure that the terminal pins not

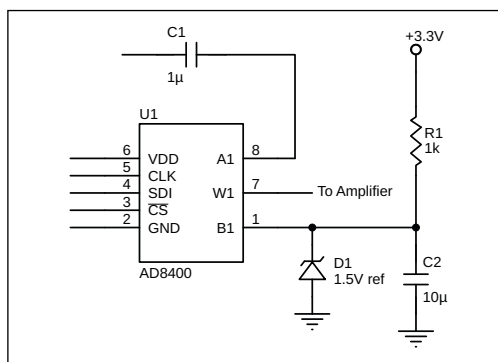


FIGURE 6
A 1.5 V reference solves the problem of coupling an audio input to the digital potentiometer.

ABOUT THE AUTHOR

Stuart Ball is a registered professional engineer with a BSEE and an MBA. He has more than 30 years of experience in electronics design. He is currently a principal engineer at Seagate Technologies.

only aren't biased beyond the supply rails, but also aren't driven beyond the supply rails over the full range of circuit operation.

The circuit in Figure 6 is relevant to another problem—for a mechanical potentiometer, the three terminals can effectively be isolated from

everything else. A mechanical potentiometer could be used to adjust something in a 24 V circuit, as long as the ratings aren't exceeded. In contrast, the digital potentiometer would fry in that kind of application, unless you floated the entire circuit—MCU and all—at the 24 V level.

Memory: A mechanical potentiometer retains its position when power is removed. Therefore, when you turn the circuit on, the setting of the potentiometer is wherever you left it. If you turn the volume knob all the way down while the amplifier is turned off, then when you turn it on, the volume will be turned off.

The AD8400 used in the example circuit doesn't have any non-volatile memory. When you turn the power on, the wiper goes to some random location between 0 and 255. If you want to remember the last position of the wiper, you must store it in MCU EEPROM or some other non-volatile memory. Some digital potentiometers, such as the Microchip Technology MCP4161T, have internal non-volatile memory to store the most recent resistance setting. But you should make sure that your design doesn't do something unexpected the first time you turn it on, even with a non-volatile part.

Current: Digital potentiometers have much lower current capacity than that of mechanical potentiometers. The AD8400 used in the example has a maximum continuous current across the two endpoint terminals of 2 mA. Clearly you aren't going to use this part to handle any significant amount of power. A mechanical potentiometer also has a maximum power rating, but it is much higher than for a digital potentiometer. In the example circuit, I could have eliminated U4 and used a 1K version of the AD8400 directly driving the transistor Q1, but the current would be 3.3 mA, exceeding the 2 mA continuous current rating of the device.

Tolerance: Like mechanical potentiometers, digital potentiometers don't have tight resistance tolerance. The AD8400 has about 20% tolerance, though the matching between values is closer to 1%, if using a version with multiple resistors in one package. A few parts, such as the AD5291, have precision as good as 1%. In some applications, the end-to-end resistance tolerance is less important than the ratio of the wiper position to overall resistance.

Audio clicks: Digital potentiometers switch between different resistance values using solid-state switches. For many applications, this is not an issue. But when a digital potentiometer is used as a volume control, it is possible to hear an audible click when the switches open and close. One way to

MAKING A LINEAR POTENTIOMETER LOGARITHMIC

A linear potentiometer has a linear relationship between the "position" of the wiper and the attenuation. The attenuation is proportional to the wiper position. If the wiper (mechanical or electronic) is at the halfway position, the attenuation is 50%, or 6 dB below the maximum value. However, the response of the human ear isn't linear, but rather logarithmic.

For a logarithmic 8-bit potentiometer, the relationship is:

$$\text{Output amplitude (in dB)} = 20 \log\left(\frac{\text{input amplitude} \times \text{wiper position}}{256}\right)$$

Control Word	Output Attenuation
0	Output off
1	-48 dB
2	-42 dB
4	-36 dB
8	-30 dB
16	-24 dB
32	-18 dB
64	-12 dB
128	-6 dB
255	0 (full on)

TABLE 1

For the AD8400, you could have 9 steps of 6 dB each with the resistance control values shown here.

Audio volume changes are measured in decibels (dB). To make a 256-step, 8-bit linear potentiometer into a log potentiometer, you don't use all 256 positions. Instead, you use resistance (wiper) values that result in a constant separation in dB. So, for the AD8400, you could have 9 steps of 6 dB each, with the resistance control values shown in **Table 1**.

If you calculate the midpoint, 128, using the logarithmic formula, you will see that it is $20 \log(.5)$ or -6 dB from maximum volume. Doing the same calculation with the next step, 64, you will find it is -12 dB below maximum.

You can do something similar to get 3 dB steps, but the more steps you want to have, the less accurate the logarithmic scale will be. A conversion of 3 dB per step with 16 steps in the AD8400 would have values like this:

0, 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 45, 64, 90, 128, 180, 255

If you do the math for the low end of the scale, you find it isn't very close to 3 dB steps. But at very low volume levels, it usually matters less.

For detailed article references and additional resources go to:

www.circuitcellar.com/article-materials

References [1] through [4] as marked in the article can be found there

RESOURCES

Analog Devices | www.analog.com

Maxim Integrated | www.maximintegrated.com

Microchip Technology | www.microchip.com

Texas Instruments | www.ti.com

avoid this is to only allow the volume to be changed on zero-crossings of the input signal. This makes the switch transitions inaudible. Some digital potentiometers, such as the Maxim Integrated's MAX5408, are designed specifically for volume control applications, with built-in zero crossing detectors. Mechanical potentiometers do not have this issue. However, as they wear, they do have a similar issue with noise on dead spots.

OTHER USES FOR DIGITAL POTENTIOMETERS

I have focused on audio applications for the digital potentiometer because they are easy to understand, but there are other applications as well. A digital potentiometer can be used to electronically compensate a bridge circuit. Coupled with a thermistor or other temperature-measurement component, a digital potentiometer can be used to compensate for temperature variation in a temperature-sensitive circuit.

A digital potentiometer in the feedback path of an op amp circuit can be used to control gain or other parameters. **Figure 7** shows how the AD8400 could be wired to control the gain of an op amp. A pair of digital potentiometers can be used to tune a low-pass or high-pass audio filter [3]. Analog Devices has an

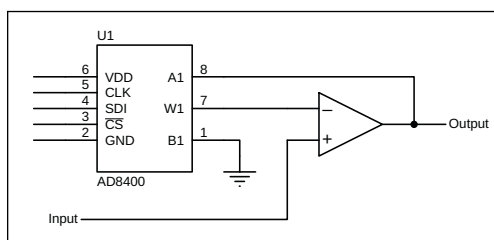



FIGURE 7

Using a digital potentiometer to control gain of an op amp.

application note describing the use of digital potentiometers to control the frequency and gain of a Wein Bridge oscillator [4].

Here, I have concentrated on applications using an MCU. Digital potentiometers such as the AD5228 have inputs for two buttons. One button steps the wiper up, the other steps it down. However, many of the advantages of an MCU-based design cannot be realized with these parts, unless the up/down inputs are connected to an MCU. For that reason, I omitted discussion of these parts from this article.

CONCLUSION

The digital potentiometer isn't a glamorous component, but it solves a lot of problems. Hopefully, this brief tutorial has provided some insight into the benefits of digital potentiometers and the things you need to plan for if you use them in your designs. 



LOOKING TO
**ADVANCE
YOUR
CAREER?**

Be a part of one of the
**top Electrical Engineering
programs in country**
and experience the
Bearcat Promise!

University of
CINCINNATI | **ONLINE**

online.uc.edu

Fall registration is
open now

The Laser Harp

A Twist on Modern Instruments

FEATURES

Normally, you'd think that taking the strings out of a harp would be a downgrade. But in this article, Cornell student Alex Hatzis presents a system that does just that—replacing the harp strings with red lasers. Phototransistors are used to detect when the beams are intercepted by a person's hand playing the harp, and some convincing real-time sound synthesis helps to create a new, high tech instrument.

By
Alex Hatzis

My lab partner, Glenna Zhang, and I had the opportunity to work together on a few final projects during our time studying electrical and computer engineering at Cornell University. Even though those projects varied wildly in both design and complexity, we always tried to tie-in one thematic constant, something for which we both share a passion: music. The application of technology has already given way to a number of new ways to produce music. The

unique sound of the electric guitar would not be possible without its amplifier, and computer synthesizers can replicate most instruments with remarkable accuracy, or can be used to produce entirely new sounds. Despite these advancements, we both feel that there is plenty of untapped potential for technology to change how we produce music. Our final project for the course "ECE 4760: Designing with Microcontrollers," is now dubbed the "Laser Harp," and it is one such attempt to tap into that potential [1].

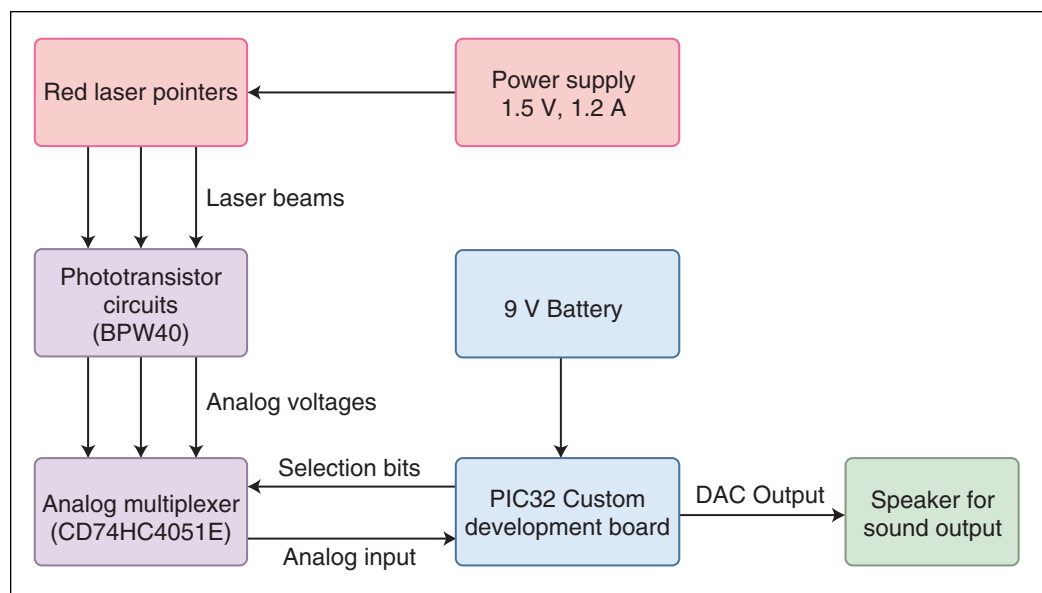


FIGURE 1
Block diagram of harp subsystems

Anyone equally interested in the world of technology and music will probably realize that our project is not the first device to be called a “laser harp.” While researching the idea, we found that laser harps already take on several different forms. The most common is a single unit placed on the ground that beams several coplanar, outward-fanning lasers toward the sky.

Our idea took closer inspiration from a display at a local museum, the Ithaca Sciencenter, which featured a traditional harp frame without any strings [2]. However, if you ran your hand through the middle, sounds would play as if the strings were still present. Our project was our own take on this example. At the highest level, our harp can be broken into a few different subcomponents: the laser emitter and receiver subsystem, the string control/sound synthesis subsystem and the physical harp frame.

Our laser harp is extremely intuitive to use. A block diagram of the harp subsystems is shown in **Figure 1**. When powered on, the eight laser pointers built into the top of the frame illuminate. They emit visible red light, but the beams themselves are not visible without the use of a fog machine or something similar. The harp is small enough to rest comfortably in one’s lap (**Figure 2**). This is how we designed it to be played—instead of resting on the floor like a typical full-size harp. Each of the lasers is carefully aligned to point at its own dedicated phototransistor circuit, which allows us to tell whether or not the corresponding laser pointer is currently shining on it. These lasers represent the strings of the harp.

In the current implementation, each of the eight “strings” corresponds to one of the notes in a C major scale. By running your hand through the harp, the laser beams are intercepted, and the phototransistor circuits register this. The control system interprets which strings are being plucked, and produces the correct waveform to be played through a speaker connected externally to the harp using a standard 3.5 mm aux cable. The sounds produced have been designed to closely replicate plucked strings, and the result is quite pleasing to hear.

Not all the design choices we ended up with were obvious from the beginning, and the changes we made in response to issues we found along the way are responsible for a few of the most distinctive aspects of the final design.

HARDWARE DESIGN

Our initial design for the harp didn’t specify the use of lasers, because the display we were inspired by at the Ithaca Sciencenter

didn’t have any lasers visible. Rather, our goal was simply to have invisible or intangible strings. Our first idea, and the first step we took in this project, was to try ultrasonic or infrared range sensors. This seemed like the obvious choice, because, even using units with mediocre accuracy, we would be able to easily register the presence of a hand in the harp. The issue arose when we began experimenting with the region of sensitivity, which, with most sensors of this type, extends outward like a cone. This meant that closer to the sensors, they were quite precise in telling whether a hand or finger was in front of them.

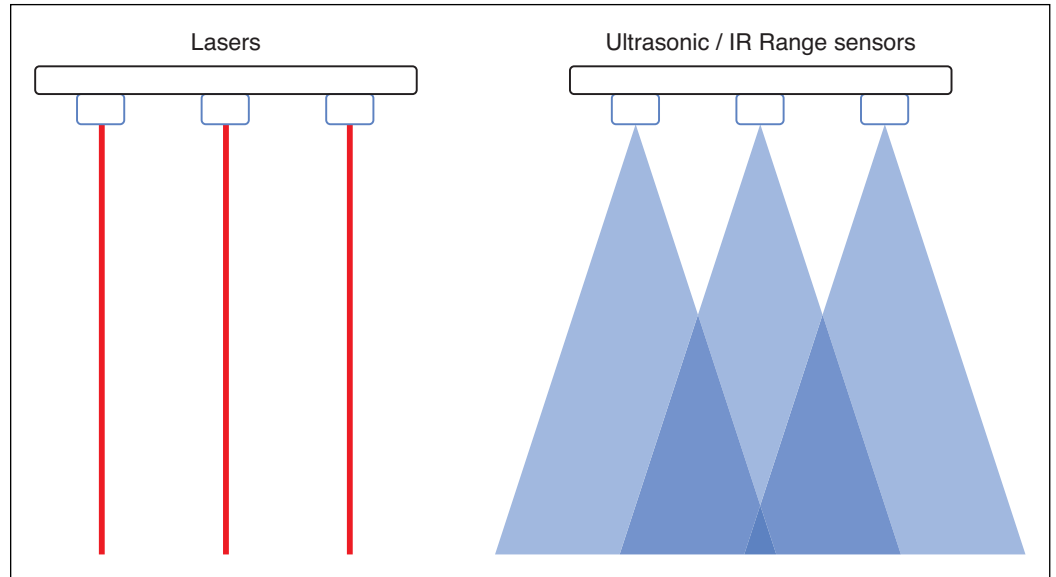
As you moved farther away from the sensors, however, the cone where the sensor could “see” became wider. For many



FIGURE 2
Finalized harp frame

FIGURE 3

Conceptual comparison of different sensor detection areas



applications, this would not be an issue, but when trying to emulate several slim strings running parallel to one another, this made differentiating among those strings very difficult. A person trying to play the middle string, for instance, might accidentally end up tripping the sensors both to its right and left. A conceptual comparison of different sensor detection areas is shown in **Figure 3**.

It was this concern that led us to begin experimenting with laser pointers. The pointers we chose were originally designed for pets, so they are not very high power. But their beam divergence was easily small enough for the dimensions of our harp. Each pointer we acquired was typically powered

with a single AA battery, but the additional length added to accommodate the battery made it difficult to fit the pointers into the frame. We used pipe cutters to remove the section of the pointers dedicated to holding the battery, and soldered wires to the leads so they could be connected to a single power supply (**Figure 4**). Each pointer runs on about 150 mA at 1.5 V, and since we wired the entire string of eight in parallel, the line runs on 1.2 A at 1.5 V. In hindsight, this was not an optimal setup and required higher-gauge wire, but the power supplies in our lab still worked with this power draw.

PHOTOTRANSISTORS

The challenge then became that these lasers had no inherent way, as range sensors did, of telling if their beams were being intercepted, so a separate system was required for checking this. For this purpose, we opted to use BPW40 phototransistors, which have a maximum response around 780 nm light. This is slightly further than ideal from the 650 nm emitted by the lasers, but when properly aligned with the beam, we found the response from the phototransistor was still enormously pronounced. The change in the circuit's voltage was so strong that we could almost treat it as a digital signal, but we still ended up working with it as an analog voltage for robustness. It did, however, make it much easier to differentiate between when the light on the phototransistor was on or off.

Each phototransistor was built into an extremely simple circuit with Vcc (supply voltage) connected to its collector terminal, and a 1 k Ω resistor to ground on its emitter terminal. When light shines on the phototransistor, current flows through it and the resistor, causing the voltage across the

**FIGURE 4**

Laser pointer wiring scheme

resistor to rise. This means that a high voltage at the node between the phototransistor and the resistor corresponds to a string not being plucked, since the beam is not being broken. Consequently, when a beam is intercepted, comparatively little light shines on the phototransistor, very little current flows and the voltage across the resistor falls. We correlate this with a string being plucked.

The full circuit schematic is shown in **Figure 5**. Each node between each phototransistor-resistor pair is connected to an eight-to-one, two-way analog multiplexer, the CD74HC4051E from Texas Instruments (TI). Analog multiplexers are much slower than their digital counterparts, but because the strings are used only for user input, this was not an issue. All eight phototransistor circuits are multiplexed to the same single ADC (analog-to-digital converter) on our microcontroller (MCU) board, which further simplified our design later on. We used simple I/O (input-output) pins to toggle the three address bits on this

multiplexer and sequentially cycle through the phototransistor circuits—checking them one at a time. Using this method, the MCU was able to read more or less continuously from a single ADC, to determine which strings were being played and which were not.

We used a PIC32 MCU on a custom development board designed for our class, ECE 4760. If our software found that one of the polled strings had been plucked by the user, then the PIC would output the waveform corresponding to that string (or multiple strings) using the DAC (digital-to-analog converter) present on the development board. That DAC output would then be fed through a very simple, passive, RC low-pass filter with a cutoff frequency of a few thousand hertz. This filter was intended to reduce some of the quantization that is inherent in the DAC. Once past the filter, the signal was passed to a 3.5 mm audio jack, which could be used to play the sound on external powered speakers.

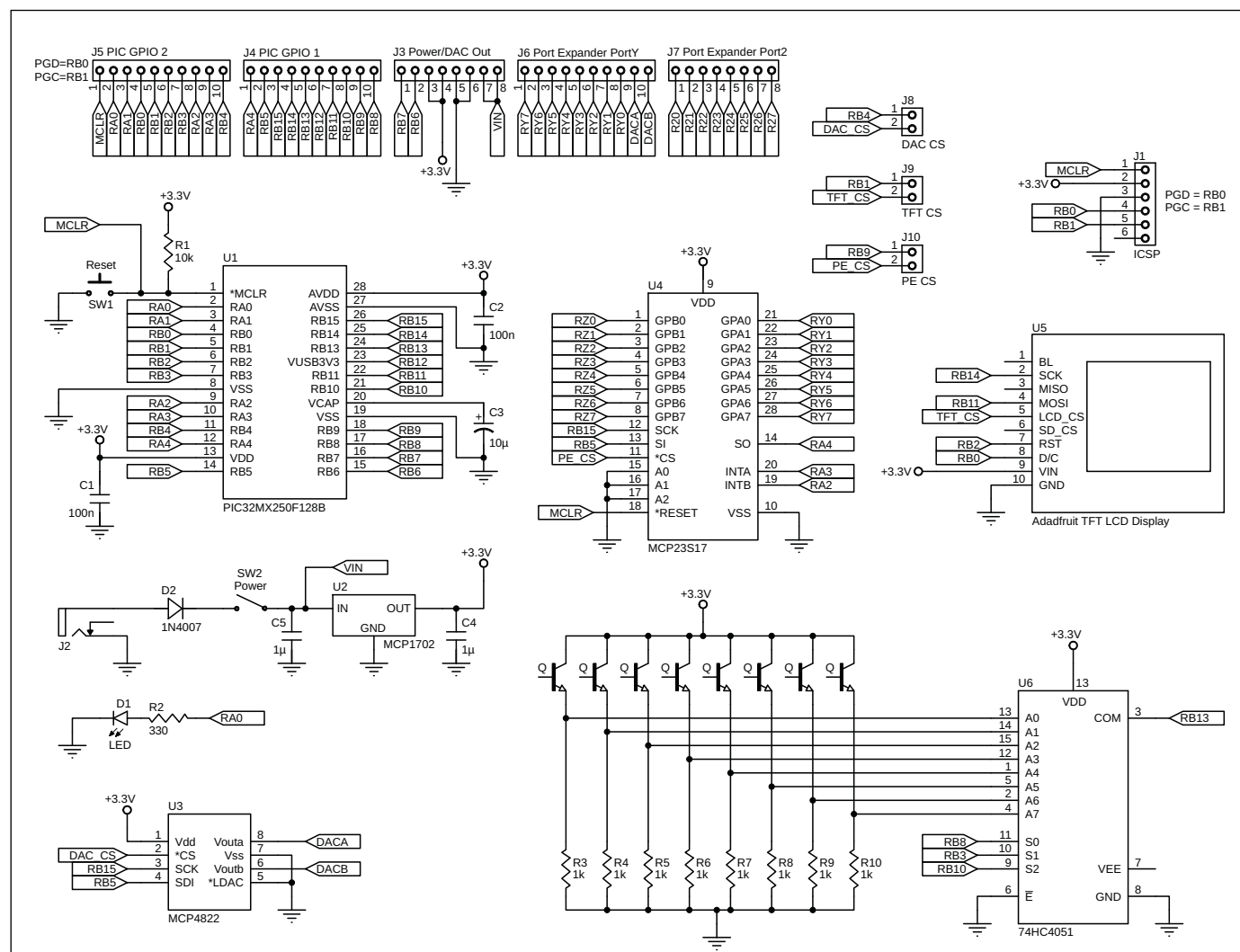


FIGURE 5
Full circuit schematic

STRUCTURAL DESIGN

Perhaps one of the most distinctive aspects of our project is its visual design. The harp has a handmade wooden frame that is designed to house all the electronics. Spending as much time as we did on the physical aspects of the harp was certainly not our original intent, but it made for a far better result than we had anticipated.

The first prototype we created was a cardboard box with holes at the top for mounting laser pointers, and spots at the bottom for affixing the phototransistors. We quickly realized that this type of frame, or any frame made from a weak material, would not be suitable for our needs. The main issue we encountered was with aligning the laser pointers to the phototransistors. The very reason we chose to use them in the first place—their precision—was making it nearly

impossible to set them up reliably. Not only did we have to align the laser pointers to have their beams land on the phototransistors, but also the phototransistors themselves have a sharp angular response curve and had to be angled precisely to get consistent, easily discernible data. The response of these particular phototransistors begins to fall off sharply after 15 degrees, and is almost nonexistent past 20 degrees.

These factors, combined, meant that the specifications we had laid out for our project demanded a much more robust frame. The frame would have to keep the parts aligned when stationary, and it should not wobble or flex significantly when held. We decided that wood was the easiest material to work with that met our requirements for strength and stability. We used a few pencil sketches and a rough idea of how large the harp should be to create a 3D CAD model of the current frame in Autodesk's Fusion 360 (**Figure 6**).

We had limited woodworking experience, so we decided it would be easiest for us to represent the harp as a series of two-dimensional pieces that we could cut out of plywood. We achieved this by projecting several of the profiles on the 3D model into a flat .dxf file (**Figure 7**). This type of file would normally allow us to cut out the pieces using a laser cutter, but the plywood we chose was too thick to allow this.

Instead, we used the profiles we generated as guides to cut out the pieces manually with a jigsaw, and used wood glue to assemble the pieces into the full frame. We sanded the sides where the different pieces did not align well, and added wood filler, mainly to help with the aesthetics, but also to make the frame more comfortable to hold. The last step was to spray paint the entire frame gold, which again was for purely aesthetic reasons.

With the frame fully assembled (pre-assembly pieces shown in **Figure 8**), we were able to install the eight laser pointers into the topmost arch, which has a gap in it designed to house them. From here, they shine down to the trapezoidal sound box. Once the lasers were fixed in place, we drilled holes for the phototransistors in the sound box, permanently aligning them with the laser pointers. The rest of the electronics also fit into the trapezoidal section of the frame, making for an extremely clean final design.

Sanding and painting the frame, along with packing all the electronics inside it, made for a much more enjoyable user experience. The harp was actually quite easy and pleasant to use as a result of several of the functional decisions we made during the assembly process.

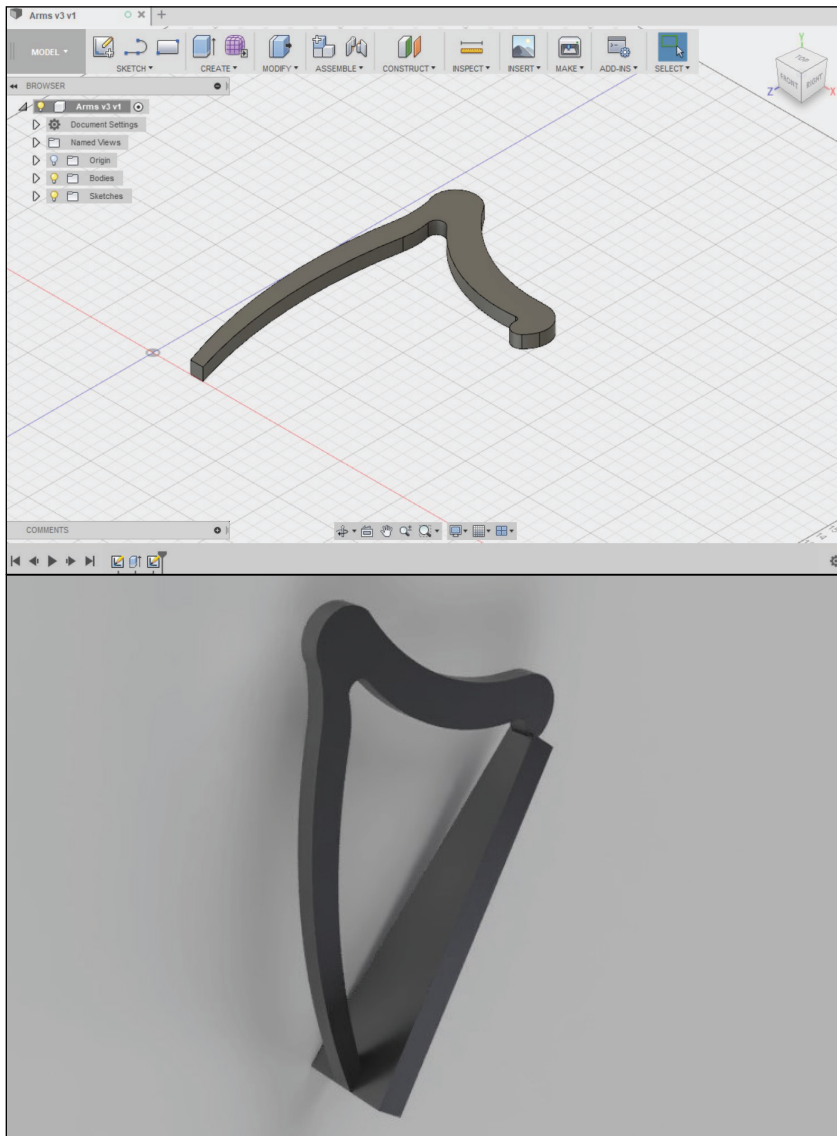


FIGURE 6
CAD renderings of the harp frame

SOFTWARE DESIGN

Many small components in the code ensure our harp is working as intended, but the most important parts can be broadly divided into two sections: string polling and sound synthesis.

String Polling: The string polling section's most basic task is exactly what the name implies—it has one thread dedicated entirely to repeatedly checking the state of the laser strings. It uses the I/O pins RB3, RB10 and RB8 to set the address bits and toggle through the eight inputs to the analog multiplexer, reading the value passed through to the ADC for each address. The way this thread is written, it reads the value on the ADC before updating the address I/O pins and yielding from the thread. This gives the ADC more time to set up during the time that the thread is yielded, ensuring that each reading is accurate. When any of the strings is determined to be plucked, the polling thread sets a variable that lets the sound synthesis section of the code know which note to play. The code checks one string every millisecond, so all eight strings are checked every 8 ms, for a polling rate of 250 Hz. This is comparatively low, but more than sufficient for reading human inputs, while leaving enough CPU time for the real-time synthesis.

Sound Synthesis: The sound synthesis section is the key to our harp's ability to create a convincing string sound in real time. We used the Karplus-Strong algorithm to synthesize our sounds. Karplus-Strong is a sort of "shortcut" for simulating an oscillating string using the wave equation [3].

The wave equation itself is a second-order, partial differential equation, which can be used to describe wave-like behavior in any situation by relating several of the features of the oscillating system. If we want to accurately model sound on a string, then the wave equation is the place to start.

$$\frac{\partial^2 y}{\partial x^2} = \frac{1}{v^2} \times \frac{\partial^2 y}{\partial t^2}$$

The wave equation in this form describes the relationship among the curvature, vertical acceleration and horizontal velocity of the wave. The details of how to use or solve this equation are beyond the scope of this article, but the important takeaway is that using a solution to this equation, we can model a vibrating string.

The algorithm takes advantage of the fact that one possible solution to the wave equation is a travelling wave on a string:

$$y(x,t) = A \sin \frac{2\pi}{\lambda} (x - vt)$$

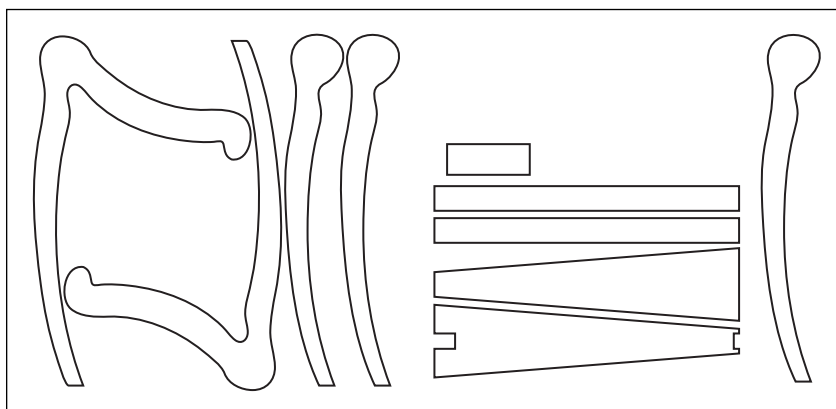


FIGURE 7

The .dxf files generated for cutting out sections of the frame from plywood

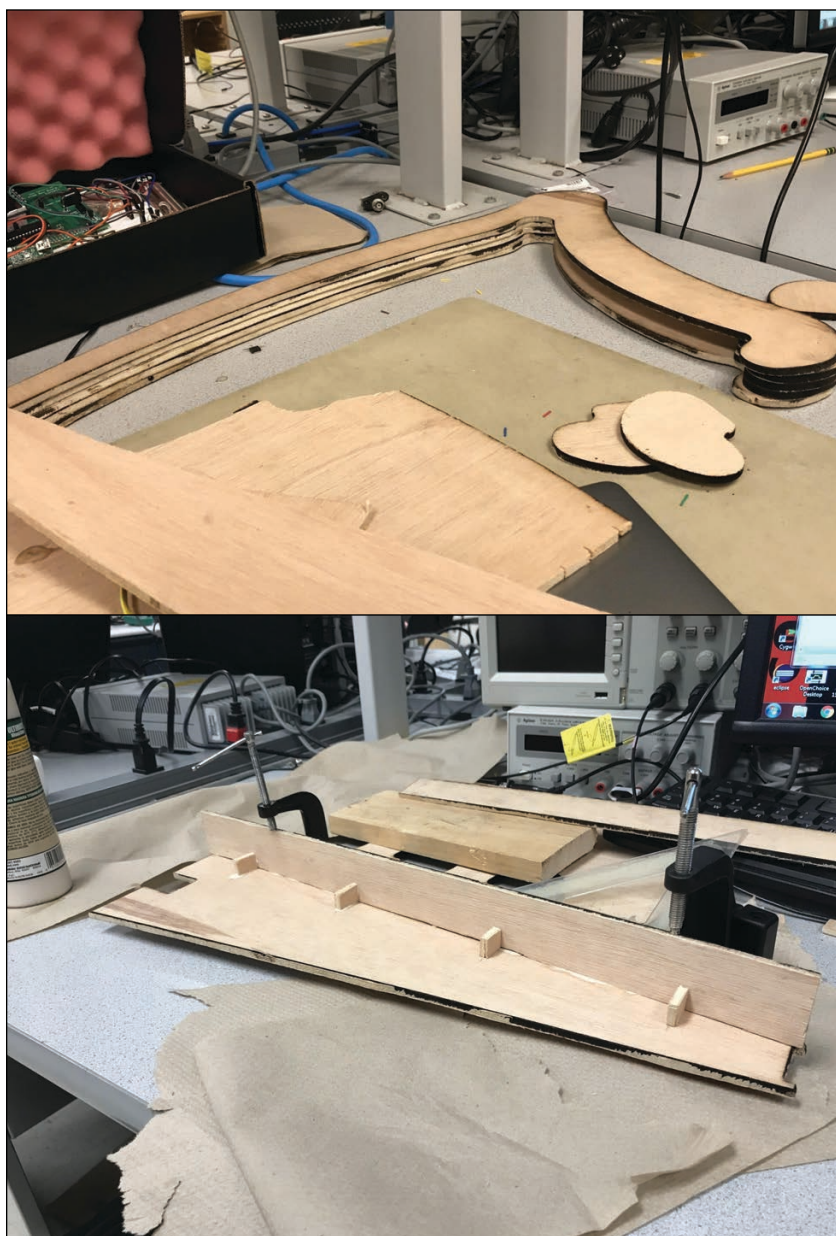
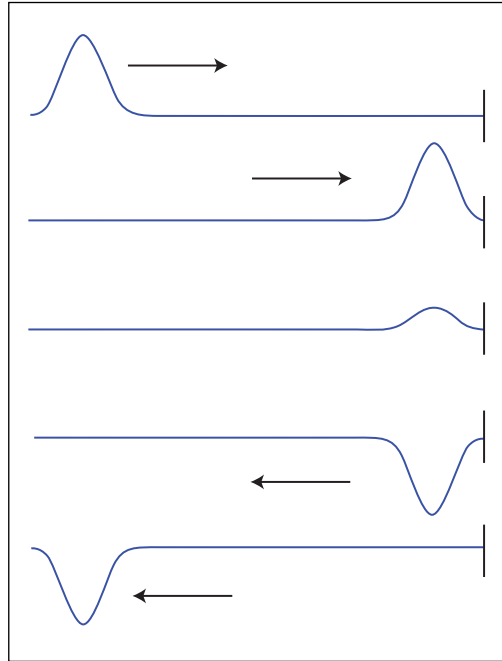


FIGURE 8

Two views of pieces of the wooden harp frame during assembly

FIGURE 9

Travelling wave reflection at a fixed end



or

$$y(x,t) = A \sin \frac{2\pi}{\lambda} (x + vt)$$

Notice that the $\sin()$ function makes the wave equation periodic. When a wave travels along a string with fixed ends (that is, the ends do not move, as would be the case in a harp or other string instrument), it reflects and inverts upon reaching the end of the string (Figure 9).

We represented this behavior in the code using a circular buffer shift register, where the length of the circular buffer is analogous to the length of the string. Using

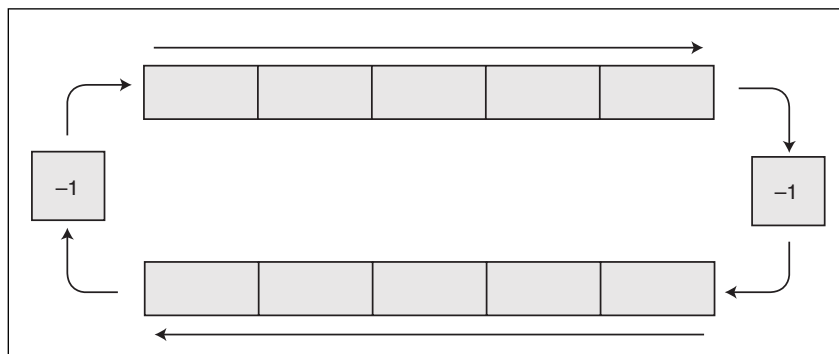


FIGURE 10

Circular buffer used to represent a travelling wave with reflections at the ends

ABOUT THE AUTHOR

Alex Hatzis is a Cornell class of 2020 student pursuing a BS in Electrical and Computer Engineering. His main interest is in embedded systems design.

this technique, we replicated the periodic nature of the travelling wave. We started with two buffers travelling in opposite directions, with a factor of -1 applied at both ends to represent the waves being inverted at a fixed end of a string (Figure 10). Any real string has a certain “resonant” frequency. When a string is plucked, energy is added to it in the form of noise comprising all frequencies. All other frequencies die out as the resonant one lingers, which gives each string its distinct tone. The resonant frequency for each of our simulated strings is defined by the length of the buffer. Because a larger buffer means that it takes more time for the travelling wave to traverse it, a larger buffer corresponds to a lower frequency.

To begin implementing this in the code, we set up a hardware timer interrupt to run at 20 kHz, which is our synthesis frequency. The buffers shift their entries each time the interrupt from the timer occurs. These multiplications at the ends of the buffer commute to form a single buffer, cycling in one direction without any inversions. To set the length of the buffer for a certain note, we divide our sampling frequency by the frequency of the note we want to play. At a high enough sampling frequency, this circular buffer could always have an integer number of cells, but for most typical synthesis frequencies—in our case 20 kHz—we require a fractional number. For example, to play a C4 note (261.6 Hz) at a synthesis rate of 20 kHz, we would need 76.45 cells. This means that in addition to the 76 integer locations in our circular buffer, we needed a fractional time delay of some kind to achieve the required 76.45 cells, to have tonal accuracy.

Finally, low frequencies die out slower than higher ones on a string, so a low-pass filter replicates this characteristic (Figure 11). Ideally, this happens to every cell at every time step, but only applying it to one transition in the circular buffer—when it loops back to the beginning—is not audibly different to a human listener, because it still happens so fast, and since mathematically these operations can commute without issue.

THE ISR

In the actual ISR (interrupt service routine), we do not constantly resize the buffer (represented by an array) to change the note played or shift all the entries in the buffer. Instead, we simply move two pointers every time the ISR is called, and a separate variable tracks when those pointers should roll back to zero. We change it as necessary, effectively setting the length of the buffer. At this loop back, we first low-pass filter the

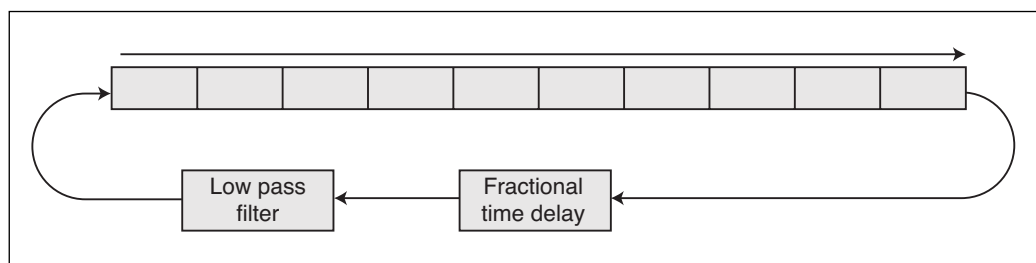


FIGURE 11
Full circular buffer with reflections commuted and additional filters added

signal by taking the difference between the previous and current index, multiplying it by a low-pass coefficient, and then adding that value to the current index. This shifts it downward if the previous index was lower, or upward if the previous index was higher, thereby smoothing the waveform.

The next step implements the fractional time offset. The goal of the code in this section is to create a phase-shift all-pass filter, where the specified phase shift results in the required fractional delay. It takes the difference between the previous index and the current one, multiplies it by a corrector value specific to the note and corresponding to the necessary fractional component, and adds it to the previous index, to obtain the corrected current index. This corrector value does not depend on the length of the

string. Instead, each time it loops through the entire buffer once, it correctly delays the waveform. This results in accurate frequency readings with a tuning app, and the notes sound correct to the ear as well. The overall string sound is quite convincing as well. You can listen to a sample of the results in our project video using the QR code in **Figure 12**.

In the final implementation of this ISR, we used fixed-point arithmetic and a few small optimizations to synthesize four strings simultaneously, with each string using about 20% of the CPU. This means that at every timer interrupt, all these steps happen separately for four different strings.

If a string is determined to have been plucked by the string polling section, then another variable is set to add drive to the plucked string. After the drive has been



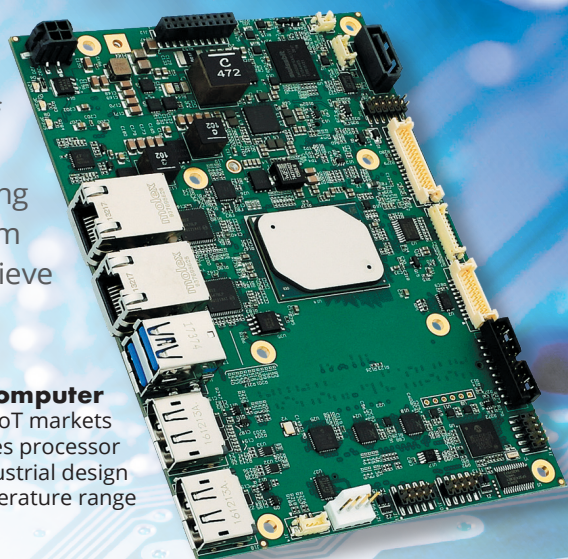
FIGURE 12
Use this QR code to link to a video showing the Laser Harp project. The video is also available on Circuit Cellar's article materials webpage.

EMBED RELIABILITY

Rugged, Reliable Embedded Computing Solutions

WINSYSTEMS understands the risk and challenges of bringing new products to market. We help technology decision makers select the optimal embedded computing solutions to reliably enable their products, giving them more time to focus on product feature design and achieve faster times to market.

SBC35-427 Small Form Factor Single Board Computer
For industrial control, transportation, energy, and industrial IoT markets
Latest generation Intel® Atom® E3900 series processor
Rugged industrial design
-40°C to +85°C operating temperature range



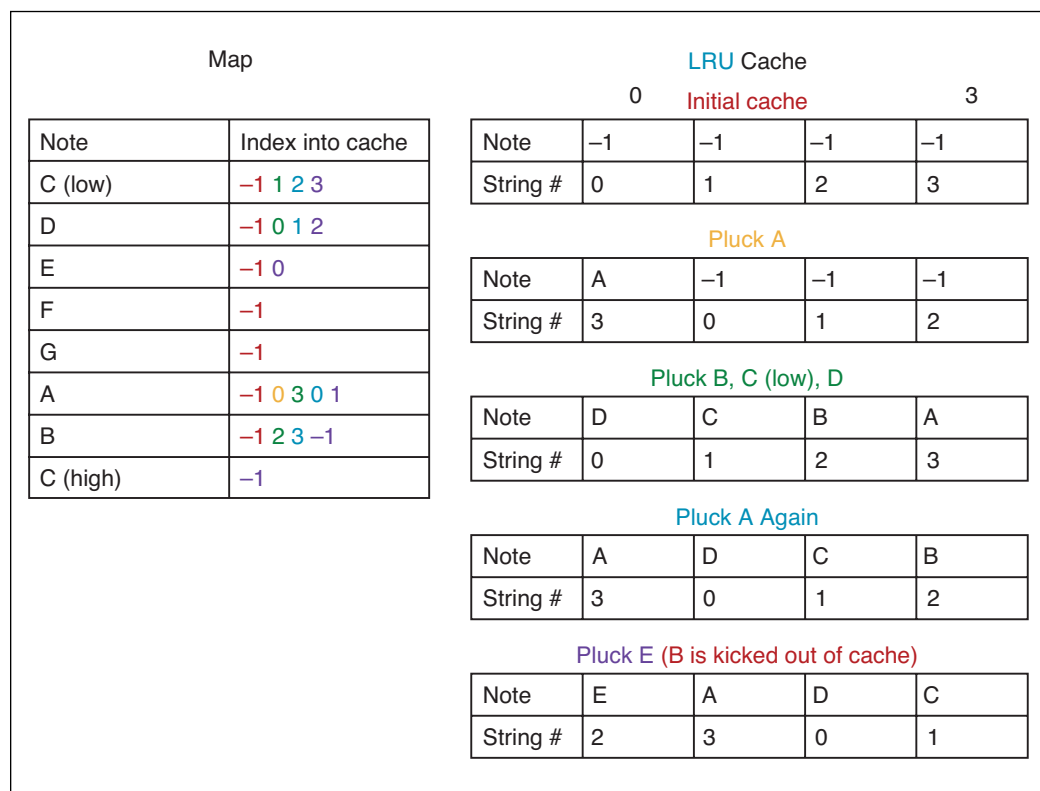
www.winsystems.com | Telephone: +1 817-274-7553

QUALITY CERTIFIED
ISO 9001:2015



FIGURE 13

Example of LRU cache and cache map usage. The colored numbers in the map on the left represent the value for each parameter during the step, with the corresponding color on the right.



added, vibrations continue to pass through the string until the damping coefficient causes them to die out. This creates a convincing string sound, ringing out for some time before dying out, independent of what the other strings are doing.

We encountered a snag at this point. Eight laser strings were built into our harp, but our MCU was limited to synthesizing only four strings simultaneously. We effectively had four “slots” that could be used for producing sound. The obvious solution seemed to be mapping two strings to each synthesis slot, so that plucking either the first or second laser string would play using the first synthesis slot, the third and fourth strings would use the second synthesis slot, and so on. The problem with this idea was that playing the two strings assigned to the same synthesis slot would

require one of them to stop playing, and it would be impossible for them to play simultaneously.

We also found that changing the note assigned to a synthesis slot produced an unpleasant audible cutoff sound, since this required stopping the sound altogether to change the size of the buffer. However, replucking a string already being synthesized did not have this issue, so we wanted to change the notes assigned to the synthesis slots as infrequently as possible. Additionally, we wanted to allow recently played notes to continue playing for as long as possible, so they could overlap realistically with other played notes, regardless of which strings were chosen.

LRU DATA STRUCTURES

To achieve these objectives, we created a system of data structures to resemble an LRU (least recently used) cache, as an expansion to the thread where we poll the laser string inputs. An LRU cache of size four, for example, remembers the four most recent unique items that were accessed. So, if you request item A, item A is also placed into the cache. If an item already in the cache is requested, that item is provided, and it becomes the most recently used. If a requested item is not already in the cache, either the least recently used item or the one that has been in the cache the longest without being used is removed, and the

For detailed article references and additional resources go to:
www.circuitcellar.com/article-materials
 References [1] through [3] as marked in the article can be found there

RESOURCES

Autodesk | www.autodesk.com

Microchip Technology | www.microchip.com

Texas Instruments | www.ti.com

newly requested item is added in the first spot. This type of priority structure is exactly what was needed for the strings and synthesis slots.

We started with the cache itself, which we implemented as a 2 by 4 array. Each column of the cache is a single entry, but for each entry we require two pieces of information—the string synthesis slot being used (ranging from 0 to 3), and the note currently assigned to that slot. If, for example, you play the notes A, B, C and D, they appear in the cache in the order of D, C, B, A. If you play A again, it moves to the front of the cache, changing the order to A, D, C, B. If you then play an E note, B is kicked out of the cache, and the cache becomes E, A, D, C.

In this example, while a note remains in the cache, the synthesis slot it uses remains the same. This minimizes the number of times we switch the synthesized notes, and also ensures that, when we do switch synthesized notes, we choose the least recently used synthesis slot. This plays more softly than any of the more recently played notes, minimizing how noticeable the cut-off sound is. It makes the harp sound more realistic, and also makes it simpler to change which notes are playing.


Finally, to see if a note is in the cache—and if so, where it is located in the cache—we have a map array. This maps each of the eight notes corresponding to the laser strings, if present, to its index in the cache. Otherwise it is mapped to the value -1. An illustration of how these data structures look after the aforementioned series of events is shown in **Figure 13**.

CLOSING REMARKS

The data structures used to control sound synthesis were, in some senses, overengineered for the tasks for which they were required. We even had presets left over in the code for several additional octaves of notes and half notes that were not currently implemented. These artifacts suggest how we originally intended the harp to have slightly more functionality. You could imagine how something like an octave key could be a logical addition to a harp like ours. There could even be some interactive way to set which notes are played by each string.

Most of our software was designed to be easily expandable for these features, but what really caused these trade-offs for us was the structural design and construction of the harp. Not only did the construction require a significant time investment on its own, but also our decision to package all the electronics neatly into the interior of the frame meant that adding an additional push button or something similar would require modifications to the frame. Nonetheless, adding such additional features is a logical extension of the project.

Even without these features, at a base level we achieved the goal that we set for ourselves during our project proposal. From a product standpoint it was even better than we had hoped. Our harp turned out to be an extremely enjoyable design challenge, as well as a product that is pleasant both to the eye and the ear.

Author's Note: My sincerest appreciation and thanks to my lab partner, Glenna Zhang, without whom this project would have been impossible to create and much less enjoyable to work on. 

IoT Hardware

Free Online Design in Minutes

Pretested, production-ready device ships in 15 Days



dream, design, deliver
geppetto.gumstix.com

Assortment of Tech Solutions Enable the Smart Home

IoT-Leveraged Living Spaces

SPECIAL FEATURE



Photo courtesy of Whirlpool®

From preventive maintenance for appliances to voice-controlled lighting, the subsystems that comprise a modern Smart Home continue to evolve. Providing the building blocks for these implementations, IC vendors are keeping pace with specialized MCUs, sensors platforms and embedded software to meet diverse requirements.

By **Jeff Child**,
Editor-in-Chief

The evolution of Smart Homes is about more than pure convenience. Smart Home technologies are leveraging IoT concepts to improve energy efficiency and security, thanks to intelligent, connected devices. The topic encompasses things like power-saving motor control systems, predictive maintenance, cloud-based voice assistance, remote monitoring and more.

Clearly the market is an attractive one. According to the latest Smart Home Device Database from market research firm IHS Markit, the global Smart Home market is forecast to grow by nearly a factor of five to reach more than \$192 billion in 2023, up from \$41 billion in 2018 (**Figure 1**). The report says that the fastest-growing device types in the market include lighting, smart speakers and connected major home appliances.

While it's impossible to cover all the bases of Smart Home technology in a single article, here we'll examine the microcontrollers (MCUs), analog ICs and special function chips that MCU vendors are developing to address Smart Home system designs.

AWARE APPLIANCES

An important piece of Smart Home technology is the idea of outfitting major home appliances with sophisticated maintenance features. With that in mind, in January Renesas Electronics launched its Failure Detection e-AI Solution for motor-equipped home appliances, featuring the Renesas RX66T 32-bit MCU. This solution with embedded AI (e-AI) enables failure detection of home appliances—such as refrigerators, air conditioners and washing machines—due to motor abnormality (**Figure 2**).

Property data showing the motor's current or rotation rate status can be used directly for abnormality detection, making it possible to implement both motor control and e-AI-based abnormality detection with a single MCU. Using the RX66T eliminates the need for additional sensors, thereby reducing a customer's bill of materials (BOM) cost.

When a home appliance malfunctions, the motor operation typically appears abnormal when running and being monitored for fault detection in real-time. By implementing e-AI-based motor control-based detection, the

failure detection results can be applied not only to trigger alarms when a fault occurs, but also for preventive maintenance. For example, e-AI can estimate when repairs and maintenance should be performed, and it can identify the fault locations. This capability provides home appliance manufacturers the means to boost maintenance operations efficiency and improve product safety by adding functionality that predicts faults before they occur in their products.

The solution uses the Renesas Motor Control Evaluation System and an RX66T CPU card. This hardware is combined with a set of sample program files that run on the RX66T MCU as well as a GUI tool that enables collecting and analyzing property data indicating motor states. In order to detect faults, it is necessary to learn the characteristics of the normal state. Using the GUI tool, system engineers can immediately begin developing AI learning and optimized fault detection functionality. Once the AI models are developed, the e-AI development environment (composed of an e-AI Translator, e-AI Checker and e-AI Importer) can be easily used to import the learned AI models into the RX66T.

MCU FOR HOME AUTOMATION

The kind of MCU technologies required for the Smart Home must be extremely small sized, and have low power levels suited to low battery life. Addressing such needs, in February Microchip Technology launched what it claims is the industry's smallest IEEE 802.15.4-compliant module that combines an ultra-low-power MCU with a sub-GHz radio. At half the size of the next smallest module on the market, the SAM R30 module is designed to meet the needs of space-conscious gear such as home automation sensors and controls (**Figure 3**).

Based on IEEE 802.15.4, the SAM R30 module supports proprietary networks that can be easily customized and configured. This is ideal for applications where interoperability is not desired due to their inherent vulnerability to remote attacks, such as alarm systems, building automation, smart cities and industrial sensor networks.

A key advantage of an IEEE 802.15.4-based network is that member devices can sleep for extended periods of time and remain part of the network. The SAM R30 module features ultra-low-power sleep modes, with wake from general-purpose I/O (GPIO) or its built-in real-time clock (RTC) while consuming

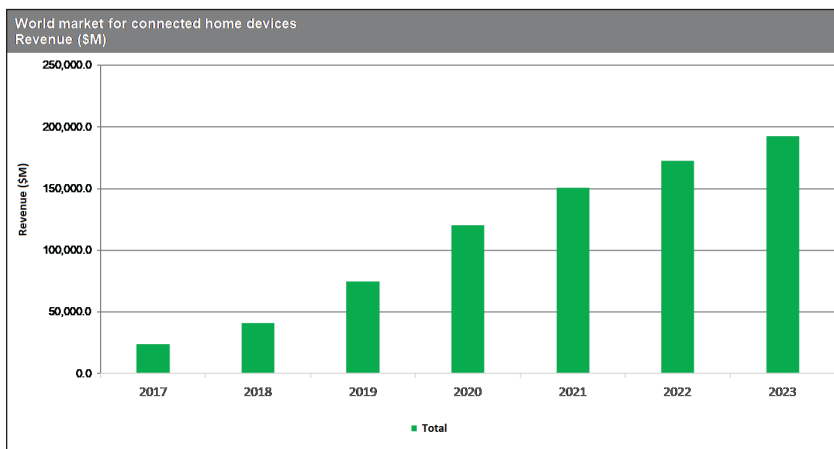


FIGURE 1

According to research from IHS Markit, the global Smart Home market is forecast to grow by nearly a factor of five to reach more than \$192 billion in 2023, up from \$41 billion in 2018.



FIGURE 2

The Failure Detection e-AI Solution with embedded AI (e-AI) enables failure detection of home appliances—such as refrigerators, air conditioners and washing machines—due to motor abnormality.



FIGURE 3

An ultra-low-power MCU with a sub-GHz radio, the tiny SAM R30 module is designed for space-conscious systems such as home automation sensors and controls.

approximately 800 nA. Devices can sleep for years, only waking as needed to transmit data.

Compact sensor designs can significantly reduce installation components and simplify device deployment. Applications often rely on simple double-sided tape for mounting, making decreasing the size of the sensor a priority. The SAM R30 module measures just 12.7 mm x 11 mm and includes the necessary features to drive any remote connected sensor, thereby eliminating the need for a separate MCU in the design. The device also offers up to 256 KB of flash and 40 KB of RAM, as well as serial data interfaces, USB and digital and analog I/O for advanced sensor development.

Operating in the sub-GHz RF spectrum, the SAM R30 module delivers two times the connectivity range and better propagation through walls and floors than similarly powered devices using the 2.4 GHz frequency band. This robustness is critical in applications such as leak detection, where the sensor may be buried deep in a remote cabinet, or for pool and spa controllers, which require reliable sub-GHz solutions that can communicate through exterior walls.

ALEXA FOR SIMPLE DEVICES

Voice control platforms are another key ingredient that define the concept of today's Smart Home. As voice recognition platforms such as Apple's Siri, Amazon's Alexa and Google's Google Assistant become more ubiquitous, Smart Home system developers have been eager to integrate those

technologies into their implementations.

Facilitating such needs, last year STMicroelectronics rolled out its X-CUBE-AVS software package that enables Amazon's Alexa Voice Service (AVS) to run on STM32 MCUs, allowing simple connected objects such as smart appliances, home-automation devices and office products to support advanced conversational user interfaces with cloud-based intelligence like automatic speech recognition and natural-language understanding (**Figure 4**).

As an expansion package for the STM32Cube software platform, X-CUBE-AVS contains ready-to-use libraries and open routines that accelerate porting the AVS SDK to the MCU. With application samples also included, it abstracts developers from the complex software layers needed to host AVS on an embedded device. AVS development usually targets more power-hungry and expensive microprocessors. In contrast, X-CUBE-AVS makes Alexa technology accessible to a wider spectrum of developers and projects.

The software handles low-layer communication and connection to AVS servers, provides application-specific services, and encapsulates the AVS protocol to ease application implementation. Connection management includes a persistent-token mechanism for directly restoring connection losses without repeated user authentication.

WAKE WORD SUPPORT

Also offering an Alexa-capable MCU solution, in February NXP Semiconductors unveiled an MCU-based voice control solution qualified with Amazon's AVS. Built on an NXP i.MX RT crossover platform, this MCU-based AVS solution enables low latency, far-field, "wake word" detection, embeds all necessary digital signal processing capabilities, runs on Amazon FreeRTOS and includes an Alexa client application (**Figure 5**).

This MCU-based AVS solution provides OEMs with a self-contained, turnkey offering that enables them to quickly add Alexa to their products. It includes the MCU, the TFA9894D smart audio amplifier, optional A71CH secure element and comes with fully integrated software. It also features noise suppression, echo cancellation, beam forming and barge-in capabilities that enable use in acoustically difficult environments.

NXP offers at its Mougins, Sophia-Antipolis facilities a product testing service for Alexa built-in products, available to its customers who



FIGURE 4

The X-CUBE-AVS software package enables Amazon's Alexa Voice Service (AVS) to run on STM32 MCUs, allowing simple connected objects such as smart appliances, home-automation devices and office products to support advanced conversational user interfaces with cloud-based intelligence.

want to test their devices before submitting to Amazon for final evaluation. If a system developer's product supports music and/or is far-field enabled and uses a "wake word" to initiate interactions with Alexa, additional testing is required prior to submitting products to Amazon for evaluation. This is where Pro-Support Audio Voice Services helps to complete the self-test checklists.

AUDIO AMPLIFIER SOLUTION

Audio amplifiers are another key technology for the voice control segment of Smart Home development. Along such lines, Texas Instruments (TI) offers digital-input Class-D audio amplifiers that enable engineers to deliver high-resolution audio in more Smart Home and voice-enabled applications. By combining high integration, real-time protection and new modulation schemes, TI's audio devices allow designers to reduce board space and overall BOM costs. These new amplifiers are designed for personal electronics applications with any power level, like smart speakers, soundbars, TVs, notebooks, projectors and IoT applications.

An example from this family of amplifiers

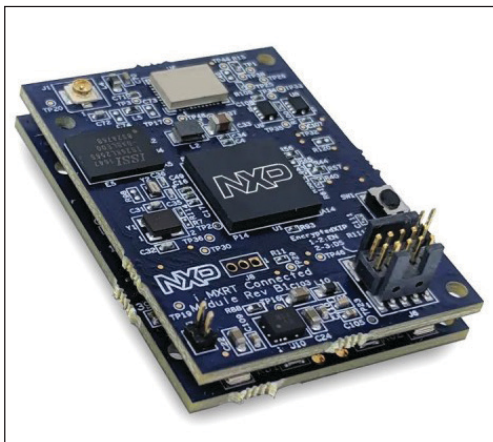


FIGURE 5

Built on an NXP i.MX RT crossover platform, this MCU-based AVS solution enables low latency, far-field, "wake word" detection, embeds all necessary digital signal processing capabilities, runs on Amazon FreeRTOS and includes an Alexa client application.

is the TAS2770 15 W audio amplifier. TI claims it as the first wide-supply I/V sense amplifier. The TAS2770 (**Figure 6**) offers state-of-the-art, real-time speaker protection when paired with TI Smart Amp algorithms. The amplifier monitors loudspeaker behavior and increases loudness while improving audio quality in applications requiring small speakers.

The TAS2770 is an audio front end (AFE) that combines a digital microphone input with a powerful I/V sense amplifier. The device captures

Make your code even faster, smaller, and smarter
while ensuring robustness and high quality.



IAR Embedded Workbench

TOTAL TOOLS

Our customers build the technology for a new world.
We supply the tools to make it happen.
One toolbox, one view, one uninterrupted workflow.
As simple as that.



www.iar.com

Learn for free at
circuitcellar.com/iar



FIGURE 6

The TAS2770 is an audio front end (AFE) that combines a digital microphone input with a powerful I/V sense amplifier. The device captures voice and ambient acoustic information for echo cancellation or noise reduction in voice-enabled applications.

voice and ambient acoustic information for echo cancellation or noise reduction in voice-enabled applications. The TAS2770 monitors battery voltage and automatically decreases gain when audio signals exceed a set threshold, helping designers avoid clipping and extend playback time through end-of-charge battery conditions without degrading sound quality. Embedded systems designers can use TI's PurePath Console software to easily configure the TAS2770. Engineers can jump-start their design with the TAS2770 Stereo Audio Subsystem Reference Design. Additional resources and reference designs are available to help engineers with their smart speaker designs.

SENSORS FOR SAFETY

For its part, Infineon Technologies provides a variety of technologies aimed at the Smart Home. For example, its range of XENSIV sensors enable highly accurate and reliable status monitoring in Smart Home devices. Meanwhile, its OPTIGA product family of hardware-based security solutions are designed to provide the right level of

security to protect both data and devices in Smart Home appliances, says Infineon. And finally, the company also provides IGBT power semiconductors in discrete or module packages, driver ICs, highly-integrated IPM solutions and XMC MCUs—all well suited for Smart Home systems.

An example of the Infineon's recent XENSIV offerings is its XENSIV PAS210, announced in June. According to the company, city dwellers often spend a large amount of their time indoors—either in an office, at school or simply at home. Urbanized spaces, however, tend to trap and develop bad air quality as they get more insulated for energy efficiency purposes. The concentration of CO₂ is an indicator for bad air quality.

Many of today's market solutions for monitoring this odorless and colorless gas are bulky and costly or simply not good enough for a widespread adoption. To address this issue, Infineon developed a disruptive CO₂ sensor technology based on the photoacoustic spectroscopy (PAS). The XENSIV PAS210 sensor makes use of this new technology. It implements a high sensitivity MEMS microphone as a detector and enables significant miniaturization of CO₂ sensors (**Figure 7**). For that reason, it is ideal for Smart Home applications and building automation—for demand-controlled ventilation, as well as various indoor air quality IoT devices such as air purifiers, thermostats, weather stations and personal assistants.

On a single PCB, the CO₂ sensor integrates the photoacoustic transducer including the detector, the infrared source and the optical filter. It also holds an MCU for signal processing and algorithms as well as a MOSFET to drive an infrared source. XENSIV PAS210 uses the high SNR (signal-to-noise ratio) MEMS microphone XENSIV IM69D130 as a detector.

The device's integrated MCU converts the MEMS microphone output into a ppm reading, which is available either via the serial I²C, UART or PWM interface. XENSIV PAS210 CO₂ measurement capabilities cover a range from 0 ppm to 10,000 ppm with an accuracy of ±30 ppm or ±3% of the reading. In a pulsed mode, the XENSIV PAS210 CO₂ sensor is designed to have a lifetime of ten years.

KIT WITH iCloud ACCESS

Among the Smart Home offerings from Cypress Semiconductor is its Wireless Internet Connectivity for Embedded Devices (WICED) Studio platform, a turnkey development platform for the IoT that simplifies the

FIGURE 7

The XENSIV PAS210 sensor implements a high sensitivity MEMS microphone as a detector and enables significant miniaturization of CO₂ sensors. For that reason, it is ideal for Smart Home applications and building automation—for demand-controlled ventilation, as well as various indoor air quality IoT devices.



integration of wireless connectivity into smart home applications (**Figure 8**). Last year the company updated WICED adding iCloud remote access support for Wi-Fi-based accessories that support Apple HomeKit. Developers can leverage iCloud support in the WICED software Cypress WICED IoT Development Kit (SDK) and Cypress' CYW43907 Wi-Fi MCU to create hub-independent platforms that connect directly to Siri voice control and the Apple Home app remotely.

Using Cypress' WICED development platform and low power CYW20719 Bluetooth/BLE MCU, developers can integrate HomeKit support into products such as smart lighting devices, leverage Siri voice control and seamlessly connect to the Apple Home app. WICED Studio provides a single development environment for multiple wireless technologies, including Cypress' Wi-Fi, Bluetooth and combo solutions, with an easy-to-use application programming interface in an integrated and interoperable wireless SDK.

The Cypress CYW43907 SoC integrates dual-band IEEE 802.11b/g/n Wi-Fi with a 320 MHz Arm Cortex-R4 RISC processor

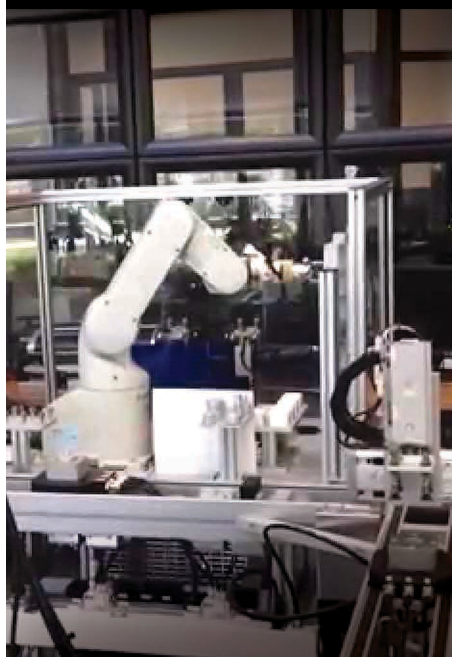


FIGURE 8

The Wireless Internet Connectivity for Embedded Devices (WICED) Studio platform is a turnkey development platform for the IoT that simplifies the integration of wireless connectivity into smart home applications. The updated WICED added iCloud remote access support for Wi-Fi-based accessories that support Apple HomeKit.

You can take it almost anywhere.

Where will it take you?



Now you can have the complete Circuit Cellar issue archive and article code stored on a durable and portable USB flash drive.

Includes PDFs of all issues in print through date of purchase.

Visit cc-webshop.com to purchase

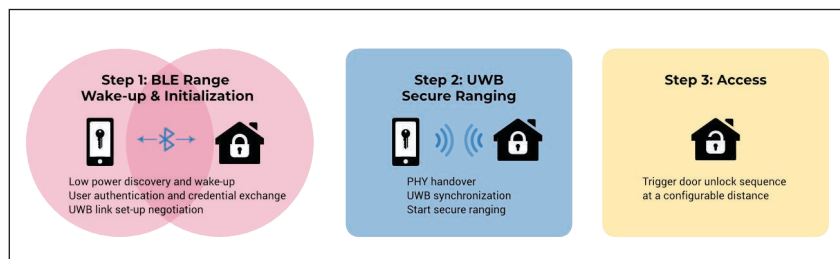


FIGURE 9

Working securely in combination with other connectivity technologies, such as Bluetooth Low Energy (BLE), UWB-driven access works in mobile phones, badges and other wearables, and can even lock or unlock doors when the device is turned off or in sleep mode.

and 2 MB of SRAM to run applications and manage IoT protocols. The SoC's power management unit simplifies power topologies and optimizing energy consumption. The WICED SDK provides code examples, tools and development support for the CYW43907.

UWB IN THE SMART HOME

Smart Home implementations rely on several wireless technologies—including Wi Fi, Bluetooth and others. But Ultra-Wideband (UWB) technology offers some unique advantages when it comes to applications where precise ranging and location determination is required. To help grow the UWB ecosystems, in August, the ASSA ABLOY Group (which includes HID Global), NXP Semiconductors, Samsung Electronics and Bosch announced the launch of the FiRa Consortium.

The new coalition is designed to grow the UWB ecosystem so new use cases for fine ranging capabilities can thrive, ultimately setting a new standard in seamless user experiences. Sony Imaging Products & Solutions, LitePoint and the Telecommunications Technology Association (TTA) are the first companies to join the newly-formed organization.

The FiRa name, which stands for "Fine Ranging," highlights UWB technology's unique ability to deliver unprecedented


accuracy when measuring the distance or determining the relative position of a target. Especially in challenging environments, UWB technology outperforms other technologies in terms of accuracy, power consumption, robustness in RF connection and security, by a wide margin. The starting point for UWB technology is the IEEE standard 802.15.4/4z, which defines the essential characteristics for low-data-rate wireless connectivity and enhanced ranging.

SEAMLESS ACCESS

While UWB is applicable to a variety of applications, there are some specific benefits for the Smart Home. The first is seamless access control. With today's secure access technologies typically you have to perform an action—such as entering a code on a keypad or placing a finger on a scanner—in order to open a doorway and enter a secured area, like your home or garage. With a UWB setup, you simply approach the door and it opens, leave the door on the unsecure (outside) and it locks. The UWB-based solution tracks your approach, verifies your security credentials and lets you pass without having to stop and tap first.

Working securely in combination with other connectivity technologies, such as Bluetooth Low Energy (BLE) and Near Field Communication (NFC), UWB-driven access works in mobile phones, badges and other wearables, and can even lock or unlock doors when the device is turned off or in sleep mode (**Figure 9**).

UWB also provides Angle of Arrival (AoA) technology whereby two UWB devices can share relative ranging and positioning data to localize each other. In Smart Homes, networked devices and appliances can respond to your presence automatically, as you enter a room. As soon as you enter a room (or even approach the entry), the lights can turn on, the thermostat can be set to your favorite temperature and your preferred playlist can start streaming on the audio system. The technology is less intrusive, requires less oversight and supports your everyday routine, while enabling a more energy-efficient and sustainable lifestyle.

Certainly, the evolution of Smart Home technology is in its early stages in terms of market acceptance. But the building blocks in terms of control, wireless connectivity and sensing solutions already exist from a variety of vendors. This trend will continue as chip vendors craft new tools and solutions aimed at building robust Smart Home implementations. 

For detailed article references and additional resources go to:

www.circuitcellar.com/article-materials

RESOURCES

Cypress Semiconductor | www.cypress.com

FiRa Consortium | www.firaconsortium.org

Infineon Technologies | www.infineon.com

Microchip Technology | www.microchip.com

NXP Semiconductor | www.nxp.com

Renesas Electronics | www.renesas.com

STMicroelectronics | www.st.com

Texas Instruments | www.ti.com



Early Bird Registration Is Now Open!
17th International System-on-Chip (SoC) Conference & Exhibit
 October 16 & 17, 2019 - University of California, Irvine - Calit2
www.SoCconference.com

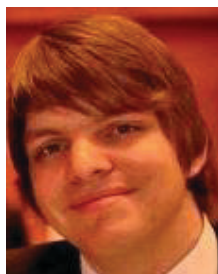
Five Enlightening Keynotes on Emerging SoC Technologies



IBM
 Jason S. Miller,
 Director, IBM Z
 Processor and Systems
 Chips Development.



Intel
 Sailesh Kottapalli, Senior
 Fellow, Graphics and
 Software Chief Architect,
 Datacenter Processor
 Architecture.



Google
 Dr. Martin Maas, Research
 Scientist, Google Research,
 Brain Team.



University of Texas-Austin
 Dr. Lizy Kurian John, P.E.
 The Cullen Trust for Higher
 Education Endowed Professor in
 Engineering No.3, IEEE Fellow.



Xilinx
 Jennifer Wong, Vice President
 of FPGA Product
 Development.

Platinum Sponsors



Farhad Mafie, President & CEO, Savant Company Inc. SoC Conference Chairman,
 Moderating three Informative Panels:

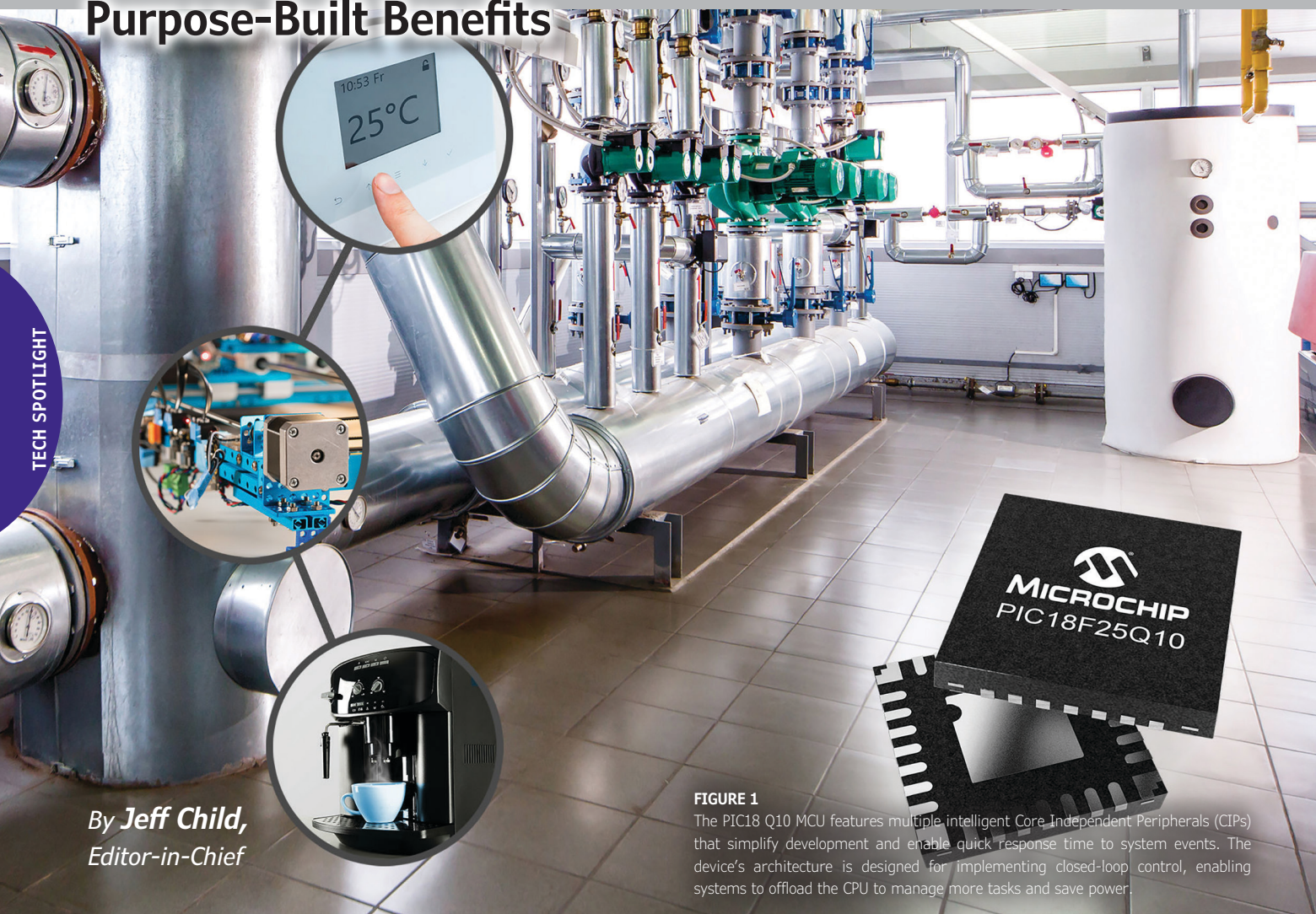
- RISC-V Realities, Opportunities and Challenges in the Complex and Crowded CPU Market.
- How AI and Machine Learning will Drive the Semiconductors Market to the Next Level.
- Is Blockchain of Things IoT 2.0? Could Blockchain of Things be the Security solution that the industry has been looking for?

Selected Presenting Companies & Universities



**For More Information or Questions, Please Contact the SoC Conference Organizing Committee at:
MySoCconference@Gmail.com or (949) 356-2399**

Application-Specific MCUs Deliver Rich Functionality Purpose-Built Benefits



TECH SPOTLIGHT

By **Jeff Child**,
Editor-in-Chief

FIGURE 1

The PIC18 Q10 MCU features multiple intelligent Core Independent Peripherals (CIPs) that simplify development and enable quick response time to system events. The device's architecture is designed for implementing closed-loop control, enabling systems to offload the CPU to manage more tasks and save power.

A class of microcontrollers has emerged that provide tailored feature sets aimed at targeted application needs. These application-specific MCUs give embedded systems developers right-sized solutions with the functionality that fits their system requirements.

In contrast to microprocessors, microcontrollers (MCUs) tend to be used for specific applications. But even among MCUs, there's distinct difference between general purpose MCUs and MCUs that are designed for very specific application segments, or even sub-segments. These application-specific MCUs usually have carefully selected on-chip IP cores and peripherals that meet the requirements unique to an application need.

Responding to needs of embedded systems developers, over the past 12 months MCU vendors have continued to roll out new MCUs that address application needs in industrial control, banking, safety-critical systems and more. While automotive MCUs are, in particular, an extremely active segment of application-specific MCUs, I don't address automotive MCUs in this article. We covered that topic recently our August Special Feature "IC Solutions Rev Up for Next Gen Auto Designs" (*Circuit Cellar* 349).

CLOSED-LOOP CONTROL

Closed-loop control ranks as one of the most fundamental tasks performed by embedded systems—required in systems ranging from pool pumps to air handling units. Feeding that specific need, Microchip Technology offers its PIC18 Q10 (**Figure 1**) and ATtiny1607 families, featuring multiple intelligent Core Independent Peripherals (CIPs) that simplify development and enable quick response time to system events. Advancements in the architecture of PIC and AVR 8-bit MCUs have optimized the devices for implementing closed-loop control, enabling systems to offload the CPU to manage more tasks and save power.

Well suited for applications that use closed-loop control, a key advantage of using the PIC18 Q10 and ATtiny1607 MCUs are the CIPs that independently manage tasks and reduce the amount of processing required from the CPU. System designers can also save time and simplify design efforts with the hardware-based CIPs, which significantly reduce the amount of software required to write and validate. Both families have features for functional safety and operate up to 5 V, increasing noise immunity and providing compatibility with the majority of analog output and digital sensors.

Offered in a compact 3 mm x 3 mm 20 pin QFN package, the new ATtiny1607 family is optimized for space-constrained closed-loop control systems such as handheld power tools and remote controls. In addition to the integrated high-speed analog-to-digital converter (ADC) that provides faster conversion of analog signals resulting in deterministic system response, the devices provide improved oscillator accuracy, allowing designers to reduce external components and save costs.

Among the CIPs in the PIC18 Q10 family are the Complementary Waveform Generator (CWG) peripheral, which simplifies complex switching designs, and an integrated Analog-to-Digital Converter with Computation (ADC2) that performs advanced calculations and filtering of data in hardware without any intervention from the core. CIPs such as these allow the CPU to execute more complex tasks, such as Human Machine Interface (HMI) controls, and remain in a low-power mode to conserve power until processing is required.

WATER TOLERANT TOUCH

User interfaces in harsh industrial environments have to contend with factors such as water and other disturbances. To address such needs, in March, NXP Semiconductor announced its 5 V KE1xZ family

of MCUs. Based on the Arm Cortex-M0+ core, the MCUs are suited for embedded control systems in harsh electrical environments and provide an integrated CAN controller and capacitive touch from 32 KB flash.

Designed for a wide range of industrial applications, the KE1xZ family offers mixed-signal integration across a range of compact memory variants. The 1-MS/s ADC and FlexTimer modules, combined with NXP's Freemaster software tools library and Motor Control Application Tuning plugin (MCAT) enable designs of Brushless DC (BLDC) and other motor-control systems.

NXP's KE1xZ MCU family offers advanced noise immunity, water-tolerant touch and low-power wake-on-touch operation—essential features for the strict electromagnetic compatibility (EMC) standards of the industrial and home appliance markets. NXP's touch IP, combined with software and tools provide a high level of stability, accuracy and ease of use, with continued responsiveness and functionality through wet conditions. It can sustain 10 V in conducted noise, in alignment with International Electrotechnical Commission (IEC) 6100-4-6 test level 3. NXP enables developers through its MCUXpresso software and tools ecosystem, along with its FRDM-KE15Z and FRDM-TOUCH development platforms (**Figure 2**).

MCU FOR CONTACTLESS BANKING

Contactless transaction capability is an important application in systems such as banking, identity, transportation and pay-television. In July, STMicroelectronics (ST)



FIGURE 2

The KE1xZ MCU family offers advanced noise immunity, water-tolerant touch and low-power wake-on-touch operation. The device is supported by NXP's MCUXpresso software and tools ecosystem, along with its FRDM-KE15Z and FRDM-TOUCH (shown here) development platforms.



FIGURE 3

The ST31P450 is a dual-interface secure MCU built on the 32-bit Arm SecurCore SC000 secure processor and meets ISO 7816 and ISO 14443 Type A smart-card and contactless standards.

announced an MCU aimed directly at that space. The ST31P450 is a dual-interface secure MCU that features the latest 40 nm flash process as well as enhanced RF technologies (**Figure 3**). The device is built on the proven 32-bit Arm SecurCore SC000 secure processor and meets ISO 7816 and ISO 14443 Type A smart-card and contactless standards. It supports the full range of MIFARE libraries including MIFARE Classic, MIFARE Plus and MIFARE DESFire.

ST's 40 nm Flash technology creates an ultra-small die for dual-interface use

cases such as banking and, with security-enhancing properties, increases safety and fraud prevention. Upgraded RF performance ensures ultra-reliable wireless connections for faster, easier contactless transactions.

In addition, the ST31P450 features new low-power cryptographic engines that minimize the energy budget and ensure superior product performance at low RF-field strength while executing cryptographic operations. The ST31P450 also comes with optimized loading firmware that makes life easier for card provisioners, offering post-issuance capability. The MCU and its associated cryptographic libraries, are expected to achieve Common Criteria EAL5+, as well as EMVCo and CUP (China UnionPay) certifications within the coming months. The ST31P450, with 450 KB non-volatile memory (NVM) and 10 KB RAM on-chip, is in production now.

SAFETY-CRITICAL MCU

Safety-critical systems—any system where failures could cause harm to life and limb—have special requirements beyond those of other embedded systems. To address such needs at the MCU level, Infineon Technologies decided on a collaborative effort. In February, Infineon announced an effort with Xilinx and Xylon to produce a new Xylon IP core called logiHSSL. It enables high-speed communication between Infineon's AURIX TC2xx and TC3xx MCUs (**Figure 4**) and Xilinx' SoC, MPSoC and FPGA devices via the Infineon High Speed Serial Link (HSSL). This serial link supports baudrates of up to 320 Mbaud at a net payload data-rate of up to 84%.

The HSSL is an Infineon native interface that is low-cost in regard to pin-count because it requires only five pins—two LVDS with two pins each and one CLK pin. So far, the HSSL interface is used to exchange data between AURIX devices and customer ASICs for performance or functional extension.

Now, the new IP core will allow system developers to combine safety and security provided by AURIX with the wide range of functional possibilities brought to the table by the Xilinx devices. Linked devices can access and control each other's internal and connected resources through the HSSL.

To support development activities, the partners are offering a starter kit. It includes a Xilinx evaluation kit, an Infineon AURIX evaluation board and a Xylon FMC board. Kit deliverables include the reference design with the test software application, Xylon's logicBRICKS evaluation licenses, documentation and technical support.

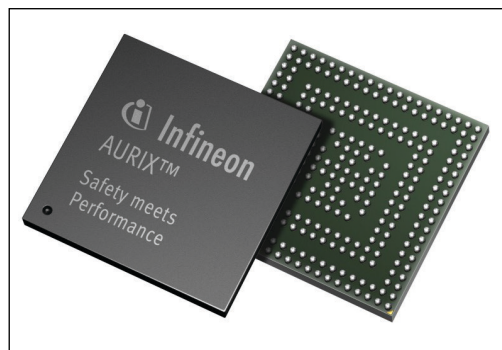


FIGURE 4

The safety-critical IP, co-developed by Xilinx and Xylon, will allow system developers to combine the safety and security provided by Infineon's AURIX MCUs with the wide range of functional possibilities brought to the table by the Xilinx devices. Linked devices can access and control each other's internal and connected resources through the HSSL.

RESOURCES

Infineon Technologies | www.infineon.com

Microchip Technology | www.microchip.com

NXP Semiconductor | www.nxp.com

Renesas Electronics | www.renesas.com

STMicroelectronics | www.st.com

Texas Instruments | www.ti.com

MCUs FOR SERVO CONTROL

MCUs play many roles in industrial applications. But there are several specific systems that have unique requirements. With that in mind, in May Renesas Electronics introduced the RX72T Group of 32-bit motor control MCUs with dedicated hardware accelerator IP to perform the complex, high-speed computations required for motor control in robots and other industrial equipment (**Figure 5**). According to the company, the RX72T Group provides superior performance, achieving an 1160 CoreMark score as measured by EEMBC Benchmarks—the highest level for a 5 V MCU operating at 200 MHz.

Servo systems are becoming increasingly popular and the demand for compact industrial robots is also rising, says Renesas. To implement the complex motions required by industrial robots, high-precision control of the position, direction, speed, and torque of the motor is required. The RX72T MCUs include dedicated accelerator hardware that enables the high-speed position control and speed control calculations required for implementing servo motor control in compact industrial robots. This allows the current control loop calculation to be performed in less than 1.5 μ s, providing a new option for users to independently develop servo systems where previously they only had the option of purchasing existing servo systems.

Implementing calculations in software can require excessive computing time. However, completely hardware-based calculations can adversely affect the flexibility to implement unique user control operations. With the RX72T MCUs, Renesas resolves the issue by implementing only the single-precision floating point trigonometric function (sin, cos, arctan, hypot) and a register bank save function in hardware as dedicated IP. This retains the flexibility while increasing the calculation speed. Furthermore, the register bank save function increases the speed and precision of interrupt handling, improving the device computation performance. The MCUs also include 200 MHz PWM inverter control timers with up to 4 channels of 3-phase control, 2 channels of 5-phase control or 10 channels of single-phase control.

MCUs WITH EtherCAT

Also targeting industrial systems, in June Texas Instruments (TI) introduced new communications capabilities on its C2000 MCUs. C2000 F2838x 32-bit MCUs enable designers to use a single chip to implement connectivity, including EtherCAT, Ethernet and Controller Area Network with Flexible Data




FIGURE 5

The RX72T Group of 32-bit motor control MCUs have dedicated hardware accelerator IP to perform the complex, high-speed computations required for motor control in robots and other industrial equipment.

Rate (CAN FD), in AC servo drives and other industrial systems.

Systems with communication interfaces often require an external ASIC or dedicated host control microprocessor, which limits the flexibility of the design architecture, adds complexity and takes up space on the board. The new C2000 F2838x MCUs do not require an external ASIC, therefore, reducing overall solution size and bill of materials.

By integrating three industrial communication protocols, the F2838x MCUs give designers the ability to tailor one MCU to the unique needs of each system. A key component in achieving this is a new connectivity manager, an Arm Cortex-M4-based subsystem, which offloads processing-intensive communications and optimizes connectivity. In addition to these capabilities, C2000 F2838x MCUs offer enhanced real-time control performance and higher flexibility than previous C2000 series MCUs.

For electrically isolated architectures, the new MCUs use a fast serial interface with eight receiving channels to facilitate chip-to-chip communication at up to 200 Mbps using minimal pins. Designers can build on this high level of integration in CAN FD designs and quickly increase the number of available CAN FD ports by pairing the F2838x with a TI system basis chip (SBC), such as the TCAN4550 SBC with integrated CAN FD controller and transceiver. 

Product Focus: Rugged SBCs and COMs

Tested for Toughness

By **Jeff Child**,
Editor-in-Chief

FIGURE 1

Rugged SBCs and COMs used in oil drills must endure hazardous environments where both extreme heat and severe shock/vibration are constant.

Embedded computers are used in a broad sweep of applications—some that must operate in harsh environmental conditions. Built for those situations, rugged SBCs and COMs offer a variety of attributes to serve such needs, including extended temperature range, high shock and vibration resilience and even high humidity protection.


If you're building an embedded system that's in an office or some other benign environment, you can afford to use any ordinary single board computer (SBC) or COM Express module that meets your performance needs. But once you get into any application where your electronics are exposed to extreme heat, vibration and other physical factors, rugged embedded computers are an absolute necessity. Fortunately, there's a rich set of rugged SBC and COM choices available from vendors with decades of experience building such products and testing them for harsh environment duties. These rugged products not only have to be designed to withstand harsh environmental conditions. They also have to be tested to guarantee such capabilities.

Good thermal management is one key attribute of rugged SBCs and COMs. As you can see from the product gallery of representative rugged boards in this article, all of them are designed for -40°C to +85°C operating temperatures. Some boards manage heat using convection cooling—basically guiding an air flow along the surface to be cooled. This is usually done using fans. For the applications on the harsher side, such solutions are not ideal. Cooling fans have a limited lifetime and also represent a risky single-point-of-failure for an embedded system. That's why many rugged

embedded boards use conduction cooling instead.

Conduction cooling uses mechanical design to optimize the thermal contact to conduct the heat from its source to the outside wall of the system's enclosure, thereby using the enclosure itself a heat radiator. Heatsinks also help manage the heat dissipation of the most power-hungry devices on the board.

Aside from heat, several other environmental factors are taken into account in rugged board-level designs. Sealed enclosures in combination with conformal coating are often needed for protection from dust or humidity. And to ensure resistance against high shock and vibration effects, components have to be firmly screwed to the based printed circuit board.

A wide range of applications depend on rugged SBCs and COMs. These include factory automation, railway systems, mining systems, aerospace electronics, military systems and more. The oil and gas industry in particular is a segment where rugged embedded computers are critical. Field computing applications in oil and gas demand precise computing to control complex equipment such as drilling rigs (**Figure 1**). Computer boards used in those systems must endure the hazardous environments of wellhead refineries where both extreme heat and severe shock/vibration are constant. 

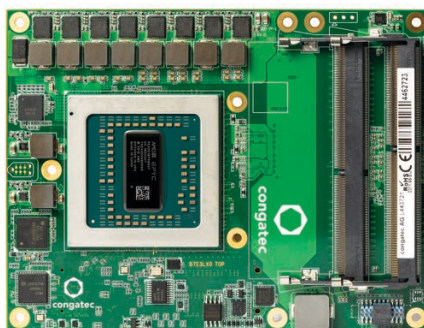


Type 6 COMe with Quad iCore and Celeron Processors

The cExpress-WL module from ADLINK Technology is a COM Express Type 6 board designed to fulfill the needs of size, weight and power (SWaP)-constrained edge computing applications. It features the 8th generation Intel Core and Celeron processors (formerly codenamed Whiskey Lake-U) with up to 4 cores and up to 64 GB memory capacity. The cExpress-WL is suited for applications such as data acquisition and analysis, image processing and 4K video transcoding and streaming at the edge.

- 8th Gen quad/dual-core Intel Core Processors
- Up to 64 GB Dual Channel non-ECC DDR4 at 2133/2400MHz
- 2x DDI channels, 1x LVDS, 1x opt. VGA
- Supports up to 3 independent displays
- Up to 8x PCIe lanes, GbE
- Up to three SATA 6 Gb/s, four USB 3.1 Gen2 and four USB 2.0
- Supports Smart Embedded Management Agent (SEMA) functions
- Operating temperature: -40°C to +85°C (optional)

ADLINK Technology
www.adlinktech.com

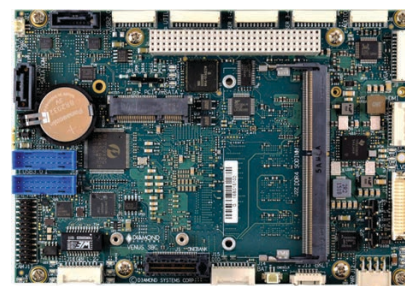


AMD EPYC Embedded 3000 Processor Rides COM Express

Congatec's conga-B7E3 is a Server-on-Module (SoM) with an AMD EPYC Embedded 3000 processor that offers up to 52% more instructions per clock compared to legacy architectures, according to the company. Use cases include Industry 4.0, smart robot cells with collaborative robotics, autonomous robotic, logistics vehicles and more.

- AMD EPYC Embedded 3000 Series
- Up to 16 Cores, 32 MB cache
- 30 W to 100 W TDP
- 4x 10 Gbit Ethernet KR Interface
- Up to 32 PCI Express generation 3 lanes
- Industrial grade variant supports -40°C to +85°C
- Optional NVMe SSD up to 1 TB capacity

Congatec
www.congatec.com



3.5" SBC Sports 2.6 GHz Skylake Processor

Venus from Diamond Systems is a rugged, small form factor (SSF) SBC featuring the Intel Skylake or Kaby Lake 6th and 7th Generation processor in an extended "3.5-inch" form factor. Venus offers a wide range of I/O plus data acquisition functionality, meeting the majority of today's connectivity requirements.

- Skylake U i7-6600U 2.6 GHz, or i5-6300U 2.4 GHz
- Kaby Lake i7-7660U 2.8 GHz
- 4 GB memory down plus socket for 4/8/16 GB DDR4 DIMM
- 3 independent displays: HDMI, VGA and LVDS
- 4x USB 2.0, 4x USB 3.0
- 4x multi-protocol serial RS-232/422/485, 3x SATA
- "3.5 inch" form factor: 5.75" x 4.0" / 146 mm x 102mm
- OneBank PCIe/104 and PCI-104 expansion
- -40°C to +85°C temperature range

Diamond Systems
www.diamondsystems.com

Rugged SBCs and COMs



Rugged, Sandwich-Style SBC has TI Sitara AM5718

The OK5718-C from Forlinx Embedded Technology is an industrial temp SBC that Linux on a "FET5718-C" module with a Cortex-A15 based TI AM5718 SoC. Other features include SATA, HDMI, MIPI-CSI, USB 3.0, CAN and mini-PCIe. It's a Sitara based board, in this case tapping TI's single-core, Cortex-A15 based Sitara AM5718.

- Arm Cortex-A15, frequency up to 1.5 GHz;
- DSP C66x with frequency of 750 MHz
- USB 3.0, PCIe 3.0, HDMI 1.4, 10x UART, 1x QSPI*1, 2x CAN, 2x Gbit Ethernet
- OS options: Linux 4.9.41 + QT5.6, TI-RTOS;
- 12-layer EINT PCB design
- Carrier board dimensions of 130 mm 190mm
- SoM with 320 pins that all functions of CPU are available
- Industrial grade temp width: -40°C to +85°C

Forlinx Embedded Technology
www.forlinx.net



COMe Board Offers 16 GB DRAM and Extend Temp. Design

The COMe-mAL10 (E2) / COMe-m4AL10 (E2) from Kontron is a series of credit card sized COMs based on the latest generation Intel Atom, Pentium and Celeron processors. The module family is available in two different designs. COMe-mAL10 (E2) uses DDR3L memory and is the perfect choice, when ECC is necessary. COMe-m4AL10 (E2) is equipped with LPDDR4 memory for even higher data transfer rate demands.

- Low-Power – Performance/Watt optimized small form factor solution
- COMe-mAL10 (E2): up to 8 GB DDR3L memory down (ECC / non ECC)
- COMe-m4AL10 (E2): up to 16 GB LPDDR4 memory down (non ECC)
- 2x USB 3.0/2.0, 6x USB 2.0, 2x SATA, eMMC flash
- Industrial grade temperature -40°C to +85°C
- Support of Kontron's Embedded Security Solution (Approtect)

Kontron
www.kontron.com

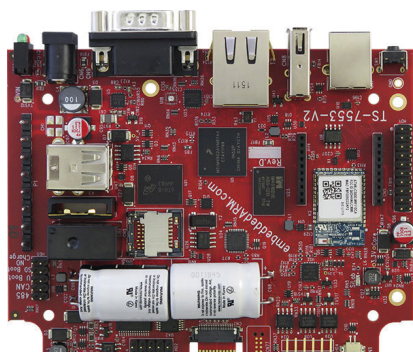


Rugged COMe Board Sports Ryzen Embedded V1000/R1000 SoC

MEN Micro's CB71C is an ultra-rugged COM Express module for rail, public transportation and industry applications, e.g. data acquisition, infotainment, transcoding and live 3D. It is 100% compatible with COM Express Type 6 Pin-Out and conforms to the VITA 59 standard, which specifies robust mechanics to ensure reliable operation even under the harshest environmental conditions.

- AMD Ryzen Embedded V1000/R1000 series
- Up to 32 GB DDR4 RAM with ECC
- Up to 4 Digital Display Interfaces (DP, eDP, HDMI, DVI)
- Hardware memory encryption
- Safety-relevant supervision functions
- Virtualization
- Up to -40°C to +85°C Tcase, conduction cooling
- VITA 59 in process, compliant with COM Express Basic, Type 6
- PICMG COM.0 COM Express version also available

MEN Micro
www.menmicro.com

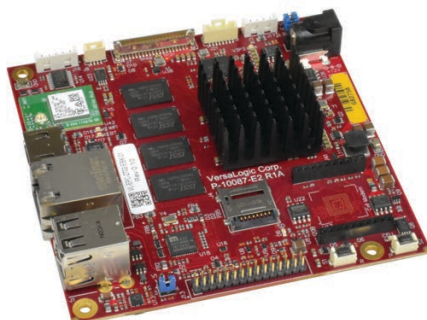


Fanless i.MX6 UltraLite SBC Targets Industrial IoT

The TS-7553-V2 from Technologic Systems is an SBC that offers the ability to communicate seamlessly with several different networks simultaneously from a single device. Using the onboard peripherals, the system can connect to Ethernet, Wi-Fi, Bluetooth, USB, RS 232, RS-485 and CAN networks or devices. Built in module interfaces like the XBee/NimbleLink socket and internal USB ports allow expansion to other networks like cellular, DigiMesh, ZigBee and others.

- NXP 696 MHz i.MX 6UltraLite processor
- Ethernet, WiFi, Bluetooth, XBee/ NimbleLink and USB radios
- TS-SILO Option Provides up to 30 s of Hold up Power
- Data Storage: FRAM, eMMC flash, microSD
- Extendable with daughter board enabling cell modem and PoE cards
- Fanless temp. range: -40°C to 85°C
- Max. power consumption: 1.66 W

Technologic Systems
www.embeddedarm.com



i.MX6-Based SBC Offers Global Cellular Expansion

The Swordtail from VersaLogic is a complete fanless SBC featuring Wi-Fi, Bluetooth and cellular options for applications that operate without a wired data connection. These Arm-based computers are available with power-efficient, dual-core or quad-core i.MX6 CPUs. The Swordtail is well suited for applications that demand rugged low-power solutions such as industrial machine automation, transportation, medical, kiosks and industrial IoT.

- Quad-core or dual-core 800 MHz i.MX6 Processor
- -40° to +85°C operation
- Compliant with 802.11 b/g/n Wi-Fi and Bluetooth 4.2
- MIL-STD-202H qualified for high shock and vibration
- Gbit Ethernet with remote boot support.
- Up to 4 GB soldered-on DRAM
- 2x USB 2.0, 1x serial port, 9x GPIO
- CAN bus, Audio (via HDMI) and I²C support.
- On-board socket supports cellular and other wireless technologies

VersaLogic
www.versalogic.com



FEMTO-ITX SBC Serves Up Intel E3800 SoC

WinSystems' ITX-F-3825 series SCB delivers a rich feature set into an ultra-small form factor Femto-ITX design at just 84 mm x 55 mm. This SBC is based on Intel's E3825 processor which provides CPU and graphics performance. The ITX-F-3825 is well suited for applications that require a very small form factor SBC such as UAVs, medical, robotics and digital signage.

- FEMTO-ITX form factor (84 mm x 55 mm)
- Intel Atom E3825 Processor
- 2 GB DDR3-LV RAM
- Gbit Ethernet
- 2x USB 2.0, 1x serial, 4x digital inputs, 4x digital outputs
- 1x full PCIe Mini Card/mSATA slot
- 1x M.2 type 2242 with B-Key slot
- -20°C to +85°C operational temperature

WinSystems
www.winsystems.com

The Darker Side

An Intro to Antenna Arrays

Elements, Interference and More

As an expert in RF technology, Robert has deep knowledge about antennas. And, in this era of IoT, his expertise is more relevant than ever. That's because every wireless device has some kind of antenna, and these antennas can often be the cause of engineering problems. With that in mind, in this article Robert discusses the math, technology and design issues that are basic to antenna arrays.

By
Robert Lacoste

Welcome to "The Darker Side." Twenty years ago, I used to describe my area of expertise as "radio frequencies" (RF for short). Then that term became a little dated, and I found I was working in the "wireless" industry for a while. A couple of years later, my wife convinced me that we could attract more customers if we presented ourselves as "connected-objects experts"—and she was right. And more recently we changed once again, and now we're in the Internet of Things (IoT) business. Of course, over all those years, nothing changed—we still do RF, but the buzzwords of the day can't be ignored. Another thing that hasn't changed is that every wireless device has some antennas, and these antennas are the root cause of many headaches. Why? Simply because antennas are not well understood by many electronics engineers.

With the main focus of *Circuit Cellar* being embedded electronics, antennas as components aren't a primary focus of this magazine. But I simply can't leave this subject in the dark. Moreover, the design of embedded systems that have antennas is very much in *Circuit Cellar's* wheelhouse.

If you're a regular reader, you know that I've already devoted some articles to antennas. These include "The Darker Side – Antenna Basics" (*Circuit Cellar* 211, February 2008) [1] and "The Darker Side – Antenna Measurement" (*Circuit Cellar* 327, October 2017) [2]. This month, my goal is to introduce you to an interesting class of antennas: antenna arrays. You will see that these antennas are based on very simple principles, and can have more than interesting applications—even for

relatively simple projects. Even if you don't need such an antenna, I hope you will enjoy learning about this topic. And, as usual, I promise I will not use any complex math—just words and illustrations.

INTERFERING WAVES

Some basics first: A radio frequency (RF) signal is an electromagnetic wave of a given frequency (F). This frequency can range from kilohertz to hundreds of gigahertz. For example, F is 2.4 GHz for Bluetooth. I'm sure you know that such a wave propagates at the speed of light. Therefore, a Bluetooth transmitter generates a signal oscillating 2,400,000,000 times per second, and this signal travels in the air at the speed of $c = 300,000,000$ m/s. That means the wavelength in the air—which is the distance between two maximums or minimums of the electric field—is, in this case, $\lambda = c/F = 0.125$ m. Very simple math, as I promised. (Don't be afraid of the Greek letter lambda, "λ". This is just the usual way to name a wavelength.)

Now it's time to introduce a very basic but fundamental concept: Interference (**Figure 1**). Imagine you have two antennas radiating an RF signal at the same frequency and at the same power (P). If they are at the same distance from the receiver, then the waves will be in phase and will add. The result will be an electromagnetic wave twice as powerful ($2 \times P$). However, if the two signal sources are not at the same distance, then the waves will not be in phase, and the resulting signal may not be as strong. The worst case is when the signals are exactly out of phase (a 180-degree phase shift). They will cancel out,

and the result will be zero. If you look again at Figure 1, you will see that this situation occurs when the distance between the two sources is half the wavelength, or $12.5 \text{ cm}/2 = 6.25 \text{ cm}$ for a 2.4 GHz system, then $3\lambda/2$ and so on.

ANTENNA ARRAYS

I have presented enough theoretical background to now explain what an antenna array is. This is simply a set of simple antennas—called “elements”—working together as a single antenna and using interferences to improve the performances. All elements are interconnected through a so-called “feed network,” which sets precise phases and amplitudes between the antenna elements. This means the designer tunes these parameters to enhance the radiated power in some directions—thanks to constructive interferences—or to nullify it in other directions. These gain and phase parameters can be either fixed or dynamically modified. More on that later.

So, an antenna array is a kind of directive antenna, in which the antenna pattern is built by an association of several small antennas interconnected by a properly designed feed network. Also keep in mind that a directive antenna focuses the energy in some direction, which means it provides a higher gain in that direction, and so a longer range. I’m talking about transmitters, but nearly all antennas work exactly the same way in transmission and reception. That means that such an antenna array will have exactly the same beam shape in both directions.

An example will be welcome, won’t it? Look at **Figure 2** where two identical antenna elements are connected through a feed network, which is nothing more than a power splitter—50% of the input power is sent to each antenna, with the same phase. What is the behavior of the resulting antenna? In the plane perpendicular to the line connecting the two antennas, the waves will be in phase, and the power will be twice the power of a single antenna. This direction is called the “main lobe.” However, in other directions, the interference will not be so powerful, and will eventually cancel out completely.

In some other directions there could be local maximums (side lobes). If you are not averse to trigonometry have a look at **Figure 3**, which provides a little more detail on what’s going on in the case of two antennas. As Figure 3 shows, the phase shift in a given direction is proportional to the distance between the two elements, inversely proportional to the wavelength of the signal and proportional to the sine of the angle.

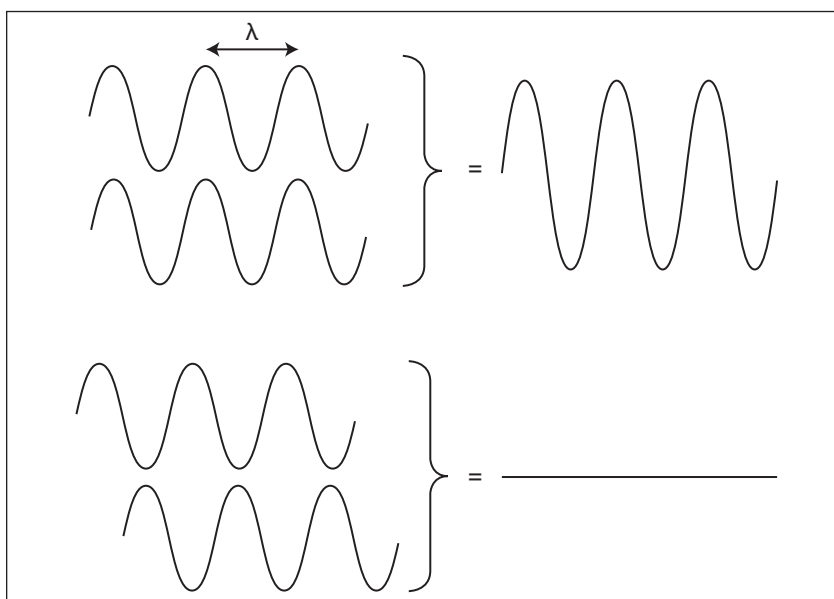


FIGURE 1 Two waves cancel out if they have the same power, but are half a wavelength apart

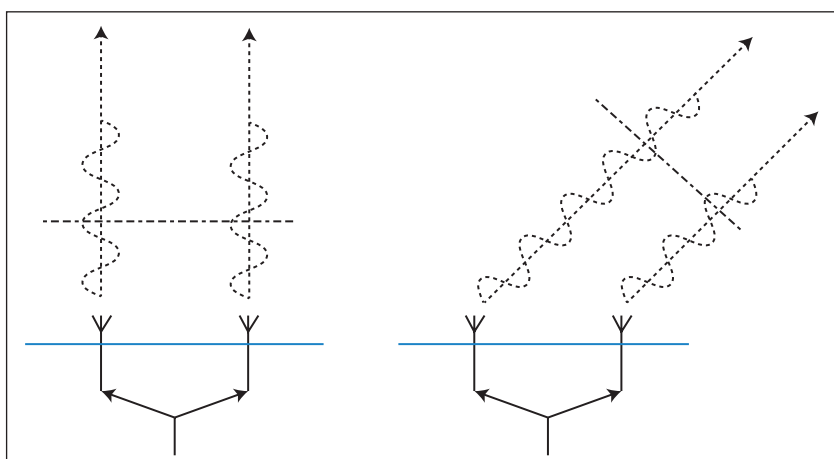


FIGURE 2 If the same signal is sent by two antennas, then radiated power will not be the same in all directions due to interferences.

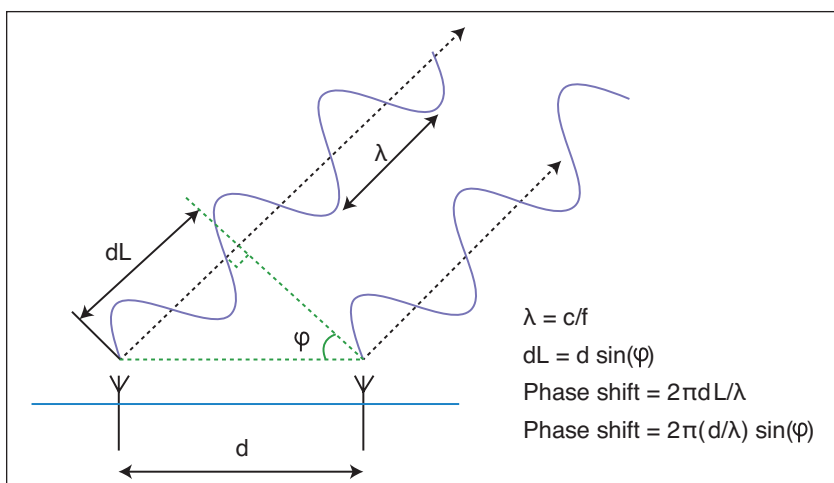


FIGURE 3 Some trigonometry, if you like ...

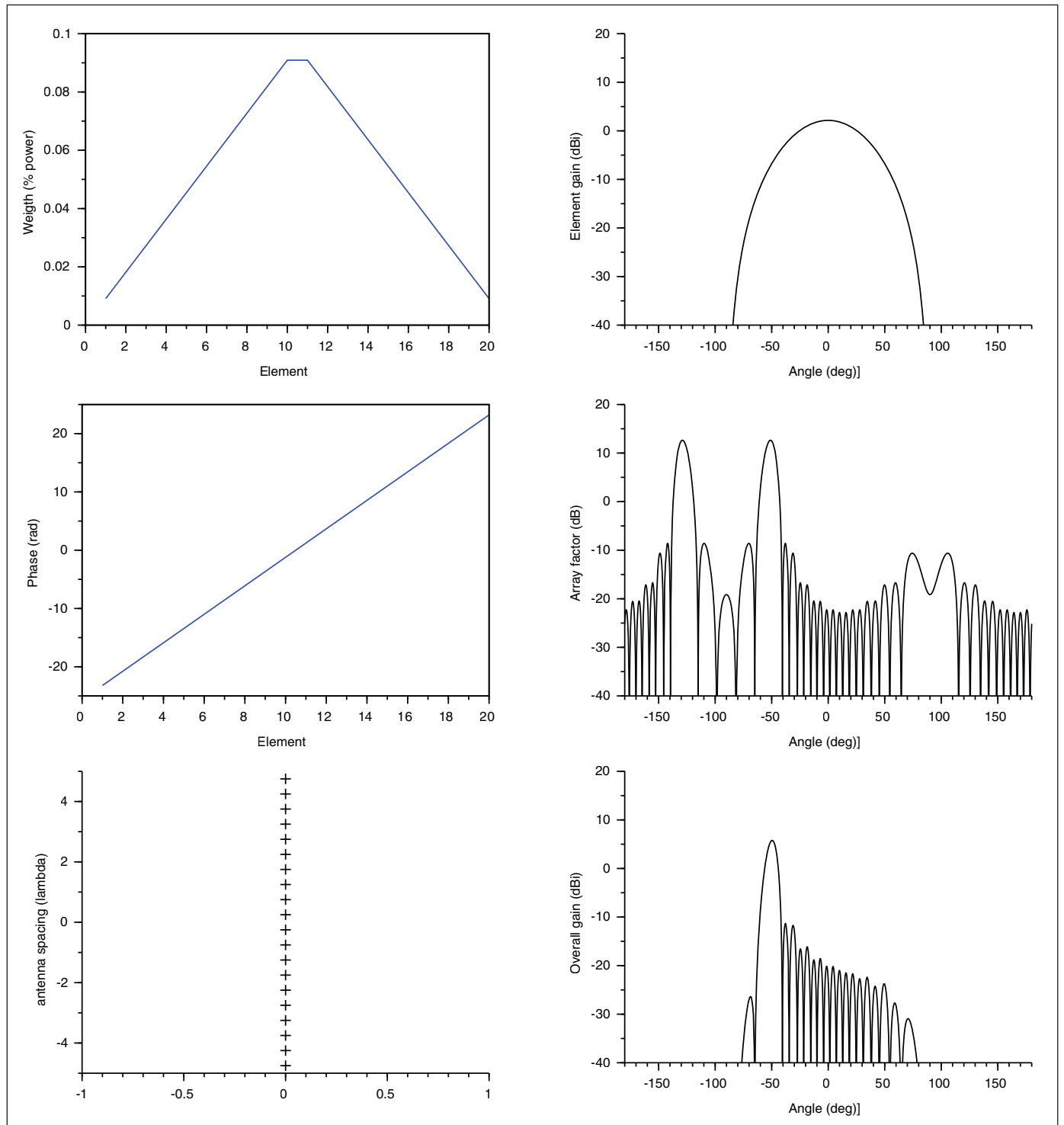
DESIGN PARAMETERS

Ok, now you know the basics. But how can an antenna array be designed? Such arrays exist in zillions of variants, but in this article, I will limit the discussion to the most basic kind. Let's assume that all antenna elements are identical. Let's also assume that these elements are aligned and regularly spaced. Now what are the remaining design parameters?

First, the distance between each pair of elements can be tweaked, and of course the number of elements can be freely set. An antenna array can have 2, 3, 4 ... or thousands of elements. The feed network can also split the input power equally among

all elements, or can implement more complex sharing methods. Signal phase between elements can be adjusted too. And finally, each element can be any kind of antenna, not necessarily the simplest one—which is the so-called “dipole.”

Rather than a theoretical presentation, I propose in this article to illustrate the effect of each of these parameters one by one. For that, I coded a small antenna array simulation software using Scilab—an open-source, numeric programming language. I will not go through the details of the code here, but don't hesitate to download it. The code is available for download on *Circuit Cellar's* article code & files download webpage. The code



is straightforward—no more than 200 lines and annotated—and should be easy to read.

Basically, each antenna element can be a dipole or a “patch” (radiating only in one half of the space), and all other parameters can be freely set—number of elements, distance between elements, input power and phase to each element and so on. The software then simply calculates the array factor—which is the effect of interferences—multiplies it by the theoretical pattern of a single element

and generates some nice plots (**Figure 4**). No electromagnetic simulation is involved, and just the effect of the array is simulated. You can use this software freely—and Scilab is free too—but no warranty is provided, because my only goal was for illustration’s sake in this article.

ELEMENT SPACING

With this software, we can now play and see the effect of each antenna array design parameter. You probably figured out

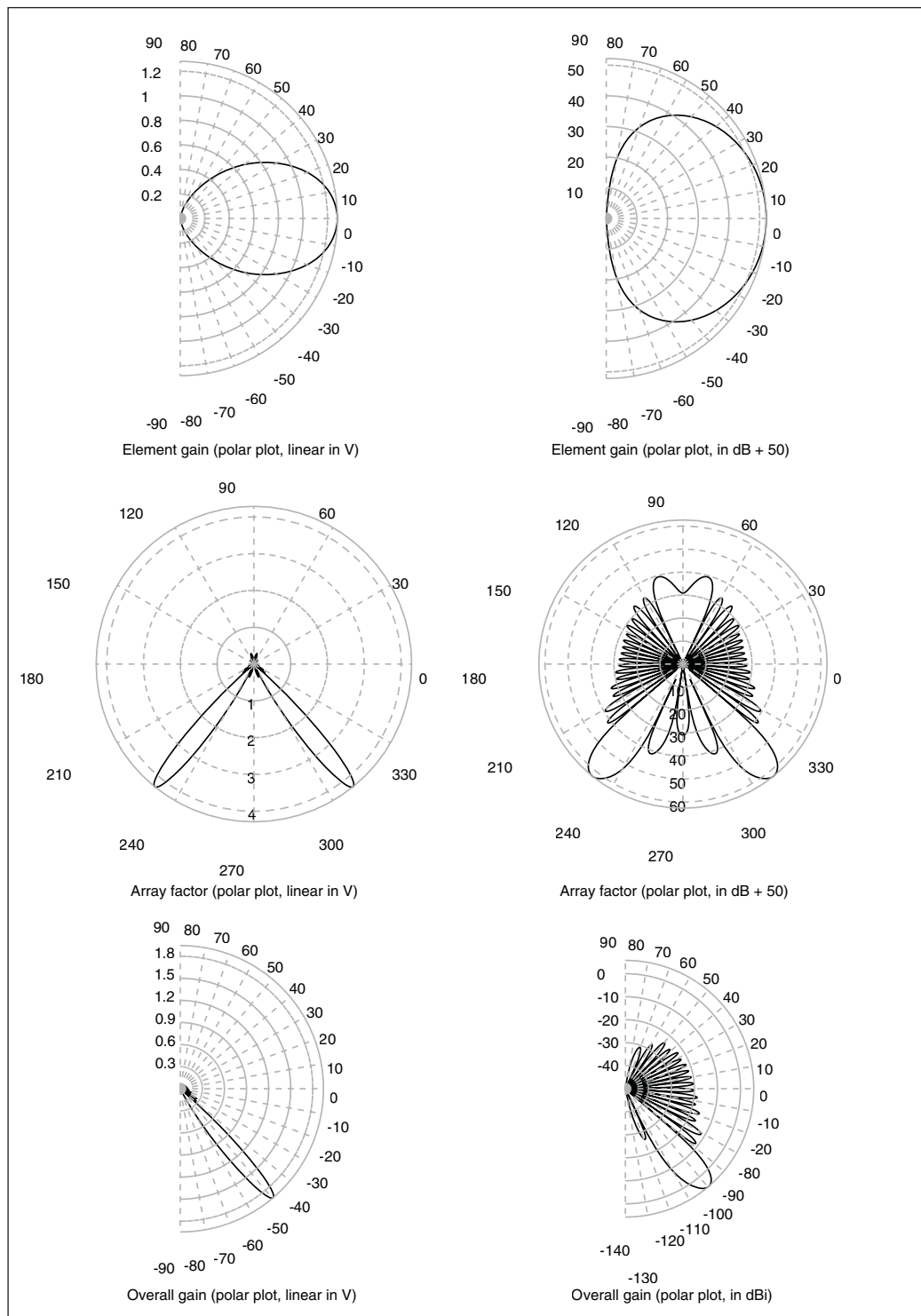


FIGURE 4 These twelve graphs were generated by my Scilab antenna array simulation software—six on this page, six on the previous. By columns, from left to right: design parameters, gain in dB, gain in volt on polar plot and gain in dB on polar plot. By rows, from top to bottom: single element, array factor and resulting antenna array.



FIGURE 5

The SCR270 World War II antenna array (photo taken by R. Haupt at the National Electronics Museum [3]).

already that the distance between elements is directly linked to the wavelength. This means that a given antenna array—used at a frequency of say 1 GHz—will have exactly the same characteristics as an antenna array used at 100 MHz (10 times less), if all lengths are multiplied by 10. That means that each element will have to be 10 times larger, and that the distance between elements also will have to be multiplied to 10.

Two photos will be better than a long explanation of which antenna array uses the highest frequency. **Figure 5** is a picture of the SCR-270 antenna installed in Hawaii during World War II to detect Japanese attacks. It worked at 110 MHz, so the elements were large

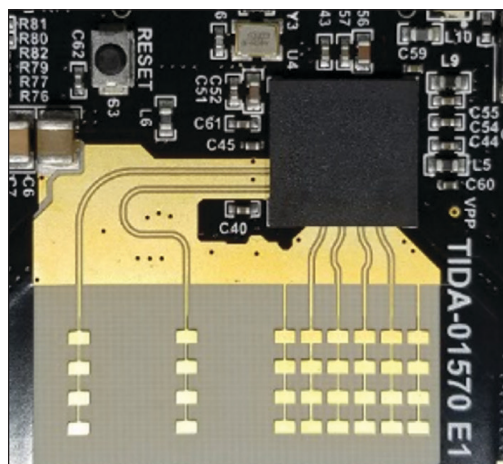


FIGURE 6

A 77 GHz antenna array on a printed circuit board (Source: Texas Instruments)

and far from each other. Now look at **Figure 6**, which is a reference design proposed by Texas Instruments (TI) for a 77 GHz antenna array for automotive radar applications. Do you see that higher frequencies means smaller elements and shorter distances between elements?

Now, how to fix the distance between elements more precisely? It can be set freely, but some distances are usually better than others. As explained above, waves cancel out if one wave is half a wavelength late compared to the other. That means that if the distance between two antenna elements is $\lambda/2$, and if both are fed by the same signal, then no power will be radiated in the lateral direction. The beam will be highly focused in one direction only. If the distance between elements is smaller than $\lambda/2$, then the main lobe will be wider. And finally, if the distance between elements is larger, for example λ or more, then several main lobes will appear.

A simulation will be helpful to explain this phenomenon. Look at **Figure 7**. Here the array is only a two-dipole antenna array, separated respectively by $\lambda/2.5$, $\lambda/2$ and λ . Based on this simulation, you will understand that $\lambda/2$ is usually a nice choice.

Now, how many elements are needed? Simple—a greater number of elements gives a narrower main lobe and a higher gain. As an order of magnitude, an antenna built with 10 elements can provide a gain of about 20 dB, whereas a 1,000-element array can provide 30 dB, which means a main lobe 10 times narrower in surface.

Another example will be welcome, won't it? Look at **Figure 8**. Here the antenna elements are still dipoles spaced by $\lambda/2$, but the number of elements are, respectively, 2, 6 and 20. If you check the simulated antenna pattern carefully, you will see that the main beam is narrower, and the gain is higher, when the number of elements is increased. The amplitude of the side lobes is reduced too. The downside is that the antenna size is larger. This is unfortunately a rule applicable to all antennas—usually a larger antenna is better than a miniature one.

A LOOK AT POWER

Up until now, all my simulations used a so-called “equipower weighting.” This means that the same power was applied to each antenna element. The input power P is split by N , and each of the N elements gets an input power of P/N . This approach is simple and provides a very narrow main beam, but at the expense of rather strong side lobes. Other power-splitting techniques allow us to reduce the side lobes, but at the expense of a slightly wider main lobe.

I illustrated two of the simplest methods in **Figure 9**: triangle power sharing and cosine

For detailed article references and additional resources go to: www.circuitcellar.com/article-materials
References [1] through [3] as marked in the article can be found there.

RESOURCES

Scilab | www.scilab.org

Texas Instruments | www.ti.com

power sharing. The principle is very simple. For triangle sharing, the power is linearly decreased from the center to the edge elements, whereas it follows a cosine shape in the last example. The difference is obvious in the figures, because

side lobes are greatly reduced in particular with a triangular weighting.

Dipoles—the most basic kind of antenna—radiate the same power in any direction on the horizontal plane. But what is the change

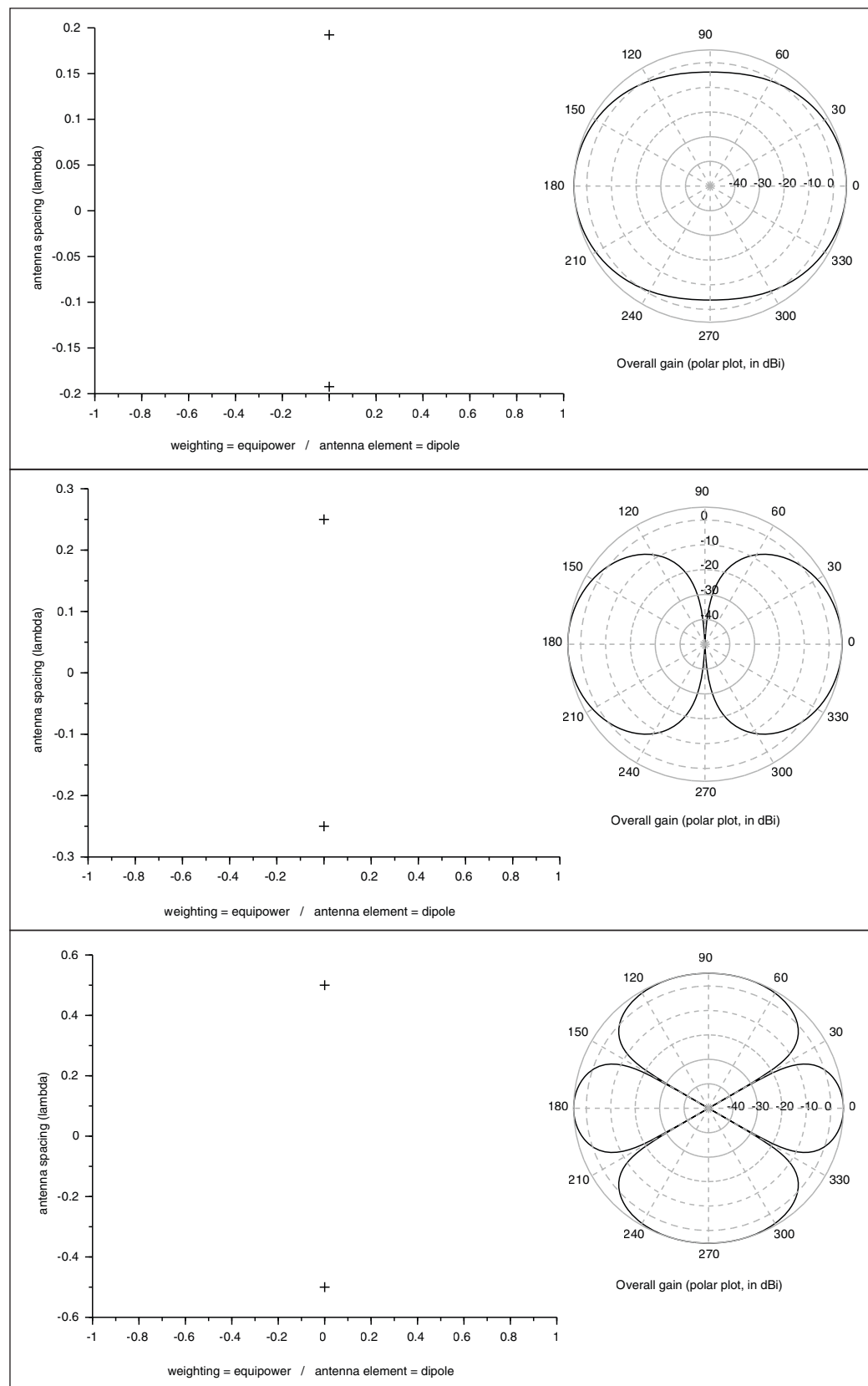


FIGURE 7 Effect of the distance between elements (from top to bottom, respectively, $\lambda/2.5$, $\lambda/2$, λ)

in the beam shape of an antenna array if the antenna element is changed to another type? In fact, the resulting antenna pattern is simply the pattern of the element alone, multiplied by the pattern of an array of dipoles (which is called “antenna array factor”).

Once again let’s consider an example, shown in **Figure 10**. The first is the antenna pattern of a 20-dipole array, whereas the second is the pattern of a 20-patch array. You can see that the main lobe and side lobe are identical, but the overall envelope is different,

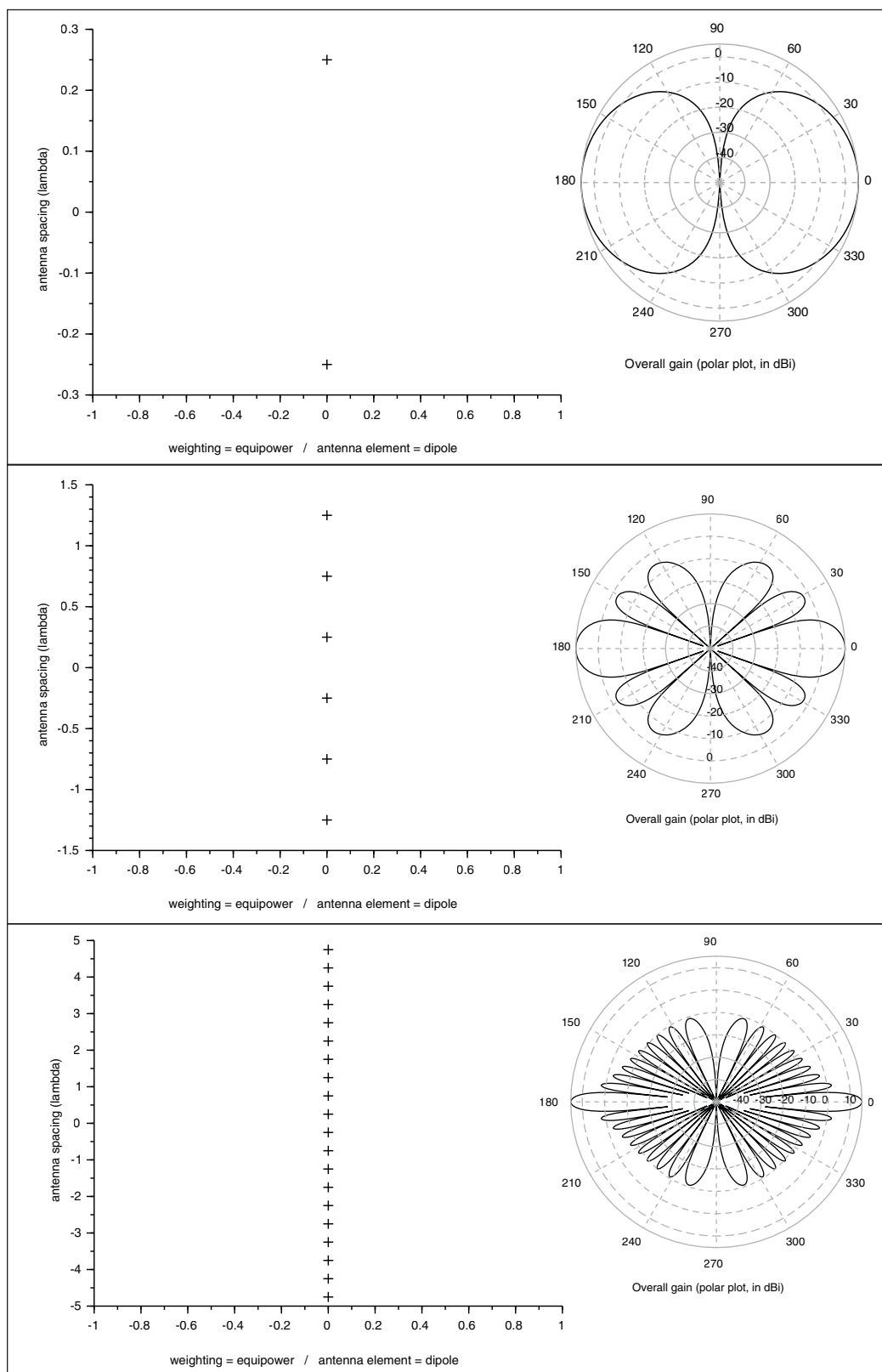


FIGURE 8

Effect of the number of elements (from top to bottom, respectively, 2, 6, 20)

and they closely map the antenna pattern of a single patch.

PHASE SHIFT & PHASED ARRAYS

Last but not least, let's talk about the phase of the feed network. Up until now, all antenna elements received a signal with exactly the same phase. What happens if this is not the case? The interferences will not be the same, and the antenna pattern will be distorted. Could it be useful? You bet it could! The first practical example is achieved when the same phase shift is applied between antennas—for example, a 10-degree phase shift between elements 1 and 2, 10 degrees more phase shift between element 2 and 3, and so on. In that case, the phase shift simply rotates the main beam. If you don't trust me, look at **Figure 11**. That shows a simulation of a 20-element patch array with triangle power weighting, and with a phase shift

between elements of 0, 40 and 140 degrees, respectively. Impressive isn't it?

Now the last and most interesting aspect is that all these parameters, and in particular the phases, can be either fixed or dynamically modified if the feed network includes electronically controllable gain and phase adjustments. Such an antenna array with computer-controlled phases and amplitudes is called a "phased array," and allows you to electronically steer the main beam direction anywhere—just by changing amplitudes and phases of the signals injected into each element.

This results in an orientable antenna, without any movement of the antenna structure by itself, and it allows steering the main beam in tens of nanoseconds. Moreover, providing several lobes simultaneously is as easy as a single lobe. Think about it. Such an antenna is a linear system, so you simply have to add the injected signals for each of the lobes you want.

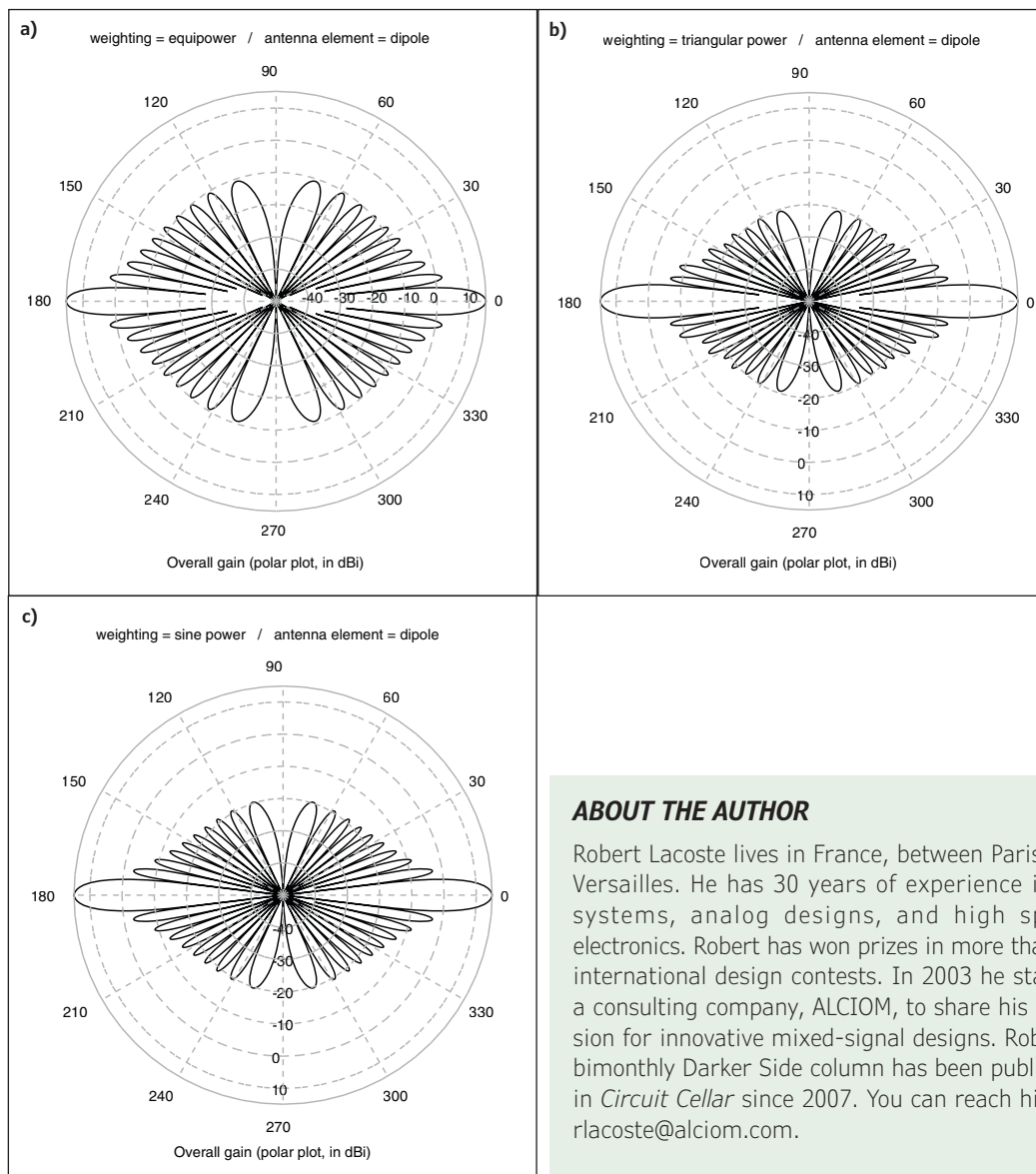


FIGURE 9 Effect of the power weighting method (respectively, (a) isopower, (b) triangle, (c) cosine)

COLUMNS

ABOUT THE AUTHOR

Robert Lacoste lives in France, between Paris and Versailles. He has 30 years of experience in RF systems, analog designs, and high speed electronics. Robert has won prizes in more than 15 international design contests. In 2003 he started a consulting company, ALCIOM, to share his passion for innovative mixed-signal designs. Robert's bimonthly Darker Side column has been published in *Circuit Cellar* since 2007. You can reach him at rlacoste@alciom.com.




WRAPPING UP

Phased arrays were invented for military radar systems, because they can quickly be targeted anywhere. According to *Wikipedia*, the largest phased array radar on earth is the so called “SBX-1,” a US defense 1,800 metric ton radar, built on a movable sea-based platform and including no less than 45,000 elements.

Antenna arrays and phased arrays also have plenty of non-defense applications.

Radio astronomy is an example. Look, for example, at the Very Long Baseline Array (VLBA), a set of 10 antennas built on an 8,000 km base worldwide. Ham radio guys are also fans of short-wave antenna arrays to increase the overall directivity and gain of their transceivers. Last but not least, look at your Wi-Fi router. Do you see that it has at least two identical antennas? Those aren't exactly an antenna array but rather a multiple input, multiple output (MIMO) system—but they are in fact closely linked techniques. However, I'm afraid that will would have to be the subject of another article.

Now, what can you do with all this information? First, I encourage you to download and play with my small array simulation software. Tweak the parameters, run the simulations—which are very fast—and discover by yourself all the details. Then, why not try using an antenna array in your next project? For example, if you are working on a robot, why not try finding its base with a phased array? Just fix a transmitter on the base, and “scan” the phased array to locate the direction. Or try using the same technique with lower frequencies—sound, for example. Take ten loudspeakers, drive them with computer-controlled phase and amplitudes, and you will be able to focus a sound in any direction. It may not be the simplest project, but for sure you will learn a lot! And you will have fun! 

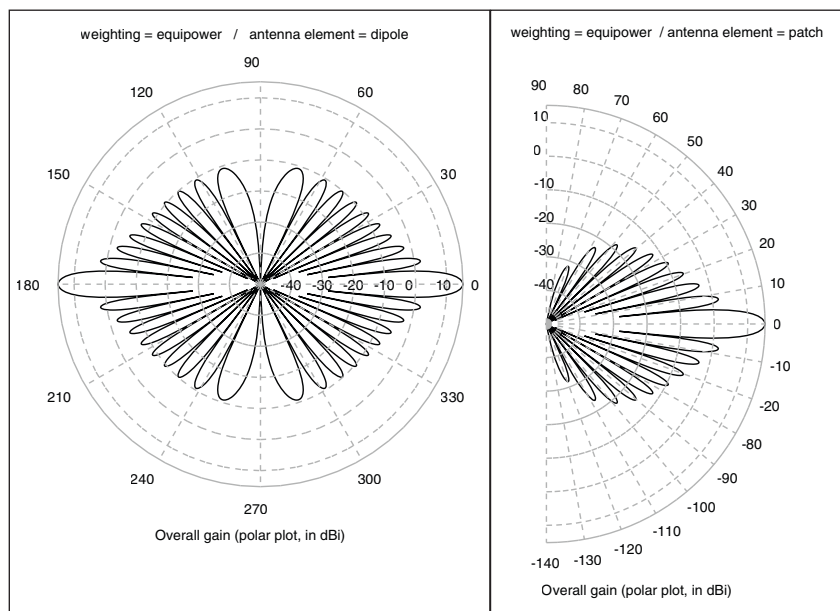


FIGURE 10
Effect of the antenna element design (respectively, dipole and patch)

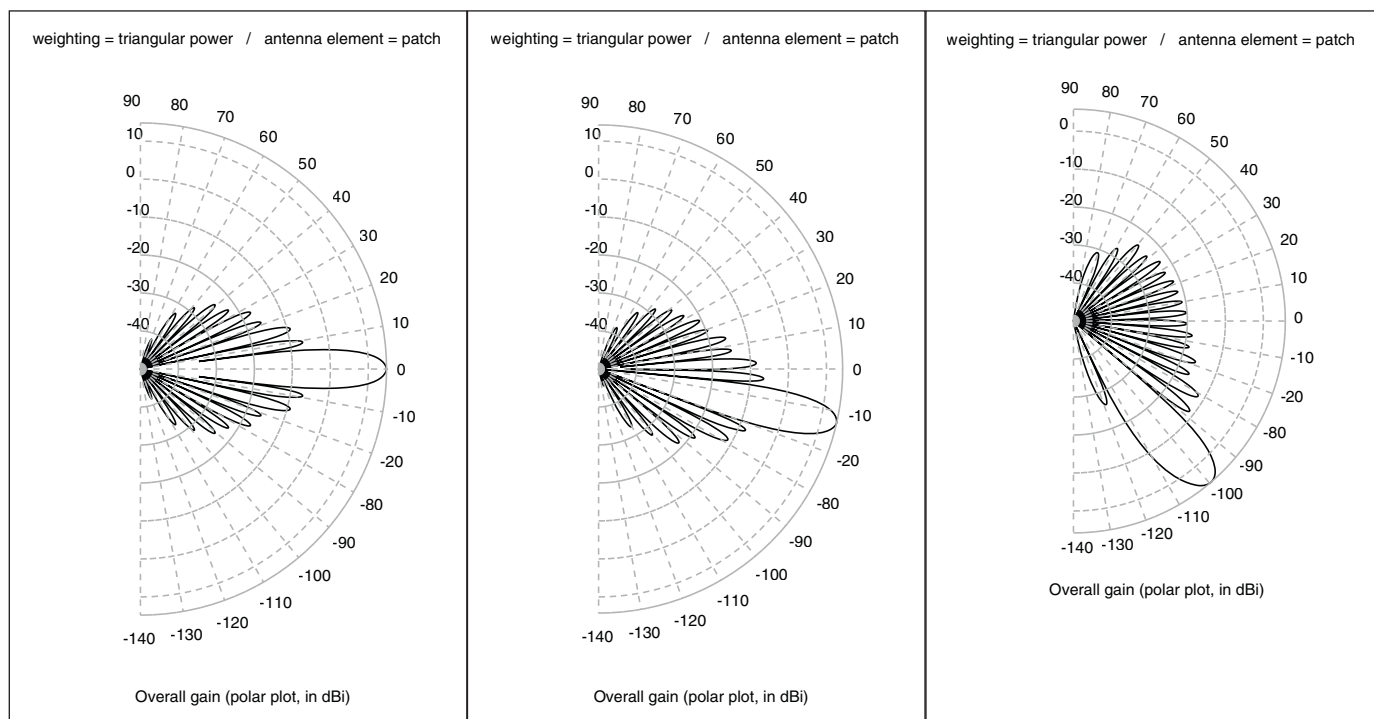


FIGURE 11
Effect of the phase shift between elements (respectively 0, 40, 140 degrees)



Messe München

Connecting Global Competence

November 12–15, 2019



Accelerating Innovation



co-located event



productronica 2019

World's Leading Trade Fair for Electronics
Development and Production

November 12–15, 2019, Messe München

productronica.com

Embedded in Thin Slices

LoRa (Part 1)

Where Does This Featherweight Fit In?

In this new article series, Bob discusses LoRa—the Long Range spread spectrum modulation technique that promises to solve a number of the key issues in fulfilling the wireless IoT requirements. In Part 1, Bob starts with an introduction to LoRa, looking at what it is, what are its limitations and how those limitations affect how we use this technology.

By
Bob Japenga

I had several friends who wrestled in high school. One friend in particular was a little wiry guy who weighed about 120 pounds. As some of you may know, high school wrestlers compete in as many as 14 weight classes. Each weight class covers a 6- or 7-pound range. My friend was always trying to get to the lowest possible class. To facilitate this, before the weigh-in, he would fast and binge on Ex-Lax. It always seemed to defeat the advantage of the lighter class by coming into the match weakened by those actions.

This begins a new article series on LoRa—the Long Range spread spectrum modulation technique that promises to solve a number of the key issues in fulfilling the wireless Internet-Of-Things (IoT) hype—especially low power and long range (**Figure 1**). For purposes of this article series, I will be talking about LoRa devices communicating on a LoRaWAN (LoRa Wide Area Network). In this series, I won't make a distinction and I'll just call them both LoRa. Jeff Bachiochi laid a great foundation for understanding LoRa in two articles in his June 2017 (*Circuit Cellar* 323) [1] and July 2017 (*Circuit Cellar* 324) [2] articles. In part 1, Jeff provided some of the technical details about chirp spread spectrum and how the data packets are formed in the RF cloud. In particular, he discussed how LoRa achieves such range. If you want to dig even deeper into "LoRa Modulation Basics," check out the Semtech Application Note by the same name [3].

At the end of part 1 and in part 2, Jeff went into the details of creating a peer-to-peer test

module for verifying the range using a single wire for an antenna. I will try to not cover ground already covered by Jeff, but rather start at a high level and work down into more details as the series progress. In particular this month, I want to ask the question: "Is LoRa like my high school buddy? Is it trimmed down so much that it has lost its ability to fulfill the objectives we require to create wireless IoT devices?" So, this month, we will briefly look at what LoRa is. Then we will look at the practical limitations to achieve these amazing distances at such low power. Finally, we will look at the kind of applications for which this very low bandwidth will be ideally suited.

WHAT IS LoRa?

The LoRa technology was patented by a French company and later acquired by Semtech. The proprietary nature of this Long Range and Low Power technology affects the costs of LoRa chipsets when compared to LTE Cat M1 and other competing wireless technologies. We have done bill-of-materials for competing designs, and the LoRa chipsets are more expensive. We will talk more about this in a later article. Unlike Bluetooth Mesh which achieves long range through a mesh network, LoRa achieves its range in a star of stars type network. Many devices that connect to one gateway and many gateways connecting to the cloud (**Figure 2**). And the range specifications are impressive (more than 10 km in rural areas). In my tests, as well as in the tests documented by Jeff in

Part 2 of his *Circuit Cellar* article, the real-world numbers—though not being close to the rural area numbers—are still impressive. Again, more on that in a later article.

LoRa uses sub-gigahertz license free radio frequency bands: the 868 MHz band in Europe, 915 MHz in North America and 433 MHz for Asia. This means that when you design your LoRa devices and your LoRa gateway, you will need to select the radios for your target area. It also means that you cannot have one device that will work in all regions.

Although LoRa devices could be used in a peer-to-peer network, the primary intent of the creators was for the LoRa devices to have a means of communicating to the cloud. For LoRa devices to communicate to the cloud, the LoRaWAN protocol was specified and is used by a number of cloud-based LoRa networks. There are three approaches to getting the data from your LoRa devices to the cloud: a private network that you create as part of your system design with your own LoRa gateways; a commercial network like MachineQ being deployed by Comcast, which gives you access to their gateways for which you pay for a data plan like you do a mobile data plan; and a community and open-source network like The Things Network (TTN), which



FIGURE 1

LoRa is the Long Range spread spectrum modulation technique created to solve a number of the key IoT issues—especially low power and long range.

provides free access to local gateways. We will explore these options in a later article.

Finally, there are three classes of traffic defined for LoRa [4]: Class A which is used for applications like those in Smart City devices that send data just a few times a day and the devices do not listen except after a transmission (**Figure 3**); Class B which is used for applications like irrigation that require the cloud to turn valves on or off with minimal latency—typically powered by batteries (**Figure 4**); and Class C which is used in applications like Smart Lighting where the LoRa device is constantly listening for pings to facility low latency actuation and is typically powered by the power grid (**Figure 5**).

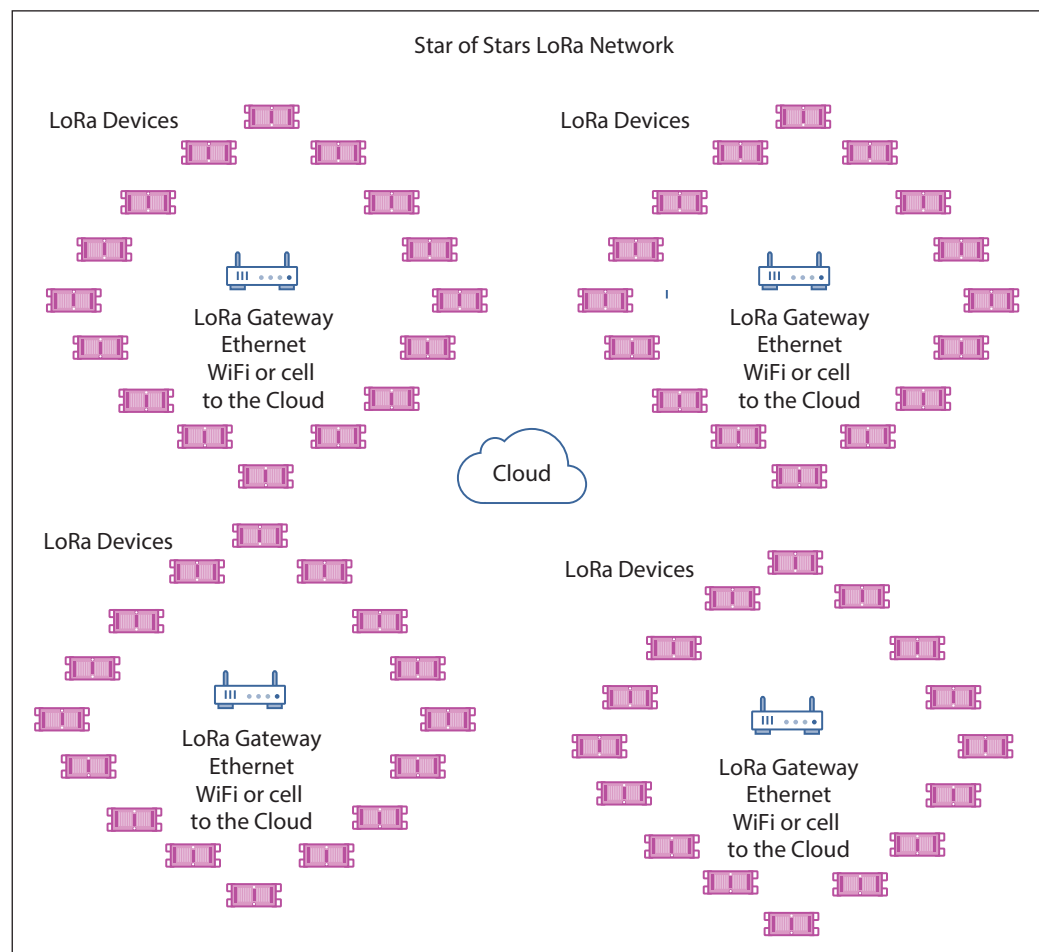


FIGURE 2

LoRa achieves its range in a star of stars type network. Many devices that connect to one gateway and many gateways connecting to the cloud.

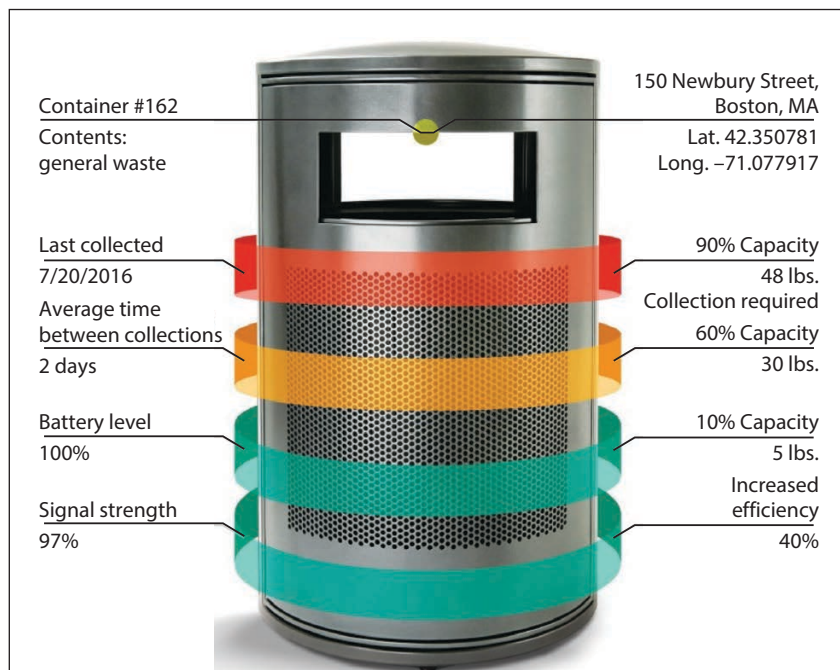


FIGURE 3

LoRa Class A: For applications like those in Smart City devices that send data just a few times a day and the devices do not listen except after a transmission.

RANGE AND BANDWIDTH

A term that is used to help us understand the theoretical limits of the range is the concept of the link budget, which is used in both wired and wireless communication systems. A link budget provides a means to take into account all of the gains and losses from the transmitter to the receiver. A very simplified equation for a link budget is to add all of the gains (in decibels) in the system to the transmitted power and subtracting all of the losses (in dB). There are some very good link budget calculators on the web [5]. From this you can see the theoretical distances that can be achieved with LoRa, which starts with a link budget of 154 dB. In comparison, the link budget for Cat M1 is 146 dB. Play around with the calculator and see how various factors like rainfall and elevation of the antenna affect the range. When you add cable attenuation and walls and trees and building attenuation you can get a much more realistic value for the maximum distance.

One of the questions that we should ask

is: How does LoRa achieve such long range using such low power transmitters? Have they invented the IoT equivalent of the free lunch? No! LoRa achieves the long distances by using a very low bandwidth. The bandwidth for a Cat M1 device is 20 MHz. The maximum bandwidth of a LoRa device is configurable at 125 kHz, 250 kHz and 500 kHz. Another way to look at it is that the maximum data rate for a Cat M1 device is of the order of 10 Mbps. For LoRa in North America, the raw maximum data rate is 27 kbps. With this reduced data rate comes smaller payload sizes as well.

Practical concerns can become legal concerns, which further reduce the practical bandwidth. In Europe, devices using the LoRa frequencies are only allowed to transmit 1% of the time. This significantly reduces your data rate by a factor of 100. Practically—since LoRa uses the same frequencies to send and receive—it is not just the legal concerns that reduce the data rate, you need your LoRa devices to leave a lot of dead air.

Once you choose the network you will use to connect to the cloud, you will be limited by their rules and guidelines. For example, if you use TTN, it is recommended that you only transmit 10 downlink messages a day (including uplink ACKs!). Further reductions happen due to the recommended maximum payload of 12 bytes. This translates to a downlink data rate of 0.001 bps!

PRACTICAL USE CASES FOR LoRa

At the end of June, 2019, the Wikipedia article on LoRa stated the following:

“Semtech’s LoRa devices and wireless radio frequency technology (LoRa Technology) is a long range, low power wireless chipset that has become the de facto technology for Internet of Things (IoT) networks worldwide” [6].

The de facto technology for IoT networks worldwide? This is not even close to being true. I complained and Wikipedia removed this statement [7]. Don’t get me wrong. I was very impressed with the Long Range and the Low Power of LoRa in my test setup. LoRa does have a bright future for certain IoT networks. But LoRa is far from the “de facto technology for IoT networks worldwide.” And the reason is that most IoT devices cannot live with the data rates allowed.

Using TTN as representative of a public network, where do these limitations affect us in the real world? According to TTN’s Fair Access Policy, you should limit your air time to 30 seconds per day. For the maximum range Spread Factor, this amounts to 200 bytes per day for sending data up to the cloud. The Fair Access Policy will limit the amount of data you

For detailed article references and additional resources go to:

www.circuitcellar.com/article-materials

References [1] through [8] as marked in the article can be found there.

RESOURCES

Semtech | www.semtech.com

can send down to your device to less than 50 bytes per day.

Let's assume that your application can tolerate this data budget. (At our company, we have yet to create an IoT device that could get by with so little data). But this leaves you with a much bigger problem. How do you handle Over-The-Air (OTA) software updates? An article in the *IEEE Communications* magazine for January 2017 documented some of the limitations of LoRa [8]. It is a useful article and I recommend it for determining the payload limitations of LoRa. But it didn't address the need for OTA software updates. How do you perform an OTA software update when you are limited to 50 bytes per day? A small 16k program would take almost a year to update.

Just recently, we worked with a company that was looking to create a Smart City application. Their device was fairly simple. It was detecting only one thing. It really only needed an uplink. They would typically deploy about 500 of them in city of 200,000 people. The city they were targeting already had a TTN network and they really wanted to leverage this existing network. They wanted to send 24 packets of 10 bytes every day. This is close to the TTN's policy and their specification could be tweaked to meet the policy. However, sending the raw sensor data to the cloud would far exceed 10 bytes. The project assumed that the device would perform a complicated, sophisticated and proprietary algorithm on the sensor data. I was not convinced that this algorithm could be perfected at the time of roll out to preclude requiring an OTA software update over the life of the product. And as we mentioned, OTA software updates are DOA with LoRa—at least on TTN.

If the sensors and actuators in your device are simple and the algorithms are in the cloud, I think you could take a chance on never having to update the software. And there are plenty of IoT examples where this is the case. Even if you design your own private gateway, to be able to do OTA software updates you would need to significantly reduce the range (by reducing the Spread Factor—more on that in a future article) and reduce the number of devices going to the gateway. The economics of scale begin to take their toll when we do that. Where I think LoRa works best is very low data rate projects with such simple software that you can guarantee that you will never need to do an OTA software update—or you have another means of doing software updates.

CONCLUSIONS

LoRa technology provides some amazing long-range wireless data transmissions at amazingly low power. But like my friend in



FIGURE 4


LoRa Class B: For applications like irrigation which require the cloud to turn valves on or off with minimal latency—typically powered by batteries.



FIGURE 5

LoRa Class C: For applications like Smart Lighting where the LoRa device is constantly listening for pings to facility low-latency actuation and is typically powered by the power grid.

high school, trimming down comes at a cost. As embedded system designers, we need to be aware of the costs up front. In my opinion, the future is not smaller and smaller programs that can be verified and expect to never be updated. The future is larger and more complex software packages that have flaws lurking in them. Often with sensors we cannot push the processing off to the cloud. We know that the IoT application has to be pretty simple for us to deploy without OTA software updates. And few are small enough to use LoRa's range at the expense of OTA software updates.

There still is an important place for LoRa in our IoT tool box. And so next month we will delve deeper into LoRa and how it can be used in your IoT project. But, of course, only in thin slices. 

ABOUT THE AUTHOR

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



From the Bench

MQ Telemetry Transport (Part 1)

Going Aloft with Mosquitto

Better known by the acronym MQTT, this lightweight messaging protocol is designed to minimize network bandwidth and device resource requirements. In this article, Jeff sets out to use MQTT via a cloud setup that he can do locally. For this, he turns to Eclipse Mosquitto, an open source message broker that implements the MQTT protocol. Jeff steps through the nitty gritty details of his implementation.



By
Jeff Bachiochi

Message Queuing Telemetry Transport. You probably know it as MQTT. It's a lightweight messaging protocol designed to minimize network bandwidth and device resource requirements. Its publish/subscribe protocol allows a server to handle all data dissemination. In its simplest form, the server can log data published from an external device, called a "node." A node requires minimal code to packetize its data before sending it to the MQTT server. Multiple nodes can send their data to the same MQTT server. The server is capable of handling a large number of nodes or clients. Packets describe how the enclosed data payload should be handled. Today, there are 16 different types of MQTT command packets. In the simplest system, we may only need three of these—Connect, Disconnect and Publish.

MQTT packets use the Transmission Control Protocol/Internet Protocol (TCP/IP) for communications. The Internet Protocol suite uses ports to identify a network service. Your HTTP Internet activity is most likely directed to port 80. Most MQTT activity uses port 1883. The use of ports simplifies the distribution packets. The Connect and Disconnect packets let the

MQTT server know who can be expected to require attention. The server will disregard any packets from clients that have not registered or those who have unregistered. The Publish packet holds the actual data that the server must deal with. If one or more clients only publishes data, then the data aren't used. There must be at least one subscriber to these data for any real work to be accomplished. Subscribing is the fourth packet type that makes this all worthwhile.

In many cases you will want to record the published data to a log file. This requires a client to subscribe to the same data being published by Client 1. Why go through all of this bending over backward just to log data that could be handled locally on the external node? If that simple scenario handles your project, then I'd take that tack. However, if you look at a bigger picture here, I'll bet you have more than the one IoT project. You may be using ThingSpeak, or any of the other cloud managers to retrieve and store data from your project. While these are convenient and may have some free storage, many require a monthly maintenance fee. I'm looking for a cloud I can implement locally. Enter Mosquitto.

Packet Type	Packet Value	Direction of Flow	Payload	Description
Reserved	0	-	-	Reserved
Connect	1	Client to Broker	Required	Connect Request
Connack	2	Broker to Client	None	Connect Acknowledge
Publish	3	Both	Optional	Publish Request
Puback	4	Both	None	Publish Acknowledge (QOS1)
Pubrec	5	Both	None	Publish Received (QOS2)
Pubrel	6	Both	None	Publish Release (QOS2)
Pubcomp	7	Both	None	Publish Complete (QOS2)
Subscribe	8	Client to Broker	Required	Subscribe Request
Suback	9	Broker to Client	Required	Subscribe Acknowledge
Unsubscribe	10	Client to Broker	Required	Unsubscribe Request
Unsuback	11	Broker to Client	Required	Unsubscribe Acknowledge
Pingreq	12	Broker to Client	None	Ping Request
Pingres	13	Client to Broker	None	Ping Acknowledge
Disconnect	14	Both	None	Disconnect Notification
Auth	16	Both	None	Authentication exchange

MOSQUITTO, AN MQTT BROKER

Eclipse Mosquitto is an open source (EPL/EDL-licensed) message broker that implements the MQTT protocol. Many brokers provide cloud-based testing for your project, including test.mosquitto.org. It is easy to move from an external cloud broker to a locally run broker. I have installed Mosquitto on one of my PCs. Mosquitto is available for Windows, Linux, Mac and Raspberry Pi. You can use external testing to get comfortable with the processes discussed here.

Let's spend a bit of time on the MQTT protocol before installing a simple client device. The MQTT protocol is communication of topics between a client and a broker. The broker (or server) handles the collection and dissemination of topic data. While multiple brokers can be connected, just a single broker is required for most systems. Clients are devices that communicate with a broker. A client may collect sensor data (for example, topic = switch status, topic = temperature, topic = position) and pass it on to the broker (publish a topic). A client may receive particular sensor data (subscribe to a topic) and perform some function (control a digital or analog output) based the topic's data. A client can be a publisher and/or a subscriber.

Unlike HTTP, which opens and closes communication for each request, the MQTT Broker controls opening and closing a connection over time. Each MQTT packet contains at least two parts—a fixed header and a variable header. Certain commands require a third part—the payload. **Table 1** shows these, with a brief description of each. Packets have three levels of confirmation.

A QOS "0" (Quality of Service, level zero) request receives no acknowledgement. A QOS "1" request receives an acknowledgement. A QOS "2" request receives an acknowledgement plus additional packets to acknowledge the acknowledgement! You can choose your level of confirmation based on your requirements.

TABLE 1
MQTT packets efficiently pass data between a Client and Broker.

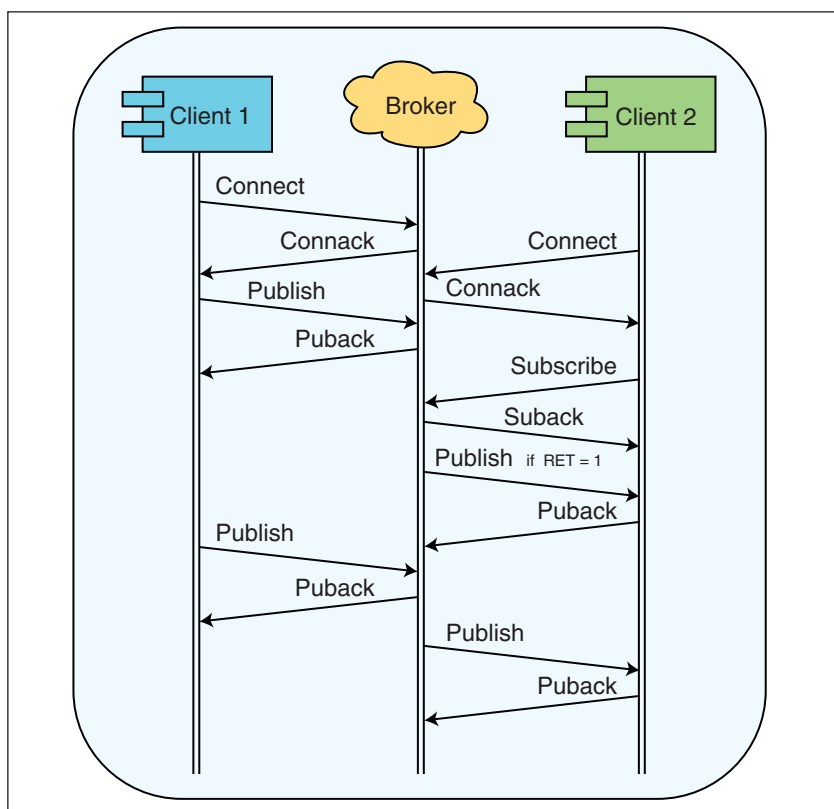
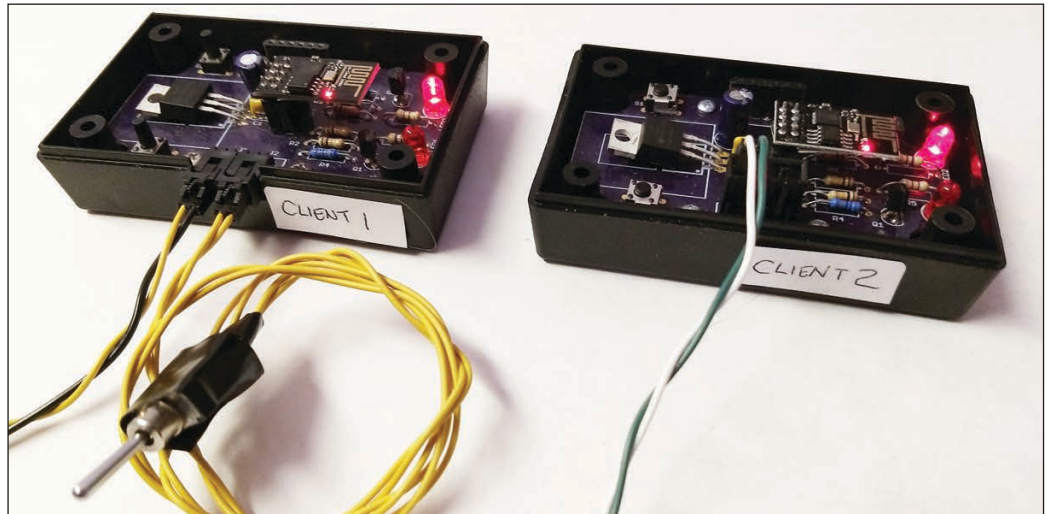


FIGURE 1
Typical command flow between clients and an MQTT broker such as Mosquitto.

FIGURE 2

Two identical circuits hosted by ESP8266 Wi-Fi modules log into my local network and connect to a Mosquitto MQTT server hosted on one of my PCs. Client 1 publishes its switch status to an MQTT topic upon any change of state. Client 2 subscribes to the MQTT topic, which instructs its output to turn ON or OFF based on the published switch status.



Besides the QoS, a publish packet contains several other flags that can be useful. The Retention flag requires the broker to hold onto this topic's data after it finishes publishing it to any clients that have subscribed to the topic. This allows new subscribers to get the last known state of the topic whenever

they subscribe. The Will (as in "last will and testament") flag requires the broker to advise all subscribers that the publish client has left or disconnected. **Figure 1** shows typical MQTT communications between the clients and the broker.

SIMPLE DEMO

About a year ago, I presented a two-part project entitled "Texting and IoT Embedded Devices," using an Espressif Systems ESP8266 to monitor the state of an input and send a text to my phone when a change occurred (*Circuit Cellar* 332 and 333, March and April 2018). I'll use the same circuit here as clients communicating with a Mosquitto broker. The first will be the input client and a second the output client (**Figure 2**). I like using the ESP8266 because it can be programmed as if it were an Arduino using the Arduino IDE. You can use the `pubsubclient` library to handle all the necessary packet issues. The ESP-01 has one I/O that can be used if you exclude TX, RX, enable, reset, power and ground. Client1 will use IO2 as a (switch) input. Inputs can have an internal pull-up resistor enabled, which will hold the input in a high state. In this example, I'll be using a toggle switch placed between the input and ground. When the switch is open, the input is at a logic high, and when it is closed, it will short the input to ground, applying a logic low to the input. Client 1 will monitor the input state and report each change of state.

The broker uses "topics" to help compartmentalize different data messages. Each topic can use a tree structure of folders. For instance, my Client 1 node is an ESP8266 with a MAC of "xxxxxx11E9B0," so I'm using this as the top folder. I'm using one subfolder to divide the client by pin function, and another to narrow this function down to a single pin number. The topic will therefore

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
//*****
// Network info
//*****
const char* ssid = "your network";
const char* password = "network password";
const char* mqtt_server = "192.168.1.216"; // pc=216,
openhanian=194
const char* switchTopic = "ESP_11E9B0/DIN/IO2";
//const char* lightTopic = "ESP_A3D0DF/DOUT/IO2";
```

LISTING 1

This code includes the necessary library files and defines the network information.

```
//*****
// setup
//*****
void setup()
{
  pinMode(switchPin, INPUT_PULLUP);
  // Initialize IO@ as a switch input using an internal
  pull up resistor
  switchState = digitalRead(switchPin);

  // Initial state of the switch
  msg[0] = (char)digitalRead(switchPin) + 0x30;
  // Initial message
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}
```

LISTING 2

Some initial setup code including setting input pin mode and the using callback function.

become “ESP_11E9B0/DIN/IO2”. This is all up to the user—I could have simply named this topic “Client1.” Client 1 will publish a message (the status of the input of IO2) using this topic. In this case, the status is simply the logic state of the input, a “1” if the input is high, and a “0” if the input is low. Sending a binary 1 or 0 is fine, but you won’t be able to “see” it (if you look at a data transmission or log it to a file), because these are non-printable characters. I chose to make them ASCII characters by adding 0x30. This means the message is either character 1 or 0, rather than binary 1 or 0. Messages can be in any format—even JSON (JavaScript Object Notation), if you wish.

CLIENT 1

Let’s look at the highlights of the Client 1 program written on the Arduino IDE for the ESP8266. The first thing we need to do is include the necessary library files and define the network information as shown in **Listing 1**.

The MQTT_server is Mosquitto running on my PC, so this address is my PC’s IP address on my network. Note: The alternate IP listed here is for openHAB’s openHABian, an open source home automation program for a Raspberry PI. This also has an MQTT broker built in, which means all your MQTT communications are accessible to it. For simplicity, we’ll stick to using the Mosquitto broker installed on my PC. Next you see the definition of the switchTopic (Client 1), so we can publish. The program requires some initial setup (**Listing 2**).

The mode of the input pin is defined and the state of the input is saved as the first message. The setup_wifi() function connects to your network. The client.setServer(mqtt_server, 1883) function uses the pubsubclient library

```
//*****
// MQTT reconnect
//*****
void reconnect()
{
  // Loop until we're reconnected
  while (!client.connected())
  {
    // Create a client ID
    char* clientId = "ESP8266Client-11E9B0";
    // Attempt to connect
    if (client.connect(clientId))
    {
      switchState = digitalRead(switchPin);
      // publish
      msg[0] = switchState + 0x30;
      client.publish(switchTopic, msg);
      // resubscribe
      client.subscribe(lightTopic);
    }
    else
    {
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

LISTING 3

The reconnect() function, shown here, connects to the server. This function will also be called to reestablish the MQTT connection, should it ever be lost.

to connect to Mosquitto (which must be running on your PC) at port 1883, the standard port for the MQTT protocol over TCP. The last function, client.setCallback(callback), lets the pubsubclient library know that it must use the user’s callback function if it receives any data. What is this callback function? This is only used when a client subscribes to a topic. For now, Client 1 is only publishing, so let’s move on to the MQTT connection routine.

Previously we set up the Wi-Fi and MQTT server info. Now we need to connect to the server. This function will also be called to reestablish the MQTT connection, should it ever be lost. Therefore, I’m naming it reconnect(), and this is shown in **Listing 3**. To make sure this device is easily identified, I’m using part of the client’s MAC address as its client ID. Once a connection is established, the switch state is sampled, a message is created, and the initial switch state is published to the broker.

And now we move on to the main loop, shown in **Listing 4**. Two basic things are happening here. The first monitors the MQTT connection status. If we only publish when a change occurs, then there may not be any transmissions for a while. The original connection request carries with it a socket

```
//*****
// main loop
//*****
void loop()
{
  if (!client.connected())
  {
    reconnect();
  }
  client.loop();
  //
  newState = digitalRead(switchPin);
  if (switchState != newState)
  {
    switchState = newState;
    // publish
    msg[0] = (char)switchState + 0x30;
    client.publish(switchTopic, msg);
  }
}
```

LISTING 4

Code for main loop is shown here.

```
//*****
// setup
//*****
void setup()
{
  pinMode(lightPin, OUTPUT);

  // Initialize IO2 as an output
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}
```

LISTING 5

Shown here, the setup() function is similar to that of Client 1, except we are defining IO2 as an output.

```

/*****
// MQTT reconnect
/*****
void reconnect()
{
    // Loop until we're reconnected
    while (!client.connected())
    {
        // Create a client ID
        char* clientId = "ESP8266Client-11E9B0";
        // Attempt to connect
        if (client.connect(clientId))
        {
            client.subscribe(switchTopic);
        }
        else
        {
            // Wait 5 seconds before retrying
            delay(5000);
        }
    }
}

```

LISTING 6

Client 2 will be subscribing to Client 1's publish topic, therefore the MQTT reconnect() function will differ slightly, as shown here.

```

/*****
// MQTT callback for subscriptions
/*****
void callback(char* topic, byte* payload,
unsigned int length)
{
    // Switch on the LED if an 1 was received
    as first character
    if ((char)payload[0] == '1')
    {
        lightState = 1;
    }
    else
    {
        lightState = 0;
    }
}

```

LISTING 7

This code shows the callback() function for subscriptions.

```

/*****
// MQTT loop
/*****
void loop()
{
    if (!client.connected())
    {
        reconnect();
    }
    client.loop();
    if (switchState != lightState)
    {
        lightState = switchState;
        if (lightState == 1)
        {
            digitalWrite(lightPin, HIGH); // Turn the port on
        }
        else
        {
            digitalWrite(lightPin, LOW); // Turn the port off
        }
    }
}

```

LISTING 8

This shows the code for the main loop for Client 2.

**ABOUT THE AUTHOR**

Jeff Bachiochi (pronounced BAH-key-AH-key) has been writing for *Circuit Cellar* since 1988. His background includes product design and manufacturing. You can reach him at:

jeff.bachiochi@imaginethatnow.com or at:
www.imaginethatnow.com.

timeout, and if the broker doesn't hear from the client within this period, then the broker initiates a *Ping* to see if the client is still there. As long as the client answers, the connection stays open (unless the client specifically requests a `pubrel`). `Client.loop()` must be called periodically, so the main loop gives up control, allowing the client to process commands received from the broker.

The last part of the main loop monitors the input and upon a change in state publishes this change to the broker. The `client.publish(switchTopic, msg)` command requires two parts, the topic and message. Here the topic is "ESP_11E9B0/DIN/IO2" and the message is the character 0 when OFF and 1 when ON.

The MQTT broker creates a temporary record of each topic's last message. If any client subscribes to it, it will (re)publish the message to that client. You don't need to do anything to the broker (Mosquitto) to make this happen, since this is the broker's intended purpose.

CLIENT 2

The program for Client 2 is basically the same as Client 1. We are including the same library files and defining the same network information. The `setup()` function (**Listing 5**) is similar to that of Client 1, except we are defining IO2 as an output. Because Client 2 will be subscribing to Client 1's publish topic, the MQTT `reconnect()` function will differ slightly, as shown in **Listing 6**.

This is a good time to discuss the `callback()` function, since we've just requested the broker to pass on any

information it receives on the `switchTopic` topic “ESP_11E9B0/DIN/IO2”. Originally, Client 1 published the state of the IO2 input using the function, `client.publish(switchTopic,msg)`. This is passed on to Client 2, which is directed by the `client.setCallback(callback)` function to use the `callback()` function, shown in **Listing 7**.

This function should be used to select a subscribed topic. Since we have only one topic here, we can simply look at the first character of the payload (received message). We know this will be a single ASCII character consisting of a 0 or a 1. The value of this character is assumed to be the state of Client 1’s switch. The `switchState` variable keeps track of this information.

```

C:\Program Files\mosquitto>mosquitto -v
1559056218: mosquitto version 1.6.2 starting
1559056218: Using default config.
1559056218: Opening ipv6 listen socket on port 1883.
1559056218: Opening ipv4 listen socket on port 1883.
1559056221: New connection from 192.168.1.220 on port 1883.
1559056221: New client connected from 192.168.1.220 as ESP8266Client-11E9B0 (p2, c1, k15).
1559056221: No will message specified.
1559056221: Sending CONNACK to ESP8266Client-11E9B0 (0, 0)
1559056221: Received PUBLISH from ESP8266Client-11E9B0 (d0, q0, r0, m0, 'ESP_11E9B0/DIN/IO2', ... (1 bytes))
1559056221: Received SUBSCRIBE from ESP8266Client-11E9B0
1559056221: ESP_A3D0DF/DOUT/IO2 (QoS 0)
1559056221: ESP8266Client-11E9B0 0 ESP_A3D0DF/DOUT/IO2
1559056221: Sending SUBACK to ESP8266Client-11E9B0
1559056222: New connection from 192.168.1.184 on port 1883.
1559056222: New client connected from 192.168.1.184 as ESP8266Client-A3D0DF (p2, c1, k15).
1559056222: No will message specified.
1559056222: Sending CONNACK to ESP8266Client-A3D0DF (0, 0)
1559056222: Received SUBSCRIBE from ESP8266Client-A3D0DF
1559056222: ESP_11E9B0/DIN/IO2 (QoS 0)
1559056222: ESP8266Client-A3D0DF 0 ESP_11E9B0/DIN/IO2
1559056222: Sending SUBACK to ESP8266Client-A3D0DF
1559056222: No will message specified.
1559056222: Received PUBLISH from ESP8266Client-A3D0DF (d0, q0, r0, m0, 'ESP_A3D0DF/DOUT/IO2', ... (1 bytes))
1559056222: Sending PUBLISH to ESP8266Client-11E9B0 (d0, q0, r0, m0, 'ESP_A3D0DF/DOUT/IO2', ... (1 bytes))
1559056236: Received PINGREQ from ESP8266Client-11E9B0
1559056237: Sending PINGRESP to ESP8266Client-11E9B0
1559056237: Received PINGREQ from ESP8266Client-A3D0DF
1559056237: Sending PINGRESP to ESP8266Client-A3D0DF
1559056253: Received PINGREQ from ESP8266Client-11E9B0
1559056253: Sending PINGRESP to ESP8266Client-11E9B0
1559056253: Received PINGREQ from ESP8266Client-A3D0DF
1559056253: Sending PINGRESP to ESP8266Client-A3D0DF
1559056661: Received PUBLISH from ESP8266Client-11E9B0 (d0, q0, r0, m0, 'ESP_11E9B0/DIN/IO2', ... (1 bytes))
1559056661: Sending PUBLISH to ESP8266Client-A3D0DF (d0, q0, r0, m0, 'ESP_11E9B0/DIN/IO2', ... (1 bytes))
1559056736: Received PINGREQ from ESP8266Client-11E9B0
1559056737: Sending PINGRESP to ESP8266Client-11E9B0
1559056737: Received PINGREQ from ESP8266Client-A3D0DF
1559056737: Sending PINGRESP to ESP8266Client-A3D0DF
1559056753: Received PINGREQ from ESP8266Client-11E9B0
1559056753: Sending PINGRESP to ESP8266Client-11E9B0
1559056753: Received PINGREQ from ESP8266Client-A3D0DF
1559056753: Sending PINGRESP to ESP8266Client-A3D0DF
1559056761: Received PUBLISH from ESP8266Client-A3D0DF (d0, q0, r0, m0, 'ESP_A3D0DF/DOUT/IO2', ... (1 bytes))
1559056761: Sending PUBLISH to mqtt_df611a7e.0a0248 (d0, q0, r0, m0, 'ESP_A3D0DF/DOUT/IO2', ... (1 bytes))

```

FIGURE 3

After Mosquitto begins listening, you can see Client 2 connecting and publishing its initial light state. Then Client 1 connects and publishes its initial switch state. The broker sends Client 2’s initial state, since Client 1 just connected, and then pings both clients because they’re idle. Each time Client 1’s switch changes state, the broker receives a publish and republishes it to Client 2. Client 2 publishes the fact that its state has changed, and the broker republishes this to Client 1.

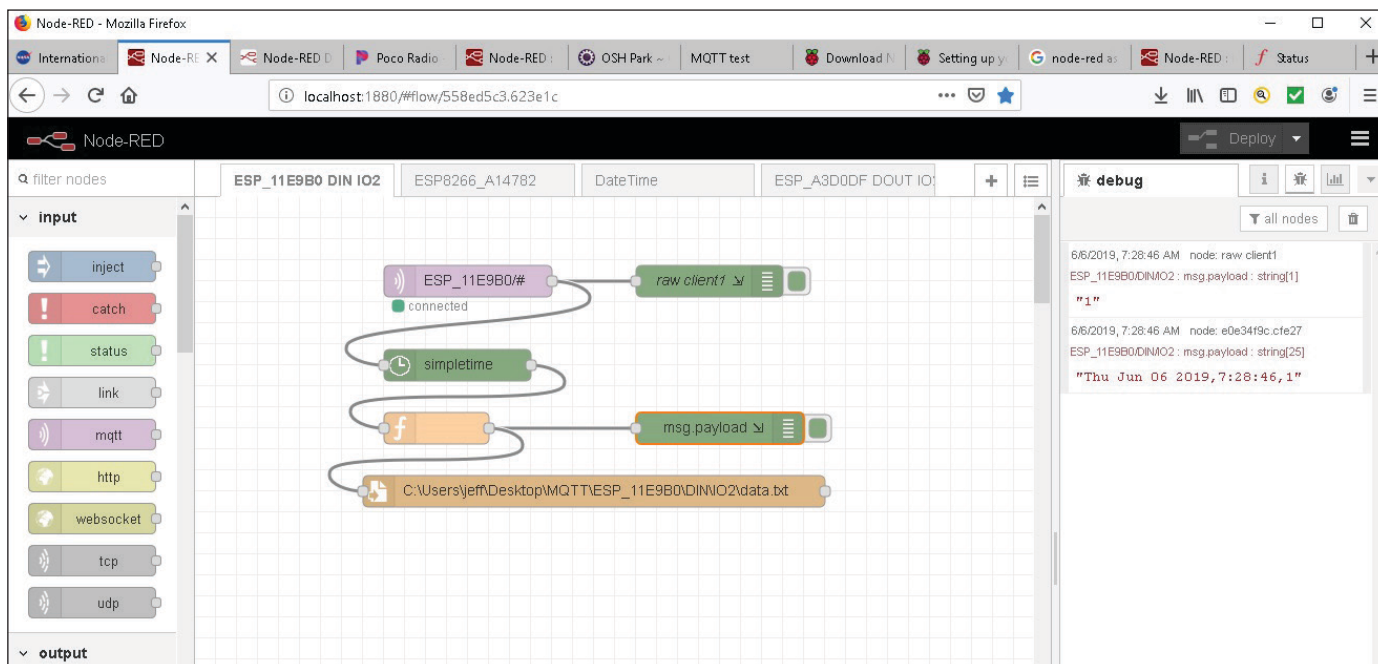


FIGURE 4

Node-RED is used as a local MQTT node that subscribes to MQTT topics and logs the messages (data).

Now on to Client 2's main loop, shown in **Listing 8**. The first part of the main loop takes care of the MQTT connection and is identical to that of Client 1. The last part of the main loop handles setting and clearing the (light) output. My circuitry has an LED that follows the state of IO2. It reflects the state of the output of Client 2 and the input of Client 1, so it can be used to monitor the actual state of IO2 on both clients.

We can call this "open loop" because there is no direct feedback that Client 1's switch state has actually affected Client 2's output. However, we are using "QOS"=2 for all this communication, which means the communication between Client 1 and the broker is assured via command handshaking. The communication between the broker and Client 2 also implements this. I can see this demonstrated once in a while, when Client 2's output fails to follow Client 1 immediately. It will, in fact, correct itself after a short delay.

Should the need arise, you can easily "close the loop" by having Client 2 publish a topic

with the state of its output, and have Client 1 subscribe to this topic. If you look back at the Network info section in Listing 1, you find the `rem'd out` statement `//const char* lightTopic = "ESP_A3D0DF/DOUT/IO2"`. Remove the `rem` slashes, and a new topic will be defined. Client 2 can then publish to this topic, and Client 1 can subscribe to it. This will pass back to Client 1 the state of Client 2's output, which can be used to confirm that Client 2's output state is the same as Client 1's input state.

MOSQUITTO

Go to mosquitto.org/download and follow the directions if you want to install Mosquitto on your PC. Mosquitto can be set as a service that is automatically run whenever your PC boots. Alternately, you can run it from the command line. If run from the command line using the `mosquitto -v` option, you'll get a feeling for what's going on in real time (**Figure 3**).

Unfortunately, Mosquitto does not supported logging to a file on the Windows system. We can get around this by adding the Node-RED application from www.nodered.org to your PC. Node-RED is a browser-based programming tool that looks like a client to a broker. In its simplest form, it can subscribe to an MQTT topic and log the topic to a data file. I'll use Node-RED here for local logging the publishes of Client 1 and Client 2. The file will record all activity of the clients.

If you want to install Node-RED on your PC follow the directions at nodejs.org/en. Node-RED can't be run as a service, so you can use the task scheduler to add this to the start-up procedure. If both Mosquitto and Node-

Additional materials from the author are available at:
www.circuitcellar.com/article-materials

RESOURCES

Espressif Systems | www.espressif.com

Mosquitto | www.mosquitto.org

MQTT | www.mqtt.org

Node-RED | www.nodered.org

openHAB | www.openhab.org

ThingSpeak | www.thingspeak.com

RED are running, type "localhost:1880" into any browser, and you will be able to add a local client to work with Mosquitto. **Figure 4** shows how easy it is to drag and drop tasks to subscribe to a topic, get the present date stamp, merge the two data streams and log the result to a file.

Note that the two tasks to the right in Figure 4 are taps into the data stream and allow the payloads (messages) to be printed to the debug screen. You can therefore see both the raw message and the message appended to the date/time stamp. This flow subscribes to any topic ("#") that begins with "ESP_11E9B0" (Client 1). Each message is appended to the file "data.txt" in the "MQTT\ESP_11E9B0\DIN\IO2" on my desktop. Another set of tasks can be added to subscribe to any topic ("#") that begins with "ESP_A3D0DF" (Client 2). The entries shown in **Listing 9** are from the two logs files showing both the actions of Client 1 (switch position) and Client 2 (light status).

You might not require a log file of the operations of this simple two-node circuit. However, these basics form a strong foundation for next month's article, which shows how I moved my IoT from an external cloud (ThingSpeak) to my local MQTT server.

MQTT has great flexibility. Multiple nodes can publish to the same topic, and multiple nodes also can subscribe to

```
from C:\Users\jeff\Desktop\MQTT\ESP_11E9B0\DIN\IO2 data.txt
Thu Jun 06 2019,7:27:16,0
Thu Jun 06 2019,7:28:11,1
Thu Jun 06 2019,7:28:41,0
Thu Jun 06 2019,7:28:46,1
```

```
from C:\Users\jeff\Desktop\MQTT\ESP_A3D0DF\DOUT\IO2 data.txt
Thu Jun 06 2019,7:27:16,0
Thu Jun 06 2019,7:28:11,1
Thu Jun 06 2019,7:28:41,0
Thu Jun 06 2019,7:28:46,1
```

LISTING 9

The entries shown in shown here are from the two logs files showing both the actions of Client 1 (switch position) and Client 2 (light status).

the same topic. If I want to control a light from multiple locations, I would use a momentary push button instead of a toggle switch with an LED, to indicate the present status of the light. This way any node could be used to toggle the output. Similarly, multiple lights, such as front and back porch lights, could be controlled from multiple locations if they both subscribe to the same topic.

So much to learn, so little time.

Note: Code for this project can be downloaded from Circuit Cellar's article code & files webpage. [E](#)

PicoScope® 2000 Series Oscilloscopes

Benchtop performance and power that fits in your pocket.



2-CHANNEL, 4-CHANNEL AND MSO MODELS

DECODE 20 SERIAL PROTOCOLS AS STANDARD

UP TO 128 MS BUFFER MEMORY

UP TO 100 MHz BANDWIDTH

USB CONNECTED AND POWERED

From
\$115

For more information visit www.picotech.com/A402

Email: sales@picotech.com. Errors and omissions excepted. Please contact Pico Technology for the latest prices before ordering

pico
Technology

The Consummate Engineer

Semiconductor Fundamentals (Part 2) Diodes in Detail

In Part 1, George examined the basic structures that make semiconductors work. But a lot more needs to be said about diodes, which are a key element of semiconductors. In Part 2, George dives deeper, this time looking at the current flow, depletion layer and electron physics that are involved in diode operations. He covers various types of diodes and the details of their operations.



By
George Novacek

Last month we looked at what happens when we join P and N materials, creating a P-N junction. To refresh your memory, free electrons from the N-atoms migrate across the junction and combine with the holes in the P-atoms, producing negative ions. This leaves positive donor ions in the N-material to which holes from the P-material tend to migrate. (See **Figure 1**, which was also shown in Part 1.) The result is that the thin layer of the P-material adjacent to the junction contains negatively charged acceptor ions, while the layer of the N material on the opposite side of the junction contains positive ones. This diffusion of electrons and holes continues until equilibrium is reached. A potential barrier around the junction is created, and the regions on either side of the junction, called the “depletion layer,” contain no free carriers.

If we apply a positive voltage to the P-side of the junction and negative voltage to the N-side, free electrons and holes will be given extra energy to overcome the potential barrier, and electric current will flow. When the polarity is reversed, the depletion layer increases in width, due to the lack of electrons and holes. Its impedance increases

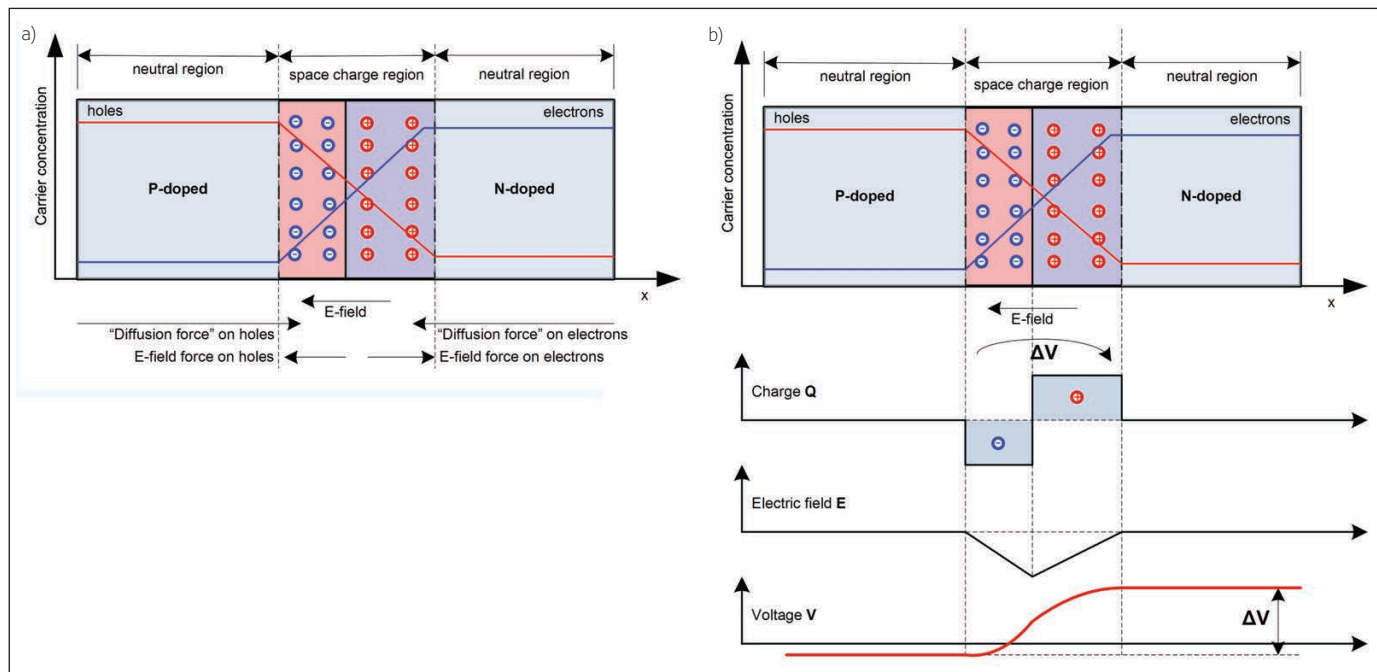
too, almost becoming an insulator, preventing current from flowing through the junction. A P-N junction, therefore, conducts electrical current in one polarity and blocks it in the opposite polarity. This is the basis of a solid-state diode operation.

DIODE BASICS

Diodes contain only one P-N junction. Nowadays, silicon is the most common material. Germanium and selenium were used in the past. You can also find diodes made of gallium arsenide. **Figure 2** is the I-V (current vs. voltage) diagram illustrating the principle of a diode operation. This characteristic is described mathematically by the Shockley diode equation:

$$I = I_s \left(e^{\frac{V_D}{nV_T}} - 1 \right)$$

where I is the diode current, V_D is the voltage across the diode, V_T is the thermal voltage and n is the ideality—also known as quality factor or emission coefficient. The n factor is usually between 1 and 2, based on the type of the diode and its manufacturing process. The thermal voltage V_T is approximately 25.85 mV



at 300°K (26.85°C = 80.334° F). It is defined as:

$$V_T = \frac{kT}{q}$$

where k is the Boltzmann constant (1.380649 × 10⁻²³ J/K (joules per Kelvin)), T is the absolute temperature and q is the magnitude of an electron charge (1.602176634 × 10⁻¹⁹ coulomb) [1].

Referring again to Figure 2, a small forward voltage results in a small forward current, until a knee voltage is reached—typically 0.7 V for silicon diodes. After that, the current will grow quickly, depending on the diode’s internal resistance which, in most cases, is very small. When reverse voltage is applied, a reverse current (leakage)—which depends on the device and varies with temperature—will flow. The leakage, when compared to the forward current, should be negligible. At some level, the reverse voltage will cause a breakdown. Exceeding the reverse breakdown voltage can damage many devices, while some, such as Zener diodes, take advantage of it. Zener diodes use this effect for voltage limiting or stabilization. Ideally, the internal resistance of a Zener diode in its avalanche region should be zero. But, because it’s not, the stabilized voltage will change slightly with the current.

There are many types of diodes tailored for a specific function. The most prevalent are ordinary signal and rectifier diodes used to conduct current in one direction only. The drawback of silicon diodes is their forward voltage of about 0.7 V—depending on the

device and the current. It affects circuits for low-level signals and power rectifiers, where it causes increased heat dissipation.

This problem is reduced by Schottky diodes with the forward voltage drop of about 0.15 V to 0.45 V. Schottky diodes are constructed from a metal-to-semiconductor contact, exhibiting faster reverse current recovery. Their capacitance is smaller than their P-N counterparts. For that reason, they can be found in switching power supplies and both

FIGURE 1 Behavior of P-N Junction. (a) in equilibrium condition; (b) when a forward voltage is applied

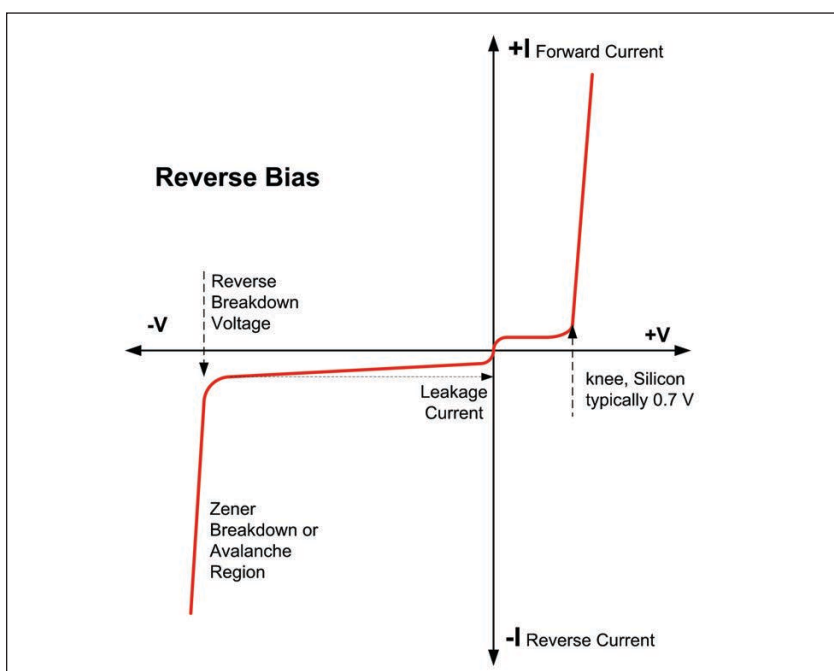
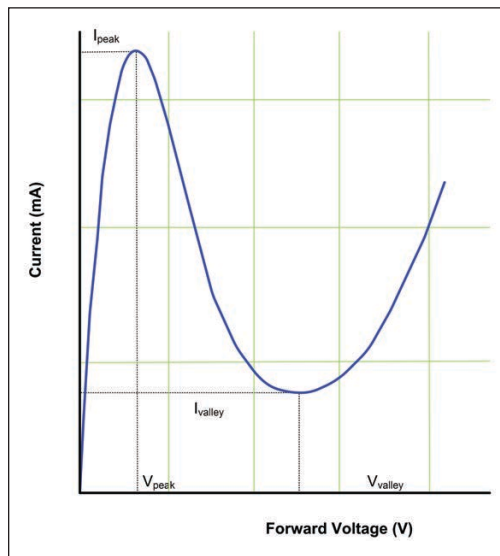


FIGURE 2 I-V Characteristic of a diode (not to scale)

FIGURE 3

Typical tunnel diode I-V graph



high-speed and radio frequency (RF) circuits. On the negative side, their reverse voltage is limited to about 50 V, and the leakage is higher than that of silicon diodes.

MORE TYPES OF DIODES

Let's look at other diode types. I already mentioned Zener diodes. Avalanche diodes are similar—often mistakenly considered Zener diodes—but their reverse voltage breakdown mechanism is a bit different. However, their temperature coefficient is of opposite polarity to Zener diodes, so that they can be successfully used in combination with Zeners for temperature compensation of voltage stabilizers.

Similar in their principle are TransZorbs made by Vishay Semiconductor—also called TVS (Transient Voltage Suppressor) diodes. They are used for overvoltage circuit

protection. TVS diodes come as unipolar— analogous to a single Zener—and bipolar, which could be modeled by two series-opposite Zeners. They are significantly faster than Zeners, and can absorb a lot of pulse energy (see my article “Overstress Protection” in *Circuit Cellar* 330, January, 2018.) Diacs— commonly used for voltage triggering—and their much faster cousins, Perun diodes, come under the same category and can be modeled by two series-opposite Zener diodes.

The “constant-current” diode is a misnomer. It is not a diode at all, but rather a JFET (junction field effect transistor), with its gate and source electrodes shorted. In conjunction with a Zener diode, it provides a very stable voltage source. Tunnel diodes, also called Esaki diodes, exhibit a negative resistance region in their I-V characteristic, which is due to quantum tunneling. These diodes are very fast and work as oscillators, small signal amplifiers or simple bi-stable circuits at high frequencies. Gunn diodes are similar and have similar applications, but are faster, allowing construction of microwave oscillators. **Figure 3** depicts the typical tunnel diode I-V characteristic.

Light-emitting diodes (LEDs) comprise a direct band-gap semiconductor, typically gallium arsenide (GaAs). Charge carriers, while crossing the junction where they recombine with the majority carriers on the other side of the junction, emit photons. Depending on the material used, different colors of light can be generated, from infrared to the near ultraviolet. Various colors, including white, can be obtained by mixing several base colors and by scintillator coating.

LEDs are sensitive to light and can be used as cheap photo detectors. Coupled with photo detectors—for example, photo diodes, photo transistors, photo SCRs and so on—they form the basis for opto-isolators. Today, efficient, powerful LEDs are replacing traditional incandescent lights. By placing a specially formulated LED in a resonant cavity, we obtain a laser. Lasers are used in high-speed communications via fiber optics, and for optical storage devices such as CDs, DVDs and Blue-rays.

All P-N diodes have their forward voltage linearly dependent on the temperature:

$$T_J = (m \times V_F) + T_0$$

where T_J is the junction temperature in $^{\circ}\text{C}$, m is a device-specific slope in $^{\circ}\text{C}/\text{V}$, V_F is the forward voltage drop and T_0 is a device-specific intercept temperature in $^{\circ}\text{C}$. As is apparent from the equation, the temperature dependence is device-specific, but it is typically $-2.3 \text{ mV}/^{\circ}\text{C}$. Many



ABOUT THE AUTHOR

George Novacek was a retired president of an aerospace company. He was a professional engineer with degrees in Automation and Cybernetics. George's dissertation project was a design of a portable ECG (electrocardiograph) with wireless interface. George has contributed articles to *Circuit Cellar* since 1999, penning more than 120 articles over the years.

For detailed article references and additional resources go to: www.circuitcellar.com/article-materials
Reference [1] as marked in the article can be found there.

RESOURCES

Vishay Semiconductor | www.vishay.com

ICs contain an integral thermal diode for monitoring their internal temperature or for external temperature measurement. Thermal diodes are also the building block of Peltier thermoelectric modules (see my two-part article “Thermoelectric Cooling” in *Circuit Cellar* 336 and 337, July and August 2018.)


Semiconductors are also sensitive to light, which is not always a desired characteristic. That’s why semiconductors are packaged in light-blocking materials, unless they are photodiodes used for the detection of light. There are many uses for photo-sensitive semiconductors, from the aforementioned opto-isolators and fiber-optic communications to light sensors, solar cells and others.

PIN diodes use a so-called “high level injection,” forming an intrinsic I region between the P and N materials that is saturated with charge carriers. The intrinsic layer is an undoped semiconductor. PIN diodes find their use in RF circuits, as ionizing radiation detectors and photo detectors. Because the central I layer can withstand high voltages, the PIN structure is used in IGBTs, power MOSFETs and thyristors.

Super barrier diodes are rectifiers based on the Schottky diode structure. Compared with the regular P-N rectifiers, they feature lower forward voltage drop, high surge-

handling capability and low reverse leakage current. Diodes doped with gold (Au) or platinum (Pt) boast improved speed of recombination of minority carriers. They can operate at RF frequencies and are faster than regular P-N diodes (though not as fast as Schottky), but their forward voltage drop is higher than regular P-N diodes. Their reverse voltage leakage is lower than that of Schottky diodes. Their typical representative is the ubiquitous 1N914.

A step recovery diode (SRD) can generate very short pulses, and finds its use in microwave electronics. In the same category, we find varistors—also called varicaps or varactors—which take advantage of their voltage-dependent capacitance between the P and N electrodes. These are used as voltage-controlled capacitors in all kinds of RF circuits. In fact, the classic tuning capacitor is almost impossible to find any more. Finally, we should mention stabistors, which are silicon diodes with a very stable forward voltage drop. These are used as low-voltage reference sources with extremely low dependency on the operating current and temperature.

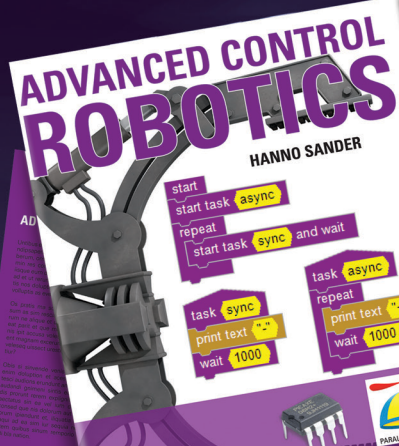
In this article, we’ve covered all-things-diode in a nutshell. Next month, in Part 3 of this article series, we’ll take a closer look at devices with more than two P-N junctions. 

When it comes to robotics, the future is now!

Advanced Control Robotics simplifies the theory and best practices of advanced robot technologies, making it ideal reading for beginners and experts alike.

With this book, you'll learn about:

- Communication
- Technologies
- Control Robotics
- Embedded Technology
- Programming Language
- Visual Debugging... and more



Get it today, cc-webshop.com



Sales@ezpcb.com

Ez PCB

Welcome

www.ezpcb.com

One-Stop PCB & PCBA Turnkey Service



PCB Online Calculator

No Need to Register
Instant Quote & Pay



1 To 40 Layers

Prototype to Mass Production
Amateur to Professional



Prototype Start At \$5/PC

2L 4" x 4" each
Free Shipping

In the past years, our PCB have been shipped to 40 countries

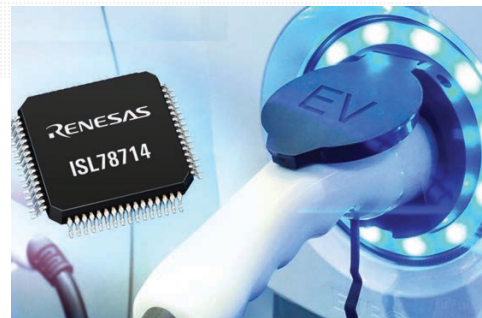
PRODUCT NEWS

IC Does Management for 14 Series-Connected Li-ion Battery Cells

Renesas Electronics has announced its fourth-generation lithium-ion (Li-ion) battery management IC that offers lifetime accuracy. The ISL78714 provides accurate cell voltage and temperature monitoring, along with cell balancing and extensive system diagnostics to protect 14-cell Li-ion battery packs while maximizing driving time and range for hybrid and electric vehicles (HEV/MHEV/PHEV/BEV).

The ISL78714 monitors and balances up to 14 series connected cells with $\pm 2\text{mV}$ accuracy across automotive temperature ranges, letting system designers make informed decisions based on absolute voltage levels. The ISL78714 includes a precision 14-bit analog-to-digital converter and associated data acquisition circuitry. The device also offers up to six external temperature inputs (two available from GPIOs) and includes fault detection and diagnostics for all key internal functions.

The ISL78714 meets the stringent reliability and performance requirements of battery pack systems for all EV variants including HEV/PHEV, with safety features



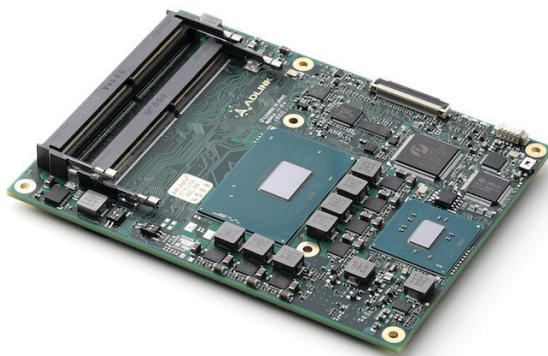
enabling automotive manufacturers to achieve the ISO 26262 automotive safety integrity level (ASIL D). Mass production quantities of the ISL78714 Li-ion battery management IC are available now in a 64-lead TQFP package.

Renesas Electronics
www.renesas.com

Boards Provide COMe Type 2 Drop-In Replacement Solutions

ADLINK Technology has announced two new COM Express Type 2 computer-on-module, based on 6th and 7th Gen Intel Core Processors (formerly "Skylake" and "Kaby Lake"). The new Express-SL2 and Express-KL2 (shown) modules support all Type 2 related legacy I/Os and thereby allow embedded systems developers to extend the production life of existing Type 2 based systems for at least another 10 years.

Type 2 remains indispensable for many system integrators and OEMs that have ongoing production systems with carriers based on the COM Express Type 2 pinout. With the recent discontinuation of the hugely popular 3rd Gen Intel Core Processor (formerly "Ivy Bridge"), many system developers are in need of a new Type 2 module



to keep existing systems up and running. They are in need of a drop-in solution at both hardware and software levels (Intel-to-Intel) with equivalent or improved performance and a better thermal envelope to simplify their transition. The new Express-SL2 and Express-KL2 Type 2 modules solve this problem.

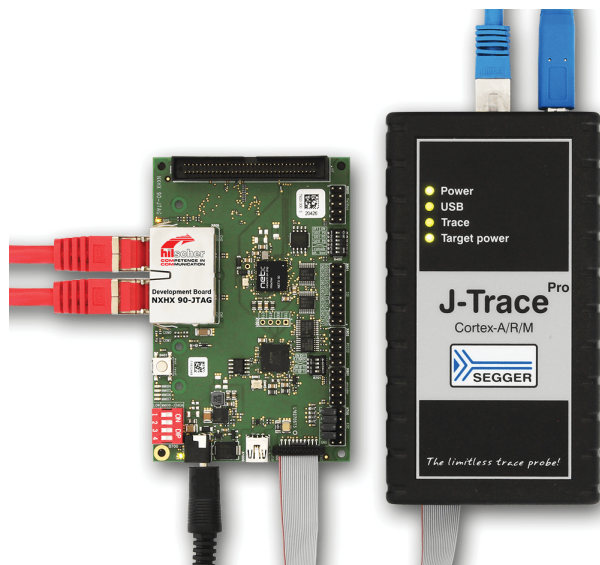
The new Express-SL2 and Express-KL2 feature the 6th and 7th Gen Intel Core Processors (as well as Celeron and Xeon options) and support the Type 2 pinout with legacy interfaces such as PCI-bus, PATA, and VGA. The Express-SL2/KL2 is available in both commercial (0°C to 60°C) and Extreme Rugged (-40°C to +85°C) versions.

ADLINK Technology
www.adlinktech.com

SEGGER Trace Probes Add Support for Hilscher NetX90 SoCs

SEGGER Microcontroller has added support for the latest Hilscher multiprotocol SoC, netX90, to its J-Trace and J-Link trace probes. Software engineers working with Hilscher's netX90 can now enjoy the high download speeds of the J-Link and advanced features, such as breakpoints in flash memory, but also the additional features brought to the table by J-Trace-PRO, such as live code profiling and coverage.

The SEGGER J-Link / J-Trace family of debug and trace probes is tool chain independent and supports GDB, LLVM/Clang as well as most commercial tool chains. J-Trace PRO works with all currently available Cortex-M devices with ETM trace output. According to SEGGER, the increasing complexity of SoCs makes system verification even more important. The ability to log the program flow over long periods of time can catch even the most elusive bugs.



SEGGER Microcontroller | www.segger.com

IDEA BOX

The Directory of PRODUCTS & SERVICES

AD FORMAT:

Advertisers must furnish digital files that meet our specifications (circuitcellar.com/mediakit).

All text and other elements MUST fit within a 2" x 3" format.
E-mail adcop@circuitcellar.com with your file.

For current rates, deadlines, and more information contact Hugh Heinsohn at 757-525-3677 or Hugh@circuitcellar.com.

Revenue Control Systems **PnP**

- Pick & Place Machines starting @ \$6,250
- Direct U.S. Sales, Support, Training, Parts, Accessories, Warranty
- PCB Fabrication Equipment
- Makerspace Specials
- Reflow Ovens

757-258-0910

RCSPnP.com






STAY SECURE WITH **RFID**

Develop RFID with a PIC16F876

- RFID Libraries in the CCS C Compiler
- Works with Read-Only and Read/Write Transponders!
- Pick up a Development Kit today!



ONLY
\$139

Dev Kit
& Single-Chip
Compiler

www.ccsinfo.com/CC1019

sales@ccsinfo.com - 262-522-6500 x 35

PIC[®] MCU is a registered trademark of Microchip Technology Inc.

ALL ELECTRONICS

Surplus & New Parts & Supplies
Since 1967


- LEDS · CONNECTORS · RELAYS
- SOLENOIDS · FANS · ENCLOSURES
- MOTORS · WHEELS · MAGNETS
- PC BOARDS · POWER SUPPLIES
- SWITCHES · LIGHTS · BATTERIES
- and many more items...



We have what you need for your next project.


Discount Prices
Fast Shipping

www.allelectronics.com



Technologic Systems

Single Board Computer



TS-7250-V2

1GHz ARM Computer with
Customizable FPGA-Driven
PC/104 Connector
and Several Interfaces
at Industrial Temp

www.embeddedARM.com

TEST YOUR EQ

Contributed by David Tweed

Answer 1— CPU utilization can be defined as the average amount of time the task requires each time it runs divided by the average period between task starts.

For example, a timer interrupt might run every 1 ms, and require about 50 μ s of CPU time each time it runs. Therefore, its CPU utilization is $50 \mu\text{s} / 1000 \mu\text{s} = 0.05$, or 5%.

Answer 2— Even when just starting the design of a real-time system, it is a good idea to define all of the tasks that will be running on the CPU, and then estimate the CPU utilization of each of them. If the total utilization of all of the tasks begins to approach 100%, it's possible that the system will fail to meet its deadlines under some conditions. You'll need to think about ways to redistribute the work—use a faster CPU, add more CPUs, employ dedicated hardware for some tasks and so on.

Answer 3— You can instrument a task to turn on a GPIO pin when it runs, and to turn it off again when it finishes.

You can then examine the activity on this pin using an oscilloscope or logic analyzer to determine whether that task is always meeting its deadlines. Depending on how many pins you have available, you can get a very detailed real-time picture of the system activity.

You can use this technique on the "idle" task (assuming your system has one) in order to get an idea of the overall CPU utilization of the system.

Answer 4— A system can monitor its own ability to meet deadlines if there's a free-running counter available (perhaps in hardware) to indicate the passage of time. Take a snapshot of the counter value at the beginning of each task. Take another snapshot at the end of the task and compute the difference. If this ever exceeds the required value (deadline) for this task, indicate an error.

For more information:
circuitcellar.com/category/test-your-eq/

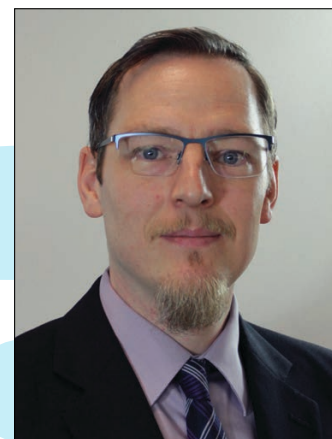
Circuit Cellar 2018 Archive

Order yours today
cc-webshop.com

2018
Issues # 330-341 — CD #23
Copyright 2018, KCK Media Corp.

The Future of Electronic Design Automation

Tool Advances Speed Design-to-Manufacture Process



By **Keith Lee**,
Software Engineer, Gumstix

In my time working in the electronics division of a research organization, I would often have colleagues from other divisions come to me asking for assistance with designing specialized hardware for their projects. Typically, they were looking for high performance embedded Linux computing with a range of sensor options. I love working on these projects. All that said, once you start working with a large team and a bureaucratic process, labor costs can quickly skyrocket well beyond the time involved to design the hardware. Sadly, for my research colleagues, this meant they were often unable to get the specialized hardware they required because their project budgets didn't allow for an additional \$20,000 to \$40,000 in hardware.

The cost to outsource this project to an electronics design contractor would have been similar, or even higher. As a result, the researchers would often end up trying to make something work with a Raspberry Pi and some breakout boards. The Raspberry Pi is a great series of boards, but, because the solutions were just wired together, they were not mechanically robust. And there are considerably fewer options for specialized sensors or op amps in the breakout board market than if we had designed them on a circuit board with a full selection of ICs available.

In 2016, I discovered Geppetto [1] during the Gumstix hiring process. Geppetto is an online circuit board design tool that's a bit different to traditional electronic design automation (EDA) packages (Figure 1). It really takes the "automation" part of EDA to the next level, removing the requirement for user knowledge of routing and circuit theory.

EMPHASIS ON AUTOMATION

For most engineers, Gumstix has been well known for its computer-on-module (COM) products. Over the past couple years, their online design program has quickly gained mindshare as well.

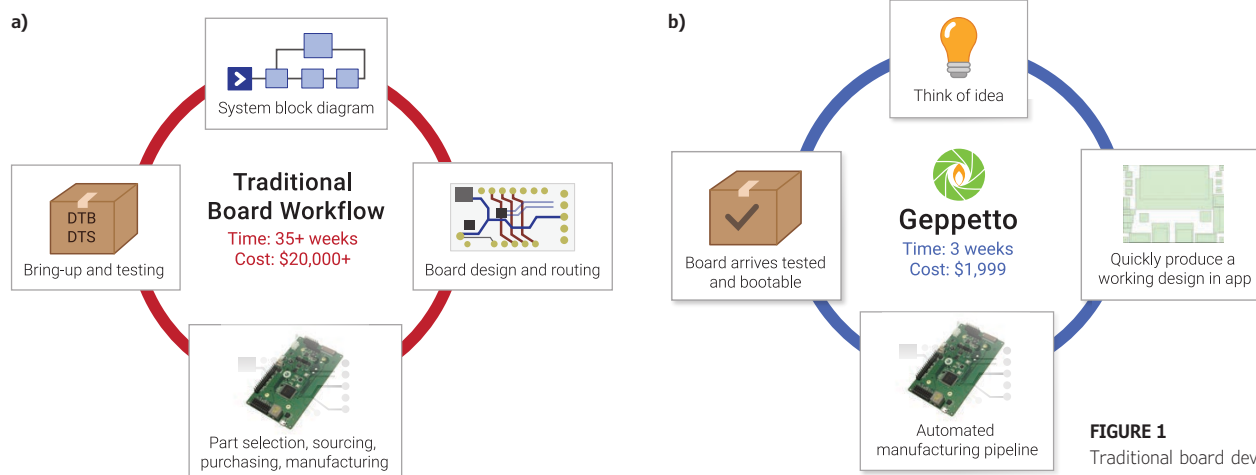


FIGURE 1 Traditional board development workflow (a) versus and Geppetto work flow (b)

TECH THE FUTURE

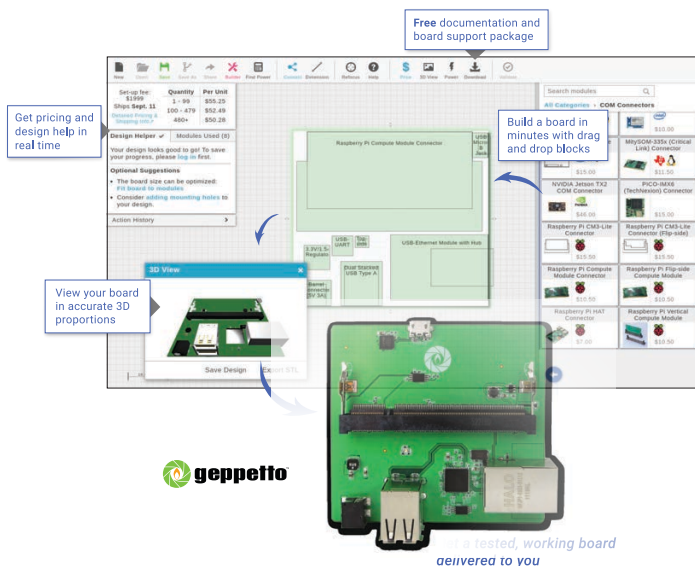


FIGURE 2

Geppetto enables you to layout your concept and receive fully working and tested hardware at the end, without ever having to deal with sourcing, manufacturing, or testing issues.

And now it enables a much wider range of components to be used. I predict we're going to be seeing a lot more tools like Geppetto in the coming years—both in the form of web browser-based EDA tools such as Altium Upverter [2] that are similar to their desktop counterparts, and more simplistic design tools that allow complex devices to be built with very little labor.

My experience tells me that Geppetto is representative of where both desktop and online electronic design software is heading. Traditional desktop ECAD software packages at all levels, from hobbyist to professional, have been making large strides towards utilizing data from services such as Octopart to assist users in designing circuit boards. This is allowing users to build higher quality component libraries, and produce lower-cost bills of materials with parts that are in mass production and readily stocked.

Some software tools provide alternatives to exporting a set of Gerber files for manufacturing to instead simply ordering a board from a list of suppliers directly in the software. By exporting data to a chosen board house directly, no design intent is lost in translation, giving even experienced engineers a higher quality, more reliable outcome. I have not seen a great deal of progress put into linking an electronic assembly

company or contract manufacturer into the electronic design software, but I feel that is the next logical step. Geppetto has taken all of these future software options into consideration from the start, offering an outstanding experience in which you layout your concept and receive fully working and tested hardware at the end, without ever having to deal with sourcing, manufacturing, or testing issues (**Figure 2**).

KEEPING PACE WITH COMPLEXITY

Electronic design tools are getting smarter every year, and are reducing the workload for engineers. However, in many applications the requirements of circuit board design are becoming ever more complex at a rate that's similar or faster than the evolution of design tools. Even as tools make routing easier, advances in embedded computing are bringing more processing power to bear in smaller, cheaper packages. Tools are racing to keep pace as we're required to route more complex boards with tighter constraints on net length and impedance matching. These increasing complexities of both circuit board and design tools—despite improved usability—increase the costs of designing specialized performance hardware.

If you are producing tens of thousands (or more) of a single device, it makes a lot of sense to be using traditional EDA software to ensure that the PCB layout uses every square inch efficiently—and that each part on the board is the most cost-effective option possible. In contrast, for applications where you might only be prototyping a concept, or producing a specialized piece of hardware for a single low volume application, the future will lean towards rapid hardware development tools that take a lot of the time-consuming complexities out of the design process.

If my research colleagues had had access to a tool like Geppetto, they could have designed and ordered their own high-performance Linux boards to match their project requirements on their own with drastically lower production costs and bring-up times. Even if they lacked the technical knowledge to identify the sensors and modules their design required, I could have made use of the EDA tool to define the board far quicker than if I was using a traditional ECAD product.

The ability to simply order a fully assembled and working product directly from the design tool is a huge advantage for researchers lacking technical expertise in electronics. Also professionals working on projects with strict timelines can take great comfort in the trend toward supplier/manufacturer integration into the design process. **E**

For detailed article references and additional resources go to: www.circuitcellar.com/article-materials
References [1] and [2] as marked in the article can be found there

RESOURCE

Altium | www.altium.com
Gumstix | www.gumstix.com

Keith Lee is a software engineer and Gadget Guru at Gumstix. He has a BSc and MSc in computer engineering and has a keen interest in embedded systems, SoCs, computer architecture, and logic design. Keith enjoys designing and creating tutorials for the latest IoT devices. You can follow his blog at gumstixgadgets.blogspot.ca.

STRONG FOUNDATION

Whether you are an
EMS, CM or OEM,
let our bare boards be the foundation
you build your reputation upon!

Technology:

Up to 50 Layers
Any Layer HDI
Sequential Lamination
Blind / Buried Vias
Laser Drilling / Routing
Heavy Copper

Materials:

Fr4
Metal Core
Isola
Rogers
Polyimide - Flex
Magtron

**We will make only what is needed,
when it's needed,
and in the amount needed.**

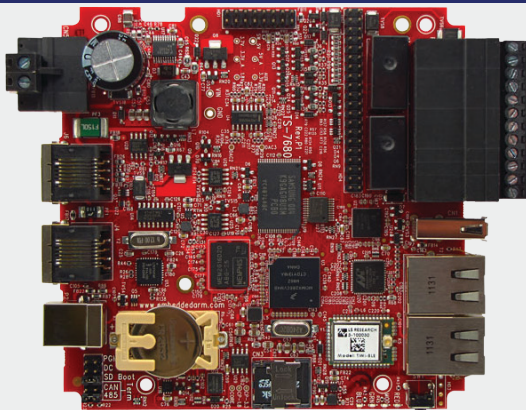
You no longer have to worry about long shelf life
or tie your capital in bare board inventory.

Accutrace[®] inc.

www.PCB4u.com sales@PCB4u.com

SAM & ITAR Registered UL E333047 ISO 9001 - 2008

FROM THE DEEP BLUE SEA TO THE WILD BLUE YONDER



TS-7680

Low Power Industrial
Single Board Computer with
WiFi and Bluetooth

\$159
Qty 100

The TS-7680 is designed to provide extreme performance for applications demanding high reliability, fast boot-up/startup, and connectivity at low cost and low power. Because there are so many features packed on to one single board computer you will see a reduction in payload weight since there is no need for additional boards, micro-controllers, or peripherals.

Rated for industrial temperature range of -40°C to $+85^{\circ}\text{C}$ the TS-7680 is deployed in fleet management, pipeline monitoring, and industrial controls and is working in some of the most demanding places on Earth.

The TS-7680 will help you perform at your very best in a variety of critical missions.



Made in USA
with Global Parts

 **Technologic**
Systems