



ISSN: 2230-9926

Available online at <http://www.journalijdr.com>

# IJDR

International Journal of Development Research

Vol. 15, Issue, 06, pp. 68607-68610, June, 2025

<https://doi.org/10.37118/ijdr.29791.06.2025>



RESEARCH ARTICLE

OPEN ACCESS

## ECU REFLASH ADVANCEMENTS USING COMBINATION OF DELTA METHODS

\*Anand Wanjari

USA

### ARTICLE INFO

#### Article History:

Received 11<sup>th</sup> March, 2025

Received in revised form

26<sup>th</sup> April, 2025

Accepted 14<sup>th</sup> May, 2025

Published online 30<sup>th</sup> June, 2025

#### Key Words:

ECU Reflash, Delta Reflash, Compression, Parallel Reflash, CAN FD, CAN XL, Over the Air Reflash.

\*Corresponding Author: Anand Wanjari,

### ABSTRACT

This paper explores the benefits of the Delta ECU reflash method and its enhancement through integration with other advanced ECU reflashing techniques. By updating only the modified segments of firmware, Delta reflash significantly reduces programming time and improves overall efficiency in ECU management. The paper examines its application across various automotive scenarios, including parallel ECU updates, OTA deployments, and secure environments, highlighting its role in minimizing vehicle downtime and enhancing service productivity. Additionally, it discusses how combining Delta reflash with technologies such as CAN FD, DoIP, CAN XL, and data compression further optimizes performance. This integrated approach supports faster, secure, and scalable ECU reprogramming, aligning with the demands of modern, software-defined vehicle architectures.

Copyright©2025, Anand Wanjari. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Citation: Anand Wanjari. 2025. "ECU Reflash Advancements Using Combination of Delta Methods". International Journal of Development Research, 15, (06), 68607-68610.

## INTRODUCTION

The traditional ECU reflash methods such as Compression, Parallel Processing, Switching Baud rate switching, Over the air programming (Bogdan, D., Bogdan, R., & Popa, M., 2016). Comes with their limitations and may not suit with the faster pace automotive environment. It often involves lengthy procedures that can be cumbersome for both technicians and vehicle owners. By implementing the Delta ECU reflash method, these processes can be streamlined significantly, offering a more efficient solution for software updates. The integration of Delta reflash techniques with traditional methods can yield significant improvements in the efficiency of ECU updates, addressing the challenges posed by increasing software complexity in vehicles. This method stands out due to its ability to streamline the reprogramming process, making it a valuable tool in modern automotive service practices. ECU reflash methods are needed to address the rising complexity of automotive electronics and the increasing frequency of software updates required for optimal vehicle performance. It enhances the overall reliability and security of firmware updates, making it a crucial advancement in automotive technology. The adoption of such advanced reprogramming techniques is vital for automotive manufacturers to stay competitive in an evolving market.

## BACKGROUND

The increasing complexity of electronic systems in vehicles necessitates the adoption of innovative reprogramming methods to ensure timely software updates and maintain optimal performance.

As the demand for efficient ECU management grows, the Delta ECU reflash method emerges as a promising solution to address these challenges and enhance the overall efficiency of firmware updates in the automotive industry. By leveraging delta update technology, manufacturers can minimize data transfer and reduce update durations, ultimately improving vehicle performance and security. This approach aligns with the ongoing transformation in the automotive sector, emphasizing the need for reliable and efficient firmware update systems to ensure safety and compliance with evolving standards. The integration of such methods not only streamlines the update process but also addresses the challenges posed by the growing complexity of vehicle electronics. This paper will also highlight the importance of adhering to industry standards for ECU reprogramming to ensure compatibility and security across various vehicle models and manufacturers.

**Delta ECU Reflash Method:** In an ECU reflash process using the UDS protocol (ISO 14229), the procedure is typically divided into three phases: Pre-Programming, Programming, and Post-Programming (A. Prasad and P. Shanthi, 2021). During the Pre-Programming phase, the reflash tool initiates communication with the ECU by entering the programming session (UDS 0x10), performing security access (0x27) using seed-key exchange or certificate-based authentication, and optionally running diagnostic routines to ensure system readiness (ISO15765-3 2004). This phase may also include digital signature verification of the incoming firmware to confirm authenticity before accepting any data. In the Programming phase, rather than transferring the entire firmware image, the Delta Reflash method is applied to compute and send only the changed memory segments using services like Request Download (0x34) and

TransferData (0x36). The data is often compressed and encrypted to ensure both speed and security during transfer (Ji Zhang *et al.*, 2018). Once all delta blocks are received and verified, the ECU exits with Request Transfer Exit (0x37) and may run an internal routine via RoutineControl (0x31) to perform checksum validation or decrypt and install the update. In the post-programming phase, the tool commands an ECU Reset (0x11) to boot into the new software, followed by restoration of preserved parameters (like trims or learned values) and functional validation. Integrating Delta Reflash into this standard UDS flow significantly reduces data size, transmission time, and flash wear, while maintaining compliance with cybersecurity practices such as encrypted payloads and signed firmware packages. The number of the delta files generated for updating the firmware is given by the next formula, where ROMsize is the data quantity that are to be programmed, RAMsize is the data quantity available for the buffer where the delta file will be received:

$$NUMBER\ OF\ DELTAS = ROUNDUP\left(\frac{ROMsize}{0.9\ RAMsize}\right) \dots\dots\dots(1)$$

In the previous formula (1), the number of delta files is obtained by dividing ROM size to RAM size. From the reserved RAM, 10% will be used by the algorithm, therefore only 90% of it will be available for buffers. That is why the division will be operated on 0.9 \* RAMsize. The CAN protocol used in the auto network achieves speed up to 1 Mbit/sec. Most auto networks use 500 Kbit/sec. Due to redundancy check the efficiency of the protocol is 50% (i.e. the real bus speed is reduced to the half). Firmware update duration is equal to size of data to be transfer divided by the speed of the bus (Bogdan, D., Bogdan, R., & Popa, M., 2016) (2).

$$t = \frac{size_{bin}}{speed_{CANbus}} \dots\dots\dots(2)$$

As an example: suppose that the firmware version X will be replaced by firmware version Y. In this case, let's define:

size<sub>x</sub> = 430 Mbit,  
 size<sub>y</sub> = 439 Mbit,  
 speed CANbus = 0.5 Mbit/sec. Then the time to update the firmware will be 30 m.  
 $t = 439\ Mbit / (0.5\ Mbit/sec)(50\%) \approx 439\ Mbit / 0.25\ Mbit/sec = 1756\ sec \approx 30min$

In the current solution, each time that the firmware has been updated (e.g. a bug fix) the whole content of the memory is replaced with the newer version.

In the proposed solution, one delta file will be generated: NUMBER OF DELTAS = ROUNDUP (64 MB / (0.9 \* 128 MB)) = 1 Then, this delta file will be sent instead of the new firmware. From our previous example, the delta file to convert firmware version X to Y will have a size of size<sub>patch</sub> = 4 Mbit, then the time required to update the firmware will be:

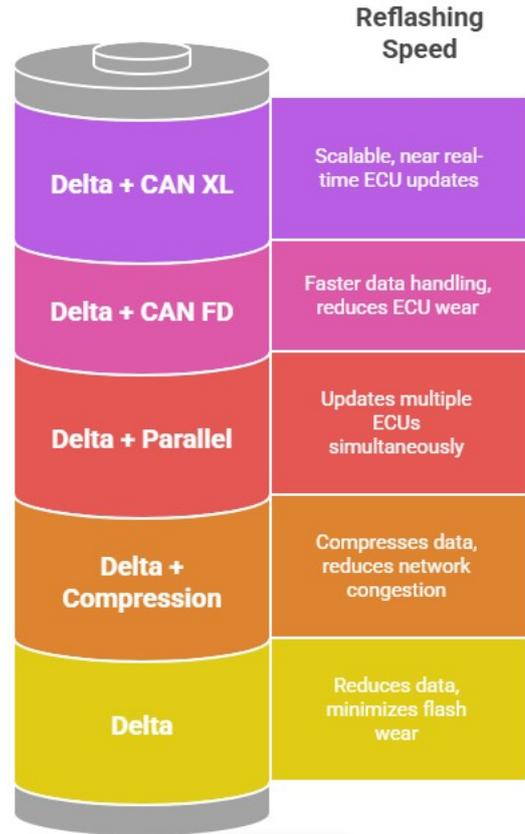
$$t = 4\ Mbit / (0.5\ Mbit/sec)(50\%) = 4\ Mbit / 0.25\ Mbit/sec = 16\ sec$$

The firmware update operation time has been reduced to a 0.1% of the original required time.

**Combination of the ECU Reflash Techniques using Delta Methods**

**Delta Compression Reflash:** The Delta Compression Reflash method is a powerful optimization strategy in ECU reflashing that combines the data size reduction of delta updates with the additional reduction from data compression algorithms. In a delta flash process, only the portions of firmware or calibration that have changed are included in the reflash package, significantly reducing the update size compared to a full image. When this delta data is further compressed using algorithms such as LZ4, gzip, or custom OEM-specific methods, the resulting payload becomes even smaller—sometimes reduced by an

additional 30–60%, depending on the content type and redundancy (A. Prasad and P. Shanthy, 2021).



**Fig. 1. Combination of the ECU Delta Methods as per Reflash Speed**

This compact, compressed delta package is transmitted to the ECU over communication protocols like CAN FD, DoIP, or even OTA, and is then decompressed by the ECU’s bootloader or application layer before being written to memory. The reduced data size lowers the number of required frames, speeds up transmission, and decreases the overall flash time. It’s particularly effective when dealing with ECUs that have larger calibration datasets, such as aftertreatment systems, or modular software structures, where only a subset of components need updating. The Delta + Compression method also reduces network congestion, flash memory wear, and power consumption—making it an ideal choice for production lines, service centers, and over-the-air (OTA) platforms (Onuma *et al.*, 2017) However, implementing this approach requires ECUs to have sufficient processing power and memory to handle decompression reliably, as well as robust error handling to detect and recover from decompression failures. As a result, this method is mostly used in modern ECUs with advanced bootloaders and secure reflash frameworks, often combined with encryption and digital signatures for integrity and security.

**Delta Parallel Reflash:** The Delta Parallel ECU Reflash method is a highly optimized strategy for updating multiple ECUs simultaneously with only the modified portions of firmware. This approach combines the efficiency of delta reflash, which transmits only the changed segments of the ECU software or calibration, with the speed and scalability of parallel flashing, where multiple ECUs are updated at the same time. Rather than reprogramming ECUs sequentially—an approach that can be time-consuming in modern vehicles with 30+ modules—parallel reflash initiates concurrent communication sessions with several ECUs, dramatically reducing total update time (Y. S. Lee *et al.*, 2015). By applying delta reflash within each parallel stream, the data volume per ECU is minimized, making the process not only faster but also more bandwidth efficient. This method is especially useful in end-of-line (EoL) manufacturing, dealer service centers, and OTA environments, where reducing vehicle

programming time directly improves throughput and customer satisfaction. For example, powertrain, body, and ADAS ECUs can all be updated in parallel with their respective delta packages, coordinated by a central flashing tool or gateway (M. H. Sarwar et al., 2019). To ensure reliability, the system often employs synchronization logic, security access management, and error-handling routines to monitor and control each flashing session independently. When implemented correctly, the Delta + Parallel Reflash method can reduce overall programming time by up to 80% compared to traditional sequential full-flash approaches, without compromising ECU data integrity or security.

**Delta CAN FD Reflash:** The Delta CAN FD Reflash method is a highly efficient ECU reflashing approach that combines the benefits of delta reprogramming with the enhanced speed and data handling capabilities of CAN FD (Controller Area Network – Flexible Data Rate) (S. Woo et al., 2016). In this method, only the modified portions of the ECU firmware—called delta segments—are identified and transmitted, rather than sending the entire flash image. This significantly reduces the data size, minimizing transfer time and lowering the risk of errors. When used over CAN FD, which supports payloads up to 64 bytes per frame and data rates up to 8 Mbps, the transmission of delta files becomes substantially faster and more reliable compared to classic CAN, which is limited to 8-byte frames and slower speeds (S. Woo et al., 2016). This combination is particularly valuable in field service or factory environments where multiple ECUs require rapid updates without full software overwrite. Delta + CAN FD also reduces ECU wear by avoiding unnecessary memory erasures, making it more hardware friendly. Additionally, by minimizing vehicle downtime during service operations, it enhances the overall user and technician experience. This method is increasingly adopted in OEM service tools and production lines, especially for powertrain and aftertreatment ECUs where frequent calibrations and software updates are needed.

**Delta CAN XL Reflash:** The DeltaCAN XL Reflash method represents a next-generation solution for high-speed, scalable ECU reprogramming, combining the minimal data footprint of delta reflashing with the advanced throughput and payload capacity of CAN XL. In this method, only the changed portions of firmware or calibration data are transmitted to the ECU, significantly reducing the overall size of the update package. This data efficiency is amplified by CAN XL's capability to transmit up to 2048 bytes per frame—far beyond the 64-byte limit of CAN FD—at speeds of up to 20 Mbps. The result is a dramatic reduction in reflash time, especially for larger ECUs such as ADAS, domain controllers, and electrified powertrain units (J. Laufenberg, 2023). Unlike classic CAN or even CAN FD, CAN XL is designed with future scalability in mind, offering built-in features like payload integrity checks, enhanced frame structures, and more efficient arbitration. When combined with delta reflashing, this allows for near real-time ECU updates across complex vehicle networks, with fewer frame transmissions and lower CPU overhead. It is particularly well-suited for centralized or zonal architectures, where multiple ECUs are connected to a high-bandwidth backbone. While Delta + CAN XL provides significant performance gains, it requires a completely new physical and protocol layer, meaning ECUs, gateways, and flashing tools must all support CAN XL. As a result, this method is currently most applicable to future vehicle platforms and OEM pilot projects, rather than legacy systems. However, as CAN XL adoption grows, this combination is expected to become a core enabler for fast, secure, and large-scale ECU reflashing, particularly in environments that demand high-speed OTA updates and minimal vehicle downtime.

**Field Service and Dealer Diagnostics:** In field service environments, such as dealer workshops or authorized service centers, Delta Reflash is primarily used to reduce service time while maintaining software consistency across the fleet. When an ECU requires an update—for example, to fix a software bug or apply a calibration enhancement—tools like Cummins INSITE, Ford FDRS, or GM GDS2 use delta files that contain only the changed calibration blocks or strategy modules. This minimizes data transfer over limited bandwidth diagnostic interfaces like CAN FD and reduces the flash time from 30+ minutes

to under 10 minutes in many cases. Field tools often integrate automated "Save and Restore" functionality to preserve important learned values during this process. Delta flashing in this context is often combined with compressed payloads and pre-validation routines to ensure compatibility before writing to the ECU.

#### Industry Use Cases and Benefits of Delta Reflash Methods:

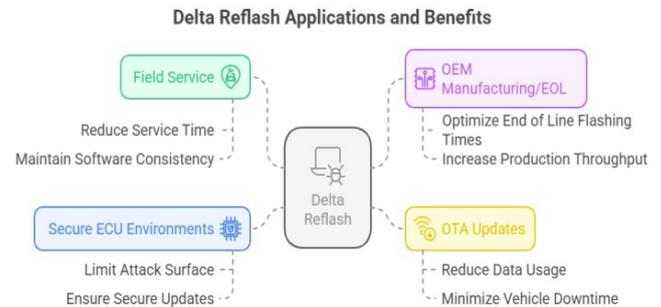


Fig. 2. Delta Reflash Application and Benefits

**OEM Manufacturing and End-of-Line (EoL) Programming:** In OEM production environments, where hundreds of vehicles are programmed daily on the factory floor, Delta Reflash is used to optimize end-of-line (EoL) flashing times. ECUs that are pre-loaded with base firmware at the supplier level may only need final configuration or calibration updates during vehicle assembly. Instead of performing a full reflash, OEMs like Volkswagen, BMW, or Hyundai use delta packages to update only the hardware-specific or customer-specific parameters (e.g., DO/SC options). Delta flashing is executed using high-speed protocols like DoIP or CAN FD, often in combination with parallel ECU flashing setups, allowing multiple ECUs to be programmed simultaneously. The resulting time savings per vehicle enable higher production throughput, reduced network congestion, and better factory-floor efficiency.

**OTA (Over-the-Air) Software Updates:** The use of Delta Reflash is especially critical in Over-the-Air (OTA) update scenarios, where vehicle connectivity is limited by network bandwidth, user patience, and safety constraints. OEMs like Tesla, Ford, and Volkswagen have adopted delta-based OTA frameworks where software is delivered wirelessly to vehicles using cellular or Wi-Fi connections. Instead of downloading full firmware images that can be hundreds of megabytes, the server computes a delta image between the installed and target versions, often compresses and encrypts it, and then transmits it securely to the vehicle. The in-vehicle update client decrypts and applies the delta using the ECU's bootloader, followed by validation and secure boot routines. This approach significantly reduces data usage, battery drain, and vehicle downtime, and is critical for fleet-wide software updates at scale.

**Secure ECU Environments (R155-compliant Systems):** In cybersecurity-hardened environments, such as ECUs compliant with UN Regulation R155, Delta Reflash is integrated with robust security features including encryption, digital signatures, and secure boot chains. Secure ECUs—such as engine controllers, ADAS modules, or telematics gateways—require all flash data, including delta segments, to be signed and verified before execution. The delta method is particularly beneficial here because it limits the attack surface by reducing the amount of memory being modified and minimizes the time the ECU remains in an unsecured programming state. These secure delta updates are often authenticated via PKI, with encrypted transfer protocols like TLS over DoIP or via a secure CAN gateway. The reflash tool or OTA client handles certificate validation, image decryption, and rollback support to ensure safe, reliable updates without compromising the ECU's trust anchors.

## RESULTS AND DISCUSSION

The integration of delta update technology within ECU reprogramming significantly enhances the efficiency and security of firmware updates, addressing the growing complexity of automotive

software systems. By minimizing data transfer and update durations, this method proves essential for maintaining optimal vehicle performance and compliance with evolving standards. As the automotive industry increasingly embraces interconnected and autonomous vehicles, the role of ECU reprogramming methods like Delta flashing will become even more critical in ensuring seamless vehicle operation and safety. The growing integration of advanced diagnostics and predictive maintenance systems necessitates that firmware updates not only enhance performance but also align with real-time data analytics to preemptively address potential failures. This synergy between ECU reprogramming and advanced diagnostics can lead to a more proactive approach in vehicle maintenance, reducing the likelihood of breakdowns and enhancing overall reliability. Furthermore, the challenges posed by cybersecurity threats in an interconnected environment highlight the importance of secure update methodologies, as the integrity of firmware is paramount in safeguarding vehicle systems against unauthorized access and manipulation. Thus, the evolution of ECU reflash techniques must continuously adapt to these emerging demands, ensuring that vehicles remain safe, efficient, and compliant with the latest technological standards.

**Limitation and Future Research:** While the advancements in Delta-based ECU reflashing offer significant benefits in terms of speed, efficiency, and scalability, several limitations remain that must be addressed for broader adoption. Firstly, Delta Reflash requires accurate tracking of existing ECU firmware versions and strict version management; mismatches between expected and actual ECU states can result in failed or incomplete updates. Secondly, not all ECUs support partial flash or memory segmentation, particularly in legacy or low-cost modules, limiting the applicability of delta techniques across the entire vehicle network. Additionally, security mechanisms such as encryption, secure boot, and digital signature validation can add overhead and complexity, especially when applied to highly fragmented delta payloads. The computational cost of decompression and decryption within constrained ECUs is another challenge, particularly in real-time systems with limited RAM or flash access bandwidth. For future research, there is a strong need to develop adaptive delta generation algorithms that can handle variable memory layouts and software modularity across ECU variants. Research into standardized delta packaging formats, compatible with AUTOSAR and UDS, could also streamline toolchain interoperability. Furthermore, exploring AI-assisted decision-making in determining when to apply delta vs. full reflash could optimize update strategies dynamically. There is also scope for advancing secure reflash protocols that combine delta delivery with post-update attestation, particularly in the context of R155 cybersecurity compliance. As ECUs evolve toward centralized domain controllers and zonal architectures, future reflashing approaches must support multi-core synchronization, cross-ECU dependency management, and OTA error recovery frameworks—all while maintaining real-time system integrity and functional safety compliance.

## CONCLUSION

The evolution of ECU reflashing strategies has been pivotal in meeting the growing demands of modern automotive systems—where speed, security, scalability, and adaptability are paramount. This paper presented a comprehensive analysis of advanced flashing techniques, emphasizing the role of Delta Reflash when combined with emerging transport layers and optimization strategies. Integrating Delta Reflash with CAN FD allows for faster updates in field and service environments by utilizing increased payload capacity and higher data rates, significantly reducing overall flash times. The combination of Delta and Parallel ECU Reflash enables simultaneous updates to multiple ECUs, greatly improving throughput in manufacturing and EoL programming.

The use of Delta with CAN XL introduces a future-ready approach for high-bandwidth applications, offering larger frame sizes and higher efficiency for complex ECUs like domain controllers and ADAS units. Moreover, Delta + DoIP has emerged as the preferred method for high-volume and high-speed updates, particularly in Over-the-Air (OTA) use cases, by leveraging Ethernet-based transfer and secure data channels. Delta with compression further enhances transmission efficiency by minimizing data size and network load, particularly in bandwidth-constrained environments. Across all scenarios, the structured use of Pre-Programming, Programming, and Post-Programming phases—combined with robust security mechanisms such as digital signatures, encryption, and secure boot validation—ensures a reliable and trustworthy update process. These strategies collectively allow OEMs and Tier-1 suppliers to deliver faster, safer, and more flexible ECU updates, aligning with regulatory standards like UNECE R155 (G. Costantino et al., 2022) and enabling seamless software lifecycle management across connected vehicle platforms. As automotive systems continue transitioning toward centralized and zonal architectures, these delta-based flashing approaches will remain foundational to achieving software-defined vehicle (SDV) goals.

## REFERENCES

- A. Prasad and P. Shanthi, "Automotive Electronic Control Unit Reprogramming Using Delta Method-A Review," *2021 Asian Conference on Innovation in Technology (ASIANCON)*, PUNE, India, 2021, pp. 1-6, doi: 10.1109/ASIANCON51346.2021.9544595.
- Bogdan, D., Bogdan, R., & Popa, M. (2016). Delta flashing of an ECU in the automotive industry. *Symposium on Applied Computational Intelligence and Informatics*. <https://doi.org/10.1109/SACI.2016.7507429>
- G. Costantino, M. De Vincenzi and I. Matteucci, "A Comparative Analysis of UNECE WP.29 R155 and ISO/SAE 21434," *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, Genoa, Italy, 2022, pp. 340-347, Doi: 10.1109/EuroSPW55150.2022.00041.
- ISO15765-3 2004 Road vehicles - diagnostics on controller area networks (CAN) - Part3: Implementation of unified diagnostic services (UDS on CAN). International Organization for Standardization
- J. Laufenberg, T. Kropf and O. Bringmann, "CAN Simulation Framework - From Classic CAN to CAN XL," *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, Bilbao, Spain, 2023, pp. 3343-3348, doi: 10.1109/ITSC57777.2023.10422342.
- Ji Zhang et al 2018 J. Phys. "Research on automotive ECU remote update and it's security" Conf. Ser. 1074 012133 DOI 10.1088/1742-6596/1074/1/012133
- 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, UK, 2019, pp. 1-5, doi: 10.23919/ICAC.2019.8895162.
- Onuma, Yutaka, Yoshiaki Terashima, and Ryoza Kiyohara. "Compression method for ECU software updates." *2017 Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*. IEEE, 2017.
- S. Woo, H. J. Jo, I. S. Kim and D. H. Lee, "A Practical Security Architecture for In-Vehicle CAN-FD," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2248-2261, Aug. 2016, doi: 10.1109/TITS.2016.2519464.
- Y. S. Lee, J. H. Kim, H. Van Hung and J. W. Jeon, "A parallel reprogramming method for in-vehicle gateway to save software update time," *2015 IEEE International Conference on Information and Automation*, Lijiang, China, 2015, pp. 1497-1502, doi: 10.1109/ICInFA.2015.7279522.