International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

# Reprogramming of Automotive Electronic Control Units: A Review on Evolution of Automotive Reprogramming

**Swarnashree S A[1], Mahaveera K[2]**

[1]*P.G student, Dept of Electronics and Communication Engineering, NMAMIT, Nitte, Karnataka, India.*
[2]*Assistant Professor, Dept of Electronics and Communication Engineering, NMAMIT, Nitte, Karnataka, India.*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract -** *Automotive suppliers together with OEMs (Original Equipment Manufacturers) follow some crucial steps to check the security of Electronic Control Units (ECU). Control units are verified to check if it is working in a secure and efficient environment. This process is assisted by updating the software of ECUs to the latest version available through a process called reprogramming. This process also helps the control units to be equipped with the latest security patches and bug fixes. This paper attempts to review the evolution trend to improvise the reprogramming practice.*

***Key Words***: **Reprogramming, Diagnostics, Bootloader, CAN, UDS, AUTOSAR, OTA.**

## 1. INTRODUCTION

Hike in the count of automobiles with increase in the urban population add to traffic congestion along with carbon emission. Air pollution being the major outcome has caused various illness in nearly all the cities. Automotive industries are held responsible for designing vehicles which are environment friendly, sustainable and fuel efficient. As driving less is the main tactic proposed to handle vehicle emissions, several ways to achieve the same are being studied. Updating the software of ECUs (Electronic Control Units) is an available process which helps in decreasing driving time of a vehicle.

Electronic Control Units as explained in Fig 1 are the embedded systems with the ability to control one or more electrical systems in a vehicle.

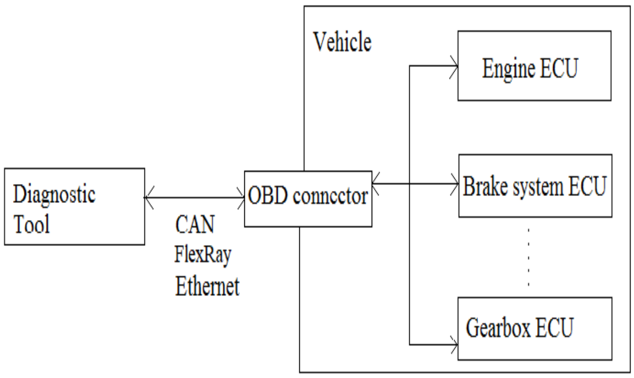


Fig-1: Various ECUs in an automobile

Firmware update may be carried out during the development phase, maintenance phase or updating after sales. Control units are verified to find out if it is working in a secure and efficient environment. This process is assisted by updating the software of Electronic Control Unit to the latest version available through a process called reprogramming. This process helps in maintaining the most recent version of software and security patches. It is a necessity to reprogram the application software of ECUs frequently at service station or in a remote location. Reprogramming is carried out through CAN (Controller Area Network), FlexRay, LIN (Local Interconnect Network) or Ethernet which are the standard communication interfaces. Occupying the ROM of ECUs, the bootloader software is held responsible for reprogramming. It automates the procedure of reprogramming in addition to directing the firmware update. To handle and manage the functions of automobiles, car manufacturers along with suppliers and developers of tools have taken up a standard called AUTOSAR (AUTomotive Open System ARchitecture). It establishes a standardized and open architecture for software of ECUs. The standard suggests that Unified Diagnostic Service (UDS) is the most appropriate protocol to implement bootloader for the purpose of reprogramming the ECUs.

## 2. LITERATURE REVIEW

Sayed et.al., [1] proposed an architecture to carry out firmware updation in a secured manner. The entanglement of automotive vehicle increases with the addition of new features. But keeping the software of control unit updated to its latest version is necessary to avoid future disaster caused due to bugs and also to improve the functionality of the system by adding new features. These needs demand for an architecture which supports the secure updating of the software. To obtain higher security levels, two copies of the required software are sent by the supplier. A random difference in the transmission time of two software copies is set and secured updating of software will be guaranteed. Copies of application software are sent from the supplier to the vehicle only after establishing the secure link at random intervals. Copies received are compared. If it's not altered by hacker, positive response is sent else mismatched packets are to be resent by supplier. By using this architecture, the security attacks which compromise the system functionality and safety are avoided.

Radovan et.al., [2] proposed an efficient method to replace manual reprogramming via electrical connection through wireless reprogramming. Algorithm to update the application software of control units wirelessly is explained. During reprogramming, the central server collects the vehicle identification number of particular set of vehicles to be reprogrammed. The software update information is multicast to the target vehicle with suitable towers. If the target vehicles miss any data packets, they unicast the information about the missed packets. A priority list is created based on how number of times the packet is missed. The packets are multicast according to the priority list. Multicast works fine when the number of audience is sufficiently large. Wireless reprogramming comes with several benefits such as updating can be carried out without the vehicle being physically present at the station, saves time and money of customer.

Karim et.al., [3] proposed a client-server based reconfiguration approach. Car manufacturer can continuously monitor the vehicle sold to get the feedback which helps to produce more reliable vehicles and also to detect if any vehicle needs software updating. For the efficient reprogramming this process can be carried out over the air with the support of AiroDiag. It is a plug and play system which helps the vehicle, phones, traffic signs, buildings to be connected to each other with the help of IP (Internet Protocol) network. With its simple design it reduces the implementation cost. It reduces the trip to garage unless physical support of technicians is needed.

Sungjoo et.al., [4] proposed dynamic firmware update for tightly integrated cyber physical systems. In Cyber Physical Systems (CPS) like unmanned aerial vehicles or smart home appliances, the major challenge is to handle the change in the operating environment where the system is expected to function as designed. Updating these deployed complex systems dynamically helps to adapt in the changing operating conditions and also fix the bugs. Dynamic Software Updating system (DSU) can be implemented to update CPS without stopping the execution of the system. The introduced EcoDSU (ETRI CPS open Dynamic Software Updater), has several advantages compared to other implementations of DSU as it uses the memory very efficiently and also protects the code from being modified.

Guoyang et.al., [5] proposed an efficient way to employ Over-The-Air (OTA) firmware updation which supports after sales maintenance. Convenient way of reprogramming can be achieved by Over The Air method, where customers can update control unit themselves. The server which updates the control units is connected to Vehicle Communication Interface (VCI) via 3G/4G and the interface is linked to the control units with the help of vehicle network. The system mainly requires a smart phone containing application to be reprogrammed, VCI, vehicle network and also remote server. Application present in smart phone is connected to the CAN (Controller Area Network) bus through VCI system and Unified Diagnostic Services is utilized for transferring the program application. Several test results proves that OTA method is dependable and it also defines an easy way of updating the software. This method achieves reprogramming quickly without the need for recalling the vehicle and reducing the maintenance fees.

Young et.al., [6] proposed parallel firmware update which efficiently reduces reprogramming time. ECUs provide the driver an efficient way to handle the vehicle. Increase in the count of ECUs and size of application software has resulted in time consuming reprogramming process. To achieve efficient communication among heterogeneous networks, in- vehicle gateway is utilized which integrates various in-vehicle networks (CAN, Ethernet and FlexRay). As gateway implementation increases the reprogramming performance, the data processing method used by it significantly effects the time taken for software updating. Parallel processing of data along with data routing technique can be integrated to achieve parallel reprogramming which handles software updating of multiple ECUs present in various vehicle networks.

Seok et.al., [7] performed an extensive analysis of the performance of communication protocols. The control units are reprogrammed through a protocol based on diagnostic communication over a network connection implemented between server and the vehicle to be reprogrammed. Network connection along with the protocol implemented plays a major role in enhancing the reprogramming performance. Conventional systems follow CAN protocol to reprogram the control units, which has limited power to reduce reprogramming time. As the complexity of the network increases, the time taken to reprogram also increases which effect the repair and development cost of ECUs. Performance of communication protocols like CAN FD (Fixed Distance) and FlexRay in reprogramming is analyzed. The reprogramming performance of FlexRay was superior compared to CAN FD. To achieve reduced reprogramming period, speed of diagnostic communication network has to be improved.

Luis et.al., [8] proposed an Wi-Fi based firmware reconfiguration technique. The automotive vehicles are improvised by adding latest available intelligent algorithms, security, wireless communications, and automatic control. As the new features are added, the software complexity also increases. This may result in bugs even after profuse debugging process. Driving to the necessary station for servicing is a time misusing procedure. To avoid this reprogramming of an automotive vehicle can be done wirelessly. A well defined architecture consisting of transceiver, server and client can be implemented. It uses the internet to download file which contains necessary information to update the software of ECUs. Transceiver is utilized to handle the interaction between various components of architecture. Reprogramming can be carried out remotely with the help of internet connectivity. It simplifies the reprogramming procedure of electromechanical systems if any software errors occur after the manufacturing process without vehicle being physically present in the service station.

Yutaka et.al., [9] proposed an architecture which employs code-difference algorithm to reduce the size of data transferred. With the increasing count of ECUs, fixing the bugs caused due to timing is a complex procedure while reprogramming. With the automotive vehicles being connected to each other through network, the time period required is considerably less in cases where the network has poor capacity, the reprogramming period plays a crucial role. PC software can be implemented efficiently with the help of Bsdiff which focus on operations on non-fixed-length. It can be improved to suit the automotive requirement which follows fixed-length operation. Reducing the data size with the help of code-difference method can assist in achieving efficient reprogramming.

Hidetoshi et.al., [10] developed an incremental update approach to reprogram resource constraint control units. With the increase in length of program code with the addition of new feature like autonomous driving, efforts of reprogramming also increases. Manufacturers have switched to OTA updating of the application instead of manual procedure. Control units which are resource constraint can be equipped with incremental update which is Bsdiff algorithm based which can reduce the data to be transferred. Three main modifications are applied on algorithm, data format serialization, block based comparison and two stage compression. CAN is used by most widely used and its narrow bandwidth forces to reduce the amount of data to be transferred. As the vehicle is not available during the remote reprogramming process, minimum time has to be engaged to update.

Marco et.al., [11] proposed an systematic approach to carry out reliable wireless firmware update. Even though wireless updating of the application software comes with plenty many advantages like ease of carrying out updating without the vehicle being recalled, cost efficient; it also brings threats of security attacks which may affect the safety of customer and also vehicle. A secure way of wireless updating has to be explored which is efficient and also trustworthy. SecUP ensures the confidentiality and also integrity of the information by implementing asymmetric and symmetric key concept which identifies each nodes present in the network. Analysis of system- centric structure is carried out to achieve configurations of a system in a secured manner. It protects the wireless vehicle interface introduced to support wireless updating which could be exploited as in-vehicle-communication system can be fully accessed.

Marco et.al., [12] proposed framework to carry out wireless updates. Modern vehicles are equipped with safety features along with functions which are vehicle dependent and comfort facilities. It is made possible by connecting a vehicle wirelessly to the road framework, adjacent vehicle and internet support. It is necessary to ensure that the process is less time consuming, safe and reliable in a way that both manufacturers and customers are benefited. Parallel updating and IEEE 802.11s network can be implemented as a wireless medium between vehicle and device carrying out diagnostics. The mesh network is flexible and reliable due to its multi-hop ability. Multicast data stream ability of the network supports efficient parallel updating. The framework is designed to support all requirements of application scenarios, mainly during the development stage followed by the maintenance phase and also during remote updating.

Hidetoshi et.al., [13] proposed an architecture to combine compression approaches to efficiently reduce time taken during updating. Adding up of new features increases the size of application software which increases the number of times the software to be updated. Delta compression can be combined with normal compression strategies to produce an incremental update procedure which can effectively reduce the firmware updating time. Patch file is sent to the ECU which has necessary information regarding bug fixes or addition of new feature to improve the existing application. Patch file is then utilized to produce a novel program application which replaces the old application. Along with OTA it is a necessity to diminish the time consumed during the reprogramming process.

Yutaka et.al., [14] proposed a method to divide the update data to achieve reduction in update timings. Remote reprogramming of automobiles has resulted in connected vehicles which increases the threats of cyber attacks. The software present in vehicles has to be improvised by the addition of latest security patches. Dividing the update data can considerably reduce the time taken to reprogram the control units. Considering in-vehicle systems with multiple ECUs, the ECUs which are not being updated at the moment are made to wait until a required ECU updating is being carried out. This time further gets added with the processing time of data if it is compressed. This can be handled by dividing the data to be updated into several portions where the ECUs are then updated through distributed manner. During update the power supply should be provided, critical functions should be reconfigured very soon and the owner has to wait until vehicle is available. So it is expected to perform the procedure with least time possible. It handles the increasing delay occurred due to addition of ECUs.

Marco et.al., [15] proposed a framework to support the updating process throughout vehicle's lifetime. The reprogramming procedure followed should be efficient enough to support the automobile throughout vehicle's life cycle. A reliable framework called SecUP with IEEE 802.11s network can be implemented which is fast and efficient. It is a reliable procedure that supports software updates to be performed parallelly or partially and is equipped with modern security options to prevent security breach. It handles the updating process from the development stage all the way through after sales maintenance.

Naoki et.al., [16] proposed an method which employs binary difference approach to reduces updating period. Introducing new features in automobiles makes the software present in the control units more complex and also increases the chances of encountering bugs after sales. Binary difference technique can be used in addition to OTA update to obtain considerable reduction in the software update timings. Deflate algorithm which combines Huffman coding and LZ77 is used to compress the update information especially when the bug fixation information is very small. Replacement of old application from ECUs with small RAM size can be effectively done where delta technologies cannot be implemented.

Muhhammad et.al., [17] proposed enhanced architecture which make use of delta file. Updating the control units demand for a sufficient amount of time during which the important functions of an automobile cannot be accessed by the user. An architecture of ECU with delta file consisting of updating information can be utilized allowing the user to access the important function when the update is running. Delta file runs the software modules to be updated and it is later updated in the current software. Addition of NAND memory increases the execution speed of update and the user can utilize the vehicle without being effected by upgrading. Even though using NAND memory increases the initial cost, in long term it reduces the cost and the time consumed. As the delta file is the difference between the old software and the new one, it can be transferred through the network very fast. Using the delta file based architecture reduces the overall time consumed and also allows the user to use required functions of the vehicle during upgrading.

Roland et.al., [18] proposed parallel updating approach of firmware to reduce time consumed. Innovations in the field of automotive industry have led to increase in the frequency of reprogramming to add modern features. It demands for an optimal design for parallel firmware updating of control units. Hybrid algorithm together with mixed-integer linear algorithm can be employed for developing efficient update process. Highly abstract model greatly improves the reprogramming time compared to sequential reprogramming. Automated parallel update assists the manufactures by cutting down the requirements of personnel and also advanced equipments required to carry out update in service stations. The two algorithms efficiently improve the computation time and also time consumed.

Nick et.al., [19] proposed dynamic firmware update through virtualization. To support the firmware update where vehicle critical options are still available during update process, an efficient architecture of control units has to be designed. Virtualization in the operating system level can be implemented as container based approach to assist dynamic ECU updates. This approach provides scalability, isolation and efficiency for the ECU design. The containers can be easily initiated, replicated or ended as no dedicated hardware emulation or operating systems. It efficiently manages the resources without requiring separate operating system. Architecture reconfiguration helps to handle increased workloads. Dynamic updates can be efficiently carried out by virtualization which handles the increasing complexity of automotive architecture.

David et.al., [20] proposed in-situ verification of firmware update through virtualization. Integration of modern features has not only raised the complexity of control units but also made reprogramming into a challenging approach. Architecture which integrates multicore virtualizations to assist in-situ verification to be carried out safely to support parallel updation of currently running software is found to be a better approach. OTA update can be prevented from obstructing the working of automobile by utilizing hypervisor and multicore virtualizations. The automobile which has to be updated through OTA update approach is turned into a Hardware-In-the-Loop (HIL) testbed which allows comparison of old along with new software to check the suitability for targeted system. Deploying in-situ approach for verification is found to be greatly cost effective. It results in highly reliable and secure automobiles as verification process is carried out in all-inclusive scenarios, states, real world conditions and platform variations.

## 3. CONCLUSION

The evolution of reprogramming trends followed to improvise the maintenance and security of automobiles has been discussed in this paper. Having equipped with millions of code lines as an outcome of introducing latest comfort features for the user, firmware updating shifts into a challenging approach. Security vulnerabilities along with code flaws have to be handled well. Many techniques like parallel software update, dynamic software update or data compression approaches have been incorporated to reduce the duration of reprogramming as the vehicle critical functions are not available during the process. Remote firmware updation has also been undertaken to aid the user and manufacturers to effectively tackle time and cost requirements. Future versions of automobiles are expected to be provided with vehicle-to-infrastructure along with vehicle-to-vehicle communications in addition to achieving complete autonomous driving capabilities.

## REFERENCES

[1] S. M. Mahmud, S. Shanker and I. Hossain, "Secure software upload in an intelligent vehicle via wireless communication links", IEEE Proceedings. Intelligent Vehicles Symposium, 2005., Las Vegas, NV, USA, 2005, pp. 588-593.

[2] R. Miucic and S. M. Mahmud, "Wireless Reprogramming of Vehicle Electronic Control Units", 2008 5th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, 2008, pp. 754-755.

[3] K. Mansour, W. Farag and M. ElHelw, "AiroDiag: A sophisticated tool that diagnoses and updates vehicles software over air", 2012 IEEE International Electric Vehicle Conference, Greenville, SC, 2012, pp. 1-7.

[4] S. Kang, I. Chun and W. Kim, "Dynamic software updating for cyber-physical systems", The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014), JeJu Island, 2014, pp. 1-3.

[5] G. Shi, Z. Ke, F. Yan, J. Hu, W. Yin and Y. Jin, "A vehicle electric control unit over-the-air reprogramming system", 2015 International Conference on Connected Vehicles and Expo (ICCVE), Shenzhen, 2015, pp. 48-5.

[6] Y. S. Lee, J. H. Kim, H. V. Hung and J. W. Jeon, "A parallel re-programming method for in-vehicle gateway to save software update time", 2015 IEEE International Conference on Information and Automation, Lijiang, 2015, pp. 1497-1502.

[7] S. J. Jang and J. W. Jeon, "Software reprogramming performance analysis of CAN FD and FlexRay protocols", 2015 IEEE International Conference on Information and Automation, Lijiang, 2015, pp. 2535-2540.

[8] L. E. González-Jiménez, R. C. Rodríguez, H. R. T. Rodriguez, H. A. Rivas-Silva and D. Sánchez-Hernández, "Wi-Fi reflashing system for electronic control units", 2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, 2015, pp. 1-5.

[9] Y. Onuma, M. Nozawa, Y. Terashima and R. Kiyohara, "Improved Software Updating for Automotive ECUs: Code Compression", 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, 2016, pp. 319-324.

[10] H. Teraoka, F. Nakahara and K. Kurosawa, "Incremental update method for resource-constrained in-vehicle ECUs", 2016 IEEE 5th Global Conference on Consumer Electronics, Kyoto, 2016, pp. 1-2.

[11] M. Steger, C. Boano, M. Karner, J. Hillebrand, W. Rom and K. Römer, "SecUp: Secure and Efficient Wireless Software Updates for Vehicles", 2016 Euromicro Conference on Digital System Design (DSD), Limassol, 2016, pp. 628-636.

[12] M. Steger, M. Karner, J. Hillebrand, W. Rom, C. Boano and K. Römer, "Generic framework enabling secure and efficient automotive wireless SW updates", 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, 2016, pp. 1-8.

[13] H. Teraoka, F. Nakahara and K. Kurosawa, "Incremental update method for vehicle microcontrollers", 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE), Nagoya, 2017, pp. 1-2.

[14] Y. Onuma, Y. Terashima and R. Kiyohara, "ECU Software Updating in Future Vehicle Networks", 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, 2017, pp. 35-40.

[15] M. Steger et al., "An Efficient and Secure Automotive Wireless Software Update Framework", in IEEE Transactions on Industrial Informatics, vol. 14, no. 5, pp. 2181-2193, May 2018.

[16] N. Suzuki, T. Hayashi and R. Kiyohara, "Data Compression for Software Updating of ECUs", 2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT), Ancona, Italy, 2019, pp. 304-307.

[17] M. H. Sarwar, M. Ali Shah, M. Umair and S. H. Faraz, "Network of ECUs Software Update in Future vehicles", 2019 25th International Conference on Automation and Computing (ICAC), Lancaster, United Kingdom, 2019, pp. 1-5.

[18] R. Herberth, S. Körper, T. Stiesch, F. Gauterin and O. Bringmann, "Automated Scheduling for Optimal Parallelization to Reduce the Duration of Vehicle Software Updates", in IEEE Transactions on Vehicular Technology, vol. 68, no. 3, pp. 2921-2933, March 2019.

[19] N. Ayres, L. Deka and B. Passow, "Virtualisation as a Means for Dynamic Software Update within the Automotive E/E Architecture", 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, United Kingdom, 2019, pp. 154-157.

[20] D. J. Coe, J. H. Kulick, A. Milenkovic and L. Etzkorn, "Virtualized In Situ Software Update Verification: Verification of Over-the-Air Automotive Software Updates", in IEEE Vehicular Technology Magazine, vol. 15, no. 1, pp. 84-90, March 2020.

## BIOGRAPHIES

**Swarnashree S A** is currently pursuing M.Tech in VLSI Design and Embedded Systems at NMAMIT, Nitte, Karnataka, India.

**Mahaveera K** received his B.E. in Electronics and Communication Engineering from Karnataka University, Dharwad, Karnataka, India and M.Tech in Microelectronics and VLSI Design from IIT Madras, Chennai, India. His research interests include Analog VLSI Design, Data Conversion, Embedded Systems and Automotive electronics.