Electronic Design®

THE AUTHORITY ON
EMERGING TECHNOLOGIES
FOR DESIGN SOLUTIONS

NOVEMBER/DECEMBER 2021
electronicdesign.com

2021 SALARY SURVEY
— THE — RESULTS ARE IN

p8

$10.00

IN THIS ISSUE

FEATURES

COVER PHOTO CREDITS:
33123442 © Dragonimages | Dreamstime.com
943331302 Foryou13 | iStock / Getty Images Plus

COLUMNS & DEPARTMENTS

(ElectronicDesign)
(@ElectronicDesgn)

EDITORIAL MISSION:
To provide the most current, accurate, and in-depth technical coverage of the key emerging technologies that engineers need to design tomorrow's products today.

ELECTRONIC DESIGN (ISSN 0013-4872) is published bi-monthly in Jan/Feb, Mar/Apr, May/June, July/Aug, Sept/Oct and Nov/Dec by Endeavor Business Media, 1233 Janesville Ave., Fort Atkinson, WI 53538. Paid rates for a one-year subscription are as follows: $60 U.S., $90 Canada, $120 International. Periodicals postage paid at Fort Atkinson, WI and additional mailing offices. Printed in U.S.A. Title registered in U.S. Patent Office. For subscriber services or to order single copies, write to Electronic Design, PO Box 3257, Northbrook, IL 60065-3257. POSTMASTER: Send change of address to Electronic Design, PO Box 3257, Northbrook, IL 60065-3257. Canadian GST #R126431964.

ElectronicDesign. **NOVEMBER/DECEMBER 2021**

EDITORIAL

SENIOR CONTENT DIRECTOR: **BILL WONG** bwong@endeavorb2b.com

SENIOR EDITOR: **DAVID MALINIAK** dmaliniak@endeavorb2b.com

MANAGING EDITOR: **ROGER ENGELKE** rengelke@endeavorb2b.com

SENIOR STAFF WRITER: **JAMES MORRA** jmorra@endeavorb2b.com

CONTRIBUTING EDITOR: **LOUIS E. FRENZEL** lfrenzel@endeavorb2b.com

ART DEPARTMENT

GROUP DESIGN DIRECTOR: **ANTHONY VITOLO** tvitolo@endeavorb2b.com

ART DIRECTOR: **JOCELYN HARTZOG** jhartzog@endeavorb2b.com

PRODUCTION

GROUP PRODUCTION MANAGER: **GREG ARAUJO** garaujo@endeavorb2b.com

PRODUCTION MANAGER: **DEANNA O'BYRNE** dobyrne@endeavorb2b.com

AUDIENCE MARKETING

USER MARKETING MANAGER: **DEBBIE BRADY** dmbrady@endeavorb2b.com

FREE SUBSCRIPTION / STATUS OF SUBSCRIPTION / ADDRESS CHANGE / MISSING BACK ISSUES

OMEDA **T** | 847.513.6022 **TOLL FREE** | 866.505.7173 **FAX** | 847.291.4816 electronicdesign@omeda.com

SALES & MARKETING

REGIONAL SALES REPRESENTATIVES:

AZ, NM, TX: **GREGORY MONTGOMERY** gmontgomery@endeavorb2b.com

AL, AR, SOUTHERN CA, CO, FL, GA, HI, IA, ID, IL, IN, KS, KY, LA, MI, MN, MO, MS, MT, NC, ND, NE, OH, OK, SC, SD, TN, UT, VA, WI, WV, WY, CENTRAL CANADA: **JAMIE ALLEN** jallen@endeavorb2b.com

CT, DE, MA, MD, ME, NH, NJ, NY, PA, RI, VT, EASTERN CANADA: **ELIZABETH ELDRIDGE** eeldridge@endeavorb2b.com

AK, NORTHERN CA, NV, OR, WA, WESTERN CANADA: **STUART BOWEN** sbowen@endeavorb2b.com

INTERNATIONAL SALES:

GERMANY, AUSTRIA, SWITZERLAND: **CHRISTIAN HOELSCHER** christian.hoelscher@highcliffemedia.com

BELGIUM, NETHERLANDS, LUXEMBURG, UNITED KINGDOM, SCANDINAVIA, FRANCE, SPAIN, PORTUGAL:
LARA PHELPS lara.phelps@highcliffemedia.com

ITALY: **DIEGO CASIRAGHI** diego@casiraghi-adv.com

PAN-ASIA: **HELEN LAI** helen@twoway-com.com

PAN-ASIA: **CHARLES LIU** liu@twoway-com.com

ARTICLE REPRINTS: reprints@endeavorb2b.com

LIST RENTALS/ SMARTREACH CLIENT SERVICES MANAGER: **MARY RALICKI** mralicki@endeavorb2b.com

DIGITAL

SENIOR DIGITAL INNOVATION & STRATEGY DIRECTOR: **RYAN MALEC** rmalec@endeavorb2b.com

DESIGN ENGINEERING & SOURCING GROUP

EVP, DESIGN & ENGINEERING GROUP: **TRACY SMITH** tsmith@endeavorb2b.com

GROUP CONTENT DIRECTOR: **MICHELLE KOPIER** mkopier@endeavorb2b.com

VICE PRESIDENT OF MARKETING SOLUTIONS: **JACQUIE NIEMIEC** jniemiec@endeavorb2b.com

ENDEAVOR BUSINESS MEDIA, LLC

331 54th Ave N., Nashville, TN 37209 USA | www.endeavorbusinessmedia.com

CEO: **CHRIS FERRELL**

CRO/CMO: **JUNE GRIFFIN**

CFO: **WILLIAM NURTHEN**

COO: **PATRICK RAINS**

CHIEF ADMINISTRATIVE AND LEGAL OFFICER: **TRACY KANE**

EVP SPECIAL PROJECTS: **KRISTINE RUSSELL**

ENDEAVOR BUSINESS MEDIA.

Electronic Design | Machine Design | Microwaves & RF | Hydraulics & Pneumatics | Source ESB | Source Today | Evaluation Engineering

Editorial

WILLIAM WONG | Senior Content Director
bwong@endeavorb2b.com

Continual Change Can Be CHALLENGING

THIS YEAR has been like no other, but I can say the same for any year. Nonetheless, with COVID-19 still raging throughout the world, there's no doubt the radical changes it's induced are just the tip of the iceberg. The disruption has been across the board, from changing supply chains to how we interact with each other.

Not all changes were good, with the tremendous number of deaths at the top of the list. Even more people are affected by the long-term impact contracting the disease. Among the positive changes are an increase in telemedicine to reduce travel, which helps the environment, and better digital cameras and internet bandwidth have made video conferencing ubiquitous. Augmented reality (AR) and virtual reality (VR) continue to grow in performance and importance, too.

NVIDIA's Omniverse Avatar allows for the creation of lifelike avatars that can be controlled by AI-based applications to interact with people.

We now have Meta, formerly Facebook, with its Metaverse and NVIDIA and their Omniverse that push AR/VR and digital twins to the max. NVIDIA's Omniverse Avatar is looking to provide 3D, AI-managed avatars that can interact with people in real-time *(see figure)*.

The recognition of these and other changes shows up in the results of our annual engineer salary survey. This includes the desire for many to continue working remotely, which was often forced upon us by COVID-19. And the growth of virtual trade shows has been significant even as in-person events start to reappear. A lot of work still needs to be done to make these virtual trade shows better, but they will not be going away. Likewise, hybrid shows may become the norm as vaccinations make in-person events practical again.

The engineering salary survey also included questions like, "What keeps you up at night?" The item at the top of the list was component availability. That's not too surprising given the shortages and supply-chain disruption in our industry that have made headlines in the regular news. Car part shortages and consumer electronic shortages are just a portion of those problems.

Other trends have been long in coming, and they've been expected, even if many developers don't put them high on their to-do list. Security is among the top items and artificial intelligence is another. One need only look at the current crop of processors to hit the market like Arm's Cortex-M33 or Cortex-M55 families that incorporate secure boot and AI acceleration. Even FPGA vendors are getting into the fray with easier-to-use development tools that cater to secure AI applications.

All of this change leads to one of the major problems engineers and developers need to address: complexity. Complexity has grown significantly as the IoT wave came crashing down upon us well before the COVID-19 wave hit. Taken as a whole, this means that changes old and new are going to make our lives more interesting and challenging. ed

WILLIAM G. WONG | Senior Content Director

SALARY & CAREER REPORT 2021
ElectronicDesign.

ENGINEERING IS STILL A GREAT JOB

As revealed in our 2021 Salary Survey, science and engineering jobs continue to be fun and profitable.

Image courtesy of 33123424 © Dragonimages | Dreamstime.com

I was excited about engineering before I was in high school. I discovered programming then and you can tell I've been around for a while—my first exposure to software was programming in BASIC on a Teletype ASR 33 with punch paper tape. Georgia Institute of Technology was the starting point of being a real engineer with co-op sessions at Burroughs Corp. working with mini and mainframe design.

I've had a lot of jobs since then and backed into the editorial side of things because I had a knack for technical writing. I still dabble with hardware and software, but I've never lost the awe and excitement of engineering.

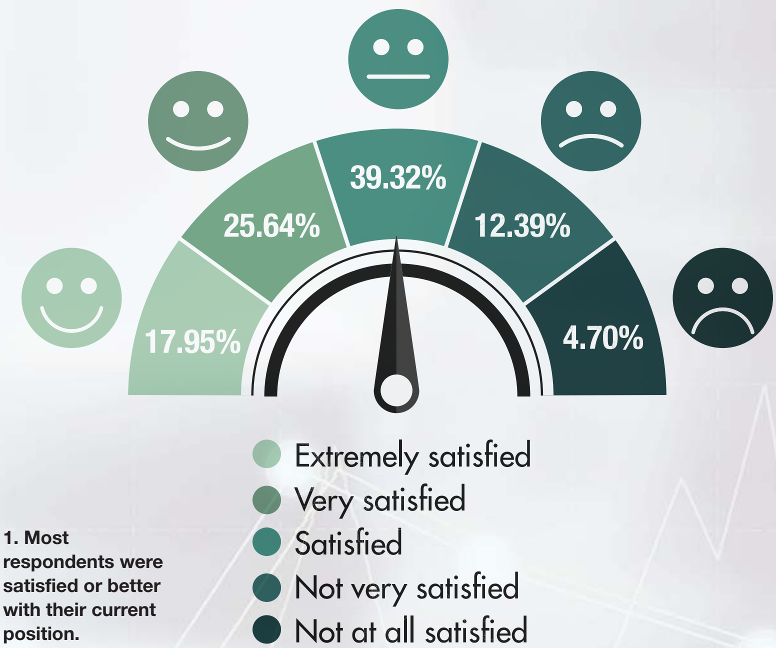
One thing that's always cropped up along the way was whether engineering was a worthwhile profession. I've always recommended it to those who were interested; I have three children who are all engineers, although none went into electric engineering or software like dear old dad or into mom's engineering field either.

I'm happy to say that most engineers and programmers want to stay in their jobs and would recommend them to others based on the results of our recent salary survey. The majority of engineers and programmers were satisfied with their current position, which is a good indication they chose the right profession *(Fig. 1)*. Likewise, the survey results show that more than 90% recommend engineering as a career choice to others. Still, getting into a good engineering school will be a challenge with an acceptance rate on the order of 8%.

Job satisfaction seems to match well with retention or it may be the sign of the times. Not a lot of engineers or developers are actively searching for a new position *(Fig. 2)*. Still, a substantial portion are keeping their eyes open for opportunities that may arise.

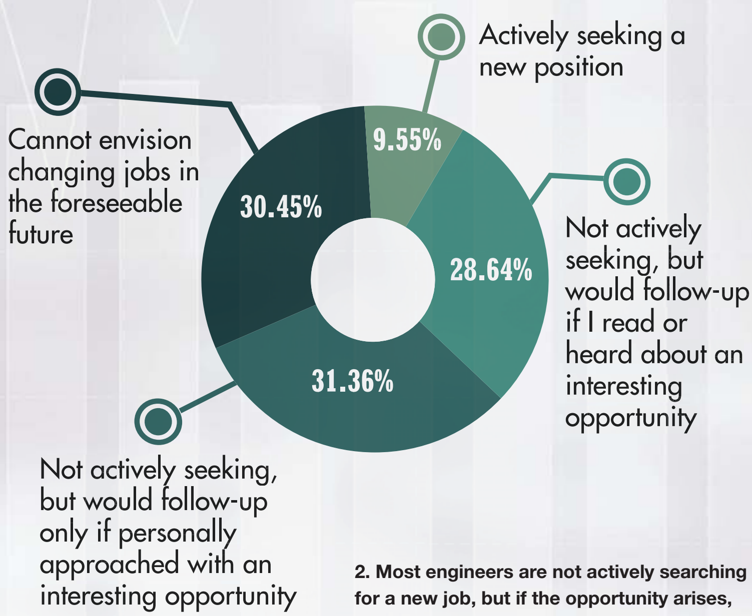
The challenge for companies that want to retain their staff include keeping them happy. Likewise, trying to attract new employees has become more

How satisfied are you in your current position?



1. Most respondents were satisfied or better with their current position.

What is your status regarding new employment?



2. Most engineers are not actively searching for a new job, but if the opportunity arises, one never knows.

interesting since location sometimes isn't a factor, given the ability to work remotely. Even developers who need to work with hardware may not be limited by remote access.

For example, Green Hills Software's Embedded Board Farm allows developers to test software on actual hardware without having it on a nearby lab bench *(Fig. 3)*. The boards are connected to power supplies, scopes, and logic analyzers that can be remotely controlled. Remote collaboration also is part of the mix these days, with access to shared debugging tools and video conferencing allowing people to work together from almost anywhere.

The outlook for engineers who want to stay put or move into new positions looks good, with most companies maintaining or increasing the number of engineers or developers on their staff *(Fig. 4)*. The fact that over 40% are looking for new employees should make those hunting for a new position happy.

On the flip side, matching up the engineers and developers to fill needed positions can be a challenge of human resources as well as engineering managers. Our results based on feedback from both our electrical/electronic and machine design respondents *(Fig. 5)* was pretty similar. Systems engineers are in demand everywhere, with mechanical engineers high on the list as well. Electrical/electronic engineering responses put a higher priority on areas like analog and RF.

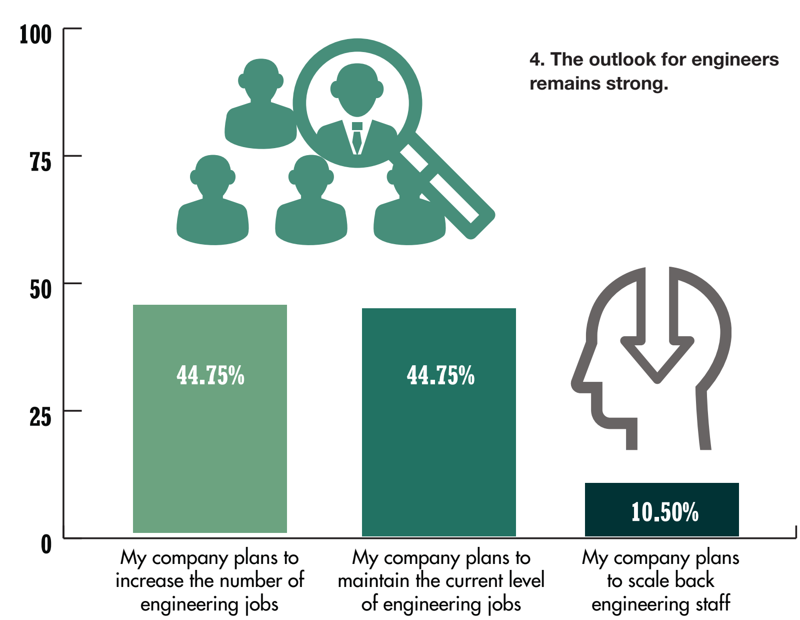
The only surprise I noted was that machine learning/artificial intelligence (ML/AI) was on the low side. I suspect that's being driven by two factors. First, ML/AI is being used in specialized areas such as automotive applications and that industrial applications are just starting to ramp up investigation or deployment. It also may mean that the solutions accessible to developers are being packaged in a way that allows their use without requiring lots of expertise. Of course, telling an engineer that he or she can't handle a job is like waving a red flag in front of a bull.

Besides, learning on the job is par for the course for engineers and programmers. But I leave that part of the story to other editors who are writing about our survey results.

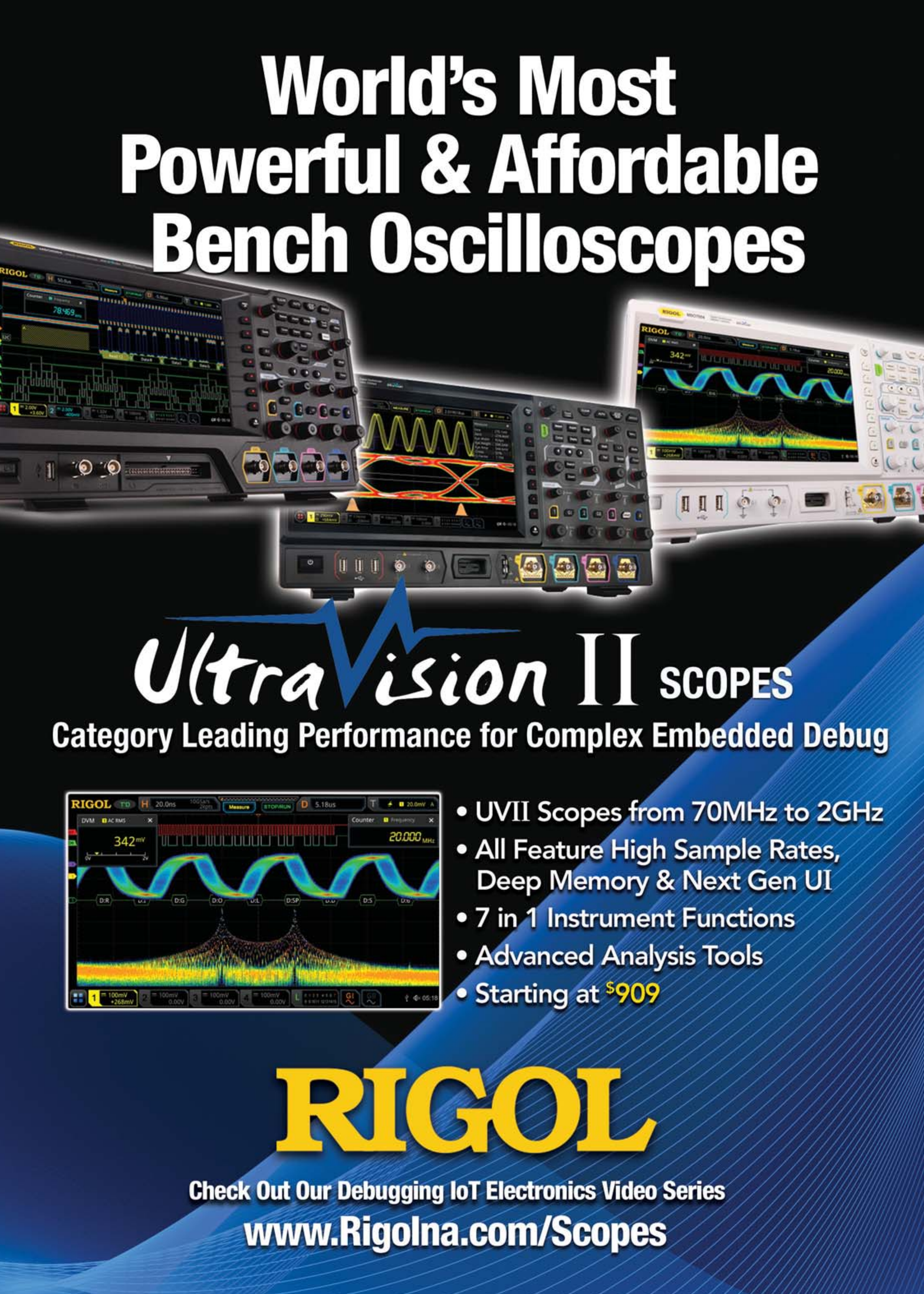
It does lead me to the chart highlighting concerns that keep engineers up at night *(Fig. 6)*. I had to contend with quite a few of those at the top end when managing a group of engineers and pro-

3. Green Hills Software's Embedded Board Farm allows remote developers to test real hardware.

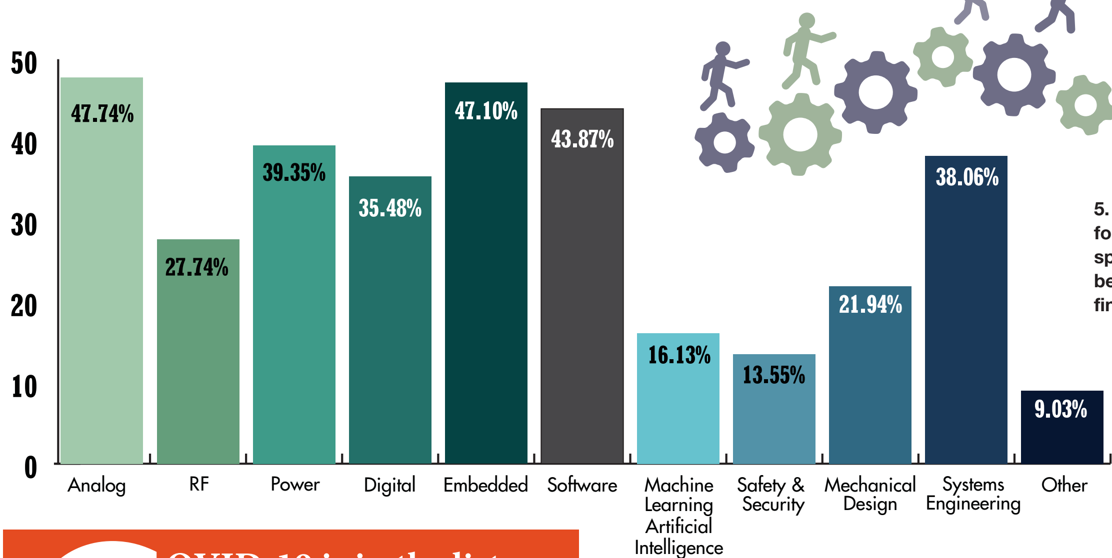
To the best of your knowledge, what is the engineering employment outlook at your company in the coming year?



4. The outlook for engineers remains strong.


World's Most
Powerful & Affordable
Bench Oscilloscopes
UltraVision II SCOPES
Category Leading Performance for Complex Embedded Debug
• UVII Scopes from 70MHz to 2GHz
• All Feature High Sample Rates, Deep Memory & Next Gen UI
• 7 in 1 Instrument Functions
• Advanced Analysis Tools
• Starting at $909
RIGOL
Check Out Our Debugging IoT Electronics Video Series
www.Rigolna.com/Scopes

For which engineering specialties are you having difficulty finding qualified candidates?



5. Engineers for these are specialties have become hard to find and hire.

> **COVID-19 is in the list and one that will likely be with us for at least the near future.**

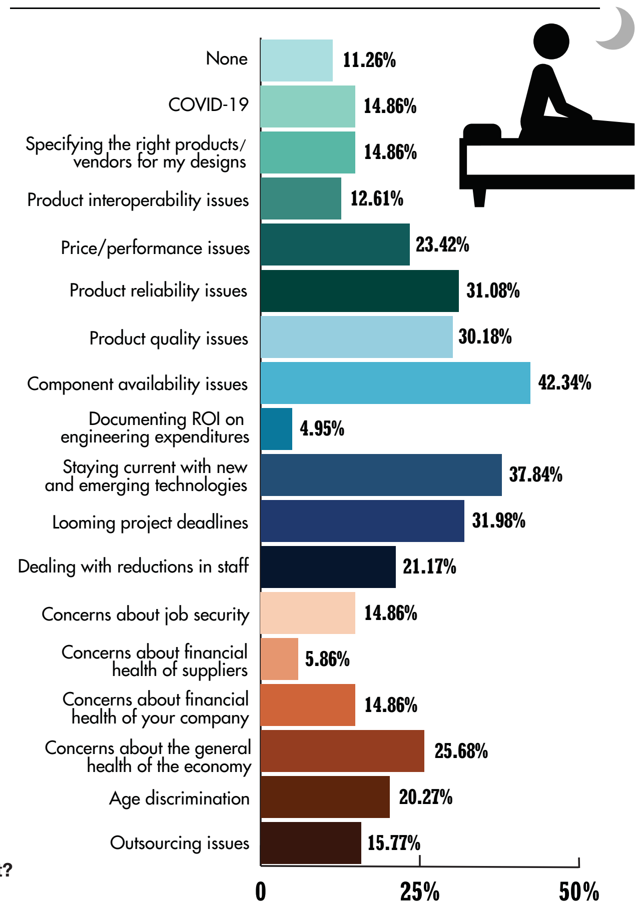
grammers, including deadlines and staying current with new and emerging technologies. ML/AI would fall in that category for many.

Product quality and availability rank high on the list. The survey was done before much of the major supply-chain issues arose, so these percentages may be increasing. We shall see how things fare in 2022. They weren't really an issue in 2020.

COVID-19 is in the list and one that will likely be with us for at least the near future. It has changed the landscape significantly for trade shows and working conditions, with remote attendance being the norm at this point whenever possible. Vaccinations are making it easier to work and meet in person, but the interest in continuing remote operations is high. As noted, it also can be a selling point for a job these days, with companies being more amendable to it versus just a couple years ago.

I help manage the local, annual Mercer Science and Engineering Fair and I like to let the students know that their interests can lead to fulfilling jobs in industry. Our survey results reinforce this view. I hope everyone's interest in science and engineering will be passed onto new and budding programmers, engineers, and developers so that they can enjoy the benefits and meet the challenges you encounter on a regular basis. ed

What are the professional issues that keep you up at night?



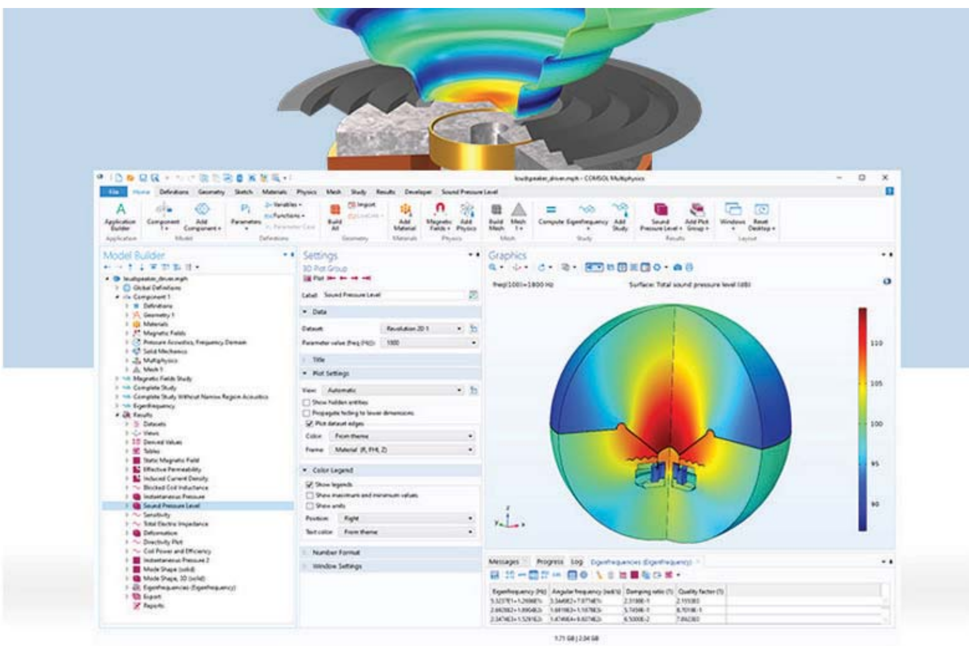
6. What issues keep engineering and software developers up at night? Quite a few, with component availability rising to the top this year.

FREE
ARCHIVED WEBINAR

Registration URL: https://www.electronicdesign.com/resources/webinars/webinar/21177019/loudspeaker-simulation-and-optimization-in-comsol-multiphysics?sti=ED-COMSOL1110

Loudspeaker Simulation and Optimization in COMSOL Multiphysics®

Tune into this webinar to get an overview of the setup and simulation of high-fidelity models of loudspeaker drivers and their interaction with waveguides and cabinets (enclosures). The modeling approaches range from lumped circuit models coupled with finite element method (FEM) domains to fully fledged multiphysics electrovibroacoustics models. The approach chosen by the designer depends on the goals of the simulation and also at which stage of product development the simulation is used. Finally, we will look at the use of optimization in loudspeaker design, for example, for the horn or speaker suspension.



SPEAKER: Mads J. Herring Jensen, Technology Manager, Acoustics, COMSOL

Mads Herring Jensen joined COMSOL in 2011 and is the technology manager for the acoustics products. Mads has a PhD in computational fluid dynamics from the Technical University of Denmark. Before joining COMSOL, he worked in the hearing aid industry for five years as an acoustic finite element expert.

Hosted with

ElectronicDesign.

COMSOL

SALARY & CAREER REPORT 2021
ElectronicDesign.

JAMES MORRA | Senior Staff Writer

Our 2021 Salary Survey revealed that around 60% of engineers expect to see their compensation go up this year, a huge improvement compared to 2020.

ELECTRICAL ENGINEERS

See COVID Pain Turn into Salary, Bonus Gains

As the world collapsed under the strain of coronavirus last year, many companies delayed raises, canceled bonuses, and paused hiring for electrical engineers to reduce costs. Now they are starting to make up for all the belt-tightening.

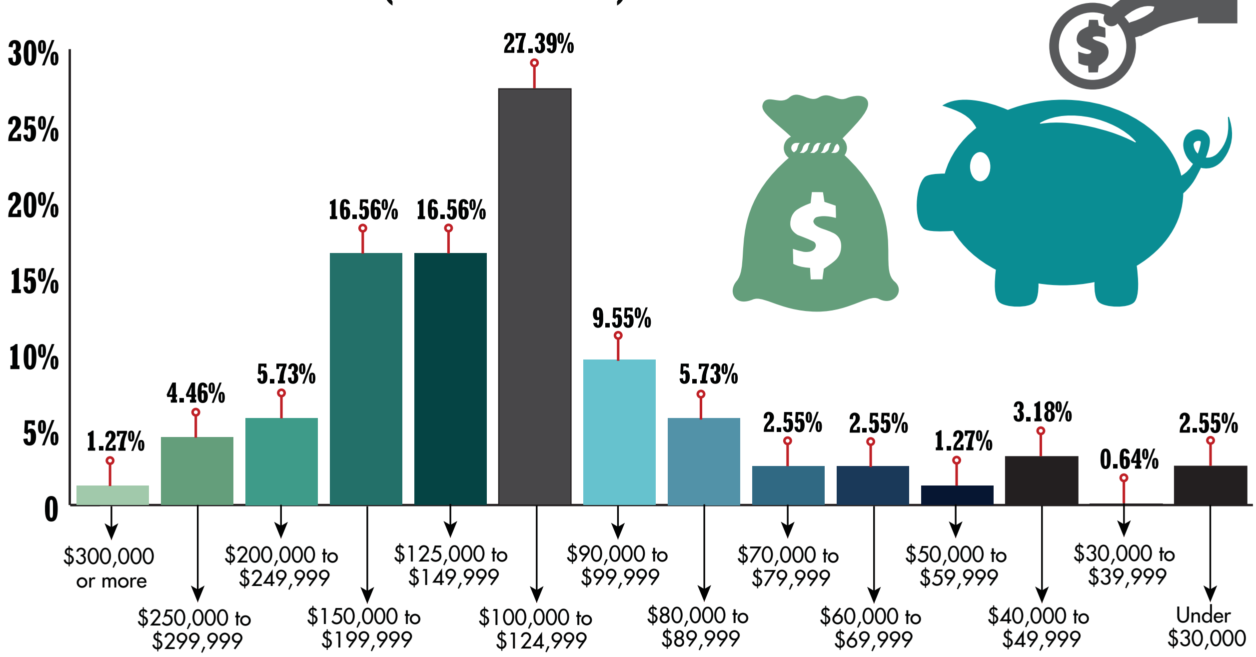
Employers are raising salaries and boosting hiring for electrical engineers as the economy roars back from the worst of the pandemic, according to the results of the latest annual survey from *Electronic Design* and Endeavor Business Media's Design Engineering Group. As employers grapple with a skills shortage, many are boosting bonuses or other perks to entice engineers from other jobs or hold on to the ones they have.

Last year, engineers made it through the pandemic better than most. Unlike large swathes of the workforce, most engineers were able to work remotely. While they continued to struggle with balancing tight deadlines, continuous education, and other challenges, they prospered for the most part. For more than 58% of survey respondents, their wages have not been directly affected by the economic turmoil caused by the pandemic.

But electrical engineers were not completely unscathed. The economic uncertainty unleashed by the virus pressured wages for many engineers, reversing some of the gains they enjoyed over the last decade or so. Last year, many engineers polled by *Electronic Design* said that, while their employers were keeping raises, bonuses, and a wide range of other perks in place, the wages gains were significantly weaker than usual.

Now companies seem to be picking up from where they left off. Around 60% of respondents say they will see their compensation rise in 2021, a stark improve-

What do you expect your total compensation to be in 2021? (BASE SALARY)



Design Note

PassThru Boost Controller for High Efficiency, Fast Transient Response Automotive and Industrial Supplies

Victor Khasiev, Senior Applications Engineer

Introduction

Boost power supply topology is increasingly popular in automotive and industrial electronics. A significant number of systems require a stable input rail where the upstream supply rail voltage can change significantly. Boost converters can be used to significantly increase application versatility. They enable new electronic equipment to seamlessly connect to any supply rail without requiring front-end redesign or multiple versions to cover various supply scenarios. Boost regulators can also support electronic devices that demand a high degree of rejection against input voltage drops. This is mostly related to automotive electronics where the supply rail can drop significantly—for example, during the cold cranking.

The LTC7804 simplifies the design of boost converters without sacrificing advanced characteristics. The main features of LTC7804 are low quiescent current, single output synchronous rectification, a wide input voltage range up to 40 V where the output can go up to 36 V, spread spectrum frequency modulation (SSFM), and an internal charge pump for high efficiency, low EMI PassThru™ operation.

Boost Converter for 12 V Input to 24 V Output

One of the advantages of a boost converter is that it can provide system immunity to front-end voltage drops, such as from a cranking automobile battery rail, in addition to providing a stable intermediate output rail.

Figure 1 shows the schematic of a boost converter consisting of a low pin count controller LTC7804, bottom FET Q1, top FET Q2, chock L1, and input/output filters. The schematic features a low component count solution, but can boost a 12 V rail to 24 V and deliver 6 A of the output current. The output current should be derated at low input voltages to keep input current below 17.5 A.

In this particular solution, the MODE pin is connected to GND, invoking Burst Mode® operation, which maintains high efficiency at light loads. The PLLIN/SPREAD pin is connected to $INTV_{CC}$, setting the switching frequency to SSFM operation, which eases the qualification against published EMI standards. This design was tested using dedicated current sense resistors, but it can operate entirely without current sense resistors, using optional DCR sensing instead. The efficiency of this solution is shown in Figure 2.

Rejection of the Input Voltage Drops and PassThru Mode Operation

One interesting application of the LTC7804 is supplying automotive audio amplifiers and preamplifiers. The goal of this application is twofold. First, the LTC7804 rejects sudden drops in the input voltage—for example, during cold cranking. Second, it essentially shorts the input to the output when input voltage rises above the output level to maximize efficiency, such as during load dumps. The voltage output of a preamplifier power

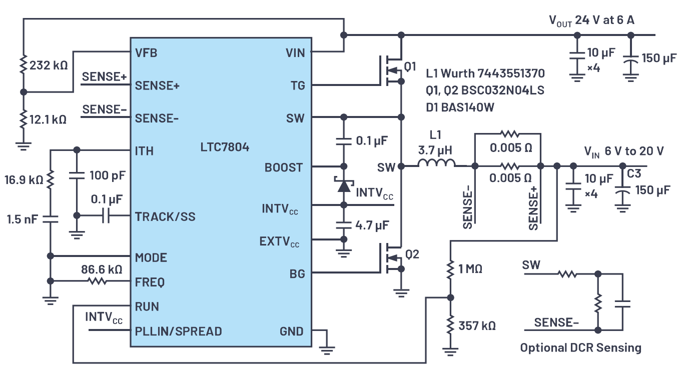


Figure 1. Electrical schematic of the boost converter based on LTC7804 with V_{IN} 6 V to 20 V and V_{OUT} 24 V at 6 A.

supply is set to a value a little less than the automotive rail voltage (around 10 V) from the typical 12 V automotive voltage rail. If input voltage is equal to or above the set value, then the input should pass directly to the output. If the input voltage drops below the desired intermediate voltage, the boost converter keeps its output at the set value. The term PassThru is used to describe this direct input-to-output mode of operation.

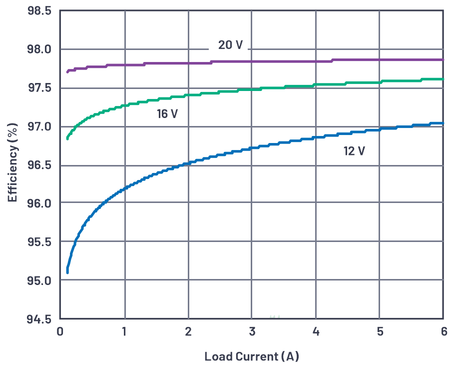


Figure 2. Efficiency curves of the boost converter on Figure 1.

Figure 3 shows a complete schematic for the boost solution. It's similar to the solution shown in Figure 1, but the connections of the control signals are slightly different. The MODE pin is connected to $INTV_{CC}$ through a 100 kΩ resistor to select pulse skip operation. Burst Mode operation is not supported in this application, because to enable PassThru operation, the top MOSFET gate charge pump must be enabled—it is disabled in Burst Mode operation. The PLLIN/SPREAD pin is tied to GND to disable the SSFM feature, as it is important for the power supplies for some audio systems to operate at a fixed frequency. If knowing the frequency is a real concern, then synchronization to an external clock through the PLLIN/SPREAD pin is advised; that, or connect the MODE pin directly to $INTV_{CC}$ to select forced continuous conduction mode at the operating frequency set at the FREQ pin.

Figure 4 shows how the solution functions through operational waveforms. In the test, the input voltage starts at 14 V, above the preset converter output of 10 V. The gate of the top MOSFET Q1 is high and Q1 is ON—it is completely enhanced. The LTC7804 internal charge pump can keep a converter in this state indefinitely. This is PassThru mode, where there's no switching action, with the input voltage passed directly to the output at 14 V. PassThru mode is enabled as long as the input is above or equal to the desired output as shown in the waveforms. The output voltage holds at 10 V even as the input drops to 5 V. The switching action begins once the input drops below a preset value to keep the output exactly at this level. The GQ1-V_{OUT} waveform is the differential voltage on the Q1 gate (GQ1 node) relative to Q1 source (V_{OUT}).

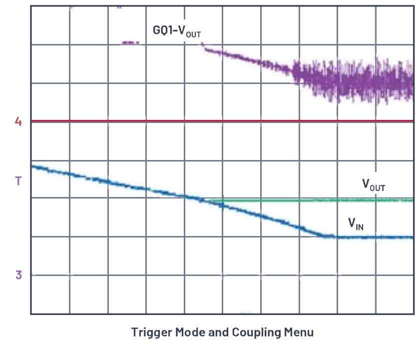


Figure 4. PassThru operation at V_{IN} > V_{OUT}. V_{IN}, where V_{OUT} 5 V/div, the time scale is 1 ms/div, and GQ1-V_{OUT} is the oscilloscope's mathematical function with 2.5 V/div.

The switching frequency of both converters is around 500 kHz to balance efficiency and size, but can be increased to 3 MHz if the inductor (L1) size must be minimized. Both solutions presented in this design note were verified and tested on DC2846A.

Conclusion

The LTC7804 controller significantly simplifies the design of highly efficient boost converters. Available output power is easily scaled by employing the same schematic and changing external components. The high switching frequency enables significant reduction in the size of the inductor. The internal charge pump and synchronous rectification guarantee maximum efficiency when the input voltage drops far below or rises far above the output level, making the LTC7804 an ideal controller for automotive electronics. Low quiescent current also preserves battery life in automotive and always-on systems.

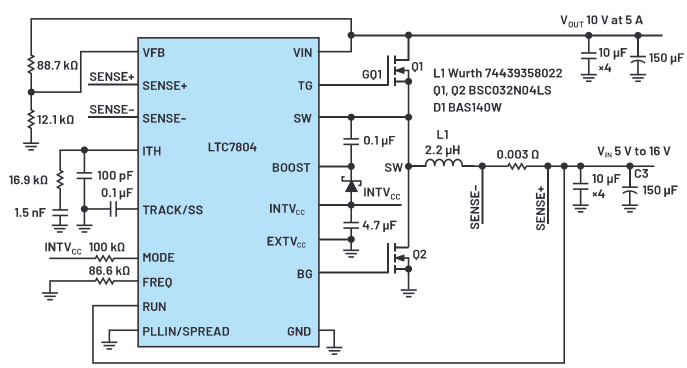


Figure 3. The boost converter is able to operate in PassThru mode, with V_{IN} 5 V to 16 V and V_{OUT} 10 V at 5 A.

ment from the 42% of engineers who saw raises in 2020.

Around 650 engineers responded to the survey, volunteering to share details about their salaries, bonuses, and other sources of compensation with *Electronic Design*, *Evaluation Engineering*, and *Microwaves & RF*. Engineers said they have gained a stronger hand as jobs are abundant, salaries are rising, and companies compete for talent. A negligible number of engineers say they are losing out on raises or bonuses in 2021.

But they remain upbeat about their prospects. Buoyed by demand for highly skilled engineers, around 70% feel the potential for salary advancement in engineering is at least as favorable now as it was pre-pandemic.

Several factors impact how engineers are compensated, including education, experience, title, seniority, age, location, and the status of the broader economy. But the survey results reveal that electrical and electronics engineers have very high-paying jobs. Engineers say that they will have a median base salary of $100,000 to $124,999 in 2021. Also, 58.5% report a median base salary in the range of $100,000 to $199,999.

Many employers plan to pay out bonuses, supplementing engineers' salaries with a median bonus of $1,000 to $1,999. Around 35% of respondents say they are in store for $5,000 or more in bonus pay this year.

At the top of the pay scale, engineers in management and executive roles not only have higher salaries, but they also earn thousands of dollars more in stocks and through employer-led share-matching programs. In contrast, rank-and-file engineers only get bonus pay and other perks on top of their base salaries. Just 30% say stocks are part of their compensation package, and approximately 12% are counting on $10,000-plus in stock awards in 2021.

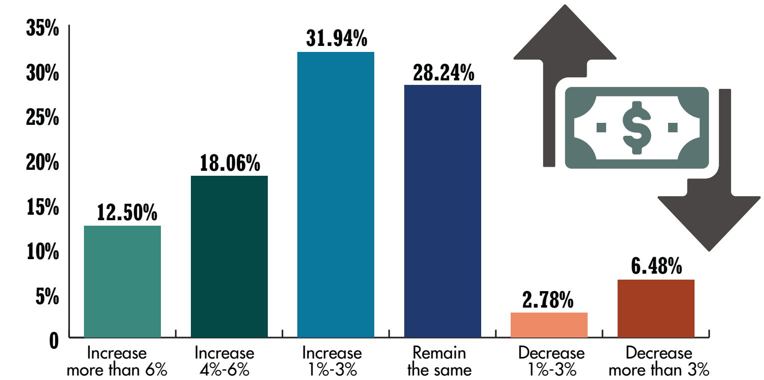
Around 14.5% said that compensations rates will jump by more than 6% in 2021, about twice the percentage last year, and about 30% of respondents revealed they are seeing an increase of 1% to 3% in compensation. According to the data, about 16% of engineers say they will see compensation boosts of 4% to 6% in 2021, signaling that electronics companies are boosting pay for engineers as the economy continues to rebound.

Only around 10% say they expect a pay reduction in 2021—a significant improvement from the 20% who last year said wages would contract due to delayed bonuses, pay cuts, or other cost-saving measures by their employers.

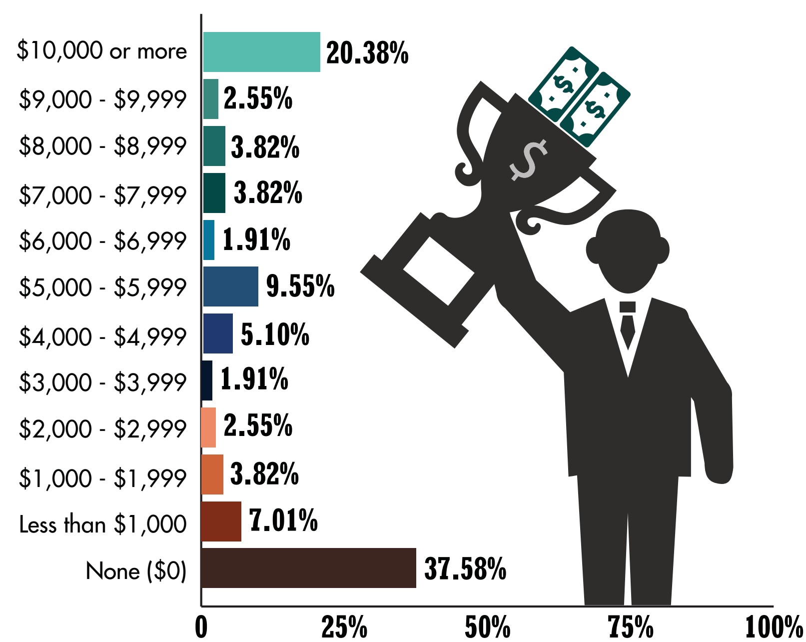
Around 30% say they are not getting raises or seeing pay cuts this year, whether because of economic pressures, business challenges, or other factors, such as older engineers reaching the top of their pay range. But at a time of rapidly rising inflation, those paychecks may not go as far as they used to.

More than 60% of respondents say their employer sufficiently compensates them for their work, and 37% feel that their compensation is as competitive as other firms pay engineers with the same job.

How will your total 2021 compensation (salary, bonuses, etc.) compare to what you earned in 2020?



Bonuses



Another 19% report they are probably better compensated than their peers at other employers, while 29% say their wages are “somewhat” less competitive than what others are willing to pay in 2021.

At the same time, given the grind of electrical and electronics engineering, others feel that these wage gains are falling short of what they should be. Many engineers grappling with long hours, tight deadlines, and the challenges of continuing education worry that their salaries are out of step with what they bring to their job. Among engineers who claim to be under-compensated, most feel entitled to a raise of 10% to 25%.

While many engineers feel as though they deserve to be making more money, around 90% say they would recommend engineering as a career with—all things considered—a promising future and competitive pay.

Only 30% of respondents have debated whether to leave engineering permanently or take a detour into another industry. Many workers report feeling burned out by the unrelenting demands of the job and want to pursue a career with a better work-life balance. Others are seeking new challenges to shake up their work situation. Among engineers who have considered leaving, approximately 31% hope to make more money.

As the economy recovers from last year’s pandemic-induced slump, employees are quitting their jobs at a record clip this year. The flood of departures is exacerbating a skills shortage in a wide range of industries, forcing companies to pay more to stand out and lure new employees. The mass resignations mark a sharp turn from last year, when workers craved stability as the virus spread, leaving economic turmoil in its wake.

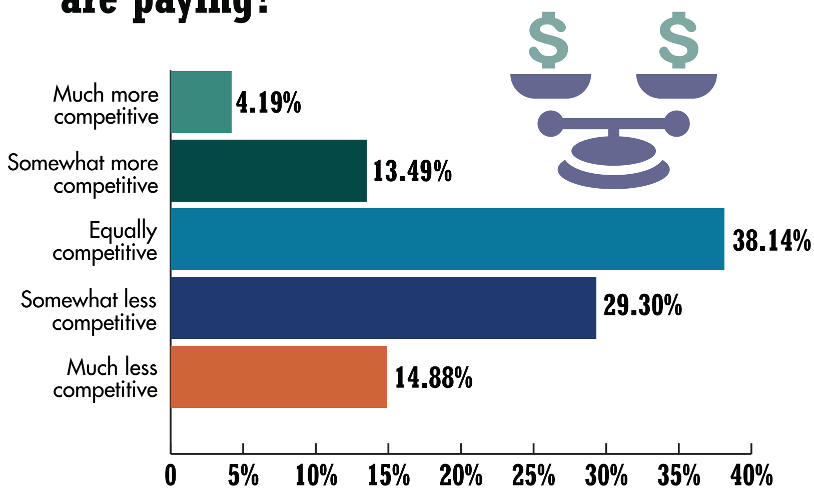
But it is unclear whether the shortages are specifically responsible for making companies in the electronics industry more willing to raise salaries or preventing them from freezing wage growth or delaying bonuses.

The struggle to find engineering talent predates the pandemic, and it could have to do with the types of jobs engineers are comfortable taking than it does with a worker shortage. Even so, high-end engineering skills continue to be coveted: Around 54% of respondents from last year’s salary survey said their companies were having hiring troubles. Now? 67% say they are struggling to locate qualified candidates for open positions.

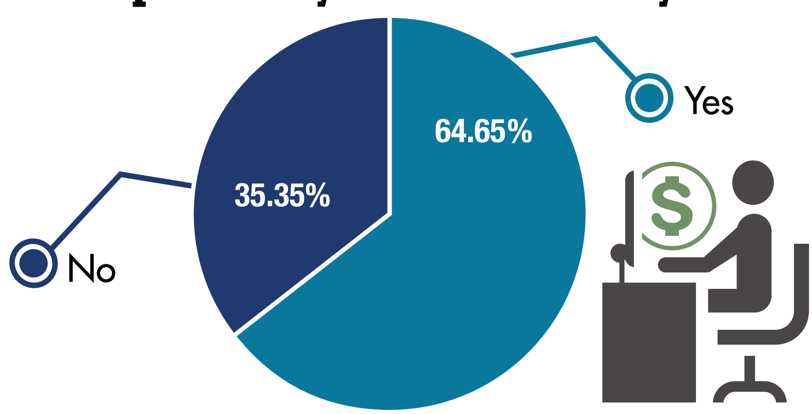
While some industries are churning workers more than others, the number of workers quitting may help explain why so many employers are struggling to plug holes in their engineering departments. More than 70% of respondents feel that there is a shortage of skilled engineers. Nevertheless, companies plan to keep hiring new workers, with 42% saying that they work for companies looking to hire more engineers in 2021.

Engineers are also keeping the door open to changing jobs. Many say that moving to a management role or changing jobs are the only ways to guarantee salary growth in engineering. Around 11% are seeking out a new job, and 31% say

Generally speaking, how do you think your compensation package compares with what other engineering employers are paying?



Do you feel your company adequately compensates you for the work you do?



that, while they are not actively looking to switch, they would follow up if personally contacted with a job. Another 29% say they would follow up if they heard about a promising opportunity.

Around 29% of engineers are staying put and have no plans to start a new job in the foreseeable future.

The survey signals that while some engineers are walking out the door and others are eyeing open positions, most are staying where they are, possibly in part due to the ongoing uncertainty around the virus and the stability of their current jobs. Only 7% of engineers changed jobs within the last year, and among them, 13% were promoted to a new position with their current employer, while 23% left to pursue another opportunity.

Around 15% of engineers who changed jobs say they landed at a new employer after losing a previous job.

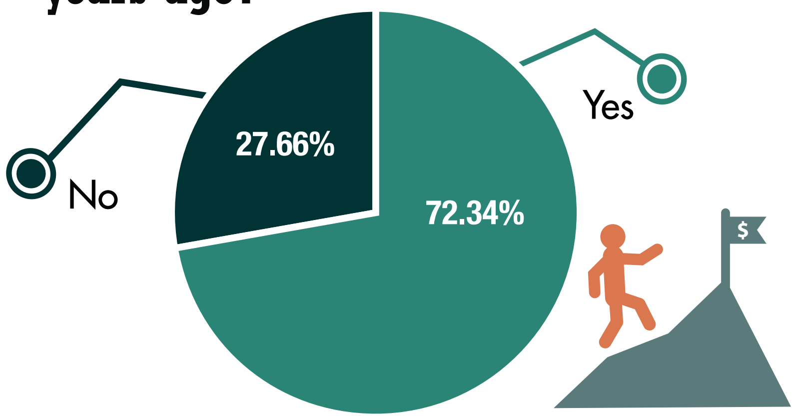
Given the challenges of scouting and hiring engineering talent, many companies are trying to hold on to the ones they already have. More than 60% of respondents say their employers are at least as focused on employee retention this year compared to last year. In the short term, the talent shortages could give engineers more bargaining power to negotiate a raise or promotion, allowing them to avoid changing jobs.

Employers are increasing non-wage compensation and other perks to remain on their engineers' good side. A number are paying for continuing education, as mounting competition for skilled engineers underscores the need to nurture new skills internally. While companies suspended travel because of lockdowns last year, engineers say some are reopening their wallets to pay for travel to industry conferences or training courses.

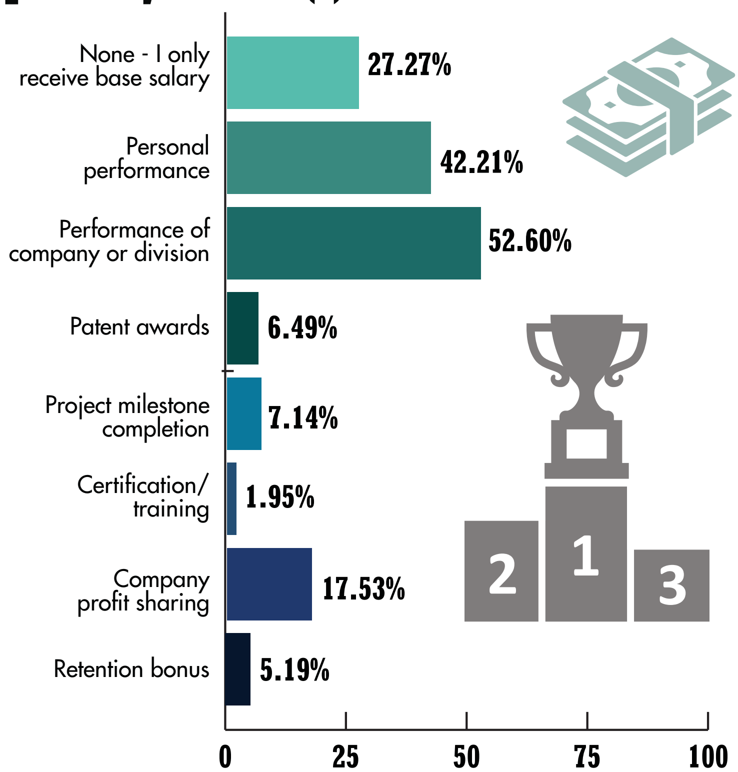
Covering the cost of healthcare continues to be one of the top priorities for employers, respondents said.

Engineers have long struggled with feeling that employers look at them as interchangeable assets. But the current engineer shortage could help bring about change long-term—and possibly better pay in the process. ed

Do you believe that a career path in engineering and the potential for salary advancement is as promising today as it was five years ago?



Of the bonuses and other direct cash payments over and above base salary that you receive, please specify the primary reason(s) for them?



DR. FREDERIC NABKI | Co-Founder and CTO, SPARK Microsystems

Vladyslav Starozhylov, Dreamstime.com

What's the Difference Between Bluetooth and UWB for High-Speed Data and Multimedia?

For nearly 20 years, Bluetooth has dominated as the short-range technology for wirelessly connected devices. But UWB's latency and power-efficiency advantages position it as a compelling alternative with faster, freer dataflow and low power consumption.

Bluetooth and ultra-wideband (UWB) short-range wireless technologies both rose to prominence at the turn of the century, and their development paths have been driven by the unrelenting need to reduce power consumption and extend battery life for an endless proliferation of wirelessly connected devices.

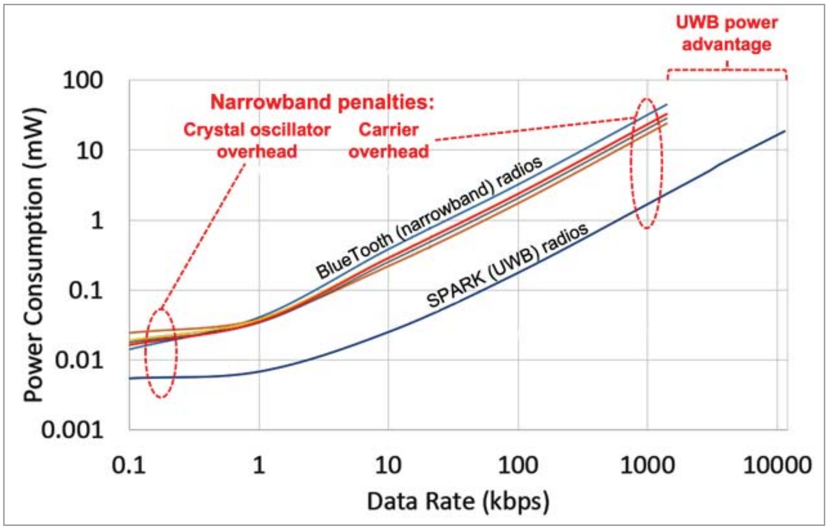
Bluetooth Low Energy (BLE) was ratified in 2006 to address the early power-consumption deficiencies of Bluetooth. More recently, Bluetooth 5.2 added features to reduce consumption for targeted applications like audio. However, these modifications are strictly incremental. Fundamentally, reductions in power consumption are physically limited by the Bluetooth architecture—a carrier-based transceiver will always require a significant amount of power to start, stabilize, and maintain its RF oscillator.

The *figure* shows the two significant power penalties inherent to all narrowband radio architectures, including Bluetooth.

First, crystal oscillator overhead *(lower left)* cripples low data-rate performance. Bluetooth uses a ~20-MHz crystal oscillator, which requires a few milliwatts to power. When efficiently optimized, UWB radios can operate with impulses that don't require a high-frequency crystal oscillator and can be designed to operate with a low timing power-consumption overhead. Much depends on the UWB optimization technique, though, so this is an area that should draw scrutiny.

Many of today's UWB technology implementations must in fact use *higher*-frequency crystal oscillators than what's required for BLE. Meanwhile, advanced UWB implementations can utilize crystal oscillators down to 32-kHz timing.

Second, the modulated carrier overhead *(upper middle in the chart above)* penalizes high data-rate performance. Transmitting a large amount of data over a narrow bandwidth channel such as that used in Bluetooth radios requires lots of time and power.



These are the two significant power penalties inherent to all narrowband radio architectures, including Bluetooth.

Large amounts of data can be transmitted with UWB far more quickly because it's spread across a wide bandwidth, keeping the transmitter on for a much shorter duration and significantly reducing power consumption. This means for the same amount of consumed power, UWB can transmit much more data *(far upper right)*.

This owes to the time-frequency duality, well encapsulated by the Fourier transform. In simple terms, this duality states that if you have an infinitely long periodic time signal, it will have an infinitely small bandwidth. On the other hand, if you have an infinitely short impulse signal, it will have an infinitely large bandwidth. In other words, you can trade time for bandwidth.

Ultra-wideband enjoys a clear inherent advantage over narrowband given its allocation and support over a large portion of radio spectrum. A UWB signal is defined as a signal having a spectrum larger than 500 MHz. In the United States, the Federal Communications Commission (FCC) in 2002 authorized the unlicensed use of UWB in the frequency range from 3.1 to 10.6 GHz.

UWB systems use short-duration (i.e., nanosecond timescale) electromagnetic pulses for high-speed transmission and reception of data over large bandwidths. They also have a very low duty cycle, which is defined as the ratio of the time that an impulse is present to the total transmission time.

Bluetooth vs. UWB for Positioning

After two decades of maturation, Bluetooth today is nearly ubiquitous in the battery-powered wireless-device market, spanning smartphones/tablets, earphones/headsets, gaming peripherals, IoT sensors, and more. For wireless apps that could get by with high latency and highly compressed audio signals, Bluetooth has delivered an acceptable user experience for some wireless apps. However, it could be argued that Bluetooth has reached its point of diminishing returns.

Today, UWB is emerging as a compelling successor to Bluetooth/BLE for the next generation of low-power short-range wireless applications. Consumer electronics manufacturers like Apple, Samsung, and others sure to follow are leveraging UWB spectrum for the delivery of electromagnetic impulses for applications like positioning for object/asset tracking, as exemplified by Apple's AirTags. This is a narrow application of UWB's technology potential, but nonetheless an effective one.

In this capacity, UWB measures time of flight (ToF): an impulse is sent from one device to another, and we measure the time it took from transmit to receive. The distance between objects is determined accordingly, and this can be measured with picosecond accuracy with UWB chips. Leveraging onboard antennas, measurements are then able to be correlated to determine a signal's angle of arrival, and UWB "tagged" objects can consequently be located with accuracy down to a mere 10 cm.

Bluetooth technology comes nowhere close to matching this precision, as it utilizes received signal strength (RSS) to measure spatial distance. RSS is a very simple technique to implement and can be used by any wireless transceiver, which explains why it's so widely used. However, it's severely limited in its accuracy: The perceived distance between two immobile objects will change according to obstacles in their direct path, and BLE typically achieves positioning accuracy only to within several meters.

Positioning technology enabled with UWB—while extremely accurate—is exceedingly complex from a design perspective and therefore extremely power-hungry. As a result, UWB chips used today for object tracking are actually less power-efficient than Bluetooth chips/radios by as much as 10X. So, while UWB is well-suited for positioning, it's a power-intensive application by nature and at the end of the day, there's no device-level power benefit delivered with UWB.

UWB for High-Speed Data and Multimedia Communication

The aforementioned time-frequency duality expresses how time and bandwidth are interchangeable. If one wants to compress in time a wireless transmission, it requires more frequency bandwidth. This property can be used to increase the accuracy of positioning and ranging, but these capabilities represent a mere sliver of UWB's potential.

Another very interesting capability enabled by the time-frequency duality is that it can reduce the latency in systems. This has huge implications for untold short-range wireless applications into the future.

Impulses delivered over ultra-wide bandwidth ensure extremely low latency—these signals can be sent in microseconds with UWB, whereas Bluetooth would take milliseconds. The end result is ultra-efficient wireless data communication. What's more, UWB implementations have demonstrated at least 10X less power consumption than BLE for non-positioning applications.

Bluetooth's latency penalties will only persist for applications like gaming, audio, and IoT, which is the chief reason why wired connectivity has lingered so stubbornly for peripherals and sensors used in these applications. We welcome the freedom of mobility that wireless affords us, but historically it's cost us quite a bit in terms of latency/delays, signal degradation, and battery drain.

Gaming

For gaming, speed is everything when it comes to outperforming one's opponents, and latency is therefore a major concern among die-hard gamers. When gamers press the mouse button, they want an instantaneous response, but Bluetooth can only deliver response speeds of 20 to 30 ms at best.

Leveraging UWB connectivity, SPARK has demonstrated sub-0.2-ms latency for UWB wireless gaming peripherals, and the company is well along the path to achieving sub-0.1 ms. This is far beyond what Bluetooth can do, and it's even faster than what many commercially available USB-wired mice can deliver today.

Audio

For audio, since Bluetooth is limited to a very narrow bandwidth, audio data compression must be applied to squeeze an otherwise bulky audio signal through a narrow pipe, which degrades the signal. Bluetooth codecs are inherently lossy in that lots of source audio data is stripped away. CD-quality audio is achieved with a 1,411-kb/s data rate—a Bluetooth codec renders that down to about 300 kb/s to be able to fit the audio stream within Bluetooth's limited data-rate capabilities.

UWB enables 10X more data throughput than BLE; thus, there's no need to compress the audio signal for wireless delivery to your UWB headset. This ensures that the sound stage one can hear with UWB headsets is considerably more detailed than what's possible with Bluetooth today, and exactly faithful to the audio source. These benefits extend to live music performance as well—UWB liberates performing musicians from cumbersome cables without sacrificing latency, allowing for wireless live performances.

IoT

The battery life of wireless sensors and devices is insufficient today for many IoT applications, leading to overly frequent recharge cycles, limited connectivity, and bulky batteries and/or costly maintenance. In addition, long latency makes wireless inadequate in applications requiring real-time sensing and communications.

With UWB, huge volumes of sensor data can be delivered with 60X lower latency and 40X better energy efficiency than legacy Bluetooth. This is hugely beneficial not only to IoT applications, but also to the myriad smart building, smart city, and AI-guided applications on the horizon that will require ultra-high-speed communication among sprawling networks of battery-powered wireless sensors.

Bluetooth technology is well-entrenched today and has served us reasonably well for the last two decades. However, UWB's stark latency and power-efficiency advantages position it as a compelling alternative for any wireless application requiring more data to flow faster and more freely with minimal power consumption. Everywhere Bluetooth resides today—across untold commercial and industrial applications, from our earphones to the edge—UWB can potentially reside tomorrow. ed

PAUL FADLOVICH | Director of Product Management, Antennas, Laird Connectivity

About **Smart Utilities IoT and Antenna Strategies**

Laird Connectivity's Paul Fadlovich discusses this fast-growing category of IoT and provides practical advice about antenna strategies for these design projects.

The utilities sector is one of the fastest-growing adopters of Internet of Things (IoT) technology, with hundreds of millions of wireless devices being deployed to support multiple areas of operations for electrical utilities, municipal water entities, and natural gas providers. These smart utilities applications have tremendous diversity, which presents many complex decisions to engineering teams, including significant challenges for connectivity that require a sophisticated antenna strategy.

1. Utilities are old-guard industries that will be slow to adopt new technologies like IoT.

Utilities have moved way ahead of the curve on IoT compared to other industries. In the most recent comprehensive report by Gartner, their research team predicted that utilities would have a total of 1.37 billion IoT endpoints (aka communication devices) by the end of last year, with aggressive growth predicted for the coming years.

Utilities are perceived as being slower adopters of new technology. However, the number of use cases and pain points for utility operations has been a huge driver for deployments where these new wireless technologies complement and augment the existing infrastructure utilities have in place.

2. Smart utilities IoT is largely focused on smart meters and smart thermostats.

Residential use cases tend to capture most of the attention when it comes to utilities IoT, but there's actually a long list of other use cases that go well beyond automated meter reading and smart thermostats in people's homes. That includes industrial IoT (IIoT) deployments in electrical generation plants and water treatment facilities. It includes IoT networks for transmission lines and pipes. And it includes vehicular IoT for fleets and other kinds of IoT deployments in numerous other complex areas of operations such as underground.

When you look at all of those collectively, it becomes clear that IoT is creating a massive digital infrastructure across nearly every aspect of a utility's operations.

3. IoT design projects for utilities tend to be concentrated on those residential devices, which are relatively straightforward Wi-Fi/ Bluetooth design projects.

Those kinds of devices are definitely an important category of projects that engineering teams will work on, but smart utilities applications go far beyond that. Your team will need to be proficient in IIoT projects for challenging RF environments like water treatment plants and generation plants. You also will need to successfully navigate the connectivity and RF challenges of outdoor implementations on towers and poles. And much more.

Smart utilities projects encompass every kind of IoT design project, which is a big part of what makes it challenging and fascinating. That variety of use cases, the range of RF environments, and the number of technologies involved also make antenna strategies such a critical factor. Engineering teams need to have different antenna strategies depending on the individual utility project.

4. The biggest antenna challenge will be all of the certifications required for the residential devices.

Certification is definitely important. Engineering teams should think hard about certification early in their design projects to avoid delays and unnecessary costs later in the design process. Pre-certified modules and antennas can dramatically simplify that process and ensure success, but there are far more vexing antenna decisions in smart utilities. For example, environmental factors that distort RF dynamics in some use cases, like antennas in the presence of large metal structures in distribution networks, can present complex decisions about antenna selection and placement.

5. Smart utilities projects may have a higher volume of endpoints because of the geographic scale of utilities deployments, but the antenna process is similar to other projects our team has worked on.

It's true that what you have learned on other projects will prove very useful for utilities-related IoT. Still, the variety and RF complexity of smart utilities applications will be the equivalent of a senior thesis that draws on everything you've done previously while also challenging you in ways that may be new.

Antenna selection is a perfect example. You may have a project that utilizes familiar elements like a combination of Wi-Fi and Bluetooth, but the RF dynamics of the utilities implementation site may require an antenna that can perform adjacent to thick concrete walls or nearby metal machinery. As another example, the IoT project at hand may involve a local network of Bluetooth-connected devices, which will be very familiar to you, but the implementation site is geographically located in an area where a cellular or LoRaWAN solution might be the only practical option for backhaul.

6. My antenna strategy will be very similar to other IIoT projects my team has done.

Your IIoT experience will be very valuable for smart utilities projects, but this vertical industry will throw you unexpected curveballs. IoT for utilities' large vehicular fleets is a great example of that. Electrical, gas, and water utilities have large fleets of specialized vehicles that must operate as mobile communications hubs for crews. They also must be equipped with all of the wireless technologies that are embedded into the IoT deployments being rolled out, so that workers can access and utilize real-time data relevant to their field work.

Antenna selection and placement is notoriously complex for vehicles because the RF dynamics are so different from model to model of even similar-looking vehicles. Two vehicles may look the same, but if one was designed with a metal roof and the other has a fiberglass roof, the RF dynamics will likely be dramatically different in ways that require very different antennas and installation locations on the vehicle body. For those reasons, it's important to conduct RF modeling, seek expert support about antenna selection and installation, and then conduct extensive testing before large-scale rollout to entire fleets.

7. Water ingress from rain is the biggest threat to IoT deployments located outdoors for these projects.

That's a major concern, but it's important to broaden the list of threats so that enough of a failsafe is designed into those devices, including the ruggedness of the antenna and quality of the antenna assembly. Water ingress from rain is a big one, but other forms of ice may create even bigger threats.

Sleet or snow that potentially builds up into a shell around the casing should be accounted for when picking antennas to ensure performance even during winter weather. Ice build-up on antennas may not only affect performance, but also damage the physical bonds between the antenna's radome and connectors from increased weight and wind loading.

Toughness of the outer casing/radome also is important for protecting the device from pecking birds or curious squirrels. All of this may mean you need an antenna that not only has strong RF performance, but also strong mechanical build qualities.

8. The antennas on my short list are all labeled as "rugged," so we'll be in good shape against those conditions.

Unfortunately, the term "rugged" is overused in the antenna industry without consistent standards for how well a solution will stand up to tough environmental conditions. For that reason, it's best to be skeptical of those labels and to work with your antenna partner to pick an antenna that's well-suited to the specific conditions you expect in a given implementation site. Be sure to also look for warranties that are long enough to demonstrate the manufacturer truly stands behind the ruggedness it's claiming. These devices need to be built to last, and the antenna can't be the weak link in the chain.

9. The antennas I'm looking at have exactly the performance I'm looking for. The datasheet said so.

It would be incredibly helpful if the performance metrics in the real world matched what's listed in datasheets, but that's often not the case. The gap between what's on the page and what you see in the field can often be very wide, which is why you need to be skeptical about those numbers, particularly for key metrics like gain. Those specs for gain can be quoted as max gain, which may be deceptively high but not pointing in a useful direction for the given applications.

The reason for the gap is simple: Those datasheets are based on testing in lab environments that are often so idealized that they no longer reflect an actual implementation site, where other RF signals, concrete walls, plastic device casings, and many other factors impact performance. To ensure the antenna you implement performs as needed, your team should take the datasheets with a grain of salt, conduct extensive testing in conditions similar to the real-world implementation, and make final selection and installation decisions based on that information.

10. I have always used off-the-shelf antennas. Those will definitely meet my needs.

There are thousands of off-the-shelf antenna options, so chances are good that you can find one meeting your needs—particularly if you work with an antenna partner who can help you look deeper than the datasheets. But some utilities IoT projects have such unique needs—often because of the RF environment and combination of technologies required—that a custom antenna might be the best, or even the only, solution because it avoids the compromises of off-the-shelf options.

11. Will all of these IoT projects for utilities lower my electricity bill? My teenagers are driving my power bill through the roof with all of their devices.

I can't promise that, but these IoT projects are increasing safety for utilities workers, helping utilities manage spikes in demand that enable the rollout of more renewable energy assets. So, you may not see a lower bill, but this is building a foundation for next-generation utilities operations that hopefully helps us consume energy and water in more sustainable ways.

Design Solution

RAFAEL TAUBINGER | Technical Marketing Specialist, IAR Systems

Getty Images | #1145394425

Techniques to **Minimize Code Footprint** in RISC-V-Based Apps

In this article, we'll look how developers can help the compiler make better decisions about what to do with their code to achieve optimizations in RISC-V-based applications.

The RV32I is the base instruction set that can get the standard extensions listed in *Figure 1*, such as:

- M (Integer Multiplication)
- A (Atomic Instructions)
- F (Single Floating Point)
- D (Double Floating Point)
- C (Compressed Instructions)
- B (Bit Manipulations)
- and so on….

Most extensions *(Fig. 1)* are ratified or frozen, but new ones are currently being worked on. An example of supported extension in various cores can be seen in *Figure 2*.

If we take the generic device RV32, we can recognize that it supports M, F, D, and C. C (Compressed Instructions) reduces static and dynamic code size by adding short 16-bit instructions for operations, resulting in average 25%-30% code-size reduction, and leading to lower power consumption and memory use. In addition, the RV32E base instruction set (embedded) is designed to provide an even smaller base core for embedded microcontrollers with 16 registers.

Designers are free to implement their own extensions for specific needs, e.g., machine learning, low-power application, or optimized SoC for metering and motor control. The purpose of the standard extensions or custom extension is to achieve faster response time from calculations and processing performed in hardware that require mostly one or just a few cycles.

Why Professional Tools for RISC-V?

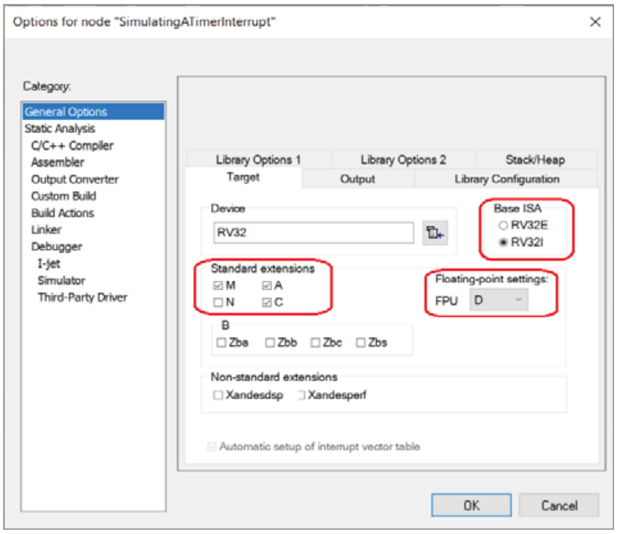
With the growth of RISC-V, the need arises for professional tools that can take full advantage of the core features and extensions. A well-designed and optimized SoC also should run the best optimized code so that companies can innovate fast, have outstanding products, and derive the best cost benefit out of it.

When it comes to code density, every byte that can be saved counts. Professional tools help to optimize the application to best fit the required needs. By optimizing the application, customers will be able to save money by using devices with smaller memory or aggregate value by adding functionality to the existing platform *(Fig. 3)*.

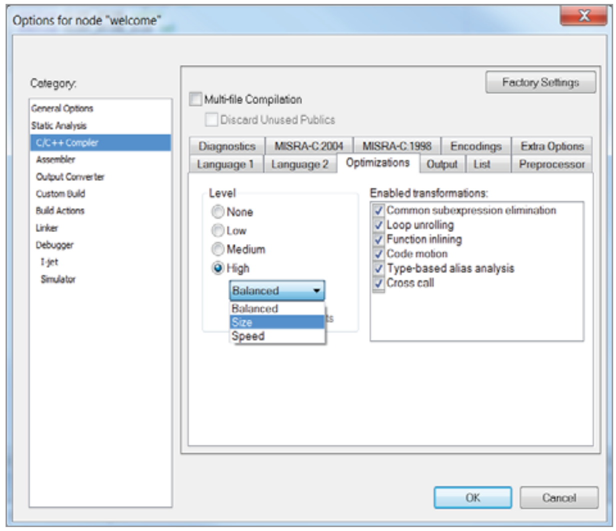
ISA base and extensions

Name	Description	Version	Status[a]	Instruction Count
Base				
RVWMO	Weak Memory Ordering	2.0	Ratified	
RV32I	Base Integer Instruction Set, 32-bit	2.1	Ratified	49
RV32E	Base Integer Instruction Set (embedded), 32-bit, 16 registers	1.9	Open	49
RV64I	Base Integer Instruction Set, 64-bit	2.1	Ratified	14
RV128I	Base Integer Instruction Set, 128-bit	1.7	Open	14
Extension				
M	Standard Extension for Integer Multiplication and Division	2.0	Ratified	8
A	Standard Extension for Atomic Instructions	2.1	Ratified	11
F	Standard Extension for Single-Precision Floating-Point	2.2	Ratified	25
D	Standard Extension for Double-Precision Floating-Point	2.2	Ratified	25
Zicsr	Control and Status Register (CSR)	2.0	Ratified	
Zifencei	Instruction-Fetch Fence	2.0	Ratified	
G	Shorthand for the IMAFDZicsr Zifencei base and extensions, intended to represent a standard general-purpose ISA	N/A	N/A	
Q	Standard Extension for Quad-Precision Floating-Point	2.2	Ratified	27
L	Standard Extension for Decimal Floating-Point	0.0	Open	
C	Standard Extension for Compressed Instructions	2.0	Ratified	36
B	Standard Extension for Bit Manipulation	0.93	Open	42
J	Standard Extension for Dynamically Translated Languages	0.0	Open	
T	Standard Extension for Transactional Memory	0.0	Open	
P	Standard Extension for Packed-SIMD Instructions	0.2	Open	
V	Standard Extension for Vector Operations	1.0RC	Open	186
N	Standard Extension for User-Level Interrupts	1.1	Open	3
H	Standard Extension for Hypervisor	0.4	Open	2
S	Standard Extension for Supervisor-level Instructions[26]	1.12	Open	7
Zam	Misaligned Atomics	0.1	Open	
Ztso	Total Store Ordering	0.1	Frozen	

1. These are the standard extensions for the RISC-V ISA. *(Courtesy of Wikipedia - https://en.wikipedia.org/wiki/RISC-V)*



2. Standard extension support in the compiler is specified by these selections.



3. Compiler optimization transformations can be selectively enabled.

A professional compiler for RISC-V can generate, on average, 7%-10% smaller code when compared to other tools.

Writing Compiler-Friendly Code for Better Optimizations

An optimizing compiler tries to generate code that's both small and fast by selecting the right instructions in the best order for execution. It does so by repeatedly applying a number of transformations to the source program. Most optimizations follow mathematical or logical rules based on a sound theoretical foundation. Other transformations are based on heuristics, where experience has shown that some transformations often result in good code or open up opportunities for further optimization.

So, the way you write your source code can determine whether an optimization can be applied to your program or not. Sometimes small changes in the source code could significantly impact the efficiency of the code generated by the compiler.

Trying to write your code on as few lines as possible, using ?:-expressions, postincrements, and comma expressions to squeeze in a lot of side effects in a single expression, will not make the compiler generate more efficient code. The best hint is to write your code in a style that's easy to read.

Developers can help the compiler make better decisions by paying attention to the following hints in the source code:

1. Make a function call only once. A compiler generally has difficulty looking into common subexpressions because the subexpressions can have side effects that the compiler may not know a priori if they're necessary. Hence, the compiler will make multiple calls to the same function when instructed to do so, which wastes code space and execution overhead. It's better to assign the function to a variable (which will most likely be stored in a register) and perform operations while it's in an easily accessed register *(Fig. 4)*.

Bad Example

```
void bad()
{
  /* Two calls to foo */
  if (foo(12) && SomeCondition)
  {
  ...
  }
  if (!foo(12) || SomeOtherCondition)
  {
  ...
  }
}
```

Good Example

```
void good()
{
  /* One call to foo */
  r = foo(12)
  if ( r && SomeCondition)
  {
  ...
  }
  if ( !r || SomeOtherCondition)
  {
  ...
  }
}
```

4. The compiler may be able to optimize redundant calls, but it's better to write the code properly.

Trying to write your code on as few lines as possible, using ?:-expressions, postincrements, and comma expressions to squeeze in a lot of side effects in a single expression, will not make the compiler generate more efficient code.

2. Pass by reference rather than by copy. When you pass a pointer to a primitive rather than the primitive itself, you save the compiler the overhead of copying that primitive somewhere in RAM or in a register. For a large array, this can save quite a bit of execution time. Passing by copy will force the compiler to insert code to copy the contents of the primitive.

Bad Example of Mixing Function Pointers and Integers

```
typedef char (* FuncPtr)(void);      /* pointer to function "char f()" */
typedef unsigned short uint16_t;     /* 16-bit value */

  /*     Macro to call a function… really?            */
  /* The argument (function name) is cast to an integer   */
  /* and then back to a function pointer, and then called */
#define Call(pF) ((FuncPtr)(uint16_t)(*(&pF)))();

char foo( void );   /* a function to call */
void main( void )
{
  Call(foo);   /* can jump to strange places if FuncPtr is > 16 bits */
}
```

5. Avoid becoming a castaway by using the correct cast.

Example

```
unsigned char gGlobal;  /* global variable */

void foo(int x)
{
  unsigned char  ctemp;
  ctemp = gGlobal;  /* should go into register */
  ...
  /* Calculations involving ctemp, i.e. gGlobal */
  bar(z);  /* does not read or write gGlobal, otherwise error */
  /* More calculations on ctemp */
  ...
  gGlobal = ctemp;   /* make sure to remember the result */
}
```

6. This is the right way to handle global variables and registers.

3. Use the correct data size. Some MCUs like an 8051 or AVR are 8-bit micros; some like the MSP430 are 16-bit; and some like Arm and RISC-V are 32-bit. When using an "unnatural" size for your core, then the compiler must create extra overhead to interpret the data contained therein, e.g., a 32-bit MCU will need to do shift, mask, and sign-extend operations to get to the value needed to perform its operation. It's therefore best to use the natural size of the MCU for your data types unless there's a compelling reason not to, i.e.. doing I/O. You also need a precise number of bits, or bigger types (such as a character array) would take up too much memory.

4. Using signedness appropriately. The signedness of a variable can affect the code that's generated by the compiler. For example, division by a negative number is treated differently (by the rules of the C language) than that for a positive number. Ergo, if you use a signed number that will never be negative in your application, you can incur an extra test-and-jump condition in your code that wastes both code space and execution time. In addition, if the purpose of a variable is to do bit-manipulation, it should be unsigned or you could have unintended consequences when doing shifting and masking.

5. Avoid becoming a castaway. C will often perform implicit casts (e.g., between floats and integers and between ints and long longs) and these are not free. Casting from a smaller type to a bigger type will use sign extend operations, casting to and from a float will introduce the need for the floating-point library (which can dramatically increase the size of your code). Naturally, you should avoid making explicit casts as much as possible to sidestep this extra overhead. This problem can be easily seen when a desktop programmer who is accustomed to using ints and function pointers interchangeably makes the jump to embedded programming *(Fig. 5)*.

6. Use function prototypes. If a prototype doesn't exist, then the C language rules dictate that all arguments must be promoted to an integer and—as previously discussed—this can link in unnecessary overhead from a runtime library.

7. Read global variables into temporary variables. If you're accessing a global variable several times within a function, you might want to read it into a local temporary variable. Otherwise, every time you access this variable, it will need to be read from memory. By putting it into a local temporary variable, the compiler will probably allocate a register to the value so that it can more efficiently perform operations on it *(Fig. 6)*.

8. Refrain from inlining assembly. Using inline assembly has a very deleterious impact on the optimizer. Since the optimizer knows nothing about the code block, it can't optimize it. Moreover, it's unable to do instruction scheduling of the handwritten block since it doesn't know what the code is doing (this can be especially damaging to DSPs). On top of that, the developer must inspect the handwritten code each time to make sure that it's correctly interspersed in the optimized C-code so that it doesn't produce unintended side effects. The portability of inline assembler is very poor, so it will need to be rewritten (and its ramifications understood) if you ever decide to move it to a new architecture. If you must inline assembler, you should split it into its own assembler file and keep it separated from source.

9. Don't write clever code. Some developers erroneously believe that writing fewer source lines and making clever use of C constructions will make the

"Clever" solution

```
unsigned long int a;
unsigned char     b;

/* Move bits 0..20 to positions 11..31
 * If non-zero, first ! gives 0      */
b |= !!(a << 11);
```

Straightforward solution

```
unsigned long int a;
unsigned char     b;

/* Straight-forward if statement */
if( (a & 0X1FFFFF) != 0)
  b |= 0x01;
```

7. Don't Write Clever Code 1: Comments are good but not an excuse for bad, clever code.

"Clever" solution

```
int bar(char *str)
{
  /* Calculating with result of */
  /* comparison.                */
  return foo(str+(*str=='+'));
}
```

Straightforward solution

```
int bar(char *str)
{
  if(*str=='+')
    str++;
  return foo(str);
}
```

8. Don't Write Clever Code 2: You're not making something more efficient by writing bad code.

code smaller or faster (i.e., they're doing the compiler's job for it). The result is code that's difficult to read, impossible to understand for anyone but the person who originally wrote it and harder to compile. Writing it in a clear and straightforward manner improves the readability of your code and helps the compiler to make more informed decisions about how best to optimize your code.

For example, assume that we want to set the lowest bit of a variable b if the lowest 21 bits of another variable are set. The clever code uses the ! operator in C, which returns zero if the argument is non-zero ("true" in C is any value except zero), and one if the argument is zero. The straightforward solution is easy to compile into a conditional followed by a set bit instruction, since the bit-setting operation is obvious and the masking is likely to be more efficient than the shift. Ideally, the two solutions should generate the same code. The clever code, however, may result in more code since it performs two ! operations, each of which may be compiled into a conditional *(Fig. 7).*

Another example involves the use of conditional values in calculations. The "clever" code will result in larger machine code since the generated code will contain the same test as the straightforward code and adds a temporary variable to hold the one or zero to add to str. The straightforward code can use a simple increment rather than a full addition and doesn't require the generation of intermediate results *(Fig. 8).*

10. Access structures in order. If you order your structure whereby you step from one element of the structure to the next instead of jumping around in the structure, the compiler can take advantage of increment operations to access the next element of the structure instead of trying to calculate its offset from the structure pointer. In a statically allocated structure, doing this will not save code since the addresses are computed a priori. However, in most applications, these are done dynamically.

Conclusion

Embedded compilers have evolved greatly over the last 30 years, especially as it pertains to their optimization capabilities. Modern compilers employ many different techniques to produce very tight and efficient code so that you can focus on writing your source in a clear, logical, and concise manner. Every developer strives to achieve the optimum efficiency in their software. Compilers are amazingly complex pieces of software that are capable of great levels of optimization, but by following these simple hints, you can help it achieve even greater levels of efficiency. ed

ALI OSMAN ÖRS | AI ML Strategy and Technologies, NXP Semiconductors, NXP Semiconductors
DAVID STEELE | Director of Innovation, Arcturus

Tuning Embedded Machine Learning to REAL-WORLD APPS

Machine learning based on deep-neural-network (DNN) architectures have demonstrated impressive results in numerous experiments, particularly in tasks such as the recognition of objects and people in images. Many of these experiments, as well as the first real-world deployments, were performed on high-performance cloud-server hardware that can deliver the required calculation throughput.

There are numerous applications where processing on cloud-server hardware is impractical. Issues such as communications latency and available bandwidth demand the use of local intelligence for processing.

Take the example of a camera system that's used to monitor activity at an automated teller machine (ATM) in situations such as social distancing during the recent COVID-19 pandemic. Banks have found it's important for them to introduce automated monitoring to check conditions by each ATM to make sure people in the queue or around the area don't stand too close together. In addition, they may want to ensure only patrons wearing masks are given access to a lobby-based ATM or the machines themselves.

AI models have demonstrated impressive results in experiments, but deploying them in real-world applications requires combining neural networks with pre- and post-processing steps. Thus the need for flexible hardware and firmware platforms.

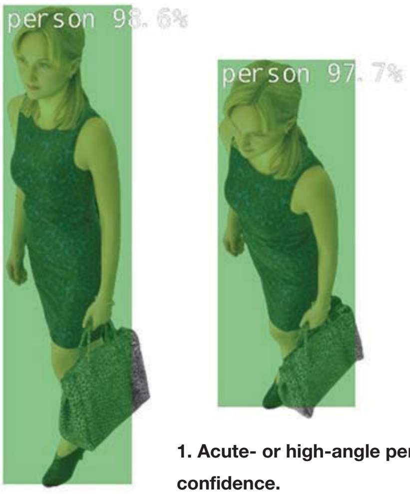
Sending live video from security cameras to the cloud is one possibility for processing the data. But this is potentially very costly and difficult to implement in all but dense urban centers. Furthermore, the communications latency makes it harder for the system to react to arrivals and crowd movements in a timely way. Processing the video data locally potentially provides much better responsiveness if hardware can satisfy the computational demands of the appropriate image-recognition pipeline.

Model Choices

Many choices are available for integrators looking for edge-computing processors designed to handle the performance challenges associated with DNNs. Some employ modified graphics processing unit (GPU) architectures for throughput. However, for DNN processing, dedicated neural processing unit (NPU) designs can offer superior performance-energy ratios.

The most important consideration is to choose hardware that offers high flexibility and not just good results for off-the-shelf benchmark DNNs such as ImageNet or MobileNet. Often preprocessing steps will need to manipulate the image data into a form that suits the application and the DNNs involved must be fine-tuned to handle the specific requirements of the application.

The ATM-monitoring example has several elements that demonstrate this need for fine-tuning and preprocessing. In its work on this application, machine-learning specialist Arcturus Networks analyzed sample image data from a banking client. The data revealed that acute camera angles common in small, enclosed ATM spaces lead to a loss of detection confidence as people move underneath the camera *(Fig. 1)*. Confidence in the results could change from more than 98% for an image where the next customer's face is clear to less than 40% if the camera sees the top of their head with little of the face.



1. Acute- or high-angle perspectives can cause a large loss in detection confidence.

The need to handle masks adds complexity. This isn't as trivial as simply training the network to recognize that there's a class of people wearing masks. The appearance of helmets or other face coverings also can be considered personal protective equipment (PPE). The network requires the ability to consider these other types of face or head coverings, with their own detection classes.

Other requirements may be placed on the system, such as being able to detect suspicious behavior (e.g., loitering) and in cases where the subject isn't always in view. They may wander in and out of the camera's field of view at different points in time. This calls for the ability to track subjects over time, rather than

2. Analysis of different types of face covering is necessary.

simply detecting mask wearers, and that the people in view are spaced adequately.

Each of these requirements call for adjustments to the model's operation as well as preprocessing steps. With live video, the assessment of whether a subject is wearing PPE is more difficult than in controlled experiments because partial occlusions and body pose will cause variability in detection results. To improve accuracy, the determination must be made using the results from multiple frames *(Fig. 2).* That, in turn, requires motion tracking to be performed on each unique person within the field of view.

In principle, motion tracking alone is an effective choice as it's relatively lightweight in terms of processing power. However, it relies on continuous detections. In a straightforward image-recognition system, occlusions, obstructions, or a person leaving and re-entering the field of view would result in the person being treated as new subjects rather than reidentifying them.

An effective approach for handling reidentification, for example, is to make use of embeddings, which are representations of the objects in the field of view that have other data encoded into them. Embeddings are commonly used in language processing. For instance, they represent words and phrases in the form of vectors, making it possible to cluster those with similar meanings in vector space.

In the case of the ATM-monitoring application, an embedding is used that not only represents visual appearance, but also information on where the object was last seen in a frame as well as what class that object was assigned. Within the bounding box used for localization, the visual appearance of the pixels inside is sampled to generate a feature vector that can be used for later comparisons.

A key advantage of embeddings is that they can be shared across multiple-camera systems, which may be employed to increase accuracy and scalability to larger spaces. The embeddings also can be used for archival searches, possibly to create active watch lists for offline analysis.

The processing overhead of tracking and reidentification does impact the processing throughput required *(Fig. 3).* In some situations, the number of people who need to be handled within a given time frame is inherently bound by the physical space available. However, with a larger field of view, computational demand may exceed a single SoC.

Flexible Architecture

What's required is a flexible architecture that can handle the different elements of a real-world machine-learning architecture. In its work, Arcturus has taken advantage of the flexible combination of processing elements in the NXP i.MX 8M Plus architecture to create an easily customizable vision pipeline. In the Arcturus approach, different stages of processing are represented by nodes. A node could be an inference model, a preprocessing or post-processing algorithm, data retrieval, or access to an external or remote service. The model is similar to the containerized approach utilized in cloud computing but adapted to the resource constraints of edge computing.

Each node is implemented as a microservice and interconnected through tightly synchronized, serialized data streams. Together, these nodes create a complete vision pipeline from image acquisition through to local actions. For basic applications, pipeline nodes can run on the same physical resource. More complex pipelines can have nodes distributed across hardware, such as the CPUs, GPUs, and NPUs in one or more i.MX 8M Plus processors, or even the cloud.

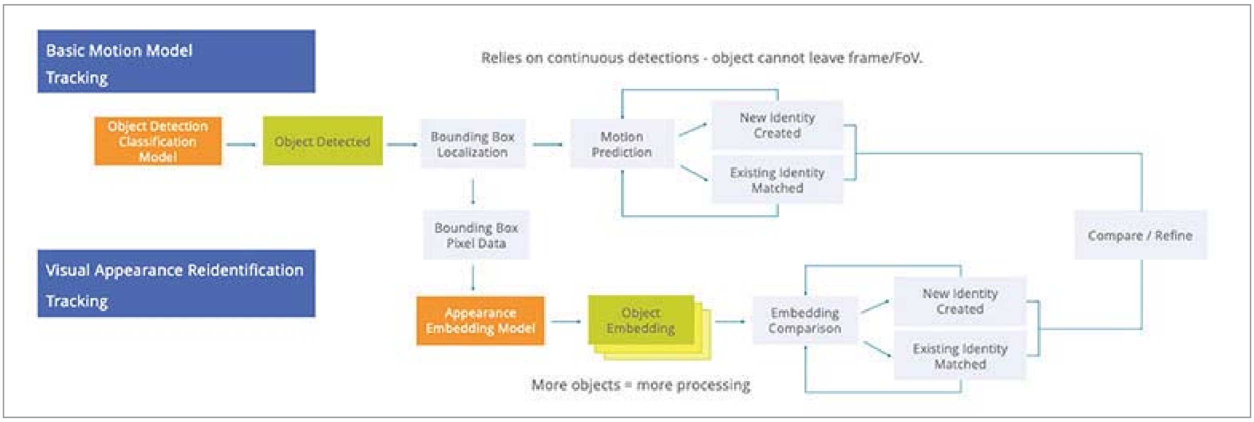
In this architecture, pipelines are orchestrated at runtime so that they can be reorganized easily as application needs change, helping to futureproof edge investment. As each node is containerized, it's simple to replace one part of the system. For example, an inference model can be updated without disrupting the rest of the system, even if model attributes change.

This pipeline architecture in the Arcturus Brinq Edge Creator SDK makes it possible to scale AI performance beyond one physical processor. An i.MX 8M Plus can generate embeddings for DNNs on one or more i.MX 8M devices that may perform detection across one or more cameras. These devices can be interconnected easily using a network fabric across one of the two dedicated Ethernet MACs on each of the processors.

As is common in machine learning, development, training, and fine-tuning can take place on a workstation, perhaps leveraging cloud-based acceleration. Once the model is trained, fine-tuned, and validated, the models will be converted for more efficient processing on NPU hardware, often by reducing the 32-bit floating-point operations needed for training to 8-bit integer calculations. Efficiency benefits further from the use of prebuilt layers and models that are optimized for the edge environment.

Arcturus provides a catalog containing prebuilt models using different precisions. These models are pre-validated to support all major edge runtimes—Arm NN, TensorFlow Lite, and TensorRT with CPU, GPU, and NPU support. Tooling is available to train or fine-tune models along with dataset curation, image scraping, and augmentation.

The results of that combination of optimized runtime, quantized model, and NPU hardware can offer a 40X performance improvement when compared with other publicly available systems running the same model. Comprehensiveness in a library like this is vital. Often, edge runtime versions don't support all



3. A comparison of the motion and visual-appearance tracking workflows.

layers required by all types of networks. Newer models that may demonstrate better performance tend to be less broadly supported than the older types used in frequently cited benchmarks.

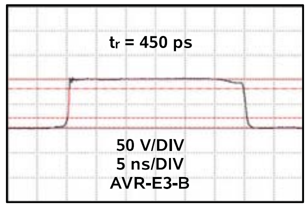
The final component is a runtime inference engine that can load the DNN model into the i.MX 8M Plus. NXP's eIQ machine-learning software development environment provides ported and validated versions of Arm NN and TensorFlow Lite inference engines.

Summary

Flexibility and performance scalability are two vital elements in real-world applications for machine learning. Each application is different and will influence not just the choice of DNN, but also the processing around it. A framework that supports this need for flexibility is vital. And it's a key reason why the combination of a microservices environment such as that developed by Arcturus and processing hardware like the NXP i.MX 8M Plus can be a powerful tool in the migration of machine learning to the edge. ed

Ad Index

This index is provided as an additional service by the publisher, who assumes no responsibility for errors or omissions.



WILLIAM WONG | Senior Content Director

GPU Delivers Ray Tracing on Mobile Devices

Impressive ray-tracing graphics are no longer limited to desktops with massive PCI Express cards.

Imagination Technologies is well-known for pushing the limits of GPUs, and its new CXT family continues this trend. One of the key features of this GPU IP is level 4 ray-tracing hardware support. At this point, ray tracing is a feature found on high-end systems that target gaming and video editing.

Ray tracing replicates real lighting and shading based on reflections and refraction of light *(Fig. 1)*. This process is data- and compute-intensive but delivers photorealistic images. It also provides visual cues we normally notice unconsciously that aren't replicating in bit-map imaging typically used to reduce processing and storage overhead.

There are five levels associated with ray-tracing hardware:

- Software using traditional GPUs
- Ray-box and ray-triangle testers
- Bounding-volume-hierarchy (BVH) processing in hardware
- BVH support plus coherency sorting in hardware
- Coherent BVH processing with BCH hardware builder

The more that can be moved into hardware with this ray-tracing level system (RTLS), the better, because it improves performance and reduces power requirements. The downside is the need for more transistors to make it work. NVIDIA's RTX 8000 with ray-tracing support is almost three years old, but that and subsequent hardware has been sitting on PCI Express cards for PCs and servers. Attaining this type of performance on mobile hardware is a bit more challenging.

The basis for Imagination's ray-tracing support is the Photon Ray Acceleration Cluster (RAC) *(Fig. 2)*. This starts with a box tester unit (BTU), a dual triangle tester unit (DTTU), and a procedural tester unit (PTU) that searches for rays intersecting with an object in 3D space. This hardware tests rays against axis-aligned boxes from the 3D scene hierarchy. It effectively implements level 2 support.

The addition of the box primitive scheduler (BPS), ray reference courier (RRC), ray store (RS), and ray task scheduler (RTS) brings the hardware up to level 3 RTLS. These dedicated blocks keep the ray date on-chip, and it handles ray traversal, tracking, and monitoring.

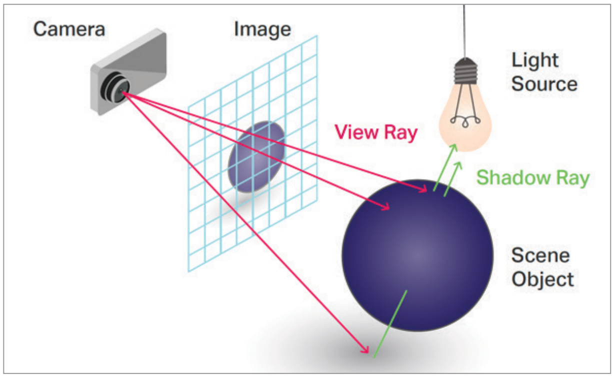
The packet coherency gather (PCG) block brings the RAC up to level 4 RTLS. It analyzes all active rays and creates groups of coherent rays. These can be tested together against the 3D scene.

A single RAC can handle 433 Mrays/s and 16 Gbox-tests/s. The hardware supports the VulkanRT ray-query and ray-pipeline interfaces. It works on all ray-traced content and enables advanced ray-tracing effects on a mobile power budget.

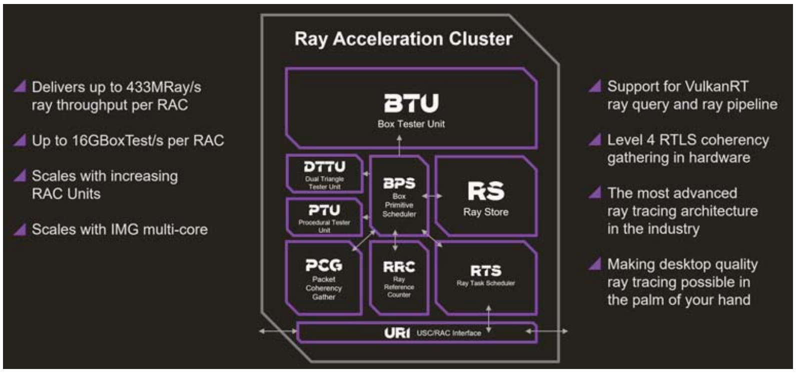
The CXT RT3 combines three Photon RAC blocks with other GPU blocks, including Tensor Processing Units (TPUs) and Unified Shading Clusters (USCs). Multiple CXT RT3s can be combined as well. This just goes to show how many different tasks use GPUs. The CXT RT3 can deliver 30 to 60 frames/s of ray-traced content at 1080p.

Expect mobile devices like smartphones to make even better use of their hi-res displays. This moves mobile gaming and user interfaces to a new level.

Imagination also provides developers with tools like PVRTune, which can examine low-level ray-tracing counters built into the hardware. ed



1. Ray tracing replicates the way light should work in a virtual world, providing a more realistic image. *(Courtesy of Imagination Technologies)*

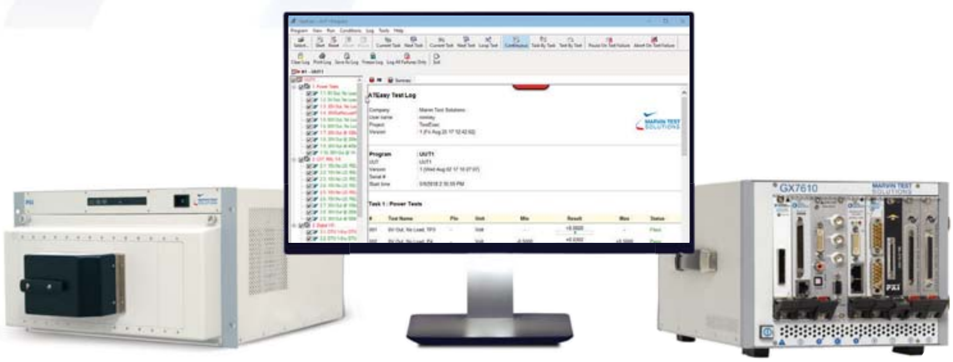


2. The Ray Acceleration Cluster (RAC) is the basis for Imagination's ray-tracing support.

Tangled up in test?

Discover how **Marvin Test Solutions'** innovative hardware and software solutions can set you free.

MarvinTest.com/WeMakeTestEasy



GENASYS	ATEasy®	Chassis | Controllers Instrumentation
High Performance Benchtop Test Platforms	Best-in-Class Test Executive and Development Studio	3U and 6U PXI/PXIe Modular Test Solutions

MARVIN TEST SOLUTIONS
We Make Test Easy™

MARVINTEST.COM