

Automotive

# Future Advances in Body Electronics

AMPG Body Electronics Systems  
Engineering Team



[freescale.com](http://freescale.com)

# Table of Contents

<b>1 Introduction</b> .....	<b>2</b>
<b>2 Networking</b> .....	<b>3</b>
<b>3 Hyper-Integration</b> .....	<b>8</b>
<b>4 Power</b> .....	<b>10</b>
<b>5 Functional Safety</b> .....	<b>14</b>
<b>6 Security</b> .....	<b>15</b>
<b>Conclusions</b> .....	<b>17</b>

## 1. Introduction

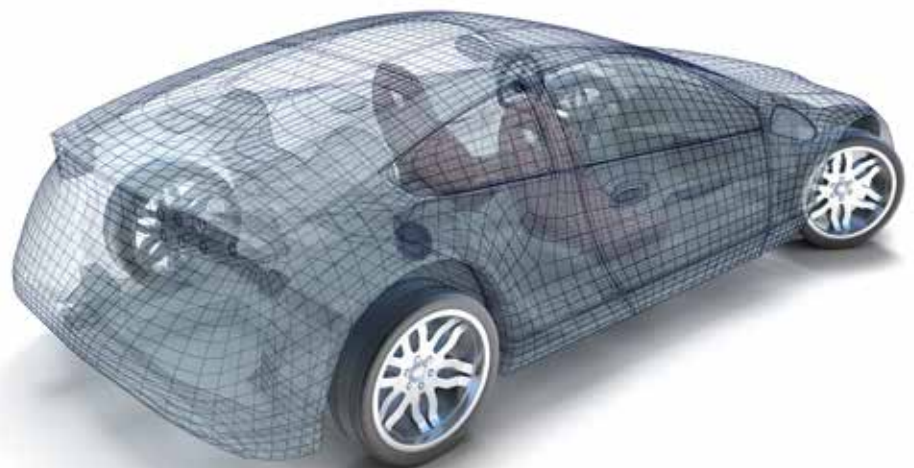
Drivers today are looking for new levels of comfort, safety, efficiency and consumer features in their vehicles. These requirements in the automotive world mirror those of society in general, with the “connected world” and machine-to-machine (M2M) communications being the current buzz.

In newer vehicles, there are multiple cameras (lane departure warning, automatic cruise control, etc.) and multiple thin-film transistor (TFT) screens for satellite navigation, reverse camera, cluster and more. Along with increased computing performance and embedded memory capacity, these new features are driving the need for high bandwidth in the vehicle network, and hence the use of Ethernet connectivity. In addition to these high bandwidth connections, the vehicle is seeing a proliferation of sensors, actuators and motors in control applications. The sensors measure gases (COx, NOx, etc.), various temperatures, vibration, wheel speed, torque, yaw and other parameters to help improve efficiency and safety. Actuators, including relays and solenoids, as well as motors controlled by MCUs, drive pumps, fans, heating ventilation and air control (HVAC), window lifts, the sunroof and more. This extra functionality increases power consumption through two routes: one, based on the direct relationship of weight of added hardware to lower fuel economy, and two, the need for more computing performance, embedded memory capacity and higher bandwidth connectivity, each consuming additional power.

Despite a push to integrate major computing resources into a small number of central domain controllers, the explosion of low data rate small sensors and actuators is resulting in many new Electronic Control Units (ECUs). In figure 1, Strategy Analytics’ data show the projection of the growing number of ECUs in vehicles for each segment. While the average vehicle today has approximately 25 ECUs, some high-end models are already over 100.

The combined results of these trends is a larger in-vehicle network with the wiring loom commonly being the second heaviest component in the vehicle (behind the engine), having over 6 km of copper wire and weighing over 70 kg.

Coupled to the demand for greater computing and networking performance, there is a major push for increased safety within the body network to cope with the growing complexity of electronics and the critical nature of the functions they enable. In addition, as wireless communication to/from the vehicle becomes more prevalent, there is an ever-greater need for security measures within automotive MCUs to prevent unauthorized access to the vehicle network for the occupants’ welfare and to safeguard the intellectual property it contains.



## 2. Networking

Modern vehicles have a large number of ECUs that deliver many functions. Those functions may be distributed among several ECUs with the majority being networked nodes that are connected to one or more system busses.

These ECUs control a range of functions, such as lighting, air conditioning, seats and the engine or transmission. The various bus systems that connect them such as controller area network (CAN), local interconnect

network (LIN) and FlexRay form a distributed network within the vehicle.

In the future, vehicle network architectures will consist of highly integrated domain controllers, which will be interconnected via higher-speed bus systems. The industry trend indicates that Ethernet will be the protocol of choice forming the "backbone" of the domain network and replacing CAN, however, there are some instances of FlexRay. sub-busses with CAN, FlexRay and LIN will provide connectivity

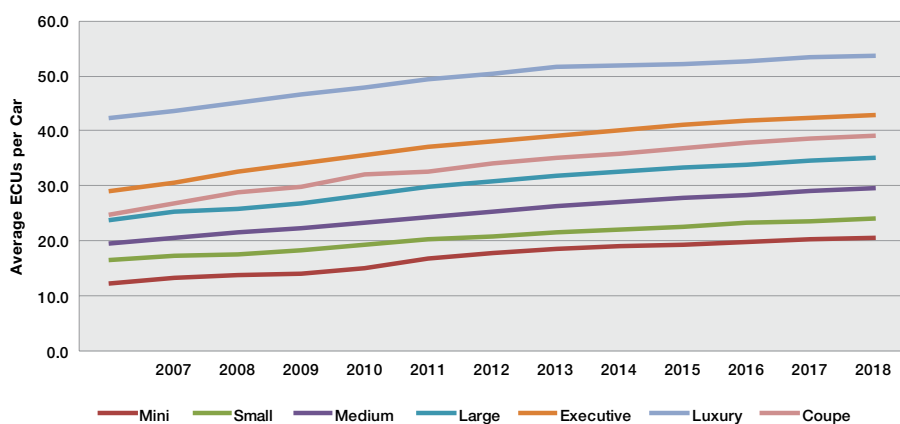
to intelligent nodes within a vehicle sub-domain. Powerful domain controllers will be required to support this highly interconnected architecture.

Figure 2 illustrates an example vehicle network partitioned into separate application domains with associated domain controllers. These domain controllers will require significant amounts of processing power coupled with real-time performance and a plethora of communications peripherals.

The Freescale Qorivva MPC5748G family introduces a highly flexible MCU suitable for operation as an advanced centralized gateway controller, a high-end body domain controller or elements of both. The vast array of communications interfaces, coupled with high-performance levels and low power capabilities make it ideally suited for these operations.

The MPC5748G MCU uses e200 cores based on Power Architecture® technology developed for automotive gateway and high-end, centralized body controller module applications. The device contains two 160 MHz e200z4 cores and an 80 MHz e200z2 core to offer a flexible power-performance solution. The MCU's salient features include 6 MB of embedded non-volatile flash memory and 768 KB of embedded SRAM, in addition to support for revolutionary new low power modes. The feature set includes an e200z0-based hardware security module (HSM) exceeding the requirements of the secure hardware extension (SHE) of the Hersteller Initiative Software (HIS) standard, and a selection of communications, analog and timer modules. The device is a SafeAssure solution (refer to Section 5), has been developed in accordance with the Automotive Functional Safety Standard (ISO 26262) and is targeted at specific safety functions of at least an Automotive Safety Integrity Level (ASIL)-B rating.

**Figure 1: The Number of ECUs is Increasing in All Car Segments**



Source: Strategy Analytics

**Figure 2: Vehicle Networks Arranged by Application Domains**

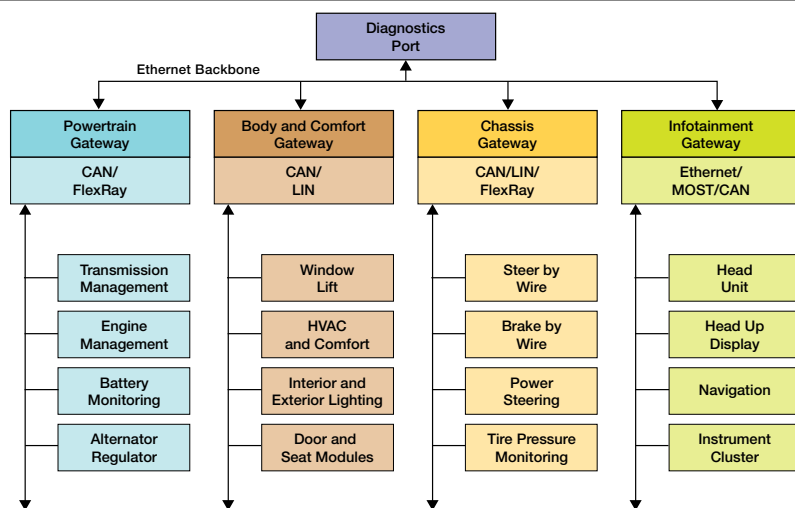


Table 1 illustrates the level of communications interfaces supported by the MPC5748G MCU, further illustrating its suitability as a domain controller in addition to its applicability in high-end vehicle body controller applications.

Due to its multicore design and associated feature set, the MPC5748G MCU is particularly well suited to support multiple applications within a single architecture. A high degree of separation and isolation between different cores and their associated resources permits isolation at the application level. This means that it is possible to dedicate some resources of the MCU, for example a core, subset of the periphery and memory, to one application while retaining another core with its own subset of the periphery and memory for a completely separate application. These MCU features are essential at the vehicle level to break the 1:1 correspondence of a vehicle feature to an ECU. To make vehicle options cost effective and manageable in a complex manufacturing environment, vehicle features need to be enabled by software on a common hardware platform. Another side benefit of this application isolation is the protection it offers the software integrator to collate software from many third-party developers, knowing that they will run independently and autonomously.

The diagram of the MPC5748G architecture in figure 3 shows some of the features that enable such a high degree of application isolation.

To demonstrate these features and the capabilities of the MPC5748G MCU in a practical real-world application, Figure 4 (next page) shows a proposed use case. In this example, the device controls two independent domains:

- A gateway domain
  - Handles classic automotive open system architecture (AUTOSAR) automotive gateway functionality.
  - Has a dedicated CPU and associated memory and peripheral resources.
  - Runs almost independently of IP domain, but is capable of safely and securely exchanging data through shared memory and interrupt messaging schemes
- An IP domain
  - Connects to the Internet and is intended for supporting applications such as distributing in-field flash downloads within the vehicle network.
  - Uses a dedicated e200z4 core, dedicated system RAM, and a portion of the flash array and runs its own operating system (OS) with its own OS timers, watchdog and system resources.

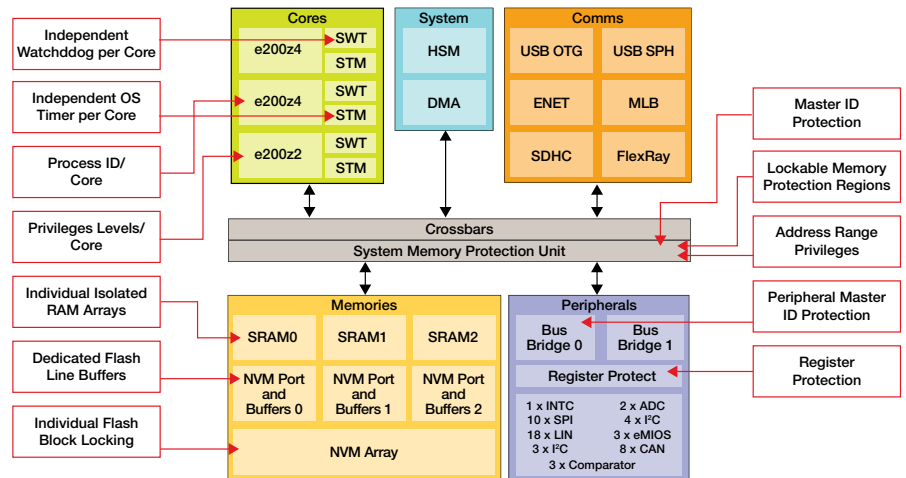
**Table 1: Communications Interfaces Supported by the MPC5748G MCU**

Communications	Bit Rate	Description
FlexCAN	CAN2.0 1 MB/s CAN FD 8 MB/s	<ul style="list-style-type: none"> <li>• CAN2.0B compliance</li> <li>• Mailbox support</li> <li>• FIFO support</li> <li>• Active and passive CAN_FD compliance</li> <li>• Low-power pretended networking filtering on one node</li> </ul>
LINFlex	20kbps	<ul style="list-style-type: none"> <li>• LIN protocol version 1.3, 2.0, and 2.1</li> <li>• 1x master/slave, 17x master supporting LIN</li> </ul>
Ethernet	100 MB/s	<ul style="list-style-type: none"> <li>• Supporting (RMII*, MII** + 1588)</li> </ul>
FlexRay	10 MB/s	<ul style="list-style-type: none"> <li>• Supporting FlexRay 2.1</li> <li>• 128 MB/s</li> </ul>
SDIO (full speed) SDIO (high speed)	25 MHz 40 MHz	<ul style="list-style-type: none"> <li>• Secure digital input output</li> </ul>
USB	480 MB/s	<ul style="list-style-type: none"> <li>• 1 x on-the-go</li> <li>• 1 x host controller</li> <li>• ULPI interface</li> </ul>
MediaLB	2048 fs ~98 MB/s	<ul style="list-style-type: none"> <li>• 3-pin and 6-pin interface</li> <li>• Speed grade up to 2048 fs</li> </ul>
SPI	40 MHz	<ul style="list-style-type: none"> <li>• Serial peripheral interface</li> <li>• Up to four with features for SPI controlled LED drivers</li> </ul>

\*RMII = reduced media independent interface

\*\*MII = media independent interface

**Figure 3: Application Isolation and Protection Mechanisms Implemented on the MPC5748G MCU**



**Vehicle Reflashing**

Flashing and reflashing of the vehicle's software content is an evolving area of advanced automotive electronics. While traditionally the vehicle is flashed under tightly controlled factory conditions and potentially updated during routine vehicle servicing, this concept is being expanded to include more user-convenient, over-the-air updates to vehicles in the field. As shown in figure 5 (next page), the average vehicle has approx 10 MB of flash memory, however, many high-end vehicles will have at least 10 times this number.

To put the scale of this challenge in perspective: modern vehicles now have in excess of 50 MB of embedded flash memory distributed across the ECUs (excluding the infotainment/multimedia segments). Original equipment manufacturers (OEMs) require the ability to safely and securely update some or all of this content in a reliable and convenient manner with minimal impact to the driver. A number of requirements and challenges need to be addressed for reflashing in the field:

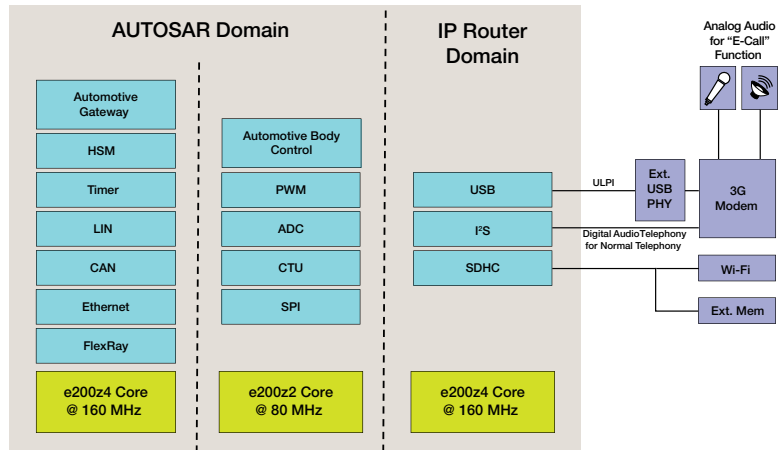
- Safety
  - New software cannot cause any system malfunction
  - The ability to roll back to previous software version is desirable
- Security
  - Updates cannot be intercepted and non-approved updaters do not have access
  - Integrity of downloaded software must be verified
- Transparency
  - Updates in the field must have minimal interference with driver's intended use of the car

Vehicle manufacturers also need the ability to download a new flash image while driving and store it securely for later programming when the car or specific ECU is in a safe operating/ non-operating mode. The MPC5748G MCU is ideally suited to such applications. It contains many features that make it ideal for supporting the range of vehicle flashing and reflashing applications, being able to receive and store the image, then send the image to the relevant node(s). Table 2 (next page) lists some of these features and their associated benefits.

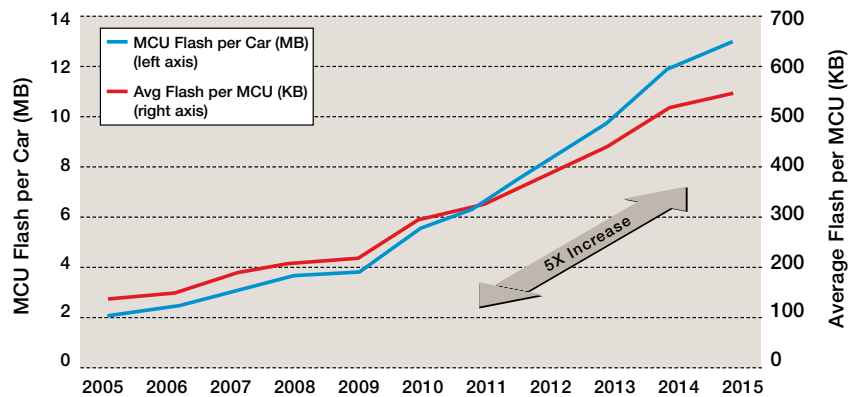
**Advanced Networking**

New applications, such as camera-based parking, in-car TV and driver assistance, result in larger program and data memories. For example, one of the latest high-end car series has more than 1 GB of embedded memory distributed among the 100+ ECUs, while the previous model had less than 100 MB of memory. Due to the increasing number of ECUs per car and embedded memory, there is an increasing need for higher network bandwidth.

**Figure 4: Multi-Domain Operation with the MPC5748G MCU**



**Figure 5: Evolution of Embedded Flash Content in the Vehicle**



Source: Strategy Analytics and Freescale analysis

**Ethernet**

With the projected increase in data volume, increase in embedded memory and move towards larger domain controller architectures, there is a need for a new high-speed interface for interconnectivity. Ethernet is an obvious choice for this high-speed network since it is used extensively in non-vehicle applications and already used in production vehicles. Initially, Ethernet was introduced into the car as a high-performance data interface for diagnostics and software downloading by some OEMs to reduce programming times in production and at service centers. This initiative has propagated to other OEMs and is expected to lead to an ISO/SAE standard that stipulates that Ethernet be used as part of the on-board diagnostic (OBD) interface, replacing

CAN, which was the previous standard. In addition, Ethernet is currently being prototyped as the network for surround camera systems. In this system, Ethernet will be used in normal runtime applications providing a significant upgrade to its use in diagnostics. The catalyst for this change was the reduction in the bill of materials (BOM) cost due to advances in the physical interface that allow electromagnetic interference (EMI) legislation to be met even with very low-cost unshielded twisted single pair (UTSP) cabling. The BOM cost savings is achieved by the use of inexpensive UTSP cabling. The reduced cost and growth of Ethernet usage at one vehicle manufacturer is shown in figure 6 (next page).

Ethernet has further advantages that make it the ideal choice for the backbone network. The key areas are:

- Increased bandwidth options (scalability)
- Possible to stay below electromagnetic compatibility (EMC) emissions limit with low-cost UTSP
- Ethernet is a well-known and mature network structure
- Many developers have Ethernet experience
- Simple integration of consumer devices
- Many suppliers offer hardware and software
- Availability of low-cost and freeware tools

Increased bandwidth options are one of the more significant advantages, since OEMs need scalable vehicle architecture due to the projected future increases in data volume. Ethernet is very strong in this area with 1 GB/s and higher capabilities. In fact, as shown in figure 7 (next page), automotive networking lags consumer electronics by several years since 1 GB/s and 10 GB/s networks are common in non-vehicle networks.

Another major strength of Ethernet is its maturity in the consumer electronics domain, leading to a large pool of developers, knowledge, suppliers, tools and software available that automotive teams can access. The advantages highlighted above, and considering that Ethernet is already used in volume vehicle production, suggest that further proliferation will occur and Ethernet will be the choice for the high-speed backbone between domain controllers.

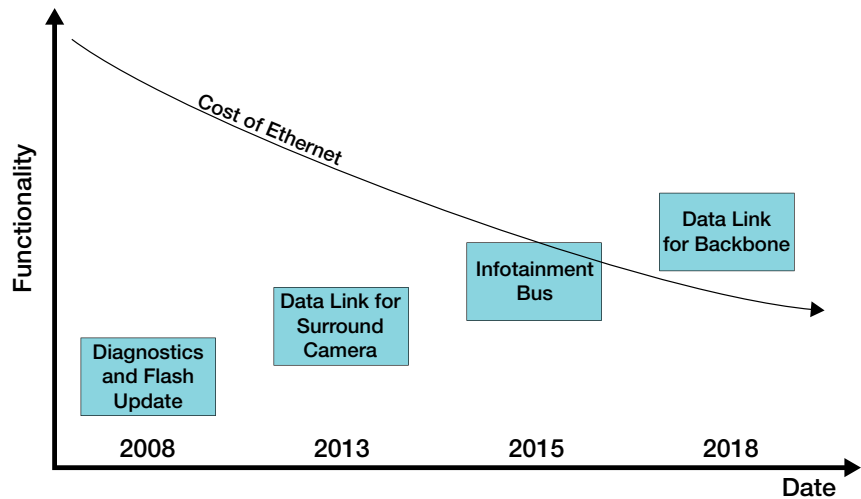
The MPC5748G MCU and its derivatives are very well positioned to meet the future needs of Ethernet. These MCUs support the ENET Ethernet complex and provide the large amount of embedded memory required for Ethernet use in body/gateway applications. The ENET complex used on the MPC5748G MCU supports 100 Mbps, but has an existing upgrade path to support Gigabit Ethernet. The Gigabit version is fully software compatible and is integrated on other Freescale products.

Additionally, the ENET complex has been updated to fully support audio video bridging

**Table 2: MPC5748G MCU Features Applicable to Vehicle Reflashing Applications**

Feature	Benefit
Up to 6 MB flash	• Enough storage to maintain the local application (i.e. BCM or gateway functionality) and also an additional image for another node in the vehicle.
Multiple flash partitions	• Ability to program an area of flash while executing from another area. Permits minimal intrusion of application while storing flash downloads.
Multicore and independent resources	• Ability to manage separate applications responsible for flash download and storage with minimal interference of the main application.
Interfaces	<ul style="list-style-type: none"> <li>• High-speed CAN_FD supported for rapid distribution of flash contents on vehicle network.</li> <li>• SDIO and USB for connection to network IP (Wi-Fi®, 3G etc).</li> <li>• Ethernet for diagnostic connections.</li> <li>• JTAG, UART for serial connection.</li> </ul>
Hardware flash remapping scheme	• Ability to swap internal flash address mapping on the fly. Includes a mechanism to apply remapping upon subsequent resets yet still permits simple rollback functionality.
Low power	• Capable of distributing an image over the vehicle network in low-power "vehicle parked" modes. CAN and LIN nodes active in new low-power unit (LPU) modes.
Fast program flash	• Highly-efficient initial factory programming and in-vehicle re-flashing. Effective EEPROM emulation scheme.
Security	• HSM for validating integrity of downloads and ensuing secure communication on the vehicle bus network.
Censorship	• Robust mechanism for securing the contents of the device NVM, avoiding exposure of updates to hackers.
Robust boot loader	• Non-erasable NVM-based boot loader supporting serial downloads under factory conditions.
Can FD	• Fast download to vehicle and within vehicle network over existing CAN bus infrastructure.

**Figure 6: Use of Ethernet at an Early Ethernet Adopting OEM**

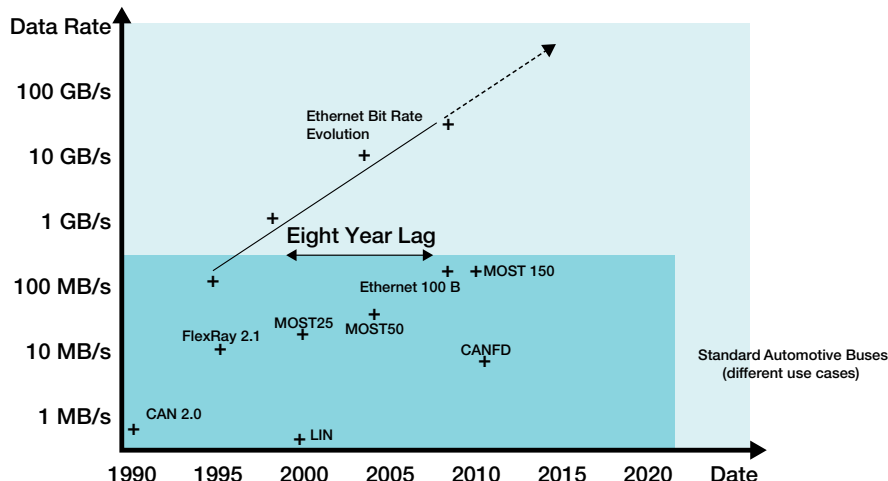


Source: Freescale

(AVB) standards with multiqueue support and traffic shaping to meet AVB quality of service (QoS) requirements. The multiqueue support for AVB can separate different traffic types in hardware, which makes the software driver more efficient offloading the central processing unit (CPU). The multiqueues and traffic shaping also allow separation of application tasks and guarantee that higher priority data is always

transmitted. An example use case would be in a domain controller that combines body and gateway functions. In principle, both can share a single MAC by partitioning the tasks to the multiple queues. This helps ensure that the more important tasks (such as those from the gateway) always get sufficient network bandwidth. Figure 8 (next page) shows a block diagram of the multiqueue ENET complex.

Figure 7: The Bit Rate Lag of Automotive Networks Compared to Ethernet



Source: Freescale

Figure 8: Block Diagram of the Multiqueue ENET Complex

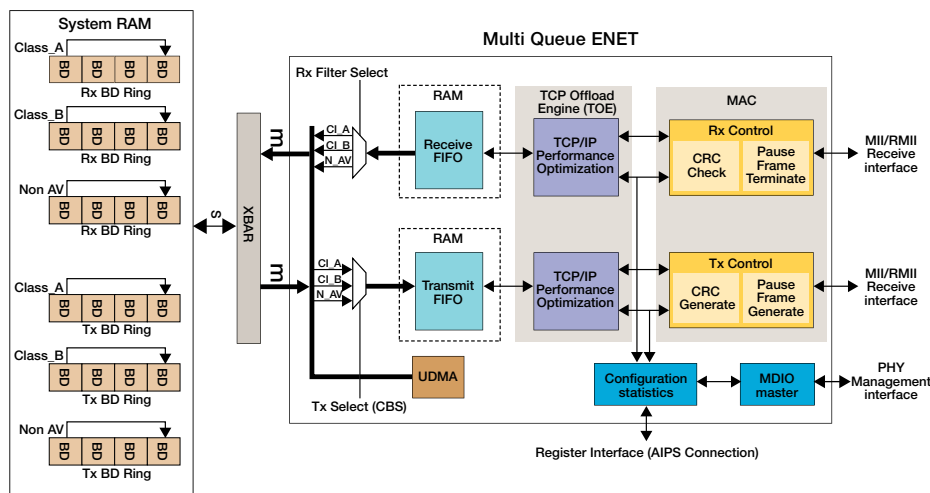
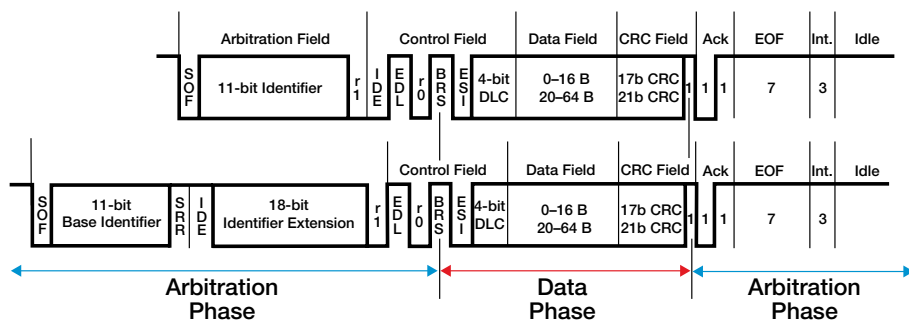


Figure 9: Structure of the CAN FD Frame (CAN FD Base and Extended Formats)



Source: Freescale

CAN FD

The larger amounts of embedded memory required in future MCUs will increase the overall programming time of the vehicle, resulting in higher production and servicing costs. Also, the more complicated ECUs need to transport more information between each other, which requires higher bandwidth on the existing networks in the car. In addition, to transitioning to high-speed data interfaces, such as Ethernet, for diagnostic ports and interconnection of domain controllers (large ECUs), the industry also wants to increase the throughput of the existing CAN2.0 networks, since this evolutionary growth helps preserve current investments. This has resulted in the creation of CAN flexible data rate (FD) (CAN FD - ISO 11898-7) that allows the baud rate of the data portion of the CAN message to increase up to 8 MB/s and the payload to increase up to 64 bytes in order to increase the throughput. Figure 9 shows the structure of the CAN FD standard and extended frame used to improve throughput.

An example of the throughput increase for CAN 2.0 versus CAN FD is shown in figure 10 (next page). Varying data phase transmission rate (4 vs. 8 MB/s) and payload size (8- vs. 64-byte) is also shown.

This example shows that significant improvements (up to 6x) can be made by the deployment of CAN FD, particularly when larger payloads are needed. This can be utilized to decrease programming times and also transport more data between ECUs.

The FlexCAN3 module used in the MPC5748G MCU family supports both CAN 2.0 and CAN FD. FlexCAN3 is a full CAN implementation with a flexible buffering arrangement. All mailboxes are able to support CAN 2.0 and CAN FD formats for both transmit and reception. The implementation is optimised to allow CAN 2.0, CAN FD and interleaved CAN 2.0 and CAN FD, so the user can effectively configure each mailbox individually.

This allows the full support of programming and functional use cases. Figure 11 shows a block diagram of the FlexCAN3 buffer arrangement.

The buffer sizes are a mixture of 8, 16, 32 and 64 bytes to accommodate the different CAN FD use cases. For example, it is anticipated that the majority of 64-byte payload frames will be used for program download.

### 3. Hyper Integration

With the continuous growth in the number of functions and the increasing complexity of the functions themselves, today's automotive systems use architectures where a handful of central ECUs communicate to each other. On the peripheral side, the actuators and sensors are also becoming increasingly powerful embedded MCUs, with communication interfaces, voltage regulators and other application-specific components.

One of the key enabling technologies for this architecture of distributed intelligent actuators and sensors is the high integration level of integrated circuits (ICs). This high integration is the first and most important step towards achieving an efficient and optimized system solution. Instead of the typical two or three ICs, only one IC is able to address the application needs. This brings various advantages, which are discussed later in this section (see The

Value of Integration), but first of all it can reduce the required printed circuit board (PCB) space by half, as shown in figure 12 (next page), by removing discrete analog components like voltage regulator, PHY and high-current drivers. Freescale LL18UHV technology combines high-voltage (40 V) analog, digital logic and non-volatile memory through package or wafer-level integration, allowing customers to realize these advantages. Implemented in the S12 MagniV mixed-signal product family, the system-in-package (SiP) design provides a highly capable and cost-effective solution for many body electronics applications. When designed into a vehicle network with the MPC5748G MCU high-end body domain controller, even higher system-level advantages can be obtained.

### Networking

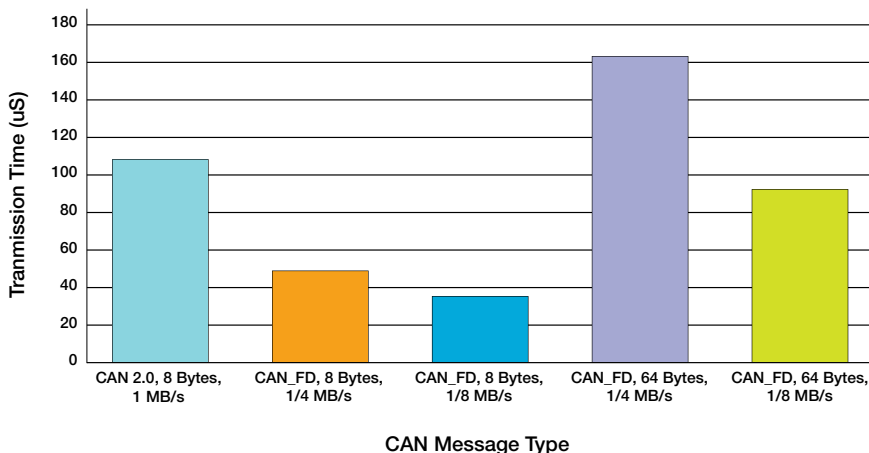
Intelligent and highly integrated actuator and sensor nodes communicate their information via their respective system network. An example of a highly integrated node is a light control switch module that transmits the driver's desire to turn on the headlights to the actuating lighting ECU. Most body electronics applications use the LIN and/or CAN communication interfaces. Application requirements such as latency and bandwidth, as well as cost, influence the selection of a specific interface. Since the

actual communication physical layer (PHY), mostly driven by electric and electromagnetic requirements (ESD, EMI, EMC) is not a negligible portion of the device area, normally either a LIN or a CAN PHY is integrated according to application needs. The general-purpose S12ZVL family of MCUs with an embedded LIN PHY and the S12ZVC family with on-chip CAN PHY address those needs.

In addition to LIN and CAN protocols, other communication interfaces for the powertrain or chassis domain, such as single edge nibble transfer (SENT) or peripheral sensor interface 5 (PSI5), are gaining interest to further reduce network costs. For example, the use of PSI5 instead of LIN reduces the number of wires and connector pins from three (LIN, VBAT, GND) down to only two (supply, GND). Even though PSI5 needs to modulate the data onto the supply line, the savings on the harness and connector side may be sufficient to account for the higher requirements on the electronics side.

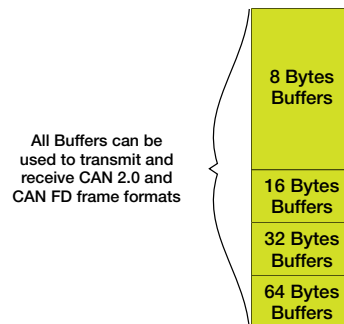
In spite of the continued development of lower cost protocols, the overall trend to LIN and CAN-based nodes has been observed for many years in automotive sensors and actuators, especially in body applications. According to Strategy Analytics, by 2018, the number of LIN nodes will exceed one billion and the number of CAN nodes will exceed

**Figure 10: CAN Frame Type vs. Transmission Type (Where x/y Represents Data Rate in Arbitration/Data Phases of the CAN Frame)**



Source: Freescale

**Figure 11: FlexCAN3 FD Buffer Arrangement**





the two billion mark. The average number of nodes per vehicle will be around 10 LIN nodes and approximately 20 CAN nodes. With 17 percent compound annual growth rate (CAGR), the expected market growth for LIN nodes is significantly higher than for CAN nodes with 13 percent CAGR. This shows that simple functions are increasingly implemented in LIN nodes. Figure 13 shows the significance of this growth.

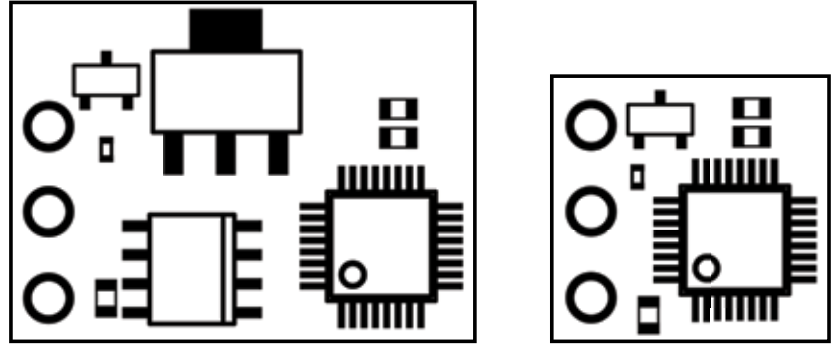
**Mechatronics—“Second Level Integration”**

Mechatronics is the combination of multiple engineering disciplines including mechanical, electrical and control engineering into one system or product. The combination of mechanics with electronics and software allows the design of very dedicated, optimized and therefore cost-effective systems into very limited space, while at the same time increasing the functionality and flexibility (programmability) of the systems. This aspect of integration does not end at the IC. However, embedding electronics in a mechatronic system requires the right building blocks and technology to produce them, application know how and application-specific ICs (ASICs).

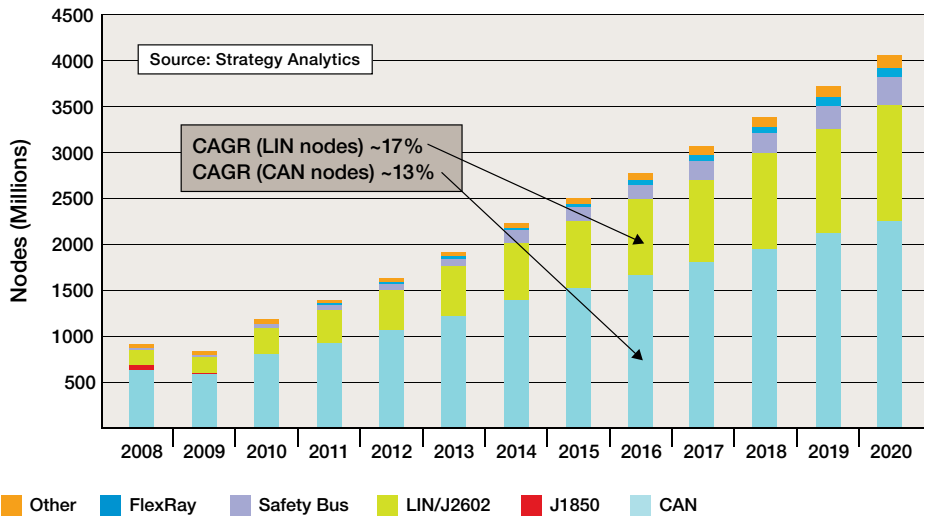
As shown in figure 14, the previously mentioned LL18UHV technology can be considered as an industry-leading technology for this trend. LL18UHV technology is an extension of proven low-leakage technology (LL18), which is already implemented in more than 200 million S12 MCUs worldwide. By adding only a few process steps for the LL18UHV (buried n-well and deep link for isolating transistors) to this base technology prior to the CMOS + NVM processing steps, basic transistor parametrics and proven reliability are unaffected.

The technology is developed to sustain 40 V for load dump from a 12 V car battery. These high-voltage circuit elements are typically not present in technologies used to manufacture MCUs, but are mandatory to integrate single-package devices into mechatronic actuator and sensor nodes.

**Figure 12: Increased Integration Significantly Reduces PCB Space**

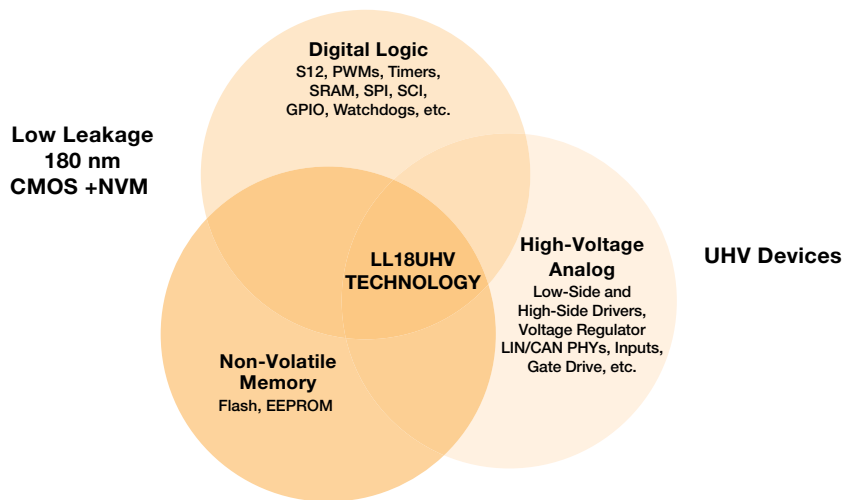


**Figure 13: The Growth of CAN, LIN and Other Nodes in Vehicles**



Source: Strategy Analytics

**Figure 14: Mechatronics Capability in LL18UHV Technology**



### Building Blocks

Typical mechatronic sensor or actuator nodes share the following common building blocks:

- MCU with standard peripherals including pulse width modulation (PWM), timer, analog-to-digital converter (ADC), general-purpose input/outputs (GPIOs), etc.
- Robust voltage regulator to achieve the appropriate power supply (such as 5 V) from the +12 V car battery
- Communication Interface consisting of physical layer (such as LIN or CAN PHY) and actual protocol controllers (SCI, MSCAN)

For the highest levels of system integration, other application-dependant blocks are added, including low-side drivers, high-side drivers, pre-drivers, high-voltage inputs or other analog and high-voltage modules as shown in figure 15.

### The Value of Integration

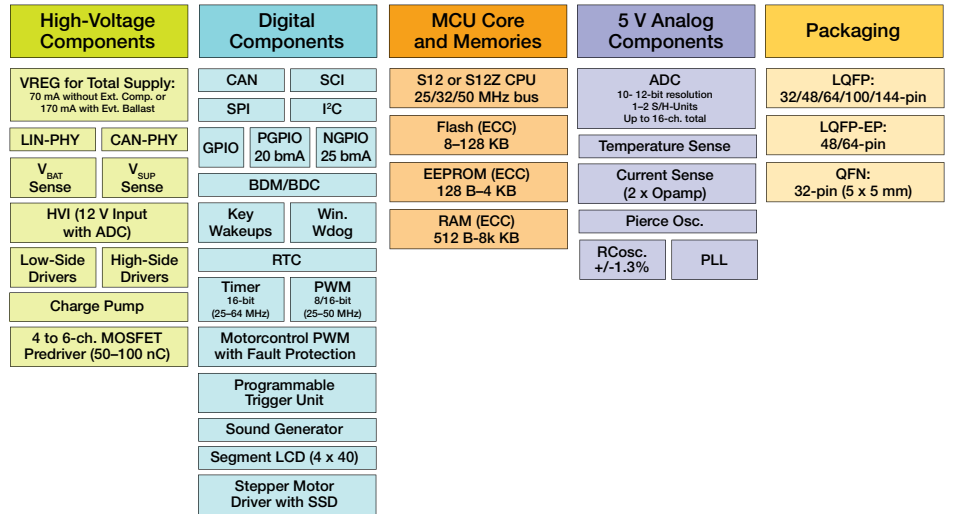
Putting a value on integration is always difficult. It certainly is more than just product cost. For example, consider a window lift system that uses an embedded MCU with Vreg, LIN PHY and more, compared to a traditional electromechanical system. Determining the actual value of the integration requires a rather thorough investigation. There are more obvious direct savings, for instance:

- Reduced BOM (less components)
- Smaller PCB size
- Reduced design space (often strictly regulated by OEM)
- Reduced size, weight, harness, less materials
- Easier mechanical interfaces
- More robust design (e.g., simpler mechanics, due to monitoring/supervising electronics)

There are also several less obvious, and not easily evaluated, indirect savings, including:

- Manufacturing cost (fewer components to mount, less testing required compared to multiple ICs)
- Improved quality (fewer solder joints, pre-tested subsystem)

Figure 15: Application-Dependent Building Blocks



- Easier logistics (fewer parts, less effort on sourcing, ordering, storage, tracking)
- Faster time to market (pre-engineered sub-system, software changes to address new, last-minute, application requirements)

When all of these elements result in a lower system cost, then integration—the integrated circuits and the mechatronic system around it—makes sense. Separate from the purely cost driven motivation to integrate, there are also some applications where space is so limited that highly integrated ICs are mandatory, so the value of integration is inherent. Being able to integrate all of the required building blocks, including robust high-voltage capable modules, onto a monolithic chip as well as adding a LIN interface, Vreg and MCU to a single 20 mA RGB LED for ambient lighting opens the door to new applications.

### Power Consumption

Current draw and energy consumption are very critical aspects of the vehicle's fuel consumption, driving range, CO<sub>2</sub> emissions and more. The importance increases with increasing the number of nodes per car. Besides the low power modes and cyclic monitoring techniques, discussed in section 4, distributed actuator nodes enable smarter load management. For example, it is possible to reduce the consumed power of an HVAC blower by electronically controlling the motor, rather than using a resistor for the speed control.

Integration is one of the most critical design tools to satisfy the growing demand for mechatronic system actuator and sensor nodes in modern vehicle architectures. At the same time, the calculation performance and memory needs are constantly increasing. High-performance 16-bit MCUs with up to 256 KB of flash memory are needed to address these application requirements.

### 4. Power

Enhancing the driver's experience is a fundamental goal of car manufacturers that typically drives the growth of electrical nodes and increases power consumption. This energy is certainly not free and, in fact, can be directly linked to fuel consumption:

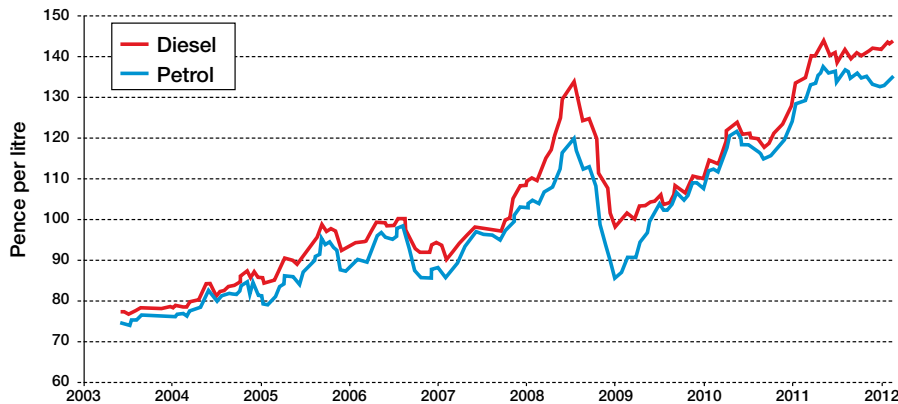
100 W electrical power ~ 0.1 litre/100 km

If the vehicle weight is also considered:

50 kg weight ~ 0.1 liter/100 km

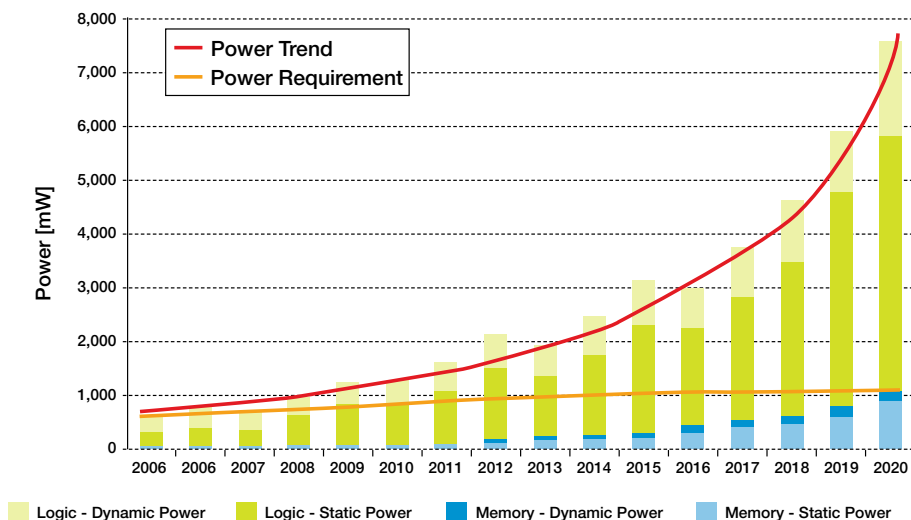
Both these factors demonstrate why minimizing electrical energy consumption and the weight of ECUs are essential in the drive to improve fuel consumption. Considering other factors, such as legislation and the desire to increase the range of electrical vehicles, power consumption is clearly a critical aspect of modern automotive design. As indicated in figure 16 (next page), the long-term trend for the cost of fuel increases the importance of any vehicle aspect that can reduce fuel consumption.

Figure 16: UK Fuel Prices



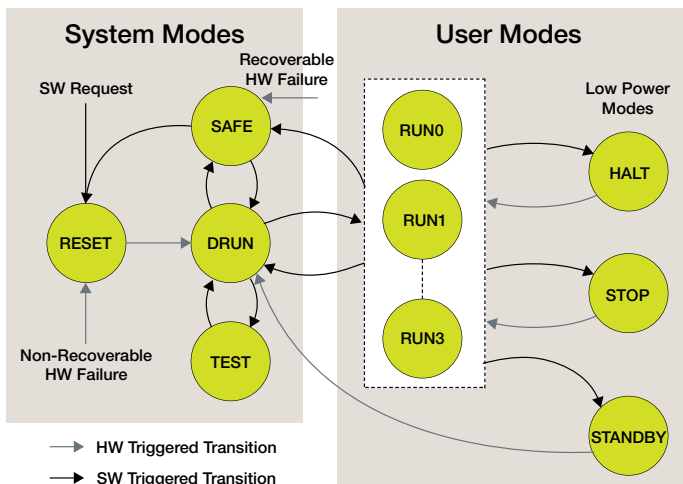
Source: The UK Department of Energy and Climate Change

Figure 17: Power Trends through 2020



Source: International Technology Roadmap for Semiconductors (itrs.net/)

Figure 18: A Variety of Operating Modes Allow Optimum MCU Power Consumption



*Emerging Low-Power Architectures*

Addressing the performance requirements of the evolving automotive industry requires investment in future technologies, but these technologies introduce many challenges for system designers, including coping with increased power consumption.

While each technology step helps devices execute faster to meet customer expectations, it also requires an innovative way of thinking to solve the power crisis. Demands for more features, increased safety and faster performance, as well as competitor differentiation and reduced weight and costs, must be addressed within a tighter power budget. Figure 17 shows the increasing demand for power.

*Thinking Differently*

Traditionally, all the elements of the system are active and powered to meet the demands of the driver. However, the numerous electrical nodes with increasing design complexity leads to a huge increase in the system power demands. Philosophies starting to sweep the industry include approaches that only power the absolute minimum number of nodes at any particular moment. Techniques from partial networking, pretended networking and super-nodes through to fully distributed integrated nodes are all gathering momentum. MCUs and distributed integrated sensors and actuators must support these emerging protocols, but also contribute their own unique approaches to compete in this challenging area.

*MCU and Distributed Integrated Sensor/ Actuators Power Modes*

Older MCUs contained only two basic states: either ON or OFF. Advanced MCU technologies allow several operating modes to cope with concerns for power consumption. As shown in figure 18, today's MCUs operating modes include:

RUN: Traditional full-execution mode, usually the highest consumption mode

HALT: All MCU elements powered, main elements clock gated

STOP: All MCU elements powered, only a subset operational

STANDBY: Only a small subsystem powered, main areas of device power gated special modes, such as RESET, TEST, etc.

The first Freescale products to introduce modes such as STANDBY into the automotive arena were part of the MPC564xB/C family. This dictated a different mindset to address the entire power reduction ecosystem, requiring applications developers to create special RAM-based routines to obtain the lowest power consumption. Fortunately, the industry has risen to the challenge and taken power consumption to new lows, proving that hardware architecture designers and software developers can work closely together to yield the optimum results.

**Next-Generation Power Modes**

While the MPC564xB/C family has been highly successful in meeting low power and other system requirements, Freescale has recognized that it is time to take solutions to a new level. Working with leading industry partners and customers, a more advanced power management concept has been introduced. The main initiative has been in four areas:

1. Introduction of a Low Power Unit (LPU)

The LPU has emerged as a real product differentiator where full RUN performance is not always required. It is an aggressive mode of device operation, allowing large sections to be completely powered off while supporting the full execution capabilities of a processor core.

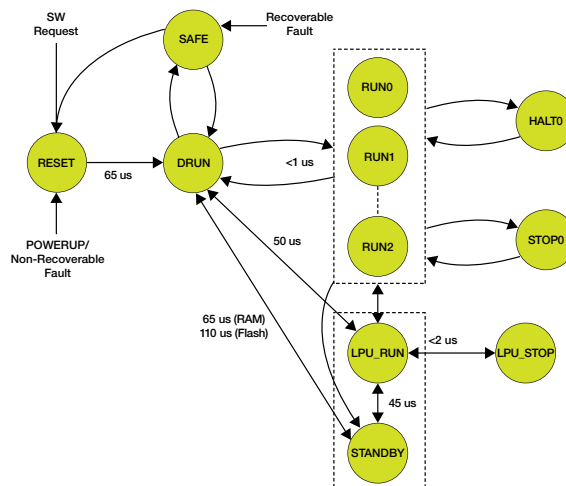
2. Combination of an analog comparator with a periodic timer

Multiple application profiles only require the periodic sampling of input pins. With a traditional approach and even on devices such as the MPC560xB/C/D MCUs, this can only be realized by moving to the device's full RUN mode. However, by intelligently interconnecting several analog comparators with an on-chip timer, all of this functionality can be realised within the STANDBY mode. This type of revolutionary approach allows aggressive low power consumption to be achieved

3. Pretended Networking

Pretended Networking has been driven by an automotive consortium and in response, Freescale has introduced an extension of the LPU modes to adhere to this emerging standard.

**Figure 19: Low Power Modes in the MPC5748G MCU**



4. Highly adaptable Distributed Integrated Sensor/Actuators

These are flexible device solutions, offering ways to distribute computing performance but also help reduce vehicle weight. (Refer to the earlier section, "Mechatronics –Second Level Integration.")

**LPU Modes**

The LPU initially introduced in the MPC5748G MCU family of products allows the application developer to choose between several new as well as traditional operating modes:

1. RUN

- o Full support for max speed/max Idd mode
- o All modules/flash powered, clocking optional

2. STOP

- o Main peripherals' states retained
- o LPU peripherals' states retained
- o Cores (e200z2 and e200z4) powered on, state retained but clock gated

3. LPU\_RUN, LPU\_STOP small micro system:

- o CAN0, LIN0, SPI0, 10 bit ADC, timer, etc.
- o Reduced frequency execution mode
- o Main cores/platform/flash, phase-locked loop (PLL), etc. all power gated off
- o Large parts of the SoC inactive

4. STANDBY

- o Required to support 8 KB RAM up to 256 KB of system RAM
- o Also supports wakeup logic, application programming interface (API)/real-time clock (RTC), 32 KHz SXOSC, 8–40 MHz FXOSC
- o Analog comparator sub-system

**Solving Customer Requirements Using Innovative Solutions**

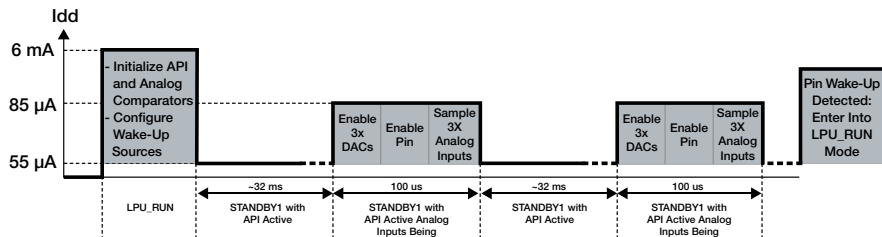
A typical application requires the periodic sampling of a number of analog inputs. As shown in figure 20 (next page), the introduction of the analog comparator and its unique capability to operate from independent analog references provides a straightforward implementation solution.

Effectively, the analog inputs are monitored on a periodic basis, thereby minimizing current consumption. Taking this innovative approach even further, the analog circuitry is also only periodically powered at precisely the time that the sample is required as shown in figure 21 (next page).

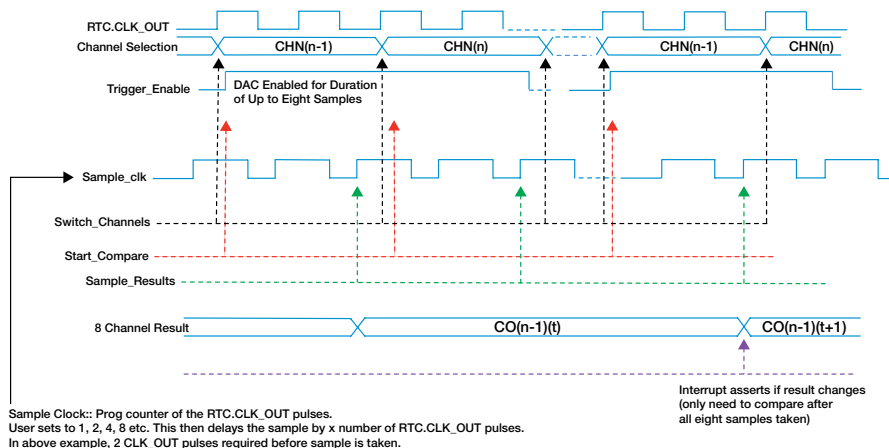
The combination of these two design approaches allows a fully autonomous solution to be realized, reducing current consumption as low as almost 50 µA while still maintaining a portion of RAM. The steps for a fully autonomous solution include:

- Configure the API to generate a 200 ms wake-up output

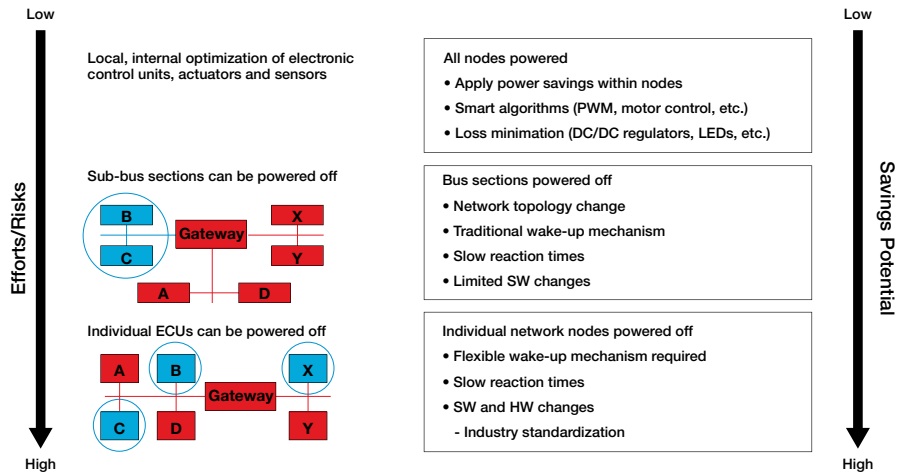
**Figure 20: Periodic Sampling of Analog Inputs Using an Analog Comparator**



**Figure 21: Periodic Powering of Analog Sampling**



**Figure 22: Power Consumption Reducing Solutions through Partial, Pretended and Locally Optimised Networks**



- Configure the API to 1ms period (RTC.CLK\_OUT)
- Configure inputs to be read inside the ANL logic
- Software enters the low power STANDBY mode
- API is free-running
- After 200 ms API asserts a 'trigger enable' to the comparator
- Read all inputs
- If different, wake-up else wait for next 200 ms time interval

**Partial and Pretended Networking**

Traditionally, an entire car network would be powered and operational but there are many examples where this is not required. For instance, when driving, some functions, such as seat movement, can be restricted. In this case, partial networking allows the complete shutdown of an ECU that is independent of the other ECUs on the network. The bottom level of figure 22 shows this methodology. There are other network solutions to help reduce this type of power consumption, depending on the network topology.

In a pretended networking approach, elements of the network determine that the level of activity has significantly decreased. Once this decision has been made, the ECU places itself in a lower power state but "pretends" that it still has a network presence. As soon as the status is required to change, the node quickly re-establishes itself within the network.

To support this type of network solution, Freescale has enhanced the CAN module to work efficiently with this emerging aspect of the AUTOSAR standard. This includes:

1. Introduced a CAN module into the LPU on the MPC5748G family
  - a. Options to wake up the LPU or full system upon valid message reception
  - b. Allows most of the device to be completely powered off
2. FlexCAN ID filtering scheme (match on exact ID) enhanced to include
  - a. Hardware (HW) ID range filtering
    - i. Match on ID above or equal to a target
    - ii. Match on ID below or equal to a target
    - iii. Match on a range of IDs
  - b. Message ID and payload filtering
    - i. Exact ID and payload
    - ii. Exact ID and different payload
  - c. Message occurrence filtering
    - i. Specific ID message match occurring X times
    - ii. Specific ID message missing for a defined quantity of time

### 5. Functional Safety

With their direct impact on human well-being, all electronic systems are experiencing increasingly stringent requirements. Designing systems inside the vehicle to meet the functional safety criteria is a challenging job, especially with increased application complexity combined with time to market urgency. The challenge is to architect the system in a manner that prevents dangerous failures from occurring or sufficiently controls them when they occur. Traditionally, functional safety standards have been applied to the vehicle safety systems, such as airbag or advanced driver assistance systems (ADAS). However, it is clear that many nodes around the car could have a significant effect on occupant well-being if a failure occurred, so these standards are now proliferating throughout the vehicle.

ISO 26262 is the adaptation of the IEC 61508 standard for electrical/electronic systems within road passenger vehicles. The standard addresses architectural and functional aspects as well as procedural aspects, including safety lifecycle to avoid and control faults considering both systematic and random HW faults.

Quoting from the standard, functional safety is about achieving “absence of unreasonable risk due to hazards caused by malfunctioning behavior of E/E systems” where hazards are defined as “potential sources of harm” and harm is defined as “physical injury or damage to the health of people.” Failures are the main impairment to safety:

- Systematic failures: “Failure, related in a deterministic way to a certain cause, that can only be eliminated by a change of the design or of the manufacturing process, operational procedures, documentation or other relevant factors.”
- Random HW failures: “Failure that can occur unpredictably during the lifetime of a hardware element and that follows a probability distribution.”

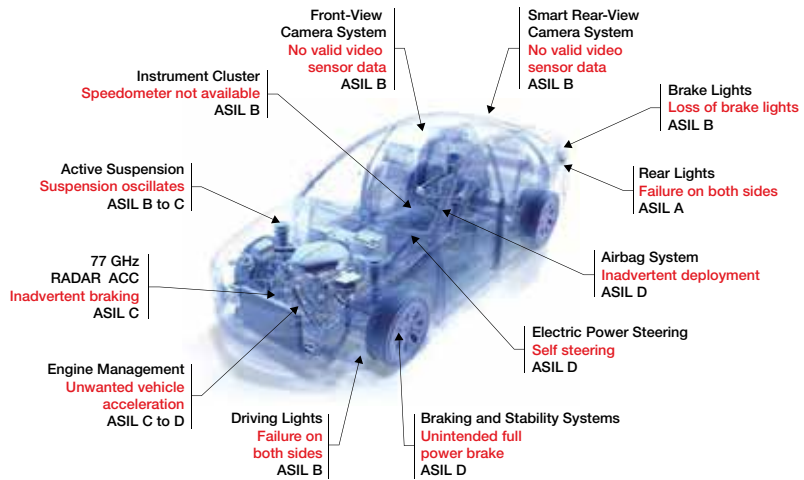
ISO 26262 specifies ASIL ratings at one of four levels (A to D) to identify the necessary requirements of the standard and the safety measures to apply for avoiding an unreasonable residual risk, with D representing the most and A the least stringent level. The appropriate ASIL is determined through hazard

**Table 3: Determining the Appropriate ASIL for an Application Function**

Class of Severity	Class of Probability of Exposure Regarding Operational Situations	Classes of Controllability		
		C1 (Simple)	C2 (Normal)	C3 (Difficult, Uncontrollable)
S1 (Light and moderate injuries)	E1 (very low)	QM	QM	QM
	E2 (low)	QM	QM	QM
	E3 (medium)	QM	QM	A
	E4 (high)	QM	A	B
S2 (Sever and life threatening injuries, survival probable)	E1 (very low)	QM	QM	QM
	E2 (low)	QM	QM	A
	E3 (medium)	QM	A	B
	E4 (high)	A	B	C
S3 (Life threatening injuries, fatal injuries)	E1 (very low)	QM	QM	A
	E2 (low)	QM	A	B
	E3 (medium)	A	B	C
	E4 (high)	B	C	D

\* QM (quality managed): No requirements from standard applied explicitly

**Figure 23: General Consensuses of Automotive OEMs and Tier 1s on the Appropriate ASIL Rating**



analysis and risk assessment performed at the vehicle level considering the portion of the mission profile for which a failure in the system may lead to a harm, (i.e., the exposure), the ability of the driver to cope with the system failures and avoid this harm, (i.e., the controllability) and the human consequences if the controllability actions fail (i.e., the severity). Table 3 shows classes of controllability versus severity and failure probability. Figure 23 identifies the ASIL rating for several vehicle systems.

The ASIL rating is applied at the system not component level, however the MCU plays a key role in the determination. Freescale has

developed many innovations in its chassis and powertrain MCUs that have now been adopted into body electronics products. Traditionally, lockstep-based dual-computational core architecture was used, including redundancy of critical elements, to reach the highest ASIL ratings. For an ASIL-A device, there is no need for a second core. However on an ASIL-B device, such as the MPC5748G MCU, that does not replicate a second core in lockstep mode, an alternate approach that can be adopted. Safety relevant functions can run on two cores at different times, achieving temporal decoupling as well, allowing the diversity in implementation on a different core.

**Table 4: A Comparison of ASIL-A to ASIL-B MCUs**

Safety measure	Feature	ASIL A (S12ZVL, S12ZVC)	ASIL B (MPC5768G)
Implementation diversity	Compute resource	Single-core	Multiple asymmetric non-lockstep cores
	RAM	Single array	Multiple RAM arrays, in different power domains
	Communications	Single instantiation of most comms modules	Multiple instantiations of comms modules
Monitor units	Clock monitor	Yes	Yes
	Voltage monitor	Low-voltage monitor	Low and high-voltage monitor
	Fault collection/control	Distributed	Consolidated
Error detection and correction	End-to-end bus transaction ECC	No	Yes
	SRAM and flash ECC	Yes (SECDED)	Yes (SECDED)
	RAM column muxing	No	Yes
	Flash margin read	Yes	Yes
	Temp sensor	Yes	Yes
Latent fault detection	Logic test	User software	L-BIST
	Memory test	User software	M-BIST
Operational interference protection	Register protection	Yes	Yes
	Memory access protection	Illegal address interrupt	System MPU
	CRC	Software	Hardware
	Watchdog	COP	Window watchdog
	Flash access protection	Accidental and unauthorized write access prevented	Accidental and unauthorized write access prevented
Robust technology		180nm CMOS with Flash and UHV Analog	55 nm CMOS with flash

The focus of the built-in functional safety features on an MCU is on detecting and mitigating random hardware failures: single point faults, latent faults and dependent faults. The target percentage of detectable faults increases for the different ASIL ratings from 60 percent for A to 90 percent for B, 97 percent for C to 99 percent for D.

Considering the intended applications, the specific technology attributes and the design features, table 4 shows a comparison of the built-in functional safety features employed on ASIL-A S12ZVL/S12ZVC MCUs and the ASIL-B MPC5748G MCU.

Functional safety is more than just the hardware features of an MCU. Safety requirements state the specific safety-related detail which will be implemented as part of the software and hardware design, and it is this combination of software and hardware design that provides a functional, safe system. With considered design, ASIL-D systems can be developed from QM products, but the

converse is also true, ASIL-D rated products can be designed carelessly into systems which will never achieve an ASIL rating. Functional safety applied to the development of an MCU is about strictly adhering to the prescribed ISO 26262 process, using robust proven software, and providing the support to allow system development engineers to use the device hardware features to achieve the desired system level safety integrity.

The Freescale SafeAssure functional safety program provides exactly this support. SafeAssure products help simplify system-level functional safety design and standard compliance and come with a rich set of enablement collateral facilitating failure analysis, hardware and software integration. Moreover, SafeAssure products provide a clear support interface to help ensure that Freescale addresses customer needs at each step of the system design and compliance process.

## 6. Security

Together with other general trends such as increased system safety, there is a strong push for advanced security features in automotive semiconductors. The new MPC5748G family supports these customer security requirements with a new security engine, the hardware security module (HSM) and other security modules.

Before discussing these modules, two use-cases, secure communication and secure flashing, demonstrate the need for security features.

### Secure Communication

As highlighted previously, safety is a very important trend and security is required to establish a safe system. For this reason, many vehicle OEMs have started to protect safety relevant CAN message with cryptographic algorithms. Some encode the whole message body, while others prevent modifications of the message with a cipher-based message authentication code (CMAC) value. The CMAC works like a secure checksum, only the owner of the security key is able to produce the right CMAC value. For this process, the CMAC identifies the communication partners.

Figure 24 (next page) shows secure data exchange among three MCUs.

In this example, the MPC5748G MCU is able to send secure messages to the MPC564xB/C and the S12 MagniV device. Two communication groups are created: blue and red and nodes in each group that share the same cipher key. No other CAN nodes are able to send a CAN message with a valid CMAC value to one of these nodes. In the MPC5748G MCU, cipher keys are managed by the HSM, which could take over additional tasks such as sending the message itself. For the S12 MagniV S12ZVC MCU, it is possible to enable flash memory and debug protection. This significantly reduces the possibility that the cipher key is exported from the device.

### Secure Flashing

Secure flashing is very similar to the previous use case. In this situation, the MPC5748G MCU receives the firmware for several sub-nodes over Ethernet and distributes the right firmware to them. Secure communication to the OEM server is established by the HSM based on a public-key algorithm implemented in software on the HSM. Later, for the real firmware download and distribution task, the advanced encryption standard (AES)-128 block can be used to increase performance.

### The New Security Architecture

MPC5748G MCUs have a comprehensive set of customer-configurable security features designed to protect code and data from unauthorized access.

The security features include:

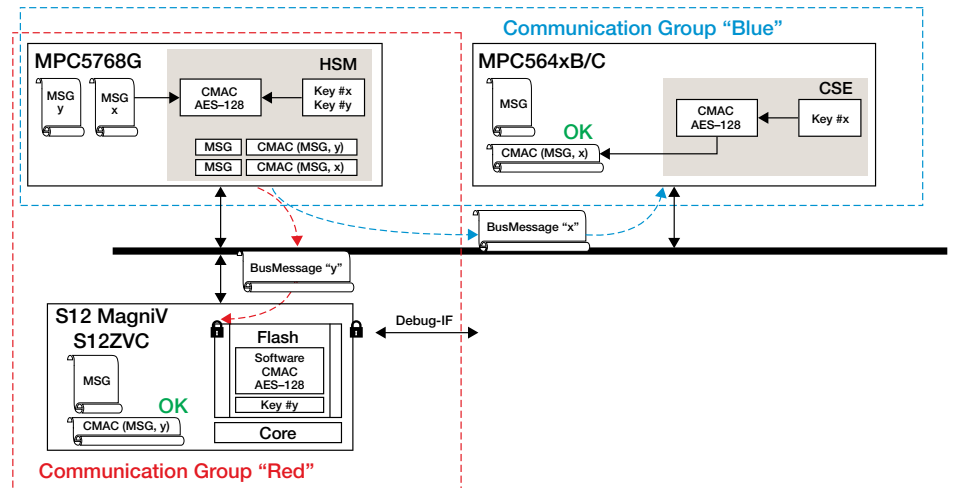
- Device censorship based on lifecycle model
- Memory security features:
  - NVM-censorship support, password protection, and one-time-programmable (OTP) flash memory areas, flash erase counter and tamper detection
  - SRAM and caches initialized to a constant value after reset
- Unique ID for each device
- Secure watchdog timer
- Basic debugger restrictions (on/off via censorship mode) and a secure debugger interface
- Trusted/secure boot support
- Hardware security module

Figure 25 shows the different security layers to protect system memory.

The three main security modules and their functions are:

- The password and device security module (PASS) receives password challenges and determines their validity. It also maintains the device security and access states.

**Figure 24: Sending Secure Messages Between MCUs with CMAC**



**Figure 25: Implementing Increasingly Higher Security in MCUs**

High Security ↑ Low Security	Tamper Detection Module (TDM)	Factory Test Mode Disable Enables the customer to disable the device's ability to be placed in factory test mode. OTP Flash Configure any flash block as OTP so it can only be programmed once. Configurable Tamper Detection Tamper detection acts as an erase counter.	Intended to provide protection against accidental changes to code and data
	Password and Device Security Module (PASS)	Debug Disable Overrides all other debug controls and disables the debug interface. The debug disable feature has higher priority than censorship, JTAG password, and secure debug-IF Secure HSM Debug Customer is able to implement his own system protection scheme and access protocol (e.g. PK based) on the HSM. The HSM is responsible to grand device debug access. JTAG Password (256-bit passwords) Secure Read Protection Read protection; group base (code, data, secure code, secure data). Secure Write Protection Write protection on individual block base; up to 4x 256-bit passwords required to change. Device Censorship (16-bit Password) Traditional protection mechanism for all Freescale MCUs. It is possible to censor and uncensor the device multiple times.	
	Platform	MPU	
	Flash Memory Controller	Non-Secure Write Protection	
	Flash Memory	Flash Blocks Must Be Explicitly Selected for Erase Operations	
		Application Code and Data	

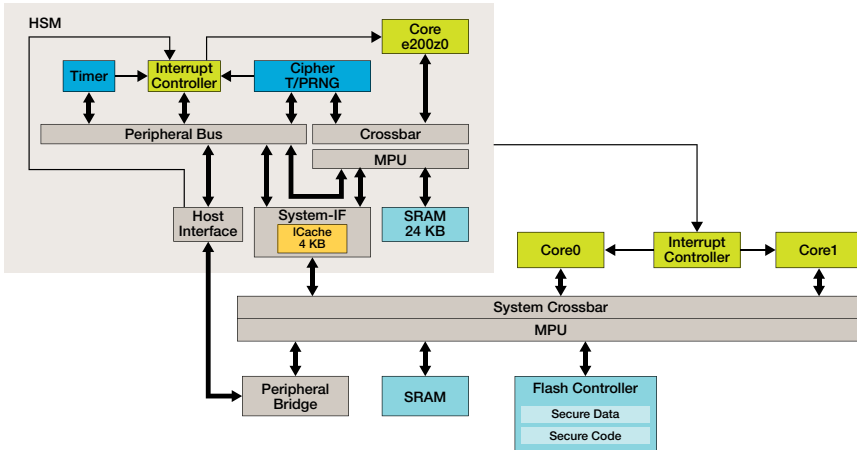
- The tamper detection module (TDM) provides a type of flash memory write protection mechanism that forces software to write a record associated with one or more blocks in a tamper detection region (TDR) before the block(s) can be erased.
- Hardware security module is the second generation of automotive security modules. The first module is the crypto service engine (CSE) in the MPC564xB/C, which implements the HIS SHE V1.1 specification. The HSM is very similar from top level view but offers much more performance and flexibility. The main difference to the CSE is that it is freely programmable by the customer.

As shown in the block diagram of figure 26 (next page), the HSM has its own core and memory blocks for data and instructions. The code is executed from dedicated flash sections, which are only accessible by the HSM.

The HSM is able to access the whole address range of the main MCU. In addition, it is possible to control and service peripheral modules directly by the HSM. This could be useful, if a CAN or Ethernet stack have a secure communication stack.



Figure 26: Hardware Security Module and its Connection to Other MCU Blocks



### Conclusions

With the ever-increasing levels of comfort, safety, efficiency and consumer features in vehicles, carmakers and their electronics suppliers must cope with the conflicting demands of electronics complexity versus power and weight.

New MCU products with higher levels of integration, performance and connectivity will result in a total reduction of the number of components and wiring which will help reduce vehicle weight and thereby enable more efficient vehicles. Additionally, innovative low-power modes help reduce current consumption of the individual components as well as to facilitate a reduction in network power.

One network standard gaining momentum in automotive applications is Ethernet. Freescale is convinced that Ethernet will be widely adopted for high-bandwidth automotive applications, but is unlikely to replace existing and application specific protocols of CAN, LIN, SENT and PSI5.

Implementing MCU solutions that meet automotive requirements involves innovative designs from experts with a long-term commitment to the automotive market. Freescale continues to demonstrate its automotive leadership through innovative body electronics solutions including:

- Qorivva MPC5748G MCU family for centralized gateway and/or high-end body domain controller applications
- S12 MagniV S12ZVL and S12ZVC MCUs, using the hyper integration LL18UHV technology, for smart sensor and actuator nodes

The wide adoption of functional safety within the chassis and powertrain areas of automotive has extended across into body electronics, with new body MCUs from Freescale developed in line with the ISO 26262 process and part of the SafeAssure program. Finally, with the growing internal and external connectivity, increased network traffic and increased vehicle embedded memory capacity, there are higher security demands and Freescale MCUs provide the trusted, hardware-based foundation to help ensure that automotive systems remain secure.



For more information, visit [freescale.com](http://freescale.com)

Freescale, the Freescale logo and Qorivva are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. and Tm. Off. MagniV and SafeAssure are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. © 2013 Freescale Semiconductor, Inc.

Document Number: BODYDELECTRWP REV 0