EMBEDDED SYSTEM

DR. SARWAN SINGH

AGENDA

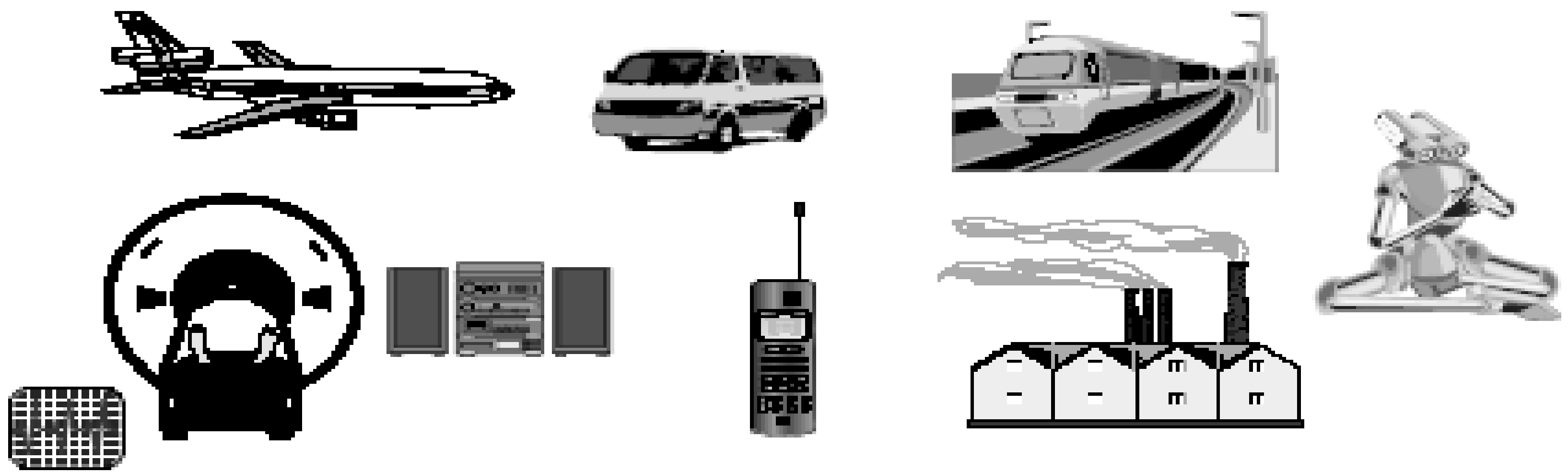
- Embedded System –
  - Introduction, components, classification
- Hardware, software, applications
- Market
- Designing Embedded System

EMBEDDED SYSTEMS: AN INTRODUCTION

- What is an embedded system?
  - More than just a computer
- What makes embedded systems different?
  - Real-time operation
  - *Many* sets of constraints on designs
    - size
    - cost
    - time
    - reliability
    - safety
    - energy
    - security
- What embedded system designers need to know?
  - The "big" picture
  - Skills required to be an "expert" in this area

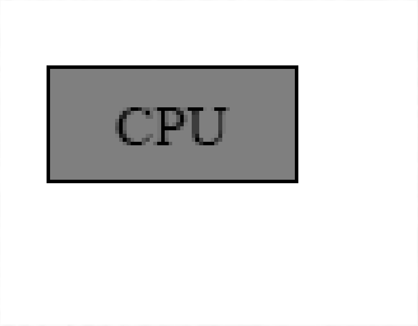
WHAT IS AN EMBEDDED SYSTEM?

- Computer purchased as part of some *other* piece of equipment
  - Typically dedicated software (may be user customizable)
  - Often replaces previously electromechanical components
  - Often no “real” keyboard
  - Often limited display or no general purpose display device
- But, every system is unique there are always exceptions



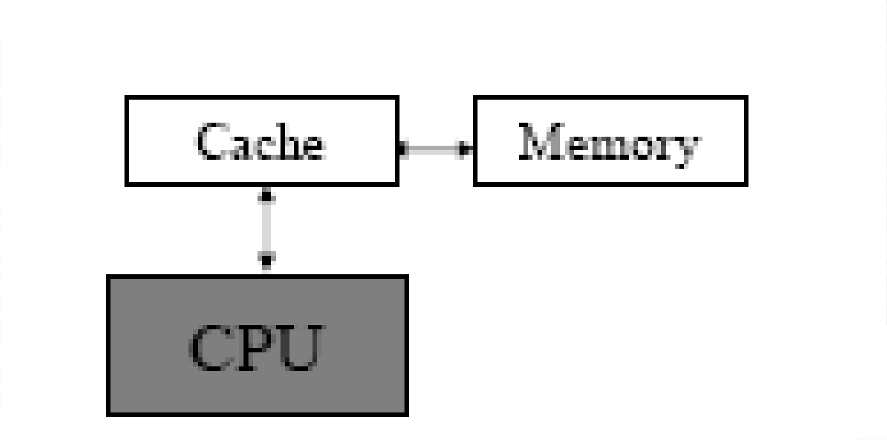
CPU: AN ALL-TOO-COMMON VIEW OF COMPUTING

- Measured by:
  - Performance



AN ADVANCED COMPUTER ENGINEER'S VIEW

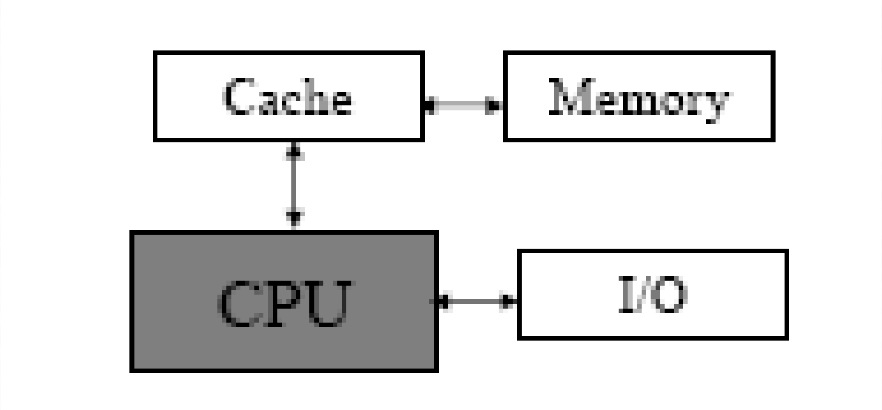
- Measured by: Performance
    - Compilers matter too...



AN ENLIGHTENED COMPUTER ENGINEER'S VIEW

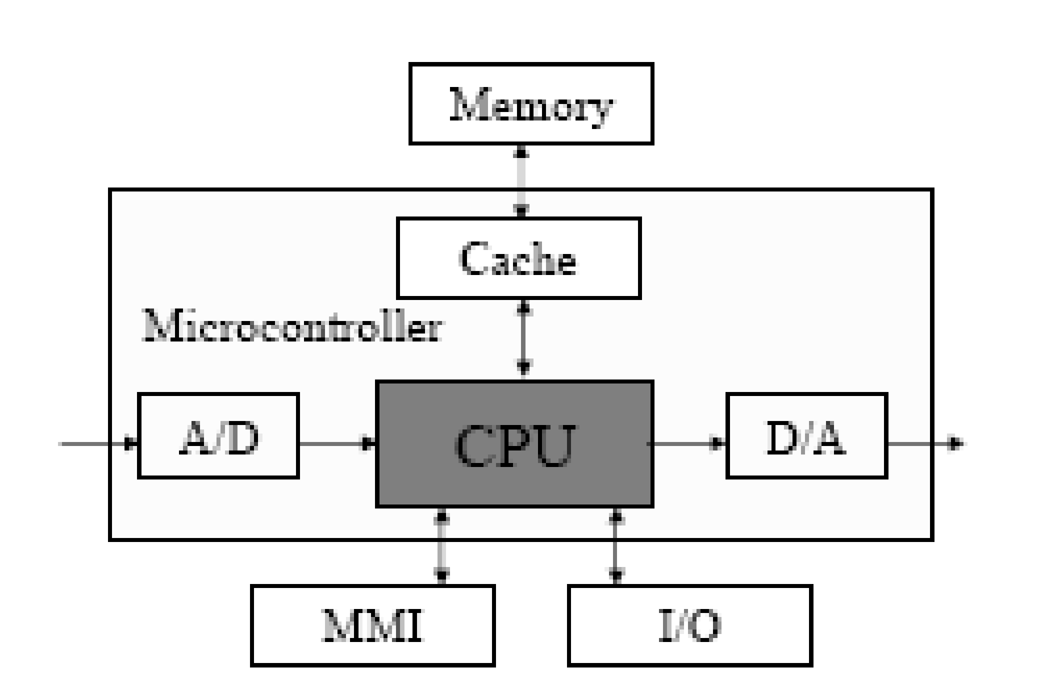
- Measured by: Performance, Cost

  Compilers & OS matters



AN EMBEDDED COMPUTER DESIGNER'S VIEW

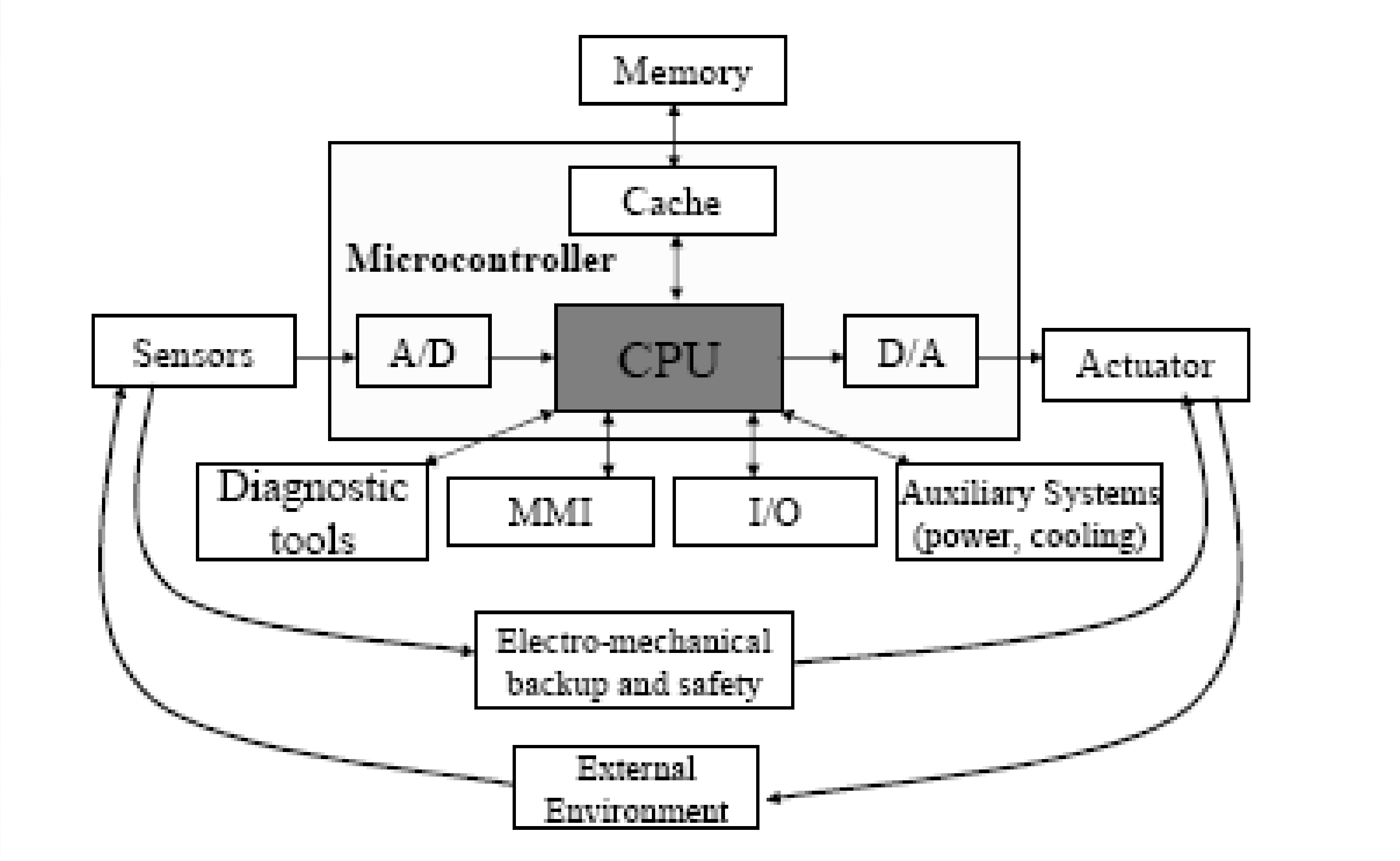
- Measured by: Cost, I/O connections, Memory Size, Performance



AN EMBEDDED CONTROL SYSTEM DESIGNER'S VIEW

- Measured by:

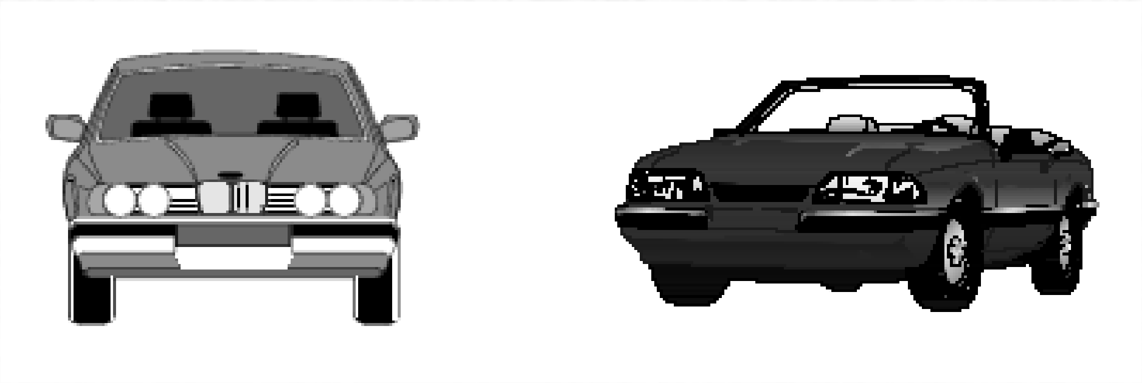
  Cost, Time to market, Cost, Functionality, Cost & Cost.



Source : users.ece.cmu.edu

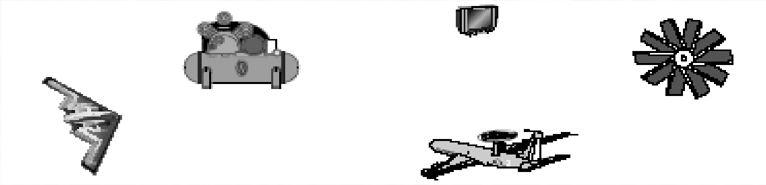
A CUSTOMER VIEW

- Reduced Cost
- Increased Functionality
- Improved Performance
- Increased Overall Dependability



SOME EMBEDDED SYSTEM EXAMPLES

- Pocket remote control RF transmitter
  - 100 KIPS, water/crushproof, fits in pocket, 5year battery life
  - Software handcrafted for small size (less than 1 KB)
- Industrial equipment controller (e.g., elevator; jet engine)
  - 110 MIPS for 1 to 10 CPUs, 1 8MB memory
  - Safety critical software; real time control loops
- Military signal processing (e.g., Radar/Sonar)
  - 1 GFLOPS, 1 GB/sec I/O, 32 MB memory
  - Software handcrafted for extremely high performance



An example of:	Signal Processing	Mission Critical	Distributed	Small
Computing speed	1 GFLOPS	10 - 100 MIPS	1-10 MIPS	100,000 IPS
I/O Transfer Rates	1 Gb/sec	10 Mb/sec	100 Kb/sec	1 Kb/sec
Memory Size	32 - 128 MB	16 - 32 MB	1 - 16 MB	1 KB
Units Sold	10 - 500	100 - 1000	100 - 10,000	1,000,000+
Development Cost	$20M - $100M	$10M - $50M	$1M - $10M	$100K - $1M
Lifetime	15 - 30 years	20 - 30 years	25 - 50 years	10 - 15 years
Environment	Vibration, Heat	Heat, Vibration, Lightning	Dirt, Fire	Over-voltage, Heat, Vibration
Cost Sensitivity	$1000	$100	$10	$0.05
Other Constraints	Size, weight, power	Size, weight	Size	Size, weight, power
Safety	—	Redundancy	Mechanical Safety	—
Maintenance	Frequent repairs	Aggressive fault detection/ maintenance	Scheduled maintenance	"Never" breaks
Digital content	Digital except for signal I/O	~½ Digital	~½ Digital	Single digital chip; rest is analog/power
Certification authorities	Customer	Federal Government	Development team	Customer; Federal Government
Repair time goal	1-12 hours	30 minutes	4 min - 12 hours	1-4 hours
Initial cycle time	3-5 years	4-10 years	2-4 years	0.1-4 years
Product variants	1-5	5-20	10-10,000	3-10
Engineering allocation method	Per-product budget	Per-product budget	Allocation from large pool	Demand-driven daily from small pool
Other possible examples in this category:	Radar/Sonar Video Medical imaging	Jet engines Manned spacecraft Nuclear power	High-rise elevators Trains/trams/subways Air conditioning	Automotive auxilliaries Consumer electronics "Smart" I/O

EMBEDDED COMPUTERS *RULE* THE MARKETPLACE

- ~80 Million PCs vs. ~3 Billion Embedded CPUs annually

– Embedded market growing; PC market *mostly* saturated

WHY ARE EMBEDDED SYSTEMS DIFFERENT?

Four General Categories of Embedded Systems

- General Computing
  - Applications *similar* to desktop computing, but in an embedded package
  - Video games, set top boxes, wearable computers, automatic tellers
- Control Systems
  - Closed loop feedback control of real time system
  - Vehicle engines, chemical processes, nuclear power, flight control
- Signal Processing
  - Computations involving large data streams
  - Radar, Sonar, video compression
- Communication & Networking
  - Switching and information transmission
  - Telephone system, Internet

THE PATRIOT MISSILE FAILURE

- http://www-users.math.umn.edu/~arnold/disasters/patriot.html

Not secure | www-users.math.umn.edu/~arnold/disasters/patriot.html

The Patriot Missile Failure

On February 25, 1991, during the Gulf War, an American Patriot Missile battery in Dharan, Saudi Arabia, failed to track and intercept an incoming Iraqi Scud missile. The Scud struck an American Army barracks, killing 28 soldiers and injuring around 100 other people. A report of the General Accounting office, GAO/IMTEC-92-26, entitled *Patriot Missile Defense: Software Problem Led to System Failure at Dhahran, Saudi Arabia* reported on the cause of the failure. It turns out that the cause was an inaccurate calculation of the time since boot due to computer arithmetic errors. Specifically, the time in tenths of second as measured by the system's internal clock was multiplied by 1/10 to produce the time in seconds. This calculation was performed using a 24 bit fixed point register. In particular, the value 1/10, which has a non-terminating binary expansion, was chopped at 24 bits after the radix point. The small chopping error, when multiplied by the large number giving the time in tenths of a second, led to a significant error. Indeed, the Patriot battery had been up around 100 hours, and an easy calculation shows that the resulting time error due to the magnified chopping error was about 0.34 seconds. (The number 1/10 equals $1/2^4+1/2^5+1/2^8+1/2^9+1/2^{12}+1/2^{13}+\ldots$. In other words, the binary expansion of 1/10 is 0.0001100110011001100110011001100.... Now the 24 bit register in the Patriot stored instead 0.00011001100110011001100 introducing an error of 0.0000000000000000000000011001100... binary, or about 0.000000095 decimal. Multiplying by the number of tenths of a second in 100 hours gives $0.000000095\times100\times60\times60\times10=0.34$.) A Scud travels at about 1,676 meters per second, and so travels more than half a kilometer in this time. This was far enough that the incoming Scud was outside the "range gate" that the Patriot tracked. Ironically, the fact that the bad time calculation had been improved in some parts of the code, but not all, contributed to the problem, since it meant that the inaccuracies did not cancel, as discussed here and here.

The following paragraph is excerpted from the GAO report.

The range gate's prediction of where the Scud will next appear is a function of the Scud's known velocity and the time of the last radar detection. Velocity is a real number that can be expressed as a whole number and a decimal (e.g., 3750.2563...miles per hour). Time is kept continuously by the system's internal clock in tenths of seconds but is expressed as an integer or whole number (e.g., 32, 33, 34...). The longer the system has been running, the larger the number representing time. To predict where the Scud will next appear, both time and velocity must be expressed as real numbers. Because of the way the Patr computer performs its calculations and the fact that its registers are only 24 bits long, the conversion of time from an integer to a real number cannot be any more precise than 24 bits. This conversion results i a loss of precision causing a less accurate time calculation. The effect of this inaccuracy on the range gate's calculation is directly proportional to the target's velocity and the length of the the system has been running. Consequently, performing the conversion after the Patriot has been running continuously for extended periods causes the range gate to shift away from the center of the target, making it less likely th the target, in this case a Scud, will be successfully intercepted.

More disasters attributable to bad numerics

Last modified August 23, 2000 by *Douglas N. Arnold*

APPLICATION AREAS

- TV
- stereo
- remote control
- phone / mobile phone
- refrigerator
- microwave
- washing machine
- electric tooth brush
- oven / rice or bread cooker
- watch
- alarm clock
- electronic musical instruments
- electronic toys (stuffed animals, handheld toys, pinballs, etc.)
- medical home equipment (e.g. blood pressure, thermometer)

Medical Systems

pace maker, patient monitoring systems, injection systems, intensive care units, …

Office Equipment : printer, copier, fax, …

Tools : multimeter, oscilloscope, line tester, GPS, …

Banking : ATMs, statement printers, …

Transportation

(Planes/Trains/[Automobiles] and Boats)

radar, traffic lights, signalling systems, …

Automobiles

engine management, trip computer, cruise control, immobilizer, car alarm, airbag, ABS, ESP, …

Building Systems

elevator, heater, air conditioning, lighting, key card entries, locks, alarm systems, …

Agriculture feeding systems, milking systems,…

Space satellite systems, …

APPLICATION AREAS

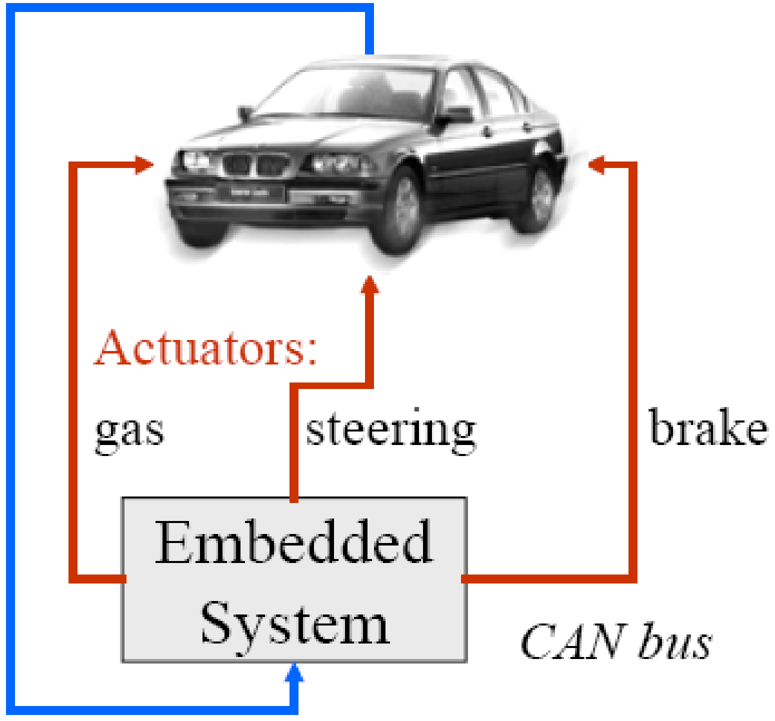
- **Facts:**
  - 1997: The average U.S. household has over 10 embedded computers (source: www.it.dtu.dk/~jan)
- 1998: 90% Embedded Systems vs. 10% Computers
  - (source: Frautschi, www.caliberlearning.com)
- 2001: The Volvo S80 has 18 embedded controllers and 2 busses (source: Volvo)

Automobiles

Autonomous cars:

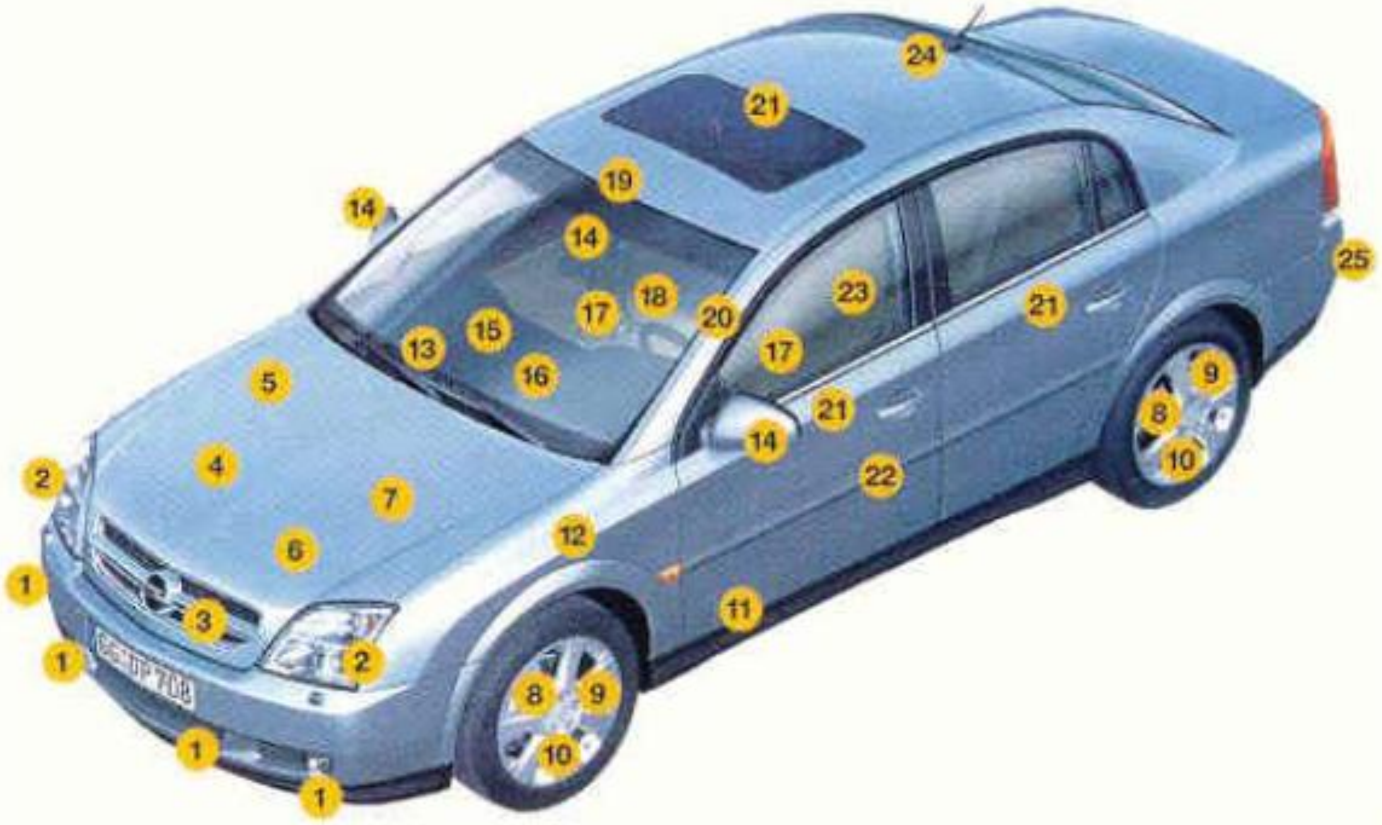
- Electronic gas
- Electronic brake
- Electronic steering

See: The Daimler Story



Sensors: Stereo-cameras, speedometer, accelerometers, signalling

Automobiles



2002: Opel Vectra has over 40 sensors (25 types)