



circuitcellar.com

circuit cellar

Inspiring the Evolution of Embedded Design

CAR INFOTAINMENT SYSTEMS TAP MCU SOLUTIONS



► Product Focus: ADCs and DACs ► Non-Standard Single Board Computers |
Inductive Sensing with Cypress' PSoC | Build a Self-Correcting LED Clock |
Building a Generator Control System | Thermal Issues in Machine Learning
► Fun with Time Protocols | Infrared Sensors | Bluetooth Mesh (Part 1) |
The Art of Voltage Probing ► The Future of Artificial Intelligence



Trust Your Application

Developing embedded systems is a challenge. Higher complexity, better performance and more connectivity brings a rapid increase of safety and security demands. This creates needs for secure workflows and full code control, extending the developers' reach all the way from design through to production.

We are dedicated to provide you with the superior technology and close-at-hand support you need to be confident in your code when building the products of today and the innovations of tomorrow. Using the right tools, you can trust your application and create for the future.



www.iar.com/ew



**IAR Embedded
Workbench**

 **IAR**
SYSTEMS

Mentor®

A Siemens Business

Speed up your
PCB design verification
by up to **90%**

IIoT Evolution: An Approach To Reuse And Scale Your IIoT Technology Investment

Learn More in this FREE White Paper

This white paper introduces an IIoT framework that complements and extends the massive investments made by cloud vendors to provide comprehensive IIoT features that can be implemented down to the hardware of the edge or end node devices – and can be ported across platforms and clouds. The benefits of using a framework such as MEIF are abundantly clear: minimize learning curves, simplify implementations, increase code reuse, reduce porting costs, and reduce testing.

www.circuitcellar.com/mentor3



OUR NETWORK



SUPPORTING COMPANIES

| | |
|----------------------------|--------|
| Accutrace, Inc. | C3 |
| All Electronics Corp. | 13, 77 |
| Avnet | 53 |
| CCS, Inc. | 77 |
| HuMANDATA, Ltd. | 27 |
| IAR Systems, Inc. | C2 |
| Mentor, A Siemens Business | 1 |
| Slingshot Assembly | 33 |
| TDK-Lambda Americas, Inc. | 29 |
| Technologic Systems, Inc. | C4, 77 |

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

CONTROLLER
Chuck Fellows

FOUNDER
Steve Ciarcia

EDITOR-IN-CHIEF
Jeff Child

TECHNICAL EDITOR
Carol Bower

GRAPHICS
Grace Chen
Heather Rennae

ADVERTISING COORDINATOR
Nathaniel Black

ADVERTISING SALES REP.
Hugh Heinsohn

PROJECT EDITORS
Chris Coulston
Ken Davidson
David Tweed

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 343 February 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

C INPUT Voltage

You've Come a Long Way, Robot

I'm sure many of you are old enough to remember when COMDEX was the huge annual show around which the "technology" industry was focused. It was at its high point in the years I began my technology journalist career back in the early '90s. It was an era when the announcement of Microsoft Windows 3.0 was a huge deal—even though it was still crash-prone—and the latest speed of dial-up modem was newsworthy.

COMDEX Las Vegas in '91 was in fact one of the first major shows I was sent to cover and I remember a Senior Editor coworker advising me to schedule my press appointments with companies at least an hour apart because getting from one side of the exhibitor hall to the other—or between buildings even—would eat up time. As I look back, I'm baffled to think how we ever got by at those kinds of megatradeshows without smartphones, Wi-Fi and Uber.

Fast forward to today and the equivalent of yesteryear's COMDEX is, of course, the Consumer Electronics Show (CES). Over the decades, CES has grown and grown as the PC shifted away from center stage to be eclipsed by smartphones, drones and a myriad of other consumer electronic devices. As I'm writing this, it's actually the week of this year's CES show in Las Vegas. With over 4,500 exhibitors and over 180,000 attendees, it's not quite up to the 200,000+ numbers of a '90's COMDEX, but close.

The CES week was kicked off by a pre-show keynote by LG Electronics (LG) President and CTO Dr. I.P. Park. He outlined the company's vision for the future in his speech entitled "AI for an Even Better Life." Artificial Intelligence (AI) seemed to infiltrate many corners of this year's CES, and Park's talk reflected that. The focus of his speech was on advocating the beneficial role of AI in consumers' lives. The keynote featured LG's AI technologies implemented in daily-life scenarios. Interestingly, LG's CLOi GuideBot robot shared the spotlight with Dr. Park during the address, becoming the first ever robot to help deliver a CES keynote.


Dr. Park began his keynote by asking the audience: "Is technology making your life better?" He continued "Over the past 100 years, household appliances such as refrigerators, washing machines and vacuum cleaners have reduced time spent on housework by around 75% but the amount of cognitive labor involved has significantly increased. The answer lies in AI—but only if we can achieve true intelligence."

During his keynote, Dr. Park presented LG's latest innovations which leveraged the power of AI: an advanced AI chip for home appliances, a washing machine with reinforced learning, and "self-healing" machines that can detect and fix malfunctions automatically without

interrupting operation. He also highlighted the importance of evolving intelligence in consumer electronics. For AI devices to go beyond simple voice recognition and automated task execution, they must be able to understand the purpose and intention behind each command. Such "contextual understanding" requires AI to evolve through accumulating interaction with users and their environments.

It's fitting that Dr. Park shared the stage with one of the company's robots because some of LG's major announcements at the show revolved around robotics. At the show, LG debuted an updated version of its wearable CLOi SuitBot alongside a lineup of improved service robots which are nearing commercialization. The wearable LG CLOi SuitBot supports the lower body to reduce stress when lifting and bending. And the enhanced CLOi service robots—PorterBot, ServeBot and CartBot—demonstrated their more advanced capabilities made possible by LG's continuously evolving AI and robotics technology. Originally unveiled last fall, the updated LG CLOi SuitBot supports the user's waist and augments their own muscle power to reduce the risk of injury and fatigue when performing physically demanding tasks such as lifting and lowering heavy loads. The wearable robot operates for four hours on a single charge and recharges in just one hour.

In the spirit of putting AI into action, part of Dr. Park's talk included using AI technology on a larger scale by connecting what have been traditionally individual units into intelligent systems. He presented LG's Robot Service Delivery Platform, for example, that can systematically coordinate what multiple robots see, hear and learn to transform how we manage our work and our environment.

Although AI applications can take many forms, its use in robotics is perhaps more visceral than in other platforms. Now I'm thinking: How many years away are we from an AI-based robot doing its own CES keynote speech solo? 



Jeff Child

COLUMNS

PRODUCT FOCUS

50 ADCs and DACs Resolution and Speed

By Jeff Child

54 Embedded in Thin Slices Bluetooth Mesh (Part 1) Alternatives Compared

By Bob Japenga

58 The Consummate Engineer Infrared Sensors Heat Lights the Way

By George Novacek

62 The Darker Side The Art of Voltage Probing Scope Savvy

By Robert Lacoste

69 From the Bench Tinkering with Time Protocols and Programming

By Jeff Bachiochi

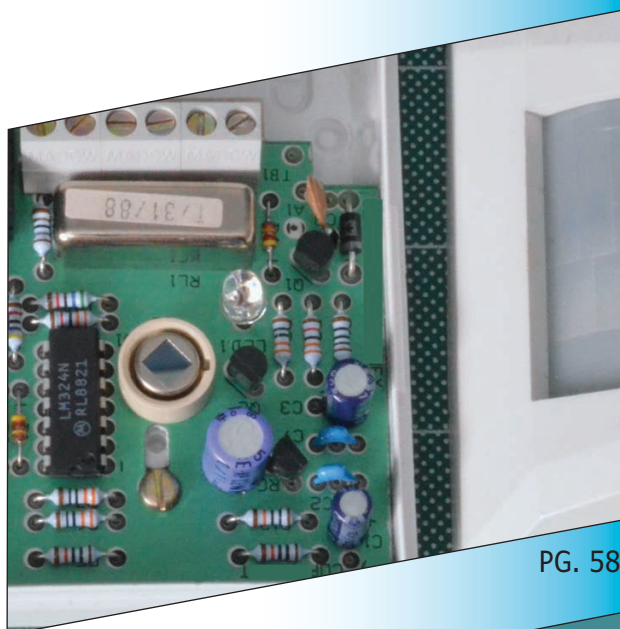
TECH THE FUTURE

79 The Future of Artificial Intelligence Intelligent Edge: Is AI Ready to Play a Role?

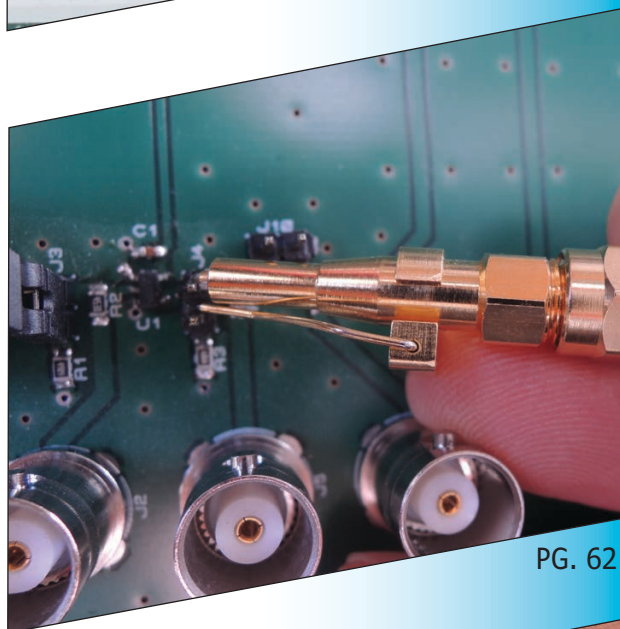
By Scott Nelson

76 : PRODUCT NEWS

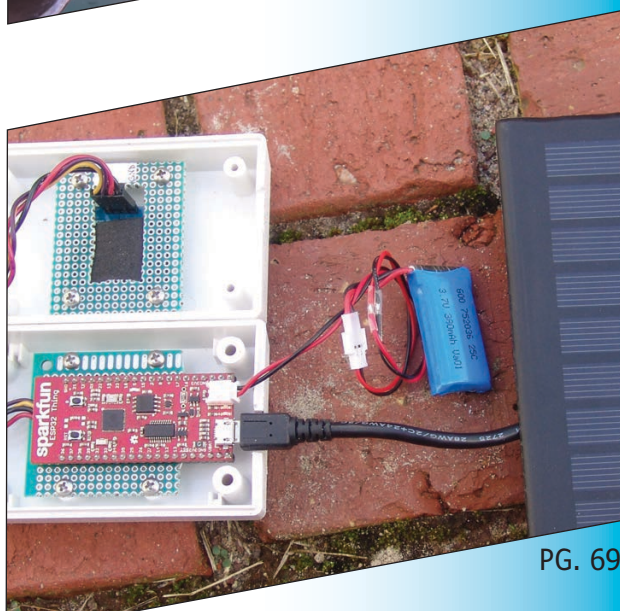
78 : TEST YOUR EQ



PG. 58



PG. 62

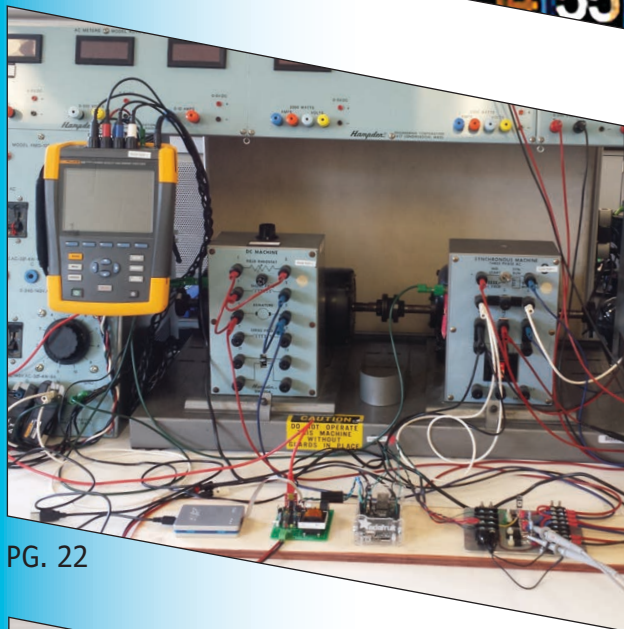


PG. 69

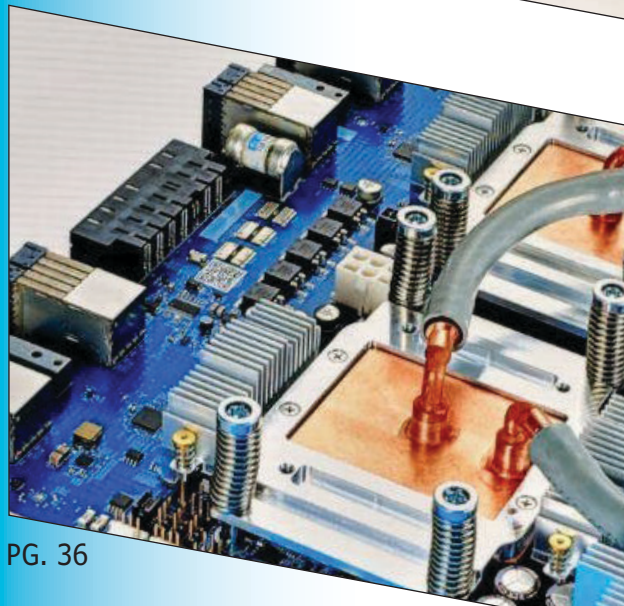
FEATURES



PG. 6



PG. 22



PG. 36

6 Build a Self-Correcting LED Clock

MCU-Driven Art

By Eldar Slobodyan and Jason Ben Nathan

14 Inductive Sensing with PSoC MCUs

Tougher Touch Tech

By Nishant Mittal

22 Building a Generator Control System

Three-Phase Power

By Kent Altobelli and Caleb Stewart

36 Thermal Management in Machine Learning

Beat the Heat

By Tom Gregory

SPECIAL FEATURE

40 MCUs Serve Up Solutions for Car Infotainment

Dashboard Dazzle

By Jeff Child

TECHNOLOGY SPOTLIGHT

46 SWaP Needs Drive Non-Standard SBC Demand

All-In-One Solutions

By Jeff Child

Build a Self-Correcting LED Clock

MCU-Driven Art

In North America, most radio-controlled clocks use WWVB's transmissions to set the correct time. WWVB is a Colorado-based time signal radio station. Learn how these two Cornell students designed and built a prototype of a Digital WWVB Clock. The project's main components include a Microchip PIC32 MCU, an external oscillator and a display.

By
*Eldar Slobodyan and
Jason Ben Nathan*

FIGURE 1

Digital WWVB Clock at 200 Water Street, Manhattan, NY (Photo Credit: Noel Y. Calingasan).

In this article we explain how we designed and built a prototype of a Digital WWVB Clock, based on the design of the building at 200 Water Street in Lower Manhattan (**Figure 1**). Jason has always wanted to pay homage to this clock, as he finds the design inspiring. In this, our final project at Cornell, we were determined to utilize the same design to create a functional piece of technology that adds color, vibrancy and time to any room.

Making a clock requires a time base, a control circuit and a display. We implemented the time base and control on a Microchip PIC32MX250F128B microcontroller (MCU), using a development board built at Cornell University [1]. The clock has an external oscillator keeping time with an accuracy of ± 1 minute per year, and a WWVB receiver for setting the time whenever the signal can be received. Time is displayed on a panel in front of 72 Adafruit DotStar LEDs. The clock takes input from three push buttons that select the mode of operation: Time, Date and Temperature. Additionally, because the project was inspired by a piece of visual art, we put extra effort into making it aesthetically pleasing, using wood instead of cardboard, and by eliminating

the leakage of light between the LEDs by creating an individual housing for each one. **Figure 2** shows the full electrical schematic of the clock and peripheral devices.

THE CLOCK BODY

We built the housing out of a combination of plywood sheets and softwood planks. The front panel was a piece of quarter inch plywood that was laser-cut at Cornell Rapid Prototyping Lab (RPL) [2] to represent the numbers. It was later drilled from the front and the back to install the buttons. The back face was a piece of quarter inch plywood that was drilled in the back for the ventilation of the power supply. The sides of the frame were 0.5" thick pieces of softwood, drilled on the bottom to hold the power supply in place, and on the left for the 120 VAC cord path.

We also secured the AC cord—consisting of line, neutral and ground cables—with zip-ties on either side of the left wall. We did this to ensure that if it were pulled, it wouldn't come out of the power supply—which would cause electrical damage and endanger the user. We hot-glued the LED strips to the back face. To create a housing for each separate LED light and prevent light from leaking into other numbers, six long horizontal ribs were

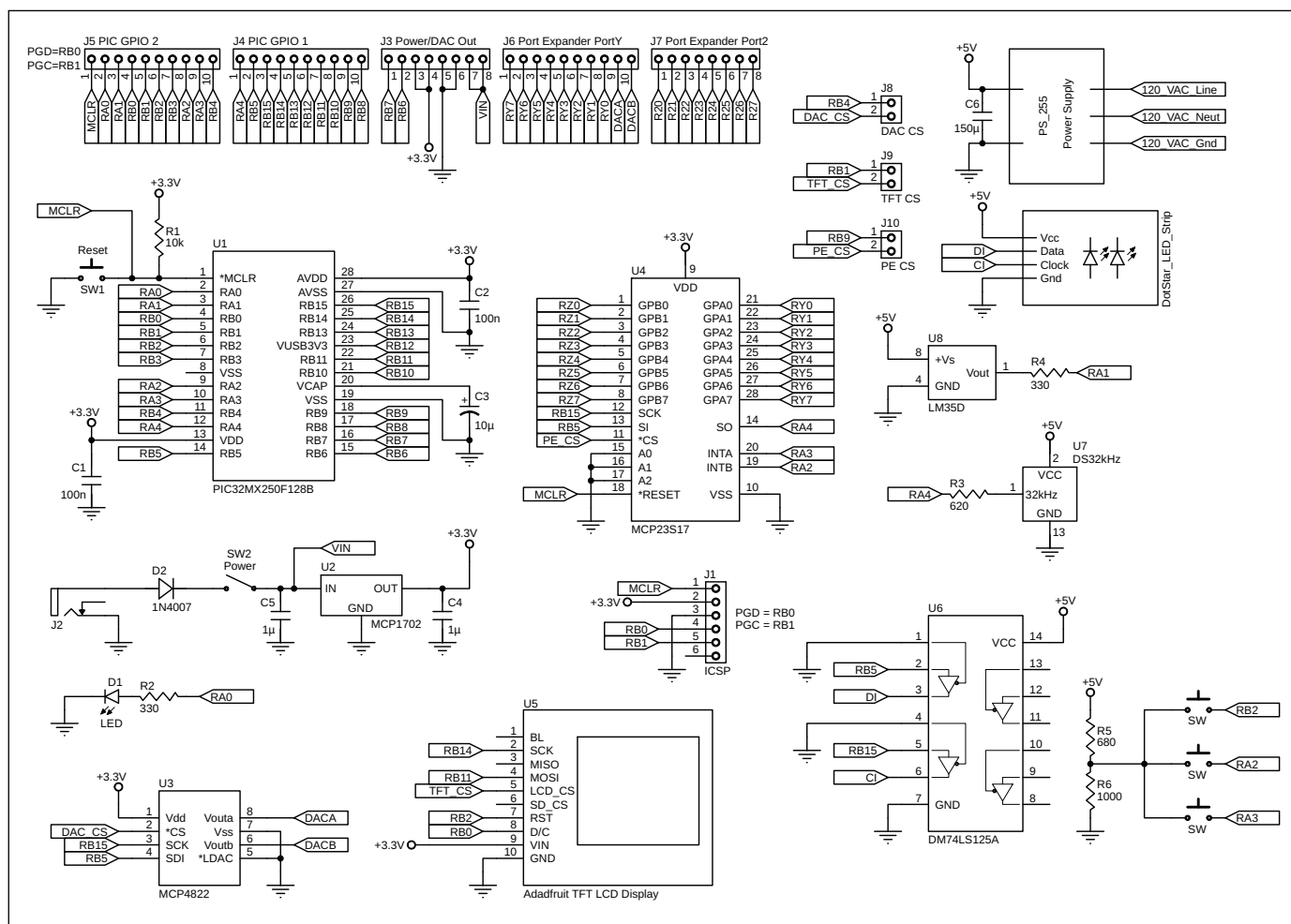


FIGURE 2

Full electrical schematic of the clock and peripheral devices

made out of quarter inch plywood and cut on a band-saw to fit into the box. Sixty-six short vertical ribs were laser-cut from pieces of leftover quarter inch plywood. The ribs were fitted to house the lights and hot-glued in place (**Figure 3**).

THE LEDS

We chose DotStar LED strips for their brilliant and vibrant colors. They had the added benefits of being easy to program and having good documentation, though they were somewhat costly. We chose strips that had 30 LEDs per meter, to space-out each LED in its own box. The DotStars support 8-bit RGB color, which allowed us extreme flexibility for color choice. Their implementation was complicated, as we had to solder them with short wires, based on research of the possible problems associated with using longer wires between cut strips.

We had the clock numbers displayed on the face of the clock ordered from left to right, top to bottom, wrapping around from

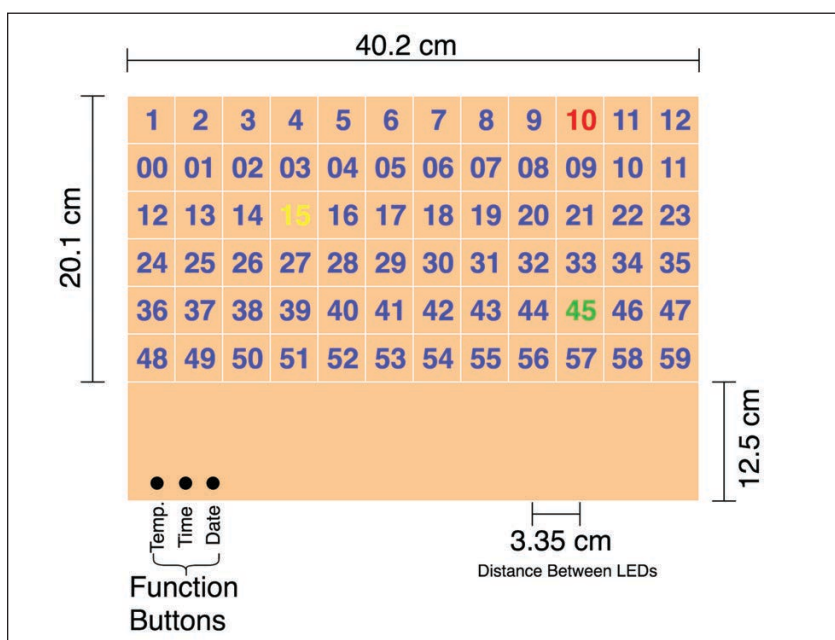


FIGURE 3

Shown here is the design for the front panel of the clock. Dimensions were chosen to fit the LED strips and power supply.

TABLE 1

LED indexing. The clock digits are in red, and LED index numbers are in blue. LED indexing began next to the location of the PIC 32 and “snaked” upward.

| | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 71 | 2 70 | 3 69 | 4 68 | 5 67 | 6 66 | 7 65 | 8 64 | 9 63 | 10 62 | 11 61 | 12 60 |
| 00 48 | 01 49 | 02 50 | 03 51 | 04 52 | 05 53 | 06 54 | 07 55 | 08 56 | 09 57 | 10 58 | 11 59 |
| 12 47 | 13 46 | 14 45 | 15 44 | 16 43 | 17 42 | 18 41 | 19 40 | 20 39 | 21 38 | 22 37 | 23 36 |
| 24 24 | 25 25 | 26 26 | 27 27 | 28 28 | 29 29 | 30 30 | 31 31 | 32 32 | 33 33 | 34 23 | 35 35 |
| 36 23 | 37 22 | 38 21 | 39 20 | 40 19 | 41 18 | 42 17 | 43 16 | 44 15 | 45 14 | 46 13 | 47 12 |
| 48 0 | 49 1 | 50 2 | 51 3 | 52 4 | 53 5 | 54 6 | 55 7 | 56 8 | 57 9 | 58 10 | 59 11 |

the right end of one row to begin at the left end of the next (**Table 1**, red). However, we couldn't have long wires stretching across the face, so we wrapped the LED strip in a “snake” pattern around the box, (**Table 1**, blue). Another good reason to use the DotStars was for their simple and reliable communication protocol via SPI. DotStars require 5 V power and 5 V command signals. Powering them was not an issue, because all our other components also used 5 V. To drive them from the PIC32 MCU, we used a voltage level translator to convert the 3.3 V PIC32 MCU signals to 5 V.

We purchased a 25 W single output power supply made by Mean Well. It takes 120 VAC from a wall outlet and outputs 5 V DC. We chose a 25 W power supply to provide a maximum of 5 A of current to power the DotStar strip, which can pull a maximum of 4.32 A.

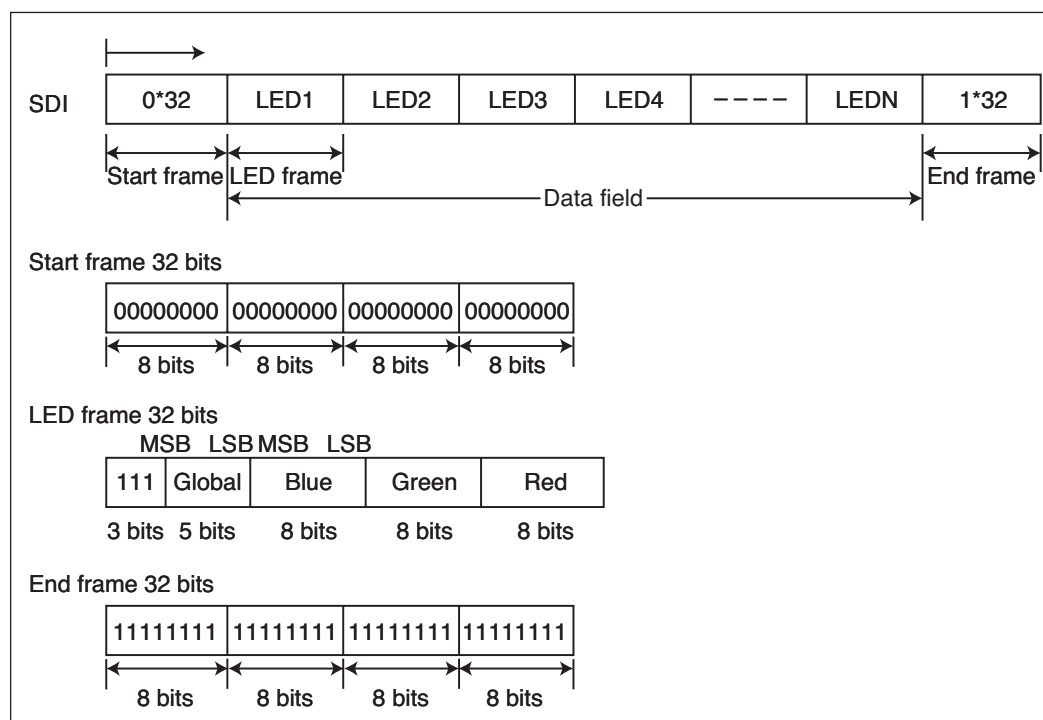
LED LOGIC

The SPI protocol used by DotStars transfers 32-bit LED data frames. The first

frame is 32 bits of 0's. Each subsequent LED has its own data frame, in which the first 3 bits are 1, followed by a 5-bit value for the intensity level, then an 8-bit value for blue, green and red. The last frame sent is 32 bits of “1”s, which indicate the end of transmission. A visual representation of the protocol is shown in **Figure 4**.

We used code from Bruce Land's Pixel Strip Displays example. The code contained a working SPI function to communicate with the DotStar LED strip of specified size and a data structure, to format the color and intensity information of the LEDs. The SPI channel also outputs square wave clock signals for the SK9822 control IC in the DotStars can synchronize with the SPI. Once the LED data array is filled, `write_pixels()` is called and the DotStars are set.

Before `write_pixels()` can be called, the LED data array needs to be filled. We used three functions, `time_to_led()`, `date_to_led()` and `temp_to_led()` to display the time, date, and temperature,

**FIGURE 4**

DotStar SPI Communication Protocol is used to light the LEDs. Good documentation was part of the reason DotStars were chosen.

respectively, on the DotStars. For `time_to_led()`, parameters for the hour, minute and second are sent. Next, we calculated which LEDs are the hour, minute and second and wrote a unique color value to each of them, while giving all the unused LEDs a different color to form a background. The function is `date_to_led()`, which is identical to `time_to_led()`, except it only takes parameters for month and day. The LED for month is set the same way that hour is set, and day is the same way minute is set. We blacked out numbers 00 and 32-59, because those are not possible dates.

The last LED function is `temp_to_led()`, which receives a single parameter for temperature. The LED for temperature in degrees Celsius is then set the same way as minute and day. An added feature of the temperature LED function is that the temperature “flickers” between the integer values to indicate the value that is nearest.

TIME BASE

We used a 32.768 kHz oscillator to implement the Real Time Clock and Calendar RTCC functionality (Digi-Key part number DS32KHZS#-ND). This specific value is convenient because it is exactly 2^{15} —in other words, $2^{15}=32,768$. Therefore, it makes a 16-bit timer overflow exactly once a second. The specifications of the oscillator indicate that its error is only ± 1 minute per year while operating under moderate climate temperatures (0 to 40°C). The oscillator outputs a 5 V square wave, which is connected in series with a 620 Ω resistor to pin 12 on the small board. Despite being a 5 V signal, the current in the oscillator’s output is only 150 μA . We decided not to use a voltage divider or level shifter to guarantee the fastest rise time.

The RTCC peripheral on the PIC32 MCU is used to keep the time and date. It keeps accurate time using an external 32.678 kHz crystal oscillator. Without the external oscillator, RTCC would need to use the internal oscillator, which is not accurate. In the `config.h`, we added `#pragma config FSOSCEN = ON` so the PIC32 knows to use the external oscillator. The RTCC peripheral is initialized in the `main()` function.

OTHER FUNCTIONS

We wanted to add more functionality to time/date keeping. We settled on temperature because it is easy to display room temperature in degrees Celsius using the minute boxes. We used a Texas Instruments LM35 analog temperature sensor to read ambient temperature. The LM35 uses a linear scale factor of +10 mV per °C, and it has ensured accuracy of 0.5°C at normal room temperatures (25°C).

| Previous Value | Current Value | Logic |
|----------------|---------------|---------|
| Off | Off | None |
| Off | On | Press |
| On | Off | Release |
| On | On | Hold |

TABLE 2

Debouncing logic for the buttons. Some of the functionality implemented here can be used for future additions.

User input is received via three buttons connected to the development board [1]. When depressed, the buttons shorted the PIC pins to V_{CC} through a voltage divider 680 Ω /1,680 Ω from the 5 V provided by the power supply, resulting in approximately 3 V. A protothread [4] was used to “debounce” the buttons, to make sure that the input is read correctly. Every 30 ms, the buttons were polled to assess whether they had been pressed. The pins were set as digital inputs. Values from the previous poll were also kept to implement the logic as shown in **Table 2**. Three buttons were implemented in this manner. The debounce functionality ended up being more robust than necessary, because only the button presses were used to switch between the three different modes.

The `main()` function was used to set up the peripherals and protothreads [4] that were used in a voluntary yield-scheduling scheme to run the necessary functions in parallel. We set up the ADC channel (for the temperature sensor), SPI channel (to send the DotStar LEDs lighting data), `Timer2` and `InputCapture1` (to calculate WWVB Pulses described below), the RTCC (to keep correct time using the external oscillator) and system-wide interrupts. The SPI was set up for channel 2 and configured to send 32-bit values. The SPI clock speed is divided by 16 (to operate within the range of DotStars) to

ABOUT THE AUTHORS

Eldar Slobodyan graduated with a B.S. in Electrical and Computer Engineering from Cornell ECE in May 2018. He now works at Curtiss-Wright Sensors & Controls and consults at Cornell Tech. He is passionate about embedded systems, robotics, machine learning, product development and entrepreneurship. He is the CTO of a startup developing smart vehicle technology that will facilitate more efficient urban transportation. His email is es876@cornell.edu

Jason Ben Nathan is originally from North Caldwell, NJ. He received his B.S. in Electrical and Computer Engineering from Cornell in December 2017. Jason is currently working as an Electrical Engineer at The Johns Hopkins Applied Physics Laboratory in Laurel, MD. He is also in the process of creating an upgraded version of this project. His email is jdb378@cornell.edu

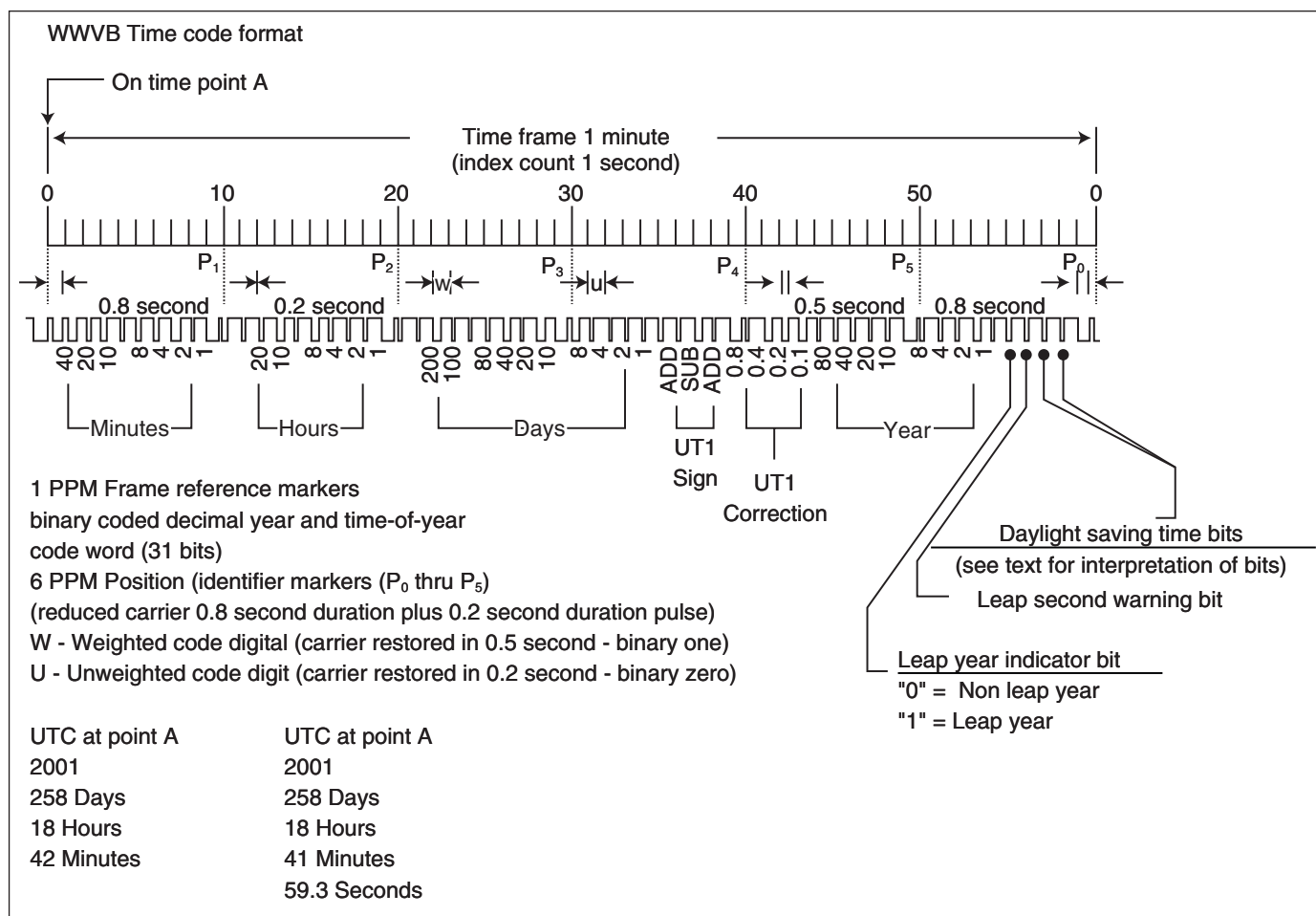


FIGURE 5

WWVB signal representation and the breakdown of the decoding. This radio signal can be used to correct the time on the clock when available.

give small relative rise and fall times after the signal is level-shifted to 5 V. Finally, we initialized two protothreads [4], `pt_timer` and `pt_buttons`, and scheduled them to run in a round-robin format.

WWVB SIGNAL

After much research and testing, it was clear that the WWVB was unreliable because it was a weak signal and too much noise was injected into that bandwidth by our location on a technically advanced campus. The only antenna/receiver available for purchase was chosen. There were no other hardware options to explore at that point. For testing the updating of the clock, the WWVB signal was provided by an Arduino that sent the correct time in WWVB format to the PIC32 MCU.

The WWVB signal is transmitted 1 bps, taking 60 s to send the current time of day and date within a century. Information is encoded using amplitude modulation with the scheme shown in **Figure 5**. If power is reduced for one-fifth of a second (0.2 s), this is a data bit with value zero. If power is reduced for one-half of a second (0.5 s), this is a data bit with value one. If power is reduced for four-fifths of a second (0.8 s),

this is a special non-data "marker," used for framing. Markers are sent during seconds 0, 9, 19, 29, 39, 49 and 59 of each whole packet. The start of the second of two consecutive markers indicates the top of the minute and serves as the on-time marker for the next frame of time code. Markers are important to allow receivers to frame the time code properly.

To receive the WWVB signal, we purchased a 60 kHz WWVB atomic radio-controlled clock receiver from Universal-Solder. It is the only WWVB receiver currently available on the market. The receiver was attached to a 100 mm ferrite-core antenna.

WWVB SOFTWARE

The WWVB transmits data by sending an amplitude-modulated signal. The length of time the pulse is low determines the value of the symbol being transmitted. We used an input capture module to trigger on the first falling edge and then on every edge thereafter. The input capture begins on a falling edge because the length of a WWVB pulse is determined by the time the signal is at -17 dB, which in our case was a digital low value. We set up an interrupt service routine

that records the time for a falling edge (the first signal that causes the input capture to trigger) and stores it into an integer `capture1`. The input capture will then wait until it sees the next edge—which will be a rising edge—and it records the current time and stores it into another integer `capture2`. Then we subtracted `capture1` from `capture2` to calculate the length of the pulse `pulse_time`. Additionally, we increment the counter that keeps track of the symbol number in the current transmission. Several flags are also set here. For the falling edge we set `ready` to 0, indicating that time is not ready to be changed. When it receives the rising edge, `ready` is back to 1. In the timer thread, the new symbol and time can only be set when `ready` is 1.

The input capture module cannot determine whether the edge it is capturing on is high or low. Therefore, we toggled a bit back and forth to indicate which type of edge was next. This was possible because we defined the first trigger to be on a falling edge.

The input capture is useful for calculating the time of small pulses using the internal timers. Because we are calculating times in hundreds of milliseconds, even setting up an internal timer with the largest prescaler value when the processor is running at 40 MHz is not sufficient. We therefore opted to use the millisecond timer from `protothreads` [4], `PT_GET_TIME()`, to calculate the time pulses. We believe that `PT_GET_TIME()` is sufficient because it takes approximately 5 weeks to overflow. If it does overflow while decoding pulses, that pulse will be an error, and the frame will need to restart.

The timer thread begins by checking to see if the input capture has triggered twice, starting with a falling edge followed by a rising edge. If it has, it will then use `pulse_time` to determine the symbol: 100 ms < `pulse_time` < 300 ms is a 0; 400 ms < `pulse_time` < 600 ms is a 1; 700 ms < `pulse_time` < 900 ms is a marker. Otherwise it sends an error. The symbols are then stored into an integer. If a marker symbol is received, we check if the previous symbol was also a marker, which indicates the start of the next frame. If this is the case, we set the bit number `bit_n`, to 0 to keep track of the position in the frame. We also check the `capture_count` to determine if time has been set. If it has not been (`capture_count == 0`), we set `capture_count` to 1 and call `input_time()`. Calling this first `input_time()` sets the time and date to default values (date: 1/1; time 1:02). The yellow seconds indicator begins to run, and it can be used as a countdown to when the entire dataframe is received and the proper

| Bit number | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|------------|----|----|----|----|----|----|----|
| Symbol | 0 | 1 | 0 | 1 | 0 | 0 | 1 |

TABLE 3

A possible partial sequence of bits from the WWVB signal to be decoded. This will read 7 o'clock p.m.



FIGURE 6

Final design of the clock with minimal image processing, showing the color and diffusion of the LEDs. The clock is on "Time" setting reading 1:39:37.

time and date are set. If the time is being recorded but has not been set (`capture_count == 1`), we set `capture_count` to 2 and call `input_time()`. Calling `input_time()` now sets the time and date to value it decoded from the WWVB signal.

After the symbol is determined from the pulse duration, the time is set by calling `set_time()`. The `set_time()` function takes in two parameters--the symbol and its number in the frame. We use 60 individual case statements that are based on the symbol number to decode time. For example, bits 12, 13, 15, 16, 17 and 18 in the frame are used for the hour values, 20, 10, 8, 4, 2 and 1, respectively (bit 14 is left at 0). The example sequence is depicted in **Table 3**.

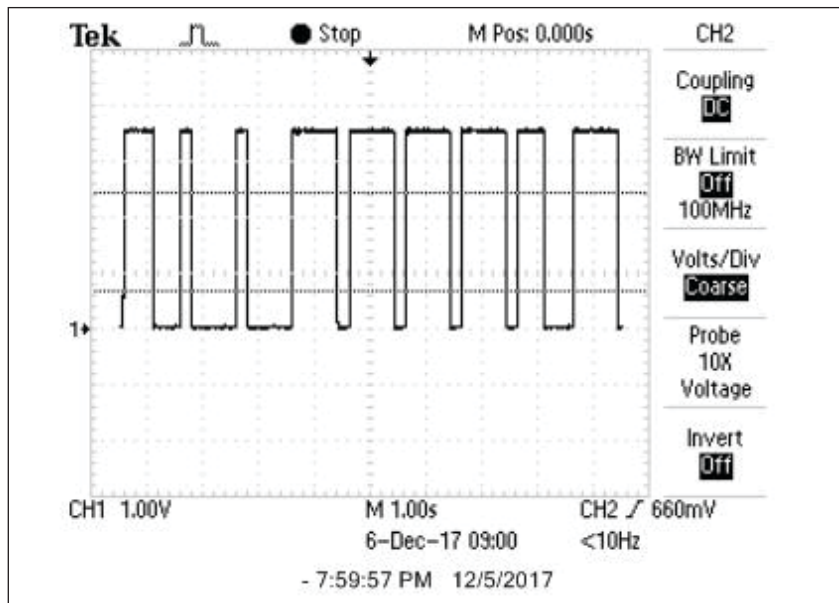
The hour will be $20 \times 0 + 10 \times 1 + 8 \times 1 + 4 \times 0 + 2 \times 0 + 1 \times 1 = 19$ or 7 o'clock PM. We also use the `set_time()` function to check for errors in the received symbols.

The `input_time()` function mentioned above is what assigns the time and date to the RTCC structures (`tm` and `dt`). The function begins by converting the day of the year value to month and day values. Then the RTCC structures for time and date are assigned using the values from the `set_time()` function and a macro that converts integers to BCD format: `#define int2BCD(a) ((a/10)<<4)|(a%10)`.



FIGURE 7

If you would like to see the clock in all its glory, here is a QR link to a YouTube video for your convenience.

**FIGURE 8**

This screen shot shows the Arduino-produced signal to model the WWVB radio signal. A simulated signal had to be used because the WWVB signal was weak and the noise was high on Cornell's EE Campus.

Next, we call `RtccSetTimeDate(tm, 1, dt, 1)` to set the time and date for the RTCC peripheral. Last, we clear all the global time and date variables for the next time `set_time()` is run.

The final part of the thread is a case statement based on the mode of operation (time, date, or temperature). Depending on which mode is chosen, a corresponding LED function is called to set the DotStar LEDs. Last, we call `write_pixels()` to send the LED color and brightness data via SPI. `PT_YIELD_TIME_msec(10)` is called to allow the Button thread time to execute.

RESULTS OF DESIGN

The laser cut numbers on the panel of the display make the numbers sharp and easy to read (**Figure 6**). The DotStars were a great choice. In Figure 6, the blue digits are at quarter power because the background,

and the LEDs carrying the information are at half power. The wood box was strong, but the ribs that separate the individual number boxes and prevent light from leaking were not totally even. This presented no visual problem, but it would be preferable for front and back panels to be completely flat. Also, the numbers could be laser cut more precisely by leaving little connector pieces for digits 0, 4, 6, 8 and 9.


It turned out that the PIC32 MCU is able to keep accurate time. We let the time run side-by-side with other clocks for several-hour periods, and did not see any difference in the time. This was important to our goal of making a fully functional and useful clock, which must remain powered on and accurate for long intervals. You can see our clock by using the QR link (**Figure 7**) to a YouTube video. The video is also posted on *Circuit Cellar's* article materials webpage.

After waiting for more than a month to receive the WWVB receiver, it finally arrived. Unfortunately, we had great difficulty using it, and were unable to pick up any trace of a signal in our building (Phillips Hall at Cornell) (**Figure 8**). Because the deadline for the project was approaching, we decided to use an Arduino board to simulate the WWVB signal. We found code online to do this. The Arduino code sends a perfectly clear pulse-modulated signal that the PIC32 MCU has no trouble decoding.

From a black-box perspective, the signal from the Arduino and the signal from the WWVB receiver look identical. The Arduino is a powerful tool that we used to test all types of time/ date inputs and to fix any errors we found. Unfortunately, we were not able to build a clock that sets the time with absolutely no user input. We plan to continue working on the clock and find a solution for receiving a WWVB signal in perhaps a less tech-centric location.

FUTURE ADDITIONS

Like any good engineering product, we designed the clock so it can be easily upgraded and improved. We compiled a list of additional functionalities that we would like to implement in the future include:

- Self-calibration for the oscillator— further improves accuracy
- Add better error correction—for instance, if a marker is the symbol that is supposed to be present at a specific bit, an error will be thrown and the entire data frame will be discarded.
- More details using the LEDs—such as AM/ PM lighting effects and decimal place for the temperature sensor
- Games—Tetris or Connect 4. 

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

Adafruit, www.adafruit.com

All Electronics | www.allelectronics.com

Digi-Key | www.digikey.com

Microchip Technology | www.microchip.com

Mean Well USA | www.meanwellusa.com

Texas Instruments | www.ti.com

ALL ELECTRONICS CORP.

New & Surplus Electronics since 1967

1000's of parts available online www.allelectronics.com

12 VDC 3 AMP POWER SUPPLY

6' output cable with 2.1mm
coax plug, center+. cULus.

CAT# PS-1272

\$6⁹⁵
each

50 for \$6.45 each
100 for \$6.00 each



300 LEDS, RED, GREEN, YELLOW, BLUE, WHITE

- 5mm LEDs, 20 pcs each color.
- 3mm LEDs, 40 pcs each color.
- Clear plastic divided box w/ locking lid.

CAT# LED-K1

\$8⁰⁰
each

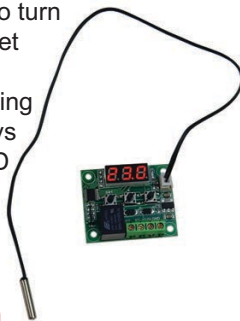


THERMOSTATIC CONTROL MODULE

Digital module can be programmed to turn
equipment on or off at specific, pre-set
temperatures. Also has hysteresis
setting, delay setting, upper-limit setting
and upper limit alarm setting. Displays
temperature in Celsius on 3-digit LED
display. On-board 20A relay.
Water-proof temperature sensor.
Operating voltage 12 Vdc.
48mm x 40mm x 14mm.

CAT# THM-4

\$6⁰⁰
each



ISD1820 VOICE RECORD/PLAY MODULE WITH SPEAKER

Based on the ISD1820 chip, this multiple-
message record/playback device
offers true single-chip voice
recording, non-volatile storage,
and playback capability for
8-20 seconds. Features on-board
microphone as well as internal audio amplifier which can
drive the included 8 Ohm 0.5W speaker directly. Supply
voltage 3.3-5 Vdc. 54mm x 38mm x 15mm high.

CAT# ME-63S

\$7⁰⁰
each



0-100V DC VOLTMETERS

3-digit LED voltmeter for 0-100VDC range.
0.28" digits on a 0.89" x 0.39" display.

Working current: 20mA Operating voltage: 4-40 VDC
Measurement rate: 200mS /Times

Red digits CAT# PM-126
Blue digits CAT# PM-126B

\$3⁹⁵
each



BATTERY LEVEL METER

Monitor battery usage in lead-acid and
lithium-ion batteries. Displays percentage
of voltage remaining in numerical and
graphic format. For use with 12, 24, 36 or 48V lead-acid
batteries or 3 to 15 cell (8-63V) lithium-ion batteries.
Battery symbol flashes when level reaches 5%.

CAT# BT-3

\$7⁷⁵
each



THROUGH-BEAM PHOTO-ELECTRIC SENSOR PAIR

SICK Optics WS15-D/A1130
Photo-electric sender and receiver
pair detects the presence of an object
when the beam between the two
is interrupted. Useful in automation,
alarm systems and door annunciators.
5 Meter range.

CAT# OSU-1130

\$11³⁵
pair



3 1/2 DIGIT LCD MULTIMETER WITH BACKLIGHT

For features, accuracy and rugged-
ness, this is the best inexpensive
multimeter we've ever seen. DC
current (10 A), DC and AC voltage
(600 V), resistance (2 M ohm), diode,
transistor tester, audible continuity and
hold button. Protective rubber shell
and test leads included.

CAT# DVM-850BL

\$17⁹⁵
each



3 AMP DC MOTOR SPEED CONTROLLER

An efficient method of changing the
speed of a DC motor without changing
voltage. Has a potentiometer with on-off
switch and a four-position screw-terminal strip for easy
connection to motor and power source. Can be used as a
lighting dimmer and similar applications with DC loads up
to 3A. 30 x 28mm pc board.

CAT# MSC-35

\$5⁷⁵
each



STEPPER MOTOR AND DRIVER BOARD

Includes our driver board, CAT# ME-16
and a 5-wire stepper motor that
plugs directly into the board.
Ideal for robotics projects.

Motor is a 4-phase, unipolar stepper, #28BYJ-48.

CAT# ME-125

\$5³⁵
set



Toll Free Phone Orders 800-826-5432

Free shipping on orders over \$75 in the continental USA

Inductive Sensing with PSoC MCUs

Tougher Touch Tech

Inductive sensing is shaping up to be the next big thing for touch technology. It's suited for applications involving metal-over-touch situations in automotive, industrial and other similar systems. Here, Nishant explores the science and technology of inductive sensing. He then describes a complete system design, along with firmware, for an inductive sensing solution based on Cypress Semiconductor's PSoC microcontroller.

By
Nishant Mittal

Touch sensing has become an important technology across a wide range of embedded systems. Touch sensing was first implemented using resistive sensing technology. However, resistive sensing had several disadvantages, including low sensitivity, false triggering and shorter operating life that discouraged its use and led to its eventual downfall in the market.

Today whenever people talk about touch sensing, they are usually referring to capacitive sensing. Capacitive sensing has proven to be robust not only in a normal environmental use cases but, because of its

water-resistant capabilities, also underwater. As with any technology, capacitive sensing comes with a new set of disadvantages. These disadvantages tend to be more application-specific. And those have opened the door for the advent of inductive sensing technology.

Inductive sensing is based on the principle of electromagnetic coupling, between a coil and the target. When a metal target comes closer to the coil, its magnetic field is obstructed and it passes through the metal target before coupling to its origin (**Figure 1**). This phenomenon causes some energy to get transferred to the metal target named as eddy current that causes

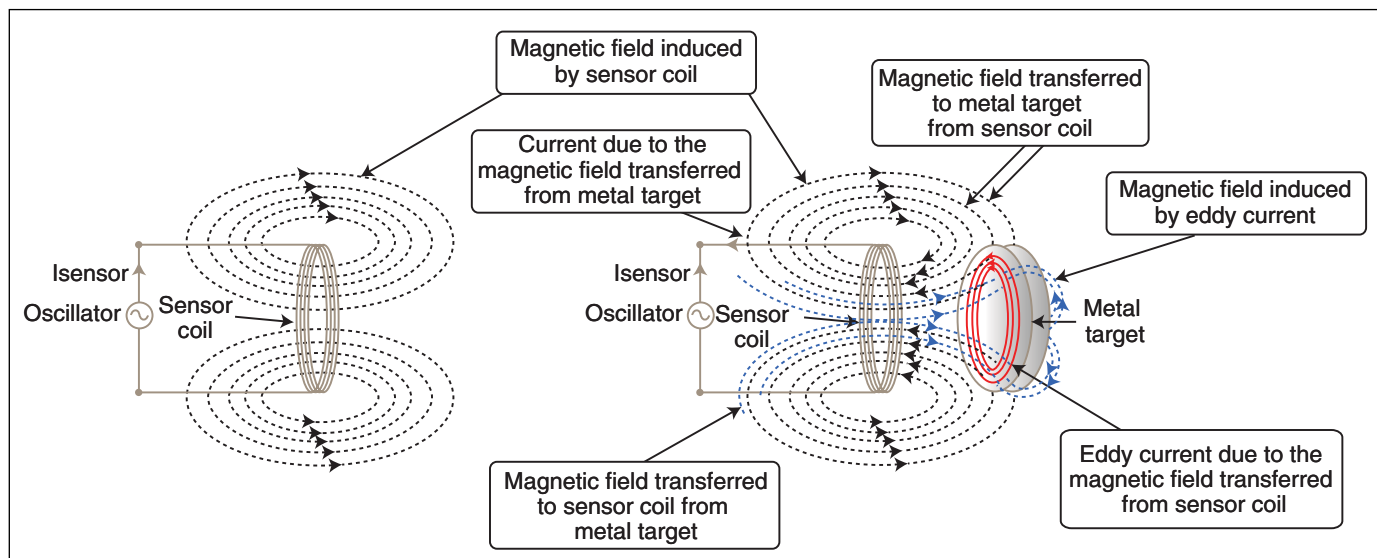


FIGURE 1

Inductive sensing technique (Source: Cypress Semiconductor application note AN219207 - Inductive Sensing Design Guide).

a circular magnetic field. That eddy current induces a reverse magnetic field, and that in turn leads to a reduction in inductance.

To cause the resonant frequency to occur, a capacitor is added in parallel to the coil to create the LC tank circuit. As the inductance starts reducing, the frequency shifts upward changing the amplitude throughout.

SOME USE CASES

Consider the use case of a Bluetooth speaker that needs to be water resistant and is intended for use in up to 2" of water for half an hour. This use case requires that the product is functional underwater. It also requires that the user can adjust the speaker in these circumstances. Such operation needs to be simple, consistent and reliable—even in the presence of water. Inductive sensing provides the solution for this. That's because it has nothing much to do with the change in dielectric, but is only concerned with the metal detection.

For this application, metal-over-touch using inductive sensing would provide a consistent and reliable user performance (**Figure 2**). A light deflection in metal can be detected as touch. Alternatively, a mechanical button and/or dial could be used. However, a mechanical interface is costly compared to a coil printed on a PCB and connected to a few passive components. Additionally, a mechanical button can break or fail, providing a much shorter useable lifespan than an inductive button would.

Consider another use case for proximity sensing using inductive sensing technology. A vehicle detection system needs to monitor when another vehicle approaches within 2 m and signal the driver on the dashboard or navigation panel. This functionality can be implemented using inductive sensing. A hardware board containing multiple coils at different locations using a flex cable, all around the dashboard, can be designed around the four corners and center of the headlight areas (**Figure 3**). Data from the inductive coils is collected by an inductive sensing controller such as the PSoC 4700S from Cypress Semiconductor. The controller would then analyze the data to determine the presence or absence of other cars in a 4-m vicinity around the vehicle.

From an engineering point of view, inductive sensing is rugged, environment-independent, and easy to design and develop. In addition, little tuning is required to achieve the desired closed loop for an application. Note: The controller need not be placed far away from the coils to improve SNR. Individual controllers can be used to optimize the design. The block diagram mentioned in **Figure 4** is a principle representation.

SYSTEM DESIGN

In this section, I'll discuss the complete system design along with the firmware design for the inductive sensing solution. To support the discussion, I will be using CY8CKIT-148

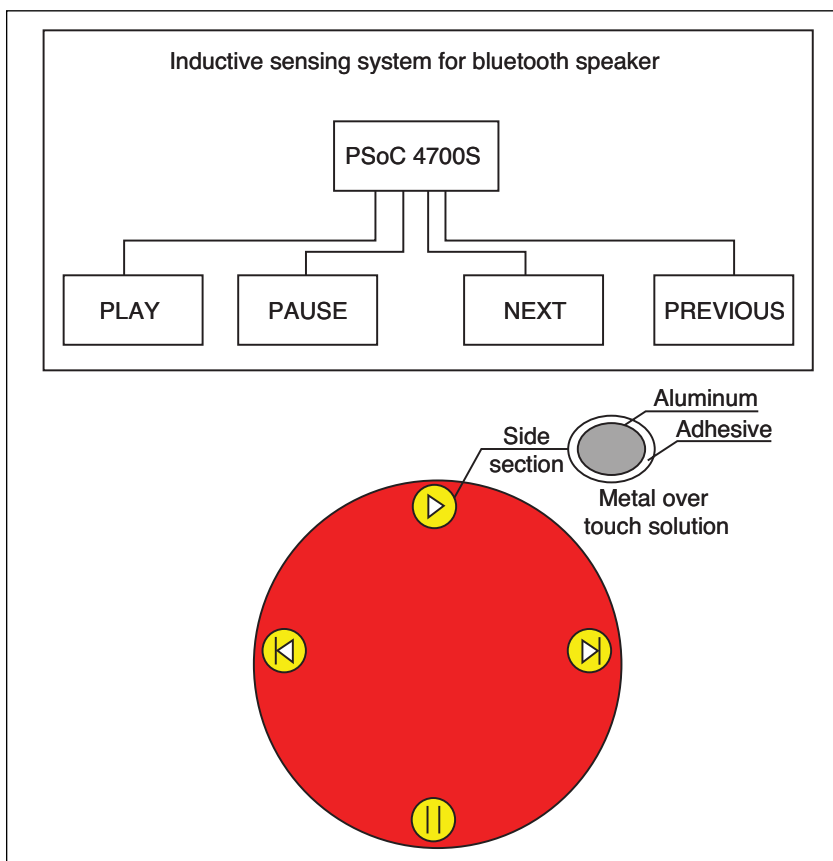


FIGURE 2

Shown here is the architecture of a water-resistant Bluetooth speaker using inductive sensing.

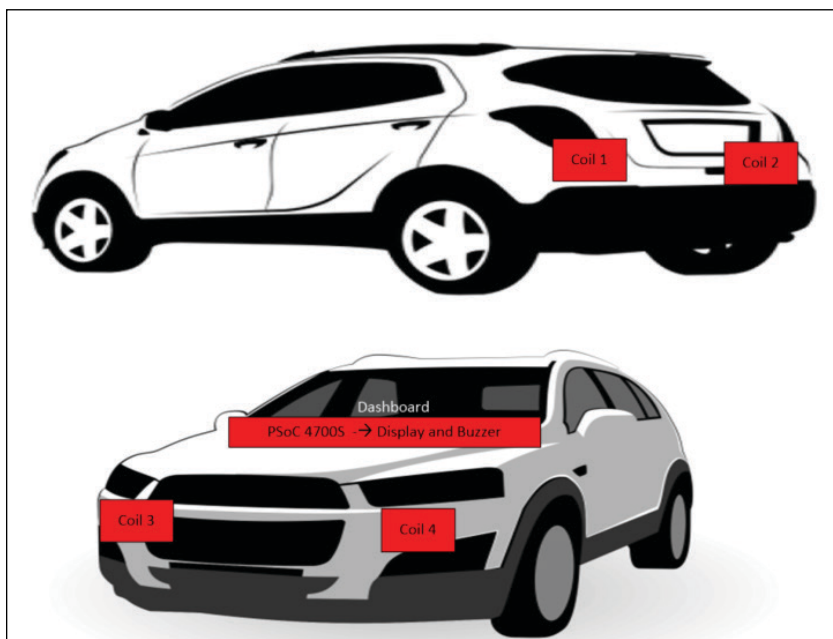


FIGURE 3

Using inductive sensing to determine vehicle proximity in an automotive application.

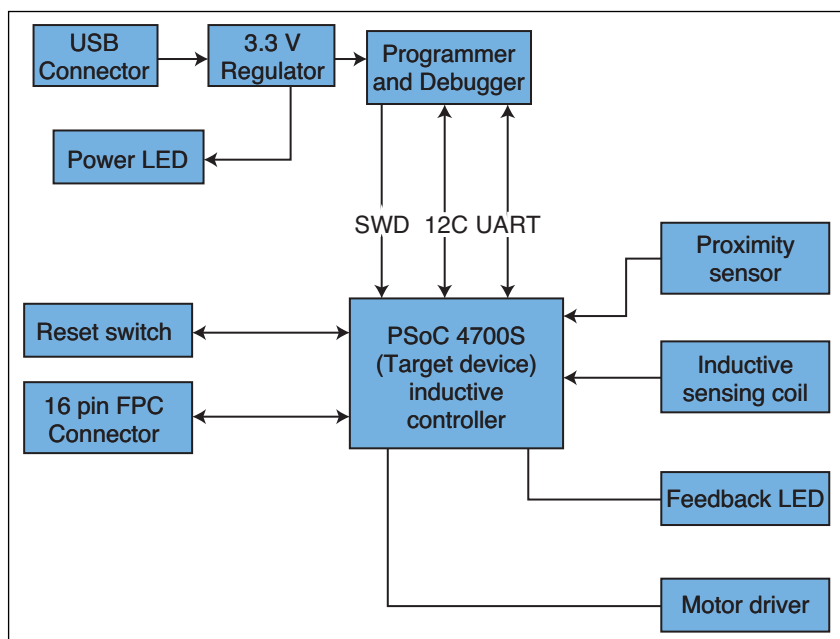


FIGURE 4
Inductive sensor block diagram

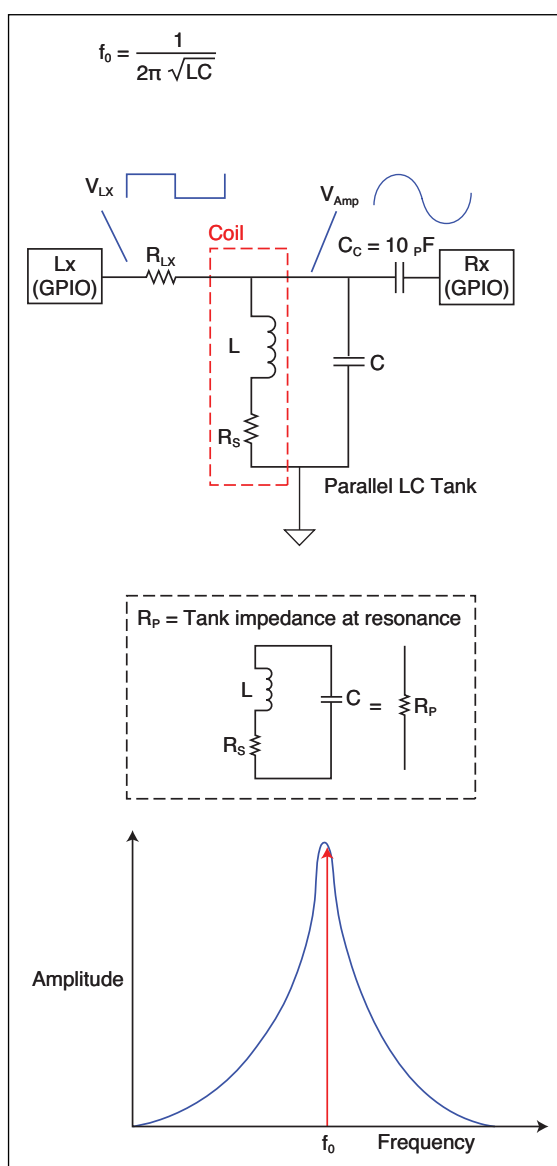


FIGURE 5
Tank circuit for Inductive Sensing example

evaluation kit from Cypress as a reference.

In general, designing an inductive sensor is straightforward (Figure 4). A typical inductive sensor requires one or more inductive coils, as determined by the requirements of the application. To learn more about designing inductive sensing boards and controllers, make sure to check out the links on the *Circuit Cellar* article materials webpage. The sensor needs to be interfaced to the controller using suitable drivers or controllers to be understood by the microcontroller (MCU). This interface can be implemented using external components. However, to reduce system design and manufacturing complexity, PSoC integrates driver and converter circuitry to convert inductive sensor data into raw counts, which can then be processed using suitable algorithms.

Figure 4 shows the complete system block diagram of a typical PSoC-based Inductive Sensing board. A typical Inductive Sensing board using PSoC would require a programming and communication device, and here we use the PSoC 5LP device family. The sensor is interfaced with PSoC 4700S device, which communicates to external world using UART/I²C or any kind of feedback interface like LEDs. To program the inductive sensing controller, we need a suitable programmer either on board or using external programmers. You need to decide the maximum power to be provided. Here we have designed the system at 3.3 V, however it can range from 1.8 V to 5 V.

Designing an inductive sensing board using PSoC is straightforward compared to some other sophisticated systems. In an inductive sensing system, we need to take care of the design of tank circuit, which plays major role in tuning the circuitry for inductive sensing. **Figure 5** shows the tank circuit involved in the functioning of the inductive sensing. Here L is the coil, Rs is the internal resistance. C is the tank capacitor, whose value is decided based on the frequency of resonance observed. Generally, the system is designed for higher frequency up to 1 MHz or 800 kHz for better performance, however lower frequency too can be chosen for it.

BUTTON OVERLAY DESIGN

The next important part in the system design of inductive sensing board is designing the metal assembly for the button. The overlay design has three major parameters that you need to decide:

1. Aluminum overlay thickness
2. Polycarbonate adhesive thickness
3. Cutout area of the sensor on the adhesive layer

Figure 6 shows the mechanical dimension of the overlay and adhesive layer for this kit. The thickness of overlay is an important parameter to decide the sensitivity of the coil response to MagSense. However, with a lower overlay thickness, the lifetime of the board reduces. A 0.5 mm thickness is typically an optimal choice from a button sensitivity and lifetime point of view. The metal target material determines the amount of deflection and

response. We recommend using an aluminum overlay for inductive sensing application because of its better deflection and response. For button applications, a higher Newton force on the overlay causes deflection throughout the overlay, leading to undesirable false triggering throughout the coils. For this use case, the user should only be able to press the buttons just enough to generate feedback. Pressing the overlay harder can even deform the overlay.

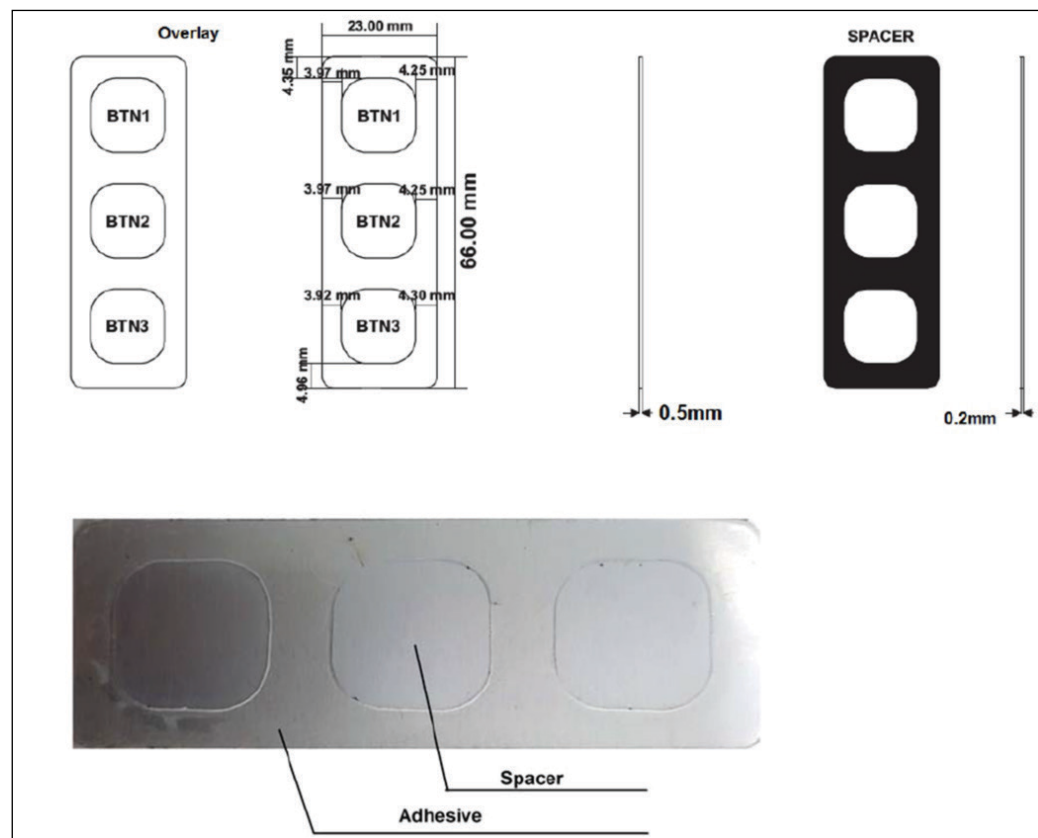


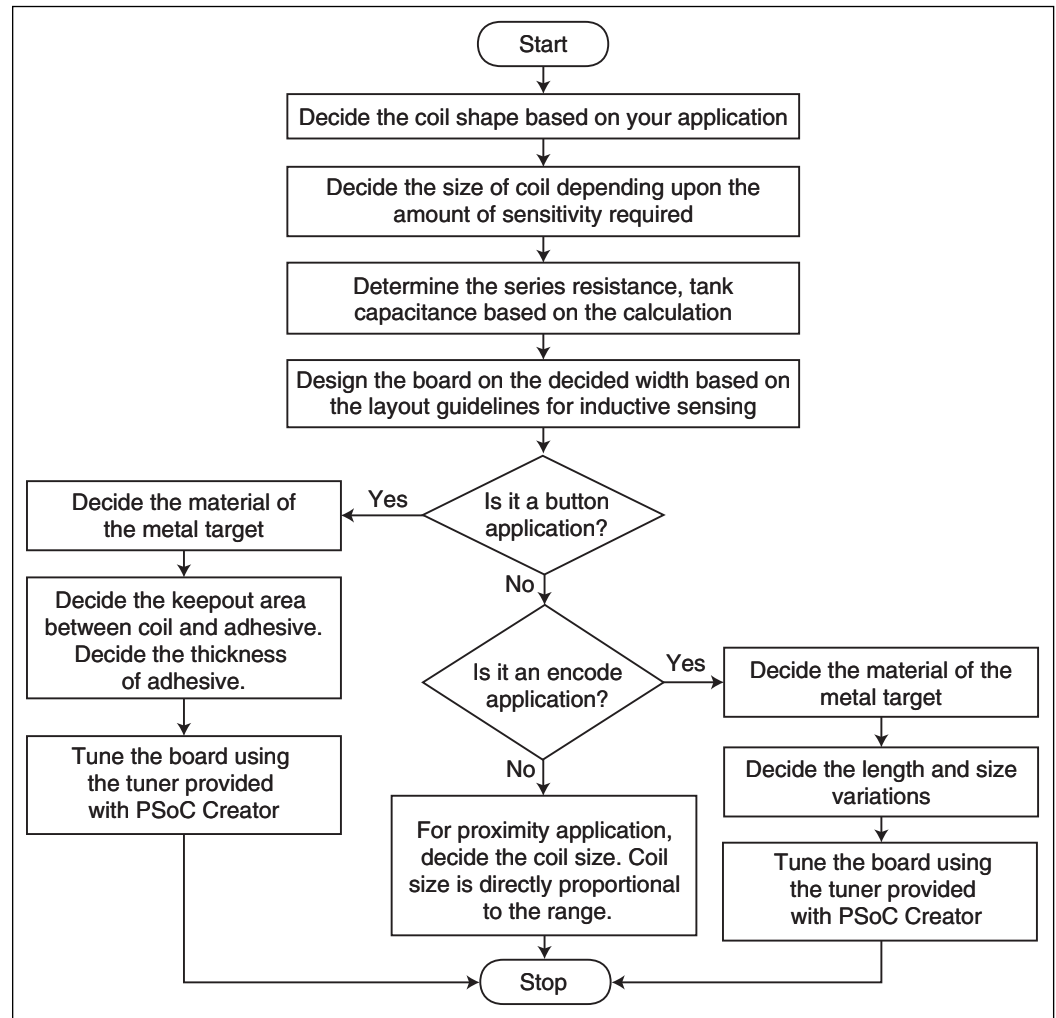
FIGURE 6
Aluminum overlay design

| Without Aluminum Target | Lp (Inductance) | Rs (Resistance) | External Parallel Capacitor | External Resistor |
|-------------------------|-----------------|-----------------|-----------------------------|-------------------|
| BTN1 | 14.1 μ H | 7 Ω | C41: 5 nF | R32: 330 Ω |
| BTN2 | 14.1 μ H | 6.78 Ω | C46: 5 nF | R39: 330 Ω |
| BTN3 | 14.13 μ H | 6.9 Ω | C51: 5 nF | R49: 330 Ω |
| PROX | 55.8 μ H | 16.9 Ω | C42: 2.2 nF | R34: 330 Ω |
| With Aluminum Target | Lp | Rs | External Parallel Capacitor | External Resistor |
| BTN1 | 5.5 μ H | 8.9 Ω | C41: 5 nF | R32: 330 Ω |
| BTN2 | 5.42 μ H | 8.5 Ω | C46: 5 nF | R39: 330 Ω |
| BTN3 | 5.51 μ H | 8.73 Ω | C51: 5 nF | R49: 330 Ω |
| PROX | 18.3 μ H | 20 Ω | C42: 2.2 nF | R34: 330 Ω |

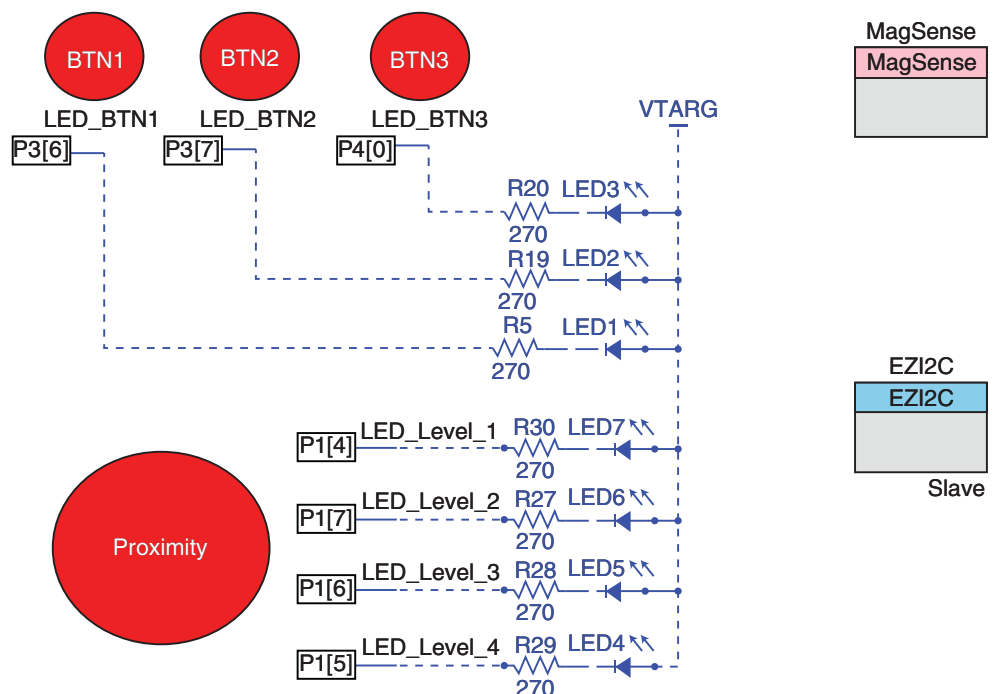
TABLE 1
Measurement of Lp and Rs using an LCR meter and values for external components

FIGURE 7

Design flow of a typical inductive sensing application



Inductive sensing example

**FIGURE 8**

Top design schematic

To tune the buttons and proximity sensor, it is necessary to measure the inductance and resistance of the coil by themselves and then with an aluminum overlay (for the buttons) and with the metal target at 2 cm (for the proximity sensor). Note here that, the proximity distance is directly proportional to the coil diameter. Once tuning has been completed, sensitivity can be adjusted by changing the resonant frequency by about 5% to 10% depending on your design. During this iteration, it is recommended to reduce the resonant frequency to detect the correct signal. In the CY8CKIT-148, there are a total of 4 inductive coils, out of which 3 are projected as buttons and 1 as proximity.

The first step in assembling the system or for the most part deciding the components for the tank circuitry, measure the inductance of the coil on board. The practical inductance observed would be different from the theoretically calculated ones.

Note: To measure the values of L and R of the coil, make sure to isolate it from other parts of the board by desoldering the series and parallel components of the coil on board. This is needed to get an accurate reading from the LCR meter. **Table 1** shows the inductance and series resistance measured for the coils present in CY8CKIT-148 kit.

Figure 7 shows the design flow involved for a typical inductive sensing application. First, you assess how sensitive the system needs to be. Sensitivity determines the coil size and its number of turns are decided. The application also impacts the shape of the coil. For example, a slider interface requires a series of squares or an elongated rectangle. The next step is to calculate the tank capacitor and the inductance based on the number of turns, spacing, width, and diameter. To understand the detailed steps, refer to the Inductive Sensing Design guide. A link is available on *Circuit Cellar's* article materials webpage.

Once all these things are intact, the board is designed and fabricated. The advantage of PSoC Creator IDE is, it provides a user-friendly Inductive Sensing Tuner GUI that can be used by designers to serve their design needs

FIRMWARE DESIGN

This example showcases the features of inductive sensing in CY8CKIT-148 PSoC 4700S Inductive Sensing Evaluation Kit. When the user presses the buttons, the corresponding LEDs glow. When you bring a metal target, provided with the kit, from up to a 2 cm distance, closer to the proximity coil, the number of LEDs glowing on the board increases. A pre-requisite for the code example is that the reader must be well

versed with PSoC Creator. To understand the PSoC Creator environment, refer to my three-part article series "Getting Started with PSoC Microcontrollers" in July, August and September 2017 (*Circuit Cellar* 324, 325, 326).

This example has a single workspace: Inductive Sensing Example. The code is available on *Circuit Cellar's* article code download page. The MagSense component is configured for three buttons and one proximity coil. The EZI2C Slave Component is

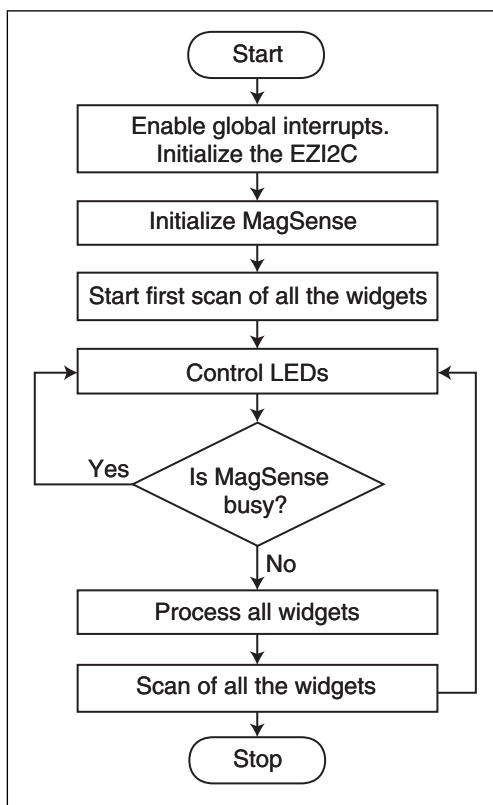


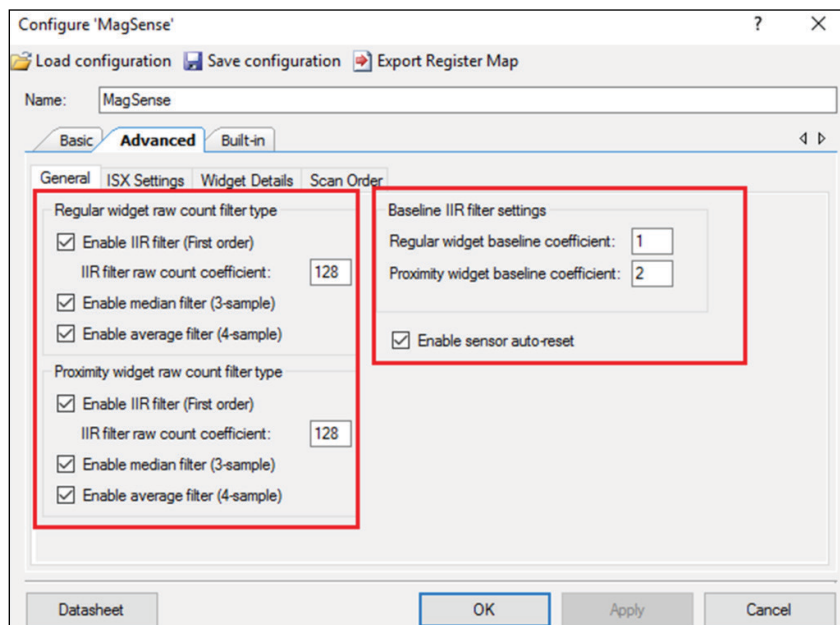
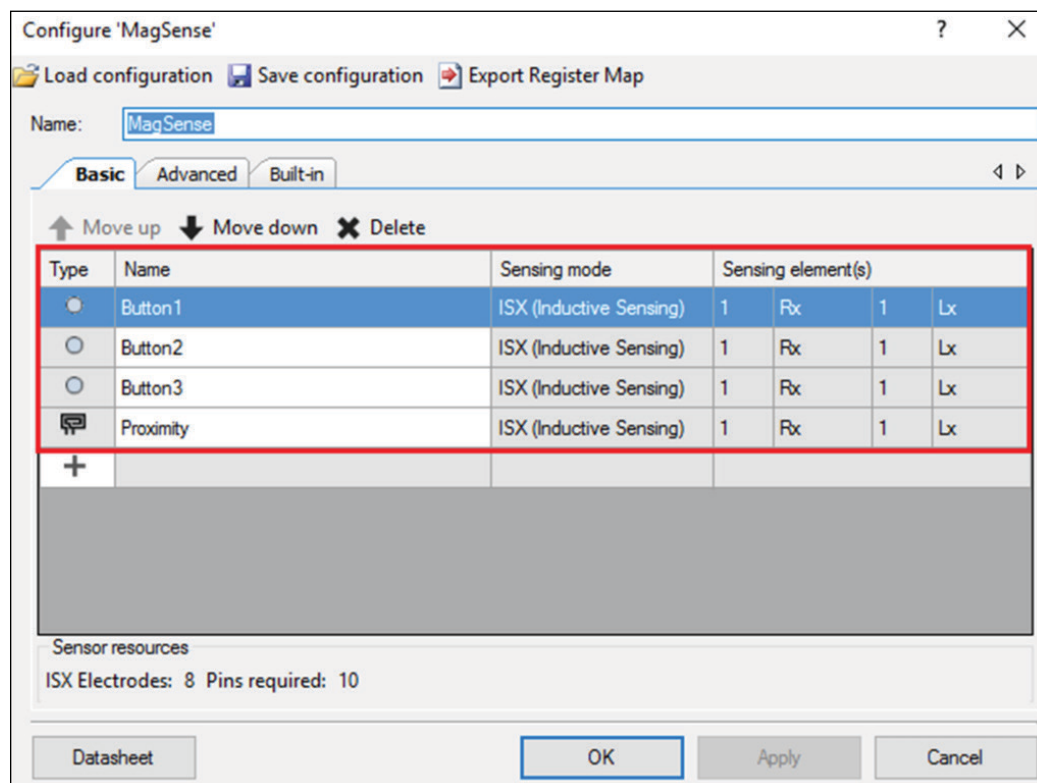
FIGURE 9
Flowchart of main.c

| Component | Instance Name | Purpose |
|---|---------------|--|
| MagSense (6.0) | MagSense | Inductive Sensor processing and configuration |
| PSoC 4 Serial Communication Block (SCB) (4.0) | EZI2C | Communication with external interfaces or with Tuner |
| Digital Output Pin (2.20) | LED_BTN1 | Provide visual feedback |
| Digital Output Pin (2.20) | LED_BTN2 | Provide visual feedback |
| Digital Output Pin (2.20) | LED_BTN3 | Provide visual feedback |
| Digital Output Pin (2.20) | LED_Level_1 | Provide visual feedback |
| Digital Output Pin (2.20) | LED_Level_2 | Provide visual feedback |
| Digital Output Pin (2.20) | LED_Level_3 | Provide visual feedback |
| Digital Output Pin (2.20) | LED_Level_4 | Provide visual feedback |

TABLE 2
PSoC Creator components

FIGURE 10

MagSense component configuration: basic tab

**FIGURE 11**

MagSense component configuration: advanced tab, general settings

ABOUT THE AUTHOR

Nishant Mittal is a Systems Design Engineer in Bangalore, India.

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

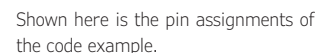
RESOURCES

Cypress Semiconductor | www.cypress.com

used to monitor sensor data on the computer using the MagSense Tuner, available in the PSoC Creator integrated design environment (IDE). **Figure 8** shows the PSoC Creator schematic for this code example. This code example uses MagSense, SCB (configured as EZI2C Slave) and pins used to drive LEDs.

Table 1 shows the measurement of the inductance and resistance of the coil from an LCR meter as well as external capacitance and resistance components on the board. Based on these measurements the frequency for buttons are calculated to be 959 kHz. Similarly, the resonant frequency of the tank circuit for the proximity coil is calculated to 793 kHz. The code example firmware is implemented in the file main.c from the project. It implements a scanning algorithm as shown in **Figure 9**. **Table 2** lists the PSoC Creator Components used in this example, how they are used in the design, and the non-default settings required so they function as intended. For information on the hardware resources used by a Component, see the Component datasheet. **Figure 10** shows the configuration for the MagSense Component.

Enabling filters are necessary to eliminate noise. **Figure 11** shows the settings for filter assignments on the component. An auto-reset feature has been enabled to prevent any kind of false triggering or stuck-on sensors. False triggering can happen due to touching the tank circuit, touching the coils or keeping the device on a metallic base. The



Clock settings of the code example in PSoC Creator

Note: When you operate the button or proximity sensor for a longer duration, the auto-reset feature will automatically disable the button or proximity and its corresponding LED.



Building a Generator Control System

Three-Phase Power

Three-phase electrical power is a critical technology for heavy machinery. Learn how these two US Coast Guard Academy students built a physical generator set model capable of producing three-phase electricity. The article steps through the power sensors, master controller and DC-DC conversion design choices they faced with this project.

By Kent Altobelli and Caleb Stewart

From left to right: Aaron Dahlen, Caleb Stewart, Kent Altobelli and Christopher Gosvener.

Three-phase electrical power is typically used by heavy machinery due to its constant power transfer, and is used on board US Coast Guard cutters to power shipboard systems while at sea. In most applications, electrical power is generated by using a prime mover such as a diesel engine, steam turbine or water turbine to drive the shaft of a synchronous generator mechanically. The generator converts mechanical power to electrical power by using a field coil (electromagnet) on its spinning rotor to induce a changing current in its stationary stator coils. The flow of electrons in the stator coils is then distributed by conductors to energize various systems, such as lights, computers or pumps. If more electrical power is required by the facility, more mechanical power is needed to drive the generator, so more fuel, steam or water is fed to the prime mover. Together, the prime mover and the generator are referred to as a generator set “genset”.

Because the load expects a specific voltage and frequency for normal operation, the genset must regulate its output using a combination of its throttle setting and rotor field strength. When a real load, such as a light bulb, is switched on, it consumes more real power from the electrical

distribution bus, and the load physically slows down the genset, reducing the output frequency and voltage. The shaft rotational speed determines the number of times per second the rotor’s magnetic field sweeps past the stator coils, and determines the frequency of the sinusoidal output. Increasing the throttle returns the frequency and voltage to their setpoints.

When a partially reactive load—for example, an induction motor—is switched on, it consumes real power, but also adds a complex component called “reactive power.” This causes a voltage change due to the way a generator produces the demanded phase offset between supplied voltage and current. An inductive load, common in industrial settings, causes the voltage output to sag, whereas a capacitive load causes the voltage to rise. Voltage induced in the stator is controlled by changing the strength of the rotor’s electromagnetic field that sweeps past the stator coils in accordance with Faraday’s Law of inductance. Increasing the voltage supply to the rotor’s electromagnet increases the magnetic field and brings the voltage back up to its setpoint.

The objective of our project was to build a physical generator set model capable of

producing three-phase electricity, and maintain each “Y”-connected phase at an output voltage of 120 ± 5 V RMS (AC) and frequency of 60 ± 0.5 Hz. When the load on the system changes, provided the system is not pushed beyond its operating limits, the control system should be capable of returning the output to the acceptable voltage and frequency ranges within 3 seconds. When controlling multiple gensets paralleled in island operation, the distributed system should be able to meet the same voltage and frequency requirements, while simultaneously balancing the real and reactive power from all online gensets.

TWO CONFIGURATIONS

Gensets supply power in two conceptually different configurations: “island” operation with stand-alone or paralleled (electrically connected) gensets, or gensets paralleled to an “infinite” bus.” In island operation, the entire electrical bus is relatively small—either one genset or a small number of total gensets—so any changes made by one genset directly affects the voltage and frequency of the electrical bus. When paralleled to an infinite bus such as the power grid, the bus is too powerful for a single genset to change the voltage or frequency. Coast Guard cutters use gensets in island operation, so that is the focus of this article.

When in island operation, deciding how much to compensate for a voltage or frequency change is accomplished using either droop or isochronous (iso) control. Droop control uses a proportional response to reduce error between the genset output and the desired setpoint. For example, if the frequency of the output drops, then the throttle of the prime mover is opened

correspondingly to generate more power and raise the frequency back up. Since a proportional response cannot ever achieve the setpoint when loaded (a certain amount of constant error is required to keep the throttle open), the output frequency tends to decrease linearly with an increase in power output. A no-load to full-load droop of 2.4 Hz is typical for a generator in the United States, but this can usually be adjusted by the user.

Frequency control typically uses a mechanical governor to provide the proportional throttle response to meet real power demand. Voltage control typically uses an automatic voltage regulator (AVR) to manipulate the field coil strength to meet reactive power demand. Isochronous mode is more challenging, because it always works to return the genset output to the setpoint. Maintaining zero error on the output usually requires some combination of a proportional response to compensate for load fluctuation quickly, and also a long-term fine-tuning compensation to ensure the steady-state output achieves the setpoint.

If two or more gensets are paralleled, the combined load is supplied by the combined power output of the gensets. As before, maintaining the expected operating voltage and frequency is the first priority, but with multiple gensets, careful changes to the throttle and field can also redistribute the real and reactive power to meet real and reactive power demand efficiently.

If the average throttle or field setting is increased, then the overall bus frequency or voltage, respectively, also increases. If the average throttle or field setting stays the same while two gensets adjust their settings in opposite directions, the frequency or voltage stay the same, but the genset that increased

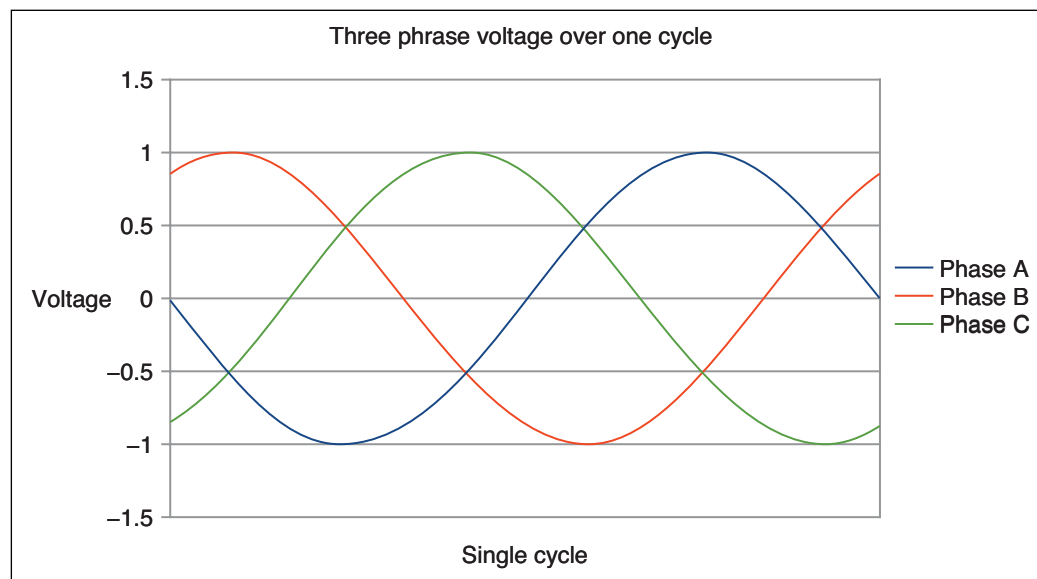


FIGURE 1

Arbitrary three phase sinusoid

their throttle or field provides a greater portion of the real or reactive power. Redistribution is important because it allows gensets to produce real power at peak efficiency and share reactive power evenly, because excessive reactive power generation derates the generator. Reactive currents flowing through the windings cause heat without producing real, useful power.

FOUR CONDITIONS

Before the breaker can be closed to parallel generators, four conditions need to be met between the oncoming generators and the bus to ensure smooth load transfer:

- 1) The oncoming generator should have the same or a slightly higher voltage than the bus.
- 2) The oncoming generator should have the same or a slightly higher frequency than the bus.
- 3) The phase angles need to match. For example, the oncoming generator "A" phase needs to be at 0 degrees when the bus "A" phase is at 0 degrees.
- 4) The phase sequences need to be the same. For example, A-B-C for the oncoming generator needs to match the A-B-C phase sequence of the bus.

Meeting these conditions can be visualized using **Figure 1**, which shows a time vs. voltage representation of an arbitrary, balanced three-phase signal. The bus and the generator each have their own corresponding plots resembling Figure 1, and

the two should only be electrically connected if both plots line up and therefore satisfy the four conditions listed above.

If done properly, closing the breaker will be anticlimactic, and the gensets will happily find a new equilibrium. The gensets should be adjusted immediately to ensure the load is split evenly between gensets. If there is an electrical mismatch, the generator will instantly attempt to align its electrical phase with the bus, bringing the prime mover along for a wild ride and potentially causing physical damage—in addition to making a loud BANG! Idaho National Laboratories demonstrated the physical damage caused by electrical mismatch in its 2007 Aurora Generator Test.

Three primary setups for parallel genset operation are discussed here: droop-droop, isochronous-droop, and isochronous-isochronous. The simplest mode of parallel operation between two or more gensets is a droop-droop mode, where both gensets are in droop mode and collectively find a new equilibrium frequency and voltage according to the real and reactive power demands of the load.

Isochronous-droop (iso-droop) mode is slightly more complex, where one genset is in droop mode and the other is in iso mode. The iso genset always provides the power required to maintain a specific voltage and frequency, and the droop genset produces a constant real power corresponding to that one point on its droop curve. Because the iso genset works more or less depending on the load, it is also termed the "swing" generator.

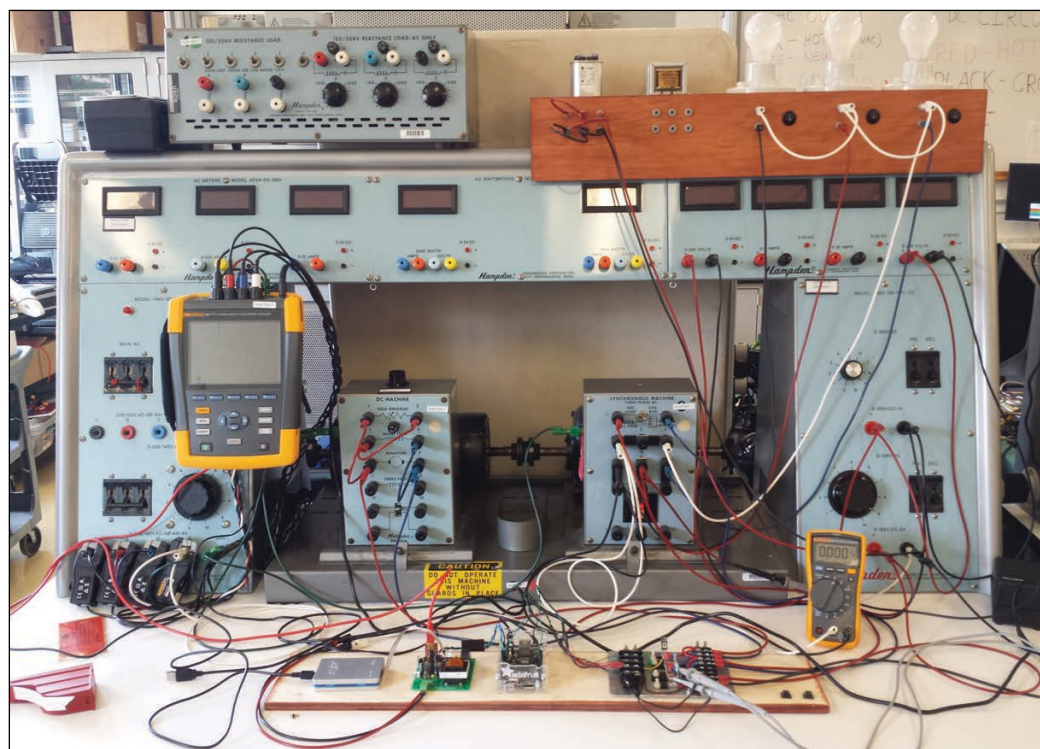


FIGURE 2

Simulated genset on the Hampden Training Bench

Finally, isochronous-isochronous (iso-iso) is the most complex. In iso-iso mode, both gensets attempt to maintain the specified output voltage and frequency. While this sounds ideal, this mode has the potential for instability during transient loading, because individual genset control systems may not be able to differentiate between a change in load and a change in the other genset's power output. Iso-iso mode usually requires direct communication or a higher level controller to monitor both gensets, so they respond to load changes without fighting each other. With no external communication, one genset could end up supplying the majority of the power to the load while the second genset is idling, seeing no need to contribute because the bus voltage and frequency are spot on! At some point one genset could even resist the other genset, consuming real power and causing the generator to "motor" the prime mover. Unchecked, this condition will damage prime movers, so a reverse power relay is usually in place to trip the genset offline, leaving only one genset to supply the entire load.

SYSTEM DESIGN

Each genset simulated on the Hampden Training Bench had a custom sensor monitoring the generator voltage, current, and frequency output, a small computer running control calculations and a pair of DC-to-DC converters to close the control loop on the generator's rotational velocity and field strength. The genset was simulated by coupling a 330 W brushed DC motor acting as the prime mover to a four-pole 330 W synchronous generator (**Figure 2**). Our power sensor was a custom-designed circuit board with an 8-bit microcontroller (MCU) employed to sample the genset output continuously and provide RMS voltage, RMS current, real power, reactive power, and frequency upon request. The control system ran on a Linux computer with custom software designed to poll the sensor for data, calculate the appropriate control response to return the system to the set point and generate corresponding pulse width modulated (PWM) outputs. Finally, the PWM outputs controlled the DC-to-DC converter to step down the DC supply voltage to drive the prime mover and energize the generator field coil. The component relationships are shown in **Figure 3**, where the diesel engine in a typical genset was replaced by our DC motor.

Since this project was a continuation of a previous year of work by Elise Sako and Jasper Campbell, several lessons were learned that required the system be redesigned from the ground up. One of the

largest design constraint from the previous year was the decision to use a variable frequency drive (VFD) to drive an induction motor as the prime mover. While this solution is acceptable, it introduces inherent delay in the control loop, because the VFD is designed to execute commands as smoothly but not necessarily as quickly as possible.

Another design constraint was the decision to power the generator field coil using DC regulated by an off-the-shelf silicon controller rectifier (SCR) chopper. Again, while this is an acceptable solution, the system output suffered from the SCR's slow response time (refresh rate is limited to the AC supply frequency), and voltage output regulation was non-ideal (capacitor voltage refresh again limited by the frequency of the AC supply).

To solve these performance constraints, we selected the responsive and easily controllable DC motor as the prime mover so the DC output from our Hampden Training Bench could be used as the power supply for both the DC motor and the generator field coil. By greatly simplifying the electrical control of the genset, we reduced implementation cost and improved control system response time.

POWER SENSOR

The power sensor provided control feedback by taking continuous voltage and current measurements on a single phase of the generator output. Based on the design from the previous year, the sensor circuit was redesigned in Eagle CAD and printed. Ultimately the two-layer board routed just over 60 components on the top layer with an unbroken ground plane on the bottom, and measured 3" by 3". The circuit features a Microchip (formerly Atmel) ATmega328p MCU clocked at 20 MHz. Software libraries were developed for the Analog to Digital Converter (ADC), Inter-Integrated Circuit (I²C) and Universal Asynchronous Receive-Transmit (UART) MCU peripheral modules.

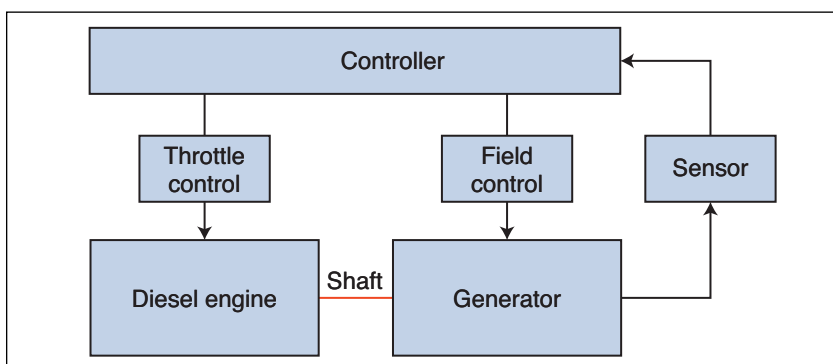


FIGURE 3
Genset component layout

A 2.5 kHz sampling rate was selected based on available clock prescalers for interrupt service routines, which gave a nearly perfect 42 samples per 60 Hz cycle. Voltage and current samples were managed with a circular buffer, and real-time RMS voltage, RMS current, real power, reactive power and frequency values for the connected phase were continuously calculated to be available immediately when data were requested.

Using the previous year's design for reference, we refined the sensor schematic and hardware for sampling speed. To measure phase voltage, a single-phase and neutral (this configuration only works for a balanced "Y"-connected load) was fed into the primary coil of a small isolation transformer. The resulting voltage differential on the secondary of the transformer was treated as a complementary signal, and each path was attenuated independently using resistor dividers collectively biased around 2.5 V to allow the signal to occupy the full 5 V range of the ADC. ADC input filtering capacitors were selected according to the voltage divider impedance to low pass filter the signal at 1.7 kHz, rejecting noise at the ADC input pins. This corner frequency was selected because it avoided significant phase shift in the 60 Hz signal of interest, the phase shift of which is especially important for reactive power calculations.

To measure current, the generator output wire from the same phase was run through a current transformer loop. The secondary of the current transformer was loaded with

a 100 Ω resistor, and the voltage difference, proportional to the current, was again treated as two complementary signals. After biasing around 2.5 V, the two signals were amplified using a rail-to-rail op amp in an instrumentation amplifier configuration to occupy the full 5 V range of the ADC. The final circuit board design shown in **Figure 4** is rated to measure up to 340 V_{RMS} and 12 A_{RMS} on a single phase.

Several aspects of the C program were optimized to ensure faster sampling rates. Lookup tables were used for finding squares in real time, and resource-intensive square root and division calculations were offloaded to the Linux computer running our control system. RMS voltage was calculated using Equation 1, and RMS current was calculated using Equation 2 for our discrete series of voltage and current samples. Real and reactive power were calculated using Equations 3 and 4, respectively, which were adapted for discretely sampled signals.

Reactive power required a unique calculation, which delayed the voltage samples by 90 degrees from the current samples to isolate the purely reactive component. Forty-two samples per cycle is not perfectly divisible by 4, so a delay of 10 samples (instead of 10.5) was used for the reactive power calculation, and we incurred a continuous 4.3 degrees phase offset error.

To determine signal frequency, the two samples adjacent to the zero crossing were linearly interpolated to find the estimated zero crossing. The estimated time stamp of the previous zero crossing was subtracted from the present one, and the period was inverted to find frequency using the relationship shown in Equation 5. After the effective smoothing of the RMS calculations, the 42 samples per cycle produced excellent results, and sampling far above the Nyquist rate (of our primary 60 Hz signal) eliminated the need for signal reconstruction to determine the generator output frequency and voltage.

$$V_{RMS} = \sqrt{\frac{\sum_{n=1}^{nsamples} v(n)^2}{nsamples}} \quad [1]$$

$$I_{RMS} = \sqrt{\frac{\sum_{n=1}^{nsamples} i(n)^2}{nsamples}} \quad [2]$$

$$P = \frac{\sum_{n=1}^{nsamples} v(n) \times i(n)}{nsamples} \quad [3]$$

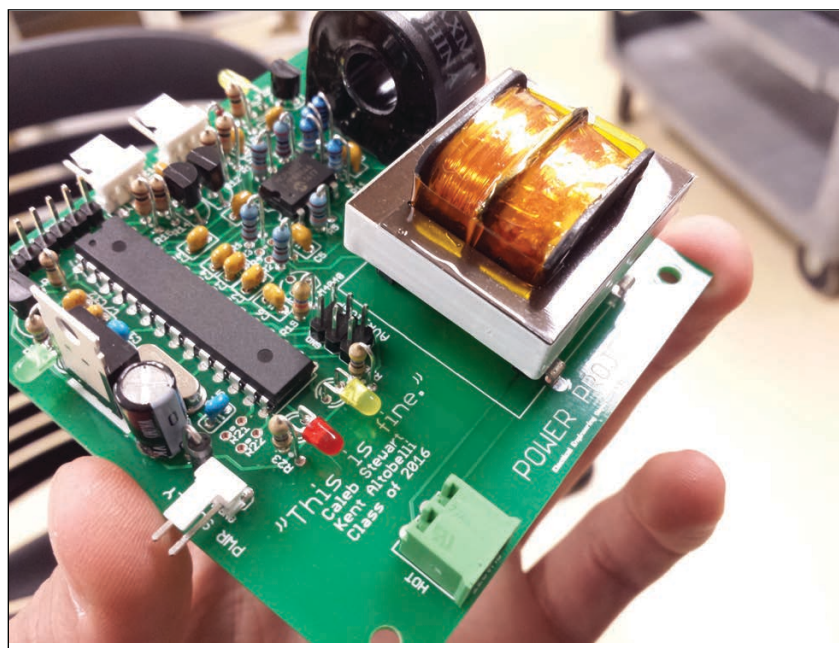


FIGURE 4
Power sensor circuit board

$$Q = \frac{\sum_{n=1}^{n_{\text{samples}}} V_{90^\circ \text{delay}}(n) \times i(n)}{n_{\text{samples}}} \quad [4]$$

$$f = \frac{1}{T} \quad [5]$$

Because sampling loop speed was of primary importance, calculations were converted to a looping summation to ensure the values were always up-to-date when polled by the Linux computer. To accomplish this with minimal processing power, a moving window was used to identify the "active" samples in the circular buffer, so the program could simply add new samples to a running summation and remove the old samples.

Integer math was used for all summations to increase calculation speeds, which conveniently removed the possibility of error accumulation that might occur over time if using floating point math. When requested, the summation was sent to the Linux computer, which completed the divisions and/or square roots necessary, and scaled the ADC units to equivalent floating-point SI units. Scaling factors were initially determined mathematically, then adjusted empirically to ensure accuracy.

Curiously, our sensor sometimes produced RMS voltage readings greater than 1,000 V, or frequencies exactly half of the true system frequency. We had various hypotheses for these anomalies, and investigated edge cases in our integer summation windowing and zero-point crossing implementation, but were unable to resolve the spurious errors. The anomalies seemed to occur randomly, from multiple times per second to several seconds in between occurrences, so ultimately, we opted to remove the outliers with very specific filters. It was disappointing to not find the true solution, but we were forced to move forward due to time constraints.

GENSET CONTROLLER

The Linux computer running the control system had to bridge the gap between the low-level I²C data input, the PWM output and the higher-level network communication that would eventually be needed for multiple gensets to coordinate. Initially, we developed our control system to run on the low-cost and well documented Raspberry Pi. However, because the Raspberry Pi only generates a single PWM output, we ultimately switched to the BeagleBone development board, which had multiple onboard PWM signal generators. The BeagleBone is a similar Linux-based platform, so nearly all the code originally developed for the Raspberry Pi was simply compiled for the new architecture. The code developed for I²C communication with the sensor board remained the same, but a small library was written to control the new PWM hardware with a 10-bit duty cycle command.

After we nailed down our choice of silicon, we implemented the control system to regulate the system output at our desired 120 V_{RMS} and 60 Hz. As the first step in any control system problem, we wanted to identify our system. Because of the perceived complication of implementing a true Multiple Input Multiple Output (MIMO) control system, however, we decided to make do with two Single Input Single Output (SISO) control systems running side by side. Ideally, frequency feedback would be used to control "throttle," and the voltage feedback would be used to control the strength of the field coil electromagnet.

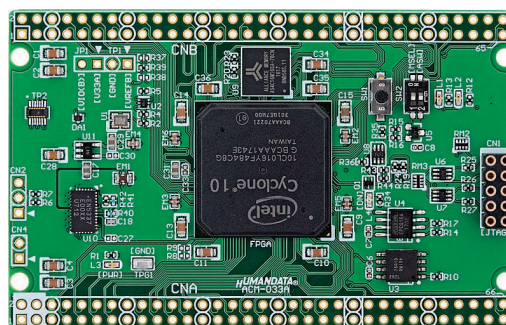
FPGA Boards from HUMAN DATA[®] JAPAN

SAVING COST=TIME with readily available FPGA boards

- Basic and simple features, single power supply operation
- Free download technical documents before purchasing

INTEL ACM-033

Intel Cyclone 10 LP F484 FPGA board



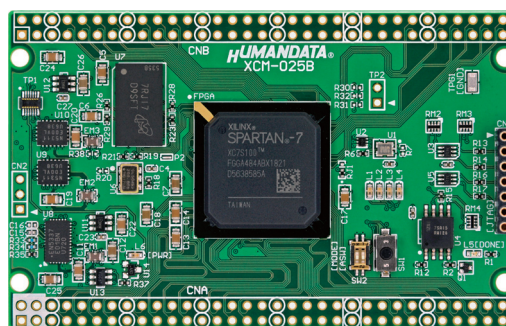
Cyclone 10 LP SDRAM SPI-Flash RoHS

SIZE : 3.386" x 2.126" (86 x 54 mm)

ACM-033 is an FPGA board with Intel high-performance FPGA Cyclone 10 LP. It's compact and very simple. 3.3V single power supply operation.

XILINX XCM-025

Xilinx Spartan-7 FGA484 FPGA board



Spartan-7 DDR3 RoHS

SIZE : 3.386" x 2.126" (86 x 54 mm)

XCM-025 is an FPGA board with Xilinx high-performance FPGA Spartan-7. It's compact and very simple. 3.3V single power supply operation.

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

www2.hdl.co.jp/CC19B



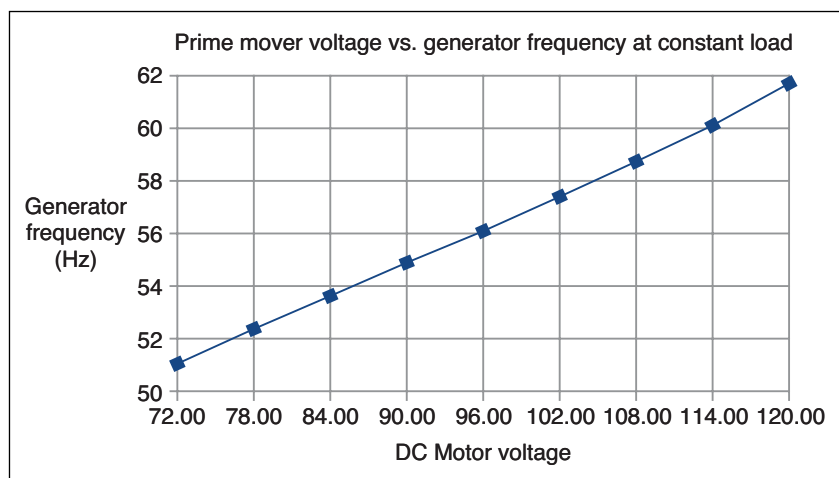


FIGURE 5
Prime mover voltage vs. generator frequency

The two parallel proportional and integral (PI) control systems were tuned separately to account for the slow mechanical time constant of the machine's rotational speed and the faster electrical time constant of the rotor field strength.

Although it was easier for prototyping, the two SISO control systems were not truly independent from each; the generator output voltage and rotational speed affect one another. Another factor that was largely ignored during empirical tuning was the dynamic asymmetry in controlling rotational speed. It was easy to spin the machine up by increasing the throttle, but the machine could only slow down when coasting under load.

DC motors are known for their nearly linear relationship between terminal voltage and rotational speed (in steady state). As shown in **Figure 5**, an increase in applied DC motor terminal voltage "throttle" resulted in a linear increase in output frequency.

Mathematically, we see the indicated proportional relationship between EMF and rotational speed (n) in Equation 6, because the motor's speed constant (k_n) and motor magnetic flux (Φ) are constants set during motor construction. Accounting for the voltage drop over the low-resistance motor windings, the relationship between terminal voltage and rotational speed are nearly linear. As the mechanically coupled generator is a synchronous machine, the electrical output frequency is approximately proportional to the terminal voltage on the DC motor.

$$EMF = k_n \phi n \quad [6]$$

Due to our haste to get the system up and running, we neglected to take measurements supporting the relationship between field coil voltage and generator output voltage, but we had experienced rough linearity when manually adjusting the machines. In lieu of data, we can analyze the governing equations to derive the expected response.

Inside a synchronous machine, the field coil is a spinning electromagnet, the current of which is proportional to the supplied voltage in accordance with Ohm's Law. Equation 7 shows the Biot-Savart Law reduced to solve for the magnetic field (B) at the center of a single loop with a constant radius (r). Assuming that the permeability (μ) of the rotor, air gap, and stator are constant throughout the rotation, we see that the magnetic field of the rotor increases linearly with an increase in the current.

The derivative form of Faraday's Law of induction in Equation 8 shows that EMF induced in a conductor is the change in magnetic flux, magnetic field (B) times area (A), over time. As differentiation is linear, any scaling of the magnetic field results in a linear scaling of the EMF experienced in the stator windings of a constant winding area. Therefore, as long as the generator's core remains unsaturated, there should be a nearly linear relationship between the voltage applied to the field coil and the output voltage on the stator coils—again assuming the resistance of the stator windings are low, so voltage drop is minimal with increases in current.

$$B = \frac{\mu I}{2r} \quad [7]$$

$$EMF = \frac{d\phi}{dt} \quad [8]$$

After showing that the independent control systems of our genset should

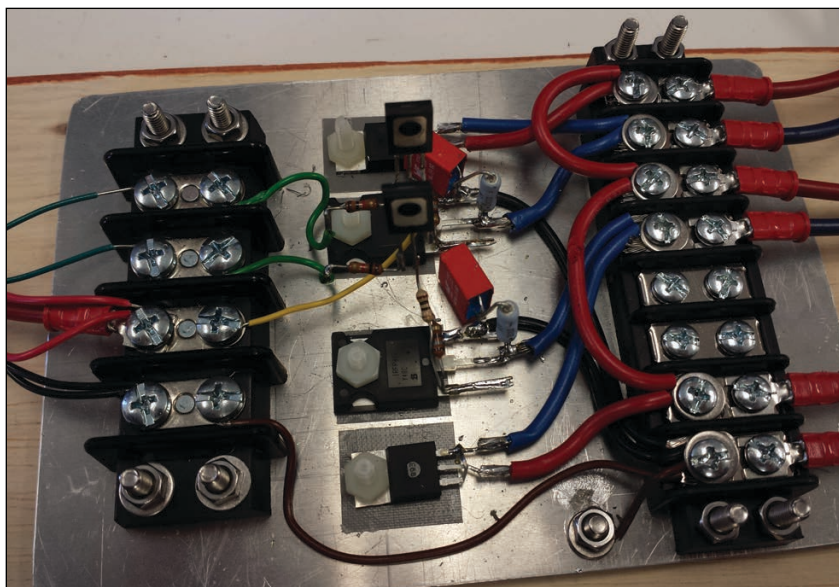


FIGURE 6
DC chopping circuit

respond linearly, we ran our two PI control loops at 20 Hz to make constant corrections to the DC motor terminal voltage and the field settings. Armed with data from our power sensor, the controller calculated the error between the present state and the setpoint. For the proportional component, the controller multiplied the error by a fixed gain coefficient to determine the action needed to return the system to the set point, just like standard droop control. The integral component kept a running integral of the system's error—the summation of error times loop period—and integrated error was multiplied by a gain coefficient and summed with the proportional component to determine the overall iso control response. Both the proportional and integral terms contributed to different aspects of the control system. Proportional response helped the genset react quickly, and integral response fine tuned the output to ensure steady state achieved the set point.

Because voltage was not produced on the genset output until the machine was spinning and no load could result in overspeed damage, we used a predetermined open loop start-up routine to ramp up speed and field voltage to 60% over the course of 5 seconds.

Once the genset reached the end of its start-up routine, the control system took over. By delaying the integral term from accumulating until the generator control loop was closed, integral wind-up was avoided.

DC-TO-DC CONVERTER

The final piece of equipment necessary to close the loop on our control system was the DC-to-DC converter to drive the DC motor and energize the generator field coil. For simplicity, we used a metal oxide semiconductor field effect transistor (MOSFET) as the power switch to "chop" the DC voltage from the Hampden Training Bench by rapidly turning the MOSFET on and off at the determined duty cycle. Several driver circuits and drive frequencies were tested before settling on a reliable design for the DC-to-DC converter. Ultimately, the open-collector of the opto-isolator receiving the PWM signal was used to drive the base of a PNP transistor in a common emitter configuration to supply the gate of our power MOSFET with the 12 V needed for saturation. Although this solution actively drove the MOSFET to conduct, the simplicity of the unipolar drive required a resistor to bleed the MOSFET gate current when

Prevent Complications!



QM7 Power Supplies - 1500W Modular Power Supplies

TDK-Lambda's latest QM7 modular power supplies feature medical and ITE safety certifications with full MOPPs isolation. Available with up to 16 outputs ranging from 2.8 to 52.8V output, the QM7 offers industry leading flexibility.

Prevent complications - check out our simple to use on-line configurator to obtain a fully optimized product configuration and part number at:
<https://config.emea.tdk-lambda.com/nv/>

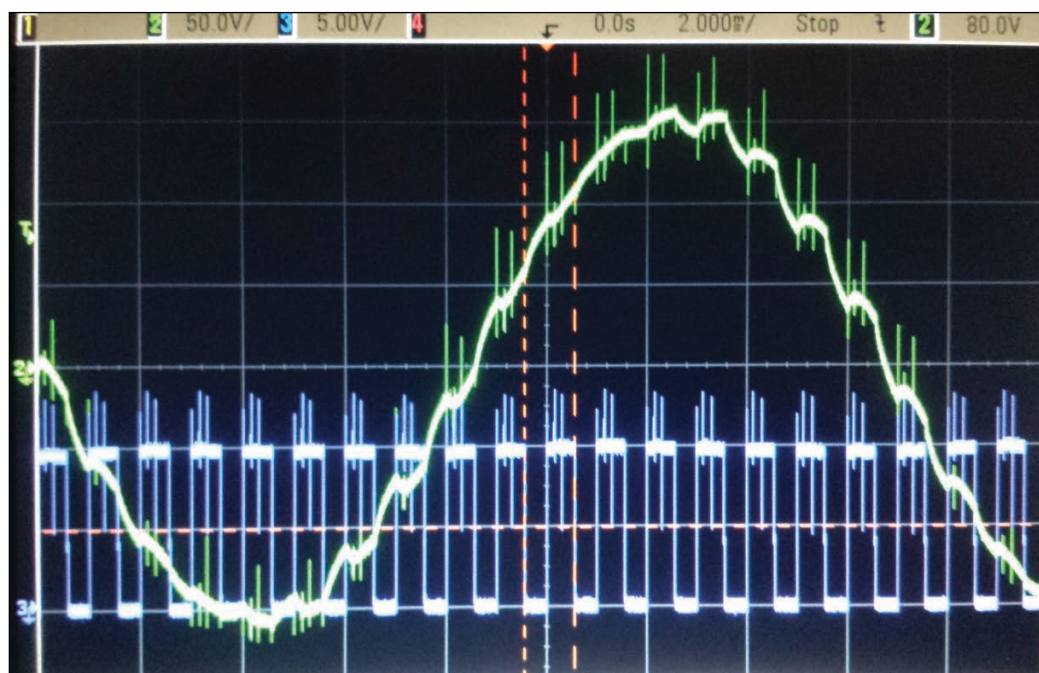
- ◆ 1200W Low AC line, 1500W High AC line
- ◆ Wide Range of outputs from 2.8V up to 60V
- ◆ 10.6 x 6.9 x 2.5 inch footprint
- ◆ High Efficiency
- ◆ Available for quick delivery
- ◆ 7 year warranty

For more information on how TDK-Lambda can help you power your unique applications, visit our web site at
www.us.tdk-lambda.com
or call 1-800-LAMBDA-4



FIGURE 7

1 kHz drive frequency distortion in generator output fixed



switched off. All conduction waste heat was easily wicked away by the aluminum sheet to which the point-to-point circuits were mounted (**Figure 6**).

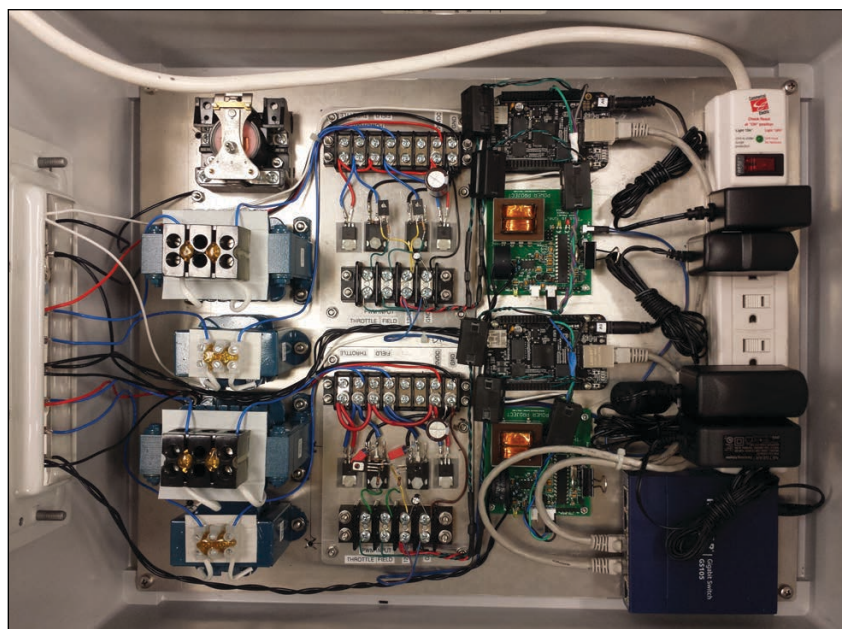
Working essentially as a buck converter using the inductive windings on the machine to smooth current from the DC power supply, several additions helped smooth the generator output and reduce switching harmonics. A snubber circuit was added across the MOSFET output to absorb switching energy from inductive spikes, and DC reactors were placed in line with the

DC motor and generator field coil drivers to increase the effective inductance of our buck converter circuit. Body diodes of some spare insulated gate bipolar transistors (IGBTs) were used as freewheel diodes and prevented the MOSFETs from releasing their magic smoke. Despite these additions, a 1 kHz drive frequency (blue) still caused spikes in the generator output (green) shown in **Figure 7**. Non-ideal artifacts of the rapid charging and discharging of the DC reactor magnetic field were also evident in the distorted 60 Hz sinusoid, so the drive frequency was increased to 5 kHz.

Before using an opto-isolator as a buffer between the high-power driver circuitry and the Linux computer, high-voltage signals propagated back through the circuit and caused mysterious problems. Because we used the BeagleBone PWM hardware instead of a robust external PWM generator, energizing the genset caused the computer to go into a “kernel panic” state that halted all operations until it was reset. We spent several hours looking for errors in the I²C libraries on the MCU and the BeagleBone. Ultimately the use of opto-isolators electrically separated the controller from the power electronics, and snap-on ferrite chokes around the braided PWM signal lines prevented the electromagnetic interference (EMI) from disrupting the computer’s operation.

MASTER CONTROLLER

Because our objective was to maintain the system output at exactly 120 V_{RMS} and 60 Hz, our original objective intentionally

**FIGURE 8**

Two genset control systems

required iso-iso paralleling, which was a lofty goal. An iso-iso configuration posed instability challenges, with each individual genset control system fighting to manipulate the bus. Unfortunately, we ran out of time, and the software was never completed, but we spent our remaining time conceptualizing a master controller that would control an arbitrary number of gensets and balance a theoretically unlimited total power output.

The framework for the master controller was implemented in C on a laptop that communicated with the individual genset controllers via a TCP socket connection. After registering themselves with the master as online and available, the genset controllers entered paralleled mode, where they initiated connections with the master controller and sent periodic updates containing their RMS voltage, real power, reactive power and frequency.

Two methods were examined for master control: (1) cut out the genset controller and use the master controller to decide on genset-level control response; or (2) use the genset control systems for instant droop correction and use the master controller for long-term cooperative load balancing.

On the one hand, the first method would produce the best stability, because the master controller could respond to disturbances using the big picture to make decisions; however, pushing time-sensitive commands to a large number of gensets would be slow and would create a single point of failure. On the other hand, utilizing the genset controllers to respond instantly in droop mode would not fully return the bus frequency and voltage to the setpoint, but it would give the best overall stability even if the master controller failed. Long-term voltage, frequency and power adjustments could then be made by the master controller raising or lowering the droop curves for iso-iso control.

While fine tuning the long-term voltage and frequency, the master controller would simultaneously redistribute real and reactive power as necessary for maximum efficiency. Additionally, because the processing demand on the master controller would increase linearly with the control of more gensets, we considered this to be an optimal use of computing resources. Load distribution on the bus could be further tailored for maximum efficiency by identifying high efficiency “base load” gensets for normal conditions and lower efficiency “peaking” gensets when demand is high. Although it was never implemented, we were confident that our implementation would work well in island operation, especially with further control system tuning.

RESULTS AND DISCUSSION

Although we never finished the master controller, we successfully implemented a genset control system that exceeded our design requirements for steady-state operation and transient load recovery. Steady-state voltage and frequency output were maintained at $120 \pm 4 \text{ V}_{\text{RMS}}$ and $60 \pm 0.5 \text{ Hz}$. Two completed genset control systems were placed in an enclosure for operation and display (**Figure 8**).

Transient load response was examined by analyzing the recovery time for a stand-alone genset. **Figure 9** shows the system step response to the switching-on of three “Y”-connected 60 W light bulbs—180 W

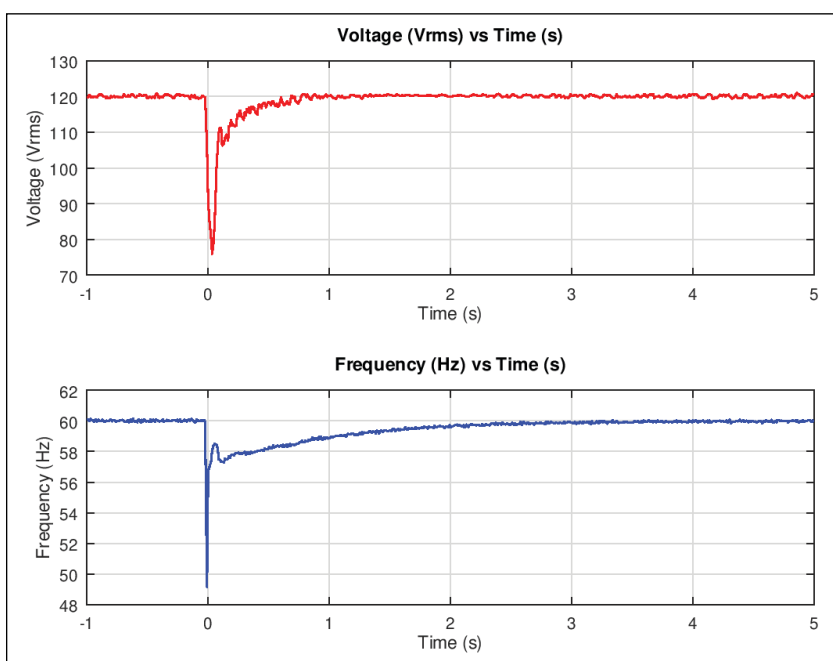


FIGURE 9
Stand-alone genset 180 W loaded

ABOUT THE AUTHORS

Kent Altobelli is from Los Angeles, California, and received his Bachelor of Science in Electrical Engineering from the U. S. Coast Guard Academy. After graduating as a Commissioned Officer in the U. S. Coast Guard, he served as a Deck Watch Officer on the Coast Guard Cutter ALDER out of Duluth, Minnesota. Kent is presently stationed in Miami, Florida supervising electronics support for Coast Guard units from South Carolina down to Puerto Rico.

Caleb Stewart is from Grandview, Texas, and received his Bachelor of Science in Electrical Engineering from the U.S. Coast Guard Academy. After graduation, he first served as a Deck Watch Officer onboard the Coast Guard Cutter BERTHOLF out of Alameda, California. Caleb is now the Infrastructure Support Branch Chief at the Coast Guard Cyber Security Operations Center. Caleb is an Associate Certified Information Systems Security Professional as well as an Offensive Security Certified Professional and is currently pursuing a Masters in Computer Science from the Georgia Institute of Technology.

is 55% of the genset power capacity. The system output dropped to roughly $75 V_{RMS}$ at 49 Hz, but the control system was capable of returning the output to $120 V_{RMS}$ at 60 Hz in about 2.5 seconds. **Figure 10** shows the step response when the 180 W load was removed and the output jumped to $145 V_{RMS}$ at 63 Hz. The output returned to steady state in about 3.5 seconds. The plots illustrate the control system's less aggressive frequency correction to address the non-ideal

dependence between frequency and voltage mentioned in the "Genset Controller" section.

Next, the control system was used to hold a genset in isochronous mode while it was paralleled against a second genset in droop mode. In paralleled island operation, a 180 W load had a much smaller effect. Although the tempting explanation is that because the overall system capacity is now 660 W, the droop genset only changes its output power when the bus frequency differs from 60 Hz. Therefore at 60 Hz the system capacity is limited to 330 W from the iso generator plus whatever power the droop generator is set to contribute.

The true benefit of this style of paralleling is that the droop genset improves the inertial response of the bus to transient loading, and also assists the iso genset by temporarily contributing more or less power to "bounce back" from large power fluctuations. **Figure 11** and **Figure 12** show that when the system was loaded and unloaded, the output only dropped to $103 V_{RMS}$ at 54 Hz or rose to $150 V_{RMS}$ at 62 Hz, respectively. Working as a swing generator, the controlled genset again responded immediately and met our objective by returning to steady state in about 3 seconds in both cases.

To demonstrate the significant increase in performance with our actively controlled genset, an 80 W induction motor (highly reactive load) was started and stopped using the same iso-droop configuration. During both tests, the 180 W resistive load remained continuously energized to provide a real component to the genset power output. As shown in **Figure 13**, with the control system, the iso-droop paralleled gensets temporary succumbed to the 90 V drop caused by the highly inductive load, but were able to spin up the motor's inertia and fully recover in about 2.5 seconds (though a small voltage integral wind-up was visible in the aftermath). **Figure 14** demonstrated the unloading of the control system, where the output rose by 30 V due to the removal of the inductive load; the system returned to steady-state in about 3.5 seconds.

While both induction motor tests show the performance of our voltage control system, they also highlight the dependence between the frequency and voltage control systems, since the majority of the disturbance was reactive. Ideally, the two exercises should have had a small impact on the frequency, aside from initially consuming real power to spin up the induction motor.

CONCLUSION

Successful hardware and software development of our genset control system demonstrated that a low-cost solution is

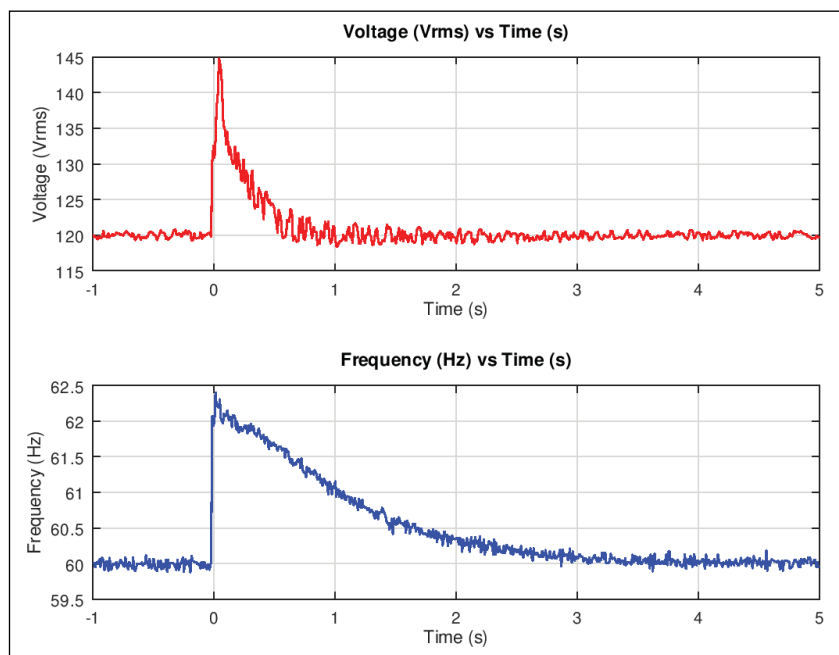


FIGURE 10

Stand-alone genset 180 W unloaded

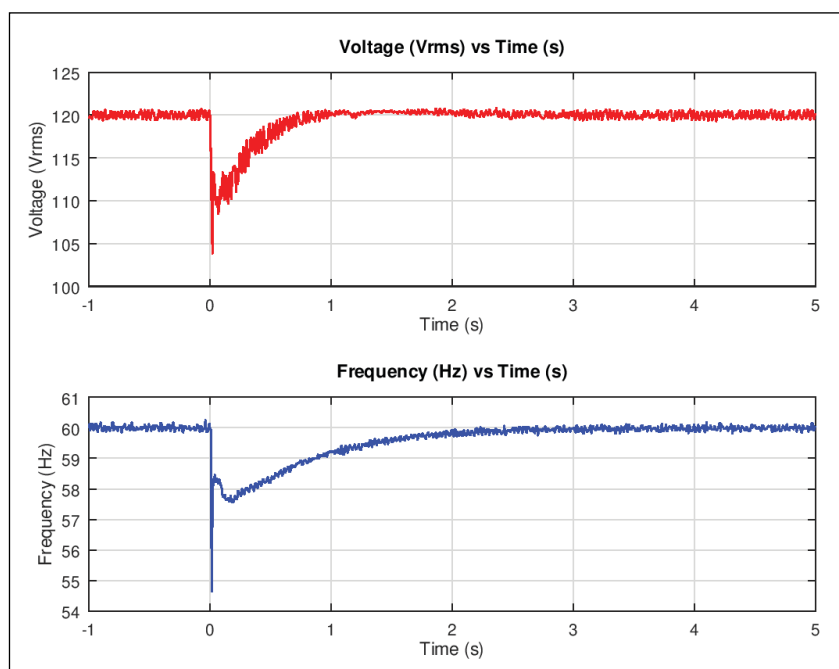
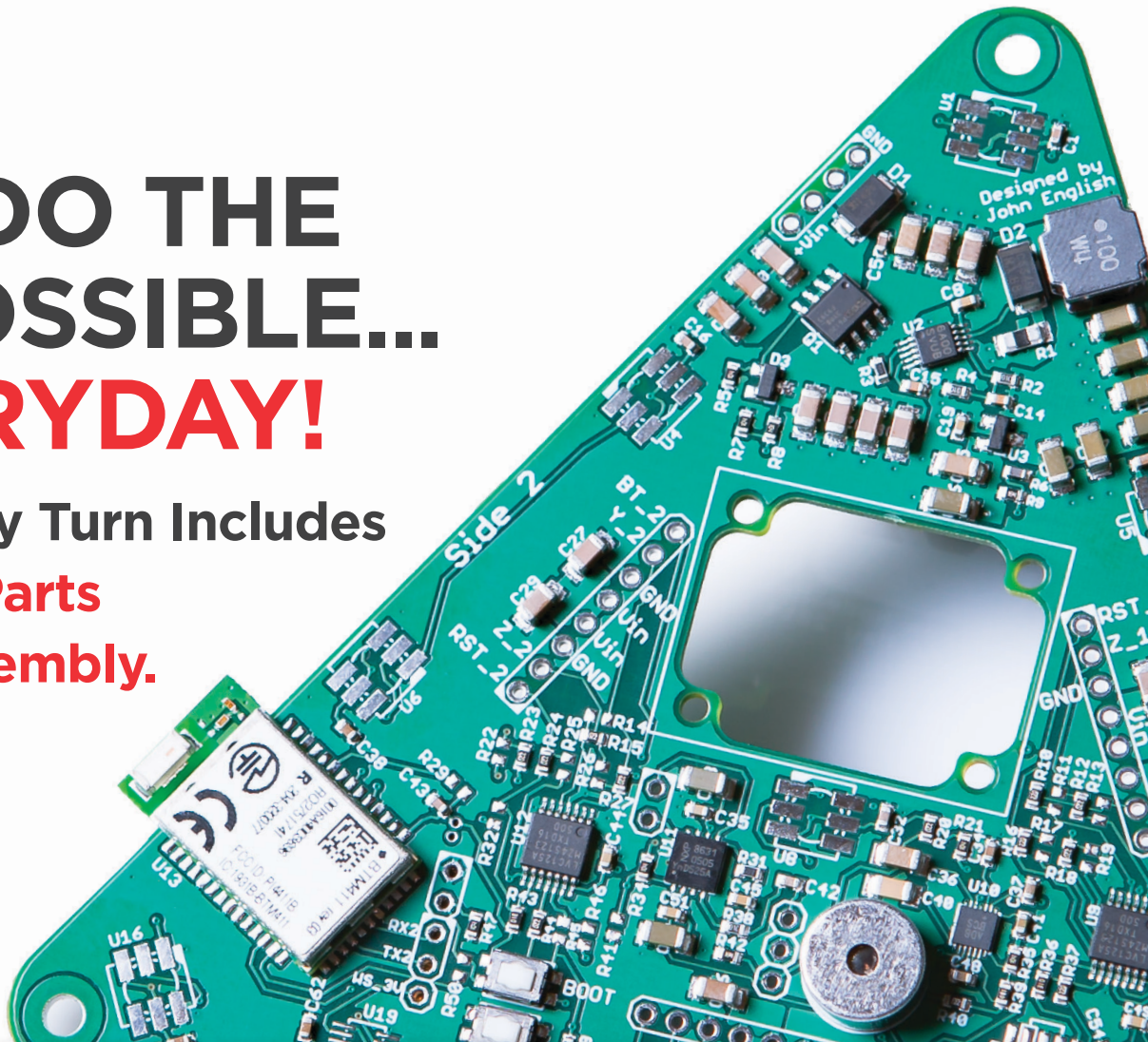


FIGURE 11

Iso-droop paralleled gensets 180 W loaded

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

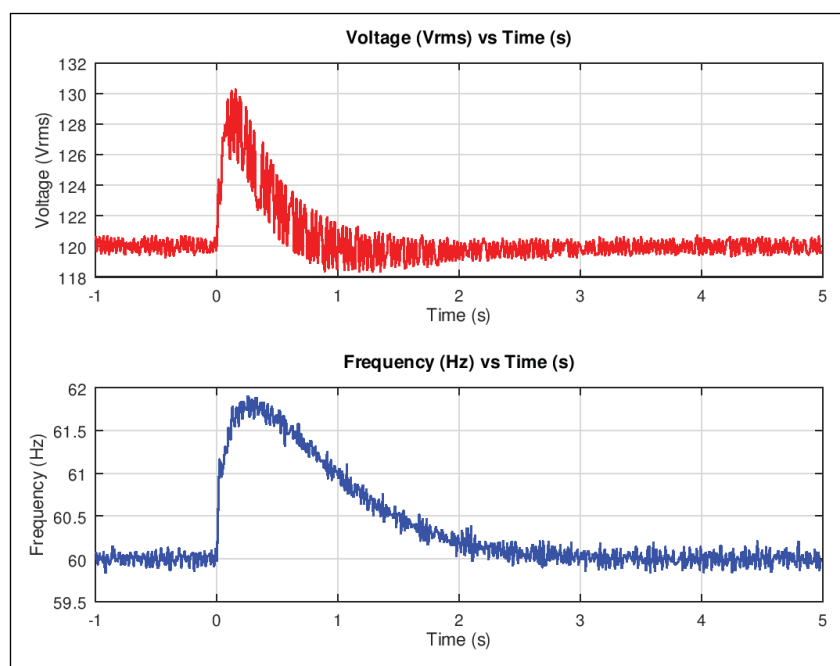


FIGURE 12

Iso-droop paralleled gensets 180 W unloaded

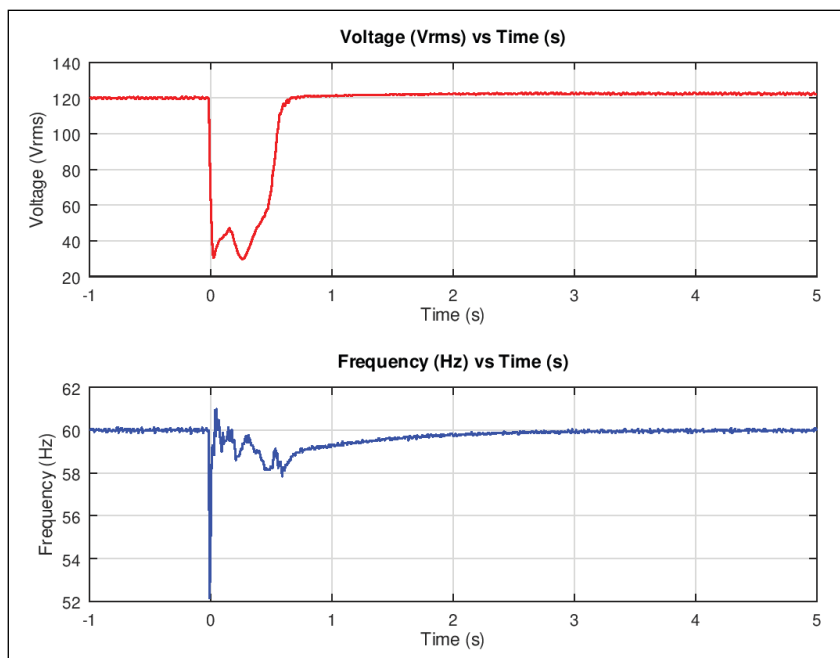


FIGURE 13

Iso-droop genset 80 W induction motor start

attainable for managing gensets. Building and testing individual modules for any project involves component-level troubleshooting and further troubleshooting when integrating the various pieces to form the complete system. Despite the challenges and our imperfect PI controllers, the project demonstrated that an industrial control system can be designed and built completely from scratch and still have acceptable performance.

We surpassed our objective to keep steady-state output within $\pm V_{RMS}$ and ± 0.5 Hz, and nearly met our objective to respond to transient loads by returning to steady-state operation within 3 seconds. Even though the master controller was not completed, our discussions culminated in exciting realizations for stable iso-iso bus control schemes and provided a deeper appreciation for power systems in the industry and on board US Coast Guard cutters.

Properly identifying the mechanical and electrical characteristics of the genset would have greatly improved system response and provided stability during the testing of the master controller. For example, feed forward compensation might be able to decouple voltage and frequency response, or a tailored Multiple Input Multiple Output (MIMO) control system could potentially use this coupling to improve system response time and stability.

To completely automate the paralleling process, a “smart breaker” could also be developed to monitor the bus and oncoming genset, to determine when to make the connection. By measuring the differential voltage and beat frequency between both the “A” phase of the bus and oncoming genset, a device could detect compliance with the first three paralleling conditions. An additional device could be used on the “B” phase to validate the first three conditions and verify phase sequence—or this could be assumed for a properly installed circuit breaker.


Throughout the project and especially when developing the smart breaker, network vulnerability needs to be addressed to ensure machinery is protected from the physical damage that could result from malicious commands. Although the system would exist behind enterprise-level security appliances, in the event of network intrusion, machinery should be capable of recognizing legitimate authority or be smart enough not to cause damage directly.

For example, the gensets should know that the master would never command the system frequency be raised to 100 Hz, and the smart breaker should never allow itself to be overridden to connect the genset when it is 180 degrees out of phase with the bus. As a real example of the importance of security, the Ukraine power grid was disrupted in 2015 by phishing and malware attacks that caused extensive physical damage.

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

RESOURCES

Microchip Technology | www.microchip.com

Regarding security in general, power distribution in the United States relies on synchrophasors, which use the low signal power global positioning system (GPS) constellation as a clock source to synchronize power grid waveforms across the country. As discussed in the Master Controller design section, gensets and the power grid can be vulnerable to instabilities, demonstrated by the cascading failures causing the 2003 Northeast Blackout, and systems need to be designed carefully to ensure robust operation. 

Authors' Note: We thank Aaron Dahlen, LCDR, US Coast Guard retired, for advising us on the project, providing insight on unexplainable measurements and inspiring us to explore the nuances of three-phase systems. We thank ETC Christopher Gosvener for meticulously maintaining the laboratory facilities, supplying all equipment needs and ensuing electrical safety standards were followed during the development and testing of our project.

This project was completed to fulfill BSEE requirements for the US Coast Guard Academy. Views expressed here do not necessarily represent the views of the United States, the Dept. of Homeland Security, or the United States Coast Guard.

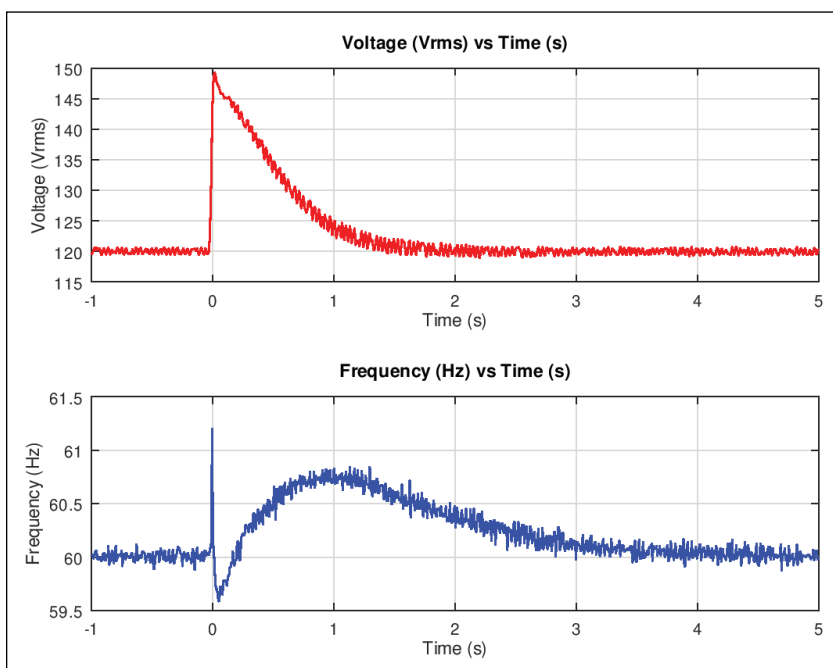


FIGURE 14

Iso-Droop 80 W induction motor stop

With the right tools designing a microprocessor can be easy.

Okay, maybe not easy, but certainly less complicated. Monte Dalrymple has taken his years of experience designing embedded architecture and microprocessors and compiled his knowledge into one comprehensive guide to processor design in the real world.

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

cc-webshop.com



Verilog HDL

Thermal Management in Machine Learning

Beat the Heat

Artificial intelligence and machine learning continue to move toward center stage. But the powerful processing they require is tied to high power dissipation that results in a lot of heat to manage. In his article, Tom Gregory from 6SigmaET explores the alternatives available today with a special look at cooling Google's Tensor Processor Unit 3.0 (TPUv3) which was designed with machine learning in mind.

By
Tom Gregory,
6SigmaET

FIGURE 1

Google's Tensor Processor Unit 3.0 (TPUv3) is designed specifically with the future of machine learning in mind. The TPU is a custom ASIC, tailored specifically for TensorFlow, Google's open source software library for machine learning and AI applications.

For much of the electronics community, 2018 was the year that artificial intelligence (AI)—and in particular machine learning—became a reality. Used in everything from retail recommendations to driverless cars, machine learning represents the ongoing effort to get computers to solve problems without being explicitly programmed to do so.

While both machine learning and the wider notion of artificial intelligence come with revolutionary new implications for the technology sector, they also require significant investment in new forms of electronics, silicon and customized hardware. But just why do we need such an extensive ground-up redesign? And why shouldn't AI firms simply keep building on the processing and acceleration technologies that are already in place?

BUILDING A BRAIN

For most of those working in AI, the end goal is to create artificial general intelligence (AGI); "thinking machines" that assess, learn, and handle nuance and inference in a way that replicates the thought processes of the human brain. Based on the current design and architecture of electronics however, this simply isn't possible.

As it stands, the vast majority of current-generation computing devices are based on the same Von Neumann architecture—a beautifully simple, but fundamentally limited way of organizing information. In this structure programs and data are held in memory, separate from the processor, with data having to move between the two in order for operations to be completed. The limitations of this structure result in what has become known as the "Von Neumann bottleneck," in which latency is unavoidable.

Unlike today's computing devices, the human brain does not separate memory from processing. Biology makes no distinction between the two, with every neuron and every synapse storing and computing information simultaneously.

While even the largest investments in AI are nowhere near recreating such a system, there is a growing demand to rethink the Von Neumann architecture and to overcome the natural bottleneck of the two-part memory/processing system. It is here that new hardware developments are proving so vital. While plenty of AI companies have had significant success simply throwing more processing power and evermore GPUs at the problem, the reality is that AI and machine

learning will never reach their full potential until a new more “biological” hardware is developed from the ground-up.

THE RACE FOR HARDWARE

This demand for new and increasingly advanced hardware has become known as the “Cambrian Explosion”—an apt reference to the most important evolutionary event in the history of life. This colossal growth in the AI market has resulted in a huge variety of new electronics, as well as an increasing number of high-end investments being made in any start-up that can offer a solution to the ever expanding disconnect between existing hardware outputs and the massive potential of machine learning technology.

Already in 2018, we saw several such investments. In March, AI chip start-up SambaNova Systems received \$56 million in funding for its new GPU replacement, designed specifically with AI operations in mind. More recently, it’s competitor Mythic received a similar \$40m investment to continue its efforts to replace the traditional GPU. And it’s not just start-ups that are getting in on the action. Tech giants such as Microsoft, Google, Nvidia and IBM are all looking for their own ‘killer app’ solution to the AI hardware problem.

While each of these different companies has its own particular solution – often attacking the problem from completely different angles – one of the most common pieces of hardware being developed is an accelerated GPU.

In traditional computing environments, CPUs have been used for the bulk of processing, with GPUs being added to ramp things up where required (such as for rendering videos or animation). However, in the age of machine learning, GPUs still aren’t enough. What’s needed now is a new, more

powerful, and more streamlined processing unit which can undertake the heavy lifting needed for machine learning – a unit that can analyze large data sets, recognize patterns and draw meaningful conclusions.

TENSOR PROCESSOR UNIT

The recently released Tensor Processor Unit 3.0 (TPUv3) is Google’s latest foray into the AI hardware market. Designed specifically with the future of machine learning in mind, the TPU is a custom ASIC, tailored specifically for TensorFlow, Google’s open source software library for machine learning and AI applications (**Figure 1**).

Unlike GPUs, which act as a processor in their own right, Google’s TPU is a coprocessor—shifting all code execution to the CPU in order to free up the TPU for a stream of machine learning-based microoperations. In purely practical terms, TPUs are designed to be significantly cheaper and to (theoretically) use less power than GPUs, despite playing a pivotal role in some pretty hefty machine learning calculations, predictions and processes. The only question is, does Google’s TPU really achieve what it claims?

While positioned as a game-changing development in the AI space, the new TPUv3.0 still faces many of the same challenges as competitor products offered by Amazon and Nvidia—in particular, the potential for thermal complications.

As with so much of the hardware developed specifically for the machine learning market, Google’s TPUv3 offers a colossal amount of processing power. In fact, according to Google CEO Sundar Pichai, the new TPU will be eight times more powerful than any of Google’s previous efforts in this area.

From an AI standpoint this is hugely

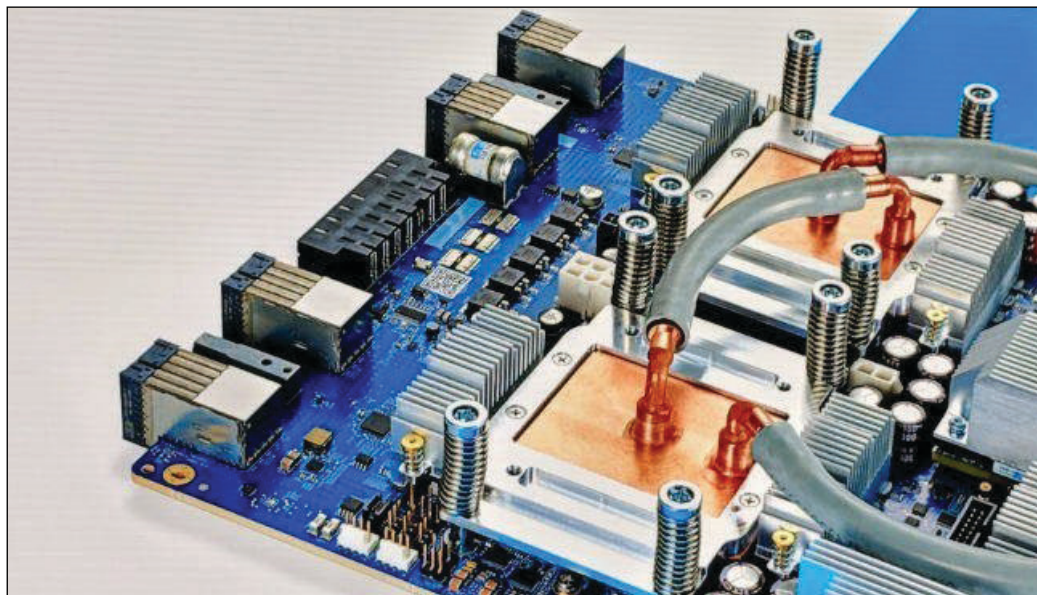
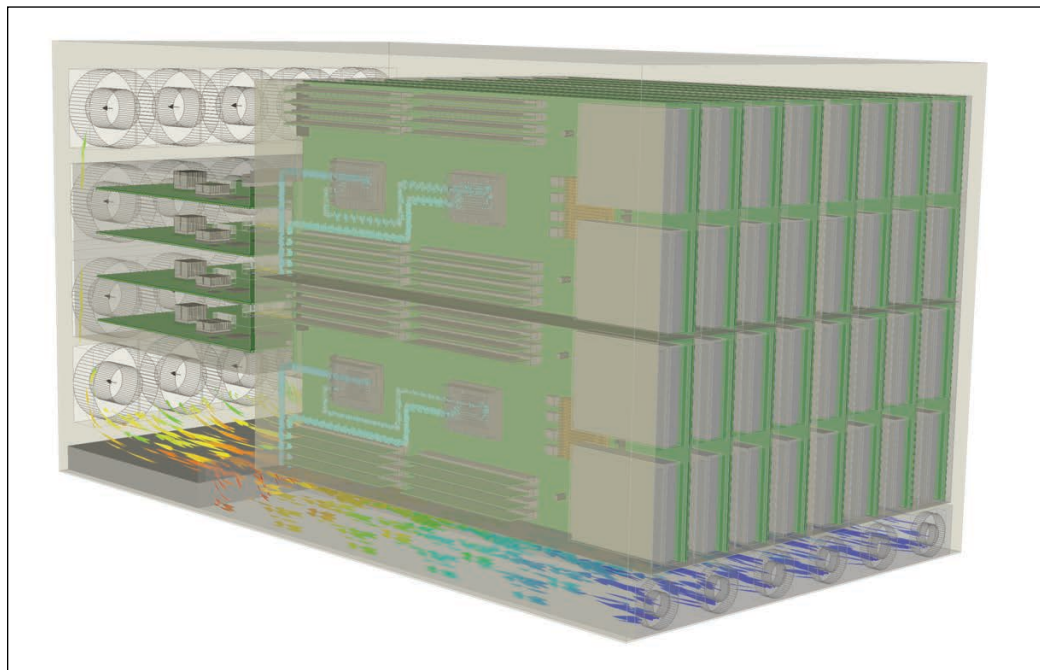


FIGURE 2
Google TPU 3.0 with liquid cooling

FIGURE 3

This graphic shows streamlines within a blade chassis with hybrid liquid and fan cooling system modelled using 6SigmaET.



beneficial, with the process of machine learning relying on the ability to crunch huge volumes of data instantaneously in order to make self-determined decisions. From a thermal perspective however, this dramatic increase in processing represents a minefield of potential complications – with increased power meaning more heat is generated throughout the device. This accumulation of heat could potentially impact performance, ultimately risking the reliability and longevity of the TPU.

In a market where reliability is essential, and buyers have little room for system downtime, this issue could prove the deciding factor in which hardware manufacturer ultimately claims ownership of the AI space.

Given these high stakes, Google has clearly invested significant time in maximizing the thermal design of its TPUv3. Unlike the company's previous tensor processing units, the TPUv3 is the first to bring liquid cooling to the chip—with coolant being delivered to a cold plate sitting atop each TPUv3 ASIC chip (**Figure 2**). According to Pichai, this will be the first time ever that Google has needed to incorporate a form of liquid cooling into its data centers.

KEEPING IT COOL

While the addition of liquid cooling technology has been positioned as a new innovation for the industry, the reality is that high powered electronics running in rugged environments have been using similar heat dissipation systems for some time.

When preparing equipment for these environments, engineers are faced with the challenge of working with limited cooling resources. As a result, they must find clever ways of dissipating heat away from the critical components. In this context, carefully designed hybrid liquid and air-cooling systems have proved vital in ensuring that servers and other critical electronics systems function reliably.

As an example, 6SigmaET thermal simulation software has been used to model liquid cooling systems for servers by The University of Texas at Arlington for their research. One significant challenge the research faced was the components other than the main processing chip within a server—like the DIMMs, PCH, HDD and other heat generating components that are not directly cooled by a liquid cooling loop. Hence, the combination of warm water and recirculated air was used in the research to cool the server to keep the critical temperatures within the recommended range.

While such liquid cooling systems are extremely effective, they should not necessarily be used as the go-to solution for thermal management. With those in the machine learning space looking to optimize efficiency—both in terms of energy and cost—it's vital that designers minimize thermal

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

RESOURCES

6SigmaET | www.6sigmaet.info

Future Facilities | www.futurefacilities.com

Google Cloud | cloud.google.com

issues across their entire designs, and do not rely on the sledgehammer approach of installing a liquid cooling system just because the option is available.

For some of the most powerful chips, such as the Google TPuv3, it may be that liquid cooling is the most viable solution. In future however, as ever more investment is placed in machine learning hardware, engineers should not grow complacent when it comes to exploring different thermal management solutions. Liquid cooling may be sufficient to dissipate heat build-up in the most high-powered components currently available. However, this may not always be the case in the future. It may be more efficient to strive for designs that do not risk such accumulations of heat in the first place.

ALL-ENCOMPASSING SOLUTION

If AI hardware manufacturers are truly going to overcome the thermal complications associated with their increasingly powerful designs, they must take every opportunity to optimize thermal management at every stage of the design process.

At the chip level, appropriate materials for substrates, bonding, die attaches and interface materials need to be selected. At the system level there are equally important decisions to be made regarding PCB materials, heat sinks and where to incorporate liquid cooling or thermoelectric coolers.

The more robust materials used in high power electronics also brings their own challenges. Compared to typical FR4 PCBs, materials like ceramic or copper have high thermal conductivity, which can be advantageous in thermal management, but at the same time these materials can also add significant cost and weight to a design if not used optimally.

While these may seem like minor considerations, it is the businesses that maximize their use of innovative solutions within their design constraints and build thermal considerations into the fabric of their designs that will be most effective at minimizing energy waste and mitigating unnecessary risk and costs due to component failure.


UP FRONT CONSIDERATION

According to 6SigmaET's research, which incorporates data from over 350 professional engineers, 75% don't test the thermal performance of their designs until late in the design process. And 56% don't run these tests until after the first prototype has been developed, while 27% wait until after a design is complete before even considering thermal complications.

Instead of relying on physical prototypes, which are expensive and time consuming to produce, more and more of today's engineers are choosing to test the thermal qualities of their designs virtually—in the form of thermal simulations.

By creating a thermal simulation model in advance, AI hardware engineers can test their designs using a wide variety of different materials and configurations—for example, switching apart from copper to aluminum at the click of a button. Simulation also enables designs to be tested in a massive array of different environments, temperatures and operating mode scenarios (**Figure 3**). This will not only help to identify potential inefficiencies, but also reduces the need for multiple real-world prototypes.

Through the early-stage incorporation of thermal simulation into the design process, it is becoming increasingly easy for engineers to precisely understand the unique thermal challenges facing AI hardware. This means that thermal considerations can be dealt with far earlier, enabling the thermal performance of TPUs and related AI hardware to be fully optimized and reducing the risk of expensive late-stage 'fixes' and unnecessary over-engineering.

With numerous industry titans, and hundreds of electronics start-ups all racing to be the first to develop truly effective and efficient AI hardware, those that fail to account for the 'little things' will quickly fall behind. When working in such a high-precision field, every wasted second or lost watt represents a significant burden on the effectiveness of the resulting system. Inevitably, those firms that produce the most elegant, efficient and—ultimately—the most reliable products, will be those that claim ownership of the space, and win their place as the leaders in the AI hardware market. 



ABOUT THE AUTHOR

Tom Gregory is a Product Manager at Future Facilities Ltd. He has been working at the company for five years and is responsible for marketing and supporting customers on the 6SigmaET side of the business. Founded in 2004, Future Facilities Ltd. is a global company that provides thermal simulation software and consultancy for the electronics and data-center industries. Driven by a team of the world's leading thermal simulation engineers and scientists, Future Facilities launched its tool for the electronics design market, 6SigmaET, in 2009.

MCUs Serve Up Solutions for Car Infotainment

Dashboard Dazzle

As automotive dashboard displays get more sophisticated, information and entertainment are merging into so-called infotainment systems. The new systems are driving a need for powerful MCU solutions that support the connectivity, computing and interfacing requirements particular to these designs.

SPECIAL FEATURE

FIGURE 1

The Cypress Wi-Fi and Bluetooth combo solution uses Real Simultaneous Dual Band (RSDB) technology so that Apple CarPlay (shown) and Android Auto can operate concurrently without degradation caused by switching back and forth between bands.

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

Microcontroller (MCU) vendors have a rich legacy of providing key technologies for nearly every aspect of an automobile's electronics—everything from the powertrain to the braking system to dashboard displays. In recent years, they've taken on a new set of challenges as demands rise for ever more sophisticated "infotainment" systems. Advanced touchscreen, processing, networking, voice recognition and more are parts of these subsystems tasked with providing drivers with information and entertainment suited to today's demands—demands that must rival or exceed what's possible in a modern smartphone or tablet. And, as driverless cars inch toward mainstream reality, that hunger for rich infotainment functionality will only increase.

In order to meet those system design needs, MCU vendors are keeping pace with highly integrated chip-level solutions and embedded software tailored specifically to address various aspects of the automotive infotainment challenge. Over the past 12 months, MCU companies have announced products aimed at everything from advanced dashboard graphics to connectivity solutions to security technologies. At the same time, many have

announced milestone design wins that illustrate their engagement with this dynamic sub-segment of automotive system development.

SMARTPHONE SUPPORT

Exemplifying these trends, in July Cypress Semiconductor announced that Pioneer integrated Cypress' Wi-Fi and Bluetooth Combo solution into its flagship in-dash navigation AV receiver. The solution enables passengers to display and use their smartphone's apps on the receiver's screen via Apple CarPlay (**Figure 1**) or Android Auto, which provide the ability to use smartphone voice recognition to search for information or respond to text messages. The Cypress Wi-Fi and Bluetooth combo solution uses Real Simultaneous Dual Band (RSDB) technology so that Apple CarPlay and Android Auto can operate concurrently without degradation caused by switching back and forth between bands.

The Pioneer AVH-W8400NEX receiver uses Cypress' CYW89359 combo solution, which includes an advanced coexistence engine that enables optimal performance for dual-band 2.4- and 5-GHz 802.11ac Wi-Fi and dual-mode Bluetooth/Bluetooth Low Energy (BLE) simultaneously for advanced multimedia

experiences. The CYW89359's RSDB architecture enables two unique data streams to run at full throughput simultaneously by integrating two complete Wi-Fi subsystems into a single chip. The CYW89359 is fully automotive qualified with AECQ-100 grade-3 validation and is being designed in by numerous top-tier car OEMs and automotive suppliers as a full in-vehicle connectivity solution, supporting infotainment and telematics applications such as smartphone screen-mirroring, content streaming and Bluetooth voice connectivity in car kits.

In October, Cypress announced another infotainment-related design win with Yazaki North America implementing Cypress' instrument cluster solution to drive the advanced graphics in Yazaki's instrument cluster for a leading American car manufacturer. According to Cypress, Yazaki selected the solution based on its unique offering of five chips that combine to drive dual displays and provide instant-on memory performance with automotive-grade, ASIL-B safety compliance. The Cypress solution is based on a Traveo MCU, along with two high-bandwidth HyperBus memories in a multi-chip package (MCP), an analog power management IC (PMIC) for safe electrical operation, and a PSoC MCU for system management support. The Traveo devices in the Yazaki instrument cluster were the industry's first 3D-capable Arm Cortex-R5 cluster MCUs.

VIRTUALIZATION EMBRACED

The complexity of automotive infotainment systems has pushed system developers to embrace advanced operating system approaches such as virtualization. Feeding those needs, last June Renesas Electronics rolled out its "R-Car virtualization support package" designed to enable easier development of hypervisors for the Renesas R-Car automotive system-on-chip (SoC). The R-Car virtualization support package includes, at no charge, both the R-Car hypervisor development guide document and sample software for use as reference in such development for software vendors who develop the embedded hypervisors that are required for integrated cockpits and connected car applications.

A hypervisor is a virtualization operating system (OS) that allows multiple guest OSs—such as Linux, Android and various real-time OSs (RTOS)—to run completely independently on a single chip. Renesas announced the R-Car hypervisor in April of 2017 and the new R-Car

virtualization Support Package was developed to help software vendors accelerate their development of R-Car hypervisors.

The company's third-generation R-Car SoCs were designed assuming that they would be used with a hypervisor. The Arm CPU cores, graphics cores, video/audio IP and other functions include virtualization functions. Originally, for software vendors to make use of these functions, they would have had to understand both the R-Car hardware manuals and the R-Car virtualization functions and start by looking into how to implement a hypervisor. Now, by following development guides in the R-Car virtualization support package, not only can software vendors easily take advantage of these functions, they will be able to take full advantage of the advanced features of R-Car. Also, by providing sample software that can be used as a reference, this package supports rapid development.

Technology partnerships have been playing a key role in automotive infotainment trends. Along just those lines, in September Renesas and OpenSynergy, a supplier of automotive hypervisors, announced that the Renesas' SoC R-Car H3 and OpenSynergy's COQOS Hypervisor SDK were adopted on Parrot Faurecia's automotive safe multi-display cockpit. The latest version of Android is the guest OS of the COQOS Hypervisor, which executes both the instrument cluster functionality, including safety-relevant display elements based on Linux, and the Android-based in-vehicle infotainment (IVI) on a single R-Car H3 SoC chip (**Figure 2**). The COQOS

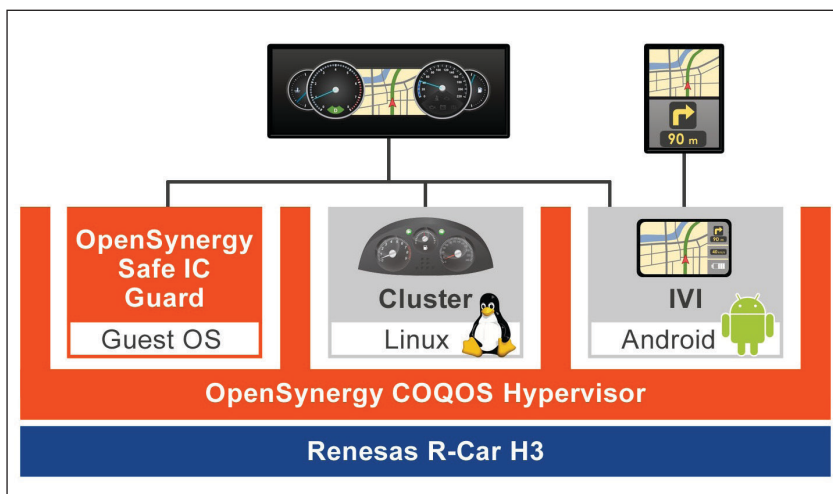


FIGURE 2

With Android as the guest OS of the COQOS Hypervisor, it executes both the instrument cluster functionality, including safety-relevant display elements based on Linux, and the Android-based in-vehicle infotainment (IVI) on a single R-Car H3 SoC chip.

FIGURE 3

The maXTouch family of single-chip touchscreen controllers is designed for screens up to 20 inches in size, and supports up to 3,000 touch sensing nodes. The devices even support multiple finger touches through thick gloves and in the presence of moisture.



Hypervisor SDK shares the R-Car H3 GPU with Android and Linux allowing applications to be presented on multiple displays, realizing a powerful and flexible cockpit system.

According to OpenSynergy's CEO Stefaan Sonck Thiebaut, the COQOS Hypervisor SDK takes full advantage of the hardware and software virtualization extensions provided by Renesas. The OpenSynergy solution includes key features, such as shared display, which allows several virtual machines to use multiple displays flexibly and safely. The R-Car H3 GPU and video/audio IP incorporates virtualization functions, making virtualization by the hypervisor possible and allowing for multiple OSs to operate independently and safely. OpenSynergy's COQOS Hypervisor SDK is built around a safe and efficient hypervisor that can run software from multipurpose OSs such as Linux or Android, RTOS and AUTOSAR-compliant software simultaneously on one SoC.

LARGE TOUCHSCREEN SUPPORT

As the content provided by automotive infotainment systems gets more sophisticated, so too must the displays and user interface technologies that interact with that content. With that in mind, MCU vendors are offering more advanced touchscreen control solutions. Dashboard screens have unique design challenges. Screens in automobiles need to meet stringent head impact and vibration tests. That means thicker cover lenses that potentially impact the touch interface

performance. Meanwhile, as screens get larger, they are also more likely to interfere with other frequencies such as AM radio and car access systems. All of these factors become a major challenge in the design of modern automotive capacitive touch systems.

Along just those lines, Microchip in December announced its maXTouch family of single-chip touchscreen controllers designed to address these issues for screens up to 20 inches in size (**Figure 3**). The MXT2912TD-A, with nearly 3,000 touch sensing nodes, and MXT2113TD-A, supporting more than 2,000 nodes, bring consumers the touchscreen user experience they expect in vehicles. These new devices build upon Microchip's existing maXTouch touchscreen technology that is widely adopted by manufacturers worldwide. Microchip's latest solutions offer superior signal-to-noise capability to address the requirements of thick lenses, even supporting multiple finger touches through thick gloves and in the presence of moisture.

As automakers use screens to replace mechanical switches on the dash for sleeker interior designs, safe and reliable operation becomes even more critical. The MXT2912TD and MXT2113TD devices incorporate self- and sensor-diagnostic functions, which constantly monitor the integrity of the touch system. These smart diagnostic features support the Automotive Safety Integrity Level (ASIL) classification index as defined by the ISO 26262 Functional Safety Specification for Passenger Vehicles.

The new devices feature technology that enables adaptive touch utilizing self-capacitance and mutual-capacitance measurements, so all touches are recognized and false touch detections are avoided. They also feature Microchip's proprietary new signal shaping technology that significantly lowers emissions to help large touchscreens using maXTouch controllers meet CISPR-25 Level 5 requirements for electromagnetic interference (EMI) in automobiles. The new touch controllers also meet automotive temperature grade 3 (-40°C to +85°C) and grade 2 (-40°C to +105°C) operating ranges and are AEC-Q100 qualified.

3D GESTURE CONTROL

Aside from the touchscreen display side of automotive infotainment, Microchip for its part has also put its efforts toward innovations in 3D human interface technology. With that in mind, in July the company announced a new 3D gesture recognition controller that offers the lowest system cost in the automotive industry, providing a durable single-chip solution for advanced automotive HMI designs, according to Microchip. The MGC3140 joins the company's family of easy-to-use 3D gesture controllers as the first qualified for automotive use (**Figure 4**).

Suited for a range for applications that limit driver distraction and add convenience to vehicles, Microchip's new capacitive technology-based air gesture controller is ideal for navigating infotainment systems, sun shade operation, interior lighting and other applications. The technology also supports the opening of foot-activated rear liftgates and any other features a manufacturer wishes to incorporate with a simple gesture action.

The MGC3140 is Automotive Electronics Council AEC-Q100 qualified with an operating temperature range of -40°C to +125°C, and it meets the strict EMI and electromagnetic compatibility (EMC) requirements of automotive system designs. Each 3D gesture system consists of a sensor that can be constructed from any conductive material, as well as the Microchip gesture controller tuned for each individual application.

While existing solutions such as infrared and time-of-flight technologies can be costly and operate poorly in bright or direct sunlight, the MGC3140 offers reliable sensing in full sunlight and harsh environments. Other solutions on the market also come with physical constraints and require significant



FIGURE 4

The MGC3140 3D gesture controller is Microchip's first qualified for automotive use. It's suited for a range for applications such as navigating infotainment systems, sun shade operation, interior lighting and more.

infrastructure and space to be integrated in a vehicle. The MGC3140 is compatible with ergonomic interior designs and enables HMI designers to innovate with fewer physical constraints, because the sensor can be any conductive material and hidden from view.

VEHICLE NETWORKING

While applicable to areas beyond infotainment, an automobile's ability to network with the outside world has become ever more important. As critical vehicle powertrain, body, chassis, and infotainment features increasingly become defined by software, securely delivering updates such as fixes and option packs over the air (OTA) enhances cost efficiency and customer convenience. Serving those needs, in October STMicroelectronics released its latest Chorus automotive MCU that provides a gateway/domain-controller solution capable of handling major OTA updates securely.

With three high-performance processor cores, more than 1.2 MB RAM and powerful



FIGURE 5

The SPC58 H Line of MCUs can run multiple applications concurrently to allow more flexible and cost-effective vehicle-electronics architectures. Two independent Ethernet ports provide high-speed connectivity between multiple Chorus chips throughout the vehicle.



FIGURE 6

INICnet technology is a synchronous, scalable solution that significantly simplifies building audio and infotainment systems, offering seamless implementation in vehicles that have Ethernet-oriented system architectures.

on-chip peripherals, ST's new flagship SPC58 H Line joins the Chorus Series of automotive MCUs and can run multiple applications concurrently to allow more flexible and cost-effective vehicle-electronics architectures (**Figure 5**). Two independent Ethernet ports provide high-speed connectivity between multiple Chorus chips throughout the vehicle and enable responsive in-vehicle diagnostics. Also featuring 16 CAN-FD and 24 LINFlex interfaces, Chorus can act as a gateway for multiple ECUs (electronic control units) and support smart-gateway functionality via the two Ethernet interfaces on-chip.

To protect connected-car functionalities and allow OTA updates to be applied safely, the new Chorus chip contains a Hardware Security Module (HSM) capable of asymmetric cryptography. Being EVITA Full compliant, it implements industry-leading attack prevention, detection and containment techniques.

Working with its large on-chip 10 MB flash, the SPC58NH92x's context-swap mechanism allows current application code to run continuously even while an update is downloaded and made ready to be applied

later at a safe time. The older software can be retained, giving the option to roll-back to the previous version in an emergency. Hyperbus and eMMC/SDIO high-speed interfaces to off-chip memory are also integrated, enabling further storage expansion if needed.

SINGLE CABLE SOLUTION

Today's automotive infotainment systems comprise mobile services, cross-domain communication and autonomous driving applications as part of in-vehicle networking. As a result, these systems require a more flexible solution for transporting packet, stream and control content. Existing implementations are either costly and cumbersome, or too limited in bandwidth and packet data capabilities to support system updates and internetworking requirements.

To address this need, Microchip Technology in November announced an automotive infotainment networking solution that supports all data types—including audio, video control and Ethernet—over a single cable. Intelligent Network Interface Controller networking (INICnet) technology is a synchronous, scalable solution that significantly simplifies building audio and infotainment systems, offering seamless implementation in vehicles that have Ethernet-oriented system architectures (**Figure 6**).

Audio is a key infotainment feature in vehicles, and INICnet technology provides full flexibility through supporting a variety of digital audio formats with multiple sources and sinks. INICnet technology also provides high-speed packet-data communications with support for file transfers, OTA software updates and system diagnostics via standard Ethernet frames. In this way, INICnet technology supports seamless integration of Internet Protocol (IP)-based system management and data communications, along with very efficient transport of stream data. INICnet technology does not require the development and licensing of additional protocols or software stacks, reducing development costs, effort and time.

INICnet technology provides a standardized solution that works with both Unshielded Twisted Pair (UTP) at 50 Mbps and coaxial cable at 150 Mbps. With low and deterministic latency, INICnet technology supports deployment of complex audio and acoustics applications. Integrated network management supports networks ranging from two to 50 nodes, as well as processor-less

RESOURCES

Cypress Semiconductor | www.cypress.com

Infineon Technologies | www.infineon.com

Microchip | www.microchip.com

OpenSynergy | www.opensynergy.com

Renesas Electronics America | www.renesas.com

STMicroelectronics | www.st.com

or slim modules where the node is remotely configured and managed. The solution's Power over Data Line (PoDL) capability saves costs on power management for microphones and other slim modules. Nodes can be arranged in any order with the same result, and any node in the system can directly communicate with any other node in the system.

SECURITY FOR CONNECTED CARS

As cars become more network-connected, the issue of security takes on new dimensions. In October, Infineon Technologies announced a key effort in cybersecurity for the connected car by introducing a Trusted Platform Module (TPM) specifically for automotive applications—the first on the market, according to the company. The new OPTIGA TPM 2.0 protects communication between the car manufacturer and the car, which increasingly turns into a computer on wheels. A number of car manufacturers already designed in Infineon's OPTIGA TPM.

The TPM is a hardware-based security solution that has proven its worth in IT security. By using it, car manufacturers can incorporate sensitive security keys for assigning access rights, authentication and data encryption in the car in a protected way. The TPM can also be updated so that the level of security can be kept up to date throughout the vehicle's service life.

Cars send real-time traffic information to the cloud or receive updates from the manufacturer "over the air," for example to update software quickly and in a cost-effective manner. The senders and recipients of that data—whether car makers or individual components in the car—require cryptographic security keys to authenticate themselves. These critical keys are particularly protected against logical and physical attacks in the OPTIGA TPM as if they were in a safe.

EARLY PHASE CRITICAL

Incorporating the first or initial key into the vehicle is a particularly sensitive moment for car makers. When the TPM is used, this step can be carried out in Infineon's certified production environment. After that, the keys are protected against unauthorized access; there is no need for further special security precautions. The TPM likewise generates, stores and administers further security keys for communication within the vehicle. And it is also used to detect faulty or manipulated software and components in the vehicle and




FIGURE 7

The SLI 9670 consists of an attack-resistant security chip (shown) and high-performance firmware developed in accordance with the latest security standard. The firmware enables immediate use of security features, such as encryption, decryption, signing and verification.

initiate troubleshooting by the manufacturer in such a case.

The SLI 9670 consists of an attack-resistant security chip and high-performance firmware developed in accordance with the latest security standard (**Figure 7**). The firmware enables immediate use of security features, such as encryption, decryption, signing and verification. The TPM can be integrated quickly and easily in the system thanks to the open source software stack (TSS stack) for the host processor, which is also provided by Infineon. It has an SPI interface, an extended temperature range from -40°C to 105°C and the advanced encryption algorithms RSA-2048, ECC-256 and SHA-256. The new TPM complies with the internationally acknowledged Trusted Computing Group TPM 2.0 standard, is certified for security according to Common Criteria and is qualified in accordance with the automotive standard AEC-Q100.

Side by side with driverless vehicle innovations, there's no doubt that infotainment systems represent one of the most dynamic subsets of today's automotive systems design. MCU vendors offer a variety of chip and software solutions addressing all the different pieces of car infotainment requirements from display interfacing to connectivity to security. *Circuit Cellar* will continue to follow these developments. And later this year, we'll take a look specifically at MCU solutions aimed at enabling driverless vehicles and assisted driving technologies. 

SWaP Needs Drive Non-Standard SBC Demand All-In-One Solutions

TECH SPOTLIGHT

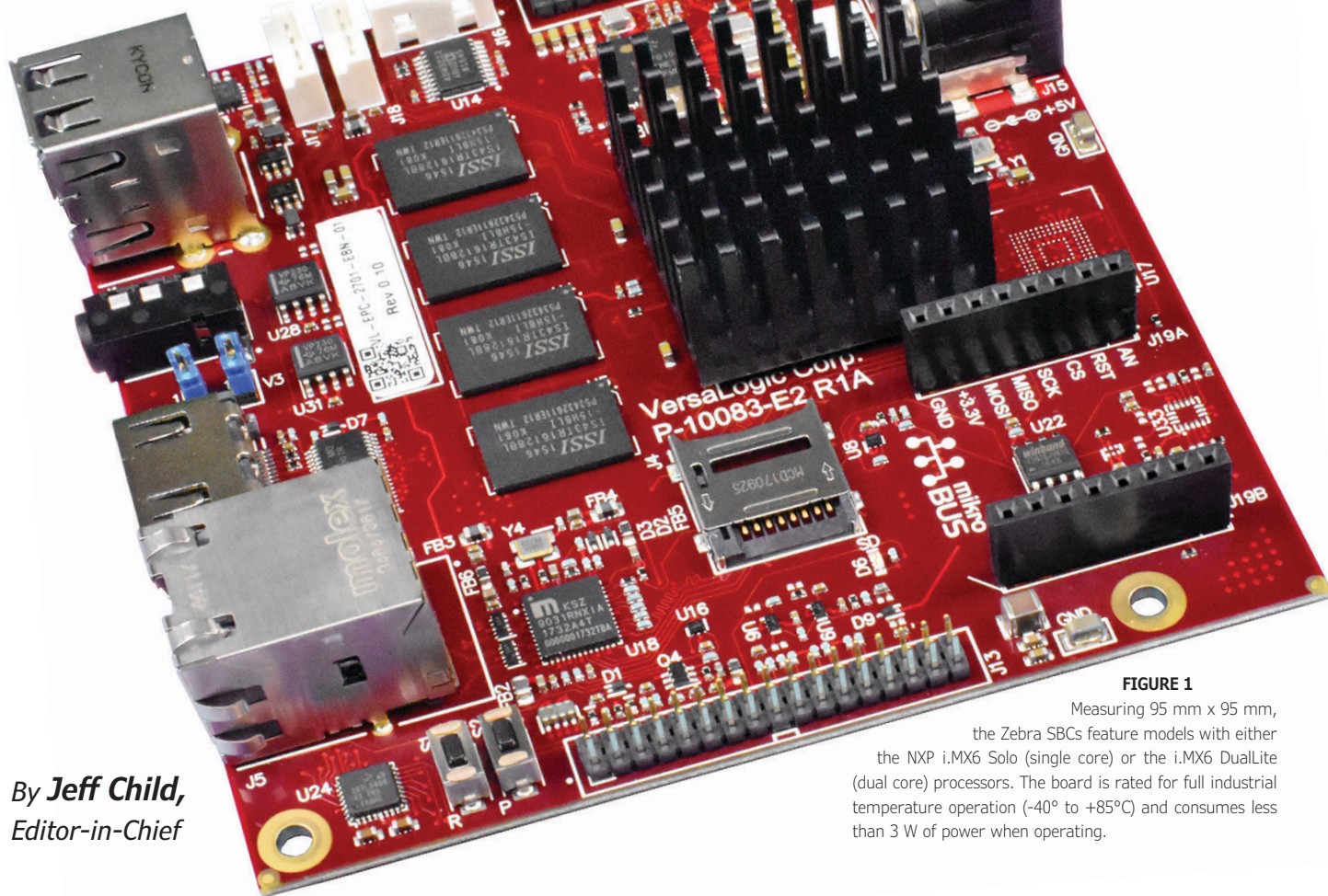


FIGURE 1

Measuring 95 mm x 95 mm, the Zebra SBCs feature models with either the NXP i.MX6 Solo (single core) or the i.MX6 DualLite (dual core) processors. The board is rated for full industrial temperature operation (-40° to +85°C) and consumes less than 3 W of power when operating.

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

Driven by embedded system requirements to shrink size, weight and power consumption (SWaP), today's non-standard single-board computers (SBCs) are serving up a rich set of functionality and features.

Thriving well on their own outside the world of standards-based SBCs, today's crop of non-standard form factor embedded computers trend toward designs that are extremely compact, and well suited for size-constrained system designs. These non-standard form factors free designers from the size and cost overheads inherent to standards-based boards that include a standard bus or interconnect architecture.

By definition the category of non-standard SBCs is open-ended and difficult to encapsulate. Among these SBCs are a variety of open-spec "hacker" style boards such as Raspberry Pi and Arduino cards. There are also SBC designs

that strive to offer the smallest form factor possible—so-called "tiny" SBCs. Given that these boards tend to be literally "single board" solutions, there's often no need to be compatible with multiple companion I/O boards.

Non-standard SBCs are distinct from Computer-on-Module (COM) boards, such as COM Express boards. COM boards are designed to provide just the computing core for a system—mainly just processor, memory and some key interface interconnect. SBCs in contrast are meant to provide complete system functionality, although that definition of "system" keeps expanding. And that's been to the benefit of system developers looking to craft more compact system designs.

COMPLETE SOLUTIONS

An example that illustrates the distinction between COM boards and complete SBCs is VersaLogic's Zebra line of production-ready Arm-based embedded computers. Introduced in May, the boards feature models with either the NXP i.MX6 Solo (single core) or the i.MX6 DualLite (dual core) processors. The Zebra is rated for full industrial temperature operation (-40° to +85°C) (**Figure 1**). The compact 95 mm x 95 mm computer board typically consumes less than 3 W of power when operating.

Unlike many Arm-based "modules," these Arm-based products are complete board-level computers. They do not require additional carrier cards, companion boards, connector break-out boards or other add-ons to function. For ease of mounting and future upgrades, the Zebra product conforms to the size and mounting points of the industry standard COM Compact format. Unlike proprietary-format Arm products, VersaLogic Arm boards provide a standardized mounting pattern for simplified upgrading in the future.

Both Zebra models include soldered-on memory and a variety of I/O. The on-board I/O includes a Gigabit Ethernet port with network boot capability, two USB 2.0 ports, serial I/O (RS-232), CAN Bus, I²C and SPI. An on-board 6-axis e-compass is optional. Zebra meets MIL-STD-202G specifications to withstand high impact and vibration. According to the company, the board is designed with careful component choices and long product life in mind. VersaLogic's 10+ year life-extension program offers long production cycles free from expensive migration changes and upgrades.

SBC WITH DATA ACQ

The level of integration possible now on a small form factor board is so high that SBC vendors have begun offering more than just the typical SBC I/O function. They're also adding extra capabilities for data acquisition and wireless support. Exemplifying these trends, Technologic Systems announced an engineering sampling program for a wireless- and data acquisition focused SBC with open specifications that runs Debian Linux on NXP's low-power i.MX6 UL SoC. The -40°C to 85°C tolerant TS-7180 is designed for industrial applications such as industrial control automation and remote monitoring management, including unmanned control room, industrial automation, automatic asset management and asset tracking.

Similar to the company's i.MX6-based TS-7970, the TS-7180 has a 122 mm x 112 mm footprint (**Figure 2**). At the heart of the board is a low-power Cortex-A7 based i.MX6 UL that enables the board to run at 0.91 W typical power consumption. The new SBC includes

an FPGA. The FPGA enables the optional, 3x 16-bit wide quadrature counters, which are accessible via I²C registers. The "quadrature and edge-counter inputs provide access to dual, optional tachometers," according to Technologic. The quadrature counters and tachometers are part of a DAQ subsystem with screw terminal interfaces that is not available on its other i.MX6 UL boards. The data acq features also include analog and digital inputs, DIO and PWM.

In terms of wireless options, the TS-7180 provides a cellular modem socket that supports either MultiTech or NimbeLink wireless modules. Also included are Wi-Fi/Bluetooth, optional GPS and a socket for Digi's XBee modules, which include modems for RF, 802.15.4, DigiMesh, and more. There are also dual 10/100 Ethernet port with an optional Power-over-Ethernet daughtercard.

The TS-7180 ships with up to 1 GB RAM and 2 kB FRAM (Cypress 16-kbit FM25L16B), which provides reliable data retention while eliminating the complexities, overhead and system level reliability problems caused by EEPROM and other nonvolatile memories, says Technologic. The board also provides a microSD slot and 4 GB eMMC, which is configurable as 2 GB pSLC mode for additional system integrity. The SBC provides a USB 2.0 host port, as well as micro-USB OTG and serial console ports. Five serial interfaces, including TTL and RS-485 ports, are available

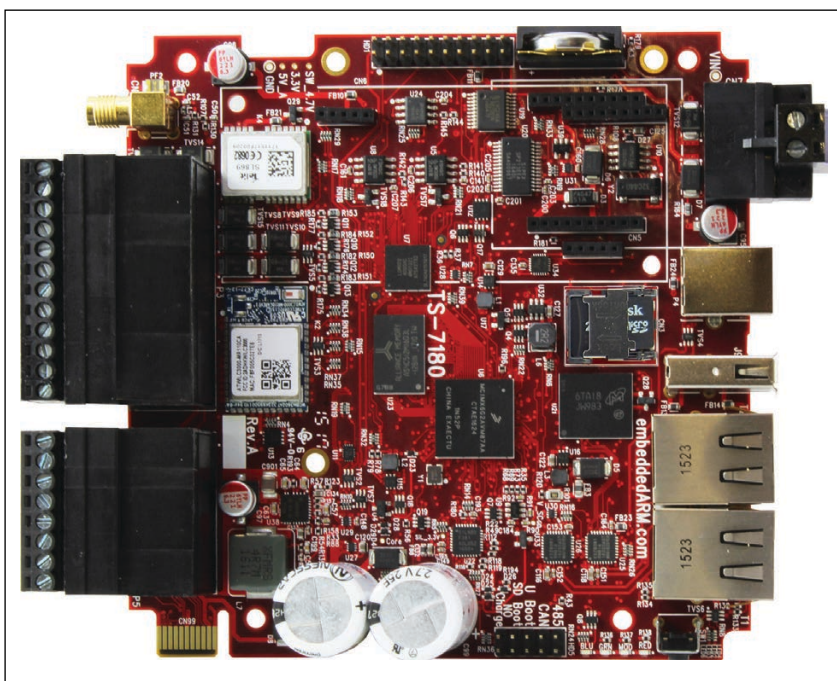
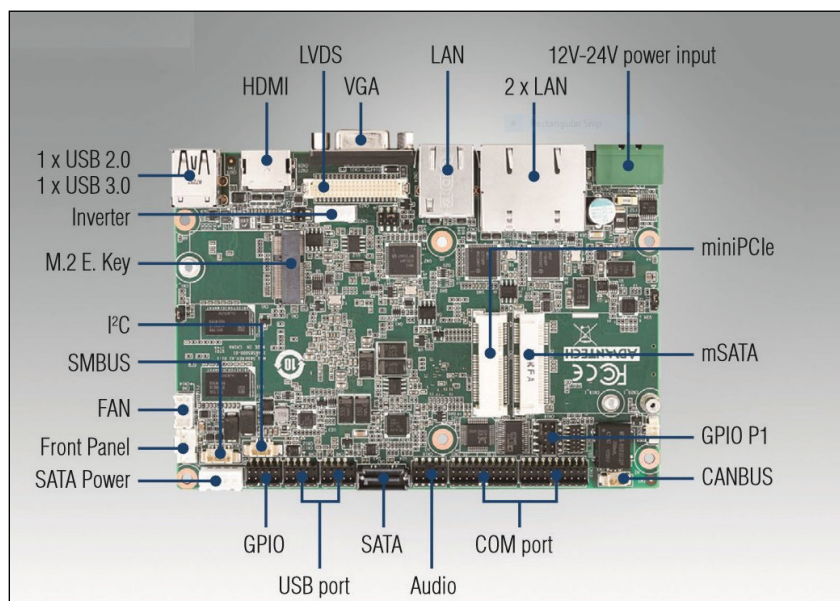
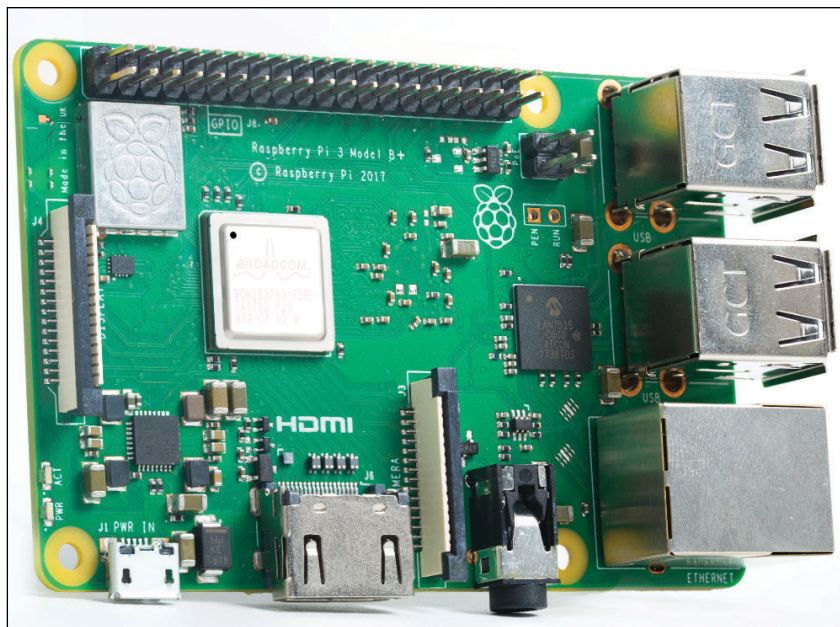


FIGURE 2

The 122 mm x 112 mm footprint TS-7180 sports a Cortex-A7 based i.MX6 UL that enables the board to run at 0.91 W typical power consumption. Beyond traditional SBC functionality, the board also adds extra capabilities for data acquisition and wireless support.

**FIGURE 3**

The MIO-5850 is a 3.5" fanless SBC that supports 4th Gen Intel Celeron J1900 and Atom E3800 series low power consumption processors with 10 W to 12 W total power dissipation. It's designed with 2 GB or 4 GB of onboard memory and anti-vibration eMMC storage. It has three GbE ports, CAN bus with isolation and 12 V to 24 V input power.

**FIGURE 4**

Compared to its predecessor, the Raspberry Pi 3 Model B+ provides a faster, 1.4 GHz Broadcom SoC and pre-certified, dual-band 802.11ac and Bluetooth 4.2. The LAN port has moved from 10/100 port to a USB-powered, up to 300 Mbps Gigabit Ethernet port.

on screw terminals along with a CAN port. Other features include an RTC and an optional enclosure and 9-axis IMU. The board runs on an 8 V to 30 V input with optional external power supply and Technologic's TS-SILO SuperCap for 30 seconds of battery backup.

3.5" ALL-IN-ONE SOLUTIONS

While not an official "standard" created by any particular standards organization, the 3.5" size remains a very popular form factor among SBCs. The 3.5" size is close to the PC/104's specification, and, as a result, many of the same companies manufacture them. But, unlike PC/104—3.5" SBCs aren't meant to be stacked. And in some cases, this 3.5" size is more suited to today's levels of integration. In the past, PC/104 stacks were necessary because not all the needed I/O functionality could fit on one board. But today, what used to require a stack of boards can now be integrated into a single 3.5" SBC.

In an example along those lines, last August Advantech announced the release of its MIO-5850, a domain focused 3.5" fanless SBC that supports 4th Gen Intel Celeron J1900 and Atom E3800 series low power consumption processors with 10 W to 12 W total power dissipation (**Figure 3**). MIO-5850 is designed with 2 GB or 4 GB of onboard memory and anti-vibration eMMC storage. Three GbE ports, CAN bus with isolation and 12 V to 24 V power make this SBC well suited for domain focused applications in industrial automation, transportation and outdoor self-service kiosks. MIO-5850 comes bundled with Advantech's exclusive iManager 3.0 utilities and WISE-PaaS/RMM cloud ready solution for remote device management.

MIO-5850 is designed to fulfill a variety of vertical application needs and adopts a rich array of I/O interface including: 3 GbE ports, 1 x CAN bus with 15 kV isolation, 1 x USB 3.0, 5 x USB 2.0, 4 x COM ports and 16-bit GPIO. For extra expansion, MIO-5850 supports 1 x full-size miniPCIe, 1 x mSATA and an optional M.2 E-key. These allow various peripheral modules like Wi-Fi, 3G/LTE and additional storage to expand functionality. MIO-5850 can operate under wide temperature settings ranging from -40°C to 85°C, making it suited for use in rugged and harsh environments such as factory automation, railways, outdoor signage and kiosks.

MIO-5850 can be used in combination with Advantech's WISE-PaaS/RMM solution and iManager 3.0. WISE-PaaS/RMM is part of Advantech's WISE-PaaS cloud solution, which provides centralized management features including HW/SW status monitoring, remote control and system backup/recovery. It supports server redundancy and hierarchical

RESOURCES

Advantech | www.advantech.com

Garz & Fricke | www.garz-fricke.com

NXElec | www.nxelec.com

Technologic Systems | www.embeddedarm.com

VersaLogic | www.versalogic.com

server management, which increases service reliability and availability and helps customers incorporate IoT cloud services.

POPULAR RASPBERRY PI

In the category of open-specification, so-called “hacker boards,” Raspberry Pi SBCs remain the most popular choice. Over the past couple years, *Circuit Cellar’s* partner LinuxGizmos.com has done a reader survey, and the Raspberry Pi 3 Model B has been the winner each time. The board is based on the Broadcom 1.2 GHz BCM2837 processor which has 4x Cortex-A53 cores and a 400 MHz Broadcom VideoCore IV GPU along with 1 GB of SDRAM. In the most recent LinuxGizmos survey last June, the RPi 3B slipped to fourth place thanks to roll out of the faster, more feature rich, and identically priced Raspberry Pi 3 Model B+.

The Raspberry Pi 3 Model B+ was announced March last year, with the same price and much the same layout and feature set of the RPi 3 Model B, but with both major and minor improvements. The 3B+ provides a faster, 1.4 GHz Broadcom SoC and pre-certified, dual-band 802.11ac and Bluetooth 4.2 (Figure 4). The LAN port has moved from 10/100 port to a USB-powered, up to 300 Mbps Gigabit Ethernet port, and there’s even a \$20 Power-over-Ethernet POE HAT option. The initial PoE HAT reportedly had regulator problems, but Raspberry Pi Trading offered a refund and a repaired model is now available. Other RPi 3B+ improvements include a better PMIC, a heat spreader and 0 to 50°C support.

TINY SIZED SBCs

As the previous examples show, today’s level of electronics has enabled small SBCs to include more functionality. The other consequence is SBC designers can offer a complete computer on ever smaller form factor boards—some in the “tiny” sized category. *Circuit Cellar* did a Product Focus roundup of these small and tiny SBCs last August (*Circuit Cellar* 337). An example of this tiny style SBC is the 80 mm x 42 mm Innostick 6 launched by Shanghai Naxing Electronics (NXElec) last July (Figure 5). The board features the low-power, i.MX6 ULL, a variation on the i.MX6 UltraLite (UL) that similarly offers a single Cortex-A7 core, in this case clocked to 900 MHz.

The board ships with a choice of Yocto Project “Morty” based Linux stacks: one with X11 and one with Qtopia. Debian Stretch is also supported. Innostick 6 integrates 512 MB DDR3L and either a 16 GB or a 32 GB eMMC. There’s no Ethernet, but you get Wi-Fi and Bluetooth. The only coastline ports are the USB 2.0 host and micro-USB OTG ports. Also included on the SBC is a resistive touch-

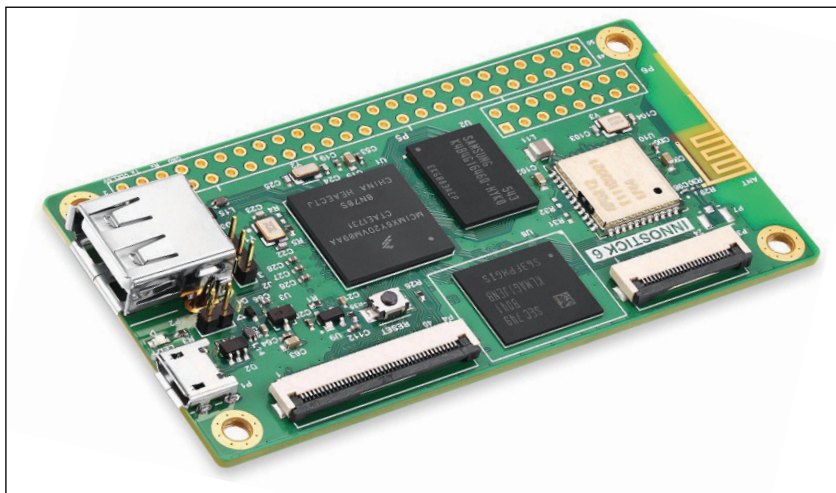



FIGURE 5

The Innostick 6 SBC features the low-power, i.MX6 ULL, a variation on the i.MX6 UltraLite (UL) that similarly offers a single Cortex-A7 core, in this case clocked to 900 MHz.

enabled LCD interface and a CSI Parallel camera interface. There are also 16- and 50-pin GPIO headers.

Another example SBC pushing the size barrier is the Nallino Core SBC from Garz & Fricke introduced in November. The SBC runs a Yocto Project based stack on NXP’s i.MX6 ULL SoC. At 113 mm x 47 mm x 18 mm, the Nallino Core is a little bigger than other “tiny” category boards, but it also offers a few more features. The 55-gram board is available in a baseline M version and a more feature-rich S model. Both clock the low power i.MX6 ULL to 792 MHz.

The Nallino Core SBC is equipped with 4 GB MLC eMMC and 512 MB RAM. The M model uses DDR3 RAM while the S provides DDR3L, which can run at 1.35 V in addition to the standard 1.5 V. The S model also offers a 1 GB DDR3L option. Both Nallino Core models provide a microSD slot, a micro-USB OTG port and a 24-bit RGB TTL display interface with support for backlit, I²C-based capacitive touchscreens up to 5 inches. You also get CAN, RS-485, RS-232 and buzzer interfaces, as well as a real-time clock.

Both versions list a 10/100 Ethernet port, although the block diagram suggests you could optionally remove this in favor of a Wi-Fi/Bluetooth option. The S version adds a USB 2.0 host port, as well as I²C, UART, SPI, and 4x4 matrix keyboard interfaces. This more fully featured model also adds a 4-wire resistive analog touch controller and a 1.5 W speaker interface. The S model supplies a wide-range 9 V to 32 V input in place of the M model’s 5 V input. Both versions support an extended, 0 to 60°C operating range. 

Product Focus: ADCs and DACs

Resolution and Speed

ADCs and DACs are two of the key IC components that enable digital systems to interact with the real world. Makers of analog ICs are constantly evolving their DAC and ADC chips pushing the barriers of resolution and speed.

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

As critical enablers serving a wide variety of embedded applications, analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) are available in many combinations of high-speed or high-resolution—or some tradeoff of both. Analog IC vendors continue to innovate, enhancing the performance and architectures of these critical devices.

For high-speed DACs the subsets of products include wideband radio frequency, intermediate frequency signal processing and general-purpose baseband classes. System designers use these DACs in wired and wireless communications, instrumentation, radar, electronic warfare and general waveform synthesis. The latest and greatest high-speed ADCs meanwhile seek to blend the high performance and optimized power consumption necessary for today's demanding receiver/data acquisition applications. These products are used in wired and wireless communications, instrumentation, radar, electronic warfare and general data acquisition.


Simultaneous sampling capability is offered in some ADCs. That enables a single ADC device to take two or more measurements at the exact same time. This is particularly useful for systems where signal levels change quickly, such as motor control, distribution automation and ultrasound. Such systems benefit from the added precision that simultaneous sampling provides.

FIGURE 1

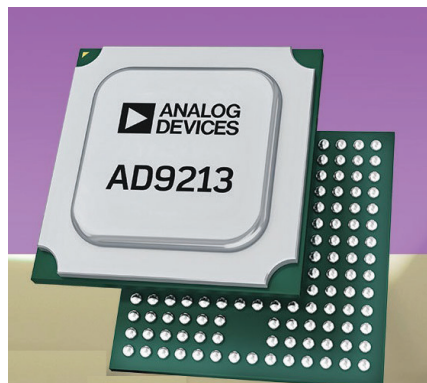
More than 100,000 ADCs from Analog Devices are used within CERN's Large Hadron Collider (LHC). ADI's AD9042 high-speed, low-power monolithic 12-bit ADCs ensure a continuous stream of ultraprecise data during LHC experiments.

Resolution overrides demands for high speed for devices like precision DACs. These operate at less than 30 MHz and are critical data conversion components used in most high-performance signal processing systems. High-precision DACs range from 8 bits to 20 bits in resolution.

More than 100,000 ADCs from Analog Devices are used within CERN's Large Hadron Collider (LHC) (**Figure 1**). Located outside Geneva, Switzerland, in a tunnel 100 meters underground and 27 kilometers in circumference, the LHC is one of the most sophisticated and expensive scientific instruments ever developed. CERN depends on ADI's AD9042 high-speed, low-power monolithic 12-bit ADCs for high reliability and performance to ensure a continuous stream of ultraprecise data during carefully orchestrated, resource intensive LHC experiments. Maintaining 80 dB spurious-free dynamic range (SFDR) over a bandwidth of 20 MHz, AD9042 ADCs support the dynamic range required to measure the energy captured by an array of 64,000 lead tungstate crystals. These scintillating crystals are designed to absorb particles formed as electrons, positrons, and photons pass through the LHC's electromagnetic calorimeter.

The product gallery displayed on the next couple of pages shows a representative snapshot of today's ADCs and DACs 

A-D Converters

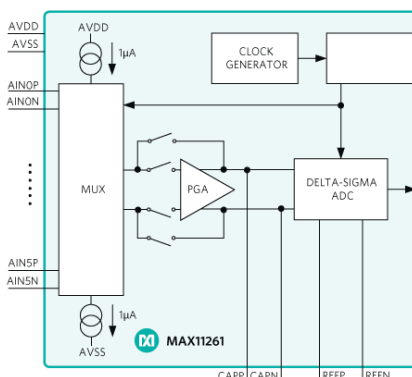


ADC Delivers 12-Bit, 10.25-GSPS RF Performance

The AD9213 from Analog Devices is a single 12-bit, 10.25 GSPS, RF ADC with a 6.5 GHz input bandwidth. The AD9213 has been optimized to support high dynamic range frequency and time domain applications requiring wide instantaneous bandwidth and low code error rates (CER). The AD9213 features a 16-lane JESD204B interface to support its maximum bandwidth capability.

- High instantaneous dynamic range
- Noise spectral density 154 dBFS/Hz
- SFDR 68 dBc (1 GHz, -1 dBFS)
- Low power consumption: 5.1 W at 10 Gsps
- Integrated Input Buffer (6.5 GHz input bandwidth)
- 1.4 V_{p-p} full-scale input with RIN=50 Ω
- Overvoltage protection
- 16-lane JESD204B output (up to 16 Gbps line rate)
- Multichip sync capable with 1 sample accuracy
- DDC NCO synchronization included

Analog Devices
www.analog.com



24-Bit, 6-Channel, Delta-Sigma ADC Sports I²C Interface

Maxim Integrated's MAX11261 is a 6-channel, 24-bit delta-sigma ADC that achieves exceptional performance while consuming very low power. Sample rates up to 16 ksps allow precision DC measurements. The device also features a 64-entry, on-chip FIFO to offload the host processor. The MAX11261 communicates through an I²C-compatible serial interface and is available in a small, wafer-level package (WLP).

- Analog supply: 2.7 V to 3.6 V
- Digital supply: 1.7 V to 2.0 V, or 2.0 V to 3.6 V
- PGA Low-noise mode: 6.2 nV/√Hz noise
- Fully differential signal and reference inputs
- Internal system clock of 8.192 MHz
- I²C-compatible serial interface
- Supports standard, fast-mode, and fast-mode plus I²C
- 64-entry on-chip FIFO
- Hardware interrupt for input monitoring and FIFO usage
- WLP 2.838 mm x 2.838 mm x 0.5 mm
- -40°C to +85°C temperature range

Maxim Integrated
www.maximintegrated.com



Precision 24-Bit Sigma Delta ADC Features Tiny Footprint

The ADS122C04 from Texas Instruments is a precision, 24-bit, ADC that offers many integrated features to reduce system cost and component count in applications measuring small sensor signals. The device features two differential or four single-ended inputs through a flexible input multiplexer (MUX), a low-noise, programmable gain amplifier (PGA), two programmable excitation current sources, a voltage reference, an oscillator and a precision temperature sensor.

- Current consumption 315 µA (typ.)
- Wide supply range: 2.3 V to 5.5 V
- Programmable gain: 1 to 128
- Programmable data rates: Up to 2 kSPS
- Up to 20 bits effective resolution
- Two differential or four single-ended inputs
- I²C-compatible interface
- 16 pin-configurable I²C addresses
- Package: 3.0-mm x 3.0-mm x 0.75-mm WQFN

Texas Instruments
www.ti.com

D-A Converters

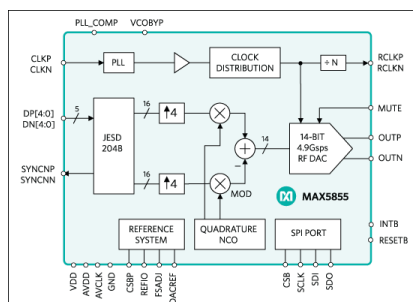


16-Bit DAC Provides Low Power Industrial Solution

Analog Devices' AD5758 is a single-channel, voltage and current output DAC that operates with a power supply range from -33 V (minimum) on AVSS to $+33\text{ V}$ (maximum) on AVDD1 with a maximum operating voltage between the two rails of 60 V . On-chip dynamic power control (DPC) minimizes package power dissipation, which is achieved by regulating the supply voltage (V_{DPC}) to the VIOU output driver circuitry from 5 V to 27 V using a buck DC-DC converter, optimized for minimum on-chip power dissipation. The C_{HART} pin enables a HART signal to be coupled onto the current output.

- 16-bit resolution and monotonicity
- DPC for thermal management
- Current/voltage output available on a single terminal
- User-programmable offset and gain
- Advanced On-chip diagnostics including a 12-bit ADC
- On-chip reference
- Output fault protection
- -40°C to +115°C temperature range
- 32-lead, 5 mm × 5 mm LFCSP package

Analog Devices
www.analog.com

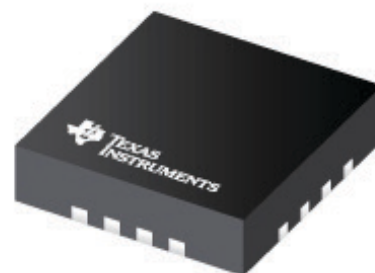


16-Bit DAC Delivers Direct RF Synthesis of 1,000 MHz Bandwidth

The MAX5855 from Maxim Integrated is a high-performance, interpolating and modulating, 16-bit, 4.9 Gsps RF DAC that can directly synthesize up to 1,000 MHz of instantaneous bandwidth from DC to frequencies greater than 2.45 GHz. The device is optimized for cable access and digital video broadcast applications and meets spectral emission requirements for a broad set of radio transmitters and modulators including DOCSIS 3.1/3.0, DVB-C/C2, DVB-T2, DVB-S2X, ISDB-T and EPoC.

- Eliminates I/Q imbalance and LO feedthrough
- Enables multi-band RF modulation
- 4.9152 Gsps DAC output update rate
- High-performance 14-bit RF DAC core
- Digital baseband I/Q with 4x interpolation
- Sub-1 Hz NCO resolution
- Integrated clock multiplying PLL + VCO
- 5-lane JESD204B input data interface
- Divided reference clock output
- SPI interface for device configuration

Maxim Integrated
www.maximintegrated.com



8-Channel, 16-Bit DAC Has Precision Internal Reference

Texas Instruments' DACx0508 is a pin-compatible family of low power, eight-channel, buffered voltage-output DACs with 16-, 14- and 12-bit resolution. The DACx0508 includes a 2.5-V, 5-ppm/°C internal reference, eliminating the need for an external precision reference in most applications. A user selectable gain configuration provides full-scale output voltages of 1.25 V (gain = 1/2), 2.5 V (gain = 1) or 5 V (gain = 2).

- Linearity: ± 1 LSB maximum at 16-bit resolution
- Integrated 2.5 V precision internal reference
- 20 mA drive with 0.5 V from supply rails
- User Selectable Gain: 2, 1 or 0.5
- Reset to zero scale or midscale
- Power supply: 2.7 V to 5.5 V
- Temperature range: -40°C to 125°C
- 50 MHz SPI compatible serial interface
- Packages: 3 mm \times 3 mm, 16-pin WQFN
- Or 2.4 mm \times 2.4 mm, 16-pin DSBGA

Texas Instruments
www.ti.com



Clarius Mobile Health
revolutionized ultrasounds.
Xilinx's technology helped
them get there.



Imaging technology that fits in a pocket and works with a smart phone is just the kind of tech to shake up the \$6 billion ultrasound industry. But only if it's executed to perfection.

See how design engineers benefit from Zynq® products. circuitcellar.com/Avnet2

Embedded in Thin Slices

Bluetooth Mesh (Part 1)

Alternatives Compared

Wireless mesh networks are being widely deployed in a variety of settings. In this article, Bob begins his series on Bluetooth mesh. He starts with defining what a mesh network is, then looks at two alternatives available to you as embedded systems designers.

By
Bob Japenga

In 2003, our company designed an energy monitor to be used in a load shedding application for electrical customers who were charged for peak usage. The target customers were universities and municipalities, which had multiple sites but a single electric bill that charged them for peak usage as a single user. This common billing paradigm made it important to know the energy usage across multiple sites—some upward to a mile away. Communication between the nodes was done using a proprietary radio to a central hub that passed the data up to the central office where the load shedding algorithms were performed. We weren't involved with the radio portion during this initial stage.

After a few years, the relationship between our customer and the radio designer went south and they came to us to design the radio. Initially we thought it would be a piece of cake. Then we discovered that the radios used a sophisticated mesh network that auto-configured the routes, minimized the hops from node to central hub and self-healed when nodes dropped out. We got gun shy about the complexity of the design as well as our ability to test the vast array of possible scenarios that can take place in a dynamic mesh network. We decided to

redesign without changing the radio software at all. But we got our first taste of a mesh network and learned a little about what we didn't know.

A few years later, we got involved in a smart-grid solar project from the ground up. One of the requirements was to provide a ZigBee interface to other devices connected to the grid. So, again, we dipped our toes into the wireless mesh network waters. But it became very clear that we were complete novices about these networks and had a lot to learn.

Over the next few articles, I want to introduce a very popular and potentially powerful entry into wireless mesh networking. In July of 2017, the Bluetooth SIG introduced Bluetooth mesh, which promised to revolutionize IoT. This month I want to introduce two of the competitors to Bluetooth mesh—ZigBee and Thread—and highlight some of the distinctions. These three network protocols are often compared since they all can use a common radio interface and are often implemented on the same chip (**Figure 1**). Although Wi-Fi mesh networks will play a major role in the future of IoT, we won't be comparing it to these three because of its power consumption. In the next set of articles we will drill down into some of the details of deploying a Bluetooth mesh network.

DEFINITION OF TERMS

Node: This is a specific device on the wireless mesh network that can send and receive data on the network.

Hop: As a network data travels from the source to its destination, it traverses between one or more nodes. The transmission between these nodes is called a hop.

Relay: When a node is transmitting some other node's data, it is a relay. (Some notations call this a router.)

Wireless Mesh Network: A wireless mesh network uses a topology where the nodes in the network work together to move the data from the source to the destination. The source RF signal does not need to be received by the destination node for the data from the source to be received by the destination. **Figure 2** shows a simple wireless mesh network. Node A's RF signal cannot be received by Nodes C, D or E. Node B's RF signal cannot be received by Node's D or E. But data from Node A can be sent to Node E with the help of Nodes B, C and D. Transmission from Node A to Node E is a four-hop transmission.

Flooding/Routing: There are two basic mechanisms for a wireless mesh network to get data from the source node to the destination node. One mechanism is called flooding. The source node does not know who is going to relay the data to get the data to the destination. All nodes configured as relays in the mesh that receive the data can forward it on (also by flooding). The other mechanism is called dynamic routing where the source node has a routing table to indicate one or more paths it can take to get its data to its destination. Bluetooth uses what is called managed flooding (more on that in a later article). Thread and ZigBee use dynamic routing tables which self-configure.

OOB Authentication: Out of Band (OOB) authentication and key exchange is where an alternate means of authenticating and exchanging keys happens outside the normal



FIGURE 1

The three network protocols—Bluetooth mesh, ZigBee and Thread—are often compared since they all can use a common radio interface and are often implemented on the same chip.

“band” of communications. For example, requiring the user to enter a passcode is an OOB method. Pressing an input on a node device X times could be used as an OOB method. Passing keys through a Near Field Communications (NFC) channel is another. Many of the ICs that are offered with Bluetooth Mesh, Thread and ZigBee built in also offer NFC capability.

TRADE-OFFS AMONG THE THREE

Let's look at some of the trade-offs when choosing a wireless mesh network.

Network Performance: Silicon Labs, which makes parts that support all three mesh networks, released a report documenting a 12-month long study of the network performance of the three mesh networks [1]. The Silicon Labs benchmark showed very little differences between the three protocols on small networks (less than 24 nodes) and small payloads (less than 10 bytes). Thread and ZigBee outperform Bluetooth in both throughput and latency as either the network grows or the payload increases in size. So, if latency and/or larger payloads are important you, may want to stay away from Bluetooth.

Routing/Flooding: This could be a deal breaker for some of your designs. The Bluetooth standard left open the door for

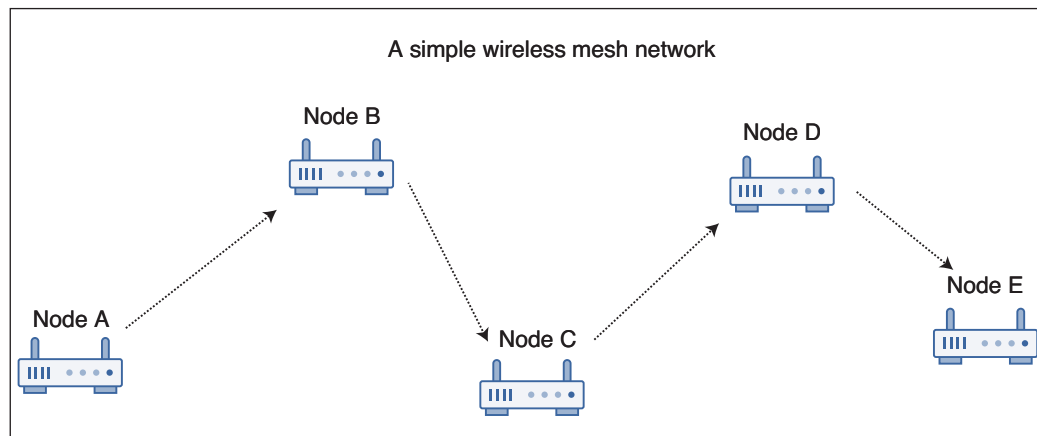


FIGURE 2

Shown here is a simple wireless mesh network. Node A's RF signal cannot be received by Nodes C, D or E. Node B's RF signal cannot be received by Node's D or E. But data from Node A can be sent to Node E with the help of Nodes B, C and D. Transmission from Node A to Node E is a four-hop transmission.

using dynamic routing, but for now doesn't support it. Thread and ZigBee support dynamic routing. Dynamic routing can be a serious problem for mobile nodes since the network could be saturated with attempts to dynamically reconfigure the system as the optimal routes keep changing. The algorithms that scared us away 15 years ago are complex and probably not foolproof especially if the nodes move around. Flooding creates problems if you have to dynamically reduce the number of relays in the system or reduce the radio power because you have too many nodes talking at the same time. For example, if you were selling a system with 150 devices spread out such that at full power and with all nodes configured as relays, most nodes could not hear each other, flooding would work fine. But what if another customer placed your devices such that all 150 nodes could hear each other? The scheme could break down. One Bluetooth mesh software development kit (SDK) says:

The flooding-based approach to message relaying can cause a lot of redundant traffic on air, which may impact the throughput and reliability of the network. Therefore, it is highly recommended to limit the number of relays in a network to restrict this effect. The rate of relay-enabled devices in the network is a trade-off between message route-redundancy and reliability. It should be tuned according to network density, traffic volumes, network layout, and requirements for reliability and responsiveness [2].

If you are creating large onesie networks and can hand tweak the relays and radio power, or if you're creating a system with a small number of nodes (less than 20), managed flooding is not an issue.

Power Usage: Each wireless mesh network allows you to minimize power with different schemes. Each one has its own advantages and disadvantages that you the designer must understand if power is an issue. With ZigBee, power can be minimized because transmit and receive are synched allowing devices to all sleep at the same time. Therefore, the ZigBee

radios can be off most of the time. At the appropriate time slot, all radios wake up and the network is alive. Bluetooth implements a friend feature, which allows the designer to have some devices that do not have power constraints to store messages for its sleeping friends. When the power constrained friend wakes up, it asks its friend for all of its messages. Thread enables the designer to designate some devices as sleepy end nodes that need not be awake for relaying and can keep the radio off until needed.

Maturity: ZigBee has been around since 2003. Thread was first released in 2015 and Bluetooth Mesh in July 2017. For those of you who have experience with the SDKs of major suppliers, you know that it takes time to work out the bugs in these complex software packages. Our experience bears this out with these three types of mesh SDKs. Be prepared to provide more hand holding the newer the technology.

SETTING UP THE NETWORK

Each of these wireless mesh network protocols is very different in how you set up the network. Each offers different options, which should be carefully considered when you choose one protocol over another. There are security trade-offs made with each method. All three provide secure encrypted transmissions. The challenge has to do with jump starting the process. It would be ideal if all this could be done at a secure factory where the network is setup, nodes authenticated and keys exchanged. But most of us don't sell systems like that. Let me try to provide the basics. But I want to emphasize that the devil is in the details of this process.

Bluetooth Setup: To securely install a node on a Bluetooth mesh network, one node must be a provisioner. The provisioner has two primary tasks: 1) Establish a secure link over which the keys can be exchanged to allow the node to talk on the network. 2) Authenticate that the new node is one that you want to add to the network. There are several ways that the specification allows this to happen. To create a secure process is complicated when designing headless nodes (when no OOB methods are available). We will look at this more in depth in our next article.

Thread Setup: Because each device is an IPV6 address on the Internet, installing a Thread node can be done with a smartphone, a tablet or PC. Although simple for a home device (the supporting organization touts Thread for use in home automation), the IoT designer needs to fully understand how this will be done if you are installing hundreds of nodes on a factory floor.

ZigBee Setup: Out of the box, with one

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

Nordic Semiconductor | www.nordicsemi.com

Silicon Labs | www.silabs.com

device designated as a ZigBee coordinator, ZigBee nodes can be easily added and can instantly create a mesh network transmitting encrypted data. The problem is the way it is done right out of the box. They use a default Trust Center link key for encryption (published widely on the web). If you use that method in your design, a hacker could easily decrypt the first transmission with this key and decrypt the transmission with the secret keys without any trouble. ZigBee provides the necessary tools to setup the network securely, but they require some OOB method.

SECURITY, RANGE AND MORE

Security: If you can accept the methods prescribed by Bluetooth and Thread, they are more secure (and more complicated) than ZigBee. But Bluetooth and Thread have also not been scrutinized as long. Even the Bluetooth security vulnerability uncovered in July 2018 (Cert Vulnerability #304725 [3]) would not affect a Bluetooth mesh implementation that used OOB methods for setting up the network. So much depends on you the developer following good practices including: 1) Making the device tamper-resistant. (for ZigBee especially, which relies on the Trust Center); 2) Network setup should pass initial security parameters using an OOB method; and 3) Refreshing Keys often. Bluetooth implementations are required to provide a function that does this and the SDK we used had a single function to perform this.

Number of Nodes: If we are thinking big, Thread and ZigBee node size limitation may make Bluetooth a big winner for large networks. Here is the breakdown:

Bluetooth Mesh: 32,767 nodes with a maximum of 126 hops

Thread: 250+ nodes with the maximum number of relays set to 32

ZigBee: 250 nodes

Range: Don't believe the stated range for any of these. One SDK states that for a Bluetooth 5.0 radio to not expect more than 100 feet. However, with a Bluetooth 5.0 radio, we did get some impressive distances through office walls compared to what we get with our Bluetooth 4.2 headsets. Here are the specs nonetheless:

Bluetooth Mesh: 100 feet – 1,000 feet


Thread: 100 feet

ZigBee: 30-300 feet

Gateway Requirements: Because the Thread mesh network nodes are IPV6 addressable, the Gateway (called "Edge Router" in Thread terminology) can be generic (in other words,

no special software). This is not true for the ZigBee or Bluetooth gateways.

CONCLUSION

As the Carpenters' song goes: "We've only just begun..." This topic is gigantic. The Bluetooth specification is hundreds of pages of thick prose. But we have started you down the path—of course—in thin slices. Next time we will look in more depth at setting up a Bluetooth mesh network. 

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.



Circuit Cellar 2018 Archive



Order yours today

cc-webshop.com

The Consummate Engineer

Infrared Sensors

Heat Lights the Way

Infrared sensing technology has broad application, ranging from motion detection in security systems to proximity switches in consumer devices. In this article, George looks at the science, technology and circuitry of infrared sensors. He also discusses the various types of infrared sensing technologies and how to use them.

By
George Novacek

Invisible infrared (IR) radiation, whose wavelength is longer than that of visible red light (for example greater than 780 nm), was discovered by William Herschel in 1800. The IR range most interesting for our topic here lies between 780 nm and 14 μ m and is called Near Infrared. Its spectrum with respect to other radiation is shown in **Figure 1**.

Physicists Planck, Stefan, Boltzmann, Wien, Kirchhoff and others, working in the late nineteenth and early twentieth centuries, defined IR's electromagnetic spectrum and the physical characteristics of its energy. To explain IR behavior, they created a theoretical Blackbody model, which absorbs all incoming radiation while reflecting or transmitting none.

Temperature is an objective measure of hot or cold and is determined with a thermometer. Some thermometers are based on the bulk behavior of thermometric materials such as mercury. Others may rely on measurement of thermal radiation, or assessment of particles' kinetic energy. Every physical body with temperature above "absolute zero" (equal to 0° Kelvin, -273.15°C or -459.666°F) emits electromagnetic radiation proportional to its surface temperature. This radiation can be detected by IR sensors.

One application of IR radiation is for detection of movement of objects whose temperature differs from the background. The first extensive use of such infrared technology was by the military for personnel detection and in missile guidance systems. After declassification, the technology was commercialized by, among others, the security industry, for the design of passive infrared intrusion detectors (PIR). PIR has now become the most popular type of an intrusion detector, because it is inexpensive to manufacture, versatile, and—unlike ultrasonic, microwave or light beam interruption devices—emits no energy, thus making its presence undetectable. IR sensors have become an inexpensive commodity that can be found in many consumer applications, such as proximity switches, security lights, toys and various gadgets.

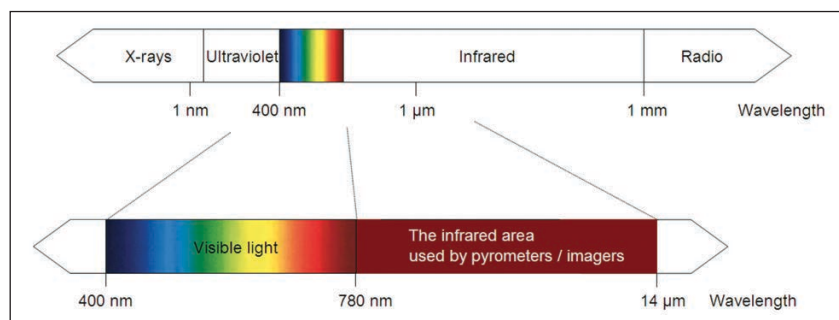


FIGURE 1

Infrared radiation spectrum

TWO MAJOR TECHNOLOGIES

The two major infrared-sensing technologies are thermopiles and pyroelectric sensors. Thermopiles comprise a stack of thermocouples—typically of bismuth and antimony—arranged on a chip around a heat-absorbing black material, integrated with a preamplifier to boost the output signal. Pyroelectric sensors are constructed of crystals such as lithium tantalate (LiTaO₃) or deuterated-triglycine sulfate (DTGS), whose exposure to IR radiation causes surface loading. This, due to the pyroelectric effect, causes a voltage to be generated across the crystal. Because of the extremely high output resistance of the pyroelectric crystal (on the order of $10^{12} \Omega$), an FET source follower is integrated into the sensor to provide low-output resistance.

Most modern pyroelectric sensors for motion detection contain two pyroelectric crystals, X1 and X2, in series or parallel-opposite configuration, as shown in **Figure 2**. They attenuate the sensor's response to common mode stimuli caused by ambient temperature variations, RF irradiation, vibration and other factors. This arrangement keeps the sensor output fairly constant when no moving object is within its field of view.

For reliable operation, the packaging of a PIR detector is crucial. In addition to holding the lens to properly focus the IR beam, the enclosure must prevent free air movement across the sensor. PIR detectors need optics to achieve good sensitivity, although a small aperture may suffice for short-distance detection. In the past, faceted mirrors were the mainstay, but eventually, Fresnel lenses replaced them, due to their low cost, low weight and easy manufacturing. You can still find mirrors in long-range, perimeter-protection devices. In my experience, the enclosure should not be hermetically sealed. In fact, it should be allowed to “breathe” to maintain the same environment inside and out.

Figure 3 illustrates detection of a moving object by a PIR. The sensor is located in the focal plane of an array of Fresnel lenses, often comprising many lens segments to achieve a desired pattern of detection field [2]. Today, there are many commercially available, inexpensive lens arrays [3]. They are usually molded from polyethylene, which, unlike glass, is transparent to near-IR radiation. IR radiation from an object depicted by the green rectangle and moving across the field of view from left to right strikes one sensor element (red rectangles) first, then the other. This causes the sensor to output two pulses. It also explains why PIR detectors are rather insensitive to the axial movement toward or

away from the sensor. Both elements get irradiated at the same time, and the resulting common-mode signal is attenuated. The dual output pulse results only when the object moves across the field of view, even at a small angle.

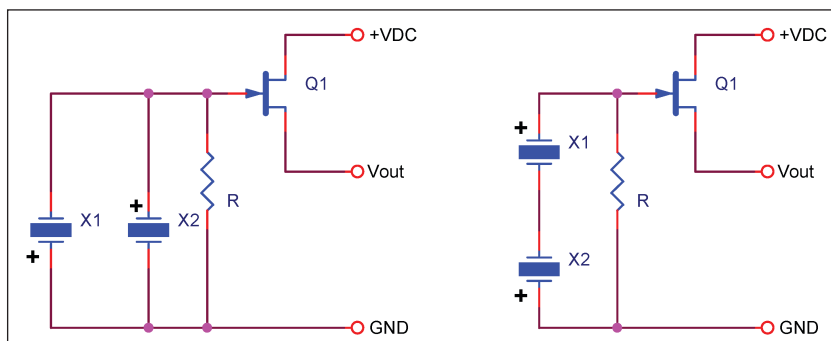


FIGURE 2

Schematics of a typical pyroelectric sensor

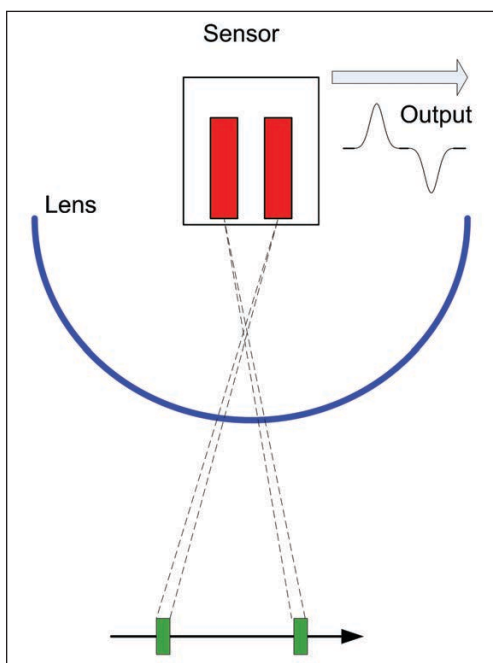


FIGURE 3

Principal internal layout of a PIR motion detector



ABOUT THE AUTHOR

George Novacek is a retired president of an aerospace company. He is a professional engineer with degrees in Automation and Cybernetics. George's dissertation project was a design of a portable ECG (electrocardiograph) with wireless interface. He's been contributing articles to *Circuit Cellar* since 1999. Contact him at gjnovacek@nexicom.net with "Circuit Cellar" in the subject line.

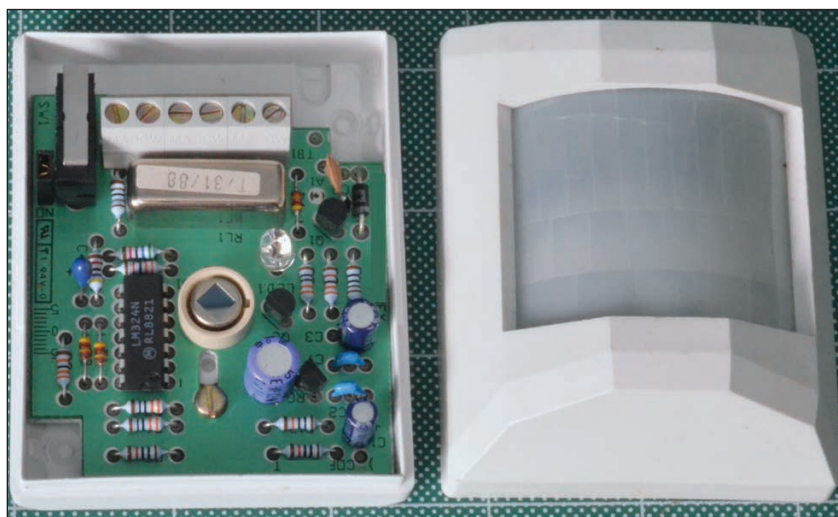


FIGURE 4

Passive infrared intrusion detector. Near the center of the device, note the plastic thermal ring surrounding the sensor.

Many commercially available Fresnel lenses provide both horizontal and vertical segments, whereas some are designed with fewer segments for a narrow field of view or for long-distance detection. Some detectors claim up to 140-degree detection view, but at 70 degrees off the center axis, the IR beam arrives at the sensor at too obtuse an angle to provide good sensitivity. A wider view is commonly achieved by arranging two sensors at 90 degrees to each other. This renders a reliable 180-degree angle of view. I purchased several devices for external security lighting claiming a 270-degree view, but none of them actually achieved that.

Figure 4 shows the inside of a PIR intrusion detector I designed some time ago. Nowadays, ICs containing all the needed circuitry are available from several vendors

[4] [5]. Nevertheless, to explain the PIRs' inner workings I'll describe the discrete circuits I utilized at the time. Before I discuss the circuitry, consider the close-fitting, albeit not hermetically sealed enclosure. Mounting holes and openings for wires allow it to breathe. Also notice the plastic ring, called the thermal ring, surrounding the sensor. It isolates the sensor from internal air movement.

THE SENSOR CIRCUITRY

The circuitry is uncomplicated and can be built with a single quad operational amplifier, such as the old faithful Texas Instruments LM324 or most modern operational amplifiers. **Figure 5** is a simplified schematic diagram in which, for clarity, I omitted some frequency-band-limiting filter components, pulse discrimination and relays with their drivers. The pulses generated by the pyroelectric sensor in response to the movement of humans range from approximately 0.1 Hz to 10 Hz. This is not critical, but suppressing frequencies outside this range avoids false triggering.

As mentioned previously, the pyroelectric sensor contains an integral FET, operating as a source follower to provide low impedance output. U1A amplifies the signal and sends it to a window comparator comprising U1B and U1C. The comparator outputs are OR-ed and buffered by U1D for further processing. U2 is a three-terminal regulator, such as the ubiquitous 780x. To avoid false alarms, it is vital to keep the power supply stable, especially within the detector's operating frequency band of 0.1 Hz to 10 Hz.

Modification of this basic PIR design for

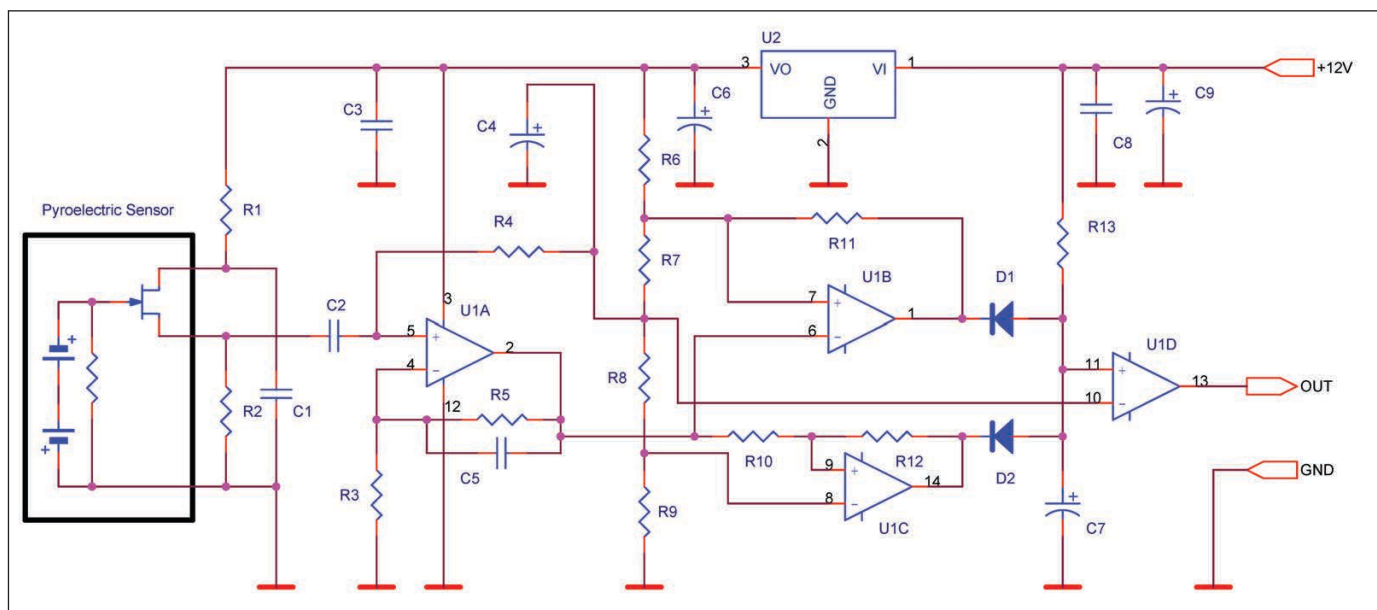
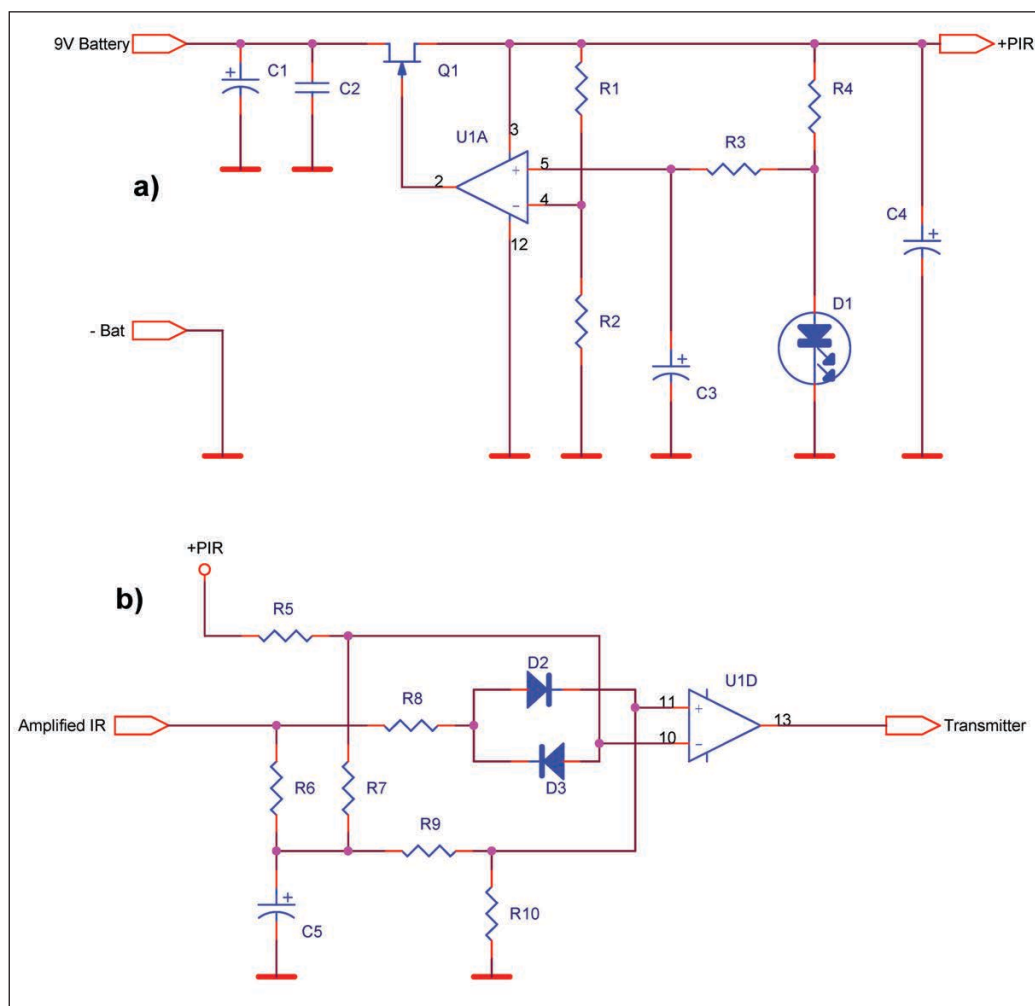


FIGURE 5

Shown there is a simplified diagram of a "wired" PIR intrusion.

**FIGURE 6**


PIR modifications for battery operation. The voltage regulator (a) and the window comparator (b)

battery operation was, at the time, a challenge. The circuit with LM324 drew around 8 mA (relays not counted), and would rapidly drain the battery. A typical 9 V alkaline battery has roughly 500 mA-hours capacity. I tested this, as no battery manufacturer published their battery capacity below about 10 mA current drain. Still, at least 1-year battery life was the design requirement.

To reduce the current, I replaced the LM324 with a programmable quad op amp LM146 and modified the voltage regulator as shown in **Figure 6a**. It relies on a JFET Q1 as a series regulator, together with one of the LM146 amplifiers. A stable, low-power voltage reference presented the biggest challenge. I tested several devices and established that a specific red LED provided a sufficiently stable reference voltage at merely 2.5 μ A current. (Being driven from stable regulator output helped.) Several storage capacitors throughout the circuit ensured the internal voltage stability during the RF transmitter operation.

Having given up one op amp for the power supply, I redesigned the window comparator as shown in **Figure 6b**. The circuit is essentially what's known as a bit slicer. C5 averages the

amplified signal of the PIR sensor, while the surrounding resistors and diodes determine the window size. The average current draw of the end product was approximately 14 μ A, including regular 477 MHz transmission bursts reporting status every 4 minutes. The 9 V alkaline battery lasted more than 1 year.

As one would expect, present day integrated circuits do all the processing transparently to the designer. Their current draw is higher, but they operate at a lower voltage. This, combined with huge capacity batteries such as LiPo batteries (not available at the time of my design), make new designs quick and efficient. Nevertheless, I am sure my old discrete circuits illustrate to you how the PIR detectors work. 

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

References [1] through [5] as marked in the article can be found there.

RESOURCES

Texas Instruments | www.ti.com

The Darker Side

The Art of Voltage Probing

Scope Savvy

Using the right tool for the right job is a basic tenet of electronics engineering. In this article, Robert explores one of the most common tools on an engineer's bench: the oscilloscope probe, and in particular the voltage measurement probe. He looks at the different types of voltage probes as well as the techniques to use them effectively and safely.

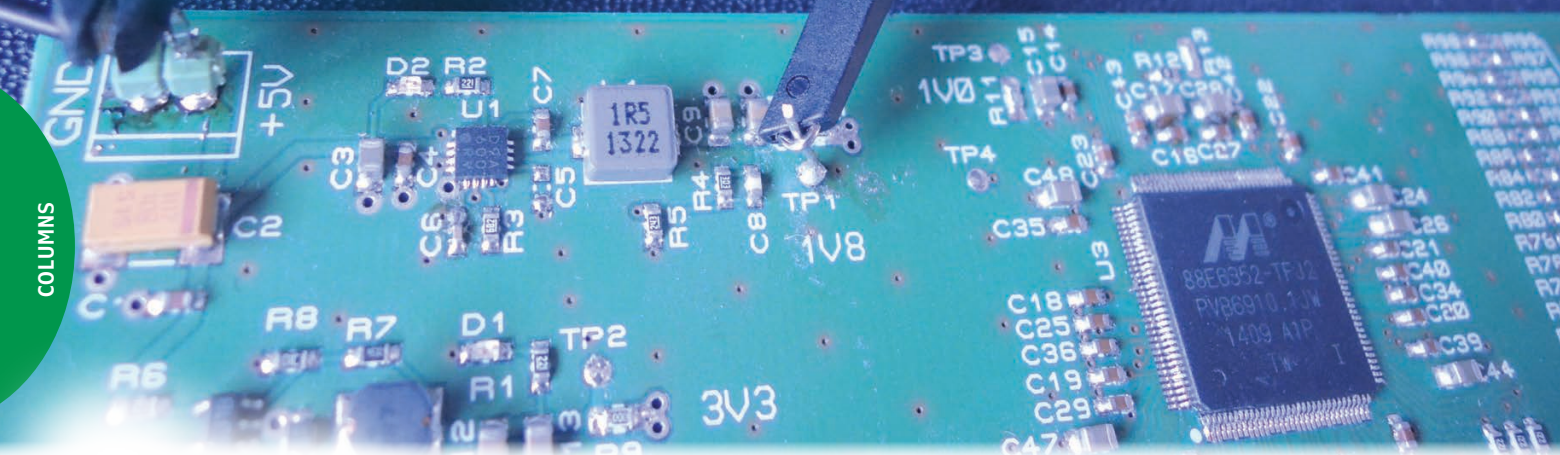


FIGURE 1

A classic 1:10 passive probe connected to measure noise on a 1.8 V power rail.

By
Robert Lacoste

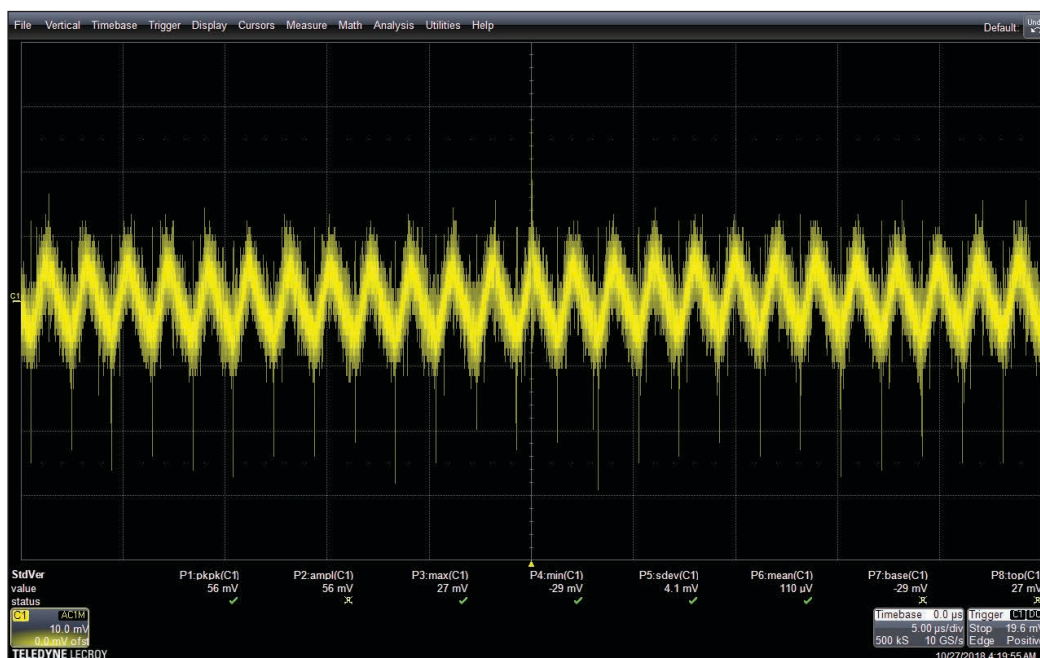
Welcome to the Darker Side. Have you ever tried to catch a needle with ski gloves? Difficult, isn't it? In a similar way, I'm sure you know you should use the right tools when measuring your electronic systems. If not, you may either fail to measure anything, or worse, think that you have measured something correctly but got a completely wrong result. And wrong measurements may lead to wrong decisions. This month I'm focusing on a very common test accessory that is often the culprit in many measurement errors: the oscilloscope probe, specifically the voltage measurement probe.

A BASIC EXAMPLE

Let's start with a very simple example. Imagine that you are in front of your latest project and want to measure the noise on one of its power supply rails, using your oscilloscope. Not rocket science, right? I did this test for you. I had on my desk a prototype of a Gbit Ethernet switch we designed some

months ago. This board includes a 5 V to 1.8 V DC-DC converter built around a Rohm Semiconductor BD9A600MUV synchronous buck converter chip. As illustrated in **Figure 1**, I simply connected the tip of the scope probe to the 1.8 V test point on the board. As you know, such a probe also has a ground lead with an alligator clip, which I simply connected to the 0 V power supply input. I did this test using the standard 500 MHz 1:10 passive probe that is provided with a high-performance 1 GHz Teledyne LeCroy WaveRunner 610Zi oscilloscope. For sure this probe and scope are adequate for measuring signals of a few megahertz. I switched on the scope, and got the plot shown in **Figure 2**. Here the vertical scale is 10 mV/division. The plot shows a 20 mV sawtooth oscillation, plus transient spikes of up to 40 mV peak to peak. So, this power supply has about 60 mV of noise, doesn't it?

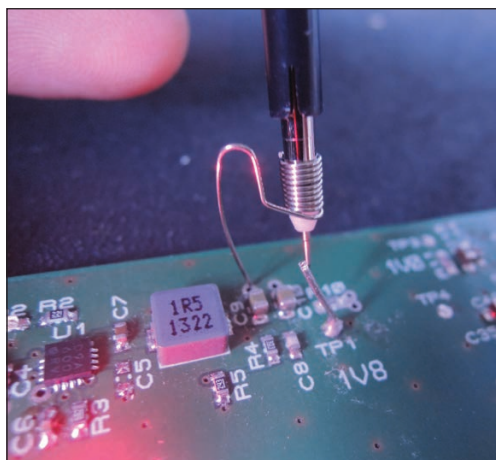
Wrong. Let's repeat the same measurement, this time without the long ground clip wire and instead using a very short ground connection.

**FIGURE 2**

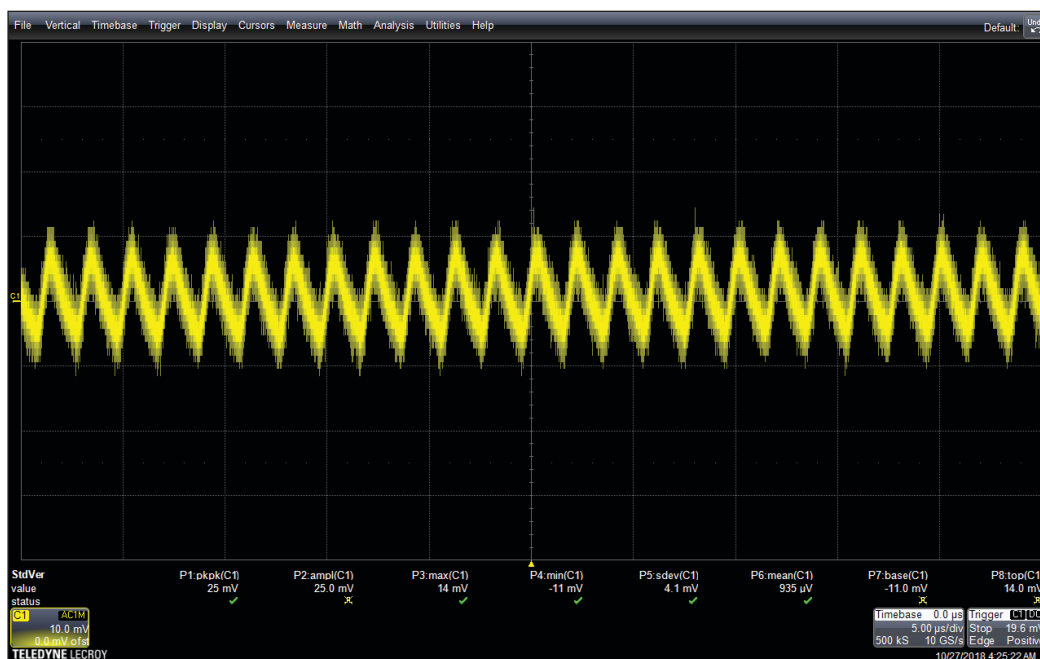
Here is the noise measured as 20 mV of sawtooth noise plus 40 mV of transient spikes.

To do this, you will need the small accessory kit supplied with your probe. You will usually find small spring-shaped wires designed to be fitted directly on the ground tube at the end of the probe (**Figure 3**). (If you don't have one, you can easily build one with a short length of wire.) I then did the measurement again, with the same scope and same probe, on the same test point, but connecting the ground to the nearest accessible grounded pad. The difference on the plot is remarkable (**Figure 4**). There is still 20 mV of sawtooth noise, but the transient spikes are gone!

What does this mean? Simply that these spikes aren't present on the 1.8 V power rail at all! They are artifacts, caused by the long

**FIGURE 3**

A much better ground connection, with as short as possible connection of the probe ground body to the nearest ground.

**FIGURE 4**

Measurement with the test setup illustrated in Figure 3, with no more transient spikes.

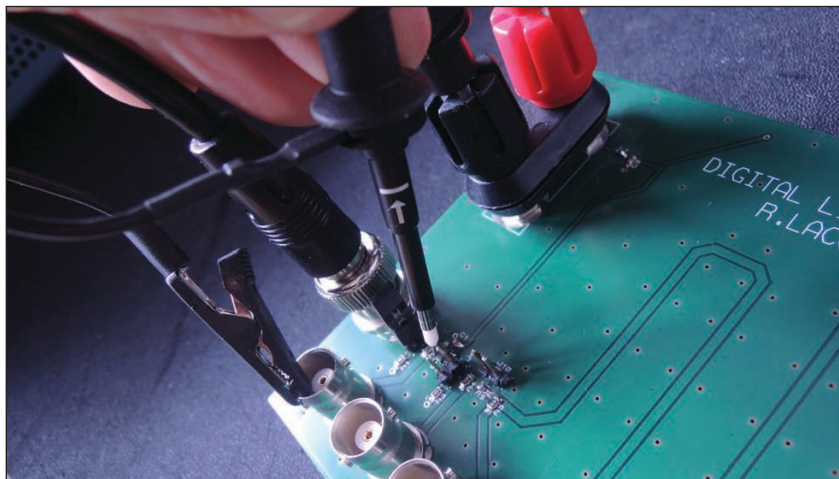


FIGURE 5

Here I measured the output of a fast logic gate with a standard probe and long ground wire.

ground connection. Looking back at Figure 1, you will understand why. The area between the ground wire and the probe is, in fact, a loop antenna that grabs all nearby signals. Here, the fast transitions in the DC-DC converter MOSFET are probably generating impulse magnetic fields, which were “received” by this unintended antenna.

This first example illustrates the first rule for using voltage probes: Always use the shortest possible connection for both the signal to be tested and also the ground!

FAST SIGNALS?

My first example was a very low-frequency application, and you saw that, even there, long or improper ground connections can have nasty effects. When the frequency

is increasing, or when signals have faster transitions (which is in fact the same), such improper ground connections could also jeopardize the measured signal itself. Why? Simply because a long ground wire acts as an inductor. And this inductor might alter the measurement.

To show you this phenomenon, I used a test board I had on hand, which basically includes only a fast 1.8 V logic gate. I drove it with a 50 MHz clock from a lab generator. I then connected the same 500 MHz 1:10 probe to the logic gate output, first with the standard long ground wire clipped to a nearby ground (**Figure 5**). I configured the scope to measure the rise time of the logic signal, and the result is given in **Figure 6**. Here this rise time is measured at 1.7 ns, but the signal shows a large overshoot and long oscillations.

Can you guess what happens if the long ground wire is replaced by a short one, still using the same probe, as shown in **Figure 7**? You guessed correctly. The oscillations and overshoot nearly disappear (**Figure 8**). The rise time is, however, still in the same order of magnitude, here 1.8 ns. Using short ground connections is also a must when dealing with fast signals or high frequencies. The overshoot and oscillations were in fact not present on the signal, but were generated by the inductance of the ground connection.

PROBE TYPES

A good probe must give a true measurement of the signal, and must not change the signal itself. However, there are plenty of different measurement situations, so not all probes share the same design. **Figure 9** shows the most common probe types.

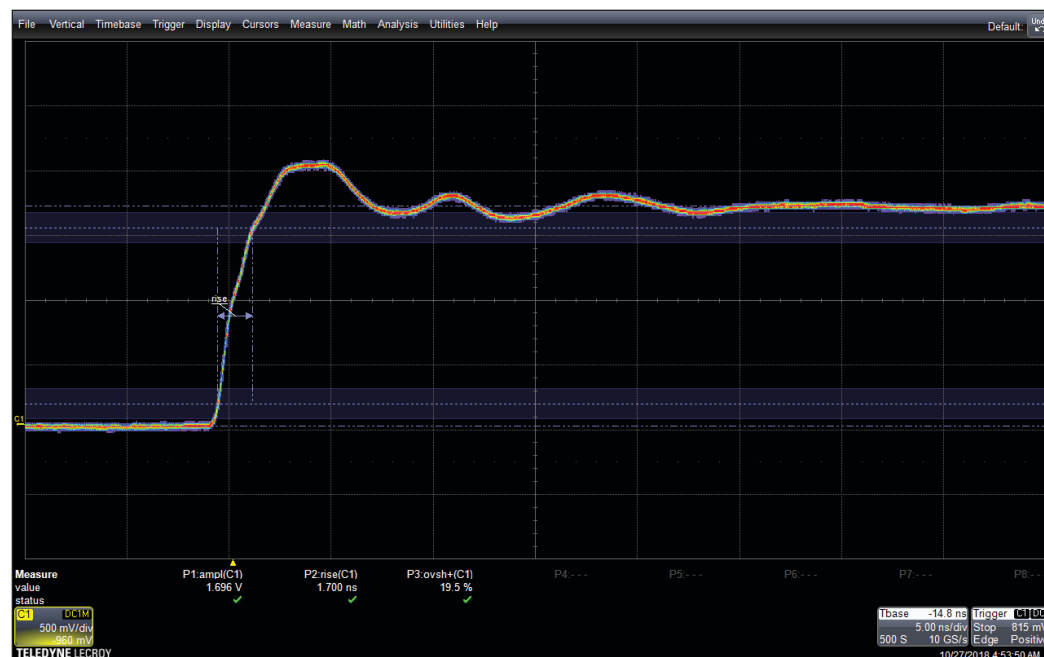


FIGURE 6

With standard probe and long ground wire, strong oscillations are visible. Rise time is measured at 1.7 ns.

Passive Probe: The simplest probe is the 1:1 probe, which is more or less a simple shielded wire, sometimes equipped with a small serial resistor. Such a probe is connected to a high impedance input of the oscilloscope (usually 1 M Ω). Therefore, the probe impedance is 1 M Ω , and grabs only a very small current on the signal. The problem with this type of probe is the parasitic capacitance applied on the circuit—around 25 pF is not uncommon for a 1 m shielded cable. A capacitance of this magnitude drastically limits the probe bandwidth to a maximum of 20 MHz. The 25 pF of capacitance are also connected to the circuit under test, and this may change its behavior dramatically as well. For these reasons, 1:1 probes are used rarely.

The most common probe is then the 1:10 probe, which, like the 1:1 probe, is passive, but limits the drawbacks of the 1:1 probe. This type of probe, which I used so far in this article, includes a 9 M Ω serial resistor. This resistor, associated with the 1 M Ω impedance of the scope input, forms a voltage divider with a 1/10 ratio. So, the sensitivity of the scope is reduced by the same amount. The good news is that the bandwidth of such a probe can be significantly higher than that of 1:1 probes, because the capacitance of the shielded cable can be accounted for in the voltage divider network, thanks to an extra adjustable compensation capacitor. When this capacitor is set to exactly 10 times the capacitance of the cable plus the capacitance of the scope input, then the effect of all these capacitors is drastically reduced. The only remaining capacitance is that of the probe tip itself, which could be a couple of picofarads. This explains why all probes above 20 MHz

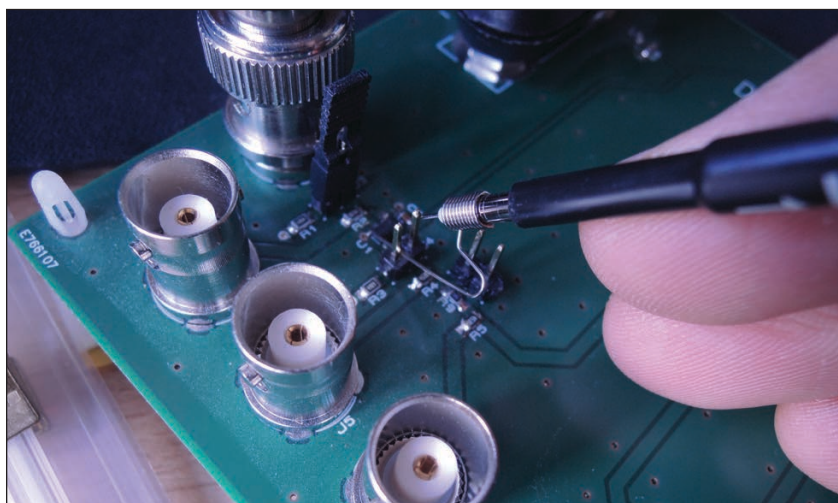


FIGURE 7

The same measurement with a short wire connection.

are 1:10 probes, and why these probes have an adjustment screw. Take care to tune this screw using the calibrator signal from the scope, as its optimal setting varies when using a different scope.

Active Probe: Now, how do we measure above 500 MHz, or signals faster than a few nanoseconds? To go higher in frequency, impedance-matching becomes a concern. The best solution is to use properly matched cables and impedances. This is not possible, however, using a 1 M Ω input, and is why most high-performance scopes have a selectable 1 M Ω or 50 Ω input impedance. If you switch it to 50 Ω , you can use a coaxial cable with a 50- Ω characteristic impedance, and the signal will not be distorted up to very high frequencies. Regardless of its length, a 50- Ω cable connected to a 50- Ω load is still a 50 Ω load, without any parasitic capacitance or inductance.

This solution is practical if you want to

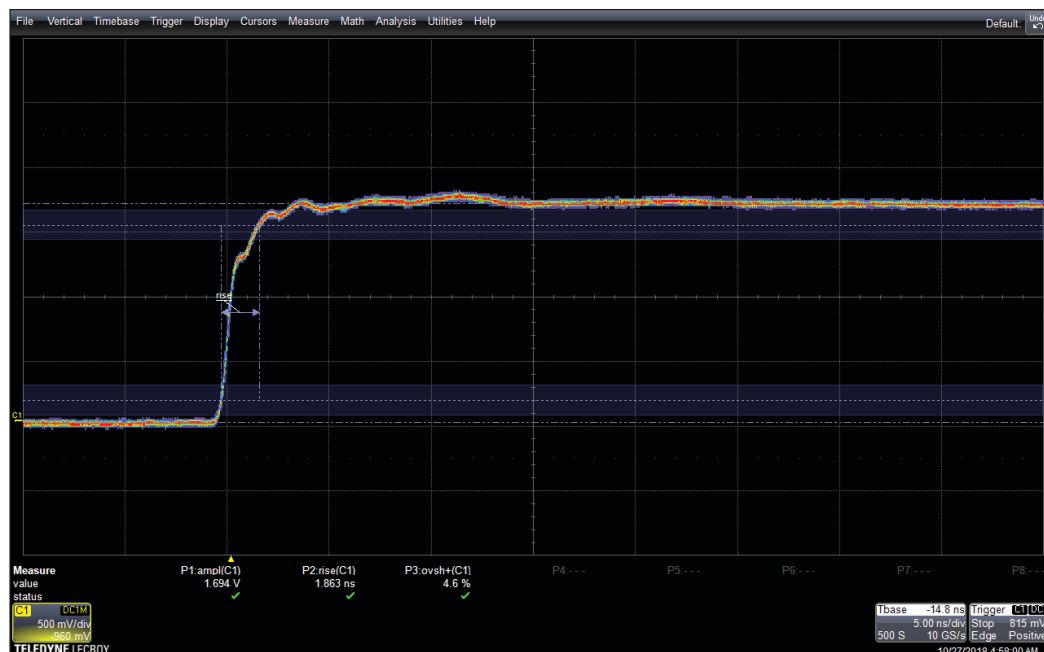


FIGURE 8

With the same 1:10 passive probe but with a short ground connection, the oscillations nearly disappear. Rise time stays similar here at 1.8 ns.

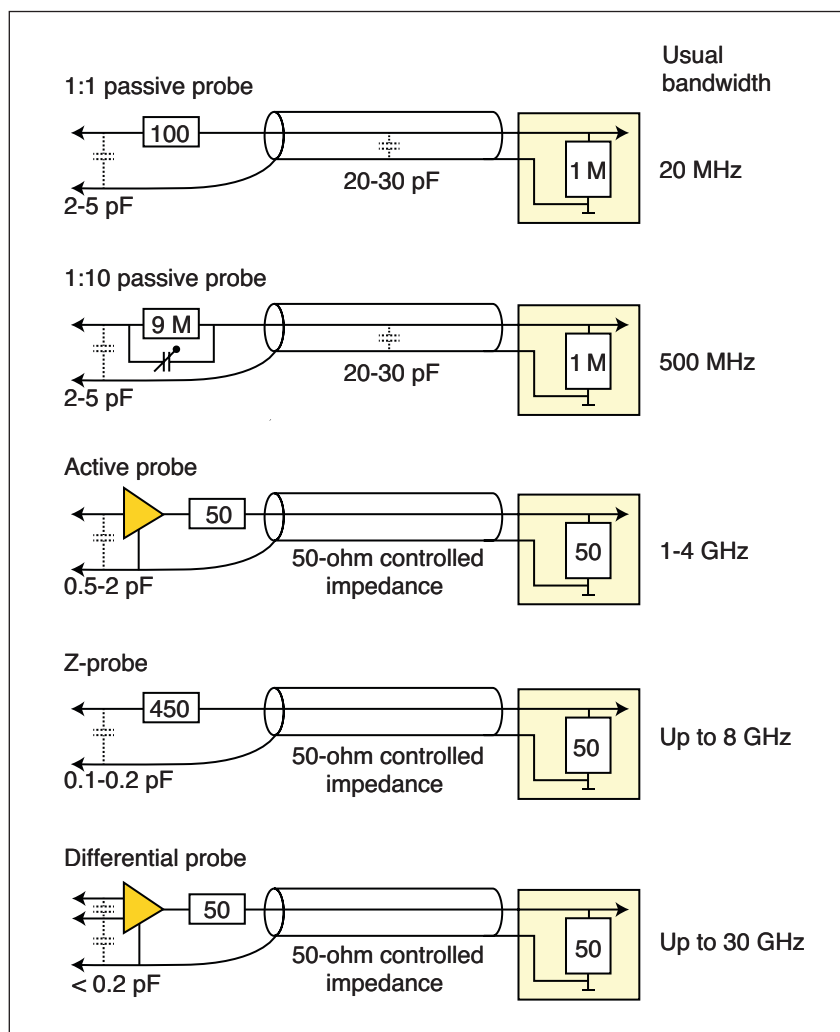


FIGURE 9

Not all probes are equal. Here are the five more common kinds of voltage measurement probes.

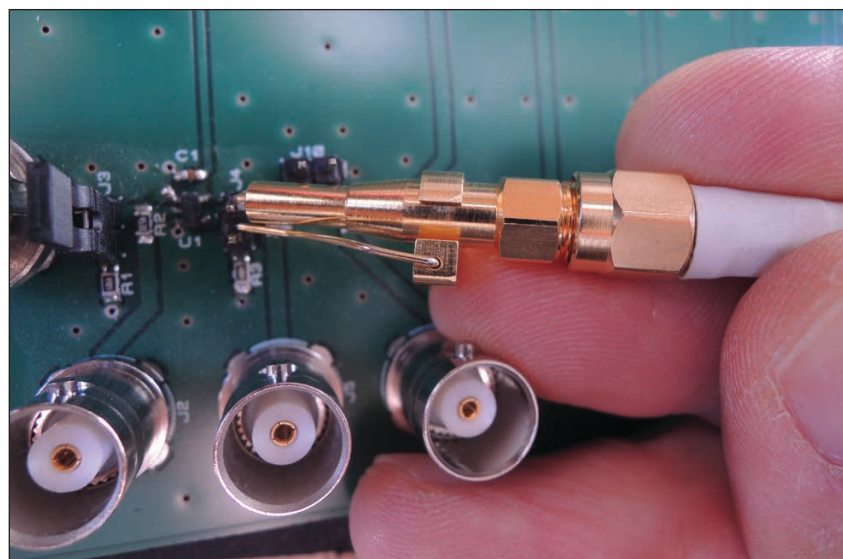


FIGURE 10

A "Z0" passive probe is nothing more than a 450- Ω resistor at the end of a 50- Ω coaxial cable. Here I used a Tektronix P6150, but a similar probe can be home-built. Note the short ground connection.

connect the scope to a circuit providing a 50- Ω output. However, you can't "probe" a signal running on a board with a 50- Ω impedance probe without overloading it. That's why there are active probes. An active probe includes a fast preamplifier, usually fitted in the probe tip itself. Its input impedance is very high, but its output impedance is 50 Ω , allowing it to be connected to a 50- Ω scope input. This type of probe is a very efficient tool, as its input capacitance can be small, down to 0.5 pF or so. Active probes up to 4 GHz are commercially available, though they aren't cheap.

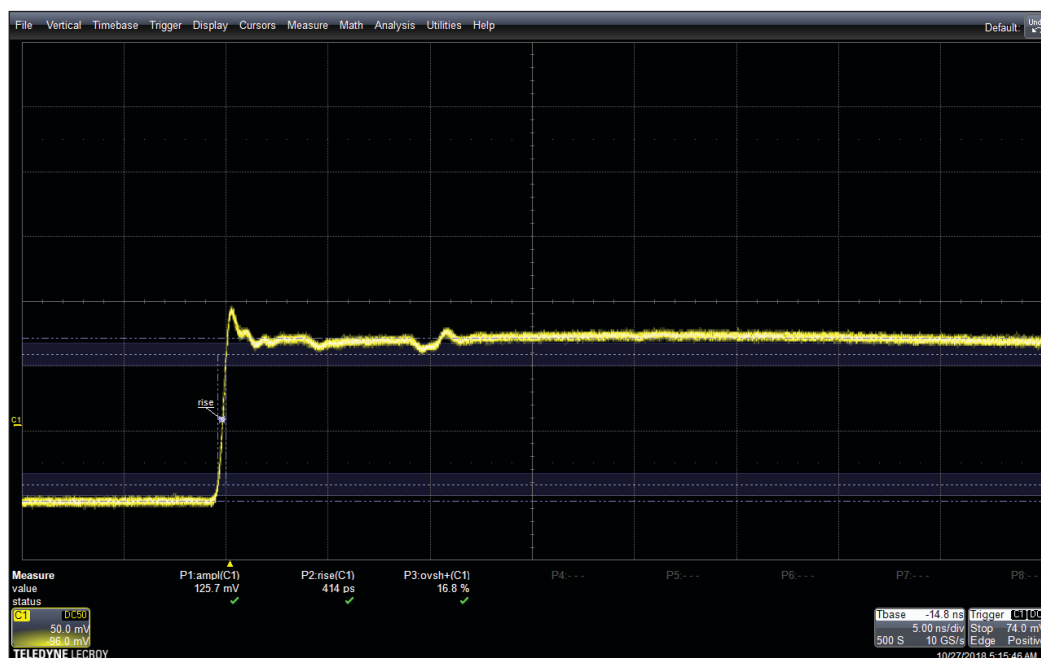
"Z0" Probe: Another very interesting probe option is the so-called "Z0" probe, which is nothing more than a 450- Ω small resistor at the end of a 50- Ω cable (Figure 9). Like the 1:10 high impedance probe, it acts as a 1/10 voltage divider when connected to a 50- Ω scope input. However, it can provide very high bandwidths—up to 8 GHz—as everything is impedance-matched. The downside is a significant load on the circuit under test, precisely 500 Ω , but this isn't usually a concern for high frequency applications.

Note that you can easily build this type of probe. Take a good 50- Ω cable, and solder a 450- Ω SMT resistor to its end. If you don't have a 450- Ω resistor on hand, a 470- Ω resistor can be substituted, but will produce a small ratio error. With such a homemade Z0 probe you will not get performance equal to that of a professional one, but a bandwidth of 1 GHz or so is easily achieved.

Differential Probe: The last type of probe I want to introduce is also the most powerful: the active high frequency differential probe. Like the active probe described above, it includes a built-in preamplifier and is designed to be connected to a 50- Ω input; In addition, it has two differential inputs, which enable it to directly measure the voltage difference between two signals, without any ground or common voltage issue. These probes are particularly well suited to differential signals like Ethernet, PCI or USB. Thanks to their structure, such differential probes are available up to 30 GHz and more. A warning: They should not be confused with another kind of differential probe—low speed insulated differential probes—which are designed for measurements on high-voltage systems. The concept is the same, but the design and applications differ.

TO THE BENCH AGAIN

Let's go back to my example of a 1.8 V fast logic gate. Remember that I measured a rise time of 1.7 ns using a 500 MHz 1:10 passive probe (Figure 8). Why not do the same measurement with a high-performance Z0 probe? I grabbed a Tektronix P6150 probe,

**FIGURE 11**

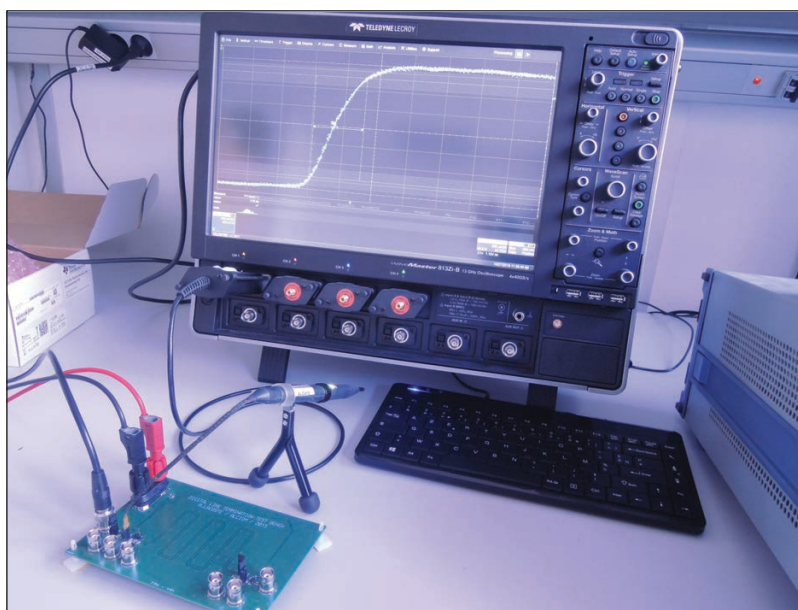
With the Z0 probe, the measurement changes completely. Now the rise time is measured at only 414 ps!

and connected it to the same test point. **Figure 10** shows this passive probe. The ground connection is very short, providing a bandwidth up to 9 GHz and a parasitic capacitance as low as 0.15 pF. Similar models are available from Keysight, such as the HP-54006A. I switched the input impedance of my scope to 50 Ω , and the result was pretty impressive (**Figure 11**). Some limited ringing is visible, but the rise time is now measured at only 414 ps! This indicates that the first measurement, done with a 500 MHz passive probe, was in fact the measurement of the probe rise time, and not the circuit rise time.

Why not do a last test with an even higher performance configuration? Well, here I must admit that I am lucky. First, I have unlimited access to my company's lab, and second, we have some serious toys there. I moved to another desk and switched on our monster scope, a 13 GHz, 4 × 40 GS/s, WaveMaster 813Zi-B, also from Teledyne LeCroy (**Figure 12**). I also grabbed a 6 GHz WaveLink D610 differential probe from the same supplier, and connected it to the output of the same 1.8 V logic gate. The resulting measurement is shown in **Figure 13**. Here the horizontal scale is 200 ps per division (yes, picoseconds), and the rise time is now evaluated at only 315 ps. There are no overshoots, no strange behaviors. That's why such a scope and such a differential probe cost a fortune!

WRAPPING UP

I know that most of you will not have access to such high-end equipment, but I wanted to show you that choosing the right probe for a given voltage measurement must

**FIGURE 12**

A last test with a 13 GHz scope and a high-end differential active probe.

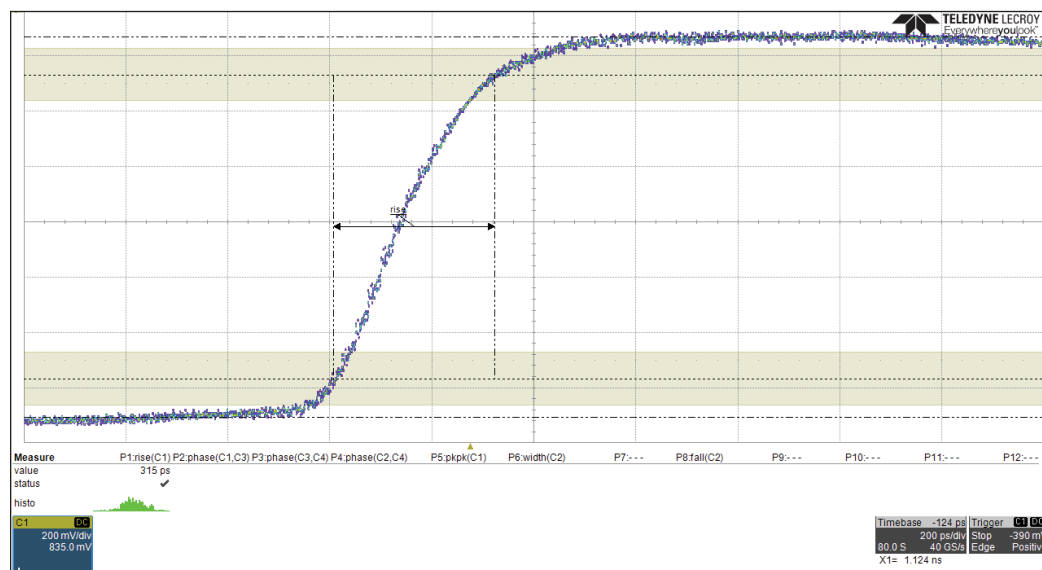


ABOUT THE AUTHOR

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.

FIGURE 13

If you have never looked at a 40 GS/s scope, here is the result. The horizontal scale is 200 ps/div, and the rise time is measured at 315 ps.



be done with care. Using it properly is also really important, even for simple tasks.

The first fundamental rule: Always use short ground connections. Throw away the long ground leads with alligator clips that are supplied with the probe, and use very short ground wires. If you have any doubt, check if you get the same results with both options.


Remember also that 1:1 passive probes are useless above 10 or 20 MHz, or for signals faster than some 100 ns. Good 1:10 passive probes could be used up to 500 MHz, but only with ultra-short grounds.

As explained, so-called Z0 probes are also very efficient, provided that the circuit can tolerate a load of 500 Ω . These probes can be homebuilt very easily with a length of 50- Ω cable and a 450 Ω SMT resistor as a tip. As noted above, a 470- Ω resistor will work too.

Finally, if you are lucky enough to find reasonably priced high-frequency active probes or high-speed differential probes, don't hesitate and buy them.

You can now see that there are probes for every application. Just for fun, I can't resist showing you a picture of an old but quite unusual probe we bought for a specific project some years ago—a Tektronix P6015 high-voltage probe (**Figure 14**). Exotic isn't it? This one is designed to work up to 40,000 V. The funniest part is that this probe is supplied with a can of fluorocarbon high-voltage dielectric fluid, with which the probe should be filled when operating above 13,000 V. Quite specialized, but useful from time to time.

A last word: The "art of probing" has been covered in depth by analog gurus far better than myself. The links provided on *Circuit Cellar's* article materials webpage are absolutely mandatory. They include readings from Jim Williams, Bob Dobkin and Bob Pease. You will learn a lot. There is even a nice 42-minute video on that subject on YouTube by Bob Pease and experts from Tektronix and TI. It's called "What's All This Scope Probe Stuff, Anyhow?"

In my next column, I will continue in the same spirit with discussions and experiments on current probing. In the meantime, just play with your scope and have fun! 

**FIGURE 14**

An example of a quite exotic old probe, a 40-kV rated Tektronix P6015.

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

RESOURCES

Analog Devices | www.analog.com

Keysight | www.keysight.com

ROHM Semiconductor | www.rohm.com

Tektronix | www.tektronix.com

Teledyne Lecroy | www.teledynelecroy.com

Texas Instruments | www.ti.com

From the Bench

Tinkering with Time

Protocols and Programming

By
Jeff Bachiochi

**FIGURE 1**

Time zone boundaries are flexible, shifted locally to keep territories from being divided wherever possible.

Many embedded systems need to make use of synchronized time information. In this article, Jeff explores the history of time measurement and how it has led to NTP and other modern technologies for coordinating universal date and time. Using Arduino and the Espressif System's ESP32, Jeff then goes through the steps needed to enable an embedded system to request, retrieve and display the synchronized date and time to a display.

It's been said that "If you are on time, you're late," or "To be early is to be on time." It's all relative. If you go to a meeting and people are already there, you feel as if you are late. If you are the first to arrive, you wonder if you've got the schedule wrong, and then you check your watch or phone for the time. Time can be troublesome for us, because the present is an ever-changing instant where the past meets the future. We cruise through life when all players reference the same moment, but should we become out of sync, the ride gets bumpy.

We can imagine that in humanity's early times the first concepts of time were cyclic periods—like day/night, seasons and life/death. Our fundamental measurement of a day directly relates to our life and history, and seems to tie all nature together. But what about those activities that occur within the confines of each day? Some way of defining the parts of a day were needed. At the time, we had one division—day/night—with most considering the start of a day to be daybreak or sunrise, and the start of night to be sunset.

Since daytime was directly related to the sun's position, the day could be divided into two parts based on whether the sun was rising in the sky or falling back toward the horizon. Observing the sun's shadow gave way to the first sundials, which provided a visual indication of time relative to sunrise and sunset without physical divisions. One such division of the day was religious in origin: canonical hours or periods

of fixed prayer at regular intervals were defined in monastic communities. At that time, our understanding of the sky was astrological and not astronomical. The latter would eventually define the breakdown of a day into hours, minutes and seconds.

For the most part, the hour was a variable concept. Around the 14th century, 12 was chosen as a practical division of the day (and the night) into equal parts. It was the most convenient number for dividing into fractions because it's divisible by 2, 3 and 4—thus giving us the 24-hour day we use today. Without the sun, sundials were worthless, so other means of recording the passage of time were invented, including water, candles and weights. These and early mechanical clocks of the 16th century were not accurate, because their mechanisms were essentially unregulated. It wasn't until the next century that the pendulum gave the mechanical clock accuracy to within 1 minute a day. Today, we have access to extremely accurate clocks. Atomic clocks measure an atom's fluctuating energy levels to produce an accuracy of ± 1 second in over a billion years.

TIME KEEPER

The International Bureau of Weights and Measures (called Bureau International des Poids et Mesures or BIPM in France) is an intergovernmental organization that was established to oversee measurement science and measurement standards. One important role for the BIPM is maintaining the accuracy of worldwide time of day. It combines, analyzes and averages

| | |
|------------------|-----------------------------------|
| UTC-04:00 (AST) | Puerto Rico and US Virgin Islands |
| UTC-05:00 (EST) | Atlantic coast |
| UTC-06:00 (CST) | Gulf Coast to the Great Lakes |
| UTC-07:00 (MST) | Mountain states |
| UTC-08:00 (PST) | Pacific coast states |
| UTC-09:00 (AKST) | Alaska |
| UTC-10:00 (HST) | Hawaii |

TABLE 1

Time zone offsets are listed here for the US daylight saving times have an additional offset of 1 hour and must be accounted for locally.

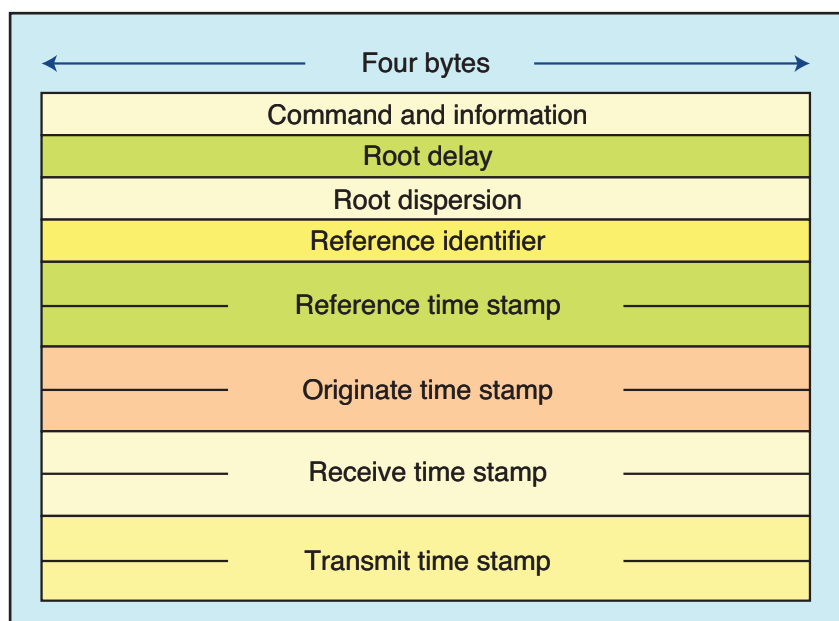


FIGURE 2

This is the format of the 48 byte packet sent to and from NTP servers. We can get away with sending a packet of “zero” data, except for the first byte as a request. A received packet will contain the total seconds since the Epoch located in the first four bytes of the Transmit Timestamp.

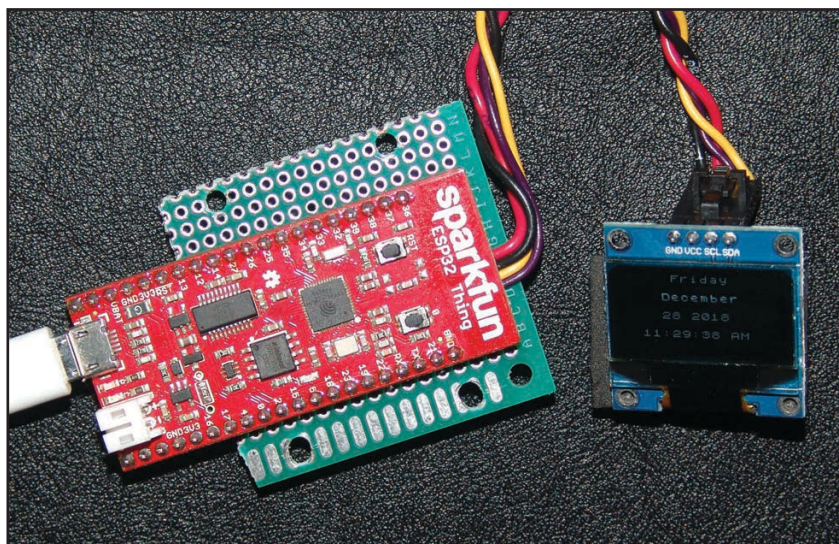


FIGURE 3

I’m using a Sparkfun ESP32 Thing, containing a USB port, Li-ion battery charger and ESP32 (the next generation ESP8266). The display seen here is only 1” wide, but can handle 8 lines of 16 characters.

the official atomic time standards of member nations around the world, to create a single, official Coordinated Universal Time (UTC). The Royal Observatory, Greenwich, England was chosen as the reference point to define the Universal day, counted from 0 hours at mean midnight, as used on the island since 1847. By 1884, the Greenwich Meridian was used for two-thirds of all charts and maps as their Prime Meridian. The world is divided into 24 time zones, each 15 degrees in width (24 hr/360 degrees). However, as shown in **Figure 1**, time zone boundaries are shifted to prevent a country from being needlessly split into separate zones.

All time on earth is related to the official time in Greenwich, England by denoting a time zone offset. Current civil time can be determined by adding or subtracting the UTC offset (number of hours and minutes). This ranges from UTC-12:00 in the west to UTC+14:00 in the east. **Table 1** lists those offsets that relate to the United States.

The US spans seven time zones. When a time zone uses daylight saving time, the ST for Standard Time is replaced by DT indicating Daylight Saving Time. Daylight Saving Time increases the regional offset by 1:00, and was implemented to shift daylight activities during the longer summer hours. Daylight Saving Time is a local shift that must be handled locally, and as such does not affect the UTC in any way.

In my youth I recall the phone company providing a number you could call to hear the current time. The first radio station, WWV in Colorado, morphed into National Institute of Standards and Technology (NIST), whose broadcast focused on developing frequency standards and eventually broadcasting time and frequency information on the 2.5-, 5-, 10-, 15- and 20-MHz shortwave bands. Today, the time is available almost everywhere, and that time is synchronized to the UTC, all thanks to the Internet.

NATIONAL STANDARD TIME

The Network Time Protocol (NTP) is used to synchronize our clocks via the Internet. The NTP architecture, protocol and algorithms provide a nominal accuracy of tens of milliseconds on WANs, sub-milliseconds on LANs, and sub-microseconds using a precision time source such as a cesium oscillator or GPS receiver. Reliability is assured by redundant tiered servers and diverse network paths. The “NTP pool” is a dynamic collection of networked computers that volunteer to provide highly accurate time via the NTP to clients (like us) worldwide. We can use one of the NTP pool servers to get UTC information. Although using the NTP protocol will assure the accuracies listed above, this is

often unnecessary and overly complicated for those applications that are only interested in whole-second times for RTC (Real Time Clocks). SNTP, a simplified subset of the NTP protocol, generally is sufficient for our needs.

SNTP uses a UDP connection to send a datagram or packet, as opposed to a TCP connection. The basic transaction is simple. We send an SNTP data structure as a UDP packet using port 123 to the server. The time server (one of the NTP pool) then sends back an SNTP data structure as a UDP packet. That's it! The structure of the datagram consists of four 32-bit words (4×32 bits = 128 bits or 16 bytes), followed by four 64-bit time stamps (4×64 bits = 256 b or 32 bytes) as shown in **Figure 2**. There can be optional data, but we won't need it. In fact, we need only to worry about the first byte of the (16 bytes + 32 bytes = 48 bytes) datagram to make a request.

This is set according to RFC 4330:

LeapsecondInformation 2 bits = "00"
disregarded

VersionNumber 3 bits = "100" 4
MODE 3 bits = "011" Client

Therefore:

First Byte = "00100011" or 0x23

The returned datagram will be in the same format. The time stamps we sent as zeros could have been used to determine the actual propagation delay in the message trip, to calculate an accurate sub-second synchronized time. We are not concerned with that level of accuracy. However, we do want to get the time from the server, and that will be populated in the last of the four time stamps in the reply. So how does this time stamp relate to the present second, minute, hour, day, month and year? Sad to say, it does not specify any of those. This time stamp gives the number of seconds from 0:00 on 1 January 1900.

Once we have this "elapsed time," we will need to figure out the rest. So, for now let's just be content to get this number. To do that I'm going to use the Sparkfun's ESP32 Thing board. This board shown in **Figure 3** contains the Espressif's ESP32 and all supporting hardware necessary to connect to your home wireless network and to be programmed using the Arduino IDE. This is the "next gen" ESP8266 and has 4 MB of external flash memory, compared to the ESP8266's 4 Mb.

You can visit the ESP32 support for Arduino on Espressif's github page [1]. In the Arduino IDE, begin installation by clicking on the File tab and selecting Preferences. Enter <https://>

```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
ets Jun  8 2016 00:22:57

rst:0x10 (RTCWDT_RTC_RESET),boot:0x13
(SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_
drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:808
load:0x40078000,len:6084
load:0x40080000,len:6696
entry 0x400802e4
Booting
Ready
IP address: 192.168.1.242
```

LISTING 1

After you compile and download the program to the ESP32 Thing, you can click on the spyglass icon to open the IDE's serial monitor and you will see the information shown here.

```
const int EEST = +3; // Eastern European Summer Time
const int EET = +2; // Eastern European Time
const int CEST = +2; // Central European Summer Time
const int CET = +1; // Central European Time
const int WES = +1; // Western European Summer Time
const int WET = +0; // Western European Time
const int GMT = +0; // Greenwich Mean Time
const int EGT = -1; // East Greenland Time
const int EDT = -4; // Eastern Daylight Time
const int EST = -5; // Eastern Standard Time
const int CDT = -5; // Central Daylight Time
const int CST = -6; // Central Standard Time
const int MDT = -6; // Mountain Daylight Time
const int MST = -7; // Mountain Standard Time
const int PDT = -7; // Mountain Daylight Time
const int PST = -8; // Mountain Standard Time
```

```
String timeZone = "Eastern Daylight Time";
long timeZoneOffset = EDT * 3600; // Zone * sec/hr
int updateInterval = 60; //secs
```

```
unsigned long lastTime;
unsigned long presentTime;
unsigned long myEpoch=timeZoneOffset * -1;
```

```
int year = 0;
int month = 0;
int day = 0;
int DOW = 0;
int hour = 0;
int minute = 0;
int second = 0;
int lastSecond = 0;
boolean newSecond = false;
```

continued

LISTING 2

After making the additions shown here, when the application is compiled, downloaded and run, we should get a display of time and date, which increments each second.

dl.espressif.com/dl/package_esp32_index.json into the “Additional Board Manager URLs” field and close Preferences. Then, click on the Tools tab and open Boards > Open Boards Manager, install the esp32 platform and close the manager. I connected my ESP32 Thing to a USB port, restarted the IDE so it can find the port, selected Sparkfun ESP32 thing from Tools>Boards, and the proper

port from Tools>Port in preparation to begin writing an application.

Sparkfun’s ESP32 thing is just one of the ESP32 platforms that were just added to the IDE. Many times, along with a platform’s installation, is a number of example programs. We can begin with one of these to get started. As a bonus here, we’ll choose the BasicOTA (Over the Air) example from the File>Examples>Examples for the Sparkfun ESP32 thing>ArduinoOTA>BasicOTA. You will need to modify two lines in the application:

```
const char* ssid = “.....”;
const char* password =
“.....”;
```

Then, substitute your home wireless network’s SSID and password—and remove the slashes from these two REM’d statements—to enable the basic password protection:

```
// ArduinoOTA.
setPassword(“admin”);
// ArduinoOTA.setPasswordHash
(“21232f297a57a5a743894a0e4a801
fc3”);
```

Save the application as “FTB343,” so you won’t overwrite the example. Compile and download the program to the ESP32 Thing. Upon download success you can click on the *spyglass* icon to open the IDE’s serial monitor and you will see the information in **Listing 1**.

Listing 2: Continued

```
void loop()
{
  ArduinoOTA.handle(); // handles the OTA stuff
  //
  presentTime = millis(); // seconds since reboot
  newSecond = false; // not a new second
  if((presentTime - lastTime)>=1000)
    // if seconds has advanced by at least 1000ms
  {
    myEpoch++;
    // we have a new second so add one to the total
    lastTime = lastTime + 1000;
    // new advanced time to look for
    newSecond = true; // we have a new second
  }
  if(newSecond) // if we have a new second
  {

    calculateTime(myEpoch+timeZoneOffset);
    // calculate date and time based on the total + timeZoneOffset
    displayTime(); // now display the calculated information
  }
}
```

NTP initialization:

```
const char timeServer[] = “time.nist.gov”; // time.nist.gov NTP server
const int NTP_PACKET_SIZE = 48; // NTP time stamp is in the first 48 bytes of the message
byte packetBuffer[ NTP_PACKET_SIZE]; //buffer to hold incoming and outgoing
```

Call to NTP functions at the end of loop():

```
if(!(myEpoch % updateInterval) & newSecond) // limit how often we request an NTP packet (required)
{
  sendNTPpacket(timeServer); // send an NTP packet to a time server
  Serial.print(“requesting packet from “);
  for(int i=0; i<sizeof(timeServer) - 1; i++)
  {
    Serial.write(timeServer[i]);
  }
  Serial.println();
}
if(Udp.parsePacket()) // NTP response packet could come at any time
{
  Serial.println(“packet received”);
  decodePacket(); // get information and sync it
}
```

LISTING 3

Code for synchronizing with NTP

Note that debug information is dumped from the ESP32 at every reset. Following this dump, you'll see the result of our print statements, Booting, Ready and IP address. This indicates the Wi-Fi connection is established at local port 192.168.1.242. If you close and reopen the Arduino IDE, you'll find an addition in the Port>List: *esp3232-[MAC] at 192.168.1.242 (Sparkfun ESP32 Thing)*. If you choose this Port selection you will be downloading any compiled application, using Wi-Fi instead of the USB. You can monitor this through the now unused USB port with a serial terminal program such as RealTerm. The Arduino's IDE serial terminal can't be used during a Wi-Fi download. You can always choose to continue to use the USB port to program, in a wired configuration. OTA is just a bonus example you may find helpful, and was a consequence of getting the UDP client installed.

SOME INITIALIZING

With the UDP client, we can send a request to one of the servers in the NTP pool. Rather than adding another library—`NTPClient`, for example—I'll add some raw code to send

```
unsigned long sendNTPpacket(const char * address)
{
    // set all bytes in the buffer to 0
    memset(packetBuffer, 0, NTP_PACKET_SIZE);
    // fill packet with 'zero' data
    // Initialize values needed to form NTP request
    // (see RFC 4330 for details on the packets)
    packetBuffer[0] = 0b00100011; // LI, Version, Mode
    packetBuffer[1] = 0;          // Stratum, or type of clock
    packetBuffer[2] = 0;          // Polling Interval
    packetBuffer[3] = 0;          // Peer Clock Precision
    Udp.beginPacket(address, 123);
                               // NTP requests are to port 123
    Udp.write(packetBuffer, NTP_PACKET_SIZE);
    Udp.endPacket();
}
```

LISTING 4

The request to NTP is simple, especially when UDP has already been installed. Shown here is the support routine to do the request.

```
void decodePacket()
{
    // read the data from the packet
    Udp.read(packetBuffer, NTP_PACKET_SIZE); // read the packet into the buffer
    //the timestamp starts at byte 40 of the received packet and is four bytes,
    // or two words, long. First, extract the two words:
    unsigned long highWord = word(packetBuffer[40], packetBuffer[41]);
    unsigned long lowWord = word(packetBuffer[42], packetBuffer[43]);
    // combine the four bytes (two words) into a long integer
    // this is NTP time (seconds since Jan 1 1900):
    unsigned long secsSince1900 = highWord << 16 | lowWord;
    Serial.print("Seconds since Jan 1 1900 = ");
    Serial.println(secsSince1900);
    // now convert NTP time into everyday time:
    Serial.print("Unix time = ");
    // Unix time starts on Jan 1 1970. In seconds, that's 2208988800:
    const unsigned long seventyYears = 2208988800UL;
    // subtract seventy years:
    unsigned long epoch = secsSince1900 - seventyYears;
    // print Unix time:
    Serial.println(epoch);
    // print the hour, minute and second:
    Serial.print("The UTC time is ");
    // UTC is the time at Greenwich Meridian (GMT)
    calculateTime(epoch);
    displayTime();
    Serial.println("My Time Zone is " + String(timeZone));
    Serial.println("It's time offset is " + String(timeZoneOffset) + " seconds");
    Serial.print("The EDT time is "); // UTC is the time at Greenwich Meridian (GMT)
    calculateTime(epoch + timeZoneOffset);
    myEpoch = epoch; // update myEpoch so I can use the system clock until I sync again
}
```

LISTING 5

Shown here is the support routine for decoding packets.

out a UDP datagram. But first we need to initialize a few things and add some code to keep track of the passing seconds. This is based on `millis()` the system variable that keeps track of the number of milliseconds that have elapsed since the application started. An alternative might be to have a hardware RTC attached, but I chose to do this all in software. Note that I've added some constants that define the time zones for the US.

Since our time reference will be UTC, we need to adjust this for our specific time zone by adding a `timeZoneOffset = zone × 3,600 seconds/zone` (one hour of seconds

for each time zone). It is suggested that you keep your NTP requests to no less than 10 requests/minute, or 6 seconds), I'll be using 1 minute here. While epoch will become the total seconds count received from a request, `myEpoch` is initialized to the negative of `timeZoneOffset`. As a result, as the application begins, `myEpoch` is initialized to the negative of my EDT `timeZoneOffset` (in seconds) (**Listing 2**). With these additions—when the application is compiled, downloaded and run—we should get a display of time and date, which increments each second.

You will note that while `myEpoch` is initialized to local zero (`timeZoneOffset × -1`), the actual date/time displayed is Thursday January 1, 1970 00:00:00. That is the standard Epoch that is used as the UNIX=0 reference. The number of seconds given by NTP will be referenced to this date/time. So naturally, zero seconds should display as this date/time. Now with all this background material in place, we can finally get in sync with NTP (**Listing 3**).

Remember, it is suggested that you do not make requests to an NTP server more often than every 6 seconds. Faster requests are considered flooding, and you may get rejected from further services. In this application, I set my `updateInterval` to once an hour. In reality, your system clock should be able to keep pretty good time without having to sync for many hours, or even just once a day. Please note here I would rather use an interrupt to ensure I don't miss updating each second. However, if you don't have any code that prevents `loop()` from executing at least once a second, all will be fine. The request to NTP is simple, especially when UDP has already been installed. **Listing 4** is the support routine to do the request.

Note that once our packet has been filled with zero data, we only need to change the first byte in the packet before sending it. Since UDP communication has no handshaking, a response isn't guaranteed. However, we can test for a response, then get data once it comes in. If a packet is received, we can find what we are looking for in packet bytes 40:43. **Listing 5** is the support routine for decoding packets.

The four packet bytes of interest are the first four of the 8-byte Transmit Timestamp shown in Figure 2. The first four are seconds, and the last four are fractions of a second. We don't need the sub-second accuracy of the last four bytes. The first four bytes show seconds, up to 4,294,967,295 seconds. Based on the Epoch of 1970, rollover will happen in 2038. Y2K anyone? While proposals have been made, presently there is no accepted solution. As in many proposals, backwards compatibility is the issue.

```

1 //-----
2 // Program Name, so I
3 //-----
4 String signOn = "ftb343
5 //-----
6 // Start of the ESP32
7 //-----
8 #include <WiFi.h>
9 #include <ESP8266.h>
10 #include <WiFiUdp.h>
11 #include <ArduinoOTA.h>
12 // A UDP instance to le
13 WiFiUDP Udp;
14 const char* ssid = "Fro
15 const char* password =
16 unsigned int localPort
17 //-----
18 // End of the ESP32 OT
19 //-----

```

```

ets Jun 8 2016 00:22:57
rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:808
load:0x40078000,len:8084
load:0x40080000,len:8996
entry 0x400800e4
Booting
ftb343_OTA_SyncTime
IP address: 192.168.1.242
Starting connection to server...
UDP Ready
Thursday January 1 1970 00:00:01
Thursday January 1 1970 00:00:02
Thursday January 1 1970 00:00:03
Thursday January 1 1970 00:00:04
Thursday January 1 1970 00:00:05
Thursday January 1 1970 00:00:06
Thursday January 1 1970 00:00:07
Thursday January 1 1970 00:00:08
Thursday January 1 1970 00:00:09
Thursday January 1 1970 00:00:10
Thursday January 1 1970 00:00:11
Thursday January 1 1970 00:00:12
packet received
Seconds since Jan 1 1900 = 3748188196
Unix time = 1539199380
The UTC time is Wednesday October 10 2018 19:22:48
My Time Zone is Eastern Daylight Time
The Time Zone offset is -14400 seconds
The EDT time is Wednesday October 10 2018 15:22:48
Wednesday October 10 2018 15:22:47
Wednesday October 10 2018 15:22:46
Wednesday October 10 2018 15:22:45
Wednesday October 10 2018 15:22:44
Wednesday October 10 2018 15:22:43
Wednesday October 10 2018 15:22:42
Wednesday October 10 2018 15:22:41
Wednesday October 10 2018 15:22:40
Wednesday October 10 2018 15:22:39
Wednesday October 10 2018 15:22:38
Wednesday October 10 2018 15:22:37
Wednesday October 10 2018 15:22:36
Wednesday October 10 2018 15:22:35
Wednesday October 10 2018 15:22:34
Wednesday October 10 2018 15:22:33
Wednesday October 10 2018 15:22:32
Wednesday October 10 2018 15:22:31
Wednesday October 10 2018 15:22:30
Wednesday October 10 2018 15:22:29
Wednesday October 10 2018 15:22:28
Wednesday October 10 2018 15:22:27
Wednesday October 10 2018 15:22:26
Wednesday October 10 2018 15:22:25
Wednesday October 10 2018 15:22:24
Wednesday October 10 2018 15:22:23
Wednesday October 10 2018 15:22:22
Wednesday October 10 2018 15:22:21
Wednesday October 10 2018 15:22:20
Wednesday October 10 2018 15:22:19
Wednesday October 10 2018 15:22:18
Wednesday October 10 2018 15:22:17
Wednesday October 10 2018 15:22:16
Wednesday October 10 2018 15:22:15
Wednesday October 10 2018 15:22:14
Wednesday October 10 2018 15:22:13
Wednesday October 10 2018 15:22:12
Wednesday October 10 2018 15:22:11
Wednesday October 10 2018 15:22:10
Wednesday October 10 2018 15:22:09
Wednesday October 10 2018 15:22:08
Wednesday October 10 2018 15:22:07
Wednesday October 10 2018 15:22:06
Wednesday October 10 2018 15:22:05
Wednesday October 10 2018 15:22:04
Wednesday October 10 2018 15:22:03
Wednesday October 10 2018 15:22:02
Wednesday October 10 2018 15:22:01
Wednesday October 10 2018 15:22:00
Wednesday October 10 2018 15:21:59
Wednesday October 10 2018 15:21:58
Wednesday October 10 2018 15:21:57
Wednesday October 10 2018 15:21:56
Wednesday October 10 2018 15:21:55
Wednesday October 10 2018 15:21:54
Wednesday October 10 2018 15:21:53
Wednesday October 10 2018 15:21:52
Wednesday October 10 2018 15:21:51
Wednesday October 10 2018 15:21:50
Wednesday October 10 2018 15:21:49
Wednesday October 10 2018 15:21:48
Wednesday October 10 2018 15:21:47
Wednesday October 10 2018 15:21:46
Wednesday October 10 2018 15:21:45
Wednesday October 10 2018 15:21:44
Wednesday October 10 2018 15:21:43
Wednesday October 10 2018 15:21:42
Wednesday October 10 2018 15:21:41
Wednesday October 10 2018 15:21:40
Wednesday October 10 2018 15:21:39
Wednesday October 10 2018 15:21:38
Wednesday October 10 2018 15:21:37
Wednesday October 10 2018 15:21:36
Wednesday October 10 2018 15:21:35
Wednesday October 10 2018 15:21:34
Wednesday October 10 2018 15:21:33
Wednesday October 10 2018 15:21:32
Wednesday October 10 2018 15:21:31
Wednesday October 10 2018 15:21:30
Wednesday October 10 2018 15:21:29
Wednesday October 10 2018 15:21:28
Wednesday October 10 2018 15:21:27
Wednesday October 10 2018 15:21:26
Wednesday October 10 2018 15:21:25
Wednesday October 10 2018 15:21:24
Wednesday October 10 2018 15:21:23
Wednesday October 10 2018 15:21:22
Wednesday October 10 2018 15:21:21
Wednesday October 10 2018 15:21:20
Wednesday October 10 2018 15:21:19
Wednesday October 10 2018 15:21:18
Wednesday October 10 2018 15:21:17
Wednesday October 10 2018 15:21:16
Wednesday October 10 2018 15:21:15
Wednesday October 10 2018 15:21:14
Wednesday October 10 2018 15:21:13
Wednesday October 10 2018 15:21:12
Wednesday October 10 2018 15:21:11
Wednesday October 10 2018 15:21:10
Wednesday October 10 2018 15:21:09
Wednesday October 10 2018 15:21:08
Wednesday October 10 2018 15:21:07
Wednesday October 10 2018 15:21:06
Wednesday October 10 2018 15:21:05
Wednesday October 10 2018 15:21:04
Wednesday October 10 2018 15:21:03
Wednesday October 10 2018 15:21:02
Wednesday October 10 2018 15:21:01
Wednesday October 10 2018 15:21:00
Wednesday October 10 2018 15:20:59
Wednesday October 10 2018 15:20:58
Wednesday October 10 2018 15:20:57
Wednesday October 10 2018 15:20:56
Wednesday October 10 2018 15:20:55
Wednesday October 10 2018 15:20:54
Wednesday October 10 2018 15:20:53
Wednesday October 10 2018 15:20:52
Wednesday October 10 2018 15:20:51
Wednesday October 10 2018 15:20:50
Wednesday October 10 2018 15:20:49
Wednesday October 10 2018 15:20:48
Wednesday October 10 2018 15:20:47
Wednesday October 10 2018 15:20:46
Wednesday October 10 2018 15:20:45
Wednesday October 10 2018 15:20:44
Wednesday October 10 2018 15:20:43
Wednesday October 10 2018 15:20:42
Wednesday October 10 2018 15:20:41
Wednesday October 10 2018 15:20:40
Wednesday October 10 2018 15:20:39
Wednesday October 10 2018 15:20:38
Wednesday October 10 2018 15:20:37
Wednesday October 10 2018 15:20:36
Wednesday October 10 2018 15:20:35
Wednesday October 10 2018 15:20:34
Wednesday October 10 2018 15:20:33
Wednesday October 10 2018 15:20:32
Wednesday October 10 2018 15:20:31
Wednesday October 10 2018 15:20:30
Wednesday October 10 2018 15:20:29
Wednesday October 10 2018 15:20:28
Wednesday October 10 2018 15:20:27
Wednesday October 10 2018 15:20:26
Wednesday October 10 2018 15:20:25
Wednesday October 10 2018 15:20:24
Wednesday October 10 2018 15:20:23
Wednesday October 10 2018 15:20:22
Wednesday October 10 2018 15:20:21
Wednesday October 10 2018 15:20:20
Wednesday October 10 2018 15:20:19
Wednesday October 10 2018 15:20:18
Wednesday October 10 2018 15:20:17
Wednesday October 10 2018 15:20:16
Wednesday October 10 2018 15:20:15
Wednesday October 10 2018 15:20:14
Wednesday October 10 2018 15:20:13
Wednesday October 10 2018 15:20:12
Wednesday October 10 2018 15:20:11
Wednesday October 10 2018 15:20:10
Wednesday October 10 2018 15:20:09
Wednesday October 10 2018 15:20:08
Wednesday October 10 2018 15:20:07
Wednesday October 10 2018 15:20:06
Wednesday October 10 2018 15:20:05
Wednesday October 10 2018 15:20:04
Wednesday October 10 2018 15:20:03
Wednesday October 10 2018 15:20:02
Wednesday October 10 2018 15:20:01
Wednesday October 10 2018 15:20:00
Wednesday October 10 2018 15:19:59
Wednesday October 10 2018 15:19:58
Wednesday October 10 2018 15:19:57
Wednesday October 10 2018 15:19:56
Wednesday October 10 2018 15:19:55
Wednesday October 10 2018 15:19:54
Wednesday October 10 2018 15:19:53
Wednesday October 10 2018 15:19:52
Wednesday October 10 2018 15:19:51
Wednesday October 10 2018 15:19:50
Wednesday October 10 2018 15:19:49
Wednesday October 10 2018 15:19:48
Wednesday October 10 2018 15:19:47
Wednesday October 10 2018 15:19:46
Wednesday October 10 2018 15:19:45
Wednesday October 10 2018 15:19:44
Wednesday October 10 2018 15:19:43
Wednesday October 10 2018 15:19:42
Wednesday October 10 2018 15:19:41
Wednesday October 10 2018 15:19:40
Wednesday October 10 2018 15:19:39
Wednesday October 10 2018 15:19:38
Wednesday October 10 2018 15:19:37
Wednesday October 10 2018 15:19:36
Wednesday October 10 2018 15:19:35
Wednesday October 10 2018 15:19:34
Wednesday October 10 2018 15:19:33
Wednesday October 10 2018 15:19:32
Wednesday October 10 2018 15:19:31
Wednesday October 10 2018 15:19:30
Wednesday October 10 2018 15:19:29
Wednesday October 10 2018 15:19:28
Wednesday October 10 2018 15:19:27
Wednesday October 10 2018 15:19:26
Wednesday October 10 2018 15:19:25
Wednesday October 10 2018 15:19:24
Wednesday October 10 2018 15:19:23
Wednesday October 10 2018 15:19:22
Wednesday October 10 2018 15:19:21
Wednesday October 10 2018 15:19:20
Wednesday October 10 2018 15:19:19
Wednesday October 10 2018 15:19:18
Wednesday October 10 2018 15:19:17
Wednesday October 10 2018 15:19:16
Wednesday October 10 2018 15:19:15
Wednesday October 10 2018 15:19:14
Wednesday October 10 2018 15:19:13
Wednesday October 10 2018 15:19:12
Wednesday October 10 2018 15:19:11
Wednesday October 10 2018 15:19:10
Wednesday October 10 2018 15:19:09
Wednesday October 10 2018 15:19:08
Wednesday October 10 2018 15:19:07
Wednesday October 10 2018 15:19:06
Wednesday October 10 2018 15:19:05
Wednesday October 10 2018 15:19:04
Wednesday October 10 2018 15:19:03
Wednesday October 10 2018 15:19:02
Wednesday October 10 2018 15:19:01
Wednesday October 10 2018 15:19:00
Wednesday October 10 2018 15:18:59
Wednesday October 10 2018 15:18:58
Wednesday October 10 2018 15:18:57
Wednesday October 10 2018 15:18:56
Wednesday October 10 2018 15:18:55
Wednesday October 10 2018 15:18:54
Wednesday October 10 2018 15:18:53
Wednesday October 10 2018 15:18:52
Wednesday October 10 2018 15:18:51
Wednesday October 10 2018 15:18:50
Wednesday October 10 2018 15:18:49
Wednesday October 10 2018 15:18:48
Wednesday October 10 2018 15:18:47
Wednesday October 10 2018 15:18:46
Wednesday October 10 2018 15:18:45
Wednesday October 10 2018 15:18:44
Wednesday October 10 2018 15:18:43
Wednesday October 10 2018 15:18:42
Wednesday October 10 2018 15:18:41
Wednesday October 10 2018 15:18:40
Wednesday October 10 2018 15:18:39
Wednesday October 10 2018 15:18:38
Wednesday October 10 2018 15:18:37
Wednesday October 10 2018 15:18:36
Wednesday October 10 2018 15:18:35
Wednesday October 10 2018 15:18:34
Wednesday October 10 2018 15:18:33
Wednesday October 10 2018 15:18:32
Wednesday October 10 2018 15:18:31
Wednesday October 10 2018 15:18:30
Wednesday October 10 2018 15:18:29
Wednesday October 10 2018 15:18:28
Wednesday October 10 2018 15:18:27
Wednesday October 10 2018 15:18:26
Wednesday October 10 2018 15:18:25
Wednesday October 10 2018 15:18:24
Wednesday October 10 2018 15:18:23
Wednesday October 10 2018 15:18:22
Wednesday October 10 2018 15:18:21
Wednesday October 10 2018 15:18:20
Wednesday October 10 2018 15:18:19
Wednesday October 10 2018 15:18:18
Wednesday October 10 2018 15:18:17
Wednesday October 10 2018 15:18:16
Wednesday October 10 2018 15:18:15
Wednesday October 10 2018 15:18:14
Wednesday October 10 2018 15:18:13
Wednesday October 10 2018 15:18:12
Wednesday October 10 2018 15:18:11
Wednesday October 10 2018 15:18:10
Wednesday October 10 2018 15:18:09
Wednesday October 10 2018 15:18:08
Wednesday October 10 2018 15:18:07
Wednesday October 10 2018 15:18:06
Wednesday October 10 2018 15:18:05
Wednesday October 10 2018 15:18:04
Wednesday October 10 2018 15:18:03
Wednesday October 10 2018 15:18:02
Wednesday October 10 2018 15:18:01
Wednesday October 10 2018 15:18:00
Wednesday October 10 2018 15:17:59
Wednesday October 10 2018 15:17:58
Wednesday October 10 2018 15:17:57
Wednesday October 10 2018 15:17:56
Wednesday October 10 2018 15:17:55
Wednesday October 10 2018 15:17:54
Wednesday October 10 2018 15:17:53
Wednesday October 10 2018 15:17:52
Wednesday October 10 2018 15:17:51
Wednesday October 10 2018 15:17:50
Wednesday October 10 2018 15:17:49
Wednesday October 10 2018 15:17:48
Wednesday October 10 2018 15:17:47
Wednesday October 10 2018 15:17:46
Wednesday October 10 2018 15:17:45
Wednesday October 10 2018 15:17:44
Wednesday October 10 2018 15:17:43
Wednesday October 10 2018 15:17:42
Wednesday October 10 2018 15:17:41
Wednesday October 10 2018 15:17:40
Wednesday October 10 2018 15:17:39
Wednesday October 10 2018 15:17:38
Wednesday October 10 2018 15:17:37
Wednesday October 10 2018 15:17:36
Wednesday October 10 2018 15:17:35
Wednesday October 10 2018 15:17:34
Wednesday October 10 2018 15:17:33
Wednesday October 10 2018 15:17:32
Wednesday October 10 2018 15:17:31
Wednesday October 10 2018 15:17:30
Wednesday October 10 2018 15:17:29
Wednesday October 10 2018 15:17:28
Wednesday October 10 2018 15:17:27
Wednesday October 10 2018 15:17:26
Wednesday October 10 2018 15:17:25
Wednesday October 10 2018 15:17:24
Wednesday October 10 2018 15:17:23
Wednesday October 10 2018 15:17:22
Wednesday October 10 2018 15:17:21
Wednesday October 10 2018 15:17:20
Wednesday October 10 2018 15:17:19
Wednesday October 10 2018 15:17:18
Wednesday October 10 2018 15:17:17
Wednesday October 10 2018 15:17:16
Wednesday October 10 2018 15:17:15
Wednesday October 10 2018 15:17:14
Wednesday October 10 2018 15:17:13
Wednesday October 10 2018 15:17:12
Wednesday October 10 2018 15:17:11
Wednesday October 10 2018 15:17:10
Wednesday October 10 2018 15:17:09
Wednesday October 10 2018 15:17:08
Wednesday October 10 2018 15:17:07
Wednesday October 10 2018 15:17:06
Wednesday October 10 2018 15:17:05
Wednesday October 10 2018 15:17:04
Wednesday October 10 2018 15:17:03
Wednesday October 10 2018 15:17:02
Wednesday October 10 2018 15:17:01
Wednesday October 10 2018 15:17:00
Wednesday October 10 2018 15:16:59
Wednesday October 10 2018 15:16:58
Wednesday October 10 2018 15:16:57
Wednesday October 10 2018 15:16:56
Wednesday October 10 2018 15:16:55
Wednesday October 10 2018 15:16:54
Wednesday October 10 2018 15:16:53
Wednesday October 10 2018 15:16:52
Wednesday October 10 2018 15:16:51
Wednesday October 10 2018 15:16:50
Wednesday October 10 2018 15:16:49
Wednesday October 10 2018 15:16:48
Wednesday October 10 2018 15:16:47
Wednesday October 10 2018 15:16:46
Wednesday October 10 2018 15:16:45
Wednesday October 10 2018 15:16:44
Wednesday October 10 2018 15:16:43
Wednesday October 10 2018 15:16:42
Wednesday October 10 2018 15:16:41
Wednesday October 10 2018 15:16:40
Wednesday October 10 2018 15:16:39
Wednesday October 10 2018 15:16:38
Wednesday October 10 2018 15:16:37
Wednesday October 10 2018 15:16:36
Wednesday October 10 2018 15:16:35
Wednesday October 10 2018 15:16:34
Wednesday October 10 2018 15:16:33
Wednesday October 10 2018 15:16:32
Wednesday October 10 2018 15:16:31
Wednesday October 10 2018 15:16:30
Wednesday October 10 2018 15:16:29
Wednesday October 10 2018 15:16:28
Wednesday October 10 2018 15:16:27
Wednesday October 10 2018 15:16:26
Wednesday October 10 2018 15:16:25
Wednesday October 10 2018 15:16:24
Wednesday October 10 2018 15:16:23
Wednesday October 10 2018 15:16:22
Wednesday October 10 2018 15:16:21
Wednesday October 10 2018 15:16:20
Wednesday October 10 2018 15:16:19
Wednesday October 10 2018 15:16:18
Wednesday October 10 2018 15:16:17
Wednesday October 10 2018 15:16:16
Wednesday October 10 2018 15:16:15
Wednesday October 10 2018 15:16:14
Wednesday October 10 2018 15:16:13
Wednesday October 10 2018 15:16:12
Wednesday October 10 2018 15:16:11
Wednesday October 10 2018 15:16:10
Wednesday October 10 2018 15:16:09
Wednesday October 10 2018 15:16:08
Wednesday October 10 2018 15:16:07
Wednesday October 10 2018 15:16:06
Wednesday October 10 2018 15:16:05
Wednesday October 10 2018 15:16:04
Wednesday October 10 2018 15:16:03
Wednesday October 10 2018 15:16:02
Wednesday October 10 2018 15:16:01
Wednesday October 10 2018 15:16:00
Wednesday October 10 2018 15:15:59
Wednesday October 10 2018 15:15:58
Wednesday October 10 2018 15:15:57
Wednesday October 10 2018 15:15:56
Wednesday October 10 2018 15:15:55
Wednesday October 10 2018 15:15:54
Wednesday October 10 2018 15:15:53
Wednesday October 10 2018 15:15:52
Wednesday October 10 2018 15:15:51
Wednesday October 10 2018 15:15:50
Wednesday October 10 2018 15:15:49
Wednesday October 10 2018 15:15:48
Wednesday October 10 2018 15:15:47
Wednesday October 10 2018 15:15:46
Wednesday October 10 2018 15:15:45
Wednesday October 10 2018 15:15:44
Wednesday October 10 2018 15:15:43
Wednesday October 10 2018 15:15:42
Wednesday October 10 2018 15:15:41
Wednesday October 10 2018 15:15:40
Wednesday October 10 2018 15:15:39
Wednesday October 10 2018 15:15:38
Wednesday October 10 2018 15:15:37
Wednesday October 10 2018 15:15:36
Wednesday October 10 2018 15:15:35
Wednesday October 10 2018 15:15:34
Wednesday October 10 2018 15:15:33
Wednesday October 10 2018 15:15:32
Wednesday October 10 2018 15:15:31
Wednesday October 10 2018 15:15:30
Wednesday October 10 2018 15:15:29
Wednesday October 10 2018 15:15:28
Wednesday October 10 2018 15:15:27
Wednesday October 10 2018 15:15:26
Wednesday October 10 2018 15:15:25
Wednesday October 10 2018 15:15:24
Wednesday October 10 2018 15:15:23
Wednesday October 10 2018 15:15:22
Wednesday October 10 2018 15:15:21
Wednesday October 10 2018 15:15:20
Wednesday October 10 2018 15:15:19
Wednesday October 10 2018 15:15:18
Wednesday October 10 2018 15:15:17
Wednesday October 10 2018 15:15:16
Wednesday October 10 2018 15:15:15
Wednesday October 10 2018 15:15:14
Wednesday October 10 2018 15:15:13
Wednesday October 10 2018 15:15:12
Wednesday October 10 2018 15:15:11
Wednesday October 10 2018 15:15:10
Wednesday October 10 2018 15:15:09
Wednesday October 10 2018 15:15:08
Wednesday October 10 2018 15:15:07
Wednesday October 10 2018 15:15:06
Wednesday October 10 2018 15:15:05
Wednesday October 10 2018 15:15:04
Wednesday October 10 2018 15:15:03
Wednesday October 10 2018 15:15:02
Wednesday October 10 2018 15:15:01
Wednesday October 10 2018 15:15:00
Wednesday October 10 2018 15:14:59
Wednesday October 10 2018 15:14:58
Wednesday October 10 2018 15:14:57
Wednesday October 10 2018 15:14:56
Wednesday October 10 2018 15:14:55
Wednesday October 10 2018 15:14:54
Wednesday October 10 2018 15:14:53
Wednesday October 10 2018 15:14:52
Wednesday October 10 2018 15:14:51
Wednesday October 10 2018 15:14:50
Wednesday October 10 2018 15:14:49
Wednesday October 10 2018 15:14:48
Wednesday October 10 2018 15:14:47
Wednesday October 10 2018 15:14:46
Wednesday October 10 2018 15:14:45
Wednesday October 10 2018 15:14:44
Wednesday October 10 2018 15:14:43
Wednesday October 10 2018 15:14:42
Wednesday October 10 2018 15:14:41
Wednesday October 10 2018 15:14:40
Wednesday October 10 2018 15:14:39
Wednesday October 10 2018 15:14:38
Wednesday October 10 2018 15:14:37
Wednesday October 10 2018 15:14:36
Wednesday October 10 2018 15:14:35
Wednesday October 10 2018 15:14:34
Wednesday October 10 2018 15:14:33
Wednesday October 10 2018 15:14:32
Wednesday October 10 2018 15:14:31
Wednesday October 10 2018 15:14:30
Wednesday October 10 2018 15:14:29
Wednesday October 10 2018 15:14:28
Wednesday October 10 2018 15:14:27
Wednesday October 10 2018 15:14:26
Wednesday October 10 2018 15:14:25
Wednesday October 10 2018 15:14:24
Wednesday October 10 2018 15:14:23
Wednesday October 10 2018 15:14:22
Wednesday October 10 2018 15:14:21
Wednesday October 10 2018 15:14:20
Wednesday October 10 2018 15:14:19
Wednesday October 10 2018 15:14:18
Wednesday October 10 2018 15:14:17
Wednesday October 10 2018 15:14:16
Wednesday October 10 2018 15:14:15
Wednesday October 10 2018 15:14:14
Wednesday October 10 2018 15:14:13
Wednesday October 10 2018 15:14:12
Wednesday October 10 2018 15:14:11
Wednesday October 10 2018 15:14:10
Wednesday October 10 2018 15:14:09
Wednesday October
```


EPOCH TO EPOCH


When this application is first run, myEpoch begins from zero. All we need to do now is make myEpoch = "Epoch" (as received from NTP) and then the Date and Time will be displayed based on this synchronized epoch (**Figure 4**). That's it. But let's add one more item. To see the date and time, we are connected via USB, and it is displayed on our PC's serial terminal application. so, let's add a small display like the 0.96" 128x64 bit I2C OLED graphics display (<\$10). This has support for 16x8 characters using the U8x8lib.h library connected via I2C. This allows the application to be stand-alone.

A four-wire cable—including power, ground, clock and data (Arduino dedicated I2C pins)—is all you need to add to the microcontroller. While 8x8 character size is approximately 0.05", I like this display's small size and the fact that it will operate at 3.3 V or 5 V. Because of the 16-character horizontal limit, I need to break the date into two pieces, one for the month and one for the day and year. This all fits nicely displaying on the even numbered lines. To make it even more pleasing, space characters are added to center the displayed data. **Listing 6** is the code I added to use this display.

You can see the result in **Figure 5**. While the library offers a number of OLED functions, I used just one function for simplicity, drawing a glyph (character) at a column, row position. The padding at front and back of the data string clears out any old characters.

IT'S TIME

It's probably clear that most of the effort required for this application is in preliminary work and not in the actual requesting, retrieving and even displaying the synchronized date and time to an OLED display. While I have shown the important tasks as user-written functions, there are libraries already written to handle most functions. Unless you dig into the library code, however, you don't learn much, and it all seems like black magic. Once you understand how the tasks work, using the libraries can save you coding time.

I mentioned earlier one reason for choosing the Sparkfun ESP32 Thing was its inclusion of a Li-ion battery connector and the ability to recharge the battery via a USB connection. Since you can find small solar-cell 10 W USB chargers online for less than \$20, you can create a totally solar-powered Date and Time display. You will need to add some code for choosing an open Wi-Fi connection, or search for one. But your application will no doubt want to do more with the date and time than just display it. So, expand on what's offered here and let me know how you use this new-found tool! 

Initialization:

```
#include <U8x8lib.h>
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_PIN_NONE);
```

Setup():

```
u8x8.begin();
u8x8.setPowerSave(0);
u8x8.setFont(u8x8_font_chroma48medium8_r);
```

Support functions in displayTime():

```
printString(0, 0, DOW[dayOfWeek]);
printString(0, 2, MOY[month.toInt()]);
printString(0, 4, day + " " + year);
if(hour.toInt()>12)
{
    //pm
    hour=String(hour.toInt()-12);
    printString(0,6, hour + ":" + minute + ":" + second + " PM");
}
else
{
    //am
    if(hour.toInt()==00) hour="12";
    printString(0, 6, hour + ":" + minute + ":" + second + " AM");
}
```

DisplayOLED:

```
void printString(int col, int row, String myString)
{
    int pad = ((128/8) - myString.length())/2;
    for(int z=0; z<pad; z++)
    {
        u8x8.drawGlyph(col+z,row,32);
    }
    for(int z=pad; z<myString.length()+pad; z++)
    {
        u8x8.drawGlyph(col+z,row,myString.charAt(z-pad));
    }
    pad=z;
    for(int z=pad; z<16; z++)
    {
        u8x8.drawGlyph(col+z,row,32);
    }
}
```

LISTING 6

Shown here is the code I add to use the I2C OLED graphics display.

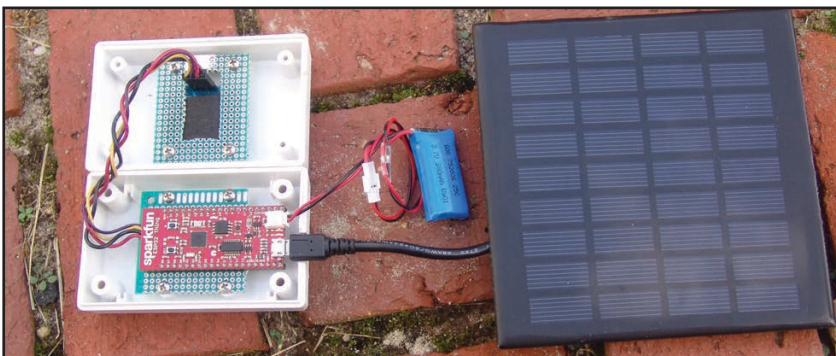


FIGURE 5

Mounted in a small enclosure, the project can be "solarized" by adding a solar module (<\$20) that is meant to charge USB devices through the USB port.

PRODUCT NEWS

CompactPCI Serial Board Delivers Four Gbit Ethernet Channels

MEN Micro has announced the G211X Ethernet interface card. It ensures fast data transmission with four X-coded M12 connectors connected to the backplane via an x4 link. The G211X is a new quad Ethernet card based on CompactPCI Serial. It can be used in combination with a CompactPCI Serial or CompactPCI PlusIO CPU board in a CompactPCI Serial or hybrid system. The four Gigabit Ethernet interfaces on the front panel are accessible via robust, X-coded M12 connectors.

All four interfaces are controlled by an Ethernet controller connected to the backplane via an x4 PCI Express connection. Each interface also supports a data transmission rate of 1 Gb/s—even if all four channels are used simultaneously. For better control, two LEDs each indicate the connection and activity status of the interfaces. The G211X is designed for the extended operating temperature and prepared for conformal coating for use in harsh and mobile environments, in particular for railway rolling stock applications.

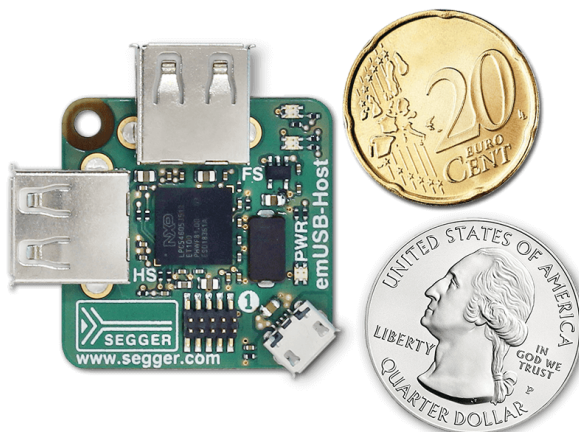
MEN Micro | www.menmicro.com



Tiny MCU-Based Development Platform Hosts Dual USB Ports

Segger Microcontroller has introduced emPower-USB-Host, a compact low-cost development board. With two USB host ports, many applications using USB peripherals can be realized with little effort. Precompiled applications for barcode and smartcard readers, as well as POS displays, LTE sticks and USB to LAN adapters are available for download, including complete projects for Embedded Studio with source code of these applications. The applications are using Segger's emUSB-Host software API, which makes accessing the different types of USB devices easy.

emPower-USB-Host uses the emLoad bootloader, pre-loaded into the flash of the MCU, to easily change applications in seconds using a USB flash drive. Development of custom applications is also supported. The board has a debug connector, providing full access to the NXP LPC54605J512 MCU with its Cortex-M4 core. Schematics and PCB layout of the board are available under a Creative Commons license. This way, the hardware can be used as a blueprint for custom devices using two USB host ports.



Segger Microcontroller
www.segger.com

i.MX ULL Processor-Based Board Drives Touch Panel Kit

MYIR has launched development board built around its MYC-Y6ULX computer-on-module, which runs Linux on NXP's low-power i.MX6 ULL SoC. The MYD-Y6ULX-HMI dev board

powers a new 7" capacitive touch-panel display called the MYD-Y6ULX-CHMI. The kit features a pre-installed Linux HMI stack and offers an optional MYB-Y6ULX-HMI-4GEXP IO wireless add-on board.

Announced a year ago, the 39 mm x 37 mm MYC-Y6ULX module offers four flavors of i.MX6 UL and three flavors of i.MX6 ULL, both of which are built on single Cortex-A7 cores. The new MYD-Y6ULX-HMI dev board uses a 528 MHz ULL variant of the module equipped with 256 MB DDR3 and 256 MB NAND flash. The RAM can optionally be expanded to up to 1 GB DDR3, and there's an option for 4 GB eMMC.

The module has a 140-pin, 1.0mm pitch "stamp hole" carrier board interface designed to work with either the standard MYD-Y6ULX development board or the new MYD-Y6ULX-HMI version designed for touch-panel applications.



MYIR | www.myirtech.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 adcopy@circuitcellar.com with your file.

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

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

ADVANCED CONTROL ROBOTICS
HANNO SANDER

Get it today, cc-webshop.com

ALL ELECTRONICS CORPORATION

Electronic and Electro-mechanical Devices, Parts and Supplies.
Many unique items.

We have what you need for your next project.

www.allelectronics.com
Free 96 page catalog 1-800-826-5432

CCS Inc.
THE SOLUTIONS YOU NEED FOR INTERNET CONNECTIVITY IN AN IoT WORLD.

EZ Web Lynx

- Capability to upload custom web pages
- View the condition of the 17 programmable I/O pins
- Change the device settings and view incoming data

Sku: 59101-694

OR

3.3V Embedded Ethernet Dev Kit
with easy to use C libraries for TCP/IP

www.ccsinfo.com/cc219

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

Circuit Cellar 2018 Archive

Order yours today
cc-webshop.com

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

Answer 1— The capacitance of the plates drops with increasing distance, so the voltage between them rises, because the charge doesn't change and the voltage is equal to the charge divided by the capacitance. At first, while the plate spacing is still small relative to their diameter. The capacitance is proportional to the inverse of the spacing, so the voltage rises linearly with the spacing. However, as the spacing becomes larger, the capacitance drops more slowly and the voltage rises at a lower rate as well.

While the plate spacing is small, the electric field is almost entirely directly between the two plates, with only minor "fringing" effects at the edges. Since the voltage rise is proportional to the distance in this regime, the electric field (in volts per meter) remains essentially constant. However, once the plate spacing becomes comparable to the diameter of the plates, and fringing effects begin to dominate, the field begins to spread out and weaken. Ultimately, at very large distances, at which the plates themselves can be considered points, the voltage is essentially constant, and the field strength directly between them becomes proportional to the inverse of the distance.

Answer 2— There is an attractive force between the plates of a capacitor created by the electric field. Physically moving the plates apart requires doing work against this force, and this work becomes the additional potential energy that is stored in the capacitor.

Answer 3— Dielectric materials are made of atoms, and the atoms contain both positive and negative charges. Although

TEST YOUR EQ

Contributed by David Tweed

neither the positive nor the negative charges are free to move about in the material (which is what makes it an insulator), they can be shifted to varying degrees with respect to each other. An electric field causes this shift, and the shift in turn creates an opposing field that partially cancels the original field. Part of the field's energy is absorbed by the dielectric.

In a capacitor, the energy absorbed by the dielectric reduces the field between the plates, and therefore reduces the voltage that is created by a given amount of charge. Since capacitance is defined to be the charge divided by the voltage, this means that the capacitance is higher with the dielectric than without it.

Answer 4— With certain dielectrics, most notably quartz and certain ceramics, the displacement of charge also causes a significant mechanical strain (physical movement) of the crystal lattice. This effect works two ways—a physical strain also causes a shift in electric charges, creating an electric field. This effect can be exploited in a number of ways, including transducers for vibration and sound (microphones and speakers), as well as devices that have a strong mechanical resonance (such as crystals) that can be used to create oscillators and filters.

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

CC Vault

Unlock the power of
embedded design.

A vault of need-to-know information in the fields of embedded hardware, embedded software, and computer applications

Order yours today! cc-webshop.com



This pocket-sized vault comes fully loaded with every issue of Circuit Cellar magazine and serves as an unparalleled resource for embedded hardware and software design tips, schematics, and source code.

*CC Vault is a 16-GB USB drive.



The Future of Artificial Intelligence

Intelligent Edge: Is AI Ready to Play a Role?

Three of the most common, buzzy phrases in the 2019 technology forecasts are edge compute, machine learning (ML) and artificial intelligence (AI). There's good reason for that. There is no question that all three are real trends that will have dramatic impacts on how the Internet of Things (IoT) is used to change our world. But a critical reader will be uncomfortable that these terms are often used interchangeably.

I know I am.

Edge compute, ML and AI are three very distinct technologies and they have very different roles for those designing products and systems that are or will become part of the IoT stack. If you are an embedded systems engineer, my guess is that you make one of two faces to someone combining AI and IoT Edge in the same sentence: rolling your eyes or twisting in horror. Let's break down the buzzwords and see if we can better prepare for what we are told is inevitable.



By
Scott Nelson
Chief Product Officer,
Digi International

EDGE COMPUTE

Edge computing is a distributed computing paradigm in which computation is largely or completely performed on distributed device nodes known as smart devices or edge devices as opposed to primarily taking place in a centralized cloud environment. – Wikipedia

The IoT stack has three levels of compute: Cloud, Aggregation Edge and the Physical Edge. **Figure 1** shows where each is deployed and offers a rough scale. The importance of understanding edge compute is that it is the resource used for any edge-application deployment and it is the platform upon which the tools of machine learning and AI may be deployed. As such, edge designers must think about both the compute capabilities they have in the functional blocks of an edge device as well as the compute capability that they should design in for future proofing.

For example, a cellular modem today has both RF and MCU processors. Forward-thinking designers have implemented code emulators—for example Python and MicroPython—within those processors and partitioned the memory to protect the functional firmware of the modem. The result is a functional computing block with a programmable engine. Applications at this scale include protocol translators, business logic and auto-configurators just to name a few. This physical edge compute capability enables interoperability, security and edge analytics.

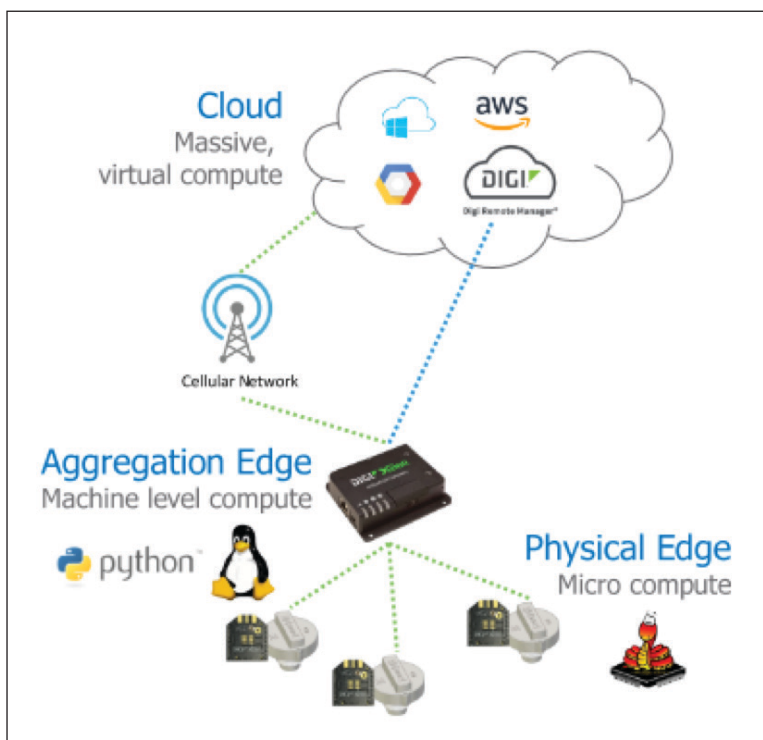


FIGURE 1

The IoT stack has three layers of compute: cloud, aggregation edge (gateways) and the physical edge (connected sensors and devices).

At the next level up, the scale increases to the point where full Linux containers can deploy PC-level applications. Virtualization becomes operative at this level and architects can look at moving cloud compute applications down to this level to take advantage of reduced data latency as well as reduced data transport costs. Systems designers leveraging Aggregation Edge compute are looking a full, closed loop control over the local wireless networks.

MACHINE LEARNING

Machine learning (ML) is the study of algorithms and statistical models that computer systems use to progressively improve their performance on a specific task. Machine learning algorithms build a mathematical model of sample data, known as “training data”, in order to make predictions or decisions without being explicitly programmed to perform the task. – Wikipedia

Training vs. learning—that is the question when we consider Machine Learning vs. Artificial Intelligence. In my book, Machine learning is not true learning. Machine learning is training. The algorithms upon which it is based make it deterministic within the constraints of the digital twins used to train them. Machine learning is very powerful for optimization and adaptation to known key parameters precisely because of its functional ability to train.

Training drives performance and optimization, whether training people or machines. As such, machine learning and digital twins are already proliferating. Digital twins can be as simple as a linear correlation algorithm and as complex as a fully parameterized emulation of a machine, including end of life. Machine learning and machine training will allow operations to “see around the corner” of time, but it still doesn’t reach the threshold of true learning.

ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and other animals. In computer science AI research is defined as the study of “intelligent agents”: any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term “artificial intelligence” is applied when a machine mimics “cognitive” functions that humans associate with other human minds, such as “learning” and “problem solving”- Wikipedia

The key difference between Machine Learning and Artificial Intelligence is that training vs. learning definition.

AI-based systems will learn outside of the statistical and physical model constraints of machine learning digital twins. AI-based systems have the cognitive capacity to question a model and experiment with it. AI-based systems can learn from mistakes.


That is the rub. Most controls applications cannot afford mistakes. AI can work in consumer preference and recommendation domain because the consequences of the mistakes are effectively virtual. I watched a show I didn’t like on Netflix the other night. I lost an hour of my time but I mitigated those consequences by working during the show. AI in consumer Internet applications may have financial consequences, but they will not crash trains, planes or automobiles.

AI, therefore, has the opportunity to get traction in areas where the learning environment can be made safe or constrained to performance levels that do not cause harm. Comfort and productivity applications are excellent candidates here. Another valuable IoT application is device behavior in the security context. AI algorithms can learn about and predict behavior which, if applied to edge devices, can help identify both physical performance issues—calibration for example—as well as security breaches.

THE FUTURE

Moore’s Law has made edge compute ubiquitous. There are two edges when we talk about edge compute—the physical edge and aggregation edge. Virtualization will happen at both and across the two, and this virtualization will drive the value of IoT deployments. The better we understand the physical world, machines and systems, the better we can “teach” machines to learn and perform better, and that is what machine learning is all about: training and optimization, not the unknown.

Artificial intelligence is the most powerful of these three trends but also the most uncertain and least understood. Designers and control engineers can feel safe with machine learning via the determinism and constraints of their models (digital twins) and algorithms. AI, on the other hand, will remain in the cloud for a while because experimenting in the virtual world of data is safer than the physical world of the IoT.

There is a reason this discussion is still somewhat philosophical. In the same way quantum mechanics was a philosophical discussion between Bohr and Einstein in the early 1900s. We cannot yet see how AI systems will transform our world and that is a bit scary—at least in the context of 2019. 

Scott Nelson is Chief Product Officer at Digi International. For more than 25 years he has led product development and entrepreneurial business growth as both a technology and business leader. Scott is formerly CEO/CTO of Reuleaux Technology, where he helped companies in both Silicon Valley and the MSP area with strategy and new business development in the Internet of Things (IoT). After beginning his career at Honeywell in the Corporate R&D center, he spent the next 15 years at Logic PD as CTO and EVP.

As technology evangelist, Scott is connected to Silicon Valley and at present is a member of multiple tech start-up advisory boards and a leading start-up accelerator the Alchemist Accelerator. He holds a Ph.D. in applied and engineering physics from Cornell University, a doctoral minor in business administration from the Samuel Johnson School of Management at Cornell University, and a B.A. degree in physics and mathematics from St. Olaf College.

RESOURCES

Digi International | www.digi.com

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.**

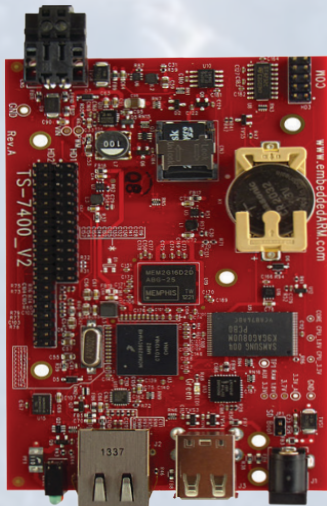
You no longer have to worry about long shelf life
or tie your capital in bare board inventory.

Accutrace[®] inc.

www.PCB4u.com sales@PCB4u.com

SAM & ITAR Registered UL E333047 ISO 9001 - 2008

INTERESTING USE #901: AUTONOMOUS POLAR ICE BUOY



TS-7400-V2

Starting at
\$84
Qty 100

When Ice911 needed to design an autonomous ice buoy to deploy to the Arctic Circle, they came to Technologic Systems. They chose the TS-7400-V2 for its low price, fanless operation, power efficiency, and -40 to 85°C industrial temperature range. The ARM9-based i.MX286 SoC can be configured down to 454MHz to save on consumption, vital when you're running on batteries.

For more information on this project go to:
www.embeddedARM.com/buoy.

We build our single board computers to thrive in any environment from the Saharan deserts to Arctic Circle.

Where does your project want to go?

