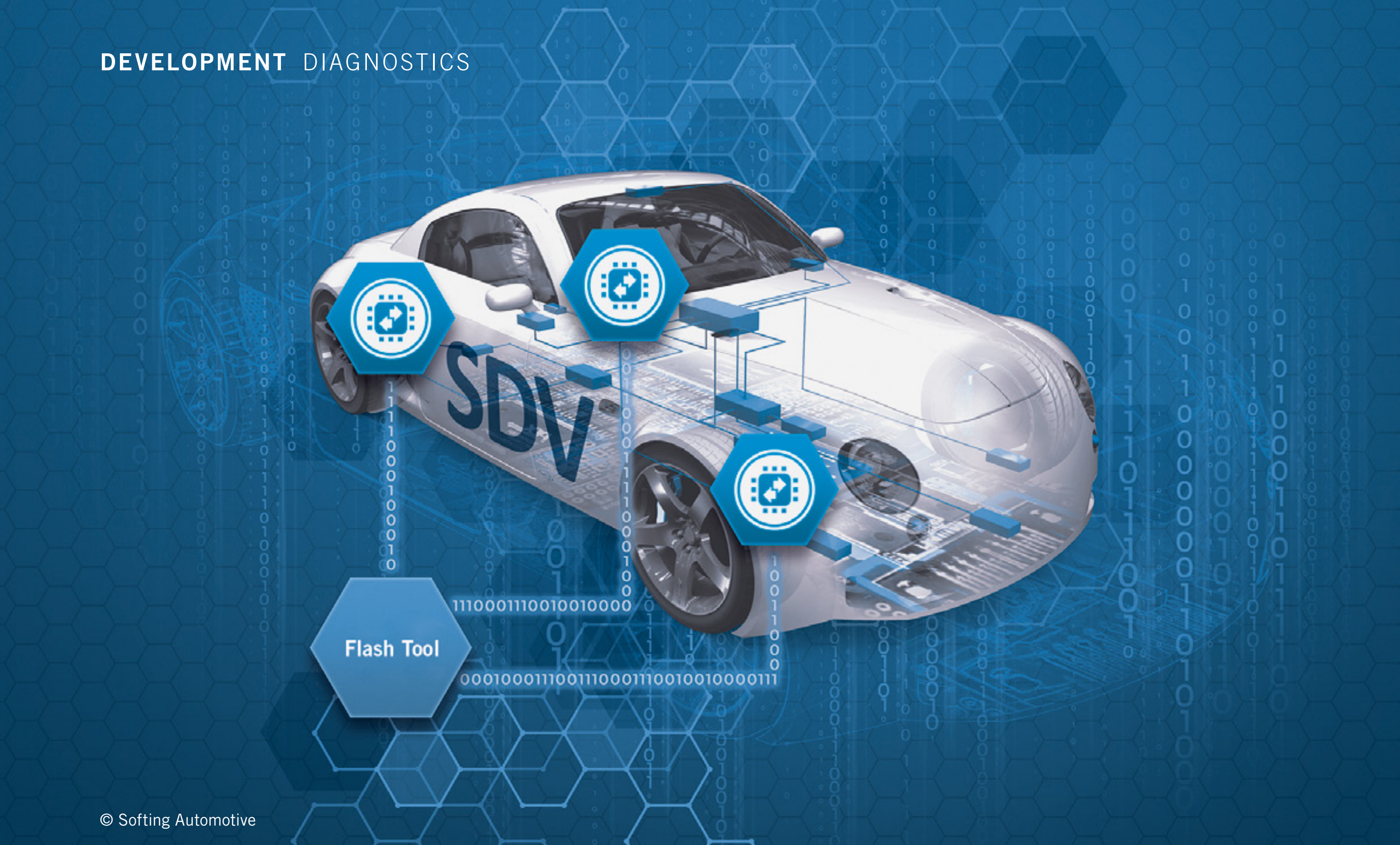


The Software-defined Vehicle – Flash Programming at its Limits

WRITTEN BY

Markus Steffelbauer
is Head of Product Management at Softing Automotive in Haar (Germany).

The increasing amount of software in the vehicle that can be attributed to functions implemented to increase efficiency, convenience and safety is leading to the software-defined vehicle. However, it is also leading to challenges in flash programming, as this is threatening to become a limiting factor. Softing Automotive shows how this can be addressed with suitable solutions.

Vehicles are becoming increasingly efficient: recuperation recovers energy which is not consumed at all when sailing with navigation support, and engines consume less through increasingly precise operating points. Vehicles are becoming more and more comfortable and convenient, for example with air suspension and features such as parking aids. They are also becoming safer and safer because they have autonomous driving functions and an emergency braking assistant. All these functions are largely implemented in software. This is why today people are already talking about the software-defined vehicle.

This will apply even more to future generations of vehicles. Not only will there be a further increase in the amount of software on the road to fully autonomous vehicles, but the infrastructure must also be increasingly considered. This includes, for example, the traffic lights in the area or the parking garage into which the vehicle parks autonomously. But it also includes the cloud, in which vehicles automatically report accidents and traffic jams, as well as abnormalities in the road surface, so that following vehicles can adapt to them immediately. Such functions also exist – in software.

CONSEQUENCES

The first consequence of the software-defined vehicle is a massively changing E/E architecture. In today's architecture, a duality of vehicle function and mechatronic system has been established over the years. A mechatronic system consists of the mechanical system and the electronic control unit (ECU). ECUs are connected to each other via bus systems, today still mostly CAN or CAN FD, so that the vehicle functions can access information from other functions. Access to the vehicle from outside is controlled via a central gateway, which also performs an internal distribution task for several bus systems. This architecture will change massively based on High-Performance Computers (HPCs), **FIGURE 1**.

There are several reasons for this. Firstly, HPCs provide the processing power required for new driving functions. Multi-core computers are envisaged, which can also support several operating systems to integrate real-time-critical and normal functions, for example. Secondly, several HPCs will be kept in the vehicle to provide the redundancy that is essential for autonomous functions. Thirdly, the HPCs are distributed appropriately in the vehicle to enable a zonal architecture. This is particularly useful to reduce the amount of copper required. And finally, in addition to the OBD jack for diagnostic access, there is a wireless connection that integrates the vehicle into the infrastructure.

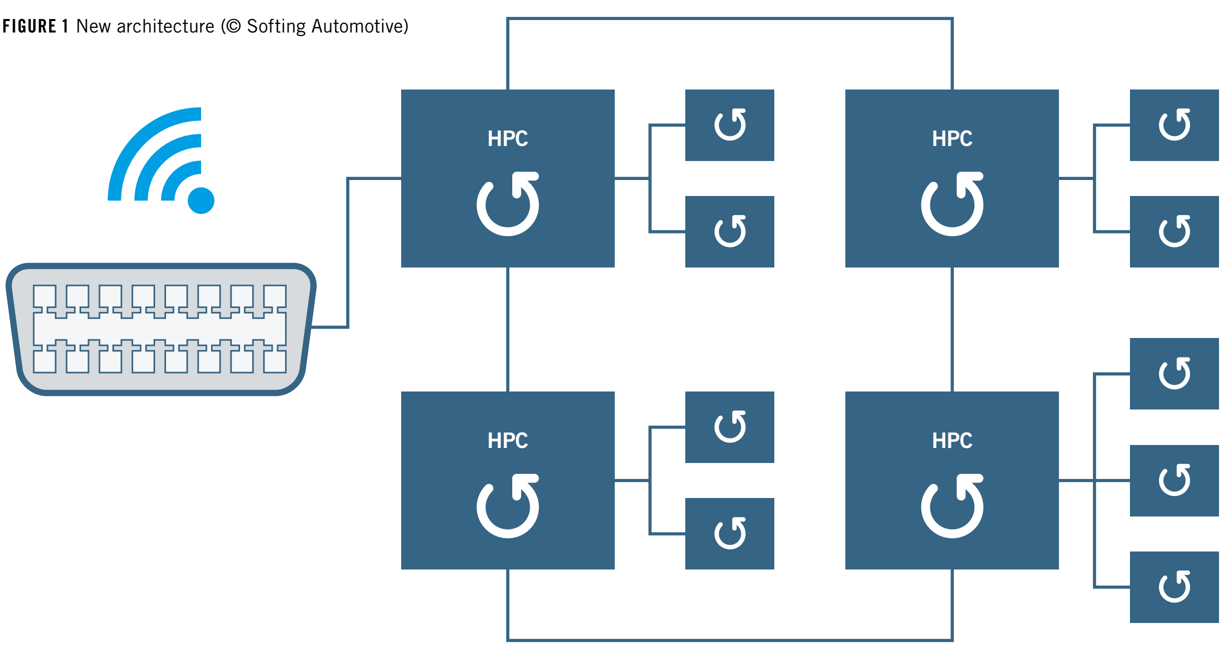
This architecture significantly reduces the number of ECUs. However, as the number of functions continues to increase, numerous functions are being implemented as software modules in the HPCs. This means that those from different suppliers must be integrated on the HPC. Since all modules must be subject to versioning and must also interact with software in other HPCs and classic ECUs, it is now logical to say that the vehicle manufacturer must become a software manufacturer.

FLASH PROGRAMMING IN MANUFACTURING

The many software modules automatically lead to a high frequency of flash programming processes. In addition to the release of the various versions, the sheer volume of software is a major challenge. To put this into perspective: whereas in 2010 there were around 10 million lines of code (LoC) in a luxury vehicle, this figure has risen to 100 million LoC in 2020. Estimates predict 1000 million LoC by 2030, a factor of one hundred in 20 years. It is estimated that around 50 GB of software will then have to be programmed.

Today, ECUs are usually programmed via the vehicle's OBD jack. CAN is specified as the bus system; CAN FD or 100BASE-TX ethernet are also possible The CAN is operated with a transmission rate of 500 kbit/s. However, a realistic user data rate is only 26 kB/s, caused by overhead from the data format (CAN ID, checksum) and the waiting cycles resulting from the protocol. For CAN FD, the user data rate is about five times as high, approximately 130 kB/s. Taking the 50 GB data volume as a benchmark, this results in programming times of 540 respectively 108 h, which is of course way too slow. If this is taken into consideration for ethernet, a realistic user data rate of 6.3 MB/s and thus a programming time of 130 min can be

FIGURE 1 New architecture (© Softing Automotive)



achieved at a transmission rate of 100 Mbit/s – significantly more header information must be transmitted here, and more waiting times must be calculated. This is still far too slow for manufacturing applications with cycle times of just minutes. Things only start becoming acceptable with Gigabit ethernet that would ensure a reasonable 13 min.

A look at modern wireless interfaces paints a mixed picture. The very high theoretical transmission rates – 10 Gbit/s are possible with WiFi 6 and even up to 20 Gbit/s with 5G – are offset by significantly reduced user data rates. In addition to the relatively large overhead here too, physics play a significant role. The user data rate that can be achieved depends not only on the theoretical transmission rate, but also on other factors: for example, the current channel assignment (how many vehicles are currently using an access point) and the field strength (including the distance from the access point). Realistically, the 50 GB are programmed into the vehicle in around 11 min with WiFi 6 and in just over 30 min with 5G.

In addition to pure programming speed, reliability plays a major role in vehicle manufacture. The radio links do not perform well here because the current user data rate depends on the position on the assembly line and the changing connection parameters to the access point. There is another point to consider on the production line. As a rule, today's production lines do not produce just one type of vehicle, but alternate between mid-range vehicles, luxury cars and SUVs. This also results in different bandwidth requirements, and you do not just have to store 50 GB of data, but the same amount of data for 5-8 series.

SOLUTION CONCEPT

Today, programming is conducted via the Vehicle Communication Interface (VCI), small computers that are plugged directly into the OBD jack and connected to the programming system via WiFi. These are not suitable for large data volumes (shielding in the footwell, non-deterministic WiFi path for large data volumes). However, wired solutions that were common 20 years ago must not be reintroduced for handling reasons.

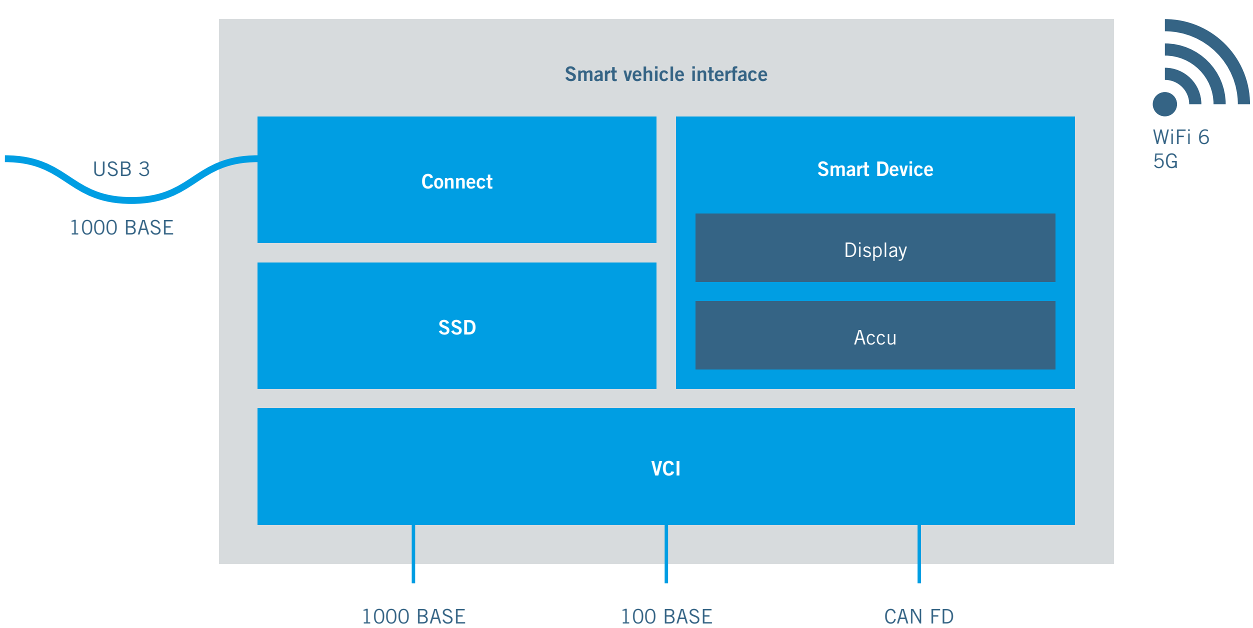
The following general requirements must be met: deterministic and fast programming with large amounts of data and the continuous supply of newly released flash data and handling of 400 GB (8 times 50) of data for several vehicle series as well as a minimum boot time for fast system availability in the vehicle, access to the OBD jack so that no additional technology is required in the vehicle, and a display for status information and operating steps.

The building blocks for a solution are all in place. Today's VCIs are excellent at real-time processing of the protocols required for vehicle communication. They need to be expanded so that they can process Gigabit Ethernet. The same applies to the vehicle side, but this is a feasible extension. All of us carry with us an example of communication via fast wireless connections in combination with a display: a smart device. Together, the basic requirements are met, **FIGURE 2**.

The solution contains the following elements:

– The VCI manages vehicle communication; in addition to flash programming via Gigabit Ethernet, there are also legacy access points to the vehicle via 100BASE-TX and CAN FD for normal diagnostics.
– The smart device provides the wireless connection technology and the display for status information and applications. The integrated battery also ensures that there are no boot times when changing vehicles. Status information includes, for example, connection quality to the vehicle and to the backend as well as the battery charge status.
– The SSD stores data for a variety of vehicle types so that any vehicle can be programmed on the assembly line.

FIGURE 2 Hardware architecture (© Softing Automotive)



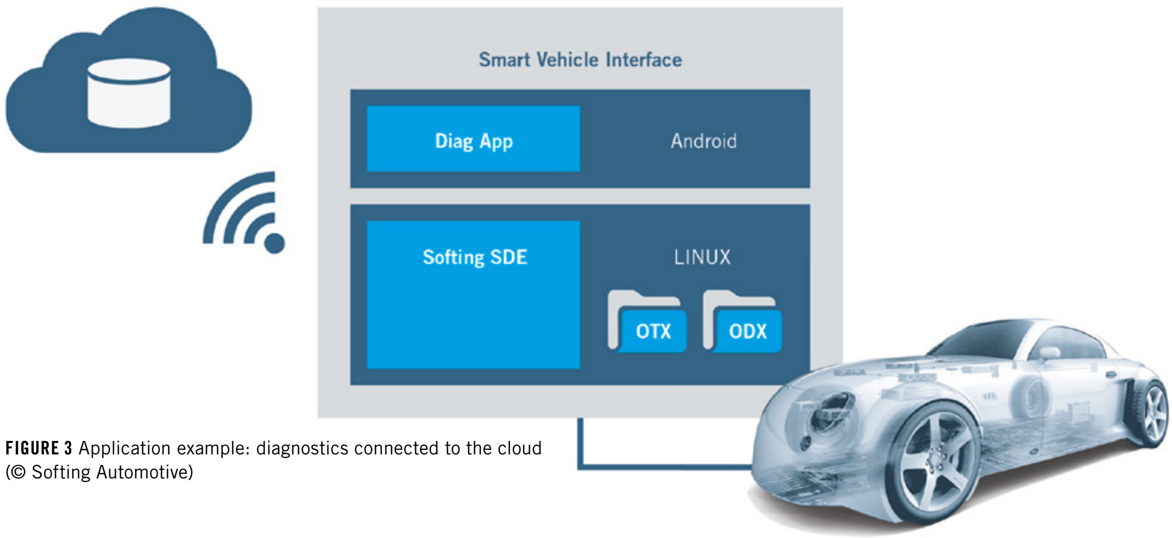


FIGURE 3 Application example: diagnostics connected to the cloud (© Softing Automotive)

It must be fast enough to store data loaded via WiFi 6 or Gigabit Ethernet without delay.

- On the one hand, the Connect module allows data to be brought into the device by cable from outside, and, on the other, it also manages the internal high-speed connections.

All modules are accommodated in a compact housing and connected to the vehicle via a special adaptation using a cable. During operation, the device is first supplied with the necessary (flash) data. It is then attached to the vehicle at the corresponding section of the assembly line. Reading out the Vehicle Identification Number (VIN) results in a unique assignment of vehicle and programming device being made so that the backend can initiate the correct programming procedure. Once the programming and diagnostic tasks have been completed, the device is unplugged again and loaded on a transport trolley and returned to the starting point. During this time, updated information can be uploaded at any time, and reports can also be downloaded for further processing in quality assurance.

FURTHER AREAS OF IMPLEMENTATION

The device described is intended for programming vehicles on the production line. However, the configuration can also be used for a variety of other applications. All you need to do is install an appropriate basic diagnostic system, such as Softing SDE, which processes the usual diagnostic data (ODX ISO 22901-1 and OTX ISO 13209). The diagnostic application is then application specific, **FIGURE 3**.

For example, the vehicle health state of several vehicles can be read out via the device during breaks in the road test and then transferred directly to the cloud for further processing via 5G. It is also possible to install new software on the affected vehicles. If smaller amounts of data are involved, for example for individual control units, this can be done directly via 5G, otherwise via the Wi-Fi at the base. Another example in after-sales service would be a damaged vehicle that is diagnosed directly on the road. This allows the necessary spare parts to be ordered in the workshop so that the subsequent repair can be conducted much more quickly.

CONCLUSION

The software-defined vehicle poses major challenges for the entire industry, both in terms of engineering and vehicle updates. The estimated data volume of 50 GB poses a technical and logistical challenge, particularly in time-critical vehicle manufacturing. The combination of a high-performance VCI and a modern smart device enables a solution that combines fast and deterministic programming with a continuous data supply. The challenges of tomorrow can thus be addressed and rolled out today.

IMPRINT
Special Edition 2024 in cooperation with Softing Automotive Electronics GmbH, Richard-Reitzner-Allee 6, 85540 Haar; Springer Fachmedien Wiesbaden GmbH, Postfach 1546, 65173 Wiesbaden, Amtsgericht Wiesbaden, HRB 9754, USt-IdNr. DE81148419

MANAGING DIRECTORS:
Stefanie Burgmaier I Andreas Funk I Joachim Krieger

PROJECT MANAGEMENT: Anja Trabusch

COVER PHOTO: © Softing Automotive Electronics GmbH

optimize! softing

DIAGNOSTIC AND TEST SOLUTIONS BY SOFTING

AUTOMOTIVE

automotive.softing.com