Make:
Sensors
Projects and Experiments to Measure the World with Arduino and Raspberry Pi
Tero Karvinen, Kimmo Karvinen & Ville Valtokari

2. Characteristics of Embedded Systems

Review

- What will you learn from this course
  - What is an embedded system?
  - What are the design issues?
  - How to design embedded software?
  - Embedded Software Examples
- What is an Embedded system?

What is an Embedded system?

- An embedded system is a large or small computer system that is built into a product, a piece of equipment or another computer system, and that performs some task useful to the product, equipment or system.

（嵌入式系统是以应用为中心，以计算机技术为基础，并且软硬件可裁剪，适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。）

简而言之，它是完成特定任务的计算机系统。

History and Future

- 传统分类法：

  巨型机、大型机、中型机、小型机、微型机

- 通用机、嵌入式

History and Future

When did embedded systems first appear?

Not before 1971! Why?

- Intel designed the world‘s first microprocessor, the 4004, in 1971!
- 4004 was designed for use in a line of business calculators produced by a Japanese company Busicom.
  - In 1969, Busicom asked Intel to design a set of IC ---one for each of their new calculator models
  - The 4004 was Intel’s response.

History and Future

- 在电子计算机发展的初期，计算机一直是“供养”在特殊的机房中的大型、昂贵的专用设备，主要是实现一些特殊的数值计算。
- 直到20世纪70年代微处理器的出现，计算机应用才出现了历史性的变化。这也使计算机摘掉神圣的光环走下了神坛，步入平民化的时代。
- 同时微处理器表现出的智能化水平引起了设备制造、机电控制等专业人士的兴趣，要求将微型机嵌入到一个控制对象的体系中，实现对象体系的智能化控制。
- 微处理器的问世极大的促进了控制领域的发展，复杂的控制系统最初只是由简单的设备组成，以微处理器这样的部件作为主要的控制和反馈器件，极大的提高了系统的可控性和智能化

History and Future

- 经过发展，到80年代初微处理器及微控制器各自已发展为一个庞大的家族，以Intel公司x86为主流的应用于个人计算机PC的微处理器格局已形成。
- 为了区别于原有使用在PC的通用计算机，把嵌入到对象体系中、实现对象体系智能化控制的微控制器的计算机，称作嵌入式计算机.
- 因此，嵌入式计算机是诞生于微处理器发展时代;
- 这也标志着计算机进入了通用计算机与嵌入式计算机两大分支、并行发展时代，从而导致20世纪末，计算机应用的高速发展并由此引发了计算机分类方式的变化。

History and Future

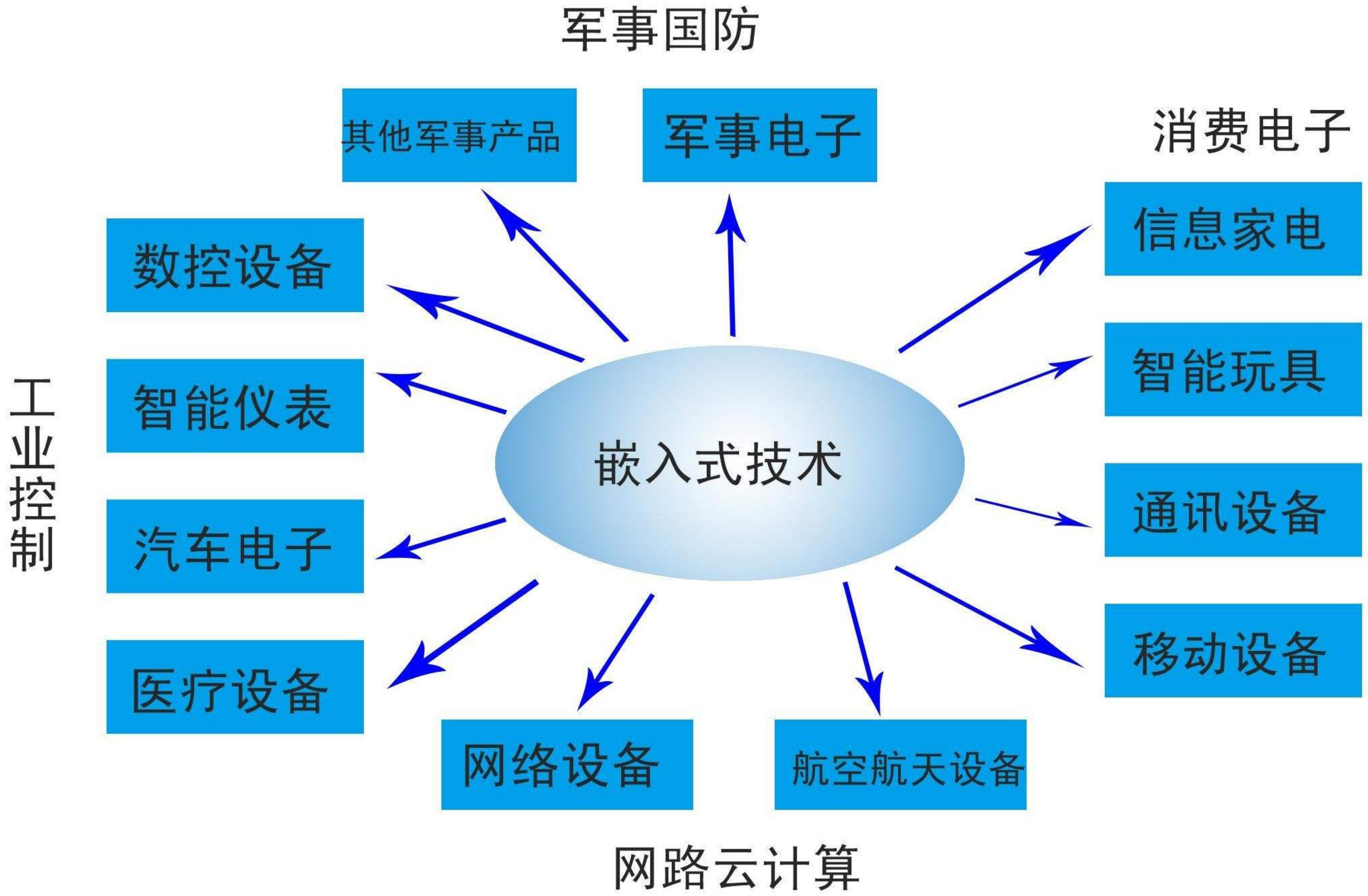
- The microprocessor was an overnight success!
- Increased use in the next decade:
  - unmanned space probes
  - computerized traffic lights
  - aircraft flight control systems
- In 1980’s, embedded systems quietly rode the waves of microcomputer age.

History and Future

Embedded systems will continue to increase

- 中国正处于由世界制造大国向制造强国转型升级的过程中，要求我们制造的产品具有更高的性能、丰富的功能和更高的附加值。这就促进了嵌入式技术在国民经济生活中各个层面的广泛应用。
- 当前嵌入式系统的应用已经涉及生产、工作、生活各个方面。

History and Future



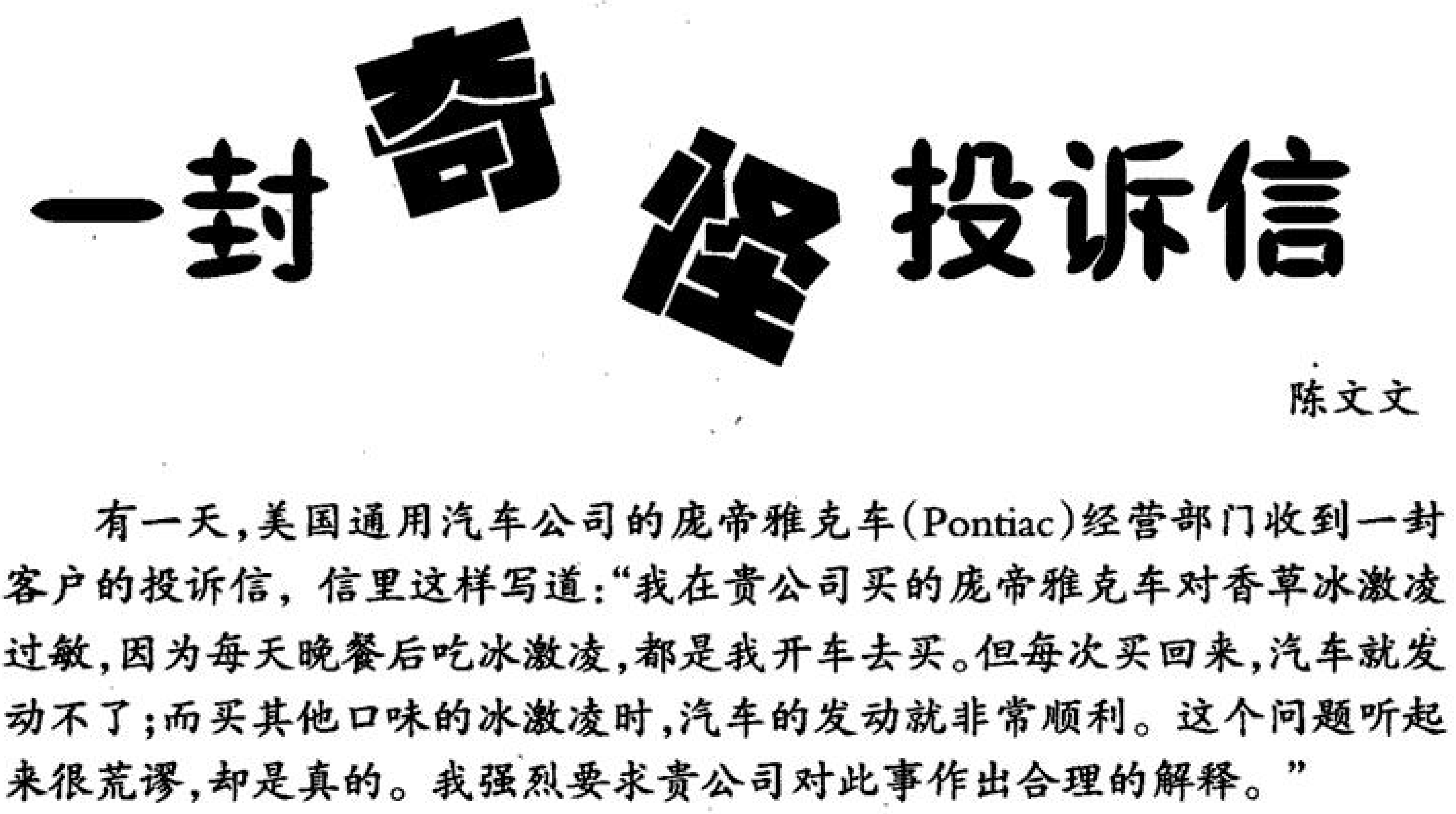
Characteristics of Embedded Systems

- Power limitations and constraints（功耗低、体积小、具有专用性）
  - Consider a device to track wild animal movements in nature
  - Low power = less heat = fewer failures
  - Low power = less energy = longer time of operation
  - Low power = smaller batteries = physically lighter and smaller
  - Usually requires both hardware and software elements to achieve

Characteristics of Embedded Systems

- Must operate in extreme environmental conditions
  - Embedded systems are everywhere and this means everywhere
  - Your PC would fail in a second if left outside on top of Everest
  - Once again, both hardware and software should be aware of this

嵌入式系统运行环境差异很大

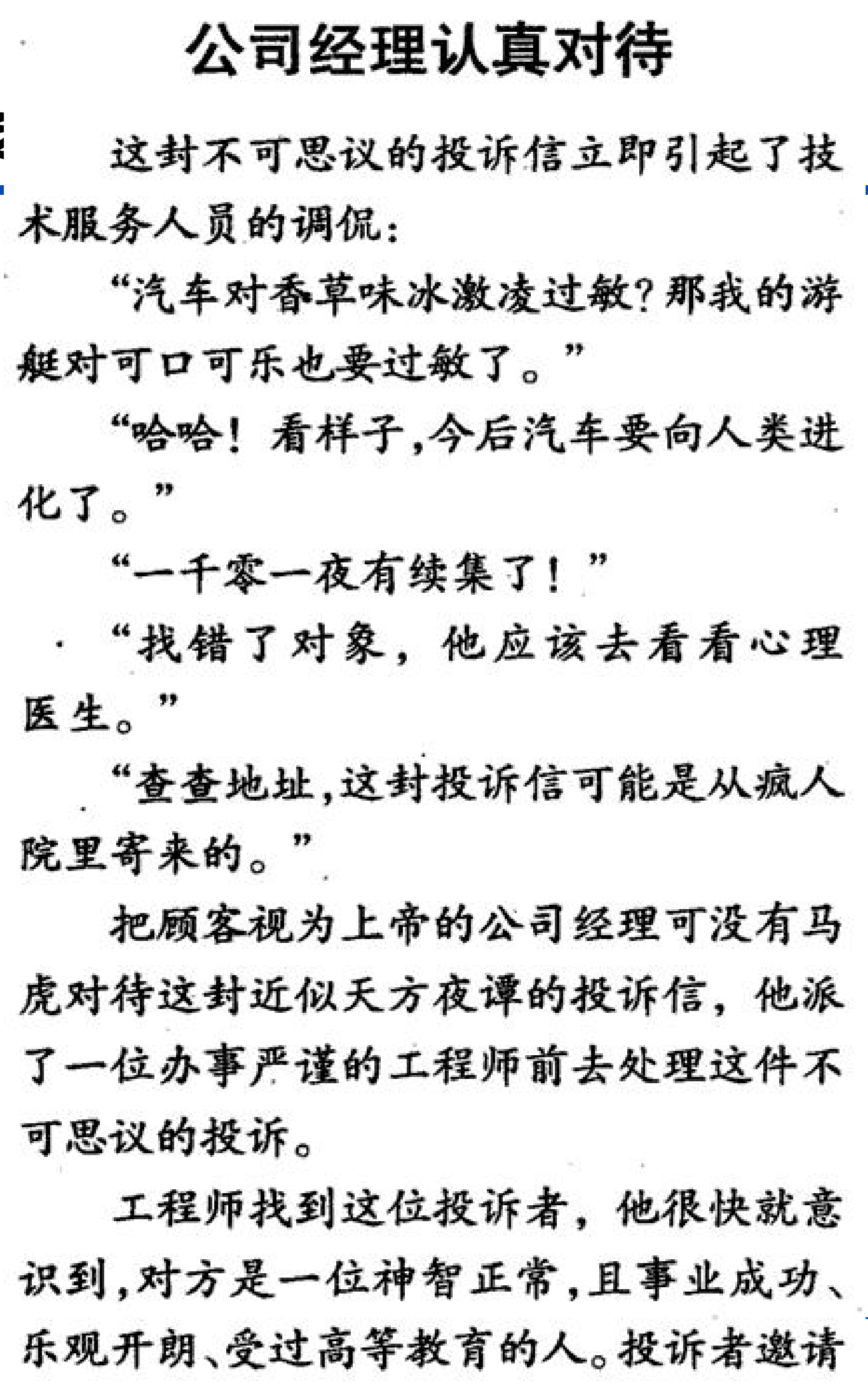


一封奇怪投诉信

陈文文

有一天，美国通用汽车公司的庞帝雅克车(Pontiac)经营部门收到一封客户的投诉信，信里这样写道：“我在贵公司买的庞帝雅克车对香草冰激凌过敏，因为每天晚餐后吃冰激凌，都是我开车去买。但每次买回来，汽车就发动不了；而买其他口味的冰激凌时，汽车的发动就非常顺利。这个问题听起来很荒谬，却是真的。我强烈要求贵公司对此事作出合理的解释。”

嵌入式系统



公司经理认真对待

这封不可思议的投诉信立即引起了技术服务人员的调侃：

“汽车对香草味冰激凌过敏？那我的游艇对可口可乐也要过敏了。”

“哈哈！看样子，今后汽车要向人类进化了。”

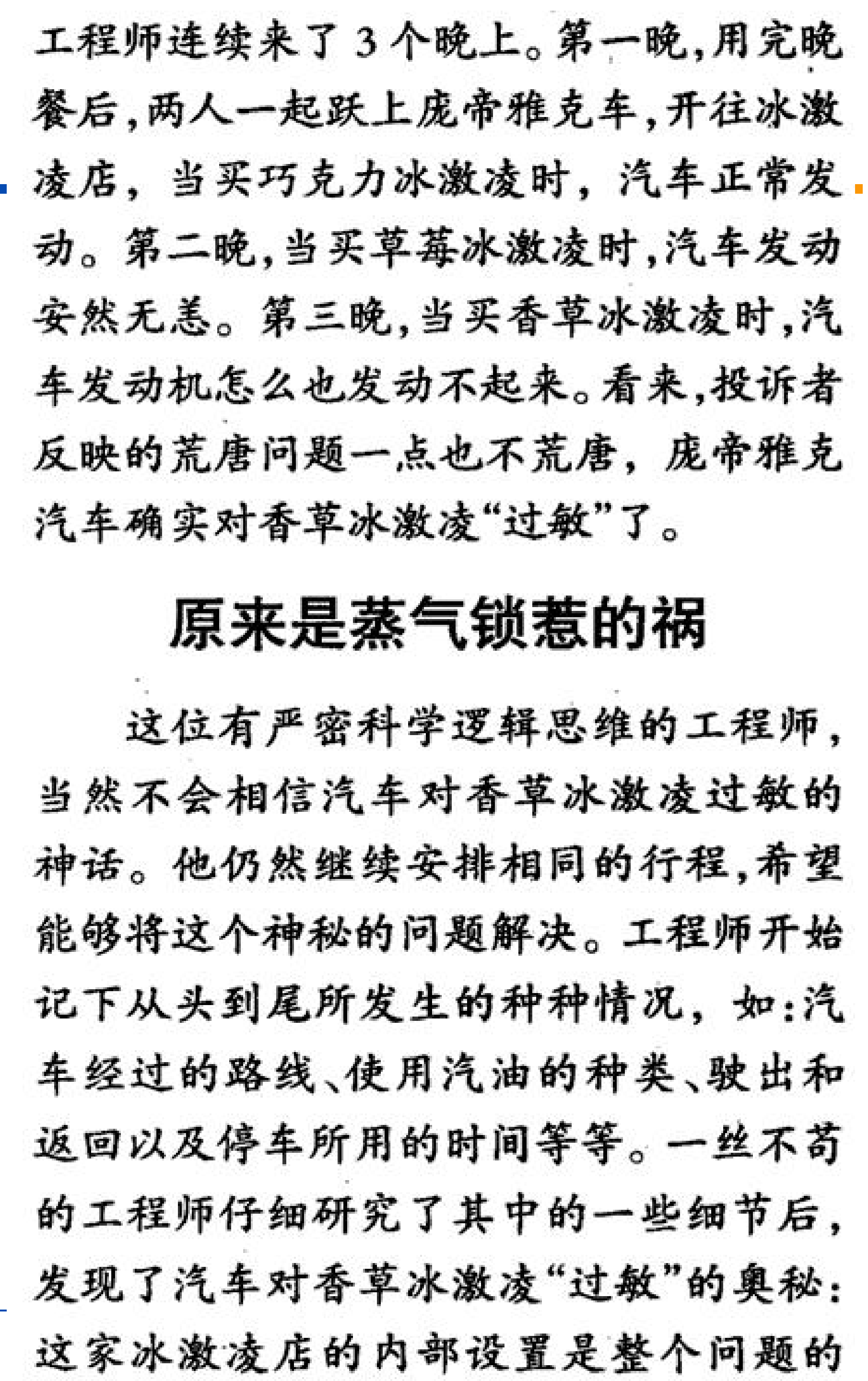
“一千零一夜有续集了！”

“找错了对象，他应该去看看心理医生。”

“查查地址，这封投诉信可能是从疯人院里寄来的。”

把顾客视为上帝的公司经理可没有马虎对待这封近似天方夜谭的投诉信，他派了一位办事严谨的工程师前去处理这件不可思议的投诉。

工程师找到这位投诉者，他很快就意识到，对方是一位神智正常，且事业成功、乐观开朗、受过高等教育的人。投诉者邀请



工程师连续来了3个晚上。第一晚，用完晚餐后，两人一起跃上庞帝雅克车，开往冰激凌店，当买巧克力冰激凌时，汽车正常发动。第二晚，当买草莓冰激凌时，汽车发动安然无恙。第三晚，当买香草冰激凌时，汽车发动机怎么也发动不起来。看来，投诉者反映的荒唐问题一点也不荒唐，庞帝雅克汽车确实对香草冰激凌“过敏”了。

原来是蒸气锁惹的祸

这位有严密科学逻辑思维的工程师，当然不会相信汽车对香草冰激凌过敏的神话。他仍然继续安排相同的行程，希望能够将这个神秘的问题解决。工程师开始记下从头到尾所发生的种种情况，如：汽车经过的路线、使用汽油的种类、驶出和返回以及停车所用的时间等等。一丝不苟的工程师仔细研究了其中的一些细节后，发现了汽车对香草冰激凌“过敏”的奥秘：这家冰激凌店的内部设置是整个问题的

嵌入式系统运行环境差异很大

- 投诉者买香草冰淇淋花的时间比买其他品种的冰淇淋花的时间要少得多。因为香草冰淇淋，店主将他们单独放在柜台边最容易取的冰柜里，品种较多，取时花的时间就要多一些。
- 接着，摆在技术员眼前的另一个问题是,为什么时间短汽车就发动不了呢？当问题从"作怪的香草冰淇淋"变成了"长短时间"，技术员很快就找到了答案。这叫做"气阻现象"，是高温导致管路内气泡积聚，使燃油或刹车油无法通过而形成阻断现象。但是，当投诉者买其他品种的冰淇淋时，因为花的时间较长，引擎冷却，气阻现象小时，故又能正常发动了。
- 一件看似荒诞的事情，经过认真分析，其实就是一个简单的现象。但是,如果因其荒诞而置之不理,简单的现象微 会变得光怪陆离,永远没有大白于天下的那一天。

Characteristics of Embedded Systems

- Far fewer resources than a general purpose computer
  - lower speed and memory
  - limited inputs and I/O etc.

Software issues

- Why C for embedded software?
  - a “very-level” high-level language
  - compact, efficient code for almost all processors
  - direct hardware control, without losing the benefits of a high-level language

Software issues

- Why C for embedded software?
  - appropriate for both 8-bit and 64-bit processors
  - for systems with bytes, KB, MB of memory
  - for design teams of 1, 12, or more people

Software issues

- Other embedded languages
  - assembly language
  - C++

Software issues

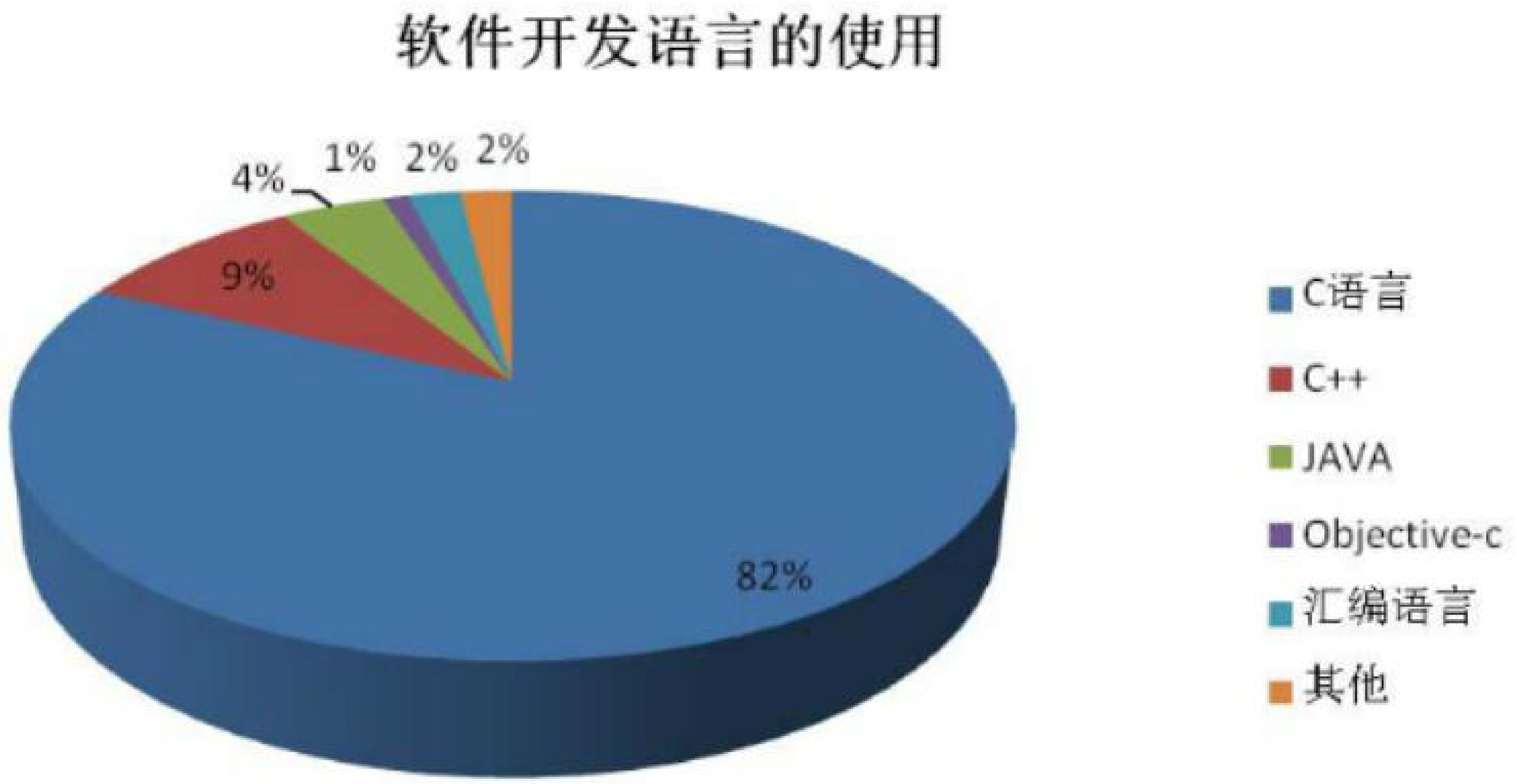
- Assembly Language
  - complete control of CPU and hardware
  - high software development costs
    - lack of code portability
    - lack of skilled assembly programmers

Software issues

- Assembly Language
  - used as an adjunct（附属的）to high-level languages, for small pieces of code that must be
    - extremely efficient or
    - ultra-compact, or
    - cannot be written in any other way.

Software issues

- C++
  - better data abstraction
  - reduce efficiency of executable programs
  - more popular with large development teams, where the benefits to developers outweighs the loss of program efficiency


软件开发语言的使用
82%
9%
4%
1%
2%
2%
C语言
C++
JAVA
Objective-c
汇编语言
其他

Software issues

- Specialized development tools
  - You will surely not be able to use printf() or popup alerts
  - Development takes places on a *host*, separate than the *target*
  - Final product usually has ROM and a very small chunk of RAM

Software issues

- Need for a Real-Time Operating System(RTOS)

- Example: An Automobile airbag system.

  When the airbag's motion sensors detect a collision, the system needs to respond by deploying the airbag within 10ms or less.

  – or the system fails!

Software issues

- Need for a Real-Time Operating System(RTOS)
  - Regular Windows or Linux not suitable for time critical tasks
  - Programming for guaranteed timeliness is hard. Deadlocks, interrupt processing, polling for external events etc. must be properly resolved and synchronized

What is Real Time?

A real time system is one in which the correctness of the computations not only depends upon the logical correctness of the computation but also upon the time at which the result is produced. If the timing constr-aints of the system are not met, system failure is said to have occurred.

—— Donald Gillies

唐纳德 吉利斯

What is Real Time?

- Real time in operating systems:

The ability of the operating system to provide a required level of service in a bounded response time.

—— POSIX Standard 1003.1

Real-Time Systems

- Consequences of a missed deadline:

severe	→	hard real-time
acceptable	→	soft real-time

Hard vs. Soft Real Time

- Hard real-time
  - Systems where it is absolutely imperative that responses occur within the required deadline.
  - E.g. Flight control systems.
- Soft real-time
  - Systems where deadlines are important but which will still function correctly if deadlines are occasionally missed.
  - E.g. Data acquisition system.(数据采集)

Real-Time System

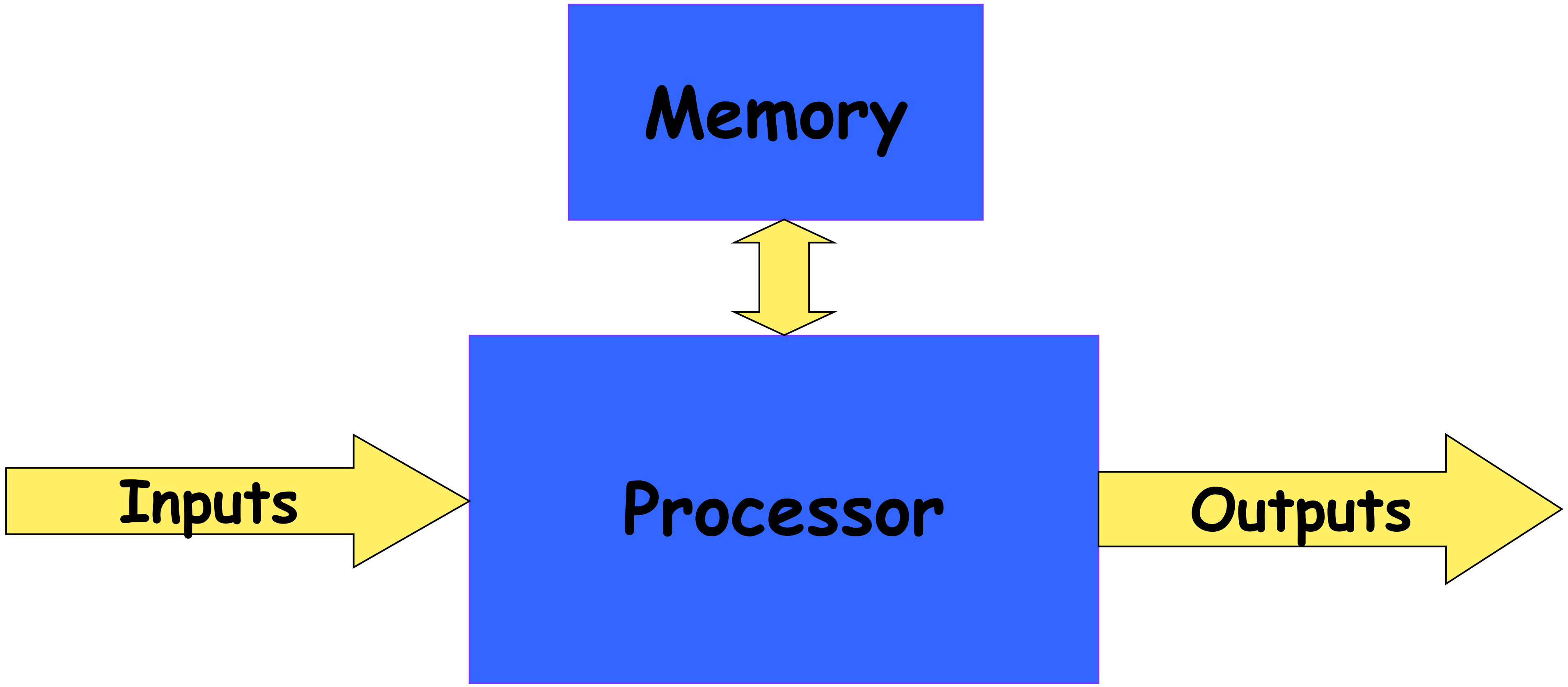
- Applications (Hard)
  - Cars (Volvo S80 has 19 computers)
  - Aeroplanes (JAS) （飞机）
  - Medical equipment （医疗设备）
  - Space vehicles (Mars lander)空间设备 (火星探测器)
  - Military systems（军方系统）
  - Industrial automation（工业自动化）

Real-Time System

- Applications (Soft)
  - Games
  - DVD (Mpeg encoding)
  - Internet video and broadcasting
  - Telecom

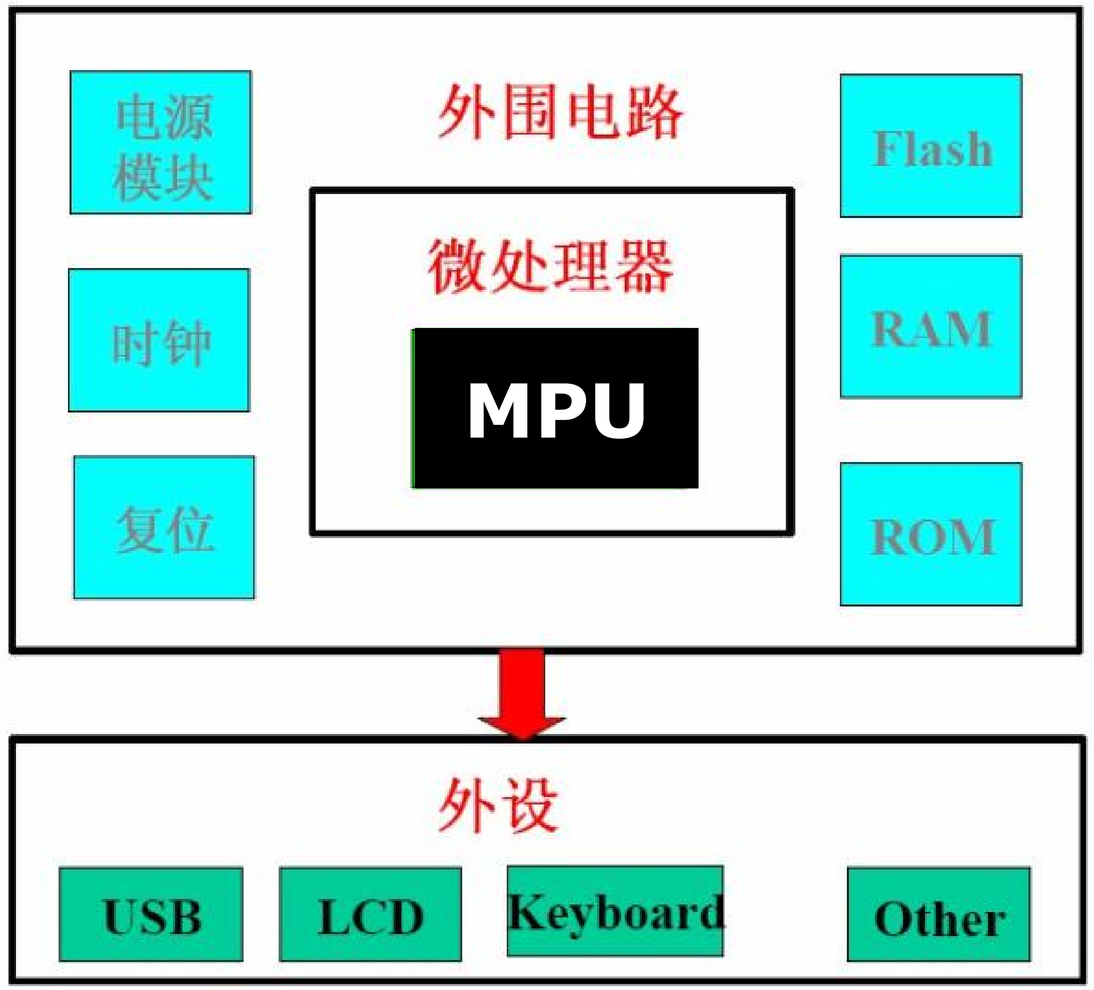
Hardware issues

- Typical Hardware



Hardware issues

- Typical Hardware
  - 嵌入式处理器
  - 外围电路
  - 接口
  - 外设



Typical Hardware

- Microprocessor: execute code
- Memory: different memories for program & data
- Embedded systems do not have the following:
  - a keyboard
  - a screen
  - a disk drive
  - CD, speakers, microphones, diskettes, modems
- Embedded systems have: serial port, network interface, sensors, actuators, etc.

Typical Hardware

- Besides CPU and software, what else is common among embedded systems?
  - Memory storage: ROM, RAM, …
  - Input: knobs, buttons, probes, sensors, communication signals, …
  - Output: human-readable display, microwave radiation, communication signals, changes to physical world

Hardware issues

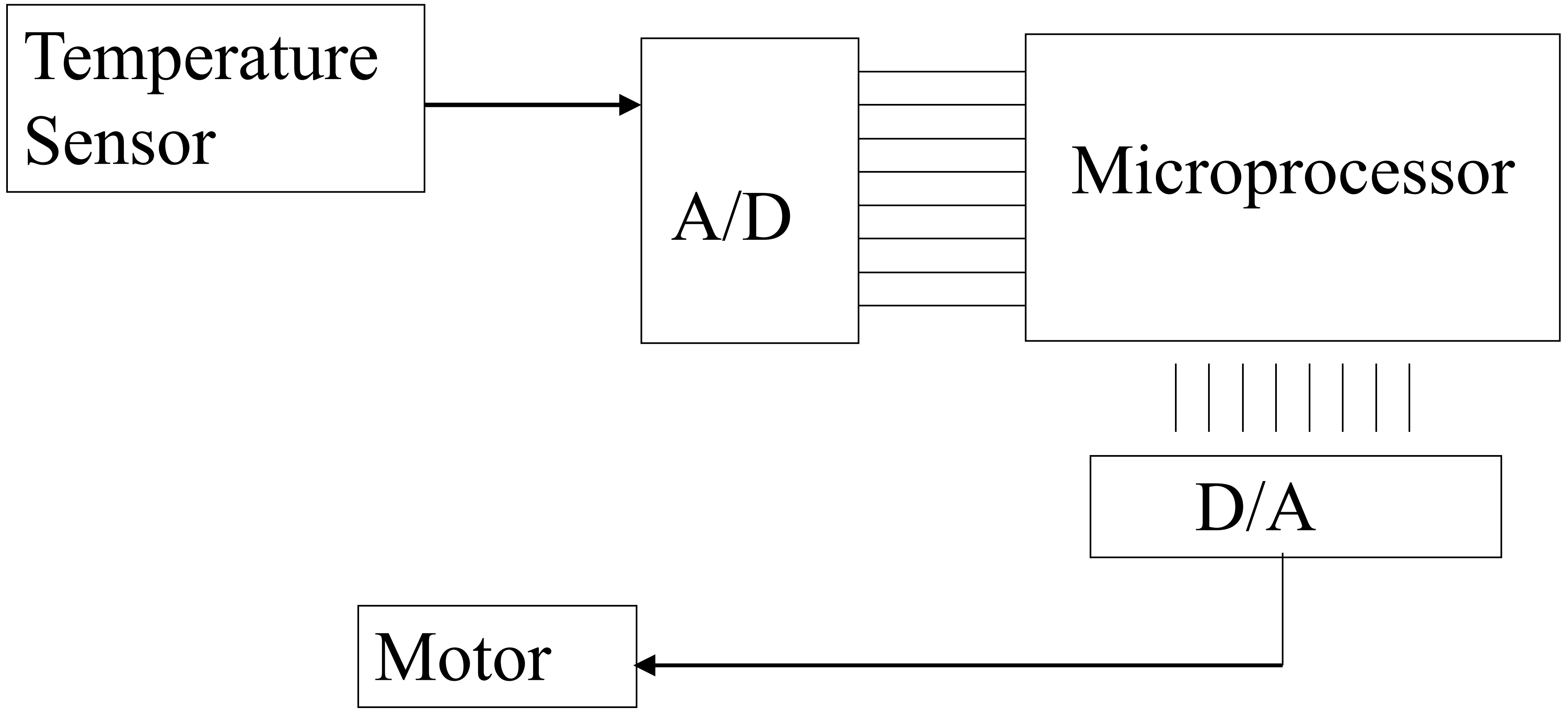
- HardWare/Electronics
  - to control Electriconic equipment
    - e.g. Motors（发动机）
  - to read Electronic sensors
    - Temperature sensor
    - Light sensor
    - GPS system

Hardware issues

- Analog / Digital
  - Real World is analog (continuously varying)
  - Computer World digital. Values only have descrete quantities
  - Reading a temperature sensor, need an analog to digital convertor.
  - Controlling an analog motor, need a digital to analog convertor.

Hardware issues

➢ Analog / Digital

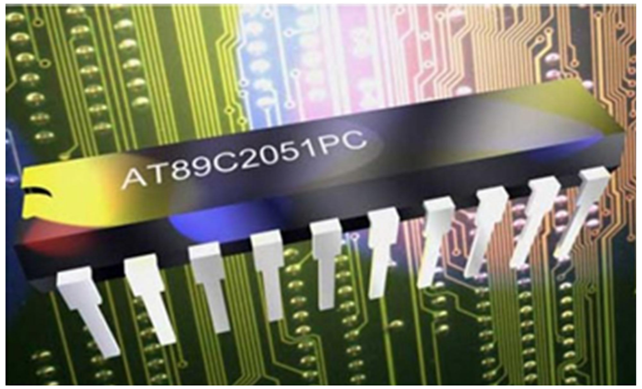


嵌入式处理器

- 嵌入式微处理器（MPU）：运算器、控制器
- 嵌入式微控制器（MCU）：片内ROM、RAM、总线、I/O口、计数器、看门狗、AD、DA、Flash
- 数字信号处理器（DSP）：哈佛结构，适用于FFT（快速傅立叶）变换、谱分析、数字滤波等操作，用于音频、视频处理
- 片上系统（SOC）：USB、GPRS、GPS、IEEE1394、蓝牙，可靠性强、开发时间短

嵌入式处理器

- MPU：Am186/88、386EX
- MCU：8051、P51XA
- DSP：TMS320系列、DSP56200系列
- SOC：M-core、CC2430



嵌入式处理器

- MPU：Am186/88、386EX
- 嵌入式微处理器是在通计算机中央处理器的基础上设计而来的，它将微处理器安装到专门设计的电路板上，只保留和嵌入式应用有关的主板功能，大幅减小了系统的体积和功耗。为了满足嵌入式应用的特殊要求，嵌入式微处理器在工作温度、抗电磁干扰、可靠性等方面都做了增强。

嵌入式处理器

- MCU：8051、P51XA
- 嵌入式微控制器又称单片机，它以某种微处理器内核为核心，将计算机系统的各个部分集成在一块芯片中，包括ROM/EPROM、RAM、总线、总线逻辑、定时/计数器、看门狗、I/O、串行口、脉宽调制输出、A/D、Flash RAM、EEPROM等。

嵌入式处理器

- DSP：TMS320系列、DSP56200系列
- DSP处理器对系统结构和指令进行特殊设计，使其适合于执行DSP算法，编译效率较高，，指令执行速度也较高。在数字滤波、FFT（快速傅立叶变换）、谱分析等方面DSP算法正在大量进入嵌入式领域。

嵌入式处理器

- SOC：M-core、CC2430
- 片上系统是采用硬件描述语言来设计各种处理器内核以及各种外设，把设计好的单元存储在器件库中，用户只需根据系统要求选用这些器件，仿真通过后就可以将设计图交给半导体工厂制作样品。

外围硬件设备

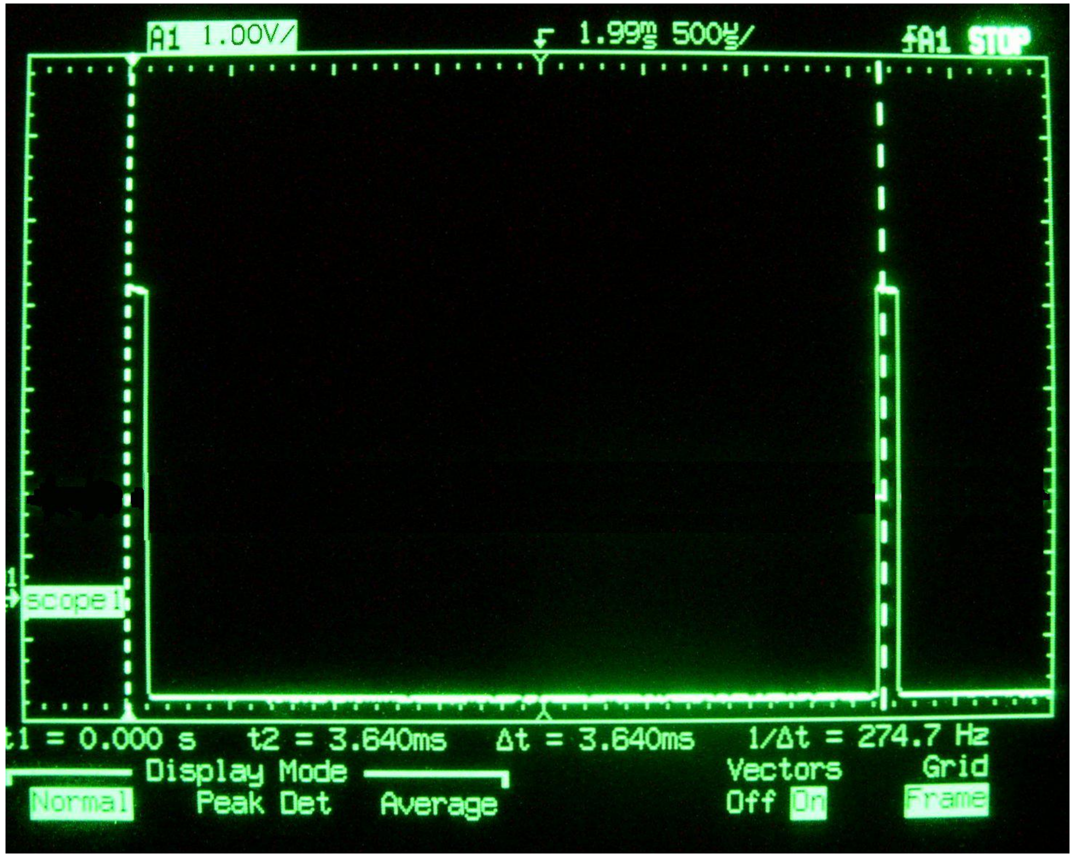
- 电源部分：交流电、电池供电
- 输入部分：键盘、鼠标、触摸屏、拨码开关等。
- 输出部分：发光二极管、LED显示器、LCD显示器、蜂鸣器等。
- 存储部分：ROM、EPROM、EEPROM、FLASH、RAM。
- 接口电路：USB接口、PS/2接口、串口、IDE接口、红外接口、1394接口、CF卡接口、网络接口、CAN总线接口、RS422接口、RS485接口等等，每一种接口一般都对应一个专用的控制芯片，例如串口一般是由8250芯片来控制。
- 其他硬件逻辑电路：AD转换电路、电机驱动电路、时钟日期生成电路等等。

Hardware methods

- ➢ Use hardware timing tools to measure execution times for critical modules

Example 1:

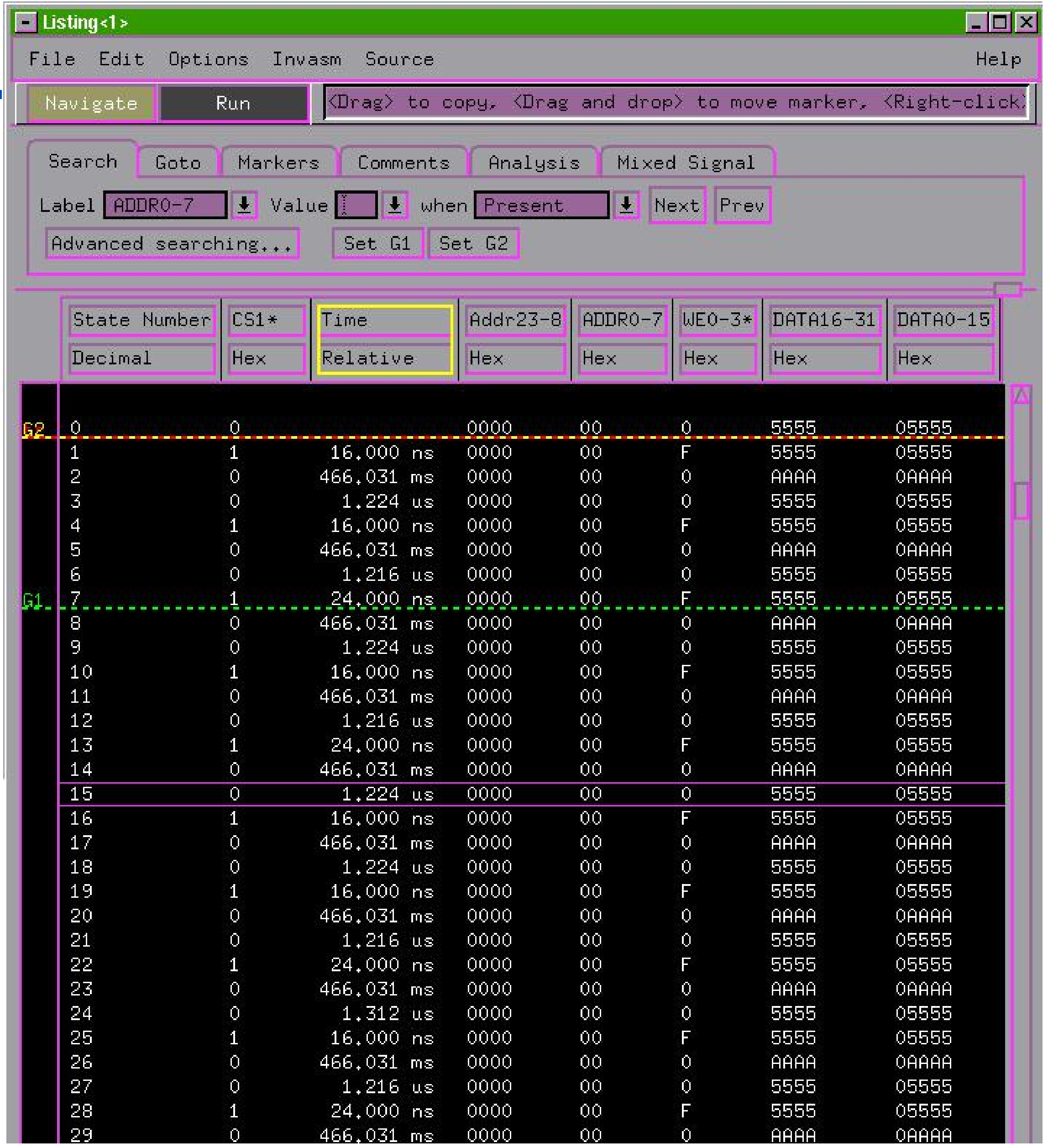
- ➢ Add small code segment to pulse I/O pin on processor at function entry and exit points
  - ❑ Use an oscilloscope to watch the I/O pin pulse



Hardware methods

Example 2:

- Use a logic analyzer Monitors the state of all processor bus signals
- Captures all data writes to a specific address
- Monitors the state of all processor bus signals


Listing<1>
File Edit Options Invasm Source Help
Navigate Run
<Drag> to copy, <Drag and drop> to move marker, <Right-click>
Search Goto Markers Comments Analysis Mixed Signal
Label ADDR0-7 Value when Present Next Prev
Advanced searching... Set G1 Set G2

	State Number	CS1*	Time	Addr23-8	ADDR0-7	WE0-3*	DATA16-31	DATA0-15
	Decimal	Hex	Relative	Hex	Hex	Hex	Hex	Hex
G2	0	0		0000	00	0	5555	05555
	1	1	16.000 ns	0000	00	F	5555	05555
	2	0	466.031 ms	0000	00	0	AAAA	0AAAA
	3	0	1.224 us	0000	00	0	5555	05555
	4	1	16.000 ns	0000	00	F	5555	05555
	5	0	466.031 ms	0000	00	0	AAAA	0AAAA
	6	0	1.216 us	0000	00	0	5555	05555
G1	7	1	24.000 ns	0000	00	F	5555	05555
	8	0	466.031 ms	0000	00	0	AAAA	0AAAA
	9	0	1.224 us	0000	00	0	5555	05555
	10	1	16.000 ns	0000	00	F	5555	05555
	11	0	466.031 ms	0000	00	0	AAAA	0AAAA
	12	0	1.216 us	0000	00	0	5555	05555
	13	1	24.000 ns	0000	00	F	5555	05555
	14	0	466.031 ms	0000	00	0	AAAA	0AAAA
	15	0	1.224 us	0000	00	0	5555	05555
	16	1	16.000 ns	0000	00	F	5555	05555
	17	0	466.031 ms	0000	00	0	AAAA	0AAAA
	18	0	1.224 us	0000	00	0	5555	05555
	19	1	16.000 ns	0000	00	F	5555	05555
	20	0	466.031 ms	0000	00	0	AAAA	0AAAA
	21	0	1.216 us	0000	00	0	5555	05555
	22	1	24.000 ns	0000	00	F	5555	05555
	23	0	466.031 ms	0000	00	0	AAAA	0AAAA
	24	0	1.312 us	0000	00	0	5555	05555
	25	1	16.000 ns	0000	00	F	5555	05555
	26	0	466.031 ms	0000	00	0	AAAA	0AAAA
	27	0	1.216 us	0000	00	0	5555	05555
	28	1	24.000 ns	0000	00	F	5555	05555
	29	0	466.031 ms	0000	00	0	AAAA	0AAAA

➢ Embedded Software Engineering

ENJOY THE COURSE !!!