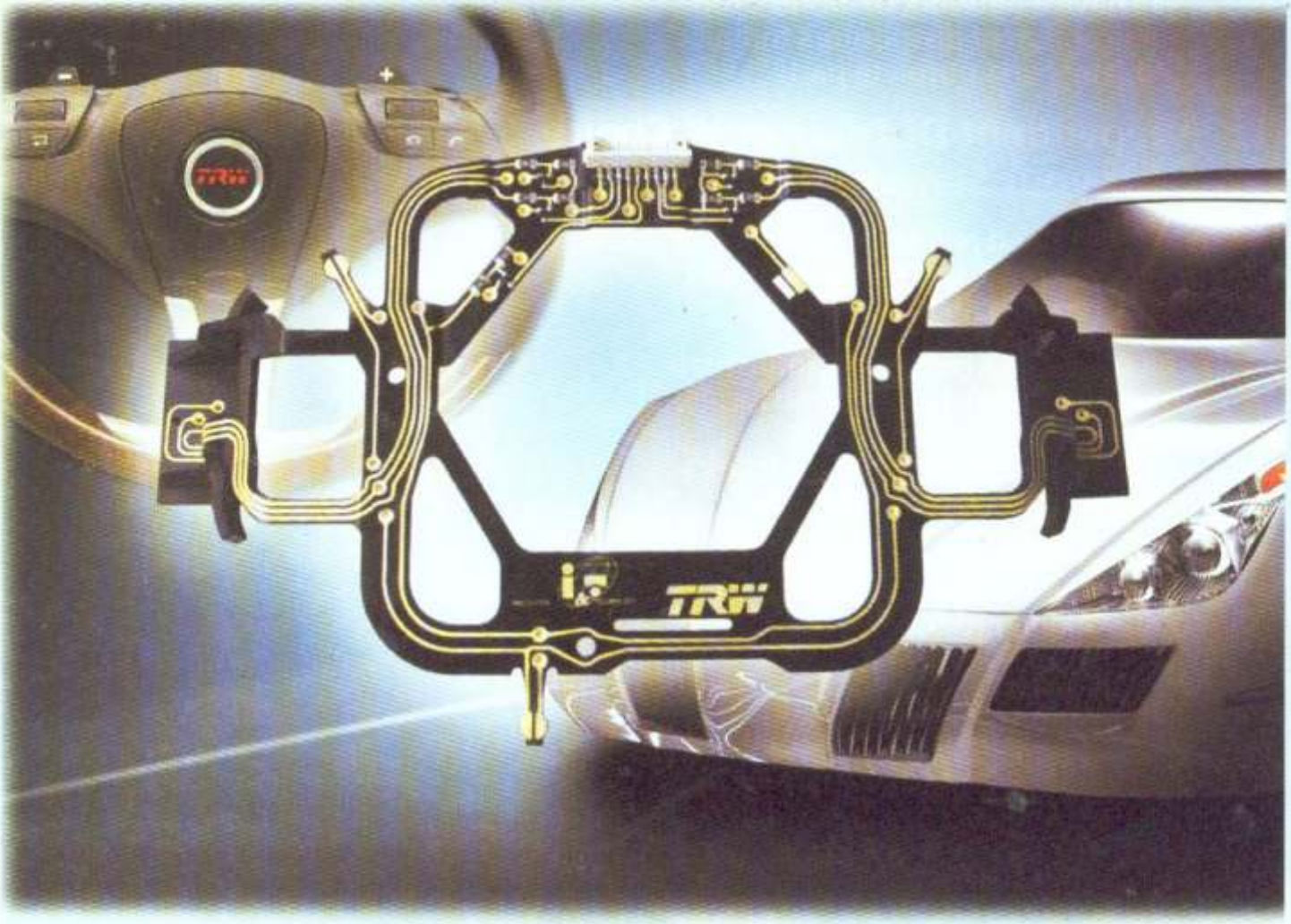
Software Tools and Developments for Automotive Electronics



Vehicles are no longer driven by pure mechanical and hydraulic systems. They are a combination of mechanical and electronic systems coined as "mechatronics". Modern day cars have 100 of ECUs (Electronic control units) with approximately 100 million lines of software code running on them. The greatest challenge the software industry faces in designing automotive software for these 'mission critical' ECUs is high degree of desired quality or "zero defects" expectations. Second challenge is the "automobile domain" understanding that is the knowledge of the operational and functional requirements of a system inside a car. The automotive software industry is increasingly prompt and vibrant enough to tackle these challenges.

RAJESH PAUL, EINFOCHIPS

Innovations in modern automobiles are mostly driven by electronic components and embedded software. The effort spent on software development for new products in the automotive industry is steeply increasing, and so is the value of the created software assets. Automotive Software Engineering has gained significant importance and visibility in research and industry in recent years as an enabling technology and driver of innovation. Providing further innovative automobile functions with highest demands on quality and, specifically, safety, while keeping cost and resource use competitive is one of the software engineering challenges of the next decade. The field requires a sustainable integration of new software engineering methods, development processes, and tools that are specifically adapted and tailored to the automotive domain.

Software Engineering and Process

A group of automobile manufacturers, suppliers and tool developers came together to define AUTOSAR standards, AUTOSAR is a standardized AUTomotive Open System ARchitecture. The objective of this consortium is to create a de-facto standard for vehicle-internal software infrastructure and architecture. It addresses issues including platform independent in-vehicle software components development, inter-operability of software from multiple vendors and reusability of software components. AUTOSAR framework is an excellent platform for automotive software developers and aspirants to get a good understanding of what software building blocks constitutes a vehicle electronic control unit for a given function. On similar lines, the Japanese automobile community also has formed the JASPAR (Japan Automotive Software Platform ARchitecture).

AUTOSAR framework is an excellent platform for automotive software developers and aspirants to get a good understanding of what software building blocks constitutes a vehicle electronic control unit for a given function

Secondly, in order to manage the highly deterministic and real time control requirements of the vehicle ECUs, automotive software are being developed on OS platforms. The industry is increasingly adopting an Open OS standard "OSEK" specifically tuned for the automotive domain. OSEK is a standards body that has produced specifications for the embedded operating system, a communications stack and a network management protocol for automotive embedded system. Many implementations exist on the OSEK standards and are being adopted as a standard Real Time Operating Systems in ECUs. Other highly deterministic RTOS platform for the automotive software includes QNX Neutrino.

Thirdly, the Automotive Special Interest Group (SIG), a joint special interest group of the SPICE User Group has defined the "Automotive SPICE". SPICE stands for "Software Process Improvement and Capability dEtermination". Intention behind developing this process was to provide a common framework for the assessment of suppliers in the Automotive Industry. Adoption to automotive SPICE has benefited software vendors to build a standardized process management in order to organize and manage software development life cycles, improve re-usability, quality, reliability, safety and capability of software and as well reduce development costs.

Many automotive software service companies are adopting and getting Automotive SPICE assessed. This way they are enhancing their software process models and at the same time gaining maturity in building automotive software. Vehicle electronics manufacturers use SPICE as a tool to perform supplier software capability assessments both before and during contract execution.

Quality

The quality requirements of automotive software

which is where the necessity for some coding standards & guidelines like MISRA C. MISRA (Motor Industry Software Reliability Association) was formed by vehicle manufacturers, component suppliers and engineering consultancies. The MISRA C coding guidelines emphasizes on safe, reliable, portable software within vehicles. It is primarily intended for those involved in the creation, procurement and support of vehicle based software. A handful of software vendors produce MISRA C Compliance check tools which are readily used by software vendors & developers to test their automotive software conformance to the MISRA rules.

Complementing the coding standards & guidelines, are the quality management systems and processes that are in place and are rigorously followed by the software industry. These internal quality processes designed for both software verification and validation satisfies the additional quality probe needs that are demanded by the critical automotive software.

Development Tools

Managing automotive software requirements and tracing it down to the final software product has become a difficult task. Easy and powerful way to manage requirements like business objective, customer needs, regulation as well as their traceability a tool called Telelogic's DOORS provide most robust and automatic traceability of requirements throughout the project lifecycle. Traceability is evident down to the source code level.

Automotive software engineers are extensively using modeling tools like Matlab and Simulink from 'MathWorks' to model and simulate their ECU software functionality and even automatically generate code from models. This provides a great flexibility in designing and testing software for vehicle ECUs well before the actual hardware is in place. This even helps testing real time situations & scenarios which otherwise couldn't have been tested.

Today's vehicles are a complex combination of Electronic control units to control various vehicle functions including the Engine management system (EMS), Body & comfort control modules, Chassis control, Powertrain control, In-vehicle Infotainment system, Anti-lock braking systems, Advanced Electronic Stabilization modules etc. All these systems communicate over some form of robust in-vehicle networking buses like CANbus, FlexRay, LIN or MOST networks based on specific needs vis-à-vis error tolerance, bandwidth, determinism, performance, flexibility, security, robustness. There is a great need for testing the software driving these network protocols. Tools like CANalyzer, LINAnalyzer are available to analyze the correctness, performance of CAN protocols integrated in the vehicle bus systems. Developers exploit these tools in their engineering labs to develop fully functional networking software even before connecting their embedded units in a real in-vehicle network.

Development tools like Vector's CANoe is available for simulation and testing of networks and electronic control units. CANoe supports through all phases like development, testing, networking and ECU analysis. Another tool CanDiva along with CANoe can automatically generate and execute the reproducible test cases for implementation and integration of the diagnostic protocol.

These tools reduce design time by as much as 50%, and ensuring compliance with important standards including AUTOSAR, MISRA C and IEC 61508.

In Summary

The embedded electronics' role in automobiles will continue to increase over the years and as well the software requirements in these embedded devices. It is anticipated the software component would itself constitute 15% of the cost of the total vehicle. This eventually calls for more and more innovative and advanced tools to meet the software development need in ensuring best quality, faster development timelines, cost effectiveness and standardized processes.

About the Author

Rajesh Paul is a Project Manager, Embedded division at eInfochips. Rajesh has over 13 years of experience in embedded software solutions for the Automation and Automobile Industries. He manages the Industrial Automation and Automotive electronics Line of Business at eInfochips.