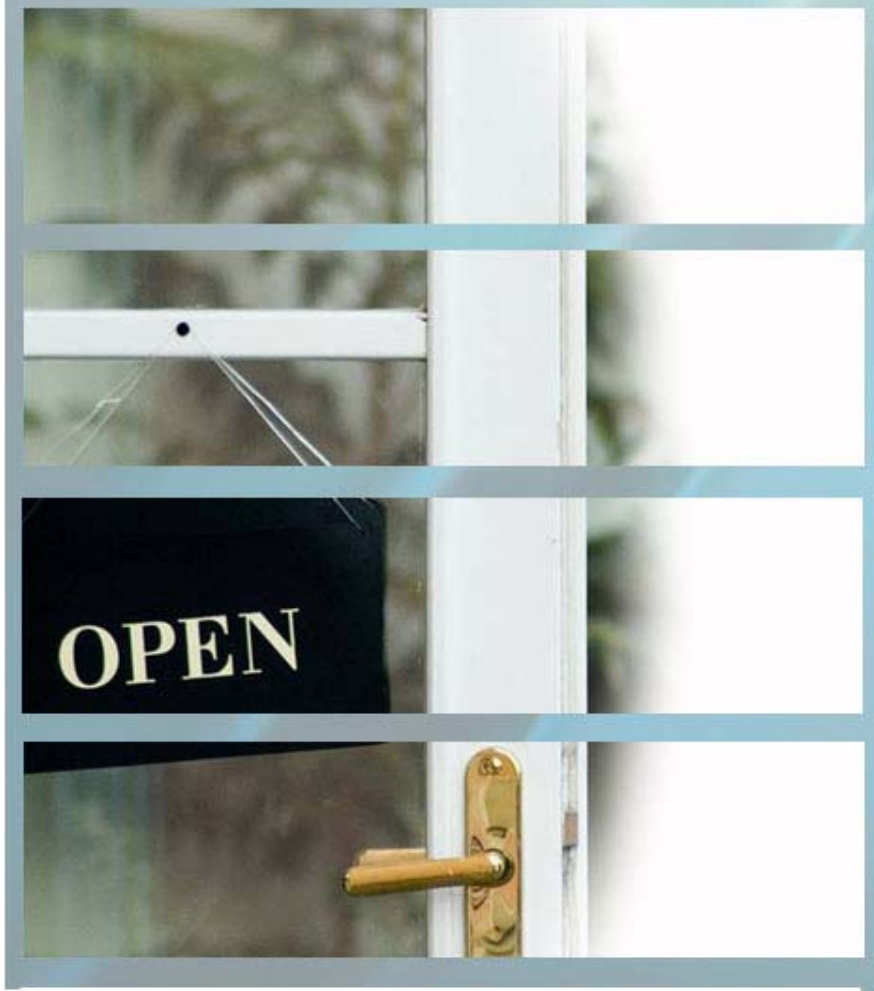
第3章
传感器技术

物联网导论

Introduction to Internet of Things



传感器作为信息获取的重要手段，与通信技术和计算机技术共同构成信息技术的三大支柱。

本章将介绍传感器的发展与应用以及软硬件平台。

内容提要

内容回顾

- 第2章介绍了常见的自动识别方法和技术，重点介绍了RFID技术
  - IC卡系统构成，一维和二维条形码
  - RFID的概念和系统组成，标签的存储方式、分类以及常用频率
  - RFID标签防冲突方法（基于ALOHA协议/基于二进制树协议）

- 本章重点介绍传感器技术，涉及传感器的基本概念和典型应用，以及常用的硬件平台和操作系统等内容。

本章内容

3.1 传感器概述
3.2 传感器技术发展史
3.3 典型应用
3.4 设计需求
3.5 硬件平台
3.6 操作系统

究竟什么是传感器？传感器有哪些部分组成呢？

3.1传感器概述

•定义

我国国家标准（GB7665-2005）对传感器的定义是：“能感受被测量并按照一定的规律转换成可用输出信号的器件或装置”。

•传统传感器的局限性

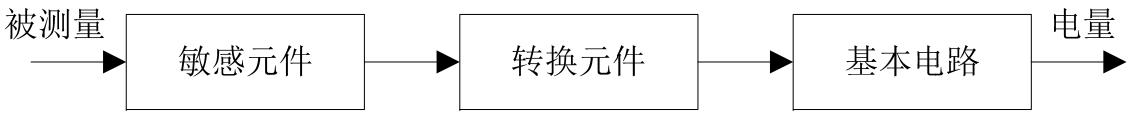
网络化、智能化的程度十分有限，缺少有效的数据处理与信息共享能力

•现代传感器

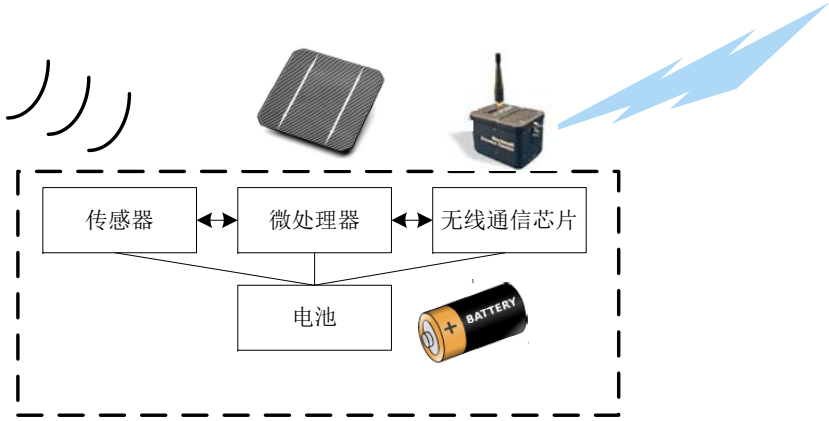
特点：微型化、智能化和网络化

典型代表：无线传感节点

无线传感节点



无线传感节点的组成：电池、传感器、微处理器、无线通信芯片；相比于传统传感器，无线传感节点不仅包括传感器部件（左上图），还集成了微型处理器和无线通信芯片等，能够对感知信息进行分析处理和网络传输。



本章内容

3.1 传感器概述

3.2 传感器技术发展史

3.3 典型应用

3.4 设计需求

3.5 硬件平台

3.6 操作系统

传感器发展的两条主线是什么？制约因素又有哪些？


3.2传感器技术发展史：两条主线

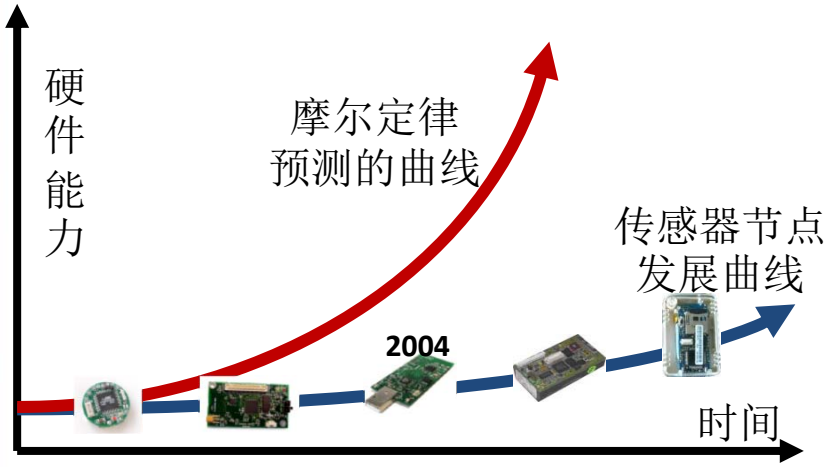
对无线传感器的研究始于20世纪90年代

- 加州洛杉矶分校LWIN项目
  低功耗无线传感节点
  - 1996年，LWIM团队将多种传感器、控制和通信芯片集成在一个设备上，开发了*LWIM节点*
  - 1998年，LWIM团队和Rockwell科学中心合作开发了*WINS节点*
- 加州伯克莱分校SmartDust项目
  微型化传感器节点
  - 1999年，该校发布了*WeC节点*
  - 之后，该校又发布了一系列节点，包括*Mica*、*Mica2*、*Mica2Dot*，*MicaZ*

3.2传感器技术发展史：缓慢提升的性能

计算机硬件的发展通常遵循**摩尔定律**：集成电路上可容纳的晶体管数量，约每隔18个月增加一倍，性能也将提升一倍。

无线传感器节点的发展并没有像摩尔定律预测的速度发展！





制约传感器性能提升的因素？

功耗的制约：无线传感节点一般被部署在野外，不能通过有线供电。其硬件设计必须以节能为重要设计目标。

价格的制约：无线传感节点一般需要大量组网，以完成特定的功能。其硬件设计必须以廉价为重要设计目标。

体积的制约：无线传感节点一般需要容易携带，易于部署。其硬件设计必须以微型化为重要设计目标。

本章内容

3.1 传感器概述
3.2 传感器技术发展史
3.3 典型应用
3.4 设计需求
3.5 硬件平台
3.6 操作系统

虽然传感器性能有限，但仍然得到广泛应用。

军事监测中的传感器：VigilNet

VigilNet是由美国弗吉尼亚大学研制的用于军事监测的无线传感系统，该系统由XSM，Mica2和Mica2Dot节点构成，其规模最大达200个节点；节点通过电池供电，铺设在道路旁边，用于检测与收集移动目标的情况。

应用特点

- 节点自主成网、多跳传输
- 节点通过电池供电，通过软件节能机制延长网络的生命周期
- 节点智能感知、协同工作，向上提供预警的功能

智能楼宇中的传感器： LoCal

每年美国用电报告显示至少有30%的电量是浪费的。这些电能浪费在何处？其中哪些是可以节省的？
由美国加州大学伯克利分校大学发起的LoCal项目试图通过在智能楼宇中部署无线传感器网络来解决这些问题。

应用特点

- 传感器能实现空间和时间上的细粒度感知，可实时跟踪到单个电器
- 传感器能实现“多功能”的感知，能推测用户的行为
- 传感器能够互联互通，通过大量连续的数据则有助于分析得出更多有用的信息

医疗监控中的传感器： Mercury

传感器的另一个重要应用是医疗监控，哈佛大学研究组改进了传统传感器，使得其外形更小，适合穿戴在身上

应用特点

- 传感器的设计十分人性化
- 传感器具有高精度的感知能力，医用的数据需要较高的采样精度供医生分析诊断
- 传感器能连续长期地采集数据
- 传感器使用无线通信方式，其数据传输是机会性的

本章内容

3.1 传感器概述
3.2 传感器技术发展史
3.3 典型应用
3.4 **设计需求**
3.5 硬件平台
3.6 操作系统

不同的应用场景对传感器软硬件提出独特的设计需求


大规模长时间部署传感器的设计需求

低成本与微型化

•低成本的节点才能被大规模部署，微型化的节点才能使部署更加容易

•节点的软件设计也需要满足微型化的需求 。例如TelosB节点的内存大小只有4KB，程序存储的空间只有10KB。因此，节点程序的设计必须节约计算资源，避免超出节点的硬件能力



大规模长时间部署传感器的设计需求

低功耗

- 在硬件设计上采用低功耗芯片

  例如TelosB节点使用的微处理器，在正常工作状态下功率为3mW，而一般的计算机的功率为200到300W

- 软件节能策略来实现节能

  软件节能策略的核心就是尽量使节点在不需要工作的时候进入低功耗模式，仅在需要工作的时候进入正常状态



大规模长时间部署传感器的设计需求

灵活性与扩展性

- 传感器节点被用于各种不同的应用中，因此节点硬件和软件的设计必须具有灵活性和扩展性
- 节点的硬件设计需满足一定的标准接口，例如节点和传感板的接口统一有利于给节点安装上不同功能的传感器
- 软件的设计必须是可剪裁的，能够根据不同应用的需求，安装不同功能的软件模块



大规模长时间部署传感器的设计需求

鲁棒性

- 鲁棒性是实现传感器网络长时间部署的重要保障
- 对于普通的计算机，一旦系统崩溃了，人们可以采用重启的方法恢复系统，而传感器节点则不行，就整个网络而言，可以适当增加冗余性，增加整体系统的鲁棒性


本章内容

3.1 传感器概述
3.2 传感器技术发展史
3.3 典型应用
3.4 设计需求
3.5 硬件平台
3.6 操作系统

结合设计需求可得出传感器节点硬件平台的基本特征

3.5 硬件平台

供能装置

•采用电池供电，使得节点容易部署。但由于电压、环境等变化，电池容量并不能被完全利用。

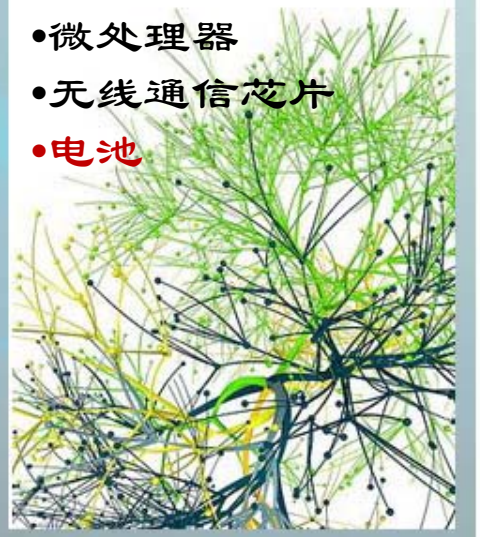
•可再生能量，如太阳能。可再生能源存储能量有两种方式：充电电池，自放电较少，电能利用会比较高，但充电的效率较低，且充电次数有限；超电容，充电效率高，充电次数可达100万次，且不易受温度，振动等因素的影响。

设计需求回顾

•低成本与微型化
•低功耗
•灵活性与扩展性
•鲁棒性

无线传感器组成

•传感器
•微处理器
•无线通信芯片
•电池



3.5 硬件平台

传感器

有许多传感器可供节点平台使用，使用哪种传感器往往由具体的应用需求以及传感器本身的特点决定

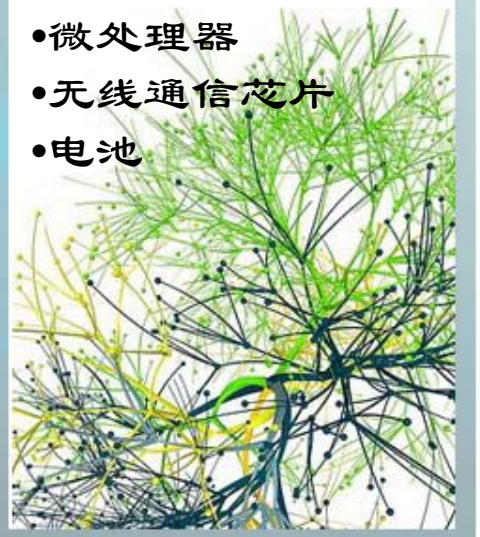
需要根据处理器与传感器的交互方式：通过模拟信号和通过数字信号，选择是否需要外部模数转换器和额外的校准技术。

设计需求回顾

- 低成本与微型化
- 低功耗
- 灵活性与扩展性
- 鲁棒性

无线传感器组成

- 传感器
- 微处理器
- 无线通信芯片
- 电池




常用传感器及其关键特性

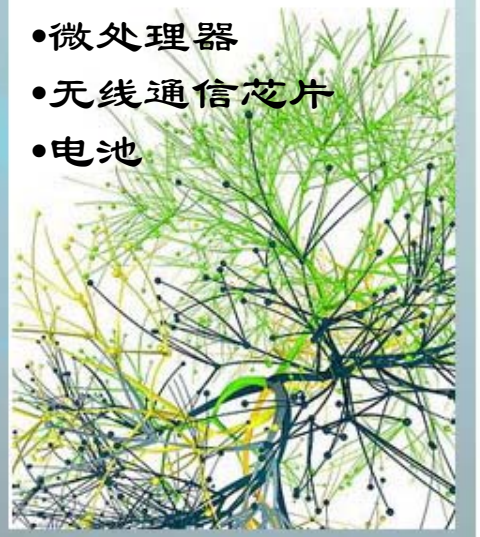
厂商	传感器	工作电压（V）	工作能耗	离散采样时间
Taos	可见光传感器	2.7-5.5	1.9mA	330us
Dallas Semiconductor	温度传感器	2.5-5.5	1mA	400ms
Sensirion	湿度传感器	2.4-5.5	550uA	300ms
Intersema	压强传感器	2.2-3.6	1mA	35ms
Honeywell	磁传感器	Any	4mA	30us
Analog Devices	加速度传感器	2.5-3.3	2mA	10ms
Panasonic	声音传感器	2-10	0.5mA	1ms
Motorola	烟传感器	6-12	5uA	-
Melixis	被动式红外传感器	Any	0mA	1ms
Li-Cor	合成光传感器	Any	0mA	1ms
Ech2o	土壤水分传感器	2-5	2mA	10ms

设计需求回顾

- 低成本与微型化
- 低功耗
- 灵活性与扩展性
- 鲁棒性

无线传感器组成

- 传感器
- 微处理器
- 无线通信芯片
- 电池



3.5 硬件平台

微处理器

微处理器是无线传感节点中负责计算的核心，目前的微处理器芯片同时也集成了内存、闪存、模数转化器、数字IO等，这种深度集成的特征使得它们非常适合在无线传感器网络中使用。

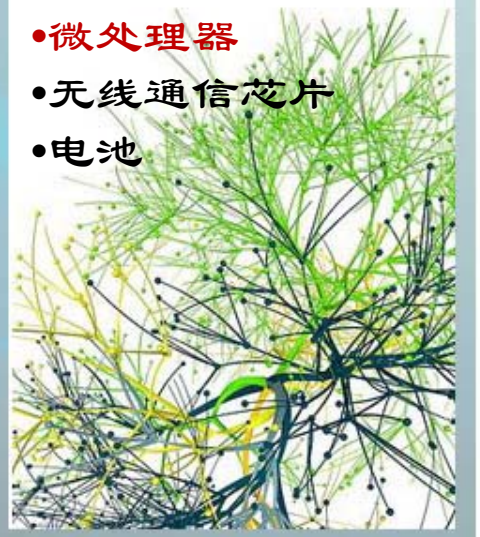
影响节点工作整体性能的微处理器关键性能包括功耗特性，唤醒时间（在睡眠/工作状态间快速切换），供电电压（长时间工作），运算速度和内存大小

设计需求回顾

- 低成本与微型化
- 低功耗
- 灵活性与扩展性
- 鲁棒性

无线传感器组成

- 传感器
- 微处理器
- 无线通信芯片
- 电池



常用微处理器及其关键特性

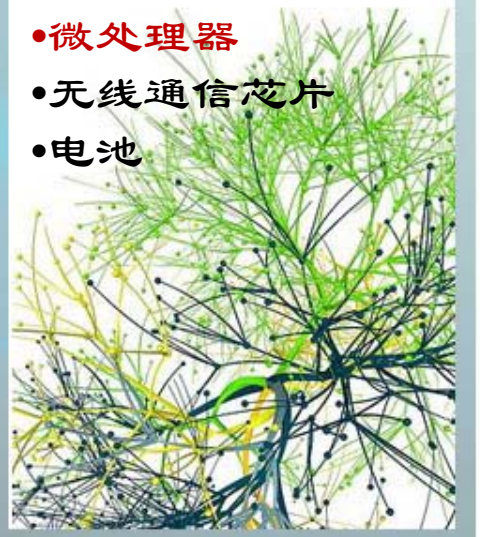
厂商	设备	发布年份	字长(位)	工作电压（V）	内存KB	闪存KB	工作能耗(mA)	睡眠能耗(uA)	唤醒时间(us)
Atmel	Atmega128L	2002	8	2.7-5.5	4	128	0.95	5	6
	Atmega1281	2005	8	1.8-5.5	8	128	0.9	1	6
	Atmega1561	2005	8	1.8-5.5	8	256	0.9	1	6
Ember	EM250	2006	16	2.1-3.6	5	128	8.5	1.5	>1000
Freescale	HC05	1988	8	3.0-5.5	0.3	0	1	1	>2000
	HC08	1993	8	4.5-5.5	1	32	1	20	4
	HCS08	2003	8	2.7-5.5	4	60	7.4	1	10
Jennic	JN5121	2005	32	2.2-3.6	96	128	4.2	5	>2500
	JN5139	2007	32	2.2-3.6	192	128	3.0	3.3	>2500
TI	Msp430F149	2000	16	1.8-3.6	2	60	0.42	1.6	6
	Msp430F1611	2004	16	1.8-3.6	10	48	0.5	2.6	6
	Msp430F2618	2007	16	1.8-3.6	8	116	0.5	1.1	1
	Msp430F5437	2008	16	1.8-3.6	16	256	0.28	1.7	5
ZiLOG	eZ80F91	2004	16	3.0-3.6	8	256	50	50	3200

设计需求回顾

- 低成本与微型化
- 低功耗
- 灵活性与扩展性
- 鲁棒性

无线传感器组成

- 传感器
- 微处理器
- 无线通信芯片
- 电池



3.5 硬件平台

通信芯片

•通信芯片是无线传感节点中重要的组成部分，在一个无线传感节点的能量消耗中，通信芯片通常消耗能量最多，在目前常用的TelosB节点上，CPU在工作状态电流仅500uA，而通信芯片在工作状态电流近20mA。

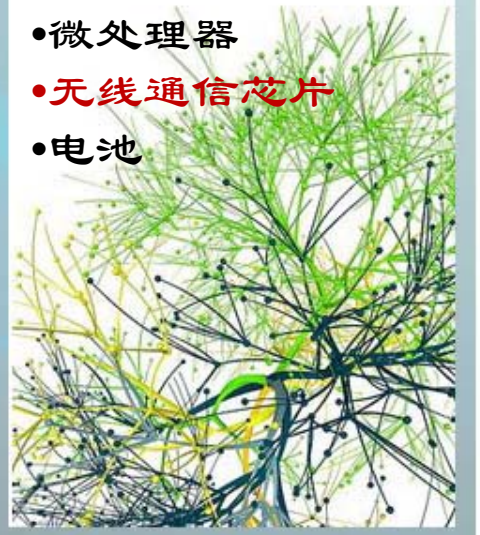
•低功耗的通信芯片在发送状态和接收状态时消耗的能量差别不大，这意味着只要通信芯片开着，都在消耗差不多的能量

设计需求回顾

- 低成本与微型化
- 低功耗
- 灵活性与扩展性
- 鲁棒性

无线传感器组成

- 传感器
- 微处理器
- 无线通信芯片
- 电池



3.5 硬件平台

通信芯片（续）

•通信芯片的传输距离是选择传感节点的重要指标。

发射功率越大，接受灵敏度越高，信号传输距离越远。

•常用通信芯片：

•CC1000：可工作在433MHz，868MHz和915MHz；采用串口通信模式时速率只能达到19.2Kbps

•CC2420：工作频率2.4GHz，是一款完全符合IEEE 802.15.4协议规范的芯片；传输率250Kbps

设计需求回顾

•低成本与微型化

•低功耗

•灵活性与扩展性

•鲁棒性

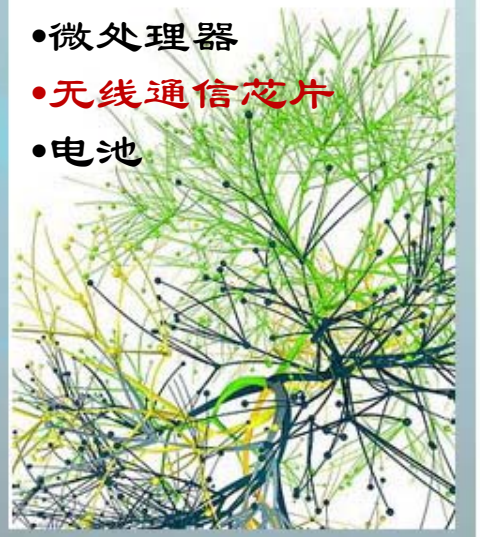
无线传感器组成

•传感器

•微处理器

•无线通信芯片

•电池



常用通信芯片及其关键特性

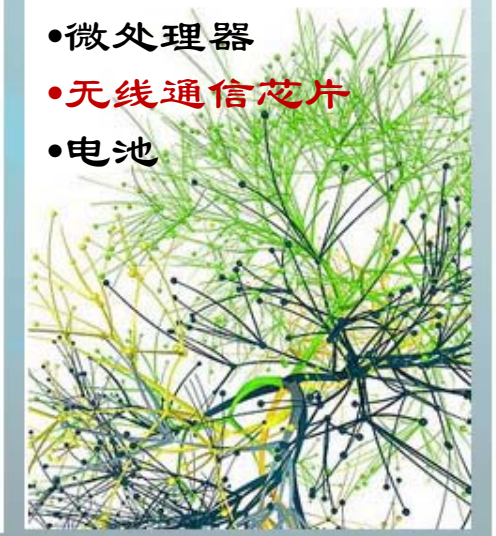
厂商	设备	发布年份	唤醒时间（ms）	接收灵敏度（dbm）	发射功率（dbm）	接收功耗（mA）	发送功耗（mA）	睡眠功耗（uA）
Atmel	RF230	2006	1.1	-101	+3	15.5	16.5	0.02
Ember	EM260	2006	1	-99	+2.5	28	28	1.0
Freescale	MC13192	2004	7-20	-92	+4	37	30	1.0
	MC13202	2007	7-20	-92	+4	37	30	1.0
	MC13212	2005	7-20	-92	+3	37	30	1.0
Jennic	JN5121	2005	>2.5	-93	+1	38	28	<5.0
	JN5139	2007	>2.5	-95.5	+0.5	37	37	2.8
TI	CC2420	2003	0.58	-95	0	18.8	17.4	1
	CC2430	2005	0.65	-92	0	17.2	17.4	0.5
	CC2520	2008	0.50	-98	+5	18.5	25.8	0.03

设计需求回顾

- 低成本与微型化
- 低功耗
- 灵活性与扩展性
- 鲁棒性

无线传感器组成

- 传感器
- 微处理器
- 无线通信芯片
- 电池

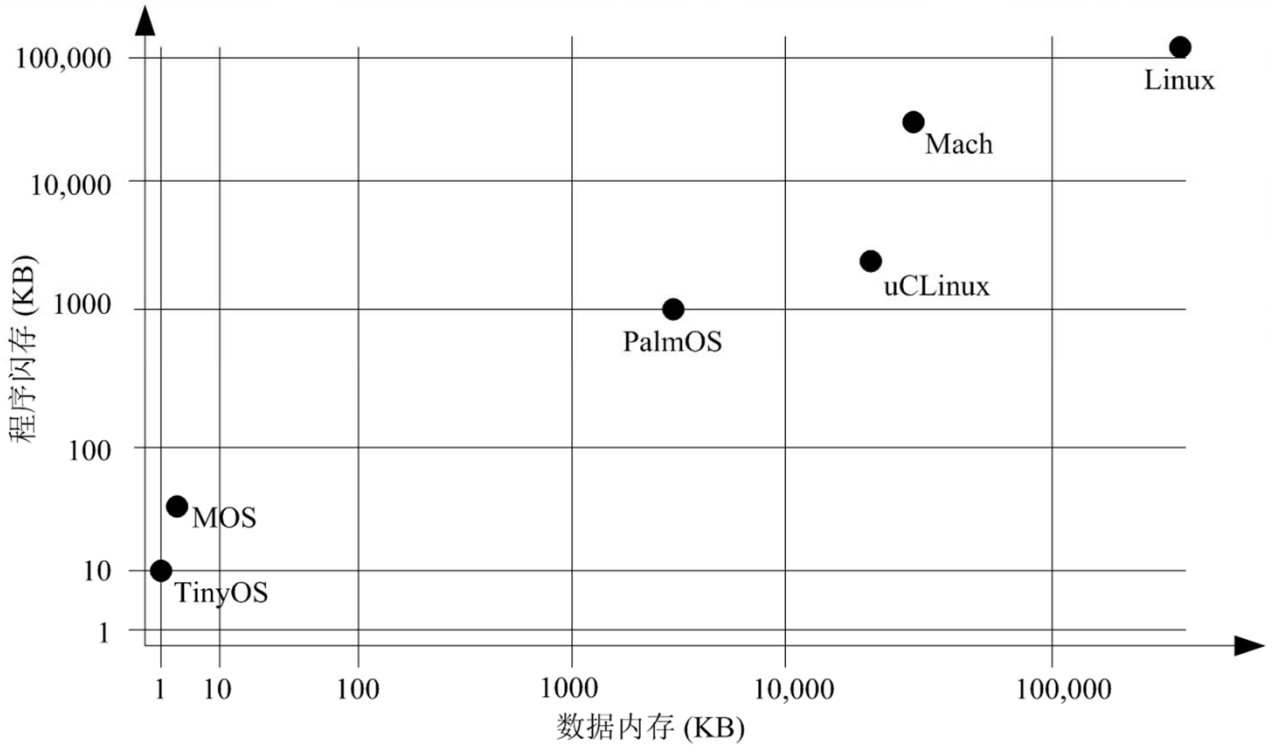


本章内容

3.1 传感器概述
3.2 传感器技术发展史
3.3 典型应用
3.4 设计需求
3.5 硬件平台
3.6 操作系统

操作系统是传感器节点软件系统的核心

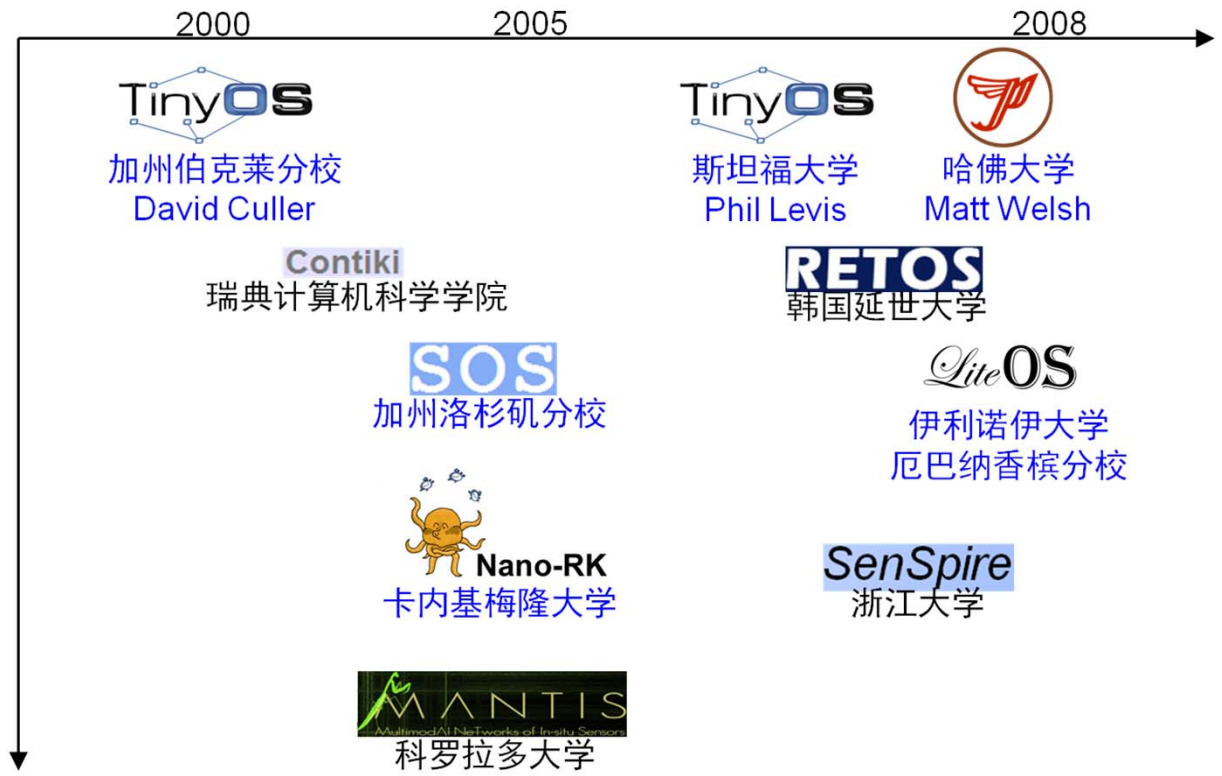
节点操作系统VS其他操作系统



由图可得，节点操作系统是极其微型化的。

节点操作系统区别于传统操作系统的主要特点是：**其硬件平台资源极其有限**

节点操作系统发展史



节点操作系统区别于传统操作系统的主要特点是：**其硬件平台资源极其有限**

TinyOS

TinyOS由加州伯克莱分校开发,是目前无线传感网络研究领域使用最为广泛的OS (http://www.tinyos.net)

TinyOS开发语言：nesC

- nesC语言是专门为资源极其受限、硬件平台多样化的传感节点设计的开发语言
- 使用nesC编写的应用程序是基于组件的
- 组件之间的交互必须通过使用接口
- 用nesC编写的应用程序一般有一个最顶层的配置文件

节点操作系统功能：

- 硬件驱动
- 资源管理
- 任务调度
- 编程接口

TinyOS代码举例

```
// BlinkC.nc
module BlinkC {
  uses interface Timer<TMilli> as Timer;
  uses interface Leds;
  uses interface Boot;
} implementation {
  event void Boot.booted() {
  call Timer.startPeriodic( 250 );
}
  event void Timer.fired() { call Leds.led0On(); }
}
```

左侧代码中：
BlinkC就表示一个组件，它使用了三个接口：Timer，Leds，Boot。

在其实现部分，它可以调用这些接口提供的服务，如Timer.startPeriodic启动一个以250ms周期触发的时钟，而Leds.led0Toggle使节点上第一个灯亮起。

在上面的代码中，注意的是，event关键字表示BlinkC组件处理的系统事件。

TinyOS代码举例

```
// BlinkCApp.nc
configuration BlinkAppC {}
implementation {
  components MainC, BlinkC, LedsC;
  components new TimerMilliC() as TimerC;
  BlinkC -> MainC.Boot;
  BlinkC.Timer -> TimerC;
  BlinkC.Leds -> LedsC;
}
```

左侧代码显示了一个典型的nesC配置文件。它必须指定当前程序使用了哪些组件。例如该程序使用了MainC，BlinkC（即代码1显示的组件），LedsC和TimerC组件。

BlinkC组件中使用的接口到底是由哪个组件提供的，例如，BlinkC组件使用的Boot接口由MainC组件提供；BlinkC组件使用的Timer接口由TimerC组件提供；BlinkC组件使用的Leds接口由LedsC组件提供。

TinyOS（续）

TinyOS任务调度

- TinyOS核心使用了事件驱动的单线程任务调度机制，这和传统OS的多线程调度机制截然不同
- 任何一个时刻，处理器只能执行一个任务。因此，如果当前正在执行一个任务，处理器必须等这个任务处理完毕，才能开始处理另一个任务
- 在单个TinyOS任务中不能有IO等阻塞的调用

节点操作系统功能：

- 硬件驱动
- 资源管理
- 任务调度
- 编程接口

其他常用微型OS对比

	TinyOS	Contiki	SOS	Mantis	Nano-RK	RETOS	LiteOS
发表会议(年份)	ASPLOS (2000)	EmNets (2004)	MobiSys (2005)	MONET (2005)	RTSS (2005)	IPSN (2007)	IPSN (2008)
静态/动态	静态	动态	动态	动态	静态	动态	动态
事件驱动/多线程	事件驱动 & 多线程 TOSThreads	事件驱动 & 多线程	事件驱动	多线程 & 事件驱动 TinyMOS	多线程	多线程	多线程
单核/模块化	单核	模块化	模块化	模块化	单核	模块化	模块化
网络层	主动消息	uIP, uIPv6, Rime	消息	“comm”层	套接字	三层架构	
实时支持	否	否	否	否	是	符合 POSIX 1003.1b	否
语言支持	nesC	C	C	C	C	C	LiteC++

节点操作系统功能：

- 硬件驱动
- 资源管理
- 任务调度
- 编程接口

传感网的研究问题-1

- **Geographic Routing in Wireless Sensor Networks**
- **Medium Access Control in Wireless Sensor Networks**
- **Localization in Wireless Sensor Networks**
- **Data Aggregation in Wireless Sensor Networks**
- **Clustering in Wireless Sensor Networks**

传感网的研究问题-2

- **Energy-Efficient Sensing in Wireless Sensor Networks**
- **Mobility in Wireless Sensor Networks**
- **Security in Wireless Sensor Networks**
- **Network Management in Wireless Sensor Networks**
- **Deployment in Wireless Sensor Networks**



传感网研究的挑战-三大科学问题

- 易感不易传和易传不易感的矛盾
  - 传感失谐
- 可见性缺失与管理系统化的冲突
  - 诊断失据
- 大规模使用对理想化研究的挑战
  - 模型失用



易感不易传和易传不易感的矛盾

- 易感不易传：图像、声音、视频等传感数据
- 易传不易感：CO2含量、光谱、地震波等
- 网络传输带宽瓶颈
- 路由协议可靠性不足
- 异构传感网的体系结构和数据管理



可见性缺失和管理系统化的冲突

- 目前传感网诊断体系的弊端不足
  - 模型封闭(适应性差)
  - 拓补耦合（网络诊断与拓补或应用相关）
  - 单目交互（诊断目标单一）
  - 网络重载（负载重）
  - 离线调整
- 传感网诊断方法应达到如下标准
  - 轻量级：低带宽、低存储、低计算量
  - 通用性：需求无依赖、应用无依赖、平台无依赖



大规模实用对理想化研究的挑战

- 基于理想模型假设的研究与真实应用存在差距
  - 已有研究成果面临挑战（如定位、覆盖、拓补控制等）
  - 实验室产品无法应用于大规模实用系统
- 仿真调试平台偏理想化
  - 忽略环境因素、无法模拟MAC层以下事件，对链路质量、节点可靠性等因素建模偏离实际情况
- 更多规模化应用问题待解决

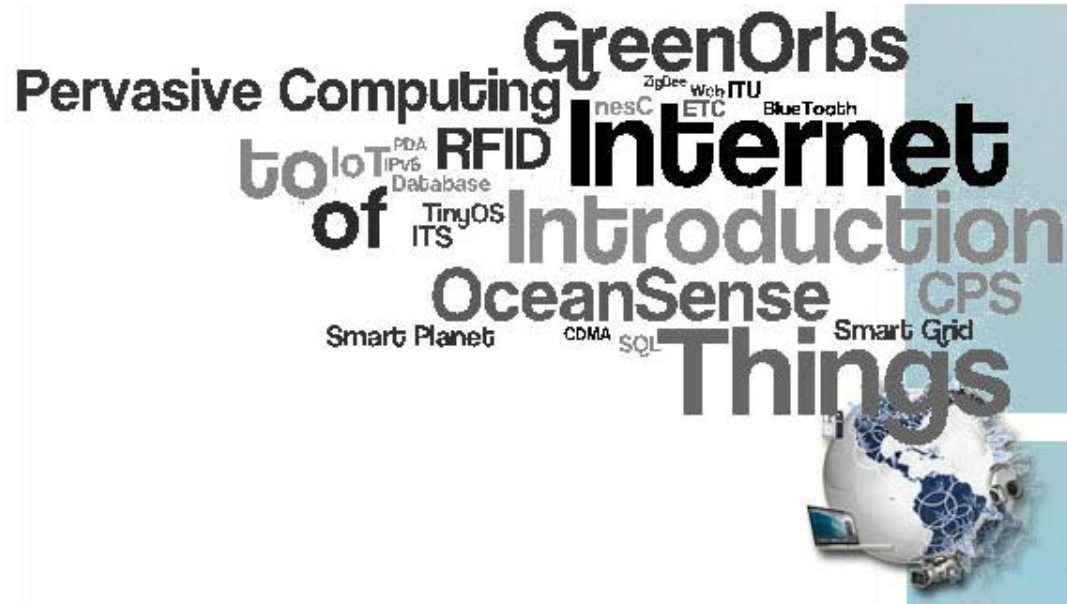

本章小结

内容回顾

本章介绍了传感器的基本概念和典型应用，并讨论了传感器的设计需求和软硬件平台，以TinyOS为例简单介绍了节点操作系统。

重点掌握

- 现代传感器的基本组成以及各部分的软硬件平台特点和需求。
- 掌握制约传感器性能提升的瓶颈以及相应的设计需求（低成本与微型化，低功耗，灵活性与扩展性，鲁棒性）
- 了解节点操作系统的主要特点以及TinyOS/nesC编程的基本框架



Thank you!

Internet of Things