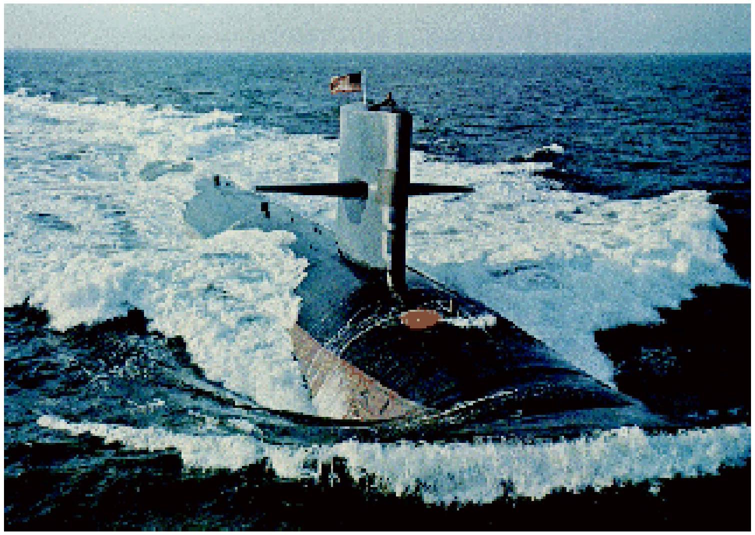
Embedded Systems

18-200

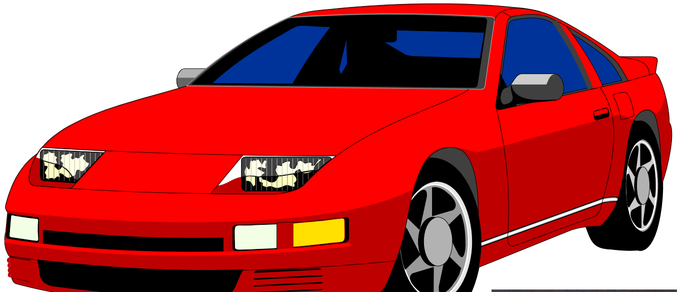
Prof. Philip Koopman

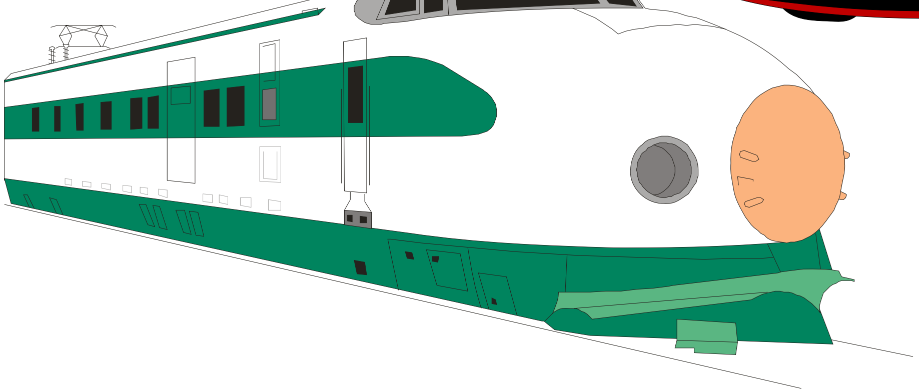
http://www.ece.cmu.edu/~koopman

Carnegie Mellon



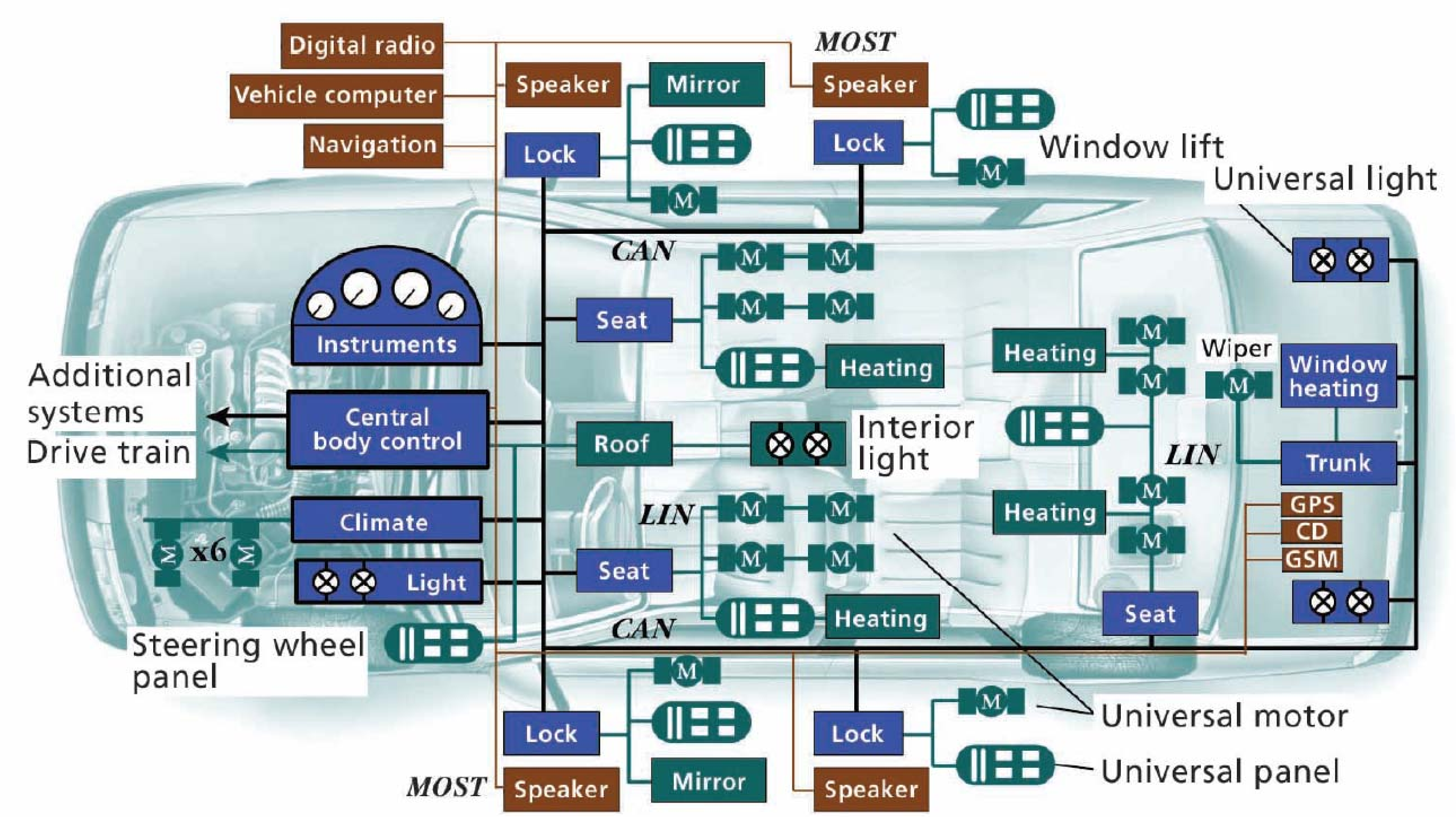
Embedded System =
Computers Inside a Product





Some Embedded Systems Have "Big" Computers

But More Often There Are Many Specialized Ones



CAN	Controller area network
GPS	Global Positioning System
GSM	Global System for Mobile Communications
LIN	Local interconnect network
MOST	Media-oriented systems transport

[Leen02]

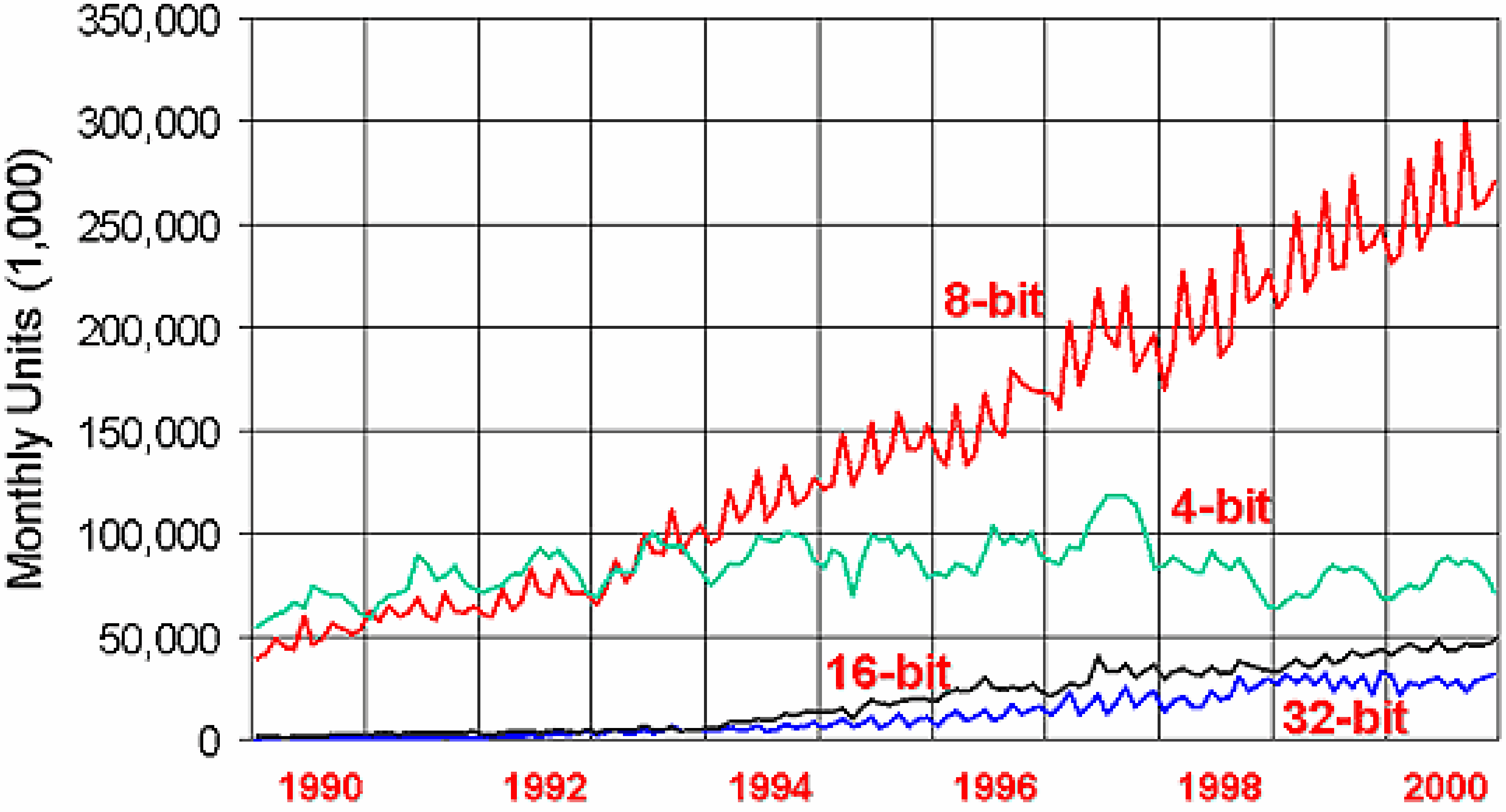
Small Computers Rule The Marketplace

- Everything here has a computer – but where are the Pentiums?

[Smolan]

Microprocessor Unit Sales

All types, all markets worldwide



[EDN] Source: WSTS

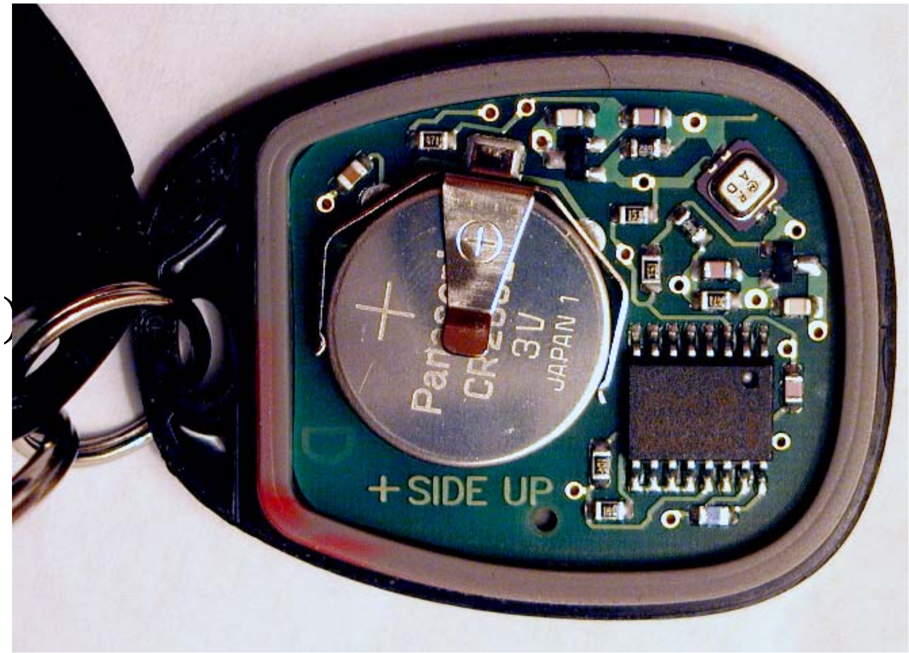
Example Embedded System Requirement

- **Remote Entry system used on General Motors and other vehicles**
  - Designed in 1994, but still in production – if it works don't mess with it!
  - Uses an 8-bit Motorola processor at < 1 MHz clock rate

- **Functions:**
  - Transmits door lock/unlock & trunk open with 110-bit encryption key
  - Performs secure resync without dealer visit if transmitter loses power (flash memory unavailable)

- **Constraints:**
  - Multiple years on one battery
  - Must work after being stepped on in in a rain puddle (don't try this at home!)
  - About 700 Bytes of Program ROM
  - About 512 bits of RAM
  - Extremely low cost

Lear Encrypted Remote Entry Unit



Typical Embedded System Constraints

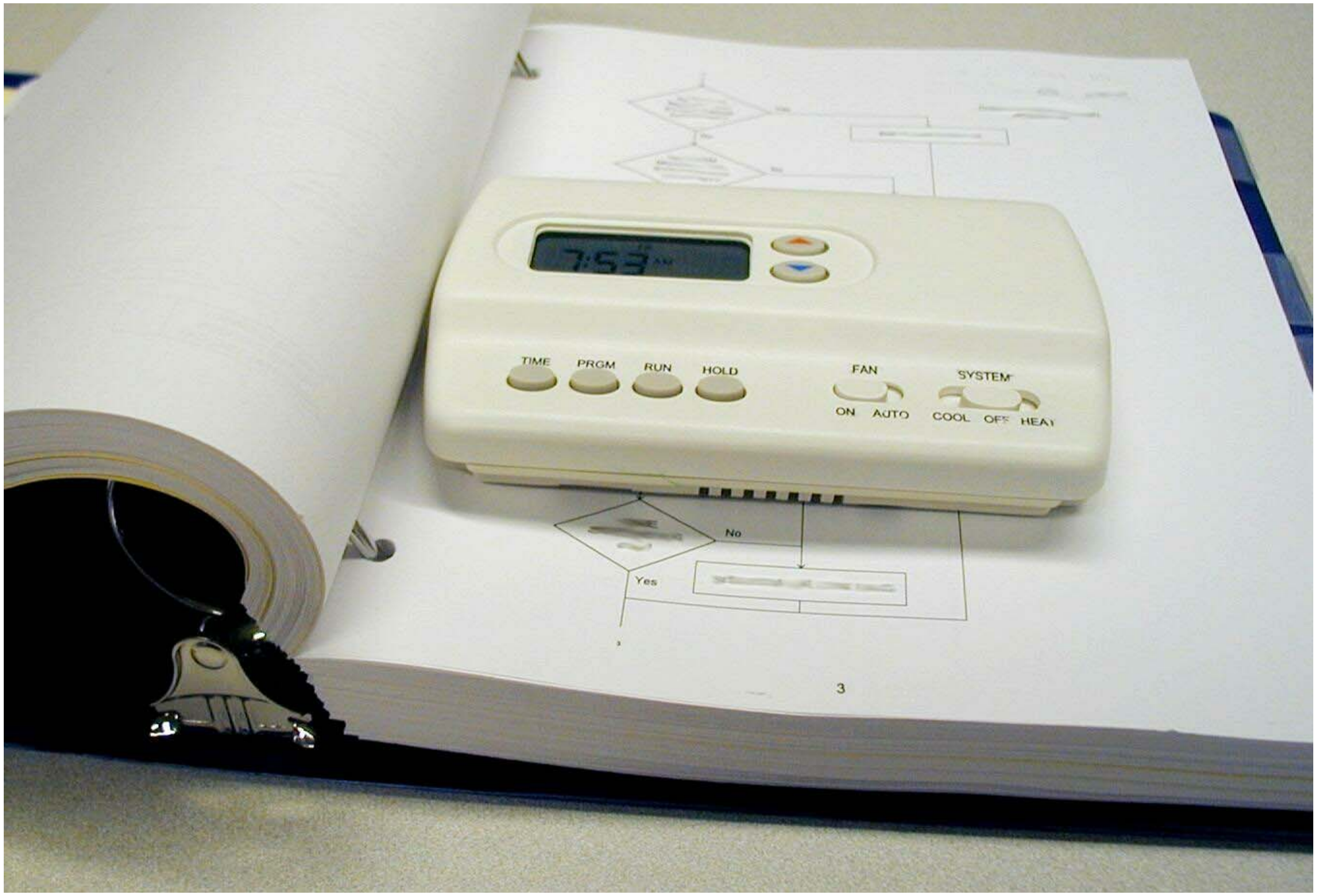
- **Small Size, Low Weight**
  - Hand-held electronics
  - Transportation applications -- weight costs money
- **Low Power**
  - Battery power for 8-100+ hours (laptops often last only 2 hours)
  - Limited cooling may limit power even if AC power available
- **Harsh environment**
  - Power fluctuations, EMI, lightning
  - Heat, vibration, shock
  - Water, corrosion, physical abuse
- **Safety-critical operation**
  - Must function correctly
  - Must *not* function *in*correctly
- **Extreme cost sensitivity**
  - $.05 adds up over 1,000,000 units

Adidas 1 Shoe

http://www.t3.co.uk/news/general/general/adidas_1_the_computer-powered_shoe!

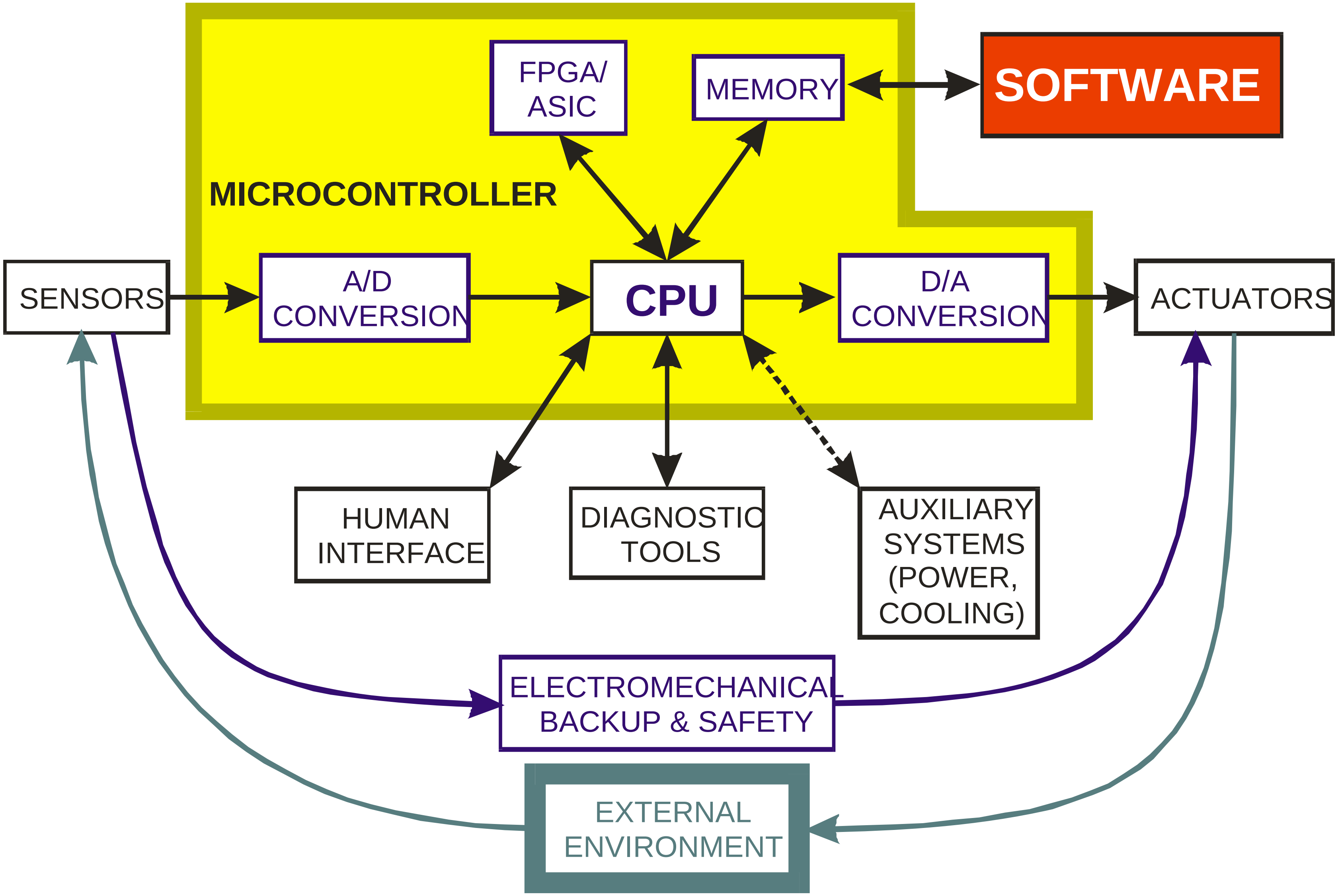
Trend: More Complex Software

- ***Reality:*** **Winning the game requires shoving 20 pounds into an 3 ounce sack**
  - Here's the design package for a household setback thermostat
  - Cars are approaching 1 Million lines of code (exclusive of infotainment)



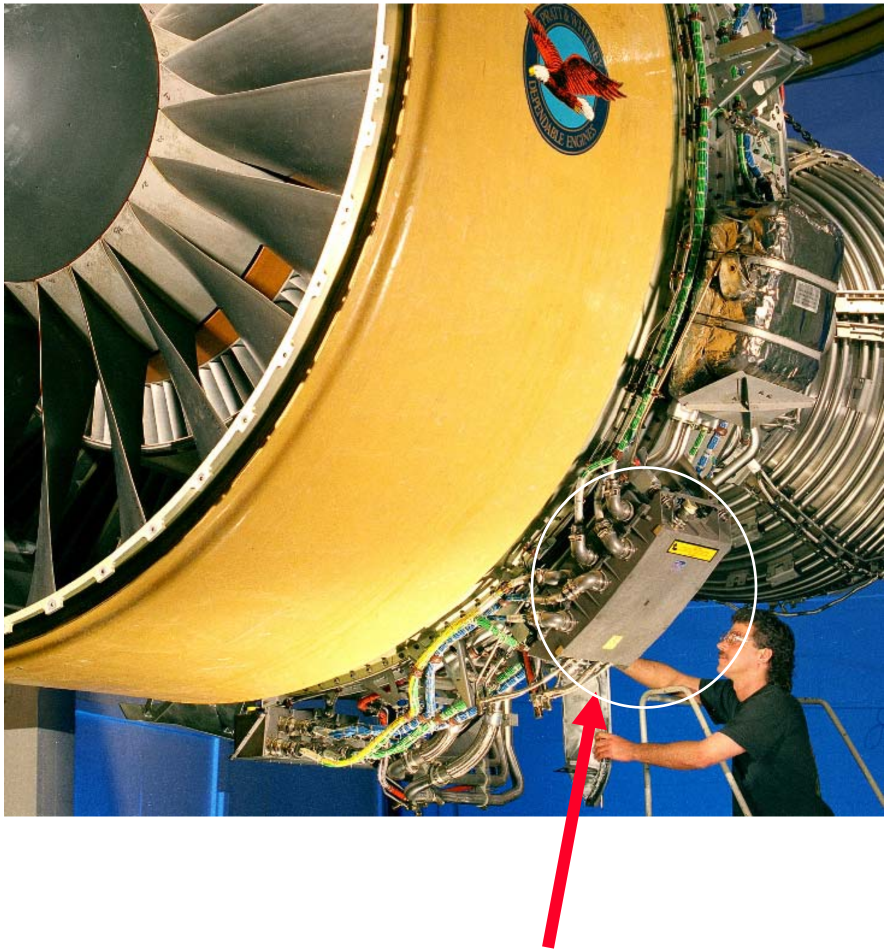
An Embedded Control System Designer's View

- Measured by: Cost, Time-to-market, Cost, Functionality, Cost & Cost.
- In many embedded systems, software is the most difficult part



Common Types of Embedded System Functions

- **Control Laws**
  - PID control, other control approaches
  - Fuzzy logic
- **Sequencing logic**
  - Finite state machines
  - Switching modes between control laws
- **Signal processing**
  - Multimedia data compression
  - Digital filtering
- **Application-specific interfacing**
  - Buttons, bells, lights,…
  - High-speed I/O
- **Fault response**
  - Detection & reconfiguration
  - Diagnosis



PW-4000 FADEC
(Full Authority Digital
Engine Controller)

Various Embedded Computing Areas – 1

Read more about this at:

http://www.ece.cmu.edu/~koopman/pubs/koopman05_embedded_education.pdf

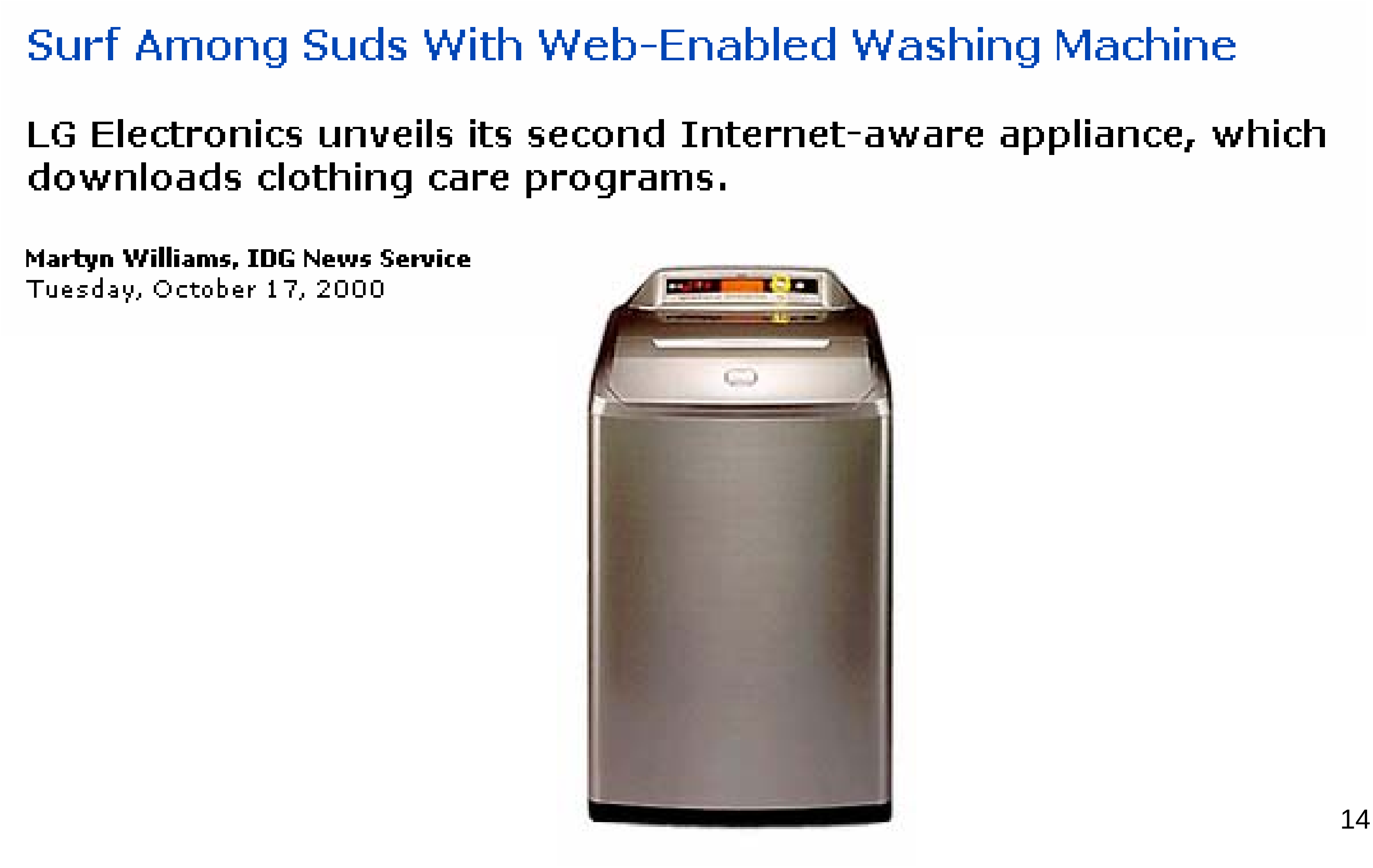
Koopman, P., H. Choset, R. Gandhi, B. Krogh, D. Marculescu, P. Narasimhan, J. Paul, R. Rajkumar, D. Siewiorek, A. Smailagic, P. Steenkiste, D. Thomas, C. Wang, "Undergraduate Embedded System Education at Carnegie Mellon," *ACM Journal Transactions on Embedded Computing Systems*, Vol 4, No. 3, September 2005.

- **Small embedded controllers** (e.g., thermostats)
  - 8-bit CPUs dominate, simple or no operating system
- **Control systems** (e.g., automotive engine control)
  - Often use DSP (Digital Signal Processing) chip for control computations
- **Distributed embedded control** (e.g., cars, elevators, factory automation)
  - Mixture of large and small nodes on a real-time embedded network
- **System on chip** (e.g., consumer electronics, set-top boxes)
  - ASIC design tailored to application area
- **Network equipment** (e.g., network switches; telephone switches)
  - Emphasis on data movement/packet flow
- **Critical systems** (e.g., pacemakers, automatic trains)
  - Safety & mission critical computing

Various Embedded Computing Areas – 2

- **Signal processing** (e.g., face recognition)
  - Often use DSP chips for vision, audio, or other signal processing
- **Robotics** (e.g., autonomous vehicles)
  - Uses various types of embedded computing (especially vision and control)
- **Computer peripherals**
  - Disk drives, keyboards, laser printers, etc.
- **Wireless systems**
  - Wireless network-connected "sensor networks" and "motes" to gather and report information
- **Embedded PCs**
  - Palmtop and small form factor PCs embedded into equipment
- **Command and control**
  - Often huge military systems and "systems of systems" (e.g., a fleet of warships with interconnected computers)

Trend: Internet-connected embedded systems



Surf Among Suds With Web-Enabled Washing Machine

LG Electronics unveils its second Internet-aware appliance, which downloads clothing care programs.

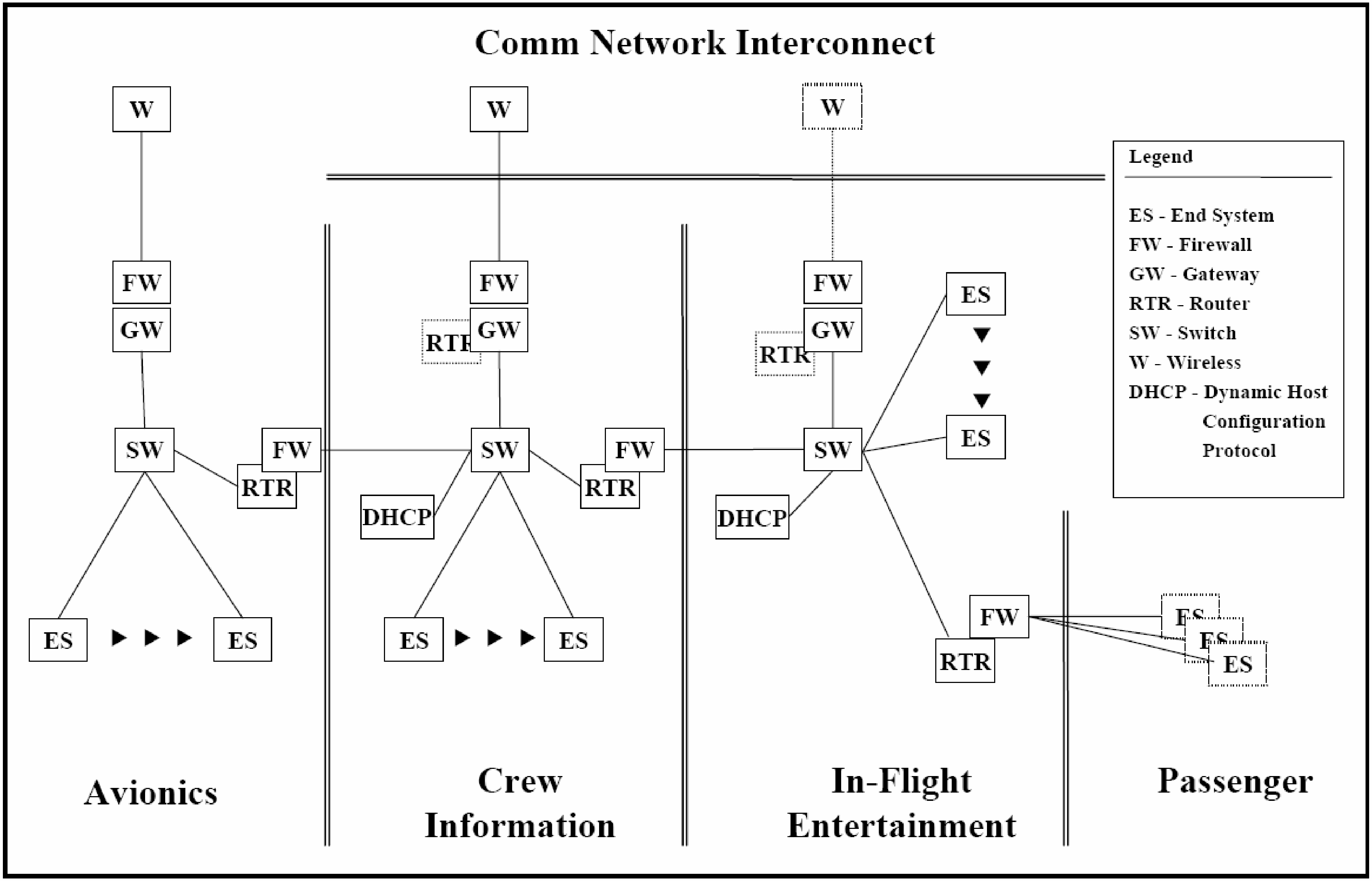
Martyn Williams, IDG News Service
Tuesday, October 17, 2000

Trend: External Connectivity

- **Safety critical subsystems will be connected to external networks (directly or indirectly)**
  - This is going to lead to security issues

Computer graphics by I3M

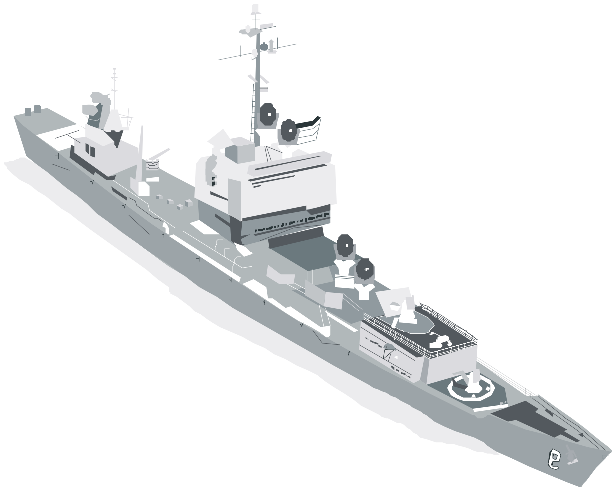
[Airbus 2004] A-380 scheduled to enter service in 2006



Wargo & Chas, 2003, proposed Airbus A-380 architecture

Trend: Desktop Software In Embedded Systems

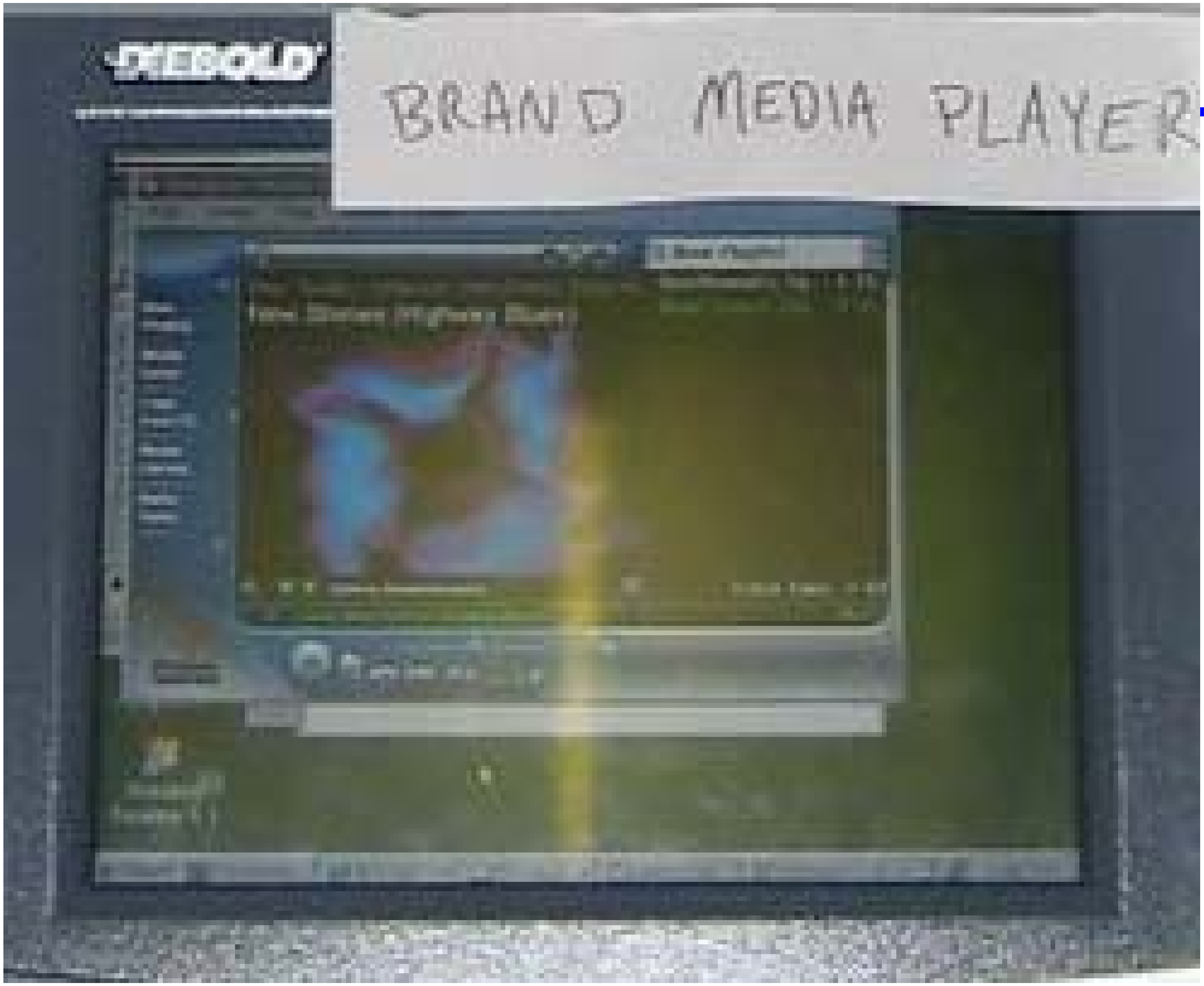
- **Highly dependable software is often required**
  - But desktop systems aren’t designed to provide that!



7/28/98:

“Windows NT Cripples US Navy Cruiser”

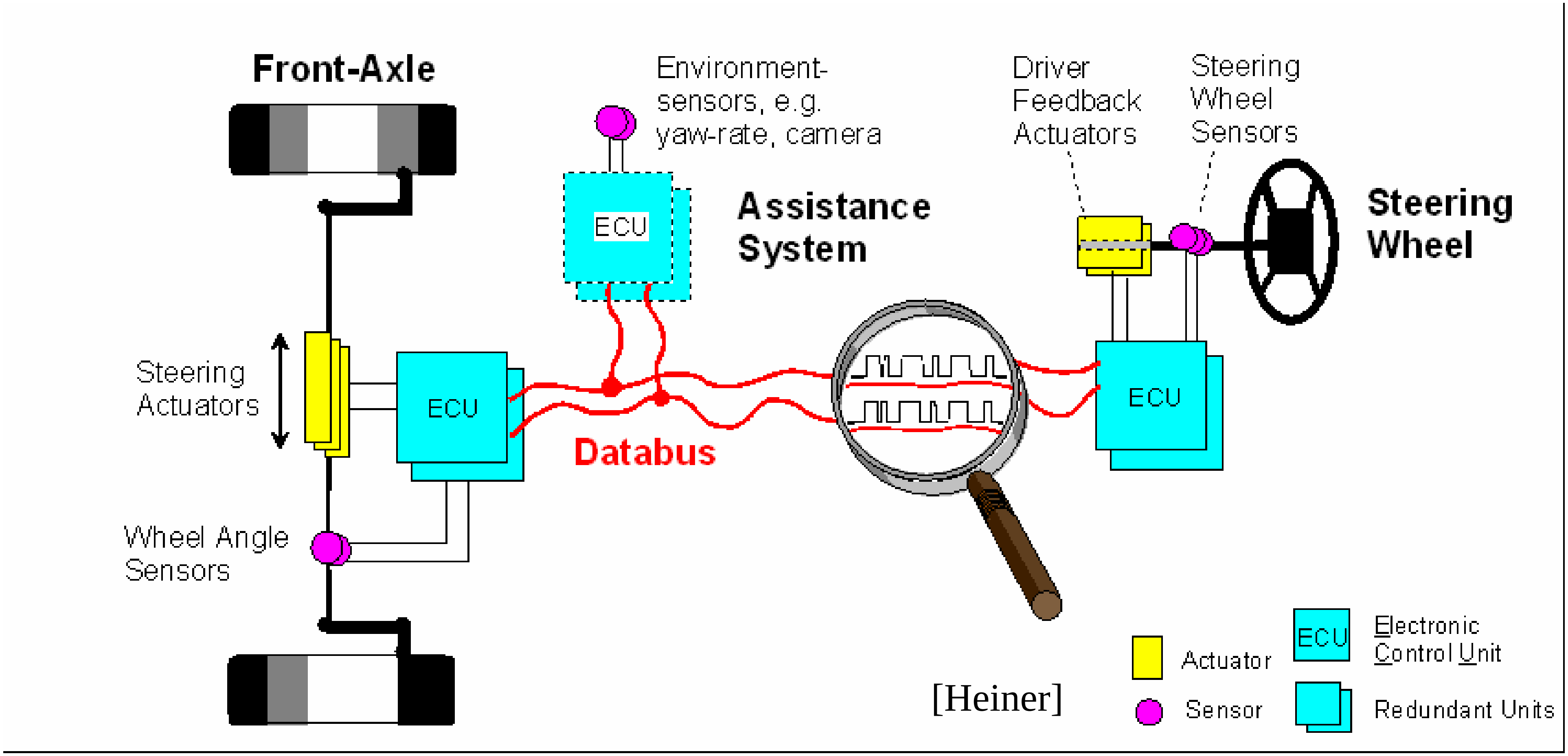
- **Diebold voting machine problems**
  - Electronic voting machines booting to windows instead of votes
  - http://catless.ncl.ac.uk/Risks/23.27.html#subj8.1
- **Automated teller machine crashes**
  - Windows error messages
  - At Carnegie Mellon, someone got an ATM to run media player



http://www.coed.org/photodb/folder.tcl?folder_id=3334
"When ATMs go bad by Carla Geisser“, March 18, 2004
(See also: http://midnightspaghetti.com/newsDiebold.php)

Application Example: X-by-Wire Is Coming Soon

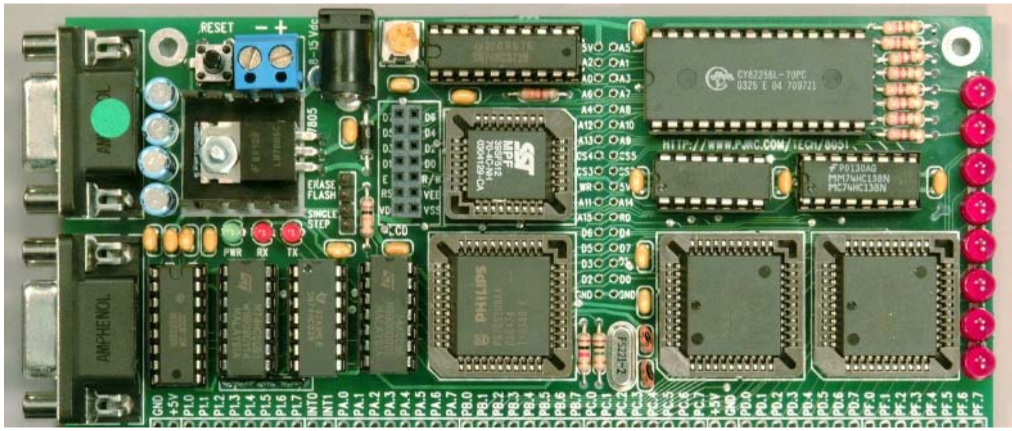
- **X-by-Wire is perhaps the ultimate automotive computer technology**
  - All embedded computers in automobile will probably interface to it
  - Has the most stringent requirements
- **We already have: throttle-by-wire; parking-brake-by-wire**



[Heiner]

Why Take Embedded Computing Courses?

- **Optimizing cost, size & speed**
  - Understanding hardware lets you do more functions with less cost
    - Sometimes you can’t spend more than $1 on a CPU, but it still has to fit everything
  - Getting good performance requires understanding some hardware details
- **Some hardware doesn’t have a lot of support software**
  - Sometimes there is no good I/O support in high level languages
  - Sometimes assembly language is the only way to get good enough code
  - Very often, debugging requires some understanding of hardware
- **Some skills are almost impossible to learn on your own**
  - E.g., ultra-dependable system design for safety critical systems



http://www.pjrc.com/tech/8051/board5/dev5_big.jpg

ECE Embedded Course Sequence:

- **Pre-reqs:**
  - 15-213 Introduction to Computer Systems
  - 18-240 Fundamentals of Computer Engineering

- **18-349 Embedded Real-Time Systems**
  - Single-CPU embedded systems
- **18-549 Distributed Embedded Systems**
  - Multiple CPUs on an embedded network; critical systems; system engineering
- **18-749 Fault-Tolerant Distributed Systems**
  - Enterprise systems with fault tolerant middleware
- **18-849 Dependable Embedded Systems**
  - Deep coverage of dependability & safety critical system research papers
- **Many other relevant specialty and related courses**
  - Controls
  - Robotics
  - Software engineering
  - …

18-349 Introduction To Embedded Systems

- **Junior-level course with significant project content**

- **Course areas:**
  - Low level system/software
    - Combining C & Assembly language
    - Software profiling and optimization
    - Memory management
  - Hardware interfacing
    - I/O
    - Buffering and DMA
    - Serial communications
    - Timers & Interrupts
  - Real time operating systems
    - Resource management
    - Rate monotonic scheduling
    - Loaders, object files
  - Interacting with the outside world
    - Basics of feedback control and signal processing
    - A/D and D/A conversion

18-549 Distributed Embedded Systems

- **Capstone design course**
  - Semester-long project with representative embedded system design cycle: Requirements / design / networking / implementation / test / fault recovery
  - Emphasis on the software side of things; survival skills for 1st year in industry

- **Course areas:**
  - System Engineering
    - Requirements, design, verification/validation, certification, management-lite
  - System Architecture
    - Modeling/Abstraction, Design Methodology, a little UML, Business Issues
  - Embedded Systems
    - Design Issues, scheduling, time, distributed implementations, performance
  - Embedded Networks
    - Protocol mechanisms, real-time performance, CAN, FlexRay, embedded Internet
  - Critical Systems
    - Analysis Techniques, software safety, certification, ethics, testing, graceful degradation
  - Case Studies
    - Elevator as capstone design project
    - Guest speakers and other discussions as available