



circuit cellar

Inspiring the Evolution of Embedded Design

POWER TAKES PRIORITY IN WEARABLE DEVICES



- Product Focus: Power Supplies for Medical Use ■ Flex PCB Design Services |
- Guitar Video Game Uses PIC32 | MCU-Based IC Links USB to Legacy PC I/O |
- Pet Collar Uses GPS & Wi-Fi | Formal Design Flow for Automotive Safety
- Impedance Spectroscopy | Fun With Current Sensors | Cooling Schemes |
- Side-Channel Power Analysis ■ The Future of IoT Connectivity





The Embedded Experts



J-Trace PRO Cortex

The Superspeed Trace Probe



- USB Superspeed and Gigabit Ethernet interfaces
- Real-time streaming at full System Clock
- Tune your application with live profiling
- Instruction-level code coverage
- Isolate and Identify hard-to-find code defects
- Supports Cortex-A/R/M targets
- Full J-Link debug functionality
- Included cross-platform Ozone debugger

segger.com

Worldwide: sales@segger.com

+49 2103 2878 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

■ **It simply works!**

Together, we are making security implementation easier for you.



Industry-leading development tools and ground-breaking security technology

IAR Systems and Secure Thingz are teaming up to make security implementation part of the development workflow with the release of Embedded Trust and C-Trust.

Embedded Trust is a security development environment, which simplifies the configuration of security, from the root of trust and key storage for a connected device to the creation of security profiles and projects.

C-Trust is an extension to IAR Embedded Workbench that enables application developers to deliver secure, encrypted code as part of their standard workflow.

Make security a natural part of your day-to-day development

Learn more at iar.com/security

OUR NETWORK



SUPPORTING COMPANIES

Accutrace, Inc.	C3
All Electronics Corp.	77
AUVSI XPONENTIAL 2019	29
Avnet	59, 77
CCS, Inc.	77
EzPCB	25
IAR Systems, Inc.	1
Labcenter Electronics	13
Mentor, A Siemens Business	47
PCBCart	33
SEGGER Microcontroller Systems	C2
Slingshot Assembly	67
Technologic Systems, Inc.	C4, 77
WilsonPro	55

NOT A SUPPORTING COMPANY YET?

Contact Hugh Heinsohn

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

THE TEAM

PRESIDENT

KC Prescott

EDITOR-IN-CHIEF

Jeff Child

ADVERTISING COORDINATOR

Nathaniel Black

CONTROLLER

Chuck Fellows

TECHNICAL EDITOR

Carol Bower

ADVERTISING SALES REP.

Hugh Heinsohn

FOUNDER

Steve Ciarcia

GRAPHICS

Grace Chen
Heather Rennae

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 344 March 2019 | ISSN 1528-0608

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

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

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

SUBSCRIPTION MANAGEMENT

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

CUSTOMER SERVICE

E-mail: customerservice@circuitcellar.com

Phone: 434.533.0246

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

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

NEW SUBSCRIPTIONS

circuitcellar.com/subscription

ADVERTISING

Contact: Hugh Heinsohn

Phone: 757-525-3677

Fax: 888-980-1303

E-mail: hheinsohn@circuitcellar.com

Advertising rates and terms available on request.

NEW PRODUCTS

E-mail: editor@circuitcellar.com

HEAD OFFICE

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

COPYRIGHT NOTICE

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

DISCLAIMER

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

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

© KCK Media Corp. 2019 Printed in the United States

INPUT Voltage

A Case for Intelligent Interconnects

As I do most every year in January, I attended this year's annual Embedded Tech Trends (ETT) event. Held in San Diego this year, ETT is an industry-wide forum where suppliers of component, board and system level solutions can meet exclusively with members of relevant industry media to discuss technologies, trends and products. Organized by the trade association VITA, ETT is a combination of TED-talk style presentations from the vendors and one-on-one meetings between us in the press and the 21 sponsoring vendors. A busy couple of days! These 21 companies are the leading makers of board, box and interconnect products designed for the more rugged end of the embedded spectrum—everything from transportation to factory automation to defense.

For me, the most interesting topic at ETT was optical interconnect technologies. While optical interconnections came up in many of the presentations, there were four with optical right in their titles.

All these presentations provided great insights and information. And I learned a lot about optical interconnects and other technologies. And in the follow up one-on-one meetings I learned still more. All that said, I must admit there was one ETT presentation that made me perk up my ears. In his talk, Ritch Selfridge, director of engineering for Amphenol Aerospace made the case for intelligent autonomous interconnects.

"I have a dream that intelligent interconnects will enable the creation of intelligent autonomous data and power delivery systems," began Selfridge, "The electronics revolution continues to enhance numerous devices and interconnects should be no exception!" He used an avionics subsystem as the example in his discussion, but presumably it could apply to any electronics subsystem.


Implementing these intelligent autonomous interconnect systems would require integrating sensors and intelligence into the interconnect system. A whole host of miniature sensors are being used today in smart devices and medical instruments, said Selfridge. These sensors can be integrated into interconnects permitting the monitoring of interconnect systems for numerous parameters such as temperature, pressure, voltage, acceleration and position. The sensor outputs are provided to the sub-systems to process the data. Smart watches and exercise trackers are constantly evolving, providing more feedback while consuming less energy. The processing and memory cores of these devices can be incorporated into interconnects along with sensors to provide autonomous interconnect

system health diagnostics without needing support from subsystems, said Selfridge.

Meanwhile, new and refined miniature devices that convert thermal and dynamic energy into electricity are emerging. These devices, coupled with miniature energy storage systems, can also be incorporated into interconnect systems to power the sensor data processing and sensor systems, eliminating the need to distribute platform power to the intelligent interconnects.

"Reliable, passive interconnects will always be in demand. But when coupled into an intelligent autonomous interconnect system, they can be streamlined to provide superior performance with less weight while not burdening other sub-systems," said Selfridge. "Many of the components required already exist and new ones are emerging on a constant basis. Dare to dream and anything is possible!" An ambitious idea to be sure, but I for one was intrigued.

The talks mentioned here and all the of the rest of presentations from this year's ETT are available at www.embeddedtechtrends.com.

Shifting gears, I am saddened to share with you that *Circuit Cellar* columnist George Novacek passed away in late January after a long illness. He will be missed. George has been contributing articles to this magazine since 1999 penning over 120 articles over the years. George's monthly column "The Consummate Engineer" began in 2010 and it was aptly named. George was the consummate engineer, but also a skilled educator, providing *Circuit Cellar* readers with insightful articles that explain complex topics with clarity while sharing his deep knowledge and engineering experience. Fortunately, *Circuit Cellar* won't feel the loss of George's contributions right away. George had several months' worth of articles written, and it was his wish that we publish them—a wish we are both happy to honor and grateful to benefit from. Thank you, George. 



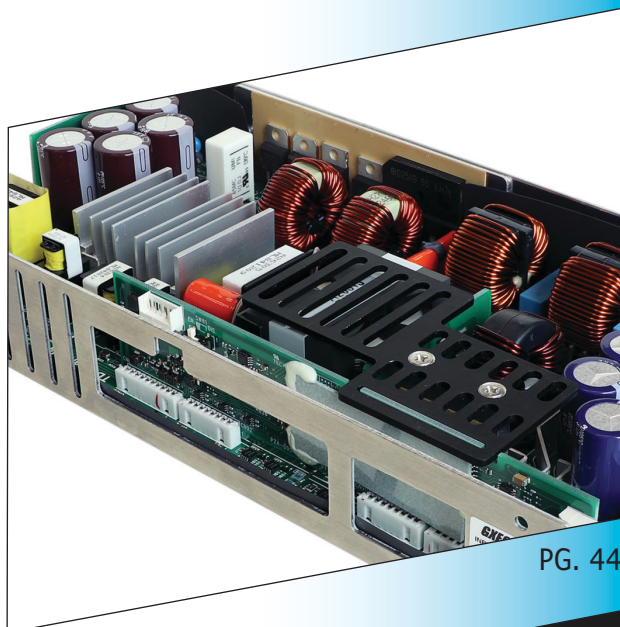
Jeff Child

COLUMNS

PRODUCT FOCUS

44 Power Supplies for Medical Use Compliance for Care

By Jeff Child



PG. 44

48 Embedded System Essentials Side-Channel Power Analysis Easy Path to Proof

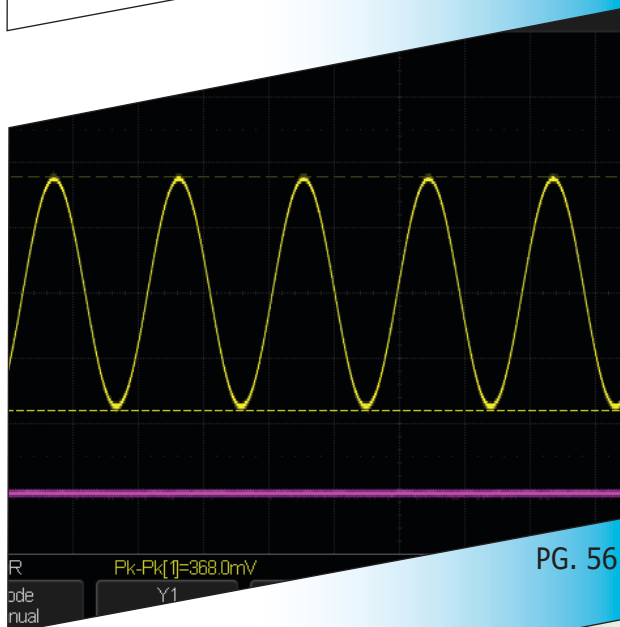
By Colin O'Flynn

52 The Consummate Engineer Cooling Electronic Systems Beat the Heat

By George Novacek

56 Picking Up Mixed Signals Impedance Spectroscopy Using the AD5933 From IC to Instrument

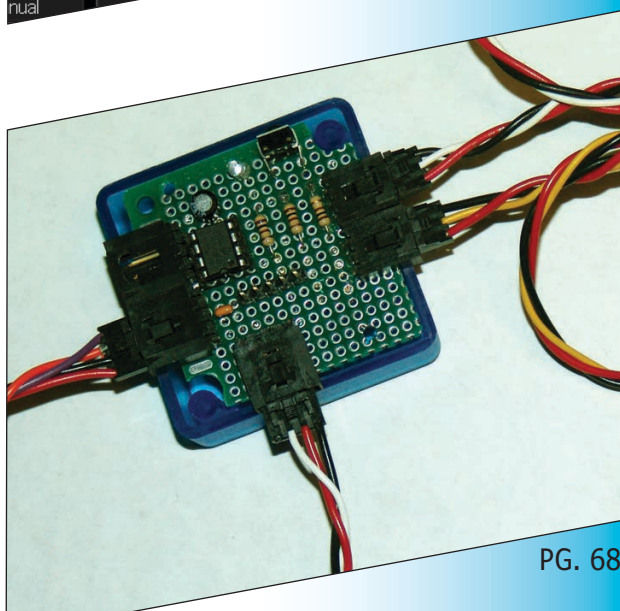
By Brian Millier



PG. 56

68 From the Bench Non-Evasive Current Sensor Pondering Probes

By Jeff Bachiochi



PG. 68

TECH THE FUTURE

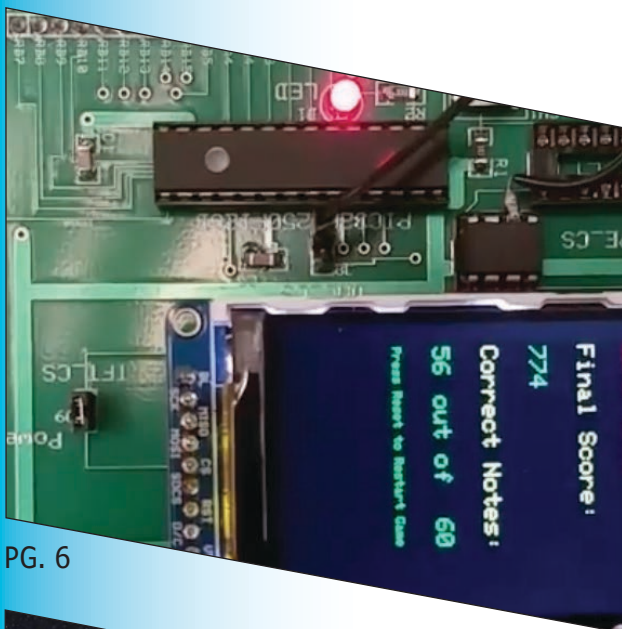
79 The Future of IoT Connectivity Cellular Technology's Role in IoT Market Growth

By Broc Jenkins

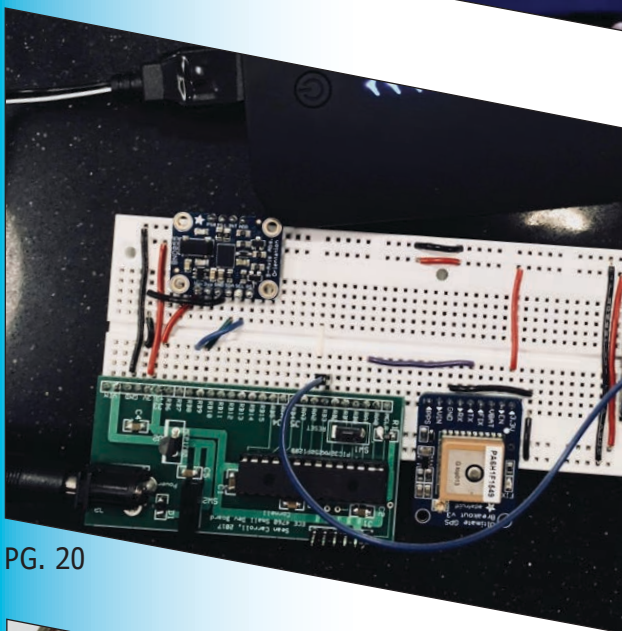
75 : PRODUCT NEWS

78 : TEST YOUR EQ

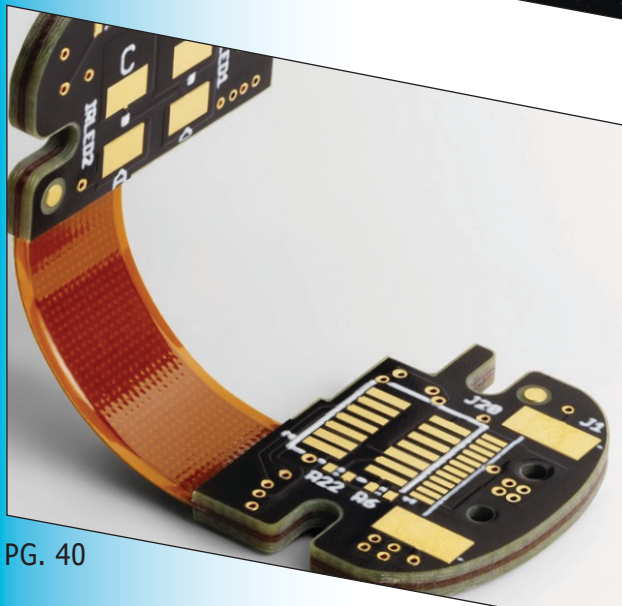
FEATURES



PG. 6



PG. 20



PG. 40

6 Guitar Video Game Uses PIC32

Realism Revamp

By Jake Podell and Jonah Wexler

14 MCU-Based IC Links USB to Legacy PC I/O

Versatile Solution

By Hossam Abdelbaki

20 Smart Pet Collar Uses GPS and Wi-Fi

PIC32-Based Design

By Vidya Ramesh and Vaidehi Garg

26 Formal Flow for Automotive Safety

Bulletproofing Car Design

By Doug Smith

34 Tailored Solutions Tackle Design Needs for Wearables

SPECIAL FEATURE

Low Power Priorities

By Jeff Child

40 Flex PCB Design Services

TECHNOLOGY SPOTLIGHT

Building Boards that Bend

By Jeff Child

Guitar Video Game Uses PIC32

Realism Revamp

While music-playing video games are fun, their user interfaces tend to leave a lot to be desired. Learn how these two Cornell students designed and built a musical video game that's interfaced using a custom-built wireless guitar controller. The game is run on a Microchip PIC32 MCU and has a TFT LCD display to show notes that move across the screen toward a strum region.

By
Jake Podell and Jonah Wexler

While many popular video games involve playing a musical instrument, the controllers used by the player are not the greatest. These controllers are often made of cheap plastic, and poorly reflect the feeling of playing the real instrument. We have created a fun and competitive musical video game, which is interfaced with using a custom-built wireless guitar controller (**Figure 1** and **Figure 2**). The motivation for the project was to experiment with video game interfaces that simulate the real-world objects that inspired them.

The video game is run on a Microchip PIC32 microcontroller [1]. We use a thin-

film-transistor LCD display (TFT) to display notes that move across the screen toward a strum region. The user plays notes on a wireless mock guitar, which is built with carbon-impregnated elastic as strings and a conducting plectrum for the guitar pick. The game program running on the PIC32 produces guitar plucks and undertones of the song, while keeping track of the user's score. The guitar is connected to an Arduino Uno and Bluetooth control center, which communicates wirelessly to the PIC32.

The controller was designed to simulate the natural motion of playing a guitar as closely as possible. We broke down that motion on a real guitar into two parts. First, users select



FIGURE 1

Front of the guitar controller. Note the strings and plectrum.

the sound they want to play by holding the appropriate strings down. Second, the users play the sound by strumming the strings. To have a controller that resembled a real guitar, we wanted to abide by those two intuitive motions.

FRET & STRUM CIRCUITS

At the top of the guitar controller is the fret board. This is where the users can select the sounds they want to play. Throughout the system, the sound is represented as a nibble (4 bits), so we use 4 strings to select the sound.

Each string works as an active-low push-button. The strings are made of carbon-impregnated elastic, which feels and moves like elastic but is also conductive. Each string is wrapped in 30-gauge copper wire, to ensure solid contact with any conductive surfaces. The strings are each connected to screws that run through the fret board and connect the strings to the fret circuit (**Figure 3**).

The purpose of the fret circuit is to detect changes in voltage across four lines. Each line is branched off a power rail and connected across a string to an input pin on an Arduino Uno. Current runs from the power rail across each string to its respective input pin, which reads a HIGH signal. To detect a push on the string, we grounded the surface into which the string is pushed. By wrapping the fret board in a grounded conductive pad and pushing the string into the fret board, we are able to ground our signal before it can reach the input pin. When this occurs, the associated pin reads a LOW signal, which is interpreted as a press of the string by our system.

Along with the fret circuit, we needed a way to detect strums. The strum circuit is similar in its use of a copper-wrapped, carbon-impregnated elastic string. The string is connected through the fret board to an input pin on the Arduino, but is not powered. Without any external contact, the pin reads LOW. When voltage is applied to the string, the pin reads HIGH, detecting the strum. To mimic the strumming motion most accurately, we used a guitar pick to apply the

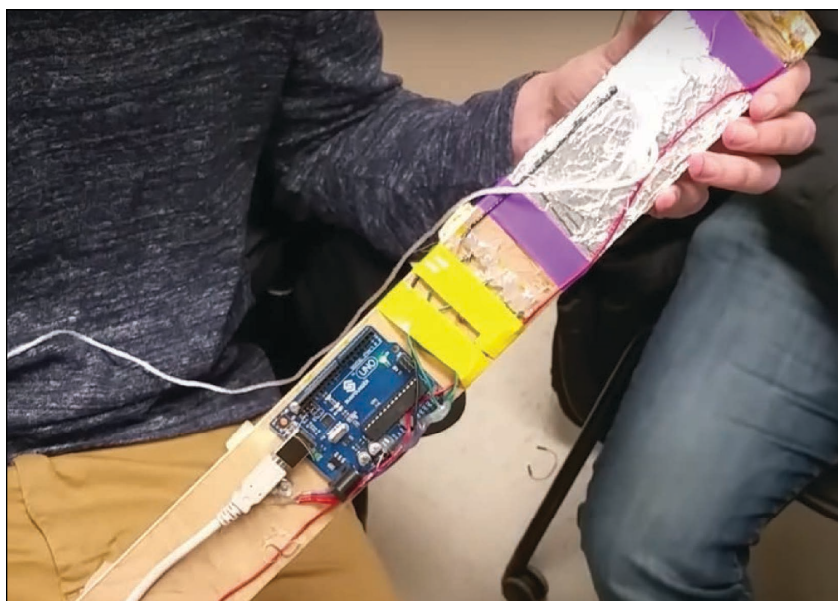


FIGURE 2
Back of the guitar controller

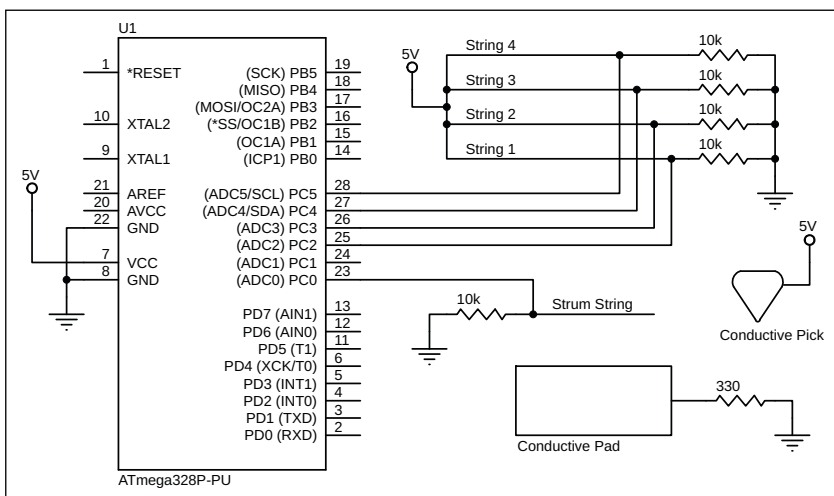


FIGURE 3
Complete controller circuit schematic (on guitar).

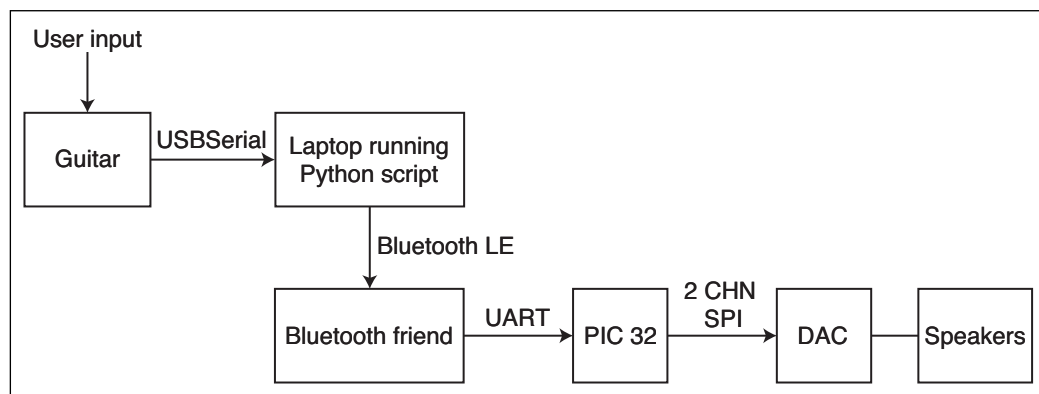


FIGURE 4
Shown here is a block diagram of the controller signals.

voltage to the string. The pick is wrapped in a conductive material (aluminum foil), which is connected to the power rail. Contact of the pick applies voltage to the string, which on a rising edge denotes a strum.

As shown in **Figure 4**, the direct user interface for the player is the guitar controller. The physical interaction with the guitar is converted to an encoded signal by an Arduino mounted to the back of the guitar. The Arduino Uno polls for a signal that denotes a strum, and then reads the strum pattern across the four strings. The signal is sent over USB serial to a Bluetooth control station, which uses a Python script to broadcast the signal to an Adafruit Bluetooth LE module. The laptop that we used as a Bluetooth control station established a link between the controller and the Bluetooth receiver, and was paramount to the debugging and testing of our system. Finally, the Bluetooth module communicated over UART with the PIC, which interpreted the user's signal in the context of the game [2].

To enable communication between the Bluetooth module and the PIC, we connected the TX and RX lines on the module to the

RX and TX lines on the PIC, respectively. These lines are shown in the diagram of the development board (**Figure 5**) as RA1 for RX and RB10 for TX. The Bluetooth module can be found as UZ.

SOFTWARE DESIGN

Animation, score tracking and song generation were accomplished in two threads in the main C program on the PIC32 [3]. The animation thread was run at a constant frame rate, such that the user experiences a uniform rate during the game. It is slow enough, however, to allow for the game processing and UART communication protocol. The game thread is run as quickly as possible to check for correct notes played and to update scoring.

The game thread handles all scoring and checking whether the correct note is played at the correct time. The thread keeps track of the current game time, and upon detecting a strum, compares it with a window of acceptable times to be strumming for the current note. For any note, this window starts at the note's "start time," and has a constant period, which we refer to as the "user error tolerance." After

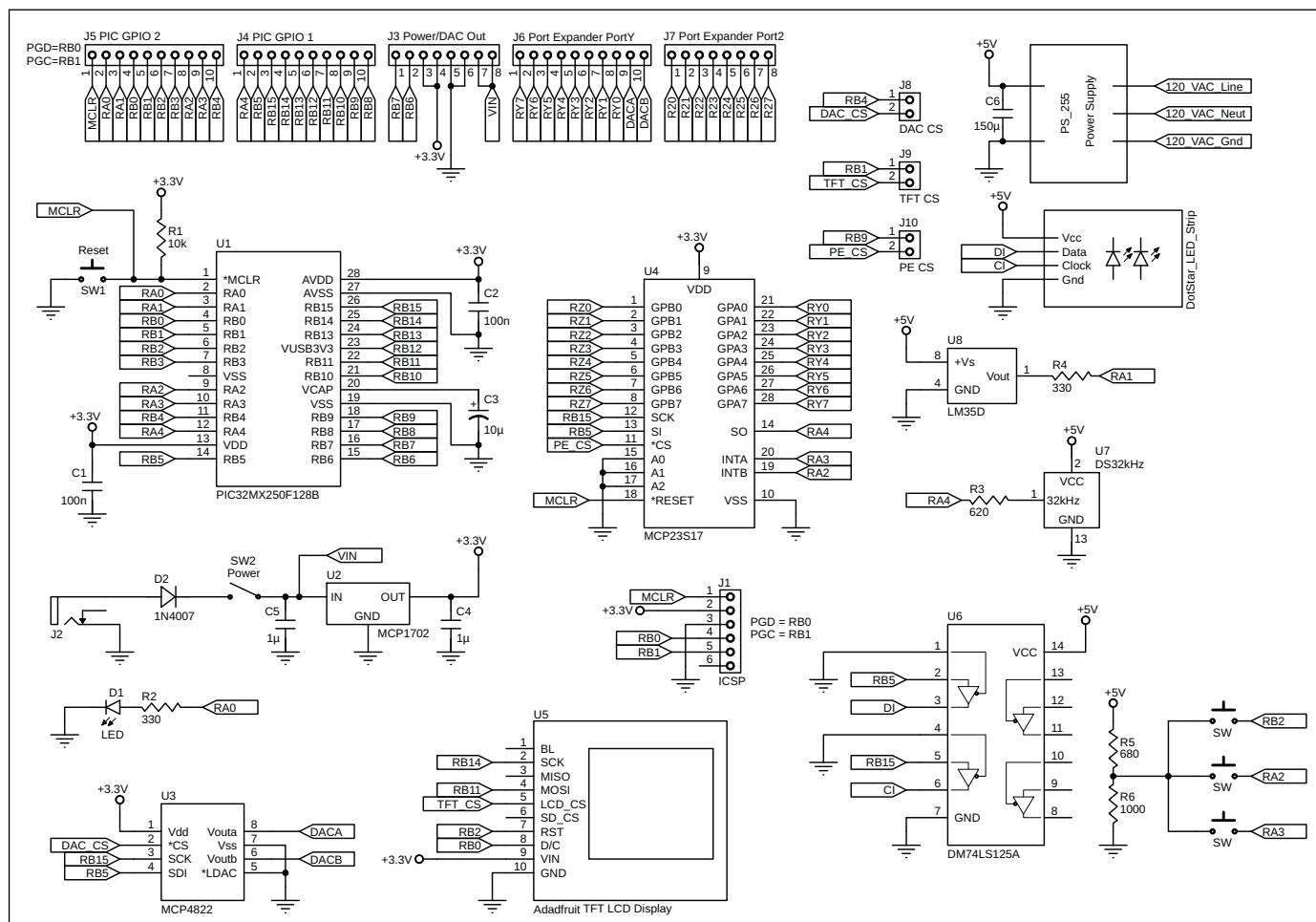


FIGURE 5
PIC32 Development Board Schematic. Note the RX (RA1) and TX (RB10) lines used for UART communication, and the DAC outputs used for sound generation.

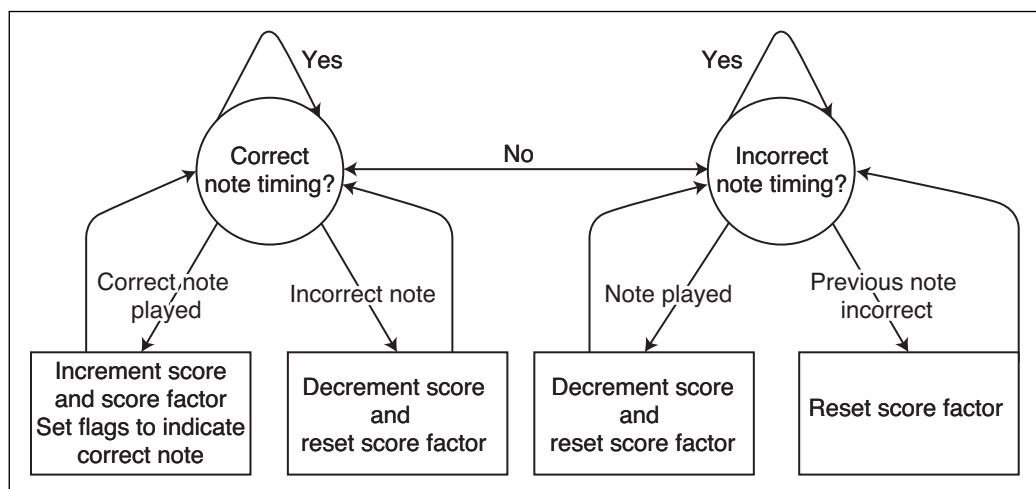


FIGURE 6

Game logic utilized in game thread.

a note's window has passed, we increment a pointer to the current note, and reset all variables related to sound generation.

Upon detecting a strum, we check that the current game time is within the current note's start time and user error tolerance. If it is, we then check whether the string pattern is equal to the correct value according to the note's frequency. If this is the current condition, we increment the score by the score factor and increment score factor by 1, unless the latter has already reached its maximum value of 30. If the note being played is a "G", which requires two strings to strum, the increment to the score is doubled. If a strum is detected at a time that is not within any note's timing window, or if the string pattern is incorrect, then we reset the score factor to 1, and decrement the score by 2, unless the score is at its minimum value of 0.

If a note's time window has passed—more specifically if the current game time is greater than the start time of the note plus the user error tolerance—and the user never made an attempt to play the note, then the user is penalized. Similar to the previous case, we reset the score factor and decrement the score.

The cases above are shown in the game logic block diagram (Figure 6), which represents the states we will be in during the game. Note that at all points in the game, we are either in a correct or incorrect time period, and in each of those periods, we continuously check the other conditions previously mentioned.

Once the last note has been played, the game state enters a final score display screen (Figure 7) and waits until the user restarts the game. The final display screen shows the correct number of notes played out of the total

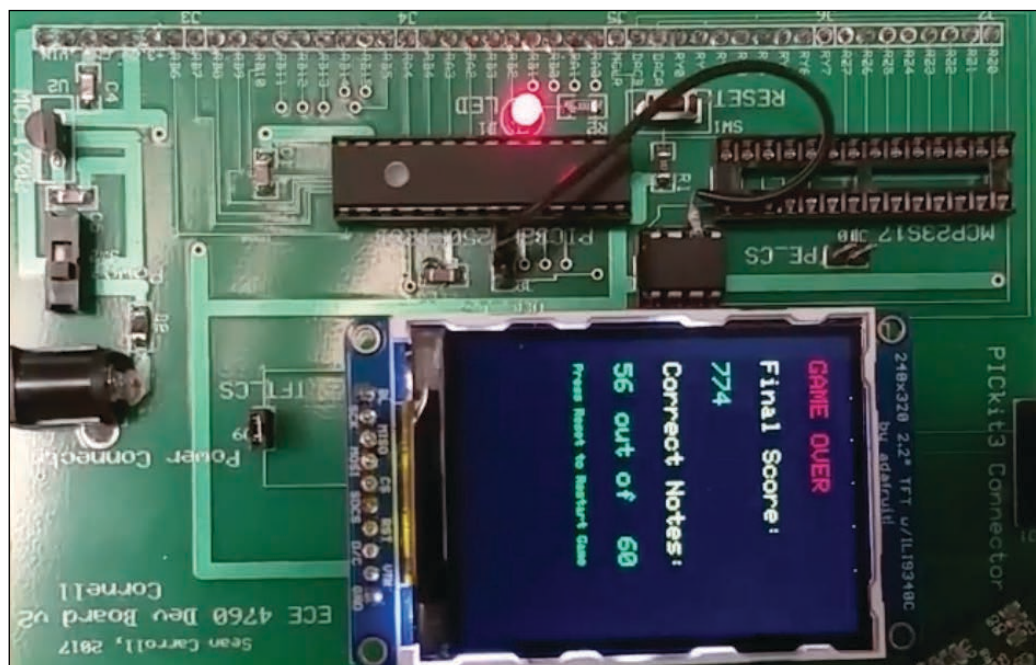
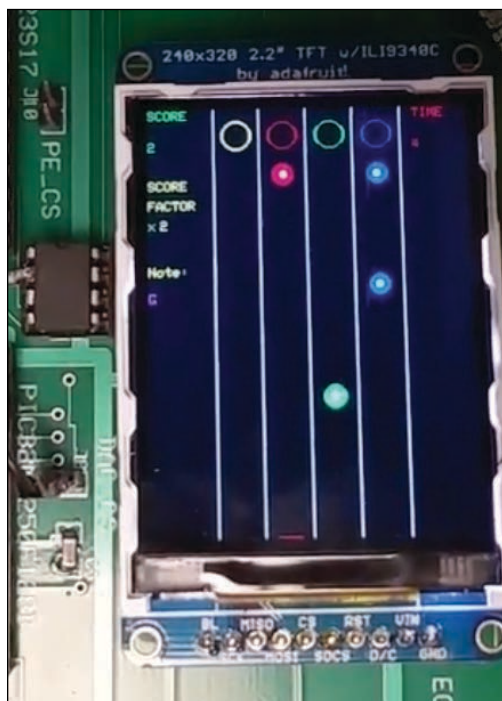


FIGURE 7

TFT display at the end of the game. Note the total score is displayed and the number of correct notes played out of the total possible notes.

FIGURE 8

TFT display mid game. Note the score, score factor, and current string pattern shown on the left side, the game time shown on the right, and the notes being displayed in the middle.



possible and the final score. It also prompts the user to restart the game if desired.

The movement of all the notes across the TFT display is managed by the animation thread. This thread begins by initializing the TFT for a vertical resolution. The notes start at the bottom of the screen and then move across the screen toward a “strum region.” When the notes hover over that area, the user is supposed to strum the correct strings. The strum region, shown in **Figure 8**, is represented by empty circles of the same color as that of the note that will be played. We display the score, score factor and current game time on the sides of the TFT. We also display the musical note representation of the current string pattern.

We had a hard requirement of at least 15 frames/s for drawing the notes and updating the scoring metrics to eliminate tearing in the display. Therefore, we determined trade-offs between the sampling frequency of the audio, processing of the game logic, and the number and frequency of updates on the TFT display. To meet the audio and animation requirements, we implemented our sound generation with a sampling frequency of 8,000 Hz instead of a

higher frequency. The final design allowed for a constant 16+ frames/s.

POSITIONING THE NOTES

Each time we enter the animation thread, we must update the y position of the next four notes to be played. The Y position follows a function that ensures that the note is in the strum region when the current game time matches the note’s start time. We erase the previous animation of a passed note by keeping track of the note’s position before Y is updated. The calculation for these positions is shown in the code in **Listing 1**. Active notes are drawn to the screen with colors matching their frequencies, as follows:

“C” = YELLOW; “D” = RED; “E” = GREEN; “F” = BLUE; “G” = RED & BLUE.

When a note has been played correctly, there is auditory feedback matching the feedback of the guitar. The sound generation occurs in an interrupt service routine (ISR) at a sampling frequency (F_s) of 8,000 Hz. Sound is generated using Direct Digital Synthesis (DDS) for the undertones of each note, and Karplus-Strong String Synthesis for the guitar plucking noise when a note is played correctly [4]. The sound is outputted on a two-channel speaker using two lines of a DAC (A for undertones and B for Karplus-Strong). We write to both lines of the DAC through SPI.

Our configuration includes setting up a timer (Timer2), which allows us to have a DDS sample rate of 8 kHz on a 40 MHz clock. We set the timer to trigger our DDS logic ISR. We set up Timer2 by first opening it to interrupt at 5,000 cycles, configuring it, and clearing the interrupt flag, as shown in **Listing 2**.

For the undertones, we looped through a 256-entry sine table using a 32-bit phaser. Because the sine table is only 256 entries long, we used bit shifting to get the top 8 bits of the phaser before doing a lookup. The phaser increment value is determined by:

$$\text{phase increment} = \frac{\text{output frequency} \times 2^{32}}{\text{sample frequency}}$$

where our sampling frequency is 8 kHz. The

```
for (i = 0; i<4; i++){
    notes[noteIndex+i].yprev = notes[noteIndex+i].y;
    notes[noteIndex+i].y = 20 - ((currentGameTime-notes[noteIndex+i].startTime)
    /10);
}
```

LISTING 1

Shown here is code for calculating the X and Y positions.

calculated phase increment is added to a phase accumulator. The output frequency is changed based on the current note the user is supposed to play, regardless of whether the note is played correctly. This allows for audio feedback if the note is played incorrectly or if the user is unable to determine visually what note to play.

We also modulated the amplitude of the sound linearly from 0 to 1 at the start of each note, to avoid any annoying clicking at the start or end of a note. We chose linear ramping because of the simplicity of the calculation, and to maintain a nearly continuous sound for the user. The undertone is outputted to channel A of the DAC and played on one channel of a connected speaker.

For the acoustic guitar string plucking, the basic filter for a Karplus-Strong implementation is shown in the block diagram in **Figure 9**. We implemented this for each note using fix16 arithmetic to utilize fewer CPU cycles in the interrupt [9]. An example of the C implementation of this filter is shown in **Listing 3**.

Each note's table is originally set to white noise, to simulate the initial randomness of an acoustic guitar string pluck that converges to a single frequency. We did not dampen, to simulate a high energy pluck and to allow the note to play continuously at the same energy level until it ends. The frequency of the note determines the correct amount of delay we require for each update cycle. Whenever a note is finished, its circular buffer array is refilled with white noise for the next time it is played.

RESULTS AND DISCUSSION

We used unit testing for each component of the project, to ensure that every part of the system worked separately. We then combined each of these components and tested edge cases to ensure the implementation worked according to the specifications we set out to fulfill.

First, we constructed the physical set-up of the mock guitar using carbon-impregnated

```
OpenTimer2(T2_ON | T2_SOURCE_INT | T2_PS_1_1, 5000);
// set up timer with priority 1
ConfigIntTimer2(T2_INT_ON | T2_INT_PRIOR_1);
mT2ClearIntFlag(); // clear interrupt flag
```

LISTING 2

We set up Timer2 by first opening it to interrupt at 5,000 cycles, configuring it, and clearing the interrupt flag, as shown in this code.

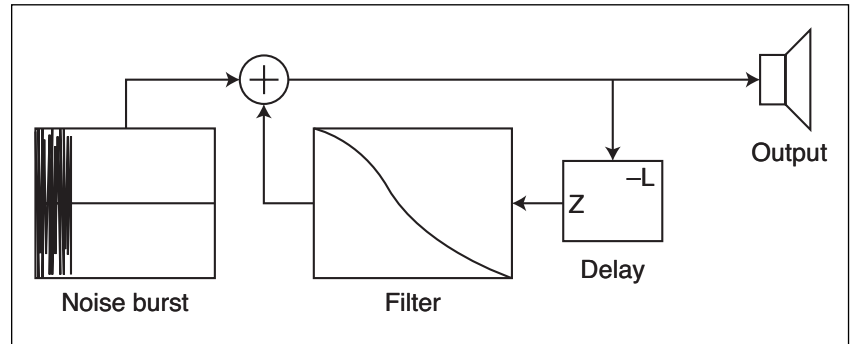


FIGURE 9

Implementation of Karplus Strong block diagram. [10]

elastic and a conducting plectrum connected to an Arduino Uno [5]. We tested the physical set-up by first making a mock-up out of cardboard, to test the length and width for user comfort. After prototyping the design, we made modifications and then constructed the final guitar out of 1/8-inch plywood, with carbon-impregnated elastic for strings and a conducting spectrum.

Second, we tested the note structures' animation and timing visually, to ensure that the notes would appear with ample time for the user to play each note correctly and enter the strum region of the TFT at the correct time. We caught timing and display errors and corrected the physics and display of the note movement accordingly.

Third, the song construction and note structures were tested by printing values to the TFT and debugging according to the specification. By using this method, we caught

```
DAC_B_data = (C_table[pluckIndexIn]>>16); // sample out, fix16 to int
lowpass_out = multfix16(damping, ((C_table[pluckIndexIn]) + (C_table[pluckIndexOut]))
    >>1);
C_table[pluckIndexIn] = multfix16(tune,(lowpass_out - last_tune_out)) + last_tune_in //
    replace current pluckIndexIn with new time-delayed averaged value with tuning
    constant
// all-pass state vars
last_tune_out = C_table[pluckIndexIn];
last_tune_in = lowpass_out;
```

LISTING 3

For the acoustic guitar string plucking, the basic filter for a Karplus-Strong implementation is used. An example of the C implementation of this filter is shown in this code.

Note	Natural Freq. (Hz)	Synth Freq. (Hz)	Error Percent
C4	261.6	258.0	1.4
D4	293.7	296.2	0.8
E4	329.6	333.3	1.1
F4	349.2	347.8	1.4
G4	392.0	400.0	2.0

TABLE 1

Frequency errors for each of the five notes

**FIGURE 10**

This QR code links to a YouTube video of the project.

timing errors when updating the note index of the song, and corrected them accordingly.

Fourth, we tested the sound generation using the oscilloscope on the DAC and listening to the undertones. By listening, we could change the fine tuning and dampening of the string generation to more closely resemble the frequency and sound of an acoustic guitar played with a high energy pluck.

Fifth, we tested the Bluetooth communication between the Bluetooth control station and the Adafruit Bluefruit LE UART Friend to the PIC32, and caught additional timing errors for updating the strum value [6]. To correct these issues, we modified the UART library by updating a timer flag.

MORE ON GAME PLAY

The final game-play runs constantly at a little over 16 frames/s and shows no tearing


or jumping in notes, scoring or timing on the display. The score factor updates consistently and accurately when notes are played correctly, an incorrect note is played, or there is no strum by the correct time. When playing a note that requires two strings at once, double score is added correctly.

The frequencies of our undertone for each of the five notes (C4, D4, E4, F4, G4) have errors of less than 1%, according to measurements from the oscilloscope. The errors for frequencies from the Karplus-Strong String Synthesis vary for any given note due to limitations in the sampling frequency caused by the use of integer left-shift registers. The synthesized frequency in relation to the notes' natural frequency and percent error are shown in **Table 1** [7].

Further error reduction is feasible by using fractional stages in the Karplus-Strong algorithm by tuning each individual note. We instead chose a fixed tuning constant to minimize error for all notes by audibly testing the sound generation.

There are many opportunities to implement further functionality in the game. Although we found a good medium for the user tolerance error in the game, adding a feature that allows the user to choose the difficulty would be possible in our current implementation. We could also extend the game to allow multiple-string chords to be played. This feature could be extended for playing multiple notes simultaneously, simply by adding and averaging the components of their Karplus-Strong buffers. There are also opportunities for extending the implementation of the mock guitar. While we used carbon-impregnated elastic to create a connection for the strings and strumming, an extension could allow fretting on the same string by measuring the resistance change when stretched.

The real bottom line for this project is that the game is exciting and engaging to play! We set out to create a fun video game designed to simulate the basic hand movements of playing a guitar. Our end result was a successful, functional video game that allows the user to play various note patterns for our implementation of "Ode to Joy" [8].

The scoring component and error noise create a competitive and educational feature that encourages the user to improve through audio-visual feedback. Scan the QR code in **Figure 10** to watch the game in action! A link to that YouTube video is also available on the *Circuit Cellar* article materials webpage. 

ABOUT THE AUTHORS

Jake Podell (jhp246@cornell.edu) is a Junior at Cornell University studying Computer Science in the School of Engineering. He is also a member of the CUAir, Cornell's Unmanned Aerial Vehicle project team, and will be interning at Facebook over the summer.

Jonah Wexler (jfw95@cornell.edu) is a Senior at Cornell University studying Electrical and Computer Engineering with a Dyson Business Minor for Engineers. Next year, he will be working at BlackRock in the Portfolio Analytics Group.

Jake and Jonah have been friends throughout college and have worked on various engineering projects with each other.

For detailed article references and additional resources go to: www.circuitcellar.com/article-materials
References [1] through [10] as marked in the article can be found there.

RESOURCES

Adafruit | www.adafruit.com

Microchip Technology | www.microchip.com

DO YOU WANT THE BEST ELECTRONICS DESIGN SOFTWARE?

The Alternatives

PROTEUS

User Friendly

Comprehensive

Integrated

Affordable

Danger

FEATURES

- Schematic Capture
- PCB Layout
- Gridless Autorouting
- 3D Visualization
- M-CAD Integration
- SPICE Simulation
- MCU Co-simulation
- Built in IDE
- Visual Programming

labcenter  **www.labcenter.com**
Electronics

Toll Free: +1 866 499 8184

Direct: 905 898 0665

Email: don.jackson@labcenter.com

MCU-Based IC Links USB to Legacy PC I/O

Versatile Solution

In PCs, serial interfaces have now been replaced almost completely by USB. But many of those interfaces are still used in control and monitoring embedded systems. In this project article, Hossam describes his ATSTAMP design. ATSTAMP is an MCS-51 (8051)-compatible microcontroller chip that can be connected to the USB port of any PC via any USB-to-serial bridge currently available on the market.

By
Hossam Abdelbaki

To interface a PC to the external world for control and monitoring applications, the computer requires an external I/O interface. Legacy ports such as parallel ports, serial ports, PS/2 or game ports were found in early PCs. Currently, USB replaces all those legacy ports and you rarely find them in modern PCs.

Compared to USB protocol, dealing with the serial port protocol is straightforward with regard to hardware and software. With that in mind, it's no wonder that considerable numbers of important communication modules—such as Wi-Fi, Bluetooth, GSM and GPS—have serial UART interfaces. Because it's such a widely used standard, the RS-232 serial port is still used in many types of industrial, communication and medical equipment.

Several companies manufacture USB-to-serial bridge ICs, which can be connected to USB ports and give the PC what is called a “virtual COM port” (or VCP). These companies include Silicon Labs (CP210x), FTDI Chip (FT232RL), Jiangsu Yuheng (CH34x) and Prolific Technology (PL2303). Each type has characteristic specs and features. Drivers for different operating systems are available from the manufacturers' websites.

Even though some microcontrollers (MCUs) do have built-in USB peripherals, using such MCUs may complicate the firmware and software. They also may require customized drivers to be developed. In this project, a Silicon Labs CP2102-based module is used as the USB-to-serial bridge. The module provides TX, RX, DTR, VCC and GND pins for connecting ATSTAMP with a PC's USB port via a micro-USB socket cable. The module I used is available from amazon.com—search

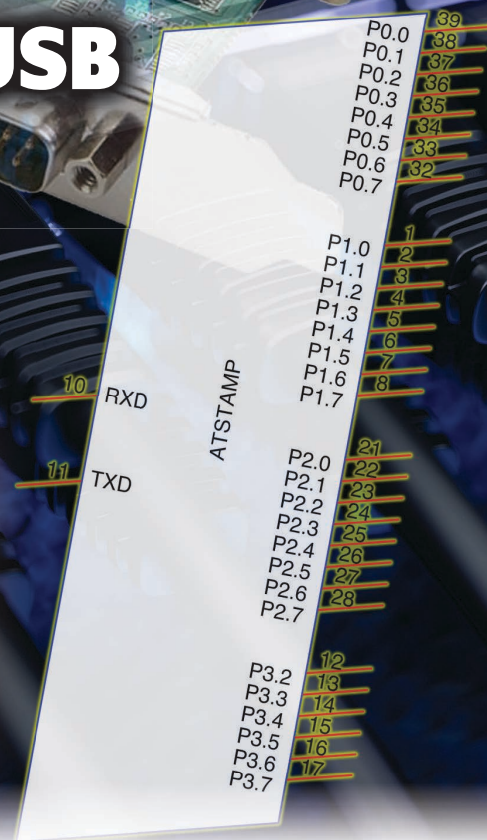
“Wingoneer CP2102 Micro USB to TTL.” The module is easy to plug into a breadboard without the need for any flying wires. Alternately, any similar module can be used, provided that it has a DTR pin for downloading firmware into the MCU, as will be discussed later in this article.

ATSTAMP has been used for several years in many educational and industrial projects. It has been used in microprocessor courses in interfacing and understanding the operation of peripheral ICs, such as 8255, 8259, 8253, 8251 and 6821. It was also used in embedded systems courses in interfacing with ICs working with different protocols, such as SPI and I²C. Operation of LCDs, keypads, ADCs and DACs has been also explored. Several types of EPROM and MCU programming kits also used ATSTAMP as the main processor.

ATSTAMP is well suited as a tool for debugging your design and verifying its functionality. By adding transistors, current drivers, relays, triacs and protection circuits, ATSTAMP can be used in a variety of industrial control and monitoring applications. An inexpensive industrial PC connected to ATSTAMP can replace expensive HMI (Human Machine Interface) screens and Programmable Logic Controllers (PLC). Any program written using the ATSTAMP-provided functions can be easily converted into embedded C language for any MCS-51-based MCU, which facilitates developing stand-alone embedded applications without the need of a PC connection.

HARDWARE DESIGN

The ATSTAMP's schematic is shown in **Figure 1**. ATSTAMP provides four ports—P0, P1, P2 and P3—that each can be



programmed as input or output. All port pins are individually programmable for input or output operation (except P3.0 and P3.1). In the schematic, an LED and a push button are connected to P1.0 and P2.0, respectively, to test the functions, as will be shown later. A prototype has been wired on a breadboard as shown in **Figure 2**. The ports are organized and bus oriented, so it's easy to use 8-way flat cable to connect the breadboard to your interface. It should be noted that the hardware is compatible with the following MCUs from Microchip Technology (formerly Atmel): AT89S51, AT89S52, AT89S53, AT89S8252 and AT89S8253. Because of the versatility of ATSTAMP—especially in the industrial field—a development board has been designed and fabricated as shown in **Figure 3**. The board also has been used as a general board for 8051-based faculty laboratory experiments. The schematic and design files for this general board are freely available from the author upon request.

ATSTAMP logic levels are 0 V and 5 V. Low current devices such as LED indicators can be controlled directly by making the output port pin sinks current (active low). Input devices such as push buttons can be connected to the port pin with a pull-up resistor. Figure 1 shows the connection of an LED and a push button to two I/O pins of ATSTAMP.

For controlling AC-DC loads working on higher voltages (currents), a relay or triac can be used, as shown in **Figure 4** and **Figure 5**. Note that clearing the IO pin activates the relay or the triac. If the input comes from an industrial process that operates on 24 V, an optocoupler may be used to interface with the ATSTAMP input pin (**Figure 6**).

DOWNLOADING THE FIRMWARE

ATSTAMP firmware is provided as a binary hex file (*astamp.hex*) on the *Circuit Cellar* article code & files download page. The firmware can be programmed into any 8051-based MCU from any manufacturer, including Microchip, Silicon Labs, STC and Maxim Integrated. Each company provides hardware and software programming tools for its MCUs. Personally, I prefer the Microchip (formerly Atmel) AT89S52 MCU because it is widely available, relatively inexpensive and comes in a DIP package.

The problem with Microchip 8051-based MCUs is that the latest available programmer from the manufacturer is based on the legacy Centronics Parallel Port. It is called AT89ISP. Some variants have a built-in bootloader (such as AT89C51ED2) and can be programmed via a serial port using Microchip's FLIP software. The developer should search for a USB clone programmer or a universal programmer for downloading the firmware into AT89S5x MCUs.

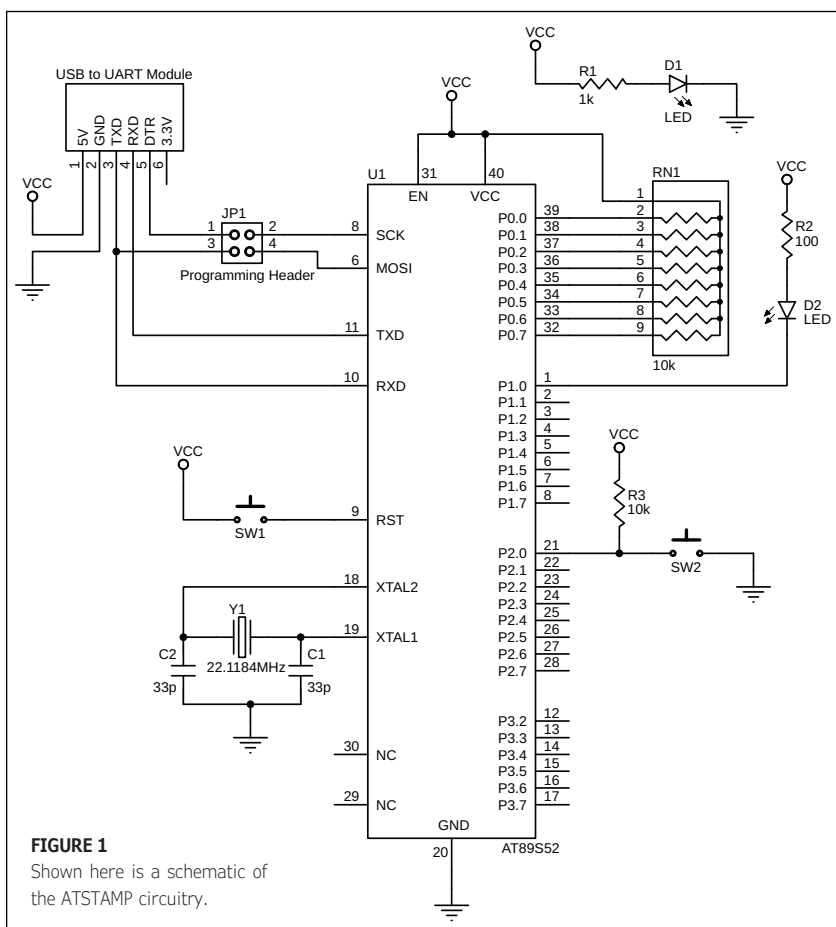


FIGURE 1
Shown here is a schematic of the ATSTAMP circuitry.

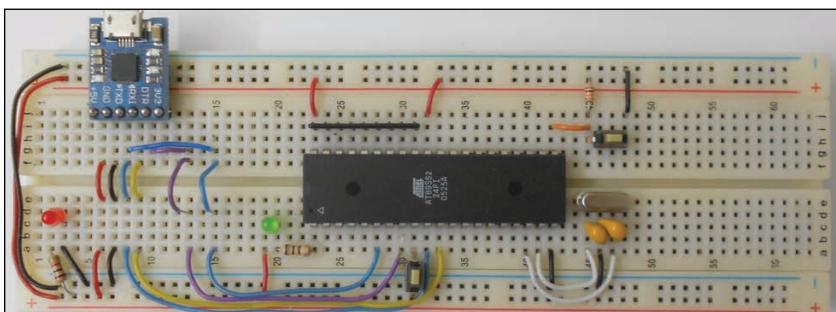


FIGURE 2
ATSTAMP breadboard prototype.

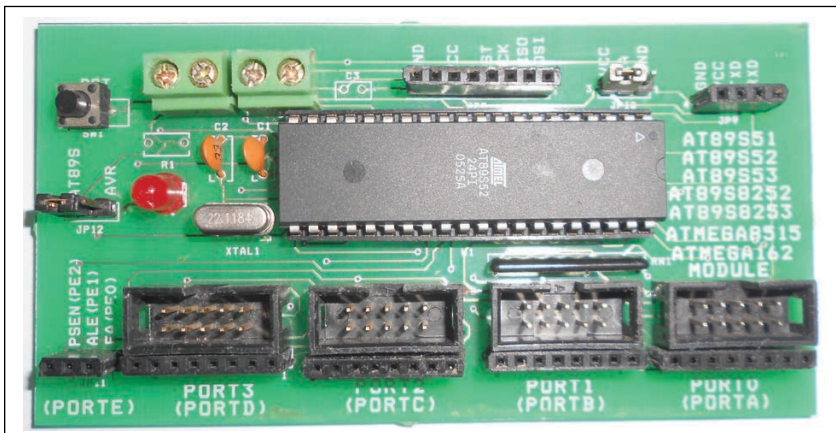
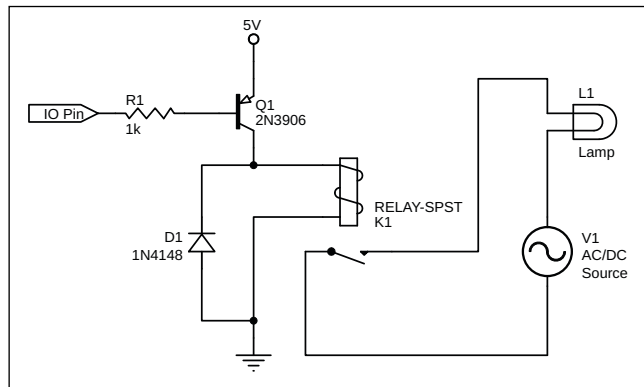


FIGURE 3
This is the ATSTAMP finished development board.

FIGURE 4
Controlling DC-AC



the PC enables you start your first ATSTAMP applications in less than an hour.

LOW LEVEL ASCII COMMANDS

The firmware of ATSTAMP is written in MCS-51 assembly language. ASCII control characters (commands) are sent to ATSTAMP with baud rate 115200, 7 data bits, no parity, one stop bit and no hardware handshake. It should be noted that each time a command is sent to ATSTAMP acting upon a certain I/O pin, the previous setting or value of that pin is overridden. ASCII commands can be sent directly from terminal emulator software, or from any device with a UART interface. Commands are made of one, two or three characters to perform different input and output port operations.

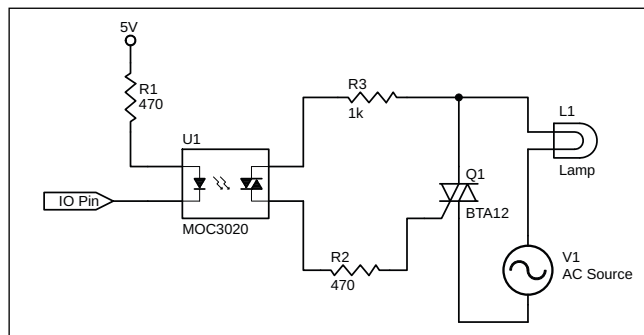
To keep the communication speed between the host PC and ATSTAMP as high as possible, no acknowledgement is returned to the host from ATSTAMP. Regarding the timing, setting or clearing a bit in a port may take around 0.5 ms—depending on the given crystal and baud rate. The ASCII commands with their arguments and return values are listed in **Table 1**.

Note that P3.0 and P3.1 are used for serial communication with the PC, so don't use them in any of the library functions. In fact, any command involving these two pins won't be recognized by the ATSTAMP firmware. If you write values to these two bits, commands will be discarded, and if you read them, ATSTAMP always will return a "1". Those functions can be converted directly into embedded MCS-51 assembly language or C language.

After connecting ATSTAMP to the PC's USB port via the CP2102 module, check the virtual COM port number associated with the module (from Control Panel > Device Manager). You will find it under Ports (COM & LPT)—assume it is COM1. Use a Terminal Emulator program such as "SerialPort Terminal," which is very efficient open-source software. Open the COM port with the settings described earlier. Now, if you send the ping command "T", ATSTAMP will reply by "HO" meaning it's alive. To switch on the LED, clear P1.0 by sending "n00". To switch it off, set the same bit by sending "N00".

To set P1 to "FF", send "XFF", and to clear it to "00", send "X00". To check the status of the push button connected to P2.0, send "G00". You should get "01", indicating the button is released. Now press the button and issue the command again, and you will receive "00". Note that to program a pin as input, you should set this bit first. Also, to program a port as all inputs, you should send "FF" to it first. This is a common practice for MCS-51 MCUs. You can practice all the commands in Table 1 to verify the proper operation of ATSTAMP.

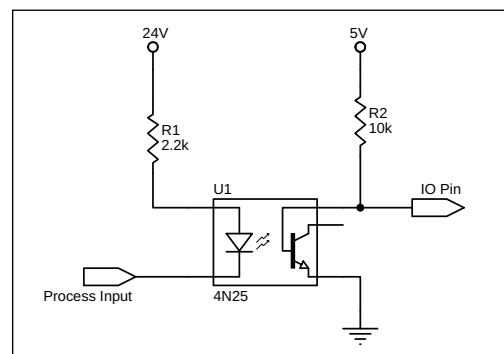
FIGURE 5
Controlling AC



I solved this problem with an unconventional technique by using the same USB-to-UART module for downloading the firmware into the AT89S52 MCU. While programming the firmware, connect the TX and DTR pins of the module to MOSI (pin 6) and SCK (pin 8) pins of the MCU, respectively, and run the provided program, *ATSTAMP_Downloader.EXE*. I wrote this program to bit bang the ISP pins of the MCU. The binary hex file is embedded into the .exe file.

Alternately, you may program the controller using a ready programmer, if you have one available. The software will ask the user to press the RST button while programming and release it when downloading is completed. A video named *atstamp_download.mp4* is provided on the *Circuit Cellar* article code & files download page. The video shows how the firmware is downloaded into the AT89S52 MCU. Using the USB-to-TTL module for programming the firmware and for communication with

FIGURE 6
Reading inputs from 24 V



C LIBRARY FUNCTIONS

On one hand, terminal emulator programs provide a nice way to check the low-level ASCII commands accepted by ATSTAMP. It's a good idea to get familiar with them so you can develop your own functions and customized applications. On the other hand, providing high-level software library functions can be used more efficiently and make you focus on the application. Currently, C++ Builder Starter Edition from Embarcadero Technologies is used for developing ATSTAMP PC-based programs. It is a fully featured free C++ IDE for creating native cross-platform applications.

Using C++ Builder software looks weird, and you may ask: Why aren't I using the fancier Visual Studio C# environment? Truth is I've been working with Borland products for three decades. Although Borland was acquired by Inprise, CodeGear then finally by Embarcadero, C++ Builder is the best environment for me because of its vast speed and

optimization. For years many embedded systems experts have known Borland C and C++ Builder software as the optimum choice. The starter edition of C++ Builder can be downloaded for free from the Embarcadero website. On the *Circuit Cellar* article code & files webpage, a video named *embar_starter.mp4* is provided for demonstrating how to download and install C++ Builder Free Starter Edition. The current version is 10.2 Tokyo. The video also shows how to start your first C++ Builder Console application.

The ATSTAMP.LIB library provided contains the definition of the main functions used for communication between the PC and ATSTAMP controller. There are seven high-level functions, which are each described here:

1. OpenCom

```
bool OpenCom(unsigned int ComPortNumber, int
BAUDRATE);
```

Port Read Commands (Check SerialInput C Function)			
ASCII Command	Argument	Action	Return
w	None	Reads P0	Hex value presented on P0
x	None	Reads P1	Hex value presented on P1
y	None	Reads P2	Hex value presented on P2
z	None	Reads P3	Hex value presented on P3
Port Write Commands (Check SerialOutput C Function)			
ASCII Command	Argument	Action	Return
W	HH	Sends the hex value HH to P0	None
X	HH	Sends the hex value HH to P1	None
Y	HH	Sends the hex value HH to P2	None
Z	HH	Sends the hex value HH to P3	None
Port Bit Setting Commands (Check setb C Function)			
ASCII Command	Argument	Action	Return
L	bb	P0.bb = 1	None
N	bb	P1.bb = 1	None
Q	bb	P2.bb = 1	None
S	bb	P3.bb = 1	None
Port Bit Clear Commands (Check clr C Function)			
ASCII Command	Argument	Action	Return
l	bb	P0.bb = 0	None
n	bb	P1.bb = 0	None
q	bb	P2.bb = 0	None
s	bb	P3.bb = 0	None
Port Bit Read Command (Check GetBit C Function)			
ASCII Command	Argument	Action	Return
E	bb	Reads bit bb in P0	P0.bb
F	bb	Reads bit bb in P1	P1.bb
G	bb	Reads bit bb in P2	P2.bb
H	bb	Reads bit bb in P3	P3.bb
Ping Commands (No relevant C Function)			
ASCII Command	Argument	Action	Return
T	None	Check ATSTAMP	HO

TABLE 1
ASCII commands for ATSTAMP

ABOUT THE AUTHOR

Dr. Hossam Abdelbaki is a faculty member at Department of Electrical Engineering, Faculty of Engineering, Alexandria University, Alexandria, Egypt. He has taught embedded and reconfigurable systems since 1995. He served as head of research and development for the Subsea Petroleum Services company, where he supervised building and delivery of three FPGA based Dynamic Positioning (DP) offshore marine vessels. Dr. Hossam also is the CTO of Technoenergy Company, which specializes in designing LED Lighting and energy-saving products. He can be reached by email at: hossam@technoenergy-eg.com or hossam.mostafa@alexu.edu.eg

The `OpenCom` function accepts a COM port number (physical or virtual) and a baud rate (which is fixed at 115200). You may get the COM port number from the Windows Device Manager, as discussed previously. You may also get it automatically in your code by reading it from the registry. This function is responsible for opening the PC COM port and establishing reliable communication. The function returns true (1) if the communication is established, and returns false (0) if there is an error in communication.

When there are no problems in communication, your program should proceed normally. If you enter a non-existing COM port number in the `OpenCom` function, you will get the following message: *Fatal Error: Cannot open COM port.* Also, this function pings ATSTAMP to check if it is connected or not. If ATSTAMP is not connected or there is any problem with its hardware, you will get the message: *ATSTAMP is not detected, Check the hardware and try again.* If you are sure that hardware is OK, press the RST button and run the program again.

Hardware problems may result from any of the following:

- Either VCC or GND is not connected to the ATSTAMP circuit.
- The serial cable (or USB cable) is not connected to the PC or the ATSTAMP circuit.
- The value of the quartz crystal is different than 22.11884 MHz

- Pin 31 of ATSTAMP is not connected to VCC.
- The crystal or the capacitors are not connected properly or have different values.
- TXD and RXD pins of ATSTAMP are swapped when connecting the circuit.

In any of the above cases, try to solve the problem and run your program again.

2. CloseCom

```
void CloseCom();
```

This function is responsible for closing the previously opened COM port, so that it can be released and used in any other application, such as the serialTerminal. This function should be invoked at the end of your C program. If you forget to include it in your program, the COM port will close automatically when you close the application.

For the following five input/output control functions, the `portnum` argument can be P0, P1, P2 or P3, and the bit argument can be 0 to 7. Remember that P3.0 and P3.1 cannot be used.

3. setb

```
void setb(unsigned int portnum, unsigned char bit);
```

This function sets any bit in any of the four ports. For example, if you want to set bit 5 of port 2, then you call the function `setb(P2, 5)`. `setb` is also used when you want to configure a bit in port as input. For example, if you want to read the logic level on pin P0.4, then you should first call `setb(P0, 4)` to make this pin input, then you can read its value using `GetBit` function. The equivalent with MCS-51 would be `setb P0.3` (in assembly language) or `P0^3 = 1` (in C language).

4. clr

```
void clr(unsigned int portnum, unsigned char bit);
```

This function clears any bit in any of the four ports. For example, if you want to clear bit 2 of port 1, then you call the function `clr(P1, 2)`. The equivalent with MCS-51 would be `clr P1.2` (in assembly language) or `P1^2 = 0` (in C language).

5. GetBit

```
bool GetBit(unsigned int portnum, unsigned char bit);
```

This function reads any bit in any of the four ports. For example, if you want to read the logic level on bit 3 of port 0, then you call the function `x = GetBit(P0, 3)`. The equivalent with MCS-51 would be `move c, P0.3` (in assembly language), where `c` is the carry flag or `x = P0^3` (in C language).

6. SerialOutput

```
void SerialOutput(unsigned int
```

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

RESOURCES

Embarcadero Technologies | www.embarcadero.com

FTDI Chip | www.ftdichip.com

Jiangsu Yuheng | www.wch.cn

Microchip Technology | www.microchip.com

Prolific Technology | www.prolific.com.tw

Silicon Labs | www.silabs.com

STC | www.stcmcu.com


```
portnum, unsigned char data);
```

This function outputs the value given by the data argument as logic levels to any of the four ports. For example, if you want to output the hexadecimal value F3 to port 2, then you call the function `SerialOutput(P2, 0xF3)`. The equivalent with MCS-51 would be `move A, #0F3H` (in assembly language) or `P2 = 0xF3` (in C language).

7. SerialInput

```
unsigned int SerialInput(unsigned
int portnum);
```

This function reads the logic levels on any of the four ports. For example, if you want to read the levels on port 1, then you call the function `x = SerialInput(P1)`. The equivalent with MCS-51 would be `mov A, P1` (in assembly language) or `x = P1` (in C language). As discussed earlier, to read a port, you should set all its bits first, so use `SerialOutput (P1, 0xFF)` before invoking the `SerialInput` function to read P1.

SAMPLE ATSTAMP APPLICATION

To start your first ATSTAMP interface application, first prepare a project folder (for example, name it `atstamp_io`). Copy the provided files, `atsamp.h` and `atstamp.lib`, to the folder project. Start C++ Builder and select Project > Add New Project > Console Application. Select File > Save Project As, save the project in the folder under same name (`atstamp_io.cbproj`). Select Project > Add to Project, point to `atstamp.lib` file. Copy and paste the program listed in **Listing 1** to the source file window. You can also get it as the Listing 1.txt file from the *Circuit Cellar* article code & files download page. Select Project > Build `atstamp_io` then Run.

The program infinitely checks the status of the push button connected to pin P2.0, and when it goes low, it flashes the LED connected to pin P1.0 five times. Note the operation is fully controlled by the commands sent to the ATSTAMP by the running program. The `atstamp_io` project is provided as a template that can be tuned to your application.

A video named *atstamp_app.mp4* is provided to illustrate creating the sample software application, and shows how it runs on the prototype circuit. Download it from the *Circuit Cellar* article code & files download page. Although the given sample application is console-based, with little a practice you can build your own GUI Windows-based applications.

CONCLUSION

ATSTAMP is a great tool for embedded system development. It may save you lot of programming and firmware uploading time. Actually, a lot of applications and products


```
// Checks status of push button connected to P2.0
// (active low)
// When pushed, LED connected to P1.0 flashes five times
#include "ATSTAMP.h"
#include <stdio.h>
#include <conio.h>
#include <windows.h>
int main()
{
    if(!OpenCom(1, 115200))
    { exit(0); }
    while(1)
    {
        if(!GetBit(P2,0))
            for(int i =0;i<5;i++)
            {
                clr(P1,0);
                Sleep(300);
                setb(P1,0);
                Sleep(300);
            }
    }
    return 0;
}
```

LISTING 1

Shown here is sample C application program for the ATSTAMP.

can be based on ATSTAMP. It has been used for more than a decade in various applications and proved to be useful and reliable. As an example, consider the development of a full-featured stand-alone digital IC tester for TTL 74xxx and CMOS 40xxx families. Without ATSTAMP you may need thousands of upload and erase cycles of your target MCU until you get it work satisfactorily. In contrast, using ATSTAMP requires no MCU programming in the development phase. Just write your test patterns on Embarcadero C++ program, and run it on the fly to test applying the patterns to the target IC and getting its response. Once you settle on the logic, you may convert your Embarcadero C++ code to the C code of your target MCU embedded compiler and upload it once. You're limited only by your imagination.

Finally, it should be noted that during reading inputs of ATSTAMP, the PC control program constantly polls the ATSTAMP to check level changes. This technique wastes CPU time and generates a lot of traffic over the serial link. It would be more efficient if the ATSTAMP firmware detected input hardware changes and automatically sent a notification to the PC software, which would trigger a call-back function. If this were the case, however, ATSTAMP firmware and the PC software library implementation would be more complex.

Using an MCU with enhanced architecture—such as AVR or Arm—would open the possibility of including functions other than digital I/O. For instance, reading analog values and generating analog voltages would be advantageous. 

Smart Pet Collar Uses GPS and Wi-Fi

PIC32-Based Design

Microchip's PIC32 MCU has proven effective for myriad applications, so why not a dog collar? Learn how these Cornell graduates built a GPS-enabled pet collar prototype. The article discusses the hardware peripherals used in the project, the setup and the software. It also describes the motivation behind the project, and possibilities to expand the project in the future.

By

Vidya Ramesh and **Vaidehi Garg**

People have a multitude of ways of staying in touch with each other. These days we can call, text, email, instant message, Snapchat or video call anyone with cell service or Wi-Fi—and write letters to anyone who doesn't. It's important for human beings to be in contact with one another, but what about their pets? There aren't as many ways of checking in on your furry friends while you're away from them. As a final project for our class on "Digital Design Using Microcontrollers" at

Cornell University, we built a wearable smart device that can be attached to a pet collar. Equipped with Wi-Fi and GPS capabilities, the collar lets you follow Lassie's adventures from wherever you are.

The system runs on Microchip Technology's PIC32 microcontroller (MCU) and is outfitted with a small army of peripherals from Adafruit. These include the Ultimate GPS breakout board, the Huzzah ESP8266 breakout board and the 9-DOF (degrees of freedom) Absolute Orientation IMU Fusion breakout board. The device displays real-time data to a webpage that the user can access over the Internet. A portable 5-V USB battery pack is used to power the system. The block diagram in **Figure 1** presents a high-level overview of the full system.

The PIC32 is mounted onto a small development board designed by former Cornell student, Sean Carroll [1]. The board takes in 5-V and regulates it down to 3.3-V, which powers the MCU. The GPIO pins and the power outputs on PIC32 are brought out onto headers, so that the board can easily be mounted onto a breadboard along with the peripherals. **Figure 2** shows the schematic for the entire system, and **Figure 3** is a photo of our prototype.

The combined surface area of all components—excluding the battery pack—can easily fit onto an average dog collar. The weight of any collar being worn by a mammal is ideally within 5% of its total body weight [2], a requirement that this design meets for most

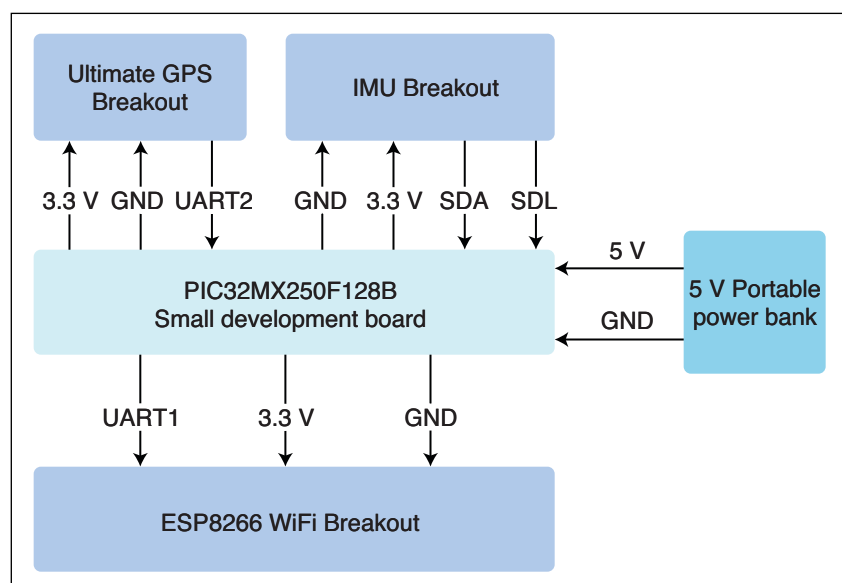


FIGURE 1

A high-level block diagram of the system for a pet collar with Wi-Fi and GPS capabilities

dogs. The USB battery pack can be replaced with a smaller battery pack or a rechargeable battery, if desired. For testing purposes, the battery pack was a convenient, if bulky, solution due to its high charge capacity. The rendering in **Figure 4** shows potential placement of the MCU and peripherals in a flexible collar. Important design aspects to be considered here include appropriately enclosing all electronics to prevent contact with water, fur or skin.

DESIGN ELEMENTS

Let's take a look at each major element of the design one by one.

Software: Our software on the PIC32 consists of three threads and a main function. Each thread is dedicated to one peripheral and handles the communication with the corresponding module. The threads were implemented using the Adam Dunkels' popular lightweight protothreads library for C [3], modified for PIC32 by Dr. Bruce Land [4]. The

GPS thread spawns a child thread to read the serial input buffer for the PIC's UART2 channel, and parses the NMEA sentence received if it is in valid GPS message format. The IMU (inertial measurement unit) thread uses a I²C helper function to read accelerometer values in the X, Y and Z directions. The values are used to determine whether the dog is active, to set flags and to update variables. Finally, the Wi-Fi thread encodes the data collected from the other modules into a string and communicates it to the Espressif Systems ESP8266 by spawning a child thread. The child thread puts the data into the serial output buffer on the PIC32's UART1 channel. The main function takes care of configuring and enabling I²C1 on the PIC32, configuring the IMU, and initializing and scheduling the threads.

GPS: We used the Adafruit Ultimate GPS breakout board to collect geolocation data, such as latitude and longitude coordinates and the time in EST. The board has a MediaTek Labs MTK3339 chipset, a high-quality and

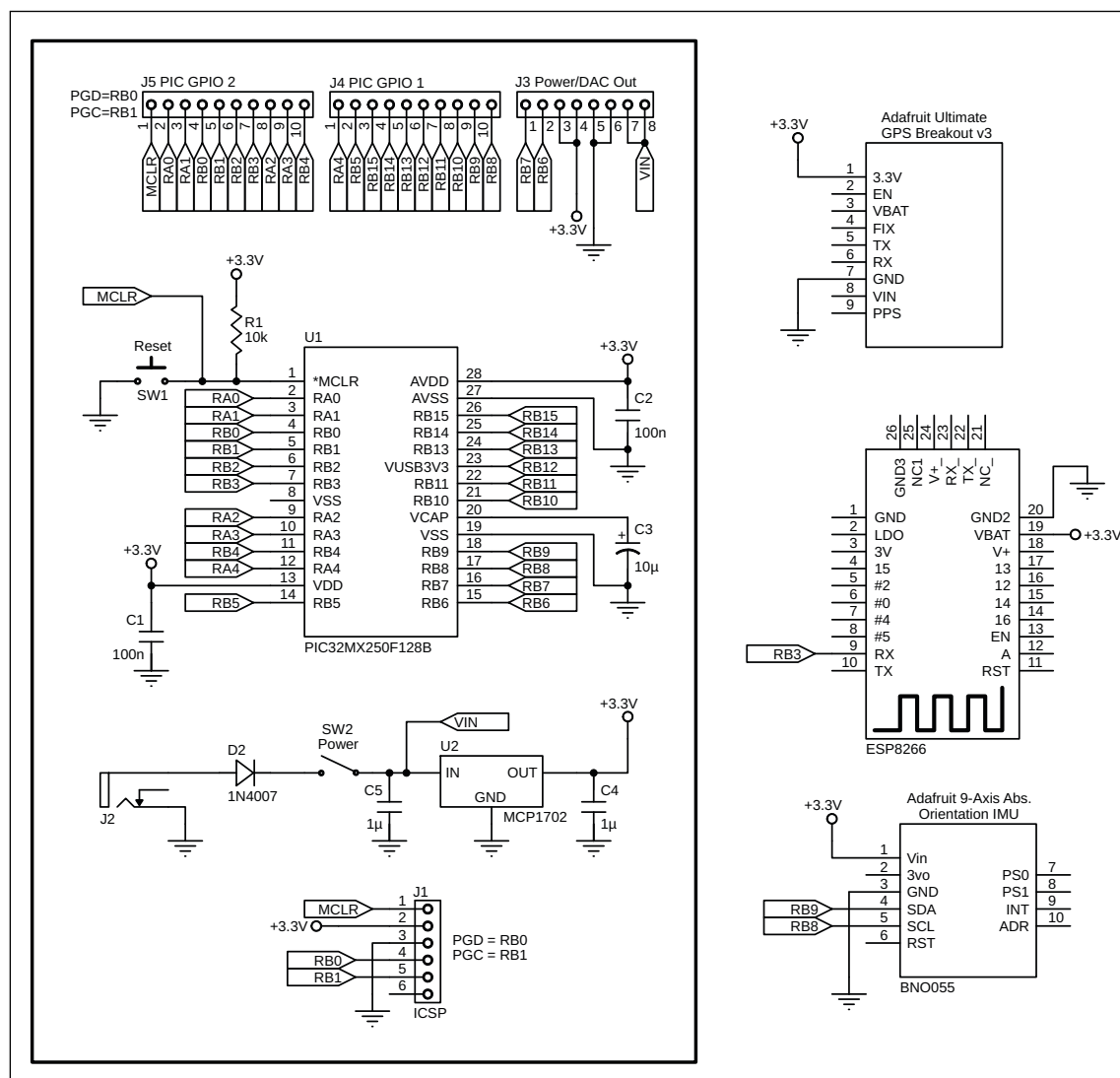
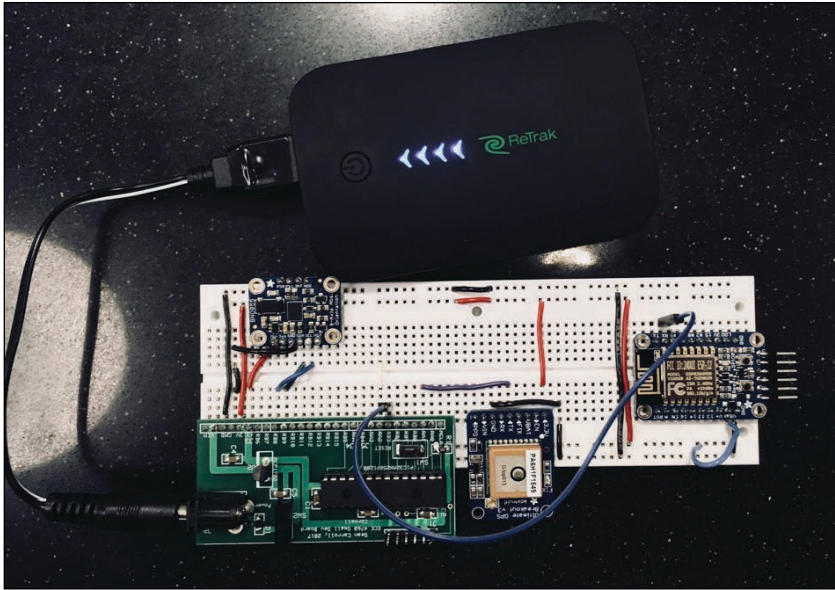
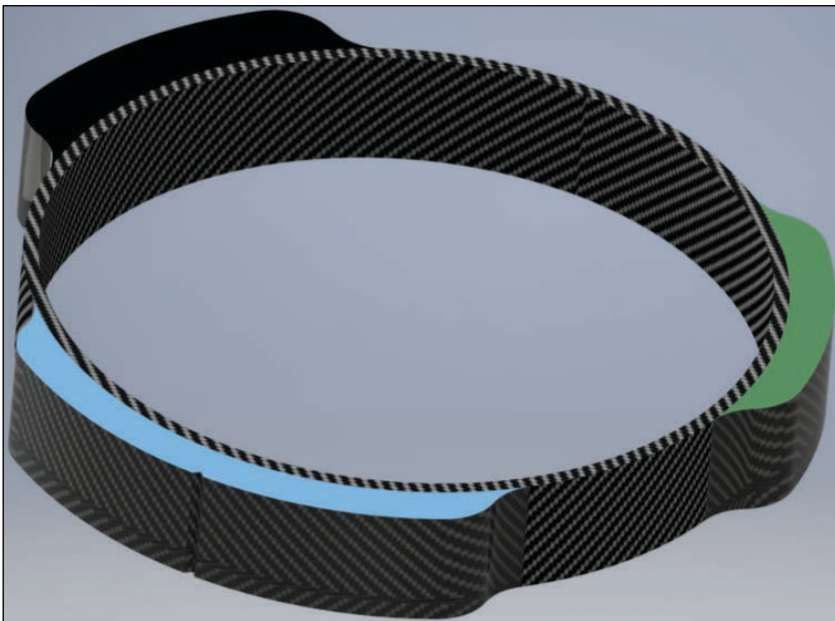


FIGURE 2

Shown here is a schematic of the system, designed in ExpressPCB using Sean Carroll's schematic [1] of the small PIC32 development board.

**FIGURE 3**

This is our prototype on a breadboard, including the battery pack used for testing.

**FIGURE 4**

Dog collar concept design (Image courtesy of Abhishek Paramasivan)

ABOUT THE AUTHORS

Vidya Ramesh received her B.S and M.Eng degree in Electrical and Computer Engineering from Cornell University in 2018. Her interests include hardware acceleration, computer architecture and embedded systems. She will begin working as a Design Verification Engineer at Apple in March.

Vaidehi Garg graduated from Cornell University in May 2018, with a major in Electrical and Computer Engineering and a minor in Computer Science. She is currently working for Acacia Communications as an Associate ASIC Design Verification Engineer. She is interested in computer architecture, embedded systems and robotics.

low-power GPS module able to track up to 22 satellites and perform location updates at a rate of up to 10 Hz. The module sends a steady stream of NMEA data to the PIC32 via UART2 at the default baud rate of 9,600 bps. The sentence is transmitted in \$GPGGA format, which refers to the Global Position System Fix Data format [5]. Once the PIC32 confirms that the NMEA sentence is in the correct format, it parses it and extracts the desired values from it. We powered the module using 3.3 V from the PIC32 development board.

The UART functions used for communication were adapted from Junyin Chen's and Haonan Liu's project, "GPS Navigator for Runners" [6], which used the same GPS module. The GPS thread spawns a child thread, which reads and parses the UART2 serial input buffer, using functions defined in the GPS UART header file. The child thread yields to other threads until data are available in the serial input buffer. Once a byte is available, it reads it and stores it in a character buffer. If the character received is a newline character, '\n', it signals the end of the string, and the buffer is zero-terminated. Otherwise, the buffer index is incremented, and the character received is added to the buffer. After the entire NMEA sentence has been collected in the buffer, it is verified and parsed to obtain the required information using a simple state machine.

IMU: To give the users a general idea of whether their pet was active, a 9-DOF BNO055 IMU was used to detect the movement of the collar. The IMU interfaces with the PIC32 using the I²C protocol. It outputs accelerometer, gyroscope and magnetometer readings that can be used to generate measurements for yaw, pitch and roll for the device on which the IMU is mounted. We had initially planned to use the IMU to facilitate step counting on the device; however, we soon realized that a device mounted on a pet's collar would not be able to provide data accurate enough for such an application. Moreover, counting a pet's steps would also have been heavily dependent on the type of animal, and would have required a more in-depth analysis of what motions would count as a "step" for that particular pet.

With all that in mind, we chose instead to use the accelerometer on the IMU to indicate whether the device was in motion. A rough velocity calculation was implemented using the accelerometer readings. A nonzero velocity value indicated that the device was in motion, whereas the velocity was consistently zero when the device was at rest.

To receive data from the IMU over I²C, we used macros and helper functions from a previous ECE 4760 project by Ashley Xue, Jessica Chen and Adam Chung [7]. The header

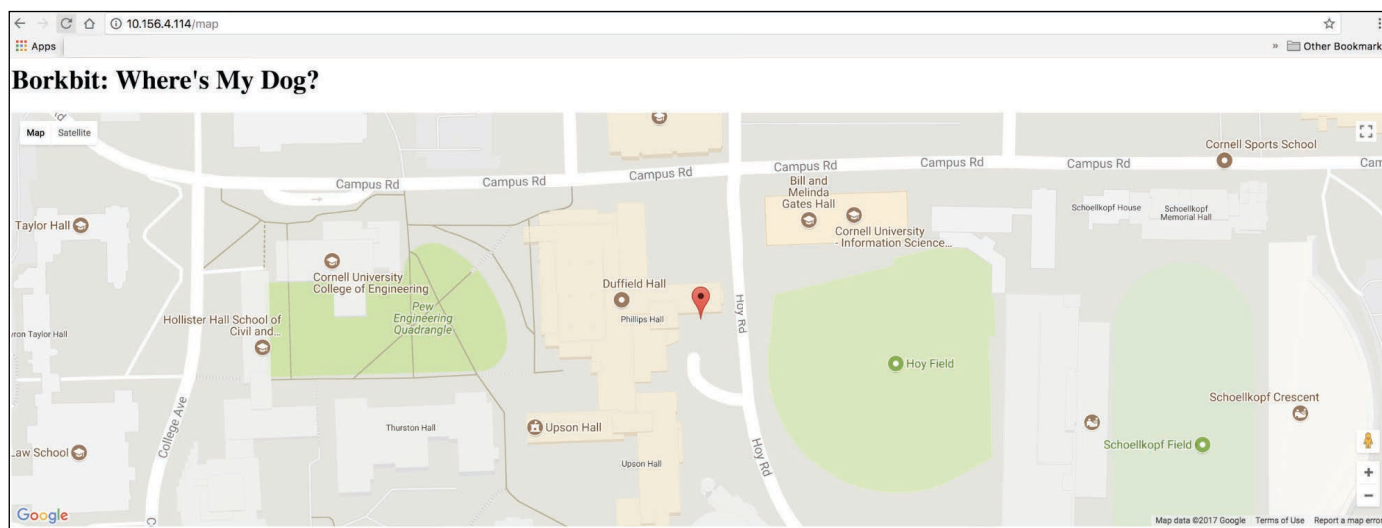


FIGURE 5

This is a screenshot of the map on the webpage when the device was located in 232 Phillips Hall at Cornell University.

file `i2cIMU.h` defines the device address and the different data registers on the IMU, and the helper function `i2c1ConsecutiveRead()` is used to read from two consecutive registers on the IMU. It does so by taking in a start register address and the number of registers to be read (which is always 2). It returns a "short" (a short is a 16-bit data type value), with the value from the first register stored in the LSB and the value from the second register stored in the MSB (most significant bit).

Wi-Fi: The Adafruit ESP8266 Huzzah breakout board was used to provide Wi-Fi connectivity to the device. Like the GPS module, the ESP8266 communicates with the PIC32 through UART. The PIC32 runs threads that communicate the parsed data from the GPS module and IMU to the ESP8266 by writing to the UART1 buffer. This is done by spawning a child thread that uses a helper function to write data to UART1 serial output buffer. This thread was implemented with the help of sample code from the Cornell ECE 4760 course website [8] and the helper function was adapted to use the UART1 channel instead of UART2. The ESP8266 was programmed using the Arduino IDE. This is discussed in further detail in the next section.

DISPLAYING DATA

To display all the data generated in a centralized location, we first had to collect it from the different peripheral modules and transmit it to the Wi-Fi module. To make our design useful, we needed to use the ESP8266's abilities to display the data to a webpage. Programming the ESP8266 can be tricky, but as with most Adafruit components, there are plenty of handy guides out there to help the process along.

Adafruit's Huzzah breakout board ships with

NodeMcu's Lua interpreter pre-programmed onto it, but it can also be programmed using the Arduino IDE. We decided to use the latter, because we were familiar with the Arduino interface and there were many more libraries available online that supported it. Neither of us had any prior experience with web development or setting up clients and servers using an MCU, but after some tinkering around we were able to connect the ESP8266 to a Wi-Fi network, start a server and write HTML code to a local webpage. The program uses the Adafruit ESP8266 libraries `ESP8266Wi-Fi` [9] and `ESP8266WebServer` [10] and Arduino's `WiFiClient` [11] to handle the backend processes of setting up a web server and client.

Our Arduino sketch reads incoming serial data from the PIC32 and saves it to different HTML files. One such file contains JavaScript for a map rendered via the Google Maps API [12]. The other HTML file simply displays raw data received from the PIC32. The ESP8266 starts a web server with two subdirectories to represent the two pages needed to display the data. The HTML files are communicated over the local server, and can be viewed on a Wi-Fi-enabled device connected to the same Wi-Fi network as the ESP8266 module.

After more experimentation with the ESP8266 module, we enabled it to scan through multiple Wi-Fi networks in a predetermined order, and connect to the first one available. We were also able to use the module to write dummy data to a remote webpage. We hope to extend the functionality of our project to include writing to a remote webpage in the future.

TESTING AND RESULTS

We verified device functionality by checking that the data returned from each

TABLE 1

Listed here are GPS coordinates vs. coordinates from the Google Maps website and the percentage error.

Location		GPS Module	Google Maps	% Error
Phillips Hall	Latitude	42.444415	42.444538	0.00028978
	Longitude	-76.481758	-76.481725	0.00004314
Gates Hall	Latitude	42.2677867	42.444940	0.41737201
	Longitude	-76.481758	-76.481192	0.00074005
Eddygate Park Apartments	Latitude	42.441200	42.4448110	0.00850751
	Longitude	-76.485900	-76.486122	0.00029024

individual peripheral were valid within a certain margin of error. For instance, when our GPS module indicated that we were in the middle of Europe, we knew there was something wrong. But when it briefly seemed to think we were on the other side of the street, we were willing to let it pass. Similarly, the activity level output by the IMU was verified by ensuring that the output was zero when the device was at rest and nonzero otherwise—where the nonzero value represented a crude velocity calculation.

The accuracy of the GPS was further verified using the Google Map embedded into our webpage. The latitude and longitude data received from the GPS was plotted onto the map and displayed. By comparing this map with our current location using the Google Maps website, we were able to confirm whether we were actually at the location indicated by the GPS. We tested our device around different parts of the Cornell campus. **Figure 5** is a screenshot of the Google Map our device produced when we tested it indoors in the Undergraduate Lounge in Phillips Hall.

A map of the various rooms in Phillips isn't included in this article, so you'll have to take our word that the lounge actually is by the point indicated. The marker on the

map actually lines up very closely with the windows of the lounge, which is where the device was set up. To calculate the percentage error, we recorded the latitude and longitude coordinates output by the module for several other locations, and compared them with the corresponding coordinates obtained using Google Maps. Our results are summarized in **Table 1**. The relatively large error for the Gates Hall latitude entry was most likely due to the measurement being taken indoors.

The device activity was detected using accelerometer readings as described in previous sections. This provided a dependable indicator of motion to convey whether the device was in motion or stationary. We tested this functionality by walking and running with the system, and monitoring the raw data displayed on our webpage. An example of our readings can be seen in **Figure 6**. The small but nonzero velocity value indicates that the device was in motion.

FUTURE ADDITIONS

Our design packs a punch because of its small, but powerful and reliable hardware. With a pocket-sized USB battery pack, it can easily be attached to the collars of pets weighing 10 pounds or more. However, the usability of the current iteration of our design is fairly limited. This is because our design requires the Wi-Fi modules on the collar and the pet owner to be connected to the same Wi-Fi network. This means that the pet owner can only monitor his/her pet if they are within the Wi-Fi range—for instance, if the pet is in a different room in the house. The system's functionality could be improved greatly by enabling the Wi-Fi module to write to a remote domain, which the user could access from any Wi-Fi network. This would allow pet owners to view their pets' data from any location and on any Wi-Fi network. We had some success experimenting with these features, but discovered that full

For detailed article references and additional resources go to:

www.circuitcellar.com/article-materials

References [1] through [13] as marked in the article can be found there.

RESOURCES

Adafruit | www.adafruit.com

Espressif Systems | www.espressif.com

MediaTek Labs | labs.mediatek.com

Microchip Technology | www.microchip.com

NodeMcu | www.nodemcu.com

integration would require time beyond our project deadline.

Another addition required to make the system practical and marketable would be the ability to configure it from a user standpoint. For instance, a potential user would want to pre-program the device to scan for a few Wi-Fi networks. They might also want to configure what data they want to receive about their pets, and the frequency at which they would like to receive the data. Further, a user login could be implemented to allow only verified users to access the data.

Other highly requested functionalities include heart rate monitoring and step counting. While working on our project, we were able to interface successfully with the hardware required for both these features. However, to make the collected data truly useful, we would have to examine aspects of our design much more closely. We would have to identify how the movement patterns detected by an IMU located on a collar of an animal can be translated into steps, and determine ideal placement of a pulse measurement device to pick up heart rate accurately through fur.

Last but not least, our final design set us back less than \$85, including a \$20 battery

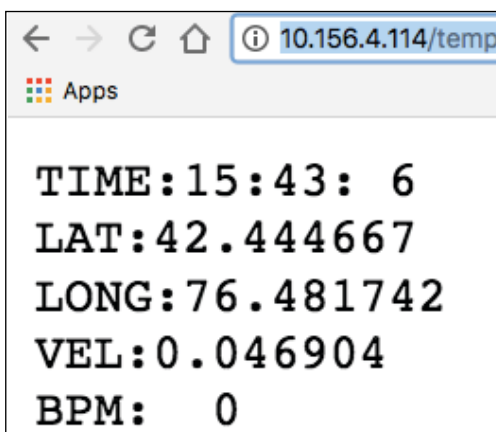



FIGURE 6

Raw data obtained from the PIC32 MCU as displayed on the webpage, after walking and running with the system. The nonzero velocity value indicates movement.

pack. The closest competitor currently on the market—that we were able to identify—is the Link AKC Smart Dog Collar [13] which costs from \$80 to \$100 depending on the size, and requires a monthly payment for a service plan. This leads us to believe that our design, once extended to include the changes mentioned above, could be a marketable product if manufactured. 

Authors' Note: We thank Divya Gupta for helping build the project, and Professor Bruce Land for his time and guidance.

When it comes to robotics, the future is now!

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

With this book, you'll learn about:

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



ADVANCED CONTROL ROBOTICS
HANNO SANDER

Get it today at
ccwebshop.com



Sales@ezpcb.com

Ez PCB

Welcome

www.ezpcb.com

One-Stop PCB & PCBA Turnkey Service



PCB Online Calculator
No Need to Register
Instant Quote & Pay



1 To 40 Layers
Prototype to Mass Production
Amateur to Professional



Prototype Start At \$5/PC
2L 4"x4" each
Free Shipping

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

Formal Flow for Automotive Safety

Bulletproofing Car Design

Formal tools can prove that a fault is safe, residual or latent—and they can quickly determine a worst-case/best-case diagnostic coverage without time-consuming analysis. In this article, Doug presents an improved formal verification flow that uses sequential logic equivalency checking (SLEC) to reduce the number of faults while simultaneously providing much higher quality of results.

By
Doug Smith,
Mentor, a Siemens Company

The ISO 26262 automotive functional safety standard [1] requires performing random hardware failure analysis of electronic devices for automotive applications. The purpose is to test that a component's safety functions and safety mechanisms work correctly. This is done after a design has been verified by randomly injecting faults and testing to see if the design still works correctly or if the faults are controlled or detected by a safety mechanism. This process of injecting faults and determining results is referred to as a fault campaign.

Fault analysis has been around for years, and many methods have been created to optimize evaluation of hundreds of concurrent faults in specialized simulators. However, there are many challenges in running a fault campaign. In most designs, the number of faults to inject is very large. Simulating thousands of tests can lengthen project schedules and compete for resources. Creating a list of faults may not be trivial. Determining if tests activate and propagate faults can be labor intensive and difficult. Dealing with third party IPs and constraining them correctly under fault conditions may be tedious. Injecting multiple faults for latent failure analysis and using multiple failure modes (stuck-at0, stuck-at1, transient) only doubles and triples the effort. Lastly, gathering results for metric calculations typically requires some form of automation.

Fortunately, there is a better method for running a fault campaign—*formal sequential logic equivalency checking* (SLEC). Mentor, a Siemens Business, has developed a flow, called *SafeCheck*, which not only automates every step of the process, but also solves many issues associated with running a fault campaign. This formal platform-based flow allows engineers to focus on automotive functional safety goals, architecture, and requirements, without getting bogged down in the details of the fault campaign.

MINIMIZING FAULTS

As a starting point, an engineer uses a failure modes, effects, and diagnostic analysis (FMEA) or creates a simple spreadsheet (CSV file) listing the safety functions, their safety mechanisms (both primary and secondary), any constraints to apply (such as disabling DFT when testing) and the *fault detection interval* (FDI) [2] required by the design's hardware requirements.

The sheer number of potential faults in a design makes the fault campaign process overwhelming. Using a formal tool to prune the number of faults separates the wheat from the chaff and makes the process more manageable. Because formal tools perform their own synthesis of the design into a Boolean representation for formal analysis, they have the ability to trace back the *cone of influence* (COI) of a signal by using the cone of influence command:

```
formal generate coi -var <signal_name>
```

A cone of influence includes all the logic that has the possibility to affect the output signal. Anything outside of the cone is by definition irrelevant, since logically it cannot affect the cone. In other words, logic outside of the cone of influence can automatically be classified as *safe* without the need for further analysis.

Overlapping the cone of the safety function with the safety mechanism reveals what safety related logic is covered by the safety mechanism, or, in other words, what logic should be treated as a *single-point fault* or a *residual fault* (or *latent faults* when overlapping cones of safety mechanisms). Safety functions and mechanisms, even on large designs, may only consist of a small number of potential faults, which makes formal analysis possible even on large ICs. We'll return to single-point faults and residual faults later in this article when we talk about analyzing faults with SLEC.

INJECTING A FAULT

Once faults are determined using the cone of influence, the faults in the cone are randomly injected. Typical failure modes considered in a fault campaign are *stuck-at0*, *stuck-at1* and *transient* faults. With formal, simple constant constraints can be specified, as with synthesis tools, causing logic to be pruned away as logic is removed. Likewise, nets and registers can be cut, also known as *cutpoints*. Here is an example of a cutpoint directive:

```
netlist cutpoint {dut.p_wbspi.clgen.cnt[8]}
```

Any signal not controlled by a driver becomes a formal control point, which formal can drive to any value (note: formal tools do not typically drive unknowns.) These cutpoints can introduce faults since the formal tool can drive either stuck-at or transient values onto them. Unfortunately, constants and cutpoints are created at compile time, which would require many compilations to add the potentially thousands and thousands of faults to inject.

Alternatively, a formal tool can create *conditional* cutpoints. For example:

```
netlist cutpoint {dut.p_wbspi.clgen.cnt[8]} -cond { fault == 100 } \
                                         -driver 0
```

Conditional cutpoints are specified for every possible fault in the design; however, the signal is only cut when the condition is true. This allows formal to run any number of targets, cut the appropriate signal to inject the fault, and require only one formal compilation.

ANATOMY OF A FAULT INJECTOR

The next step is to control the condition used in the cutpoint. The way to give formal control of a signal is to have no driver for it. This is referred to as a *non-deterministic (ND)* input since the formal engines can assign it to any value. Care must be taken to ensure that only one fault is selected per formal evaluation, otherwise multiple faults will be injected by the formal tool. Simple assumptions can constrain the fault selected:

```
bit [N-1:0] fault;
...
asm_stable_fault:assume property ( $stable(fault) );
```

Not all failure modes, however, call for a fault to be continuously stuck-at, and some requirements may even need to have the stuck-at faults injected at some time after reset. To accommodate this, a second ND-input is created to indicate when the fault should be injected:

```
bit nd_inject;
```

Getting assumptions to cover all cases can be a little tricky, so some straightforward SystemVerilog code is used to register when formal chooses to inject the fault, which makes writing cutpoint conditions, as well as other properties, easier:

```
bit injected;
always @($global_clock)
    injected = injected || nd_inject;
```


Notice that `$global_clock` is used here since formal automatically infers the clock and reset, which is handy when automating a fault injection flow. If the clock and reset are known, then it would be best to use those instead. Now the formal assumption becomes:

```
asm_stable_fault: assume property ( injected |-> $stable(fault) );
```

And the conditional cutpoint is:

```
netlist cutpoint { dut.p_wbspi.clgen.cnt[8] } \
    -cond { injected && fault == 100 } \
    -driver 'sb0
```

This method works for stuck-at faults as the injected signal remains asserted once the fault is injected and the cutpoint remains active from that point onwards. For transient faults, however, the cutpoint should only be active for a limited number of cycles. Instead of using `injected`, we can create another `inject` signal and constrain it for use with transient faults such that it creates a one-time pulse of a specified width:

```
netlist cutpoint { dut.p_wbspi.clgen.cnt[8] } \
    -cond { inject && fault == 100 } \
    -driver 'sb0
```

In this example, there is no reset on the `always` block so formal may try to inject faults during reset:

```
bit nd_inject;
bit inject;
bit injected;
always @($global_clock) begin
    inject = nd_inject;
    injected = injected || inject;
end
// -----
// Transient faults
// -----
asm_inject_width : assume property ( $rose(inject) |-> inject[*WIDTH] ##1 !inject );
asm_inject_stable : assume property ( $rose(inject) |-> $past(!injected));
```

Coding the `always` block with a reset solves this problem, or it can be solved using a special type of assumption—one that is called inside of an initial block. This forces the assumption to be applied only once formal evaluation begins, which happens when reset is released, and has the added benefit of indicating when a transient fault should be injected. The initial assumption could look like this:

```
property prop_init_seq;
    ( !inject[*START_MIN:START_MAX] ##1 inject ) and
    ( !injected[*START_MIN:START_MAX] ##1 injected );
endproperty
initial begin : INIT_SEQ
    asm_init_seq: assume property ( prop_init_seq );
end
```

Once the fault injector is created, it should be bound into the design or compiled in as a second top module.

PROPAGATING FAULTS WITH SLEC

With formal tools, stimulus generation is automatic. The difficulty comes in holding it back from using illegal or undefined input values. Usually, this involves writing lots of formal constraints and debugging lots of false negatives.

However, with formal SLEC, constraining the design becomes automatic. Unlike logic equivalency checking used to determine RTL-to-gate equivalency, which checks that the logic

ABOUT THE AUTHOR

Doug Smith is a functional verification consultant for Mentor, a Siemens Company, with expertise in UVM and formal technologies. Doug holds a master's degree in Computer Engineering from the University of Cincinnati and a bachelor's degree in Physics from Northern Kentucky University.

THE POWER OF UNMANNED

Tap into the power that 8,500 technologists, regulators and users of unmanned and autonomous technology bring to **AUVSI XPONENTIAL 2019**. Connect the dots, gain inspiration and gather game-changing information over four days of collaborative learning and collective progress with colleagues across industries.

- ▶ **Discover the power of innovation** at the largest showcase of unmanned technology in existence—with 100+ exhibits focused on engineering solutions.
- ▶ **Uncover the power of knowledge** with actionable insight and real-world solutions gleaned from 150+ educational sessions, including topics such as communications, remote sensing, machine learning and data management.
- ▶ **Harness the power of connection** as you network, share ideas and successes, and build new relationships with peers from 20+ vertical markets.



MCCORMICK PLACE | CHICAGO

April 29 – May 2 | Educational Program

April 30 – May 2 | Exhibits



UNLEASH THE POWER OF UNMANNED.

▶ REGISTER NOW AT **XPONENTIAL.ORG**

between registers is equivalent, SLEC uses formal methods to clock the design and determine if two entire designs are *functionally* equivalent. In other words, the functionality can be implemented in totally different ways with different timing, but the outputs remain the same. SLEC has many applications, such as checking that a design ported from VHDL to Verilog remains the same, or that a new feature does not break existing functionality.

With fault injection, a design can be compared with itself, with one instance containing the fault. If the two designs produce equivalent outputs even when a fault is injected, then clearly the fault does not affect the design; in other words, it can be classified as *safe*. SLEC automatically ties the inputs of the two instances together, so in essence the original design becomes the reference model that constrains the fault injected model. Formal constraints are no longer required and the entire legal state space can be explored without additional user input.

SLEC targets equivalency points, which are automatically inferred based on path names, so no manual intervention is needed when two instances of the same design are compared. If the outputs form the safety function, then the safety paths will automatically be checked for equivalency. If the safety function is a combination of signals or need qualified (like with a strobe or enable signal), an additional signal can be defined in SystemVerilog and a *SLEC map* can be created using a SLEC directive:

```
slec map { slec_wrapper.spec.top.safety_func } \
        { slec_wrapper.impl.top.safety_func }
```

In this example, the *slec_wrapper* is a top level wrapper created automatically by the Questa SLEC App. *spec* represents the specification or unmodified instance, and *impl* refers to the *implementation* or fault injected model.

SLEC mappings create SLEC targets, but it can fire from any fault being injected into the design. For ISO 26262 metric calculations, each fault must be classified so specific targets are required for each fault. Adding a condition to the SLEC mapping, as well as giving it a name, allows tracking of the fault results:

```
slec map { slec_wrapper.spec.top.safety_func } \
        { slec_wrapper.impl.top.safety_func } \
        -name SLEC_safe_100_sa0 \
        -cond {slec_wrapper.impl.fi.fault == 100 && \
              slec_wrapper.impl.fi.injected }
```

The *fi* instance and injected signal come from the fault injector described in the previous section. In this example, if target *SLEC_safe_100_sa0* fires, it means that formal SLEC has injected fault 100, which corresponds to a conditional cutpoint. A SLEC firing means that the two designs are not equivalent. A proof means that they are. In other words, a proof demonstrates that the fault is safe (**Figure 1**); whereas a firing demonstrates that the fault can propagate and that more analysis is required to see if it can be detected by a safety mechanism.

ANALYZING FAULTS

Injecting faults with SLEC to see if a fault can propagate is referred to as safe fault analysis in the Mentor *SafeCheck* flow. Simple SLEC mappings, as shown previously in the article, typically evaluate quickly, and the Questa SLEC App is optimized to find equivalency points and collapse logic so large designs and state spaces can be handled efficiently. However, proving that a fault can propagate does not prove that it can be detected.

By definition, safety related logic not covered by any intended safety mechanism is considered a *single-point fault (SPF)*. SPFs are easy to find using nothing more than a structural cone of logic. For everything else protected by a safety mechanism, the cones of influence can be overlapped to see what is covered and what is not. Anything not in the cone of the safety mechanism is structurally a *residual fault*. Even so, there is no guarantee that a fault outside of a safety mechanism cone can propagate, which is where safety fault analysis comes in. For all faults within the cone of influence of a safety mechanism, *residual fault analysis* needs to be performed. With this analysis, not only are we looking for a violation of the safety goal, but we

Status	Property	Fault Injected	Failure Mode
safe (113)			
Proven	SLEC_safe_516_safety1_sa0	dut.p_wbspi.pos_edge	sa0
Proven	SLEC_safe_250_safety1_sa0	dut.p_wbspi.clgen.cnt[7]	sa0
Proven	SLEC_safe_324_safety1_sa0	dut.p_wbspi.shift.p_in[22]	sa0
Proven	SLEC_safe_683_safety1_sa0	dut.p_wbspi.spi_tx_sel[1]	sa0
Proven	SLEC_safe_152_safety1_sa0	dut.p_wbspi.shift.data[119]	sa0

FIGURE 1

Proven SLEC results showing that the injected fault is safe—in other words, it does not affect the safety outputs

are also looking to see if the fault is detected by the safety mechanism within a specified time window, called the *fault detection interval* [2] (**Figure 2**). Note: this differs from the *fault tolerant time interval (FTTI)*, which is a system-level time interval, usually on the order of microseconds.

While simple SLEC mapping targets could be used and the results crossed, they do not adequately capture the residual fault scenario. SLEC targets by themselves do not include the concept of a window in time, as shown in Figure 2, and once a SLEC violation fires, any assumptions or subsequent behavior is ignored. Instead, SLEC supports SystemVerilog Assertions (SVA) properties, where any specific behavior can be described. So, for residual fault analysis, a simple cover property will suffice, along with some SystemVerilog code to simplify the writing of the property:

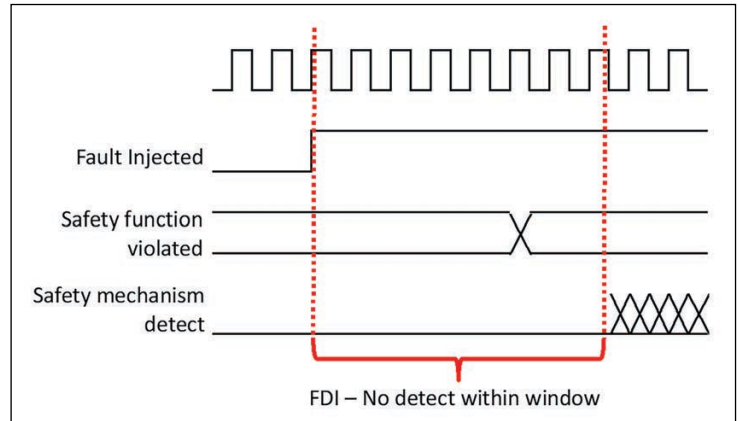


FIGURE 2

Residual fault example where a fault causes a violation but goes undetected

```

bit violation;
bit detection;

default clocking cb @($global_clock); endclocking

always @($global_clock) begin
    ...
    if ( injected && (safety_function_violated)) violation <= 1;
    if ( injected && (safety_mechanism_expression)) detection <= 1;
end

cov_res: cover property ( fault == 100 && violation ##[FDI] !detection );

```

In this example, the time window is specified by the parameter FDI. Technically, the ISO standard defines the FDI as the time from when the fault is injected until it is detected, but practically speaking, it makes more sense to treat it as the time from the violation to the point of detection, since propagation delays through the design will greatly vary from path to path. If formal finds a way to cover the property where the safety function is violated and the fault goes undetected, then a residual fault has been detected. If the property is uncoverable, then it will be treated as a dual-point fault (DPF) for later analysis of latent faults.

Latent fault analysis involves injecting two faults into the analysis—one in the safety function known to be detectable and the other in the primary safety mechanism. If a design has only one safety mechanism, then all state bits in the primary safety mechanism are by definition latent; i.e., faults in the safety mechanism cannot be detected. Most designs will have a secondary mechanism, such as power-on self-test, which can find stuck-at faults in the primary safety mechanism. (Note: the ISO standard does not require transient fault analysis for latent faults.)

Crossing safety function faults with safety mechanism faults increases the number of testable combinations exponentially. With formal, however, adding more faults reduces analysis. Since a fault is created by cutting a signal, adding faults creates additional control points for formal. The more control points, the more degrees of freedom formal has, making it easier to find a solution for the target under analysis.

By constraining formal to use only DPFs in the safety function, latent fault SLEC targets can be written similar to residual fault targets with just one modification. We will create a *fault ID* variable for the safety mechanism called `sm_fault` and rename the fault signal as `sf_fault` for the safety function fault ID. Then the latent fault target becomes as follows:

```

always @($global_clock) begin
    ...
    if ( injected && (safety_function_violated))
        violation <= 1;
    if ( injected && (primary_safety_mechanism_expression ||
                    secondary_safety_mechanism_expression))
        detection <= 1;
end

cov_latent: cover property ( sm_fault == 32 && violation ##[MPFDI]
                             !detection );

```

FIGURE 3

Example report summing up a formal regression run

SAFECHECK	All Faults (All Safety Critical Paths)				
	Fault Type	Previous Analysis		Current Analysis	
		Unverified	Verified	Unverified	Verified
	Safe faults (outside cone of influence)	0	287	0	287
	Safe faults (fault detected by a safety mechanism)	0	0	0	120
	Single-point faults (no safety mechanism)	0	0	0	0
	Residual fault (not covered by safety mechanism)	8	0	8	0
	Dual point fault (detected/perceived) in safety function	202	0	192	0
	Dual point fault (detected/perceived) in the safety mechanism	2035	0	1841	0
	Dual point fault latent	0	0	0	84
Subtotal	2245	287	2041	491	
Total	2532		2532		
Number of randomly sampled faults	2245 (88.7%)				
Design bits	2245 (unsafe) / 2532 (all)				
Residual Diagnostic Coverage	92.10% - 100.00%				
Latent Diagnostic Coverage	23.97% - 96.68%				

An additional conditional cutpoint is added for the second fault in the primary safety mechanism:

```
netlist cutpoint { dut.err_h0.spi_error } \
    -cond { injected && sm_fault == 32 } \
    -driver 'sb0
```

SVA assumptions are added to constrain the analysis to only two faults, using only known DPFs:

```
asm_dpfs: assume property ( sf_fault inside
    {[1:709],[854:855],[1387:1394],[1430:1431]} );
asm_stable_sm_fault: assume property ( injected -> $stable(sm_fault);
```

An advantage of this approach is that formal automatically chooses the DPF to use, and the appropriate fault is injected by the conditional cutpoints used during residual fault analysis.


An injected fault will have an effect down multiple paths within a design. Suppose a design has two ports, A and B. When a fault has no effect on port A, this implies it is a *safe* fault. However, the same fault may propagate down path B and go undetected, making it *residual*. In an IC, which is likely to have shared logic, prioritizing the fault classification is important. In this example, since there exists a path (to port B) where the fault is not detected, the fault should be classified as *residual*, not *safe*, even though it had no effect on port A. The fault priorities can be inferred from the flowchart found in figure B.2 of ISO26262-5:2018.

Once all the formal SLEC targets are evaluated, the results can be summed up in a report, as seen in **Figure 3**. More detailed reports can be created to evaluate missing safety coverage in the design. Any faults whose properties could not converge in formal can be exported as a fault list and used in a simulation or emulation regression to complete the safety metrics.

BEYOND FORMAL

Perhaps the biggest advantage of using a formal-based flow is that very quickly worst-case/best-case diagnostic coverage can be determined based on nothing more than structural analysis. As formal is able to verify results, the diagnostic coverage range narrows and confidence in the results increases.

Unlike other methods, when formal proves a fault is either safe, residual, or latent it is absolute and exhaustive. Anything formal cannot prove can be punted over to simulation and emulation. Likewise, software safety mechanisms are not an ideal candidate for formal, so

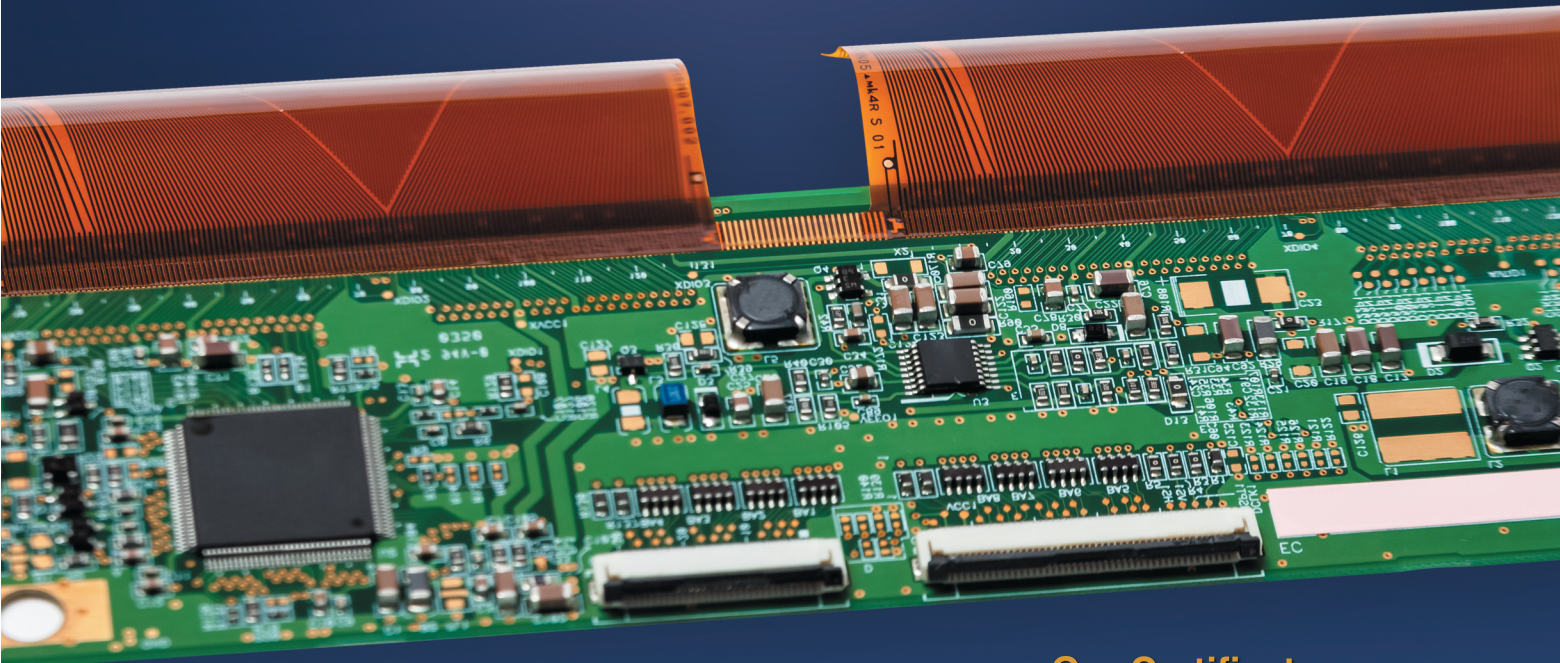
these need another solution, such as emulation. In any case, formal easily reduces the fault analysis effort by pruning the fault lists and focusing on the cone of influence. With little more than a few inputs describing the safety functions and mechanisms, the entire formal flow is automated. Using this approach is really the best method for proving something is functionally safe. 

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.

RESOURCES

Mentor, a Siemens Company | www.mentor.com

Flexible Manufacturing Solutions for Flex & Flex-rigid PCBs



Our Certificates:



Our Competences:

- IPC 2 custom Flex & Flex-rigid PCB fabrication
- Consigned and turnkey circuit board assembly
- Valor manufacturing system solution
- FREE valor DFM check and multiple test & inspection options
- Any order quantity from 1 to 10,000+
- Lead time varies from 2 days to 5 weeks as required
- Online quoting and ordering system

Pcbcart

China-based Electronics Manufacturer
Serving Globally since 2005



www.pcbcart.com



sales@pcbcart.com

Tailored Solutions Tackle Design Needs for Wearables

Low Power Priorities



For wearable devices, every drop of power is precious. That's driving designers of these embedded systems to attack the power challenge from multiple angles. Fortunately, a slew of analog, power and system ICs have emerged that address the wearable market's particular needs.

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

While power is an important issue in any embedded system design, it's especially critical in wearable devices. Today's generation of wearable electronics require longer battery lives, more functionality and better performance—all in extremely small form factors. Wearables comprise a wide variety of products including smartwatches, physical activity monitors, heart rate monitors, smart headphones and more.

Today's wearable electronic devices share some common design priorities. First, they have an extremely low budget for power consumption. And because they're not suited to being powered by replaceable batteries, they usually require a way for the unit to be recharged. Meanwhile, most modern wearables require some kind of wireless connectivity.

Feeding those needs, chip vendors—primarily from the microcontroller (MCU) and analog sectors—over the past 12 months have announced a generous mix of solutions to help keep power consumption low, to aid recharging and to enable new capabilities while maintaining narrow power

constraints. Chip and platform solutions aimed at wearables span the range from specialized power management ICs (PMICs), data converters and power regulator chips, to wireless charging solutions and even complete reference design platforms specially for wearables.

WRIST-WORN HEALTH GEAR

Wearables have evolved from being more than just fun devices for health and fitness. Using sophisticated sensors and other capabilities, devices are being designed to do virtual care monitoring, assess chronic conditions and evaluate overall well-being. Along just those lines, in September Maxim Integrated announced its Health Sensor Platform 2.0 (HSP 2.0) (**Figure 1**). This wrist-worn platform can be used for rapid prototyping, evaluation and development. It provides the ability to monitor electrocardiogram (ECG), heart rate and body temperature from a wrist-worn wearable, saving up to six months in development time, according to Maxim.

In the past, system developers have found it challenging to derive precise ECG

monitoring from the wrist—most alternatives require a wearable chest strap. Getting accurate body temperature typically requires using a thermometer at another location. Maxim has overcome these challenges in the HSP 2.0, by using its proprietary sensor and health monitoring technology.

Enclosed in a watch casing, the wrist-based form factor enables HSP 2.0 to provide basic functionality out of the box, with body-monitoring measurements starting immediately. Data can be stored on the platform for patient evaluation or streamed to a PC for analysis later. Unlike other wearables, the data measurements collected by the HSP 2.0 can be owned by the wearer. This alleviates data privacy concerns and enables users to conduct their own data analysis. Also, because HSP 2.0 is an open platform, designers can evaluate their own algorithms on the board. In addition, the modular format is future proof to quickly accommodate new sensors over time.

HSP 2.0 includes the following Maxim chips: the MAX32630 DARWIN low-power MCU for wearables; the MAX32664 ultra-low-power biometric sensor hub with embedded heart-rate algorithm; the MAX20303 PMIC; the MAX30205 human body temperature sensor with $\pm 0.1^\circ\text{C}$ accuracy; the MAX30001 single-channel integrated biopotential and bioimpedance analog front-end (AFE) solution; and the MAX86141 optical pulse oximeter and heart-rate sensor.

ENERGY CONTROLLER FOR WEARABLES

For its part, Renesas Electronics has been working on meeting extreme low power demands by applying innovations in semiconductor process development. In November the company unveiled an innovative energy-harvesting embedded controller that can eliminate the need to use or replace batteries in a device. Developed based on Renesas' SOTB (silicon-on-thin-buried-oxide) process technology, the new embedded controller achieves extreme reduction in both active and standby current consumption. The extreme low current levels of the SOTB-based embedded controller enables system designers to completely eliminate the need for batteries in some of their products through harvesting ambient energy sources such as light, vibration and flow (**Figure 2**).

Although the solution was developed with



FIGURE 1

The Health Sensor Platform 2.0 is a wrist-worn platform that can be used for rapid prototyping, evaluation and development. It provides the ability to monitor electrocardiogram (ECG), heart rate and body temperature from a wrist-worn wearable.

IoT devices in mind, the controller is more broadly aimed at what they call the new market of maintenance-free, connected IoT sensing devices with endpoint intelligence. This includes health and fitness apparel, shoes, wearables, smart watches and drones. Renesas' first commercial product using SOTB technology, the R7F0E embedded controller, is a 32-bit, Arm Cortex-based embedded controller. The device is capable of operating up to 64 MHz for rapid local processing of sensor data and execution of complex analysis and control functions.

The R7F0E consumes just 20 $\mu\text{A}/\text{MHz}$ active current, and only 150 nA deep standby current, approximately one-tenth that of conventional low-power MCUs. According to the company, samples of the new R7F0E embedded controller are available now for beta customers, and samples are scheduled to be available for general customers from July 2019. Mass production is scheduled to start from October 2019.

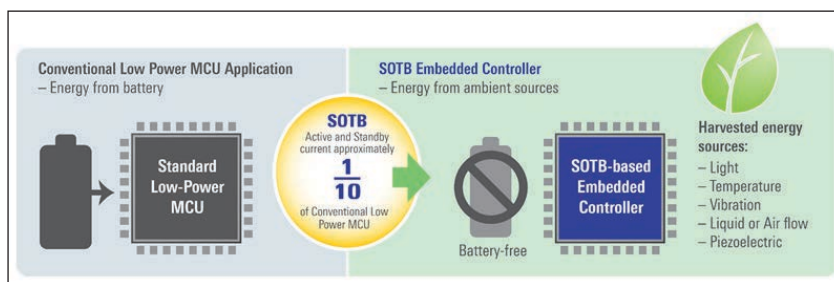


FIGURE 2

The extreme low current levels of the SOTB-based embedded controller enables system designers to completely eliminate the need for batteries in some of their products through harvesting ambient energy sources such as light, vibration and flow.

FIGURE 3

With an ultra-low I_Q of 250 nA versus the approximately 1 μ A operation of traditional devices, the MCP1811 LDO regulator saves battery life, enabling end users to recharge or replace batteries less often.



LDO REGULATOR FOR WEARABLES

Achieving longer battery lives is a problem that can be attacked from many angles. Power regulator electronics are among those. With that in mind, Microchip Technology in October introduced a linear Low Dropout (LDO) regulator that extends battery life in portable devices up to four times longer than traditional ultra-low quiescent (I_Q) LDOs. With an ultra-low I_Q of 250 nA versus the approximately 1 μ A operation of traditional devices, the MCP1811 LDO reduces quiescent current to save battery life, enabling end users to recharge or replace batteries less often (**Figure 3**).

Well suited for IoT and battery-operated applications such as wearables, remotes and hearing aids, the LDO reduces power consumption in applications by minimizing standby or shutdown current. Reducing standby power consumption is critical in remote, battery-powered sensor nodes, where battery replacement is difficult and operating life requirements are high. Available in package options as small as 1 mm x 1 mm, the MCP1811 consumes minimal board space to meet the needs of today's compact portable electronic designs. Depending on the application and number of LDOs, designers can take advantage of the extra board space with a larger battery to further increase battery life.

An additional benefit the MCP1811 offers is faster load line and transient response when compared to other ultra-low I_Q LDOs. Faster response times can accelerate wake-

up speed in devices such as monitors or sensors that require immediate attention. Faster transient response can help designers avoid undervoltage and overvoltage lockout measures used in sensitive applications where transient spikes can lead to catastrophic results.

SECURE PAYMENTS WITH WEARABLES

An important capability in a certain class of wearables is the ability to support electronic retail transactions directly from the wearable device. While this is arguably a whole separate technology category in itself, we'll touch on a couple developments here. In November, Infineon Technologies announced an EMV-based payment solution for key chains, rings, wristbands, bracelets and other wearable devices.

The SECORA Pay W for Smart Payment Accessories (SPA) combines an EMV chip with the card operating system, payment applet as well as the antenna directly on the unit. As a turnkey solution it allows card vendors, device manufacturers, financial institutions or event organizers to quickly and cost-efficiently introduce fashion accessories for payment and even access.

Infineon's SECORA Pay solutions portfolio comprises the SECORA Pay S for standard Visa and MasterCard payment cards, SECORA Pay X for applications with extended features such as multi-application, national debit and white label schemes or access management and SECORA Pay W for payment accessories. All SECORA turn-key solutions are pre-certified by Mastercard and Visa and will accelerate the deployment of contactless payment. The EMV Chip Specifications (www.emvco.com) define globally valid requirements for chip-based payment solutions and acceptance terminals. They enable secure contact- and contactless applications and the use of other emerging payment technologies.

COMPLETE PAYMENT SoC

Likewise a player in the contactless transaction market, STMicroelectronics (ST) back in October announced teaming up with Fidesmo to create a turnkey active solution for secure contactless payments on smart watches and other wearable technology. The complete payment system-on-chip (SoC) is based on ST's STPay-Boost IC, which combines a hardware secure element to protect transactions and a contactless controller

featuring proprietary active-boost technology that maintains reliable NFC connections even in devices made with metallic materials. Its single-chip footprint fits easily within wearable form factors (**Figure 4**).

ST's proprietary NFC-boosting active load modulation technology simplifies RF design and accelerates time to market by ensuring superior performance with little or no circuit optimization needed. A small-size antenna can sustain robust and reliable wireless connection, permitting smaller overall product dimensions and lower power consumption resulting in longer battery life.

Fidesmo's MasterCard MDES tokenization platform completes the solution by allowing the user to load the personal data needed for payment transactions. Convenient Over-The-Air (OTA) technology makes personalization a simple step for the user without any special equipment. Kronaby, a Sweden-based hybrid smartwatch maker, has embedded the STPay-Boost chip in its portfolio of men's and women's smart watches that offer differentiated features such as freedom from charging and filtered notifications. The SoC with Fidesmo tokenization enables Kronaby watches to support a variety of services such as payments, access control, transportation and loyalty rewards.

DATA CONVERTERS

Data converters also have role to play in efforts to meet the extreme low power needs of wearable devices. Along such lines, in December Texas Instruments (TI) introduced four tiny precision data converters (**Figure 5**). The new data converters enable designers to add more intelligence and functionality, while shrinking system board space. The DAC80508 and DAC70508 are eight-channel precision digital-to-analog converters (DACs) that provide true 16- and 14-bit resolution, respectively.

The ADS122C04 and ADS122U04 are 24-bit precision analog-to-digital converters (ADCs) that feature a two-wire, I²C-compatible interface and a two-wire, UART-compatible interface, respectively. The devices are optimized for a variety of small-size, high-performance or cost-sensitive electronics applications such as wearables.

Both DACs include a 2.5-V, 5-ppm/°C internal reference, eliminating the need for an external precision reference. Available in a 2.4-mm-by-2.4-mm die-size ball-grid array (DSBGA) package or wafer chip-scale package



(WCSP) and a 3-mm-by-3-mm quad flat no-lead (QFN)-16 package, these devices are up to 36% smaller than the competition, says TI. Meanwhile, the tiny, 24-bit precision ADCs are available in 3-mm-by-3-mm very thin QFN (WQFN)-16 and 5-mm-by-4.4-mm thin-shrink small-outline package (TSSOP)-16 options. The two-wire interface requires fewer digital isolation channels than a standard serial peripheral interface (SPI), reducing the overall cost of an isolated system. These precision ADCs eliminate the need for external circuitry

FIGURE 4

The STPay-Boost IC combines a hardware secure element to protect transactions and a contactless controller featuring proprietary active-boost technology that maintains reliable NFC connections even in devices made with metallic materials. Its single-chip footprint fits easily within wearable form factors.

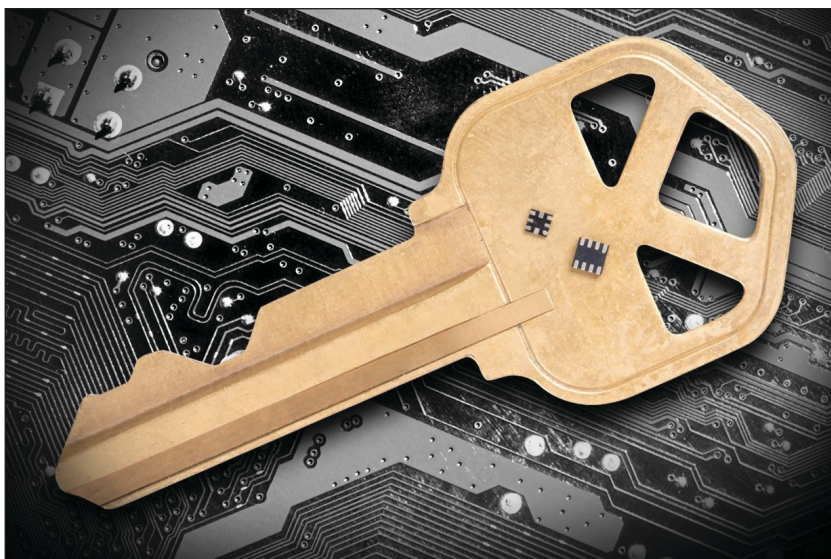
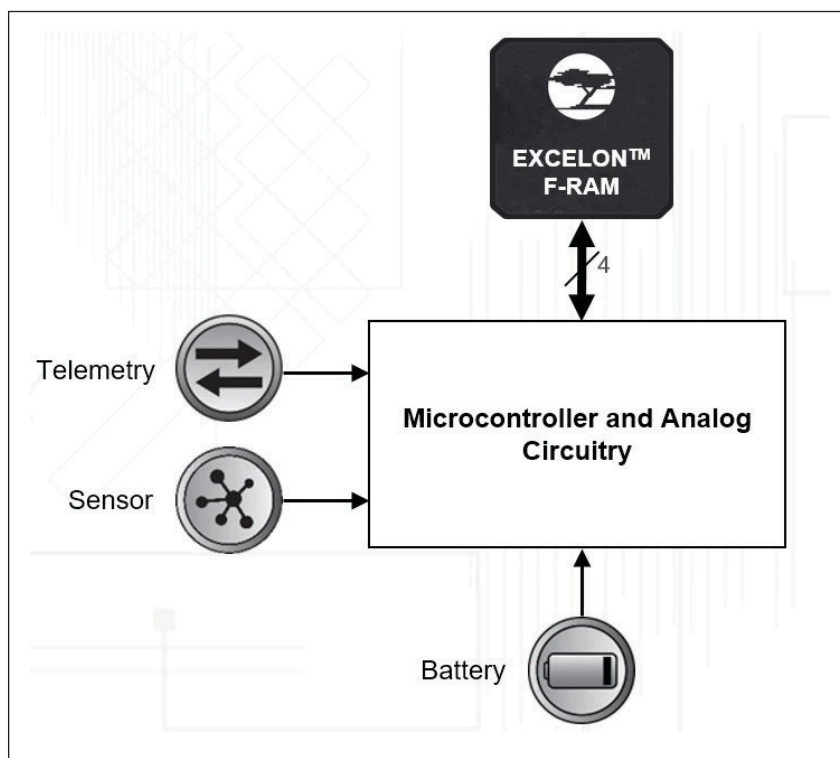
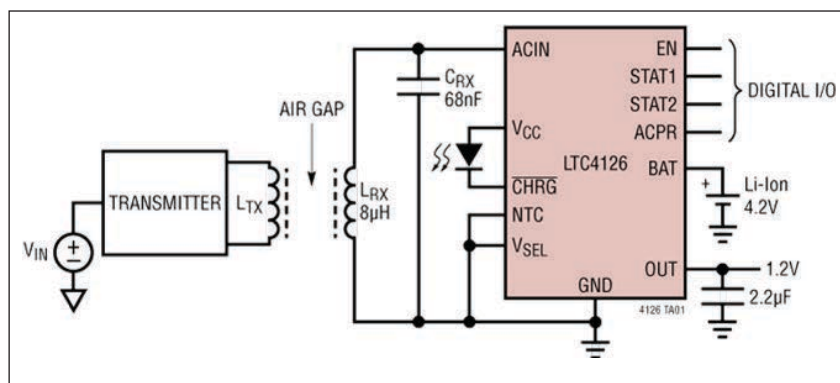


FIGURE 5

The DAC80508 and DAC70508 are eight-channel precision DACs that provide true 16- and 14-bit resolution, respectively. The ADS122C04 and ADS122U04 are 24-bit precision ADCs that feature a two-wire, I²C-compatible interface and a two-wire, UART-compatible interface, respectively.

**FIGURE 6**

The Excelon LP FRAM provides instant-write capabilities with virtually unlimited endurance. This enables wearable systems to perform mission-critical data logging requirements while maximizing battery life.

**FIGURE 7**

The LTC4126 combines a wireless powered battery charger for Li-Ion cells with a high efficiency multi-mode charge pump DC-DC converter, providing a regulated 1.2 V output at up to 60 mA.

by integrating a flexible input multiplexer, a low-noise programmable gain amplifier and other circuitry.

MEMORY INNOVATIONS

Among the latest innovations aimed at wearables from Cypress Semiconductor is an FRAM (ferroelectric random access memory)-based data logging solution. In November, Cypress introduced a nonvolatile data-logging solution with ultra-low power consumption. This solution is well suited for portable medical and wearable devices that demand nonvolatile memories to continuously log an increasing amount of user and sensor data while using as little power as possible.

Cypress' Excelon LP FRAM is an energy-efficient device that provides instant-write capabilities with virtually unlimited endurance (**Figure 6**). This enables wearable systems to perform mission-critical data logging requirements while maximizing battery life. The Excelon LP series is available in a low-pin-count, small-footprint package that is suited for space-constrained, wearable applications.

The Excelon LP series offers 4-Mb and 8-Mb industrial and commercial-grade densities with 50 MHz and 20 MHz Serial Peripheral Interface (SPI) performance. The series reduces power consumption with 100 nA hibernate and 1 µA standby modes that greatly improve a battery-powered product's user experience by extending system operating time. The device's inherent instant writes also eliminate power failure "data-at-risk" due to volatile data buffers in legacy memories. The family features wide voltage operation from 1.71 V to 3.6 V and is available in RoHS-compliant industry-standard packages that are pin compatible with EEPROMs and other nonvolatile memories. Excelon LP F-RAMs provide 1,000-trillion (10^{15}) read/write cycle endurance with 10 years of data retention at 85°C or 151 years at 65°C.

CHARGING WEARABLES

A common aspect of wearable devices is that they tend not to be suited for replaceable batteries. As a result, they typically need to be recharged. Wireless (cordless) battery charging is beginning to take hold as a solution. Feeding such needs, in October Analog Devices announced its Power by Linear LTC4126 as an expansion of its offerings in wireless battery charging. The LTC4126 combines a wireless powered battery charger for Li-Ion cells with a high efficiency multi-mode charge pump

RESOURCES

Analog Devices | www.analog.com

Cypress Semiconductor | www.cypress.com

Infineon Technologies | www.infineon.com

Maxim Integrated | www.maximintegrated.com

Microchip Technology | www.microchip.com

Renesas Electronics America | www.renesas.com

STMicroelectronics | www.st.com

Texas Instruments | www.ti.com

DC-DC converter, providing a regulated 1.2 V output at up to 60 mA (**Figure 7**).

Charging with the LTC4126 allows for a completely sealed end product without wires or connectors and eliminates the need to constantly replace non-rechargeable (primary) batteries. The efficient 1.2 V charge pump output features pushbutton on/off control and can directly power the end product's ASIC. This greatly simplifies the system solution and reduces the number of necessary external components. The device is ideal for space-constrained low power Li-Ion cell powered wearables such as hearing aids, medical smart patches, wireless headsets and IoT devices.

The LTC4126, with its input power management circuitry, rectifies AC power from a wireless power receiver coil and generates a 2.7 V to 5.5 V input rail to power a full-featured constant-current/constant-voltage battery charger. Features of the battery charger include a pin selectable charge voltage of 4.2 V or 4.35 V, 7.5 mA charge current, automatic recharge, battery temperature monitoring via an NTC pin, and an onboard 6-hour safety charge termination timer. Low battery protection disconnects the battery from all loads when the battery voltage is below 3.0 V. The LTC4126's charge pump switching frequency is set to 50 kHz/75 kHz to keep switching noise out of the audible range, ideal for audio related applications such as hearing aids and wireless headsets. The IC is housed in a compact, low profile (0.74 mm) 12-lead 2 mm x 2 mm LQFN package. The device is guaranteed for operation from -20°C to 85°C in E-grade.

KIT FOR WIRELESS CHARGING

Also facilitating building wireless chargers for wearables, ST for its part offers a kit-level solution. The ST plug-and-play wireless battery-charger development kit (STEVAL-ISH045V1) lets users quickly build ultra-compact chargers up to 2.5 W with a space-saving 20 mm-diameter coil, for charging small IoT devices and wearables such as smart watches, sports gear or healthcare equipment (**Figure 8**).

Built around the STWBC-WA wireless charging-transmitter controller, the kit comprises a charging base unit containing a transmitter board with the 20 mm coil already connected and ready to use. Getting started is easy, using the PC-based STSW-STWBCGUI software to configure the STWBC-WA and

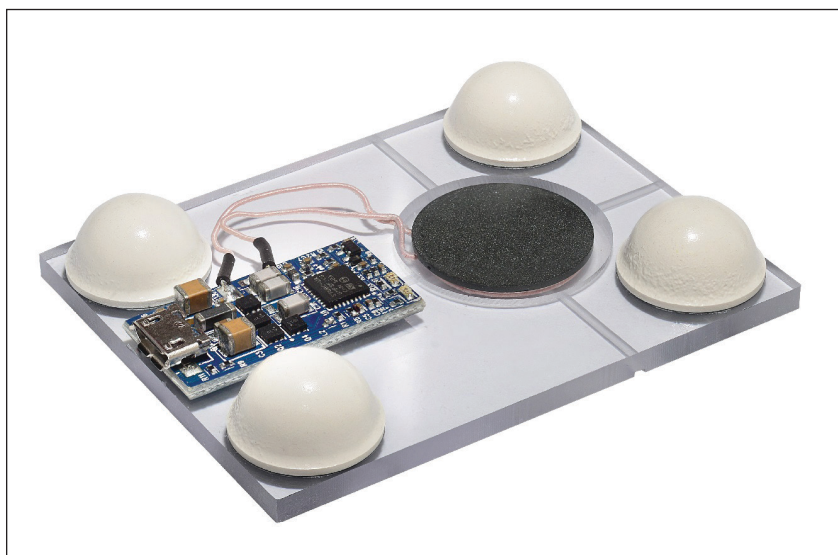



FIGURE 8

The ST plug-and-play wireless battery-charger development kit lets users quickly build ultra-compact chargers up to 2.5 W with a space-saving 20 mm-diameter coil, for charging small IoT devices and wearables such as smart watches, sports gear or healthcare equipment.

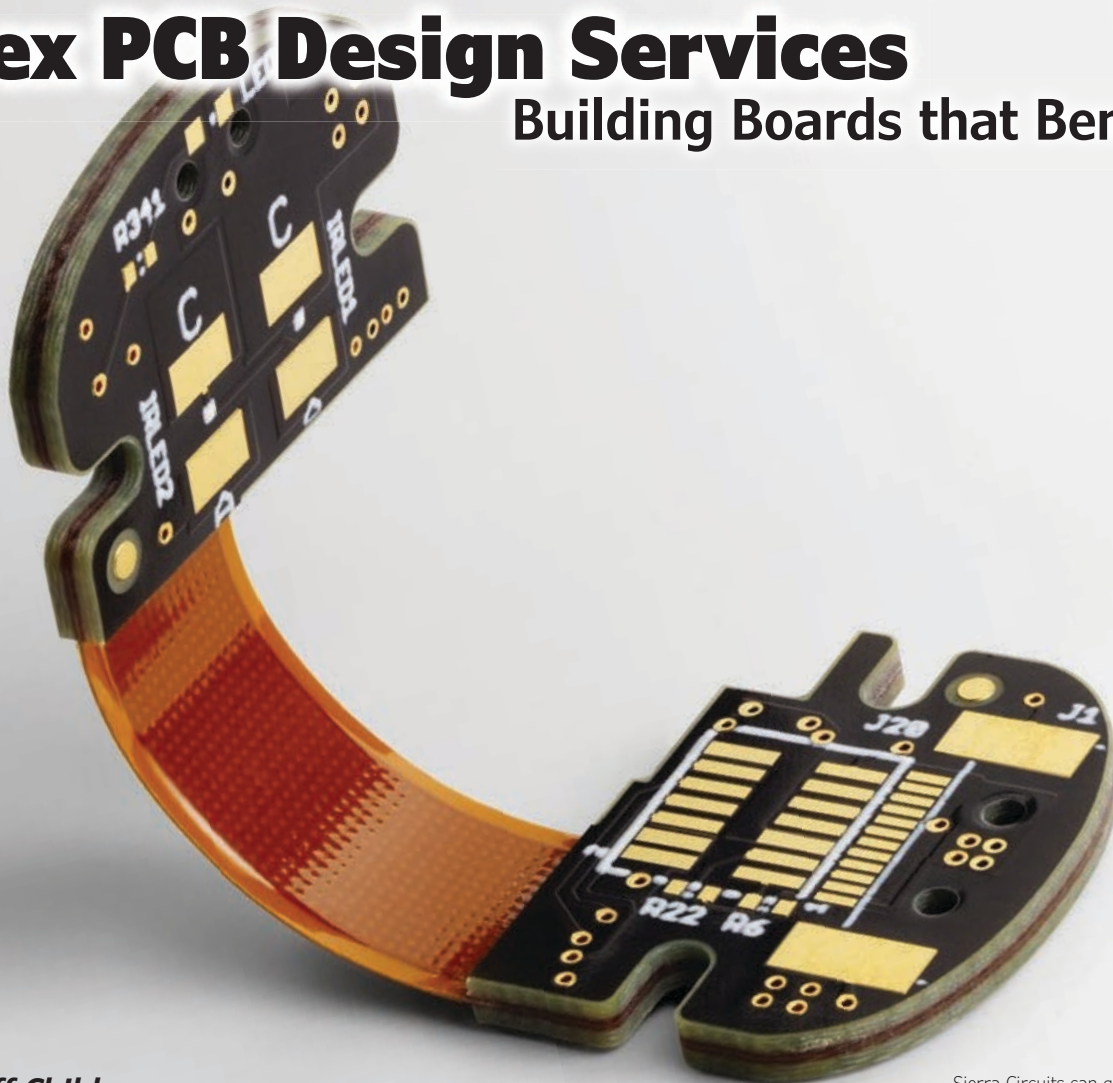
monitor runtime information such as power delivered, bridge frequency, demodulation quality and protocol status. The kit includes a dongle for running the GUI. The supporting ecosystem includes certified reference boards, software and detailed documentation to help developers quickly design chargers for wearables.

The STWBC-WA controller chip contains integrated drivers and natively supports full-bridge or half-bridge topologies for powering the antenna. The half-bridge option allows charging up to 1 W with a smaller-diameter coil for an even more compact form factor. The chip supports all standard wireless-charging features, including Foreign Object Detection (FOD) and active presence detection for safe charging, and uses digital feedback to adapt the transmitted power for optimum efficiency at all load conditions. Two firmware options give users the choice of a fast turnkey solution or customizing the application using APIs to access on-chip peripherals including an ADC, a UART and GPIOs.

Clearly there are many facets and angles to address the low power needs of wearables. As demands for more functionality rise, system developers will need to remain ever mindful of keeping battery life at the same lengths or longer. Fortunately, there seems to be no stopping the innovation among chip vendors targeting this growing wearables market. 

Flex PCB Design Services

Building Boards that Bend



TECH SPOTLIGHT

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

Sierra Circuits can quickturn
Flex PCBs in two days or less.

While not exactly a brand-new technology, flexible printed circuit boards are a critical part of many of today's challenging embedded system applications from wearable devices to mobile healthcare electronics. The PCB production service vendors listed here feature a variety of Flex PCB manufacturing services to help bring your embedded system design from prototype to production.

ACCUTRACE

www.pcb4u.com

Location: Santa Clara, CA

Accutrace serves the electronics industry's needs for quick turn PCB fabrication from prototype to production. They can manufacture only what is needed, when it's needed—and in the amount needed. No need

to worry about long shelf life or tying up your capital in bare board inventory.

Services Summary: PCB capabilities: Capabilities: Up to 50 layers; any layer HDI; blind and buried vias; 2-mil line width and spacing; copper up to 16 oz. Materials supported: FR4, Rogers, ISOLA, Polyimide-flex, Metal core, Taconic and Magtron

Flexible PCB Services: Flex and Rigid-Flex

BETA LAYOUT

uk.beta-layout.com

Location: Shannon, Ireland

Beta Layout's PCB-POOL operation is Europe's largest manufacturer of PCB prototypes, with over 36,000 customers worldwide. PCB-POOL cost sharing principle enables system designers to benefit from low prices. You can calculate the prices instantly online using the PCB-POOL's Online price calculation. PCBs using special technologies and materials can also be sourced directly from the company, from prototype to series production quantities.

Services Summary: Prototype PCBs or small series PCBs in 1-6 working days; Laser-cut SMD-Stencils in 1-3 working days; Fully populated SMD and THT prototype boards in 2 working days; Support for PCBs with embedded UHF RFID modules

Flexible PCB Services: All of Beta Layout's Flex-Pool PCBs are manufactured exclusively in Germany. Only polyimide foils are used as the base material. These provide different benefits compared to other foils, differing substantially from rigid PCB materials.

EzPCB

www.ezpcb.com

Location: Shenzhen, China

EzPCB is an online provider of PCB manufacture and PCB assembly products and services. Based in China, the company's worldwide business has been growing rapidly since its launch in 2004. EzPCB has supplied high quality PCBs—and related products and services—for over 2,000 customers from more than 40 countries. Its customers include amateurs, small businesses, universities and many world-class companies and organizations including Jet Propulsion Laboratory, Micron, Siemens, STMicroelectronics and Rohm.

Services Summary: PCB manufacturing, PCB assembly, cabling, enclosures, keypads, stencils and components, one-stop turnkey services and design consultation

Flexible PCB Services: Flex PCB support: Components mounted on flexible plastic substrates, such as polyimide, PEEK or transparent conductive polyester film. Additionally, flex circuits can be screen printed silver circuits on polyester. Flexible electronic assemblies may be manufactured using identical components used for rigid printed circuit boards, allowing the board to conform to a desired shape, or to flex during its use. Rigid-Flex PCB support: Flexible and rigid board structures are bonded together in the same units as one part of so called rigid-flex boards. Most rigid flex boards consist of multiple layers

of flexible circuit substrates attached to one or more rigid boards externally and/or internally, depending upon the design of the application. Rigid-Flex PCBs contain a flexible part with an insulating layer and a conductor layer, and a rigid part with circuit wiring.

EPEC

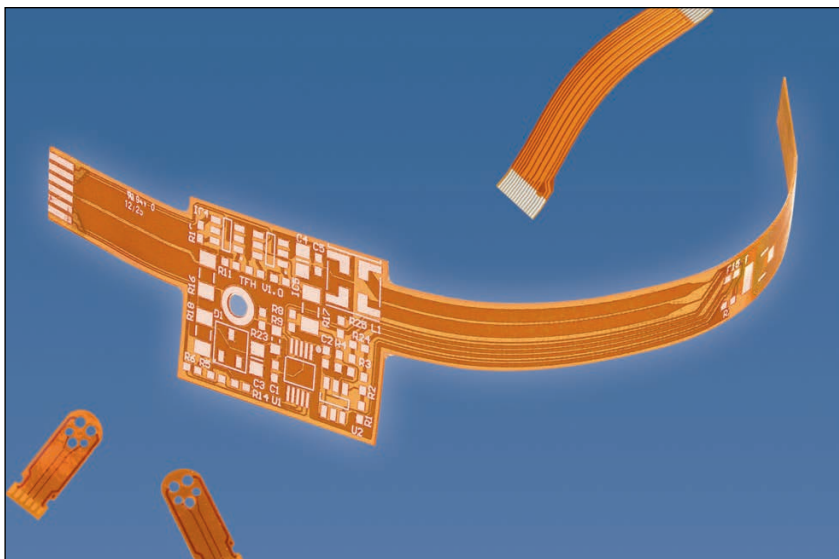
www.epectec.com

Location: New Bedford, MA

Epec specializes in custom, build to print electronics. It has a global team of engineers, designers, R & D innovators, product managers, manufacturing/supply chain professionals, quality assurance personnel and sales/customer service staff, all of whom are experts in their fields. Rather than limit its production capacity to its US and UK manufacturing centers, Epec has developed UL certified world class production facilities which are ISO-9001, QS-9002, TS-16949, with aligned technology roadmaps and quality systems.

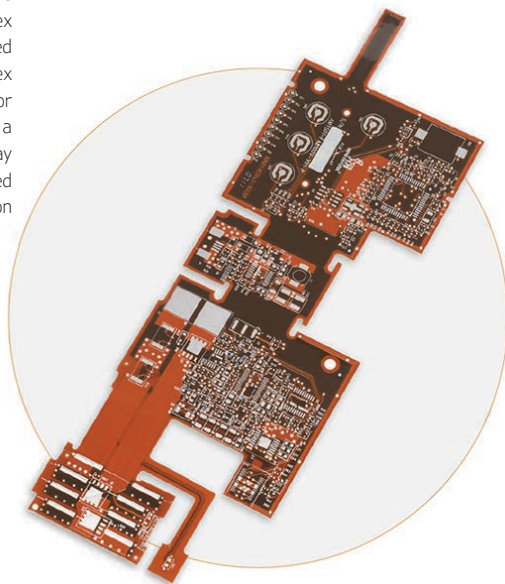
Services Summary: Provides complete engineering and design services, from concept through production, in a quick and efficient time frame. Capabilities include: battery pack design and assembly, PCB electronic design, cable assembly design, flex and rigid-flex circuit design, user interface and more.

Flexible PCB Services: Epic's flex and rigid-flex solutions and capabilities span from single- or double-sided circuitry to higher technology multilayer designs up to 6 layers. Its flexible circuits are built to withstand the rigors of aerospace, medical and military applications. As a high reliability replacement for wire and wire harness assemblies, flex circuits provide a significant cost savings with no reduction in performance.



Beta Layout's Flex-Pool PCBs use polyimide foils as their base material.

Shown here is an example of Flexible Circuit Technology's multi-layer flex PCB. It combines several single-sided or double-sided circuits with complex interconnections, shielding and/or surface mounted technologies in a multi-layer design. The layers may or may not be continuously laminated together throughout the production process.



FLEXIBLE CIRCUIT TECHNOLOGIES

www.flexiblecircuit.com

Location: Minneapolis, MN

Flexible Circuit Technologies (FCT) has an internationally recognized flex circuit Application Engineering and Design Team offering over 250 years of combined industry expertise, allowing FCT to provide customers with superior design services and attentive overall support. By co-chairing the IPC (Association Connecting Electronics Industries) Flexible Circuit Design committee, FCT stays abreast of industry standards and helps set standards the industry.

Services Summary: Supports multiple types of flexible PCBs; value added Assembly, Testing Services to complete box builds; Design consultation, optimization, DFM (Design for Manufacturing), to final flex design; Supply chain management for components and materials; Prototypes through production; Low mix-high volume and high mix-low volume program support; Assembly services; Testing Services.

Flexible PCB Services: Supports single, double, multi-layer flex, rigid flex, flexible heaters, flat flex and membrane switches. Heat sinks can be laminated to flex circuits to dissipate heat away from sensitive components such as LEDs. Supports through hole and surface mount capabilities on Flex PCBs, as well as in circuit testing, conformal coating, and electrostatic protective packaging. Graphic overlays can be placed over flexible circuits, rigid boards or membrane switches.

IMAGINEERING

www.pcbnet.com

Location: Elk Grove Village, IL

Imagineering acts as a reliable source for high quality and on time PCBs. Its quick turn prototypes not just intended for testing and verification of designs. Every one of its boards meet IPC-A-600 F (Class2) standard, be it prototype or production. Specializes in quick turn prototypes, as well as rapid turn production.

Services Summary: PCB assembly capabilities summary: SMT, Flip chip, thru hole, Flex circuit assembly, cable assemblies, lead-free assembly and wire harness assembly. Full Turnkey service for all customers if needed as well as partial turn-key and consignment orders. PCB capabilities include: 22-layer fabrication, hole sizes down to 8 mil plated and 5 mil laser drilled, 3 mil line width and spacing, 6 oz. copper and a maximum PCB thickness up to .300".

Flexible PCB Services: Supports Multi-Layer, Rigid and Flex PCBs.

OURPCB

www.ourpcb.com

Location: Shijiazhuang Hebei, China

OurPCB is a multi-national PCB manufacturing and PCB assembly company that provides global service and support while using its own Chinese manufacturing capabilities. Company has provided professional PCB production and assembly services for more than 2,500 customers around the globe.

Services Summary: Assembly capabilities include BGA, LGA, QFN, QFP, DIP and SIP. The smallest SMT footprint it can mount is 0201. Factory can also provide programming and wiring as well as injecting and conformal coating services.

Flexible PCB Services: Can fulfill flex PCB orders of 20 layers maximum; high-quality, quick turn flexible PCB manufacturing services. Processes and services include:

- Line Width/Space in the Inner Layer: 3/3 mil
- Min. Drilling Hole Diameter: 6 mil
- Tolerance for Impedance Control (+/-): 10% rts

PCBCART

www.pcbcart.com

Location: Hangzhou, China

PCBCart is a professional PCB production service provider with more than 10 years of experience in the electronics manufacturing industry. It has manufactured printed circuit boards for more than 10,000 companies and over 80 countries around the world. Fast,

affordable prototype assembly they take your unique PCB designs, prepare them for the assembly process and perform comprehensive testing to ensure they meet your precise performance requirements. Can provide a complete turnkey PCB prototype assembly featuring a one-stop shop approach.

Services Summary: Development, manufacturing, assembly and testing of custom PCBs; Rapid PCB prototyping; Circuit boards manufacturing; PCB assembly; Components sourcing services

Flexible PCB Services: Specializes in the fabrication and assembly of flexible PCB products for use in the manufacturing of medical devices and various types of telecommunication and industrial equipment. Can also work with customers to produce a custom flex PCB prototype.

Specifications:

- Quality Grade Standard IPC 2
- Number of Layers 1 - 8layers
- Order Quantity 1 pc - 10,000+pcs
- Build Time 2 days - 5 weeks
- Material DuPont PI, Domestic Shengyi PI
- Board Size Min. 6 mm x 6 mm, Max. 406 mm x 610 mm
- Board Thickness 0.1 mm - 0.8 mm

PCB UNLIMITED

www.pcbunlimited.com

Location: Tualatin, OR

In 2003, PCB Unlimited's sister company Stencils Unlimited pioneered the internet SMT stencil quote and order process. In 2008, PCB Unlimited took it one step further by providing a one-stop-shop where engineers can quote and order online US and offshore PCB services and everything else they need for their PCB projects.

Services Summary: US PCB fabrication including Rigid, Flex and Rigid-Flex PCBs; Offshore PCB Fabrication; US Quick-Turn Prototype PCB Assembly and Low Volume Production followed by our offshore operation to service your high-volume production needs

Flexible PCB Services: Flex and Rigid-Flex PCBs from prototype to production.

Specifications:

- Trace and space capabilities down to .003"/.003" (0.5 oz.), .004"/.004" (1 oz.)
- Microvia diameter from .003"
- Lead times as short as 5 days
- Multiple offshore partners quoting system to offer excellent pricing (ISO:9001 certified)
- Selective plating of gold on the same flexible circuit
- Minimum tolerance of ZIF connector to edge of flex +/- .003"

SAN FRANCISCO CIRCUITS

www.sfcircuits.com

Location: San Mateo, CA

Specializes in the complex, advanced technology of PCB fabrication and assembly, producing high quality multi-layered PCBs from elaborate layouts. Expertise in circuit boards is not limited to PCB manufacturing, but extends to PCB layout & PCB assembly.

Services Summary: PCB fabrication, PCB assembly and PCB layout services. Supports Mil-SPEC PCBs, Flex PCBs, Metal Core PCBs and many others.

Flexible PCB Services: Supports Flex PCBs and Rigid-flex PCBs. Can provide heavy copper flex circuit boards that can also be coupled with flexible thin copper layers designed to handle heavier power currents. Your heavy copper boards can be built with flex portions for lower voltage additions and applications.

Flex PCB specifications:

- Drill to copper – Outer layers 8 mil
- Drill to copper – Inner layers 10 mil
- Copper Thickness 0.5 oz. - 4 oz.
- Minimum Hole Size 0.004"
- Minimum Trace/Space 0.003"/0.003"
- Minimum Pitch 0.5 mm

SIERRA CIRCUITS

www.protoexpress.com

Location: Sunnyvale, CA

Sierra Circuits is an ISO 9001:2015, ISO 13485:2016 and MIL-spec MIL-PRF-55110 certified, Silicon Valley-based, high-technology PCB manufacturer and assembler. It specializes in quick turn PCBs and medium production. By owning its manufacturing and assembly facilities, the company controls every aspect of the production schedule and quality.

Services Summary: Micro/fine line PCBs, PCB assembly, flexible PCBs, lead-free PCBs, high-reliability PCBs, no-touch PCBs, burn-in PCBs, R&D PCBs, MIL-spec PCBs, PCB design

Flexible PCB Services: Latest offerings include Flex PCBs, Rigid-flex PCBs, and microelectronics PCBs down to 1 mil trace/space (some restrictions apply). Offers high-quality, high-reliability quickturn flexible printed circuit boards in two days or less.

Flex PCB specifications:

- Max. Layer Count 12
- Min. Core Thickness .001"
- Min. Starting Cu Foil Thickness 0.1 9 micron
- Max. Finished Cu Weight I.L. 2 oz.
- Smallest Mech. Drill Diameter .0079"
- Smallest Laser Drill Diameter .003"

Product Focus: Power Supplies for Medical Use

Compliance for Care

Over the past year, there's been an increasing trend toward new products that have some sort of application or industry focus. That means supplies that include either certifications, special performance specs or tailored packaging intended for a specific application area such as medical.

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

If you look at the power supply products released over the last 12 months, there's been a definite uptick in new products that have some sort of application or industry focus. While this hasn't diminished the role of general-purpose power supplies, the trend has been toward supplies that include either certifications, special performance specs or tailored packaging intended for a specific application—and medical is just one such application area.

Medical applications face strict regulatory requirements. Power supplies used in these applications must provide the characteristics to meet the necessary standards for medical use in a hospital or residential home (**Figure 1**). The worldwide ES/IEC/EN 60601-1 3rd Edition standard provides guidelines. According to RECOM, these are more stringent compared to guidelines for commercial or industrial appliances. Higher safety standards and lower electromagnetic interference (EMI) are required.

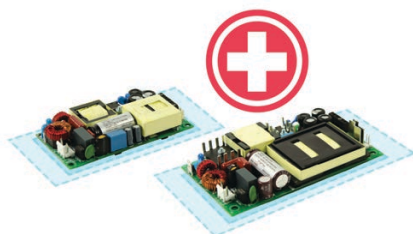
To feed the needs of medical equipment developers, power supply vendors are

building medical grade DC-DC and AC-DC power supply series that provide reinforced isolation with two means of patient protection (2xMoPP), low leakage (BF and CF ratings) and other medical specific features. Reinforced isolation provides an additional level of safety beyond the standard functional isolation to comply with the medical safety standard ES/IEC/EN 60601-1 3rd Ed. The 3rd edition of IEC 60601 focuses on the safety of operating personnel and patients. To assess medical electrical equipment, a new classification system MOP (means of protection) was introduced.

The product falls into category MOPP when there will be direct physical contact between the device and the patient and then must meet extremely stringent safety standards, especially in regard to insulation. To protect patients against electric shock, such products must feature two separate insulation barriers. The closer the contact between the device and the patient, the lower the permissible leakage current. Application devices are, therefore, classified according to the type of contact with the patient. **E**

FIGURE 1

Medical applications face strict regulatory requirements. Power supplies used in these applications must provide the characteristics to meet the necessary standards for medical use in a hospital or residential home.



AC-DC Power Supplies Feature Small Footprint

The VMS-180 series from CUI is a 180 W high density open frame AC-DC power supply certified to the medical 60601-1 edition 3.1 safety standards for 2xMoPP applications and 4th edition EMC requirements. Packaged in a low profile, industry standard 2" x 4" footprint with high efficiency up to 92%, these medical power supplies are well suited for space-constrained medical and dental applications where reliability and regulatory compliance are major concerns.

- Up to 180 W continuous power
- -40°C to +70°C temperature range
- Input voltage: 80 V to 264 V
- Efficiency up to 94%
- Industry standard 2" x 4" footprint
- 12 V auxiliary fan output
- Medical 60601-1 4th edition safety approvals
- Designed for 2xMoPP applications

CUI | www.cui.com

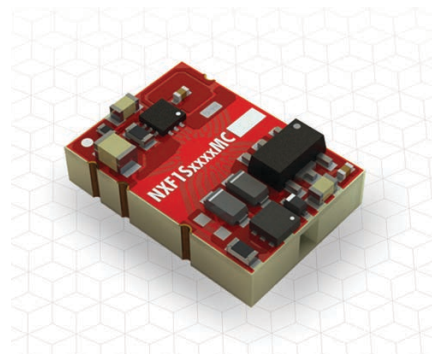


1 W Converters in Encapsulated SIP-7/SMD Packages

MINMAX Technology's MINMAX MAU01M/MSCU01M series is a new range of high performance 1 W medical safety approved DC-DC converters with encapsulated SIP-7 and SMD packages. They are specifically designed for medical applications. The series includes models available for input voltages of 4.5 VDC to 5.5 VDC, 10.8 VDC to 13.2 VDC, and 21.6 to 26.4 VDC. The I/O isolation is specified for 4,000 VAC with reinforced insulation and rated for a 300 V_{RMS} working voltage.

- Industrial standard SIP-7 package
- I/O isolation 4,000 VAC with reinforced insulation
- Low leakage current less than 2 µA
- Operating ambient temp. range -40°C to 95°C
- Short circuit protection
- Medical EMC Standard with 4th Edition of EMI EN 55011 and EMS EN 60601-1-2 Approved
- 2xMOPP per 3rd Edition of IEC/EN 60601-1 & ANSI/AAMI ES60601-1 approved with CE marking

MINMAX Technology
www.minmaxpower.com



Regulated 1 W Converters Feature Low-Profile SMT Package

The NXF1 series of DC-DC converters from Murata Power Solutions is used where a tightly regulated supply is required. They are ideal for situations where the input voltage is not tightly controlled. The single rail regulated output makes the ideal choice to power sensors, such as pressure transducers, Hall effect sensors and mass airflow sensors.

- UL60950 recognition pending
- Output regulation less than 1%
- Single isolated output
- 3 kVDC isolation "Hi-Pot Test"
- 3.3 V and 5 V input
- 3.3 V and 5 V output
- SMD construction
- Patent protected
- Short circuit protection

Murata Power Solutions
www.murata-ps.com

Power Supplies for Medical Use

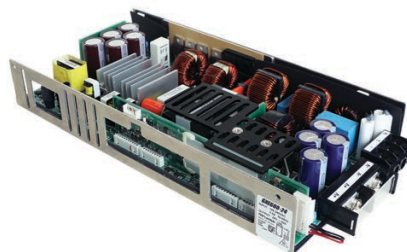


DC-DC Converters Meet Critical Medical Certifications

RECOM's REM15-W, REM20-W and REM30-W series are medical-grade DC/DC modules in a power range from 15 W to 30 W. They can be easily integrated into medical designs due to their compact case size and full medical certifications. They feature reinforced isolation of 5 kVAC, low leakage current, while offering extra wide 4:1 input voltage ranges.

- 60601-1 3rd Edition Certification
- 5 kVAC reinforced isolation rated at 250 VAC
- Case sizes: 1.6" x 1" and 2" x 1"
- Extra-wide 4:1 input voltage range
- Efficiency of up to 90%
- -40°C to 105°C operating temperature
- Single output options: 5 V to 24 V
- Dual outputs either 5 V, 12 V or 15 V

RECOM
www.recom-power.com



600 W Programmable Power Supplies Certified to Medical Standards

TDK-Lambda's GXE600 AC-DC programmable power supplies are a convection cooled 600 W-rated series is certified to the medical IEC 60601-1 and industrial IEC 60950-1 and IEC 62368-1 standards. The GXE600 is suitable for medical, industrial, test and measurement applications requiring wide-range output voltage and current programming, with sensitivity to noise from cooling fans. The GXE600 is available with a nominal 24 V or 48 V output and can be programmed, using the RS-485 interface (Modbus RTU protocol) or an external 0 to 6 V analog signal, for constant voltage and constant current (CVCC) operation.

- Convection cooled
- Up to 95% efficient
- Modbus RTU (RS-485) communication
- Constant voltage and constant current modes
- Monitoring and programming functions
- Digital or analog programming
- Seven year warranty

TDK-Lambda Americas
www.us.tdk-lambda.com



Tiny 150 W AC-DC Supply Meets Med Approvals

The EPL150 series from XP Power is a range of high density, low profile, open-frame AC-DC power supplies aimed at space-critical, cost-sensitive, industrial, ITE and healthcare applications. The series features comprehensive safety agency approvals including IEC60950-1-1 for ITE, IEC62368-1 for ITE and IEC/ES60601-1 for medical. Medical approvals are for 2xMoPP from input to output and 1xMoPP from both input to earth and output to earth, suitable for BF applied-part applications.

- 100 W convection/150 W forced-cooled ratings
- 2" x 4" footprint
- Low 0.99" profile
- Class I and Class II operation
- 12 V fan output
- High efficiency, up to 94%
- ITE and Medical (BF) approvals
- High power density
- Less than 0.5 W no load input power

XP Power
www.xppower.com

Mentor®

A Siemens Business

Speed up your
PCB design verification

by up to **90%**

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

Learn More in this FREE White Paper

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

www.circuitcellar.com/mentor3



Embedded System Essentials

Side-Channel Power Analysis

Easy Path to Proof

Side-channel power analysis is a method of breaking security on embedded systems, and something Colin has covered extensively in his column. This time Colin shows how you can prove some of the fundamental assumptions that underpin side-channel power analysis. He uses the open-source ChipWhisperer project with Jupyter notebooks for easy interactive evaluation.

By
Colin O'Flynn

This month I thought I'd bring you an introduction to side-channel power analysis (again). I've covered this in past articles, but it's been a few years and I know new readers are going to be picking up this issue of *Circuit Cellar*. But don't worry—I'm doing more than just giving you a rehash of old material. My open-source ChipWhisperer project has recently had the Version 5.0 release, which uses a new interactive Python interface (using something called Jupyter notebooks). As part of this release, several new tutorials are available, and some of them cover aspects I haven't previously shown you.

In particular, I'm going to show you how

some of the fundamental assumptions around side-channel power analysis can be easily proven. It's not something for which you have to take my word. It's something you can test yourself, and experiment with the differences that show up for various firmware code you might be running.

SIMPLE POWER ANALYSIS

My intro is going to push through all sorts of examples. The first thing we'll talk about is simple power analysis (SPA). This form of power analysis commonly refers to the fact that you can see the flow of data through a system. This can be used to break code that has an execution path that depends on the secret data being processed. What sort of code might that be? We'll take a look at a simple password check as shown in **Listing 1**. That might look straight-forward—but what if you could see the loop execution time? Power analysis lets us do exactly that, meaning that we could discover which character of our password was incorrect.

The code from Listing 1 also contains a `trigger_high()` and `trigger_low()` call. Those actually provide an added piece of instrumentation being used only for our demonstration. Using a resistor in the power pin, we could see how the power is varying, as in **Figure 1**. I'm doing that with my ChipWhisperer platform, but you could use an oscilloscope or other similar piece of gear. You can see in **Figure 2** the loop has an obvious pattern, and we see four iterations through the loop.

How does that help us crack a password? We could monitor the power consumption

```
char passwd[32];
char correct_passwd[] = "h0px3";
my_read(passwd, 32);

uint8_t passbad = 0;

trigger_high();

for(uint8_t i = 0; i < sizeof(correct_passwd); i++){
    if (correct_passwd[i] != passwd[i]){
        passbad = 1;
        break;
    }
}
```

LISTING 1

An example of a simple password check, where if one could figure out the loop count, one could recover the password byte-by-byte.

of the device and send every possible first character of the password. When we see a change in the power trace, we know that suddenly another code path was taken. Most likely this “other code path” is in fact the loop going into the second iteration. We don’t need to be clever or look for a specific signature. We just look for “different.”

It’s hard to hide this difference. If we add a random delay afterward, we can still see the time at which the power traces changed. We can notice that at this point in time whether it seems to go into a busy-wait loop or continues processing data. If you don’t believe me, there is an exact example of this in the open-source ChipWhisperer Jupyter.

So, if you think you’re clever, you’ll implement the code as in **Listing 2**. This takes the same amount of time, no matter what code is executed. Let’s see how to break that.

DATA AFFECTS POWER

What if I told you that the very data being processed affect the power consumption? The theory behind this is fairly simple. Internal to the device, a data bus consists of wires over a ground plane. Changing the voltage on this is equivalent to charging and discharging a capacitor. As a nice feature, most internal data-buses go to an intermediate state between valid data transmissions. These intermediate states mean that every time we send a value across the data-bus, we have to charge a certain number of data lines to the ‘1’ state. If we looked at the power consumed on the VCC rail, we would expect to see spikes related to the data being sent across a bus. If all the bits of the data-bus were going high, we would expect to see larger spikes than when only one or two lines went high. You can see our expected results in **Figure 3**.

But how could we test that? We could send some data to a chip, and try to find a location where, for example, we see a strong difference in the power being used that depends on the data. Since we expect our signal to be very weak, we might need to average a number of such traces over time.

Let’s try that experiment! To do this, I’m actually going to target a specific area of an encryption algorithm. Why not just plot the location of the received data directly? While we could do that, there is a good chance we’d trick ourselves into thinking the signal is stronger than it really is. In particular, if I use a serial interface to send 0xFF on repeat, I would also expect a strong signal because this serial interface is being driven to one level! So instead, I’m going to use something like the setup in **Figure 4**, where the input data are passed through a random look-up table. This look-up table is reversible, which is to say that every one of the 256 inputs maps

to only one output.

Including this look-up table means that to get a certain number of ones at the output, we are sending an unrelated number of ones in the input. For example, to get seven bits set to one, we could look at the output 0xF7 or 0x7F. In our particular look-up table that would mean that 0x26 and 0x6b, respectively, get sent—those with 3 bits and 5 bits. Therefore, the messages going over the serial

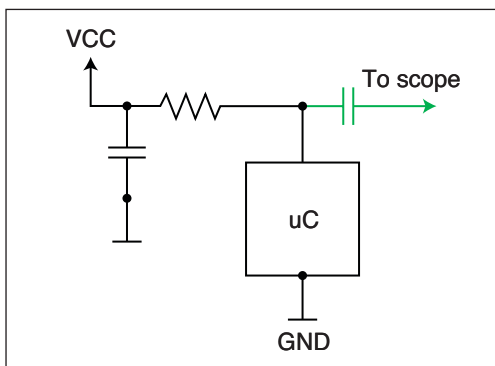


FIGURE 1

Power consumption can be measured with a resistor in the VCC line of the device. Here I’ve also removed decoupling capacitors to improve the strength of the signal. We AC-couple the measurement to remove the high-DC bias, since we are looking at small variations only.

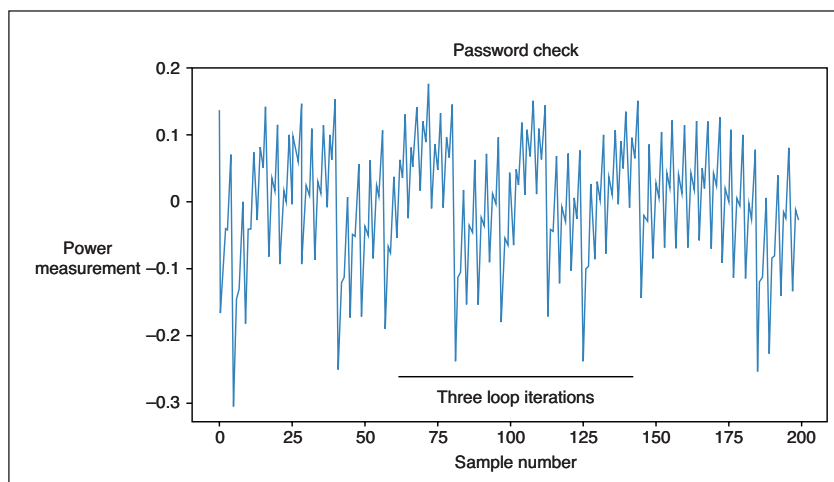


FIGURE 2

A power trace of the loop execution helps you understand how many iterations through the loop your code ran, which could break the password check in Listing 1.

```
char passwd[32];
char correct_passwd[] = "h0px3";
my_read(passwd, 32);

uint8_t passbad = 0;

trigger_high();

for(uint8_t i = 0; i < sizeof(correct_passwd); i++){
    passbad |= correct_passwd[i] ^ passwd[i];
}
```

LISTING 2

An example of a time-independent password check that could be broken by looking at power consumption of the device.

port wouldn't have the same number of bits set. As a result, we wouldn't expect to see as much influence on our signal from just the serial data line.

How do we know where the actual output is? Our evaluation path will be to send hundreds or even thousands of random bytes to firmware that is executing the code in Figure 4. Because we know the input to the look-up table—we sent it after all—we know the output of the look-up table. This means we could put every possible trace into a “group”

based on the number of ones in the output of the look-up table. This number of ones will be the “Hamming weight” of the data.

VALIDATION TIME

Based on my claims from Figure 3, I would expect that at some point in time, there would be a measurable difference in the power consumption. Let's try to validate that now. If we plot the average of each group, there is a point where the averages noticeably diverge, as in **Figure 5**. The final confirmation is to plot the value at this point in time versus the Hamming weight of the data, which has an almost linear fit (**Figure 6**). Note that we are actually measuring a one-sided voltage drop across a shunt resistor, which we AC-couple to remove the DC bias. Importantly, if we were drawing no current, we actually would have a zero voltage. As we increase our instantaneous current draw, we will see a higher negative value due to the AC-coupling. Therefore, in this graph we see a lower voltage measurement for more bits set to one. It's a minor detail, but in case you were wondering why things are “flipped,” there is a reason behind it.

What is really interesting is that all “other” points have very little difference. This makes sense if you consider that any part not within our data of interest has been effectively randomly sorted into different groups. If you take the average of each group, it makes sense that the average power consumption at these other points is the same. So, we not

FIGURE 3

Different numbers of bits being set to '1' on the power trace result in different power consumption on each clock cycle.

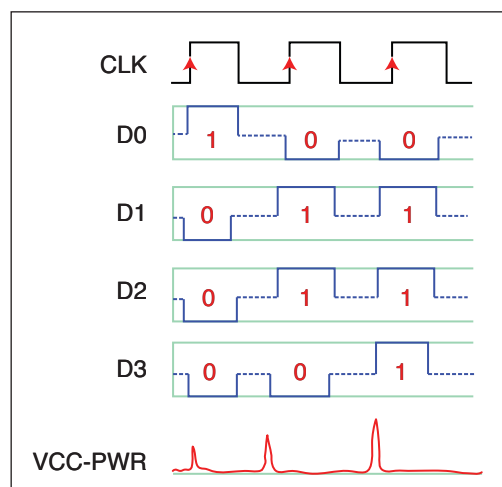


FIGURE 4

A very simple piece of code that simplifies our validation of the power consumption is related to number of bits set to one on a data bus.

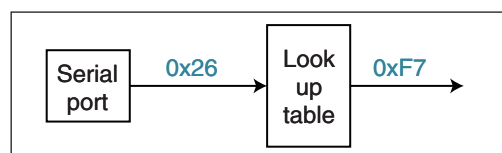
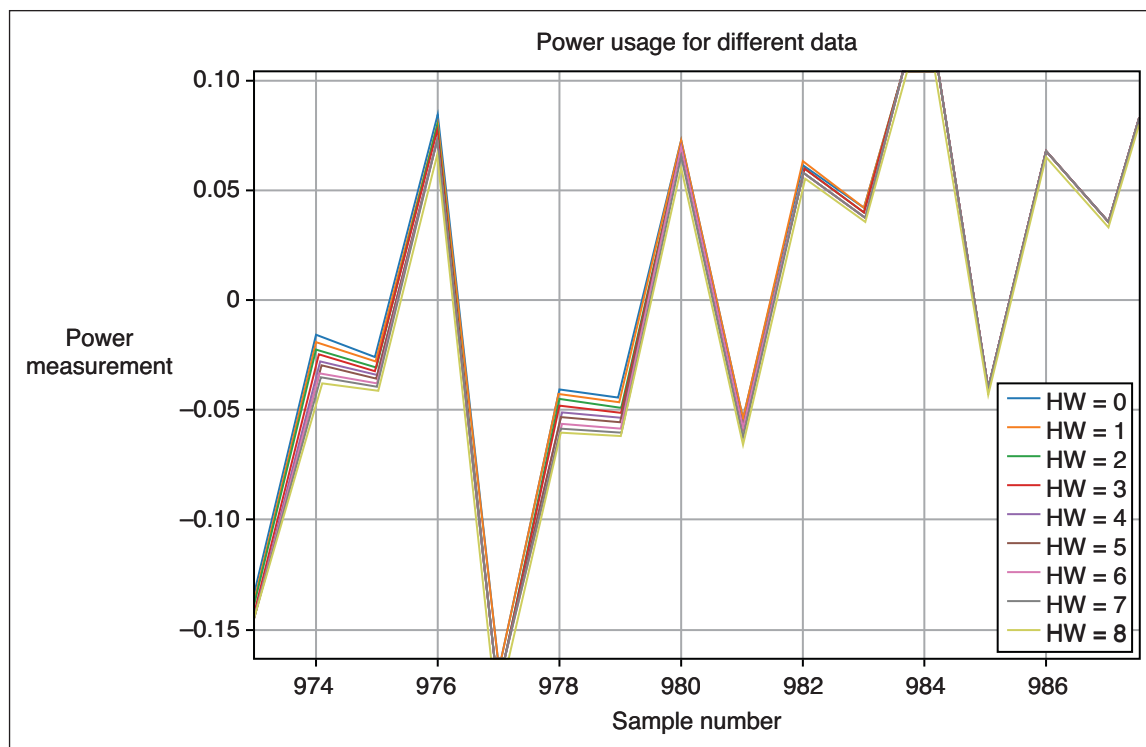


FIGURE 5

The average power consumption for different output Hamming weights (bits set to one) of the look-up table in Figure 4.



only see the relationship between data being processed and power consumption, but we also see the point in time where our look-up table was used.

You might wonder how this works for an attack. It turns out the start of the AES algorithm looks very much like Figure 4, except that, instead of just taking the input data, the input data is XOR'd with a byte of the secret key (**Figure 7**). We know from Figure 5 that we can detect the averages diverging if we group each power trace into the correct Hamming weight, and from Figure 6 that there is a strong linear relationship.


The “attack” is that we can recover bytes of the encryption key, one byte at a time. This means we can break a 16-byte AES-128 or a 32-byte AES-256 key in reasonable time—even just a few minutes. This is done by sending random (known) input data. We don't need to observe the output of the algorithm in this example.

We could then guess every possibility for 1 byte of the secret key marked in Figure 7. For each of the 256 possibilities for this byte, we could perform the same procedure and group power traces into “possible” Hamming weight groups. If our byte of the secret key matches the real secret key byte, we should be able to create plots similar to those in Figure 5 and Figure 6. If the guessed key does not match, we expect the average of each power trace to overlap as it did for all non-interesting positions.

A GOOD MATCH?

We can quantify how good the “match” is simply by looking for the linear relationship that we found from our experiment that created Figure 6. This is done by measuring the correlation between Hamming weight and power at each point in time. The correlation can be thought of as a measure of how close to a straight line the data points create. Plotting the correlation over time for each guess, you would see that at the same point in time that we saw the output of the lookup table, we also see a strong correlation. This strong correlation suggests that there is a solution to the problem of “what value of the secret key would allow us to properly group the power traces?”

The solution to that problem becomes a single byte of the encryption key. We can repeat this for each successful byte, and suddenly you've broken an AES-128 encryption key. This simple idea underpins one way that side-channel power analysis works to derive encryption keys by monitoring only the input data and the power consumption. The fun part is trying to prove the basics yourself. If you need more hints, check out the ChipWhisperer 5.0 release, which includes

several Jupyter-based tutorials that will help you recreate the figures used in my article—except you'll be recreating them with real measurements, not just simulations or assumptions. 

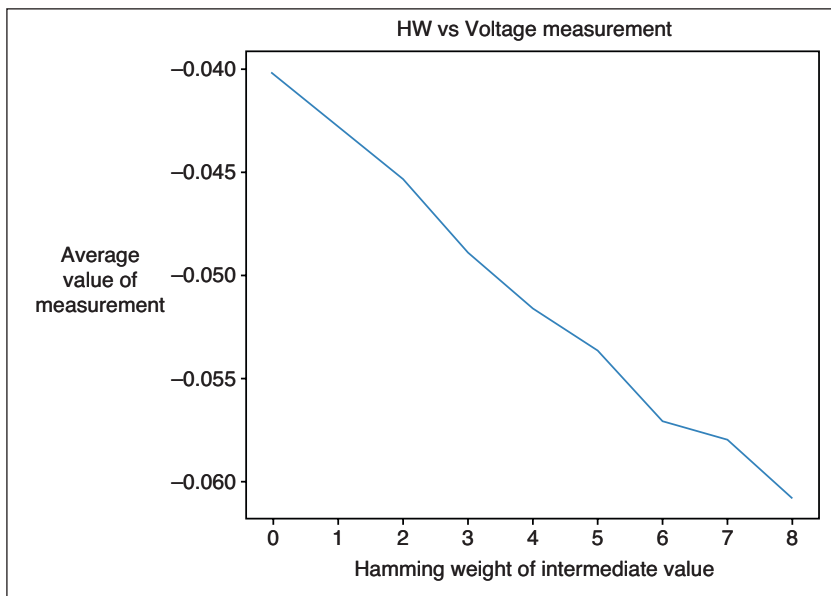


FIGURE 6

We can see a strong linear relationship between the number of bits set to one on the bus (Hamming weight) and the power consumption at a specific instant in time.

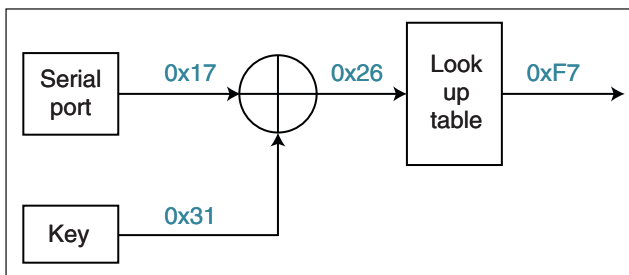


FIGURE 7

The beginning of the real AES algorithm involves a secret key being XOR'd with some input data, before being passed through a look-up table (Substitution Box).



ABOUT THE AUTHOR

Colin O'Flynn (colin@oflynn.com) has been building and breaking electronic devices for many years. He is an assistant professor at Dalhousie University, and also CTO of NewE Technology both based in Halifax, NS, Canada. Some of his work is posted on his website at www.colinoflynn.com.

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

RESOURCES

NewE Technology | www.newae.com

The Consummate Engineer

Cooling Electronic Systems

Beat the Heat

Any good embedded system engineer knows that heat is the enemy of reliability. With that in mind, it's no surprise that an increasing amount of engineering mindshare is focusing on cooling electronic systems. Here, George examines the math and science around cooling, and looks at several cooling technologies—from cold plates to heat pipes.

By
George Novacek

Every engineer worth his salt considers heat the reliability killer. And as new systems cram more functionality at higher speeds into ever smaller packages, it is no wonder that a whole engineering discipline has arisen specializing in electronics packaging and cooling. In this article we'll review some of the essentials of this engineering expertise.

The reduction of reliability of typical integrated circuits (ICs) is illustrated in **Figure 1**. The red trace represents the reliability of a digital CMOS, and the blue trace of an analog IC plotted as the Mean Time To Failure (MTTF) hours with respect to the operating temperature. I also plotted the respective failure rates λ (Greek letter Lambda) per operating hours, where λ is the reciprocal $1/\text{MTTF}$. The MTTF is more intuitive for the majority of readers, but reliability engineers prefer to think in terms of the failure rate. The orange trace is for the digital, and the violet trace is for the analog ICs. Clearly, the loss of reliability is significant.

For the purpose of this analysis, we assume the product reliability is affected by its operating temperature only. We do not consider parameter drifts, overloading, insufficient derating or otherwise poor design, potentially leading to inferior performance or a failure. By treating reliability as a function of the operating temperature only, we can appreciate how it can be improved by cooling.

THREE WAYS TO COOL

Electronic components can be cooled by conduction, convection and radiation. Conduction

is the most common method of heat transfer, occurring through a physical contact—where the heat is transferred through direct molecular collision. The amount of transferred heat depends on the temperature gradient, cross-section of the material, the length of the travel path and physical properties of the material. The transfer ceases when the two materials are at the same temperature. The rate of conduction is shown by the equation:

$$Q = \frac{k \times A \times \Delta T}{d}$$

where Q is the heat transfer rate, k is the thermal conductivity, A is the heat transfer area, ΔT the temperature difference and d is the thickness of the barrier. A typical example of conduction is heat transfer from a semiconductor to a heatsink.

Convection occurs when a fluid substance (air, water and so forth) absorbs energy from a source and removes it. For example, the air heated by a heatsink rises and carries the heat energy away. Therefore, a typical heatsink employs conduction on one side and convection on its other side.

$$Q = h_c \times A \times (T_s - T_f)$$

where Q is the heat transfer rate, h_c the heat transfer coefficient, A is the heat transfer area, T_s temperature of the surface and T_f the temperature of the fluid—which can be the air or a liquid.

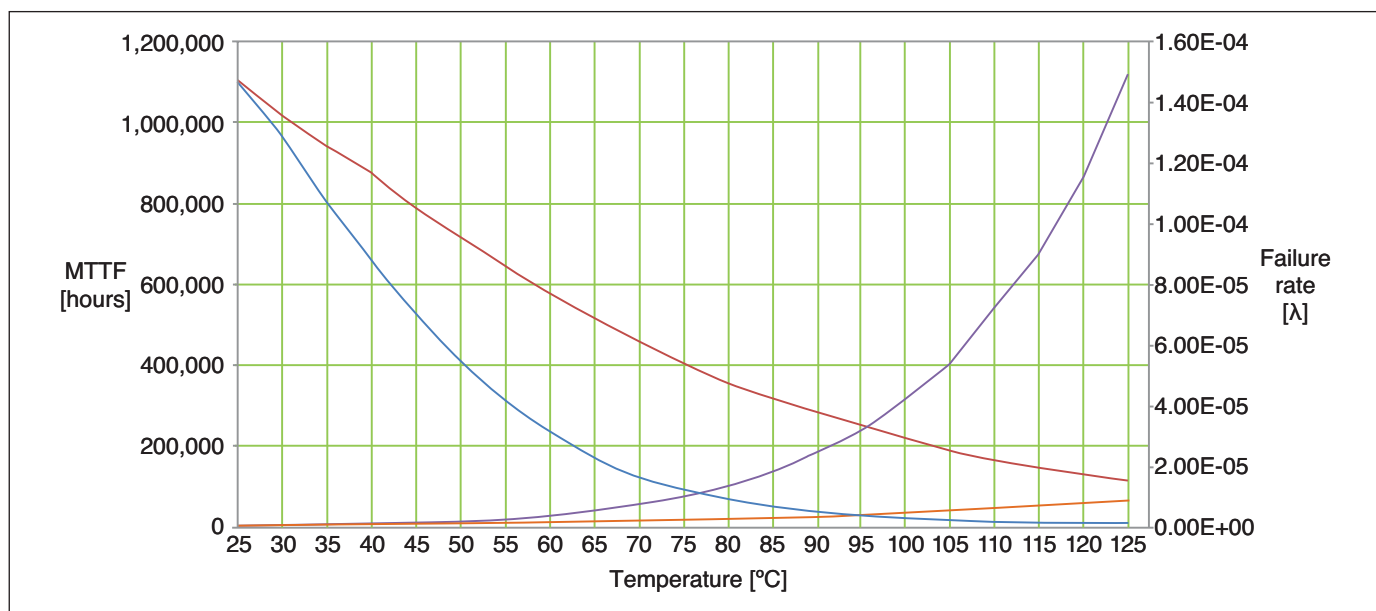


FIGURE 1
Reliability and failure rates of digital (red and orange) and analog (blue and violet) ICs versus temperature.

Thermal radiation emanates from all hot objects and is carried away through media—such as vacuum, air and so forth—as is defined by the Stefan-Boltzmann law:

$$P = e \times \sigma \times A \times (T_r^4 - T_c^4)$$

where P is the net radiated power, A is the radiating area, T_r the radiator temperature and T_c the temperature of the surroundings, e is emissivity and σ (Greek letter Sigma) the Stefan-Boltzmann constant. Radiation requires Blackbody and is rarely efficient. The most popular cooling methods used by the industry are listed in **Table 1**.

Natural convection by heatsinks has been around for a long time. It is easy to implement, robust, highly reliable due to the absence of moving parts and inexpensive. All that said, it is limited in its use by its ability to dissipate no more than about 300 W of heat—and that is quickly making it inadequate for the high-power densities of present-day equipment.

COLD PLATES

When natural convection cannot do the job, Cold Plate (Base Plate) cooling can dissipate up to about 500 W. Cold plates are essentially heatsinks that are a part of the chassis, extending over the entire electronic assembly and exposed to the ambient temperature (**Figure 2**). Forcing air or a cooling liquid through the cold plate can increase its cooling capacity manifold.

By adding internal fans or using an external source of air, the cooling effectiveness of a system can be increased up to approximately

2,000 W. A desktop PC is a good example of such a cooling method. Remember, though, that to cool the entire assembly evenly without creating hot spots requires careful placement of fans and internal components. The unfortunate side effect of forced-air cooling is noise.

Forced-air cooling capacity can be doubled by adding to the enclosure liquid conduction cooling. There, enclosure walls are cooled by liquid circulating through tubing embedded in the chassis. The flow channels in the chassis walls are optimized for the available liquid pressure and the internal heat source distribution.

Air flow-through (AFT) and liquid flow-through (LFT) methods provide a sealed environment, which also serves to protect critical processing modules from contamination by dust, humidity and other factors. Each processing board is physically



ABOUT THE AUTHOR

George Novacek is a retired president of an aerospace company. He is a professional engineer with degrees in Automation and Cybernetics. George's dissertation project was a design of a portable ECG (electrocardiograph) with wireless interface. He's been contributing articles to *Circuit Cellar* since 1999.

Cooling Approach	Chassis Cooling Method	Module Cooling Method	Approx. Module Power Dissipation [W]	Limit for Chassis Power Dissipation [W]
Natural convection	Natural air convection	Conduction cooled modules	50 W	300 W
Cold plate	Conduction with cool plate	Conduction cooled modules	75 W	500 W
Forced air conduction	Convection by forced air. Using internal fans or external air supply	Conduction cooled modules	150 W	2,000 W
Liquid conduction	Liquid cooled through chassis	Conduction cooled modules	200 W	3,000 W
Cooled air flow	Air flow through (AFT) chassis	Air flow through AFT modules	200 W	3,000 W
Cooled liquid flow	Liquid flow through (LFT) chassis	Liquid flow through LFT modules	600 W	8,000 W
Forced cooled air	Convection cooling by forced air	Air cooled modules	150 W	3,000 W
Cooled by air/air heat exchanger	Convection cooling by forced air	Air cooled modules	100 W	1,000 W
Liquid/air heat exchanger	Liquid cooled through heat exchanger	Air cooled modules	100 W	1,500 W
Spray cooled	Direct impingement on ICs	Modules compatibility with coolant	700 W	8,000 W

TABLE 1

Typical cooling methods of electronics



FIGURE 2

Cold plate chassis. The fan has been removed from the round hole.

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

RESOURCES

Digi-Key | www.digikey.comMouser Electronics | www.mouser.com

separated from the others and has its own inlet and outlet for the cooling medium. To maximize the cooling efficiency, cooling channels bring the coolant to the high-power components first, and the coolant is chilled in an external heat exchanger. The exchanger can be air-to-air or liquid-to-air. On some aircraft, for example, the fuel tank is used as a coolant for the heat exchanger.

A single specific component may sometimes need effective cooling in excess of what the existing cooling method can provide without addition of a costly, often complicated mechanical channeling method described above. That's when thermoelectric cooling with a Peltier module may be the right solution. Peltier modules—also called thermoelectric coolers (TECs)—are like electronic heat pumps with no moving parts. TECs are available from a variety of vendors, including Digi-Key and Mouser Electronics. At some point the amount of heat they generate cannot be dumped easily without significantly increasing the ambient temperature. Meanwhile, TECs consume considerable amounts of energy, which may not be an acceptable solution for the given system.

HEAT PIPES

Enter heat pipe technology—a two-phase cooling system popular in laptop computers (**Figure 3**). To begin, a tubular heat pipe is filled with a cooling fluid. Its thermal conductivity is several hundred times that of a copper rod of the same dimensions, but it weighs less. The heat pipe transfers the heat from a component, such as a microprocessor, to an external heat sink radiator. It consists of a vacuum-tight pipe, a wick and a working fluid. The evacuated pipe is filled with a small amount of the working fluid—just enough to saturate the wick—and sealed. In consequence, an internal equilibrium of vapor and liquid develops.

Heat from the cooled component warms up the evaporator end of the pipe. This upsets the equilibrium by

generating more vapor, which is at a slightly higher pressure. The higher-pressure vapor travels to the condenser, which is at a lower temperature, where it condenses, giving up its latent heat. The condensed fluid then travels back to the evaporator through the capillary action of the wick made of sintered powder. It works even against gravity.

The heat pipe operation is passive—there are no moving parts. The pipe is energized by the heat extracted from the cooled device, requiring no additional source of energy. It is simple to design and manufacture with the MTTF estimated well in excess of 100,000 hours. There are various coolant fluids in use, water and ethanol (alcohol) being the common ones. Fluorocarbon FC-77, for example, is suitable for high performance operation from about -95°C to +200°C (-139°F to +392°F) ambient. The heat pipe, however, is not a heat pump like the TEC. Its condenser must always be at a lower temperature than the evaporator for it to work.

We've touched on the mainstream methods for keeping your electronic controllers cool. Driven by ever increasing cooling demands, new cooling methods are being developed every day for embedded systems. Large systems—such as communications centers—

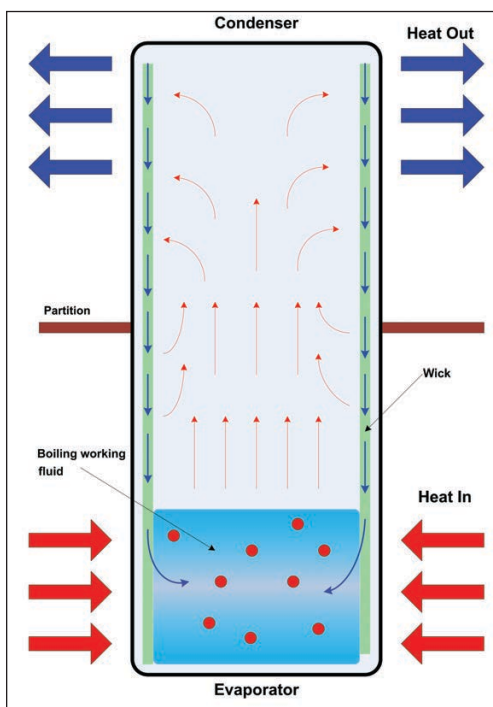



FIGURE 3
The heat pipe principle of operation

are a different story. They are often furnished with independent air or liquid cooling generation systems, and are often located in fully air conditioned buildings. 

We're raising the bar
on the possibilities for
cellular IoT.

With the dawn of 5G, cellular IoT will connect billions of appliances, physical objects, and devices to the internet—using the same mobile networks as smartphones.

WilsonPro offers enterprise-level solutions designed to give you strongest cell signal possible, including cellular signal in the 5G spectrum.

Learn more at **WilsonPro.com**

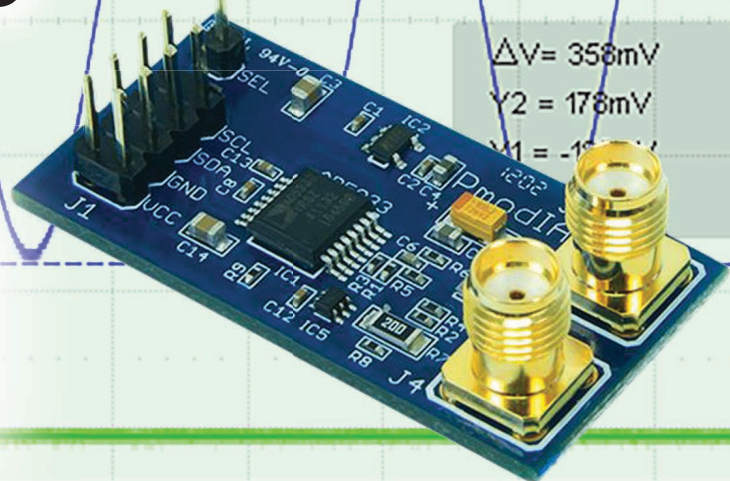


Picking Up Mixed Signals

Impedance Spectroscopy Using the AD5933

From IC to Instrument

Brian discovered an Analog Devices chip that enables you to perform impedance spectroscopy—in other words, it can measure impedance over a range of frequencies. In this article, he explores the technology, circuit design and software that serve these efforts.



By
Brian Millier

I've built and serviced many instruments for university chemistry teaching labs. Some of them involve electrochemistry. As a result, I've designed many voltammographs and conductivity meters over the years. I recently came across a chip made by Analog Devices that performs nearly all the functions needed to do impedance spectroscopy.

Impedance spectroscopy is the measurement of a device's impedance—or resistance—over a range of frequencies. For electronics enthusiasts, common uses would be measuring the impedance of transducers

such as speakers and microphones. However, it can also be used to measure electrochemical processes, in which case it is called Electrochemical Impedance Spectroscopy (EIS). In the biomedical field, its uses include detection and identification of specific forms of viruses in blood samples.

EIS is also useful for monitoring corrosion in metals, such as aluminum and steel. In situations where it's difficult to use the common visual inspection method, EIS can be used instead. Such measurements can be particularly viable if the EIS instrument is made at a low cost and miniaturized. I

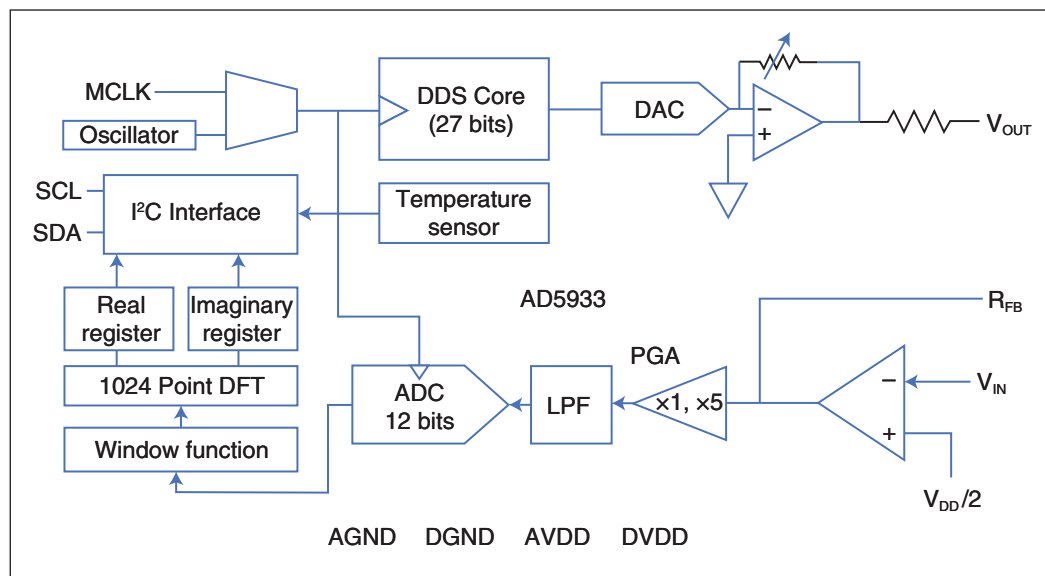


FIGURE 1
Functional block diagram of the Analog Devices AD5933 Impedance Converter chip

am guessing that this last use case was what Analog Devices engineers had in mind when they designed the AD5933 Impedance Converter chip.

IMPEDANCE CONVERTER

The Analog Devices AD5933 contains almost everything you need to do impedance spectroscopy for certain applications. However, for other applications, its excitation output amplifier has some shortcomings, as does the transimpedance amplifier used for the current-measuring input. Later, I'll discuss the additional circuitry that can be added to the AD5933 to minimize these shortcomings.

The basic AD5933 functional block diagram is shown in **Figure 1**. The AD5933 generates the required excitation sine wave using a Direct Digital Synthesis (DDS) block. Analog Devices makes DDS devices that can produce sine wave signals up into the hundreds of megahertz range. The 100,000 Hz upper frequency limit of the AD5933's DDS would have been a cinch for them to design.

For impedance spectroscopy, you generally don't need an extremely accurate excitation waveform frequency. Therefore, the AD5933 contains an internal 16.776 MHz oscillator to clock the DDS block. This internal oscillator is quite accurate. The datasheet shows that when quantities of the chips were tested at 25°C, the least-accurate samples were all greater than 16.6 MHz and less than 16.95 MHz. The oscillator's temperature coefficient is 30 ppm/°C. If better frequency accuracy is needed, there is an internal MUX switch, which allows an external, crystal-derived clock signal to be selected under program control. The maximum clock speed in this case is also 16.776 MHz.

The DDS block's phase accumulator is 27 bits wide, but the Start Frequency register is only 24 bits wide to facilitate loading it as three 8-bit bytes. Even with this 24-bit limitation, the DDS can generate frequencies up to 100,000 Hz with a resolution of 0.1 Hz.

The equation for the Start Frequency is:

Start Freq. Code=

$$\frac{\text{Required Start Frequency} \times 2^{27}}{\text{MCLK} / 4} \quad (1)$$

Example: using the 16.776 MHz internal clock, to generate a Start Frequency of 10 kHz would require the following Start Frequency code:

$$\text{Start Freq. Code} = \frac{10,000 \times 2^{27}}{16.776 \text{ MHz} / 4}$$

Range	Excitation Voltage	Output DC Bias
1	1.98 Vpp	1.48 V
2	0.97 Vpp	0.76 V
3	383 mVpp	0.31 V
4	198 mVpp	0.173 V

TABLE 1

These are the four output voltage ranges/bias levels available from the AD-5933 for $V_{DD} = 3.3 \text{ V}$.

Start Freq. Code=320,023=0x04E217

The 24-bit word corresponding to the Frequency Increment value is similarly derived using Equation (1), but substituting the Incremental Frequency for the Starting Frequency. The maximum number of frequency increments per sweep is a 9-bit number loaded into Register address 0x88 and 0x89 and is a maximum of 511 points.

The digital output of the DDS block is fed into a DAC. The analog output from the DAC is fed into a programmable-gain output amplifier, with the voltage ranges shown in **Table 1**. **Figure 2a** shows the DAC's output at a frequency of 12.8 kHz. In **Figure 2b**, you can see that the DDS is producing a much less accurate sine wave at 119 kHz. At 119 kHz, there will only be about 35 samples per cycle, given that the DDS is being clocked at 4.194 MHz.

For chemical and biological purposes, impedance spectroscopy is performed using voltages less than 2 V_{pp} , so the AD5933's output voltage ranges are well-suited to the task. However, as I'll discuss later, the AD5933's input transimpedance amplifier (TIA) is referenced to $V_{DD}/2$, which is generally 3.3 V / 2 or 3.0 V / 2. What this means is that you cannot use all four output voltage ranges directly, since the sine wave signal being generated is referenced to one of the four different bias values shown in Table 1. You could use Range 1 (bias of 1.48 V) when $V_{DD} = 3.0 \text{ V}$ (making $V_{DD}/2 = 1.50 \text{ V}$), but all other ranges wouldn't work properly, because the TIA is referenced to $V_{DD}/2$. I'll discuss the external circuitry needed to handle this later.

The current passing through the device under test (DUT) is fed directly into the negative input of the AD5933's input amplifier (V_{in} pin). A feedback resistor from the amplifier's output must be connected to the V_{in} pin. This amplifier's positive input is connected to $V_{DD}/2$, so the incoming sine wave signal must be referenced to this $V_{DD}/2$ voltage. In this configuration, the input op amp acts as a current-to-voltage converter or TIA. Following this is a programmable

gain amplifier, with x1 and x5 amplification options. The amplified sine wave signal is then fed into a low-pass filter to remove any high-frequency noise beyond the Nyquist frequency dictated by the ADC's 1 MHz sample rate.

The amplified, filtered signal is then fed to a 12-bit ADC. The ADC has a full-scale range equal to AV_{DD} . The ADC's digital output is fed into a Discrete Fourier Transform function (DFT) performed by an on-chip DSP. This is a 1,024-point DFT—that is, 1,024 samples of the current waveform are processed for each frequency sample. Both the Real and Imaginary portions of the current are available as 16-bit two's-complement numbers. A window function is performed on the data before they undergo the DFT operation. No details of this window function are given in the datasheet, but in general—with signals whose period is not equal to the sample time for the 1,024 samples—a window function will improve the accuracy of the DFT result.

The built-in oscillator has a known temperature dependence of 30 ppm/°C. For higher accuracy, this could be compensated for if you knew the temperature of the AD5933. To address this, Analog Devices includes an on-die temperature sensor with a $\pm 2^\circ\text{C}$ accuracy. All communications with the AD5933 are via a standard I²C port. Its I²C address is 0x0D. The AD5933 is powered by a single power supply that can range from 2.7 to 5.5 V. Its operating current is typically 10 mA at 3.3 V up to 17 mA at 5.5 V.

IMPROVING THE I/O CIRCUITS

As discussed in the previous section, the AD5933 can implement the measurement part of a complete impedance spectrometer all on its own. However, for most use cases, it can be enhanced considerably with some additional analog circuitry. This topic is covered in Analog Devices CN-0217 Circuit Note. Briefly, there are two main limitations to the AD5933 when used on its own:

1. Since the device uses only a positive power source, its AC excitation waveform must be biased at some point between AV_{DD} and ground. The TIA amplifier is internally referenced to $AV_{DD}/2$, so it would make sense for the excitation waveform to be biased there as well. From Table 1, this criterion is met for Output Range 1, but all three other ranges are biased differently. Because you generally need to use all four ranges, external circuitry is needed to correct this.

2. Both the excitation amplifier and the TIA must be able to handle enough current to measure the device under test at its minimum expected impedance. For example, using the 0.97 V_{pp} excitation voltage, a DUT with a minimum impedance of 100 Ω would require an excitation current of 9.7 mA. Not only would the excitation amplifier be required to supply this, but its effective output impedance would have to be much lower in value than 100 Ω , or it would adversely affect the accuracy of the impedance reading. The AD5933's output amplifier has, at best, a 200 Ω output impedance on Range 1, with significantly higher values on the other ranges. Also, it can only supply ± 5.8 mA on Range 1—and much less on the other ranges.

Because it would be difficult for me to solder the AD5933's 16-pin SSOP packages to a PCB by hand, I had to find some form of evaluation/breakout board that contained an AD5933 and preferably the external analog circuitry needed to offset the above limitations.

Two such boards are available. Analog Devices sells the EVAL-AD5933EBZ and Digilent



FIGURE 2

Shown here are screen captures from my scope for excitation frequencies of 12.8 kHz (a) and 119 kHz (b), respectively.



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



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

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

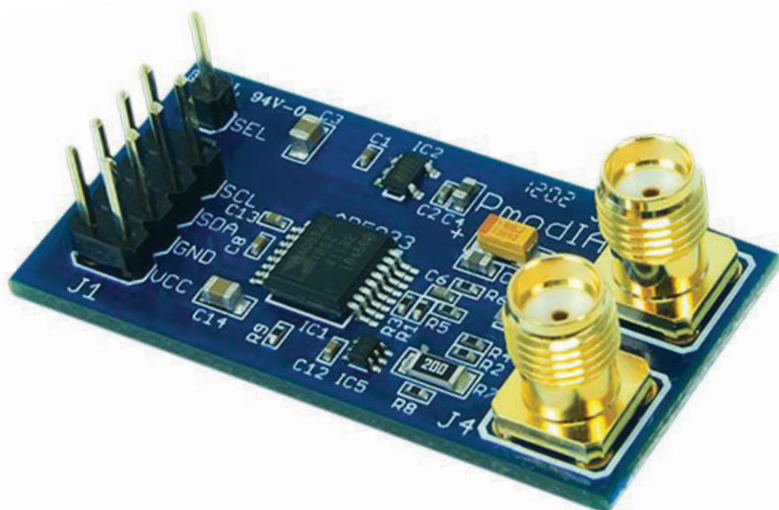


FIGURE 3 Digilent's PmodIA impedance converter module

sells a PmodIA impedance analyzer board. Both boards contain the additional analog circuitry—recommended by Analog Devices—to address the two limitations listed above. I'll discuss this analog circuitry later, but first I'll outline the other features and disadvantages of each board. Incidentally, both boards are not a whole lot more expensive than the price of the AD5933 chip alone, which was a pleasant surprise.

ADI EVAL-AD5933

The EVAL-AD5933 board contains a Cypress Semiconductor CY7C68013 MCU, which was specifically designed to interface a

USB bus to a high-speed parallel port and an I²C port. This chip is somewhat unique in that it runs firmware downloaded from the host PC via the USB port, after which time the MCU is reset and the firmware is executed. It's somewhat like an FPGA in that regard, except the CY7C68013 gets its firmware from USB rather than the SPI EEPROM chip that FPGAs typically use for program storage.

The CY7C68013's I²C port interfaces to the AD5933 directly. Analog Devices provides a Windows driver for the CY7C68013 MCU. They also provide a DLL to allow your PC application to download the CY7C68013 firmware, along with a hex file containing that firmware. The Analog Devices PC application software is written in Visual Basic 6 (VB6), which is outdated and has long been superseded by VB.NET. This would be a problem for most people, but I have used VB6 for a decade or so and am familiar with it. However, their sample program uses some custom third-party components that AD doesn't supply. I had no luck finding the source of these components. The example program that Analog Devices provides would not compile without these components being in place and licensed.

The CY7C68013 is used in many devices—notably Saleae’s Logic Pro 16 logic analyzer and its many clones. If you already have experience interfacing the CY7C68013 to your PC applications, then the EVAL-AD5933 might be a good choice for you.

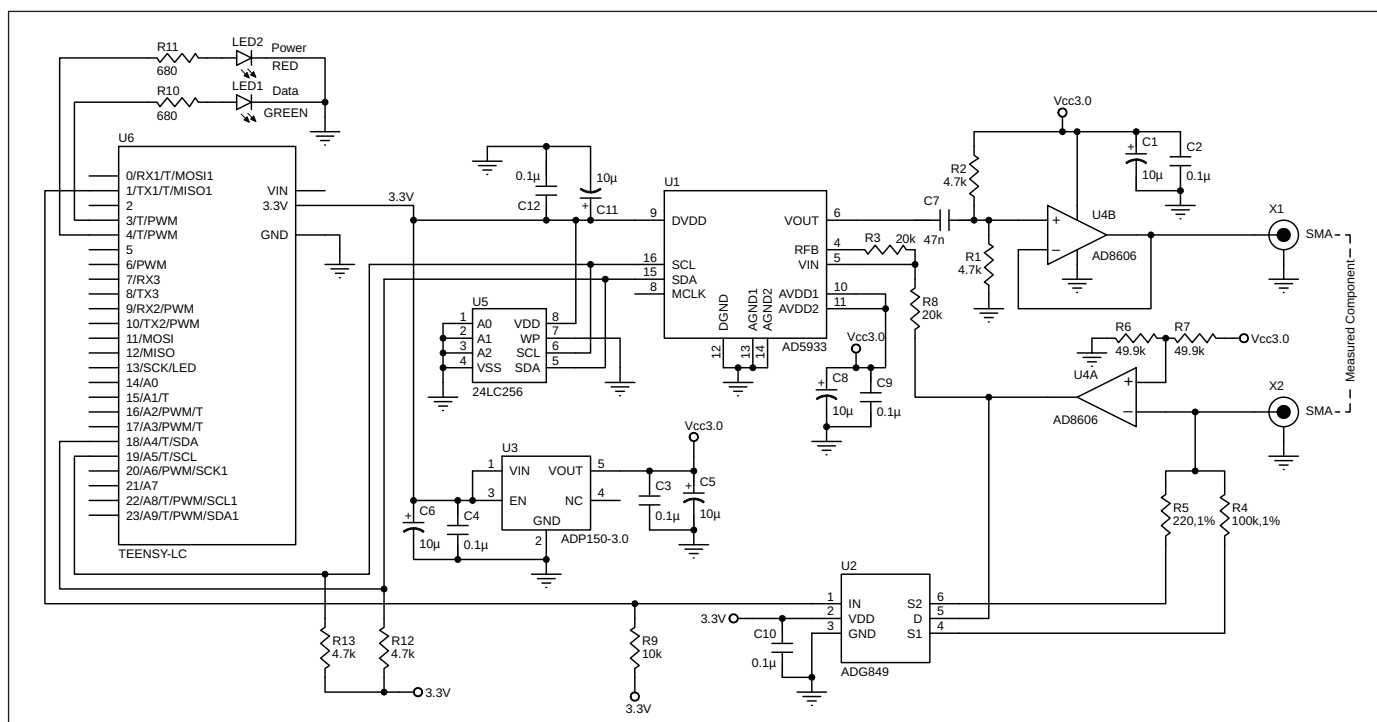


FIGURE 4
This is the schematic of the instrument that I built, using the Digilent PmodIA module, Teensy LC module, a 24LC256 EEPROM and a few LEDs.

The external TIA circuit on the EVAL-AD5933 includes only a single current-to-voltage conversion range. I feel that this is too limiting. It may be possible to remove the feedback resistor on the board, and circuitry added to support two (or more) TIA conversion ranges. I didn't have this board to try this however. This board contains a 16 MHz crystal module to provide the AD5833's clock signal, and has jumpers to allow another clock source to be used. The board is relatively large, and although it uses SMD components, it appears that some components could be swapped for alternate values—if desired.

I chose to use Digilent's PmodIA Impedance Analyzer module. It does not contain an MCU controller, which was an advantage for me. For that functionally, I prefer to use a Teensy LC development module from PJRC, which contains both a USB and I²C ports—and enough GPIO to perform some other functions. The PmodIA is less expensive than the EVAL-AD5933, even adding in the cost of the Teensy MCU module.

The external TIA circuit on this module has two ranges. The two feedback resistors are switched by an ADG849 SPDT MUX switch with an extremely low on-resistance of 0.5 Ω . To handle a wide range of impedances, this two-range TIA is a necessity so I'm glad Digilent added it. The module contains two SMA sockets, which makes it easy to add two coaxial cables to connect to the DUT. At 1.6" \times 0.8", the module is very small—I wanted to change one of the TIA range resistors, and it happened to be a large enough SMD package for me to unsolder easily. That said, nearly all the other resistors and capacitors are so tiny that it would be difficult or impossible to swap them out by hand for a different value.

Figure 3 shows the PmodIA module.

EXTERNAL ANALOG CIRCUITRY

Refer to **Figure 4** for the project's schematic. Apart from the Teensy LC module, a 24LC256 EEPROM and a few LEDs, all the other circuitry is contained on the Digilent PmodIA module. The AD5933's excitation sine-wave output is capacitor-coupled (C7) to an AD8606 unity-gain output buffer. The AD8606's input is biased at AV_{dd} (regulated 3.0 V from U3) divided by two. The capacitor coupling removes the AD5933's output bias, which varies with the chosen output voltage range. The AD8606 has a 10 MHz bandwidth and a very low output impedance of 1 Ω up to 1 MHz.

Rather than using the AD5933's input amplifier in a TIA configuration, an external TIA is used (the second section of the AD8606). It is similarly biased at $AV_{dd}/2$ by R6, R7. The current-to-voltage conversion factor is

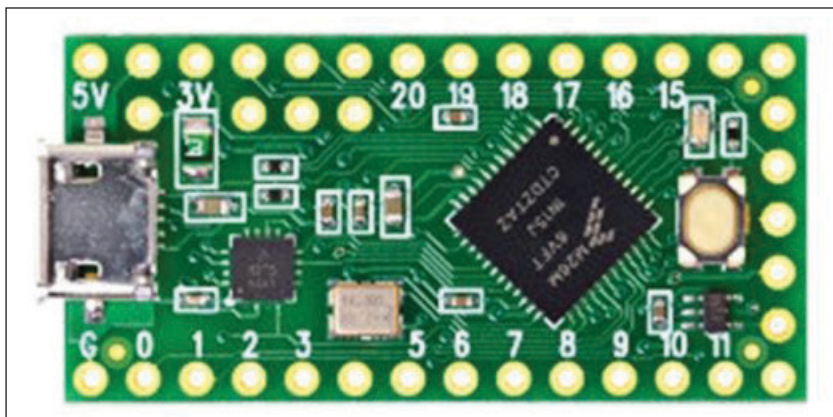


FIGURE 5

Shown here is the teensy LC MCU module used in my design.

determined by feedback resistors R4 (100 k Ω) or R5 (220 Ω)—depending upon the value of the logic signal presented to the ADG849's IN pin (which is connected to the Teensy D1 GPIO pin). I changed the PmodIA factory value of R5 from 20 Ω to 220 Ω , because that provided a better resolution for the impedance ranges I wanted.

The external TIA output signal is connected to the AD5933's input amplifier, which is now configured as an inverting voltage amplifier with a gain of 1. All the external analog circuitry, together with the analog circuitry within the AD5933 is powered by a low-noise, 3.0-V regulator (ADP150-3.0). Note that the part designations that I used for Figure 4 are not the same as those found in Digilent's PmodIA schematic. Also, note that the Teensy USB port is not shown on the diagram. The micro USB socket is mounted directly on the Teensy LC module. It does not appear on the Eagle library part that I used for the Teensy LC. The Calibration arrays are stored in U5, an I²C EEPROM chip.

TEENSY LC CONTROLLER

In the case of the EVAL-AD5933 board, the CY7C68013 MCU was basically just a dumb USB to I²C bridge. All control and measurement functions were handled by the host PC. I decided to use a Teensy LC module and have its firmware do all the control/measurement functions. In this case, the PC only performs the user interface/graphical display and data file storage functions.

At \$11, the Teensy LC is the lowest-cost Teensy module. It features an Arm Cortex-M0 MCU and a USB port. It runs at 48 MHz and contains 62 KB of flash storage. Its 8 KB of RAM is enough for this application, since the array of calibration values are stored in an external I²C EEPROM (24LC256). The Teensy LC module is shown in **Figure 5**.

The “raw” impedance values provided by the AD5933 at each frequency point must be multiplied by a calibration factor. This calibration factor varies with frequency, and is also affected by the excitation output voltage range, the TIA conversion factor and the AD5933’s internal PGA gain setting. My instrument has six discrete ranges, from 100 Ω to 1 M Ω . For each frequency sweep range, I decided that 200 data points were sufficient. Therefore, I needed six ranges \times 200 calibration values. These are single-precision floating point values, each occupying 4 bytes in memory, for a total of 4,800 bytes of EEPROM storage. Currently, my software calculates impedance but not phase. A phase calibration array would take up another 4,800 bytes of EEPROM space—still well within the 32 KB of EEPROM storage.

The Teensy LC’s USB port is enumerated by the PC host as a USB serial device, so no special Windows drivers are necessary. The Teensy LC also controls two LEDs. A bi-color LED glows RED when the instrument is idle and turns to GREEN during both the standard frequency sweep and a calibration run.

The firmware running on the LC is basically a command interpreter, accepting commands/parameters through the USB-emulated Serial port. All commands consist of a single ASCII character followed by a single parameter expressed in string format. A few commands need no parameter. The command character and the parameter (if required) are echoed back to the host. This simple protocol makes it easy to test out the instrument by running a Serial Monitor on the PC, and simply typing in the commands/parameters. In the case of both the frequency sweep and calibration commands, the respective data arrays are sent

back using an ASCII string format, with one line per frequency sample. Here again, it’s easy to quickly determine if the data being returned “look right” during early debugging.

Table 2 shows the complete command protocol. The P, R, K and Z commands enable you to set parameters individually. However, specifying the calibration range using the B command will set up all the P, R, K and Z parameters for that range simultaneously. The Teensy firmware refers to the calibration ranges as “Bands”—hence the “B” command designator.

SOME MATH EQUATIONS

For each frequency point in the data collection sweep, the AD5933 returns two 16-bit, two’s-complement values corresponding to the Real and Imaginary portions of the DUT’s impedance at that specific frequency. Note that at this stage, these are only relative numbers, which must be multiplied by a gain factor that has been pre-determined and stored in the calibration array. There is a separate gain factor for each 500 Hz section within the 1,000 to 101,000 Hz sweep range (200 values). Also, there is a separate gain factor array for each of the six calibration ranges between 100 Ω and 1 M Ω .

The impedance is defined as follows:

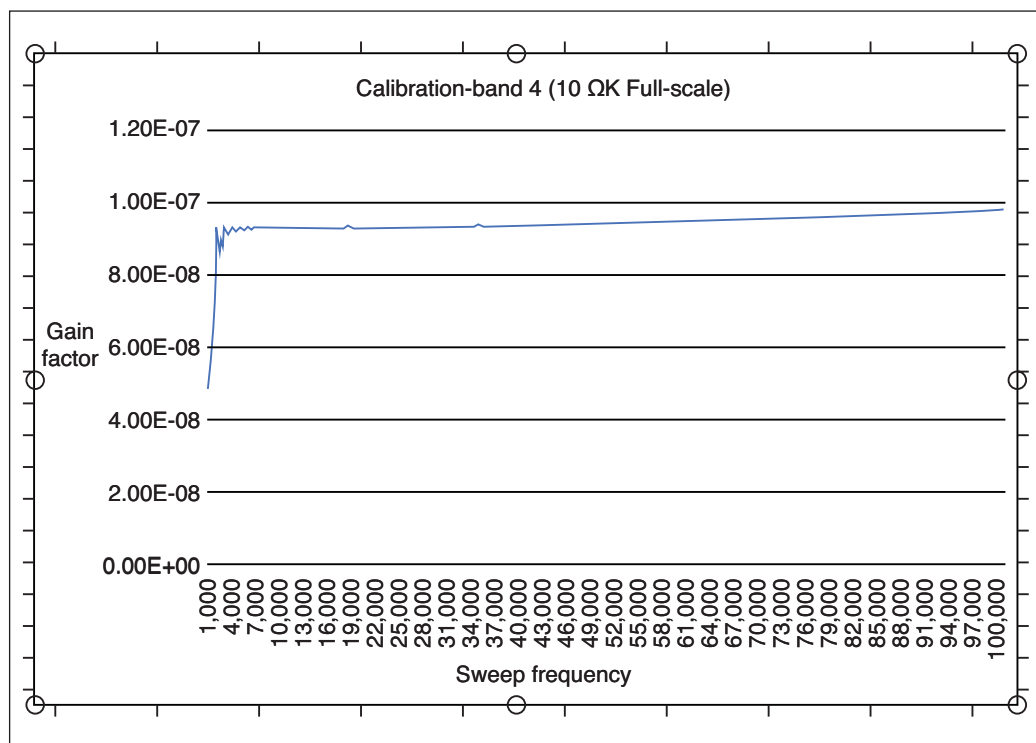
$$\text{Impedance} = \frac{1}{\text{Gain Factor} \times (\text{Real}^2 + \text{Imag}^2)^{\frac{1}{2}}} \quad (2)$$

The firmware running in the Teensy LC returns the “raw” Real and “raw” Imaginary terms, along with the actual impedance—calculated using Equation (2) and the proper Gain Factor from the calibration array. During

Command/Parameter	Description
S iiiii	Set starting frequency (32-bit integer value)
I iiiii	Set frequency increment (32-bit integer)
N iiiii	Set number of increments (9- bit integer)
P i	Set AD5933 PGA gain (1 or 5)
R i	Set output voltage range (1 through 4)
T iiiii	Set number of cycles settling time (1-2044)
G	Perform a frequency sweep and return data
K i	Set TIA conversion Gain (0 or 1)
C	Perform a calibration run using parameters from B command
B i	Set up all parameters for a given Calibration band (1 to 6)
Z iiiii	Define what reference resistance value is used for calibration

TABLE 2

Complete command set implemented by the Teensy LC MCU

**FIGURE 6**

Excel graph of the Gain Factor versus Sweep Frequency for calibration band 4 (10 kΩ full-scale)

the calibration procedure—performed on each of the six calibration bands—a known 1% tolerance resistor is connected as the DUT. Equation (2) can be rearranged. Substituting the known resistor's value for impedance, the Gain Factor can be determined for each 500 Hz section of the 1,000 to 101,000 Hz range. The 200 single-precision floating-point Gain Factor values are stored in the EEPROM using 4 bytes per value.

You might wonder why I chose to make the Teensy LC calculate and store (in external EEPROM) all the calibration data. It would be easier to let the host PC do that, with the calibration data stored in disk file(s). Most of my instrument designs are targeted at chemistry under-graduate teaching labs. I've built multiple instruments of that type, each of which is later connected to any one of many PCs located on the lab benches. With that in mind, it's important that the instrument's calibration data remain within the instrument itself.

Figure 6 is a graph of the Gain Factor versus Sweep Frequency for calibration band 4 (10 kΩ full-scale). This is basically a straight line with a slight upward slope, if you ignore the “damped oscillation” at the extreme left. This “oscillation” occurs on all impedance ranges. It results from the ADC's sampling at a fixed 1 MSPS rate and the DFT's operating on 1,024 samples. With a 1,000 Hz excitation, the 1,024 -point DFT will be sampling a little more than one complete cycle. With a 500 Hz frequency increment per sample—over the next few thousand hertz—the DFT will be

sampling a non-integral number of cycles and there will be an error introduced. As you can see, this error decreases rapidly as the frequency increases. I would guess it's a sinc rate, because that function comes up frequently in Fourier transforms on sampled data.

While the calibration provides a Gain Factor correction—at the lower end of the frequency sweep where the “oscillation” occurs—the accuracy of the data won't be as good as it is in the higher-frequency regions of the sweep. The only way to overcome the low-frequency accuracy issue is to feed the AD5933 with an MCLK of a lower frequency than the 16.776 MHz internal clock. This reduces the ADC's sample rate and drops the sinc “oscillation” proportionately lower in frequency. Slowing down MCLK would not be particularly hard to do, given that the PmodIA board has provisions on the bottom of the board for mounting an external SMD crystal module. This can be switched in/out by sending the appropriate command to the AD5933. The lower MCLK signal could alternately be provided by the Teensy LC.

Another issue arises, however, if you try to extend the low-frequency limit too much lower than 1,000 Hz. Referring to Figure 4, you can see that C7—labeled C6 on the PmodIA datasheet's schematic—is a 47 nF capacitor. At 1 kHz, this has a reactance of 3,386 Ω, which then feeds R1/R2 and presents a 24.95 kΩ load to the signal. At 1 kHz, C7's 3,386 Ω series reactance will attenuate the signal somewhat and introduce some phase

shift. If you try to extend the instrument's lower frequency limit significantly below 1 kHz, the value of C7 would need to be increased proportionately.

In practice this is very hard to do, as this capacitor is too tiny to swap out by hand. Additionally, the low-pass filter contained in the AD5933 is designed to filter out signals beyond the ADC's Nyquist frequency. The datasheet doesn't provide any details about this low-pass filter, but I assume it is designed to work when the ADC is running at 1 MSPS (with MCLK= 16 MHz). When you slow down MCLK, the ADC slows down proportionately, and I think that an external low-pass filter of a proportionately lower frequency would need to be added.

Although I didn't need it for my project's application, the phase can also be calculated using the "raw" Real and Imaginary values:

$$\text{Phase(rads)} = \tan^{-1} \left(\frac{\text{Imaginary}}{\text{Real}} \right) \quad (3)$$

The arctangent function returns the correct standard phase angle only when both the sign of the Real and Imaginary values are positive. When they're not, you must correct the phase angle accordingly. This is described in Table #7 in the AD5933's datasheet. Also, Equation (3) returns the phase angle in rads, which would generally be expressed as degrees (1 rad = 57.2958 degrees).

CALIBRATION PROCEDURE

The AD5933 device itself has four excitation voltage ranges and two ADC PGA gain settings. Also, the external TIA has two ranges. With this combination of excitation voltage and current measuring ranges, I settled on six discrete impedance ranges. A separate calibration run is performed for each of these ranges. You can see the

full-scale impedances for the six ranges, ranging from a low of 100 Ω up to 1 MΩ, in **Table 3**. For each of the six ranges, there is a combination of TIA feedback resistor, DAC voltage range (excitation voltage) and PGA gain. Each range's combination is chosen to present the ADC with about two-thirds of its full-scale range—3.0 V with the PmodIA. This gives the best possible resolution, while still providing some "headroom" for the ADC, so that it doesn't saturate from an over-range input current.

The calibration can be performed by sending the required commands to the instrument, using a serial monitor application or by using the PC application that I wrote to perform both calibration and measurements. I'll describe the first option here. First, the desired range is selected using the "Bx" command. This command sets up the excitation voltage, PGA gain, TIA feedback resistor selection and specifies the value of the "calibration" resistor that you will be connecting up as the DUT.

I decided to have the calibration done over a frequency range of 1,000 to 101,000 Hz, in 500-Hz increments. This sweep range is set by default when the instrument is powered up, so it doesn't need to be specified—unless you change it using the Start Frequency ("S" command) and/or the Frequency Increment ("I" command).

Next, you connect a 1% resistor to the DUT terminals. Sending a "C" command will perform the calibration. You will see a series of data points in the serial terminal. Each line contains the excitation frequency, the "raw" Real and "raw" Imaginary readings from the DFT and the calculated Gain Factor. An array of Gain Factors—one for each 500-Hz frequency band—is stored in the 24LC256 EEPROM chip. As a result, you don't need to do anything with the Gain Factor readings manually. When the Calibrate function is run

Cal Range	F.Scale Impedance	TIA Setting	TIA Feedback Resistor	DAC Volt Range	PGA Gain	ADC Input	Gain Value
1	100 Ω	1	220 Ω	1 V _{pp}	1	2.2 V	1.02E-6
2	500 Ω	1	220 Ω	1 V _{pp}	5	2.2 V	2.13E-7
3	1 kΩ	1	220 Ω	2 V _{pp}	5	2.2 V	9.8E-8
4	10 kΩ	0	100 kΩ	0.2 V _{pp}	1	2 V	1.28E-8
5	100 kΩ	0	100 kΩ	2 V _{pp}	1	2 V	1.18E-9
6	1,000 kΩ	0	100 kΩ	2 V _{pp}	5	1 V	9.7E-9

TABLE 3

These are the various parameters used for the full-scale impedances for the six different calibration bands of the Analog Devices EVAL-AD5933 evaluation board.

from my PC application, the 200 Gain Factor readings can be saved to a disk file, using the Save Current Run menu item. The contents of this file could then be imported into Excel and used to observe, for example, the anomaly at low-frequency readings that I mentioned earlier (Figure 6).

This procedure must be performed for all six ranges, placing the proper resistor in place for each range. To reduce noise in the reading, the Calibration routine performs four measurements at each frequency, and takes the average. In operation, different starting frequencies and sweep ranges can be specified by the user. In this case, the frequency at which readings are taken will usually not match those used in the calibration. Therefore, during normal measurement, a selection of the calibration data points is used to interpolate the measurement frequency with the frequency of the closest calibration data point. No fancy math is done here, as the Gain Factor vs. Frequency is a straight line with only a small slope (Figure 6).

COMPILING THE SOURCE CODE

All members of the Teensy module family use Freescale (now NXP) Arm processors. PJRC, the company that makes these modules, has written a “plug-in” for the Arduino IDE, which allows you to write Teensy programs in the Arduino IDE using Arduino’s simplified “C” syntax. This plug-in is called “Teensyduino” and can be downloaded from the PJRC site. The link is available on the *Circuit Cellar* article materials web page. Meanwhile, my source code (AD5933.ino) is available on *Circuit Cellar*’s article code download webpage.

When you run the Teensyduino plug-in’s installation program, you will have to specify the folder where your Arduino program is stored. After Teensyduino has installed, when you open the Arduino IDE and check the Tools-Board menu you will see a separate section called “Teensyduino.” It contains all Teensy modules. For this project, you would select the Teensy LC board. When I did this project, I was using Arduino version 1.8.5 along with Teensyduino version 1.42—both the latest, stable versions available at the time.

There is an important option that you must configure for this program to work properly. Under the Tools-Optimize menu, select “Faster.” The regular measurement routine sends out floating-point numbers for the actual impedance values, and the Calibration routine sends out floating-point Gain Factor values. Initially my Optimize setting was set for “smallest code.” I believe that must be the default, because I did not choose that setting. The “smallest code” setting prevented these

(lines 86-91 in the code)

```
If InStr(fullComPortName, "COM") <> 0 Then
    'MsgBox(fullComPortName)
    cpindex = InStr(fullComPortName, "COM")
    comPortNumber = fullComPortName.Substring(cpindex
- 1, Len(fullComPortName) - (cpindex))
    tbComportnumber.Text = comPortNumber
End If
```

LISTING 1

If you have multiple USB-serial devices attached to the PC, you would have to change this VB.net code in the program to look for the specific COM port number assigned by Windows.

formatted floating-point numbers from being sent out the USB serial port.

As I mentioned earlier, Analog Devices provides sample software to run their EVAL-AD5933 evaluation board. This software is provided as a .EXE file which runs under Windows. I didn’t purchase the EVAL-AD5933 board, so I couldn’t try out this sample program, but I assume it must work. Also, as I noted earlier in the article, I couldn’t compile the VB6 source code because third-party VB libraries were needed but not supplied. Without being able to edit/compile

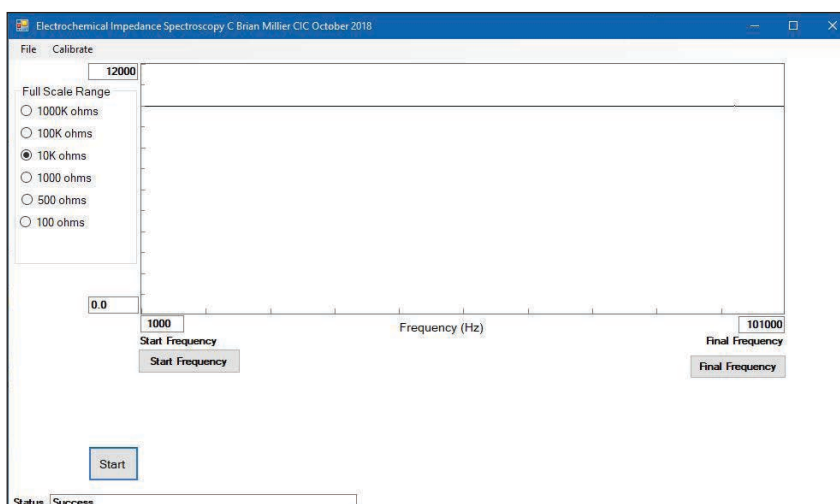


FIGURE 7

Screen shot of my VB.net PC application

ABOUT THE AUTHOR

Brian Millier runs Computer Interface Consultants. He was an instrumentation engineer in the Department of Chemistry at Dalhousie University (Halifax, NS, Canada) for 29 years.

**FIGURE 8**

Here is the finished unit, mounted in a small Hammond extruded aluminum enclosure.

Analog Devices' VB6 source code, it would have made it difficult for me to modify their sample program the way I wanted.

Given this complication, I decided against using the Analog Devices evaluation board. Instead, I used the Digilent PmodIA board with my own Teensy MCU controller. For the PC host application, rather than using the outdated VB6 compiler, I wrote it using VB.net. VB.net is part of Visual Studio Community 2015, which is a full-featured IDE with various options for C/Basic toolchains, all supplied by Microsoft at no charge.

The only "tricky" part of this program concerns its interface to the USB serial port used by the Teensy LC for communications. The Teensy's USB serial port shows up in Windows Device Manager in the Ports (COM and LPT) section as a "USB Serial Device (COMx)"—where "x" is the COM port number that Windows has assigned it. It doesn't show up as a "Teensy LC," as you might expect.


I've used USB-serial bridge adapters that show up in the Device Manager as "Prolific USB-to-Serial Comm Port" or "Silabs CP2103." These are more descriptive names that you could designate in VB.net to select the proper device, assuming you have identified its descriptive name in Device Manager.

In this case, I've written the VB.net code to search for the occurrence of "COM" in the device name. I then open the first COMx port that Windows has enumerated. This COM port number is displayed in a text box on the main program screen. If you don't have any other USB-serial devices plugged into the PC, this COM port number should be the same as the one that the Device Manager displays, and everything will work fine. If you have multiple USB-serial devices attached to the PC, you would have to change the VB.net code shown in **Listing 1** in the program to look for the specific COM port number assigned by Windows.

A screen capture of this application is shown in **Figure 7**. At the lower right of the screen, you can see that the Instrument COM port is COM9, with a green block indicating the presence of the instrument. The full-scale impedance values can be chosen from the group box to the left of the data graph area. Start and Final frequencies are set with the appropriate buttons. The six ranges are calibrated using the Calibrate Menu item. The current run can be saved to a disk file using the File menu.

In collecting data, you would start out by picking a Full-Scale Range lower than what you expect the impedance to be, and then incrementally increasing the range until a useable data graph is produced. The graph will be scaled to 120% of the full-scale value selected to allow for some "headroom." If any part of the frequency sweep produces an impedance that is beyond this 120% limit, that portion of the plot will be colored red. In this screen capture, I had a 10 k Ω , 1% resistor connected as the DUT and you can see the straight line representing the Impedance vs. Frequency.

CONCLUSIONS

Figure 8 shows the finished instrument. It's small—only about 5" long. I built it "on-spec" to see if it could be integrated into the Physical Chemistry teaching labs at Dalhousie University, where I worked for 30 years (I'm now retired). For their purposes, I expect that Phase won't be needed, so I didn't add the calculations to measure it. However, the instrument provides both the "raw" Real and "raw" Imaginary values of the impedance, and the Phase can easily be calculated from Equation (3) shown earlier. In addition, one would need to save six Phase calibration tables in EEPROM. That's because the AD5933's internal signal path contains circuitry that results in an inherent phase shift that requires compensation. The 24LC256 EEPROM that I used has plenty of room to store this extra Phase calibration data. 

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

RESOURCES

Analog Devices | www.analog.com

Cypress Semiconductor | www.cypress.com

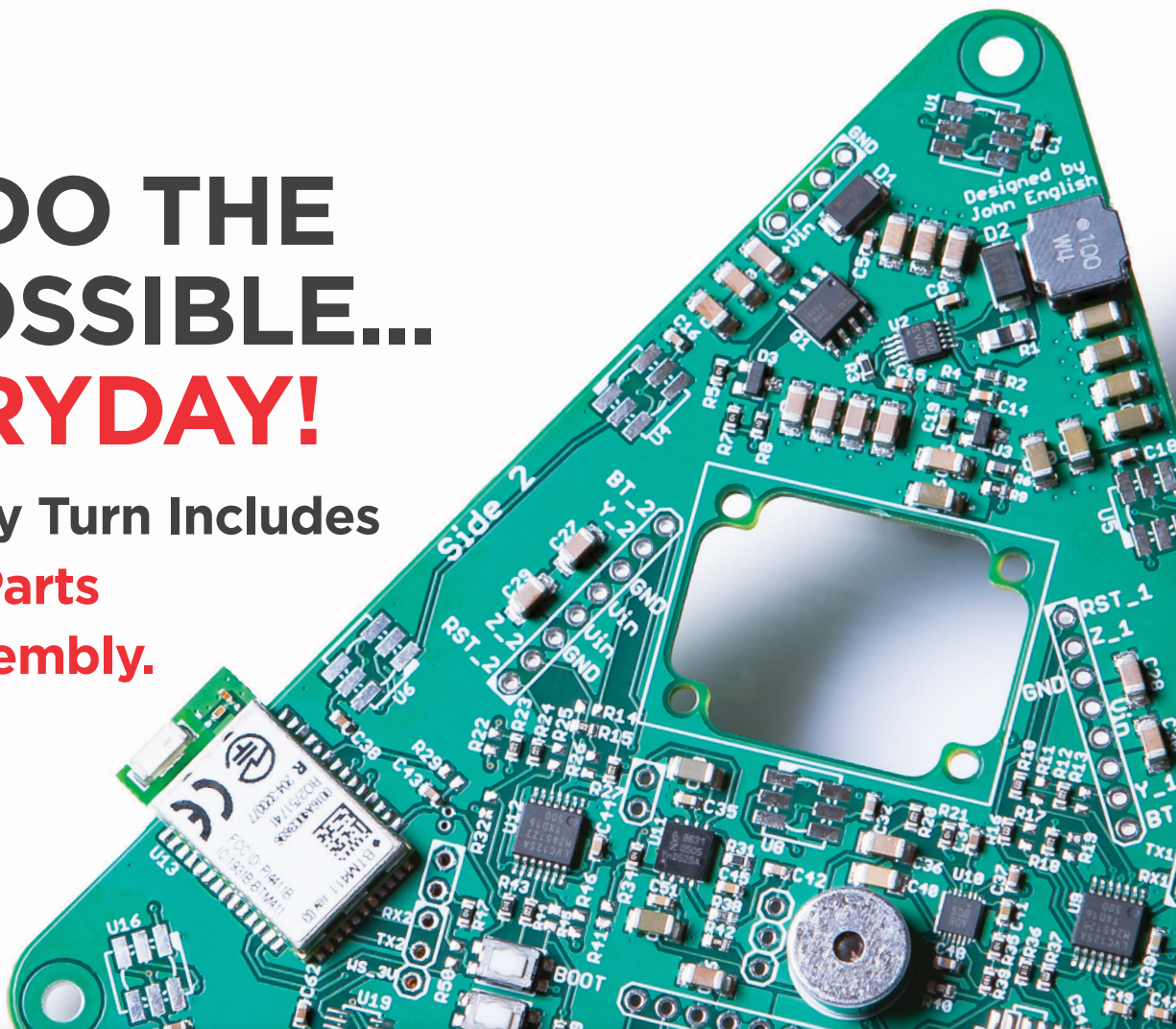
Digilent | www.digilentinc.com

Microsoft | www.microsoft.com

PJRC | www.pjrc.com

WE DO THE IMPOSSIBLE... EVERYDAY!

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



FREE LABOR

TRY SOMETHING DIFFERENT!

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

OUR ASSEMBLIES START AT \$250

DOWNLOAD YOUR OFFER CODE HERE: Circuitcellar.com/SlingShot

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

No more sacrificing quality for speed or price.

We are your **PCB ASSEMBLY SPECIALISTS!**

 **SlingShot**
ASSEMBLY



Find out why we're different at SlingShotAssembly.com/Different

Call for details: **720.778.2400** or Email: sales@sassembly.com

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

©2019 COPYRIGHT SLINGSHOT ASSEMBLY

From the Bench

Non-Evasive Current Sensor

Pondering Probes

Gone are the days when you could do most of your own maintenance on your car's engine. Today they're sophisticated electronic systems. But there are some things you can do with the right tools. In this article, Jeff talks about how using the timing light on his car engine introduced him to non-contact sensor technology. He discusses the types of probes available and how to use them to read the magnitude of alternating current (AC).

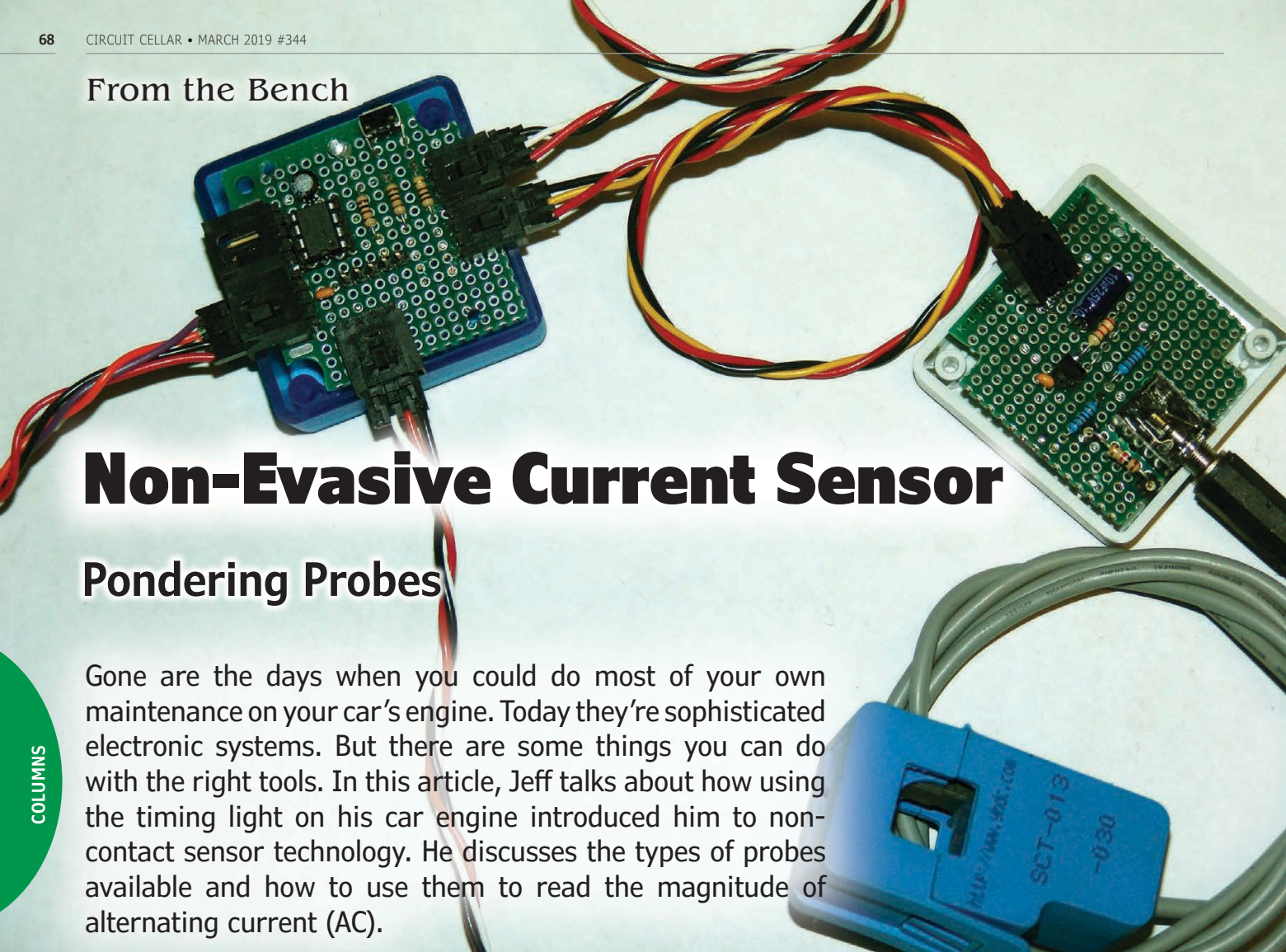
By
Jeff Bachiochi

No doubt you've looked under the hood of your automobile to find every conceivable piece of real estate crammed with mechanical and electrical components. Believe it or not, there was a time when you could see clear to the ground around the engine. People use to change their own oil, spark plugs, and thermostat, doing all the periodic maintenance themselves. Today we pass this all off to the local mechanic. Annually I would change the points, plugs, condenser and adjust the distributor using a timing light. The timing light was a strobe that was triggered by clamping a probe onto one spark plug. Whenever the plug fired, its high voltage pulse was picked up by the probe, which triggered the timing light flash. You point the flash at a timing mark on the engine and watch the groove in the engine shaft's pulley. With the distributor loosened, you rotate the distributor to adjust the point in the piston

cycle where the spark plug would fire to ignite the gas mixture (**Figure 1**).

Those of you who want to tune-up your own engine can stop reading this now. I don't claim to be a grease monkey. Mind you I use the term grease monkey as I would nerd—it's my professional description of someone who loves to work on automobiles. For me, the timing light is one of the tools of that trade that introduced me to non-contact sensor technology. The probe that clamped around the spark plug wire could be capacitive or inductive. This discussion will center around using inductive technology as a current probe.

The probe is an electrical device that surrounds an electrical conductor. This allows measurement of the current through the conductor without the need to make physical contact with it. This type of current probe comes in two varieties. First there's the closed probe—which requires the conductor to be disconnected, passed through the toroid, and



reconnected. Next is the clamp probe—which can be placed around the conductor without disconnection. Obviously, it is easier to use a clamp probe, but the closed probe is smaller and often less expensive. In this project, we will be using these probes to read the magnitude of alternating current (AC).

CURRENT PROBE TEARDOWN

The current probe is essentially a transformer that uses the conductor as a one-turn primary coil, as shown in **Figure 2**, unless you wrap the conductor so it goes through the core multiple times. The core is usually a ferrite toroid (one-piece, **Figure 3a**) as used in a closed probe, or a split-ferrite toroid, as used in a clamp probe (**Figure 3b**). Current moving through any conductor creates a magnetic field around the conductor. The magnetic field is happy to flow through the ferrite. Since the transformer secondary is also wound on the ferrite, the magnetic field induces a current in the secondary. This current passes through a load resistor, which develops a voltage across it in proportion to the secondary current and the magnetic field in the transformer core.

When dealing with voltage, the ratio of the primary to secondary windings of the perfect transformer directly determines the ratio of input to output voltage, with more secondary turns producing higher output voltages. While the output voltage increases, output current is reduced by the turns ratio.

The load resistor placed on the output of the current probe is called the “burden resistor.” If we design for a 1 V maximum output voltage on the burden resistor of

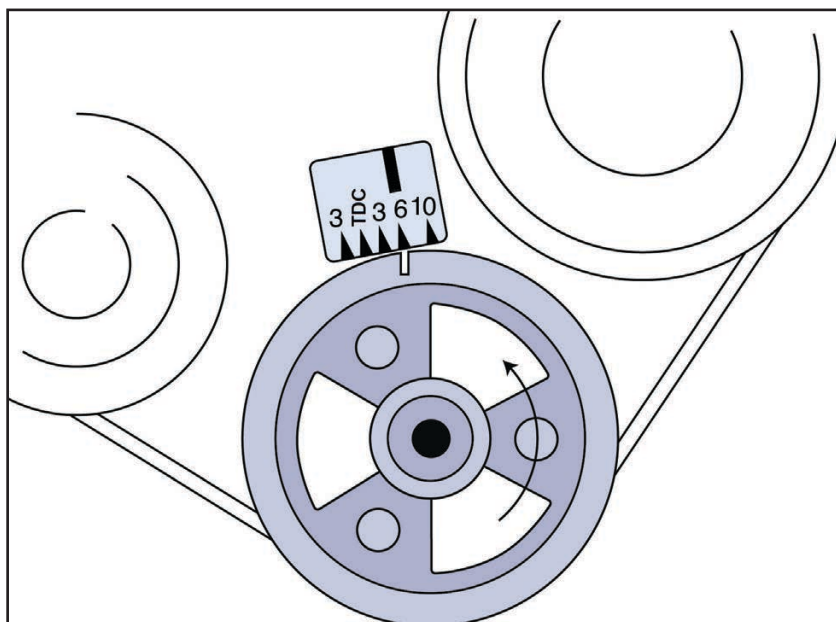


FIGURE 1

A timing light is used to adjust the point of cylinder firing by pointing its flash tube at the timing mark on an engine's block. A probe placed on the spark plug wire of cylinder 1 triggers the flash tube. By rotating the distributor while you watch the timing mark, you affect the point in the combustion cycle where the gasoline is ignited.

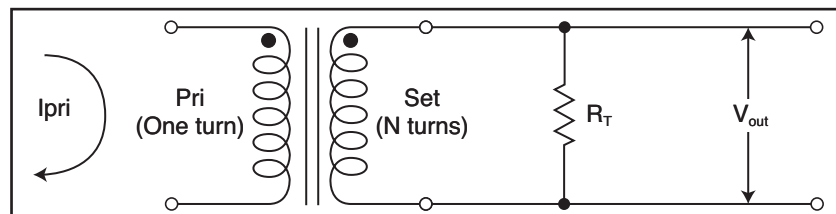


FIGURE 2

This simple schematic shows the split-core current probe, consisting of a 1-to-n turn winding transformer and output burden resistor. The one-turn primary produces a lower current in the multi-turn secondary. The burden resistor is selected to produce 1 V across it when the maximum design current is available.

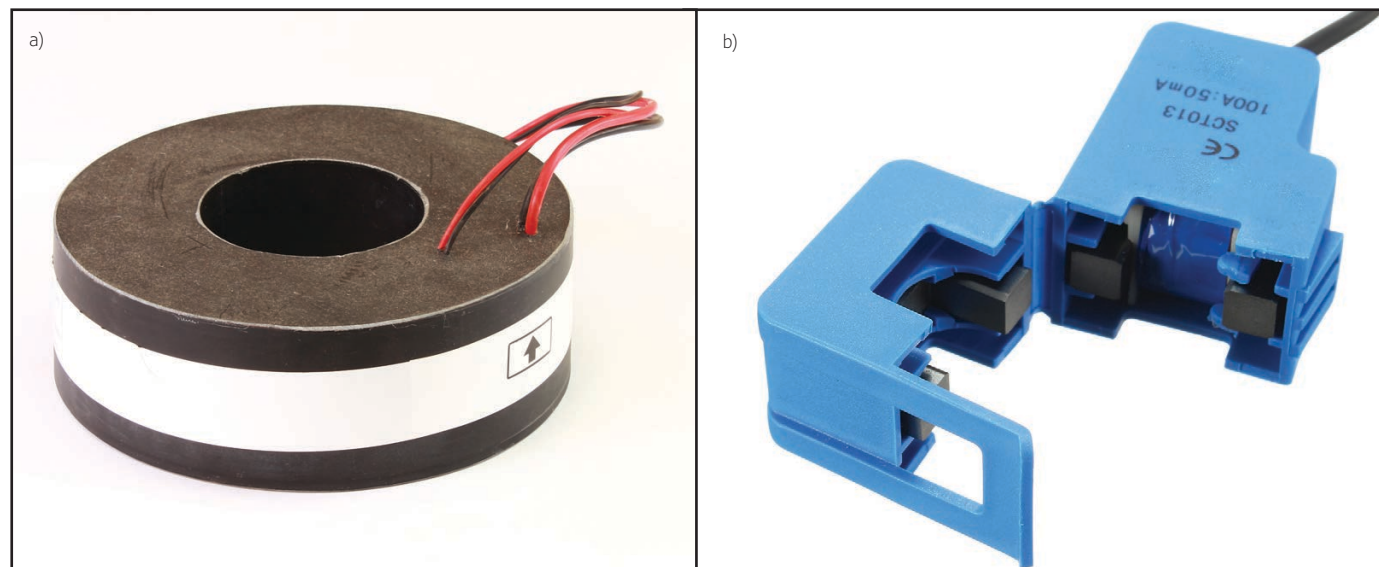


FIGURE 3

The current transformer normally has a solid core as in (a), the solid-core probe. To allow a probe to be attached to existing wiring without disconnection, a split-core probe can be used as in (b).

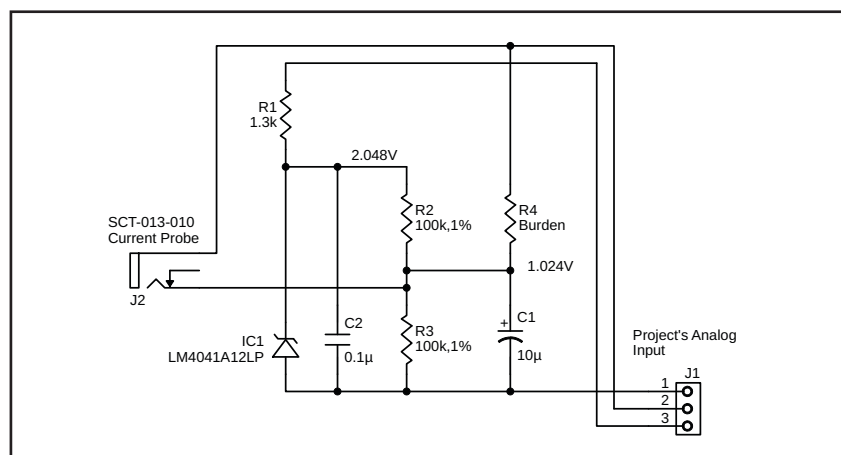


FIGURE 4

Because we want to measure an AC signal with a circuit that accepts only positive signals referenced to ground, this circuit is used to bias the measurement up by 1.024 V. The design of the probe is such that full-scale output is ± 1 V. The biased full-scale output will vary between 0.024 V and 2.024 V. When the probe has no output, the circuit output is only the bias voltage, 1.024 V.

100 Ω , we will have a current through this resistor of $1\text{V}/100\Omega = 10$ mA. If we want this maximum of 10 mA in the secondary to equal 1 A of current through the primary, then we will need a turns ratio of $1\text{A}/0.01\text{A} = 100$ or 1 primary turn to 100 secondary turns, 1:100. Both transformer core losses and saturation issues must be taken into consideration. Probe manufacturers have done all the design work for you and offer datasheets and application notes to make this all “a piece of cake.”

You can purchase low-cost probes that already contain the necessary transformers and burden resistors to ease your design. Many are designed for ± 1 V output for rated current. The SCT-013 is available without a burden resistor, and you can measure any current range up to 100 A. Use a burden resistor of 10 Ω , and at 50 mA you will measure ± 1 V for a full scale of 100 A. Use a 100 Ω resistor, and you get a range of 0-10 A. A 1,000 Ω resistor will give you 0-1 A. This line also has separate products available with burden resistors for a full scale ± 1 V output of 5, 10, 15, 20, 30, 50, 60 and 100 A.

When measuring AC current, the voltage presented will alternate in polarity. For most analog-to-digital converters (ADCs) that are an internal peripheral in a microcontroller (MCU) this is an issue, because they tend to be referenced to ground. To use these probes, you need to apply the probe's voltage to a bias voltage, which will raise it sufficiently to prevent negative voltages. This design will allow the probes to be used with a 5 V or 3 V supply. I've selected a Texas Instruments (TI) LM4040 Precision Micropower Shunt Voltage Reference to provide this bias. It's available in a variety of different reference voltages from 2.048 V to 10 V. Either the 2.048 V or 2.5 V reference could be used here—depending on the maximum current you expect and the references available to the ADC in your MCU.

The device is a precision Zener for which a series resistor sets the maximum current

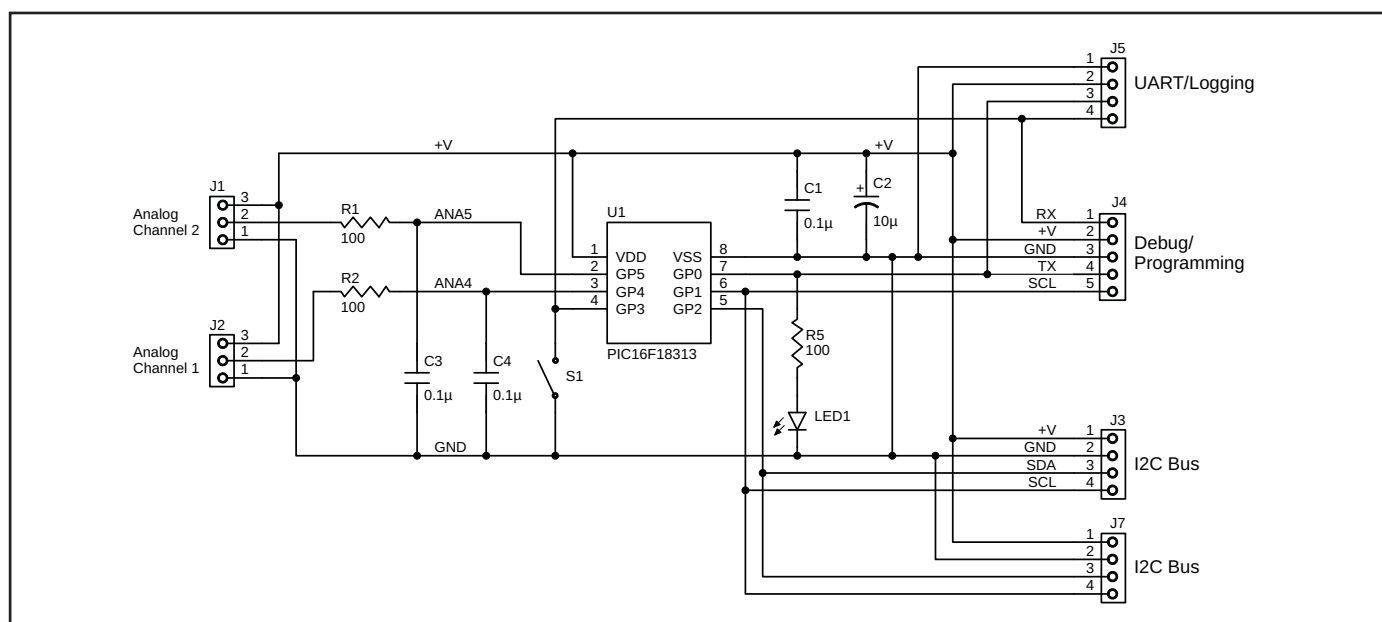


FIGURE 5

This project's schematic uses an 8-pin microcontroller to monitor two analog inputs and deliver real time current measurements from its UART to a PC, LCD or OpenLog data logger. It also acts as an I²C Slave, which gives access to its measurements to any I²C Master.

that can be handled by the Zener. In this case, 15 mA is suggested. To operate correctly, it must have a minimum of approximately 200 μ A of current. Since whatever you connect as a load across the Zener will share the current through the series resistor, you must make sure your load does not require more than the design current—15 mA minus the minimum regulation current—or else the reference cannot regulate properly.

The basic load across the Zener is a voltage divider that will provide a reference of one-half the Zener value. The lower leg of the divider has a capacitor across it to further prevent this reference from wavering under changing load conditions. **Figure 4** shows the schematic of this probe's preconditioning circuitry. The probe is connected across the upper leg of the voltage divider. It is therefore referenced to the bias voltage. With no signal from the probe, the voltage presented to the ADC is one-half the Zener reference. This is the bias that lets the probe's output rise above and fall below it by 1 V without going negative. Using a 2.048 V LM4040, a full-scale probe output will vary from 0.024 V, during the negative output of the probe (1.024 V to 1 V), to 2.024 V, during the positive output of the probe (1.024 V + 1 V).

PROBE MEASUREMENT

Because this probe is being used on 60 Hz AC, we know that any output is going to be continually changing at this rate. A single sample could get a measurement from anywhere in the cycle. We could look for zero-crossings by physical or measurement means, and then resample after a 4.1 ms (peak), or just sample during a 16.6 ms period (1 full cycle), looking for maximum and minimum values (peaks). I will use the second option here.

This project (**Figure 5**) uses a Microchip Technology PIC16F18313 MCU, whose ADC can be referenced to V_{CC} , an external input or one of the internally generated reference voltages. I want to use the same reference as the Zener in the pre-processing circuitry, 2.048 V. I am using two inputs of this 8-pin MCU's six I/O pins for analog voltages. This means I will be alternating inputs to the ADC. It is recommended that whenever a channel is selected (changed), you wait some acquisition time for the input's internal capacitor to charge properly. The acquisition time depends on the input impedance presented to the ADC, in our case approximately 2 μ s. We will want to take at least 100 samples in 16.6 ms, which is 160 μ s between samples. An acquisition time of 160 μ s is adequate. We need to see if we can

handle processing the measurements in that period.

Using RA2 as a temporary output, a toggle pin routine is placed in the ADC interrupt routine—at the entry and exit points—so we can see the execution of that routine. **Figure 6** shows the routine takes about 10 μ s. This happens every 250 μ s for a total of 25 ms for each channel. If we compress this, we get **Figure 7**, which shows the channel conversions happening at 1 second intervals. I chose 1 second for the ADC/display refresh rate.

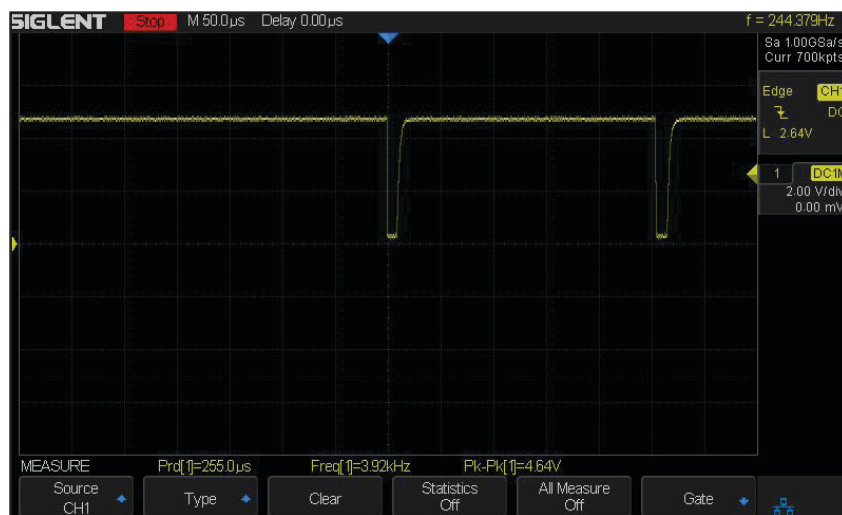


FIGURE 6

The pulse width seen here is approximately 10 μ s, which is the execution time for the ADC interrupt. A conversion has taken place, and the routine replaces the maximum or minimum values saved from previous conversions, if appropriate. New conversions are made every 250 μ s for more than 1 AC cycle time.



FIGURE 7

If Figure 6 is compressed, we see that the measurements for both analog inputs channels are completed in about 50 ms, and this happens once each second



FIGURE 8

After the analog measurements are complete, the display routine must calculate the peak-peak voltage (0-2 V), peak voltage (0-1 V), which also represents a peak current (0-1,000 mA). This scope pulse shows an execution time of 900 μ s for the calculations and the TX1Buffer being filled with display data.

While we are looking at execution timing, take note of **Figure 8**. I moved the toggle pin routine to measure all the time spent in the calculations done on the two channel measurements. This is about 900 μ s for both channels. With measurements taking 50 ms and the calculations for each measurement requiring 1 ms, we could increase the sampling rate more than 10 times and still have plenty of horsepower. I'm happy with 1 per second for this project, because in either of the uses for this project there is no need for faster sampling. The first application for this project is that of a monitor/logger. The second application will be as a slave sensor module. I'll discuss each of these later in the article.

MONITOR/LOGGER

The calculations for each measurement are handled by the main loop, which essentially waits for the 250 μ s timer to overflow 4,000 times. The ADC interrupt is responsible for collecting the maximum and

minimum voltage measurements for each channel over the 25 ms sampling period for each channel. Once per second, we have to calculate what these measurements mean. With each measurement on the load of our probe biased at 1.024 V, current through the probe's conductor will be reflected as an addition to or subtraction from this bias voltage.

Our maximum and minimum measurements are the most positive voltage above the bias and most negative voltage below the bias, respectively. Assuming we've chosen the probe correctly, this should never go above 2.024 V or below 0.024 V, and is the peak-to-peak voltage on the load resistor. We can find this by subtracting the minimum from the maximum measurements. Peak voltage will be one-half of this calculated value. The calculated peak value and the scale value are used to arrive at the actual probe current.

We know that each probe is designed to have a full-scale output of 1 V. However, each probe is designed for a different current range. The range is normally part of the probe's part number. In this case SCT-013-010 designates 10 A full scale. A scale value of 10 is used to transform the 0-1,000 mV calculated value to its proper probe value. We are using the UART in this MCU to send out formatted data to a PC or LCD for display in the format:

```
I2C Twin Analog Probes
I2C Address = 0x10
Scale = 0x0A
000.00 Amps 000.00 Amps
000.00 Amps 000.00 Amps
000.00 Amps 000.00 Amps
000.00 Amps 000.00 Amps
```

Note that after a sign-on message and a display of an I2C Address and Scale values, the currents for each channel are displayed at a rate of once per second. Not only can this be used to monitor the active currents, but also by connecting an OpenLog module, the data are automatically logged into a file on the attached SD card.

While this four-wire UART port (+V, Gnd, TX and RX) has the potential of bidirectional communication, this project does not make use of user input through the port. It could be used to set various function of the project such as Scale of the connected probe. I'll leave this up to the user, because the second application already has this.

SLAVE SENSOR MODULE

So far, we've used four of the six I/Os available in this 8-pin MCU. The remaining I/Os will be used for the slave I2C sensor module. In this application, the probe's data are made



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.

available to an I²C master. The I²C bus was initially designed for interfacing to other circuitry on the same PCB. This eliminated the need for the main processor to use separate I/Os to communicate with a potentially large number of devices. Devices such as LCDs, GPS and accelerometers still offer this interface. Many times, I've needed to measure currents in a project, and this offers a quick way to get measurements without having to deal with AD conversions.

I treated this project like a registered I²C device. In other words, it uses a pointer requesting a reply from a number of possible registers. The pointer is set by the first value, in a write to the device. Subsequent values are placed into the register pointed to—the pointer automatically is advanced for each new value. It is important to note that you should set the pointer with a write before reading the device, as the pointer is used for both reading from and writing to specific registers. If you try reading more registers than there are, the device will normally wrap the pointer back to the beginning. On a Write this could be disastrous, so additional writes are discarded.

In this device, there are six registers of interest, and these are located using a pointer value of 0-5, as follows:

Pointer	Register Description
0	Channel 1 MSB
1	Channel 1 LSB
2	Channel 2 MSB
3	Channel 2 LSB
4	Scale
5	I ² C Address

The Scale and I²C Address will default to Scale=1 and an I²C Address=0x10, if you hold down the push button (RX input) while powering up the board. Otherwise, any values written to those locations will be used by the application and saved to EEPROM memory for subsequent power-ups.

Each channel will always be a value of 0-1,000, representing millivolts. The scale factor allows one to put the channel measurements into perspective. I use Arduino for many projects. It has support for I²C and is easy to program. **Listing 1** is a sketch to allow an Arduino to read and write to an I²C slave device.

The output from this sketch is as follows:

```
I2C Dual Current
Amps: 0.00 0.00 Scale: 10
Amps: 0.00 0.00 Scale: 10
Amps: 0.00 0.00 Scale: 10
Amps: 0.00 0.00 Scale: 10
Amps: 0.00 0.00 Scale: 10
```

```
const String SignOn = "I2C Dual Current";
#include <Wire.h>
int Index = 0;
unsigned char Data[6];
int I2C1 = 0x10;
int Current1 = 0;
int Current2 = 0;
int Scale = 0;
int I2CAddress = 0;

void setup()
{
  // I2C Start
  Wire.begin();           // join i2c bus
  // I2C End
  Serial.begin(9600);     // start serial for output
  Serial.println(SignOn); // print message
}

void loop()
{
  // Get the Currents
  SampleCurrent();
  DisplayCurrent();
  delay(1000);
}

void DisplayCurrent()
{
  // Display the Currents
  Serial.print("Amps: "); // print the character
  Serial.print(Current1*Scale/1000.0); // print the character
  Serial.print(" ");      // print the character
  Serial.print(Current2*Scale/1000.0); // print the character
  Serial.print(" Scale: "); // print the character
  Serial.println(Scale);    // print the character
}

int SampleCurrent()
{
  ReadCurrent(I2C1);
  Current1=Data[0]*256;
  Current1=Current1+Data[1];
  Current2=Data[2]*256;
  Current2=Current2+Data[3];
  Scale=Data[4];
  I2CAddress=Data[5];
}

int ReadCurrent(int Address)
{
  Wire.beginTransaction(Address); // transmit to device
  Wire.write(0x00);               // sends pointer value byte
  Wire.endTransmission();         // stop transmitting
  Wire.requestFrom(Address, 6);   // request 6 bytes
  from slave device
  int result = 0;
  while (Wire.available())
  {
    Data[result] = Wire.read(); // receive a byte as character
    result++;
  }
  return result;
}
```

LISTING 1

Shown here is an Arduino sketch to allow an Arduino to read and write to an I²C slave device.

Device Pin	Default Function	Final Function
1	V _{cc}	V _{cc}
2	AN5	AN5
3	AN4	AN4
4	MCLR	RX
5	SDA	SDA
6	ICDCLK	SCL
7	ICDDAT	TX
8	V _{ss}	V _{ss}

TABLE 1

Once all routines were finished, the I/O was reassigned and the debugging feature was disabled to make use of its default I/O. This table shows the final assignment of the pins.

While the scale has already been applied to the Amps value, it is displayed just for confirmation that you are using the correct probes. Also note that with no probe attached (no burden resistor), the inputs will float, giving a non-true value.

8-PIN CHALLENGES

With a low pin count MCU, there are a number of barriers you need to understand before assigning all the I/O pins to application tasks. First off, to use the debug interface requires two predefined I/O pins, which means while debugging an application these are not available to you. In the case of the PIC16F18313, they are pins 6 and 7. By default, two analog inputs, RX (UART receive) and SCL (I²C clock) share these pins. However, this device has the PPS (Programmable Pin Select) feature, which allows many of the peripherals to be assigned to alternate locations. Without this feature you couldn't possibly use this device to the extent I have in this project. I used this feature to temporarily reassign specific peripheral I/O to unused pins RA2, RA4 and RA5. This allowed me to write and debug code for each peripheral, one at a time. First the UART code, so I would have a way of displaying information. Second, the analog conversion code, so I would have something to display. Finally, the I²C code could be written and debugged.

If you look at the source code, you'll see multiple routines that were only written to help debug the I²C code—by displaying received bytes in real time—and also toggle an output bit routine to allow a scope to grab execution timing. Once all routines were finished, the I/O was reassigned and the debugging feature was disabled to make use of its default I/O. The final assignment is shown in **Table 1**.


Like many of my projects, using the internal oscillator has the advantage of gaining two I/O pins. Using an external crystal requires an input and an output pin. These internal oscillators are being improved continuously, and while their accuracy for software real-time clocks (RTCs) may be barely acceptable, for most applications they are entirely adequate $\pm 2\%$ (with the extremes, over V_{ss} and temperature, falling off to $\pm 5\%$).

I had a stumbling block early on in getting RA5 to output any value. Most of the PIC MCUs I've used in the past require a single configuration of the oscillator being used. This device requires two—one to enable the internal oscillator, and one to disable the external oscillator. In failing to disable the external oscillator, it remained defined as an external crystal input and would not allow a redefinition of its TRIS value (direction control of the pin).

CONCLUSION

At a cost of less than \$10, split-core current probes are a great way to measure AC currents of various capacities. The same circuitry can be used independent of the current range necessary, since each probe is designed to present the same output. Simple scaling is done on the out value to achieve the correct current range.

The versatility of low-pin-count devices continues to grow. When first introduced on the market, the laughter from most people was inescapable. Even though these were smaller than most logic chips at the time, most lost sight of the power inside. Granted, in the beginning few peripherals were available, but as engineers showed their skillfulness manufacturers responded with more peripheral power.

Even though the hoops required to use multiple peripherals in this tiny device made me feel more like a circus contortionist than a programmer, the end result is very satisfying. 

Additional materials from the author are available at:

www.circuitcellar.com/article-materials

RESOURCES

DeChang Electric Company | www.yhdc.com

Microchip Technology | www.microchip.com

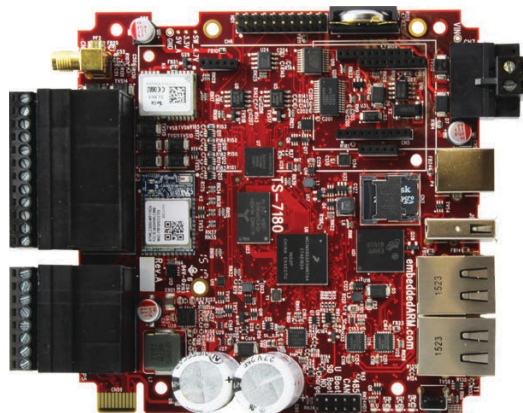
Texas Instruments | www.ti.com

PRODUCT NEWS

i.MX6 UL-Based SBC Boasts DAQ and Wireless Features

Technologic Systems has announced an engineering sampling program for a wireless- and data acquisition focused SBC based on NXP's low-power i.MX6 UL SoC. The -40°C to 85°C tolerant TS-7180 is designed for industrial applications such as industrial control automation and remote monitoring management, including unmanned control room, industrial automation, automatic asset management and asset tracking.

The TS-7180 has a 122 mm x 112 mm footprint. Like its 119 mm x 94 mm TS-7553-V2 SBC and sandwich-style, 75 mm x 55 mm TS-4100, it features the low power Cortex-A7 based i.MX6 UL, enabling the board to run at a typical 0.91 W. Like the TS-4100, the new SBC includes an FPGA. The FPGA enables the optional, 3x 16-bit wide



quadrature counters, which are accessible via I²C registers. The quadrature counters and tachometers are part of a DAQ subsystem with screw terminal interfaces that is not available on its other i.MX6 UL boards. The board has a cellular modem socket that supports either MultiTech or NimbLink wireless modules. You also get Wi-Fi/BT, optional GPS, and a socket for Digi's XBee modules, which include modems for RF, 802.15.4, DigiMesh and more.

The TS-7180 is available in an engineering sampling program for \$315 with 512 MB RAM or \$381 model with 1 GB RAM, GPS and IMU. 100-unit pricing is \$254 and \$320.

Technologic Systems
www.embeddedarm.com

Low-Power PMIC Enables High Sensitivity Optical Measurements

Maxim Integrated has introduced its latest tiny, highly integrated power-management IC (PMIC). The ultra-low-power MAX20345 integrates a lithium charger and debuts a unique architecture that optimizes the sensitivity of optical measurements for wearable fitness and health applications. In wearables, optical-sensing accuracy is impacted by a variety of biological factors unique to the user. Designers have been striving to increase the sensitivity of optical systems, in particular the signal-to-noise ratio (SNR), to cover a broader spectrum of use cases.

According to Maxim, the MAX20345 features a first-of-its-kind buck-boost regulator based on an innovative architecture that's optimized for highly accurate heart-rate, blood-oxygen (SpO₂) and other optical measurements. The regulator delivers the desired low-quiescent current performance without the drawbacks that degrade SNR and, as a result, can increase

performance by up to 7dB. The MAX20345 is available for \$4.45 (1,000-up, FOB USA) in a 56-bump, 0.4 mm pitch, 3.37 mm x 3.05 mm wafer-level package (WLP).

Maxim Integrated | www.maximintegrated.com

MAX20345: ultra-low-power PMIC for wearables and IoT



504-W Rated AC-DC Power Modules are Conduction Cooled

TDK has announced its conduction cooled TDK-Lambda PFH500F-28 AC-DC power modules. The power supplies are rated at 28 V, 504 W, feature a compact 4" x 2.4" footprint and have optional Read/Write programming and communication through a PMBus interface. Accepting an 85 to 265 VAC



input, the modules deliver 28V at 18A and can be adjusted from 22.4 to 33.6 V using the trim pin or PMBus interface. Baseplate cooling allows operation at temperatures ranging from -40°C to +100°C.

Features and options include a 12 V standby voltage with 200 mA (or 2.4 W) capability, remote on/off, pre-biasing start-up, droop mode current share, a DC Good signal, various protections (OVP, UVP, OCP, OTP) and a PMBus interface. The interface can be used to program (read-write) the output voltage and fault management functions and monitor the unit's operating status. The series has been certified to the IEC/UL/CSA/EN 60950-1 safety standards (62368-1 pending) and carries the CE mark for the Low Voltage and RoHS Directives.

TDK Lambda | www.us.tdk-lambda.com

PRODUCT NEWS

Secure Cellular Router Serves Industrial and Transportation Needs

Digi International has announced the Digi WR54, a rugged, secure, high-performance wireless router for complex mobile and industrial environments. With dual cellular interfaces, Digi WR54 provides immediate carrier failover for near-constant uptime and continuous connectivity, especially as vehicles move throughout a city or for locations with marginal cellular coverage. Together with a hardened MIL spec-certified design and built-in Digi TrustFence security framework, this LTE-Advanced router is designed specifically to meet the connectivity challenges inherent in multi-location, on-the-move conditions, from rail and public transit to trucking fleets and emergency vehicle applications.

Digi WR54 provides an LTE-Advanced cellular module built on a platform that supports higher speeds to optimize bandwidth today while also being positioned for the future as network capabilities improve. Aside from the LTE-Advanced cellular module, the system provides four Gbit Ethernet ports for wired systems and the latest 802.11 ac Wi-Fi which combine to support the needs of any user.

Digi International | www.digi.com



Mini-ITX SBC Sports AMD Ryzen APU SoC

WIN Enterprises has announced the MB-73480 which supports the AMD Ryzen Embedded V1000 processor family. The AMD processors combine the performance of the AMD "Zen" CPU and "Vega" GPU architectures in an integrated SoC solution. In addition, the AMD Ryzen processors deliver discrete-GPU caliber graphics and multimedia processing. Compute performance clocks to 3.61 TFLOPS with thermal design power (TDP) as low as 12 W and as high as 54 W.

The advanced AMD Ryzen CPUs and its other features make the MB-73480 well suited for applications requiring high performance graphics and advanced processing power. Applications include: gaming machines, digital signage, medical imaging, industrial control/automation, thin client, office automation and communication infrastructure. WIN Enterprises will customize the PL-81280 based on a customer's more specific market needs.

WIN Enterprises | www.win-ent.com



Tool Extension Enables Neural Networking on STM32 MCUs

STMicroelectronics has extended its STM32CubeMX ecosystem by adding advanced Artificial Intelligence (AI) features. AI uses trained artificial neural networks to classify data signals from motion and vibration sensors, environmental sensors, microphones and image sensors, more quickly and

efficiently than conventional handcrafted signal processing. With STM32Cube.AI, developers can now convert pre-trained neural networks into C-code that calls functions in optimized libraries that can run on STM32 MCUs.

STM32Cube.AI comes together with ready-to-use software function packs that include example code for human activity recognition and audio scene classification. These code examples are immediately usable with the ST SensorTile reference board and the ST BLE Sensor mobile app. The STM32Cube.AI extension pack can be downloaded inside ST's STM32CubeMX MCU configuration and software code-generation ecosystem. Today, the tool supports Caffe, Keras (with TensorFlow backend), Lasagne, ConvnetJS frameworks and IDEs including those from Keil, IAR and System Workbench.



STMicroelectronics | www.st.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.



Myth: Embedded vision can be complex.
Fact: It doesn't have to be.

Speed past this and other myths
as Avnet and Xilinx move you from
complexity to clarity.



Watch now at circuitcellar.com/avnet



Power of a PIC® MCU in your USB port...

Rapid USB Prototyping Stick and Single-Chip C Compiler

\$65



Capabilities and Features

Pushbutton, Two LEDs & I/O Cable

Text I/O from PIC to PC Screen

Quick Way to Log/Graph Analog Data

Easy Graphing of PIC® MCU Data

www.ccsinfo.com/CC319

sales@ccsinfo.com – 262-522-6500 x 35

Microchip Premier Partner
PIC® MCU is a registered trademark
of Microchip Technology Inc.



Technologic Systems

Single Board Computer

TS-7250-V2
1GHz ARM Computer with
Customizable FPGA-Driven
PC/104 Connector
and Several Interfaces
at Industrial Temp

www.embeddedARM.com

ALL ELECTRONICS

**Surplus & New Parts & Supplies
Since 1967**

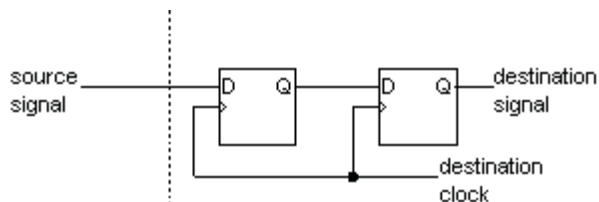
LEDs · CONNECTORS · RELAYS
SOLENOIDS · FANS · ENCLOSURES
MOTORS · WHEELS · MAGNETS
PC BOARDS · POWER SUPPLIES
SWITCHES · LIGHTS · BATTERIES
and many more items...

We have what you need for your next project.

Discount Prices
Fast Shipping

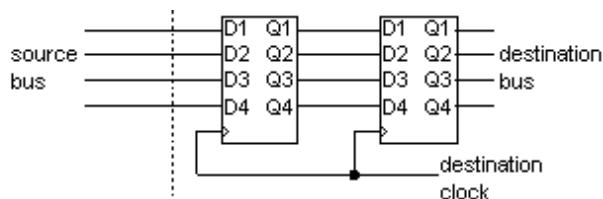
www.allelectronics.com

Problem 1— It's standard practice to use a pair of flip-flops to resynchronize a signal that passes from one synchronous clock domain to another, in order to minimize problems with metastability.



Source Clock Domain Destination Clock Domain

Sometimes, it's necessary to pass multiple-bit values (for example, a bus) between two clock domains, for example, when processing 8-bit pixel values in a digital video stream inside an FPGA. Why is the following obvious approach a bad idea?



Source Clock Domain Destination Clock Domain

TEST YOUR EQ

Contributed by David Tweed

Problem 2— Most FPGA vendors offer a true dual-clock FIFO primitive that can be used to solve this problem in the general case. However, in cases where the data changes relatively slowly with respect to the destination clock, there's a much simpler approach. What is it?

Problem 3— Two networks that allow message collisions with detection and recovery are Ethernet and Controller-Area Network (CAN). However, the methods used are quite different, and CAN offers one guarantee that Ethernet cannot. What is it?

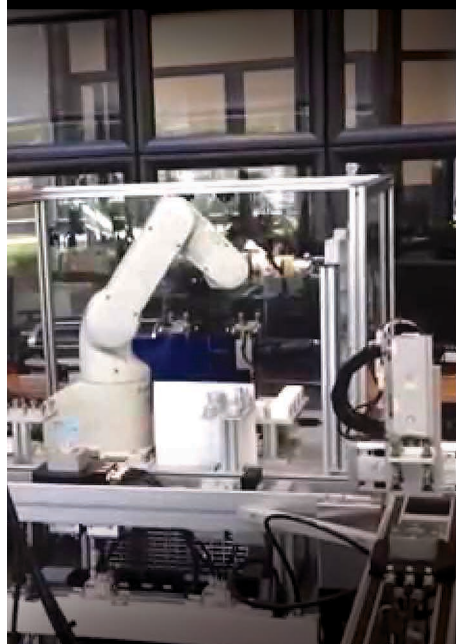
Problem 4— On the other hand, Ethernet's method has what advantage?

For more information:

circuitcellar.com/category/test-your-eq/

You can take it almost anywhere.

Where will it take you?



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

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

Visit cc-webshop.com to purchase

The Future of IoT Connectivity

Cellular Technology's Role in IoT Market Growth

When it comes to the growth of the Internet of Things (IoT) market, there are many astounding predictions. Some estimate it to reach \$1 trillion in revenue within the next decade. A recent study by Bain & Company—the Boston-based global management consultancy firm—suggests the combined markets of IoT will surge to about \$520 billion in 2021, more than double the \$236 billion spent in 2017.

But are such projections founded in reality? Although the IoT and M2M (machine-to-machine) industry has continued to evolve and grow significantly over the last few years, it has yet to fully realize the surging “hockey-stick growth” once enthusiastically forecasted by tech observers.

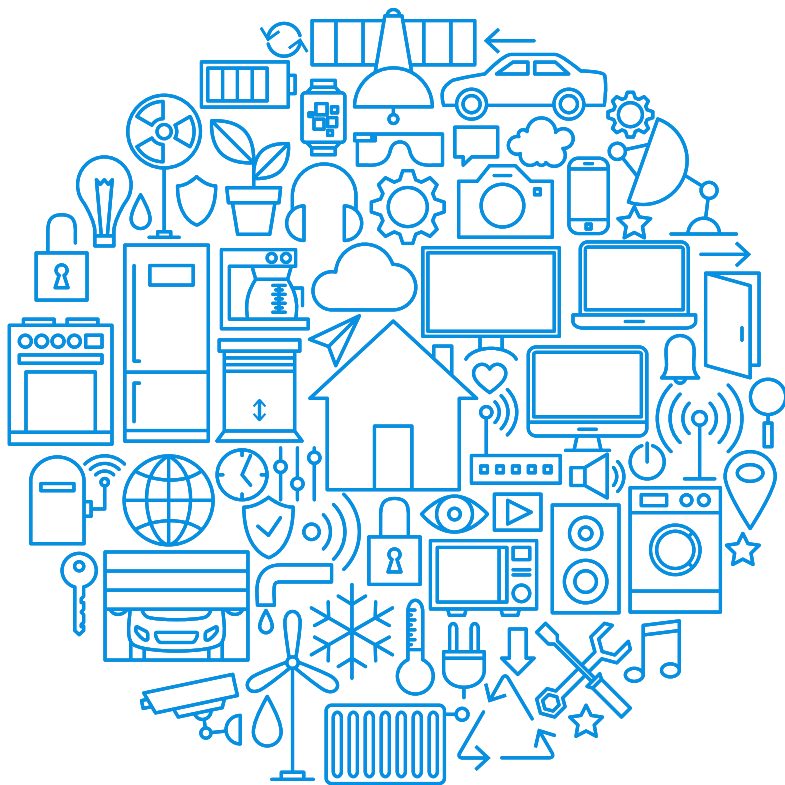
However, once you consider some of the technologies and solutions that are being deployed—as well as the problems that are solved through various types of connected technology—this unwavering enthusiasm for the IoT market remains justified. What is truly amazing is that we have only scratched the surface of IoT's potential. The IoT ecosystem expands across so many verticals in both industrial and consumer markets and has so many parts, including hardware devices, software platforms and connectivity. While the IoT market is still segmented, mergers and acquisitions continue to create more and more “one-stop-shops” that will help fuel quicker growth and adoption.

Another factor that is fueling the growth (and enthusiasm) for the IoT market is cellular connectivity. As the primary backbone for most IoT and M2M solutions, cellular connectivity offers a unique range of data-throughput capabilities—making it increasingly affordable and viable. As a result, cellular carriers have embraced IoT's vision and are making considerable strides in their own IoT solution offerings and support. The advent of 5G technology and new IoT-dedicated LTE networks (such as LTE Cat-M and NB-IoT) will be the next catalyst for rapid IoT growth. These advances will work to support data-intensive applications using 5G, while LTE Cat-M and NB-IoT will help support those IoT solutions with low data requirements and lower power restrictions.

Before rolling out to mobile handsets and devices on lower, mid-band frequencies, 5G will deploy primarily as a fixed wireless solution using millimeter wave frequencies. 5G is expected to be capable of providing peak data speeds up to 20 Gbit/s, with an expected average user speed of 1 Gbit/s. Those IoT applications



By
Broc Jenkins,
Wilson Electronics



which promise to benefit most from the increase to 5G speeds are cellular as primary and failover solutions, as well as connected and autonomous vehicles.

5G AND 4G LTE

Despite the eager anticipation of the arrival of 5G, dependence on 4G LTE networks should continue into the foreseeable future. With widespread 5G availability still years away, 4G LTE remains a viable cellular-signal solution that carriers intend to stay the course with into the next decade.

After all, not every IoT use case requires the blazing speeds promised by 5G frequencies or even those of the current 4G LTE networks. Many solutions such as remote monitoring, sensors and asset tracking may only require (and transmit) very little data, thus demanding a very low power consumption solution. This is where LTE cat-M and NM-IoT come into play. These new IoT-focused networks, as well as the hardware and chipsets that support them,

will allow for a more affordable and easily-scaled solution.


Despite all the advancements in cellular technology in terms of speeds and feeds, there is still one challenge that must be considered and addressed—signal strength and quality. A cellular-connected IoT solution is only as effective and reliable as its signal. Cellular networks are transmitted over radio frequencies (RF) which remain susceptible to a variety of things that may affect the signal strength, quality and ability for a connected device to communicate reliably. It is not only the distance from a tower that dictates how much signal is available, but other variables such as terrain, foliage, weather and man-made obstructions which collectively block or weaken transmitted signal.

The frequency at which a network is transmitting will also have an effect, as lower frequencies in the cellular spectrum (600-900 MHz) travel farther and have much better penetration compared to higher frequencies (1,900-2,500 MHz). 5G will be utilizing millimeter wave frequencies in the 28 GHz range, which will have even less ability to penetrate building materials.

ANTENNAS AND AMPLIFIERS

Solving these signal challenges can be as easy as using high-performance external antennas connected to the cellular device. These antennas not only add more power and receive sensitivity but can also be connected to a cable and placed in better signal areas—such as outside of a building or up on a pole—to bypass many signal blocking factors.

Bi-directional amplifiers are also viable solutions for both fixed and mobile applications, servicing a single cellular device or multiple devices simultaneously. However, current amplifier options are limited to current 4G LTE cellular network frequencies and legacy voice and data networks. As new frequencies are implemented for 5G, amplifier and antenna hardware will become available to support them.

Undoubtedly, the future being shaped right now is one of connectivity. It will be exciting to take part in the technological advancements that will cause the IoT market to live up to its most lofty expectations. 

Broc Jenkins is a national sales manager at Wilson Electronics. His primary responsibilities involve driving the sales and strategy of Wilson Electronics' CEDIA Channel, as well as overseeing the sales and business development of the WilsonPro M2M product line. Since joining Wilson Electronics in 2009, Broc has held positions in technical support, product line management and key account sales.

RESOURCES

Wilson Electronics | www.wilsonelectronics.com

STRONG FOUNDATION

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

Technology:

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

Materials:

Fr4
Metal Core
Isola
Rogers
Polyimide - Flex
Magtron

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

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

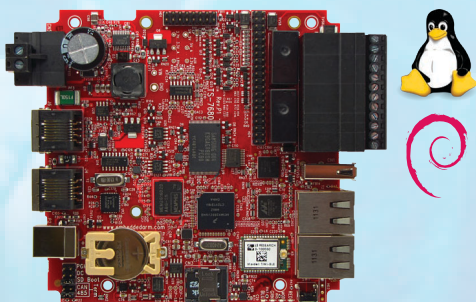


www.PCB4u.com sales@PCB4u.com

SAM & ITAR Registered UL E333047 ISO 9001 - 2008

BUILT FOR EXTREMES

Single Board Computer



TS-7680

Low Power Industrial
Single Board Computer with
WiFi and Bluetooth

\$159
Qty 100

From the the Canadian tundra to the deserts of Africa, no terrain is too demanding for our boards. Deployed in fleet management, pipeline monitoring, and industrial controls, our single board computers are working in some of the most demanding places on Earth.

The TS-7680 is designed to provide extreme performance for applications which demand high reliability, fast boot-up/startup, and connectivity at low cost and low power. Powered by a 454 MHz ARM CPU the TS-7680 offers a great balance between industrial features and high end capabilities.

