AUTOMOTIVE ELECTRONICS
AUGUST 2019
ISSUE 349

circuitcellar.com

circuit cellar

Inspiring the Evolution of Embedded Design

AUTOMOTIVE ELECTRONICS GEAR UP FOR NEXT GEN CARS

Product Focus: Tiny Embedded Boards | Displays for Embedded Systems |
MCU-Based Portable Music Synthesizer | Building a Twitter Emote Robot |
ToastBot Uses PIC32 | Build an RGB LED Controller | Inference Engines in AI
Energy Monitoring (Part 3) | The Fundamentals of Fuseology |
Bluetooth Mesh (Part 4) | Watt's Up with LEDs? | The Future of IoT Standards

$9.00US $10.00CAN
08
0 74470 75349 0

You can take it almost anywhere.

Where will it take you?

CC VAULT

Now you can have the complete Circuit Cellar issue archive and article code stored on a stylish, durable and portable USB flash drive. You can easily keep your CC Vault archive up to date by purchasing subsequent issues from our webshop or by downloading issues with a Circuit Cellar Digital Subscription. Issues appear in searchable PDF format.

Complete archive includes PDFs of all issues in print through date of purchase.

SIGNAL PROCESSING
OCTOBER 2017
ISSUE 327
circuitcellar.com
circuit cellar
Inspiring the Evolution of Embedded Design
DRONE DESIGNS REACH
NEW PROCESSING HEIGHTS
Q&A: Design for Manufacturing
Solutions for the Signal Chain
Building a Robot Hand | Emulating Legacy Bus Interfaces
Logger Device Tracks Amp Hours in Home
Positive Feedback Loops
Gas Monitoring and Sensing | Measuring Antenna Performance
Fixed-Price Contracts
The Future of PCB Design
Retro TV Remote Design
circuitcellar.com

Visit cc-webshop.com to purchase

SEGGER — The Embedded Experts

Debug Probes | Software Tools | RTOS | File System | Compression | Connectivity | Security & IoT | User Interface | In-Field-Upgrades | Production

emCompress-ToGo

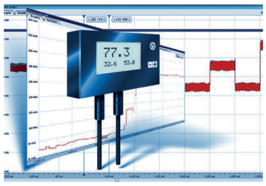
Compress Data in Real-time on any Embedded System!

More Storage

More Bandwidth

Faster Updates

One Professional Compression Solution for All Applications



Data Loggers

Internet of Things

Space / Avionics

Networking

Medical Devices

Consumer Electronics

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

Worldwide: sales@segger.com
☎ +49 2173 99312 0

U.S. East Coast: us-east@segger.com
☎ +1 978 874 0299

U.S. West Coast: us-west@segger.com
☎ +1 408 767 4068

segger.com

OUR NETWORK

circuit cellar

audioxpress

VOICE COIL

LinuxGizmos

LIS LOUDSPEAKER INDUSTRY SOURCEBOOK

SUPPORTING COMPANIES

Accutrace, Inc.	C3
All Electronics Corp.	77
CCS, Inc.	77
congatec, Inc.	19
EzPCB	25
Hackerboxes	71
Measurement Computing Corp.	61
SEGGER Microcontroller Systems	1
Siborg Systems, Inc.	63
SlingShot Assembly	37
SoC Conference 2019	57
Technologic Systems, Inc.	C4, 77
VersaLogic Corp.	15

THE TEAM

PRESIDENT
KC Prescott

CONTROLLER
Chuck Fellows

FOUNDER
Steve Ciarcia

EDITOR-IN-CHIEF
Jeff Child

SENIOR ASSOCIATE EDITOR
Shannon Becker

TECHNICAL COPY 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 349 August 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

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.

INPUT Voltage

Sensors and Embedded at Home Together

In many ways my trip to California in late June was like coming home again. I've traveled to that area of the country more than any other over years. There's really no other place where you can zip around in a rental car and visit so many technology companies—all a short drive from one another.

This recent trip was to attend the Sensors Expo show in San Jose. A new twist on the show was the launch of a new Embedded Technologies Expo and Conference co-located and sharing an exhibition hall with Sensors Expo. The Embedded Expo's 53 exhibitors added to Sensors' over 350 made for a significant event. There's a natural synergy between sensor technology and general embedded technologies, so it was certainly a smart move to co-locate.

I was particularly pleased to see that PICMG was exhibiting (or the PCI Industrial Computer Manufacturers Group if the acronym doesn't ring a bell). PICMG is a not-for-profit consortium of companies and organizations that collaboratively develop open specifications. As the group's name implies, since its birth those specifications have revolved around PCI, with the PCI Express-based COM Express, for example, among its many specs. In recent years, PICMG has ramped up its efforts on broader systems-oriented specs in keeping with today's market demands—the centerpiece being the organizations' work on Industrial IoT (IIoT) specifications.

Sensors play a big part of those efforts, and PICMG took great advantage of their presence at Sensors Expo by performing live a demonstration at their booth to showcase its efforts to create plug-and-play sensor interoperability for IIoT. The demo was comprised of three main components: (1) a control and visualization station, (2) IP-enabled smart sensors and binary-enabled smart sensors, and (3) a multilingual smart sensor gateway. Tied with the demo was a Smart Sensors Challenge whereby PICMG invited anyone with a 4-20 mA sensor to come to their booth to take the challenge.

The visualization station was responsible for monitoring and control of the factory installation. It was implemented with standard hardware and used IT technologies to gather data and take actions. PICMG is collaborating with the DMTF to extend their Redfish management and control specification for use in Industrial IT.

On a factory floor, hundreds (or thousands) of sensors and actuators gather information and alter operation under the control of the control/visualization station. In order for these sensors to participate, they must be able to communicate over the installation's IP network. Some autonomous sensors (or sensors with higher levels of functionality) may do this directly though their own dedicated ethernet port. These IP-Enabled smart sensors plug directly into the installation's network and present themselves as micro-servers of the PICMG/DMTF Redfish sensor data model.

PICMG's CTO Doug Sandy at the show remarked that other, generally lower-cost, sensors may present sensor data in binary fashion over a serial data link. "This greatly reduces the cost of the associated controller yet requires an extra gateway in the system to translate from the binary coding to the PICMG/DMTF Redfish format on the IP network. Current PICMG initiatives include binary coding of the data model and the operation of the multi-lingual gateway, and a small form-factor 'Postage-Stamp' module for creating sensor endpoints," said Sandy.

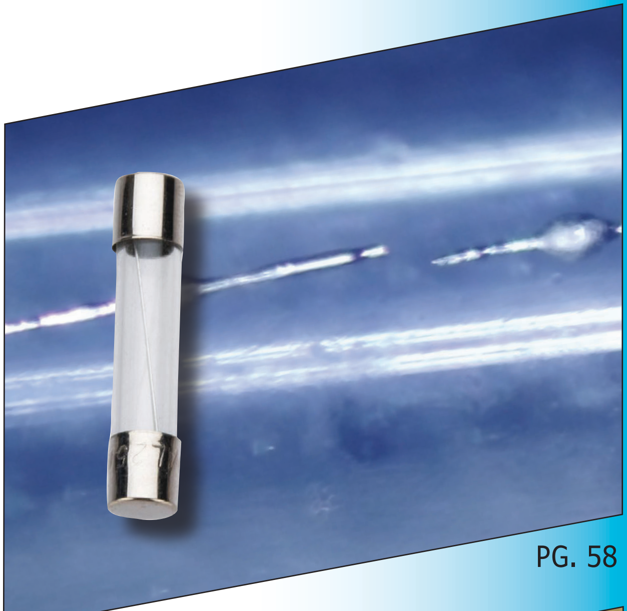
As sensors were added to PICMG's booth demonstration system, the sensors automatically reported their capabilities to the system without need for reprogramming. Enabling plug-and-play capability over the IP network is a central feature of PICMG's IIoT scheme. The PICMG IIoT network architecture model, currently beginning development, will not dictate specific hardware to be used. That said, according to PICMG, COM Express and CompactPCI Serial may both be good choices for many installations due to their rugged designs and ability to withstand harsh environments.

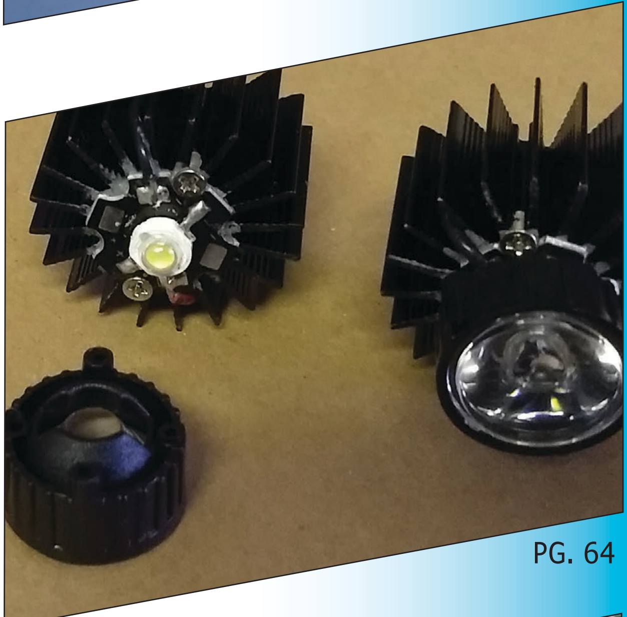
PICMG's Sensors Challenge was just one of many examples at the co-located event of the synergy between sensor technology and embedded computing technologies in today's connected IoT age. So, while the trip may have felt like coming home for me, it was crystal clear that the two technologies and two shows felt right at home together.

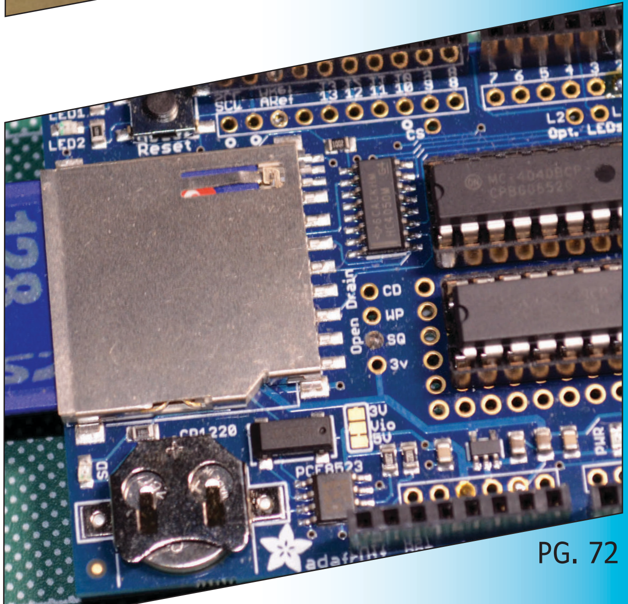
Jeff Child

COLUMNS

@editor_cc
@circuitcellar
circuitcellar

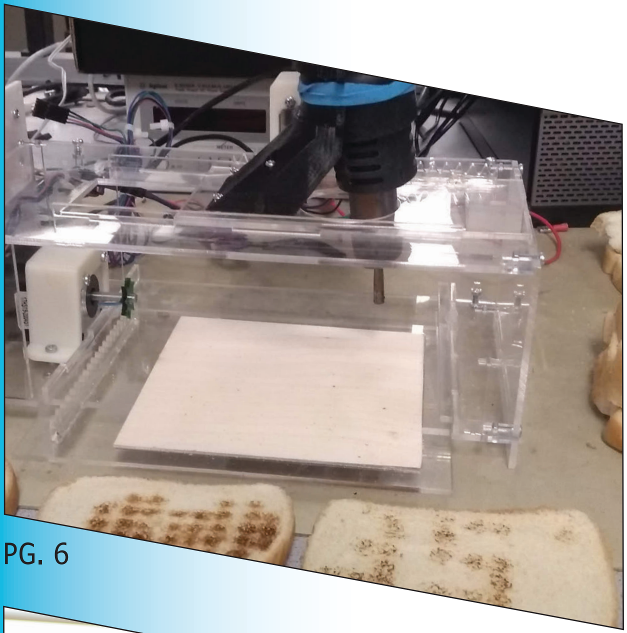

PG. 58


PG. 64



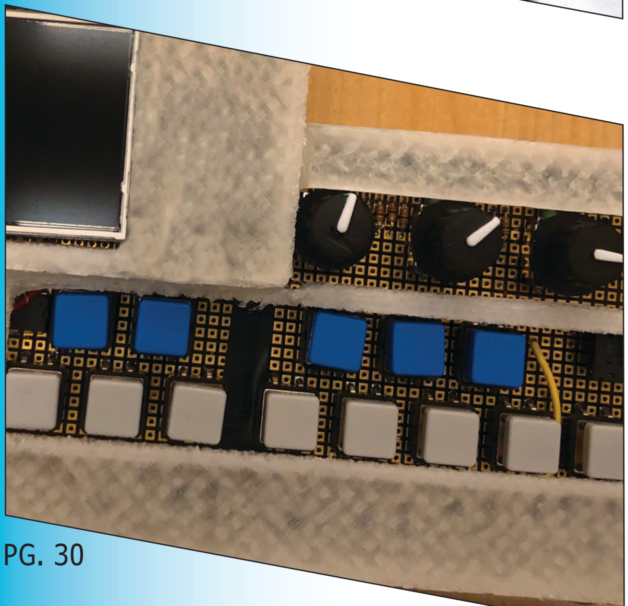
PG. 72

FEATURES



PG. 6

PG. 20



PG. 30

Automating the Art of Toast

With a Side of Raspberry Pi

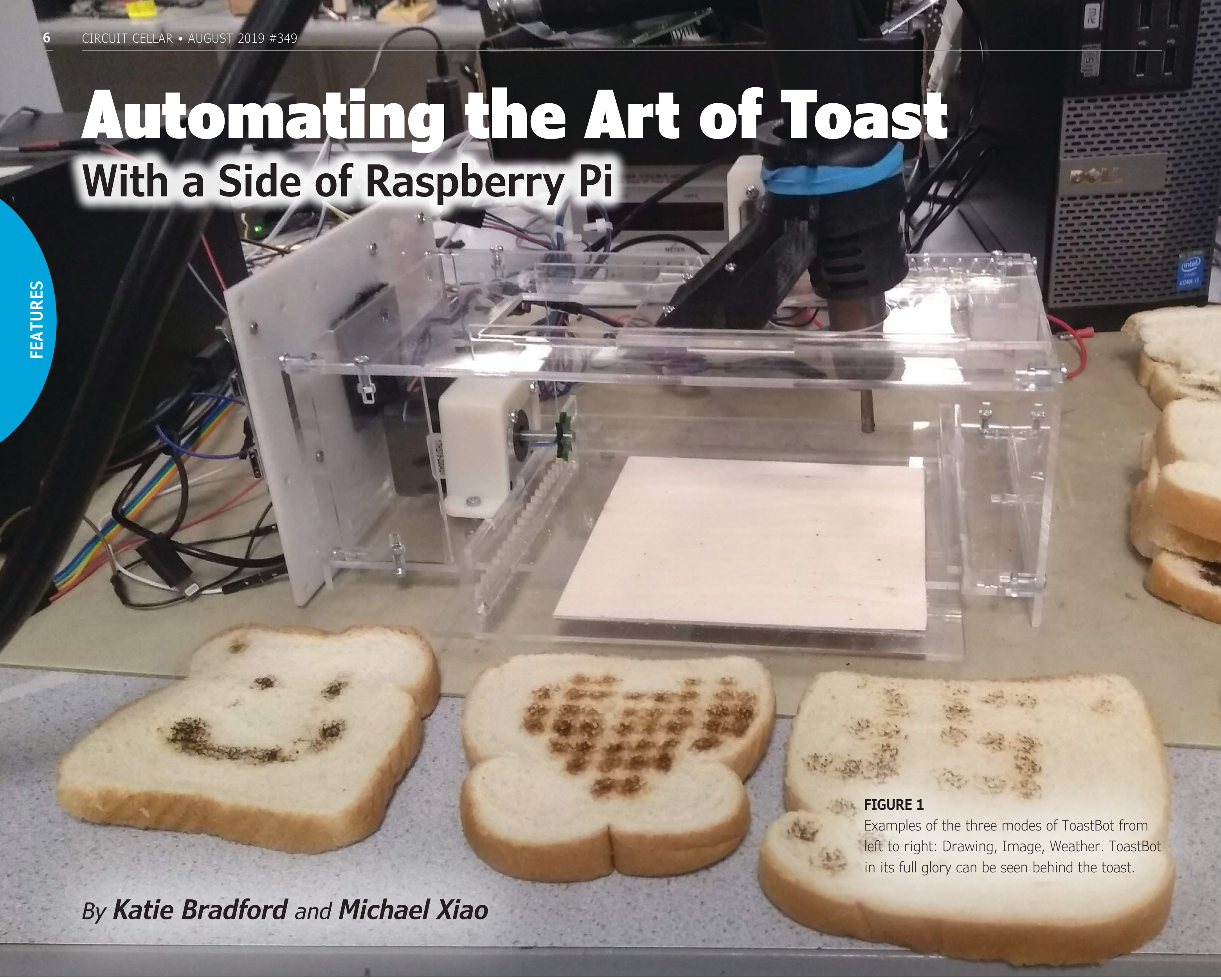


FIGURE 1
Examples of the three modes of ToastBot from left to right: Drawing, Image, Weather. ToastBot in its full glory can be seen behind the toast.

By *Katie Bradford* and *Michael Xiao*

The emergence of culinary robotics and automation has already begun to revolutionize the way we prepare our meals. In this article, learn how these two Cornell undergraduates designed an advanced toaster that's able to toast any pattern—image, text or even today's weather—onto a piece of bread. The project makes use of Microchip's PIC32 MCU and a Raspberry Pi Zero W board.

ToastBot is an automated toaster equipped with a heat gun and a PIC32 microcontroller (MCU). It is able to toast any image, text or even the weather onto a slice of bread (**Figure 1**). The device uses two servos to control an acrylic, two-axis gantry with a 3D-printed heat gun mount. A Raspberry Pi board is used to gather image or weather data from the Internet to be sent over to the PIC32 MCU to control the motors in real time. ToastBot has three modes of operation: Image, Sketch and Weather. In Image mode, a user can upload an image from the Internet through SSH into the Pi with a phone or computer, which is then processed into a print sequence. In Sketch mode, a four-button controller is used to manually control ToastBot to create a work of edible art. In Weather mode, the device automatically collects the temperature and conditions via Yahoo!'s weather API and prints the weather onto the toast.

Setting the mode and processing an image or displaying weather conditions is done via a Python script hosted on a Raspberry Pi Zero W board [1], which then passes the information to print onto the PIC32 through a makeshift serial bit-banging process. In the two automatic modes (Weather and Image), the two stepper motors move the heat gun in a lawnmower pattern through a 10×10 array, either choosing to toast a pixel or not, based on the information sent from the Pi. In Sketch mode, a four-button switch controls the motors, either moving the toasting point up, down, left or right. To darken a pixel, the user can let the heat gun remain still for a few seconds. The high-level design is shown in **Figure 2**.

Our idea for ToastBot was conceived through several layers of brainstorming. The project was created for Cornell's ECE 4760 Microcontrollers class, focused around using a PIC32 to implement real-time control of various systems.

Two themes that we were enthusiastic about implementing were creating a device with a significant mechanical component and that incorporated food. ToastBot fit these criteria, and was both a challenging endeavor and a fun project. The device is also pragmatic, in that it can combine some of our morning routines autonomously. If a user runs ToastBot upon waking up, it can provide a warm breakfast and an overview for how the user should dress or prepare for the day ahead. Additionally, there is a lot of potential for further fun and creativity within the device itself, including sending texts via toast, sending a loved one a picture, or another novel avenue for drawing and sketching.

MECHANICAL DESIGN

As shown in **Figure 3**, before we began laser cutting or 3D printing, we used Autodesk Fusion 360 to make a detailed 3D model of our design. As an overview, the chassis has two layers. The bottom layer houses one stepper motor that moves a plate holding the bread along the Y-axis. The top layer houses another stepper motor that moves a plate along the X-axis with a hole in it, such that the heat gun can be mounted through the plate to access the bottom layer. To translate the stepper motor's rotational motion into linear motion, we designed our own laser-cut rack-and-pinion tracks, shown in Figure 3.

The majority of our pieces were designed to be manufactured using a laser cutter due to its faster rate of prototyping compared to production using 3D printing. As shown in Figure 3, the two side pieces support the two layers with built-in slots. Each layer has a sliding piece—one holding toast and the other holding a heat gun—along with some railings to restrict the movement to just one axis each. These layers must be strong enough to support the weight of the stepper motors and endure the minor vibrations from the motor's

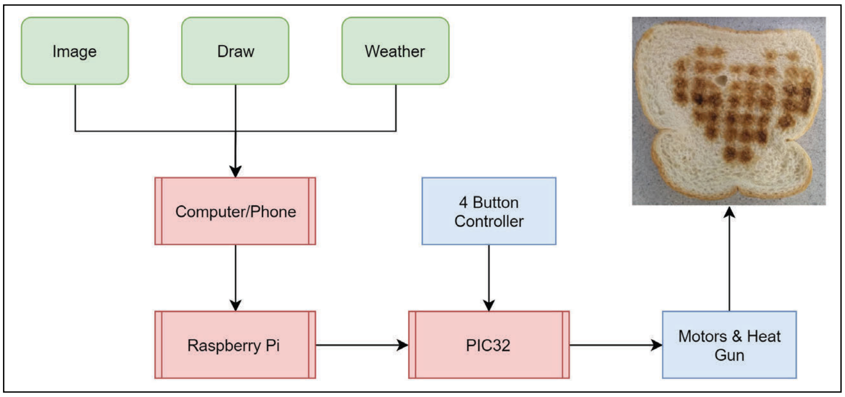


FIGURE 2
The block diagram for ToastBot's functionality. Input, shown in green, comes from the user, through the Raspberry Pi, and is communicated to the PIC32. The PIC32 then operates the heat gun accordingly to make the toast, taking additional input from the controller in Draw mode.

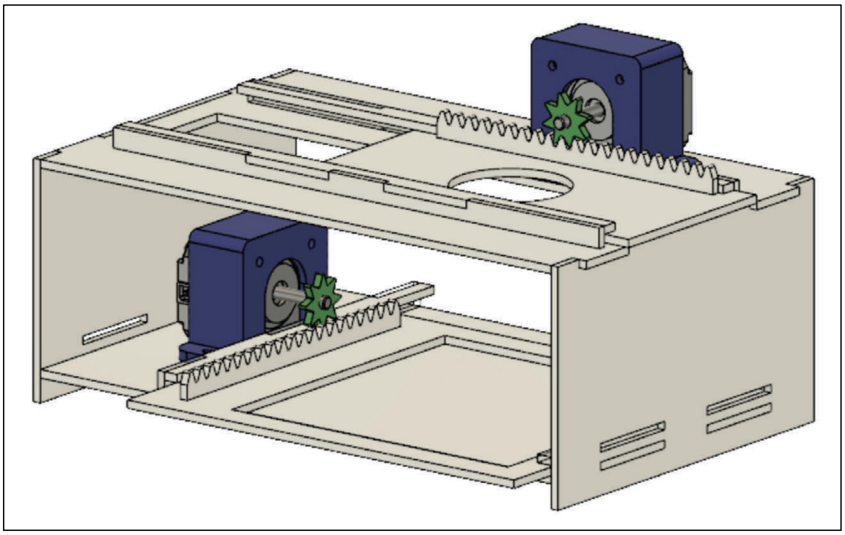


FIGURE 3
A prototype CAD model of ToastBot. The two motors with motor holders controlling the rack-and-pinion systems can be seen on the two axes.

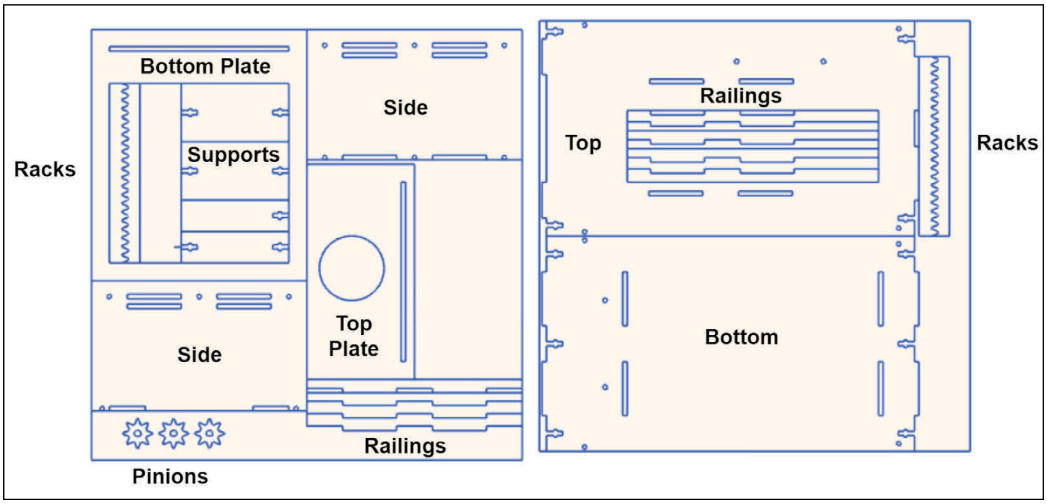


FIGURE 4
Labeled DXF design for ToastBot chassis. All parts can be produced with two 12" × 12" sheets using a laser cutter. These files are available on the Circuit Cellar article materials webpage.

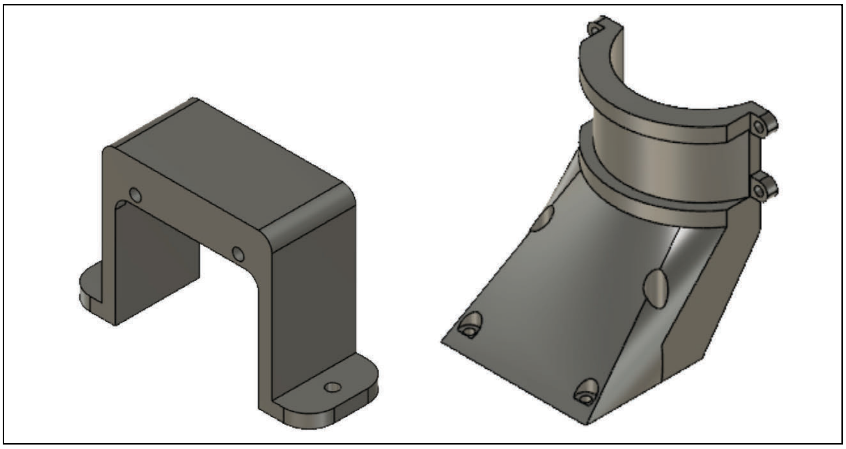


FIGURE 5
CAD models of 3D-printed pieces. On the left is the stepper motor mount, and on the right is the mount for holding the heat gun perpendicular to the toast.

operation. Our final design measured about 10" × 6" × 4".

One of the first challenges we had to overcome was finding a method to join 2D acrylic pieces together into a 3D assembly. Our first version was made with slots and secured with glue. However, we found that, although this held well initially, it was not nearly as robust as we wanted.

Upon further experimentation, we designed our own screw joint, which built off a slot-and-tab design by adding an additional hole on the slot side and a nut holder on the tab side. This essentially built in threads onto our 2.5 D model, offering several advantages. Among these are ease of assembly, no cure times, very strong joints and modularity of individual components for redesigns or replacements. Our final laser-cut design can be fitted onto two 12" × 12" acrylic sheets as shown in **Figure 4**.

Despite having most of the physical design laser cut, there were a few more complex 3D pieces that we chose to 3D print. The first was a simple motor holder, shown to the left in **Figure 5**, which secures the motor to the frame with M3-size screws. The second was the heat gun mount, which is screwed into the sliding top plate, as shown on the right of Figure 5. The heat gun fits into the cylindrical cutout and is secured with some strong elastic bands. This mount keeps the heat gun securely fastened at a proper angle and distance from the toast as the device moves it.

ELECTRICAL DESIGN

Our electrical hardware was centered around the PIC32 MCU. Eight of the MCU's GPIO pins were dedicated to operating the two motor drivers, four more were used for the button controller and two more were used for communication with the Raspberry Pi. The detailed schematic is shown in **Figure 6** [2].

A large part of our preparation for the project consisted of learning how both our

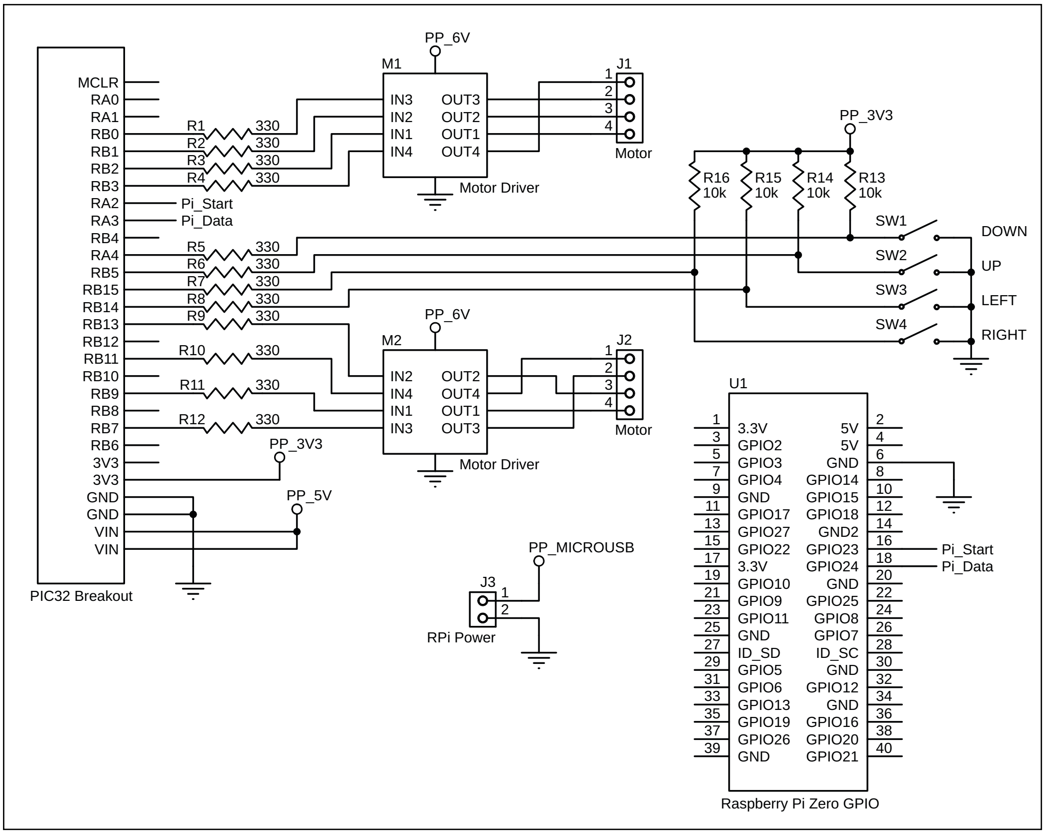


FIGURE 6
Full schematic of ToastBot. Four buttons from Draw-mode controller, Raspberry Pi Zero W breakout board, PIC32 small breakout board [2] and motor drivers/connectors are all shown.

stepper motors and motor drivers worked. Stepper motors have a magnetic core with "fingers" sticking out with alternating polarity. Each of the electromagnets surrounding the core are toggled on and off, drawing the finger to the next magnet, "stepping" it forward or backward, depending on the signal. Stepper motors are precise compared to other motors, but require quite a bit of current to operate well.

We used two STMicroelectronics (ST) L298N motor drivers [3], which have four data inputs, three power pins and four data outputs. We powered the drivers off a 6 V line. There is a built-in voltage regulator on the motor driver board for up to 12 V input. The drivers were common-grounded with the PIC32 and the Pi. After carefully examining the Arduino `stepper.h` library, we found the sequence for the four inputs to move the motor forward, and mimicked the behavior on the PIC32.

Running this sequence with a short delay of 5 ms turns the stepper motor forward a step. To run in reverse, we could simply run this in reverse starting with step 4 .. 3 .. 2 .. 1 in the code. Because the load was not heavy, we were able to run the motors with no built-in acceleration time. Testing this and understanding how the steppers worked helped us throughout the project with troubleshooting and debugging, since we knew what to expect as outputs on our motor drivers. We found that on operation our stepper motors drew more than 1 A while running at 6 V. Luckily, we were able to find a surplus power supply that was rated for 2.5 A at 6 V. To make our power system even more robust, we chose in our movement algorithm to have only one motor move at a time. This helped to minimize any issues with power consumption.

One last addition we made to our design was creating the hardware bed shown in **Figure 7**. This bed combines our various electrical components and neatly secures them to a frame, so that they are not loosened or damaged with movement. In the design, there are holes for mounting the two motor drivers, the PIC32 and its breakout board and the Raspberry Pi Zero W. The bed was attached with Velcro onto one of the sides of the ToastBot, for easy removal for debugging and for adding features in the future.

PIC32 SOFTWARE DESIGN

ToastBot's software runs on both a PIC32 MCU and a Raspberry Pi Zero W, communicating through a custom bit-banging protocol. The Raspberry Pi is used to retrieve data wirelessly on the user's wireless commands and the current weather conditions, to process images, and then to relay these data to the PIC32. The PIC32 is then responsible for appropriately processing the data and moving the heat gun accordingly.

Our software for the PIC was divided into three parts: waiting for a command, manual mode and automatic mode. Before entering any of the motor-controlling code, our main function was run and the GPIO pins were initialized for the motor drivers and the Raspberry Pi data ports.

Before the motors are allowed to move, a start signal from the Pi must be received by the PIC32. To get the Pi and PIC32 communicating, we decided to implement our own version of serial communication by bit-banging the data through the GPIO pins. This was a feasible option for us because our data was only 100 bits, and the clocks on both devices were accurate enough to stay synchronized for such a short transmission.

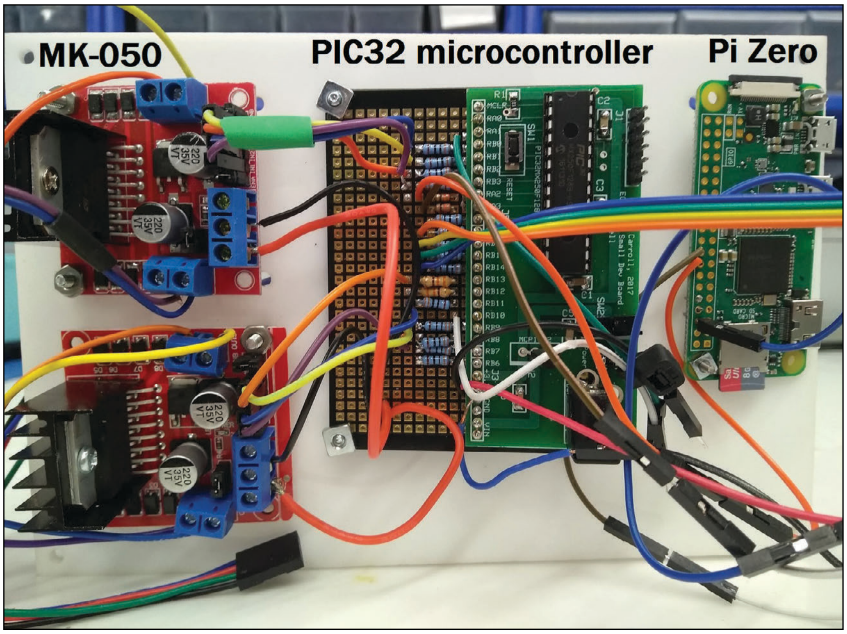


FIGURE 7
Completed hardware bed with components mounted. From left to right; MK-050 motor drivers, PIC32 small board breakout and Raspberry Pi Zero.

ABOUT THE AUTHORS

Katie Bradford (keb278@cornell.edu) is a senior in electrical and computer engineering at Cornell University. She enjoys robotics, analog electronics and rapid prototyping. Outside of the classroom, she enjoys attending concerts and exploring.

Michael Xiao (mfx2@cornell.edu) is a senior studying electrical and computer engineering at Cornell University. His interests include robotics, design and ergonomics and embedded systems. Outside of the classroom, Michael can be found rock climbing or perusing local thrift stores.

We used one output from the Pi to signal the start of transmission and another output from the Pi to indicate "0"s or "1"s in the data array, separated by a 5 ms delay. On the PIC32 side, these data were read every 5 ms and stored in the PIC32's memory. The biggest issue we had with this was calibrating the timing. In our first run, several pixels looked shifted one or two rows over. After measuring the signal from the Pi with an oscilloscope, we found that the delay from the Pi was actually slightly more than 5 ms. To minimize this discrepancy, we extended this delay to 20 ms and added an additional 15 ms every 20 reads, to keep both sides of the transmission calibrated. After further testing, we found that this effectively solved our calibration issues, despite being a relatively messy solution. With this in mind, the full data transmission takes approximately 2 seconds to occur (100 bits × 20 ms), which is a reasonable amount of time for operation.

After receiving the start signal and data array from the Pi, the PIC32 decides to operate either in manual mode or automatic mode. It does this by counting the number of zeros sent in the array. If—and only if—the entire 100 bits were 0, the PIC32 enters manual Draw mode. In this mode, the motors are controlled very simply by the four buttons on the wired controller. These buttons are tied low to ground over a 10 kΩ resistor. The device can move in only one direction at a time, X or Y, due to our motor power constraints. To exit Draw mode, the device can be reset to wait for a new transmission across the Pi communication lines.

If the data array received from the Pi is not all zeros, then the device knows that either an image is being printed or that the weather is being printed, and therefore, operates in automatic mode. In this mode, the PIC32 moves the heat gun in a lawnmower pattern—shifting 10 steps to the left for each of the 10 pixels, returning 100 steps to the right (giving full range of motion along the x-axis) and shifting down 10 steps. This cycle repeats 10 times to give the full range on the Y-axis as well. After completing a thorough run-through of the entire drawing surface, the motors return the device to its original zero position. While it is performing this lawnmower pattern, the PIC32 also reads the data array, checking the index of the array to see if there is a 1 or a 0. If there is a 1, the PIC32 delays for 2 seconds on its current position, which is long enough for the heat gun to toast that particular spot. If there is a 0 in the index of the data array, the PIC32 does not wait, not giving the necessary time for a darker toast. This pattern toasts the entire surface slightly, and the desired pattern is toasted more intensely. After finishing this cycle, the device is ready for its next command.

RASPBERRY Pi SOFTWARE

The Raspberry Pi is used for Wi-Fi connectivity and image processing, enabling ToastBot to be controlled and operated remotely. First, the user accesses the Pi via SSH from a host device, such as a phone or computer. Next, if the user wishes to upload an image for toasting, the image can be saved, and PSCP or PuTTY Secure Copy client can be used to send the image over to the Pi. If intending to use the device in Weather or Sketch mode, this step can be skipped.

Next, the user simply has to run the Python script over SSH, along with a keyword indicating the mode. There are three keywords associated with the modes: weather, image and sketch.

The command would look something like this:

```
python output.py weather
```

After this command has been sent, the Pi runs a processing script according to the parameter sent, updating a 10×10 array of 1s and 0s to be sent to the PIC32. If Image mode is selected, the Pi will downsample the image to a 10×10 image. Taking this down-sampled image, the Pi compares each pixel to a threshold value to decide if it is dark or light, appending a 1 or a 0 to the data array accordingly. Some examples of down-sampled graphics are shown in **Figure 8**.

If weather is sent as the parameter, the Pi uses a Python weather API to collect data from Yahoo! Weather, particularly noting the temperature and condition. We wrote custom bitmaps for the values of the temperature to be written on the top half of the toast, and predefined graphics to represent the conditions windy, raining, snowing and sunny. Unfortunately, the sunny condition was never

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

Adafruit | www.adafruit.com

Microchip Technology | www.microchip.com

Raspberry Pi | www.raspberrypi.org

Sparkfun | www.sparkfun.com

STMicroelectronics | www.st.com

used, because testing occurred during an upstate New York winter. The layout and icons for conditions are shown in Figure 8.

Finally, if Sketch mode is selected, the output data array is set to all zeroes. When this is sent to the PIC32, it is interpreted as Sketch mode, letting the user move the heat gun freely using the controller.

RESULTS

We were pleased with the results of our project. ToastBot was successfully operated remotely using Wi-Fi, was equipped with several modes of operation and user interaction, and produced novel and edible pieces of toast. **Figure 9** shows the fully assembled ToastBot, ready to decorate some bread. You can see in **Figure 10** the end products from using ToastBot operated successfully in all three modes, and the resulting images were recognizable and easy to read. We were quite proud of our mechanical design—the device looked great and felt robust.

Despite these successes, there were a few issues we would like to address in a future iteration. The main problem in the user experience was how long the device took to operate. The full data transmission stream from a host device to the Pi to the PIC32 takes only a few seconds, which was negligible for the user experience as far as we could tell. However, the time it takes ToastBot to finish a piece varies with the number of pixels that need to be toasted. For a less complex design, ToastBot could finish in under a minute, but for a denser or darker design, particularly common in Image mode, ToastBot could take up to 3 minutes to finish.

Additionally, we would like to make the user interface much more professional by making an app to connect to the Raspberry Pi, rather than using SSH. This would broaden our audience and make the device much easier and intuitive to use. Finally, we aim to broaden our data structure on an analog scale, to dictate the level of "toasted-ness" for each pixel. Rather than burning for 2 seconds over the dark pixels, the device could vary the time spent on each pixel between 0 and 3 seconds, to produce an effect of shading and gradient, thus improving the clarity and resolution of our results.

For code and details, go to the *Circuit Cellar* code download webpage and *Circuit Cellar* article materials webpage. There you can view our results and two videos where we demonstrate the use of ToastBot and show its results. The product was fun and easy to use. All you need is a Wi-Fi connected device and a dream, and you're ready to toast.

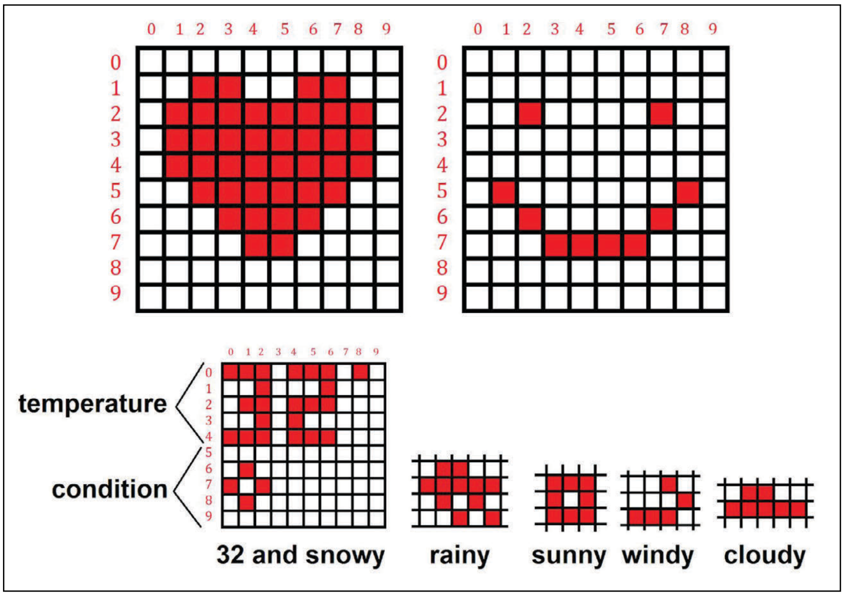


FIGURE 8
Top: Sample images showing the output for a heart and smiley face. Bottom: Positions of weather conditions in the 10×10 matrix. Temperature is printed on a 3×8 matrix in the top left corner of the pixel array, with conditions printed below.

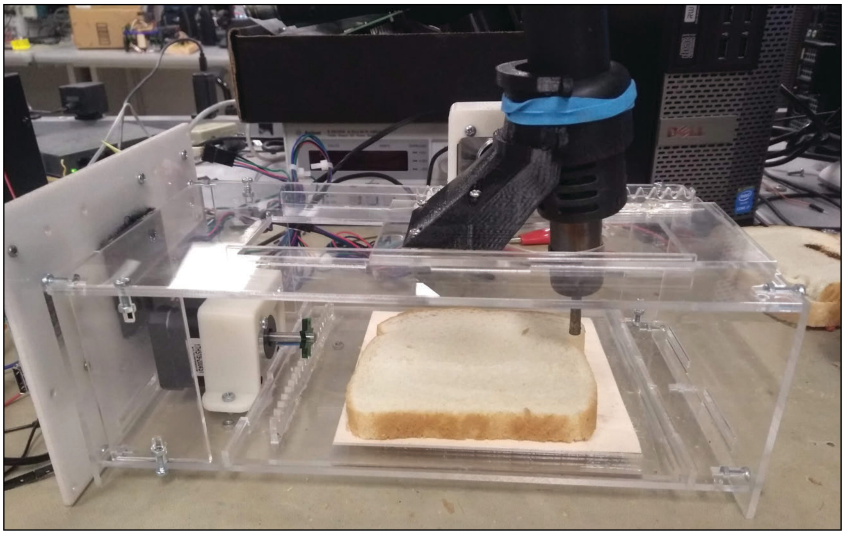


FIGURE 9
Fully assembled ToastBot, ready to decorate some bread.

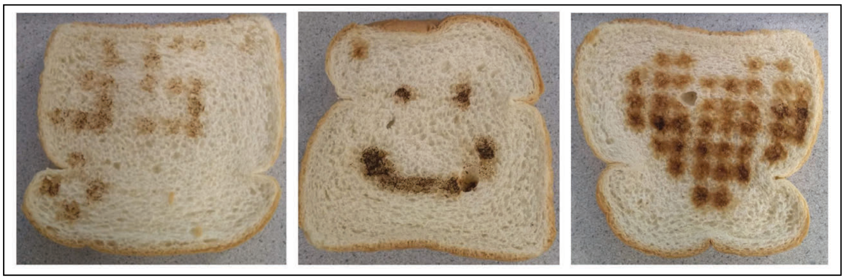
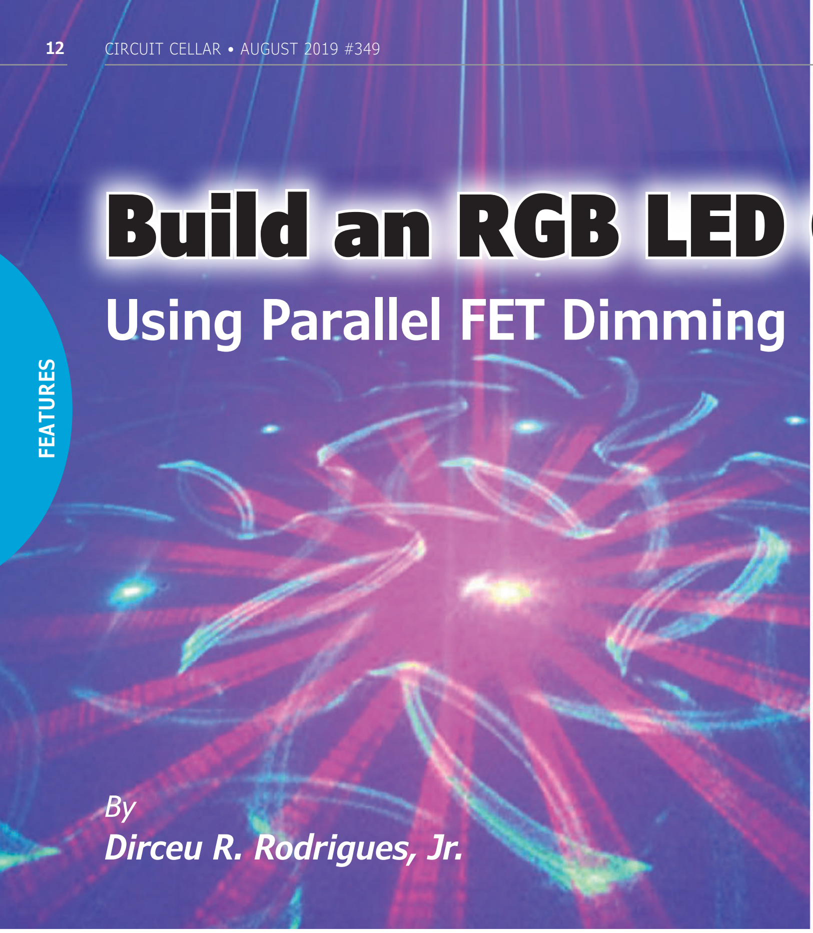


FIGURE 10
Three successfully completed pieces of toast. From left to right: Temperature mode, Draw mode, Image mode.



Build an RGB LED Controller

Using Parallel FET Dimming

By
Dirceu R. Rodrigues, Jr.

There are a lot of fun and interesting things you can do with LEDs and the different ways to control them. In this article, Dirceu describes an alternative approach to control RGB LEDs, using the parallel FET dimming technique. He steps through his efforts to design and build an alternative lighting system based on power RGB LEDs. To control them he goes very old school and uses an 8-bit MCU and the BASIC programming language.

Nowadays, applications involving advanced processors like Arm and Espressif ESP-32 are commonplace. But I thought it would be cool to test some fun lighting sequences that are controlled by an 8-bit microcontroller (MCU) programmed using an ancient language: BASIC. Although using pulse-width modulation (PWM) to dim LEDs with MCUs is a long-established idea and there's a plethora of such products on the market, my approach differs from others regarding the drive method used. The benefit will be a relatively shorter BOM, but is also of particular interest to embedded system designers involved with LEDs because it will be possible to experiment with alternative configurations for the control stage.

LEDs are inherently nonlinear devices. Their brightness depends primarily on the current flowing through them, even though the voltage on terminals don't vary that much. To achieve a constant LED current, there are two approaches: linear or switched current regulation. A linear regulator is preferred in situations where the noise due to commutation would be unacceptable—or for example, in high-precision measurement equipment. When efficiency is the main concern, a switched regulator or driver usually is chosen.

A commercially available driver usually operates above 1 MHz, providing hysteretic regulation for the LED current. To implement the required dimming, a common solution is to apply a PWM signal to an enable pin of the regulator. Because the entire component is switched continuously, the delay due to the soft start function must be taken into account. The disadvantage of this mode is, therefore, the limitation at low frequencies, usually 100 Hz. Other drivers, such as the ZXLD1350 from Diodes Inc. (used here), have a similar input named ADJ, capable of accepting a PWM signal up to 1 kHz.

As outlined in **Figure 1**, my application takes a different approach. Rather than applying PWM pulses to a dedicated regulator pin, these signals are used to "short-circuit" the LED. So, when a switch is closed, the corresponding LED is off. This technique—known as parallel FET dimming—does not pose a problem itself, since the driver is based on a current source. Regardless of the state of each LED, the same current always flows through the entire circuit. For an independent control of three RGB LEDs, traditionally three drivers are employed, each with its own inductor, Schottky diode and sensor resistor, as shown in **Figure 2a**.

My alternative configuration to reduce the number of components is to connect the three LEDs in series, each with its own switch driven by PWM (**Figure 2b**). Note that, in this case, the ADJ pin from the single ZXLD1350 stays floating, and the three PWM signals are moved to the gate of MOSFETs. Therefore, it is possible to control three LEDs using only one set, consisting of driver, sense resistor, flyback diode and inductor.

THE CIRCUITRY

The circuit shown in **Figure 3** is based on an ATmega8 MCU from Microchip Technology (formerly Atmel) running at 16 MHz, 5 V. The power is provided by a 12 VDC wall adapter. The PWM outputs are OC1A and OC1B pins from Timer 1 and an OC2 pin from Timer 2. Six low RDS_{ON} MOSFETs (in fact three double MOSFETs) act as level shifters and parallel switches for the LEDs. The operating mode or lighting profile can be changed by pressing push button S2 or wirelessly through an AX-1838HS infrared receiver. For a lower consumption, a seven-segment display shows the program number only when changing it (by blinking three times in approximately 1 second).

Note that a PWM signal is not applied to the ZXLD1350 ADJ pin, as several applications using this driver suggest. Instead, it only serves to keep all three LEDs off when desired. This avoids wasting energy, because otherwise the current would flow even with the three LEDs short-circuited. During normal operation, the ADJ pin is at high level. The ZXLD1350, along with the external inductor L1 and current sense resistor Rs, form a self-oscillating, continuous-mode buck converter. Due to hysteretic operation, the output current of the driver assumes a triangular waveform with a DC value of I_{AVG_ON} and a ripple named here as ΔI_{ON}, so that:

$$I_{AVG\_ON} = \frac{100\text{ mV}}{Rs}$$

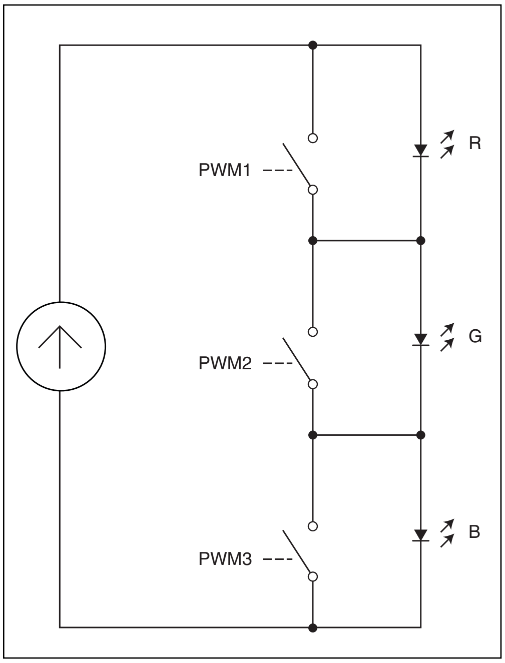


FIGURE 1
Shown here is the basic idea for the design.

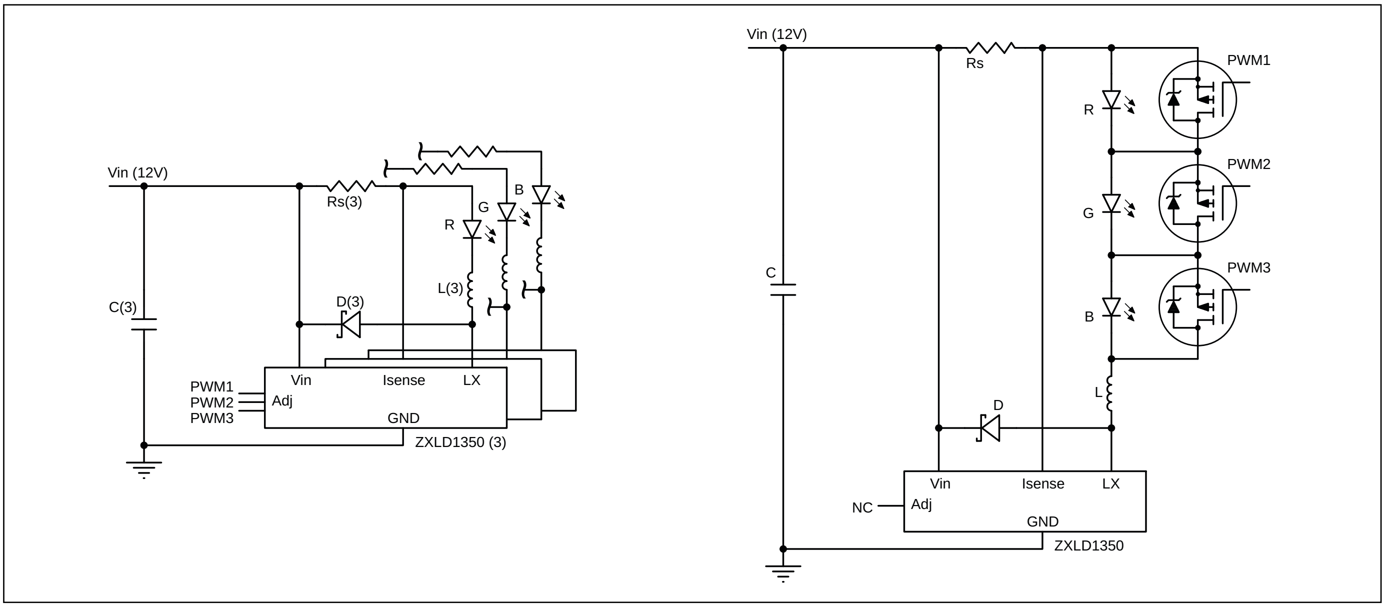


FIGURE 2
(a) shows a traditional configuration for driving 3 LEDs. (b) shows my alternative configuration, which reduces the number of components by connecting the three LEDs in series.

and

$$\Delta I_{ON} = \frac{\pm 15 \text{ mV}}{\text{Rs}}$$

with a Rs = 0.33 Ω, we get:

$$I_{AVG\_ON} \cong 303.03 \text{ mA}$$

and

$$\Delta I_{ON} \cong 91 \text{ mA}$$

For this application, we chose an RGB LED rated to 350mA/3W. In practice, each LED is controlled (short-circuited) independently according to a predetermined lighting profile, and this operation influences the shape of the current within the profile period, T. The current is always flowing through the circuit, even when all LEDs are short-circuited. This is not a concern, because we are dealing with a current source. (What changes is only the frequency of the triangular wave.) At any given time, the current is passing through the LED or through the associated MOSFET's

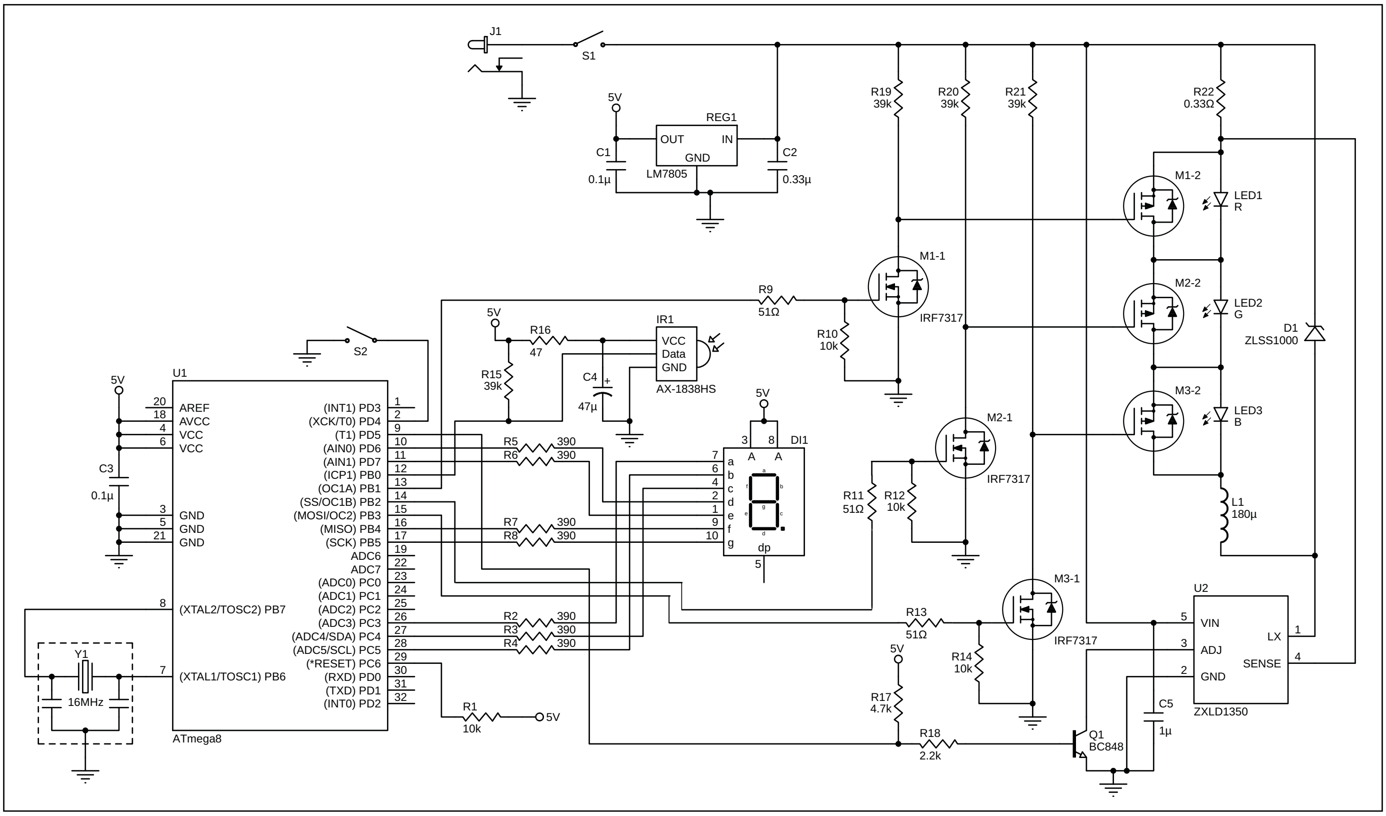


FIGURE 3
Schematic showing the circuitry connected to an ATmega8 MCU.

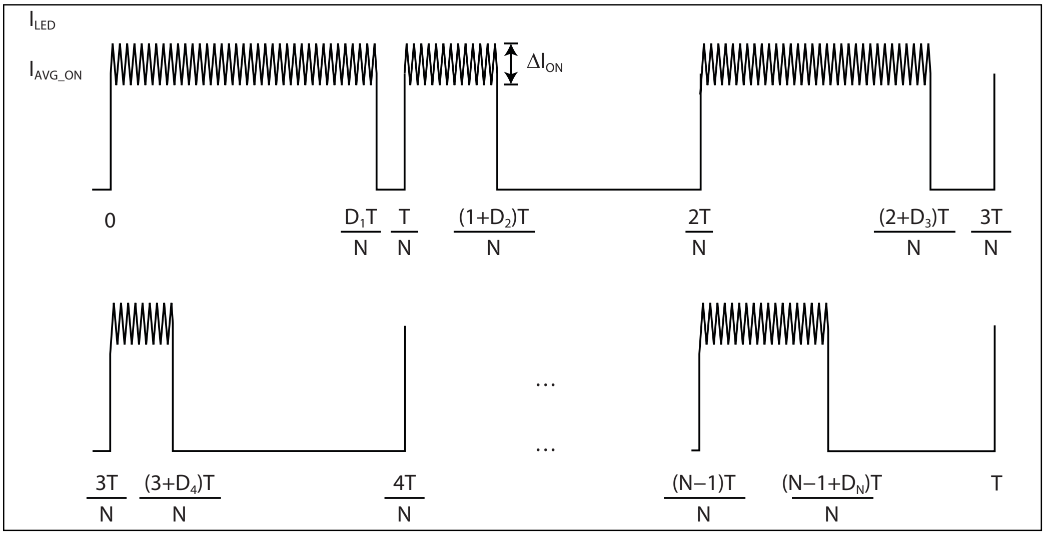
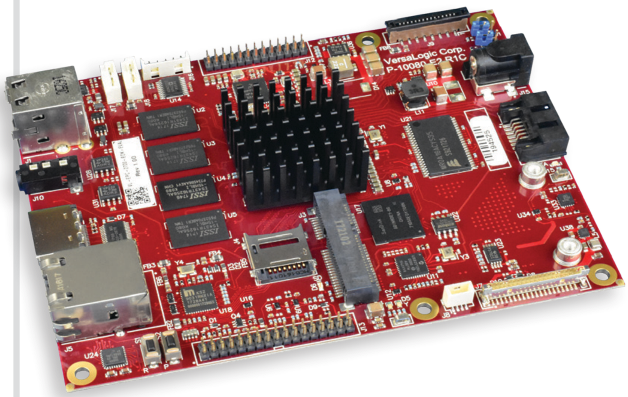


FIGURE 4
In this current waveform, you see the period divided into N intervals, each one associated with a different duty-cycle "D" when an LED is on.



VersaLogic Corp.

RDS_{ON} resistance. Next, we observe the behavior of RMS current (I_{RMS}). **Figure 4** shows that period divided into N intervals, each one associated with a different duty-cycle D when an LED is on. At the bottom of page 10, the math in box entitled "Derivation of RMS Current" shows that the RMS current, Equation (2), depends on the average value of the duty cycles belonging to a given profile. For our case:

$$I_{RMS\_ON} \cong 304.16\text{ mA}$$

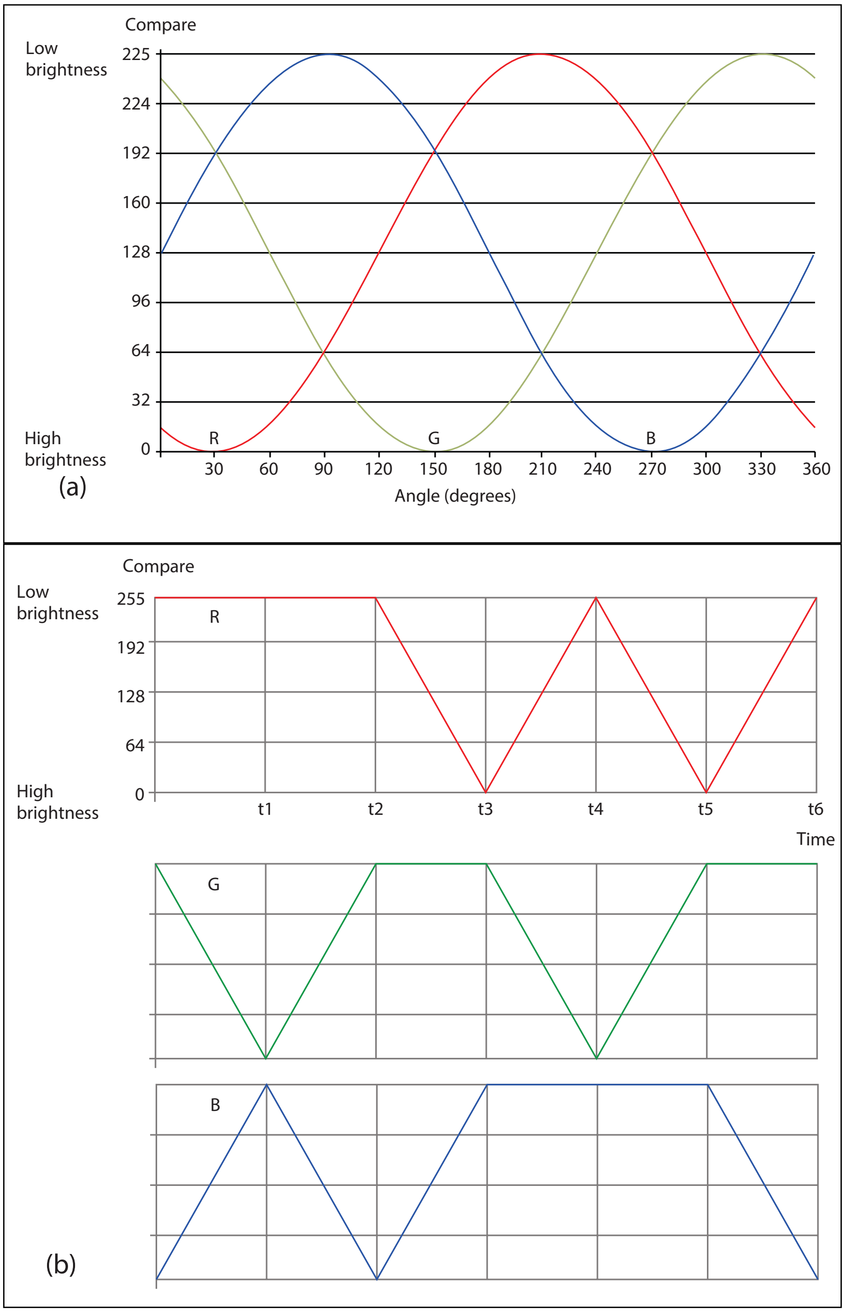


FIGURE 5
(a) Lighting Profile 1. (b) Lighting Profile 2.

FIRMWARE

The companion code associated with this article is available from *Circuit Cellar's* article code download webpage. As you will see from that source code, the PWM generation is accomplished with two outputs from TIMER1 and one output from TIMER2. With a pre-scale value of 256, these two timers are clocked at 62.5 kHz. Although TIMER1 is a 16-bit counter, only half of this was used. That's because TIMER2 is 8 bits in length, and the same resolution for all outputs is desired. Using parallel FET dimming, the switching frequency, could reach some tens of kilohertz, but the three 8-bit PWM outputs operate here at:

$$\frac{62.5\text{ kHz}}{2\times(2^8-1)} \approx 122.55\text{ Hz}$$

Note that in mode Phase-Correct/Clear-Up, the PWM frequency is halved due to up and-down counting.

The low frequency is a suitable trade-off between avoiding visible flicker and getting a nonlinear response due to PWM signal integration from the ZXLD1350 internal low-pass filter. Due to a pre-scale value of 1,024, the 8-bit TIMER0 is clocked at 15.625 kHz, and its overflow interrupt—occurring every 16.384 ms—allows the creation of three general-purpose time counters by the software. Note that due to MOSFET operation, as shown in Figure 2b, high compare values lead to low LED brightness.

The firmware, though written in the popular BASIC language, has its executable code generated by a high-performance compiler, the BASCOM-AVR from MCS Electronics. It's able to access all internal peripherals present on AVR MCUs, including handling of interrupts and execution of assembly instructions using proper directives. The application code

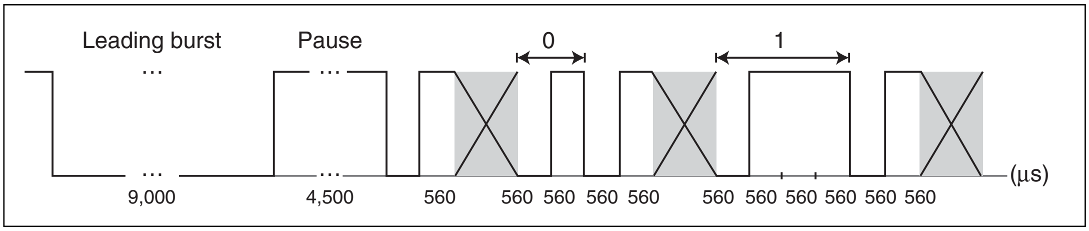


FIGURE 6
Interpretation for bit 0 and bit 1

Key	Address	Command
CH-	0x0	0x45
CH	0x0	0x46
CH+	0x0	0x47
Prev	0x0	0x44
Next	0x0	0x40
Play/Pause	0x0	0x43
VOL-	0x0	0x07
VOL+	0x0	0x15
EQ	0x0	0x09
0	0x0	0x16
100+	0x0	0x19
200+	0x0	0x0D
1	0x0	0x0C
2	0x0	0x18
3	0x0	0x5E
4	0x0	0x08
5	0x0	0x1C
6	0x0	0x5A
7	0x0	0x42
8	0x0	0x52
9	0x0	0x4A

TABLE 1
This table shows the relationship between keys and the hexadecimal values for address and command bytes.

comprises the decoding for infrared NEC protocol. To accept a wider range of remote controls—including the older ones—the code provides minimum and maximum tolerances (±10%) for the relevant timings of that protocol. Because an internal pull-up resistor on the MCU is enabled for a PD4 pin, there's no need to use this type of component on the push-button circuit. Four lighting profiles are implemented, numbered as follows:

0. LEDs off, with a low level on the ZXLD1350 ADJ pin
1. Sinusoidal three-phase with 120-degree offset, using a lookup table for sine function, as shown in **Figure 5a**
2. Two ramps (six phases, where one LED is off, one rising, and the other decreasing), as shown in **Figure 5**
3. "Flashlight" or "white LED" mode

In profiles 1 and 2, the keys "VOL+" and "VOL-" from the remote controller increase/decrease the time step. In profile 3, these keys change the power of white light, while

```
Function GetTicksBetweenEdges(ByVal maxTicks As Byte)
As Byte
  Local ticksBegin As Byte , ticksDiff As Byte

  ticksBegin = TIMER0
  ticksDiff = 0

  While PinIR <> nextEdge
    ticksDiff = TIMER0 - ticksBegin
    If ticksDiff >= maxTicks Then
      GetTicksBetweenEdges = 0  ' Error
      Exit Function
    End If
  Wend

  GetTicksBetweenEdges = ticksDiff
End Function
```

LISTING 1
This code excerpt shows one of the functions used for decoding the NEC protocol, where PinIR is an alias for PB0 pin, and the nextEdge variable is updated in an outer function.

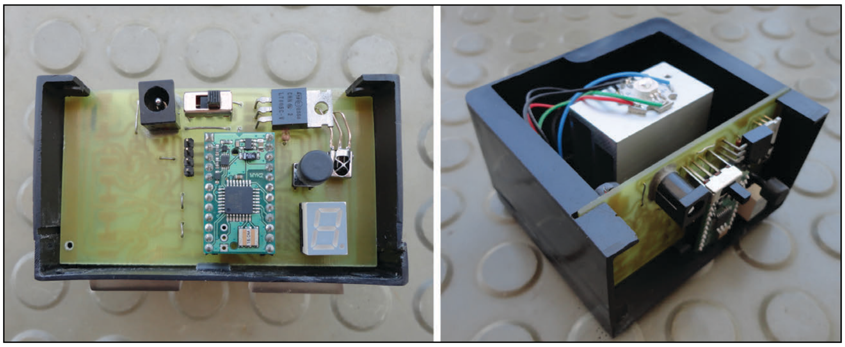


FIGURE 7
The components fit onto a small, home-made PCB, and the board was placed in a plastic box.

Derivation of RMS Current

$$I^2_{RMS} = \frac{1}{T}\sum_{n=1}^{N}\int_{(n-1)\frac{T}{N}}^{(n-1+D_n)\frac{T}{N}} i^2_{LED}(t)dt \quad \text{or} \quad I^2_{RMS} = \frac{1}{N}\sum_{n=1}^{N} I^2_{RMS\_ON}D_n$$

where:

$$I^2_{RMS\_ON} = \frac{1}{\frac{T}{N}D_n}\int_{(n-1)\frac{T}{N}}^{(n-1+D_n)\frac{T}{N}} i^2_{LED}(t)dt$$

For a triangular waveform:

$$I_{RMS\_ON} = I_{AVG\_ON}\sqrt{1+\frac{1}{12}\left(\frac{\Delta I_{ON}}{I_{AVG\_ON}}\right)^2} \quad (1)$$

therefore:

$$I_{RMS} = I_{RMS\_ON}\sqrt{\frac{1}{N}\sum_{n=1}^{N} D_n}$$

$$\text{or:} \quad I_{RMS} = I_{RMS\_ON}\sqrt{D_{AVG}} \quad (2)$$

keeping equal duty-cycle for all LEDs. The NEC IR transmission protocol uses pulse distance encoding of the message bits, at a carrier frequency of 38 kHz. On the MCU input (PB0), the received IR message, provided on AX-1838HS, consists of a 9 ms leading pulse burst and a 4.5 ms space. Each bit starts with a low time and is followed by a high time. The low time is always 560 µs, and the high time can be 560 µs (bit 0) or 1.68 ms (bit 1) (**Figure 6**). Taking this arrangement into account, each message is 4 bytes in length—address, inverted address (bitwise), command and inverted command. With the remote controller used, the relationship between keys and the hexadecimal values for address and command bytes are shown in **Table 1**.

The Timer 0 is also used to calculate the number of ticks between the edges of the IR signal. **Listing 1** is a code excerpt showing one of the functions used for decoding the NEC protocol, where `PinIR` is an alias for PB0 pin, and the `nextEdge` variable is updated in an outer function. The codes in Table 1 might appear different from those used in popular IR decoding libraries. The reason is that I decided to maintain compatibility with the NEC protocol, which sends each byte from the LSB to MSB.

The components fit into a small, home-made PCB, except the power LED, which was attached to the heatsink using a two-part, thermally conductive, adhesive epoxy. The board was placed in a plastic box in an upright position, so that the interface elements were in front of the user (**Figure 7**). A common switched-mode 12 V × 1 A wall adapter provided the power.

It is possible to place a plastic globe or other structure from an old Luminaire over the box, to spread the colored light and create a nice look. **Figure 8** shows a waveform for a typical current in one LED, and a sample of soft-start action. The datasheet for the ZXLD1350 describes the expressions for T_{ON} and T_{OFF} times, which depend on how many LEDs are on in any given instant. The waveform in Figure 9 shows a duty-cycle change for profile 1 (T_{ON} refers to descending signal).

CONCLUSIONS

Since the beginning, the main concern has been the power consumption when comparing the parallel FET dimming approach with the traditional use of three LED controllers. The minimum resistance R, common for all profiles, is assumed as 810.9 mΩ (0.33 Ω from Rs plus 480.9 mΩ from the inductor's DCR). The number of MOSFETs in plain switching on "sinusoidal" and "two-ramps" profiles are three and two, respectively. So, an

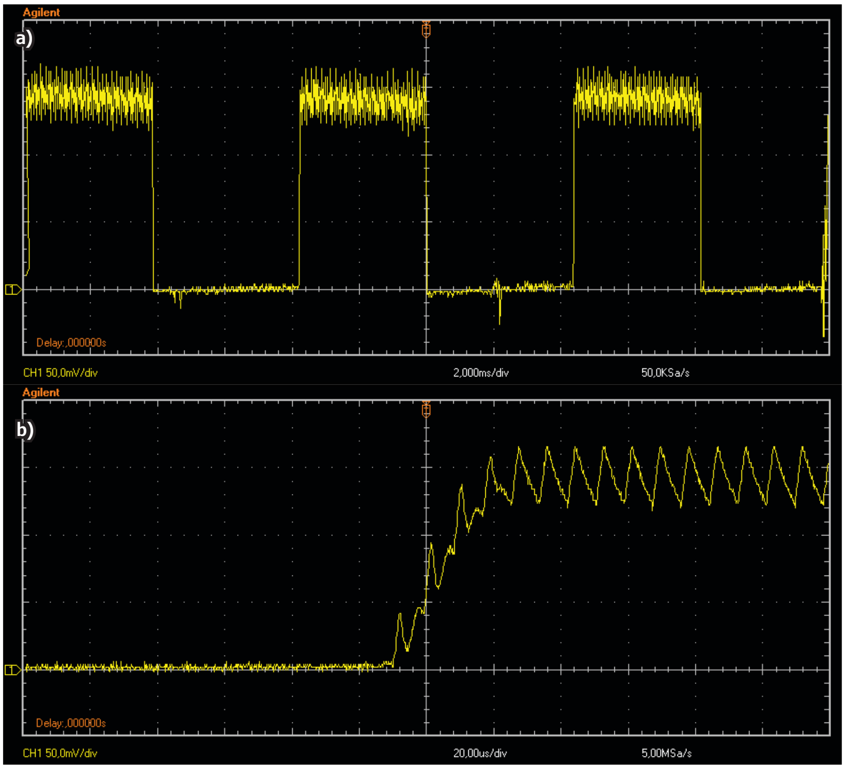


FIGURE 8
Shown here are the waveforms for (a) a typical current in one LED (the R LED), and (b) a sample of soft-start action.

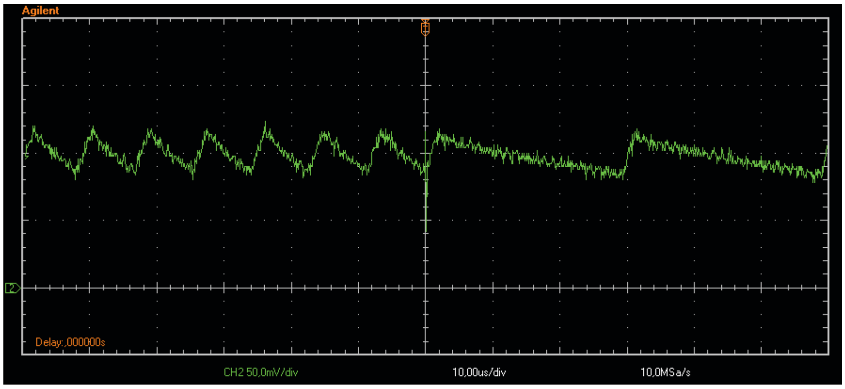


FIGURE 9
This is the waveform of duty-cycle change for profile 1.

ABOUT THE AUTHOR

Dirceu R. Rodrigues, Jr. has been a computer engineer since 1994, holding a MsC Electrical Engineering (Control Systems). He is interested in hardware and software for industrial, scientific instrumentation and embedded product design.

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

RESOURCES

Diodes | www.diodes.com

Microchip Technology | www.microchip.com

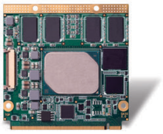
MCS Electronics | www.mcselec.com

Mode	Power Loss	
	Sinusoidal profile $D_{AVG} = 128/255$	**Two-ramps profile** $D_{AVG} = \dfrac{(255 - \frac{2}{3}255)}{255} = \dfrac{1}{3}$
Three independent controllers (PWM on ADJ pin)	$3\left(I_{RMS\_ON}\sqrt{D_{AVG}}\right)^2 R = 113 \text{ mW}$	$3\left(I_{RMS\_ON}\sqrt{D_{AVG}}\right)^2 R = 75 \text{ mW}$
Parallel FET dimming (High level on ADJ pin)	$3\left(I_{RMS\_ON}\sqrt{1-D_{AVG}}\right)^2 RDS_{ON} + I^2{}_{RMS\_ON} R = 83 \text{ mW}$	$2\left(I_{RMS\_ON}\sqrt{1-D_{AVG}}\right)^2 RDS_{ON} + I^2{}_{RMS\_ON} R = 82.2 \text{ mW}$

TABLE 2
Calculation of power consumption

equivalent number of additional series RDS_{ON} resistances (58 mΩ each) must be taken in account when calculating the power losses. We can see from **Table 2** that the calculated losses are similar.

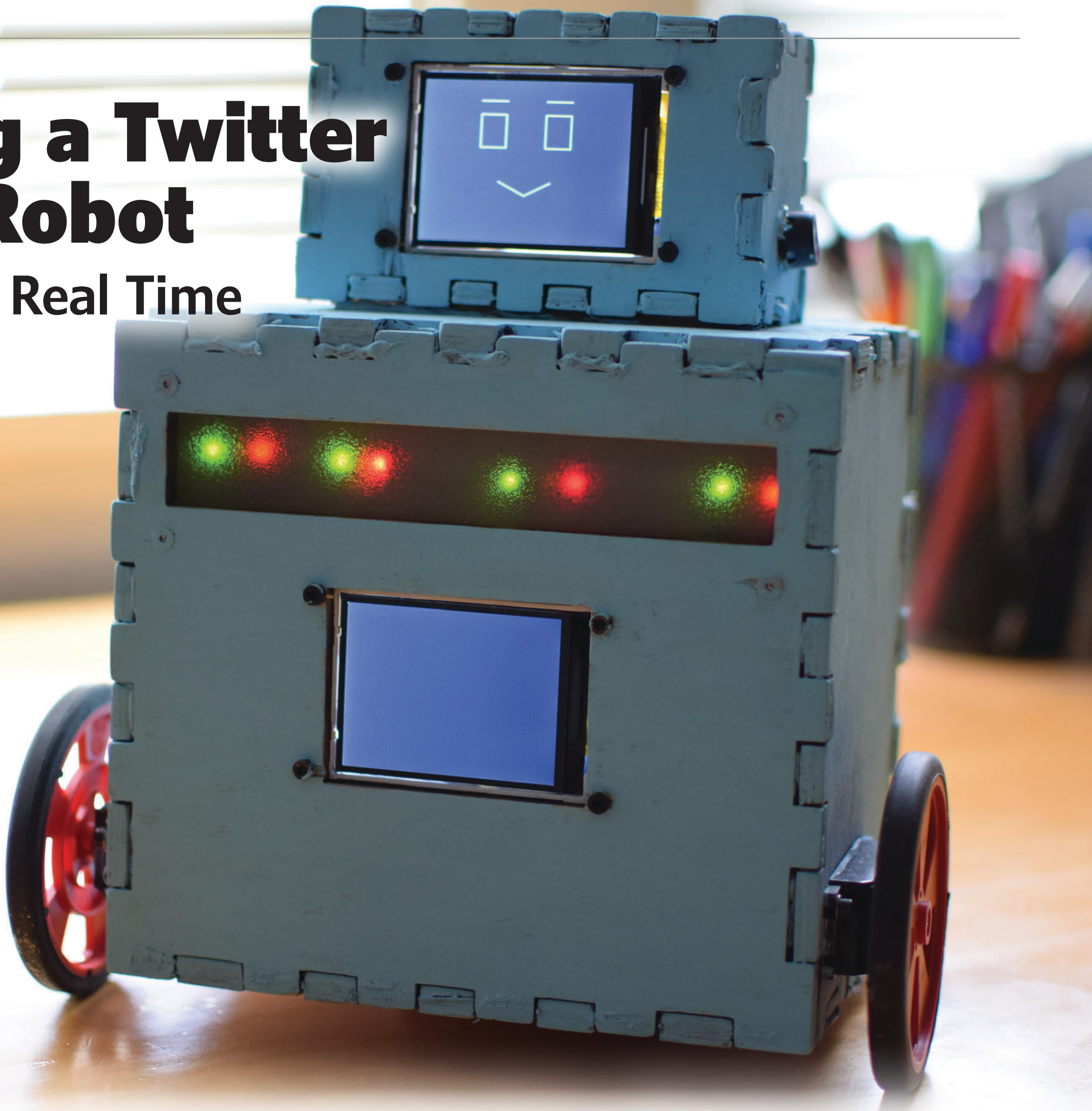
In addition to not suffering from the effect of the soft-start action, the use of parallel FET dimming reduces the bill of materials, regarding several components, to about one-third. But, in practice, this is offset by the six MOSFETs (three parts due to IRF7317 double MOSFET in the SO-8 package), at least in this implementation. You can consider alternative ways to short-circuit the LEDs, to use regulators other than the ZXLD1350 and also to provide other modifications. The number of RGB LEDs can be increased somewhat, still using only three PWM outputs. For this, replace each current LED with more LEDs in series, all of the same color. Note that this procedure could also require changes in the power supply.



Building a Twitter Emote Robot

Reactions in Real Time

By *Ian Kranz*, *Nikhil Dhawan* and *Sofya Calvin*



Social media is so pervasive these days that it's hard to image life without it. But digital interactions can be isolating, because the physical feedback component gets lost. These three Cornell students built an emotionally expressive robot that physically reacts to tweets in a live setting. Users can tweet to the robot's Twitter account and get instant feedback as the robot shares its feelings about the tweet via physical means such as sounds, facial expressions and more.

FIGURE 1
The robot is a stand-alone unit intended for tabletop use. When in rest mode, it displays a positive face and fades the red and green lights.

Social Media outlets like Twitter and Facebook have become dominant players in the field of human interaction. Indeed, many interactions have become mediated by digital technology. We believe the loss of the physical component of interaction has had negative effects on human relationships overall. In fact, according to an article in the *American Journal of Preventive Medicine* [1], people become more isolated through the use of social media.

Our project aimed to explore potential solutions to the lack of physical feedback in the world of social media. We built an emotionally expressive robot that physically reacts to tweets in a live setting. Users can tweet to the robot's Twitter account and receive nearly instant feedback, as the robot plays a sound, moves on its surface, displays the tweet text, shows a facial expression and lights up with different colors and intensities to convey its feelings about the tweet.

HIGH-LEVEL DESIGN

The robot is a stand-alone unit intended for use on a desk or table (**Figure 1**). Users communicate with the robot via Twitter by including the robot's Twitter handle—@BotCornell—in a tweet. A server application

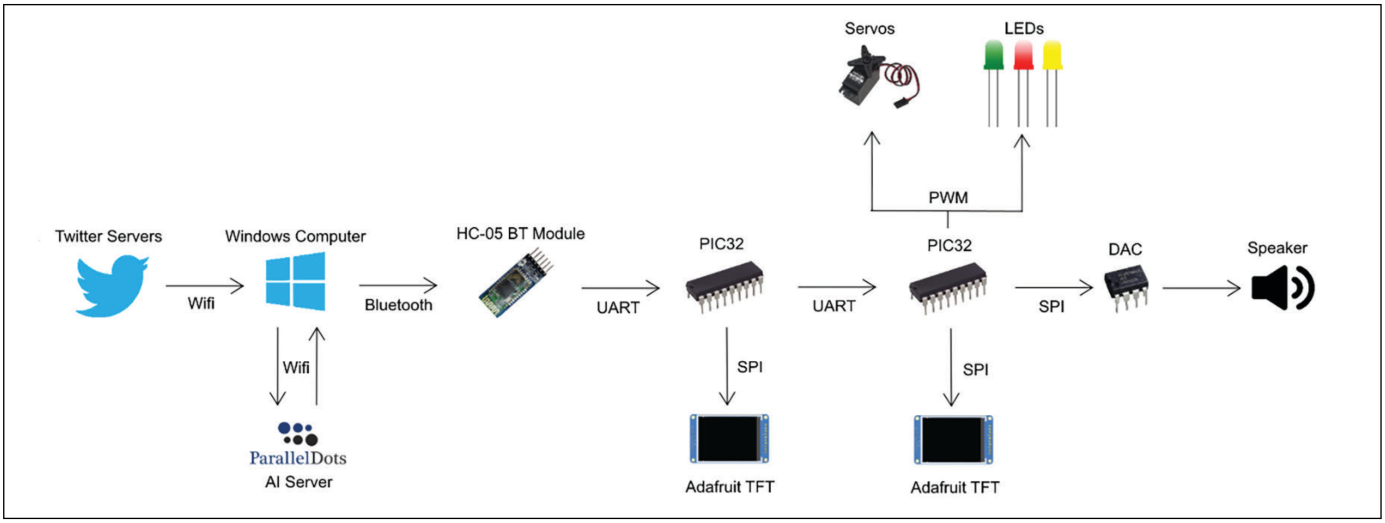


FIGURE 2
Top-level diagram depicting the connections among components in the system. The system originates on Twitter and ends with physical reactions from the robot.

written in Python runs on a laptop computer and monitors the robot's Twitter account in real time. When a tweet is received, the server processes the content, and fits the text to an emotion. The Twitter user ID, tweet content and emotion are all sent to the robot via a Bluetooth serial connection, and are processed by two Microchip PIC32 microcontrollers (MCUs). Upon receipt, the tweet content is displayed on an LCD screen on the robot's body. Simultaneously, the robot reacts to the tweet's emotion by displaying the corresponding pre-programmed response. A visualization of the system is shown in **Figure 2**.

The robot's responses are designed to physically embody the detected emotion of the received tweet. The main response is a change in facial expression, displayed on an LCD screen on the robot's face. Other responses include playing a Twitter notification sound, spinning in a circle and changing colors and intensities of lights. The robot reacts to the tweet for 20 seconds, giving the user enough time to read the tweet and experience the robot's emotional reaction. After the 20 seconds, the robot returns to a rest state, where it waits for another tweet to arrive.

When considering the final design of the robot, it was important that it have human-like traits, so that users could connect with the robot in a personal way. We therefore gave it a head and a body, both rectangular for simplicity. With intentions to keep the robot portable, we designed the body box to have dimensions of only 6" × 5.5" × 4" and the head box to be 4" × 2.5" × 2". The structure was made using laser-cut plywood and hot glue, resulting in a sturdy and finished product.

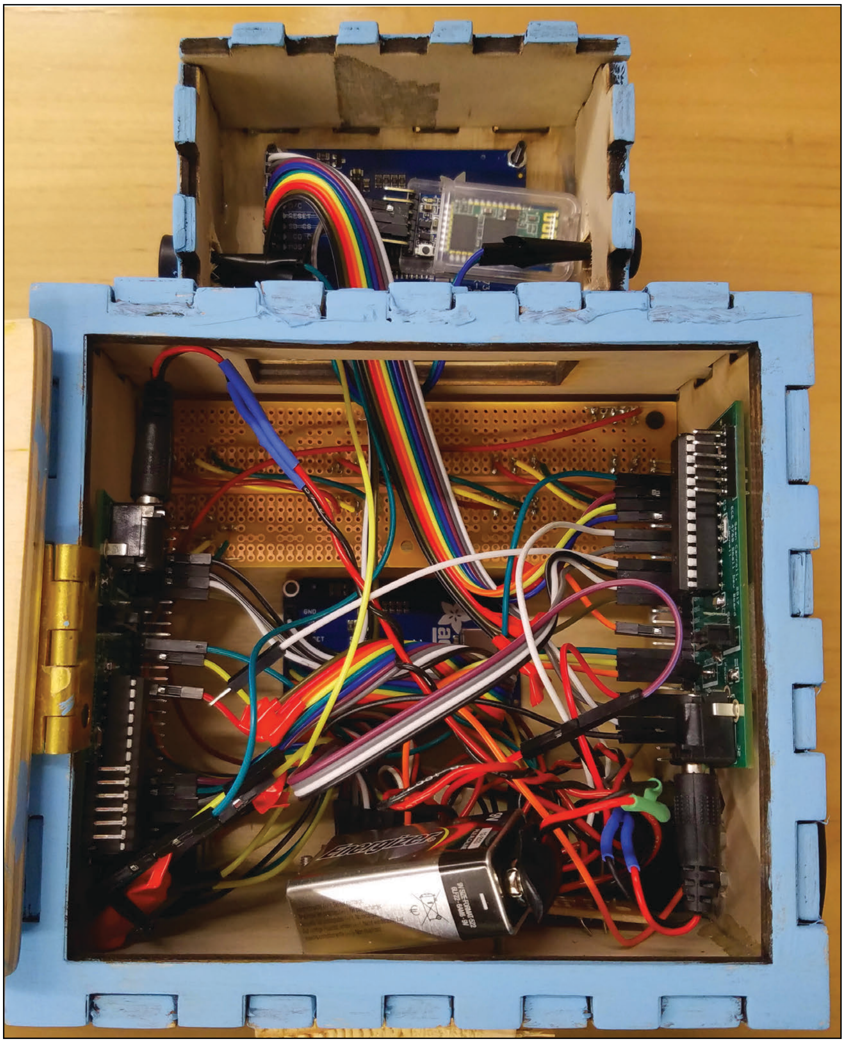


FIGURE 3
The circuitry is accessible via a rear access door. When the door is closed, no circuitry is exposed. Components are secured such that when the robot moves, connections remain closed.

The robot's face is an Adafruit Color TFT LCD Screen mounted on the head piece with screws [2]. Two speakers are attached to the sides of the head to resemble ears, further contributing to the anthropomorphic design. On the body, we added a second TFT screen identical to the one used for the face, but intended to display the tweet content to the user. Separating the two TFT screens is a rectangular cut-out, in which we placed a series of LEDs to create visual appeal during notifications. The LEDs are covered by a piece of frosted plastic to protect the circuitry and create a softened effect by dispersing the light. Finally, the robot rests on two wheels attached to the sides of the robot and controlled by servo motors. The internal circuitry of the robot is shown in **Figure 3** and the schematic in **Figure 4**.

TWEET FETCHING & ANALYSIS

Operation requires real-time access to Twitter, and the ability to pull tweet content and user ID for any tweet that mentions the robot's handle immediately after the tweet is posted. This is accomplished using Tweepy, a library created for easily accessing the Twitter API in the Python programming language [3]. We first generated API keys using a Twitter development account, which was subsequently used by Tweepy functions to connect to the Twitter API. Our PC application then spawns a thread, which asynchronously retrieves tweets in real time using Tweepy functions, and enqueues them onto a thread-safe internal buffer.

A second thread is also spawned, which monitors the internal buffer to retrieve and process tweets in the order in which they are enqueued. It is important to note that maintaining an internal buffer is crucial to the scalability of the system, as it enables the application to process tweets while simultaneously retrieving new tweets. Once a tweet has been dequeued, the script determines its emotion by making an API call to a text-to-emotion library called ParallelDots [4]. There are several text-to-emotion libraries available, however we chose ParallelDots because it is free and

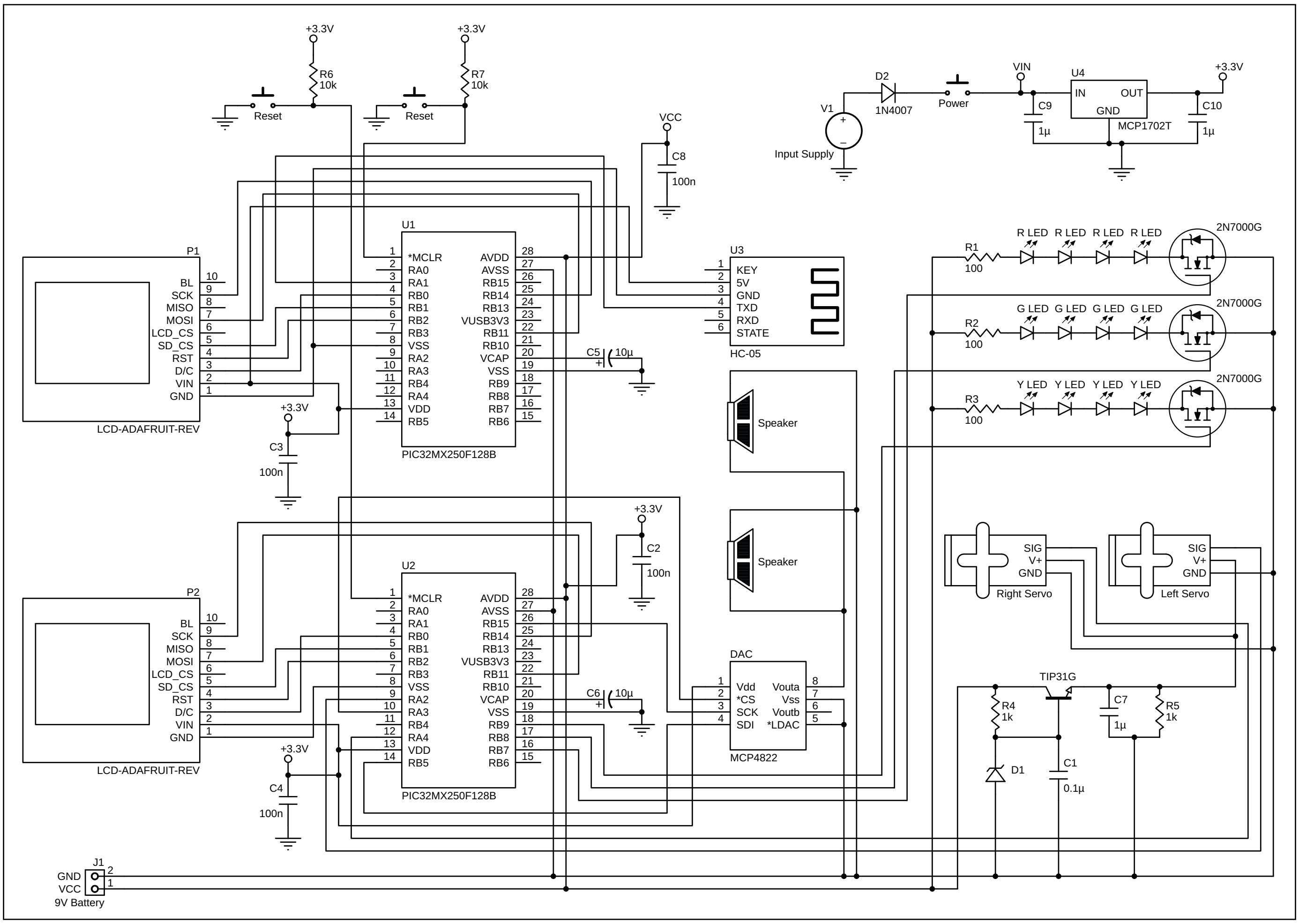


FIGURE 4

The full schematic for on-board circuitry shows the use of two PIC32 MCUs on custom PCBs. These MCUs communicate with a series of peripherals to create emotion outputs.

allows up to 1,000 API hits per day at a rate of 20 hits per minute.

The server sends tweet content, user ID and emotion choice to the robot via a serial Bluetooth channel. On the server end, we used the internal Bluetooth module on a laptop computer, controlled by the server application using the pySerial library. On the robot end, we use an HC-05 Bluetooth module with standard settings, interfaced with a PIC32 MCU mounted on a PCB via a wired serial UART connection [5][6]. Data are sent in only one direction, from server to robot, making the required software relatively simple.

To correctly and easily parse the data for each tweet on the receiving end, we send it in chunks of characters. First, we send the user ID, followed by the tweet emotion represented by a single character, and ending with the tweet content divided up such that a separate message is sent for each group of 63 characters. Of course, a tweet may contain up to 280 characters. So, for a higher degree of robustness, five chunks of tweet content are sent with each tweet. In the event that more than one tweet is sent, each tweet is sent sequentially such that the time between each full transmission allows the robot to completely react to each tweet.

The PIC32 MCU receives the data by reading the serial line from the Bluetooth HC-05 module. This is accomplished in software using a peripheral device library (provided by Bruce Land of Cornell University) to read from the UART buffer [7]. Once all the data have been transferred to the robot, the process of reacting begins, as described in the following section.

REACTION OUTPUT MAPPING

Affixed to the robot body are a series of output devices, all controlled by two PIC32 MCUs and powered by an on-board 9 V battery. In a sense, these output devices give the robot life through sound, movement, light and countenance.

After sitting still for some time waiting for an incoming tweet, the robot first grabs the user's attention with a Twitter notification sound resembling the high-pitched chirping of a bird. We acquired the sound, which was sampled at 44,100 Hz, from an online sound effects library, and extracted the raw samples using a simple MathWorks MATLAB program. The samples were scaled between 0 and 4,095, because we synthesized the sound digitally with a 12- bit DAC, and values outside this range would have been misinterpreted by the hardware. To output the audio samples to the DAC, we set up a DMA channel to send the samples over an SPI channel. This involved

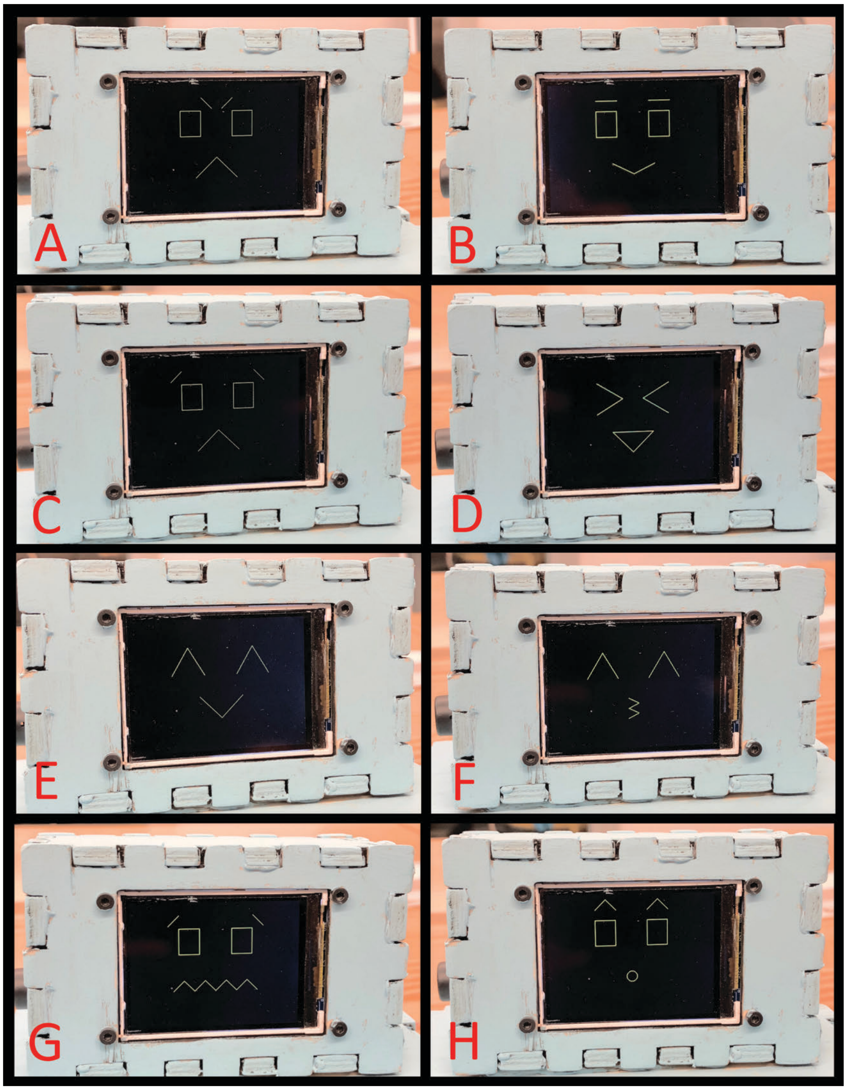


FIGURE 5
The robot's facial expressions for different emotions. A. Angry B. Resting C. Sad D. Excited E. Happy F. Bored G. Afraid H. Sarcastic. These expressions were drawn based on emojis.

ABOUT THE AUTHORS

Ian Kranz (imk33@cornell.edu) is a recent graduate of the Cornell University College of Engineering. He is currently pursuing a career in Electrical Engineering while keeping up some hobbyist electronics projects on the side.

Nikhil Dhawan (nd353@cornell.edu) is a recent graduate of the College of Engineering at Cornell University, where he obtained a Bachelor's degree in Computer Science with minors in Electrical and Computer Engineering and Business. Currently, Nikhil is a software engineer at Microsoft in Redmond, WA.

Sofya Calvin (sec293@cornell.edu) graduated from Cornell University with a Bachelor's degree in Electrical and Computer Engineering. She is currently a Technology Consultant at Accenture.

setting the DMA channel to framed mode, which allows the DMA channel to move whole blocks of memory at a time into the SPI buffer. The transfer is initiated immediately after the robot receives a Bluetooth transmission for a new tweet.

The TFT screens are driven by the PIC32 MCUs over a serial connection, using a TFT software library provided by Syed Tahmid Mahbub [8]. Functionality of the library includes line-drawing functions, which are used to draw the robot's facial expressions, and a write string function, which is used to print the tweet content to the body screen. Due to the large number and nature of the I/O lines to drive one screen, two PIC32 MCUs must be used to drive the two screens. The MCUs communicate emotion data over a serial UART line, allowing them to simultaneously display the correct emotion.

The content screen is updated by writing the user ID and tweet content to the body screen with formatting, so that the text is readable. The face screen works on a case statement that selects a pre-programmed facial expression based on the underlying emotion (as determined by text-to-emotion analysis), and draws it to the face screen (**Figure 5**).

PROGRAMMING THE LEDs

To achieve more visual stimulation, we programmed the LEDs on the robot to constantly change intensity. The robot has 12 LEDs (4 green, 4 yellow, 4 red), with all LEDs of the same color connected in series with resistors to limit current. Each group is controlled individually by 2N7000 MOSFETs acting as a low-side switches. 2N7000 MOSFETs are made by a variety of companies, including ON Semiconductor. The gates of the MOSFETs are connected to digital PWM signals from the PIC32 MCU, allowing individual control of the brightness of each color group.

The lights are employed in the design by implementing four states for each color: on, off, blinking and fading. The blinking function toggles the group of lights at a settable frequency. The fading function is slightly more complex and fades the light off and on at a settable frequency, according to a discrete exponential function—causing the light to have an apparent "breathing" effect. These effects, in combination with the colors, are used to create a level of intensity associated with each emotion the robot displays.

Although the inclusion of visual and sonic stimulation allows for a wonderful means of expressing emotion, we felt that the robot was lackluster without some element of motion to catch the user's attention. We added continuous rotation servos and wheels as a final component, to enable the robot to move on its surface. Upon receiving a tweet, the robot spins in place a full 360 degrees. We chose spinning in place for its simplicity, and because we wanted the robot to be functional in areas with limited space. To enable this movement, we power the servos with 5 V, which is stepped down from the 9 V battery by a DC-DC converter circuit. A PWM signal from the PIC32 controls the rotation direction and speed of the wheels. The signal is put on a timer, so that the wheels turn only for the time it takes to complete one full rotation—the amount of time was determined empirically.

RESULTS AND USER EXPERIENCE

When deployed, our robot responded successfully to tweets under a reasonable load, while maintaining user engagement. The system also demonstrated relatively low latency when in use. After the user tweets, and the robot has no other tweets enqueued in its internal buffer, it takes an average of 6.89 seconds to respond—effectively real time. The main bottleneck during this process is the amount of time it takes for Tweepy to retrieve an incoming tweet, which typically is 6.25 seconds on average. After the tweet is received, the robot reacts almost instantaneously from the user's point of view.

To capture the user's attention, we carefully designed the robot's facial expressions, lighting, sound, and motion. As shown in Figure 5, the robot was capable of displaying eight distinct facial expressions: resting, happy, sad, angry, excited, afraid, bored and sarcastic. We modeled the robot's facial expressions after emojis, which we expected the user base to easily identify.

Lighting also had a major impact on the user's perception of the robot. At rest, we wanted the robot to convey a generally relaxed yet friendly and up-beat demeanor. Consequently, we programmed the green

FIGURE 6
This QR code links to a video demonstration of the robot. The video includes a high-level description of our project. The video is also available on Circuit Cellar's article materials webpage.

For detailed article references and additional resources go to:
www.circuitcellar.com/article-materials
References [1] through [8] as marked in the article can be found there

RESOURCES

Adafruit | www.adafruit.com

DSD Tech | www.dsdtech-global.com

MathWorks | www.mathworks.com

Microchip Technology | www.microchip.com

ON Semiconductor | www.onsemi.com

ParallelDots | www.paralleldots.com

lights to fade on and off slowly, such that the robot appeared to be "calmly breathing." In general, we mapped the sentiment—positive, negative or neutral—of the emotion to the color of the LEDs, and the intensity of the emotion to the blinking speed. For example, when the robot was angry, it quickly flashed the red LEDs—conveying an intense negative feeling. Alternately, when the robot was happy, the yellow and green LEDs blinked on and off somewhat quickly.

Finally, the robot's motion was chosen to be as unobtrusive as possible. Spinning in a circle allows the robot to be placed in a relatively confined space. To see the robot in action, use the QR code in **Figure 6** to link to our demonstration video. The video is also available on *Circuit Cellar's* article materials webpage.

FUTURE IMPROVEMENTS

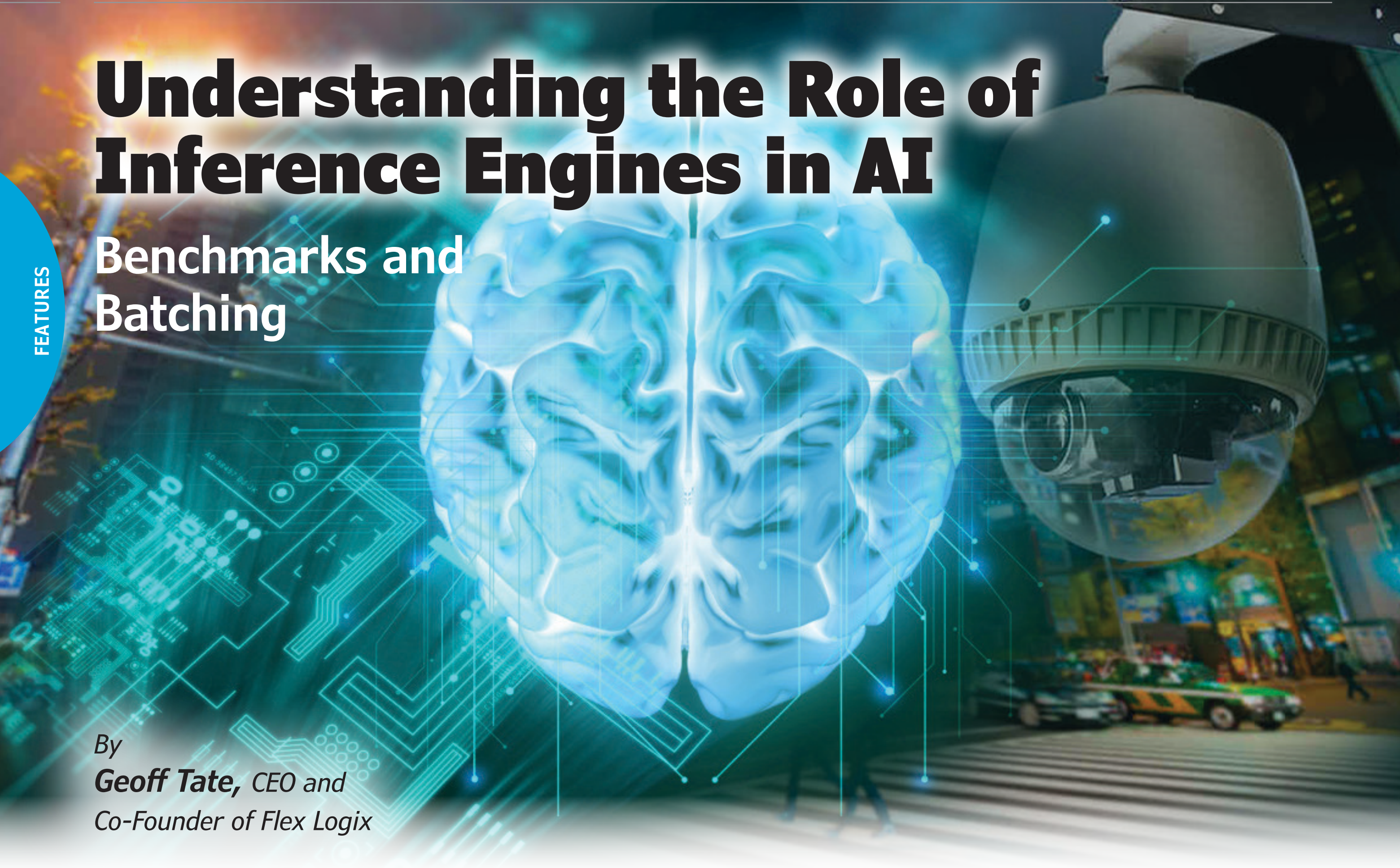
All in all, we were extremely satisfied with our final robot, which was able to successfully implement all of its intended features and looked like a finished product. That being said, we were severely limited in our device choices by a budget of only $120. If we were to move forward and revise our project with increased funding, there are many ways we could improve the user experience overall to create a more polished and professional product.

One of the most effective ways to enhance the quality of our robot would be to make changes to the two TFT displays. With a higher resolution screen and additional time, we would be able to improve the graphics on the face tremendously. This would include making the faces more realistic, and enhancing the expression of emotion by animating the faces. An example of this might be programming the face to shed a tear when a sad tweet is identified. Furthermore, the tweets that were displayed on the second screen were rather small and not formatted to be visually appealing. Therefore, we would edit these to be more eye-catching and easier for the user to read at a glance.

Animating the resting state would have also improved the user experience. However, implementing such animations would require a change to the protothread scheduler. We initially planned to add resting animations such as blinking, but that proved to be difficult because of the cooperative scheduler that managed all the running threads. For example, if we added a thread to handle blinking, the thread waiting on emotion information could miss an incoming UART transmission due to cooperative scheduling.

Another potential improvement is to enhance the appeal to emotions by currently implemented features. This can be done in two ways. First, we could use multicolor LEDs to allow for the inclusion of more colors. This would be an important enhancement, because there are many studies that relate certain colors to certain emotions. With that in mind, including more colors in addition to red, yellow and green would allow us to associate the color display more strongly with the intended emotion.

In a similar fashion, we could program the motion of the robot to be more sophisticated and act as a further reflection of emotion. For example, on a happy tweet, it could spin in a circle, but for a sad tweet, make a subtle scooting motion back and forth. The combination of these improvements would make the overall user experience more luxurious and memorable as a physical embodiment of social media.



Understanding the Role of Inference Engines in AI

Benchmarks and Batching

By
Geoff Tate, *CEO and Co-Founder of Flex Logix*

Artificial Intelligence offers huge benefits for embedded systems. But implementing AI well requires making smart technology choices, especially when it comes to selected a neural inferencing engine. In this article, Flex Logix CEO Geoff Tate explains what inferencing is, how it plays into AI and how embedded system designers can make sure they are using the right solution for their AI processing.

Clearly, both software and hardware companies alike recognize that AI—if done right—represents a huge market potential. The question is: How do you do it right? The answer lies in the neural inferencing engines being developed that will power AI in the future. Similar to an engine in an automobile, the inferencing engine determines how well, how fast and how efficient the vehicle will run.

AI, machine learning and deep learning are all terms for neural networks which are designed to classify objects into categories after a training phase. What is becoming clear is that the traditional Von Neumann processor architecture is not optimal for neural networks. Artificial intelligence [1] requires powerful chips for computing answers, which is called inferencing, from large data sets, which is the training part. Inference is the part of machine learning when the neural net uses what it has learned during the training phase to deliver answers to new problems. This is particularly important in edge applications, which we define as anything outside of the data center.

The edge inferencing market is expected to be one of the biggest over the next five years. Typical applications may include smart surveillance cameras and real-time object recognition, autonomous driving cars (**Figure 1**) and other IoT devices. In the past, most inferencing engines were developed for the data center. However, the movement of AI to the edge of the network requires a new generation of specialized processors that are scalable, cost effective and consume extremely low power.

Picking the right inferencing engine is a critical factor in developing effective AI solutions. When looking at AI, it's all about throughput and good inferencing engines provide very high throughput. The problem however, is that companies don't know how to distinguish a good inferencing engine from a bad one. Without the establishment of standard benchmarks, companies have been throwing out random performance figures that don't really matter to the overall throughput. Unless a designer has spent significant time in this space, they are going to have a hard time figuring out what benchmarks really matter.

POPULAR BENCHMARK: TOPS

When the performance of inferencing engines come up, vendors often cite benchmarks such as TOPS (tera-operations/second) performance and TOPS/W. System/chip designers looking into these soon realize that these figures are generally meaningless. What really matters is the throughput an inferencing engine can deliver for a model, image size, batch size and process and PVT (process/voltage/temperature) conditions. This is the number one measurement of how well it will perform, but amazingly very few vendors provide it.

The biggest problem with TOPS is that when a company says their engine does *X* TOPS, they typically quote this without stating what the conditions are. Without knowing this information, they erroneously believe that a *X* TOPS means it can perform *X* trillion operations. In reality, a company quoting 130 TOPS may only deliver 27 TOPS of useable throughput.

Another benchmark being used, but less commonly, is ResNet-50. The problem with this benchmark is that most companies quoting it don't give batch sizes. When they don't provide this information, a chip designer can assume it will be a large batch size that maximizes their hardware utilization percentage. This makes ResNet-50 not very helpful as a benchmark. In contrast, YOLOv3 for example requires 100 times more operations to process a 2 Megapixel image. Hardware utilization will be even more challenged on "real-world" models.

The biggest difference between ResNet-50 and YOLOv3 is the choice of image size. For example, if ResNet-50 is run using 2 Megapixel images like YOLOv3, the MACs/image increase to 103 million and the largest activation to 33.6 MB. On large images, ResNet-50's characteristics looks close to YOLOv3.

KEY FACTORS

There are several key factors to look at when evaluating neural inferencing engines. The first requirement is to define what an operation is. Some vendors count a multiply (typically INT 8 times INT 8) as one operation and an accumulation (addition, typically INT 32) as one operation. Therefore, a single multiply-accumulate equals 2 operations. However, some vendors include other types of operations in their TOPS specification so that must be clarified in the beginning.

It's also important to define the operating conditions. If a vendor gives a TOPS value without providing the conditions, they are likely using room temperature, nominal voltage and typical process. Usually they will mention which process node they are referring to, but operating speeds differ between different vendors, and most processes are offered with 2, 3 or more nominal voltages. Since performance is a function of frequency, and frequency is a function of voltage, a chip designer can get

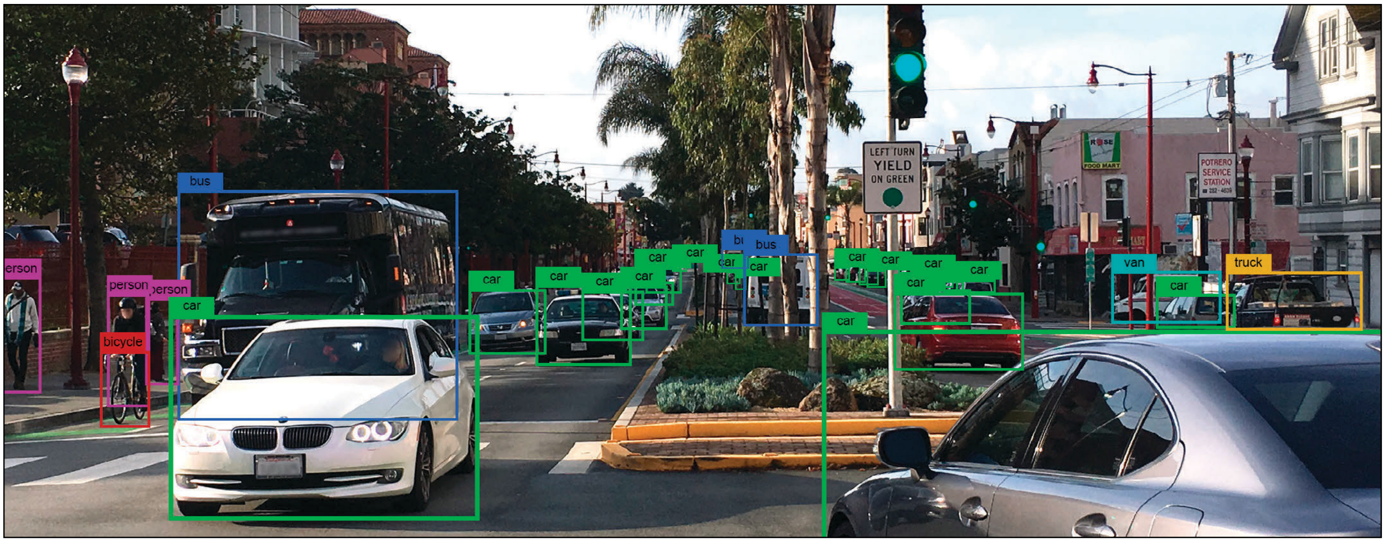


FIGURE 1
Applications requiring edge inferencing technology may include smart surveillance cameras and real-time object recognition, autonomous driving cars and other IoT devices.

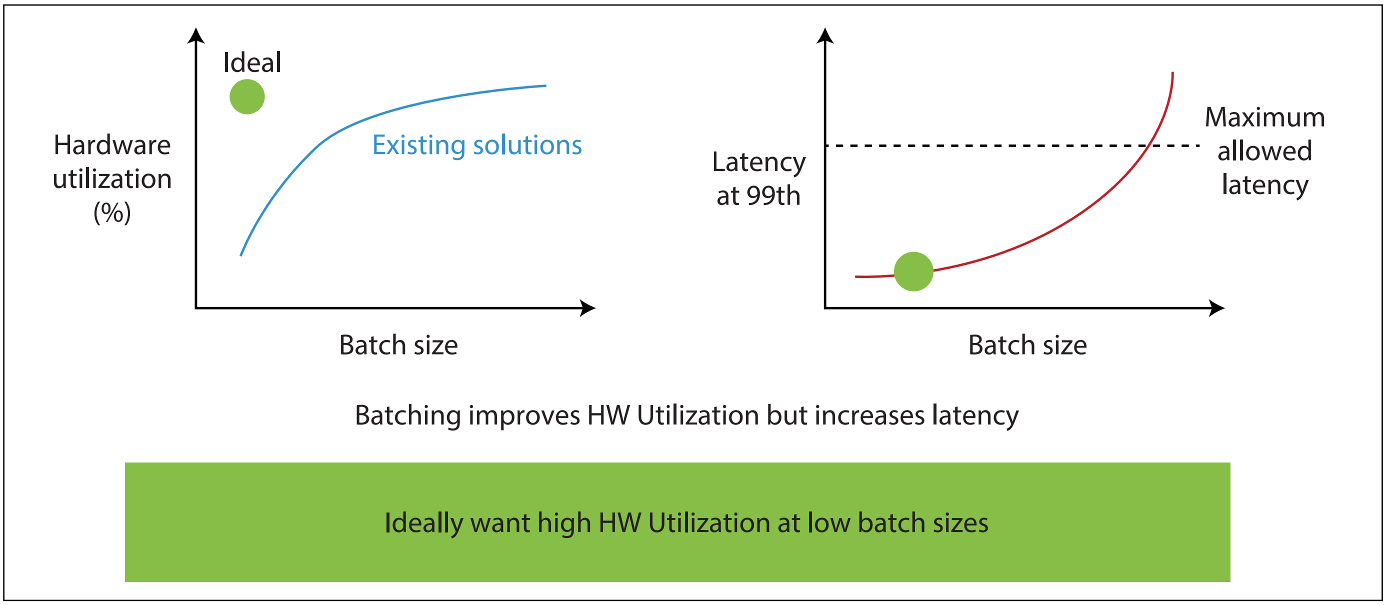


FIGURE 2
Batching improves throughputs, but the trade-off is longer latency.

more than twice the performance at 0.9 V than at 0.6 V. Frequency varies depending on the conditions/assumptions.

Batch size is also important. Even if a vendor provides worst-case TOPS, chip designers need to figure out if all of those operations actually contribute to computing their neural network models. In reality, the actual utilization can be very low because no inferencing engine has 100% utilization of all of the MACs all of the time. That is why batch size matters. Batching is where weights are loaded for a given layer and process multiple data sets at the same time. The reason to do this is to improve throughput, but the give-up is longer latency (**Figure 2**). ResNet-50 has over 20 million weights, YOLOv3 has over 60 million weights and every weight must be fetched and loaded into the MAC structure for every image. There are too many weights to keep them all resident in the MAC structure.

When it comes to MAC utilization, not all neural networks behave the same. It's important to find out the actual MAC utilization for the neural inference engine for the neural network model you want to deploy, at the batch size you require.

MOVEMENT OF MEMORY

When it comes to inferencing engines, it's important they properly manage the movement of data in memory in order to first keep the MACs supplied with the weights and activations to achieve the highest hardware utilization and second do this using the least power possible.

Table 1 depicts two popular neural network models to examine the challenge of memory in inference. We'll assume the Winograd Transformation is used for the popular 3×3, stride 1 convolutions. As Table 1 shows, caching has little benefit for neural models, which is very different than traditional processor workloads where processing is huge. The 22.7 million weights are cycled through and are not being re-used until the next image. A weights cache needs to hold all the weights, and a smaller weights cache just flushes continuously. Similarly, with activations, in some models, some activations are used again in later stages, but for the most part activations are generated and used immediately only to feed the next stage.

ABOUT THE AUTHOR

Geoff Tate is CEO/Co-founder of Flex Logix Technologies. He earned a BSc in Computer Science from the University of Alberta and an MBA from Harvard University. Geoff was Founding CEO of Rambus in 1990 and before that served as Senior Vice President of Microprocessors and Logic at AMD.

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

RESOURCES

Flex Logix Technologies | www.flex-logix.com

	ResNet-50	YOLOv3
Image Size	224×224 pixels	2048×1024 pixels
Number of stages/layers	50	>100
Number of weights	22.7 million	61.9 million
Number of MACs per image	2.5 billion	227.3 billion
Largest intermediate activation	0.8 MB	64 MB
Size of all activations	9.3 MB	475 MB

TABLE 1
The table depicts two popular neural network models to examine the challenge of memory in inference.

Therefore, for each image processed by ResNet-50, the memory transactions required are listed below, assuming for now that all memory references are to/from DRAM:

- 0.15 MB input image size read in
- 22.7 MB weights read in (assuming 8-bit integers which is the norm)
- 9.3 MB of activations are written cumulatively at the end of all of the stages.
- All but the last activation is read back in for the next stage for another almost 9.3 MB

This gives a total of 41.4 MB of memory read/writes per image. We are ignoring here the memory traffic for the code for the accelerator since there is no data available for any architecture. Code may benefit from caching. Memory references to DRAM use about 10-100x the power of memory references to an SRAM on-chip.

To reduce DRAM bandwidth there are two options for ResNet-50. First, you add enough SRAM to store all 22.7 MB of weights on chip. Second, you add SRAM on chip to store intermediate activations so stage *X* writes to the activation cache and stage X+1 reads from it. For ResNet-50 the largest intermediate activation is 0.8 MB so 1 MB of SRAM eliminates about half of the DRAM traffic.

Now let's look at YOLOv3 to see the DRAM traffic needed without on-chip SRAM:

- 6 MB input image size (remember each pixel has 3 bytes for RGB).
- 61.9 MB weights read in
- 475 MB activations generated cumulatively as output of all of the stages written to DRAM.
- 475 MB activations read back in for the next layer.

This gives a total of 1,108 MB = 1.1 GB of DRAM traffic to process just one image! Much more SRAM is required to reduce DRAM bandwidth. 62 MB is required for weights caching and, since the largest intermediate activation is 64 MB, another 64 MB is required for activation caching. This would eliminate DRAM bandwidth, but 128 MB in 16 nm is about 140 mm^2, which is very expensive.

The practical options for cost-effective designs are an activation cache big enough for most layers. Only 1 layer has a 64 MB activation output, 2 layers have 32 MB activation outputs, and 4 layers have 16 MB activation outputs—and all the rest are 8 MB or less. Thus, there is a tradeoff here between activation cache size and DRAM bandwidth.

For weights there is no trade-off: either storage all 61.9 MB on chip or have them all on DRAM. You can see why YOLOv3 doesn't run faster with batches greater than 1. Multiple batches require saving multiple activations and the activations are too big.

The industry is trending toward larger models and larger images, which makes YOLOv3 more representative of the future of inference acceleration. Using on-chip memory effectively will be critical for low cost/low power inference.

CONCLUSION

Today, there are many vendors promoting inferencing engines but none of them provide ResNet-50 benchmarks. The only information they state typically is TOPS and TOPS/W. These two indicators of performance and power efficiency are almost useless by themselves unless you know what throughput an inference engine can deliver to you for your model, your image size, your batch size and your process and PVT. As a rule of thumb, when you are looking at these engines, it's important to remember that a good inferencing engine has high MAC utilization, consumes very low power and lets you keep things small.

Portable Digital Synthesizer

Music Using an MCU

By
T.J. Hurd** and **Ben Roberge

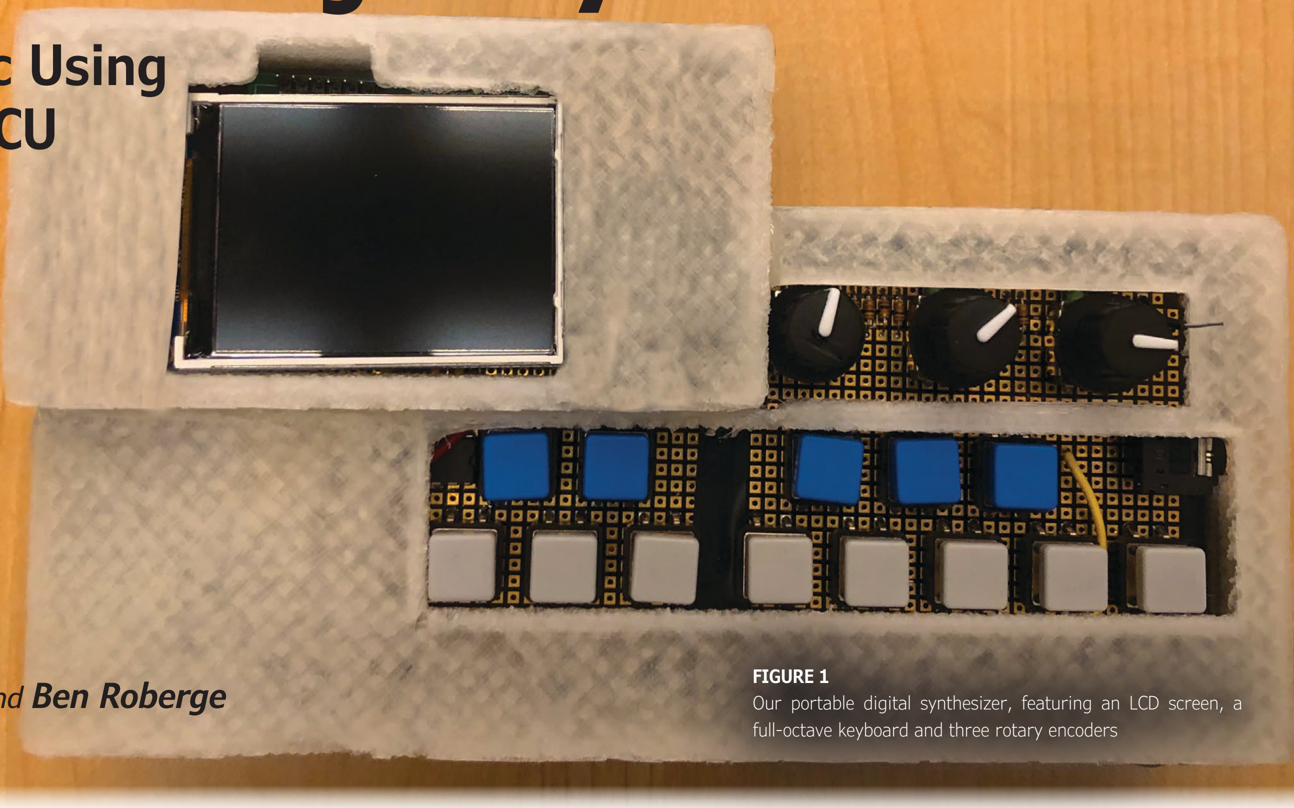


FIGURE 1
Our portable digital synthesizer, featuring an LCD screen, a full-octave keyboard and three rotary encoders

Gone are the days when even a basic music synthesizer was a bulky system requiring highly specialized design knowledge. These two Cornell students developed a portable music synthesizer using a Microchip PIC32 MCU. The portable system performs digital audio synthesis on the fly, and produces sounds that range from simple sine waves to heavily modulated waveforms.

We developed a small but powerful music synthesizer for musicians who want to make music on the go. Many current synthesizers are large and bulky, and contain a limited number of preset instruments. These limitations can obstruct creative moments and hinder experimentation. With that in mind, we wanted to create a portable device that can generate an infinite array of sounds. This way, we could provide musical inspiration for our users, wherever they may be. Our two main goals were to design compact hardware and create a powerful and flexible sound synthesis algorithm.

The system contains a full-octave (13 key) keyboard, and allows the user to play two notes simultaneously. We based the sound synthesis algorithm on frequency modulation (FM) synthesis, because it is simple to implement, yet it is capable of generating complex waveforms. The sound synthesis has 15 user-adjustable parameters, which could dramatically affect the resulting wave. These parameters are displayed on an LCD screen and can be adjusted with three rotary encoders that double as push buttons (**Figure 1**). The user needs only to plug in a power cable and either speakers or headphones to start experimenting with sounds!

An important consideration in our design was how the user interacts with the system. We wanted the user interface to be intuitive for a musician. We organized the 13 key keyboard in the shape of a traditional piano keyboard, starting at the C note. Additionally, we used rotary encoders for all variable inputs. These are used in many synthesizers and other musical devices to adjust parameters.

On our LCD screen, three parameters are displayed at a time, which correspond to the three rotary encoders. The 15 adjustable parameters were organized into four different categories or "screens," which are: Main Settings, Waveform Designer, Main Envelope and FM Envelope (**Figure 2**). The user can toggle through these four screens by pressing down on two of the rotary encoders simultaneously. Individual parameters on each screen can be toggled through by pressing down on a single rotary encoder. That allows the user to easily cycle between the screens, and quickly find an individual parameter to adjust.

HARDWARE DESIGN

Our project is based on the Microchip PIC32MX250F128B microcontroller (MCU) [1]. The PIC32 provided all the computational power necessary to interpret user input and produce the appropriate output. This MCU was integrated into a development board created by Sean Carroll [2], and the entire development board was enclosed within our project. Beyond the PIC32 MCU, several noteworthy hardware elements were integrated into our design. A full schematic of our system's hardware is shown in **Figure 3**.

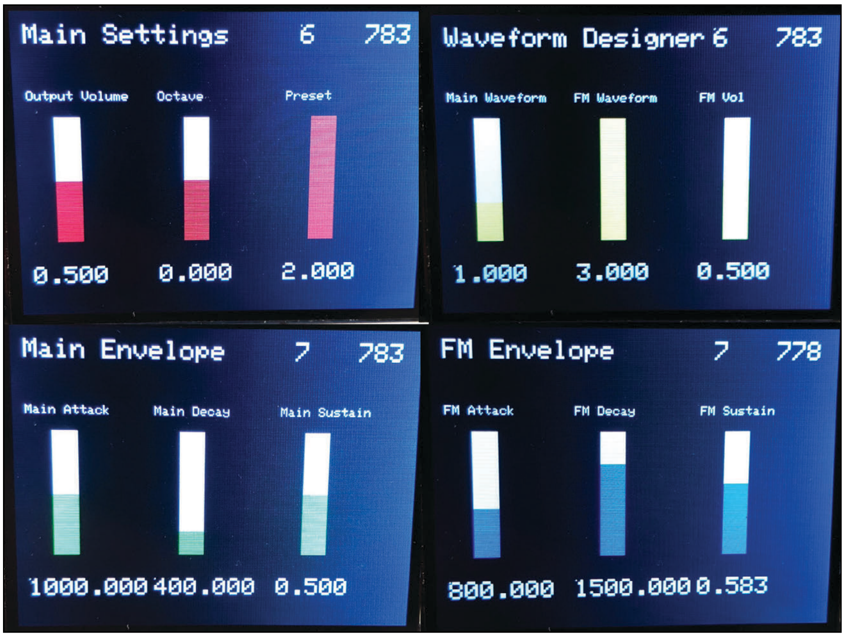


FIGURE 2
Shown here are the four screens on the synthesizer LCD display, which organize 15 user-adjustable parameters into Main Settings, Waveform Designer, Main Envelope and FM Envelope.

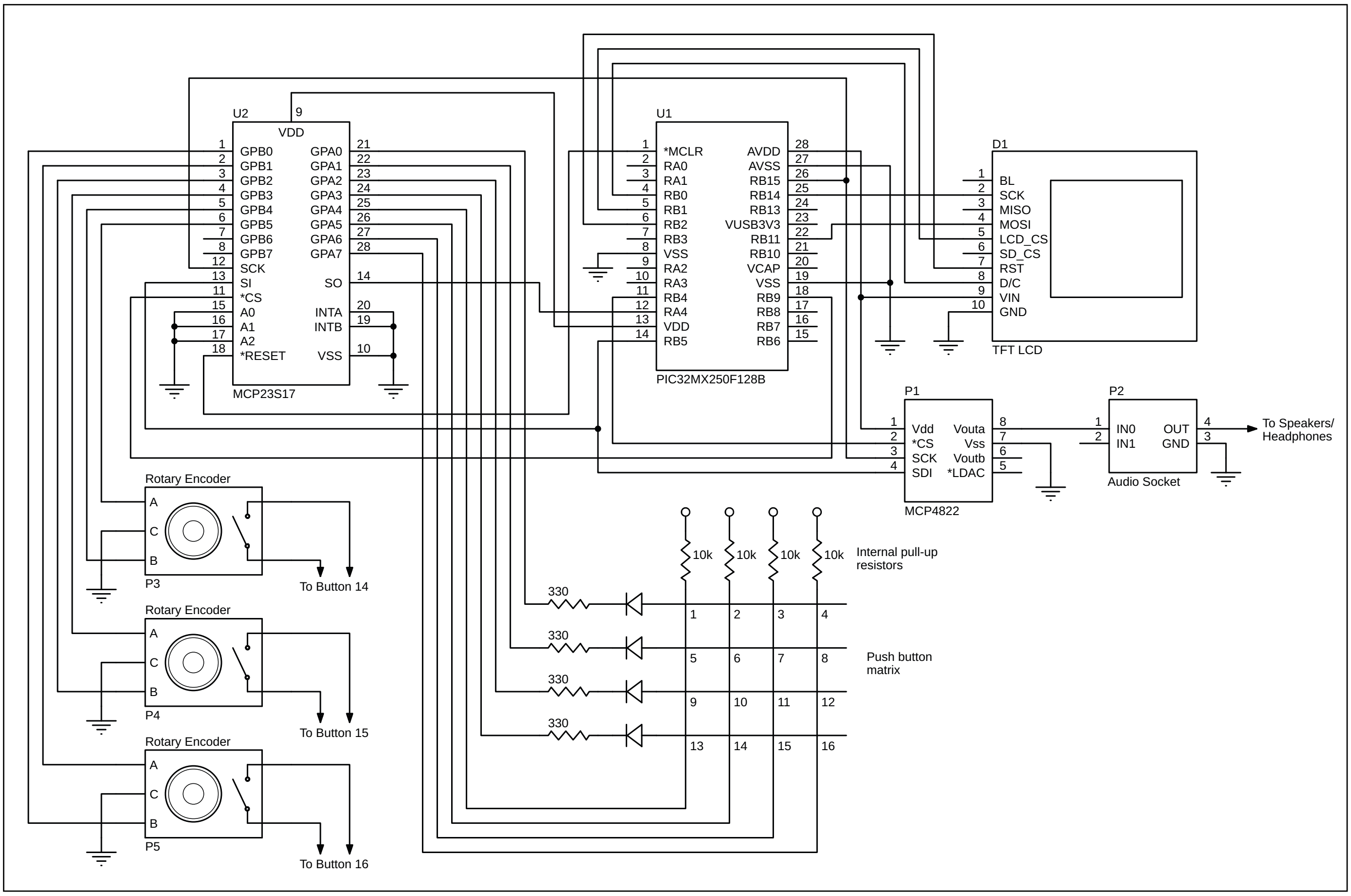


FIGURE 3
Full schematic of hardware in our portable digital music synthesizer

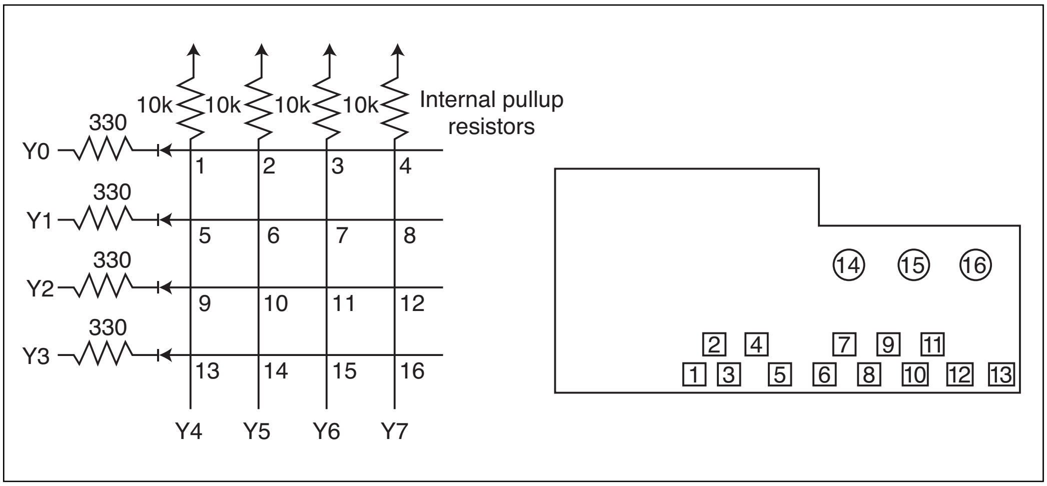


FIGURE 4
Push-button matrix schematic and diagram of push-button layout of the design. The intersections labeled 1-13 were used for the keyboard push buttons, and the intersections labeled 14-16 were used for the built-in push buttons on the rotary encoders.

The first elements were the push buttons, which enable the user to play notes and navigate on the LCD screen. We used 13 push buttons for our full-octave keyboard [3]. An additional three push buttons were built into the rotary encoders. To reduce the number of I/O pins required to handle 16 push buttons, we wired the push buttons in a 4 × 4 matrix configuration. A schematic of our push-button matrix, along with a diagram of how the push buttons were arranged in our final design, are shown in **Figure 4**. While the buttons were wired as a matrix, they were physically arranged in a more linear manner to represent a keyboard.

The four rows in the Figure 4 schematic were each connected to a separate port expander I/O pin configured as an output. The four columns were connected to separate port expander I/O pins configured as inputs. Each intersection represents a different push button. To detect button presses, logic-low pulses were sequentially sent onto the rows. The input pins connected to the columns continuously monitored for logic-low signals.

When an input pin detected a logic-low signal, the system knew a button had been pressed. To identify which button had been pressed, the system simply had to note which output pin was sending the logic-low pulse when the signal was detected at the input pin. Therefore, the button at the intersection of the row connected to the output pin and the column connected to the input pin must have been the button pressed.

ROTARY ENCODERS

In addition to the push buttons, we used rotary encoders to accept user input [4]. Their primary function in our design was to control the adjustment of system parameters. As shown in Figure 2, each system screen displayed three parameters. Each parameter could be adjusted by rotating the corresponding rotary encoder through its 24 equally-spaced, angular positions.

The rotary encoders had three pins: two signal pins and one ground pin. For each rotary encoder, we configured two I/O pins as inputs, and connected them to the two signal pins, for a total of six I/O pins. To decode the output from these encoders, we connected internal pull-up resistors to each of the six input pins. This meant these pins would be at logic-high when the encoders were in an idle state. When the user rotated one of the encoders to its next angular position, each signal pin contacted the ground pin during the transition. The critical detail here is that the signal pins do not come into contact with the ground pin at exactly the same time. During a clockwise rotation, signal pin A comes into and out of contact with the ground pin before signal pin B. During a counterclockwise rotation, the opposite is true—B comes into and out of contact with the ground pin first. The state of each signal pin during rotations is shown in **Figure 5**.

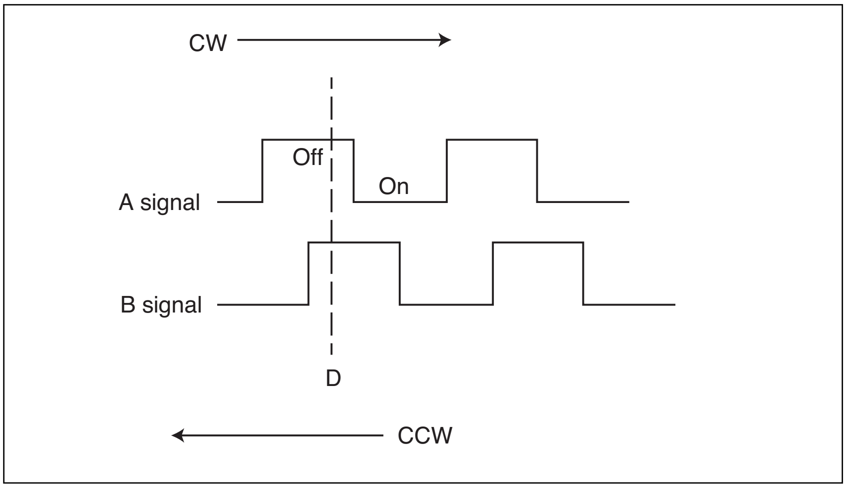


FIGURE 5
Example image of typical A signal and B signal waveforms during rotations of rotary encoder [4].

The logic to distinguish between clockwise and counterclockwise rotations is fairly simple. When a transition from logic-high to logic-low is detected on pin A, the state of pin B is checked. If B is logic-high, the rotation is clockwise. If B is logic-low, the rotation is counterclockwise. Once a rotation has been detected and distinguished, it can be used to increase or decrease the corresponding system parameter. This type of encoding scheme is known as "quadrature encoding."

Another important consideration was how to ensure that the system had enough I/O pins. To avoid running out, we added a port expander to our design [5]. This port expander used SPI to communicate with the PIC32 MCU, and provided 16 extra I/O pins. We could read it about a million times a second, which was more than adequate for our purposes. We used it for all eight pins needed for our push-button matrix, and for the six signal pins of the rotary encoders.

There were three key elements for output in our design: an SPI DAC, an audio socket and a TFT LCD. The digital audio signals produced by our system were output to the DAC via SPI [6]. The DAC, itself, was responsible for converting the digital signals into analog signals capable of being played on a set of speakers or headphones. Once converted, the analog signals were sent to a basic 3.5 mm audio socket. The user could then plug speakers or headphones into the audio socket and hear the synthesized sounds. We used a TFT LCD for visual output in our design [7]. The LCD had 320 × 240 color pixels, and communicated with the PIC32 via SPI.

The last noteworthy hardware element of our design was the enclosure. Creating a fully enclosed system was another one of our primary design goals. We felt a custom enclosure was needed to make the system as compact as possible and to protect the internal electronics. We 3D-printed the enclosure, and made the design using Autodesk Tinkercad. To give an idea of the compactness of the final enclosure, the maximum dimensions were 18.8 cm × 10.6 cm × 2.3 cm, and some sections were smaller than these maximums. We were pleased to be able to enclose all system components in such a small package, because this made the system easily portable.

SOFTWARE DESIGN

The software was organized into three major threads: The Sound Generation Interrupt Service Routine (ISR), the Input Thread and the Update Screen thread (**Figure 6**). The main focus of the software design was to optimize the execution time of each thread, since we were generating the audio in real time. Therefore, we had to make our FM synthesis algorithm very fast. Additionally, we used quick debouncing algorithms to accurately read the rotary encoder inputs. We also moved slow calculations with floating-point numbers and divisions to the slower Input Thread, which only needed to be updated on a millisecond scale within human reaction time. Finally, we updated the LCD screen only when necessary.

Sound Generation ISR: The integral part of our synthesizer is the fast Direct Digital Synthesis (DDS) [9] algorithm that is run in the Sound Generation ISR. The ISR is triggered by a hardware timer to run every 2,000 cycles, which results in a 20 kHz sampling rate. At this sampling rate, we can

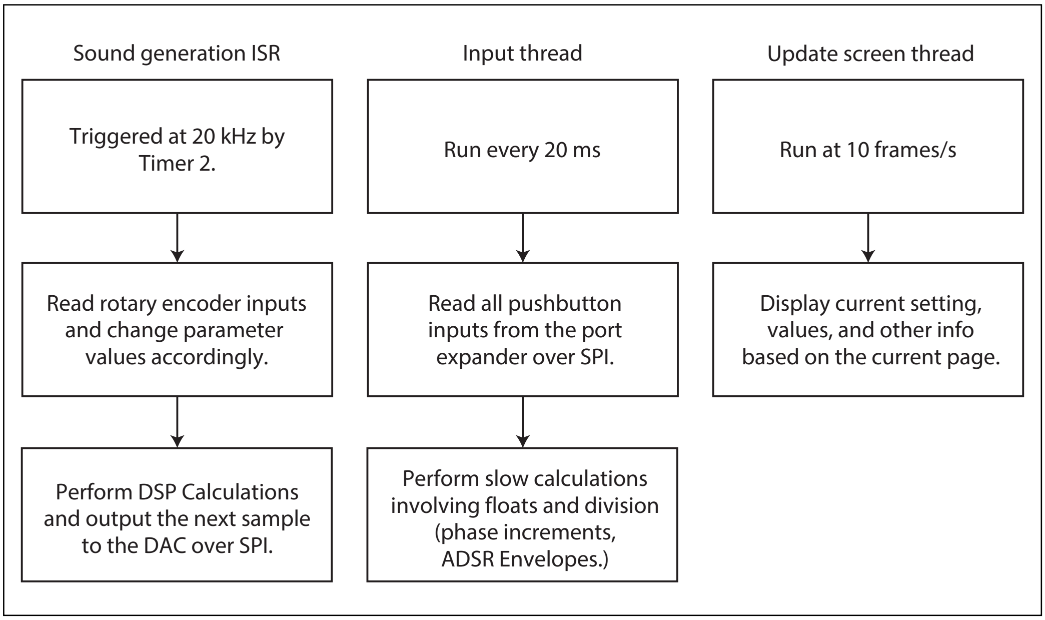

Sound generation ISR
Input thread
Update screen thread
Triggered at 20 kHz by Timer 2.
Run every 20 ms
Run at 10 frames/s
Read rotary encoder inputs and change parameter values accordingly.
Read all pushbutton inputs from the port expander over SPI.
Display current setting, values, and other info based on the current page.
Perform DSP Calculations and output the next sample to the DAC over SPI.
Perform slow calculations involving floats and division (phase increments, ADSR Envelopes.)

FIGURE 6
Block diagram illustrating the three parallel threads of our software. Protothreads [8] was used to implement the threads.

generate frequencies up to 10 kHz. Although humans can hear up to 20 kHz, we chose this rate because most people can't discern differences in the 10-20 kHz range. Additionally, this allows our notes to range up to a C8 note (4,186 Hz), with room for harmonic frequencies. Every 2,000 cycles, the ISR reads all the current values of the various parameters, and performs quick, fixed-point arithmetic to calculate the next sample to output to the DAC over SPI.

But how do we calculate the next sample to output? To create a wide variety of sounds, we use FM synthesis [10]. Our FM synthesis algorithm is based on two oscillators: a carrier and a modulator. The carrier wave corresponds to the main pitch of the note we want to generate—for example, 440 Hz is an A4 note. The modulator wave alters the frequency of the carrier wave, which creates harmonic frequencies and drastically affects the timbre of the note. To allow even more variation, we apply amplitude modulation to both the carrier and modulator, which affects the volume of the sound over time. This amplitude modulation is defined by an ADSR envelope [11], which stands for "attack, decay, sustain and release."

The function:

$$y(t) = A_c(t)wave_c(2\pi f_c t + A_m(t)wave_m(2\pi f_m t))$$

determines the output of our FM synthesis. The frequency of the carrier wave function (*wave*$_c$) is affected by the modulating wave (*wave*$_m$). Both the carrier and modulator waves have amplitudes that change over time.

Therefore, to perform FM synthesis, we need to create two waves. Various waveforms (sine, square, sawtooth and noise) are stored in wave look-up tables. To choose the correct value in the wave table, we use 32-bit phase accumulators and phase increments for both the carrier and modulator. To generate a fixed-frequency wave, the constant phase increment is added to the phase accumulator every ISR call. A large phase increment steps quickly through the wave look-up table, generating a high-frequency wave, and a small phase increment creates a low-frequency wave. Using this method, we create the modulator wave. The carrier wave is generated in a similar way, but a scaled value of the modulator wave is added to the phase increment of the main wave. This is how the modulator wave changes the frequency of the carrier wave.

The use of phase accumulators/increments makes our sound generation very efficient. All the phase increments are calculated in a separate thread, since they involve slow float division. Therefore, the ISR only needs to do a quick integer addition, a bit shift and a look-up to an array. Due to the many user-adjustable parameters, the ISR also performs several multiplications to determine the output volume, modulator volume and ADSR envelopes. The values of these Params range from 0 to 1, and are GCC standard, 32-bit, fixed-point variables with 16 integer bits and 15 fraction bits [12]. Fixed-point multiplication was used to decrease the execution time. Multiplication with fixed-point variables takes about 28 cycles, compared to about 55 cycles for float multiplication.

In addition to calculating the next sample to output, we also read the rotary encoder pins in the ISR. This was necessary because reading in a millisecond scale thread was too slow to capture every turn of the encoder. The rotary encoders are connected to the port expander, so we read the bits over SPI. However, rotary encoders are noisy, so we needed to use a fast digital debounce filter. Each time the ISR is called, the reading of the A terminal of the rotary encoder is shifted left into an integer value.

We continuously check if this integer is equal to -4,096, meaning there was a 1, followed by 12 zeros. This state ensures that we've received a single logic-high reading, followed by twelve logic-low readings [13]. With that, we know the signal is stable and no longer bouncing. Once this check passes, we can tell in which direction the encoder was turned, based on the B terminal. As explained in the Hardware Design section, if the B terminal is high, the turn was clockwise. Otherwise, the turn was counter-clockwise. Finally, parameter values for the sound synthesis algorithm are updated based on these rotary encoder readings.

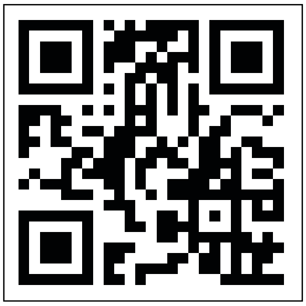


FIGURE 7
This QR code links to a video of our fully functional system. The video is also posted on Circuit Cellar's article materials webpage.

THE THREADS

Input Thread: After sufficiently optimizing the ISR, we next focused on the Input Thread. This thread reads the push buttons so the correct notes are played, and also performs slower calculations like float multiplication and division. These tasks are grouped together, because they don't need to be run as quickly as the ISR, but fast enough so the latency isn't noticeable. Pianos inherently have a delay between when a key is pressed and when the hammer hits a string, so a latency of approximately 20 ms is acceptable to most musicians [14].

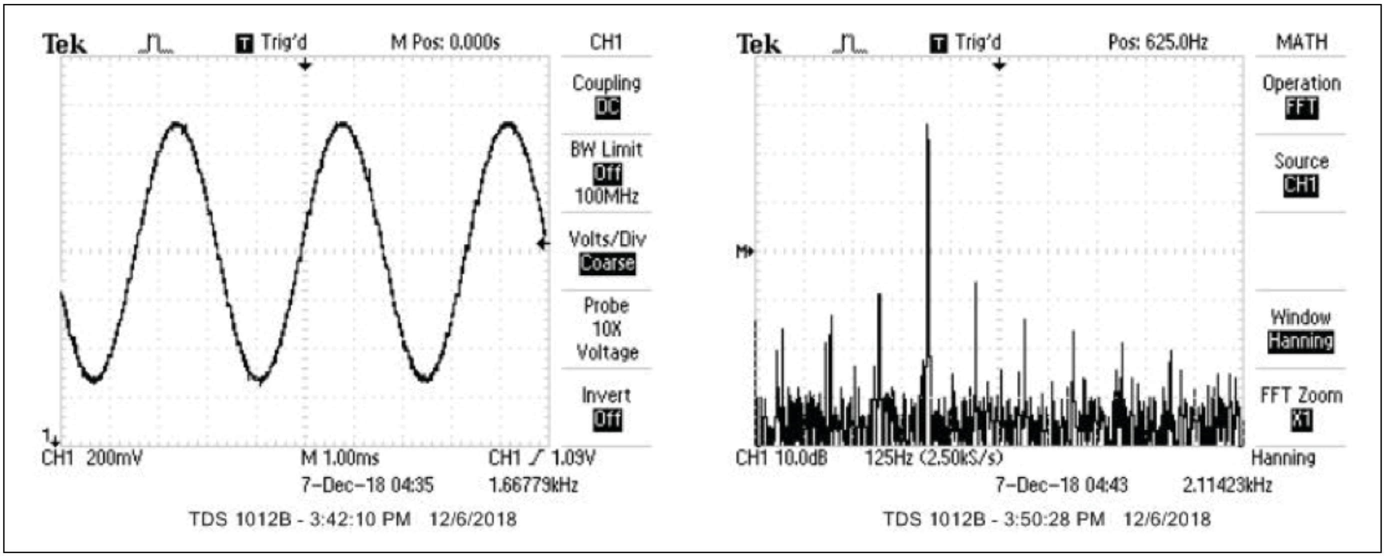


FIGURE 8

Oscilloscope screen captures of a sine wave produced by our system (left), and an FFT for the system's A-440 sine wave (right). Note that the fundamental frequency was 30 dB greater than the harmonics.

Because both the push buttons and DAC data are output through SPI, we need to create a critical section whenever we access the port expander. The critical section is enabled by turning the interrupt off for Timer2. We then read all the push buttons on the keypad matrix, and the push buttons connected to the rotary encoders. This information is then used to play specific notes or to change what parameters the rotary encoders affect.

Update Screen Thread: The last thread that had to be optimized was the Update Screen thread. This was especially important, because if the calculations took too long, the screen would visibly jitter and not update correctly. The screen is updated at 10 frames/s, or once every 100 ms. Drawing to the screen takes many cycles, so we optimize by drawing only what is necessary. To do this, we keep track of which parameters have changed since the last time the frame was drawn. If nothing has changed, we don't update the screen at all, which saves a lot of execution time. Otherwise, we draw any necessary changes.

With these many optimizations, we were able to create a powerful and flexible system. Audio was generated in real-time with a 20 kHz sample rate. A user can play notes and make changes to the audio synthesis algorithm on the fly, with no discernable delay. The LCD screen also updates immediately to reflect any changes the user makes to the various parameters.

RESULTS & CONCLUSIONS

A QR code that can be used to view a video of our fully functional system is given in **Figure 7**. After we finished constructing our system, we needed to evaluate its performance. The first method used was to get visual confirmation of the waveforms produced, using an oscilloscope. **Figure 8** shows the oscilloscope screen capture we obtained for a sine wave produced by our system, and a screen capture of a Fast Fourier Transform (FFT) for an A-440 note. Clearly, the system produced a clean sine wave with only small amounts of noise. The FFT shows that the magnitude of the fundamental frequency was about 30 dB greater than the magnitude of the harmonic frequencies. The oscilloscope screen captures also confirmed that the system produced square waves and sawtooth waves with very little noise.

We should mention here that another consideration in our design was whether

ABOUT THE AUTHORS

T.J. Hurd (jjh353@cornell.edu) is currently an undergraduate student at Cornell University studying Electrical and Computer Engineering. He's passionate about music and the technologies behind it.

Ben Roberge (bjr73@cornell.edu) is currently an undergraduate student in the School of Electrical and Computer Engineering at Cornell University. His technical interests are in MCUs and embedded systems.

or not to include a low-pass filter between the SPI DAC and the audio socket. It would have been relatively easy to add a simple RC, low-pass filter into our design, if we felt there were too many high-frequency components distorting the output. However, by considering the waveforms and FFTs observed on the oscilloscope, we determined that the output did not require a low-pass filter.

The next evaluation method was to quantify the frequency accuracy of our system. We used the FFT function on an oscilloscope to confirm the frequency content of our outputs. These oscilloscope readings showed us the fundamental frequency of each tone produced. We measured the fundamental frequencies of all 13 notes in an octave, and compared them to their target frequencies. By doing this, we found that the system's average percent error between the target frequencies and the actual frequencies produced was only 0.423%. We were very pleased with this level of accuracy.

We also wanted an estimate of our maximum CPU load, to give us an idea of how efficiently our code ran. To do this, we calculated the number of cycles the system took to execute the Sound Generation ISR, and saved this value in a variable. As stated in the Software Design section, this ISR was configured to run at 20 kHz. This meant a timer interrupt triggered its execution every 2,000 cycles. By printing the variable on the LCD, we found that a maximum of 1,200 cycles were needed to execute the ISR. This maximum execution time occurred when two notes were being played simultaneously. Our maximum CPU load was roughly 1,200/2,000 or 60%.

SCREEN FRAME RATE

The final method of evaluation was measuring the frame rate of the screen under various system conditions. Our goal was to maintain a frame rate of at least 10 frames/s. The maximum frame rates for an assortment of system conditions are summarized in **Table 1**. Our desired frame rate could be met for every condition, except when changing between system screens. However, these screen changes occurred so quickly that any frame rate lag was almost undetectable by the user.

By using these methods to evaluate the system's performance, we were able to confirm the desired functionality quantitatively. Overall, we achieved many of our initial goals. We created a system capable of producing a wide variety of high-quality, user-generated sounds. We believe the user interface was easy-to-use, and it demonstrated immediate responsiveness to user input. The system also was easily portable and simple to start up.

With all that said, extensions to the project are also possible. For example, the push-button matrix in our design was subject to something known as the "ghosting problem." This prevented the user from reliably pressing more than two buttons simultaneously. One solution to this problem involves adding a diode at each intersection in the push-button matrix. This would enable the user to play chords by pressing as many buttons as desired, without getting unintended responses from the system. Other potential extensions include enabling the user to save sound presets, which would allow easy replication of sounds the user discovers, and adding a looping feature, which would allow the user to create a complete song.

Condition	Time to Draw Screen (ms)	Maximum Frame Rate (FPS)
No Notes Played	7	142.9
1 Note Played	9	111.1
2 Notes Played	11	90.9
No Notes w/ Screen Change	125	8
No Notes w/ Param Change	60	16.7
1 Note w/ Screen Change	155	6.5
1 Note w/ Param Change	77	13
2 Notes w Screen Change	170	5.9
2 Notes w/ Param Change	82	12.2

TABLE 1
Frame rate data for a variety of system conditions. "Param" change means that a single parameter has been cycled. "Screen" change means the entire screen has been cycled to the next one. These drops in frame rates were brief and nearly undetectable.

For detailed article references and additional resources go to:
www.circuitcellar.com/article-materials
References [1] through [14] as marked in the article can be found there

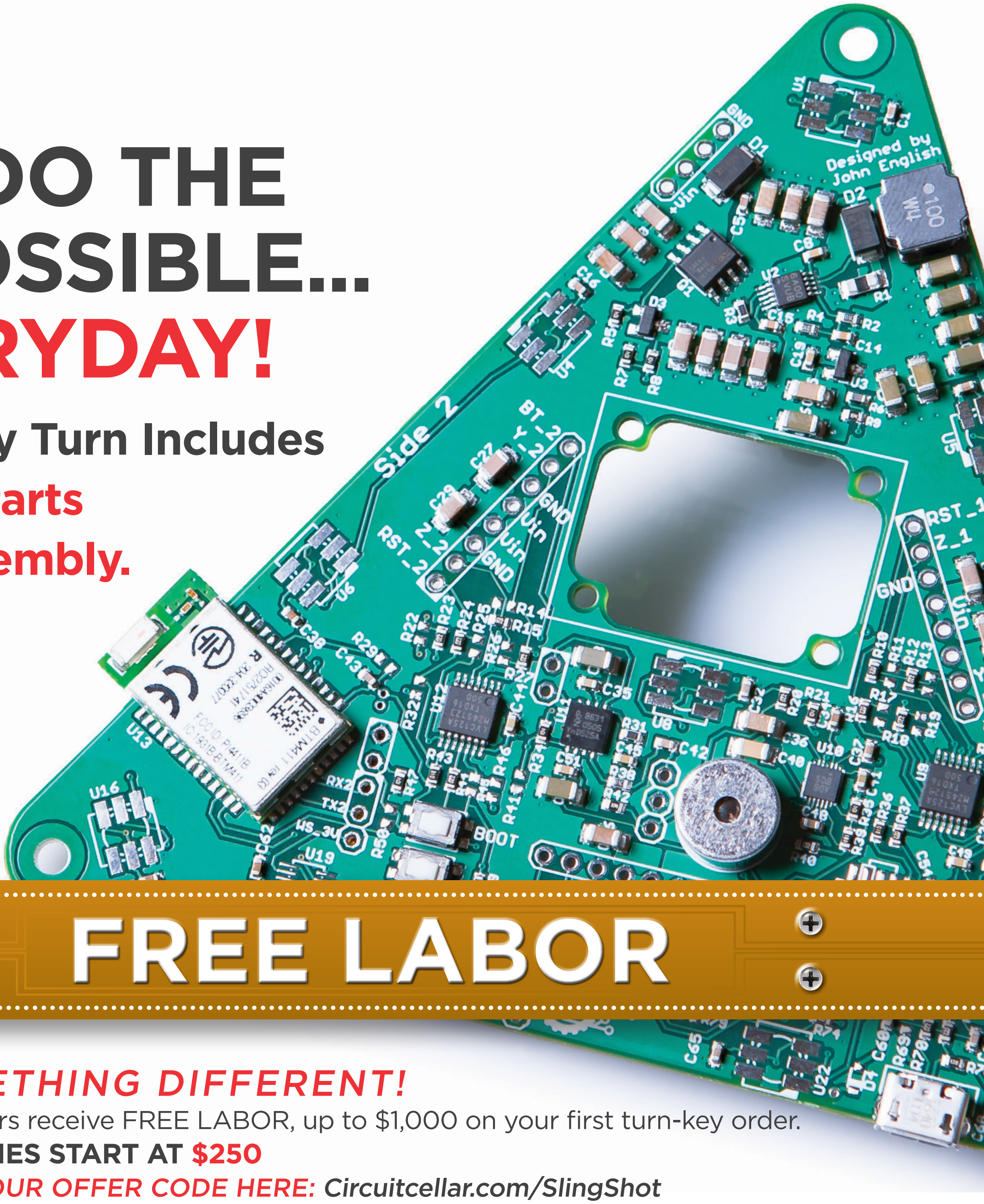
RESOURCES

Adafruit | www.adafruit.com

Autodesk | www.autodesk.com

Microchip Technology | www.microchip.com

Monoprice | www.monoprice.com



IC Solutions Rev Up for Next Gen Auto Designs

MCUs, Analog ICs and More

Automotive electronics are evolving to facilitate the shift from driver assisted vehicle controls to full autonomous driving—but that's only part of all that's happening. To meet a variety of design challenges, MCU and analog IC vendors are developing innovative solutions for automotive systems.

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

There's perhaps no more vivid example of the impact of embedded electronics than the continuing advances in automotive technologies. Today, those advances are set within an era of great innovation in the industry as car makers evolve their driver assistance technologies in parallel with their autonomous vehicle solutions, while at the same time improving the performance of full electric and hybrid electric vehicles. On top of all that, car infotainment systems are moving to an entirely new level.

To meet these system design changings automotive IC makers, continue to roll out chip, development system and software solutions aimed at next-gen automotive designs. Over the past 12 months, chip vendors, primarily microcontroller (MCU) and analog IC vendors, have announced a variety of powerful System-on-Chip (SoC), MCU and analog ICs solving all kinds of problems. Leveraging their long histories of serving the automotive market, the leading MCU vendors have taken the lead facilitating driverless car systems with not just chips, but also sophisticated development platform solutions for advanced driving assistance systems (ADAS), battery management and other automotive subsystems.

FLASH FOR VIRTUALIZATION

Some of the advances in automotive electronics over the past 12 months have revolved around embedded flash solutions aimed directly at automotive system designs. In an example along those lines, in February, Renesas Electronics announced what it claims as the world's first MCU with embedded flash that integrates a hardware-based virtualization-assisted function while maintaining the fast, real-time performance of the RH850 products.

This hardware-based virtualization assist technology can support up to ASIL D level of functional safety, providing greater levels of system integration. The RH850/U2A MCU (**Figure 1**) is the first member of Renesas' cross-domain MCUs, a new generation of automotive-control devices, designed to address the growing need to integrate multiple applications into a single chip to realize a unified electronic control units (ECUs) for the evolving electrical-electronic architecture (E/E architecture).

Based on 28 nm process technology, the 32-bit RH850/U2A MCU builds on key functions from Renesas' RH850/Px Series for chassis control and RH850/Fx Series for body control to deliver improved performance and implement a virtualization-assisted function to support operation in chassis/safety, body, domain control and low-end/mid-range gateway applications. The RH850/U2A MCU is equipped with up to four 400 MHz CPU cores in a dual core lock-step structure. Each CPU core integrates a hardware-based virtualization-assisted function, while maintaining the same fast real-time performance provided by the RH850. To support ASIL D, the MCU includes self-diagnostic SR-BIST (Standby-Resume BIST) functions with minimized current fluctuation rate.

The hardware-based virtualization-assisted function allows multiple software systems with varying ISO 26262 functional safety levels to operate independently without interference during high performance. It also reduces the virtualization overhead to maintain real-time execution. This enables users to integrate multiple ECU functions into a single ECU while maintaining safety, security and real-time operation requirements.

The RH850/U2A MCU is equipped with up to 16 MB of built-in flash ROM and 3.6 MB of SRAM, offering users the flexibility for future function expansion. The MCU includes security functions that support Evita Light up through Evita Full for enhanced protection against cyber-attacks, enabling the device to support safe and rapid Full No-Wait Over-the-Air (OTA) software updates as security requirements evolve.

FAIL-SAFE STORAGE

In other automotive flash technology news, in April Cypress Semiconductor announced that automotive supplier DENSO selected Cypress' Semper fail-safe storage for its next-generation digital automotive cockpit applications with advanced graphics. Based on an embedded Arm Cortex-M0 processing core, the Semper family is purpose-built for automotive environments.

The Cypress Semper family offers high density serial NOR flash memory up to 4 Gb and leverages the company's proprietary MirrorBit process technology. The family also features EnduraFlex architecture, which achieves greater reliability and endurance. Semper fail-safe storage devices were the first in the industry to achieve the ISO 26262 automotive functional safety standard and are ASIL-B compliant, says Cypress. According to Cypress, the Semper fail-safe storage products exceed automotive quality and functional safety requirements with ASIL-B compliance and are ready for use in ASIL-D systems. Cypress' 512 Mb, 1 Gb and 2 Gb Semper devices are currently sampling.

DOMAIN CONTROLLERS

For its part, STMicroelectronics (ST) also rolled out a new automotive-focused MCU offering back in February. Called the Stellar automotive MCU family, these devices support next-generation car architectures, which rely on broad "domain controllers" for areas such as the drivetrain, the chassis, and Advanced Driver Assistance Systems (ADAS). These domain controllers enable the transition toward software- and data-oriented architectures by providing data fusion from connected sensors while reducing harness complexity and electronic-component weight.

Built on a 28 nm FD-SOI process, major

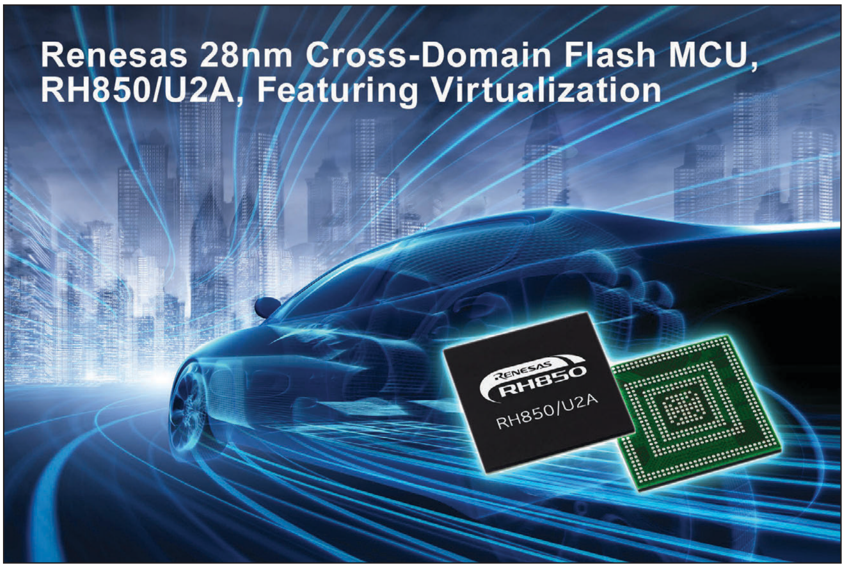
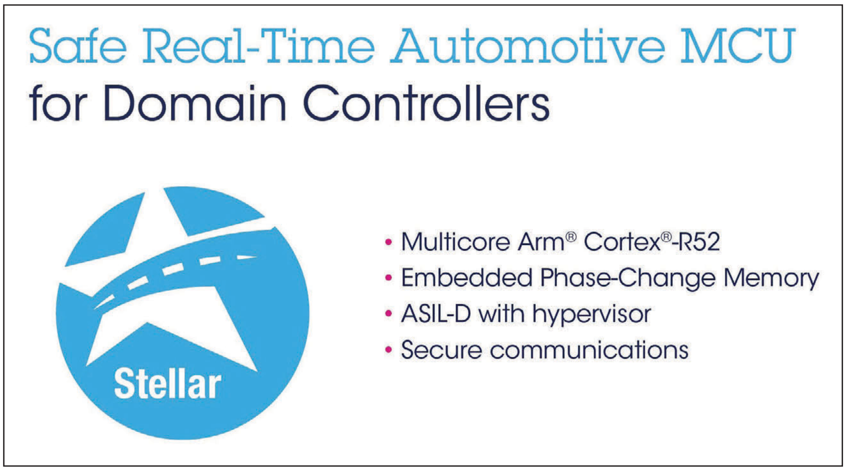

Renesas 28nm Cross-Domain Flash MCU, RH850/U2A, Featuring Virtualization
RENESAS
RH850
RH850/U2A

FIGURE 1
The RH850/U2A MCU is equipped with up to four 400 MHz CPU cores in a dual core lock-step structure. Each CPU core integrates a hardware-based virtualization-assisted function.

applications for Stellar MCUs include smart control for hybrid powertrain, the broad electrification of car systems with on-board chargers, battery-management systems and DC-DC controllers, as well as smart gateways, ADAS and enhanced Vehicle Stability Controls. The MCUs feature six Arm Cortex-R52 cores clocked at 400 MHz, 16 MB of Phase-Change Memory (PCM) and 8 MB of RAM, all in a BGA516 package (**Figure 2**). Stellar-based control units are currently undergoing road tests with lead customers.



Safe Real-Time Automotive MCU for Domain Controllers

- Multicore Arm® Cortex®-R52
- Embedded Phase-Change Memory
- ASIL-D with hypervisor
- Secure communications

FIGURE 2
The Stellar MCUs feature six Arm Cortex-R52 cores clocked at 400 MHz, 16 MB of Phase-Change Memory (PCM) and 8 MB of RAM, all in a BGA516 package.

FIGURE 3
The maXTouch TD family of touch controllers features a differential mutual signal acquisition method that significantly increases the Signal-to-Noise Ratio (SNR).

Stellar satisfies the automotive industry's demanding ISO26262 Automotive Safety Integrity Level (ASIL) Leve D safety qualification by extending the Cortex-R52 cores with lockstep capabilities. To further enhance functional safety and reliability, Stellar features a hypervisor for software separation and memory protection. Bolstering the multicore Cortex-R52 performance, Stellar also packs three Arm Cortex-M4 cores with a floating-point unit and DSP extensions to provide application-specific acceleration.

The MCUs leverage ST's advanced embedded PCM, which is compliant with AEC-Q100 Grade 0. Safe and rugged, the 16 MB PCM assures performance, data retention up to 165°C and supports Software OTA to manage multiple firmware images. The convenient eMMC and HyperBus interfaces offer additional external storage.

The devices also featured a Hardware Security Module (HSM) with EVIT FULL support and, by operating at more than 200 MHz, Stellar is designed to maximize data throughput. The combination of HSMs with multi-bus routing across Stellar's wide set of automotive interfaces—including Ethernet, CAN-FD and LIN—meet the requirements of automotive OEMs for security and connectivity to their time-sensitive car networks.

TOUCHSCREEN CONTROLLERS

Developers of automotive touchscreens face tough electromagnetic interference (EMI) and electromagnetic compatibility (EMC) challenges. Addressing those needs, in June Microchip Technology announced three new maXTouch touchscreen controllers along with optimization services. The new TD family of touch controllers features a differential mutual signal acquisition method that significantly increases the Signal-to-Noise Ratio (SNR) (**Figure 3**). This allows the use of very thick glass or plastic cover lenses and multi-finger thick-gloved touch support up to the equivalence of 4.5 mm polymethyl methacrylate (PMMA).

The MXT1067TD, MXT1189TD and MXT1665TD devices add several variants that are cost optimized for 9" to 13" automotive touchscreens to Microchip's portfolio and join the recently-introduced MXT449TD, MXT641TD, MXT2113TD and MXT2912TD devices supporting up to 20" touchscreens. Each device addresses aspects of the increasing demand for functional safety features and is designed in accordance with the Automotive SPICE Level 3 capability and ISO 26262 ASIL B requirements.

All devices in the TD family feature a unique waveform shaping capability to optimize the performance of the touch controller's radiated emissions through an EMI optimization tool. Working with product experts in Microchip's worldwide application design centers, this tool allows developers to enter user-defined RF limits and tune the shape of the transmitted burst waveform used for the touch sensing acquisition.

Waveform shaping is achieved through firmware parameters derived from the tool and helps designers to position the fundamental burst frequency to work together with other in-vehicle applications, such as the remote keyless entry system. The resulting parameters are then simply added to the maXTouch configuration file, which customizes the touch controller performance to the individual customer design. This process can save the designer many hours, or even weeks, of expensive EMC test chamber time by eliminating experimentation with different configuration settings to achieve the desired EMI/EMC performance. The maXTouch EMI optimization service will be made available as part of the system support provided by one of Microchip's worldwide application design centers.

HV/EV REFERENCE DESIGNS

Another design challenge for automotive system developers is improving the performance of hybrid electric vehicles and electric vehicles (HEV/EVs) In May Texas Instruments (TI) introduced a set of fully tested reference designs for battery management and traction inverter systems, along with new analog circuits with advanced monitoring and protection features to help reduce carbon dioxide emissions and enable HEV/EVs to drive farther and longer.

Scalable across six to 96 series cell supervision circuits, TI's new battery management system (BMS) reference design (**Figure 4**) features the advanced BQ79606A-Q1 precision battery monitor and balancer. Engineers can get their automotive designs to market quickly using the reference design, which implements the battery monitor in a daisy chain configuration to create a highly accurate and reliable system design for three to 378 series, lithium-ion battery packs from 12 V up to 1.5 kV.

The highly integrated BQ79606A-Q1 accurately monitors temperature and voltage levels and helps maximize battery life and time on the road. Additionally, the BQ79606A-Q1 battery monitor features safe-state communication that helps system designers meet requirements up to ASIL D, which is the highest functional safety goal defined by the ISO 26262 road vehicles standard.

POWERTRAIN PROTECTION

To protect powertrain systems—such as a 48 V starter generator— from overheating, TI introduced the TMP235-Q1 precision analog output temperature sensor. This low-power, low-quiescent-current (9 μA) device provides high accuracy (±0.5°C typical and ±2.5°C maximum accuracy across the full operating temperature from -40°C to 150°C) to help traction inverter systems react to temperature surges and apply appropriate thermal management techniques.

The TMP235-Q1 temperature sensing device joins the recently released UCC21710-Q1 and UCC21732-Q1 gate drivers in helping designers create smaller, more efficient traction-inverter designs. These devices are the first isolated gate drivers to integrate sensing features for insulated-gate bipolar transistors (IGBTs) and silicon carbide (SiC) field-effect transistors, enabling greater system reliability in applications operating up to 1.5 kV$_{RMS}$ and with superior isolation surge protection exceeding 12.8 kV with a specified isolation voltage of 5.7 kV. The devices also provide fast detection times to protect against overcurrent events while ensuring safe system shutdown.

To power the new gate drivers directly from a car's 12 V battery, TI has released a new reference design demonstrating three types

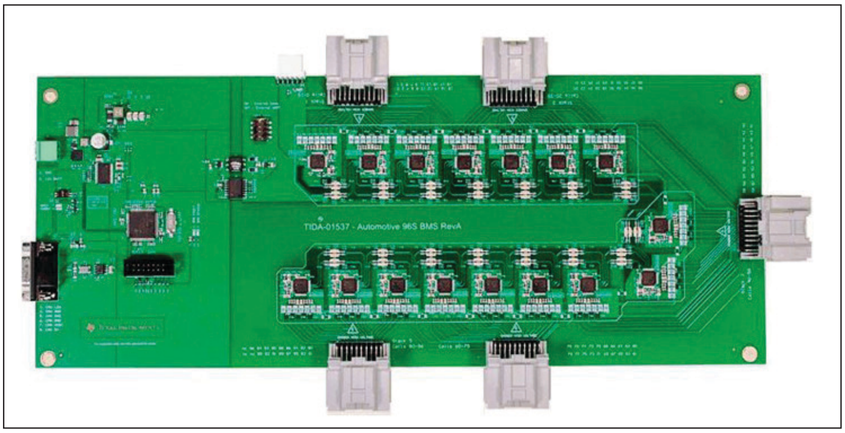


FIGURE 4
Scalable across six to 96 series cell supervision circuits, this battery management system (BMS) reference design features the advanced BQ79606A-Q1 precision battery monitor and balancer.

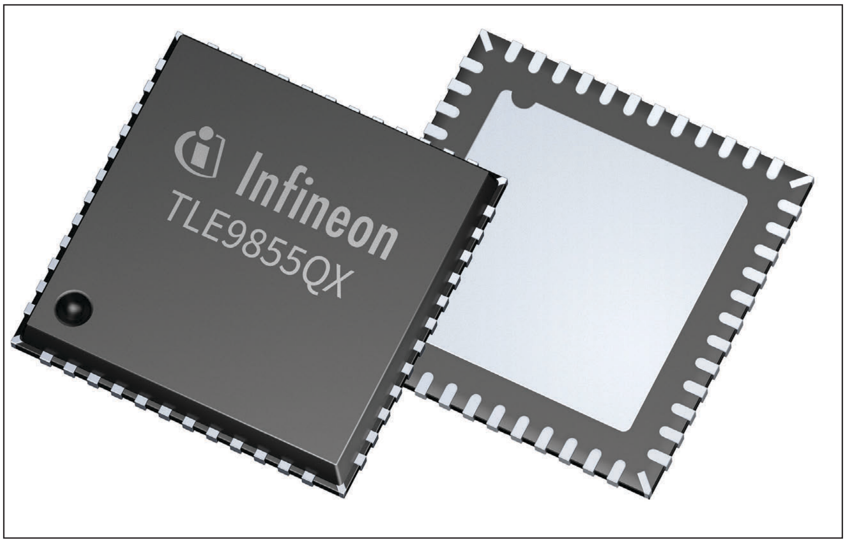


FIGURE 5
The TLE985x series provides highly integrated, AEC Q-100 qualified H-bridge driver motor control solutions for 2-phase DC and single-phase brushless DC motors. It will support automotive system designers' ability to replace relays in low-end motor control applications such as sunroof and window lift.

of IGBT/SiC bias-supply options for traction inverter power stages. The design consists of reverse-polarity protection, electric-transient clamping and over- and under-voltage protection circuits. The compact design includes the new LM5180-Q1, which is a 65 V primary-side regulation flyback converter with a 100 V, 1.5 A integrated power MOSFET.

LOW-END MOTOR CONTROL

Aside from the engine itself, automobiles are filled with a variety of low-end motors performing functions like controlling windows. Feeding those needs, Infineon Technologies launched a family of embedded power ICs. The TLE985x series provides highly integrated, AEC Q-100 qualified H-bridge driver motor control solutions for 2-phase DC and single-phase brushless DC motors (**Figure 5**). It will support automotive system designers' ability to replace relays in low-end motor control applications such as sunroof and window lift.

By switching from relays to MOSFETs, the higher level of integration reduces system costs. Additional advantages are that the PWM control and the integrated current sense amplifier, which is calibrated, allow the motor currents to be adapted and thus the mechanics and motor to be optimized toward the application requirements. The circuit board and the motor become smaller. At the same time, the noise behavior improves.

TLE985x devices integrate an Arm Cortex-M0 processor and peripherals for motor control, power supply and communication. Two integrated measurement units (analog-to-digital converters) for monitoring temperature, battery voltage and four monitoring inputs help to save pins. These inputs can be operated directly with battery voltage, which reduces costs on additional components such as external voltage dividers or shutdown transistors. Furthermore, the chips are equipped with two full duplex serial interfaces (UART) with LIN support.

A new feature in the TLE985x family is its adaptive MOSFET driver. The control algorithm is able to compensate MOSFET parameter spread in the system by automatically adjusting the gate current according to required switching times. This allows an optimization of the system concerning EME (electro-magnetic emissions, slow slew rates) as well as power dissipation (short dead times) simultaneously. The product family includes five devices with different flash sizes (48 KB to 96 KB) and temperature ranges (T_j up to 175°C). In addition, different numbers of half-bridge drivers for uni- or bidirectional DC motor applications are offered. The devices come in a leadless VQFN package with a footprint of 7 mm x 7 mm.

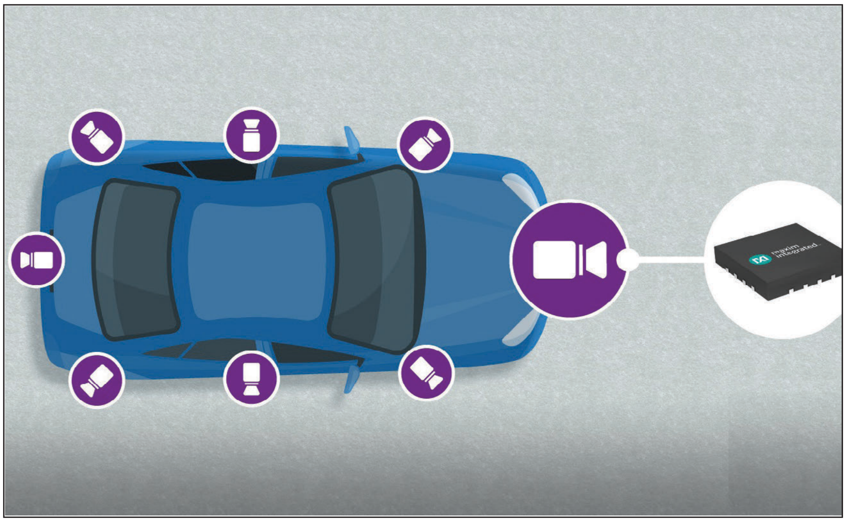


FIGURE 6
The MAX20049 PMIC is designed to meet vehicle camera needs by integrating four power supplies into a tiny footprint. The device offers many options to support various output voltages, while also providing fault mitigation by flagging faults and shifts in output voltages.

PMIC FOR CAR CAMERAS

For its latest automotive IC offering, Maxim Integrated in March introduced a compact MAX20049 power management IC (PMIC) designed to meet vehicle camera needs by integrating four power supplies into a tiny footprint (**Figure 6**). The device offers many options to support various output voltages, while also providing fault mitigation by flagging faults and shifts in output voltages.

Automotive camera modules tend to be size-constrained, so designers are constantly in search of a power management solution that can pack the necessary power and functionality into a small form factor. The 4-channel MAX20049 power management IC is almost 30% more compact than competitive solutions and offers the highest efficiency among other quad-power power management ICs in its class, says Maxim.

The chip offers many options to support modules that need various output voltages for different mixes of sensors and serializers, enabling designers to make changes in layout as needed or to fine-tune the IC to meet specific application requirements. The MAX20049 provides fault mitigation, a feature required by designers to help flag faults and shifts in output voltages to ensure that the cameras are working as needed.

DRIVER MONITORING SOLUTION

Last month, NXP Semiconductors and Momenta, a provider of software solutions for autonomous driving, announced their joint effort on automotive-grade Driver Monitoring Solutions (DMS). These solutions form the basis of the systems that monitor driver attentiveness, and play important roles in increasing safety on the road and helping carmakers address upcoming NCAP requirements. The collaboration between NXP and Momenta aims to enable car makers to deploy DMS applications into mass automotive production.

According to NXP, Driver Monitoring Systems are one of the interrelated parts of ADAS and are essential for Level 3 and higher driving systems. The systems use deep learning algorithms to visually monitor and detect a driver's lack of attention to the road and can offer pre-collision warnings. Euro NCAP, a European car safety performance assessment program, has made DMS a primary safety function slated for NCAP incorporation by 2020.

The first solution from the NXP and Momenta collaboration will combine the high performance, power-efficient architecture of NXP's Open Vision Platform (S32V2) (**Figure 7**) with Momenta's deep learning software and expertise. The solution aims to optimize, compress and accelerate deep neural networks so that they can run efficiently on an automotive-grade DMS embedded platform.

The integrated automotive-grade hardware accelerators in the NXP S32V2 are ideal for deep neural network processing

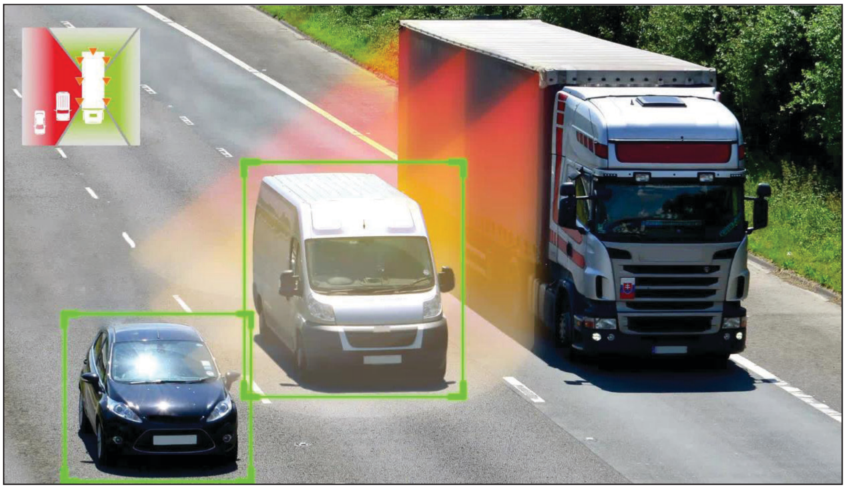


FIGURE 7
The S32V234 Vision Processor supports computation intensive ADAS, NCAP front camera, object detection and recognition, surround view, automotive and industrial image processing, also machine learning and sensor fusion applications.

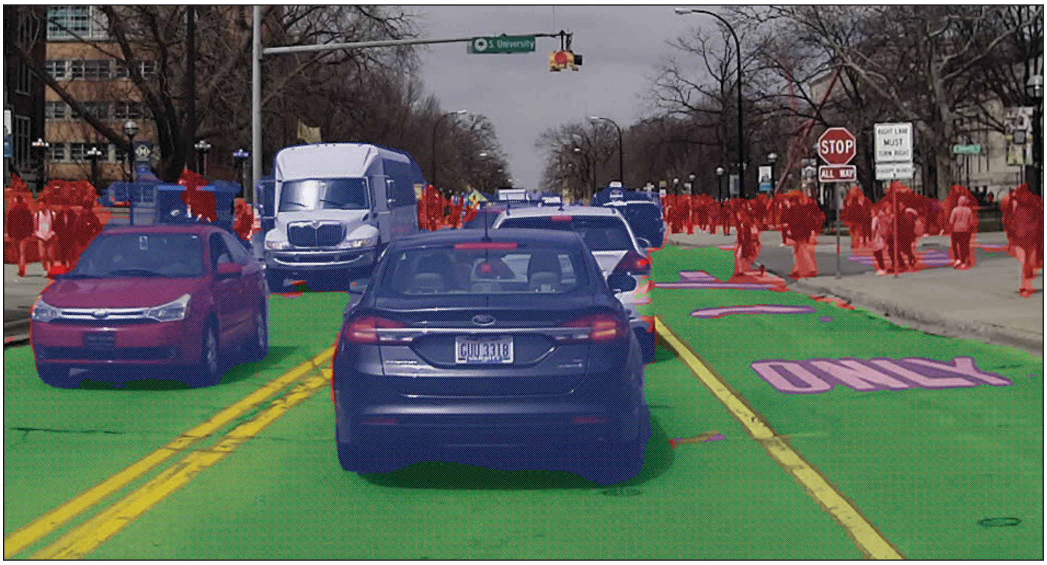


FIGURE 8
The RFD reference software uses CNN-IP, CV-E, IMR, and a versatile pipeline engine (IMP) to identify drivable free space, lanes (crossable and uncrossable), road boundaries and distances to lanes and nearest objects to support NCAP 2020.

because they can reduce CPU usage and save computing resources, says NXP. This can offer more performance for other vision processing tasks within the vehicle and reduce costs for system developers. By combining the hardware architecture of the NXP S32V2 platform and the NXP AI enablement, Momenta's deep-learning software algorithms can be deployed quickly and run efficiently on a low-power-consumption, automotive-grade chip.

DETECTION TECHNOLOGIES

Also taking aim at ADAS system development needs, in June Renesas Electronics released its Perception Quick Start Software based on its R-Car V3H SoC. The solution delivers reference software for camera obstacle detection (COD), Lidar obstacle detection (LOD) and road feature detection (RFD). According to Renesas, those are three key recognition areas for sensor-based Level 2+ autonomous vehicle systems.

The R-Car V3H SoCs provide a mix of high computer vision performance and artificial intelligence processing at low power levels, providing an optimized embedded solution for automotive front cameras in Level 2+ autonomous vehicles. To achieve state-of-the-art recognition technology, Renesas designed the SoCs with dedicated hardware accelerators for key algorithms including convolutional neural networks, dense optical flow, stereo disparity and object classification.

The new perception software provides an end-to-end pipeline reference for developers working with these complex yet cost-effective and power-efficient accelerators, which allows customers to kickstart their application designs whether they are experts at using the accelerators or have limited experience. The reference software covers input from sensor or recorded data, all stages of processing and display output on a screen.

The COD reference software uses convolutional neural network (CNN) IP, a computer vision engine (CV-E) and image rendering (IMR) technology to detect 2D objects such as cars, trucks, buses and pedestrians. The COD achieves approximately 30 frames per second (FPS). The LOD software uses CNN-IP and CV-E to detect 3D objects, including cars and trucks. The LOD achieves approximately 15 FPS with 3D bounding boxes at 50 meters. Finally, the RFD reference software (**Figure 8**) uses CNN-IP, CV-E, IMR, and a versatile pipeline engine (IMP) to identify drivable free space, lanes (crossable and uncrossable), road boundaries and distances to lanes and nearest objects to support NCAP 2020. The RFD achieves approximately 30 FPS.

RESOURCES

Cypress Semiconductor | www.cypress.com

Infineon Technologies | www.infineon.com

Maxim Integrated | www.maximintegrated.com

Microchip | www.microchip.com

Momenta | www.momenta.ai

NXP Semiconductor | www.nxp.com

Renesas Electronics America | www.renesas.com

STMicroelectronics | www.st.com

Texas Instruments | www.ti.com

Display Solutions Enhance Embedded Designs

System-Level Functionality

Integrating a display into your embedded system can be a tricky task. Fortunately, a variety of modular display systems are available today in all shapes and sizes. These solutions provide high-resolution screens with touchscreen capabilities and rich interface support.

By *Jeff Child,*
Editor-in-Chief

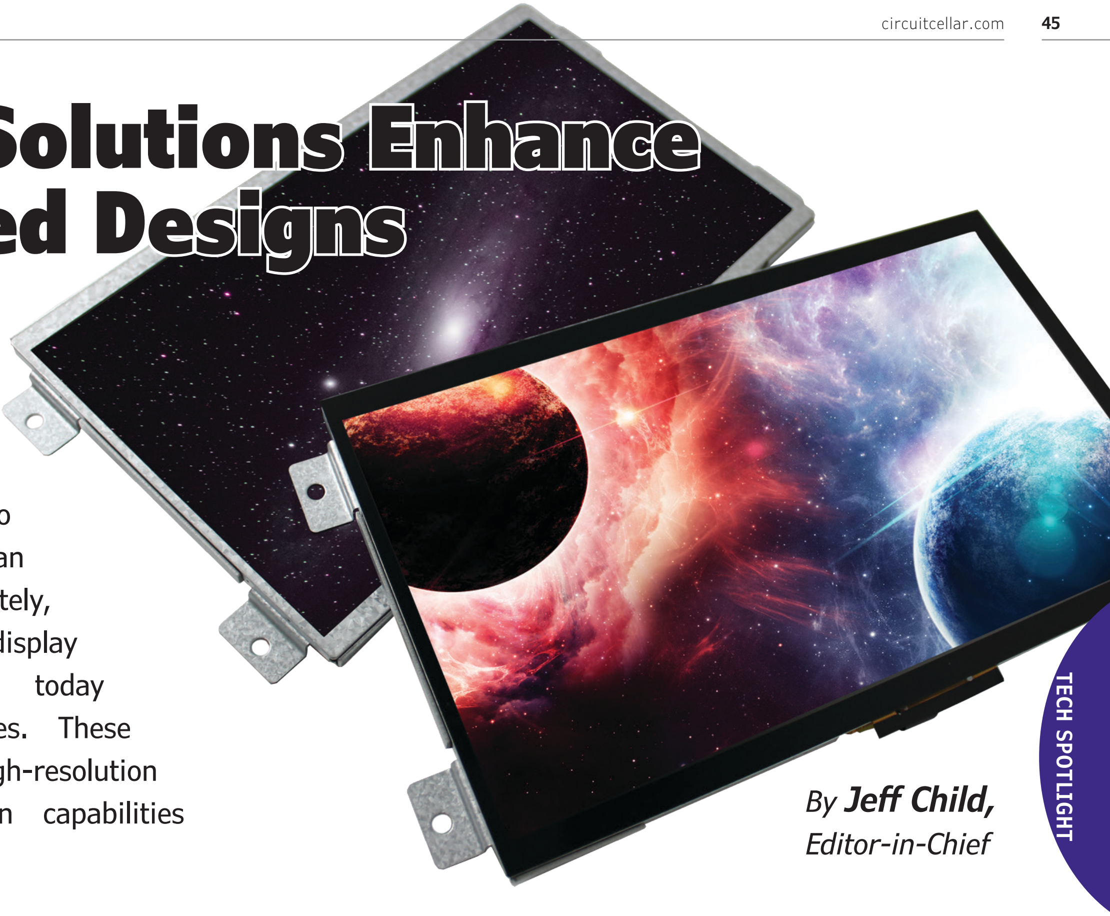


FIGURE 1
These display modules feature a 10.1" TFT paired with a custom PCB designed for HDMI compatibility. The module is available in capacitive touchscreen or non-touchscreen options—both come standard with a steel mounting bracket.

While it's true that not all embedded systems require any kind of human interface, there's a large segment that do. In fact, the sphere of applications that can accommodate high-resolution, touchscreen display technologies keep expanding thanks to the availability of easy-to-integrate display modules designed for embedded use.

Over the past 12 months, display module vendors have developed a variety of new solutions providing smaller sizes, enhanced performances, new levels of ruggedness and rich sets of interface support. Support for Raspberry Pi and improved e-paper displays are also part of the latest display trends. So that you don't have to face the complexity of interfacing a raw display with an embedded computer system, today's crop of display modules either make such integration easy, or provide complete pre-integrated display-computer subsystems.

HDMI FOR EMBEDDED

While HDMI display connectivity has become standard in consumer displays, it was slower to move into the embedded space. Now it's become entrenched in the embedded realm, with many display module vendors offering next-generation solutions. Along these lines, in September last year, Newhaven Display announced an expansion to its HDMI TFT product line to include two new 10.1" modules. These display modules feature the company's 10.1" TFT paired with a custom PCB designed for HDMI compatibility. The module is available in two options differentiated by a capacitive touchscreen version, and a version with no touch panel **(Figure 1).** Both options come standard with a durable steel mounting bracket featuring server-rack-height threaded mounting holes.

Both of the 10.1" HDMI TFT Modules are 1024×600-pixel resolution and sport MVA technology for 75-degree viewing from all angles. The boards on these products also have a Texas Instruments (TI) audio amplifier embedded to enable the use of the audio channel already supported by HDMI. The product provides simple, standard HDMI interfacing.

Similar to Newhaven Display's other HDMI TFT modules, the process of connecting to the display is designed to be easy. There's no need to work around any ribbon cables or extra controller boards. Embedded system developers can just connect a standard HDMI cable directly to the TFT Module. Setup for the capacitive touch panel version is easy as well. You connect the touch panel via USB cable and the USB-HID driver recognition will quickly initialize the touch panel for use right away.

The durable steel bracket included with both 10.1" HDMI TFT options provides a secure and stable way to mount the display within any application. The bracket is server rack-unit height making it an easy fit for standard server racks. These mounting brackets bring a vibration-resistant and reliable method of installation already packaged with the 10.1" HDMI TFT modules.

GLOVE TOUCH SUPPORT

The latest TFT module offering from Noritake is its GTWV050VHB00P product, an embedded touch TFT module designed to smoothly integrate into any project. The module consists of a 5" high brightness TFT panel, digital video interface (DVI) with HDMI connector, 5 VDC single power supply, FLETAS Metallized Projective Capacitive Touch (MPCT) screen and all necessary drive circuitry (**Figure 2**). The mounted FLETAS touch panel has high touch sensitivity that can operate accurately in demanding environments. Touch works with gloves and up to a 5 mm acrylic overlay (with 0.5 mm air-gap).

The GTWV050VHB00P can work with commonly used micro-USB and HDMI cables.

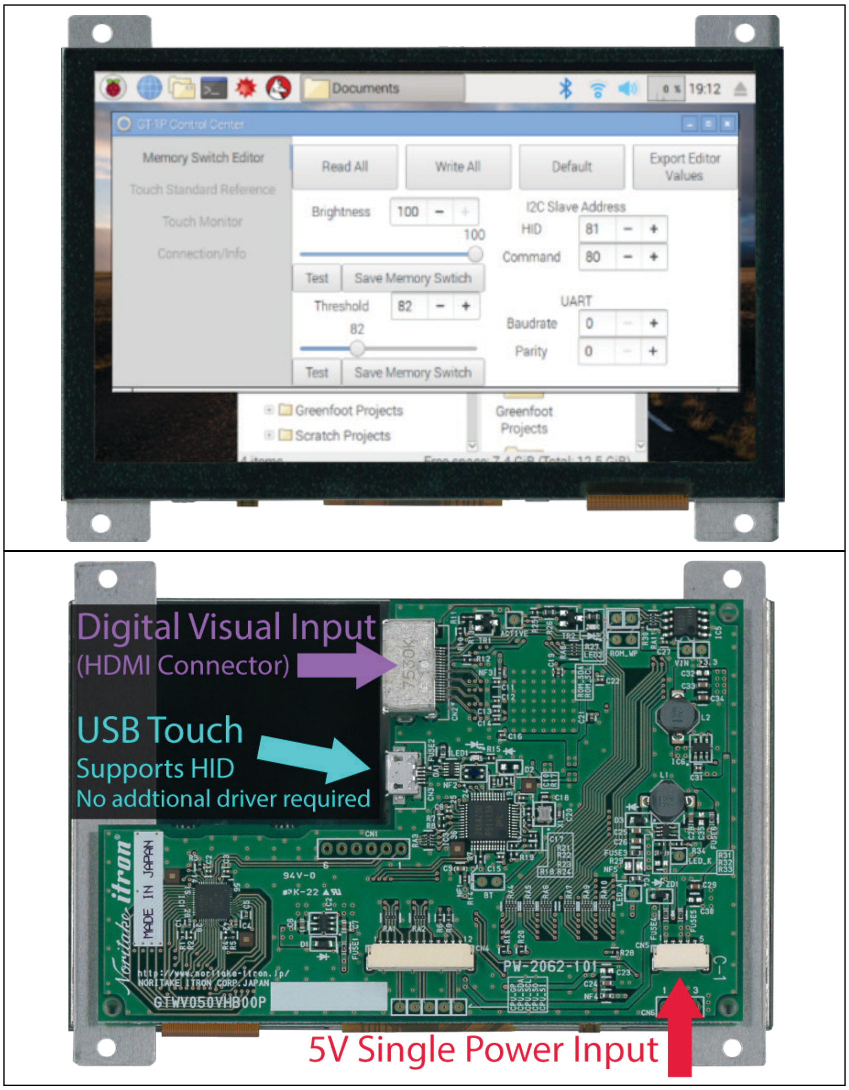


FIGURE 2
The GTWV050VHB00P is an embedded touch TFT module consisting of a 5" high brightness TFT panel, DVI with HDMI connector, 5 VDC single power supply, FLETAS MPCT screen and all the necessary drive circuitry. Top image shows the front, bottom image shows the back.

RESOURCES

4D Systems | www.4dsystems.com.au

Axiomtek | www.axiomtek.com

Newhaven Display | www.newhavendisplay.com

Noritake | www.noritake-elec.com

Pervasive Displays | www.pervasivedisplays.com

The power supply connector uses a JST SM05B-GHS-TB connector. All connectors are located on the backside of the display module as shown in **Figure 2**. Touch is supported via USB or I²C and is HID-compliant (no driver required). The module supports wide VGA (800×480 pixels) in 5" screen size. A high brightness 1,000 cd/ m² 5" TFT panel is used for this design. (Actual brightness: 810 cd/ m² typical.) The FLETAS MPCT screen has adaptive sensitivity that works with gloved hands, water droplets and thick overlays. Touch sensitivity is adjustable via simple commands over USB, I²C and UART.

RASPBERRY Pi DISPLAY

Raspberry Pi embedded computers have become extremely popular in recent years. And while the technology is often used in hobbyist projects, an ecosystem has evolved around Raspberry Pi providing a wealth of supporting products available for professional embedded system developers. Along just such lines, among the latest offerings from 4D Systems is its gen4-4DPI series of LCD display modules specifically designed to support the Raspberry Pi family of SBCs.

The display modules are available for the Raspberry Pi A+, B+, 3, Zero and Zero W with resistive or capacitive touch control options and provide a compact and elegant HMI display solution. The displays are also powered directly from the Raspberry Pi, eliminating the need for an external power supply. The gen4-4Dpi display modules are available in three screen sizes: 4.3", 5.0" and 7.0"—and connect to the Raspberry Pi through a 30-pin FPC cable and an adapter board that conforms to the Raspberry Pi expansion header pin-out and Pi's HAT device identification standard.

The m GEN4-4DPI-43CT-CLB version of the module features an integrated resistive touch panel or capacitive touch panel, enabling the gen4-4DPi to function with the Raspberry Pi without the need for a mouse. Communication between the gen4-4DPi and the Raspberry Pi is interfaced with a high speed 48 MHz SPI connection, which utilizes an on-board processor for direct command interpretation and SPI communication compression and features a customized DMA-enabled kernel. This combination allows this display to output high frame rate compared to other SPI display solutions, when displaying a typical image/ video, and can achieve higher depending if the image can be compressed.

The gen4-4DPi is designed to work with the Raspbian Operating System running on the Raspberry Pi, as that is the official Raspberry Pi operating system. It is also compatible with Pixel and Scratch. Mounting of the gen4-4DPi is achieved with the 4 mm × 4 mm mounting holes present on the resistive touch display modules, enabling standard screws to fasten the Gen4-4DPi as required, or via the adhesive supplied on the cover lens bezel, which is part of the capacitive touch version.

E-PAPER SCREEN ADVANCES

No longer a newcomer on the embedded scene, electronic paper displays (EPD) continue to evolve, with suitable products designed for embedded needs. Exemplifying that trend, in June Pervasive Displays (PDi) announced its new range of rugged tri-color displays, aimed at use in portable equipment designed for use in demanding conditions. Because such systems are used in in situations where impacts are common, a display's individual components must be exceptionally robust. Pervasive Displays' line of rugged black, white and red displays—available in 3.7", 4.2", 4.37" and 7.4" models—include a resin board attached to the glass substrate to protect the screen from breaking when bumped, dropped or knocked (**Figure 3**).

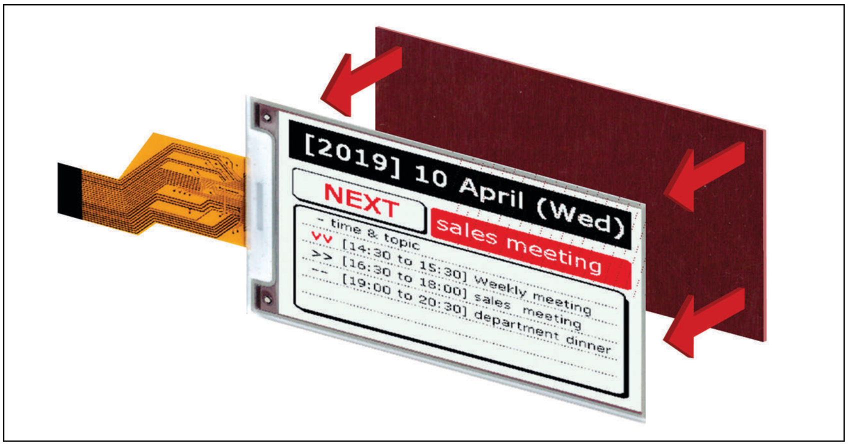

[2019] 10 April (Wed)
NEXT
sales meeting
- time & topic
vv [14:30 to 15:30] weekly meeting
>> [16:30 to 18:00] sales meeting
-- [19:00 to 20:30] department dinner

FIGURE 3
This line of rugged black, white and red e-paper displays—available in 3.7", 4.2", 4.37" and 7.4" models – include a resin board attached to the glass substrate to protect the screen from breaking when bumped, dropped or knocked.

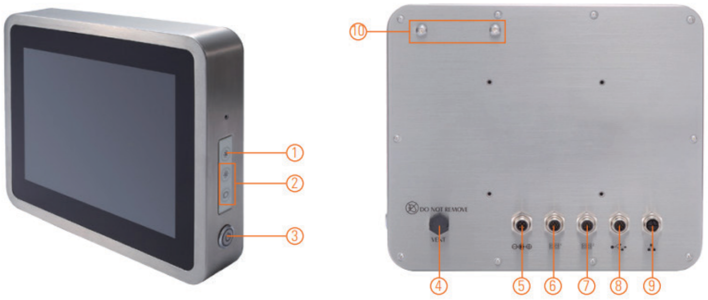
Example applications requiring this level of durability are wearable systems, logistics container labels, or tags on moving equipment. Where a conventional glass display could shatter in these situations, these new displays are designed to resist the impact by being up to 50% stronger, helping ensure the screen remains readable.

The display's ruggedness adds to the already-attractive characteristics of EPDs for use in challenging and/or remote conditions. The way EPDs display text and images, using physical ink particles that reflect ambient light, means they remain readable even in bright sunlight or other harsh lighting conditions. Their near-180° viewing angle further enhances their ease of readability. Moreover, their exceptionally low-energy demands mean they don't require grid power, and are instead able to run for months or even years using a small coin cell battery.

The rugged displays' built-in timing controller (iTC) also minimizes the need for peripheral circuitry, resulting in smaller overall devices or extra space for batteries and other components. The displays' high pixel density—ranging from 117 dpi to 130 dpi, depending on the screen size—state-of-the-art driving waveform and the ability to render both text and images in red, white and black, mean embedded system designers can display rich, sharp and detailed information to users.

FANLESS PANEL PC

So called Panel PCs are a category of display systems that tend to be larger and are meant to be mounted on a factory wall or on the side of an industrial machine. An example along those lines is Axiomtek's the GOT810-845, a 10.4" stainless steel fanless touch panel computer powered by the Intel Celeron processor N3060. The rugged touch panel PC has an XGA TFT LCD display with 350 nits of brightness, as well as two touchscreen options: 5-wire flat resistive touch and projected capacitive multi-touch (**Figure 4**).



1. Backlight on/off
2. Brightness adjust
3. Power switch (ATX)
4. Ventilation valve
5. DC power connector (M12 A-coded)
6. 1 x configure RS-422/485 (default RS-232, M12 A-coded)
7. 1 x RS-232 (M12 A-coded)
8. 2 x USB 2.0 (M12 A-coded)
9. 1 x Ethernet (M12 X-coded)
10. 2 x Antenna opening

FIGURE 4
The GOT810-845, is a 10.4" stainless steel fanless touch panel computer powered by the Intel Celeron processor N3060. The rugged touch panel PC has an XGA TFT LCD display with 350 nits of brightness, as well as two touchscreen options: 5-wire flat resistive touch and projected capacitive multi-touch.

The customizable GOT810-845 is designed for operation in industrial and outdoor environments with its IP66/IP69K-rated enclosure and IP66-rated M12-type connectors, making it well suited for food processing and heavy-duty applications that require ruggedness and water/dust resistance for harsh operating conditions.

The GOT810-845's five IP66-rated M12- type I/O connectors are located at the rear bezel and include one Gbit Ethernet port, one RS-232/422/485 port, one RS-232 port, two USB 2.0 ports and one DC power connector. For operational reliability, it can operate under wide temperature settings ranging from -10°C to +50°C and withstand vibration up to 2 G.

The expandable touch panel computer has two PCI Express Mini Card slots for wireless connectivity through 3G, 4G and LTE. The GOT810-845 comes with one DDR3L-1333 SO- DIMM for up to 8 GB of system memory, as well as one mSATA and one SATA for storage. It runs on Windows 7, Windows 8/8.1 and Windows 10 operating systems.

Product Focus: Tiny Embedded Boards

Petite Processing

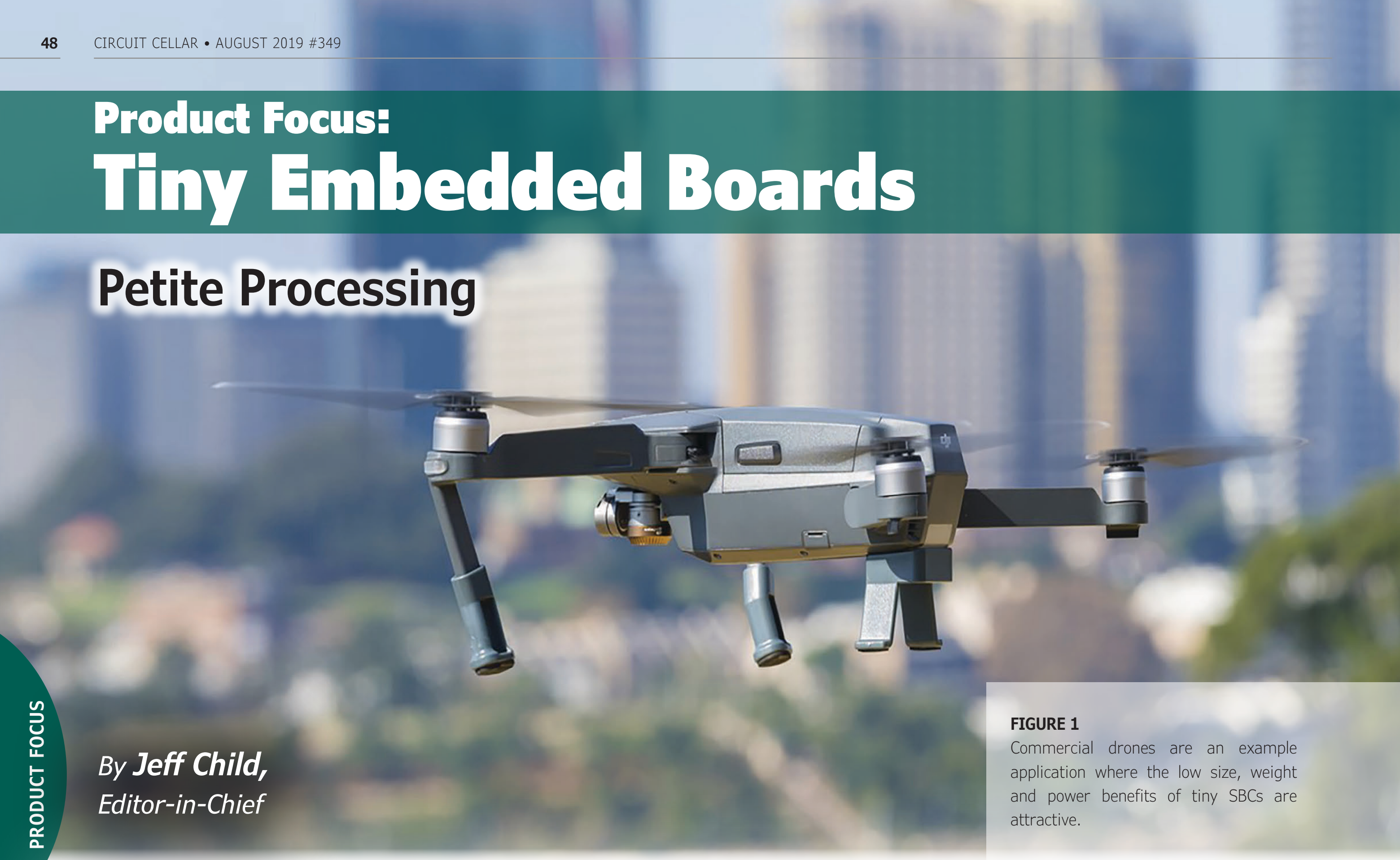


FIGURE 1
Commercial drones are an example application where the low size, weight and power benefits of tiny SBCs are attractive.

By *Jeff Child,*
Editor-in-Chief

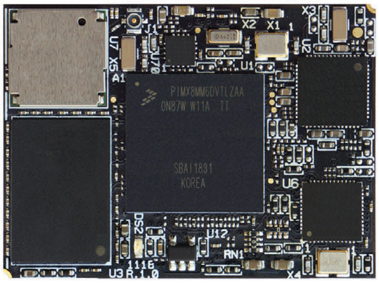
With today's level of integrated chip technology, board-level embedded computers can now literally fit in the palm of your hand. This "tiny" category of board-level computing products meets the needs of applications where extremely low SWaP (size, weight and power) is a priority over other requirements.

Long gone are the days when a complete embedded computer required a PC motherboard-sized PCB with processor, memory modules and I/O cards. The magic of semiconductor integration has brought powerful levels of functionality down to one or a handful of chips. Makers of board-level embedded computers have leveraged that trend to achieve board-level computers that now occupy extremely small form factors. Many of these are non-standard form factors. Non-standard form factors free designers from the size and cost overheads associated with including a standard bus or interconnect architecture. That said, standard form factors such as SMARC, SODIMM and COM Express Type 10 Mini are within the size range of this "tiny" category of embedded processor boards.

For purposes of this product round-up, we've selected representative products that are the smallest available on the market—most under 50 mm × 80 mm, and going as tiny as 28 mm × 38 mm. Side by side comparisons should take into account that some of these products are meant to be used as compute modules attached to an I/O baseboard, while some are true complete single board computers (SBCs).

For embedded applications where small size is critical, often the size and volume of the board takes precedence over the need for standards. Instead the priority is on cramming as much functionality and compute density onto a single board solution. Some, but not all, of these modules are true SBCs solutions, so there's no need to be compatible with multiple companion I/O boards. These tiny form factor boards seem to be targeting applications where slot-card backplane or PC/104 stacks wouldn't be practical. Many boards in this article's product gallery are based on processors like NXP's i.MX6, i.MX7 and i.MX8M SoC, as well as the newer i.MX8M Mini. The Rockchip RK3308 SoC gets some representation as well.

The list of applications targeted by these tiny embedded boards is long and varied. Included are systems such as professional handheld devices, autonomous drones (**Figure 1**), portable instrumentation with HMI, wearables, medical devices, digital signage, industrial HMI, home automation and audio/video streaming devices, scanning/imaging, building automation, smart homes and machine vision.



Tiny SBC Sports NXP i.MX8M Mini SoC

Compulab's 38 mm x 28 mm UCM-iMX8M-Mini module features an i.MX8M Mini, Wi-Fi/BT and up to 4 GB RAM and 64 GB eMMC. The COM also ships on a sandwich-style SBC-iMX8M-Mini SBC. CompuLab is aiming its new module at applications including professional handheld devices, autonomous drones, portable instrumentation with HMI and wearables, including medical devices.

- Quad-core Arm Cortex-A53 CPU, 1.8 GHz
- Real-time Arm Cortex-M4 coprocessor
- Up to 4 GB LPDDR4 and 64 GB eMMC
- MIPI-DSI interface, Wi-Fi 802.11ac, BT 4.2
- GbE, PCIe, 2x USB, 4x UART, 85x GPIO
- Wide temperature range of -40°C to 85°C
- Form factor: 28 mm × 38 mm × 4 mm

Compulab
www.compulab.com

Module Marries NXP i.MX6 with Wi-Fi/BLE

The eSOMiMX6PLUS from e-con Systems is a high performance and low power Computer/System-on-Module based on NXP semiconductor's i.MX6 processor with the basic peripherals in a small compact form factor. The eSOMiMX6PLUS has an i.MX6 processor running at 1/1.2 GHz with Arm Cortex-A9 and DDR3L SDRAM configurable up to 4 GB.

- 1.2 GHz i.MX6 QuadPlus/DualPLus/ Quad/Dual/DualLite/Solo Arm Cortex-A9
- DDR3L ranging from 2 GB to 4 GB
- eMMC NAND flash 8 GB to 64 GB
- Over-the-Air (OTA) Firmware Upgrade Solution
- Wi-Fi 802.11 a/b/g/n/ac and Bluetooth V4.2/BLE
- Gbit Ethernet (x1) - optional
- Form factor: 70 mm × 45 mm× 4.4 mm

e-con Systems
www.e-consystems.com

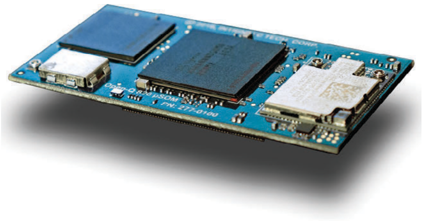
Tiny Module Runs Linux on i.MX8M Mini

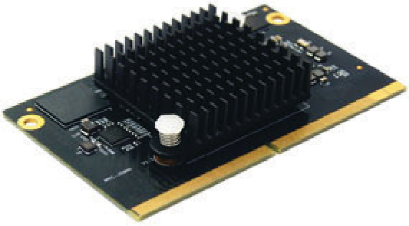
The PicoCore MX8MM from F&S Elektronik Systeme is module that runs Linux on an up to quad-A53, 1.8 GHz i.MX8M Mini with up to 8 GB LPDDR4. The NXP i.MX8M Mini is an i.MX8M variant with lower video resolution, but a faster clock rate. The PicoCore MX8MM has the same 40 mm x 35 mm footprint as other F&S PicoCore models such as the i.MX7-based PicoCore MX7ULP.

- i.MX8M Mini with 1.6 GHz 800 MHz Arm Cortex
- Up to 8 GB LPDDR4 RAM
- 512 MB NAND flash, up to 32 GB eMMC
- Wi-Fi 802.11 ac and Bluetooth 5.0
- I/O includes USB OTG, 4x UART, 4x I2C, 4x SPI and 8x PWM
- Power consumption: 3 W typical
- Form factor: 40 mm × 35 mm
- 0 to 70°C and -40°C to 85°C options

F&S
www.fs-net.de

Tiny Embedded Boards





Snapdragon 820E Card Has Long Lifecycle Support

Intrinsyc' Open-Q 820Pro μSOM is a pin-compatible drop-in replacement for the two-year old Open-Q 820 μSOM and offers a similar layout and 50 mm × 25 mm footprint. The biggest difference is an upgrade from Qualcomm's Snapdragon 820 to the faster, second-gen Snapdragon 820E, an embedded-focused version with long lifecycle support.

- Qualcomm Snapdragon 820E (APQ8096SG) Processor
- Adreno 530 GPU and Hexagon 680 DSP
- 4 GB LPDDR4, 32 GB UFS flash storage
- Wi-Fi 802.11a/b/g/n/ac and Bluetooth 4.2
- HDMI 2.0, 2x MIPI 4-lane DSI and touch panel
- Form factor: 50 mm × 25 mm × 4.5 mm
- 1x USB 3.0, 1x USB 2.0, SPI, UART, I^{2}C
- Single GPIO, PWM, SPI and PCIe

Intrinsyc
www.intrinsyc.com

SODIMM Module Serves Up i.MX8M Mini or Nano

iWave's iW-RainboW-G34M-is an SODIMM System on Module (SOM) featuring the NXP i.MX 8M. It is aimed at applications such as digital signage, industrial HMI, home automation, audio/video streaming devices and general embedded applications. The board offers support for either NXP's i.MX8M Mini or the upcoming i.MX8M Nano.

- Cortex-A53 at up to 1.6 GHz
- Cortex-M4F at 400 MHZ
- 1080p60 VP8, 1080p60 VP9, 1080p60 Encoder
- 1080p60 Decoder VPU
- Wi-Fi 802.11a/b/g/n/ac and BLE v4.2
- Dual 1000/100/10 Mbps Ethernet
- 2 GB LPDDR4 memory (Expandable)
- Form factor: 67.6 mm × 37 mm

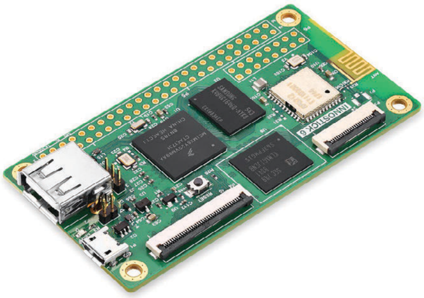
iWave Systems Technologies
www.iwavesystems.com

i.MX8M CPU Module Offers Multimedia Solution

MYIR's MYC-JX8MX CPU module provides an embedded solution for scanning/imaging, building automation and smart home, HMI, machine vision and other consumer and industrial applications which require high multimedia performance. The card is based on NXP i.MX 8M quad processor featuring 1.3 GHz quad Arm Cortex-A53 cores and a real-time Arm Cortex-M4 co-processor.

- NXP i.MX 8M based on 1.3 GHz Arm Cortex-A53 and 266 MHz Cortex-M4 Cores
- 1 GB / 2 GB LPDDR4
- 8 GB eMMC flash, 256 Mbit QSPI flash
- On-board Gbit Ethernet PHY
- ROHM Power Management IC (PMIC)
- 0.5 mm pitch 314-pin MXM 3.0 Expansion Connector
- Operating temperature: -30°C to 80°C
- Form factor: 50 mm x 82 mm

MYIR Tech Limited
www.myirtech.com

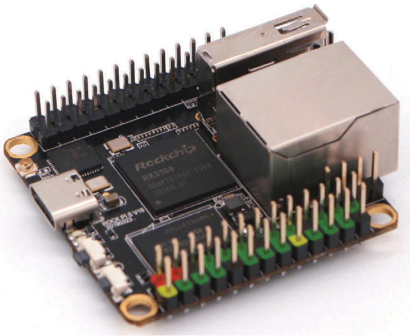


SBC Sports Low-Power 900 MHz i.MX6 ULL SoC

NXElec's Innostick 6 SBC is based on NXP/Freescale's i.MX6ULL Arm Cortex-A7 processor with up to 900 MHz CPU clock. It is integrated with on board DDR3 memory and eMMC storage in a tiny form factor: 80 mm x 42 mm. The INNOSTICK 6 provides a variety of interfaces and connectivity options, all packaged at an optimized power, size and cost.

- NXP i.MX6 ULL; 1x Cortex-A7 at 900 MHz
- 512 MB DDR3L; 16 GB or 32 GB eMMC
- Wi-Fi and Bluetooth 4.0
- 24-bit RGB LCD interface with resistive touch support
- 8-bit CSI Parallel input
- USB 2.0 host, Micro-USB 2.0 OTG, 16-pin and 50-pin expansion headers
- LED; reset button; boot and power jumpers
- Form factor: 80 mm x 42 mm

NXElec
www.nxelec.com

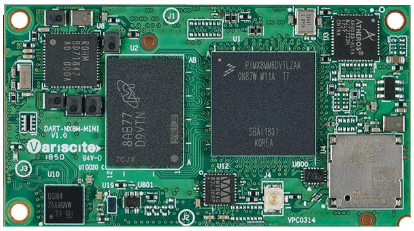


Tiny SBC is Based on Rockchip RK3308 SoC

The Rock Pi S is an SBC from Radxa that runs Linux. Aimed at IoT and audio applications, the headless SBC measures only 43 mm x 43 mm. The RK3308 SoC features four low-power, 64- bit Cortex-A35 cores. The board has extensive audio interfaces that include I^{2}S, PCM, TDM, PDM, SPDIF and HDMI ARC.

- 64-bit Rockchip quad core RK3308 processor
- USB, Ethernet, Wireless connectivity
- Voice detection engine
- 256 MB or 512 MB DDR3
- µSD card for OS and storage
- Optional on-board flash storage version
- Form factor: 43 mm x 43 mm

Radxa
wiki.radxa.com

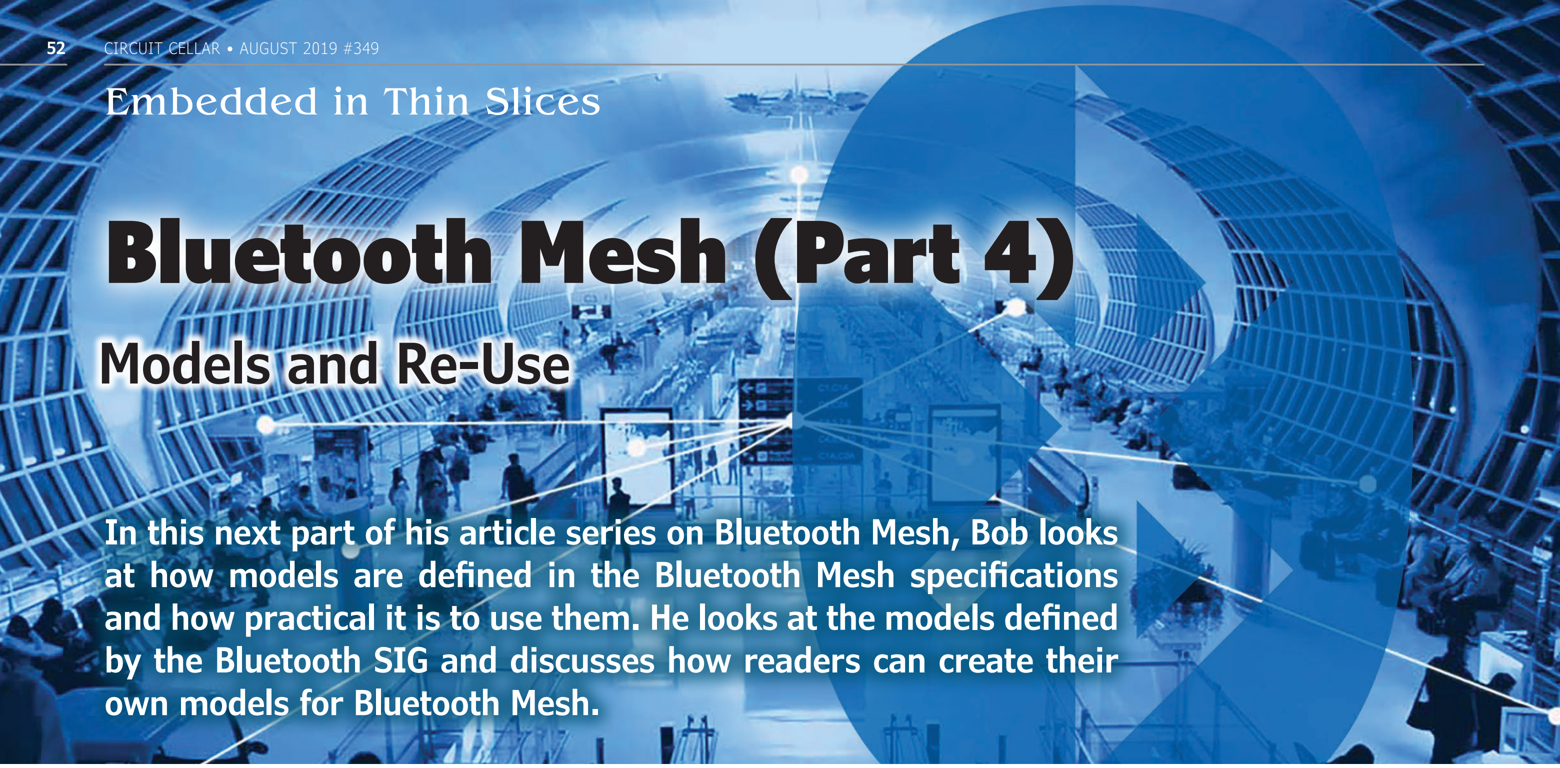


Compute Module Delivers i.MX8M Mini SoC

Variscite's DART-MX8M-MINI System on Module is based on NXP's i.MX8M Mini with up to 2 GHz Quad-core Arm Cortex-A53 plus 400MHz Cortex-M4 real-time processor, leveraging advanced low power silicon process technology to provide optimized power consumption while maintaining a high-performance bar. The module provides a pin-to-pin scalable option to the DART-MX8M.

- Up to 2 GHz Quad Cortex-A53 NXP i.MX8M Mini
- Certified Wi-Fi/Bluetooth 4.2/BLE
- Temperature range: -40°C to 85°C
- Up to 4 GB LPDDR4 and 8 GB to 64 GB eMMC
- Gbit Ethernet controller
- 1080p60 encode/decode and display, HQ audio, HD GPU
- 2x USB 3.0 OTG, 4x UARTs, 3x I^{2}C, 3x SPI and PCIe 2.0 x1
- Form factor: 55 mm x 30 mm

Variscite
www.variscite.com



Bluetooth Mesh (Part 4)

Models and Re-Use

In this next part of his article series on Bluetooth Mesh, Bob looks at how models are defined in the Bluetooth Mesh specifications and how practical it is to use them. He looks at the models defined by the Bluetooth SIG and discusses how readers can create their own models for Bluetooth Mesh.

By
Bob Japenga

Some of you might remember Grady Booch. Grady is well known for his part in developing the Universal Modeling Language (UML) that many of us use to create quality software specifications. I remember Grady from his days when he was promoting software re-use using the programming language Ada. As a software engineering visionary, he saw the re-use of Ada packages as the silver bullet to slay the vampire of software development cost and schedule overruns.

Grady envisioned a world of Ada packages that would revolutionize software development. Perhaps he never read that Fred Brooks, in his book *The Mythical Man-Month*, said that the silver bullet would never be found. (I am sure he did read this classic.) Is anyone out there re-using Ada packages today? Let me know! Perhaps his move to develop a better way of specifying and developing software through UML may have been motivated by the failure of the re-use of Ada packages. Certainly, creating clear and more deterministic specifications—which UML does—has improved our ability to create reliable software on-time and on-budget.

The promise of re-use remains elusive to those of us in the real world of software development. Certainly, some things get re-used, but we are far from Grady's vision. Most re-use takes place in the rich libraries that are provided by our operating systems and our Software Development Kits (SDK).

During the past few months, we've been looking at Bluetooth Mesh in this article series. One of the things that the Bluetooth SIG created in the Bluetooth Mesh specifications is a set of models for communication to a variety of Bluetooth devices. Hundreds of pages of the specifications are devoted to defining how to communicate to lights, sensors, batteries and generic on/off devices. It also includes models for timing and scheduling. This month we will look at what the specification provides and what is actually available from the Bluetooth community to keep us from re-inventing the wheel and to perhaps fulfill a tiny bit of Grady's expansive vision for re-use.

MODELS IN BLUETOOTH MESH

The Bluetooth SIG created a paradigm in the specification to describe what a device can do on the mesh network. Two of the building blocks in that specification are the concept of the element and the model. An element is that part of the device that has been assigned a unique address at the time of provisioning. A given device on the mesh network has one or more elements each with a unique address. Each element must support one or more models.

Meanwhile, a model defines a collection of functionality and behaviors for the element. There are three categories of models: Server, Client and Controller. A Controller contains both Client and Server functionality as well as control functionality. There are four types of models defined in the specification: the generic model, the sensor model, the lighting model and a model of various timing functions and behaviors. Much like inheritance in object-oriented programming languages, new models can inherit behavior from other models. A model that doesn't inherit from any other model is called a root model.

With all that in mind, each device on the network can have multiple elements in them and each element can have multiple models. For example, imagine a light fixture that has three lamps. One lamp is a simple on/off. Another lamp is dimmable. The third lamp can provide a range of colors as well as being turned on or off and dimmable.

In 1999, the International Standards Organization proposed a standard set of layers of network functionality (**Figure 1**) [1]. The Bluetooth SIG created a slightly different set of layers for looking at Bluetooth Mesh network functionality (**Figure 2**) [2]. The Foundation Model Layer is responsible for implementing the models that are needed to configure and manage the network. The Model Layer is defined by the specifications and is an integral part of all Bluetooth Mesh Networks.

Another concept that is very closely related to the Model is the Scene. A Scene is a collection of states of the device. Using our light fixture example, let's assume that the light fixture is used in a theatrical play. Each scene in the play would have different lamp intensity, different lamps on and different colors. The message structure for this is present in the specification to allow you to create different Scenes.

A Bluetooth Mesh Scene would correspond to a scene in the play. Another example of using the Scene concept would be a device containing a number of sensors. The sensors may be configured for three different modes: one full-power, one half-power and one off. Each of these power levels could be described as a Scene. The specification defines a Scene Server, a Scene Client and a Scene Setup Server, which includes all the commands needed to create Scenes.

DRILLING DOWN

Bluetooth SIG has done a good job and defined a number of models and scenes that will cover a lot of the devices we will be developing. **Table 1** provides a list of the comprehensive and extensible models that are defined. Bluetooth module vendors can create other models. As implementers, we are free to create other models.

Let's look more in depth at our light fixture with three different types of lamps (**Figure 3**) [3]. Our light fixture has a single Bluetooth LE radio. Each of the separate lamps is called an element. Each element uses one or more different models defined by the Bluetooth Mesh Specification. Each element has a unique address assigned to it during provisioning. Each of the three lights has states, messages and models as defined in the Bluetooth Mesh Model specification.

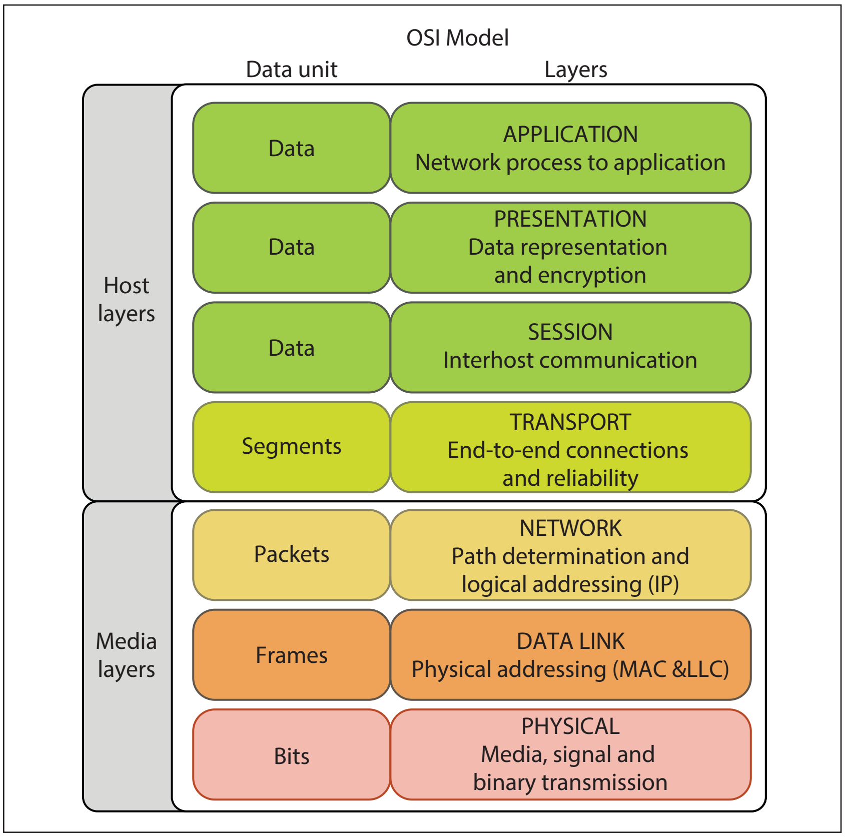

OSI Model
Data unit
Layers
Host layers
Data
APPLICATION Network process to application
Data
PRESENTATION Data representation and encryption
Data
SESSION Interhost communication
Segments
TRANSPORT End-to-end connections and reliability
Media layers
Packets
NETWORK Path determination and logical addressing (IP)
Frames
DATA LINK Physical addressing (MAC &LLC)
Bits
PHYSICAL Media, signal and binary transmission

FIGURE 1
ISO created this Open Systems Interconnection (OSI) standard, a set of layers of network functionality.

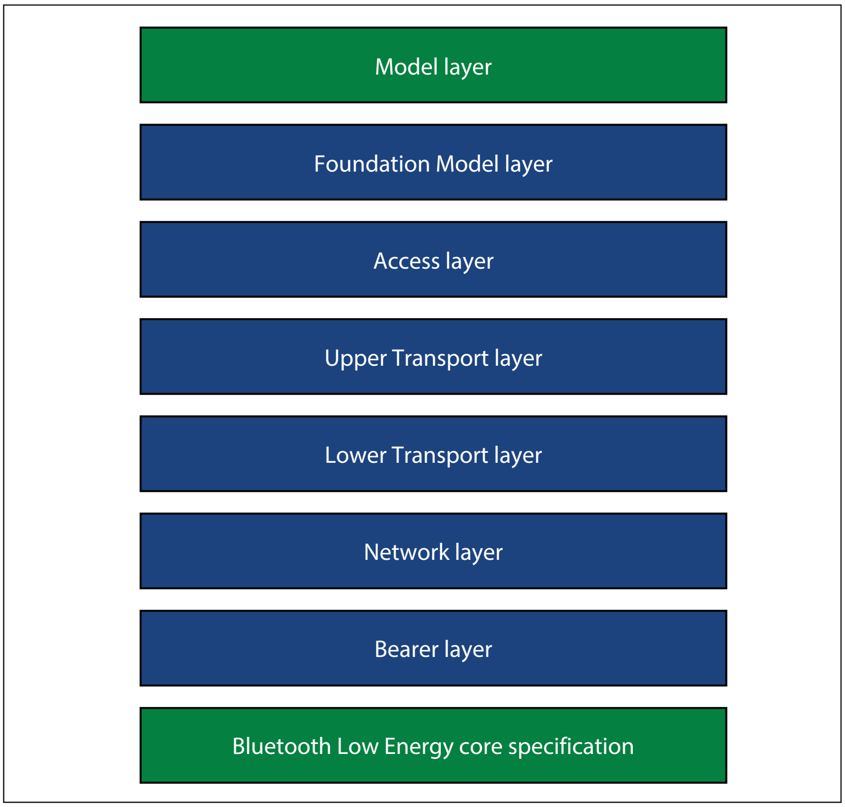


FIGURE 2
In contrast to the ISO's set of layers shown in Figure 1, the Bluetooth SIG created a slightly different set of layers for looking at Bluetooth Mesh network functionality.

The functionality of the simple lamp element is defined by the Generic OnOff Server Model. It can be controlled by another device on the mesh network that uses the Generic OnOff Client Model. The functionality of the dimmable light element in the light fixture is defined by the Light Lightness Server model. Lightness is similar to brightness—lightness is perceived reflectance, brightness is perceived luminance. It seems to me that they are controlling brightness. This model inherits functionality from several generic models like the Generic OnPowerup model, the Generic OnOff model and the Generic Level model.

The Light Lightness Model also picks up some unique features like the default settings,

Model Group	Model Name	Description
Generic	Generic OnOff Server	Used for responding to the request to turn an element on or off
	Generic OnOff Client	Used for turning an element on or off
	Generic Level Server	Used for responding to the request to set an element to a certain level
	Generic Level Client	Used for setting an element to a certain level
	Generic Default Transition Time Server	Used for responding to the request to set how long an element shall take to transition from a present state to a new state
	Generic Default Transition Time Client	Used for setting how long an element shall take to transition from a present state to a new state
	Generic Power OnOff Server	An extension of the Generic OnOff Server to encapsulate the powering on/off of the element in the node
	Generic Power OnOff Setup Server	Used to set up the timing characteristics of power on/off for the element in the node
	Generic Power OnOff Client	An extension of the Generic OnOff Client to encapsulate the powering on/off of the element in the node
	Generic Power Level Server	An extension of the Generic Level Server to encapsulate the powering on/off of the element in the node
	Generic Power Level Setup Server	Used to set up the timing characteristics of power levels for the element in the node
	Generic Power Level Client	An extension of the Generic Level Client to encapsulate the powering on/off of the element in the node
	Generic Battery Server	Used for sending battery status and battery parameters
	Generic Battery Client	Used for requesting battery status and battery parameters
	Generic Location Server	Used for responding to the request for latitude, longitude and altitude
	Generic Location Setup Server	Used for responding to requests to set latitude, longitude and altitude
	Generic Location Client	Used to send or request latitude, longitude and altitude
	Generic Admin Property Server	Used to respond to requests for administrator level properties
	Generic Manufacturer Property Server	Used to respond to requests for manufacturing level properties
	Generic User Property Server	Used to respond to requests for user properties
	Generic Client Property Server	Used to obtain properties of the client
	Generic Property Client	Used to set and request properties for all levels
Sensors	Sensor Server	Used for responding to requests for sensor data
	Sensor Setup Server	Used for responding to requests to setup sensors
	Sensor Client	Used for setting sensor parameters and obtain sensor readings

the range of brightness and the last state of brightness. The specification provides all of the messages needed to control a dimmable light. All of the parameters needed to set up this dimmable light are handled by the Light Lightness Setup Server model. The light can be controlled by a device that implements the Light Lightness Client model.

The functionality of the colored light element in our light fixture is also handled via models defined in the specification. Various hues, saturation levels and brightness of the colors of the light are all controllable with the HSL (Hue, Saturation and Lightness) model. Both client and server models are defined for this model.

WHERE'S THE RE-USE?

Before we look at this question, let me provide our history in using these models. Last year, we were involved with a company that was implementing a very simple lighting device using Bluetooth Mesh. The Bluetooth vendor (who will remain anonymous)

Model Group	Model Name	Description
Times, Schedulers and Scenes	Time Server	Note: Mesh uses International Atomic Time, which is different than UTC. The server delivers time, time zone, time authority information to Time clients.
	Time Setup Server	Used for setting up time, time zone, known uncertainty, time authority and other time related parameters.
	Time Client	Used for obtaining time, time zone and other time related parameters
	Scene Server	Scenes are used to group setup parameters to create a particular scene (e.g., Lighting controls that set a particular mood or ambiance). The server used for responding to requests for particular scenes
	Scene Setup Server	Used for setting the parameters for particular scenes
	Scene Client	Used for setting and requesting the particular scenes
	Scheduler Server	Used to schedule changes to the states and properties of an element at a particular time and date
	Scheduler Setup Server	Used to set up schedules
Lighting	Light Lightness Server	Used to respond to requests for the level of brightness
	Light Lightness Setup Server	Used to set up the Lightness settings
	Light Lightness Client	Used to control the level of brightness in a server
	Light CTL Server	Used to respond to requests for the color temperature
	Light CTL Setup Server	Used to set up the color temperature settings
	Light CTL Client	Used to control the color temperature in a server
	Light CTL Temperature Server	Used to respond to setting of the color temperature of tunable white light
	Light HSL Server	Used to respond to requests for the Hue, Saturation and Lightness settings
	Light HSL Setup Server	Used to setup the Hue, Saturation, and Lightness settings
	Light HSL Client	Used to control the Hue, Saturation and Lightness in a server
	Light HSL Hue Server	Used to respond to requests for the Hue settings
	Light HSL Saturation Server	Used to respond to requests for the Saturation settings
	Light xyL Server	Used to respond to requests for the color
	Light xyL Setup Server	Used to set up the color settings
	Light xyL Client	Used to control the color in a server
	Light LC Server	Used to respond to requests for a Lighting Controller
	Light LC Setup Server	Used to set up the Lighting Controller
	Light LC Client	Used to control the Lighting Controller

TABLE 1
The Bluetooth SIG has defined a number of models and scenes that will cover a lot of devices. This table provides a list of the comprehensive and extensible models that are defined.

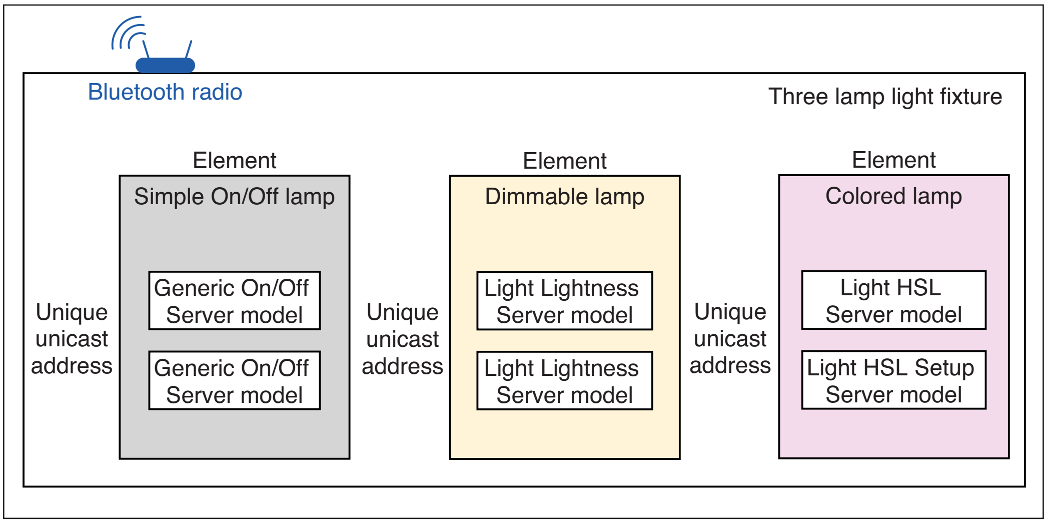


FIGURE 3
This example light fixture has a single Bluetooth Low Energy (BLE) radio. Each of the separate lamps is called an element. Each element uses one or more different models defined by the Bluetooth Mesh Specification. Each element has a unique address assigned to it during provisioning. And each of the three lights has states, messages and models as defined in the Bluetooth Mesh Model specification.

agreed to implement the Client side of the project and help us with the Server side of the project. The project started with a pilot program to demonstrate feasibility to a very large customer. After several months of glacial progress, we initially thought that the problem was that the developers did not have enough time to put into the project. However, when our deadline was approaching and the work was still not getting done, it became very clear to us that the problem was that the developers had to develop all of the lighting models themselves.

Very few of the models needed were implemented in the vendor's Software Development Kit (SDK). None of the lighting models were implemented—although there was a simple lighting example that did not use any of the lighting models! In fact, only 12 of the 51 models were implemented in the SDK. Most of the models needed for our implementation needed to be developed. So much for re-use!

The Bluetooth SIG site lists 139 products that have Bluetooth Mesh networking capability and have successfully completed the Bluetooth Qualification Process. A few of the vendors have implemented all of the models. Some, like the one we used, implemented a small subset. One vendor is clear on their webpage that they only support some of the models—foundation, generic and some of the lighting models—but not the sensor model nor the timing models. This is confirmed in their SDK. Others are not clear and you have to drill down into the SDK to find out what is provided. If we want to minimize development time and cost, the bottom line for us as developers is to find a vendor that provides the models that we need.

CONCLUSION

Re-use and interoperability are wonderful in concept but are extremely difficult to implement. Re-use takes a different form today than was envisioned by Grady Booch. Today, re-use happens when we can draw upon rich libraries provided by the vendors for the chips we are using. In an ideal world, I would love to have a Bluetooth Mesh network with lights and sensors from a variety of suppliers and have them all be interoperable. I would love to be able to create my own light fixture and have the Bluetooth chip supplier's SDK provide most of the code. The Bluetooth SIG has laid out a paradigm for that to happen. Right now, just two years after the specification was released, it may be too early to tell. But two years is a long time in the world of IoT.

In the past few articles I've introduced you to Bluetooth Mesh. Clearly, we have explored it in thin slices. Although there is much more that I could write about, I want to move on next time to explore the very, very low bandwidth wireless data communication technology known as LoRa.

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.

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.

SoC 17th 2019
INTERNATIONAL SYSTEM-ON-CHIP CONFERENCE

Early Bird Registration Is Now Open!

17th International System-on-Chip (SoC) Conference & Exhibit

October 16 & 17, 2019 - University of California, Irvine - Calit2

www.SoCconference.com

Four Enlightening Keynotes on Emerging SoC Technologies

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

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

University of Texas-Austin
Dr. Lizy Kurian John, P.E. The Cullen Trust for Higher Education.

Xilinx
Jennifer Wong, Vice President of FPGA Product Development.

Platinum Sponsors

UCI Division of Continuing Education
Handycash Buy it cash with a top.
circuit cellar OFFICIAL PARTNER
Calit2

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

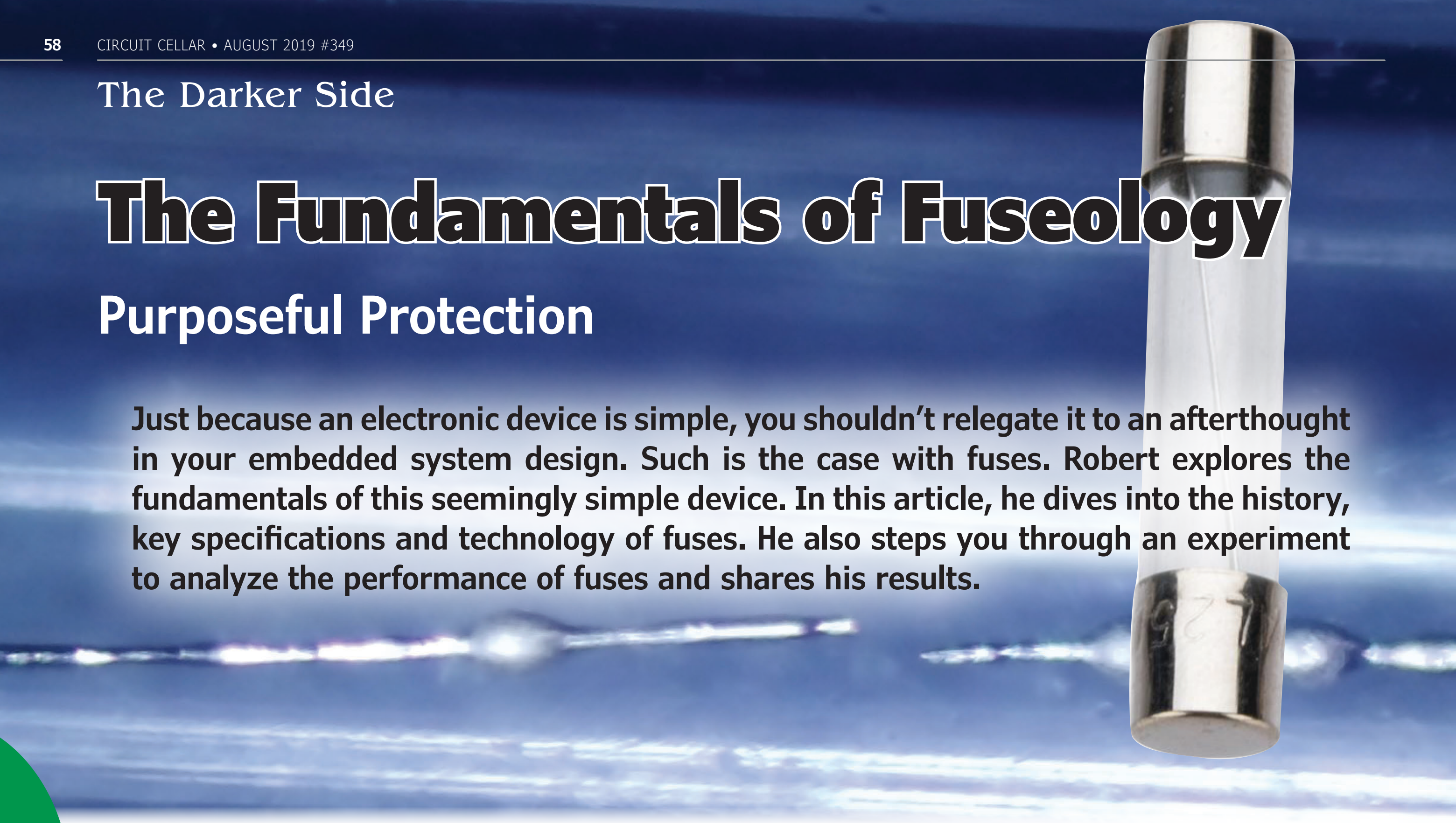
- RISC-V Realities, Opportunities and Challenges in the Complex and Crowded CPU Market.
- Will AI and Machine Learning Drive the Next Semiconductor Revolution?

Selected Presenting Companies & Universities

UCI University of California, Irvine · SAVANT COMPANY INC. · GOWIN · IBM · intel · Stanford University · IEEE SSCS
University of Bologna · skywater TECHNOLOGY FOUNDRY · NANTERO · ARM · XILINX · AONdevices AI | DSP | ASIC · TEXAS The University of Texas at Austin
Mellanox TECHNOLOGIES · netObjex · FUJITSU · tsmc · Georgia Institute of Technology · Leica MICROSYSTEMS · CRC Press Taylor & Francis Group
Handycash Buy it cash with a top. · SEGGER · INTRINSIC ID · IHS Markit · UC San Diego Jacobs School of Engineering · University of Bologna · Calit2
LedgerDomain · UCI Samueli School of Engineering University of California, Irvine · University of Bologna · circuit cellar OFFICIAL PARTNER · UNIVERSITY OF MICHIGAN · PALMCHIP · I TECH

For More Information or Questions, Please Contact the SoC Conference Organizing Committee at:

MySoCconference@Gmail.com or (949) 981-1837 & (949) 356-2399



The Darker Side

The Fundamentals of Fuseology

Purposeful Protection

Just because an electronic device is simple, you shouldn't relegate it to an afterthought in your embedded system design. Such is the case with fuses. Robert explores the fundamentals of this seemingly simple device. In this article, he dives into the history, key specifications and technology of fuses. He also steps you through an experiment to analyze the performance of fuses and shares his results.

By
Robert Lacoste

Welcome back to the "Darker Side." As electronic system designers, we've become used to dealing with some fantastic and ultra-complex pieces of silicon in our projects—microcontrollers running at hundreds of megahertz, multi-core processors with billions of transistors, wireless transceivers with Gbps of throughput, miniature power converters with close to 100% efficiency and so on. Ok, of course some small discrete parts are still required around those key building blocks, but we're inclined to disdain such components in the design phase. That's because they represent a very small portion of the overall bill of materials and have low perceived value.

All that said, if you are a regular reader of this column, you already know that's a bad choice. Some electronic components seem very simple—passives in particular. But such devices may be the source of incredible trouble if you don't understand the intimate details of their behavior. If you have any doubt, go and re-read my articles on capacitors—for example, *Circuit Cellar* 283 "Dielectric Absorption;" *Circuit Cellar* 317 "Decoupling Capacitors and RLC Networks;" *Circuit Cellar* 321 "All Ceramic Capacitors Aren't Equal."

This month, I will talk about another very simple part that isn't as simple as it seems: The fuse.

WHAT'S A FUSE?

Of course, you've all seen a fuse before. Fuses are as old as electricity. According to *Wikipedia*, their first documented use was in 1864 for telegraph installations [1]. The first patent on a fuse was registered by Thomas Edison (him again?) in 1890. Today, fuses are everywhere, and range from ultra-miniature, surface-mounted devices to massive units used in nuclear-powered generators. Let's restrict the discussion to small fuses common in electronic devices, such as the ubiquitous 20 mm x 5 mm fuse cartridge, illustrated in **Figure 1**. The picture speaks for itself—a fuse is nothing more than a wire. It is designed to be a protection device, and open the circuit in case of overcurrent. The wire is designed to melt above a given current threshold and to open the circuit.

Let's spend a few minutes on these words: "protection device." What does this mean? What is protected by the fuse? The answer to this question is not as obvious as it seems, because a fuse serves two purposes. First, it helps to protect the components of the device itself—meaning the device after the fuse—from extensive damage in case of a fault. For example, a fuse at the input of a power supply could save sensitive parts from destruction if the power supply malfunctions. Second, a fuse isolates the device from the outer world when the device is faulty, and this helps to prevent greater damage to other equipment.

"Protection device" also means that a fuse should not be, by itself, a potential source of hazard. When the wire in a fuse is melting, it will be hot and liquid, and could start a fire without adequate precautions. That's why a fuse wire is always hermetically sealed, like the glass tube in Figure 1. That's a requirement. Fuses are regulated by standards, mainly IEC 60269 [2] (for residential or large fuses) and IEC 60127 (for miniature fuses like my 20 mm × 5 mm example). Ok, Americans prefer UL248, which is a different standard—but the spirit is the same. In any case, these standards state that a fuse should not allow any external sign when a fault occurs. In other words, that means that everything should be contained within the fuse body. No smoke or other material is expelled. This is true as long as the fuse is used within its specifications. More on that in a minute.

The term "overcurrent" also needs some explanation. What is an overcurrent? Is it a current just above the nominal current? For how long? Or a short circuit with thousands of amperes? Let's dig into more details ...

FUSE SPECIFICATIONS

At this point, I encourage you to look for the datasheet of any standard fuse, and to read it carefully. Of course, you will find that a fuse is first specified by its package type and rated current. The rated current, written on the fuse, is simply the maximum current that it can continuously conduct without any problem.

The second key characteristic of a fuse is its speed. How fast will it blow in case of trouble? As you might expect, this depends on several parameters, and the first is the current. The greater the current passing through the fuse, the faster the wire will melt and cut the link. What are the tolerated limits? For miniature fuses, two speed grades are available and specified by EIC60127-2: Quick acting ("F" type, for "fast") and time-lag ("T" type). Typical values, their respective minimum and maximum breaking times, depending on the effective current are given in **Table 1**. A caution here: Standards are evolving, so always consult the latest official version of the standards for any precise information. Now, look again at Table 1. You will see, for example, that a quick-acting miniature fuse, when a current 275% higher than its rating is applied, must cut the wire in less than 2 seconds, but not less than 10 ms. These durations become respectively 10 seconds and 600 ms for a time-lag version.

FIGURE 1
A typical 20 mm x 5 mm miniature fuse, nothing more than a wire in a sealed glass tube.

Two important points here. First, look again at the second line of Table 1. It might be surprising. When the current is 210% of the fuse rating (for example, 4.2 A for a 2 A fuse), the only requirement is that the fuse must blow in less than 30 minutes. And 30 minutes is an eternity, compared to the lifetime of a stressed transistor. That's why fuses usually don't protect semiconductors!

The second point is that this table comes from IEC60127-2, which, as noted, is the international standard. Unfortunately, it isn't used by the US, Canada or Japan, which instead use UL/CSA248-14. Both standards are more or less similar, but are clearly not

IEC 60127-2:2014 5 mm × 20 mm fuse						
	Quick acting High-breaking capacity 50m A to 4 A		Quick acting Low-breaking capacity 100 mA to 6,3 A		Time-lag Low breaking capacity 100 mA to 10 A	
Percent of rating	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
150.00%	1 h	/	1 h	/	1 h	/
210.00%	/	30 min	/	30 min	/	2 min
275.00%	10 ms	2 s	50 ms	2 s	600 ms	10 s
400.00%	3 ms	300 ms	10 ms	300 ms	150 ms	3 s
1000.00%	/	20 ms	/	20 ms	20 ms	300 ms
Breaking capacity	1500 A		35 A or 10× rate		35 A or 10× rate	

TABLE 1
Here are some typical miniature fuse tripping times, depending on current and fuse type (from IEC60127-2:2014).

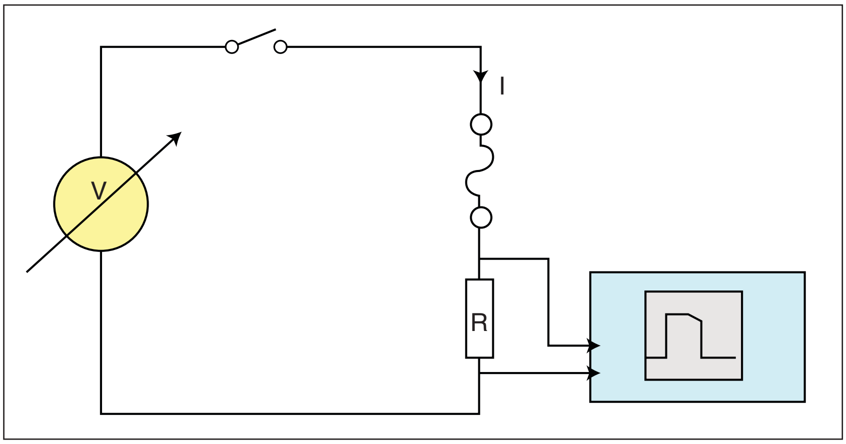


FIGURE 2
A very simple experiment to measure fuse breaking time with an oscilloscope

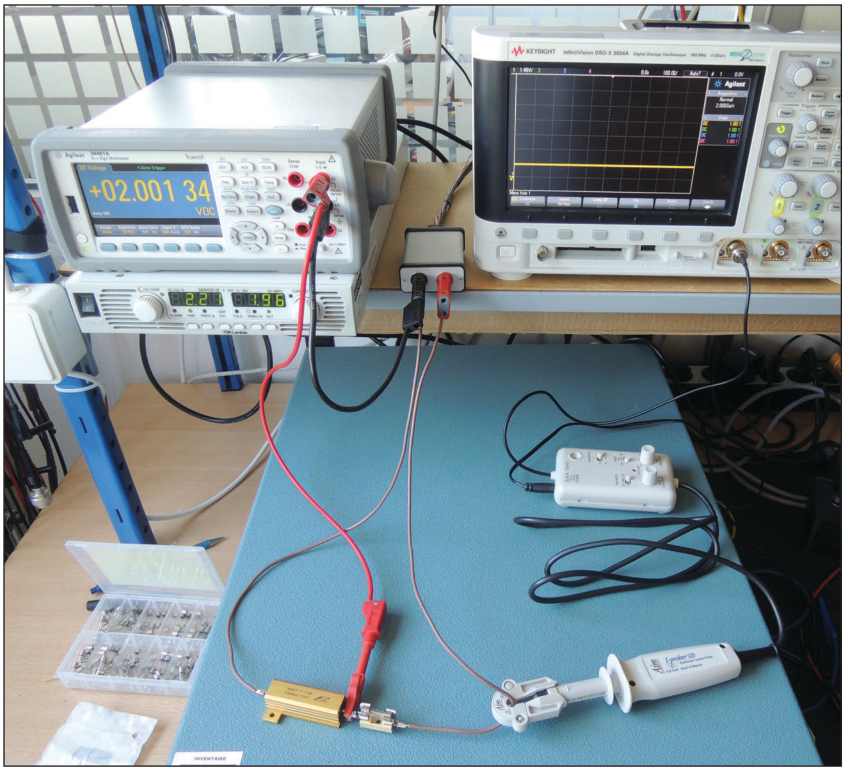


FIGURE 3
Here is my lab setup for the test. Thanks to my company, ALCIOM, for the equipment!

identical. That means that a 2 A, 20 mm × 5 mm "F" fuse according to the IEC standard is not the same part as its UL equivalent! The differences may seem minor, but they exist, and you should be aware. The specific reference standard should be defined in the bill of materials. You'll find more information on this topic in the selection guide document "Fuseology" by Littelfuse [3].

AMBIENT TEMPERATURE

Another difficulty is that ambient temperature must be taken into account. Of course, metal melts more easily if the ambient temperature is high. This parameter—namely the temperature derating—is defined by the fuse manufacturer and should not be overlooked. A 10% current trip-off variation from minimum to maximum temperature is not unusual. Once again, this may or may not be an issue, but if some customers complain that their fuses blow when others don't, you'd better check their ambient temperature. If the first lives in Hawaii while the second is in Iceland, then you might have a clue why.

Another important specification of a fuse is its so-called "breaking capacity." This is the maximum instantaneous current at which a fuse will safely cut. Why such a limit? Because physics is, from time to time, insidious. Take, for example, a standard miniature fuse like the one shown in Figure 1, and apply a huge current, say 10,000 A. The wire will vaporize instantaneously, and the metal of the wire will probably be deposited on the inside of the glass tube.

If you are unlucky (and Murphy's law states you will be), this deposit will make a conducting path between the two electrodes and the current won't be cut. That's why standard miniature fuses have a limited breaking capacity. IEC standard states it should be at minimum 35 A or 10 times the rated current, whichever is higher. If this could be an issue in your design, you'd better know that miniature fuses with high breaking capability do exist. These models are specifically designed to guarantee a 1,500 A breaking capacity, and are usually filled with sand. The sand limits the risk of metal deposition.

Finally, fuses have a rated voltage, which is simply the maximum safe voltage that a fuse can cut. If there is a higher voltage between the two electrodes, then there is a risk of arcing. A plasma will be formed between the electrodes, and will conduct some current even if the wire is cut. This rated voltage must, of course, be higher than the circuit's working voltage. A caveat here—you might think you're clever and try to connect fuses in series to try to increase their rated voltage. This simply doesn't work. One of the fuses might blow and may be arcing, while the

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

RESOURCES

AIM-TTi | www.aimtti.com
Keysight | www.keysight.com
Littelfuse | www.littelfuse.com
TDK-Lambda | www.tdk-lambda.com

second might still conduct a lower current. Just select and buy the appropriate fuse.

AN I^2t EXPERIMENT

Let's come back to the melting time. As explained earlier, it is linked to the current going through the fuse. But how? What is the relationship? How can you estimate this current breaking time for a given fuse? Let's think about it. You might remember this version of Ohm's law: $P = R x I^2$. That means that the power dissipated in a fuse, which acts as a resistor, is proportional to the square of the current. You might also remember that energy is a power multiplied by a time. So, Energy = $P \times t$, and therefore Energy = $R \times I^2 \times t$. And in which situation will the fuse wire melt? When the energy dissipated in the wire is enough, meaning that Energy > threshold. This means that $R \times I^2 \times t >$ threshold, or simply, assuming the resistance R is constant for a given fuse:

$I^2t >$ threshold

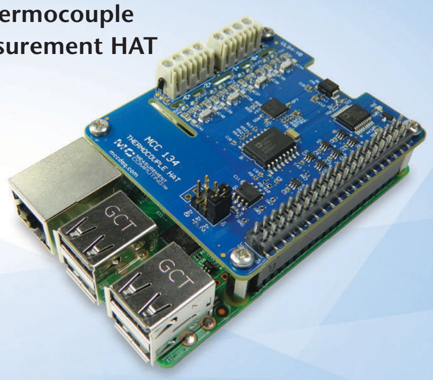
So, the melting time of a given fuse is, in fact, defined by a constant—namely the square of the current multiplied by the melting time, or I^2t. If you double the current, then the melting time theoretically will be divided by 4 (the square of 2), and so forth. Simple rule, isn't it? If you look at the datasheet of a given fuse you will find this so-called I^2t value.

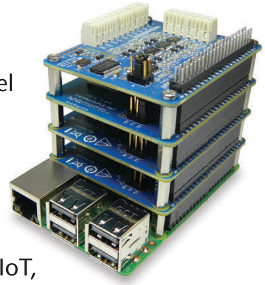
Why not experiment with this? With that in mind, I decided to destroy some 2 A, fast, 20 mm × 5 mm fuse cartridges for you. The required test setup is very simple, so you can easily reproduce it at home (**Figure 2**). You will need a lab power supply, a fuse holder, a low-value resistor and an oscilloscope connected to the resistor. And a box of fuses, of course. Put a fuse in place, set the power source to a given level, switch it on and measure both the current (proportional to the voltage measured on the scope) and the tripping duration (simply read on the scope's horizontal axis).

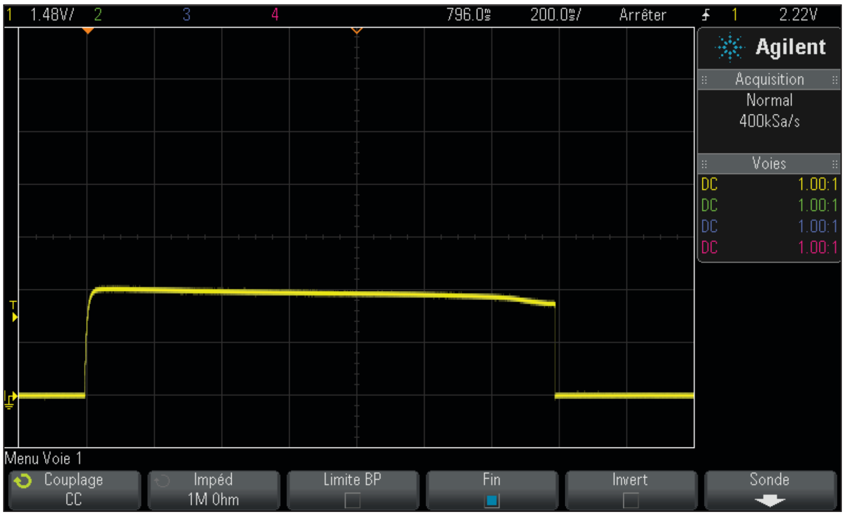
My actual experiment is shown in **Figure 3**. I used a powerful TDK-Lambda GENH20-38 power source, a DC-DC compact unit able to supply up to 38 A and 20 V. I also used a 1 Ω, 10 W power resistor, enough to get up to 20 A for short durations (even if this creates 40 W of instantaneous current, as $P= RI^2$). I measured the voltage on the 1 Ω resistor

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.

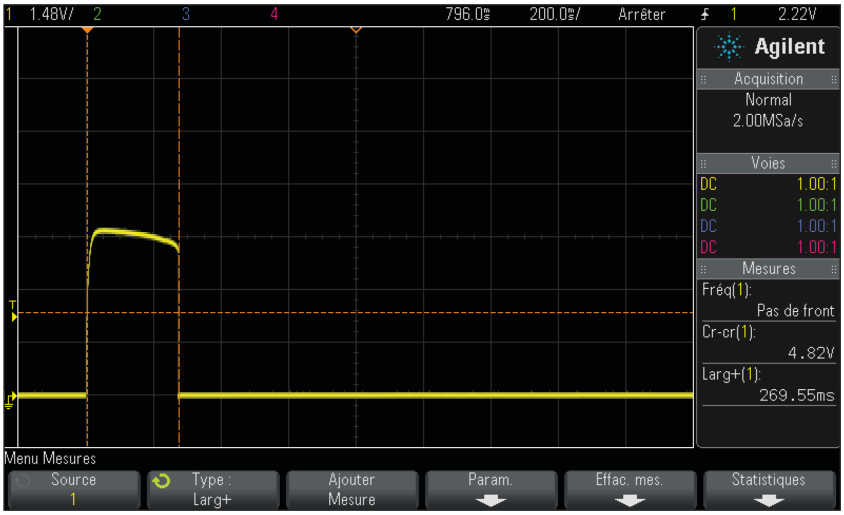






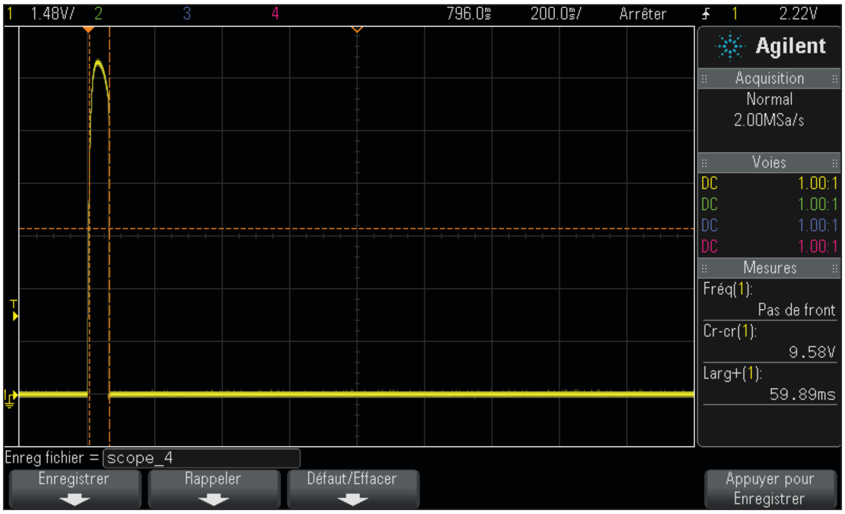
Acquisition: Normal 400kSa/s
DC 1.00:1
DC 1.00:1
DC 1.00:1
DC 1.00:1
Couplage: CC
Impéd: 1M Ohm

FIGURE 4
With 4.1 A of current, the fuse breaks in 1,380 ms. The current is decreasing a little just before breaking, meaning that the fuse is melting and its resistance is increasing.



Acquisition: Normal 2.00MSa/s
DC 1.00:1
DC 1.00:1
DC 1.00:1
DC 1.00:1
Fréq(1): Pas de front
Cr-cr(1): 4.82V
Larg+(1): 269.55ms
Source: 1
Type : Larg+

FIGURE 5
With 6.51 A of current, the fuse breaks much faster, here in 269 ms.



Acquisition: Normal 2.00MSa/s
DC 1.00:1
DC 1.00:1
DC 1.00:1
DC 1.00:1
Fréq(1): Pas de front
Cr-cr(1): 9.58V
Larg+(1): 59.89ms
Enreg fichier = scope_4

FIGURE 6
The breaking time reduces to 59.89 ms for a 12.95 A current.

with a Keysight 34461A multimeter, while the current through the wire was measured with a Keysight DSO-X 3034A oscilloscope and an AIM-TTi I-prober 520 current probe. Yes, I could have just connected the oscilloscope voltage probe to the 1 Ω resistor, but using a current probe was, well ... more fun.

THE RESULTS

What happened? Look at the oscilloscope plots reproduced in **Figure 4**, **Figure 5** and **Figure 6**. On these three plots, the current was respectively 4.1 A, 6.51 A and 12.95 A. The durations required to blow the fuses were 1.38 seconds, 0.269 seconds and 0.06 seconds, respectively. I performed the test with five different currents, set by different voltages on the power supply. My results are given in **Table 2**. This spreadsheet also gives you the calculated I^2t value—once again nothing more than the square of the current multiplied by the time required by the fuse to cut the circuit. And these results are impressively close to the theory. For all tests from 6.51 to 12.95 A, the measured I^2t value is nearly constant, close to 10 A^2-seconds.

However, for a lower current, here 4.1 A, the fuse time is far higher, more than double what was expected. Why? Probably because, as the fuse was heating slowly, this heat had enough time to dissipate through the fuse holder and ambient air, delaying the fuse melting. Interesting, isn't it? Another interesting result of this experiment is, in particular, visible on the first scope plot (Figure 4). As you see, the current is starting to decrease a little just before the cutoff. What's happening? My interpretation is that the wire was starting to become very hot or maybe to melt, and its resistance was increasing a little, reducing the current thanks to Ohm's law. If you don't agree with this interpretation, just contact me.

For your convenience, I've also graphed these current and tripping time measurements (**Figure 7**). As you will see, the measurement points—except the first one—are nearly perfectly aligned on the graph. As you will have noticed, I plotted this graph using logarithm scales. The fact that the measurement points on such a log/log scale are aligned is a direct consequence of the constant I^2t property. (Note: the log of a product is the sum of the logs.)

WRAPPING UP

So, here we are. My only goal was to show you that things may look strange, but are easily explainable by simple laws of physics. Fuses are not magical, and shouldn't be considered as more than what they are: simple wires that melt under proper conditions. By the way, if

Rated 2 A		
Current (A)	**Fuse time (ms)**	**I^2xt**
4.10	1380	23.20
6.51	269	11.41
8.84	132	10.31
10.93	84	10.04
12.95	60	10.06

TABLE 2
This table summarizes the measured tripping time for various currents. The last column shows the calculated I^2t value.

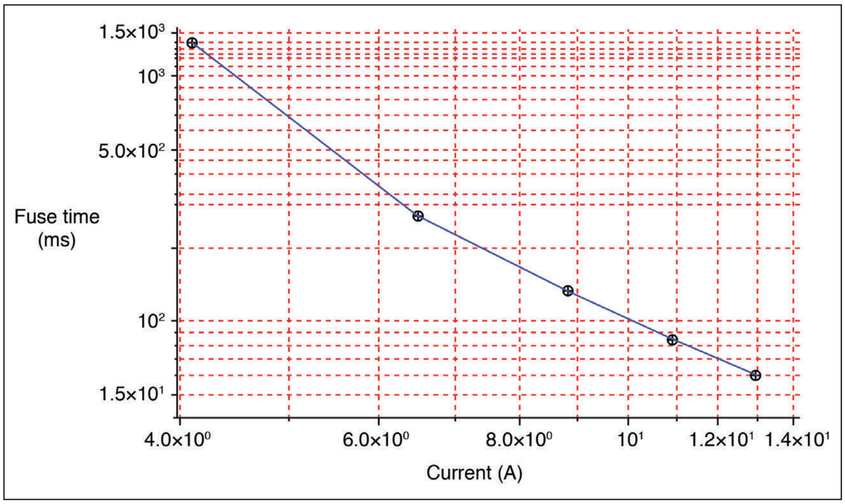


FIGURE 7
When the tripping time and current are plotted on a log/log scale, the result is a straight line except for low currents.

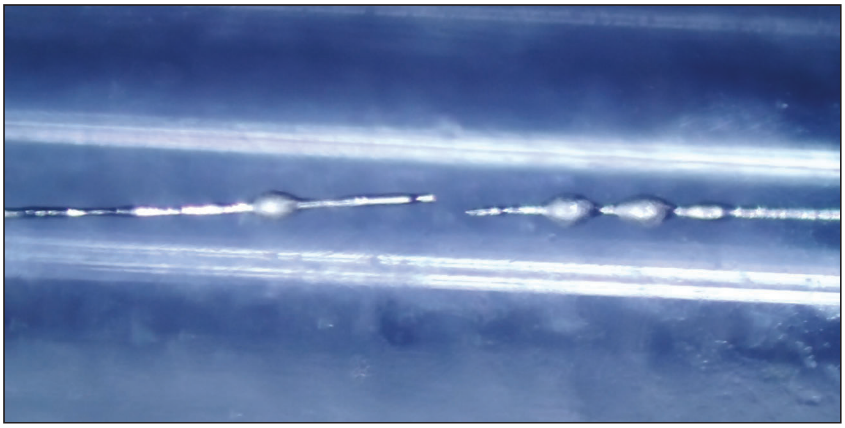
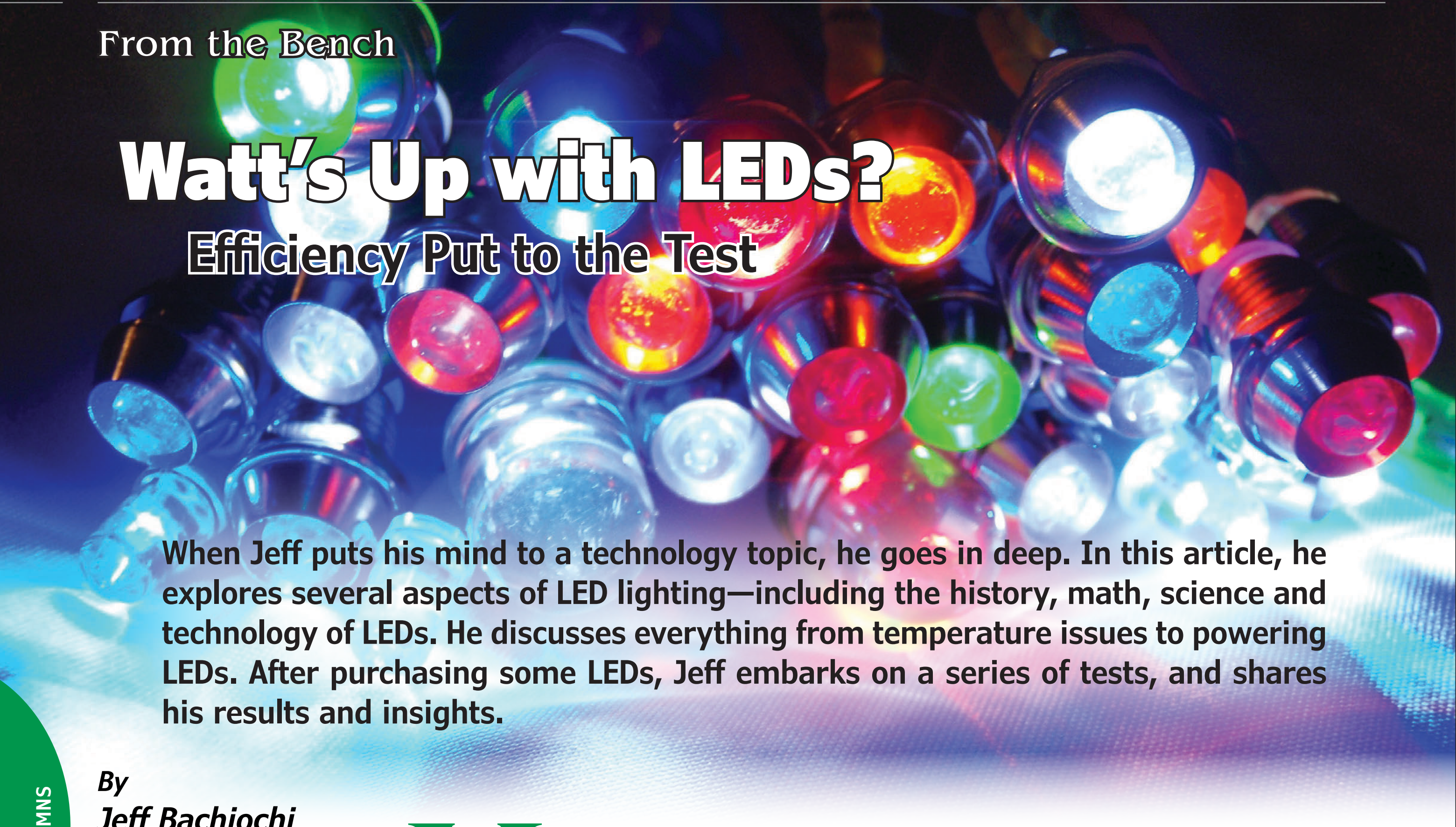


FIGURE 8
A blown fuse. Pretty isn't it?

you have any doubt, just find a blown fuse and look at the wire under a microscope. You will easily see melted metal, and the melting depends on the current. I did the experiment for you, and one of my best pictures is shown in **Figure 8** for your pleasure.

As usual, I've only scratched the surface of the subject. In particular, I didn't talk about the different kinds of resetable fuses. That is another interesting but long topic. Maybe in another article? In the meantime, remember that the best way to learn is to practice. Fuses are inexpensive, so just do some tests by yourself! And have fun!

From the Bench



Watt's Up with LEDs?

Efficiency Put to the Test

When Jeff puts his mind to a technology topic, he goes in deep. In this article, he explores several aspects of LED lighting—including the history, math, science and technology of LEDs. He discusses everything from temperature issues to powering LEDs. After purchasing some LEDs, Jeff embarks on a series of tests, and shares his results and insights.

By
Jeff Bachiochi

My first LED required 10 mA and glowed red. The body was tinted red, and red was the only color available. But it glowed. Imagine a solid-state device emitting light! In college, I turned in my slide rule for a TI SR-10 calculator. It had a multi-digit red LED display (**Figure 1**). Red LEDs were hot. Texas Instruments (TI) even manufactured some of the first LED watches (**Figure 2**).

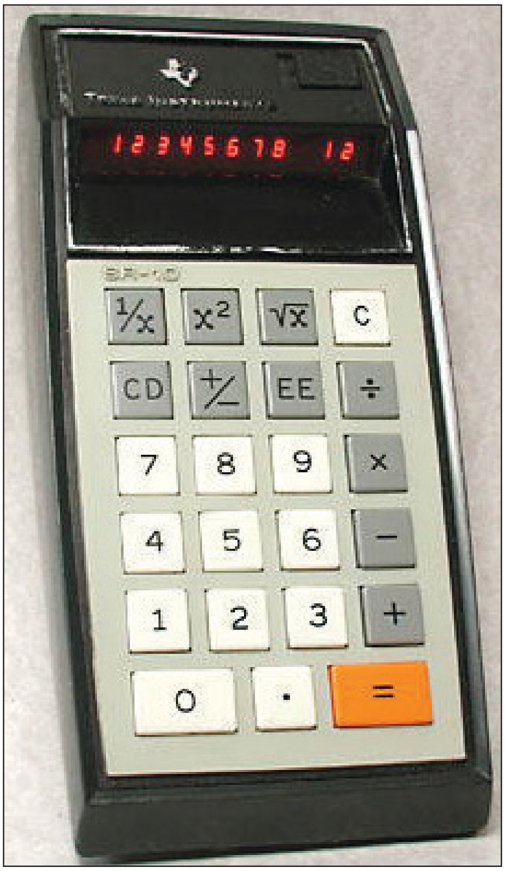


FIGURE 1
My first Texas Instrument calculator, the SR-10. Note the tiny red LED numerals showing eight significant digits plus two-digit exponent!

The first evidence of electroluminescence (emission of photons) was discovered in the early 1900s by a British experimenter working on a radio project that involved a crystal of silicon carbide and a cat's-whisker detector (diode). No useful purpose was found for this phenomenon until decades later by RCA engineers, yet TI drew the first patent for an infrared diode in the early 1960s. Turns out the emissions were in the infrared spectrum due to the material, gallium arsenide, that was being used.

A diode is a device made up of materials that forms a "p-n" (positive-negative) junction. The "n" material has been doped with impurities that contain atoms with extra "free" electrons, while the "p" material's impurities have extra holes or a lack of electrons. At the junction of the two, there is a tendency for the free electrons to migrate to the "p" side to fill in the holes. During this process, the electrons end up changing their state, that is, moving from an outer orbit with higher potential energy to a lower orbit requiring less potential energy. During this state change, energy is given up in the form of a photon. The greater the energy release, the higher the frequency of the light photon given off, thus changing the color. As implied earlier, the frequency of the photon is dependent on the materials used.

The two main types of LEDs presently

used for lighting systems are: 1) aluminum gallium indium phosphide alloys for red, orange and yellow LEDs; and 2) indium gallium nitride alloys for green, blue and white LEDs. Today, LEDs can produce a full spectrum of colors by combining RGB and sometimes W LEDs into the same package. If we vanquish mood lighting (RGB) from the mix, we end up with white. White is usually considered as the inclusion of all visible colors. It brings with it a variety of colors or temperatures—from warm (yellowish) to daylight (blueish). A warmer yellowish tint is close to that of the incandescent light, while the bluer tint approaches natural sunlight. **Figure 3** illustrates how temperature can affect ambiance.

LIGHTING INDUSTRY

What began as an industry that revolved around incandescent bulbs has expanded into specialty technologies with halogen and high-intensity discharge (HID), and into energy efficient engineering with fluorescent and CFL bulbs. It's also jumped onto the even more efficient bandwagon of LEDs. When it comes to the amount of energy (or electrical power) that a light bulb uses, the lower the watts, the lower the electric bill.

That's it in a nutshell, isn't it? The terms "watts" in this industry began by describing the energy usage of the incandescent bulb. We're used to relating the brightness of an incandescent bulb by its wattage—higher wattage, higher energy, higher brightness. But this is not the best way to describe brightness. Although the brightness of an incandescent bulb is proportional to its wattage, it is all based on the efficiency of the incandescent bulb in converting power into brightness. When we bring other technologies into the mix, each brings with it its own efficiency. Therefore, we need a different way of comparing the "brightness" of the light emitted.

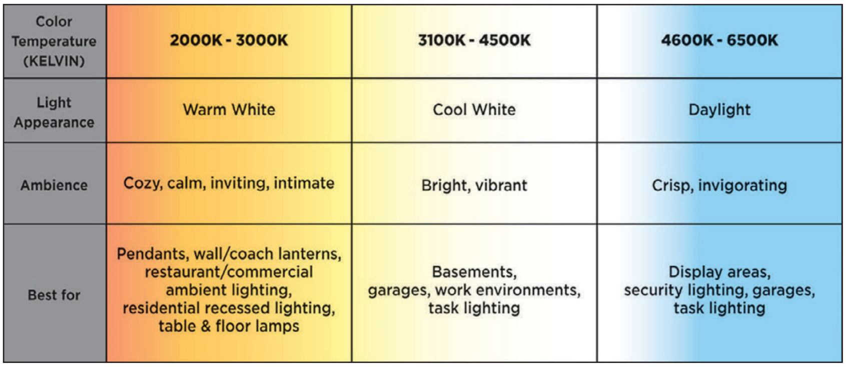
In early research, "candle power" or "candela" was defined as a standard measure of light intensity seen 1' away from a wax candle. While this standard was eventually redefined by other means, it's still used today. Enter the "lumen." The lumen is a measure of the total quantity of visible light emitted by a source. It is weighted according to a model of the human eye's sensitivity to various wavelengths. "Lux" is a term used for lumens collected per square meter. Both lux and candela describe the amount of light falling on a particular area—1 candela (cd) = 10.764 lux, keeping in mind that the candela uses square feet and lux uses square meters.

Given that the usefulness of a light source is based on our visual perception, it makes sense to base all comparisons on the lumen. The watt is more appropriately used as an indication of efficiency. Take a look at these units and their relation to technology vs. brightness in **Figure 4**. No wonder LEDs are being touted as the greenest of the green!

IT'S HOT

When I am working with heatsinks, I often relate the heat given off from a Christmas tree bulb (7.5 W) as a maximum temperature for my heatsinks. A 100 W bulb is hot. If you've ever unscrewed an incandescent bulb while it's still on, you know what I mean—your fingertips are easily burned by the heat they emit. When using LEDs in my projects, one word I would never use for them is "hot." They are generally rated in millicandelas (mcd), or thousandths of the power of one candle. These are generally run with only 10 mA, or 1 mA for some narrow-angle LEDs.

FIGURE 2
The time or date with just the push of a button, right on your wrist! Who'd a thunk that today we'd have the world on our wrist?

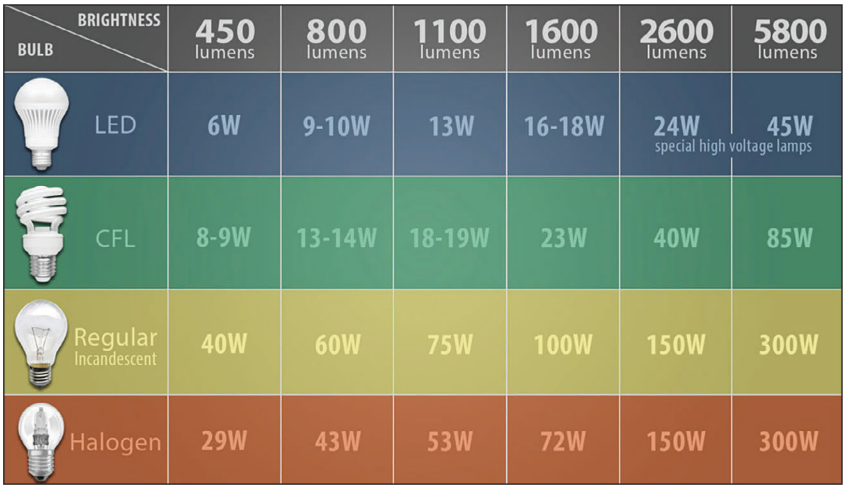


Color Temperature (KELVIN)	2000K - 3000K	3100K - 4500K	4600K - 6500K
Light Appearance	Warm White	Cool White	Daylight
Ambience	Cozy, calm, inviting, intimate	Bright, vibrant	Crisp, invigorating
Best for	Pendants, wall/coach lanterns, restaurant/commercial ambient lighting, residential recessed lighting, table & floor lamps	Basements, garages, work environments, task lighting	Display areas, security lighting, garages, task lighting

FIGURE 3
There really are shades of white. The color temperature has a lot to do with how we feel. And that's not even counting the individual RGB colors.

Consider a typical LED indicator that has a forward drop of about 3 V. If you run 10 mA of current through it, the power produced in the junction would be 30 mW. The specs for this device call out a maximum of 68 mW for the device. So, you can see that any current over 23 mA will exceed the acceptable power dissipation. And therein lies the issue. Unlike incandescent filaments, an LED cannot dissipate the high power. It becomes an issue of being able to get rid of heat.

If you've worked with LEDs you know you can trick them into producing the same average heat by pulsing them ON and OFF at higher current levels. A 10% duty cycle at 100 mA should produce on average the same heat as a 100% duty cycle at 10 mA. If the frequency of the PWM is higher than our persistence of vision (about 30 Hz), then we see it as always ON. While average power is within limits, we must be mindful of instantaneous power. The efficiency with which we can remove heat from the junction is limited by its design. By building the junction upon an efficient heat sink material, LEDs are now able to exceed 3 W, based on providing adequate heatsinking.



BRIGHTNESS / BULB	450 lumens	800 lumens	1100 lumens	1600 lumens	2600 lumens	5800 lumens
LED	6W	9-10W	13W	16-18W	24W special high voltage lamps	45W
CFL	8-9W	13-14W	18-19W	23W	40W	85W
Regular Incandescent	40W	60W	75W	100W	150W	300W
Halogen	29W	43W	53W	72W	150W	300W

FIGURE 4
Our newer technologies bring with them an increased efficiency—more lumens for less watts.

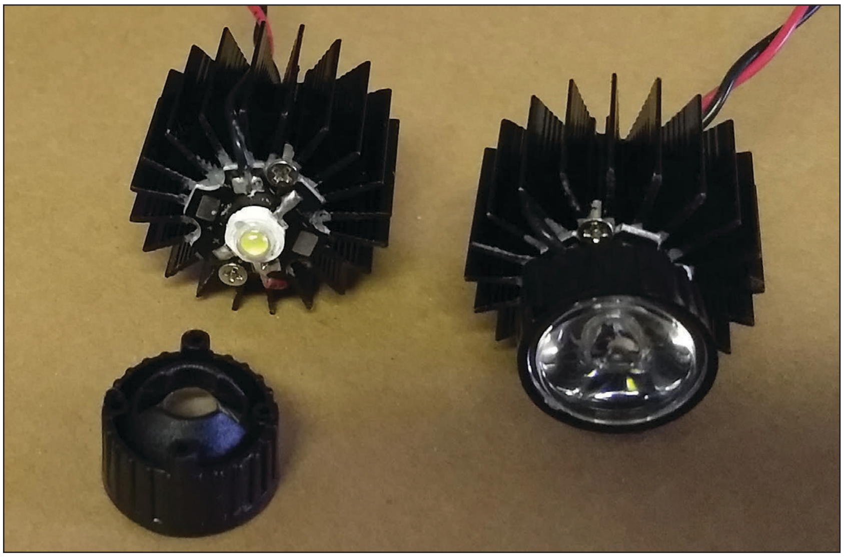


FIGURE 5
Here are the 3 W LEDs I used in some of my tests. The star-shaped plate acts as a transitional heatsink to the surface-mounted LED. An additional heatsink is still required to keep the p-n junction temperature low. The LED has a 120-degree beam angle. To use this as a spotlight, a lens is required to redirect the photons into a more focused beam.

I purchased a number of 3 W white LEDs from Addicore.com. These are already mounted on a "star" aluminum heatsink. The "star" is a way to make both electrical connection to the LED and thermal connection to an appropriate heatsink. A mating 3 W heatsink is also available (see **Figure 5** for the completed module). Note: The heatsinks cost more than the LED, regardless of where you buy them. Another advantage of using Addicore is that they offer lenses for these LEDs. More on that later. For now, let's talk a bit about heatsinks.

THROW ME AN LED SAVER!

The best way to kill an LED is by allowing it to heat up. An LED's output is specified at room temperature. Its output decreases with increasing temperature. Typically, its life is rated with a maximum junction temperature of 125°C. As the ambient temperature of its environment goes up above 60°C, it must be derated, or run at a reduced current, because it just won't be able to get rid of its heat no matter how big the heat sink.

Assuming the ambient temperature is cooler than the LED junction, heat will flow from the junction to the air. Thermal resistance is a measure of this, calculated by dividing the temperature difference (temperature), by the heat flow (power) between the two points. The heat must flow through all materials in contact with the junction. Each material the heat flows through offers a different resistance to the flow. Thermal resistance is the total of all these and we show it as °C/W.

While this can be a complicated calculation, you might want to check out the Luxeon [1] heatsink calculator. If you plug in values like 3 W for the LED, 125°C for the maximum junction temperature, and 25°C as the typical ambient temperature, you'll get something like a 12.9°C/W heatsink requirement. I encourage you to do this, because you can then change things like "lowering the junction temperature or raising the ambient temperature" and see how the thermal resistance is affected. This will give you a feel for how this is all connected. As you change the parameters and the thermal resistance required goes down, you'll want to know about the physical size of the heatsink required. **Table 1** shows a list of some sinks,

Part#	Diameter	Height	°C/W	Surface Area
DUALLED-13050	5.1"	2"	1.0533	? in^2
SA-LED-176E	2.8"	3"	2.49	108 in^2
SA-LED-113E	2.8"	0.5"	5.33	18 in^2
SV-LED-314E	0.8"	0.5"	9	5.3 in^2

TABLE 1
The addition of heatsink is required to radiate more heat, and that causes the unit's physical size increase. Although the physical size grows, it's really all about how much surface area can be provided to increase radiation. Listed here are some heatsinks, their sizes and thermal resistance specs.

their sizes and thermal resistance specs.

This discussion deals strictly with cooling by unobstructed air flow and will not get into the necessity or ability to lower thermal resistance by other methods such as fans or water cooling. If you Google for "LED heatsinks," you find many advertised. Note that no specifications are given for several of the sinks. Reputable manufacturers will list specs for each of their products, so you can calculate proper sizing. No specs came with the heat sinks from Addicore. However, I can estimate their surface area with some quick calculations, as follows:

Fin size = 0.4 × 0.9
Fin area = 0.4 × 0.9 = 0.36 in^2
Fin count = 20

Total fin area = 0.36 × 20 = 7.2 in^2

Using Luxeon's heatsink calculator, I found the minimum size heatsink requirement was 12.9°C/W. Comparing the estimated surface area calculated for the Addicore heatsink of 7.2 in^2 to the list in **Table 1** we can estimate its thermal resistance of somewhere between 9°C/W and 5.33°C/W. This is certainly within the minimum requirements. Assuming we can take care of the heat, now let's look at powering an LED.

VOLTAGE OR CURRENT

What should be the most obvious decision in powering an LED, or multiple LEDs, is to use a supply that can provide the total wattage of all your LEDs. It's a good idea to include a 20% safety margin in these computations. If we want to power a 3 W LED, for example, the supply should be at least 4 W. If you look at the specifications for any LED, you'll see that light output increases with the current through it, but so does the voltage drop. Also note that, as the junction temperature goes up, the light output is reduced. So, there is a sweet spot where all three are at their maximum values. For Addicore's high-power LEDs, the maximum specifications are 3 W and 750 mA steady-state current. Well, at 750 mA, the voltage drop is 3.43 V, for a power of 0.75 × 3.43 or 2.6 W. We could squeak out a bit more lumens by modulating the supply at up to 1.5 A, as long as we don't exceed 3 W average power (according to the specs), but we aren't going to delve into that here.

If our power supply were a regulated 5 V, we could limit the current through the LED to 750 mA by adding a series resistor to the LED. If we can drop 5 V minus 3.43 V, or 1.57 V across the resistor, then by Ohms law, the current will be limited to 750 mA. The resistor value must therefore be 1.57V/0.750A or 2 Ω. With 750 mA current through 2 Ω, this must withstand 1.125 W. Using a regulated 12 V supply, this resistor would need to be 12 V minus 3.43 V = 8.57 V and 8.57V/0.750A or 12 Ω at 7 W. Notice how the wasted heat (in the series resistor) goes up with the supply voltage. If we lowered the supply voltage to 3.43 V, no series resistor would be needed!

In reality, we could use either a constant voltage or a constant current scheme to protect our LED. The direction you choose to go will certainly depend on the number of LEDs you want to power, and whether they will be in series or parallel. With LEDs in series, you might want to use a constant current supply, since the same current is passing through each LED. This requires no series resistor, but the voltage of the supply must be at least 3 V above the total voltage

ABOUT THE AUTHOR

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

jeff.bachiochi@imaginethatnow.com or at:

www.imaginethatnow.com.

Additional materials from the author are available at:
www.circuitcellar.com/article-materials
Reference [1] as marked in the article can be found there.

RESOURCES

Addicore | www.addicore.com

On Semiconductor | www.onsemi.com

Texas Instruments | www.ti.com

TABLE 2
My first test is with a linear constant current regulator, an LM317 and a series resister that sets the current. Here are the results.

	Voltage	Current	Power	Efficiency
Battery	12.2 V	0.75 A	9.15 W	27%
Load	3.39 V	0.75 V	2.54 W	

TABLE 3
For the next test, I swapped in a switching regulator for the LM317. The switching circuitry uses a MC34063A, step-up/down/inverting switching regulator. The results are shown in this table.

	Voltage	Current	Power	Efficiency
Battery	12.4 V	0.22 A	2.78 W	87%
Load	3.32 V	0.73 V	2.42 W	

of all the LED drops. With LEDs in parallel, you might choose a constant voltage supply of 5 V. Each LED should have its own series resistor of 2 Ω at 2 W. The maximum number of LEDs is thereby limited by the current of the supply.

THE BASEMENT TESTS

The first set of tests use a 3 W LED with heat sink. The setup includes a 12 V Gel Cell battery, a 3 W LED, twin power meters and a regulator. I have one power meter between the battery and regulator, and the second meter between the regulator and LED. This allows me to record the power used and the power required. These power values give an effective efficiency of the regulator circuitry.

The first test is with a linear constant current regulator, an LM317 voltage regulator and series resister that sets the current. The circuit limits the current to a maximum of 1 A with a series resistor and a 100 Ω potentiometer, so this can be adjusted down to about 10 mA. **Table 2** shows the results. LM317 voltage regulators are available from Texas Instruments and On Semiconductor.

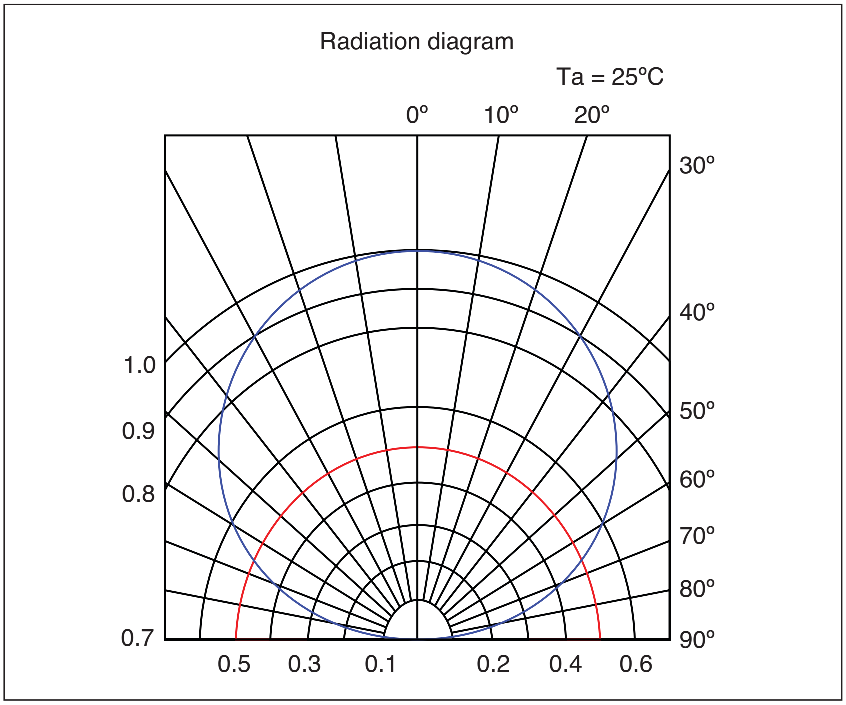


FIGURE 6
Radiation diagram showing how the luminous energy is dispersed. The LED's junction is at the center bottom pointing up (0 degrees). Angles mark the rays originating from the LED. Concentric circles around the LED mark off luminescence values (fractions of the reference 1.0). The circular pattern is a shadow of the LED's illumination. Note the reference luminous intensity 1.0 of that pattern intersects at 0 degrees. Also note its intersection with 0.5 (half of the reference intensity) and 65 degrees. The angle where the luminous intensity drops to 50% is designated as half the beam width. Because it's the same on either side, the total beam width is 2×65 degrees = 130 degrees.

Next, I swapped out a switching regulator for the LM317. The switching circuitry uses a TI MC34063A, step-up/down/inverting switching regulator. Components set the current to ~700 mA. The results are shown in **Table 3**. A switching circuit is a bit more costly, but its efficiency is worth the cost. The efficiency means a lack of wasted heat, and since we've got a heat issue with the LED, it's double payback.

The next tests deal with lux output versus beam width. Because LEDs are more of a point source, they usually have a specification that deals with beam width. The combination of junction architecture and plastic lens defines an LED's radiation pattern. This is normally conically shaped and denoted in degrees. LEDs can be classified by beam width: a spot (4-19 degrees), a flood (20-35 degrees), a wide flood (36-49 degrees) and a very wide flood (50- 120+ degrees). **Figure 6** is a diagram of the pattern of the 3 W LED I am using.

The LED is physically located at the center bottom of the diagram, pointing toward the top. The lines circling the LED mark fixed distances 0.0 to 1.0 from the LED where the luminous flux is measured. 1.0 is a normalized luminous output flux and its associated forward current. I've chosen 10' as my measurement distance. The circular area from the LED and centered on the 0-degree line (up) denotes the area that is lit by the LED. Note that its extremes—to the right and left of 0 degrees—are bounded by the maximum and minimum deviations from 0 degrees, where the luminous flux fades to 50% (0.5) or half of the amount measured at the 10' point (at 0 degrees). This max (or min) angle is 65 degrees, for a total angle of 2×65 degrees = 135 degrees. This angle places this LED in the "very wide flood" category.

When using power from either of the

aforementioned supplies, I used an app [1] on my cell phone to measure the LED's luminous flux in lux or candela at 0 degrees, 10' from the LED (hypotenuse). Then I moved along the 10' circumference away from 0 degrees to a point where the lux fell to 50%. A measurement of the distance to the 0 line is the opposite side of a triangle, which, along with the hypotenuse, can be used to find the inclusive angle. This angle deviation is half the beam angle. **Table 4** shows the measurements for the 3 W LED.

Now that we have a baseline of the 3 W LED, we can add lenses to the LED and see how the beam angle and flux output are affected. I have 90-, 60-, 30- and 15-degree lenses from Addicore for this LED. Let's look at the measurements shown in **Table 5**. It looks like the lenses do, in fact, reduce the width of the beam. I was a bit surprised at the efficiency loss of these plastic lenses as the beam width was narrowed and the lux output of 3 W LED increased. Does this spell doom for using these for larger requirements? Well... yes and no.

HEADLIGHT

Many standard auto headlights use a replaceable Halogen H4 bulb in their glass headlamps. I removed the headlamp from my Honda Shadow motorcycle and set it up on my test bench. Using the same procedure that I used previously, I was able to measure its light output. The output from the 1/2" filament depends on the mirrored glass to focus the light output in a narrow beam. These results are shown in **Table 6**.

LED replacements are available in the H4 configuration for plug-and-play connection (**Figure 7**). I purchased a few of these to play

Lux	Hypotenuse length	Opposite length	Arcsine O/H	Degrees 2×Arcsin(O/H)
41	10'	0'	0	0 degrees
21	10'	9'	0.9	128 degrees

TABLE 4
Shown are the measurements for the 3 W LED.

90-degree lens				
Lux	**Hypotenuse length**	**Opposite length**	**Arcsine O/H**	**Degrees 2×Arcsin(O/H)**
44	10'	0'	0	0 degrees
21	10'	7.5'	0.75	98 degrees
60-degree lens				
Lux	**Hypotenuse length**	**Opposite length**	**Arcsine O/H**	**Degrees 2×Arcsin(O/H)**
55	10'	0'	0	0 degrees
25	10'	4.5'	0.45	54 degrees
30-degree lens				
Lux	**Hypotenuse length**	**Opposite length**	**Arcsine O/H**	**Degrees 2×Arcsin(O/H)**
77	10'	0'	0	0 degrees
35	10'	3.3'	0.33	38 degrees
15-degree lens				
Lux	**Hypotenuse length**	**Opposite length**	**Arcsine O/H**	**Degrees 2×Arcsin(O/H)**
90	10'	0'	0	0 degrees
45	10'	1.5'	0.15	18 degrees

TABLE 5
I tested 90-, 60-, 30- and 15-degree lenses from Addicore for the 3 W LED. Here are the measurements.

Voltage	Current	Power	Lux	
11.7 V	4.94A	57.8 W	4520	
Lux	**Hypotenuse length**	**Opposite length**	**Arcsine O/H**	**Degrees 2×Arcsin(O/H)**
2200	10'	0.5'	0.05	6 degrees

TABLE 6
Next, I removed the headlamp from my Honda Shadow motorcycle and set it up on my test bench. The results are shown here.

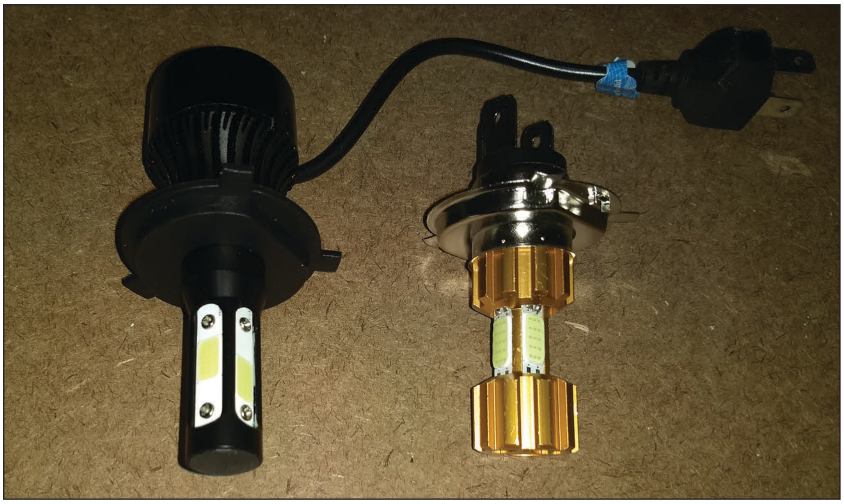


FIGURE 7
H4 bulbs use a tri-tabbed circular mount. You can see about 3×5=15 LEDs mounted on each COB module. The golden H4 uses tri-mounted modules in a passive heat sink arrangement. The black H4 has quad-mounted modules and requires active heat sinking, with a rear mounted micro fan. H4 bulbs depend on the headlamp's interior reflector to redirect their perpendicular output.

with. I like this arrangement, because they fit into the existing glass lamp and therefore will use the same reflector in the tests. The tri-sided replacement uses natural convection on the heatsink, which extends outside of the semi-sealed glass lamp. The results are shown in **Table 7**. A similar H4 bulb uses a miniature fan to actively cool the heatsink. Its configuration is four-sided and gave the results shown in **Table 8**.

Had I known what level of output to expect from the Halogen bulb, I would have paid more attention to LED bulb ratings. Because many available bulbs come in pairs (you need two for most cars), they are advertised by the total output of both bulbs. You have to look closely to find the actual specs/bulb. Most of the higher wattage bulbs have integrated fans to actively cool the heatsinks. I've found replacement bulbs up to 100 W.

The last head lamp in this test is a complete sealed replacement for the Shadow (**Figure 8**). This does not use a replaceable H4 bulb, but rather a number of discrete LEDs (4-5-4)—much like the 3 W LEDs I purchased. Note that while the original Halogen headlight had a higher lux at 10′, it had a very narrow beam width covering only 6 degrees. This one, which has a lower output at 10′, has a much greater beam width. Test results are shown in **Table 9**.

MY 2 CENTS

When I see the words "high-power LEDs," I now realize this is in reference to those we use as indicators in many of our microcontroller projects. They run at few milliamps, and there is no measurable heat produced. Once we start pumping amps through them, heat becomes a large issue. Carefully designed mounting with adequate surface area is necessary to keep these little ones cool, so they don't just burn up. The heat also shortens their life, and that is their biggest bragging right—so be careful.

Incandescent bulbs have 360-degree coverage, whereas LEDs cover about 120 degrees. So, as a ceiling light, they're fine. But in a table lamp, you'll need multiple LEDs to cover 360 degrees. If you wish to

TABLE 7
The tri-sided LED replacement uses natural convection on the heatsink. The measurement results are shown here.

Rated 2000 lumens				
Voltage	**Current**	**Power**	**Lux**	
12.0 V	1.71 A	20 W	1508	
Lux	**Hypotenuse length**	**Opposite length**	**Arcsine O/H**	**Degrees 2×Arcsin(O/H)**
750	10'	0.5'	0.05	6 degrees

TABLE 8
A similar H4 bulb uses a miniature fan to actively cool the heatsink. Its configuration is four-sided and gave the results shown here.

Rated 2500 lumens				
Voltage	**Current**	**Power**	**Lux**	
12.0 V	1.84 A	22 W	2000	
Lux	**Hypotenuse length**	**Opposite length**	**Arcsine O/H**	**Degrees 2×Arcsin(O/H)**
1000	10'	0.5'	0.05	6 degrees

TABLE 9
The last head lamp to be tested is a complete sealed replacement (Figure 8) for the Shadow headlamp. Test results are shown here.

Rated 5000 lumens				
Voltage	**Current**	**Power**	**Lux**	
11.9 V	2.72 A	32.4 W	3000	
Lux	**Hypotenuse length**	**Opposite length**	**Arcsine O/H**	**Degrees 2×Arcsin(O/H)**
1500	10'	2'	0.2	23 degrees

funnel an LED's lumens into a narrower beam, it requires special optics to make that conversion. My last test subject lamp did a pretty good job at lensing discrete LEDs for a drop-in replacement of the Halogen headlight. That's one approach. The other approach, H4 replacements, uses the perfectly designed reflector for the (fundamentally) point-source filament and tries to replicate said filament. Placement and size are essential to make use of the reflector properties. This puts a burden on the design, because the heat must exit through the body to get out of the head lamp. It's no wonder small fans are necessary to help remove the heat.

All in all, I'm not thrilled with what I see. Yeah, we've moved on from CFLs (containing mercury) to (greener?) LEDs. But, let's be truthful here. There isn't anything more simple than a glowing filament, even if it took hundreds of experiments to find the ideal material. You can see from Figure 3 that LEDs are the most efficient so far in turning power into light. While there seems to be a little extra circuitry involved in getting the LED to conform to the energy source available, with the right design you can forget about ever having to replace an LED. You can expect 10 years of continuous service.

We're at the point now where we are able to control our indoor environment more closely. We can control the color of each light's emission, and this has an effect on the way we feel. Imagine eliminating depression just by tuning the color of your environment. If it were only that easy.

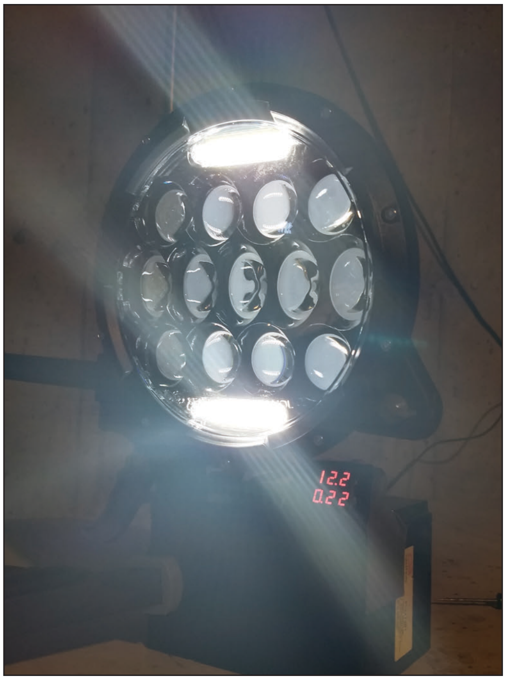


FIGURE 8
This sealed replacement headlamp uses individual LEDs each with its own lens to focus the light emission into a 23-degree high beam and a 46-degree low beam. Note it also incorporates running lights, twin rows of LEDs without lenses.

The Consummate Engineer

Energy Monitoring (Part 3)

Natural Gas and More

By
George Novacek

This is the final installment of George's energy monitoring article series. He discussed the solar power supply in Part 1 and the utility power data acquisition in Part 2. In Part 3, he wraps up the series by looking at the remaining modules that comprise his home energy monitoring setup, including the sensors, the natural gas monitor and the real-time clock.

Once again, I purchased readily available modules with the functions I needed. For the temperature and relative humidity (RH) sensing, I selected an Adafruit Si7021, ID 3251 break-out board based on the Si7006-A20 chip from Silicon Labs. The Si7006-A20 is a monolithic humidity sensor integrating humidity and temperature sensor elements. The humidity sensor relies on the change of electrical resistance of a conductive K-polymer dielectric material. Resistivity-based humidity sensors are affected by the temperature and, therefore, a temperature sensor is included in the package. This is an added benefit of the Si7006-A20, which you can use as a humidity as well as a temperature sensor. The Si7006-A20 is a CMOS IC and its block diagram is illustrated in **Figure 1**.

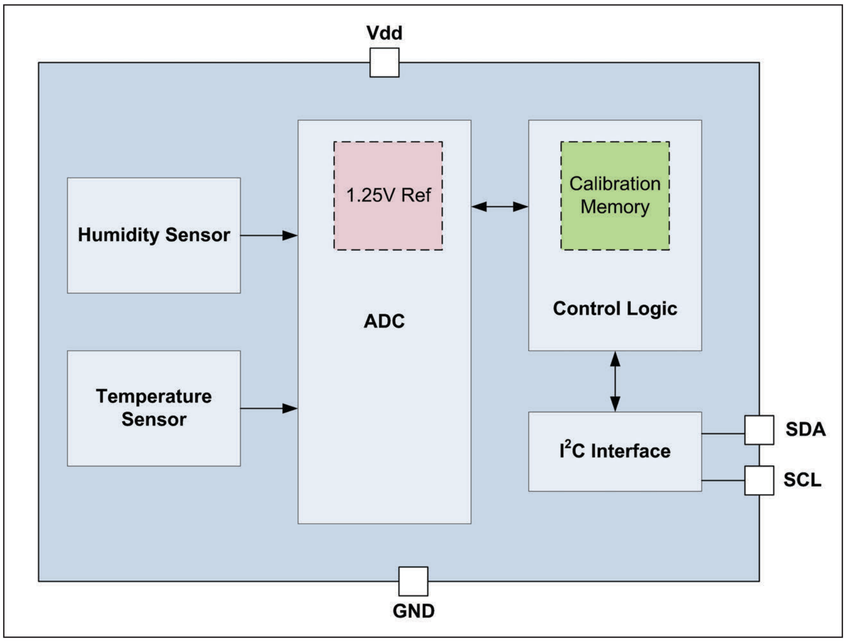


FIGURE 1
Si7006-A20 block diagram

Because the Si7006-A20's is a CMOS IC, its standby power consumption is quite low—typically 0.06 µA to 0.6 µA depending on its setup. In the course of temperature and RH conversion the current draw rises up to 90 µA to 180 µA, depending on the data being converted. Conversion time is approximately 4.5 ms. The Si7006-A20 sensor module can be used through the temperature range of -40°C to +125°C (-40°F to +257°F).

Based on my tests using telephone quad wire, the Si7021's I^{2}C interface works well up to about 1 m (around 40") in length. This is sufficient for my required 60 cm (around 24") wire run. For a longer run you will need to use a bus extender/booster such as the Texas Instruments (TI) P82B715 or rethink the interface completely. P82B715 draws 60 mA from a 5 V supply, increasing significantly the demands on your solar power supply.

DATA LOGGING

For current time, date and data logging I used the Arduino Data Logger Shield, available from Adafruit Industries, product ID 1141 (**Figure 2**). Similar shields can be obtained from other sources, such as the WIG-12772 Logomatic V2-Serial SD Datalogger from SparkFun Electronics, Universal Solder and others. The logged data is stored on a regular SD card.

An integral part of the logger shield, talking with the microcontroller via I^{2}C bus, is a Real

Time Clock (RTC). When its backup battery is inserted and the unit first powered-up the RTC gets synchronized with the Internet clock. The Data Logger library does not provide for the standard and daylight saving time changes, but I prefer to date-stamp the log at standard time anyway. Removal of the battery and a temporary power-down causes the clock to stop and reset upon reinsertion of the battery and its subsequent power-up.

In addition to being the source of the current date and time, the RTC generates 1 second hardware interrupts on its SQW/OUT pin. The pin needs a 10 kΩ pull-up resistor as the Arduino's internal pull-up seems too large for reliable pin toggling.

The interrupt drives the execution of all the input and output functions. This is when average values are calculated and, if the LCD is enabled, displayed. Once every hour on the hour, a comprehensive data package is written to the SD log. Then, every once in a while, I import the log into an Excel spread sheet for graphical analysis.

The LCD display backlighting draws a fair bit of power, so to keep the power consumption low, I enable the display manually by depressing the appropriate push button. It then cycles through values of the outside temperature, relative humidity, current flow of each phase, natural gas consumption and the current time and date, displaying each for five seconds. The display runs as long as the push button continues to be depressed. The different variables are shown with different background color schemes. This is not a necessary design requirement, but my LCD module does have this capability, so why not use it?

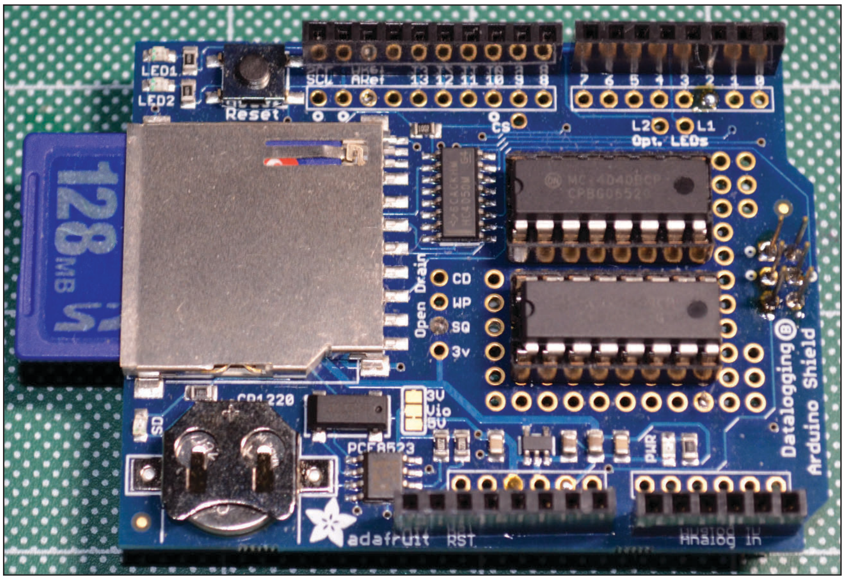


FIGURE 2
RTS and data logger shield

GAS USAGE MONITORING

For a realistic view of my household energy consumption, I also wanted to monitor my natural gas usage. We have a gas furnace, a gas water heater, a gas fireplace, a gas clothes dryer and a gas cooking stove top. Measuring the gas flow, however, proved to be a challenge. I haven't found a reasonably priced small gas flow gauge with an electrical output. Worse, all such devices—assuming they can be obtained—would require installation into the gas line by a properly trained and licensed technician. And that entails a large cost. Fortunately, an indirect approach by monitoring secondary electrical signals as shown in **Figure 3**, became a viable solution.

Aside from my stove top, my natural gas appliances all use electrical signals for their operation. The water heater has a fan

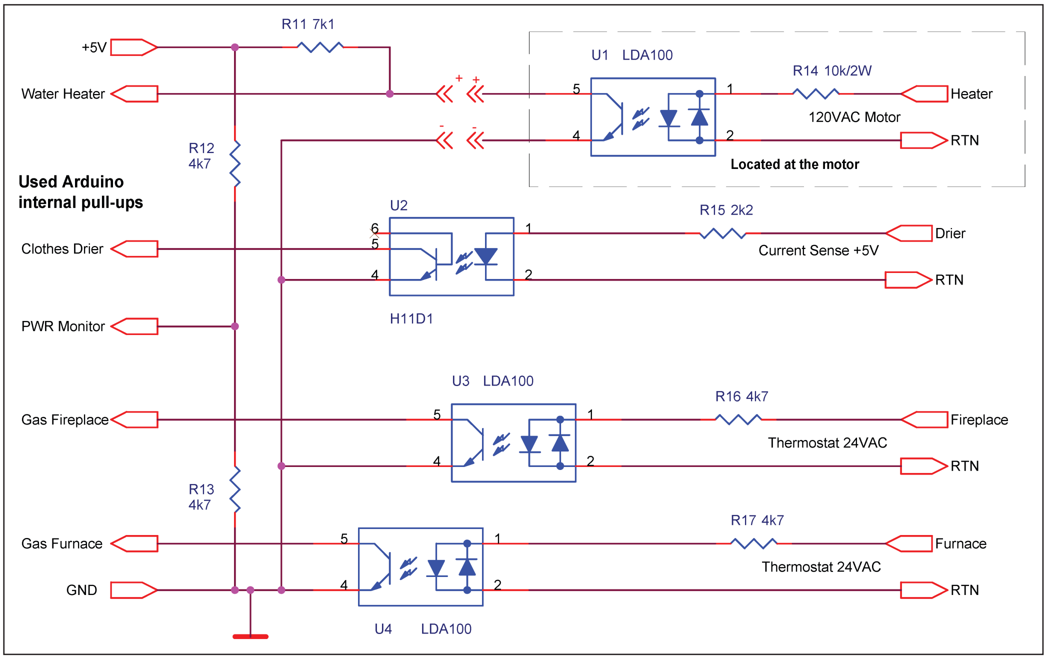


FIGURE 3
Natural gas consumption monitoring

installed in its exhaust, running whenever the heater should be on, or else the safety logic prevents it from being ignited. I attached a bi-directional opto coupler, U1, parallel to the 120 VAC motor terminals. When the fan runs, an active low signal pulls down one Arduino input with its internal pull-up enabled. For electrical safety, the opto coupler is located close to the motor and the interface to the monitor is low voltage.

In one of my previous projects, I monitored the operation of a gas-fired clothes dryer by installing a current transformer on its 120 VAC power line to the motor. The current transformer output is processed by another embedded controller, but there is a 5 VDC output signal available when the dryer is running. The U2 opto coupler is a Darlington unipolar device, which also provides an active low signal to the Arduino.

The furnace and the fireplace are controlled by 24 VAC thermostats. The two bi-directional opto couplers U3 and U4 generate the active low interface signal for the Arduino when those appliances run.

Gas appliances state their gas consumption on their name plates, but it is confusing—at least to me. The gas flow seems to depend on the altitude, temperature and perhaps some other influences that I found rather incomprehensible. The point is that I need relative values because they affect each other and are affected by the time and weather. The gas supplier takes care of accurate, absolute consumption measurement. So, I first calibrated the gas flow by running each appliance for 30 minutes and recording the readings in cubic meters (m^3). Then I normalized the readings to obtain the trends and ratios.

I haven't found a solution to establishing the stove top consumption. I tried several ideas but none of them worked to my satisfaction. But I'll continue looking. For the moment, I compared our monthly gas usage recorded by the monitor with the consumption on the gas company's invoice. I found that the month-to-month gas usage attributed to the stove top remains fairly constant through the year, and, more importantly, it is insignificant when compared with the overall gas usage. Given that my interest lies in normalized rather than accurate values, the stove top consumption may be safely ignored, for the time being. So that's it for my efforts to monitor (and hopefully to reduce) my energy consumption. Am I a hog? Or can I reduce my needs without some drastic or impractical measures? I don't know. It's too soon to tell.

A LAST MINUTE UPDATE

The RTC proved to be a disappointment. I used an older logger shield with a DSL1307 RTC. It was gaining some 4 minutes every week. That's obviously an unacceptable performance because it would have required regular time adjustments every few months. Hopeful that another RTC would be better, I purchased another logger with a purportedly better RTC, the NXP Semiconductor PCF8563. This one, unfortunately, is running too slow.

To fix the problem I have several options to consider which, let me emphasize, preclude any regular time adjustment. To begin with, there are RTCs claiming better accuracy, but that would entail purchasing another piece of hardware—something I have tried to avoid as one of the project's goals. I might be able to trim the RTC oscillator, by hardware or software.

But there's another relatively straightforward and inherently accurate solution. Some time back I built a WWVB repeater based on GPS and described in one of my past articles [6]. The date and time data are broadcast within my home by an XBee transceiver. Because I have a spare XBee transceiver, I can modify the monitor architecture and use this interface in place of the data logger shield. The total power draw is not likely to increase appreciably, if at all.

I need to do some analyses and design reviews before making a decision which way to go. But that is a topic for a future report once the decision has been made and the changes implemented.

ABOUT THE AUTHOR

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

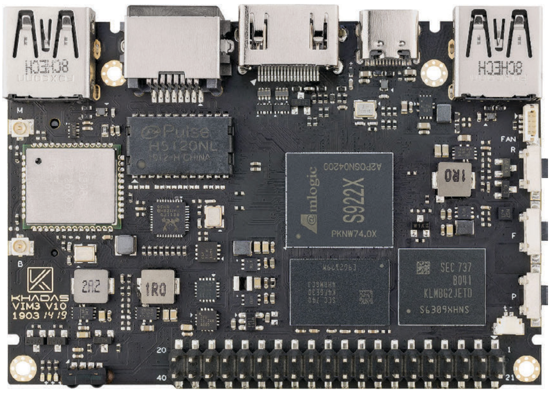
For detailed article references and additional resources go to:
www.circuitcellar.com/article-materials

RESOURCES

Adafruit | www.adafruit.com
Microchip Technology | www.microchip.com
NXP Semiconductor | www.nxp.com
Texas Instruments | www.ti.com
Silicon Labs | www.silabs.com
Sparkfun | www.sparkfun.com
Universal-Solder | www.universal-solder.ca

SBC Serves Up Cortex-A73 SoC and NVMe Support

Khadas has unveiled a "Khadas Vim3" SBC that runs Linux on an Amlogic S922X with 4x -A73 and 2x -A53 cores, with a future model featuring a neural processor. You get up to 4 GB RAM and 32 GB eMMC plus expansion via 40-pin GPIO, PCIe and M.2 with NVMe. The board is the 2nd Khadas hacker board to offer Amlogic's hexa-core S922X after Hardkernel's Odroid-N2. The Khadas Vim3, follows the quad-core Amlogic S905X based Khadas Vim1 and octa-core Amlogic S912 Khadas Vim2, has high-end features like NVMe storage and a combo interface that can be used for either PCIe or USB 3.0.



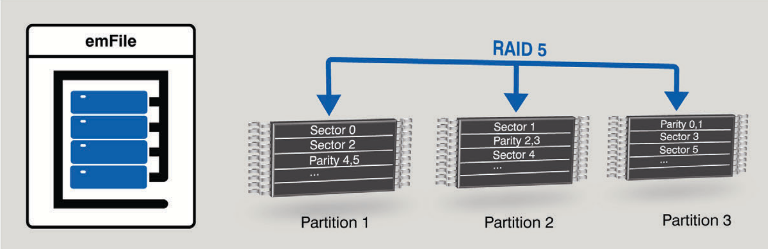
The Khadas Vim3 has the same, somewhat Raspberry Pi-like 82 mm x 58 mm x 11.5 mm footprint and layout as the earlier Vim boards, and similarly offers a 40-pin GPIO. The Khadas Vim3 ships in Basic (2 GB LPDDR4 and 16 GB eMMC 5.1) or Pro (4 GB/ 32 GB) models, both of which offer 16 MB SPI flash. Unlike the Vim2 (or Odroid-N2), there's an M.2 2280 socket, and it supports high-speed NVMe storage. Another novelty is the new PCIe 2.0 x1 interface, which is accessible via a combo socket that can switch to USB 3.0. Dual simultaneous displays are now available via the 4K-at-60 ready HDMI 2.1 port and the new 4-lane MIPI-DSI interface with touch-panel support. There's also a new 3-axis accelerometer.

Khadas | www.khadas.com

Embedded File System Adds RAID5 Support

SEGGER Microcontroller has added a RAID5 option to its emFile embedded file system to maximize data integrity and reliability. RAID5 delivers a high level of data integrity by providing an additional layer of fail-safety on top of the CRC and/or error correction used by the underlying driver or storage medium. With RAID5, the storage device is divided into several partitions, one of which holds parity information allowing data recovery in case of a partition fail. It can be used on any storage medium and the parity information can be stored on the same storage device (in a separate partition) or on a separate device.

The add-on has a tiny memory footprint, typically using about 1 KB of RAM and 3 KB of flash memory. RAID5 is available as an additional option and as an alternative to RAID1. RAID1 has been in the market for many years, and proven its value over and over, but does use 50% of the raw storage capacity for data security. The amount of storage used by RAID5 is configurable, typically using 5% or less. Being part of the emFile storage layer, the new add-on can also be used for USB Mass storage as well as other applications.



SEGGER Microcontroller | www.segger.com

Rad-Tolerant PWM Controller and GaN FET Driver Feature Plastic Packages

Renesas Electronics has announced what it claims is space industry's first plastic-packaged, radiation-tolerant PWM controller and Gallium Nitride (GaN) FET driver for DC/DC power supplies in small satellites (smallsats) and launch vehicles. The ISL71043M single-ended current mode PWM controller and ISL71040M low-side GaN FET driver are ideal for isolated flyback and half-bridge power stages and motor control driver circuits in satellite buses and payloads.

The ISL71043M PWM controller provides fast signal propagation and output switching in a small 4 mm x 5 mm SOIC plastic package, reducing PCB area up to 3x compared to competitive ceramic packages. In addition, the ISL71043's 5.5 mA max supply current reduces power loss more than 3x, and its adjustable operating frequency—up to 1 MHz—enables higher efficiency and the use of smaller passive filter components. The ISL71043M and ISL71040M are characterization tested at a total ionizing doze (TID) of up to 30 krads(Si), and for single event effects (SEE) at a linear energy transfer (LET) of 43 MeV•cm^2/mg. Both devices operate over an extended temperature range of -55°C to +125°C. The ISL71040M low-side GaN FET driver safely drives Renesas' rad-hard GaN FETs in isolated topologies and boost type configurations.

Renesas Electronics | www.renesas.com

PRODUCT NEWS

Fanless Industrial IoT Gateway Boasts Small Form Factor

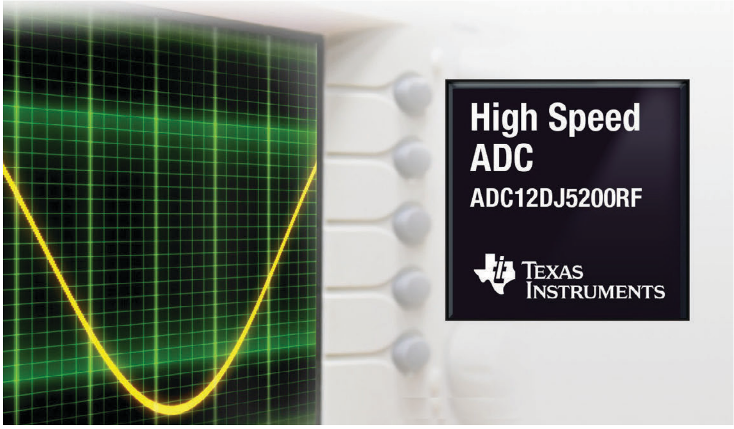
WIN Enterprises has announced the PL-80580, a fanless, small form factor for use as an Industrial IoT (IIoT) Gateway, and for networking applications requiring the small footprint and temperature tolerance of industrial applications. The unit supports a -10°C to 60°C operating temperature range. The small footprint of the PL-80580 (216 mm x 142 mm x 37.5 mm) also provides a good fit for robotics, cart-based medical and digital signage applications.

The PL-80580 features a choice of three Intel Atom E3800 3-D processors with Tri-gate design in single-, dual, and quad-core versions with 2x GbE LAN ports. The Intel processor is high performance, low-power consuming at 5 W, 7 w or 10 W. The E3845 SoC provides up to 1.91 GHz performance with its quad-core design. CPUs are partnered with the Intel i210AT GbE LAN controller. System I/O includes 1x USB 3.0, 2x USB 2.0, 2x Intel PCIe GbE, and 1x RS-232/422/485 & 3x RS232, plus expansion capabilities. ThePL-80580 is RoHS, FCC and CE compliant. WIN Enterprises will customize the PL-80580 based on customer's specific market requirements.

WIN Enterprises
www.win-ent.com

8 GHz 12-bit ADC Boasts 10.5 GSPS Sampling Rate

Texas Instruments (TI) has introduced a new ultra-high-speed ADC with what it claims is the industry's widest bandwidth, fastest sampling rate and lowest power consumption. The ADC12DJ5200RF helps engineers achieve high measurement accuracy for 5G testing applications and oscilloscopes, and direct X-band sampling for radar applications, says TI.



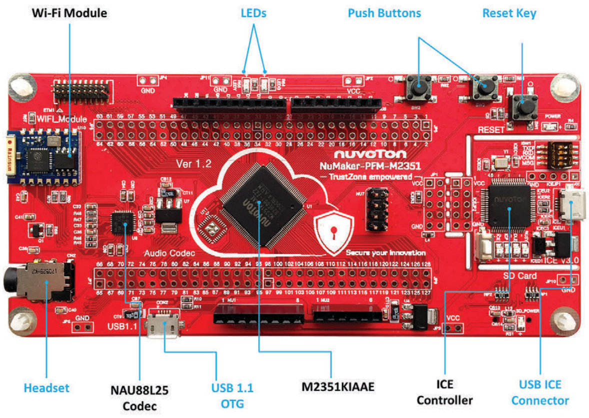
The ADC12DJ5200RF's 8 GHz bandwidth enables engineers to achieve as much as 20% higher analog input bandwidth than competing devices, which gives engineers the ability to directly digitize very high frequencies without the power consumption, cost and size of additional down-conversion. In dual-channel mode, the ADC12DJ5200RF samples at 5.2 GSPS and captures instantaneous bandwidth (IBW) as high as 2.6 GHz at 12-bit resolution. In single-channel mode, the new ultra-high-speed ADC samples at 10.4 GSPS and captures IBW up to 5.2 GHz. The device is in a 144-ball, 10-by-10-mm flip-chip ball grid array (FCBGA) package.

Texas Instruments | www.ti.com

Arm Cortex M23-Based MCUs Feature FreeRTOS Kernel Support

Nuvoton Technology has announced that it is demonstrating the capability of FreeRTOS kernel support with the NuMicro M2351 Series. According to the company, the M2351 is one of the first Arm Cortex-M23 based MCUs that has a preconfigured example that embedded developers can use to run FreeRTOS on the officially supported Armv8-M architecture. Amazon Web Services (AWS) released the latest FreeRTOS kernel that includes a preconfigured example project for the Nuvoton NuMaker-PFM-M2351 evaluation board (shown).

At the beginning of 2019, the M2351 Series had achieved with Arm PSA (Platform System Architecture) Level 1 Certified and PSA Functional Certification. PSA Certified enables device makers to achieve the security required for their use cases through three progressive levels of security assurance, each requiring increasingly rigorous hardware and software evaluation, which are assigned by analyzing the use case threat vectors.



Nuvoton Technology | www.nuvoton.com

IDEA BOX

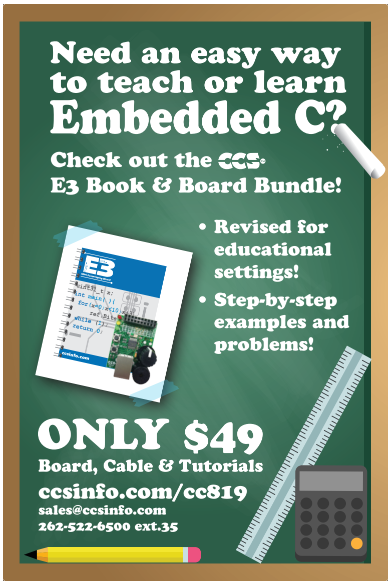
The Directory of PRODUCTS & SERVICES

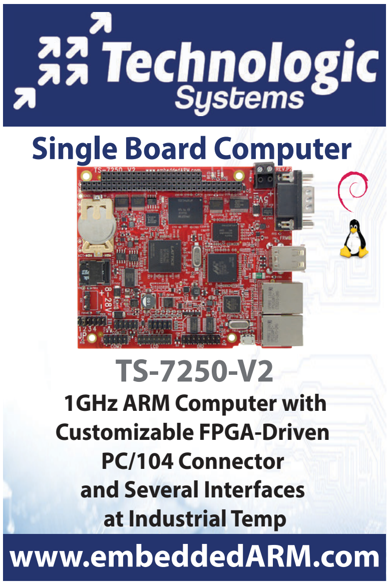
AD FORMAT:

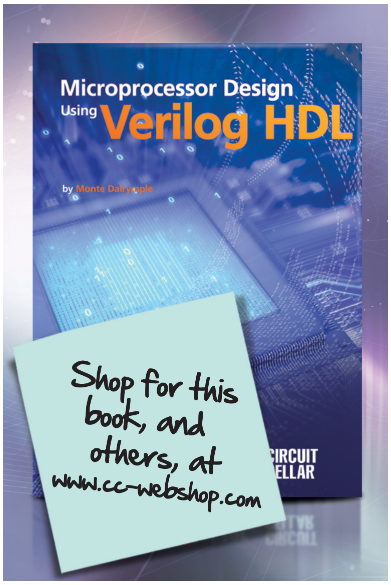
Advertisers must furnish digital files that meet our specifications (circuitcellar.com/mediakit).

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.







TEST YOUR EQ

Contributed by David Tweed

Answer 1— The R-C combination in question is being used to shorten the active time of the write-enable pulses going to the memory chip in order to meet hold-time requirements when the CPU is writing to memory.

When the clock makes a low-to-high transition while WR is high, the output of the NAND gate immediately goes low. But at the same time, the capacitor starts to charge up through the resistor, and at some point, the voltage at the input of the gate drops below its switching threshold, which forces the output high again.

Answer 2— This trick only works if the clock high time is significantly longer than the R-C time constant.

10 nF × 1 kΩ = 10 µs

So presumably the clock is something less than 50 kHz.

Actually, the 74189 is not a slow part. The minimum write pulse width is a few tens of nanoseconds—so the R-C time constant could be much shorter, by a couple of orders of magnitude.

Answer 3— Salt and pepper noise is the visual effect you get when noise flips random bits in a digital image. Isolated pixels become either much brighter or much darker than their neighbors, which gives rise to the name.

Answer 4— The usual solution is to use a median-value filter, which runs a 3×3 kernel over the image, replacing the center value with the median of the 8 surrounding pixels. The median filter does a good job of preserving edges and gradients in an image while removing the errors, unlike a simple low-pass filter, which simply blurs them out along with everything else in the image. A median filter does remove fine (noise-like) detail in an image, however.

Unfortunately, finding the median value in a list of values requires sorting the list, which is not a simple operation, especially if you want to do it quickly. This operation needs to be done individually for every pixel in the image, and when you're processing HD video at 148.5 Mpixels/s, you need a solution involving a deep pipeline that can produce a result on every clock cycle.

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

TESTS YOUR EQ

The Future of IoT Standards

Unlock IoT:
What's West of Westeros?

By
Cees Links,
General Manager,
Wireless Connectivity
Business Unit, Qorvo

I often find myself in conversations that involve someone wanting to know how to choose between Zigbee, Thread, Bluetooth or Wi-Fi technology? Or maybe LoRa? Or is it better to wait for 5G and NB-IoT (**Figure 1**)? They all have different standards, which is made more confusing by the (sometimes) unsubstantiated marketing claims about new standard capabilities, like "latency in the milliseconds!" (As if that's important. Most applications can deal with a latency of seconds. Even live TV delayed a few seconds is still live TV...)

So, how does one answer the question, "What standard should be used?" Usually, I answer this question with, "How do you make money?" Because typically, the determining factor isn't the radio standard. It's the application, that runs over the wireless.

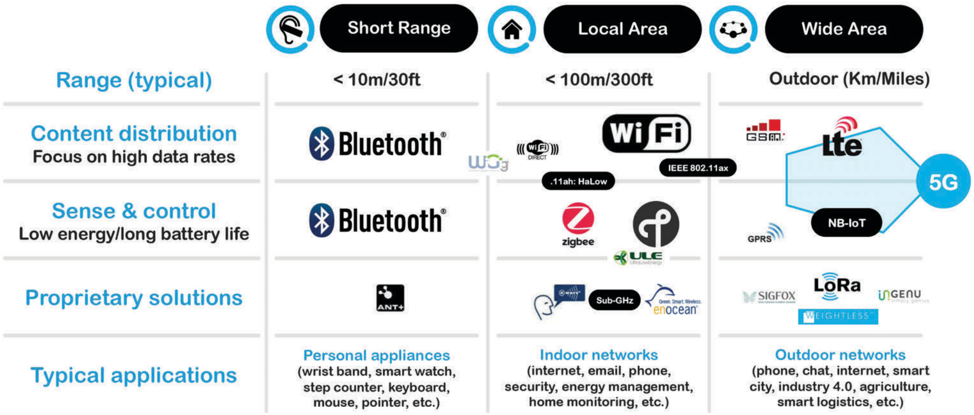
But maybe I'll invoke Game of Thrones and start saying "What's west of Westeros?" instead. Because keep in mind that whatever wireless standard is leading the pack today, in 5-10 years things will have change anyway, so upgrading your network will be important. Waiting for the "final" or "permanent" wireless networking technology will just be an exercise in waiting.

But can we channel our inner Arya (or Bran maybe?) and look toward the future to develop a sense of where things are going? I believe we can.

3 RADIOS FOR 3 SPACES

You may not have thought of it this way, but it's true. Our smartphones have 4G, Wi-Fi and Bluetooth technology—three radios. It's technically possible for a smartphone to only have one radio. Not cost-effective today, but possible. For example, you listen to music on your phone with a Bluetooth headset, but, technically, it would be possible to connect both the phone and a headset to the cellular network.

But for now, anyway, we've gotten used to having three radios in our smartphones. And we use them in our "spaces" in different ways. We use a Bluetooth radio for our immediate, bodily, personal



	Short Range	Local Area	Wide Area
Range (typical)	< 10m/30ft	< 100m/300ft	Outdoor (Km/Miles)
Content distribution Focus on high data rates	Bluetooth®	WiGig WiFi DIRECT WiFi IEEE 802.11ax .11ah: HaLow	GSM lte 5G
Sense & control Low energy/long battery life	Bluetooth®	zigbee ULE UltraLowEnergy	GPRS NB-IoT
Proprietary solutions	ANT+	Z-WAVE Sub-GHz Green. Smart. Wireless. enocean	SIGFOX LoRa INGENU WEIGHTLESS
Typical applications	Personal appliances (wrist band, smart watch, step counter, keyboard, mouse, pointer, etc.)	Indoor networks (internet, email, phone, security, energy management, home monitoring, etc.)	Outdoor networks (phone, chat, internet, smart city, industry 4.0, agriculture, smart logistics, etc.)

FIGURE 1
An overview of IoT wireless standards

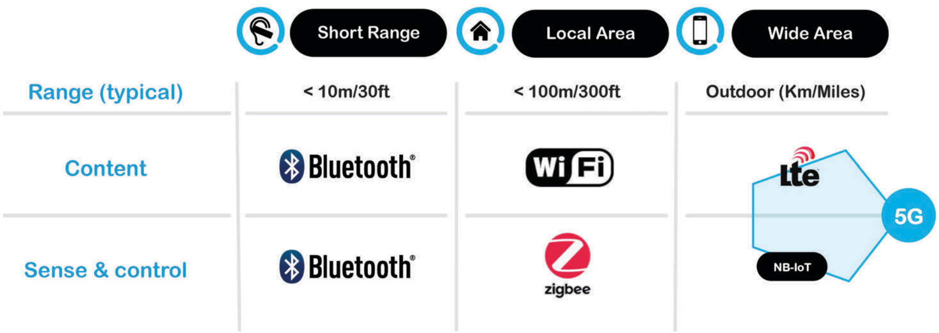


FIGURE 2
This consolidated view of low-power IoT wireless standards makes sense because it lines up with the three radios in our smartphone and how we those radios.

space—like that Bluetooth headset for listening to music. We use Wi-Fi for our private spaces like home and office—places that are private territory versus public. And we use 3G/4G—and soon 5G—for access in the public space. This division, by the way, just fell into place over the years. Anyone who has been involved in the "war of standards" knows there was no higher committee deciding how to neatly divide different radios and standards over these three spaces.

In the past 20-30 years, every standard has battled for maximum usage space—and some of these battles are still going on. 5G, the next public space standard, is claiming that it works well indoors and may make Wi-Fi redundant. Bluetooth initially made serious claims for the Wi-Fi indoor market (never successful), while Wi-Fi has been eyeing the Bluetooth market with Wi-Fi Direct. Although Wi-Fi Direct isn't dominant, it isn't dead either—for drones, Wi-Fi Direct has found an interesting niche.

APPLYING THIS TO STANDARDS

There's a parallel situation developing for low-power IoT standards. Remember that the real focus of these low-power standards is on long battery life—not data rates—without compromising range. Usually, low-power standards are used to connect devices—sensors for example—to the internet (IoT) for the purpose of sharing sensor data, and the data rate required for this is orders of magnitude lower than for "normal" Internet usage or watching videos. Essentially, low-power standards exchange data rate for battery life.

Interestingly, these low-power standards are rallying around the same three ranges mentioned earlier:

- For our personal space: Low-power Bluetooth technology (sometimes called Bluetooth Low Energy)
- For our private space (such as home, office or hotspot): Zigbee (IEEE 802.15.4), which is essentially a low-power Wi-Fi standard
- For the public space: NB-IoT/Cat-X as part of 4G/5G

Combined, **Figure 2** shows that system of standards would look like. Not incidentally, this low-power/sense-and-control alignment makes all the sense in the world because it lines up with how we experience spaces with the three radios in our smartphone.

NOT SO FAST...

But remember, nothing is for sure. Technologies continue to try venturing into "other spaces." These initiatives can vary from ways to maximize the reuse of existing technology or trying to answer simple questions—like, can NB-IoT/Cat-X also be used for indoor networks? Similarly, there's an effort to expand Bluetooth Low Energy into the networking space and adding meshing capabilities to displace Zigbee—in the same way that Bluetooth technology tried to displace Wi-Fi around 20 years ago. These efforts may not be successful, but big companies are spending serious dollars on them.

And there may be a fourth space to consider. Something that falls between the local area (indoor) and the wide area (public). There are large outdoor spaces that are still semiprivate, such as campuses, harbors, airports and convention centers. Do we need another standard that is focused on this "in-between" domain? The future will tell. Maybe the existing standards are flexible (and cost-effective) enough to serve this space well. Or maybe an emerging standard will be better qualified. In this respect, it's interesting to see that LoRa, as a pseudo standard for outdoor, is very visible in this zone between private and public, in harbors and airports.

For standards venturing into other areas, it's also interesting to mention that the boundary between Wi-Fi (high data rate) and Zigbee (low power) is not very definite. There are applications that require higher data rates and need to run on batteries, thus requiring low power. For some time now there have been significant efforts to make Wi-Fi truly low power. True low-power implementations of Wi-Fi 4 (.11n) and Wi-Fi 5 (.11ac) might be interesting alternatives for Zigbee technology, if they reach low enough power—in other words, long enough battery life.

THE TAKEAWAY

Ultimately, the usefulness of the application creates the value for the end user, not the wireless "wire." Only time will tell which standards will be the most successful, but no point in waiting. Go ahead and make money now.

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

RESOURCE
Qorvo | www.qorvo.com

Cees Links was the founder and CEO of GreenPeak Technologies, which is now part of Qorvo, and is now the General Manager of the Wireless Connectivity Business Unit. He was involved in the establishment of numerous standards efforts including the IEEE 802.11 standardization committee and the Wi-Fi Alliance. He was also instrumental in establishing the IEEE 802.15 standardization committee to become the basis for the ZigBee sense and control networking. He was recognized as Wi-Fi pioneer with the Golden Mousetrap Lifetime Achievement Award [1] and more recently inducted into the Wi-Fi NOW Hall of Fame [2].

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.

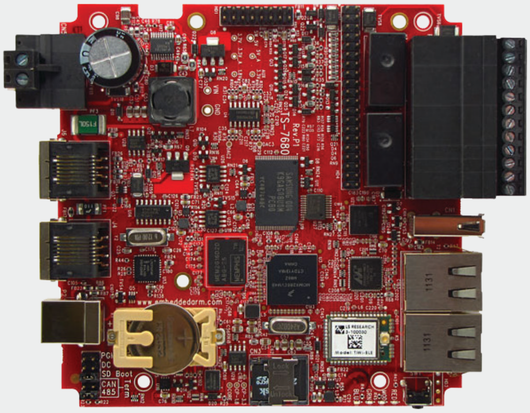
You no longer have to worry about long shelf life
or tie your capital in bare board inventory.

Accutrace inc.

www.PCB4u.com sales@PCB4u.com
SAM & ITAR Registered UL E333047 ISO 9001 - 2008

www.embeddedARM.com

FROM THE DEEP BLUE SEA TO THE WILD BLUE YONDER



TS-7680

Low Power Industrial
Single Board Computer with
WiFi and Bluetooth

$159
Qty 100

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

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

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

Made in USA
with Global Parts

Technologic Systems