International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163
Volume 1 Issue 10 (November 2014) www.ijirae.com

Secure Software Update Mechanism for Automotive ECU

Karthik Shanmugam
Department Electronics and Telecommunication Engineering,
Bharath University

Abstract— The explosive growth of electronic systems in the modern automobile has made them a complex network of individual computers with large software content. As with any software systems, keeping the software up-to-date is very important to provide reliable and efficient functionalities. Normally, updating the software in an automobile ECU has been a cumbersome process. When a software issue is detected that lead to more expensive recalls for the OEM's to update the software. This paper proposes to present an easy and secure way to update the software for automotive ECU's either by dealerships or customer by using a dedicated Software Update ECU connected to the Vehicle CAN bus. The method described can even be implemented as an off-the-shelf feature to existing automobiles.

Keywords— Automobiles, Software Update, Security, CAN, Electronic Control Unit (ECU)

I. INTRODUCTION

The modern automobiles are a complex network of independent embedded computing systems connected over a network. A Mercedes S-Class automobile consists of over 200 microprocessors running up to 65 million lines of code making it one of the most complex software systems. As with any complex software system, keeping the software up-to-date is very important to provide critical bug fixes or new features to provide reliable and efficient functionalities.

Normally, updating software for an automotive ECU needs expert technicians available in Dealerships or Service centres. The customers has to take the automobile to the service centre and it make take few days to get the vehicle back from the service centre. Many a times and critical software bug fix lead to very expensive vehicle recalls for the automotive manufactures. Since, a malfunctioning automotive ECU can have devastating consequences leading to loss of life security in such systems is paramount and cannot be overlooked. Thus the system to manage the software in automotive ECU's has terrible user experience, prone to human error and time consuming.

The proposed system fixes the short comings of the existing methods to provide a reliable, secure and smooth management of software systems in the automobiles. The system consists of a Software Update ECU connected to the vehicle CAN network with Wi-Fi functionality. The Software Update ECU functions as the gateway to the vehicle communications network and manages the software update functionality for each of the connected ECU's. The update software are signed and encrypted to provide the security and assurance that no wrong or malicious software can be programmed to the ECU's.

The proposed system takes advantage of the fact that most of the ECUs are upgraded over CAN Vehicle network bus using the OBD2 port, the proposed system directly connects to the vehicle CAN bus as an ECU and uses he existing firmware upgrade capability over CAN bus to upgrade the program memory of the targeted ECU.

II. PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture takes care of several critical architectural blocks that manage the software update process from the OEM's server to the ECU's programmable memory. The system architecture describes the functionalities of each of the architectural block of the system from the OEM Server that hosts the software update packages to the final ECU that needs to be programmed.

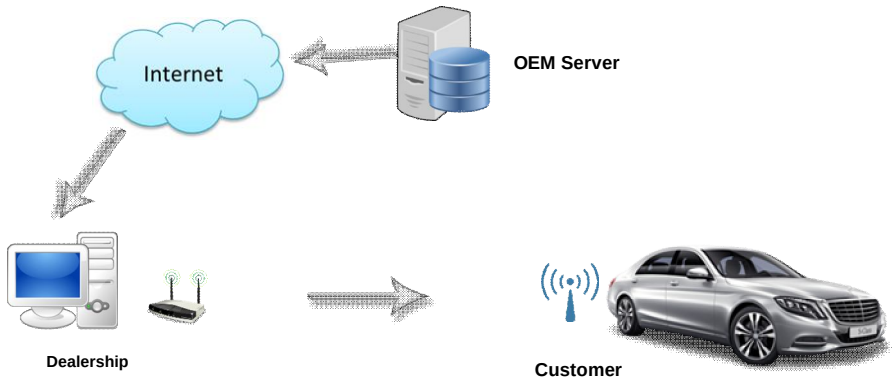

Fig 1: Top-Level Data Flow Diagram

The system aims to simplify the following the pain points in the existing systems:

a. User Interface to manage the software update
b. Securely Transferring the software from the OEM's to Dealership
c. Verification of the integrity of the downloaded software
d. Updating the ECU with the downloaded software
e. Feedback to the Manufacturer's server

A. User Interface to manage the software update

The system specifically uses the Wi-Fi instead of Mobile data to give more control to the users to manage the software update process. The rationale being having a 3G/4G modem inside the car needs cellular operator support to provide network support and it leads to fracturing of solution or unnecessary complexity and it also weakens the security opening the car to another attack vector from malicious hackers.

The automobiles software update ECU's Wi-Fi is enabled only when required and turned-off when it is not in-use. Once the Dealership's computer is connected to the automobile's software update ECU over Wi-Fi. The Dealership's computer will contact the OEM's Server to get the latest software list for all the supported Cars models. Once the software update ECU is connected to the Dealership computer. The Dealership computer queries the current software version and checks if the software needs update and presents the user with the option to update the software.

B. Securely Transferring the software from the OEM's to Dealership

Secure transfer of right software is crucial to ensure the successful update of the ECU software. To enable such secure transfer the following process may be followed. The OEM generates the firmware update packages. Note an update package may contain multiple firm-wares to ensure compatibility between ECU's. The OEM then signs the update package with secure private key. The Dealership computer queries the OEM server and downloads the software update package over secure connection. The update packages MD5 checksum is also transmitted to ensure that the downloaded software update package is not corrupted during the transfer.

C. Verification of the integrity of the downloaded software

The automobiles software update ECU is connected to the Dealership computer using Wi-Fi. Once connected, the software update package is sent over to the software update ECU. The software update ECU is pre-programmed with the OEM's public key to decrypt the software package. This ensures that the software update package is not tampered during the transfer over internet or at the Dealership computer. The software update ECU then decrypts the update package using the public key and ensures the packages integrity

D. Updating the ECU with the downloaded software

The software update ECU initiates the firmware upgrade for the ECU's connected to the vehicle communication network. The commonly used vehicle networks are Controller Area Network (CAN), FlexRAY and MOST. All automotive ECU's support firmware upgrades using proprietary protocol over the vehicle communication network. The software update ECU updates the necessary ECUs firmware from the software update package and refreshes the latest version updated and the success or failure status of the update.

E. Feedback to the OEM's server

The software update ECU then sends the update status to the Dealership computer which then relays the data back to the OEM server for diagnostics and book keeping purposes. If the software update is failed for some reason, the corresponding debug data is also sent to further analyse the issue and implement the corrective action.

III. VEHICLE SYSTEM ARCHITECTURE

A typical Automotive Vehicle Network is described below. Most of the automobiles have two network architecture called MS-CAN which is Medium Speed CAN and HS-CAN which is High-Speed CAN. The data throughput of the MS-CAN is 100Kbps whilst the throughput of HS-CAN can be up-to 1Mbps. The Gateway ECU, a role typically satisfied by the Vehicle's Instrument Cluster which translates the messages between Medium Speed and High Speed CAN Bus.

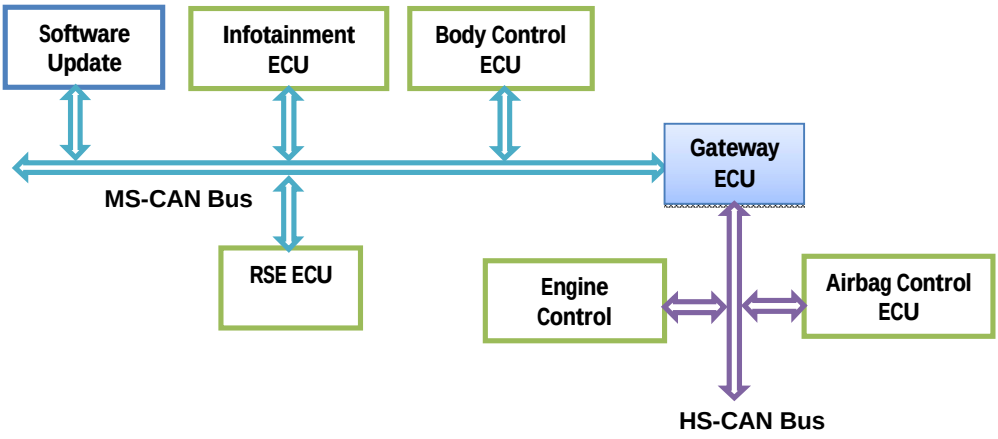

Fig 2: Vehicle Network Architecture

The Medium Speed CAN Bus connects non-essential or non-time critical ECUs such as Infotainment, Body control, Rear Seat Entertainment Systems etc. The High Speed CAN bus connects the time and safety critical ECUs such as engine control ECU, Airbag ECU, etc. The software update ECU initiates the firmware update sequence by sending a targeted CAN message to the selected ECU. The ECU upon entering the CAN boot-loader mode is ready to receive the firmware to upgrade its internal memory.

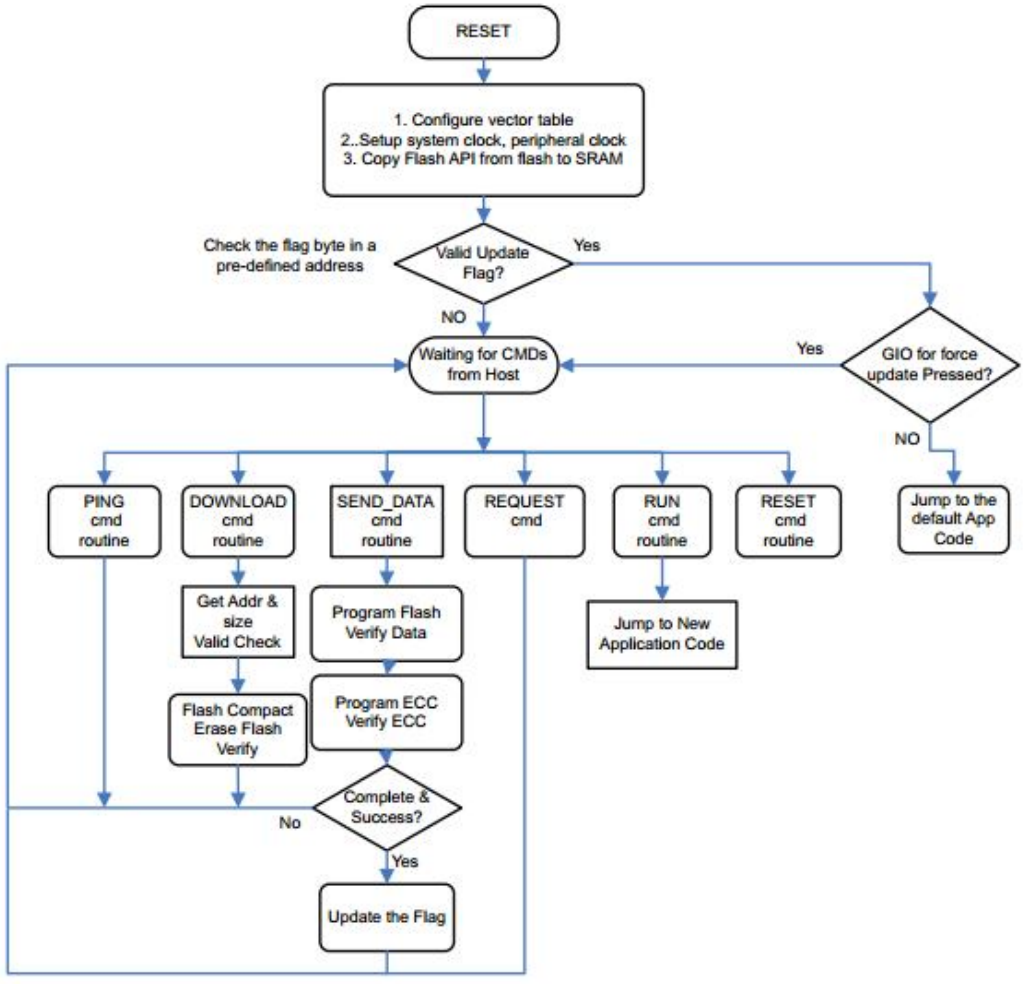

Fig 3: CAN Boot-loader Operating Sequence for TMS470M MCU

The firmware is then transmitted to the selected ECU over CAN bus as CAN Frames. The selected ECU upon receiving the CAN frames starts updating its internal program memory.

Once the entire firmware is transmitted, the software update ECU starts the verification process by reading back the firmware from the selected ECU. The software update ECU then compares the firmware transmitted and firmware received to ensure that there is no corruption during the transfer. If the verification fails, the software update ECU again attempts to update the ECU.

A typical firmware update over CAN using a boot-loader is described below

The critical component in the entire upgrade cycle is the CAN boot-loader present in all the ECUs which enables to update the program memory of the ECU over CAN Bus. The software update ECU and the CAN boot-loader of the selected ECU must agree upon an accepted handshake messages to properly transfer the firmware. Upon the successful completion, the updated ECU executes the new firmware. The verification process is carried out by the software update ECU and the results were communicated to the OEM's server for diagnostics and book-keeping purposes.

IV. CONCLUSIONS

The proposed system to manage the software upgrade process in the automotive system overcomes the short comings in the current practice. The system gives the OEM the required control to properly manage the software for various model year automobiles and successfully update/patch software as and when required. The system also ensures that the security and integrity of the software update process is not compromised at any point during the entire upgrade cycle thus guaranteeing a verified method to keep the software up-to-date in safety critical environments.

REFERENCES

[1] A Comprehensible Guide to Controller Area Network, by Wilfried Voss
[2] Understanding and Using the Controller Area Network Communication Protocol: Theory and Practice, by Marco Di Natale (Author), Haibo Zeng (Author), Paolo Giusto (Author), Arkadeb Ghosal (Author), 2014
[3] CAN Bus Boot-loader for TMS470M MCU, Texas Instruments.
[4] Controller Area Network Prototyping with Arduino, by Wilfried Voss, 2014.

