Research on Vehicle OTA Design Based on Electronic and Electrical Architecture

Ma Yunlin, Liu Jun, Cai Chunmao, Wang Zhongcai, Wen Yong, Sun Xiaowu
Chongqing Changan Automotive Software Technology Co., Ltd.

Abstract: OTA is an over-the-air upgrade technology, which provides the ability to continuously update software through iterations. Vehicle OTA is limited by the electronic and electrical architecture, the upgrade time is long, and the controller is difficult to control. In order to improve the stability of the vehicle OTA, shorten the upgrade time, and upgrade the large version of the vehicle, this paper proposes an OTA architecture design scheme, which realizes the standardization of the vehicle OTA upgrade, the unified control of the upgrade process, and the platformization of the architecture. This design scheme can be applied to cross vehicle OTA design and can solve many problems such as mixed ECU controllers, long upgrade time, and instability. This solution has become a complete cloud to vehicle full link continuous update capability, which has great promoting significance for enhancing the value of intelligent connected vehicles.

Key words: Vehicle OTA, upgrade time, unified control, platformization

Introduction

OTA refers to the over-the-air download technology, which realizes the remote management of software through the interface of mobile communication. OTA is a channel for car software upgrades, and its value is to remotely flash new software into the car. Software-defined vehicles have gradually become the consensus in the industry. There are two trends in automotive software: First, the car delivered by the OEM will no longer be a product with solidified functions, but a continuously updated smart device. Continuously support software iterative upgrades; second, with the increase of software volume, software bugs will become potential risks. OTA can effectively solve software failures, reduce software risks caused by short development cycles through software upgrades, complete software vulnerability repairs, and reduce recalls caused by software problems. OTA remote upgrade technology has gradually become the standard configuration of intelligent networked vehicles, constantly endowing the car with the potential to increase its value, and continuously updating software functions through continuous iterations, thereby driving a new operation and business model in the automotive industry.

1 Demand analysis

Vehicle OTA is limited by the electronic and electrical architecture, but there are many difficulties. With the improvement of automotive electronic technology, the electronic control unit ECU occupies the entire car, covering the fields of power, chassis, body, cockpit and automatic driving. The number of ECUs is as many as dozens or even hundreds. These ECUs are provided by different suppliers and run a variety of operating systems and application software. The OTA of the whole vehicle means that all related controllers have to update the software version in one upgrade process, so the total upgrade time and The success rate is the two major difficulties that OTAs face. Due to the long OTA time of the whole vehicle, in order to preserve the stability of the upgrade, most of the functions of the car will be disabled, requiring the vehicle to be turned off, and the long-term upgrade will lead to the consumption of battery power, which is also a major difficulty faced by OTA.

In order to improve the stability of the vehicle OTA, shorten the upgrade time, and realize the large version upgrade of the vehicle, a platform-based OTA architecture design scheme is proposed based on the electronic and electrical architecture of the vehicle.

2 Overall scheme design

Vehicle OTA is mainly to coordinate and control different types of controllers on the vehicle to jointly complete the software update process in one upgrade task. Therefore, it is the basic idea of the scheme design to propose a unified upgrade function specification and centralized upgrade control for the controller. The design of the scheme follows the principle of "centralized control, divide and conquer", which is reflected in the following three aspects:

1) Classification of objects. The controllers on existing vehicles are mainly divided into two categories: the first category is intelligent systems with operating systems capable of self-upgrades, such as car machines and instruments in the cockpit domain, and automatic driving control in the driving domain. The second type is the traditional electronic control unit ECU, which has no operating system and completes the software update by flashing the host computer. Therefore, the upgrade objects are divided into two categories: intelligent systems and brushed controllers, and the corresponding technical requirements must be

followed. The intelligent system requires version information maintenance, file storage, self-upgrade, and self-recovery capability after abnormal upgrade; the flashed controller is required to meet the flashing specification of UDS (Universal Automotive Diagnostic Service). Upgrade object classification, divide the many controllers on the car according to their software update characteristics, put forward consistent OTA technical requirements and functional specifications for the same type of controllers, and achieve standardized OTA control.

2) Segmentation of the process, OTA is divided into download deployment process and installation process. During the download and deployment process, the OTA server notifies the vehicle of the upgrade task, and the car side downloads the upgrade package from the server side to the local. This process can be carried out while the vehicle is running without affecting the use of the vehicle. During the installation process, each controller on the car performs software update. This process needs to keep the vehicle in a specific state, such as stop, engine stall, etc., and the car cannot be used normally. The execution objects, execution conditions and control strategies of the download deployment process and the installation process are different, and the OTA process can be centrally controlled by segmenting the process.

3) Separation of responsibilities, 4 roles are divided in the system architecture of OTA.

a) OTA Server: Responsible for upgrade package management, vehicle software version management, upgrade tasks and upgrade strategy management. b) OTA Client: Complete the version collection of all controllers on the vehicle, interact with the server to obtain upgrade tasks and report the upgrade status, download the upgrade package, and distribute the upgrade package and upgrade information to the controllers that perform the upgrade, responsible for the function of human-computer interaction. c) OTA Master: Check the vehicle installation conditions, maintain the installation status, execute the installation strategy and control the associated rollback. d) OTA sub-controller (Sub Master): It has the ability to read and write files, can complete self-upgrade, and can perform software flashing on connected controllers or hardware modules without self-upgrade ability through the bus or other physical channels. On the automotive side, the three roles of OTA client, OTA master and OTA sub-controller can integrate components into different controller entities according to the electronic and electrical architecture and the capabilities of the controller. The componentization of functional modules can realize the reuse of different electronic and electrical architecture models, so that the solution has a good platform.

Based on the above-mentioned scheme of "two types of objects, two processes, and four roles", the overall system design is shown in the following figure 1:

The OTA Server role is implemented on the server side, including management platform, file and data management. The role of OTA Client is implemented on a controller with networking functions and human-computer interaction functions, and requires

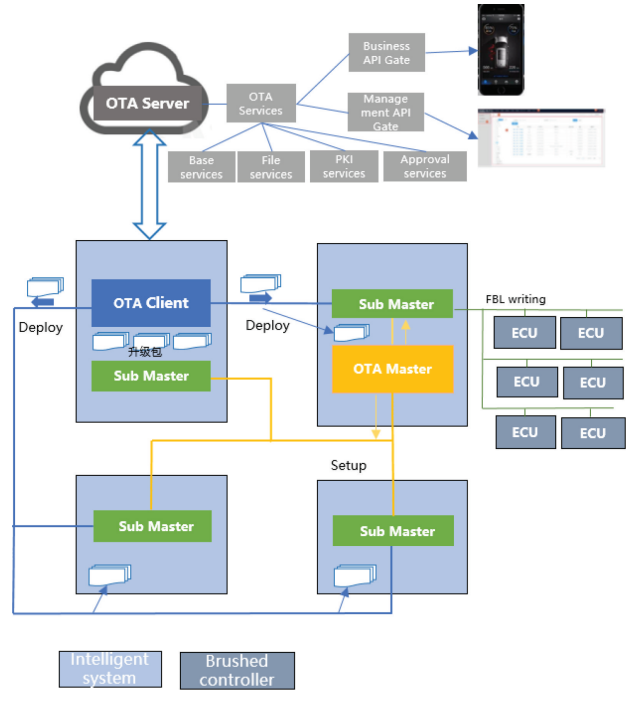

Figure 1 Overall design scheme

a large number of file storage and file distribution capabilities, usually integrated into the car or T-BOX. The role of the OTA Master is at the central node, which needs to centrally coordinate the installation of each controller, usually integrated into the gateway. Each Sub Master role is usually an intelligent controller with an operating system such as Android, Linux, QNX, etc., such as vehicle machine, instrument, automatic driving controller, gateway, etc.

On the vehicle side, the download and deployment process (steps 1.1 and 1.2 in Figure 1) is initiated by the OTA Client. After the file is downloaded from the server side, the upgrade information and upgrade package are distributed to each controller node that performs the installation one by one. The installation process (steps 2.1 and 2.2 in Figure 1) is initiated by the OTA Master to determine whether the vehicle installation conditions are met, and to maintain the vehicle installation status. According to the installation sequence, an installation command is issued to each Sub Master, and the Sub Master updates the software according to their respective software methods., respectively, to perform the installation. Since the installation execution of each Sub Master is independent, the OTA Master controls the installation in parallel to realize the parallel upgrade of the whole vehicle.

3 System Design

According to the three principles of "two types of objects, two stages, and four roles" in the scheme design, combined with the EE architecture of the vehicle, the vehicle upgrade objects and functional modules are divided. The vehicle OTA system architecture is shown in Figure 2.

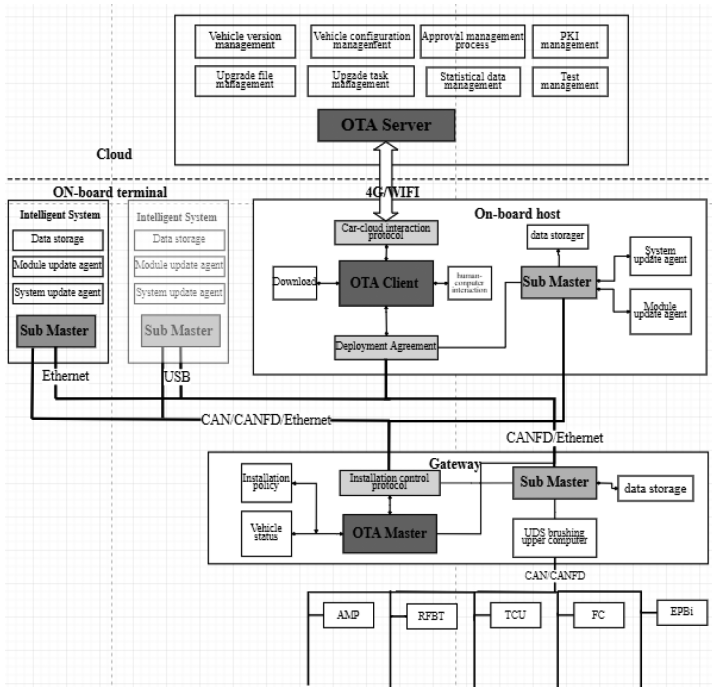

Figure 2 System Architecture Design

3.1 Server-side design

On the server side, OTA Server is the data and management center of OTA on the server side. Among them, model configuration management, vehicle version management and upgrade task management are indispensable functions for vehicle upgrade.

1) Each car series needs to set a model configuration group, one group corresponds to one or more model configurations, and a model configuration can only correspond to a unique group. Each group needs to configure the software set of all controllers, and maintain the corresponding relationship with the software of the controller through the software ID.

2) The management granularity of the vehicle version is the model configuration group. The software version of each model configuration group is managed according to the major version of the vehicle. The major version consists of the software versions of all controllers under the model configuration group.

3) The upgrade task includes the upgraded controller object, installation strategy, and upgrade scope. When adding a task, you need to set the car series, model configuration group and the corresponding target vehicle version. Each upgrade object can set how its software will be updated, the estimated installation time, and the maximum number of exception handling. The installation strategy includes installation conditions, installation order, and dependencies on the software version of the upgrade object. a) Installation conditions include: driving gear, battery power range, lower temperature limit, power gear, etc. b) The installation sequence includes parallel and serial installations, which are automatically generated when an upgrade task is created according to the inclusion relationship between the SubMaster and the controller on the car side, and the installation dependencies between the controllers. c) The software version dependency of the upgrade object includes multiple associated groups. If the installation of a new version of a controller fails in each group, other associated controllers should also be rolled back to the previous software version at the same time. Automatically generated when an upgrade task is created, based on the association between controllers.

3.2 Automotive Design

3.2.1 Functional Design

In the process of downloading, deploying and installing on the car side, there is a control subject to ensure unified control in both processes, and according to the processing capability of the platform, try to use a concurrent method to achieve the goal of shortening the entire OTA time.

1) In the download and deployment process, the OTA Client synchronizes the download process and the distribution process, so that the two processes can be executed concurrently, so as to shorten the overall time for the download and deployment of the upgrade package and reduce the demand for storage space. According to different hardware channels, the OTA Client preferentially downloads the upgrade package of the SubMaster node with low data transmission rate, and finally downloads the upgrade package of the SubMaster node where the OTA Client is located; the file deployment can be performed after the download of the upgrade package of a single SubMaster. If the vehicle is turned off during the download and deployment process, the download and deployment will be stopped, and the execution will continue at the breakpoint after the next ignition. The whole

process can be performed while driving, without affecting the user's use of the car.

2) The installation process is controlled by the OTA Master. After the user confirms that the installation is initiated, the OTA Master checks whether the installation conditions of the whole vehicle are met according to the installation conditions in the upgrade information, and keeps the installation state after the installation is initiated, such as the power supply gear and the driving gear. bit, vehicle OTA status, etc. The OTA Master sends installation requests to each SubMaster in turn; after receiving the installation requests, the SubMasters execute the installations individually to achieve the effect of concurrent installation. As the SubMaster, the gateway is the flashing host computer of the ECU, which realizes parallel flashing between each network segment, and the controller on the same network segment is serially flashing.

The OTA Master is executed according to the installation sequence in the upgrade task. The installation sequence includes multiple sub-tasks. The sub-tasks are executed serially, and the network segments within the sub-tasks are installed in parallel. Therefore, if there is a dependency on the installation order of the controllers, the dependent controller is divided into the next subtask. The total time to install is given by the following formula:

$$T=\sum_{i=1}^{n}\left(\mathrm{Max}\left(t_1,t_2,\cdots\cdots,t_n\right)\right)$$

Among them, t_n is the longest SubMaster installation time on network segment n in the subtask.

3.2.2 Interface design

In order to meet the data interaction between different roles in the OTA process of the car, two sets of protocols are designed in Figure 3.

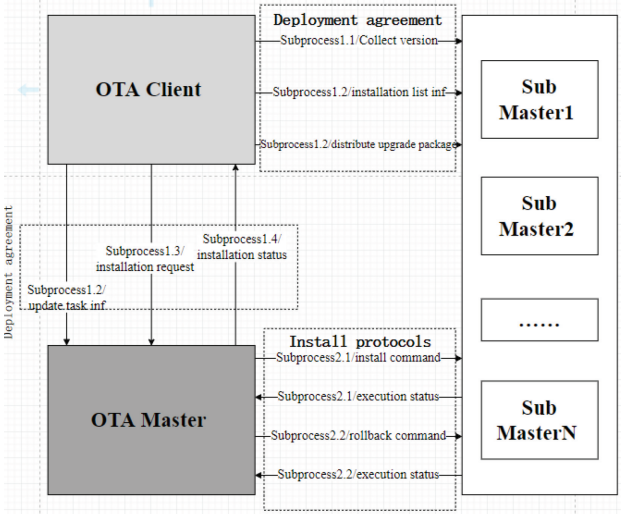

Figure 3 Vehicle-end control process

1) Deployment protocol, which is mainly used in the deployment process and human-computer interaction process. The protocol adopts the form of server/client, OTA Client as client, Sub Master/OTA Master as server. Physical channels include CANFD, Ethernet, USB, etc. The interaction of the deployment protocol covers four sub-processes: a) Version collection: The OTA Client requests version collection from each Sub Master. The scope of the SubMaster collection version includes the software version of the controller where it is located, and it is responsible for flashing or controlling other submodules/Software version of the controller. b) Upgrade information and upgrade package distribution: After the OTA Client is downloaded, it transmits the upgrade package and installation object information to each Sub Master. At the same time, the OTA Client sends the upgrade task information to the OTA Master. c) After the first two processes are completed, the OTA Client sends an installation request to the OTA Master according to the result of the human-computer interaction. d) Installation status and progress query: OTA Client queries the installation conditions, installation execution status, installation progress and installation results of the OTA Master to meet the needs of human-computer interaction.

2) The installation control protocol is mainly used in the installation process. The protocol data is carried on the diagnosis protocol, and all the controllers of the whole vehicle are reached by means of the diagnosis channel. The interaction of the installation control protocol covers two sub-processes: a) After the installation preparation is completed, the installation execution is initiated, the OTA Master upgrades the installation sequence in the task information, reads the installation sub-tasks in turn, sends the installation command to each SubMaster, and queries the installation execution status, until all subtasks are executed. b) After the installation is completed, the OTA Master reads the installation results of each SubMaster, determines whether any objects have failed to install, and reads the software version dependencies in the upgrade task information. If the failed upgrade object depends on other successfully upgraded objects, then Issue a rollback command to the Sub Master of the successfully upgraded object, and query the rollback execution status until all associated groups in the dependency are traversed.

4 Applications

Based on the above scheme design of vehicle OTA, an upgrade channel of vehicle OTA is built on the controller of a certain model project. The car machine acts as the OTA Client, and the gateway acts as the OTA Master to upgrade the controllers in the cockpit domain, body domain, power domain, chassis domain, and autonomous driving domain. The OTA scheme in Figure 4 realizes the upgrade of 18 controllers, including 4 intelligent system devices and 14 flashed controllers; a total of 5 SubMaster nodes are deployed.

According to the upgrade scope of the controller, the installation sequence of the server-side upgrade task is set as shown in Figure 5.

The key OTA upgrade channel scheme is tested according to all the controllers in the upgrade sequence in Figure 5. The time of the download deployment process and the installation process is

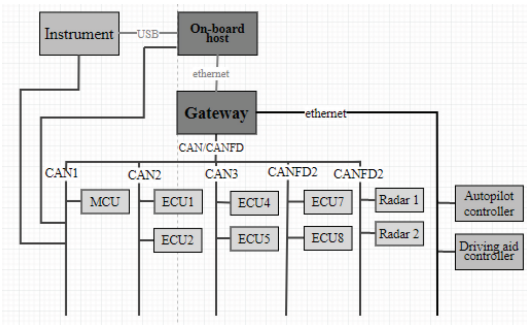

Figure 4 OTA upgrade channel scheme of a project vehicle

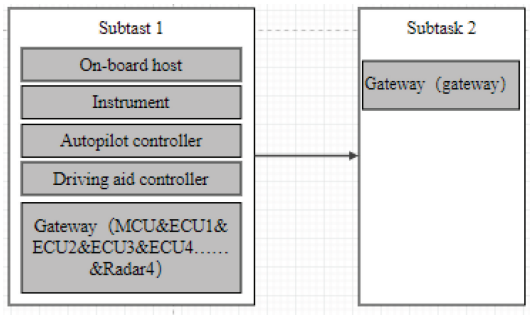

Figure 5 OTA installation task of a project vehicle

shown in Table 1 and Table 2. The entire OTA process takes 44 minutes, and the installation time that the user can perceive is shortened to 28 minutes. Compared with the traditional serial installation of the controller, the overall installation time is greatly saved, and the OTA experience is significantly improved.

It can be seen from Table 2 that the total installation time bottleneck lies in the network segment with the longest installation time in a subtask. Therefore, if the controller is appropriately added to other network segments, it will not affect the overall installation time; if the overall installation time is to be shortened, the controller on the network segment with the longest installation time can be optimized, such as reducing the size of the upgrade package., Improve it into a self-upgradable intelligent system, adjust the network segment location and other feasible measures.

Table 1 Download and Deployment Time

No	SubMaster	Download and Deployment/min
1	On-board host	16
2	Instrument	5
3	Gateway	8
4	Autopilot controller	2
5	Driving aid controller	5
Total download and deployment time		16

Table 2 Installation time

Subtask	Network segment	Installation time/min
1	CAN1	15
	CAN2	11
	CAN3	10
	CANFD1	25
	CANFD2	5
	Ethernet	8
2	CAN2	3
Total installation time		28
Total serial installation time		77

5 Conclusion

This architecture solution realizes the standardization of vehicle OTA upgrade objects, the unified control of the upgrade process, and the platformization of functional modules. Judging from the effect of the test, the effect of parallel upgrade and unified control of the upgrade process has been achieved, providing an implementable and platform-based solution for the realization of vehicle OTA. This solves the problems that the controller is difficult to manage and the upgrade process is uncontrollable during the OTA process of the vehicle, shortens the time for vehicle upgrade, and improves the stability of the upgrade.

References

[1] WANG D L, et al. Research on OTA function design of intelligent networked vehicles [J]. Automotive Technology, 2018.

[2] SHI Q G, et al. OTA upgrade scheme for intelligent connected vehicles [C]//Proceedings of the 2018 China Society of Automotive Engineers Annual Conference. Beijing: China Machine Press, 2018.

[3] YUAN J Y, et al. Virtual Simulation Test Platform for Vehicle OTA System [J]. Shanghai Automotive Practical Technology, 2020.

[4] XUE M G, WANG W. Security Concepts for the Dynamics of Autonomous Vehicle Networks [J]. Sandip Roy-Auto-matica, 2014, 3.

[5] ZHONG S, ZHANG Y. How to Select Optimal Gatewayin Multi-Domain Wireless Networks: Alternative Solutions without Learning [J]. IEEE Transactions on Wireless Communications, 2013, 12 (11).