Application Scenarios of AUTOSAR Classic Platform and Adaptive Platform

Chen Huang

[1]*Electrical and Computer Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China*
**Corresponding author. Email: hsjaccount@sjtu.edu.cn*

ABSTRACT

Nowadays, complexity in design of car systems is increasing. To make the design of software more manageable, AUTOSAR provides two platforms serving as middleware between hardware and software: AUTOSAR classic platform and AUTOSAR adaptive platform. Generally, both two platforms will be used in a car at the same time, taking care of different functions required by a car. Since the two platforms are using completely different communication methods, operating systems, and hardware, they are suited for different application scenarios. This project qualitatively analyzes the differences between the two platforms in terms of their communication methods, operating system and hardware. For communication methods, bandwidth and safety will be three aspects mainly discussed. For operating system and hardware, time management and hash-rate will be discussed respectively. After comparison, a conclusion is drawn that in scenarios where safety of communication is required or real-time control is directly related to safety, AUTOSAR classic platform is preferred; and in scenarios where performance and flexibility are more important, AUTOSAR adaptive platform is preferred.

Keywords: *AUTOSAR, classic platform, adaptive platform, application scenario*

1. INTRODUCTION

With improving of the car system, the onboard control system of a car is becoming more and more complicated. Increasing complexity is in not only software and hardware design but also in the integration of software and hardware. Reusability of a software is required so that the efforts in coding and transplanting software to new hardware or operating system can be saved. AUTOSAR (Automotive Open System Architecture) is a standard software architecture developed by several automobile manufacturers, which standardizes the design of vehicle control systems and makes design of software independent of hardware. As a central standard, AUTOSAR enables reuse of software components as well as their interoperability [1].

So far, AUTOSAR provides the architecture of two platforms: AUTOSAR's classic platform provides a layered architecture for the integration of application and hardware. The core of AUTOSAR's classic platform is RTE (Run-time Environment). RTE realizes APIs of VFB (Virtual Functional Bus). Communication between SWC (software component) is based on VFB, therefore it is independent of ECU (electronic control unit). AUTOSAR adaptive platform provides a middleware called ARA (Runtime for Adaptive Application). ARA is composed of several FCs (Function Clusters). Because ARA provides APIs for the POSIX operating system, any operating system that adheres to the POSIX standard can be used to design the functions and services. In comparison to the AUTOSAR Classic Platform, the AUTOSAR Runtime Environment for the Adaptive Platform dynamically links services and clients during runtime [2].

Nowadays, many car manufactures are using both the two platforms in car system design. Traditionally, AUTOSAR's classic platform is used in control system, while AUTOSAR adaptive platform is used in the entertainment field. Since there will be more and more functions in future cars, a generalized guideline for applying which platform in a certain scenario is required. This project qualitatively analyzes the differences between the two platforms in terms of their communication methods, operating system and hardware. After analysis and comparison, a conclusion is drawn about the classifying application scenarios of AUTOSAR classic platform and AUTOSAR adaptive platform. This project can serve as a reference for applying the two platforms in application scenarios in car system design.

Y. Jiang et al. (Eds.): ICEDBC 2022, AEBMR 225, pp. 1667–1671, 2022.
https://doi.org/10.2991/978-94-6463-036-7_249

2. COMPARISON BETWEEN AUTOSAR CLASSIC PLATFORM AND AUTOSAR ADAPTIVE PLATFORM

2.1 Communication

Generally, AUTOSAR classic platform adopts CAN (Controller Area Network) communication. CAN is developed by Bosch, a German company as a serial communication protocol. CAN bus is used for data transmission between multiple ECUs, conforming to the serial communication protocol.

In a communication based on CAN (Figure 1), all the involved nodes are connected to the same medium. The data transmission is completed by signal broadcast on the medium, which means the signal can influence every node connected to the medium.

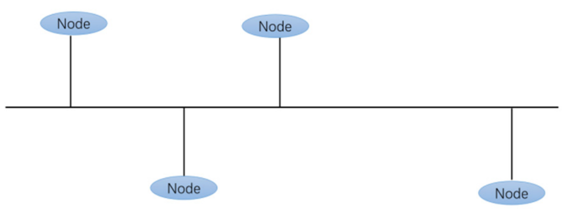

Figure 1. CAN Structure

AUTOSAR adaptive platform defines an API named ara::com to realize the communication between applications. It sets several rules about communication, for instance, communication between applications must use ara::com module instead of IPC (Inter Process Communication) and communication between processes in an application must uses ara::com module instead of IPC.

Different from CAN communication based on the signal, AUTOSAR adaptive platform is a SOA (Service-Oriented Architecture). As is shown in Figure 2, communication is realized through a designed service interface based on Ethernet. Features of service interface include field, method, and events. The SOA data transmission within the adaptive platform is realized through the Ethernet, based on SOME/IP (Scalable Service-Oriented Middleware over IP) protocol. SOME/IP is a communication protocol in the application layer of the TCP/IP model. SOME/IP guarantees that data will only be transmitted when receiver have demands. Receiver can subscribe to an event of a certain service to receive the required data. In other words, SOME/IP communication makes ECUs servers and clients. Server ECU and client ECU can communicate by using service interface.

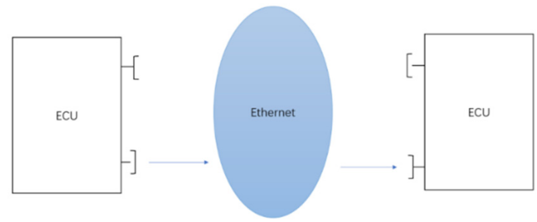

Figure 2. SOA communication

2.1.1 Bandwidth

CAN communication generally uses twisted pair as a transportation medium, whose maximum bandwidth is 1Mbit/s.

SOME/IP communication is carried out through a switch, which allows simultaneous data transmission between several pairs of nodes. Therefore, it can theoretically achieve higher bandwidth than CAN communication. Besides, the maximum bandwidth of Ethernet is generally 10 or 100Mbit/s.

2.1.2 Safety

2.1.2.1 Data Verification during Transmission

CRC (Cyclic Redundancy Check) field of a CAN frame (Figure 3) is used to perform data error detection. Based on CRC verification result, receiver will give a corresponding ACK response in ACK field.

Figure 3. Illustration of a CAN frame

The identifier is used for arbitration and therefore cannot be encrypted while the data might be encrypted. As a result, the identifier exposes information about the type of data to attackers due to the fixed coupling of identifiers and data in the automotive domain. The used acronyms are SOF - Start of Frame, RTR - Remote Transmission Request, CRC - Cyclic Redundancy Check, ACK - Acknowledgment Slot, EOF - End of Frame [3].

SOME/IP is above TCP/UDP layer in OSI (Open System Interconnection Model) seven-layer model, which means the SOME/IP frame is put in TCP/UDP payload field. The data verification method of TCP/UDP can also serve SOME/IP data transmission, including CRC and ACK. However, relying on the data verification method of TCP/UDP also means that the data verification of SOME/IP itself isn't designed independently.

2.1.2.2 Diagnostic Testing

Since CAN communication is based on a shared medium, the traditional diagnostic can be carried out by monitoring bus traffic. All the data transmission is done by broadcast in the medium so that monitoring the bus can hear all the whole traffic. However, it is not easy to monitor all the data traffic in SOME/IP communication, because the end-to-end communication supported by the switch (Figure 4) determines that monitoring one line can only hear data traffic of a certain pair of nodes.

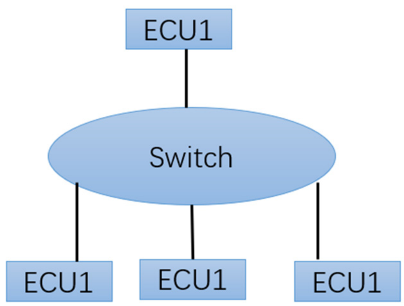

Figure 4. Ethernet Communication with Switch

2.1.2.3 Encryption

Encryption of data transmission can prevent cars using CAN networks from some attacks. Since the ID of arbitration field is used for resolving contention, it may not be encrypted as easily as the payload. Therefore, the encryption of CAN frame must be considered in terms of payload and arbitration ID respectively. Recent developments towards CAN with flexible data rate (FD) increase the payload of a single frame up to 64 bytes, enabling security features like encryption and key exchanges, for instance [4]. As to the encryption of arbitration field, research suggests an approach that determines specific message ID per vehicle for CAN systems while still satisfying all real-time end-to-end latencies [3]. Thus, the relation between identifier and data is different for each vehicle, and attackers have to face an additional obstacle when implementing exploits that are supposed to affect an entire fleet of vehicles.

As to encryption of SOME/IP data transmission, since the SOME/IP frame is put in TCP/UDP payload, it can be taken care of by encryption of TCP/UDP or other layers below TCP/UDP.

2.2. Operating System

AUTOSAR classic platform adopts OSEK OS as the core of the operating system. It is a real-time operating system (RTOS). Compared with traditional RTOS, it provides additional functions, such as memory protection and time protection. Time protection can assign deadline to tasks, and when a certain task fails to be completed before its deadline, the time protection mechanism will be triggered. Basically, three kinds of time protection are supported by AUTOSAR OS: the Execution Budget, which prevents timing errors by using execution time protection to guarantee a statically configured upper bound; the Lock Budget, which prevents timing errors by using locking time protection to guarantee a statically configured upper bound; and the Time Frame, which prevents timing errors by using inter-arrival time protection to guarantee a statically configured lower bound [2].

AUTOSAR adaptive platform can run on any operating system which accords with POSIX (Portable Operating System Interface) standard. This improves the portability of AUTOSAR adaptive platform, while the time management mechanism doesn't set rules about task deadlines as strict as OSEK OS.

2.3. Hardware

AUTOSAR classic platform is generally implemented on 8/16/32 bit MCU (Micro Controller Unit), whose hash-rate is less than 1000 DMIPS. AUTOSAR adaptive platform hardware is generally implemented on 64-bit MPU (Microprocessor Unit), whose hash rate is larger than 2000 DMIPS.

3. ANALYSIS OF APPLICATION SCENARIO OF CLASSIC PLATFORM AND ADAPTIVE PLATFORM

For future cars, two main features distinct from traditional cars will be increasing connectivity and the vision of autonomous driving [5]. This means for future cars, a high hash rate to support data transmission through Ethernet and complicated functions such as real-time path planning will be required. Hash rate of traditional MCU which is used for AUTOSAR classic platform will not be enough for these high demanding tasks. Therefore, used of adaptive platform for these tasks will be necessary.

Another reason for applying AUTOSAR adaptive platform in connectivity and vision of autonomous driving is that adaptive platform provides more flexibility for reuse of software components. Although AUTORSAR classic platform also serves as a middleware for integration of software and hardware, it is a deeply embedded system. The operating system has to be AUTOSAR OS (OSEK OS) and the implementation of the whole platform must obey the strict layered structure. For adaptive platforms, APIs for basic software components, for instance, ara::com, ara::exec, ara::rest, are not divided by strict layers. Also, the SOA communication through ara::com module instead of IPC makes the adaptive platform independent of its operating system. Different from classic platform's restriction of using OSEK OS, adaptive platform can use any operating system conforming to POSIX standard. For the design of applications that need frequent updates, using an adaptive platform can save efforts in coding and transplantation as long as the applications are designed corresponding to specifications and requirements offered by AUTOR adaptive platform. In addition, some sportful applications require high bandwidth when running, which cannot be satisfied by the classic platform. Using an adaptive platform for sportful applications can provide higher bandwidth and convenient online updates.

AUTORSAR classic platform as a deeply embedded system cannot be replaced by an adaptive platform in some functions which require strict certainty of delay and time protection. These functions usually are related to traditional ECUs on cars such as ECM (Engine Control Module), VCU (Vehicle Control Unit), BMS (Battery Management System), MCU(Microcontroller). For classic platforms, both certainties of delay and time protection serve as mechanisms for safety concerns. Certainty of delay can be realized because CAN communication has its specific contention rules for sharing the medium among every node. Therefore, the worst-case delay can be determined in advance. Time protection contributes to safety by setting a deadline for each task when applications are running, therefore accident delays in real-time communication won't impede other running applications. Besides, the diagnostics testing is easier to carry out because all the data transmission is done in the same medium. Adaptive platform cannot replace the classic platform in these scenarios because adaptive platform isn't RTOS. On adaptive platform, applications are not bound any more to a very strict and static scheduling and memory management but are free (within well-defined boundary conditions) to create and destroy tasks and to allocate memory depending on their current need [5].

In summary, making choice between classic platform and adaptive platform in various scenarios is trading off between certainty and flexibility. In scenarios where certainty is directly related to safety, AUTOSAR classic platform is preferable. In scenarios where the safety of communication is required or real-time control is directly related to safety, AUTOSAR classic platform is preferred; and in scenarios where performance and flexibility are more important, AUTOSAR adaptive platform is preferred.

4. DISCUSSION ABOUT INTEGRATION OF CLASSIC PLATFORM AND ADAPTIVE PLATFORM

As an extension to support future cars' need for connectivity and vision of autonomous driving, the adaptive platform cannot completely replace the classic platform. Therefore, the integration of a classic platform and adaptive platform on a car is necessary. However, one problem is that classic platforms and adaptive platform are using two different communication methods. The communication within the classic platform is based on signal, while the communication within the adaptive platform is service-oriented, transmitting through SOME/IP. Because of the inconsistencies in communication methods, communication between the classic platform and adaptive platform cannot be carried out directly. One possible solution for this problem is to design an extra module for converting signal to service, or vice versa. Another option is to combine the traditional and adaptive platforms on the same hardware. However, both solutions should take into account the encryption of CAN frame and SOME/IP frame. As is mentioned in section 2.1.2.3, the encryption of CAN frame is different from SOME/IP frame because its arbitration field is required for solving contention. When conversion between signal and service is carried out, the capability of two different encryption methods should be designed carefully.

Besides, the design of communication interface between classic platform and adaptive platform should also consider the problem of complexity. As is mentioned in a survey about benefits and drawbacks of AUTOSAR, the main drawback is its complexity [6]. Therefore, when designing a new module to fulfill some functions, complexity should be bear in mind.

5. CONCLUSION

AUTOSAR classic platform and AUTOSAR adaptive platform are both serving as middleware to make application design independent of basic hardware. By achieving agreement on specifications and requirements for the platforms, relative industries can focus on software design without taking basic hardware into account. The main differences between the classic platform and adaptive platform are their communication methods and the operating systems. Communication within classic platform is generally using signal-based CAN communication. CAN communication shares the same medium among all the nodes, and the bandwidth is up to 1Mbit/s. Communication within an adaptive platform is generally service-oriented, using SOME/IP protocol and is transmitted through Ethernet. It is end-to-end communication through the switch; therefore, it can achieve higher bandwidth. However, the diagnostics testing is more easily to be completed on CAN bus than on SOA, because monitoring the CAN bus can hear all the data transmission. Besides, the data verification and encryption of CAN communication are designed independently, which enhances the safety of data transmission. Classic platform uses deep embedded RTOS (OSEK OS), featuring its strict time protection mechanism, while relatively poorer flexibility. Adaptive platform can be implemented on any operating system conforming to POSIX standards. It is more flexible while less strict with time protection. Due to these properties, classic platform can be used in scenarios where demand for real-time and data transmission safety is higher, while demand for bandwidth is lower.

In conclusion, in scenarios where safety of communication is required or real-time control is directly related to safety, AUTOSAR classic platform is preferred; and in scenarios where performance and flexibility are more important, AUTOSAR adaptive platform is preferred. In the future, to realize the integration of the two platforms, an extra module that completes the conversion between signal and service should be designed. And if encryption is taken into consideration, the module should be able to handle the two different kinds of encryptions (signal-based or service-oriented) properly.

REFERENCES

[1] Soltani, M., & Knauss, E. (2015). Challenges of requirements engineering in Autosar Ecosystems. 2015 IEEE 23rd International Requirements Engineering Conference (RE). https://doi.org/10.1109/re.2015.7320445

[2] Autosar.com. Retrieved October 16, 2021, from http://www.autosar.com/.

[3] Lukasiewycz, M., Mundhenk, P., & Steinhorst, S. (2016). Security-aware obfuscated priority assignment for automotive can platforms. ACM Transactions on Design Automation of Electronic Systems, 21(2), 1–27. https://doi.org/10.1145/2831232

[4] Hartwich, F. (2012). CAN with Flexible Data-Rate. Retrieved October 16, 2021.

[5] Furst, S., & Bechter, M. (2016). Autosar for connected and autonomous vehicles: The Autosar Adaptive Platform. 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W). https://doi.org/10.1109/dsn-w.2016.24

[6] Martínez-Fernández, S., Ayala, C. P., Franch, X., & Nakagawa, E. Y. (2015). A survey on the benefits and drawbacks of autosar. Proceedings of the First International Workshop on Automotive Software Architecture. https://doi.org/10.1145/2752489.2752493