

# Chương 2: Kiến Trúc Nút Đơn

Tổng Quan về Nút Cảm Biến trong  
Mạng Cảm Biến Không Dây (WSN)

# Mục đích của chương

- ▶ Chương này giải thích phần tử cơ bản của WSN là nút cảm biến.
- ▶ Nút cảm biến thực hiện nhiều nhiệm vụ quan trọng trong mạng, từ tính toán, lưu trữ đến giao tiếp và cảm nhận/vận hành.

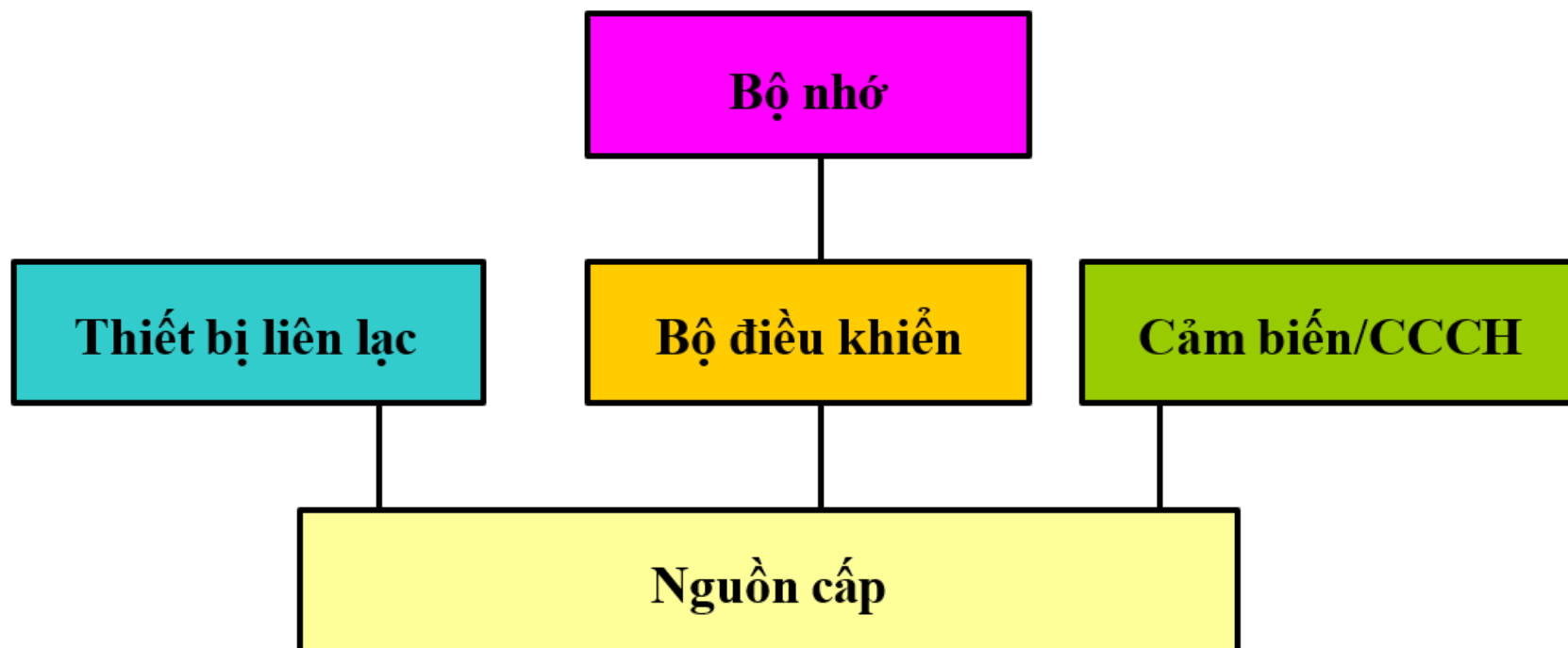
# Mục đích của chương

- ▶ Các bộ phận cần thiết để thực hiện các nhiệm vụ này bao gồm bộ điều khiển, bộ nhớ, cảm biến và các thiết bị liên lạc.
- ▶ Tiêu thụ năng lượng của các bộ phận này cũng được quan tâm đặc biệt.

# Yêu cầu của WSN

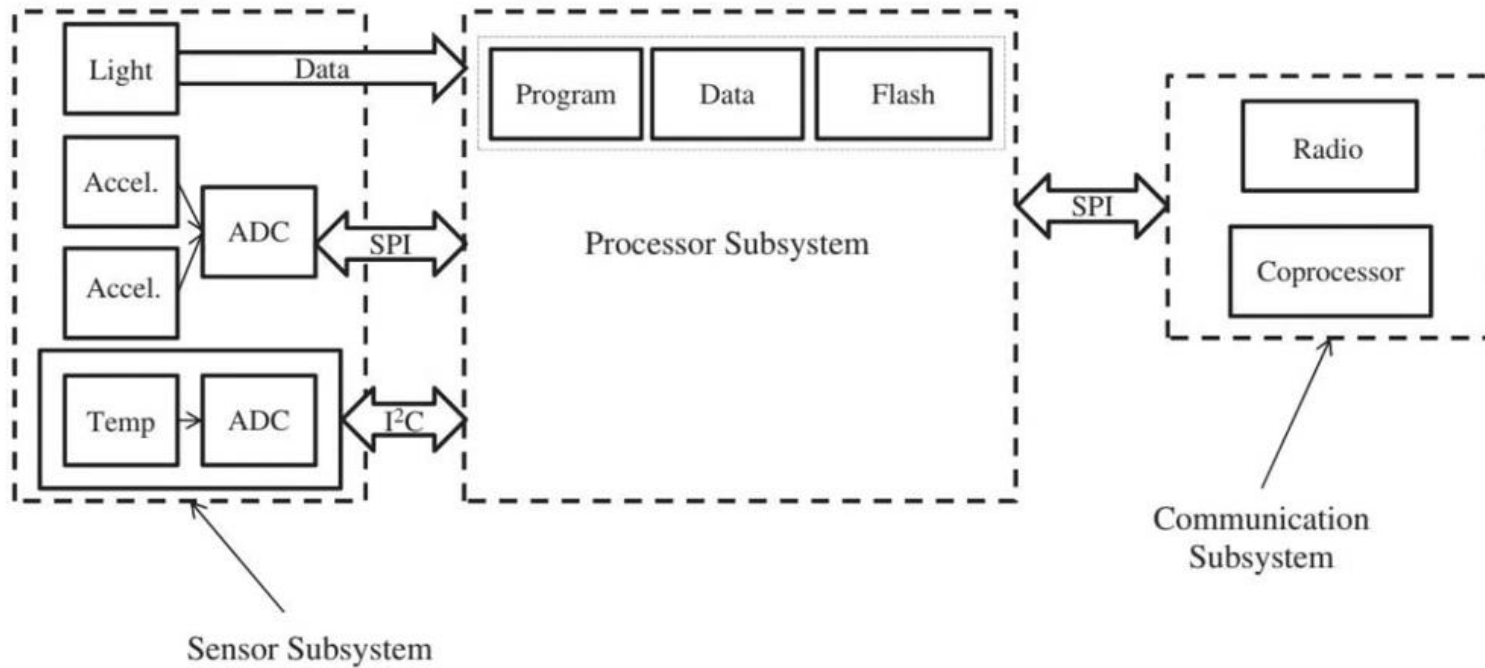
- ▶ Các yêu cầu đối với nút cảm biến bao gồm kích thước nhỏ gọn, giá thành thấp, và tiết kiệm năng lượng.
- ▶ Các yêu cầu này giúp đảm bảo tính khả thi của WSN trong nhiều ứng dụng khác nhau.

# Cấu trúc phần cứng của nút cảm biến



# Yêu cầu của WSN

## Kiến trúc chung của cảm biến không dây



# Cấu trúc phần cứng

- **Bộ điều khiển:** Bộ điều khiển xử lý tất cả dữ liệu có liên quan theo mã nhị phân.
- **Bộ nhớ:** Thông thường, bộ nhớ được dùng để lưu giữ chương trình và dữ liệu.
- **Cảm biến và cơ cấu chấp hành:** Đây là giao diện với môi trường vật lý. Các thiết bị này có thể quan sát hoặc giám sát các thông số vật lý của môi trường.
- **Thiết bị liên lạc:** Các nút trong mạng đòi hỏi một thiết bị để gửi và nhận thông tin qua kênh vô tuyến.
- **Nguồn cấp:** Thông thường, không có sẵn nguồn cấp điện nên cần phải có pin để cung cấp năng lượng, thậm chí cả pin có thể sạc từ môi trường (pin mặt trời).

# Cấu trúc phần cứng của nút cảm biến

- Bộ điều khiển là cốt lõi của một nút cảm biến không dây. Nó thu thập dữ liệu từ các cảm biến, xử lý dữ liệu này, quyết định thời gian và địa điểm gửi dữ liệu đến, nhận dữ liệu từ các nút khác tác động đến cơ cấu chấp hành.
- Các bộ xử lý này thường được gọi là các **bộ vi điều khiển**.



# Tổng quan về các bộ phận phần cứng

- ▶ Các bộ phận này phải hoạt động một cách tối ưu để đảm bảo giảm thiểu tiêu thụ năng lượng.
- ▶ Hệ thống quản lý năng lượng giúp duy trì hoạt động lâu dài của nút cảm biến.
- ▶

# Bộ điều khiển

- ▶ Bộ điều khiển chịu trách nhiệm xử lý dữ liệu từ các cảm biến, tính toán và đưa ra quyết định về việc gửi/nhận dữ liệu.
- ▶ Bộ điều khiển cần có khả năng hoạt động hiệu quả với các cảm biến và các bộ phận khác.
- ▶

# Bộ điều khiển

- ▶ Các bộ vi điều khiển tiết kiệm năng lượng là yếu tố quan trọng trong thiết kế nút cảm biến.
- ▶ Các vi điều khiển phổ biến là Intel StrongARM, Texas Instruments MSP 430 và Atmel ATmega.

# Vi điều khiển và các loại

Vi điều khiển Intel StrongARM cung cấp khả năng xử lý nhanh với tốc độ lên đến 206 MHz, thích hợp cho các ứng dụng cao cấp.



# Vi điều khiển

## ► Texas Instruments MSP 430

Texas Instrument cung cấp một họ các bộ vi điều khiển dưới tên gọi MSP 430. Không giống như StrongARM, nó rất rõ ràng là dùng cho các ứng dụng nhúng.

## ► Atmel Atmega

Atmel ATmega 128L là một bộ vi điều khiển 8 bit, cũng thường dùng cho các ứng dụng nhúng, được trang bị các giao diện bên ngoài để dùng cho các thiết bị ngoại thông thường.

# Vi điều khiển mới

## ❖ 1. Vi điều khiển tiêu thụ điện năng thấp (Low-Power MCU)

- ARM-Cortex-M0/M0+/M3/M4/M33 (STMicroelectronics, NXP, Nordic, Silicon Labs, Texas Instruments) – phổ biến nhờ hiệu năng cao, tiêu thụ năng lượng thấp, hỗ trợ nhiều giao thức không dây.
- TI MSP430 (thế hệ mới) – vẫn được dùng cho các ứng dụng yêu cầu siêu tiết kiệm năng lượng.
- Atmel/Microchip SAM D, SAM L – dựa trên ARM Cortex-M, thay thế cho dòng AVR cũ.

# Vi điều khiển mới

- ARM Cortex-M0:

- Là một trong những bộ xử lý 32-bit tiết kiệm năng lượng nhất, có kích thước silicon nhỏ và kích thước mã tối thiểu.
- Phù hợp cho các ứng dụng nhúng có giới hạn về tài nguyên và chức năng đơn giản.

- ARM Cortex-M0+:

- Là phiên bản nâng cấp của Cortex-M0, mang lại hiệu suất CPU được cải thiện và các tính năng như Bộ bảo vệ bộ nhớ (MPU), giao diện I/O chu kỳ đơn và Bộ đếm theo dõi vi mô (MTB).
- Lý tưởng cho các thiết bị nhúng cần hiệu suất 32-bit với mức tiêu thụ năng lượng rất thấp.

- ARM Cortex-M3:

- Một bộ xử lý mạnh mẽ hơn, cung cấp hiệu suất cao hơn Cortex-M0/M0+ với kiến trúc ARMv7-M.
- Thích hợp cho các ứng dụng yêu cầu hiệu năng và khả năng xử lý tốt hơn, nhưng vẫn giữ mức tiêu thụ năng lượng hợp lý.

# Vi điều khiển mới

- **ARM Cortex-M4:**

- Là bộ xử lý hiệu năng cao, có thêm tập lệnh DSP (Xử lý tín hiệu số) và FPU (Bộ xử lý dấu phẩy động) tùy chọn.
- Phù hợp cho các ứng dụng xử lý tín hiệu, điều khiển động cơ và các ứng dụng đòi hỏi hiệu năng cao hơn.

- **ARM Cortex-M33:**

- Sử dụng cho các ứng dụng nhúng tiên tiến, Internet of Things (IoT), và các hệ thống yêu cầu mức độ bảo mật cao.

Tóm lại:

Dãy Cortex-M từ M0 đến M33 là một phân cấp của các bộ xử lý 32-bit, trong đó các lõi M0/M0+ tập trung vào hiệu quả năng lượng và chi phí, còn các lõi M3, M4 và M33 được nâng cấp về hiệu năng, bộ nhớ và tính năng an ninh để phục vụ các ứng dụng ngày càng phức tạp.




## ❖ 2. Vi xử lý tích hợp module không dây (SoC – System on Chip)

- Nordic Semiconductor nRF52/nRF53 – tích hợp Bluetooth Low Energy, Thread, Zigbee.
- ESP32/ESP32-C3/ESP32-S3 (Espressif) – tích hợp Wi-Fi + Bluetooth, giá rẻ, dễ lập trình.
- Silicon Labs EFR32 (Gecko series) – hỗ trợ Zigbee, Thread, Bluetooth, Sub-GHz.
- TI CC26xx/CC13xx – hỗ trợ BLE, Zigbee, Sub-1 GHz.

# Vi điều khiển mới

## [MCU] Một Số Dòng Vi Điều Khiển Phổ Biến

1 đang bắt đầu một dự án IoT, robot, thiết bị đeo hay hệ thống công nghiệp tự động hóa? Dưới đây là một số dòng MCU phổ biến mà bạn nên biết để lựa chọn phần cứng phù hợp nhất! 

### 1. STM32 (STMicroelectronics)

ARM Cortex-M0/M3/M4/M7

Hiệu năng cao – Tiết kiệm điện – Giao tiếp đa dạng

Ứng dụng: Thiết bị y tế, robot, IoT, hệ thống nhúng

### 2. ATmega (Microchip)

Kiến trúc AVR 8-bit

Dễ học – Lập trình bằng Arduino IDE – Cộng đồng lớn

Ứng dụng: Arduino Uno, giáo dục, DIY

### 3. PIC (Microchip)

8/16/32-bit đa dạng

Dùng MPLAB X IDE – Linh hoạt và mạnh mẽ

Ứng dụng: Điều khiển công nghiệp, ô tô, thiết bị gia dụng

### 4. ESP32 (Espressif)

Tích hợp Wi-Fi & Bluetooth

Mạnh mẽ, giá tốt, tiết kiệm điện

Ứng dụng: IoT, nhà thông minh, thiết bị đeo

# Vi điều khiển mới

## 5. ESP8266 (Espressif)

Wi-Fi tích hợp – Giá siêu rẻ

Lập trình dễ – Phù hợp dự án IoT nhỏ

Ứng dụng: Cảm biến, điều khiển từ xa

## 6. 8051 (Nhiều hãng)

Kiến trúc cổ điển 8-bit

Học dễ – Tài liệu nhiều

Ứng dụng: Thiết bị gia dụng, giáo dục

## 7. MSP430 (Texas Instruments)

16-bit RISC – Tiết kiệm pin cực tốt

Ứng dụng: Thiết bị y tế, đeo tay, cảm biến không dây

## 8. NXP

ARM Cortex-M

Hiệu năng cao, tích hợp tốt

Ứng dụng: Ô tô, Công nghiệp, IoT

## 9. Renesas RX

32-bit CISC – Bảo mật tốt

Ứng dụng: Ô tô, công nghiệp, thiết bị y tế

## 10. Silicon Labs EFM32

Siêu tiết kiệm điện – ARM Cortex-M

Ứng dụng: Thiết bị đeo, IoT dùng pin

# Vi điều khiển mới

Dòng MCU	Kiến trúc	Đặc điểm nổi bật	Ứng dụng chính
STM32	ARM Cortex-M	Hiệu năng cao, đa dạng giao tiếp	Công nghiệp, robot, IoT
ATmega	AVR 8-bit	Dễ lập trình, cộng đồng lớn	Arduino, giáo dục, DIY
PIC	8/16/32-bit	Đa dạng, công cụ phát triển mạnh	Công nghiệp, tự động hóa
ESP32	Xtensa 32-bit	Wi-Fi/Bluetooth tích hợp	IoT, nhà thông minh
ESP8266	Xtensa 32-bit	Wi-Fi tích hợp, giá rẻ	IoT đơn giản, cảm biến không dây
8051	8-bit	Kiến trúc cổ điển, dễ học	Giáo dục, thiết bị gia dụng
MSP430	16-bit RISC	Tiêu thụ điện năng thấp	Thiết bị đeo, cảm biến không dây
NXP LPC	ARM Cortex-M	Hiệu năng cao, tích hợp nhiều tính năng	Công nghiệp, IoT
Renesas RX	32-bit CISC	Hiệu năng cao, bảo mật tốt	Công nghiệp, ô tô, y tế
Silicon Labs EFM32	ARM Cortex-M	Tiêu thụ điện năng cực thấp	Thiết bị đeo, IoT chạy bằng pin

# Bộ nhớ

- ▶ Bộ nhớ trong nút cảm biến bao gồm RAM (truy cập nhanh) và ROM, EEPROM hoặc FLASH.
- ▶ Bộ nhớ RAM giúp lưu trữ dữ liệu tạm thời và nhanh chóng, trong khi ROM/FLASH lưu trữ mã chương trình.
- ▶ Bộ nhớ cần được lựa chọn sao cho phù hợp với các yêu cầu về tốc độ và dung lượng, đồng thời giảm thiểu tiêu thụ năng lượng.

# Môi trường truyền dẫn

- Thiết bị liên lạc được sử dụng để trao đổi dữ liệu giữa các nút riêng rẽ.
- Các lựa chọn khác như giao tiếp dựa vào tần số vô tuyến (RF - Radio Frequency) là thích hợp nhất, nó cũng phù hợp với hầu hết yêu cầu của các ứng dụng WSN:
- Với hệ thống dựa vào RF vô tuyến thực tế, tần số liên lạc nằm trong khoảng từ 433 MHz - 2.4 GHz.

# Các bộ thu phát

Hiện nay, liên lạc yêu cầu cả bộ phát và bộ thu trong một nút cảm biến. Nhiệm vụ cốt yếu là biến đổi luồng bit đến từ bộ vi điều khiển (hoặc dãy byte, hoặc khung) thành/từ các sóng vô tuyến.

Loại bộ thu phát có giá thành thấp hữu dụng về phương diện thương mại sẽ được kết hợp trong tất cả các mạch được yêu cầu cho việc phát và thu – điều chế, giải điều chế, các bộ khuếch đại, bộ lọc, bộ trộn...

# Môi trường truyền dẫn

## Phục vụ các lớp cao hơn

Bộ thu phát không hoạt động độc lập mà phải hỗ trợ cho các lớp giao thức ở mức cao hơn, phổ biến nhất là **lớp điều khiển truy cập môi trường (MAC – Medium Access Control)**.

- Ở mức cơ bản, bộ thu phát chỉ cung cấp **giao diện bit hoặc byte** cho vi điều khiển.

- Ở mức cao hơn, nó có thể hỗ trợ **giao diện khung (frame interface)**, cho phép lớp MAC dễ dàng quản lý việc truyền và nhận gói tin.

- Trong một số thiết kế, bộ thu phát còn cho phép lớp MAC **truyền trực tiếp gói tin** vào bộ nhớ của nút cảm biến (dưới dạng khung, chùm byte, hoặc bitstream) mà không cần vi điều khiển phải xử lý chi tiết.

- Với gói tin đến, dữ liệu thường được **lưu tạm trong bộ đệm (buffer)** của bộ thu phát, để lớp MAC có thể truy cập và xử lý khi cần.



# So sánh một số bộ thu phát trong WSN

Đặc điểm	CC2420 (TI)	nRF24L01 (Nordic)	CC1000 (TI)
Chuẩn hỗ trợ	IEEE 802.15.4 (ZigBee)	Giao thức riêng, 2.4 GHz	Không theo chuẩn, 300-1000 MHz
Giao diện dữ liệu	Khung (frame interface)	Byte (qua SPI)	Bitstream (chuỗi bit)
Bộ đệm (buffer)	Có (FIFO)	Có (FIFO nhỏ)	Không đáng kể
Gánh nặng cho vi điều khiển	Thấp (MAC được hỗ trợ nhiều)	Trung bình (phải xử lý khung)	Cao (phải xử lý toàn bộ)
Mức độ “thông minh”	Cao	Trung bình	Thấp
Ứng dụng tiêu biểu	TelosB mote, ZigBee WSN	IoT giá rẻ, DIY projects	Các WSN nghiên cứu tần số thấp

# Nguồn cấp

- ▶ Nguồn cung cấp cho nút cảm biến có thể là pin sạc hoặc không sạc.
- ▶ Nguồn này cần đáp ứng yêu cầu về dung lượng, trọng lượng và hiệu quả sạc.
- ▶ Các công nghệ như pin mặt trời, năng lượng từ gió hay từ môi trường có thể được tích hợp vào nút cảm biến để cung cấp năng lượng.



# Các loại cảm biến

- Cảm biến thụ động là loại cảm biến không tác động vào môi trường khi đo đại lượng vật lý.
- Ví dụ: cảm biến ánh sáng, nhiệt kế.
- Cảm biến thụ động đo các đại lượng vật lý mà không thay đổi môi trường.
- Những cảm biến này rất hữu ích trong việc thu thập thông tin mà không tác động đến môi trường xung quanh.



# Các loại cảm biến

- ▶ Cảm biến chủ động phát tín hiệu để dò tìm môi trường.
- ▶ Những cảm biến này giúp thu thập dữ liệu môi trường hiệu quả hơn.
- ▶ Cảm biến chủ động phát tín hiệu để đo đạc các yếu tố môi trường. Ví dụ: cảm biến siêu âm, radar.
- ▶ Những cảm biến này có thể phát hiện các vật thể hoặc sự kiện trong môi trường bằng cách phát tín hiệu và nhận lại phản hồi.

# Cảm biến thụ động

Cảm biến	Độ chính xác	Tính hoán đổi	Tốc độ mẫu (Hz)	Khởi động (ms)	Dòng (mA)
Điện trở quang	N/A	10 %	2000	10	1.235
Nhiệt độ I2C	1 K	0.2 K	2	500	0.15
Khí áp	1.5 mbar	0.5 %	10	500	0.01
Bar. áp suất. nhiệt độ	0.8 K	0.24 K	10	500	0.01
Độ ẩm	2 %	3 %	500	500-3000	0.775
Pin nhiệt điện	3K	5 %	2000	200	0.17
Nhiệt kế	5 K	10 %	2000	10	0.126

# Cơ cấu chấp hành

- ▶ Cơ cấu chấp hành thực hiện các hành động đơn giản như đóng/mở chuyển mạch hoặc điều khiển các thiết bị vật lý.
- ▶ Các cơ cấu này có thể hoạt động mà không cần giao thức truyền thông phức tạp.
- ▶ Các cơ cấu này có thể được sử dụng để điều khiển động cơ, bóng đèn hoặc các thiết bị khác trong mạng cảm biến không dây.



# Nguồn cung cấp năng lượng

- ▶ Nguồn cung cấp năng lượng có thể là pin hoặc năng lượng từ môi trường. Năng lượng từ môi trường có thể được thu thập từ các nguồn như ánh sáng mặt trời hoặc gió.
- ▶ Việc sử dụng các nguồn năng lượng thay thế giúp giảm chi phí và tăng tuổi thọ cho các nút cảm biến trong mạng.



# Nguồn cung cấp năng lượng

Với các nút cảm biến, nguồn cung cấp là một bộ phận không thể thiếu. Nó có hai khía cạnh cơ bản:

- *thứ nhất* là năng lượng dự trữ và nguồn cung cấp theo các dạng yêu cầu;
- *thứ hai* là cố gắng bổ sung nguồn năng lượng đã tiêu thụ bằng cách “tìm kiếm” nó từ nguồn cung cấp bên ngoài nút theo thời gian.

Năng lượng dự trữ thường dùng pin. Pin AA thường chứa khoảng 2.2 – 2.5 Ah ở 1.5 V.



# Nguồn cung cấp năng lượng

## Năng lượng dự trữ: Pin

### *Pin truyền thống*

Nguồn cung cấp của nút cảm biến là pin, nó có thể không sạc lại được (pin chính), hoặc sạc lại được (pin phụ) nếu thiết bị tìm kiếm năng lượng ở trên nút.

Ở một số dạng khác, pin dự trữ năng lượng dạng điện hoá – các chất hoá học là nhân tố xác định chính trong kỹ thuật của pin.

Pin có các yêu cầu sau:

***Dung lượng:*** Chúng phải có dung lượng cao, trọng lượng nhỏ, thể tích nhỏ, giá thành thấp. Đơn vị chính là năng lượng /thể tích ( $\text{J}/\text{cm}^2$ ).

# Nguồn pin

Pin chính			
Năng lượng hoá học (J/cm <sup>2</sup> )	Pin kẽm không khí 3780	Lithium 2880	Alkaline 1200
Pin phụ			
Năng lượng hoá học (J/cm <sup>2</sup> )	Lithium 1080	NiMHd 860	NiCd 650

# Nguồn pin

## 1. Pin chính (Primary battery)

- Là loại pin **không sạc lại**, có mật độ năng lượng cao (ví dụ: pin kẽm–không khí, lithium, alkaline).
- Cung cấp **nguồn năng lượng lâu dài và ổn định** để duy trì hoạt động cơ bản của nút cảm biến: vi điều khiển, cảm biến, truyền nhận dữ liệu.
- Thường dùng trong môi trường khó thay thế pin (ví dụ: cảm biến ngoài trời, cảm biến dưới đất).

## 2. Pin phụ (Secondary battery)

- Là loại pin **sạc lại được** (Lithium-ion, NiMHd, NiCd).
- Dung lượng nhỏ hơn pin chính, nhưng có thể **tái nạp nhiều lần**.
- Pin phụ thường kết hợp với **nguồn năng lượng tái tạo** (như tấm pin mặt trời, thu năng lượng từ rung động, sóng radio, nhiệt...) để kéo dài tuổi thọ hệ thống.
- Giúp nút cảm biến hoạt động **ổn định** ↓ / **cả khi pin chính yếu dần** hoặc trong giai đoạn cần dòng điện cao (phát sóng truyền thông, xử lý dữ liệu).

# Pin và dung lượng

- ▶ Pin trong nút cảm biến cần phải có dung lượng lớn và hiệu quả sạc tốt. Pin có thể phục hồi dung lượng khi không sử dụng, giúp kéo dài tuổi thọ.
- ▶ Các yêu cầu về pin: dung lượng cao, trọng lượng nhẹ và khả năng sạc hiệu quả.
- ▶ Pin cần được tối ưu hóa để đáp ứng nhu cầu sử dụng năng lượng của nút cảm biến trong mạng.

# Tiết kiệm năng lượng

- ▶ **Những thành phần “ngốn” pin nhiều nhất**
  - **Bộ vi điều khiển (MCU):** xử lý dữ liệu.
  - **Bộ thu phát không dây (radio transceiver):** gửi/nhận dữ liệu.
  - **Bộ nhớ:** lưu dữ liệu, tuy ít nhưng cũng cần năng lượng.
  - **Cảm biến:** phụ thuộc loại (cảm biến hình ảnh, âm thanh tiêu tốn nhiều hơn so với cảm biến nhiệt độ).

# Tiết kiệm năng lượng

## ► Cách giảm tiêu thụ năng lượng

1. **Thiết kế phần cứng tiết kiệm điện:** dùng chip công suất thấp.

👉 Nhưng chỉ phần nào thôi, nếu sử dụng sai vẫn lãng phí.

2. **Tận dụng trạng thái “ngủ/nghỉ”:**

- Phần lớn thời gian nút cảm biến... chẳng làm gì cả .
- Vậy thay vì để nó bật hoài → cho nó **ngủ**.
- Khi cần (có tín hiệu từ ngoài hoặc đến lịch đo dữ liệu) → nó **thức dậy**.

# Tiết kiệm năng lượng

- ▶ **Các trạng thái năng lượng (giống như máy tính/laptop)**
  - **Hoạt động (Active):** làm việc đầy đủ.
  - **Nghỉ (Idle/Standby):** vẫn bật nhưng giảm hoạt động, sẵn sàng bật nhanh.
  - **Ngủ (Sleep/Deep Sleep):** tắt nhiều chức năng, tiết kiệm điện hơn, nhưng cần nhiều thời gian và năng lượng để bật lại.

# Tiết kiệm năng lượng

## ► Ý tưởng chính

- Bộ vi điều khiển (MCU) trong nút cảm biến **không phải lúc nào cũng chạy hết công suất.**
- Nó có nhiều **trạng thái hoạt động rời rạc** (Active, Idle, Sleep...), mỗi trạng thái dùng **mức năng lượng khác nhau.**
- Nhờ vậy, có thể **giảm đáng kể năng lượng tiêu thụ**, kéo dài tuổi thọ pin.



# Tiết kiệm năng lượng

## ▶ Hình dung dễ hiểu

- Hãy tưởng tượng MCU giống như **đèn trong phòng**:
  - **Bật sáng hết cỡ (Active)**: dùng nhiều điện.
  - **Chế độ nghỉ (Idle)**: giảm ánh sáng, tiết kiệm nhưng vẫn sẵn sàng.
  - **Chế độ ngủ (Sleep)**: tắt gần hết, chỉ để đèn ngủ nhỏ xíu (đồng hồ).
- Tùy nhu cầu, ta chọn mức sáng phù hợp → vừa đủ dùng, vừa tiết kiệm pin.

# Tiết kiệm năng lượng

- ▶ 🔧 Ví dụ minh họa theo từng loại chip
- ▶ 1. Intel StrongARM
  - **Bình thường (Normal):** chạy full → tiêu thụ ~ **400 mW**.
  - **Nghỉ (Idle):** CPU clock dừng, chỉ ngoại vi hoạt động → tiêu thụ ~ **100 mW**.
  - **Ngủ (Sleep):** chỉ đồng hồ thời gian thực chạy, chờ ngắt đánh thức → tiêu thụ cực thấp ~ **50  $\mu$ W**.
    - 👉 Nghĩa là nếu không cần xử lý thì “cho CPU đi ngủ”, chỉ để đồng hồ báo thức.

# Tiết kiệm năng lượng

## ► 2. Texas Instruments MSP430

- **Hoạt động (Active):** ~ 1.2 mW ở 1 MHz, 3V.
  - **Có 4 chế độ ngủ (Low Power Mode – LPM).**
    - **LPM4 (ngủ sâu nhất):** chỉ thức khi có tín hiệu bên ngoài, tiêu thụ ~ 0.3  $\mu$ W.
    - **LPM3:** clock vẫn chạy để tự hẹn giờ, tiêu thụ ~ 6  $\mu$ W.
- 👉 MCU này nổi tiếng vì siêu tiết kiệm điện, rất hợp cho cảm biến chạy pin nhỏ.

# Tiết kiệm năng lượng

## ► 3. Atmel ATmega 128L

- Có 6 chế độ năng lượng.
- Trạng thái hoạt động/nhàn rỗi: tiêu thụ ~ **6–15 mW**.
- Chế độ “power-down” (tắt nguồn sâu): chỉ còn ~ **75  $\mu$ W**.  
👉 Linh hoạt: vừa có chế độ tiết kiệm điện vừa có tốc độ khởi động lại nhanh.

# Các ví dụ của nút cảm biến

- ▶ Mica Mote: Dùng vi điều khiển Atmel, hỗ trợ nhiều cảm biến và mở rộng dễ dàng cho các ứng dụng WSN.



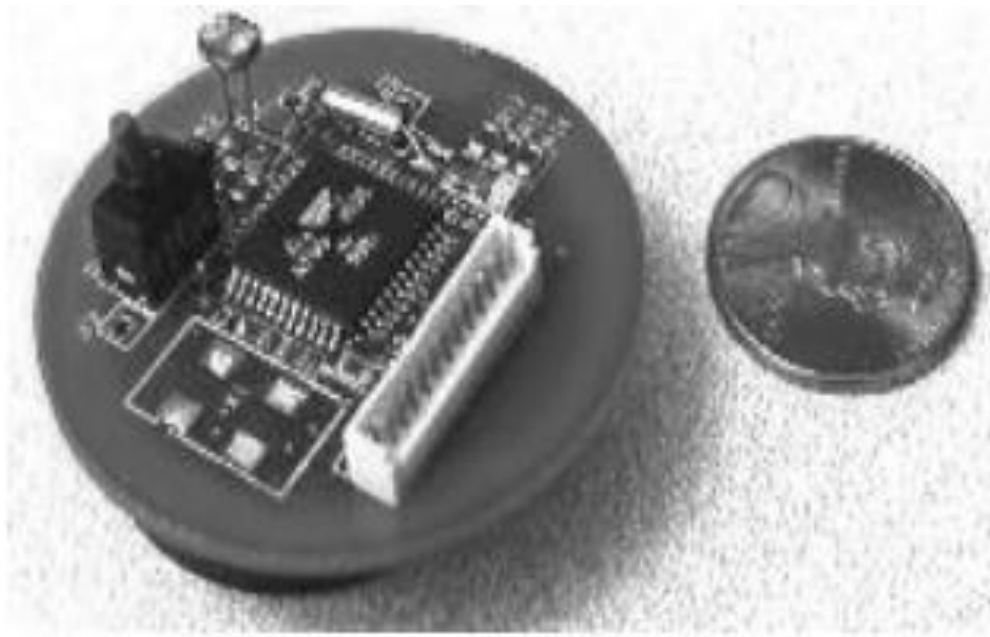
**Mica Mote** (hay **Mica2/MicaZ mote**) là một loại **node cảm biến không dây** (wireless sensor node) được phát triển bởi Crossbow (Mỹ), thường dùng trong nghiên cứu và triển khai mạng cảm biến không dây (**WSN – Wireless Sensor Network**).

•**Phần cứng chính:**

- Vi điều khiển **Atmel ATmega128L** (8-bit RISC, tần số ~8 MHz).
- Bộ nhớ Flash khoảng 128 KB, RAM 4 KB.
- Giao tiếp **radio (RF)** dùng chip thu phát tần số vô tuyến (thường 433 MHz, 868 MHz hoặc 2.4 GHz).
- Hỗ trợ nhiều loại cảm biến (nhiệt độ, ánh sáng, độ ẩm, chuyển động...).
- Chạy bằng pin, nhỏ gọn, dễ triển khai thành mạng lớn.

# Các ví dụ của nút cảm biến

- ▶ Mica Mote: Dùng vi điều khiển Atmel, hỗ trợ nhiều cảm biến và mở rộng dễ dàng cho các ứng dụng WSN.



# Các ví dụ của nút cảm biến

- ▶ EYES: Sử dụng MSP430 của Texas Instruments, hỗ trợ báo cường độ tín hiệu và kết nối USB.

**EYES mote** (còn gọi là **Eyes node**) là một loại **node cảm biến không dây** được phát triển ở châu Âu trong dự án **EYES (Energy Efficient Sensor Networks)**.

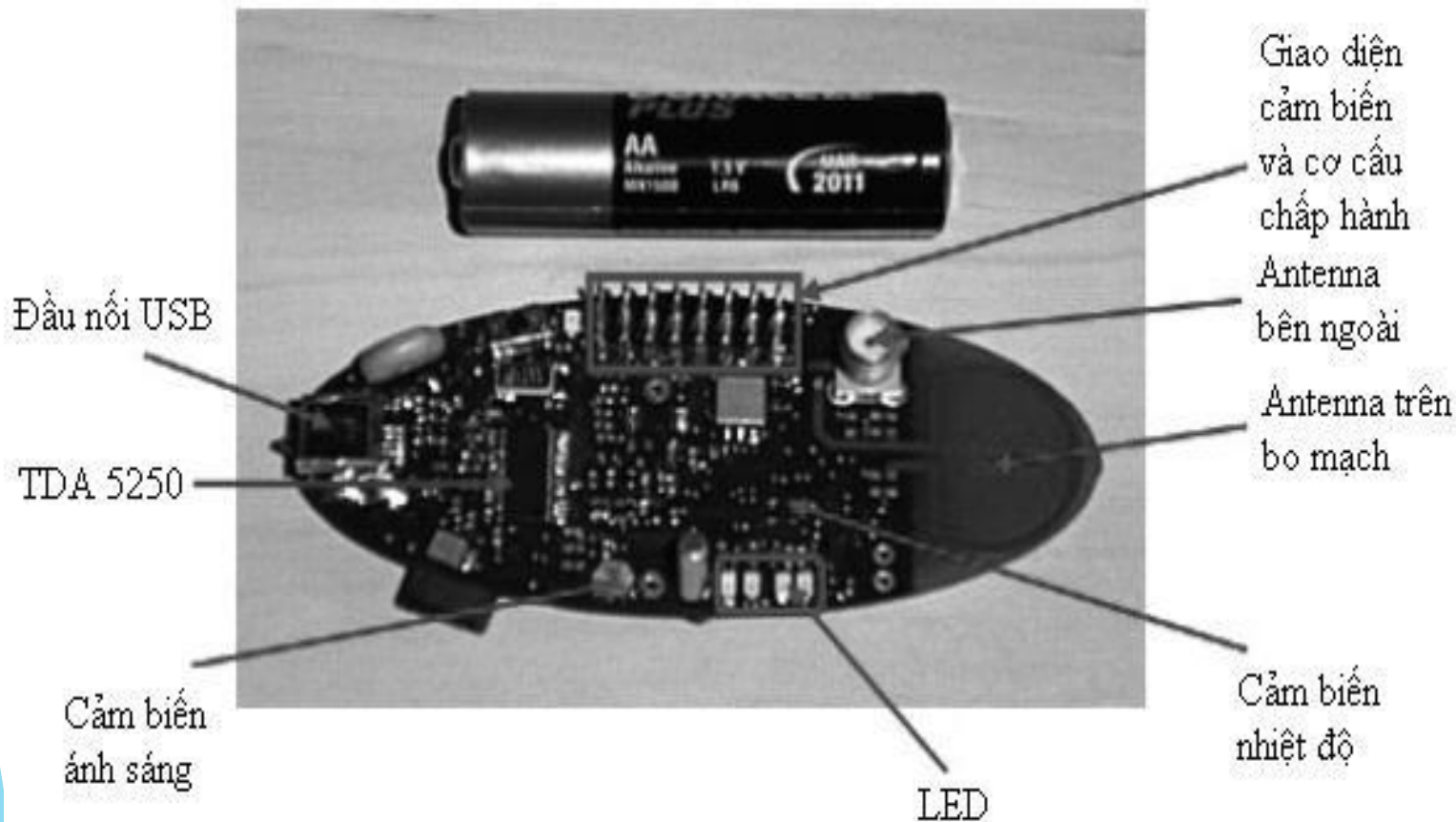
• **Bộ vi điều khiển:** dùng **MSP430** của **Texas Instruments**. Đây là dòng vi điều khiển nổi tiếng **tiết kiệm năng lượng**, rất phù hợp cho mạng cảm biến không dây (WSN).

• **Đặc điểm chính:**

- MSP430 là vi điều khiển 16-bit, hoạt động với mức tiêu thụ điện năng cực thấp.
- Có nhiều chế độ **low-power mode**, cho phép node cảm biến "ngủ đông" khi không cần thiết, kéo dài tuổi thọ pin.
- Tích hợp ADC (chuyển đổi tín hiệu tương tự—số) để đọc dữ liệu cảm biến.

# Các ví dụ của nút cảm biến

- ▶ EYES: Sử dụng MSP430 của Texas Instruments, hỗ trợ báo cường độ tín hiệu và kết nối USB.





# Các ví dụ của nút cảm biến

- ▶ Btnode: Sử dụng Bluetooth và có bộ nhớ lớn.



**BTnode** (Bluetooth Node) là một loại **node cảm biến không dây** được phát triển tại ETH Zurich (Thụy Sĩ).

## •Đặc điểm chính:

- Sử dụng **Bluetooth** làm phương thức truyền thông thay vì sóng radio chuyên dụng như Mica mote hay TelosB.
- Có **bộ nhớ lớn** hơn nhiều mote thế hệ đầu, thuận lợi cho việc lập trình, lưu trữ dữ liệu cảm biến, hoặc chạy các thuật toán phức tạp.
- Kiến trúc phần cứng thường có:
  - **Vi điều khiển Atmel ATmega128L** (giống Mica mote).
  - Module **Bluetooth Class 2** (tầm khoảng 10m, tốc độ cao hơn so với radio CC1000/CC2420).
  - Nhiều bộ nhớ Flash và RAM hơn để phục vụ ứng dụng cần xử lý nhiều.

# Các ví dụ của nút cảm biến

- ▶ Btnode: Sử dụng Bluetooth và có bộ nhớ lớn.
- ▶



# Các ví dụ của nút cảm biến

## Giới thiệu TelosB

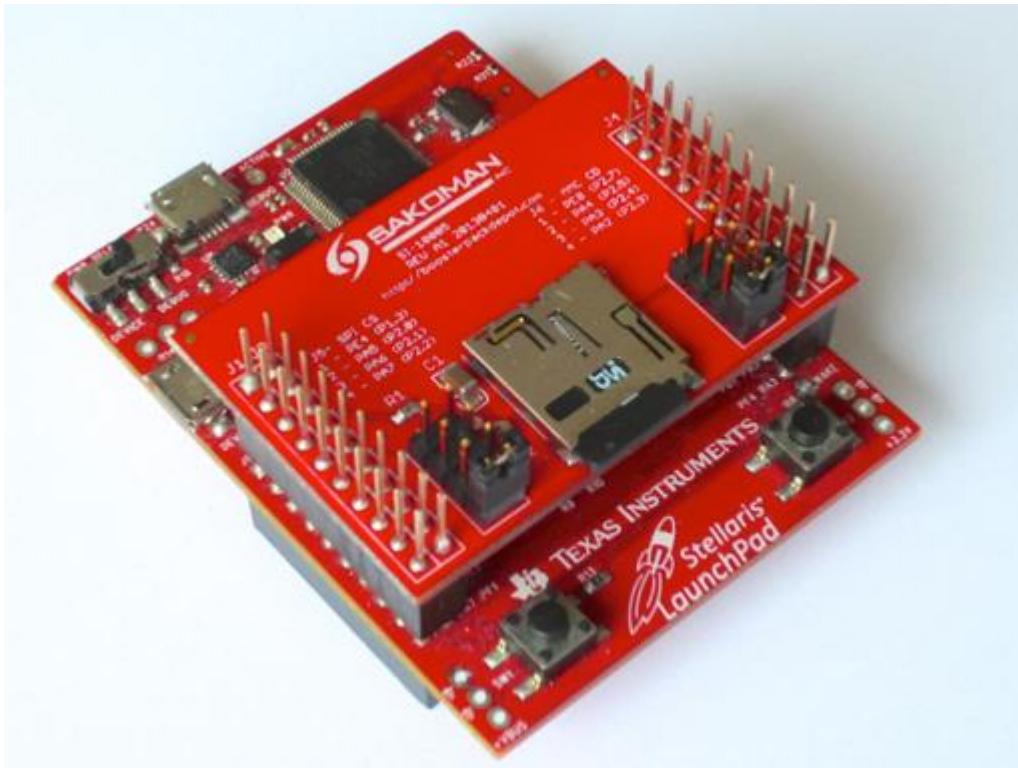
- **TelosB** là một **nút cảm biến không dây (WSN mote)** được phát triển bởi **UC Berkeley**, sau này do **Crossbow** sản xuất.
- TelosB là một trong những **nền tảng chuẩn mực IoT, WSN**.

## Ứng dụng thực tế

- Nghiên cứu và giảng dạy về **mạng cảm biến không dây (WSN)**.
- Triển khai thử nghiệm **mạng IoT trong nhà thông minh (smart home)**.
- Các ứng dụng giám sát môi trường: nhiệt độ, độ ẩm, ánh sáng, nông nghiệp thông minh.
- Hệ thống cảnh báo cháy, an ninh, theo dõi tòa nhà.

**TelosB = mote chuẩn mực của WSN hiện đại,**

- Vi điều khiển MSP430 (tiết kiệm năng lượng),
  - Radio IEEE 802.15.4 (chuẩn ZigBee),
  - Cảm biến tích hợp sẵn,
  - Kết nối USB tiện lợi,
- rất phù hợp cho nghiên cứu & ứng dụng IoT.



Đặc điểm	Mica Mote	EYES mote	BTnode	TelosB
Vi điều khiển	Atmel ATmega128L (8-bit, 8 MHz)	MSP430 (TI, 16-bit, siêu tiết kiệm năng lượng)	Atmel ATmega128L + Bluetooth	TI MSP430 (16-bit, tiết kiệm năng lượng)
Cảm biến	Hỗ trợ nhiều loại: nhiệt độ, ánh sáng, độ ẩm, gia tốc (qua board mở rộng)	Tích hợp cơ bản, hỗ trợ mở rộng; tập trung nghiên cứu tiết kiệm năng lượng	Hỗ trợ mở rộng, bộ nhớ lớn để lưu dữ liệu	Tích hợp cảm biến ánh sáng, nhiệt độ, độ ẩm, onboard USB
Khả năng kết nối	Radio CC1000/CC2420 (433/868 MHz hoặc 2.4 GHz)	Radio tiết kiệm năng lượng cho WSN lâu dài	Bluetooth Class 2 (~10m), tốc độ cao, dễ kết nối laptop/điện thoại	IEEE 802.15.4 (2.4 GHz), phù hợp chuẩn ZigBee/WSN
Ưu điểm nổi bật	Phổ biến, dễ lập trình, dùng rộng rãi trong nghiên cứu WSN	Siêu tiết kiệm năng lượng, phù hợp triển khai mạng lâu dài	Bộ nhớ lớn, dễ tích hợp thiết bị di động, tốc độ truyền cao	Tích hợp sẵn cảm biến, USB, chuẩn ZigBee, nghiên cứu phổ biến
Hạn chế	Năng lượng hạn chế, pin ngắn hơn EYES	Tài nguyên tính toán hạn chế so với BTnode	Bluetooth tiêu thụ nhiều năng lượng, phạm vi ngắn	RAM và Flash hạn chế so với mote đời mới