

## This Car Runs on Code

By Robert N. Charette



IMAGE: DAIMLER

The avionics system in the F-22 Raptor, the current U.S. Air Force frontline jet fighter, consists of about 1.7 million lines of software code. The F-35 Joint Strike Fighter, scheduled to become operational in 2010, will require about 5.7 million lines of code to operate its onboard systems. And Boeing's new 787 Dreamliner, scheduled to be delivered to customers in 2010, requires about 6.5 million lines of software code to operate its avionics and onboard support systems.

These are impressive amounts of software, yet if you bought a premium-class automobile recently, "it probably contains close to 100 million lines of software code," says Manfred Broy, a professor of informatics at Technical University, Munich, and a leading expert on software in cars. All that software executes on 70 to 100 microprocessor-based electronic control units (ECUs) networked throughout the body of your car.

Alfred Katzenbach, the director of information technology management at Daimler, has reportedly said that the radio and navigation system in the current S-class Mercedes-Benz requires over 20 million lines of code alone and that the car contains nearly as many ECUs as the new Airbus A380 (excluding the plane's in-flight entertainment system). Software in cars is only going to grow in both amount and complexity. Late last year, the business research firm Frost & Sullivan estimated that cars will require 200 million to 300 million lines of software code in the near future.

Even low-end cars now have 30 to 50 ECUs embedded in the body, doors, dash, roof, trunk, seats, and just about anywhere else the car's designers can think to put them. That means that most new cars are executing tens of millions of lines of software code, controlling everything from your brakes to the volume of your radio [see table, "How and Where Is Software Used in Cars?"]

"Automobiles are no longer a battery, a distributor or alternator, and a carburetor; they are hugely modern in their complexity," says Thomas Little, an electrical engineering professor at Boston University in Massachusetts, who is involved in developing intelligent transportation systems. "The goals to save energy, reduce [emissions], and improve safety have driven the specialization and adoption of electronics in particular."

I have experienced that complexity myself recently. Last year I bought a new car and was staggered to discover a 500-page manual explaining its operations, along with a 200-page companion manual for the GPS and radio systems. One of the new features touted was the much larger glove compartment, but the size was probably dictated by that of the required manuals.

My new car also comes with front and side passenger air bags. The car's air bag electronic controller—along with the dozen or so sensors that provide it with data—have to be able to operate for years in temperatures ranging from the dead of a freezing Minnesota winter to the blazing heat of an Arizona summer sun.

Most of the time the air bag system is just monitoring the car's condition, but if the air bags are triggered by, say, a multiple vehicle collision, the software in the ECU controlling their deployment has 15 to 40 milliseconds to determine "which air bags are activated and in which order," says Broy.

In the near future, Broy says, air-bag control systems will use more than just crash impact information. For example, BMW has just announced that many 2009 models will be equipped with its BMW Assist system, which features a "risk of severe injury" calculation based on information gathered from the car's air-bag controller and its other ECUs, which will inform accident response teams not only where the accident took place, but the likelihood of passengers being severely

injured.

The current amount of software in cars is pretty amazing, given that the first production automotive microcomputer ECU was a single-function controller used for electronic spark timing in the 1977 General Motors Oldsmobile Toronado. In 1978, GM offered as an option its Cadillac Trip Computer on the Cadillac Seville. The computer, a modified Motorola 6802 microprocessor chip, displayed speed, fuel, trip, and engine information. However, the chip served another function: It was used by GM to test how well a microprocessor could control multiple functions such as port fuel injection, electronic spark timing, and cruise control.

By 1981, GM was using microprocessor-based engine controls executing about 50 000 lines of code across its entire domestic passenger car production. Other car companies quickly followed suit.

Jonas Bereisa, a GM engineer, wrote in a 1983 article in *IEEE Transactions on Industrial Electronics* that "software development will become the single most important consideration in new product development engineering." How right he was. Broy estimates that more than 80 percent of car innovations come from computer systems and that software has become the major contributor of value (as well as sticker price) in cars. The cost of electronics as a percent of vehicle costs climbed from around 5 percent in the late 1970s to 15 percent by 2005 (excluding final assembly costs). For hybrids, where the amount of software needed for engine control alone is nearly twice as great as that for a standard car, the cost of electronics as a percent of vehicle costs is closer to 45 percent. Within 10 years, some experts predict that the percentages relating to the cost of electronics as a percent of vehicle cost are expected to rise to 50 percent for conventional vehicles and 80 percent for hybrids.

For today's premium cars, "the cost of software and electronics can reach 35 to 40 percent of the cost of a car," states Broy, with software development contributing about 13 to 15 percent of that cost. He says that if it costs US \$10 a line for developed software—a cost he says is low—for a premium car, its software alone represents about a billion dollars' worth of investment.

John Voelcker, *IEEE Spectrum*'s automotive editor, wrote in April 2007 about [the GMC Yukon hybrid automobile and its Two-Mode Hybrid automatic transmission](#). Voelcker told me that "of all the staff hours in the entire program to build the Two-Mode Hybrid transmission...some 70 percent...were devoted to developing the control software."

As Voelcker pointed out in his story, that control software logic analyzes hundreds of inputs every 10 milliseconds, including vehicle load, engine operations, battery parameters, and the temperatures in the high-voltage electric components.

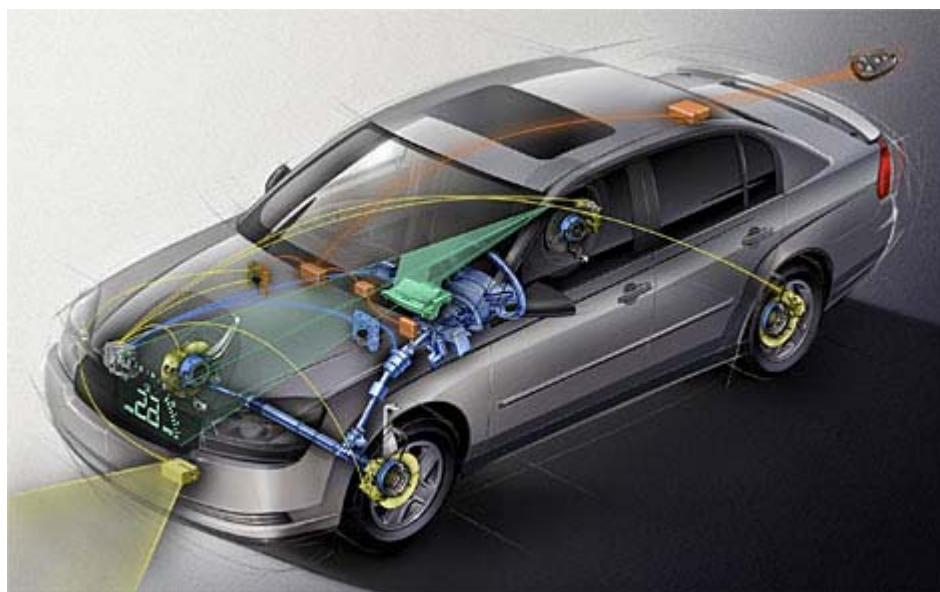


IMAGE: GENERAL MOTORS

Such complexity brings with it reliability issues. IBM claims that approximately 50 percent of car warranty costs are now related to electronics and their embedded software, costing automakers in the United States around \$350 and European automakers €250 per vehicle in 2005.

In 2005, Toyota voluntarily recalled 160 000 of its 2004 and some early 2005 model year Prius hybrids because of a software problem that caused the car to suddenly stall or shut down. The time needed to repair the software was estimated at about 90 minutes per vehicle, or about 240 000 person-hours. Even at cost, that is a lot of money.

Last year alone, there were several automotive recall notices related to software problems. For example, in May 2008, Chrysler recalled 24 535 of its 2006 Jeep Commanders because of a problem in the automatic-transmission software. Then in June, Volkswagen recalled about 4000 of its 2008 Passats and Passat Wagons and about 2500 Tiguanas for a problem in the engine-control-module software that could cause an unexpected increase in engine revolutions per minute when the air-conditioning is turned on. In November, GM recalled 12 662 of its 2009 Cadillac CTS vehicles for a software problem within the passenger-sensing system that could disable the front passenger air bag when it should be enabled or enable it when it should be disabled. It is a tribute to the automotive software developers, though, that there aren't many more recalls, given all the software

in cars.

The increased use of software has not only affected car warranty costs but has also made cars harder to repair—so much so that insurance companies increasingly find it cheaper to declare cars damaged in accidents total losses than it is to fix them.

It is not hard to understand why. “In a premium car you have 2000 to 3000 singular functions that are related to software,” Broy says. These are then combined into the 250 to 300 functions used by the driver and passengers to operate the car’s systems.

And unlike most commercial aircraft, which have strict firewalls between critical avionic systems and the in-flight entertainment systems, there is more commingling of information between the electronic systems used to operate the car and those for entertaining the driver and passengers.

According to a Wharton Business School article entitled “[Car Trouble: Should We Recall the U.S. Auto Industry?](#),” a few years ago, some Mercedes drivers found that their seats moved if they pushed a certain button; the problem was that the button was supposed to operate the navigation system.

Roughly one-third of all the software in cars is devoted just to diagnostics, according to a former automotive engineer I spoke to. But even with all that diagnostic information produced, car mechanics often cannot determine the exact cause of the trouble.

Broy told me that more than 50 percent of the ECUs that mechanics replace in cars are technically error free: They exhibit neither a hardware nor a software problem. Mechanics replace the ECUs simply because they don’t have a better way to fix them, he says.

“The garages and the maintenance people are really at a point where repairing a car is too complex and demanding [for them],” says Broy. Remote diagnostics and repair are likely to render mechanics obsolete for many tasks.

In the not-so-distant future, says Broy, when you have a problem with the computer system in your car, you will go to your garage, where your car will be connected to a network so that off-site OEM specialists can download data, do the analysis, and then upload a software correction.

Voelcker says he wouldn’t be surprised to see onboard systems like BMW’s Assist, Ford’s Sync, and GM’s OnStar soon begin routinely feeding operating data parameters back to centralized systems run by the car manufacturers that will analyze the data for parts drifting out of spec or for software that needs updating and automatically inform the driver that the car needs to be brought in for repair.

Besides monitoring their own internal health, cars are beginning to analyze the world around them. “We’re getting into this era where in addition to sensing what’s going on inside the car, we are using things like radars to detect the presence of external objects, lasers to measure distance for cruise control, and video and ultrasonics to detect objects behind you,” says Little. “The trend will be to extract information external to your vehicle about other vehicles and then exploit this information” to improve safety. For example, cars in front of you will let your car know whether there is ice on the highway or an accident.

Says Little, “We are giving up little pieces of control in exchange for safety. The interesting question is, at what point will you and I be willing to say, ‘Okay. I am not going to drive the car; it is going to drive me.’ ”

## About the Author

Robert N. Charette, an *IEEE Spectrum* contributing editor, is a self-described “risk ecologist” who investigates the impact of the changing concept of risk on technology and societal development. Charette also writes *IEEE Spectrum Online’s* The [Risk Factor](#).