MAY 2015
VOLUME 13
#3
EMBEDDED-COMPUTING.COM

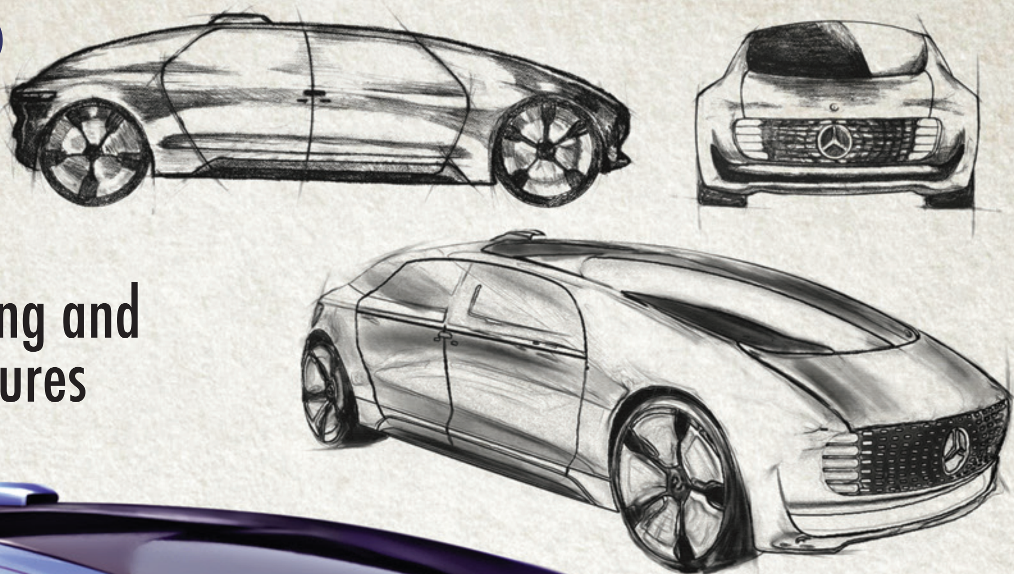
Embedded COMPUTING DESIGN

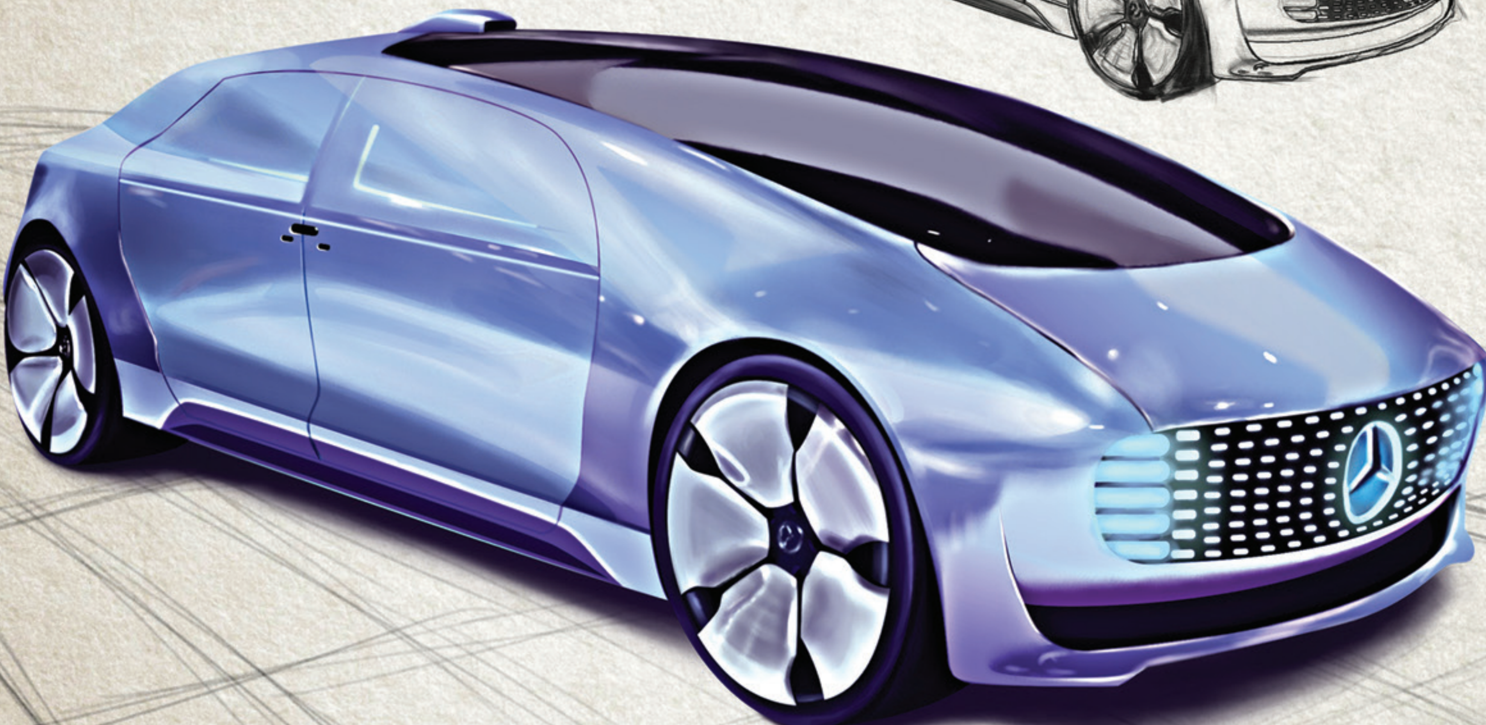
Connecting Silicon, Software, and Strategies for Intelligent Systems

AUTOMOTIVE ISSUE
PAGE 29

CONCEPTS BECOME REALITIES

Autonomous driving and advanced car features





+ PLUS

IOT INSIDER

FIVE KEYS TO SECURING THE IOT DATA PIPE
PG. 8

EMBEDDED TECHCON

PRACTICAL AND HANDS-ON
PG. 2

Embedded TechCon: Practical and hands-on

By Rich Nass, Embedded Brand Director

rnass@opensystemsmedia.com

I'm afraid it happens far too often – you're instructed to tackle a particular design or add a new feature to an existing design, and you don't even know where to begin. Some of the points may be obvious, like Googling the subject and then contacting the vendors in the space, but there's got to be a better way. Wouldn't it be great if you could get the hands-on training you need, right from the subject matter expert? Even better, what if that expert gave you a piece of hardware and some code that put you on the right path?

That scenario is exactly what you'll find at the inaugural Embedded TechCon, which takes place at the Moscone Center in San Francisco, Ca., on June 8-10, 2015. The conference is co-located with the Design Automation Conference (DAC). Embedded TechCon consists of a series of hands-on, practical classes that will provide you with the knowledge and information that you need to immediately impact your current project (and your career!).

For example, the first 50 registrants to attend the "Bring up an RTOS" class will receive a free Renesas Demonstration Kit for RX63N (valued at $100) and a one-month free access to the µC/Probe visual tool (valued at $50). Presented by Jean Labrosse, Founder/President/CEO of Micrium, this two-hour, hands-on workshop will show you how easy it is to get a real-time multitasking application up and running.

Are you designing a platform that may end up in the automotive sector? Because wireless car connectivity, or V2X, presents compelling capabilities for next-generation automobiles, including improved collision avoidance, autonomous operation, predictive maintenance and lower total cost of ownership, improved driver and passenger experiences, and more. David Kleidermacher, Chief Security Officer for BlackBerry, will dive into this topic in detail. Not to scare you, but security researchers have demonstrated that attack vectors across these extravehicular networks pose a clear and present danger to the same safety and productivity functions they seek to foster. As they say, "don't let this happen to you."

In the hands-on class "Rethinking Embedded Linux," you will learn how to adapt some workstation-tested strategies for the embedded space. Bill Gatliff, who's been training embedded developers for decades, will teach attendees how a lightly-modified, Debian-based Embedded Linux operating system can help keep you focused on your application development while it takes care of things like file system bootstrapping, configuration management, system state and event management, developer workflow, and maintenance upgrades. The bottom line is, stop wasting time rebuilding your embedded Linux files system from source code.

Did you need to add Bluetooth connectivity? Embedded TechCon has you covered. One of the industry leaders, Anaren, will provide you with a Bluetooth Smart development kit and the instruction required to get your project out the door.

FPGAs are a very popular embedded architecture. That's stating the obvious. But they can be intimidating to some. Hence, we've put together a class of best practices that'll walk you from initial concept through the design tradeoffs. Specially, our expert instructors will point out the common design challenges, risks, and pain points, showing their impacts and how to manage each condition. And we can ensure that your design will stand the test of time by projecting design obsolescence, staying ahead of device and tool obsolescence issues, and the things you need to consider when migrating from family to family or tool to tool.

Obviously, there's too much to cover here, but we've got just about every aspect of embedded development covered at Embedded TechCon. And remember, you'll leave with knowledge, hardware, and software: hands-on and practical, not theoretical.

Register:
opsy.st/EmbeddedTechConRegistration

For more on classes, speakers, and giveaways visit embeddedtechcon.com

Embedded TechCon

SCHEDULE

Day 1 – Monday, June 8

		Room
9:00-10:00	**DAC Keynote Address:** Google Smart Lens: IC design and beyond, with Brian Otis, Co-founder of the Google Smart Lens project	Gateway Ballroom
10:15-11:15	The Intel IoT Platform, taught by Kevin Williams and Victor Webb, Solutions Architects, Intel	307
11:30-12:30	Remote management, configuration, and diagnostics of embedded systems and devices, taught by Matthias Huber, Vice President, ADLINK	307
2:00-3:00	How to secure the Industrial Internet of Things, taught by Alan Grau, President/Co-founder, Icon Labs	307
3:15-5:30	Rethinking Embedded Linux, taught by Bill Gatliff, Expert Consultant	307

Day 2 – Tuesday, June 9

		Room
9:00-10:00	**DAC Keynote Address:** The critical role of design automation in automotive electronics, with Jeffrey Owens, Chief Technology Officer for Delphi Automotive	Gateway Ballroom
10:15-11:15	Use Buetooth Smart to control your embedded device with a mobile device, taught by Vijay Parmar, Business Development Manager, Anaren	307
10:15-11:15	Synchronizing mechatronic systems in real-time using FPGAs and Industrial Ethernet, taught by Sari Germanos, Ethernet Powerlink Standardization Group	308
11:30-12:30	**Panel Session**: Identify and secure your IoT weak links, moderated by Patrick Mannion, Founder/Managing Director, ClariTek **Panelists:** David Kleidermacher, BlackBerry; Skip Ashton, Silicon Laboratories; Gary Davis, McAfee; and Dr. Jakob Engblom, Wind River	307
11:30-12:30	Connecting your IoT devices to the infrastructure, taught by Philippe Chevalier, Director, Technology Platforms, Kontron	308
2:00-3:00	Understanding and dealing with security in the automotive sector, taught by David Kleidermacher, Senior Vice President, Head of Product Security, BlackBerry	307
2:00-3:00	Build vs. buy when it comes to IoT platforms, taught by Stefan Milnor, Technical Fellow, Kontron	308
3:15-5:30	**Hands-on Lab:** Bring up an RTOS, taught by Jean Labrosse, Founder, President, CEO, Micrium	307
3:15-5:30	Best practices for efficient and effective FPGA design, taught by RC Cofer, Field Applications Engineer, Avnet	308

Day 3 – Wednesday, June 10

		Room
9:00-10:00	**Keynote Address:** Cyber threats to connected cars: Staying safe requires more than following the rules of the road, with Jeff Massimilla, Chief Product Cybersecurity Officer, GM, and Craig Smith, OpenGarages.org	Gateway Ballroom
10:15-11:15	Making continuous integration a reality using simulation, taught by Dr. Jakob Engblom, Product Line Manager, Wind River	307
10:15-11:15	Making sense of sensors for IoT applications, taught by Prem Kumar, VP, Technology Platforms, Kontron	308
11:30-12:30	Finding low-level security vulnerabilities with static analysis, taught by Paul Anderson, VP of Engineering, GrammaTech	307
11:30-12:30	How to keep your IoT platform secure, taught by Eric Sivertson, EVP, Head of Avionics, Trans., and Defense BU, Kontron	308
2:00-3:00	A survey of performance, memory, and power optimization techniques for embedded system software, taught by Rob Oshana, Director of Global Software R&D and Enablement, Freescale Semiconductor	307
2:00-3:00	Developing IPs and Subsystems for automotive infotainment and ADAS applications, taught by Charles Qi, System Solutions Architect, Cadence	308
3:15-5:30	The era of machines that see: Opportunities and challenges in embedded vision, taught by Jeff Bier, President, BDTI	307

MAY 2015
VOLUME 13 #3

embedded-computing.com | @embedded_comp | in opsy.st/ECDLinkedIn

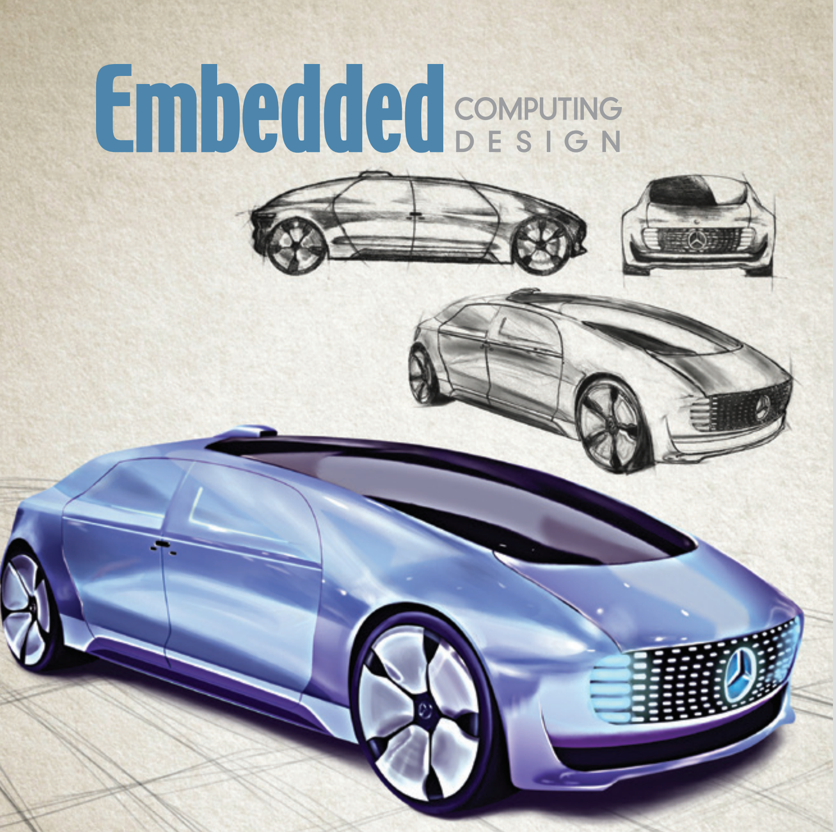


Illustration by Chris Rassiccia based on the Mercedes-Benz F 015 concept car at CES 2015.

Departments

Special Features

APP EXTRAS

Download the Embedded Computing Design app:
iTunes: itun.es/iS67MQ
Kindle Fire: opsy.st/kindlefireamaz

Silicon

Software

Strategies

ADVERTISER INDEX

Get your free digital edition at embedded-computing.com/emag

Subscriptions
embedded-computing.com/subscribe
subscriptions@opensystemsmedia.com
opensystemsmedia.com/subscriptions

ISSN: Print 1542-6408, Online: 1542-6459

ENVIROINK
The inks used to print the body of this publication contain a minimum of 20%, by weight, renewable resources.

Embedded COMPUTING DESIGN

OpenSystems media.

ECD Editorial/Creative Staff

Rich Nass, Brand Director
rnass@opensystemsmedia.com

Curt Schwaderer, Editorial Director
cschwaderer@opensystemsmedia.com

Monique DeVoe, Managing Editor
mdevoe@opensystemsmedia.com

Brandon Lewis, Assistant Managing Editor
blewis@opensystemsmedia.com

Rory Dear, Technical Contributor
rdear@opensystemsmedia.com

David Diomede, Creative Services Director
ddiomede@opensystemsmedia.com

Konrad Witte, Senior Web Developer
kwitte@opensystemsmedia.com

Chris Rassiccia, Graphic Designer
crassiccia@opensystemsmedia.com

Sales Group

Tom Varcie, Sales Manager
tvarcie@opensystemsmedia.com
(586) 415-6500

Rebecca Barker, Strategic Account Manager
rbarker@opensystemsmedia.com
(281) 724-8021

Eric Henry, Strategic Account Manager
ehenry@opensystemsmedia.com
(541) 760-5361

Kathleen Wackowski, Strategic Account Manager
kwackowski@opensystemsmedia.com
(978) 888-7367

Asia-Pacific Sales
Elvi Lee, Account Manager
elvi@aceforum.com.tw

Regional Sales Managers
Barbara Quinlan, Southwest
bquinlan@opensystemsmedia.com
(480) 236-8818

Denis Seger, Southern California
dseger@opensystemsmedia.com
(760) 518-5222

Sydele Starr, Northern California
sstarr@opensystemsmedia.com
(775) 299-4148

Reprints and PDFs

republish@opensystemsmedia.com

EMEA

Rory Dear, Technical Contributor
rdear@opensystemsmedia.com

James Rhoades-Brown – Europe
james.rhoadesbrown@husonmedia.com

Christian Hoelscher, Account Manager – Europe
christian.hoelscher@husonmedia.com

Gerry Rhoades-Brown, Account Manager – Europe
gerry.rhoadesbrown@husonmedia.com

OpenSystems Media Editorial/Creative Staff

Embedded COMPUTING DESIGN | Military EMBEDDED SYSTEMS | SIGNAL PROCESSING DESIGN | INDUSTRIAL EMBEDDED SYSTEMS | PC/104 small form factors | PICMG SYSTEMS & TECHNOLOGY | VITA TECHNOLOGIES

John McHale, Group Editorial Director
Military Embedded Systems
PC/104 and Small Form Factors
PICMG Systems & Technology
VITA Technologies
Signal Processing Design
jmchale@opensystemsmedia.com

Joe Pavlat, Editorial Director
PICMG Systems & Technology
jpavlat@opensystemsmedia.com

Jerry Gipper, Editorial Director
VITA Technologies
jgipper@opensystemsmedia.com

Steph Sweet, Creative Director

Joann Toth, Senior Designer

Lisa Daigle, Assistant Managing Editor
Military Embedded Systems
PC/104 and Small Form Factors
ldaigle@opensystemsmedia.com

Sally Cole, Senior Editor
Military Embedded Systems
scole@opensystemsmedia.com

Brandon Lewis, Assistant Managing Editor
Industrial Embedded Systems
PICMG Systems & Technology
Signal Processing Design
blewis@opensystemsmedia.com

Jennifer Hesse, Managing Editor
VITA Technologies

Joy Gilmore, E-cast Manager
jgilmore@opensystemsmedia.com

Mariana Iriarte, Associate Editor
Military Embedded Systems
miriarte@opensystemsmedia.com

Corporate

opensystemsmedia.com

Patrick Hopper, Publisher
phopper@opensystemsmedia.com

Rosemary Kristoff, President
rkristoff@opensystemsmedia.com

John McHale, Executive Vice President
jmchale@opensystemsmedia.com

Rich Nass, Executive Vice President
rnass@opensystemsmedia.com

Wayne Kristoff, CTO

Emily Verhoeks, Financial Assistant

Headquarters – ARIZONA:
16626 E. Avenue of the Fountains, Ste. 201
Fountain Hills, AZ 85268
Tel: (480) 967-5581

MICHIGAN:
30233 Jefferson, St. Clair Shores, MI 48082
Tel: (586) 415-6500

Driving cars and their wireless communication forward

By Rory Dear, Technical Contributor

rdear@opensystemsmedia.com

The self-driving movement is gaining traction in the UK due to relaxed regulations, and so is the wireless technology that is behind this revolution.

Whilst the self-driving vehicle movement picks up speed in many places, the pace of implementation of this exciting revolution in the U.S. is hampered by strangling regulation and red tape. Laws governing the opening up of roads in the U.S. to self-drive testing are determined at state rather than federal level, so a patchwork of ambiguous and contradictory rules exist either side of state lines. The public naturally remains nervous in handing over control of their steering wheels to a "computer," and necessary long-distance testing is restricted by the state of U.S. regulations.

Conversely, the UK has taken the opposite approach. By this I do not mean any regulation at all as the automobile industry will always be life critical; our Department of Transport puts it beautifully, "The aim is to achieve a light-touch, non-regulatory approach which provides the clarity industry needs to invest in further research and development while maintaining safety". Major car manufacturers invested in self-driving are now flocking to our shores and new players are getting in on the scene like Google, or even Apple.

With Embedded World 2015 a recent memory, this got me thinking about how the wireless protocols on show fit within the vehicular space, in fact some new protocols are even designed exclusively for this upcoming metamorphosis of our daily vehicles.

Intra-vehicle

The need for intra-vehicle (in-vehicle) communication is born primarily from the expense and complexity of the current wiring loom – in a typical car this equates to around 5 miles of wiring! For vehicle manufacturers, specifying, purchasing and installing this vast array of copper massively impacts build times and, of course, cost. For car owners that have had suspected issues with their vehicle's wiring loom, they'll also know this complexity corresponds to titanic repair bills!

Tomorrow's vehicles will do away with this predominantly CAN bus driven wiring loom, abolishing the "fly-by-wire" system popularized by aircraft and once the height of technology with a "fly-by-wireless" – a term I hoped I'd just coined but it transpires not…

Protocols you're probably already familiar with are fighting for implementation within this sub-application, including IEEE 802.15.1 (Bluetooth), IEEE 802.15.4 (ZigBee), and IEEE 802.15.3 (UWB) – with varying ranges, bandwidths, and associated costs. As the land lies today UWB stands most cost-effective at $1 per chip, with ZigBee and Bluetooth at $2 and $5 respectively.

This would suggest that UWB is being deployed en-masse, but actually it appears Bluetooth, as the proven technology with voice and data transfer capability, will be the first to appear. Expect to see hybrid systems with localized wired clusters first emerge. There are also some fundamental concerns relating to wireless interference that must be satisfied before this is truly implemented – when controlling vehicle functions this is inherently more safety critical than the following two categories.

Inter-vehicle (V2V)

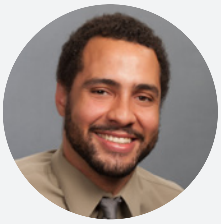
Scenarios benefiting from V2V communication range from peer-to-peer arrangement of safe distances in motorway convoys to warning vehicles of a road traffic collision up ahead. Today V2V communication is restricted to warning lamps of potential danger, but tomorrow's self-driving vehicles must also take action.

A seamless combination of Wi-Fi (802.11P/AC), DSRC, and LTE are likely to be combined to produce the reliability and information integrity that is required to trust onboard computers to prevent collisions in fast moving traffic.

Naturally given the opportunity for a malicious party to cause such catastrophe by injecting rogue data, it's critical that a combination of protocols, rather than purely the vehicles own P2P mesh, provide that redundancy to cross reference reported information – and all in fractions of a second else it's too late!

Vehicle to infrastructure (V2I)

Whilst vehicles' satellite navigation systems are increasingly supporting 3G/LTE access to provide traffic information, the connected infrastructure revolution connecting vehicle to roadside will increase bandwidths substantially using the 63 GHz band, enabling passengers in self-driven vehicles to work remotely, browse the web and video conference whilst on the move even in heavy traffic. The bottleneck of stopping for toll booths will also become a thing of the past with V2I allowing that transaction electronically at motorway cruising speed. ECD



Five keys to securing the IIoT data pipe

By Brandon Lewis, Assistant Managing Editor *blewis@opensystemsmedia.com*

Industry is realizing the missed opportunities that have resulted from squandered connections between machine data and analytics in those conventional configurations, and have begun pushing for designs that leverage the Internet and the cloud more heavily. Obviously, opening these networks to the broader Internet comes with inherent security risks, and further complicates the ongoing tradeoff developers must make between performance, reliability, and security.

In response, companies are turning to the data distribution service (DDS), a middleware standard governed by the Object Management Group (OMG) that provides a scalable, secure data pipe for Industrial Internet systems. While the DDS standard itself was originally developed by Real-Time Innovations (RTI) and Thales Group in 2001 and approved by the OMG in 2003, vendors of the middleware recently realized the need for more comprehensive definitions of the security model and service plugin interface (SPI) architecture in the standard, and subsequently began work on the DDS Security Specification (DDS-SECURITY) in 2014. After successfully demonstrating interoperability of DDS-SECURITY at OMG meetings last month, David Barnett, VP of Products and Markets at RTI, agreed to walk me through the five key tenets of the spec.

1. **Authentication –** In the DDS-SECURITY spec, authentication is used to verify that every device or user participating in a distributed system is who he, she, or it attests to be. The DDS Authentication SPI provides faculties for performing mutual authentication between participants so that "shared secrets" can be established, and sets the table for Access Control.
2. **Access Control –** Access Control may be the most critical security component of Industrial Internet/IIoT systems, as it defines who can publish or subscribe to a certain type of data or metadata across a DDS network. DDS provides very fine-grained access control, and includes a "discovery" feature that enables applications to be written in such a way that when new sensors are added to a network they can automatically be identified without any additional coding required. The Access Control SPI also gives developers control over the kinds of data/metadata those applications have visibility into, which combined with discovery is key to building and deploying scalable distributed networks.
3. **Cryptographic –** The Cryptographic SPI, in other words encryption, ensures that data moving across DDS networks in private. However, unlike typical forms of encryption, the elegance of cryptography in DDS is that it allows developers to choose what data needs to be encrypted and what doesn't. The ability to select only certain data for encryption is important for the resource and bandwidth constraints of industrial systems and networks. For those messages that are not encrypted, the DDS Security Specification also includes provisions for secure digital signatures so that senders can be authenticated without the overhead of encryption itself, which helps protect systems that may be vulnerable to spoofing or man-in-the-middle attacks.
4. **Logging –** Logging supports auditing of all security-related events on a distributed network. This gives administrators the ability to monitor any attempts to break into the system, including instances when the content of a message doesn't match its signature, usually an evidence of a man-in-the-middle attack.
5. **Tagging –** Finally, tagging is a feature that allows developers to inject metadata into a message that designates the content's security level. As an extension of access control, this makes it possible for sensor data from a wind turbine, for example, to be classified by "confidential" or "unrestricted" so that the appropriate data can be made available only to specific participants in large distributed networks.

RTI provides these DDS-SECURITY SPIs in their out-of-the-box Connext product that uses standard algorithms for encryption and public key infrastructure (PKI) for authentication, and also implements a plug-in approach for organizations that already have already policies and practices in place, or that need to meet the certification requirements of a particular industry. The flexibility of the DDS domain-specific modeling paradigm is one of the elements of the middleware that set it apart from other low-level messaging standards such as MQTT and CoAP, and a major enabler of the five keys to a secure IIoT data pipe.

OMG members expect to approve the DDS-SECURITY specification by the end of the year. ECD

Autonomous driving: In research and on the road

By Monique DeVoe, Managing Editor
mdevoe@opensystemsmedia.com

The automotive industry is inching closer and closer to getting autonomous driving on the road. There have been impressive demonstrations, such as Delphi's Roadrunner car that completed a 3,400-mile autonomous test-drive in April, and Tesla Motors will soon have autonomous driving features in cars that are actually out on the road today. It's becoming clearer that these types of autonomous driving models are where cars are heading.

A team of researchers at the University of Waterloo (uwaterloo.ca) in Ontario, Canada led by Steve Waslander, Assistant Professor, Department of Mechanical and Mechatronics Engineering, and Michael Worry, CEO and Founder at electronic product design and manufacturing company Nuvation Engineering (www.nuvation.com) are three years into a four-year partnership of working on autonomous vehicle development. Funding from the Canadian federal and provincial government levels allows for five faculty and five or six students working at the same time on different autonomous driving projects. Through this partnership, Nuvation and the University of Waterloo bring together the perspectives of what's possible in function and what's viable in business.

"Steve and his team work on what I call the string theory of autonomous driving – how do we figure out a design that can handle every situation regardless of weather or conditions that come up on the road or anything else?" Worry says. "Whereas myself, as an entrepreneur and a capitalist, I look at it from a perspective of what's the minimum viable product? What can be done here that can be carved off that can cover enough situations that it's a product that someone will pay money for? And somewhere in between those two perspectives represents this project work that we do."

Instead of focusing on building features already in OEMs' pipelines, Waslander says they're working on what can be available in vehicles in a few years.

"We're looking at the collaborative driving future where we have multiple autonomous vehicles communicating with each other," Waslander says. "We're trying to simultaneously exploit the fact that they can share information and they can autonomously make decisions and sense things around them."

The research is split into four topic areas – two focus on multi-vehicle coordination and two focus on perception systems, like those seen in the Google autonomous cars and the Mercedes Benz prototypes. Autonomous platooning involves teams of vehicles coordinating on the highway telling each other how fast they want to go, which lanes they want to travel in, how far they want to get along the highway, and they work together to maximize efficiency and driver comfort. Evasive maneuvers or emergency maneuvering is the other part of multi-vehicle coordination that allows the platoon on the highway to rapidly communicate emergency situations through the chain of vehicles behind them and take appropriate actions, such as slamming on the brakes, switching lanes, or moving off to the shoulder. One perception topic involves generalizing vision-based algorithms for lane detection to be used in every condition a vehicle might encounter, such as intersection detection, merging lane detection, and any other markings on the road. The cars need to figure out how to get from the entry point of the intersection to the correct lane when exiting the intersection. Starting this year the team is researching using computer vision and stereo vision in particular to do robust tracking of other vehicles, pedestrians, and bicycles in all sorts of weather systems.

Those projects Waslander's team are working on make up the research sphere, where researchers are able to demonstrate a feature can be done. Worry is concerned with the production sphere, where companies can demonstrate they're willing to sell the feature and guarantee it'll work. There's still a huge divide between the two and a lot of work left to do, but Waslander and Worry are working to solve those challenges. ECD

Steven Waslander, Michael Worry, and I talked further about the various challenges researchers and production are facing now and will face in the future, as well as some applications of autonomous driving that are closer to real-world deployment. Read more at opsy.st/UWaterlooNuvation.

DIY meets autonomous driving

By Monique DeVoe, Managing Editor mdevoe@opensystemsmedia.com

When you get a team of professional engineers in a group doing projects for fun, you can get some pretty awesome making going on. Michael Worry at Nuvation Engineering (www.nuvation.com) clued me in on a pretty unique DIY project in the realm of autonomous driving.

After-hours at Nuvation, engineers take on exciting projects for fun at a program called Nuvation Garage.

"During the day we do engineering because we're paid for it and after-hours we do engineering for the love of the game," Worry says.

One Nuvation Garage project they dubbed "DiscoFish." It's a 65-foot long, 38,000-pound "mobile dance party" vehicle in the shape of an anglerfish that the Nuvation team takes to the art festival and experimental community event Burning Man in the Black Rock Desert in northern Nevada. Other features include a pro audio sound system, 3,000 individually addressable light-up LED scales, a dance stage, a 20-foot flamethrower, and an 11 kW searchlight visible from four miles away.

"We have office parties on it, and we have a lot of fun with it," Worry says (I think this is an understatement).

What's more, it's an autonomous vehicle.

"One of the things we wanted to do very early on was to use it as a platform in order to advance autonomous research, because I really wanted to be able to have a beer and have an art car at Burning Man," Worry says. If you aren't familiar with Burning Man art cars, do a Google image search and you'll find some other near large-scale maker projects – it's better to see them than to read about them.

Worry leveraged Steve Waslander, Assistant Professor, Department of Mechanical and Mechatronics Engineering at the University of Waterloo, with whom Nuvation has partnered for research and development of collaborative autonomous vehicle systems, to develop DiscoFish's autonomous driving capabilities.

It has an Ubuntu Linux computer running Robot Operating System (ROS), motors on the gas, brake, and steering wheel of the vehicle, LIDAR in front and a GPS unit on the roof. These systems are hooked up to two screens accessible to the driver; one displays the forward-looking LIDAR data being used for obstacle avoidance, and one shows the GPS-managed waypoint route that DiscoFish is following. "You can reach above your head and tap on the screen, 'I want to go here, here, and here' and click 'GO' and the vehicle starts turning and peels off and follows the route," Worry says. "It's super fun!"

Nuvation Garage's DiscoFish. Image courtesy of Nuvation Engineering. All rights reserved.

The environment at Burning Man includes thousands of people moving about on foot and on bicycles, plus other art cars. It provides excellent examples of some of the more challenging and unpredictable conditions in which an autonomous vehicle may need to maneuver. Worry says it's taught them a lot about how people might use autonomous driving capabilities.

"I found there were times where I wanted to manually control the gas and brake while keeping the steering autonomous so DiscoFish would keep following the pre-programmed route," Worry says. "There were other times where I could see a mild obstacle that I wanted to steer around, but I wanted it to not fall out of its navigation, so I would briefly take it out of autonomous steering and nudge around an obstacle, then let it continue on its way. So we started to discover all these different use cases that were really interesting."

It's not your usual maker project, but it sounds very cool to play with and at the same time it's advancing research in the exciting area of autonomous driving. Keep on making with a purpose, Nuvation Garage! ECD

See DiscoFish in action (opsy.st/DiscoFishInAction) and more technical details about the system's design (opsy.st/DiscoFishTechSpecs).

Multichip package memory enabling next-generation Internet of Things connectivity in automotive

By James Malatesta

The Internet of Things (IoT) has created new requirements for memory in space- and resource-constrained applications that are adding connectivity where little to none existed before. In particular, the advent of the IoT is driving the use of multichip package (MCP) memory solutions in verticals such as automotive, where the ability to stack non-volatile memory (NVM) and volatile memory (RAM) into a single package is helping reduce footprint, lower ball count, improve performance, and ease design implementation for a range of applications in the connected car.

With the development of Internet of Things (IoT) from industrial to consumer and the resulting explosion of new embedded devices, multichip packages (MCPs) are rapidly evolving to meet the needs of this market. Defined as the ability of machines to connect together and communicate through various wireless protocols and the Internet, the IoT drives a multitude of usage applications such as home security, wearables, smart metering, automotive infotainment, and other space-constrained applications where MCPs are well aligned.

Initially introduced and optimized at the dawn of the cellular phone era, MCPs offer many advantages compared to external individual memory types. Memory MCPs stack non-volatile memory (NVM) (which delivers boot-up/application, operating system, and other critical code/data execution) and volatile memory (RAM) (which serves as high-speed temporary memory) together in one package (Figure 1). In addition to reducing area footprint, providing a lower ball count and increasing performance and density, MCPs ease design considerations by offloading the embedded memory of a microcontroller (MCU) using industry standard JEDEC interfaces and memory types. All of these criteria are driving the adoption of MCPs in IoT applications such as automotive infotainment and active safety systems, which we will examine in more detail.

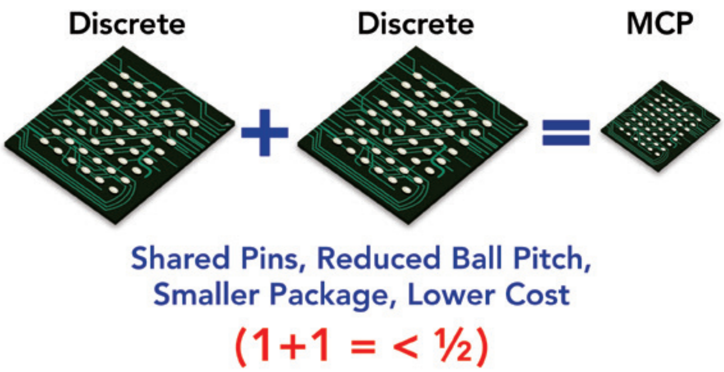


Figure 1 Multichip packages integrate non-volatile memory (NVM) and volatile memory (RAM) in a single package that reduces area footprint and ball count while increasing performance and density.

Automotive memory requirements

IoT is now reaching into automobiles, with the integration of smartphone-like functionality that enables users to stream music, make restaurant reservations, and purchase movie tickets all from the comfort of the driver's seat. As a result, in-vehicle memory requirements are also beginning to grow, with a large contributor to memory growth in the connected car being the communications modules contained in infotainment systems that allow users to be connected at all times, even when on the go.

However, driving high temperature and quality standards in today's memory solutions is key. Connected car applications require specific memory solutions due to the stringent quality, reliability, and operating temperature requirements of the automotive market, as the safety-critical nature of these systems leaves minimal margin for error. The automotive market's unique set of feature requirements include:

- **Zero defect approach –** Targeting no failures over the product lifecycle
- **Continuous improvement process –** Persistent focus on improving the overall quality of products, including legacy products
- **Automotive-grade selection –** Strict selection criteria in fabrication, assembly, and test to ensure the highest quality product
- **Long lifecycle –** 10 to 15 year product availability and support
- **Burn-in flow –** Simulating the first year of product life to improve overall quality, which statistically is when marginal products fail
- **Automotive certification of fab and assembly sites –** Fab and assembly certification to ISO/TS 16949
- **AEC-Q100 –** A failure mechanism-based stress test qualification for integrated circuits

Automotive documentation such as:

- **Production part approval process (PPAP) documentation –** Additional documentation stating where die is fabricated, parts are assembled, and testing is conducted so there is a formal return merchandise authorization (RMA) trail
- **8D failure mode and effects analysis (FMEA) support –** In-depth support with guaranteed timelines and clear steps for improvement

In addition to meeting these requirements, from a system designer's perspective, the IoT implementation in automotive applications necessitates unique memory configurations with the right mixture of density, power, size, performance, temperature, reliability, cost, and support. With the ability to support the increasing memory needs of applications such as in-vehicle infotainment and active safety systems with more memory density, MCPs are able to fill the gap of using two discrete components while greatly reducing area.

MCPs in the connected car

To meet the memory density and small size needs of next-generation automotive applications, automakers are using MCPs with proven process nodes and design stability.

Infotainment

Infotainment systems have expanded to include managing and playing audio and video content, utilizing navigation for driving; delivering rear-seat entertainment such as movies, games, social networking, etc.; listening to incoming and sending outgoing SMS text messages; making phone calls; and accessing Internet-enabled or smartphone-enabled content such as traffic conditions, sports scores, and weather forecasts. Future systems look to make infotainment systems the center point of the vehicle with additional features like 3D maps.

These infotainment systems need a connection to the Internet, which is provided by 2.5G, 3G, and 4G connectivity modules that are located in a space-constrained area of the automobile, with small, thin modules that are sometimes the size of a postage stamp. NAND MCPs utilize densities ranging from 1 Gb to 8 Gb to meet the different memory requirements of the various infotainment systems. In particular, SLC NAND Flash provides better data retention and program/erase (P/E) cycles over temperature as compared with other forms of NAND Flash to enable NAND MCPs that meet the needs of very harsh reliability and temperature environments.

Active safety and driver assistance

Another fast growing automotive sub-segment is advanced driver assistance systems (ADAS). These systems are moving from detection-only, backup sonar beeping to providing assistance to the driver that helps keep vehicle occupants safe. As

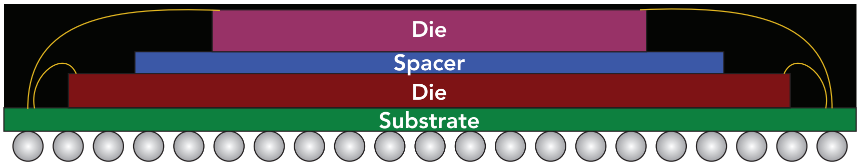


Figure 2 Multichip package (MCP) designs can comprise NAND Flash, NOR Flash, or DRAM memory depending on the application.

the complexity of these systems grows so do the memory requirements, which is further complicated by ADAS modules (like the connectivity modules) having to fit into space-constrained areas of the vehicle.

In order for the electronics in the car to keep up with the functionality being added to ADAS for tasks such as collision avoidance, lane detection, obstruction recognition, and other critical functions, manufactures are looking to MCP memory solutions that meet the needs of these systems. For critically fast response time, many existing sensor modules that employ high-performance and high-density LPDDR1 are moving to more centralized NOR + LPDDR2/LPDDR4 solutions as complexity, intelligence, and density requirements increase. Traditional parallel NOR MCPs utilize densities ranging from 32 Mb to 512 Mb + PSRAM (pseudo SRAM) or LPDDR1, while also offering high-performance synchronous burst mode to deliver fast boot up/application, operating system, and other critical code/data execution through an execute-in-place (XiP) system design that allows burst read speeds from 133 MBps to 266 MBps.

The other memory within MCPs is LPDRAM (Figure 2). LPDRAM in MCPs utilizes densities ranging from 512 Mb to 8 Gb on the LPDDR1 and LPDDR2 side in order to boost the LPDRAM performance in MCPs from 800 MBps for x16 bus width LPDDR1 clocked at 200 MHz to more than 4 GBps for the faster x32 bus width LPDDR2 clocked at 533 MHz. All of this performance is packed into the MCP with high-frequency signal integrity using the traditional multibus interface MCP design.

A bright outlook for MCPs

While MCPs have been below the radar for some time and mainly used for cellular phones, their evolution in features and functionality has set the stage for many new IoT applications within the embedded market, especially in the automotive industry. Micron's automotive MCP offerings, available in several densities of NAND+LPDDR or NOR+RAM, meet the low defects per million (DPM) and longevity requirements of the automotive industry without altering board or module requirements.

While existing MCP memory solutions already meet the demands of tomorrow's automotive memory subsystems, they continue to increase in performance and density while reducing size and power consumption per bit to pave the way for not only next-generation infotainment and ADAS requirements, but also vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) automotive systems. There's little need to design new technologies or enable complete ecosystems that satisfy IoT and automotive memory requirements because they exist now and are moving fast down the evolutionary highway. MCPs place system designers in the fast lane with easy-to-integrate solutions for applications such as state-of-the-art ADAS and infotainment systems that require high-performance and high-density memory in a small package.

With over 20 years in the electronics industry, James Malatesta has held technical and marketing positions at Intel, Numonyx, and Micron. His primary focus has been application engineering, new technology enabling, and product/technical marketing within the memory technology segment.

Micron Technology, Inc.
www.micron.com @MicronTech opsy.st/MicronFB
linkedin.com/company/micron-technology plus.google.com/explore/Micron

Auto industry highlights: Innovation in green, smart, and safe

By Curt Schwaderer, Editorial Director

cschwaderer@opensystemsmedia.com

The auto's journey to smarten up can be seen in the vision, challenges, and solutions of advanced driver assistance systems (ADAS).

Advancements in the auto industry are always a popular topic of conversation. From fuel efficiency to infotainment, cars are seen as transportation, entertainment, and status symbols all over the world. Embedded technologies are driving key initiatives in the industry:

- **Green –** Environmentally friendly and fuel efficient
- **Smart –** Greater visibility, alters using sensor, actuator, and camera technologies
- **Safe –** Collision safety features to reduce injury and road fatalities

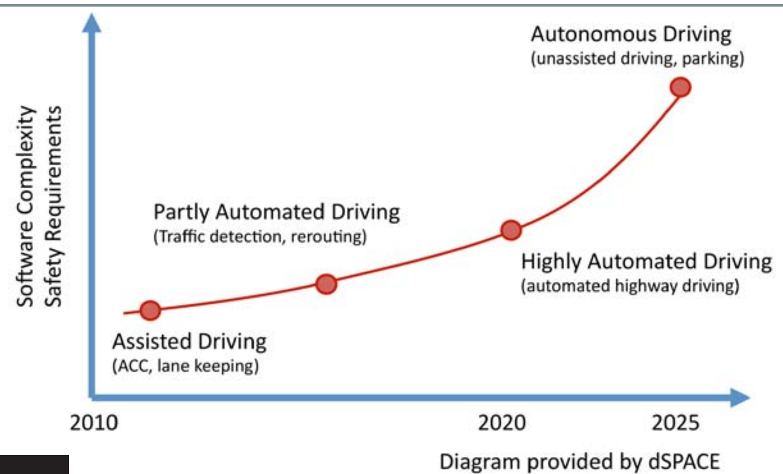


Figure 1 Autonomous driving vision.

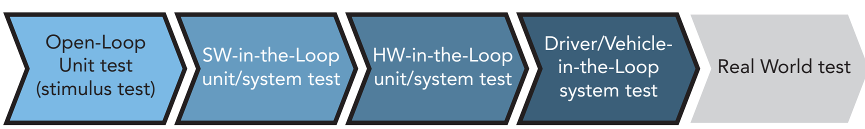


Figure 2 Testing ADAS and active safety systems.

Focusing on the "smart" initiative, advanced driver assistance systems (ADAS) are expected to advance from assisted driving today to highly automated driving in 2020 to autonomous driving by the year 2025 (Figure 1). If this vision is realized, in ten short years, autonomous driving will be a reality – though Tesla claims it'll have autonomous driving in its cars by summer 2015!

ADAS are the stepping stones toward autonomous driving. An interconnected network of cameras, video analysis, sensors, and global positioning systems provide the building blocks upon which the auto of the future will be built.

Automotive standards

The European New Car Assessment Programme (Euro NCAP) is a driving force in the assessment and advancement of these key initiatives in the auto industry. Euro NCAP performs crash tests as well as test protocols for validating active safety systems through simulation. These test simulations are used to assess the five-star rating system assigned to various makes and models.

Environmental regulations are becoming tougher and higher safety standards are being applied. In order to address these challenges, auto developers and their suppliers need access to the right embedded tools and simulation environments in order to achieve the desired ratings.

Simulation to real-world test

I got the opportunity to talk with Mahendra Muli, Director of Marketing and New Business Development at dSPACE, about Euro NCAP and the challenges involved with the testing required to ensure functional safety.

Mahendra describes ADAS and active safety systems development as a pipelined process shown in Figure 2.

Open-loop testing provides an environment where engine controller unit (ECU) algorithms can be developed and validated within a realistic context. Closed loop simulation is used in the early development stages to provide a higher quality production candidate. Simulation should also be able to occur in real-time or faster than real-time.

These models can then be used to perform early integration testing. Integration tests can be run with virtual ECUs at this point and the tests can be prepared and validated for use within the hardware-in-the-loop (HIL) testing.

Once the software-in-the-loop (SIL) testing is complete, the same tools, models, layouts, and tests can be utilized within the HIL testing. At this point the ECU tests can also be automated.

The process and simulation environment provides for testing of actuators, radar sensors, and camera sensors involved in things like lane departure warning, emergency braking, and pedestrian detection. Mahenda cited test cases dSPACE has been involved in using their ControlDesk, MotionDesk, and AutomationDesk simulation environment suite for testing lane departure warning (LDW) systems and autonomous emergency braking (AEB).

Euro NCAP test scenarios

The Euro NCAP test scenarios for autonomous emergency braking involve approaching a stationary target in city and urban environments, approaching a slower target, and approaching a braking target. Each scenario involves a vehicle speed between 50-80 km/h with distance to target calculations and operation that provides controlled braking to eliminate or minimize impact.

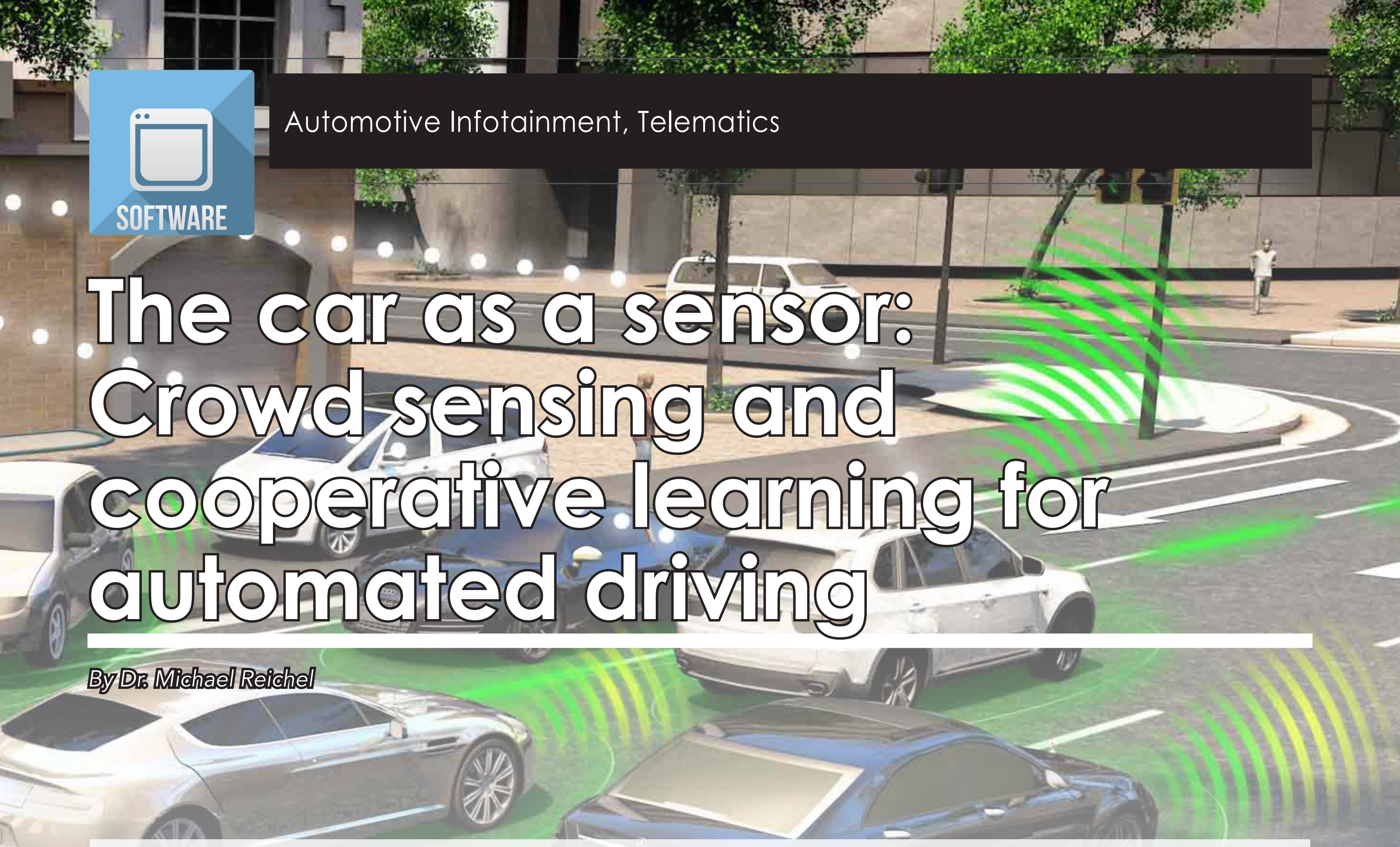
There are also similar test scenarios for AEB involving pedestrian or vulnerable road users (VRU) like bicycles or motorcycles. Mahendra mentioned that these test scenarios are not finalized and may be subject to change by Euro NCAP. Mahendra mentioned that the dSPACE simulation environment provides a library with Euro NCAP test scenarios that allow the user to execute the tests and generate score results. The same test framework can be used for model, software, and hardware testing.

Improving vehicle safety and functionality

ADAS and active safety systems are gaining importance and new challenges are emerging to ensure functional safety of these systems. Virtual test drives and early simulation and testing are becoming a critical factor in providing better modeling, higher quality algorithms, and faster development of advanced driver assistance systems for the automobile. ECD

Reference

Mahendra Muli, Active Safety Symposium 2014, Automated Testing of Active Safety Systems According to Euro NCAP Test Methods.



The car as a sensor: Crowd sensing and cooperative learning for automated driving

By Dr. Michael Reichel

Today's driver assistance systems support drivers in a variety of driving conditions, from hazardous to monotonous. In manageable traffic conditions, such as a freeway traffic jam, some systems are already capable of assuming control of the vehicle. However, their point of view is restricted to the vehicle's direct environment. This view is too limited for the advanced driver assistance that will be required for autonomous driving. Additional details, including information about geographical, topological, weather, and road conditions, will be necessary. These can be provided in the form of real-time infrastructure information in combination with digital map data from vehicles. Information will be gathered from sensor data, as well as from the cloud.

As we move toward automated driving, it will be increasingly necessary to share real-time infrastructure information and digital map data with other road users, in addition to using it within an individual vehicle. The shared data will supplement classic map data and provide drivers with a comprehensive picture of the road network.

Automated driving will also necessitate high data precision, as well as information about relevance and validity. Conventional navigation systems – which are not cloud connected and only updated occasionally when the vehicle goes in for a service – cannot deliver this.

Crowd sensing

In the future, therefore, it will be critical for data from different traffic participants to be bundled (distributed perception and mapping). Sensor data won't just be collected for individual vehicles, but will also be made available to all road users. Vehicles will communicate with each other and exchange data with other vehicles in their immediate environment. Vehicle data will also be stored and evaluated in the cloud for purposes such as map optimization when a section of road is surveyed. Many drivers are already participants in this process of indirect data collection because they make data available via their smartphones. One example of this is data on traffic jams used for traffic reports.

The quality of the information gained from the cloud depends, among other things, on the volume of data. The more vehicles providing information, the more precisely the environment can be mapped. Traffic sign recognition is a good example of this. Even the latest navigation systems don't have information on speed limits for all roads. They usually only have this information available for main roads. They are also unable to provide information on recently implemented changes to speed limits. This information gap can be closed by a road sign recognition function in the front camera. However, the net recognition rate isn't ideal. And in poor weather and bad lighting conditions, it is even lower.

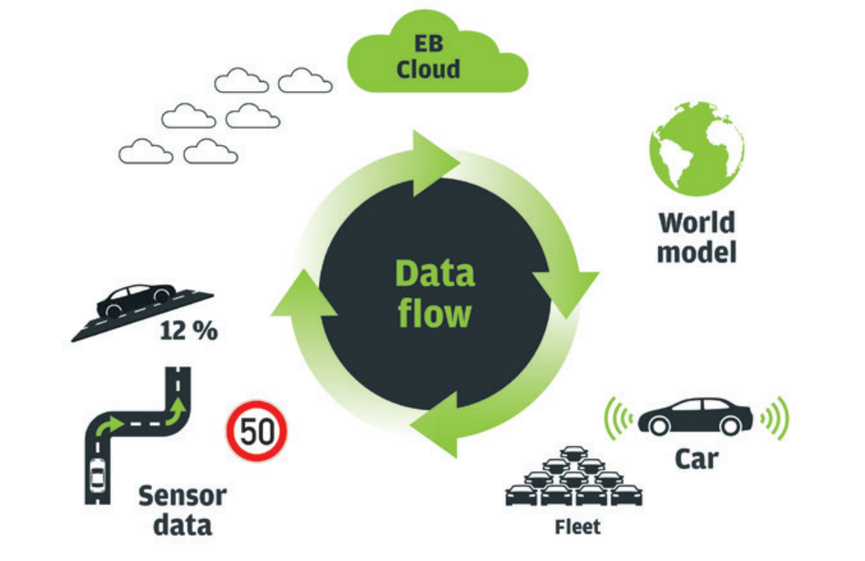


Figure 1 Many vehicles have sensors that record data on speed, incline, curves and other information that is stored in the cloud. This information is used to supplement map material, thereby providing a more comprehensive picture of the worldwide road network.

On the other hand, if a fleet of several hundred thousand vehicles regularly sends its data to a central cloud repository for evaluation, the net recognition rate improves vastly. This is called "crowd sensing," and the sheer volume of data provides a far more precise and up-to-date picture than any map provider's special survey vehicles. In addition to traffic sign data, information about the route that is relevant for transmission and drive strategies will be necessary for hybrid and electric vehicles, and cars with a proactive chassis will need data on road conditions such as slickness or iciness. For all types of vehicles, additional information about curves, lanes, and traffic routing will be necessary (Figure 1).

Map information must be supplemented by metadata so that the vehicle can check the data's validity. For example, today's maps don't include the age of the data or an aging model (i.e., they don't provide information about how old data can be to be classified as reliable). All this information is necessary for autonomous vehicles because different features of a map age at a different rate. For example, data on black ice sensed by the vehicle won't be valid after several hours or even minutes. In contrast, information about a tunnel will probably still be valid even after several years.

From the vehicle to the cloud

A modern, well-equipped vehicle generates several gigabytes of sensor data every minute. Not all of this data can be transferred due to limited network capacity, so the volume of data has to be reduced. To achieve this, the sensor data is initially compared with the vehicle's existing map data in the cloud. For example, in the case of traffic sign recognition, the system checks whether the sign has an identical counterpart in the map material. If it doesn't, the information is uploaded to the cloud.

Even if there is a data match, it might still make sense to upload the information so that the map material can be validated. The decision on whether to transfer the data or not depends on the aging model, the objective being to transfer as little data as possible and as much as necessary.

Once the relevant data is in the cloud repository it has to be preprocessed, grouped, sorted, and interpreted. In the preprocessing stage, obviously incorrect features are discarded. Then, the information on individual map features is grouped. When 100 vehicles provide information on a specific road sign, there will initially be 100 data records that will deviate slightly as to the sign's precise location or the perceived maximum speed limit. Data mining is then used to establish whether the 100 data records all relate to exactly the same sign, where the sign is located, and what kind of a sign it is (Figure 2). Then a calculation is performed to see whether the feature can be assigned to existing map data. After that, the system interprets the data to establish whether the sign is new or not, and whether the aging model has to be adapted. This can happen, for instance, with variable message road signs.

From the cloud to the ECU

When the information in the cloud has been analyzed, it has to be transferred back to the vehicle. Various protocols are necessary for the transfer process. When incremental updates are made over relatively long time intervals, the Navigation Data Standard (NDS) format is used[1]. There are also several proprietary formats, as well as the OpenLR Standard for shorter interval updates[2].

Data dissemination is modularized on different levels to separate fast-changing information from rarely updated information. This reduces bandwidth requirements during data transfer and simplifies the update process. At the same time, it ensures that the in-vehicle systems always have the latest information available. Combining this information with meta-data for relevance also makes it possible to assess data reliability.

The information from the cloud is first integrated into the navigation system's map material. Then it is transferred to the relevant ECUs so that the assistance functions can use it via a protocol such

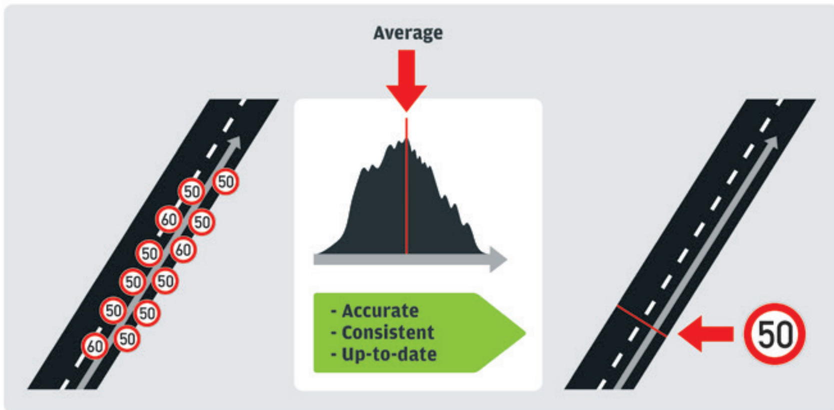


Figure 2 Several vehicles record and transfer data to the cloud about a specific road sign. These data records, which vary in terms of exact location and perceived maximum speed limit, are then analyzed by data mining. The objective is to update or verify existing sign information and to possibly include new road signs in the map material.

as Advanced Driver Assistance Systems Interface Specifications (ADASIS). Anticipatory driving systems, such as the Elektrobit electronic horizon solution, use an ADASIS Reconstructor to receive the data sent by the ADASIS Provider (within the navigation system), and then sort it correctly into the ECU's data structure (Figure 3). The ADASIS protocol ensures that the components interact properly.

Authenticity and data protection

The technical function of crowd sensing must comply with data protection legislation. To this end, the sensor data from vehicles is initially reduced and compressed, and then it is typically anonymized, signed, and encrypted before being transferred to the cloud.

Anonymity protects the drivers' privacy. After the anonymization process, it is necessary to ensure that consecutive information of the same type can be assigned to a sender in the cloud. For example, if the system receives the message "vehicle stationary" several times within a short period of time, it has to know whether the information relates to one or several vehicles. If it is sent by several vehicles, it can be assumed that they are in a traffic jam. The vehicle's identity is irrelevant, as is the fact that different information types relate to one vehicle.

However, there are some exceptions. For example, if the vehicle sends a technical defect message, it needs to be identified correctly by its manufacturer. So, with all information, it is necessary to find a balance between anonymity and identification. The data is signed so that its authenticity and credibility can be assessed and verified. It is also encrypted for secure transmission with standard encryption methods, such as symmetric and asymmetric encryption.

Self-learning systems

Many vehicle manufacturers already store sensor data in the cloud. However, they are still having problems using it efficiently. Their analyses tend to focus on specific functions and are performed manually. In the future, the data to be collected and evaluated will become increasingly complex. In order to exploit the potential of sensors, it will be necessary to increase the degree of automation along the data processing chain. Suitable systems will cover the entire process, from data mining and evaluation to data transfer and use. The vehicle manufacturers will retain ownership of the data at all times.

We are still in the very early stages of vehicle sensor data use. Growth in the number of driver assistance functions in the mid- and lower-price segment vehicles will cause the volume of analyzable data to increase over the next few years. The technologies described allow data from the cloud to be used for both comfort and safety functions. Crowd-sensing information will provide a precise picture of the road network, which will be indispensable as the basis for automated driving.

References

[1] Navigation Data Standard. www.nds-association.org
[2] OpenLR. www.openlr.org
[3] ADASIS. adasis.ertico.com

Dr. Michael Reichel is Senior Manager, Head of Technology and Innovation, Driver Assistance at Elektrobit Automotive.

Elektrobit Automotive
automotive.elektrobit.com
@EB_Automotive
linkedin.com/company/elektrobit-eb-automotive
www.facebook.com/EBAutomotiveSoftware
www.youtube.com/user/EBAutomotiveSoftware

Figure 3 Analyzed data is integrated in the navigation's map material and made available for driver assistance features via a transceiver mechanism (provider/reconstructor).

ADAS: On the road to widespread adoption

By Tina Jeffrey

Advanced driver assistance systems (ADAS) come in many shapes and forms but they share a common goal: to aid the driver and prevent accidents. Examples include systems that perform road sign recognition, blind spot monitoring, driver monitoring, pedestrian detection, forward collision warnings, lane departure warnings, adaptive front lighting, parking assistance, and autonomous emergency braking. Until recently, ADAS systems were, because of their high costs, found mainly in luxury vehicles, putting their benefits beyond the reach of many drivers. They are now poised to become mainstream, however, thanks to government calls for vehicle safety standards, technology advances that make ADAS solutions more useful and cost-effective, and growing consumer interest in cars that can avoid crashes.

Regulatory bodies are beginning to mandate the deployment of ADAS systems, much as they mandated seatbelts and airbags in the 20th century. For example, in an effort to reduce back-over accidents, the U.S. National Highway Transportation Safety Authority (NHTSA) has decreed that, by May 2018, all new vehicles under 10,000 pounds must have a rear-view camera. NHTSA is also assessing the safety benefits of autonomous emergency braking, vehicle-to-vehicle (V2V) communications, and other technologies.

Meanwhile, a number of countries, including the U.S., have implemented government-regulated New Car Assessment Programs (NCAPs), which evaluate the crashworthiness of new vehicles. The Euro NCAP, for example, offers an advanced rewards program for ADAS systems that have been scientifically proven to provide a safety benefit to consumers. Case in point: the "Collision Prevention Assist" system in the 2014 Mercedes-Benz V-Class van, which received a five-star safety rating from the Euro NCAP. The system combines radar-based forward-collision warnings with adaptive brake assist, which applies the optimum braking force to avoid or mitigate a forward collision while also ensuring that vehicles behind have a chance to stop safely.

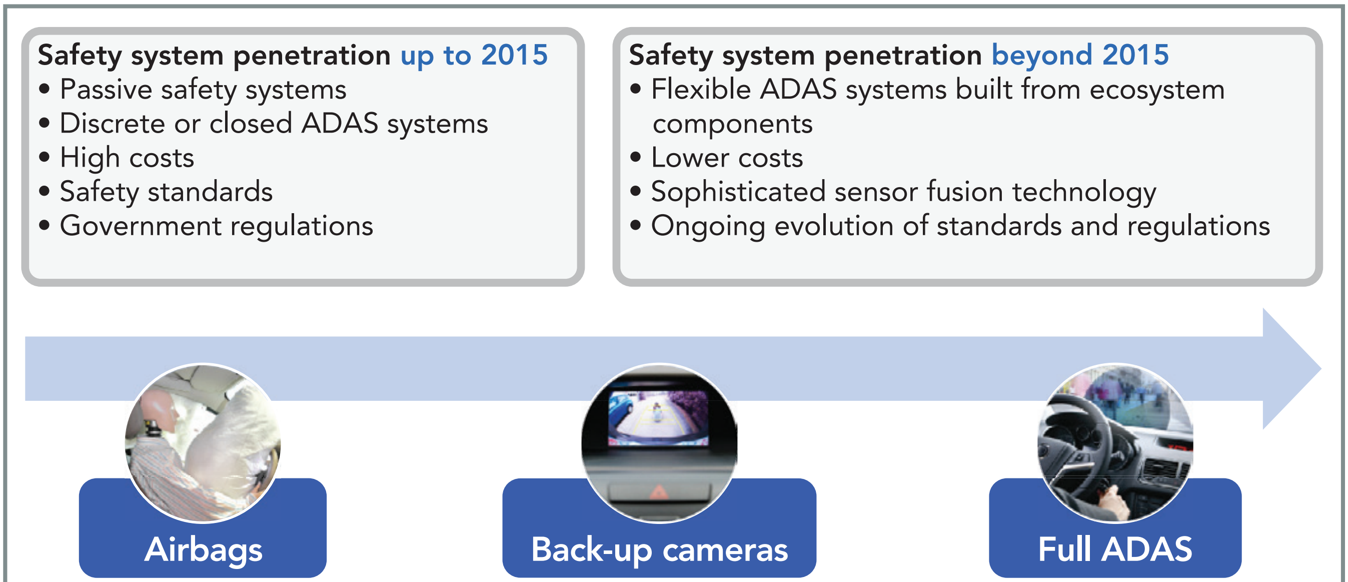


Figure 1 Safety systems have evolved from passive designs that protect occupants once a crash has occurred to active systems that can control steering, braking, and engine throttle to prevent or reduce the severity of a crash.

Active safety

As the automotive industry continues on the path to autonomous driving, systems that offer "passive" assistance by alerting drivers to potential danger, such as a motorcycle in the vehicle's blind spot, have evolved into "active" safety systems (Figure 1). These typically discrete systems can control steering, braking, or engine throttle, using input from radar or camera sensors. Next-generation active systems now in development will be much more effective at preventing crashes. They will integrate multiple heterogeneous networked sensors and use sophisticated decision-making algorithms to interpret the state of the vehicle and its surroundings more accurately. The systems will use new, purpose-built ADAS SoCs that provide support for the multiple sensor technologies. The wide availability of these deep-submicron SoCs, built by companies such as Freescale, Intel, NVIDIA, Renesas, and Texas Instruments, will spur system designers to replace discrete, single-function ADAS systems implemented on low-end microcontrollers with smarter, consolidated multi-sensor systems that can handle high data throughput in real time.

The new SoCs can manage a large amount of data from multiple streams, including vision, infrared, LiDAR, ultrasonic, and radar (long, medium, and short-range), allowing the ADAS system to achieve highly accurate detection and recognition of pedestrians, vehicles, and other objects (Figure 2). These SoCs often incorporate multiple CPU cores, digital signal processors, general-purpose graphics processing units (GPGPUs), or vision acceleration engines, along with multiple camera inputs and display outputs. The SoCs will offer automakers increased flexibility and control over how their ADAS systems are designed, enabling commercially attractive and differentiated solutions. Rather than adopt a shrink-wrapped hardware/software system as many do today, automakers will have the freedom to leverage a full ecosystem of ADAS suppliers, with best-of-breed technologies making their way to the forefront – much the same way that infotainment systems evolved.

Sensor fusion

ADAS systems must function under a variety of weather and lighting conditions. A vision-based system, for example, should have the intelligence to detect a poor visibility condition, such as snow, heavy fog, or sunlight shining directly into the lens. The system could then disable itself and warn the driver that it is non-operational, or, better yet, prioritize complementary sensor data, such as radar for forward-collision detection, that isn't affected by inclement weather. Another example is an ultrasonic parking sensor that becomes prone to false positives when encrusted with mud. In that case, the system could disregard ultrasonic sensor data and use short-range radar or camera data instead. By combining the results

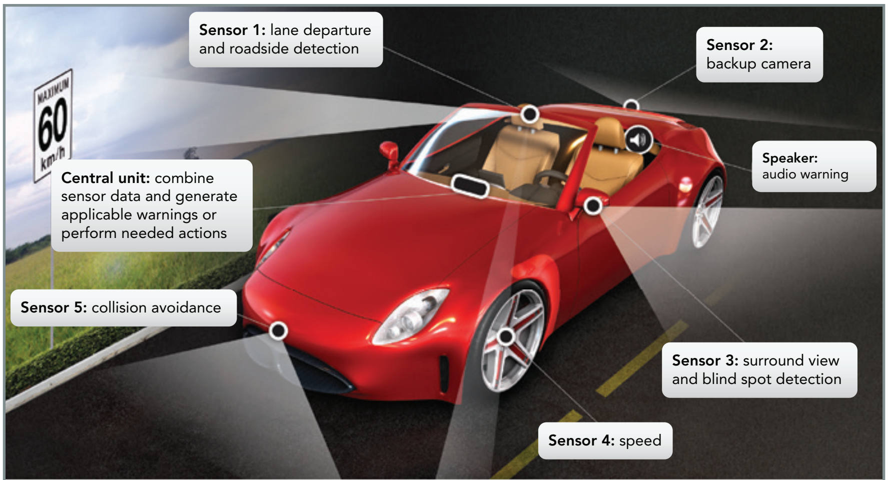


Figure 2 ADAS systems combine data from different sensors to interpret the vehicle's surroundings and perform ac-tions or provide warnings that help avoid collisions.

of different sensors or different sensor technologies through sensor fusion, system designers can create a more effective solution than by using a single technology in isolation.

As systems become more integrated and present more data to the driver, they can potentially cause driver information overload. This overload could in turn result in a high cognitive workload, reducing situational awareness and mitigating the effectiveness of ADAS. System designers must therefore devise easy-to-use systems that make use of the most appropriate modalities (visual, manual, tactile, aural, haptic) for the task at hand. To ensure an optimal user experience, designers must also establish a clear specification of the driver-vehicle interface to ensure the correct balance of user and system requirements.

Testing and validating ADAS systems is, arguably, one of the greatest challenges automakers face. Prior to deploying a commercial ADAS system, the development team must amass hundreds, if not thousands, of scenarios for testing in a regression test database, in an effort to test all scenarios. The ultimate goal is to achieve 100 percent accuracy and zero false positives under all possible conditions, regardless of traffic, weather, or number of obstacles or pedestrians in the scene. But how can the team be sure that the test database comprises all test cases? The reality is they cannot – which is why suppliers spend years testing and validating systems, and performing extensive field trials in various geographies prior to commercial deployment.

A matter of compliance

Compliance with multiple safety-related standards has become a minimum requirement for ADAS systems. It allows an automaker and its suppliers to demonstrate that they use consistent, auditable processes in their product development and manufacturing, and in their risk assessment of hazardous operational scenarios. For example, compliance with ISO 26262, an automotive adaptation of the IEC 61508 functional safety standard, demonstrates that a system is designed, implemented, and maintained to meet its automotive safety integrity level (ASIL), which can range from level A (representing the lowest degree of hazard) to D (representing the highest). The system manufacturer determines the ASIL according to the severity, probability, and controllability of the risks involved in the system.

Unlike IEC 61508, ISO 26262 doesn't declare precise values for acceptable probabilities of failure. Rather, it assesses risks in a qualitative fashion and defines safety measures to avoid, control, or mitigate the effects of systematic or random failures of the overall system. The standard covers all parts of the automotive supply chain, including the SoCs, operating systems, middleware components, and algorithms developed by Tier 2 suppliers; the hardware and software developed by Tier 1 suppliers; and the final systems built by OEMs.

Safety certification of ADAS systems can be a long and arduous process. Using pre-certified components can make system-level certification much easier and contribute to a greater level of safety. The QNX OS for Automotive Safety, for example, has been pre-certified for use in ADAS systems that comply with ISO 26262, up to ASIL D. This certification makes the OS suitable for a wide variety of ADAS systems, from PRNDL displays to pedestrian avoidance systems.

Besides ISO 26262, ADAS systems may need to comply with additional criteria dictated by the Tier 1 supplier or automaker. On the software side, these criteria may include AUTOSAR (a standardized automotive software architecture for electronic control units), OpenCL (a software framework that simplifies parallel computing tasks), or MISRA (vehicle-based software development guidelines for the C and C++ languages). On the hardware side, ADAS systems need to pass AEC Q100 qualification, which involves reliability testing of automotive-grade integrated circuits (ICs) at various temperature grades. ICs must function reliably over temperature ranges that span from -40 °C to +150 °C, depending on the system.

Stretching the model

While legislation, industry standards, and technology advancements are all helping the ADAS market reach economies of scale, each of these factors must continue to evolve to address the demands of a changing industry. For example, the emergence of SoCs with multiple ARM cores, GPUs, or vision acceleration cores may stretch the traditional AUTOSAR software model beyond its current capabilities. Ensuring optimal inter-core communication and shared-resource utilization in a multicore environment is a complex problem to solve. Add to that the requirements imposed by ISO 26262, and a complete overhaul of AUTOSAR or perhaps even a new generation of OS with established inter-process communication for processes running on different cores may be required.

Before ADAS systems can become as commonplace as the steering wheel, the industry must also address other challenges, including a lack of interoperability specifications for radar, laser, and video data in the car network. For audio/video data alone, automakers use multiple communication standards, including Ethernet AVB, LVDS, and MOST. Consequently, ADAS systems must support a multitude of interfaces to ensure widespread adoption. They may also need additional interfaces for V2V and vehicle-to-infrastructure (V2I) data. The industry needs working models that will enable complementary, redundant sensors to work in concert and thereby increase the efficacy of ADAS solutions.

The road to adoption

ADAS systems are commercially available today and next-generation systems promise great technological advancements. Consumer demand is high and the road to widespread adoption is being paved. In fact, Visiongain forecasts that the global ADAS market will experience double-digit growth between 2014 and 2024, from a baseline estimate of $18.2 billion[1]. Some hurdles to bringing next-generation ADAS systems to mainstream vehicles remain, but given the remarkable strides made over the last few years and the burgeoning ADAS ecosystem, the industry can and will overcome these challenges.

Reference

[1] Visiongain: Automotive Advanced Driver Assistance Systems (ADAS) Market 2014-2024.

Tina Jeffrey is an automotive product manager at QNX Software Systems, where she supports infotainment, ADAS, and acoustics products.

QNX Software Systems
www.qnx.com
@QNX_News
opsy.st/QNXLinkedIn
www.facebook.com/QNXSoftwareSystems
YouTube: www.youtube.com/user/QNXcam

How design and verification technologies can ensure functional safety of automotive SoCs

By Adam Sherer

Functional safety is critical for the automotive systems-on-chip (SoCs) that are the technology backbone for advanced driver assistance systems (ADAS), infotainment equipment, and other in-car systems. However, meeting the various safety standards can be time-consuming and labor-intensive, involving voluminous amounts of data that changes as the standards evolve.

Following certain methodologies can make it more efficient for designers to ensure an automotive system will behave as anticipated, even if something unplanned or unexpected occurs. A set of design and verification technologies that automate fault injection and result analysis for intellectual property (IP), SoC, and system designs can reduce automotive ISO 26262 compliance efforts by up to 50 percent.

What does functional safety involve?

Functional safety is the concept that a system will remain dependable and function as intended even in the face of an unplanned or unexpected occurrence. If a system is functionally safe, then it is assumed that the system is able to avoid unacceptable risk of physical injury or damage.

There are two foundational requirements of a functionally safe system:

- Redundancy provides multiple processing paths that limit the risk that any one error will disrupt the system
- Checkers monitor the systems and trigger error response and recovery features when needed

As SoCs move into smaller process nodes they become more susceptible to errors. For example, phenomena including radiation sources, magnetic fields, and internal wear can all be disruptive to an advanced-node SoC. To assure that an SoC is functionally safe, a designer would typically need to establish a functional verification environment where errors (faults) could be injected into the system. Redundant logic would vote on the correct data to eliminate errors and maintain continuous operation. Checkers would monitor the erroneous data within specified time periods and apply error corrections.

Meeting the ISO 26262 safety standard

ISO 26262 addresses the functional safety of electrical and electronic systems installed in series production passenger cars. An adaptation of IEC 61508, ISO 26262 affects all systems with software- or hardware-based electrical, electronic, or electromechanical components. The standard covers many aspects of safety-related automotive software production, including the qualification of tools used in the development process.

Complying with the safety integrity level outlined in ISO 26262 involves collecting and analyzing a voluminous amount of data. By voluminous, we're talking about potentially tens of person-years in the development cycle for an automotive product line.

With competitive and time-to-market pressures, designers can't afford to spend years addressing functional safety. Yet, for the sake of the end customers, corners can't be cut. However, there are some methodologies that can make it more efficient to comply with functional safety standards.

Classifying faults to set ASIL

The safety verification process involves classifying faults into the safe, dangerous, and dangerous-detected categories; codifying this classification into the safety plan; and executing a verification program to determine the ratio of dangerous-undetected to dangerous faults. The result of this sets the Automotive Safety Integrity Level (ASIL).

In many ways, functional safety verification mirrors functional verification. Typically, in a functional verification methodology, the design under test (DUT) is held as a control while a wide range of stimulus is applied. In a typical safety verification methodology, the stimulus is controlled to a few typical sequences while a wide range of faults are applied to the DUT.

What's challenging about safety verification is that the DUT logic can't actually be changed – changing this logic would invalidate the concept of verifying faults in the actual design. This kind of change would also invalidate the ISO 26262-required tool confidence level (TCL) assessment of the verification tools used. Given these scenarios, safety verification must share both testbench and DUT code and the flow must execute concurrently with the functional verification flow.

The set of monitoring points for the fault detection circuits provide the starting point for safety verification. These points are strobed during the actual design execution, so the same effect must be modeled in safety verification. A small set of functional test sequences stimulate the DUT during safety verification. Once this environment is established, the design nodes must be automatically discovered and then collapsed to create a fault dictionary for safety verification.

The safety verification methodology then iterates over the fault dictionary, injecting permanent and single event upset (SEU) faults. Through this process, a detection condition for each fault is reported. Faults that are reported as undetected or potentially detected require additional debug before they can be classified, as they may be dangerous.

Including safety verification in a functional verification flow

For a small design, it could be feasible to run safety verification using sampled input for the testbench and then analyze the results manually. However, with more complex systems, it makes sense to integrate safety verification as part of the functional verification flow. With this approach, designers can use sophisticated testbenches to control the fault injection and support the debug process. For similar reasons, it also makes sense to use the same simulator for both processes. Doing so would eliminate the efficiency loss that results from debugging result differences that occur when a modified DUT or different simulation engine are used.

A safety simulation process could involve as many as hundreds of thousands or even millions of temporal faults. That's why automated regression verification, as established by metric-driven verification, can increase the efficiency of identifying the undetected and potentially detected fault simulations as well as automatically aggregate the safe faults from the unsafe faults. By applying this approach, safety verification efforts can be reduced by up to 50 percent.

An end-to-end functional safety solution that reduces the automotive ISO 26262 compliance effort is available based on the Cadence Incisive functional verification platform. It includes the Incisive Functional Safety Simulator and a functional safety regression capability in the Incisive vManager solution. Figure 1 shows a screenshot of the Incisive vManager solution, which can help highlight potential and undetected fault runs for debugging. Overall, the functional safety solution automates fault injection and result analysis for IP, SoC, and system designs. For safety requirements tracing, it integrates permanent and transient fault simulation.

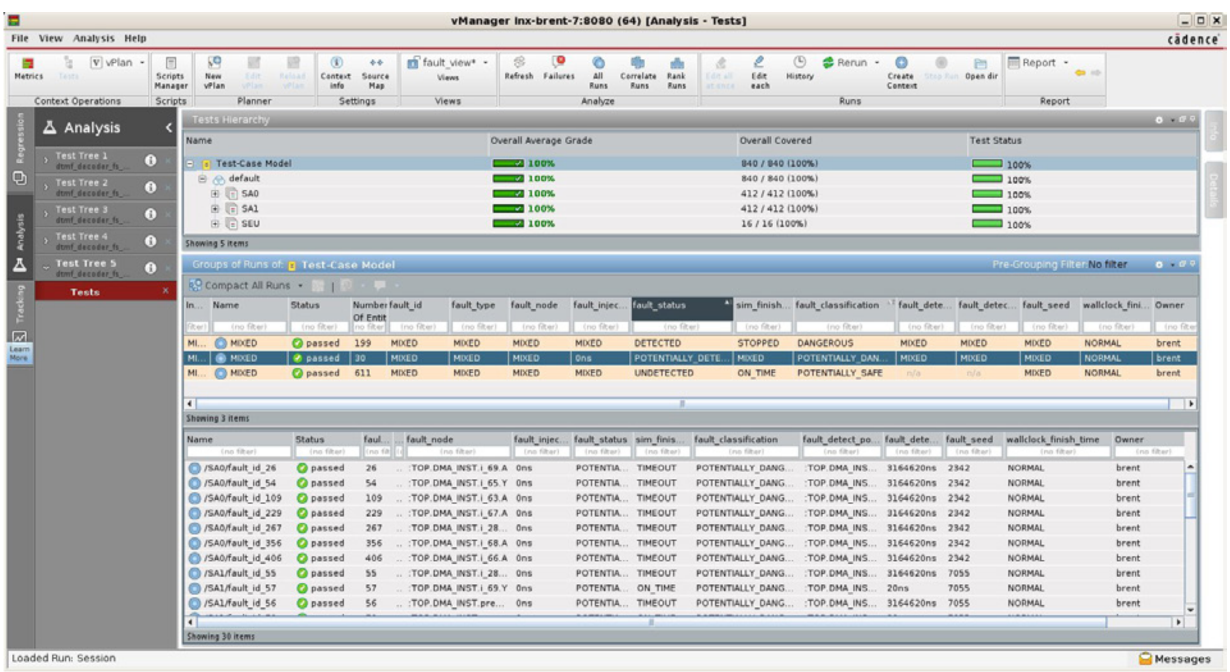


Figure 1 Metric-driven verification can provide a comprehensive functional safety regression analysis.

The Incisive Functional Safety Simulator simulates the unaltered DUT. Faults are injected during simulation and can propagate through SystemC, analog transistor or behavioral models, and assertions. Engineers can reuse their functional and mixed-signal verification environments to speed up the time to develop safety verification. With Incisive vManager, its functional safety analysis capability automatically generates a safety verification regression from the fault dictionary created by the simulator. The solution can then track millions of detected, potentially detected, and undetected faults introduced into simulation to verify a design's safety systems. Figure 2 shows a functional safety verification flow based on the Incisive environment.

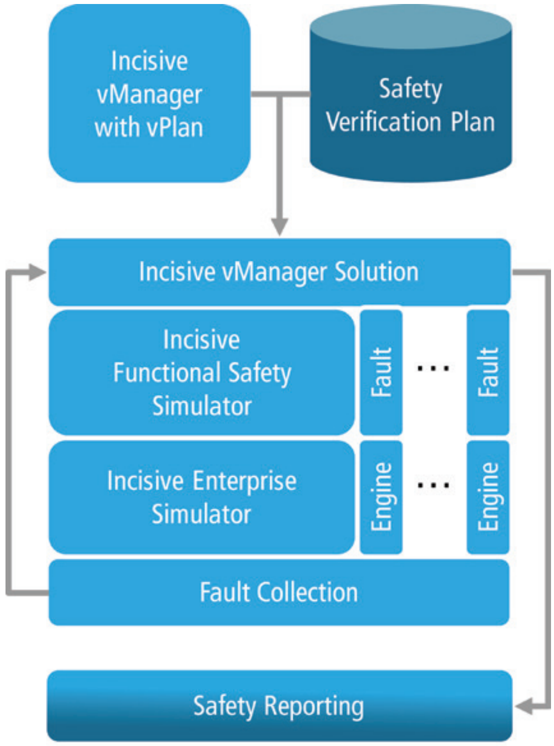


Figure 2 Functional safety verification flow.

Ensuring that an automotive SoC is functionally safe also gives drivers and passengers confidence in their vehicles. Integrating safety verification into the functional verification flow can be an effective way to speed up the process and manage the effort of complying with standards such as ISO 26262. Using functional verification and fault simulation technologies can also minimize your safety verification effort. With these methodologies and technologies, companies can spend more time creating safe and unique automotive designs.

Adam Sherer is Project Management Group Director at Cadence Design Systems.

Cadence Design Systems
www.cadence.com
@Cadence
linkedin.com/company/cadence-design-systems
www.youtube.com/user/CadenceDesign
www.facebook.com/CadenceDesign
opsy.st/CadenceGooglePlus
community.cadence.com

MORE ON... Automotive

ARTICLE
ISO 26262: A static analysis tools perspective
By Paul Anderson, GrammaTech
opsy.st/ISO26262StaticAnalysis

BLOG
V2X technology makes cars safer
By Gene Carter, Security Innovation
opsy.st/V2XCarsSafer

BLOG
Autonomous vehicles on the road: Delphi and Tesla Motors
By Monique DeVoe, Managing Editor
opsy.st/DelphiTesla

E-CAST
Improving transportation safety, efficiency, and the customer experience
Presented by Kontron and Intel
ecast.opensystemsmedia.com/547

Leveraging automotive development standards to mitigate risk, part 1

By Arthur Hicken and Adam Trujillo

ISO 26262, MISRA, and other standards seek to normalize software development for automotive applications by providing a foundation for implementing engineering concepts in software development processes. Some organizations view compliance with ISO 26262 and other standards as an overhead-boosting burden, but the truth is that the cost of failure associated with software defects is much, much greater than the cost of ensuring quality.

When average non-engineer consumers think of electronic systems in automobiles, they likely think of integrated GPS, infotainment systems, and probably some vague notion that there is a computer somewhere in the car controlling some of the safety features. Of course, the reality is that modern cars are significantly more complex with software playing an increasingly larger role in all facets of functionality, including many safety-critical functions. In fact, cars have been leveraging electronic systems for critical functionality for decades, and market changes, such as the push toward an "Internet of Things," have nudged automakers toward embedding a greater number of complex computer systems that run the gamut of criticality.

The business structures and supply chains associated with system development further adds to the complexity. It's rare, if it happens at all, that a manufacturer engineers and builds every component and subsystem in their cars from the ground up, leading to potential integration issues. A transmission is taken from this model, a good braking system from that one. While they may have worked well in their previous environment, in a totally new complex system they may well have unintended and unexpected results. As a result, automotive software is often a complex hodgepodge of systems that may or may not have been sufficiently tested. Implementing components in an ad hoc manner without proper testing, especially in safety-critical applications, can be extremely costly.

The upside, though, is that there are known practices for helping automakers mitigate the risk of failure by building software quality into their development processes. In this article, we'll discuss some of the issues contributing to automotive software complexity, as well as the risks associated with automotive software development. We'll also discuss how implementing known development best practices, such as ISO 26262, help organizations mitigate those risks.

Does more code inject more risk?

According to some estimates, a standard mid-range car can have well over a hundred electronic control units (ECU) processing millions of lines of code – and this number is increasing. It's not uncommon for a manufacturer to have several models of cars with over 100 million lines of code.

There is a perception that the more expensive the car, the more software is embedded – and that most of the software is dedicated to high-end infotainment components. While it's true that these systems become increasing complex as you move up the model line, even introductory lines of cars use software to control steering, brake systems, electrical power distribution, and so on. And even seemingly minor shifts in features, such as Bluetooth, climate control, cruise control, etc., lead to exponential growth of code.

We can assume that more code translates to more complexity – and therefore risk – but the impact may not necessarily be significant. A larger contributor to business risk associated with automotive software is the integration of code developed from a variety of sources across multiple tiers. Most components, including ECU-based components, are subcontracted to second-tier providers who subcontract to third-tier providers and so on. Each preceding tier has specific requirements associated with the component they're developing. Organizations often (but not always) have practices in place for analyzing incoming code to ensure that the components function as expected.

But this assumes that every component along the supply chain is new development. In reality, downstream tiers are branching off code written for a specific make, model, and year. The mutation and reuse of code takes place throughout the supply chain, which leads to a testing problem. How does the manufacturer implement end-to-end testing in such a chaotic ecosystem of software development? When the ECU in the steering wheel was originally developed for one vehicle and the ECU in the dashboard was developed for another vehicle, and neither ECU were designed for the vehicle their currently embedded in, what's the impact? How can you ensure that the complete system functions as expected? It is entirely possible for both systems to pass testing as functional but be unable to communicate properly in all situations. What is the risk associated with this gap?

The cost of software quality

When organizations attempt to measure the cost of software development, they tend to look at general metrics: development time for the engineers; testing time for QA; "building materials" in the form of acquiring tool licenses, compilers, and other infrastructure components. These are important metrics, but often overlooked are the costs of failure.

If the software in the braking system fails, what will it cost the business in terms of rework, recalls, audits, litigation, and loss of stock value? What if there is a loss of life? We argue that the cost of quality is the cost of developing and testing the software, including all the normal metrics we identified plus the very tangible costs associated with a failure in the field.

Defects cost automakers a lot of money. NHTSA estimates that recalls and fixes across the industry cost automakers $3 billion annually. When it comes to the cost of software-related issues, a 2005 estimate from IEEE put the cost to manufacturers at $350 per car[1]. When you consider the low profit margins across a line of vehicles, it's conceivable that a serious enough software defect can severely hurt the business.

The bottom line is important, but even more important is that people can become seriously injured or even die as a result of a software defect. And it doesn't matter how far down the supply chain the defect may originate, defects and all their associated consequences become the responsibility of the automaker. As such, any cost analysis around software development needs to take the potential costs of failure into consideration.

The current state of software development

We've argued that the complexity of the tiered supply chain for automotive software contributes to the overall risk associated with safety-critical systems. We've also reiterated the potential costs to automotive businesses. But there's another dimension to this issue that resides in the cultural difference between engineering and software development.

Software development is almost never engineering. That is, certain concepts from engineering principles, such as repeatability, well-exercised best practices, and reliance on building standards, have yet to become firmly established in software development. Additionally, training for software developers can be inconsistent – even non-existent – and organizations would have to go through great lengths to verify that their developers possess adequate knowledge to build safety-critical software.

This is in contrast to engineering in which the attitudes, mindsets, and history of the discipline enforce a process that is less prone to defects when compared to software development. That is not to say that engineers know what they're doing and software developers don't. Rather, it's to say that automotive engineering as a field is twice as mature as software development, and that the intangible, temporal nature of software perpetuates a cavalier attitude in which if it works, then it's done.

The emphasis in software development is around faster delivery and functional requirements – how quickly can we have this functionality? There is little incentive from management to implement sound engineering practices into the software development lifecycle. Achieving functional safety in software requires operationalizing certain engineering principles:

- Functional safety must be proactive
- Processes must be controlled, measured, and repeatable
- Defects should be prevented through the implementation of standards
- Testing must be effective and deterministic
- Testing should be done for complex memory problems

The good news is that the attitudes around software development have been evolving. ISO 26262, MISRA, and other standards seek to normalize software development for automotive applications by providing a foundation for implementing engineering concepts in software development processes. Some organizations view compliance with ISO 26262 and other standards as an overhead-boosting burden without any direct value, but the truth is that the cost of failure associated with software defects is much, much greater than the cost of ensuring quality. As in electrical standards that specify a specific gauge of wire to carry a known voltage, coding standards can provide the guidelines that help avoid disaster.

Editor's note: read part 2 at opsy.st/ParasoftAutoPart2.

Reference
[1] IEEE "This car runs on code" http://spectrum.ieee.org/green-tech/advanced-cars/this-car-runs-on-code

Arthur Hicken is Chief Evangelist at Parasoft.

Adam Trujillo is a technical writer at Parasoft.

Parasoft
www.parasoft.com
@Parasoft
linkedin.com/company/parasoft
www.facebook.com/parasoftcorporation?ref=s
plus.google.com/+Parasoft/posts
www.youtube.com/user/ParasoftCorp

AUTOMOTIVE

Q&A

AUTO INDUSTRY OUTLOOK: SCOTT MCCORMICK, PRESIDENT, CONNECTED VEHICLE TRADE ASSOCIATION

With the groundwork being laid for the vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications infrastructure of the future, Scott McCormick, President of the Connected Vehicle Trade Association (CVTA) and advisor to U.S. Secretary of Transportation Anthony Foxx, gives an overview on V2X technology challenges and projections. Edited excerpts follow.

Q What are some of the V2X challenges facing the industry right now?

The bigger issues technically are the security aspects. There's a $10 million federal program with the automakers to secure and harden connected electronic control systems in the car, and it's not just a question of a malicious threat getting into one of these portals because the reality is that's much less of a concern than software collisions. The Boeing Dreamliner has 4 million lines of code; the latest luxury vehicles have 20 million lines of code. There's no way automakers could possibly test for all the software collisions. Microsoft couldn't for Windows 8 – there were 100,000 known bugs when it was released – knowing that they would get fixed over time. Well, that's a whole different beast if you're talking about an embedded system than when you're talking about what goes on in your laptop or your phone.

Also, most of the major changes in the car are going to be electronic and 60 percent of people are now not buying a car if it doesn't have the tech they want in it. So, where before the automakers themselves would decide on a common denominator that everyone wanted and put that in everything, like with satellite radio, now they're having to say, "What is it that *you* want?" Somebody may want sports scores whereas somebody else may want stock quotes where somebody else may want their tunes. So the killer app is now essentially a browser to get what I want rather than what somebody else wants to give me.

That's a fundamental change because historically the quality of service is what drives why people buy and stay with a particular brand. Now the automakers have extended their quality of service to include something they have no control over, which is the quality of the communication protocol – I'm in an urban canyon, does it work? I'm in the middle of Oklahoma and I don't have cell service, what am I going to do? So all of a sudden you have to be more conscientious as an OEM or a supplier or even as an aftermarket producer because people are going to gauge your product on things that you may not have control over.

Q What are your projections for V2X and autonomous driving over the next 5-10 years?

We forecast, and we've talked to SAE and they agree, 500,000 jobs coming up in this space in the next four years alone. As insurers, telecoms, and electronics entities turn their attention to V2X and then the automated driving space, we fundamentally feel that we're at a tipping point, and that the changes in this space could be equivalent to the economic changes that occurred with the advent of broad public web access. It takes a long time to get to that point, but I think now we're at the point where we're going to start to see the convergence of the Internet of Services, the Internet of Things, connectivity, and autonomy that we have to have to go forward.

For more on the CVTA's history and objectives, V2X communications landscape, and how the move toward autonomous driving is transforming the principles of industry, government and end users alike, read the full interview with Scott McCormick: opsy.st/McCormickAutoOutlook

Security and safety in embedded software

By Rod Cope, Chief Technology Officer, Rogue Wave Software

Automotive recalls and events such as Heartbleed and GHOST hit home in 2014, linking real safety and security issues to the phrase "it's a software bug." In the past ten years the number of data breaches in the United States has climbed and will reach a predicted peak of 800 instances in 2015. As companies try to one-up each other with continual innovation, software complexity has gone far beyond our ability to find bugs effectively.

The cornerstone of embedded development is software, and software is where most errors are introduced. Not only has the volume of delivered code increased, the complexity and variety of architectures, platforms, and protocols has increased too. This pushes the number of permutations of state, behavior, interactions, and outputs well beyond our capabilities to test.

Another challenge is the fact that software products are the result of many suppliers, vendors, open source (OS) repositories, and legacy code coming together in a mix of different processes, standards, and cultures. Each input offers a chance to introduce safety, security, or performance-related errors, and most integrators struggle to achieve comprehensive testing across all inputs.

Some industries are just now recognizing that standards give valuable goals to achieve and measures of how to improve. Automotive companies have been using coding and safety standards, such as MISRA and ISO 26262, for some time now but they are just starting to investigate security standards. Adopting common, community-driven security standards such as OWASP, CWE, and DISA STIGs are essential for both educating development teams on what makes code secure and measuring how secure their code actually is.

Most companies use OS to optimize engineering costs without realizing the potential risks to security, technical quality, or licensing liability. Moreover, many companies may not even know where OS is used or delivered, as it's fairly easy for any developer or supplier to include code without anyone knowing about it.

To address this, companies should adopt OS policies and governance platforms that formalize the acquisition, provisioning, and tracking of OS code. This helps eliminate inconsistencies in versioning and licensing and tracks where packages are deployed. Organizations can also adopt OS scanning tools to identify where both the known and unknown packages are, to identify potential risks, and better inform testing activities.

The threat of hackers, data loss, and system downtimes persist across all industries, and with the advent of more communications and connections embedded systems are not as protected as they once were. An effective protection strategy is: don't trust inputs coming into the system, place strict controls on suppliers, and ensure that all inputs are validated and restricted to protect code from malicious data and control.

Automated static code analysis (SCA) is a specific tool that looks for potential flaws and is adept at understanding patterns and behaviors in code, across multiple compilation units and developers, to reveal security holes such as buffer overflows, suspicious incoming data, and unvalidated inputs. More sophisticated SCA tools can also compare code against common security standards, such as OWASP and CWE, to determine gaps in coverage or generate compliance reports.

For open source, using a governance platform that alerts teams to security vulnerabilities in OS packages is an effective method for identifying problems and preventing flaws from getting into the released product.

The rapid growth in automotive complexity, connectivity, and the software supply chain emphasizes the importance of getting security, safety, and reliability under control as soon as possible. Embracing techniques and adopting tools that are proven in other industries will help create systems that stay out of the headlines and deliver a solid path for future innovation.

AUTOMOTIVE

Automotive Product Spotlights

Wind River® Automotive: Advanced Software Solutions to Enable Automotive Innovation

WIND™

Automotive Profile for VxWorks® – An AUTOSAR-Compliant, ISO 26262-Certifiable Platform for Automotive Safety-Critical Applications

- World's #1 real-time operating system for safety- and security-critical applications
- Engineered in conformance with AUTOSAR for ADAS, piloted, and autonomous driving
- Safety capabilities certifiable to ISO 26262 up to ASIL D
- Impenetrable security based on Intel® Platform Trust Technology (fTPM) and ARM® TrustZone®
- (Bluetooth and Wi-Fi)

Learn more at auto.windriver.com

RTPatch®: COTS Binary Diff (Delta) Software Update Tools

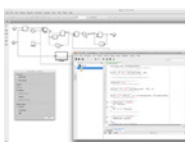
- For all firmware, data and software updating, including FOTA
- Proven safe and reliable with billions of updates applied since 1991
- Typical reduction 90-99%
- Supports any OS and hardware combination, or without OS
- Small footprint, customization available, C source code included
- From industry leader Pocket Soft, established in 1986

Pocket Soft®

pocketsoft.com

1. QGen reinforces the position of the Simulink® and Stateflow® environments as the preferred solutions for model-driven development of safety-critical control systems
2. QGen enforces a safe subset of Simulink® blocks and Stateflow® statecharts, with optional checks for compliance with the MISRA® guidelines on Simulink® feature usage
3. QGen includes a model-based static analysis tool that checks for the absence of run-time errors (such as overflows and division by zero) and logical errors (dead execution paths) and which verifies safety properties modeled with Simulink® assertion blocks
4. QGen supports the generation of highly optimized MISRA-C® and SPARK source code
5. QGen is integrated with AdaCore's flagship tools for cross-compilation, processor-in-the-loop testing and structural coverage analysis without code instrumentation
6. A QGen qualification kit for ISO 26262 will provide a clear specification of which code is generated for every supported Simulink feature

QGen



DesignWare® ARC® Processors Deliver Performance- and Power-Efficient Solutions for Automotive Applications

- **ARC EM Safety Enhancement Package (SEP):** ASIL-D fault detection features in an efficient embedded processor
- **Certified ASIL-D Ready MetaWare Compiler:** ISO 26262 ready software development
- **Power- and area-efficient ARC EM Processors:** Sensor fusion and body control applications
- **DSP and Floating Point options:** Power-efficient signal processing capabilities
- **ARC HS Processors:** High-performance infotainment and connected vehicle applications
- **ARC Metaware Development Tools:** Complete debugging and real-time trace capability

SYNOPSYS®

www.synopsys.com/arc-automotive

For more on latest news, articles, blogs, white papers, and products on Automotive visit embedded-computing.com/topics/automotive

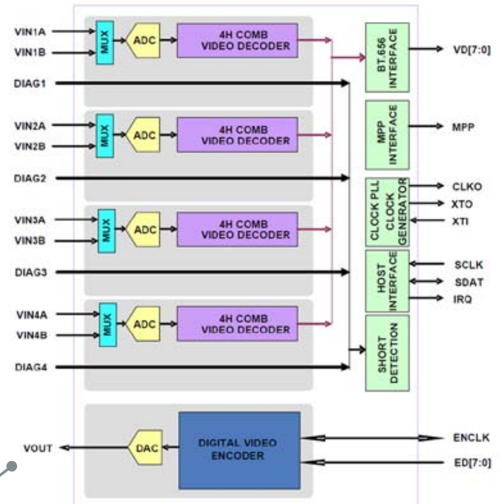
Microchip | www.microchip.com
embedded-compuing.com/p372641

Intelligent network interface controller supports MOST150

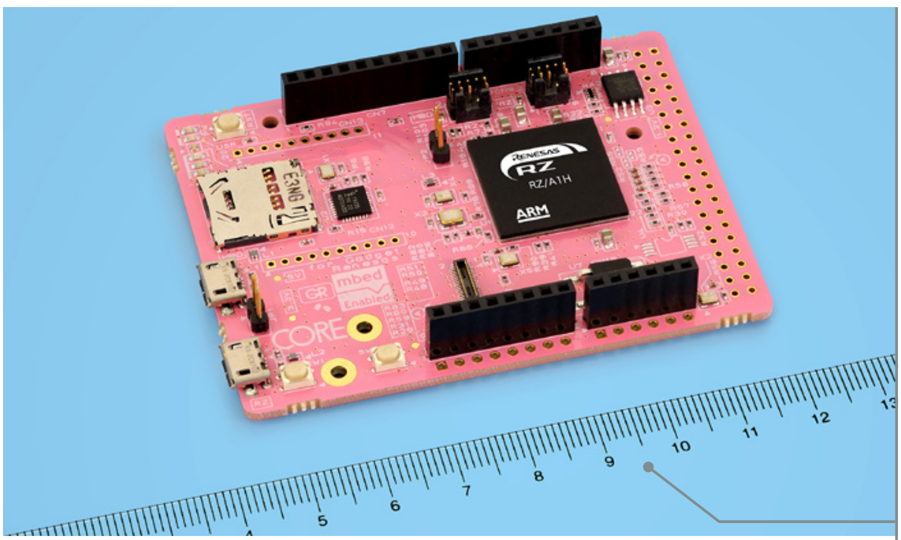
The OS81118 Intelligent Network Interface Controller (INIC) from Microchip provides support for the MOST150 network technology. MOST150 adds support for video and an Ethernet packet channel for transparent transport of Ethernet, making all types of IP communications possible while providing extended support for audio, video, and high-speed control data. The OS81118 INIC is also targeted for applications involving in-car mobility and Wi-Fi connectivity. This INIC includes MediaLB/I2C, USB, and I2S interfaces for seamlessly interfacing with the most current SoCs in the vehicle central processing unit (VCPU), supporting multimedia applications involving audio, video, and IP data, as well as advanced driver-assist applications such as top-view camera systems.

4-channel video encoder/decoder for automotive ADAS applications

The Intersil TW9984 includes NTSC/PAL/SECAM video decoders with auto-detection that convert analog composite video signals to digital component YCbCr data. It integrates four video analog anti-aliasing filters and 10-bit CMOS analog-to-digital converters with differential or single-ended inputs. The TW9984 also includes image enhancement techniques like IF compensation filter, CTI, and programmable peaking.



Intersil | www.intersil.com
embedded-computing.com/p372642



Renesas | am.renesas.com
embedded-computing.com/p372643

ARM Cortex-A processor targets maker industry

Renesas is bringing their RZ/A1 ARM Cortex-A9 based microprocessor to the masses by launching an ARM mbed IoT Device platform to enable faster development of high performance embedded systems that need interconnectivity. The mbed IoT platform delivers a free operating system and licensable software platform incorporating Internet protocols, security, and standards-based manageability. Target applications include home appliances, industrial and medical panels, and automotive applications.