SOFTWARE AND HARDWARE DESIGN CHALLENGES IN AUTOMOTIVE EMBEDDED SYSTEM

Rajeshwari Hegde[1], Geetishree Mishra[1], K S Gurumurthy[2]

[1]BMS College of Engineering, Bangalore, India
rajeshwari.hegde@gmail.com, geetishree@gmail.com
[2]UVCE, Bangalore, India
drksgurumurthy@gmail.com

ABSTRACT

Modern automotives integrate large amount of electronic devices to improve the driving safety and comfort. This growing number of Electronic Control Units (ECUs) with sophisticated software escalates the vehicle system design complexity. In this paper we explain the complexity of ECUs in terms of hardware and software and also we explore the possibility of Common Object Request Broker Architecture (CORBA) architecture for the integration of add-on software in ECUs. This reduces the complexity of the embedded system in vehicles and eases the ECU integration by reducing the total number of ECUs in the vehicles.

KEYWORDS

AUTOSAR, CORBA, ECU, OEM

1. INTRODUCTION

The increasing use of electronic systems in automobiles brings about advantages by decreasing their weight and cost and providing more safety and comfort. The last decade has seen a phenomenal increase in the use of electronic components in automotive systems, resulting in the replacement of purely mechanical or hydraulic-implementations of different functionalities [1]. There are many electronic systems in modern automobiles like antilock braking system (ABS) and Electronic Brake force Distribution (EBD), Electronic Stability Program (ESP) and Adaptive Cruise Control (ACC). Such systems assist the driver by providing better control, more comfort and safety. In addition, future x-by-wire applications aim to replace existing braking, steering and driving systems. The developments in automotive electronics reveal the need for dependable, efficient, high-speed and low cost in-vehicle communication [2]. In the earlier days of automotive electronics, each new function was implemented as a stand-alone ECU, which is a subsystem composed of a microcontroller and a set of sensors and actuators. This approach quickly proved to be insufficient with the need for functions to be distributed over several ECUs and the need for information exchanges among functions [3]. Therefore, fundamental architecture of integrated electronic systems in an automobile is important to be designed in order to optimize the total function, cost and productivity [4]. Today, 90% of all innovations are driven by electronics and software. 50-70% of the development costs for an ECU are related to software. Today, premier cars have up to 70 ECUs, connected by 5 system buses and up to 2500 signals are exchanged by these ECUs [5]. This growing number of ECUs increases the complexity and cost of vehicle development system [6]. Because of the growing system complexity, cost plays a significant role in the development of vehicles. To reduce the vehicle development cost, automotive industrial

DOI : 10.5121/vlsic.2011.2314

consortiums are working on standards for automotive electronic systems and software architecture. These standards would increase the commonality and reusability of software in ECU design and reduce the system cost accordingly. The cost, flexibility, extensibility and the need for coping with increased functional complexity in vehicles are changing the fundamental paradigms for the definition of automotive architectures [7]. The complexity of the vehicle systems is also increasing due to the increase in the amount of hardware and software. Instead of isolated functionality on separate ECUs, distributed systems located on several ECUs with a high degree of interaction are introduced [8]. For accurate and efficient development of electronic systems, well defined processes and powerful tools must be used. The transition from the conventional software development approach is being replaced by the model based approach, in order to meet the development time and time to market. The paper is organized as follows. Section 2 deals with the data processing in vehicles. Section 3 deals with the requirements and challenges to be addressed in automotive domain. Section 4 deals with the hardware and software in vehicles. Section 5 explains the migration to AUTOSAR. Section 6 deals with the CORBA Middleware Architecture to reduce the complexity of electronics and software in vehicles. The paper is concluded in section 7.

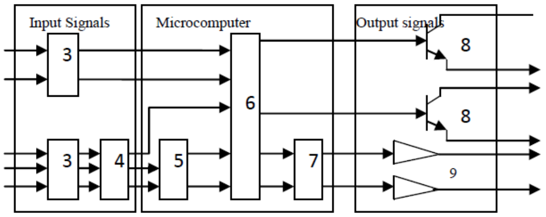
2. DATA PROCESSING IN VEHICLE

2.1. Requirements

Highly sophisticated state-of-the-art open-loop and closed-loop control concepts are essential for meeting the demands for function, safety, environmental compatibility and convenience associated with the wide range of automotive subsystems installed in modern-day vehicles.

ECUs developed for use in vehicles all have a similar design. Their structure can be subdivided in the conditioning of input signals, the logic processing of these signals in the microcomputer, and the output of logic and power levels as regulation or control signals [9]. ECUs generally process signals in digital form. Rapid, periodic, real time signals are processed in hardware modules specifically designed for the particular function. This procedure substantially reduces the CPU's interrupt response time requirements.

Originally, data exchange between the ECUs took place via separate wires. However, this type of point-to-point connection is only suitable for a limited number of signals. The introduction of automotive-compatible communication networks for serial transmission of information and data between ECUs has expanded the data transfer capabilities and represents the logical development of autonomous "microcomputers" in vehicles. The generic block diagram of any ECU is shown in the figure3.



1-Digital input signals 2-analog input signals 3-Protective circuit
4-Amplifier, filter 5-ADC 6-DSP 7-DAC 8-Circuit Breaker 9-Power Amplifier

Figure 1. Signal Processing in the Control Unit

The amount of time available for calculations is determined by the control systems. The software contains the actual control algorithms. Depending on the data, an almost unlimited number of logic operations can be established and data records stored and processed in the form of parameters, characteristic curves and multidimensional program maps. For more complex requirements in the field of image processing, the use of digital signal processor is becoming more widespread [9].

3. Requirements and Challenges to Be Addressed in Automotive Domain

The multitude of embedded system applications, especially for automotive domain is subjected to following specialized requirements and challenges to be addressed:

1.Design and implementation perspective towards systems designed for one or more of applications mentioned before, especially the mission critical applications pose high levels of technical challenges in terms of scarce computer resources, which predominantly include memory, timing and speed.
2. To add to the complexity, the scenarios that a vehicle is subjected in a real life situation are numerous and each of them is unique in its entirety.
3. The production lines of the automobile industry are, not only inherently expensive but also expensive to change frequently. As such, products need to be totally fail-safe before placing them for production. The automotive industry has addressed these issues by adapting the following technologies and networking models:

1. Standard protocols have been originated and used for communicating among different subsystems of the automobile. This shall incorporate certain standardization and addresses the ease of communication among automotive subsystems. Examples of such protocols include CAN, LIN, JTAG etc.
2. The standardized protocols and communication mechanisms have encouraged what are called tier 1 vendors for automobile manufacturers who would be, independently working on different subsystems of an automobile and be vendors to
automotive industry. This working model is possible on account of their ability to provide industry standard interfaces for their subsystems. Major tier1 vendors are Delphi Automation, Valeo Electronics, Visteon, Alps Automotive, etc.
3. Interconnection of subsystems from different Tier 1 vendors is possible again on account of standardization of the interfaces.
4. Addressing the mission critical nature of automotive subsystems, testing has become an important activity in the development of automobile subsystems. As indicated earlier, since the scenario that the subsystem and automobile as a whole is subjected is numerous and are unique themselves. This has necessitated the need of automated test systems. Again adaptation of standard protocols and interconnection mechanisms have paved way for specialized activity of design and implementation of hardware/software driven automated test systems.
5. The evolution of technologies, need of the day supplemented with the need of global standardization has resulted in increasing research potential in the domain. While many companies are engaged in development of test tools that aid the core development of testing tools, in general and also specific to automotive requirements has seen good progress. To make the situation more complex, as the advantages of these technologies are being felt in other domains as well [10].

4. Hardware and Software in Vehicles

Modern cars of today carry more hardware and computation power than the Apollo spaceship that flew to the moon. They carry up to 80 controllers connected by up to 5 different bus systems connected to numerous sensors and actuators as well as a multimedia human machine interface and external devices such as mobile phones, personal digital assistants [11]. Many automotive ECUs implement most of their functionality in real-time software systems. Thus, ensuring the availability of the software system is essential to guaranteeing the dependable operation of the ECU. Each ECU is a system embedded with software. It's no longer possible to study an ECU as a stand-alone system. The automotive industry is facing the challenge of the rapidly growing significance of software and software-based functionalities. Research has shown that software complexity is a major reason for project delay and cost overrun. AUTOSAR is a very recent international effort to address the issue of complexity management of highly integrated ECUs for future requirements[12]. The watchdog is a major design mechanism being used in ECUs to compensate for transient system failures and maintain availability. It is an external circuit/processor that monitors the CPU of an ECU. The application software running on the CPU periodically sends a signal to the watchdog indicating that it is still functioning. If a failure occurs in the ECU software, it would not be able to send this signal, and thus the watchdog would determine that a system failure has occurred. Once the watchdog detects a failure, it triggers a system reset to recover the system and resume normal operation[13]. Table 1 shows ECU software categories [9]. The emission related control ECU includes engine control and transmission control ECUs. They are safety critical and need to collaborate to produce the driving power and distribute it appropriately to the wheels.

Table 1:ECU software categories

Sl. No	**ECU software category**	**Attributes**
1	Emission related control ECU	-Safety critical
		-Emission regulation compliance -Less user interface -Long product life cycle
2	Non Emission related control ECU	-vehicle usability concern -safety regulation(concern) -less user interface -long product life cycle
3	Telematic ECU	-service commitment -conditional user interface complexity -medium product life cycle
4	Infotainment ECU	-less safety related -fancy user interface -neat feature size -adapting consumer electronic features -shorter product life cycle
5	Off-board ECU	-interacts with on-board system -adaptive to multiple vehicle models

It consists of software components that include the RTOS, network control, application control, sensor control, actuator control, on-board diagnostic and the self-test component. Non emission related ECU controls the vehicle electrical and mechanical parts to fulfill non-emission related vehicle system functions. Being with control-oriented software attribute, the architecture of this ECU is similar to the emission related control ECU. Telematic ECU furnishes vehicle driver with additional value-added services, it needs to fulfill the service commitment promoted by the carmaker or the service provider. Off-board ECU interacts with the on-board electronics through vehicle bus to examine their operation status. It hooks to the vehicle bus through the cable that connects to the vehicle test connector[9].

4.1. Autoelectronic Innovations

The automotive electronics market has been growing faster than the overall electronics market and much faster than actual vehicle production. For the next several years, research predicts that automotive electronics will grow at a rate of more than seven percent. Over the course of this decade, the worldwide market for automotive electronics is expected to double [14]. In fact, many industry observers expect electronic components to account for 40% of total car production costs in the near future. Electronic components currently comprise some 20-30% of total costs for all car categories, and this figure is expected to reach 40% or so by 2015 [15]. A generic presentation of any automotive subsystem shall be as depicted in Fig.2.

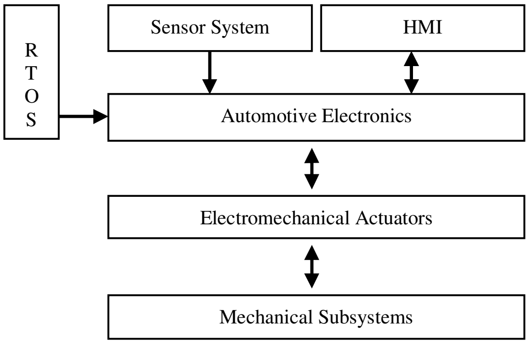


Figure 2. A generic Automotive Subsystem

The main factor behind the rapid increase in the proportion of electronic components used in motor vehicles is the crucial role that electronics plays in developing optimal technological solutions to the four main issues that automakers face today: 1) improving drivability, 2) enhancing safety features, 3) lowering environmental burden, and 4) realizing greater operational reliability. Drivers have always demanded safety and reliability, but now that the rate of car ownership is reaching unprecedented heights, they are also insisting on ease of driving. Automakers must now also address environmental issues, which have become a topic of growing concern. The effective application of electronics technology is absolutely vital to the automotive industry as viable solutions to these four key issues.

As a result of technological progress promoted by electronics, electronics technology has become indispensable to ensuring reliability. In particular, as the code size of software for microcomputer control continues to expand, ensuring the reliability of software has become a crucial matter for the automakers. The proportion of man-hours devoted to software development has been rising sharply. This forces the OEMs to redesign their entire development systems to allow efficient development of highly reliable, large-scale software programs. As part of this initiative, OEMs are promoting standardization of software development implementing AUTOSAR standards.

5. MIGRATION TO AUTOSAR

The increasing complexity of software implementations parallels increasing supply-chain complexity. Software developers design their components based on requirement definitions from the OEMs or Tier 1 suppliers, who are later responsible for their integration. The AUTOSAR development partnership which includes several OEM manufacturers, Tier 1 suppliers, and tool and software vendors, has been created to develop an open industry standard for automotive software architectures. To achieve the technical goals of modularity, scalability, transferability, and function reusability, AUTOSAR provides a common software infrastructure based on standardized interfaces for the different layers [16].

The migration to AUTOSAR in vehicles does not happen at once. Instead, every OEM is applying various migration scenarios depending on what kind of products are suitable at the developing phase of the specific models. BMW already started migration by applying a network and an ECU migration process. The migration of a vehicle's E/E-network to AUTOSAR will follow a step-by-step approach. Starting with a few ECUs, especially those with a new hardware platform, more and more ECUs will be migrated over time [17]. The great advantage of the AUTOSAR standard is the possibility to integrate a high amount of functionality into one ECU in a controlled way. Such development projects involve multiple suppliers and address many cross-domain interfaces.

The need to concentrate on a common stack of infrastructure software is addressed by AUTOSAR with well-specified, standardized basic software that closes the gap between microcontroller hardware and application software. The technical concept of the AUTOSAR approach is a layered model, which is new in the software design for automotive applications. [18]. Fig. 3 shows the layered software architecture.

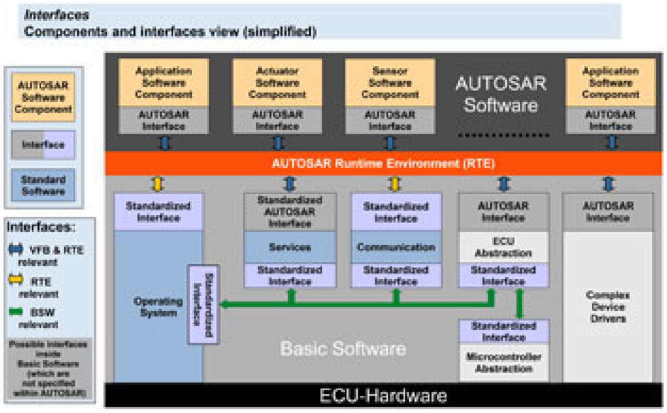


Figure 3. AUTOSAR layered software architecture [19]

The AUTOSAR layered architecture is offering all the mechanisms needed for software and hardware independence. The upper layer is dedicated to the applications; the lower part, the infrastructure, is containing the basic software layer and the Run Time Environment (RTE). The basic software layer containing the 53 Basic Software Modules is organized in 3 layers providing the different levels of abstraction from the hardware (Fig.3): the ECU and the microcontroller ; the upper layer, hardware independent, is providing services to the applications software via the RTE [20].

6. CORBA MIDDLEWARE TO REDUCE HARDWARE AND SOFTWARE COMPLEXITY IN VEHICLES

To reduce the complexity of electronics in modern vehicle, OEMs are struggling to integrate as many software components as possible into the existing ECU, without degrading the performance of the ECU. In this paper, we propose a new architecture to support ad-on software to the existing ECU, using CORBA middleware which is language, location and platform independent [21]. Fig. 4 shows the proposed model to add the new software to the existing ECU.

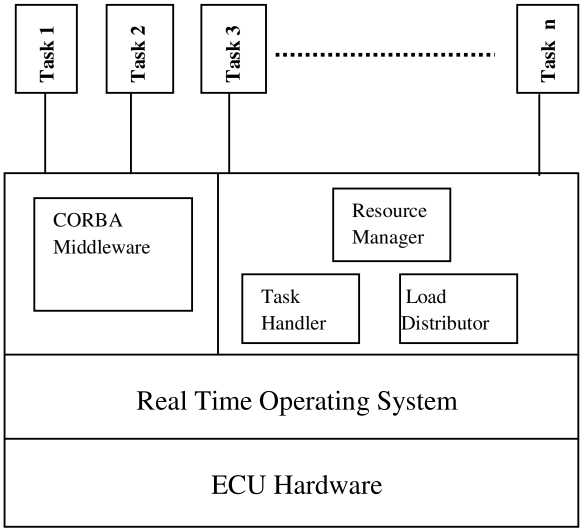


Figure 4. CORBA Middleware Architecture

In the proposed architecture, OSEK Real Time Operating System (RTOS) has been considered with a focus on reusability of application software, and transferable improvement(Hardware, network a non-depending interface). CORBA middleware is a software layer that connects and manages different tasks running on an ECU. It consists of three components, resource manager, load distributer and task handler. When a new task is to be run on the existing ECU, it is detected by the task handler. The task handler of each ECU communicates with other ECUs to know the status of the available resources. The resource handler allocates the resources required for the new task. Depending on the availability of the resources, load distributor assigns the task to the specific ECU. Each ECU is given a unique ID. The ECU with the highest ID is the master and is responsible for the control of the ECUs in the network. If the master ECU fails, a new master will be chosen with the aid of the Bully algorithm which is a method for dynamically selecting a coordinator by the ID. When a new task is detected by all the ECUs, the ECU with less CPU load and the required hardware will execute the new task.

6.1 Load Balancing and Load Sharing in Automotives

Distributed systems such as automotives can suffer from poor performance due to a bottleneck at overloaded ECUs. To address this performance bottleneck, an adaptive load balancing is used to distribute the load from densely loaded ECUs to scarcely loaded ECUs. Not much research has been done on keeping the load balanced across ECUs. To achieve good performance, it is essential to maintain a balanced work load among all the ECUs. Sometimes the load can be balanced statically. However, in many cases, the load on each ECU cannot be predicted a priory. Dispatching tasks from densely loaded ECUs to scarcely loaded ones to improve the overall performance of the vehicle is both logical and feasible [22]. A schedule of the work load that should be moved between any two ECUs , such that each ECU will have the same load on completion is a challenging task. One way to balance the load is to dispatch the job immediately upon arrival. The best load balancing status occurs when all ECUs are at the point of full utilization, without saturation. Each ECU's work load is proportional to its capacity. Allocating more jobs to a fully utilized ECUs might cause imbalance without improving the overall throughput. Since the data movement between ECUs incurs communication cost, the schedule should give balanced load with minimal data movement. Restricting the data movement to the neighboring ECUs might reduce communication cost [23]. Load balancer attempts to ensure that loads are balanced across a group of ECUs. The fig.5 shows the Novel Load balancing model.

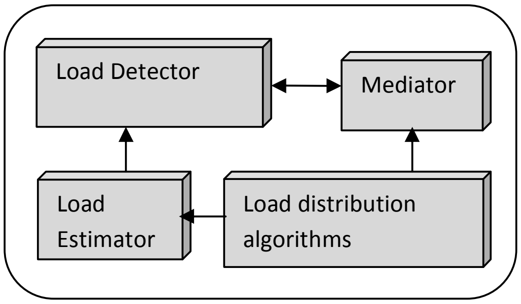


Figure 5 Components of a Load Balancer

The Load Estimator estimates the load on each ECU and intimates the load detector about the overload and under load. The detector detects load imbalance and initiates a trigger when an imbalance occurs. The trigger from the detector initiates the mediator to establish a load balancing session between the ECUs, namely offloading ECU (ECU with the higher load doing the offloading) and the load-accepting ECU (ECU accepting load from the offloading ECU). Depending on which performance metric is to be balanced, one of the offloading algorithms is invoked.

Load sharing mechanism is a software approach to transfer the load from overloaded ECU to underloaded/idle ECU in order to improve the performance of the ECUs. In order to achieve this goal, a load sharing mechanism should have the following features.

1. It generates as little traffic overhead as possible while keeping the load information updated.
2. It creates a low overhead on the CPU for running the load sharing algorithm.
3. Load sharing is done in reasonably short time intervals [24].

7. CONCLUSION

In this paper, we addressed the issues on the deployment of electronics and software in vehicles. It is expected that 40% of total cost of the vehicles will be due to the electronics and software. OEMs are working towards improvising the legacy software using AUTOSAR. The load sharing and load balancing mechanisms are finding way to reduce the complexity of ECU hardware in vehicles to improve the performance. We also presented CORBA middleware architecture for executing Add-on software, which reduces the need for a new ECU when new features are to be added to the existing system, thereby reducing the complexity of the electronics and software involved in vehicles.

REFERENCES

[1] Samarjit Chakraborty, S. Ramesh, “Programming and Performance Modelling of Automotive ECU Networks”, 21st International Conference on VLSI Design, IEEE 2008.

[2] Ugur Keskin, “In-Vehicle Communication Networks: A Literature Survey”, July 28, 2009.http://alexandria.tue.nl/repository/books/652514.pdf

[3] Nicolas Navet, Yeqiong Song, Francoise Simonot-Lion, Cedric Wilwert, “Trends in Automotive Communication Systems”, Proceedings of the IEEE Vol.93, No. 6, June 2005.

[4] Yoshimi Furukawa, Seiji Kawamura, “Automotive Electronics: System, Software, and Local Area Network”, CODES+ISSS'06, October 22–25, 2006, Seoul, Korea.

[5] Hans-Georg Frischkorn, “Automotive Software: An emerging Domain, A Software Perspective”, Automotive Software Workshop, 2004.

[6] Win-Bin See, “Vehicle ECU Classification and Software Architectural Implications”, http://dspace.lib.fcu.edu.tw/jspui/bitstream/2377/3458/1/ce07ics002006000008.pdf.

[7] Marco Di Natale, Alberto Luigi Sangiovanni-Vincentelli, “Moving From Federated to Integrated Architectures in Automotive: The Role of Standards, Methods and Tools”, Invited paper, IEEE 2010.

[8] Alexander Michailidis, Uwe Spieth, Thomas Ringler, Bernd Hedenetz, Stefan Kowalewski, “Test Front Loading in Early Stages of Automotive Software Development Based on AUTOSAR”, IEEE 2010.

[9] BOSCH Automotive Handbook, Bentley Publishers, 7th Edition.Page-1086-1087.

[10] Rajeshwari Hegde, K S Gurumurthy, “Programming Environment for CANoe and its Implementation for Automotive Development Application”, 2009 IEEE International Advanced Computing Conference, March 2009, Patiala, India.

[11] Manfred Broy, “Automotive Software and Systems Engineering”, IEEE 2005.

[12] Gia Nghia Vo, Richard Lai, Mohit Garg, “Building Automotive Software Component within the AutoSAR Environment - A case study, Ninth International Conference on Quality Software, 2009.

[13] Charles Shelton, Christopher Martin, “Using Models to Improve the Availability of Automotive Software Architectures”, Fourth International Workshop on Software Engineering for Automotive Systems (SEAS'07),IEEE 2007

[14] B S Nagabhushana, Rajeshwari Hegde, Veena Hegde, “Avenues and Technologies for Information in Automotive Industries”, National Conference on ICSA, Chennai, 2006.

[15] Automotive industry seeking electronic solutions to four main issues, Available at: http://e2af.com/trend/071210.shtml

[16] Alberto Sangiovanni-Vincentelli, Marco Di Natale, “Embedded System Design for Automotive Applications”, IEEE 2007.

[17] Simon Fürst, “Challenges in the Design of Automotive Software”, EDAA 2010.

[18] Helmut Fennel, Stefan Bunzel, Harald Heinecke, Jürgen Bielefeld, Simon Fürst, Klaus-Peter Schnelle, etl, “Achievements and exploitation of The AUTOSAR development partnership”, CTEA, 2006.

[19] Available at : www.autosar.org

[20] Gerulf Kinkelin, Alain Gilberg, Bertrand Delord, Harald Heinecke, Simon Fürst, Juergen Moessinger etl, " AUTOSAR on the Road", CTEA 2008.

[21] Rajeshwari Hegde,Kargal Satyanarayana Gurumurthy, "CORBA Architecture for the Integration of Add-on Software in Automotive Systems", Journal of Communication and Computer, ISSN 1548-7709, USA, October 2010, Volume 7, No.10 (Serial No.71)

[22] Rajeshwari Hegde, K S Gurumurthy, "Challenges in Integration of ECUs in Automotives",IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6–7March 2009.

[23] Rajeshwari Hegde, K S Gurumurthy, "Load Balancing Towards ECU Integration", International Conference on Advances in Recent Technologies in Communication and Computing, IEEE 2009.

[24] LAN Youran, " A Dynamic Load Balancing Mechanism for Distributed Systems", Journal of Computer Science and Technology, Vol11, No 3, May 1996.

Authors

Rajeshwari Hegde completed her Bachelor of Engineering in Electronics and Communication Engineering from National Institute of Engineering, Mysore and Master of Engineering in Electronics from BMS College of Engineering, Bangalore and perusing her research in the field of automotive embedded system at Bangalore University, India, under the guidance of Dr K S Gurumurthy. Her major field of study includes embedded system, signal processing and wireless communication.
She is currently working as Assistant Professor in the Department of Telecommunication Engineering, BMS College of Engineering, Bangalore, India. She has published 33 research papers in international conferences, national conferences and reputed journals. Her research interests are automotive embedded system and signal processing. Prof Hegde is the member of IET.

Geetishree Mishra completed her AMIE in Electronics and Communication Engineering from The Institution of Engineers, India, Master of Technology in Digital Communication from BMS College of Engineering, Bangalore, India. She has seven years of industry experience from Radiant Telesystems Limited Bhubaneshwar, Orissa and teaching experience of three years. Currently she is working as a lecturer in the department of Electronics and Communication Engineering, BMS College of Engineering, Bangalore. She has published four papers in International conferences and journals. Her research interests include Automotive Embedded Systems, Signal Processing, In-Vehicle networks.

Dr K S Gurumurthy completed his Bachelor of Engineering in Electronics and Communication , from Mysore University and Master of Engineering in Solid State Electronics from U.O.R., Roorkee (now IIT, Roorkee), and PhD from Indian Institute of Science, Bangalore, India. His major field of study includes VLSI and microelectronics.He has two years of industrial experience and more than 30 years of research and academic experience. He is currently working as Professor and Head, Department of Electronics and Communication Engineering, University Visvesvarayya College of Engineering, Bangalore, India. He has given more than 50 guest lectures on different topics of VLSI VLSI DESI at various colleges, WIPRO,CRL and at IETE. He has published more than 75 research papers in various international journals and conferences. His research interests are VLSI Design, Embedded System and Communication.
Dr Gurumurthy is the professional member of IEEE and ISTE. He is the recipient of KHOSLA AWARD for the BEST Technical Paper presented at UOR, Roorkee, India.