ISSN: 2583-9888
DOI: doi.org/10.51219/JAIMLD/veera-venkata-krishnarjun-rao-adabala/587

URF PUBLISHERS
connect with research world

Journal of Artificial Intelligence, Machine Learning and Data Science
https://urfpublishers.com/journal/artificial-intelligence

Vol: 1 & Iss: 1

Research Article

Plant Mode Software for Enhancing Vehicle Production Line Reliability in the Automotive Industry

Veera Venkata Krishnarjun Rao Adabala*

Citation: Krishnarjun Rao Adabala VV. Plant Mode Software for Enhancing Vehicle Production Line Reliability in the Automotive Industry. *J Artif Intell Mach Learn & Data Sci 2023* 1(1), 2802-2807. DOI: doi.org/10.51219/JAIMLD/veera-venkata-krishnarjun-rao-adabala/587

Received: 03 January, 2023; **Accepted:** 28 January, 2023; **Published:** 30 January, 2023

***Corresponding author:** Veera Venkata Krishnarjun Rao Adabala, USA, E-mail: adabalaveera2@gmail.com

ABSTRACT

Modern automotive manufacturing relies heavily on complex embedded software systems that govern a wide range of vehicle functions. As these systems become increasingly sophisticated, they also present challenges during the assembly process, particularly when software modules are incomplete, malfunctioning or undergoing final calibration. To address these issues and maintain uninterrupted production flow, manufacturers utilize Plant Mode Software a dedicated operational framework within the vehicle's control architecture. This mode enables essential functionality while bypassing or isolating faulty or non-critical components, allowing vehicles to continue through the production process with minimal disruption. This paper explores the architecture, role and design of plant mode systems in supporting production line reliability. It identifies typical functional failures encountered during manufacturing such as inter-module communication issues, sensor integration problems and configuration mismatches and discusses how plant mode provides mitigation pathways. Key implementation aspects, including integration with automotive safety (ISO 26262), software maturity models (Automotive SPICE) and security considerations, are examined. The paper also evaluates the advantages of plant mode, including reduced line stoppages and enhanced production efficiency, while acknowledging trade-offs such as increased software design complexity and testing overhead. This study aims to support automotive OEMs and suppliers in developing robust plant mode strategies for streamlined vehicle production.

Keywords: Plant mode software, automotive manufacturing, production line reliability, vehicle diagnostics, embedded systems, software faults, functional safety, ECU integration, diagnostic capabilities, fault handling, production downtime, software validation, cybersecurity, Industry 4.0, predictive diagnostics, modular software architecture, quality control, controller area network (CAN), machine learning, manufacturing execution system (MES)

1. Introduction

The evolution of automotive technology has transformed modern vehicles into highly sophisticated cyber-physical systems, with embedded software now serving as a core enabler of functionality, performance and safety. Today's vehicles integrate dozens of electronic control units (ECUs) that manage a wide array of subsystems ranging from powertrain and chassis control to infotainment and ADAS. As software complexity and interdependencies among these components increase, so too does the potential for integration-related issues during vehicle assembly. In a high-throughput production environment, even minor software faults such as a failed ECU flash, incomplete calibration data or a misconfigured communication protocol can cause production bottlenecks, requiring manual intervention or even halting the assembly line. These disruptions can significantly impact manufacturing efficiency, quality assurance and cost-effectiveness. Moreover, as the automotive industry shifts toward electrification and software-defined vehicles, the frequency and criticality of such disruptions are expected to grow. To address this challenge, automotive manufacturers have developed and deployed a specialized operational layer known as plant mode software. This mode represents a predefined software state that allows for basic, controlled operation of the vehicle or subsystem under test, even in the presence of incomplete or degraded software functionality. By activating plant mode, the production system can bypass non-critical faults, enable targeted diagnostics and support minimal viable vehicle behavior such as limited mobility or module initialization necessary to move the vehicle through key production stages. Plant mode software serves several strategic functions. It allows for fault isolation and containment, ensures that vehicles can be safely moved within the assembly plant and reduces the dependency on full-feature software maturity before start-of-production (SOP). Additionally, it supports parallel workflows, enabling rework or software updates to be performed at later production stages without disrupting upstream processes. This paper investigates the architecture, implementation and operational role of plant mode in vehicle production. Key objectives include identifying typical functional anomalies encountered on production lines, examining how plant mode software mitigates these issues and assessing its contribution to overall production line reliability. The paper also discusses relevant standards such as ISO 26262 (functional safety) and Automotive SPICE (process capability), which influence the design and validation of plant mode functions. Through this analysis, the study aims to provide practical insights for OEMs and Tier-1 suppliers seeking to enhance manufacturing resilience through software-based solutions **(Figure 1)**.

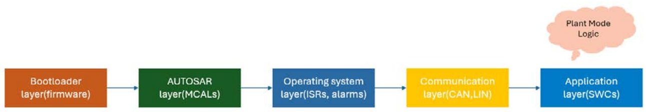

Figure 1: High level Architecture of Plant Mode Integration in ECUs.

2. Overview of Plant Mode in Automotive Software

In the context of automotive manufacturing, plant mode represents a purpose-built operational state embedded within a vehicle's software architecture. Unlike the standard driving or diagnostic modes intended for end-users or service technicians, plant mode is uniquely tailored for use within the controlled environment of the assembly plant. Its primary function is to support continuity of production workflows by allowing vehicles to be tested, moved and configured-even in the presence of incomplete, immature or faulty software modules.

A. Purpose and scope

Plant mode enables a minimal yet robust execution environment that emphasizes safety, control and diagnosability during the production process. It serves as a controlled fallback mechanism to prevent production line stoppages due to software issues. This is particularly important during pre-SOP (Start of Production) phases, where final software packages may still be undergoing integration or validation. By enabling limited but sufficient functionality, plant mode ensures that vehicles can progress through various stations on the production line without compromising safety or workflow timing.

B. Key features of plant mode

Plant mode software typically incorporates a modular and configurable set of features designed to accommodate varying stages of production and test requirements. These features may include:

- **Selective subsystem activation:** Enables or disables specific ECUs or software modules based on production-stage needs. For instance, non-essential systems like infotainment may remain inactive, while core safety and mobility functions are initialized.
- **Diagnostic data access:** Provides deep access to onboard diagnostic data, including DTCs, sensor outputs and communication bus statuses. This aids in rapid root-cause analysis of integration faults or hardware/software misalignments.
- **Initialization and calibration control:** Facilitates the staged execution of calibration routines e.g., for steering angle sensors or camera alignment which may be necessary only at specific assembly points.
- **Safety interlock overrides:** Allows temporary deactivation or simulation of certain safety conditions (e.g., seatbelt sensors, brake pedal state) for testing purposes, while ensuring compliance with safety protocols via access control and logging.
- **Real-time logging and reporting:** Collects and stores event logs, test results and execution metrics that are fed into production quality control systems for traceability and process optimization.

C. Embedded integration and architecture considerations

Plant mode is typically embedded within the bootloader or operating system layers of critical ECUs, ensuring that it is accessible during early software startup phases even when other modules are non-operational or have failed. The activation of plant mode can be triggered through configurable conditions such as CAN messages, physical switches or backend instructions via OTA tools. The implementation architecture may follow either a centralized model (where a master ECU governs plant mode behavior across subsystems) or a distributed model (where individual ECUs implement local plant mode logic). Each approach has implications for synchronization, communication and software validation.

D. Advantages in production context

By leveraging plant mode, OEMs can:

- **Prevent line stoppages:** Vehicles can be pushed through assembly even in a degraded software state, avoiding expensive delays.
- **Improve fault visibility early:** Issues that would otherwise go undetected until end-of-line (EOL) testing are identified and corrected in earlier stages.
- **Facilitate flexible workflows:** Plant mode allows deferred testing or updates post-assembly, supporting asynchronous manufacturing steps and software versioning.

E. Limitations and challenges

Despite its utility, plant mode adds a layer of complexity to software development and testing. Ensuring that plant mode is fully isolated from consumer-facing functionality is critical to avoid leakage into production use. Additionally, security mechanisms must be in place to prevent unauthorized access or misuse during and after production.

Table 1: Comparison of Normal Mode Vs Plant Mode.

Feature	Normal Operating Mode	Plant Mode
ECU Activation	Full	Selective
Diagnostic Access	Limited (User-Level)	Full (Factory-Level)
Safety Interlocks	Fully Enforced	Simulated
Fault Response	Fail-safe	Fault-tolerant
Vehicle Mobility	Full	Limited
Purpose	End-User Operation	Manufacturing Support

3. Functional Challenges in Vehicle Production

As vehicle systems become more reliant on complex, distributed software architectures, automotive production lines increasingly face functional issues that originate not from mechanical defects, but from software initialization failures, integration inconsistencies and electronic component misbehavior. These challenges, often arising during pre-final software stages or due to multi-supplier integration efforts, can significantly disrupt the efficiency and predictability of vehicle manufacturing.

A. Software initialization and flashing failures

One of the most frequent issues encountered during production is the failure of ECUs to initialize correctly, often resulting from incomplete or corrupted flash programming. Flash errors may stem from unstable power conditions during flashing, mismatched software-hardware versions or corrupted binaries. As vehicles often contain 50–100 ECUs, even a single module failing to boot properly can prevent downstream systems from initializing or communicating, leading to production halts. In some cases, ECUs enter a “bricked” state, requiring manual recovery or complete replacement. The lack of effective feedback mechanisms in traditional flashing processes further exacerbates the issue, making it difficult to isolate failures quickly.

B. Sensor misalignments and hardware variability

Modern vehicles integrate a wide array of sensors for powertrain, braking, steering and ADAS functions. During production, physical misalignments (e.g., of radar sensors, cameras or wheel speed sensors) can trigger fault conditions in the software. These are often misinterpreted as sensor failures rather than alignment issues, complicating diagnosis.

In addition, component-level variability such as signal drift in analog sensors, supplier-specific firmware differences or late-stage calibration deviations can cause systems to fail built-in self-tests, even if the hardware is not defective. This variability necessitates a flexible software environment that can tolerate minor discrepancies during assembly without blocking progression.

C. Communication bus errors

Inter-ECU communication relies on bus systems such as CAN, LIN, FlexRay or Automotive Ethernet. During production, bus errors can arise from loose connectors, incorrect wiring harness configurations or timing mismatches between communicating ECUs. For example, a late-responding module may cause timeouts, disrupting functional sequences such as powertrain initialization. In hybrid or electric vehicles, even momentary communication loss can prevent the inverter or BMS from reaching a safe operational state, halting production.

D. Safety system activation and line stoppages

Automotive safety systems such as airbag modules, ABS and ESC are designed to enforce strict fail-safe conditions. In a production environment, unintended activation or fault detection in these systems may result in the vehicle entering a failsafe mode, disabling key functions like drivability or steering assist. While this behavior is correct for consumer safety, it creates a bottleneck on the assembly line, requiring manual override or ECU reset procedures. Without software mechanisms like plant mode, the ability to bypass or suppress non-critical faults during controlled manufacturing stages is limited.

E. Troubleshooting complexity and time constraints

Traditional fault resolution on the production line relies heavily on manual inspection, offline diagnostic tools or error code scanning, which may not provide sufficient granularity for software-layer issues. Furthermore, many problems are transient or appear only during specific initialization sequences, making root-cause identification difficult. As production operates under strict takt times (time per vehicle at each station), delays of even a few minutes per vehicle can cascade into substantial downtime and increased operational costs. The complexity of diagnosing software-related issues in a high-speed, multi-variant production environment emphasizes the need for proactive, embedded mechanisms such as plant mode to isolate, diagnose and mitigate functional anomalies on the fly.

4. How Plant Mode Software Overcomes Functional Issues

Plant mode software serves as a dynamic mitigation layer in vehicle production environments, offering targeted mechanisms to manage and resolve functional issues that arise during assembly. By operating in a controlled, diagnostic-focused mode, plant mode helps manufacturers bypass transient software faults, stabilize vehicle behavior and maintain the pace of high-throughput production lines.

A. Enhanced diagnostic capabilities

One of the primary functions of plant mode is to provide real-time access to diagnostic data across vehicle systems. Traditional diagnostics may require full system initialization or access to service-level tools, which are not always feasible during early production stages. In contrast, plant mode embeds diagnostic interfaces directly within the ECU firmware, enabling:

- Live monitoring of ECU status, sensor inputs and actuator outputs.
- On-demand retrieval of DTCs, including historical and current fault states.
- Visibility into network traffic and communication statuses across CAN, LIN and Ethernet buses.

This granular access allows production engineers to pinpoint issues without the need for disassembly or off-line testing, significantly reducing mean time to repair (MTTR) and facilitating proactive fault resolution.

B. Controlled subsystem activation

To prevent cascading system failures, plant mode allows for selective enabling or disabling of subsystems, based on the production stage or fault context. For example, during initial assembly, non-essential systems like infotainment or HVAC can be deactivated to reduce complexity and power load. If a fault is detected in a safety-critical module such as the steering ECU, plant mode can:

- Isolate the module from the powertrain and braking systems.
- Enable limited operational modes, such as reduced-speed movement or diagnostic-only interfaces.
- Suppress false safety triggers that would otherwise stop production unnecessarily.

This form of fault containment ensures that the vehicle remains operable in a known, stable state, while engineering teams resolve the underlying issue offline.

C. Robust error handling and fault tolerance

Plant mode incorporates built-in error-handling logic that allows the vehicle to gracefully manage fault conditions without entering unrecoverable states. These mechanisms include:

Fail-silent and fail-operational modes, where non-critical functions are disabled while critical paths are maintained.

Predefined fallback routines for components such as electric drive systems, sensors and body control modules.

Recovery sequences triggered by software watchdogs or reset conditions, ensuring that faults do not propagate across modules.

These features are crucial for maintaining production line uptime, as vehicles can continue along the assembly path while flagged systems await resolution in downstream quality control stations.

D. Persistent data logging and analytics

Another key function of plant mode is the continuous logging of operational parameters, fault events and system transitions. These logs are stored either locally within ECUs or uploaded to a centralized manufacturing execution system (MES) or cloud-based analytics platform. This capability enables:

- Post-production failure trend analysis to identify recurring issues by part number, software version or supplier.
- Early detection of systematic integration errors across model variants or production batches.
- Closed-loop feedback into the software development and validation cycle, improving future ECU software robustness.

Log data also supports traceability an important requirement for compliance with quality standards such as IATF 16949 and functional safety frameworks like ISO 26262.

E. Production and quality assurance benefits

Through its integration in the embedded software architecture, plant mode contributes to:

- Reduced downtime by keeping vehicles in motion despite incomplete software readiness.
- Improved diagnostic accuracy, minimizing trial-and-error debugging.
- Higher first-pass yield at end-of-line testing due to early fault detection and isolation.
- Greater modularity in production, allowing for deferred updates or calibrations without rework.

5. Implementation Considerations

The successful deployment of plant mode software in an automotive manufacturing environment requires careful attention to both technical and operational parameters. Given its role in facilitating controlled vehicle behavior during production, plant mode must be designed to function robustly across diverse hardware platforms, comply with safety and cybersecurity regulations and integrate seamlessly into production workflows. This section gives overview for implementing an effective and reliable plant mode system.

A. Software architecture and configurability

A robust modular software architecture is foundational to plant mode implementation. The design must support:

- Configurable logic based on vehicle variants, production stations and feature sets.
- Scalability across multiple ECU platforms and powertrain types (e.g., ICE, hybrid, BEV).
- Layered abstraction, separating plant mode logic from consumer-facing functionality to prevent unintended interactions.

Modularity also enables targeted updates without affecting the broader system, which is essential in production where software changes must be traceable and isolated. Integration with existing middleware (e.g., AUTOSAR) or operating systems should be considered to maintain compatibility across suppliers.

B. Validation and testing strategies

Plant mode software must undergo rigorous validation under real-world production scenarios, including fault injection, partial ECU flashing and module misalignment simulations. Unlike consumer-facing features, plant mode functions must be tested under non-ideal and degraded system conditions, such as:

- Loss of network connectivity (e.g., CAN bus interruptions).
- Unavailable or corrupted configuration data.
- Delayed or missing ECU startup signals.

Validation strategies may include:

- Hardware-in-the-loop (HIL) testing to simulate ECU interactions and failures.
- End-of-line (EOL) test cases to verify plant mode activation and deactivation under various fault conditions.
- Trace-based analysis of plant mode transitions to ensure timing and dependency constraints are met.

Documentation of validation outcomes is also critical for compliance with Automotive SPICE capability levels and ISO 26262 safety requirements, particularly when plant mode interacts with safety-critical systems.

C. Hardware compatibility and integration

For plant mode to function effectively, it must integrate with a wide range of hardware components across the vehicle's electronic architecture. This includes:

- Ensuring ECU firmware supports plant mode hooks or bootloader entries.
- Compatibility with diagnostic communication protocols, such as UDS (Unified Diagnostic Services), to allow triggering and monitoring of plant mode status.

- Interfacing with factory test equipment, including flashing stations, robotic tools and MES platforms, to receive instructions and report status.

Because ECUs are often sourced from multiple suppliers, a standardized plant mode interface specification should be established across the vehicle platform to avoid integration inconsistencies.

D. Safety and cybersecurity requirements

Given the functional power of plant mode software particularly its ability to override normal behaviors and safety interlocks security and safety must be central to its design. Key considerations include:

- Access control mechanisms, such as password protection, hardware tokens or secure certificates, to prevent unauthorized plant mode activation.
- Role-based access during production, limiting certain commands to qualified operators or automated systems.
- Post-production lockout protocols that automatically disable plant mode capabilities after successful end-of-line validation, preventing misuse in the field.

From a safety perspective, plant mode must ensure that any overrides do not lead to unsafe vehicle states, even under controlled factory conditions. For example, brake overrides used for calibration must not disable actual braking functionality if a fault occurs. These constraints must be validated against functional safety guidelines (ISO 26262 ASIL levels).

E. Cross functional collaboration and process integration

The successful implementation of plant mode requires close collaboration between multiple engineering domains, including:

- Software developers, who define and code the logic based on plant requirements.
- Hardware engineers, who ensure electrical compatibility and diagnostic tool support.
- Production teams, who validate that plant mode aligns with real-world assembly line workflows and timing constraints.

Additionally, coordination with quality assurance, safety compliance and supply chain management is critical to ensure that all variants and component configurations are supported and tested.

To institutionalize this collaboration organizations may adopt cross-functional product teams or agile processes, such as Automotive SAFe (Scaled Agile Framework), to ensure that plant mode development evolves alongside changes in vehicle design and production planning.

6. Benefits and Limitations

While Plant Mode Software is increasingly recognized as a strategic enabler of robust and uninterrupted vehicle production, its adoption introduces both significant operational benefits and engineering trade-offs. This section explores these aspects in detail, providing a balanced evaluation of its impact on modern automotive manufacturing processes.

A. Benefits of plant mode implementation

- **Reduced production downtime and increased throughput:** One of the most critical advantages of plant mode is its ability to maintain production line continuity in the face of transient or non-critical software faults. By enabling vehicles to proceed through the manufacturing process without requiring immediate resolution of all system errors, plant mode significantly reduces line stoppages. This minimizes takt time variance and maximizes throughput, which is especially beneficial in high-volume facilities where even brief delays can cause cascading bottlenecks.
- **Early detection and resolution of software faults:** Plant mode supports real-time diagnostics and system monitoring, allowing production engineers to identify and address software-related issues much earlier in the manufacturing lifecycle. This capability reduces the reliance on end-of-line (EOL) inspections or post-production recalls, improving first-pass yield rates and reducing overall quality control costs. Moreover, by facilitating early root-cause analysis, plant mode contributes to more efficient validation of new ECU configurations or calibration data.
- **Enhanced quality control and traceability:** With features such as persistent logging, fault history recording and event tracking, plant mode contributes to enhanced traceability across the production line. This supports compliance with industry standards like IATF 16949, enabling manufacturers to audit software performance by batch, variant or station. Additionally, traceable logs provide vital feedback for continuous improvement processes and help close the loop between design, validation and manufacturing.
- **Flexibility across platforms and production lines:** Modern automotive production environments involve a multitude of vehicle variants, powertrain architectures (ICE, hybrid, BEV) and regional compliance configurations. Plant mode's modular design and configurability allow it to be tailored for diverse platforms and plant setups. Whether deployed in a traditional assembly plant or a software-focused EV gigafactory, plant mode enables consistent diagnostic and functional behavior across different ecosystems, enhancing platform scalability and reuse.

B. Limitations and challenges

- **Increased software complexity:** The integration of plant mode adds another operational state to the vehicle's software lifecycle, increasing the overall complexity of the ECU firmware. Developers must manage multiple software pathways, ensure isolation between plant and normal operating modes and validate all possible transition conditions. This necessitates additional development effort, more exhaustive test cases and extended validation timelines, particularly when functional safety (ISO 26262) is involved.
- **Security vulnerabilities and access control risks:** If not properly secured, plant mode presents a potential attack vector. Unauthorized access especially post-production could allow malicious actors or unqualified personnel to override safety-critical behaviors or disable essential functions. Security concerns are especially pronounced in vehicles with remote diagnostics or OTA update capabilities, where plant mode may be triggered without physical access. Implementing secure activation protocols, such as cryptographic authentication, access logging and hardware-based trusted platform modules (TPMs), is essential to

mitigate this risk.

- **Continuous maintenance amid evolving architectures:** As automotive software stacks evolve e.g., transitioning toward centralized computing, service-oriented architectures and domain controllers plant mode software must continuously adapt. Each new platform introduces updated protocols, hardware dependencies and diagnostic requirements, requiring plant mode implementations to be regularly updated and revalidated. This results in an ongoing resource commitment across the vehicle development lifecycle.
- **Mitigation strategies:** To balance these challenges, manufacturers should adopt a combination of technical and organizational best practices, including:
- **Secure access control:** Implement layered security mechanisms for plant mode activation, including digital signatures, role-based authentication and automatic deactivation post-EOL testing.

Continuous Integration and Validation (CI/CV): Integrate plant mode testing into CI pipelines, with automated fault injection, regression testing and hardware-in-the-loop (HIL) simulations.

Cross-functional Training: Ensure production staff, quality assurance personnel and software engineers are trained on the operation, safety constraints and diagnostic use of plant mode to prevent misuse or misinterpretation.

7. Conclusion and Future Work

A. Conclusion

Plant mode software constitutes a critical enabler for modern automotive manufacturing, effectively bridging the gap between complex embedded vehicle systems and the practical demands of high-volume production environments. By embedding specialized diagnostic, control and fault-tolerant capabilities directly within the vehicle software architecture, plant mode facilitates rapid identification and mitigation of functional anomalies that would otherwise disrupt assembly line operations. This functionality ensures minimal production downtime, enhances throughput and supports stringent quality assurance objectives ultimately contributing to the reliable delivery of vehicles that meet both safety and performance standards.

The adoption of plant mode represents a shift toward more adaptive and resilient manufacturing workflows, where software plays an active role in dynamically adjusting to hardware and integration variances. This adaptability is increasingly important as vehicles become more software-intensive, incorporating ADAS, electrification technologies and interconnected components. The modular and configurable nature of plant mode software makes it suitable for diverse vehicle platforms and production configurations, underscoring its value as a scalable solution in the evolving automotive landscape.

B. Future work and research direction

- **Integration of machine learning for predictive diagnostics**: Future iterations of plant mode can leverage ML and AI algorithms to transition from reactive fault detection to predictive diagnostics. By analyzing large volumes of sensor data, communication logs and fault histories collected during production, ML models can identify subtle patterns and early indicators of potential failures. This capability would enable preemptive interventions, reduce unplanned stoppages and optimize maintenance schedules, thus further increasing production efficiency.
- **Enhanced cybersecurity frameworks:** As vehicles become more connected and plant mode interfaces potentially accessible via remote or networked tools, strengthening cybersecurity defenses is paramount. Research into lightweight cryptographic protocols, secure boot mechanisms and IDS specifically tailored for plant mode contexts will be essential. These measures will safeguard against unauthorized access, tampering and cyber-physical attacks during manufacturing and throughout the vehicle lifecycle.
- **Seamless integration with industry 4.0 ecosystems:** The ongoing evolution toward Industry 4.0 production systems, characterized by IoT-enabled equipment, real-time data analytics and autonomous manufacturing lines, opens new avenues for plant mode enhancement.

This integration promises to create fully adaptive, self-optimizing production environments where plant mode plays a pivotal role in orchestrating vehicle diagnostics and control within a holistic digital manufacturing ecosystem.

8. References

1. Kiencke JW, Nielsen L. Automotive Control Systems: For Engine, Driveline and Vehicle, 2nd ed., Springer, 2005.
2. Gries T, Schiele H, Hube R. Diagnostics and Functional Safety in Automotive Software Development. SAE Technical Paper, 2016.
3. AUTOSAR Consortium. AUTOSAR Release 4.2 Specification, 2016.
4. ISO 26262, Road vehicles - Functional safety, International Organization for Standardization, 2011.
5. Steuer F, Knoll A, Lienkamp M. On the Implementation of Fail-Safe Mechanisms for Automotive ECU Software in Proc. IEEE Int. Conf. Vehicular Electronics and Safety, 2014: 21-26.
6. Anwar S, Kyamakya K. Software Diagnostics in Automotive Manufacturing in Proc. IEEE Int Conf. Industrial Informatics, 2017: 758-763.
7. Elsayed SM, Irwin MJ. Quality Control in Automotive Manufacturing Using Software Logging. IEEE Trans. Automation Science and Engineering, 2013;10(2): 382-393.
8. Zeng M, et al. CAN Bus Communication in Automotive Systems: Protocol and Security. IEEE Commun Surveys & Tutorials, 2018;20: 1737-1766.