Technical Article

Automotive Gateways: the Bridge between Communication Domains

Heather Lothamer

Cars use communication protocols such as Controller Area Network (CAN), Local Interconnect Network (LIN), FlexRay, Media Oriented Systems Transport (MOST), and Ethernet to communicate between different electronic control units (ECUs). For example, a body control module (BCM) communicates with a rain sensor using LIN. Similarly, an onboard diagnostic scan tool communicates with the ECUs in a car using CAN or Ethernet.

In order for these ECUs to communicate and exchange data, they need a way to bridge across different domains and protocols. That's where gateways come in. As Figure 1 shows, gateways connect different ECUs, translating data from one protocol to another before forwarding it on to the intended recipient.

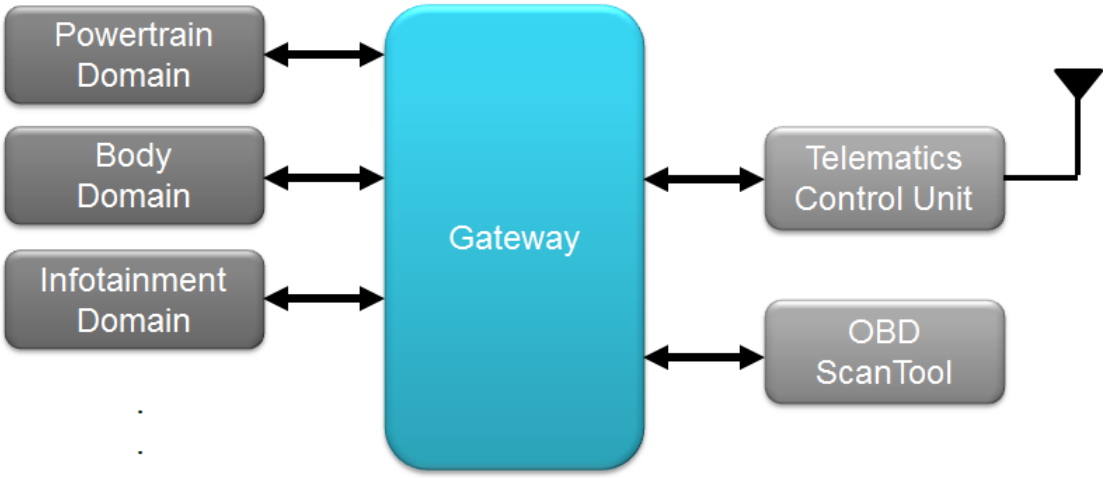

Figure 1. A Typical Vehicle Network Configuration Showing the Gateway

Figure 2 shows a typical block diagram of a gateway module. As you can see, a gateway is a collection of communication interface physical layers (PHYs), along with a processor and corresponding power-management devices.

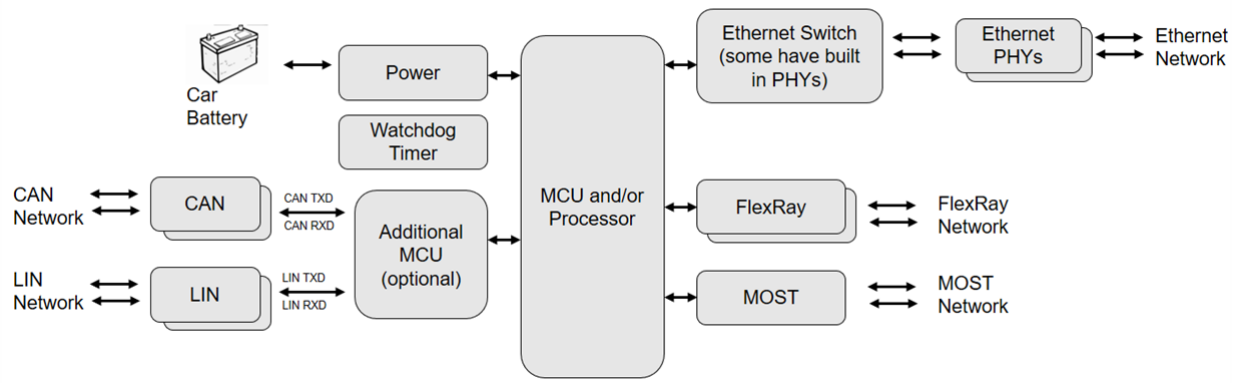

Figure 2. Automotive Gateway Block Diagram

As cars become more connected they require more data, which has led Ethernet to become more prevalent in cars; gateways evolved to meet this need. Whereas older gateways have smaller, simpler microcontrollers (MCUs) as their controllers, newer gateways use processors, sometimes with a supplementary MCU. The difference between a processor and MCU is the memory – processors have external memory, while MCUs store everything on-chip.

Why the shift to processors? Part of it is from a support standpoint. Many processors have Ethernet integrated, as well as the software to support it. Processors with more memory can run familiar operating systems such as Linux, which enable greater portability and cut down on development time. Additionally, some ECUs have gateway functionality integrated. By using a multicore processor from the Jacinto family, for example, you can divide the tasks for the ECU's functionality from the gateway functionality. Divvying up the tasks gives each core a smaller load, allowing their clocks to run at a lower speed while still letting them meet the real-time requirements of a gateway.

Another reason for the shift from MCUs to processors is the amount of processing power needed to translate data between different communication domains. Ethernet is much faster than either CAN or LIN, with the single-twisted pair 100BASE-T1 leading the way into automotive Ethernet. Gateways must be able to keep up with 100Mbps and eventually gigabit speeds. Plus, some gateways may contain eight or more CAN ports and 10 to 12 Ethernet ports. Add to that the time-critical nature of communications and you'll find you need a lot more processing power than an MCU can provide.

As I mentioned, many gateways have several ports for each protocol – more than a processor. While an external Ethernet switch may be able to take care of Ethernet, other protocols such as CAN and LIN will need another solution. A supplementary MCU can provide those ports for the gateway and forward the data to the processor, while still leveraging the computing power of a processor to translate and forward packets from one domain to another.

The automotive stand-alone gateway with Ethernet and CAN reference design provides a building block for Ethernet- and CAN-capable gateways. It features a Jacinto™ DRA710 processor with two Ethernet ports (one 100BASE-TX protocol and one 100BASE-T1), two CAN ports, and an associated power tree.

Additional Resources

- Learn more about the Automotive Stand-alone Gateway with Ethernet and CAN Reference Design
- Learn more about Jacinto infotainment processors
- Learn more about Body Control Modules

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265