Academic Journal of Emerging Technologies
ISSN Print: 3104-4417
ISSN Online: 3104-4425
CD ISSN: 3104-4433
https://doi.org/10.63313/AJET.9049

Collaborative Design Strategies for Electronic and Electrical Architectures in the Intelligent Driving Domain

Wei Miao
School of Automotive Engineering, Suzhou Vocational Institute of Industrial Technology, Suzhou, 215104, China
Email: szmiaowei@163.com

How to cite this paper: Miao, W. (2026). Collaborative design strategies for electronic and electrical architectures in the intelligent driving domain. Academic Journal of Emerging Technologies, 2(3), 77 - 88. ISSN Print: 3104-4417; ISSN Online: 3104-4425.
https://doi.org/10.63313/AJET.9049
Published: 2026-04-30

Abstract

The evolution of intelligent driving technology toward higher levels imposes new demands on electronic and electrical architectures, requiring high computing power, high real-time performance, and high functional safety levels. Traditional development models face challenges such as a disconnect between architecture and functionality, inefficient cross-team collaboration, and lengthy validation cycles. This paper focuses on the intelligent driving domain and establishes a three-tier collaborative design strategy system comprising the application layer, service layer, and hardware layer. Through layered decoupling, it achieves separation of software and hardware as well as standardized interaction between domains; through model-driven approaches, it supports cross-team parallel development; and through digital twins, it advances verification and enhances scenario coverage. A development project for a Level 2+ highway pilot assist function at a certain automaker demonstrates that this framework can shorten the development cycle by 25%, improve validation efficiency by 70%, and achieve a product safety score of 94%, providing a reusable methodological foundation for the development of intelligent driving electronic and electrical architectures.

Keywords

Intelligent Driving Domain; Electronic and Electrical Architecture; Collaborative Design; Hierarchical Decoupling; Digital Twin

1. Introduction

This template, The accelerating pace of automotive intelligence is driving the transition of autonomous driving technology from Level 2 to Levels 3 and 4, with the electronic and electrical architecture emerging as the core bottleneck hindering its implementation. In 2024, the Ministry of Industry and Information Technology (MIIT) launched a pilot program for the market access of intelligent connected vehicles, marking the official entry of Level 3 and 4 autonomous driving into the commercialization phase and imposing even stricter requirements on vehicle electronic and electrical architectures. The complexity of intelligent driving functions has surged dramatically. Modules such as multi-sensor fusion, real-time decision-making and planning, and vehicle dynamics control are deeply coupled, presenting three major challenges in terms of computing power, communication bandwidth, and functional safety: high-performance heterogeneous computing, high-reliability communication, and high functional safety levels. Traditional distributed architectures follow a "one function per ECU" model, but advanced levels of intelligent driving require the integration of dozens of controllers, leading to a sharp increase in architectural complexity and making sustained iteration difficult.

The current development model suffers from structural contradictions. Architecture teams predefine hardware topologies based on experience, while algorithm teams develop functions within fixed frameworks. The lack of real-time collaboration mechanisms between these teams leads to distorted requirement communication, slow response to changes, and prolonged validation cycles. Typical issues include inaccurate computing power assessments causing performance bottlenecks in later stages, sensor interface mismatches leading to hardware revisions, and hardware-software integration problems surfacing during vehicle testing, resulting in significant rework costs. Existing research primarily focuses on generalized analyses of vehicle-wide electronic and electrical architectures, lacking specialized strategies tailored to the high-computational-power, high-real-time, and high-safety requirements of the intelligent driving domain. Research on deep collaboration mechanisms between architecture and functionality is insufficient, and the linkage between digital twins and the architecture design phase remains unclear. Therefore, establishing a collaborative mechanism between architecture design and functional development to achieve real-time alignment and parallel iteration of requirements, architecture, and implementation is an inevitable choice for improving development efficiency and reducing technical risks.

This paper focuses on the specific functional domain of intelligent driving, analyzes the unique requirements of its electronic and electrical architecture, dissects the core pain points of traditional development models, and proposes three collaborative strategies: layered decoupled architecture design, model-driven collaborative development, and rapid verification using digital twins. The paper validates the effectiveness of these strategies through typical case studies, providing methodological support for the development of electronic and electrical architectures in the intelligent driving domain.

2. Analysis of Requirements and Challenges for the Electronic and Electrical Architecture of the Intelligent Driving Domain

2.1. Technical Characteristics and Architectural Requirements of the Intelligent Driving Domain

The intelligent driving domain is the core functional domain responsible for environmental perception, decision-making and planning, and motion control, and its technical characteristics vary significantly across different levels. Sensor configurations and computing power requirements vary significantly across different levels. Level 2 ADAS typically includes 1 millimeter-wave radar and 1 camera, with a computing power requirement of 2 to 10 TOPS; Level 2+ highway piloting systems include 5 millimeter-wave radars and 5 to 12 cameras, with a computing power requirement of 30 to 100 TOPS; Level 3 urban piloting systems are configured with 5 millimeter-wave radars, 12 cameras, and 1 LiDAR or more, requiring 200 to 500 TOPS of computing power; Level 4 Robotaxi systems are configured with multiple LiDARs and high-performance computing platforms, requiring over 1,000 TOPS of computing power.

A comparison of mainstream automakers' architecture solutions shows that Tesla's HW 4.0 uses a centralized computing architecture with approximately 300 TOPS of computing power and a sensor configuration of 8 cameras plus 1 radar; Li Auto's AD Max uses dual Journey 5 domain controllers with 256 TOPS of computing power and a sensor configuration of 5 millimeter-wave radars, 12 cameras, and 1 LiDAR; Xpeng's XNGP uses dual Orin-X chips, delivering 508 TOPS of computing power, with a sensor configuration of 5 millimeter-wave radars, 12 cameras, and 2 LiDAR units. Domestic solutions offer advantages in terms of cost-effectiveness and localization, but the maturity of their toolchains and the completeness of their ecosystems still lag behind international manufacturers such as NVIDIA.

The intelligent driving domain imposes four core requirements on the electronic and electrical architecture. Regarding computing power, a heterogeneous architecture must be adopted, comprising AI-specific chips (NPUs or GPUs), general-purpose processors (CPUs), and safety controllers (MCUs). AI computations account for over 70% of the total, and 30% redundancy must be reserved to support algorithm iterations. For communication requirements, the raw data bandwidth for 8MP cameras ranges from 8 to 16 Gbps, while LiDAR point cloud data ranges from 0.5 to 1 Gbps. Multi-sensor time synchronization accuracy must be less than 1 millisecond, and support for TSN (Time-Sensitive Networking) is required to ensure deterministic transmission. In terms of functional safety, the system must meet the highest level, ISO 26262 ASIL-D, and implement a defense-in-depth approach that includes perception algorithm failure detection, decision output monitoring, and end-to-end protection of control commands. For continuous iteration, the OTA cycle for algorithms has been shortened to monthly intervals; the architecture must support hardware-software decoupling, A/B partitioned upgrades, and closed-loop data feedback.

2.2. Core Pain Points of Traditional Development Models

Traditional development models exhibit three structural flaws when addressing the above requirements. Architecture definitions are disconnected from functional requirements; architecture teams rely on experience to pre-set computational power (e.g., reserving a 30% margin) without quantitative assessment methods for actual algorithm demands. This leads to inaccurate computational power evaluations, mismatched sensor interfaces, and memory bandwidth bottlenecks. For one vehicle model, insufficient computational power for perception algorithms in complex scenarios was discovered prior to mass production, forcing a reduction in image resolution that compromised the user experience. For another model, inconsistencies between the GMSL2 camera interface and the FPD-Link domain controller interface required the addition of an adapter board, extending the development cycle by 2 to 3 months. In yet another case, a DDR memory bandwidth bottleneck during concurrent processing of multi-sensor data arose because data flow simulation was not performed during the architectural design phase, leaving limited room for optimization later on.

Cross-team collaboration is inefficient; requirements are primarily communicated via Word documents rather than executable models. The negotiation process for interface definitions is lengthy, and progress discrepancies lead to situations where algorithm development is completed only to wait for hardware availability or, after hardware is frozen, for algorithm requirements to change. When troubleshooting issues, software and hardware teams often pass the blame onto each other. In one project, the algorithm team and the architecture team had differing interpretations of real-time requirements. The architecture team believed that a 20-millisecond CAN communication latency met the requirements, while the algorithm team actually required 10 milliseconds. This discrepancy was not revealed until the integration testing phase, resulting in software rework and project delays.

The verification phase was delayed and lacked sufficient coverage. Real-vehicle testing was constrained by the difficulty of scenario setup and lengthy cycles; reproducing a single hazardous scenario, such as a "cut-in," required extensive road testing, making it difficult to cover extreme weather and edge cases. Hardware and software integration issues exposed during the real-vehicle phase resulted in massive rework costs. During road testing of a certain vehicle model, insufficient thermal management of the domain controller was discovered, leading to thermal throttling. This required redesigning the housing and airflow channels, delaying mass production by six months and incurring rework costs exceeding 10 million yuan.

2.3. Core Pain Points of Traditional Development Models

To address these pain points, the development of electronic and electrical architectures for intelligent driving domains urgently requires the establishment of four mechanisms. A mechanism for ensuring requirement consistency guarantees that algorithmic functional requirements are communicated to the architecture design in real time and with accuracy; a mechanism for predictable resource assessment evaluates the support level of architectural solutions for functional implementation at an early stage; a rapid response mechanism to changes supports the swift coordination of algorithm iterations and architectural adjustments; and a mechanism for forward-looking and closed-loop verification moves verification activities to the design phase and enhances efficiency through virtual simulation.

3. Collaborative Design Strategies for the Intelligent Driving Domain

3.1. Overall Framework for Three-Layer Collaborative Design

This paper proposes a three-layer collaborative design strategy framework comprising the application, service, and hardware layers. The application layer implements perception, decision-making, planning, and control algorithms; the service layer provides basic services such as communication, data management, security, and diagnostics; and the hardware layer provides computing, storage, communication, and sensor/actuator resources. Interaction between layers is achieved through standardized interfaces, separating functional logic from hardware implementation and supporting cross-platform algorithm portability. These three types of strategies mutually reinforce one another. Layered decoupling provides modeling boundaries and interface definitions for model-driven development, while model-driven development provides executable algorithmic models for the digital twin. Verification results from the digital twin are fed back into the architecture and algorithm optimization, forming a closed-loop system.

3.2. Layered Decoupling Architecture Design Strategy

This strategy divides the intelligent driving domain architecture into four layers. The application layer implements algorithm modules such as multi-sensor fusion perception, high-definition map positioning, behavioral decision-making, path planning, and motion control. Perception algorithms primarily use BEV combined with Transformer networks, while decision-making and planning employ a hybrid approach of rules and deep learning. The service layer adopts a hybrid architecture combining AUTOSAR Adaptive and ROS2, providing DDS communication services to support QoS policies and real-time performance guarantees; a sensor data bus to ensure a unified data format and time synchronization management; functional monitoring and encryption security services to monitor algorithm outputs, detect anomalies, and enable TLS or DTLS secure communication; and UDS diagnostic services to manage fault codes, enable remote diagnostics, and support OTA updates. The base software layer utilizes a dual-system architecture with QNX for safety-critical tasks and Linux for AI computing tasks; Hypervisor virtualization enables resource isolation; and BSP hardware abstraction encapsulation supports cross-platform portability by accommodating chip variations. At the hardware layer, the main computing chip uses a heterogeneous SoC such as the Journey 5, which includes a 4-core A55 CPU, a dual-core BPU, and a GPU. The security controller is deployed independently to meet ASIL-D requirements. The network interface supports TSN with a synchronization accuracy of less than 1 microsecond, while the sensor interface enables synchronized triggering and fault detection for 8 GMSL2 cameras.

Function deployment within the domain follows the principle of mapping computational tasks. AI perception and inference are deployed on the SoC's NPU or GPU to leverage high-parallel computing capabilities; decision-making and planning algorithms run on the SoC's CPU big cores to meet complex logic and floating-point computation requirements; functional safety monitoring is independently deployed on the MCU to achieve physical isolation, hard real-time response, and fault self-diagnosis; vehicle control commands are output via the MCU to the chassis domain to meet ASIL-D requirements and ensure end-to-end protection. Sensor data stream optimization employs zero-copy technology to transmit data directly to the NPU via shared memory, reducing CPU intervention; time synchronization uses the gPTP protocol to ensure multi-sensor data timestamp errors are less than 1 millisecond; bandwidth management reduces raw data bandwidth through ISP preprocessing and applies region-of-interest (ROI) encoding to critical areas.

Regarding inter-domain communication, the Intelligent Driving domain and the Chassis domain exchange control commands and status feedback via dual channels—redundant CAN FD and Ethernet. Control commands include target steering angle or steering torque, target acceleration, or braking force, with latency controlled within 20 milliseconds; end-to-end (E2E) protection is employed to meet ASIL-D requirements. Chassis status feedback includes actual steering angle, wheel speed, and yaw rate, with a latency of less than 50 milliseconds to meet ASIL-B requirements. The interaction with the body domain is based on a service-oriented architecture, using the SOME/IP protocol to enable bidirectional transmission of human-machine interaction status and vehicle status, supporting driver authority handover decisions and functional activation condition assessments.

3.3. Model-Driven Collaborative Development Mechanism

This strategy establishes a unified modeling environment based on SysML, AUTOSAR, and MATLAB/Simulink. The architecture team uses AADL or EA for modeling software components and hardware topology; the algorithm team uses Simulink for modeling perception, decision-making, and planning algorithms; both teams utilize the FMI (Functional Mock-up Interface) standard to facilitate model exchange and co-simulation. The FMI implementation process includes exporting Simulink algorithm models as FMU functional models (i.e., C code) and exporting AUTOSAR architecture models as ARXML software configuration data. Joint simulation platforms, such as CarSim with VTD, load the FMU and ARXML files to perform integrated simulation verification.

The collaborative workflow is divided into four phases. The conceptual design phase (Weeks T0 to T4) involves the architecture team conducting a comprehensive evaluation based on the computational requirements model provided by the algorithm team: first, using the TensorRT Profiler to perform performance profiling of each operator to obtain the computational load and memory access volume for each layer; second, employing a roofline model to analyze computational power and bandwidth bottlenecks to determine the theoretical performance upper bound; and finally, verifying DDR bandwidth and latency through data flow simulation under typical scenarios. Based on the above three-pronged evaluation, chip selection is finalized, and both parties reach consensus through a requirements alignment review meeting; the detailed design phase (Weeks T4 to T12) involves both teams working in parallel on software architecture and algorithm modeling. The architecture team designs the software component architecture, communication matrix, and deployment plan, while the algorithm team completes Simulink modeling and detailed interface design, with alignment achieved through a model integration review meeting; The virtual validation phase (Weeks T12 to T16) involves early integration and debugging using a V-ECU. The algorithm team provides test cases and expected outputs, with 80% of algorithm debugging completed before hardware arrival. The hardware-in-the-loop (HIL) phase (Weeks T16 to T20) verifies the real-time performance of algorithms on the target hardware. Following successful HIL acceptance testing, the project proceeds to in-vehicle integration.

Change management enables automated impact analysis. When a requirement change—such as the addition of nighttime scene recognition—is triggered, the system automatically identifies affected algorithm modules, performs computational power assessments, and determines interface changes. Relevant personnel are notified via the ALM platform to synchronize updates, reducing the average response time from 5 days in the traditional model to 1 day. Version control enables traceability between architecture and algorithm model versions, while model baseline management supports change diff comparisons. Code reviews and architecture specification checks ensure consistency.

3.4. Rapid Validation Method for Digital Twins

This strategy establishes a five-layer digital twin system encompassing the scenario layer, environment layer, sensor layer, vehicle layer, and algorithm layer. The scenario layer builds a library of millions of scenarios based on real-world driving data and regulatory requirements, including a real-world driving scenario library based on millions of kilometers of data, a hazardous scenario library (e.g., cut-in and emergency evasion), a regulatory scenario library (e.g., C-NCAP test procedures), and virtual scenarios generated using adversarial generative networks for edge cases; the environment layer constructs static and dynamic environments using high-definition maps and traffic flow simulation (e.g., SUMO or VISSIM), and implements weather and lighting simulations based on physical rendering in Unreal or Unity; The Sensor Layer establishes camera twins, including distortion models, ISP models, HDR models, and LED flicker models; radar twins, including RCS models, multipath effects, and resolution models; and LiDAR twins, including point cloud generation models, reflectivity models, and rain/fog attenuation models; The Vehicle Layer deploys CarSim high-precision vehicle dynamics models, virtual ECU (V-ECU) instruction-level simulation, and cache and memory models; the Algorithm Layer implements SIL simulation of perception, decision-making, and planning algorithms.

A phased validation strategy integrates validation activities throughout the entire development process. During the architecture design phase, network simulation (e.g., RTaW or CANoe) is used to evaluate topology performance. In one project, a 15-millisecond delay at the camera data aggregation node was found to exceed the threshold; after optimizing the topology, the delay was reduced to 8 milliseconds. During the algorithm development phase, tens of thousands of scenarios are tested via SIL simulation. In another project, a scenario library was built using 100,000 kilometers of real-world driving data to test 20,000 scenarios, identifying 120 issues, 30 of which were edge cases difficult to reproduce in actual vehicles; During the fault injection phase, 50 types of sensor failures and communication anomalies were simulated to validate the effectiveness of safety mechanisms. This included camera obstruction simulated via image region masking to mimic raindrops, mud, and strong light; radar false alarms simulated via point cloud noise injection and false target injection; and LiDAR failure simulated via point cloud data interruption and sudden resolution drop. Time synchronization failures were simulated via timestamp shifts and jumps, achieving a fault detection rate of 100% and a false positive rate of less than 2%; the in-vehicle testing phase focused on high-risk scenarios and long-tail issues identified through digital twin screening, reducing the test mileage from the planned 80,000 kilometers to 30,000 kilometers.

The digital twin model's accuracy calibration employs a real-vehicle data feedback mechanism. By collecting real-vehicle sensor data and algorithm outputs via shadow mode, comparing twin predictions with actual performance, and calibrating model parameters using least squares and Kalman filtering, the error between simulation results and real-vehicle performance is kept below 5%. This strategy reduces the single-iteration algorithm validation cycle from 2–3 weeks to 2–3 days and shifts the architecture defect detection phase from the traditional in-vehicle testing stage to the detailed design stage, advancing the process by approximately 6–8 months compared to the traditional model.

4. Case Study and Performance Evaluation

4.1. Project Background and Constraints

This paper uses a Level 2+ highway pilot-assisted driving feature development project by a domestic automotive manufacturer as a case study. The project's time-to-production-from initiation to SOP—was 18 months, a 25% reduction compared to the industry average of 24 months. The hardware platform utilizes the domestically produced Horizon Journey 5 chip with 128 TOPS of computing power, and the sensor configuration consists of 5 millimeter-wave radars and 12 cameras, without LiDAR; The functional scope covers automatic lane changing, automatic overtaking, and automatic on- and off-ramp maneuvers in highway scenarios; the system must also achieve a five-star safety rating in the 2024 C-NCAP certification. Project challenges included the lower maturity of domestic chip toolchains compared to international vendors such as NVIDIA, efficiency pressures resulting from a 25% reduction in the development cycle, and technical risks associated with the high safety requirements of highway scenarios.

4.2. Application of Collaborative Design Strategies

In terms of the application of a layered decoupled architecture design strategy, the project adopts a single Journey 5 SoC combined with an independent safety MCU architecture. Perception algorithms are deployed on the SoC's NPU, accounting for 70% of the computing power, while functional safety monitoring is deployed on the MCU to achieve physical isolation and meet ASIL-D requirements. The algorithm team participated in camera selection-specifically an 8-megapixel front-facing main camera and a 3-megapixel side-view camera-to ensure ISP performance met nighttime detection requirements. Based on algorithm model simulations, computational power allocation was determined as 70% for perception, 20% for decision-making, and 10% for redundancy to avoid overdesign. The chassis team was involved early in interface definition to ensure control command response times were less than 20 milliseconds. Key collaboration milestones include the M1 phase (Weeks T0-T4), during which the algorithm team provided a TensorRT Profiler computational demand model to finalize chip selection; and the M3 phase (Weeks T4-T12), during which a joint review optimized the sensor data flow architecture and adjusted DDR bandwidth allocation to eliminate data bottlenecks, changing the original single-channel aggregation to dual-channel aggregation and reducing bandwidth utilization from 85% to 60%.

Regarding the application of model-driven collaborative development strategies, the project utilized MATLAB/Simulink, dSPACE SYSTEMA, and a proprietary requirements management platform to build a unified modeling environment. Hardware and perception algorithms were developed in parallel, with early integration via V-ECU enabling 80% of algorithm debugging to be completed before hardware was available. The OEM architecture team and Horizon's chip team established a joint technical support group comprising 12 members from both sides-including architects, algorithm engineers, and toolchain experts-ensuring model issues were addressed within 24 hours and reducing the algorithm porting cycle from an estimated 6 weeks to 3 weeks. A typical collaborative scenario occurred during the mid-project phase (Week T8), when the perception algorithm team proposed upgrading the front-facing camera resolution from 2 megapixels to 8 megapixels. Through rapid evaluation using the unified model, the algorithm team updated the Simulink model and verified an improvement in detection accuracy—specifically, the recognition rate for small targets at long distances increased from 75% to 92%. The architecture team simultaneously assessed data bandwidth and confirmed that the upgraded single-channel 4 Gbps interface (previously supported by GMSL2) had sufficient headroom to accommodate 6 Gbps. Both teams confirmed within two days that only software configuration adjustments were needed, with no hardware redesign required, thereby avoiding the three-month hardware redesign cycle and millions of yuan in rework costs associated with traditional approaches.

Regarding the application of the digital twin rapid validation strategy, the project utilized RT-over-Wi-Fi (RTaW) communication network simulation to optimize topology and latency. A SIL simulation environment was established, and a scenario library was built using 100,000 kilometers of real-world driving data to complete testing across 20,000 scenarios. Fifty failure modes were simulated to verify the effectiveness of safety monitoring mechanisms. The digital twin identified high-risk scenarios to guide real-vehicle test route planning, reducing the required real-vehicle test mileage to 30,000 kilometers. The digital twin development project required an investment of 33 million yuan, including 20 million yuan for the scenario library, 5 million yuan for sensor modeling, and 8 million yuan for simulation platform deployment. The investment was recouped with the first vehicle model. The verification cycle for individual functions was shortened by 60%, saving approximately 40 million yuan in time costs; optimized sensor selection reduced BOM costs by approximately 3 million yuan per vehicle; and the reduction of five vehicles in the physical testing fleet saved approximately 10 million yuan in testing expenses.

4.3. Quantitative Assessment of Implementation Results

The project achieved significant results across three dimensions. In terms of development efficiency, the project cycle of 18 months was 25% shorter than the industry average; the algorithm iteration and validation cycle of 3 to 5 days was 70% shorter than the traditional 2 to 3 weeks; the response time for architectural changes of 1 to 3 days was 80% shorter than the traditional 1 to 2 weeks; cross-departmental issue coordination meetings were held once a week, a 70% reduction from the traditional 3 to 5 times per week; and the joint project team adopted daily 15-minute stand-up meetings to replace traditional lengthy meetings.

In terms of product quality, the availability of the highway piloting feature reached 97.5%, exceeding the 95% target; the system failure rate was 0.3 per 1,000 kilometers, lower than the 0.5 target; the C-NCAP active safety score of 94% earned a Five-Star Plus rating; and the user complaint rate six months after launch was 3.2%, below the 5% target. In a horizontal comparison with industry benchmarks, Tesla's FSD development cycle is approximately 12 months but relies on a mature toolchain, while Xpeng's XNGP development cycle is approximately 20 months. This project achieves mass production in 18 months within the domestic chip ecosystem, demonstrating a comparative advantage. In terms of collaboration capabilities, an assessment based on the CMMI-DEV model achieved Level 3 maturity, with requirements development and management, technical solutions, verification and validation, and process quality assurance all reaching Level 3, and configuration management reaching Level 2.

5. Conclusions

This paper addresses the core challenges in developing the electronic and electrical architecture for the intelligent driving domain. It identifies the specific requirements of this field in terms of computing density, communication real-time performance, and functional safety levels, and establishes a three-tiered collaborative design strategy framework comprising application, service, and hardware layers. Through validation in a Level 2+ project at an automotive manufacturer, the approach achieved a 25% reduction in the development cycle, a 70% increase in validation efficiency, and a product safety score of 94%. The innovation of this paper lies in its focus on developing specialized, layered decoupling strategies for the intelligent driving domain, proposing a cross-team parallel development process based on a unified model, and integrating the digital twin into the architecture design phase to form a closed-loop system. The limitations of this study include the need to expand the representativeness of the case studies and to deepen the technical analysis. Future directions include the evolution of collaborative design for central computing architectures, AI-driven intelligent development platforms, closed-loop optimization of digital twins and real-vehicle data, and the establishment of cross-industry standardization.

References

[1] Guo, D,S. Li, H. and Han, M.C.(2026) A Study on Bandwidth Allocation Strategies for In-Vehicle Passive Optical Networks. Automotive Engineering, 48 (03): 598-606

[2] Qin, X.Y.(2026) A Collaborative Method for Optimizing Automotive Electronic and Electrical Architectures and Functional Design. Volkswagen, 01: 31-33.

[3] Guo, C.Y. Chen, H and Zhou. K.(2026) Design of a High-Power Adaptive Power Supply System Based on an Autonomous Driving Domain Controller. Journal of Chongqing Technology and Business University (Natural Science Edition), 12:1-14

[4] Tian, P.F. (2025) Research on Online Monitoring Strategies for In-Vehicle Controllers Based on the AUTOSAR Specification. Automotive Journal, 11: 70-72.

[5] Luo, S.Y. Zhang, S.F. and Tang, L.T.(2025) Investigation of Computing Platform Technologies and Core Components for Intelligent Vehicles. Automotive Electrical Equipment, 12: 57-59.

[6] Zhang, J.F. Wang, Y. and Liu, D.W, et al. (2025) Optimization and Development of the Intelligent Chassis Electrical Architecture for New Energy Commercial Vehicles. Special Purpose Vehicles, 09: 36-38.

[7] Zhu, J.(2025) Analysis of the Application of Ethernet Camera Bridging Technology in Automotive EEA Architecture. Automotive Electrical Systems, 03: 1-5.

[8] Dong, W.L.(2025) Research on Upgrade Pathways for the Electronic and Electrical Architecture of Intelligent Connected Vehicles. Automotive Test Report, 02: 25-27.