

GIÁO TRÌNH

LẬP TRÌNH HỆ THỐNG NHÚNG

Trường Cao Đẳng Lý Tự Trọng TP.HCM
Giảng viên : Trần Việt Khánh



Nội dung môn học



Kiến thức:

- ❖ Trình bày các khái niệm về hệ thống nhúng
- ❖ Trình bày và xây dựng được các ứng dụng nhúng chạy trên các hệ thống khác nhau.
- ❖ Nhận biết và so sánh được các ứng dụng nhúng trên các hệ thống khác nhau.

Kỹ năng:

- ❖ Xây dựng và cài đặt được các ứng dụng nhúng cho từng hệ thống khác nhau.
- ❖ Sử dụng được các kiến thức về lập trình nhúng để tạo ra các ứng dụng chạy trên từng hệ thống cụ thể.



Phương pháp học - Kiểm tra & đánh giá



➤ **Phương pháp học**

- ❖ Cần chú ý nghe giảng, rèn luyện khả năng tự nghiên cứu, tự học, rèn luyện kỹ năng thực hành một cách nghiêm túc và tỉ mỉ
- ❖ Sử dụng các công cụ mô phỏng để xây dựng và cài đặt một ứng dụng nhúng chạy trên một hệ thống cụ thể.
- ❖ Mô phỏng cách tạo ra một ứng dụng nhúng theo một hệ thống cụ thể nào đó, giúp cho người học dễ dàng tiếp thu kiến thức. Từ đó người học có thể tự học và tự xây dựng được các ứng dụng nhúng trên các hệ thống khác nhau.

➤ **Kiểm tra và đánh giá**

- ❖ Kiểm tra thường xuyên: Thực hành/Tự luận/ Bài tập – Điểm đánh giá: 20%
- ❖ Kiểm tra giữa kỳ: Lý thuyết/Thực hành – Điểm đánh giá: 30%
- ❖ Kiểm tra cuối kỳ: Thi thực hành trên máy tính – Điểm đánh giá: 50%



Chương 1: Giới thiệu về hệ thống nhúng



1 Khái niệm hệ thống nhúng

2 Lịch sử hệ thống nhúng

3 Đặc điểm hệ thống nhúng

4 Xu thế phát triển các hệ thống nhúng



Chương 1: Giới thiệu về hệ thống nhúng

1. Khái niệm hệ thống nhúng



- **Hệ thống nhúng (Embedded system):** là một thuật ngữ để chỉ một hệ thống có khả năng tự trị được (tự động điều khiển) được nhúng vào trong một môi trường hay một hệ thống mẹ.
- Là các hệ thống tích hợp cả phần cứng và phần mềm để thực hiện một hoặc một nhóm chức năng chuyên biệt cụ thể.
- Khác với các máy tính đa chức năng, hệ thống nhúng chỉ thực hiện một hoặc một vài chức năng nhất định, thường đi kèm với những yêu cầu cụ thể và bao gồm một số thiết bị máy móc và phần cứng chuyên dụng mà ta không tìm thấy trong một máy tính đa năng nói chung. Vì hệ thống chỉ được xây dựng cho một số nhiệm vụ nhất định nên các nhà thiết kế có thể tối ưu hóa nó nhằm giảm thiểu kích thước và chi phí sản xuất.



Chương 1: Giới thiệu về hệ thống nhúng

1. Khái niệm hệ thống nhúng



- Các hệ thống nhúng thường được sản xuất hàng loạt với số lượng lớn. Hệ thống nhúng rất đa dạng, phong phú về chủng loại.
- Đó có thể là những thiết bị cầm tay nhỏ gọn như đồng hồ kỹ thuật số, máy chơi nhạc MP3, hoặc những sản phẩm lớn như đèn giao thông, bộ kiểm soát và điều khiển trong nhà máy hoặc hệ thống kiểm soát các máy năng lượng hạt nhân, v.v...



Chương 1: Giới thiệu về hệ thống nhúng

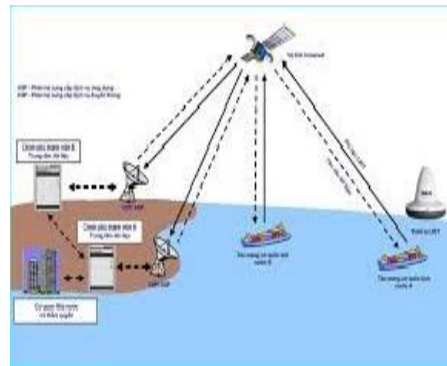
1. Khái niệm hệ thống nhúng



- Một số thiết bị hệ thống nhúng



Industrial Robots



GPS Receivers



Digital Cameras



MP3 Players



Wireless Routers



Gaming Controllers



Photocopiers



Set – top boxes

Chương 1: Giới thiệu về hệ thống nhúng

2. Lịch sử các hệ thống nhúng



- Hệ thống nhúng đầu tiên là *Apollo Guidance Computer* (Máy tính dẫn đường Apollo) được phát triển bởi *Charles Stark Draper* tại phòng thí nghiệm của trường đại học *MIT* năm 1960.
- Vào giữa thập niên 80, kỹ thuật mạch tích hợp đã đạt đến trình độ cao dẫn đến nhiều thành phần có thể đưa vào một chip xử lý. Các bộ vi xử lý được gọi là các vi điều khiển. Với giá cả thấp, các vi điều khiển đã trở nên rất hấp dẫn để xây dựng các hệ thống nhúng chuyên dụng cho tất cả các lĩnh vực trên thị trường.
- Vào cuối những năm 80, các hệ thống nhúng đã trở nên phổ biến trong hầu hết các thiết bị điện tử, thiết bị gia dụng và khuynh hướng này vẫn còn tiếp tục phát triển cho đến ngày nay.



Chương 1: Giới thiệu về hệ thống nhúng

3. Đặc điểm của các hệ thống nhúng



❖ Giao diện

- Các hệ thống nhúng có thể không có giao diện (đối với những hệ thống đơn nhiệm) hoặc có đầy đủ giao diện giao tiếp với người dùng tương tự như các hệ điều hành trong các máy tính để bàn.
- Đối với các hệ thống đơn giản, thiết bị nhúng sử dụng nút bấm, đèn LED và hiển thị chữ cỡ nhỏ hoặc chỉ hiển thị số, thường giao diện là một hệ thống menu đơn giản.
- Còn trong một hệ thống nhúng phức tạp hơn, giao diện có một màn hình đồ họa, cảm ứng hoặc có các nút bấm ở lề màn hình.



Máy chơi game joystick button

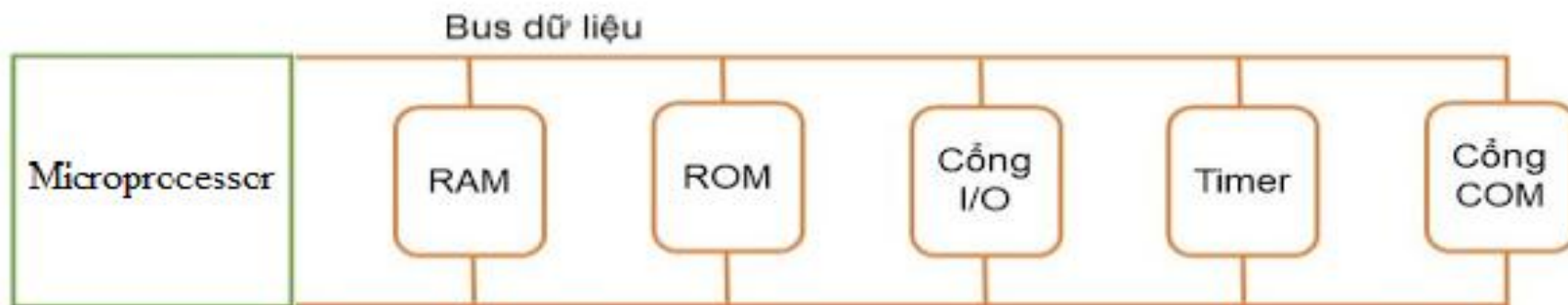
Chương 1: Giới thiệu về hệ thống nhúng

3. Đặc điểm của các hệ thống nhúng



❖ Bộ vi xử lý trong các hệ thống nhúng

- **Vi điều khiển:** là một hệ thống bao gồm một chip xử lý (Microprocessor) có hiệu suất đủ dùng và giá thành thấp kết hợp với các thiết bị ngoại vi như bộ nhớ, các thiết bị nhập / xuất, ...



Sơ đồ đơn giản của một khối vi điều khiển



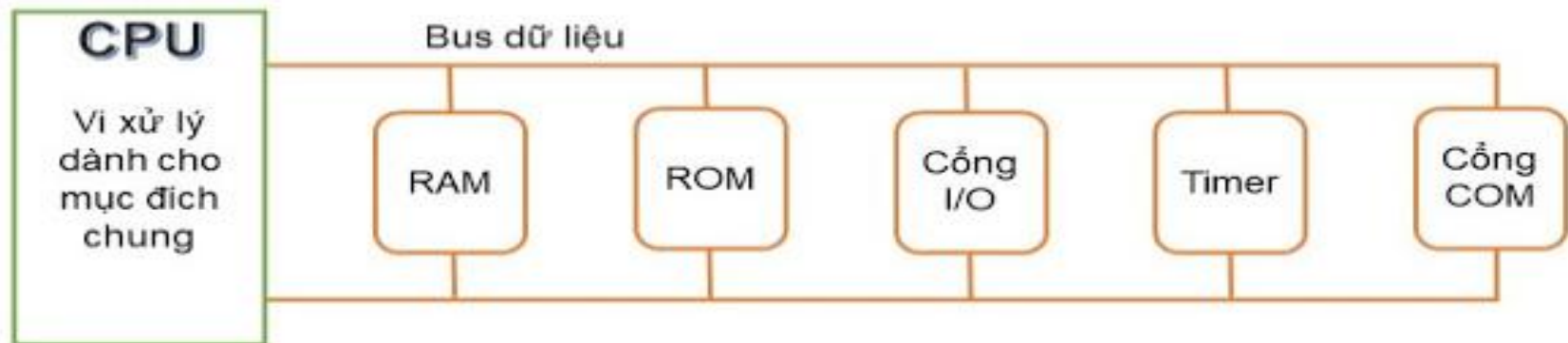
Chương 1: Giới thiệu về hệ thống nhúng

3. Đặc điểm của các hệ thống nhúng



❖ Bộ vi xử lý trong các hệ thống nhúng

- **Vi xử lý:** là một chip đơn có CPU. Ngoài các thiết bị ngoại vi, nó còn có những đơn vị điều khiển khác như: đơn vị xử lý toán học, đơn vị xử lý thời gian,... giúp xử lý các câu lệnh nhanh chóng hơn.



Sơ đồ khối đơn giản của một Vi xử lý



Chương 1: Giới thiệu về hệ thống nhúng

So sánh Vi xử lý và Vi điều khiển



Vi xử lý

- Đa nhiệm. Có thể thực thi nhiều nhiệm vụ. Ví dụ như máy tính có thể vừa phát nhạc vừa soạn văn bản.
- RAM, ROM, các cổng nhập / xuất có thể được thêm từ bên ngoài và có thể thay đổi số lượng.
- Người thiết kế chỉ mua sẵn, không quyết định được thành phần mà tối thiểu mình cần.
- Hệ thống sử dụng Vi xử lý sẽ xử lý nặng hơn và tốn kém chi phí hơn.
- Các thiết bị gắn ngoài đòi hỏi nhiều không gian và tiêu tốn nhiều năng lượng hơn.

Vi điều khiển

- Đơn nhiệm. Chỉ thực hiện một nhiệm vụ. Ví dụ như máy giặt chỉ giặt đồ không phát video giải trí được.
- RAM, ROM, các cổng nhập / xuất được nhúng vào chip, không thể thêm vào hoặc thay đổi số lượng của chúng.
- Số lượng cố định của các đơn vị thành phần, giúp cho Vi điều khiển thật lý tưởng để thực hiện các công việc đã xác định.
- Vi điều khiển xử lý nhẹ hơn và chi phí rẻ hơn so với Vi xử lý.
- Hệ thống sử dụng Vi điều khiển tiêu tốn ít năng lượng và chiếm ít không gian hơn.

Chương 1: Giới thiệu về hệ thống nhúng

3. Đặc điểm của các hệ thống nhúng



❖ Các thiết bị ngoại vi

- Hệ thống nhúng giao tiếp với các hệ bên ngoài thông qua các thiết bị ngoại vi, ví dụ như:
- Serial Communication Interfaces (SCI): RS-232, RS-422, RS-485...
- Synchronous Serial Communication Interface: I2C, JTAG, SPI, SSC và ESSI
- Universal Serial Bus (USB)
- Networks: Wireless, Internet,...



Chương 1: Giới thiệu về hệ thống nhúng

3. Đặc điểm của các hệ thống nhúng



❖ Các công cụ phát triển

- Tương tự như các sản phẩm phần mềm khác, phần mềm hệ thống nhúng cũng được phát triển nhờ việc sử dụng các trình biên dịch (compilers), chương trình dịch hợp ngữ (assembler) hoặc các công cụ gỡ rối (debuggers).
- Bộ công cụ lập trình: Keil C, AVR Studio.
- Bộ công cụ mô phỏng giả lập hệ thống nhúng: Proteus.



Chương 1: Giới thiệu về hệ thống nhúng

3. Đặc điểm của các hệ thống nhúng



❖ Các đặc điểm khác

- Các hệ thống nhúng thường nằm trong các cỗ máy được kỳ vọng là sẽ chạy hàng năm trời liên tục mà không bị lỗi hoặc có thể khôi phục hệ thống khi gặp lỗi. Vì thế, các phần mềm hệ thống nhúng được phát triển và kiểm thử một cách cẩn thận hơn là phần mềm cho máy tính cá nhân.
- Hệ thống nhúng không thể ngừng để sửa chữa một cách an toàn, ví dụ như ở các hệ thống không gian, hệ thống dây cáp dưới đáy biển, các đèn hiệu dẫn đường,... Giải pháp đưa ra là chuyển sang sử dụng các hệ thống con dự trữ hoặc các phần mềm cung cấp một phần chức năng.
- Hệ thống nhúng phải được chạy liên tục vì tính an toàn, ví dụ như các thiết bị dẫn đường máy bay, thiết bị kiểm soát độ an toàn trong các nhà máy hóa chất,... Vì vậy nếu hệ thống gặp sự cố thì giải pháp đưa ra là lựa chọn backup hệ thống.
- Nếu hệ thống nhúng ngừng hoạt động sẽ gây tổn thất rất nhiều tiền của ví dụ như các dịch vụ buôn bán tự động, hệ thống chuyển tiền, hệ thống kiểm soát trong các nhà máy ...



Chương 1: Giới thiệu về hệ thống nhúng

4. Xu thế phát triển các hệ thống nhúng



❖ Xu thế phát triển các hệ thống nhúng ngày nay:

- Các hệ thống nhúng ngày càng phức tạp hơn đáp ứng các yêu cầu khắt khe về thời gian thực, tiêu tốn ít năng lượng và hoạt động tin cậy ổn định hơn.
- Các hệ thống nhúng ngày càng có độ mềm dẻo cao đáp ứng các yêu cầu nhanh chóng đưa sản phẩm ra thương trường, có khả năng bảo trì từ xa, có tính cá nhân cao.
- Các hệ thống nhúng ngày càng có khả năng hội thoại cao, có khả năng kết nối mạng và hội thoại với người sử dụng.
- Các hệ nhúng ngày càng có khả năng tiếp nhận năng lượng từ nhiều nguồn khác nhau (ánh sáng, rung động, điện từ trường,...) để tạo nên các hệ thống tự động tiếp nhận năng lượng trong quá trình hoạt động.



Chương 1: Giới thiệu về hệ thống nhúng

3. Câu hỏi & bài tập



- Demo

- Video cài đặt phần mềm mô phỏng hệ thống nhúng
- Video cài đặt IDE Keil C lập trình điều khiển hệ thống nhúng



Chương 2: Hệ điều hành trên các hệ thống nhúng



1 Các yêu cầu chung của hệ điều hành nhúng

2 Bộ nạp khởi tạo (Boot – Loader)

3 Hệ điều hành thời gian thực

4 Các hệ điều hành nhúng

5 Hệ điều hành Windows CE

6 Hệ điều hành Embedded Linux, hệ điều hành Android

C2: Hệ điều hành trên các hệ thống nhúng



1. Các yêu cầu chung của hệ điều hành nhúng

- Tại sao cần có một hệ điều hành cho các hệ thống nhúng ?
 - Để phát triển một hệ thống nhúng lập trình viên cần phải hiểu rất rõ về phần cứng của từng loại hệ thống, cụ thể là cần phải nắm rõ kiến trúc và tập lệnh của bộ vi điều khiển đang sử dụng, các cổng giao tiếp với các thiết bị ngoại vi, các ngõ điều khiển in/out của bộ vi điều khiển.
 - Để hệ thống nhúng hoạt động được lập trình viên phải viết code để điều khiển tất cả các vấn đề trên, rồi biên dịch ra file hex và nạp xuống hệ thống.
 - Phần mềm nhúng ra đời giúp cho lập trình viên có thể dễ dàng phát triển các ứng dụng và bớt lệ thuộc vào phần cứng, đồng thời quản lý, điều phối và đồng bộ các tiến trình trong hệ thống giúp hệ thống chạy an toàn và chính xác.
 - Phần mềm nhúng như vậy được gọi là hệ điều hành nhúng.



C2: Hệ điều hành trên các hệ thống nhúng



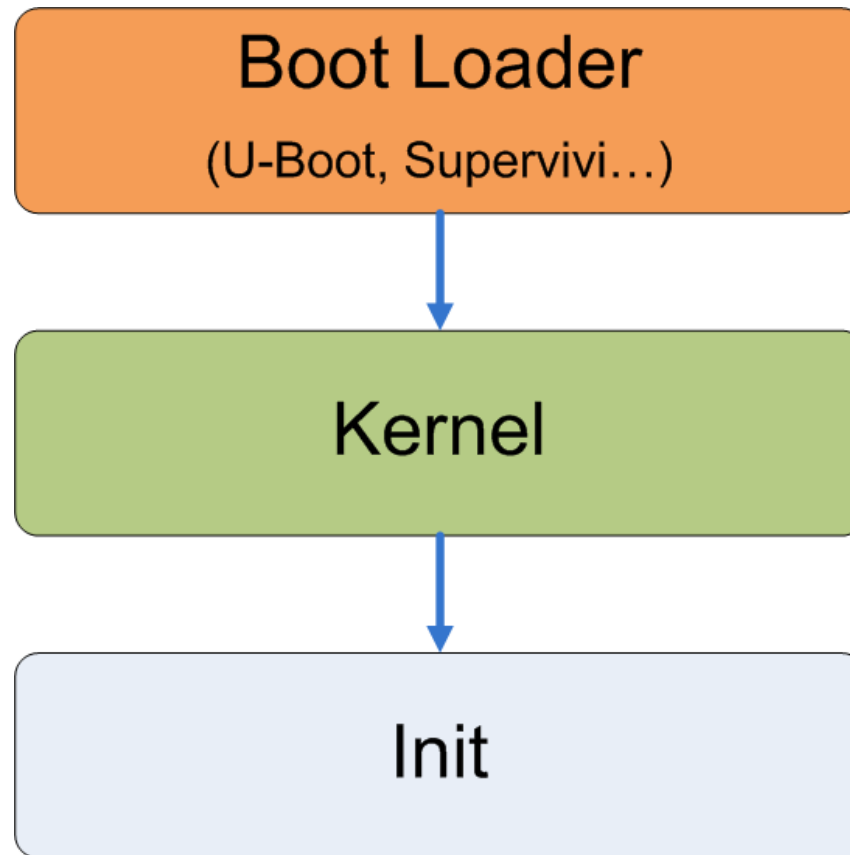
1. Các yêu cầu chung của hệ điều hành nhúng

- ❖ Khái niệm hệ điều hành nhúng (Embedded OS): là hệ điều hành cài đặt cho các hệ thống nhúng (Embedded System) đòi hỏi có các yêu cầu sau:
 - Việc thiết kế và phát triển hệ điều hành nhúng phải đảm bảo:
 - Compact.
 - Efficient.
 - Reliable.
 - Khả năng tương thích mềm: dễ dàng nâng cấp hay thu gọn để tương thích với nền tảng của hệ thống.
 - Thu gọn, đòi hỏi ít bộ nhớ hơn. Có thể hỗ trợ khởi động từ bộ nhớ ROM, Flash (những hệ thống không có ổ đĩa cứng).
 - Cung cấp các cơ chế lập lịch (scheduler) với sự hỗ trợ của hệ điều hành thời gian thực (Realtime Operating System – RTOS)



C2: Hệ điều hành trên các hệ thống nhúng

2. Bộ nạp khởi tạo (Boot-loader)



C2: Hệ điều hành trên các hệ thống nhúng

2. Bộ nạp khởi tạo (Boot-loader)

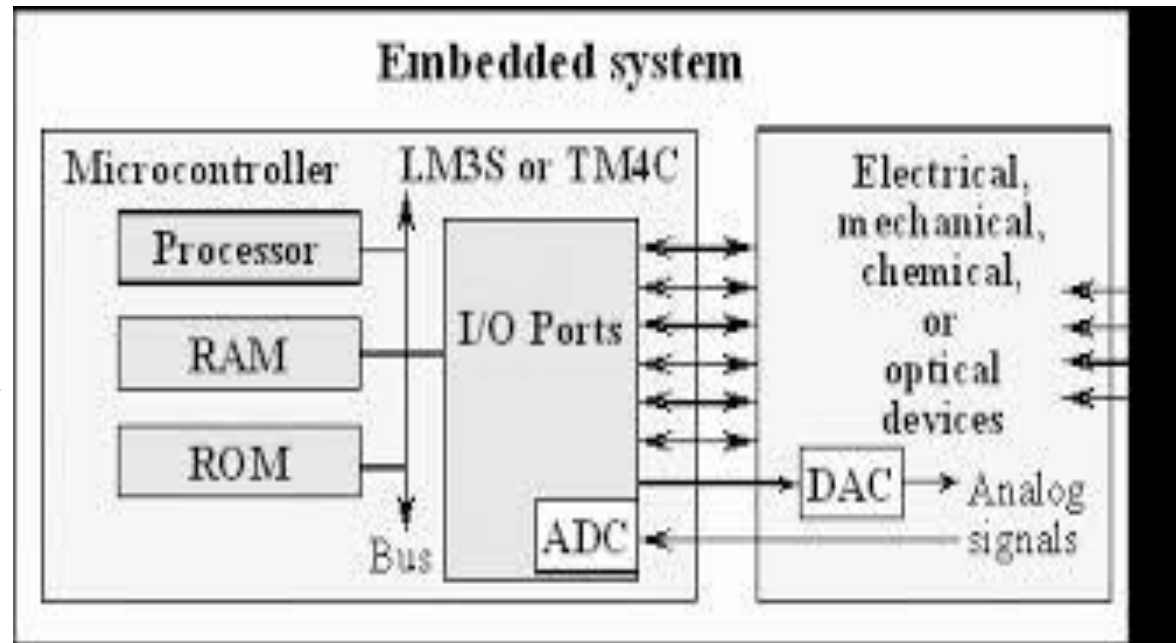
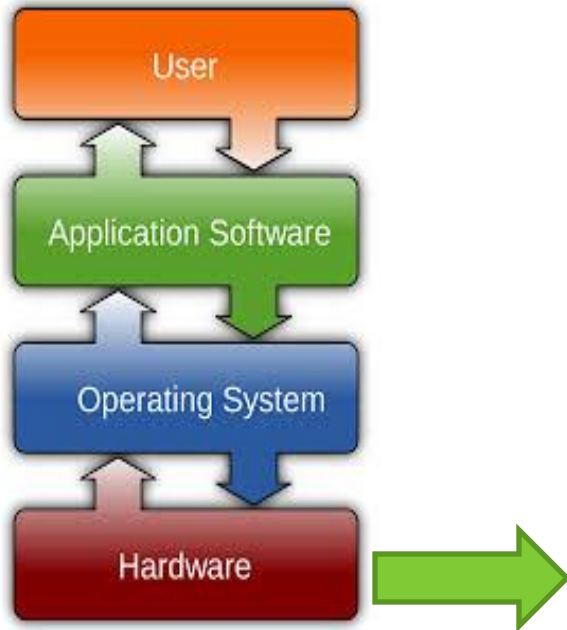


- **Boot loader:** chương trình mồi, thực hiện kiểm tra phần cứng hệ thống và nạp nhân (kernel) của hệ điều hành. Nói cách khác là nạp hệ điều hành vào hệ thống nhúng để điều khiển các hoạt động.
- **Kernel:** nhân hệ điều hành, chứa các thành phần cơ bản nhất của hệ điều hành.
- **Root file system:** hệ thống file, chứa các modules bổ sung và các phần mềm ứng dụng.



C2: Hệ điều hành trên các hệ thống nhúng

2. Bộ nạp khởi tạo (Boot-loader)



Hình ảnh hệ điều hành được nạp vào hệ thống nhúng để điều khiển các hoạt động



C2: Hệ điều hành trên các hệ thống nhúng

3. Hệ điều hành thời gian thực

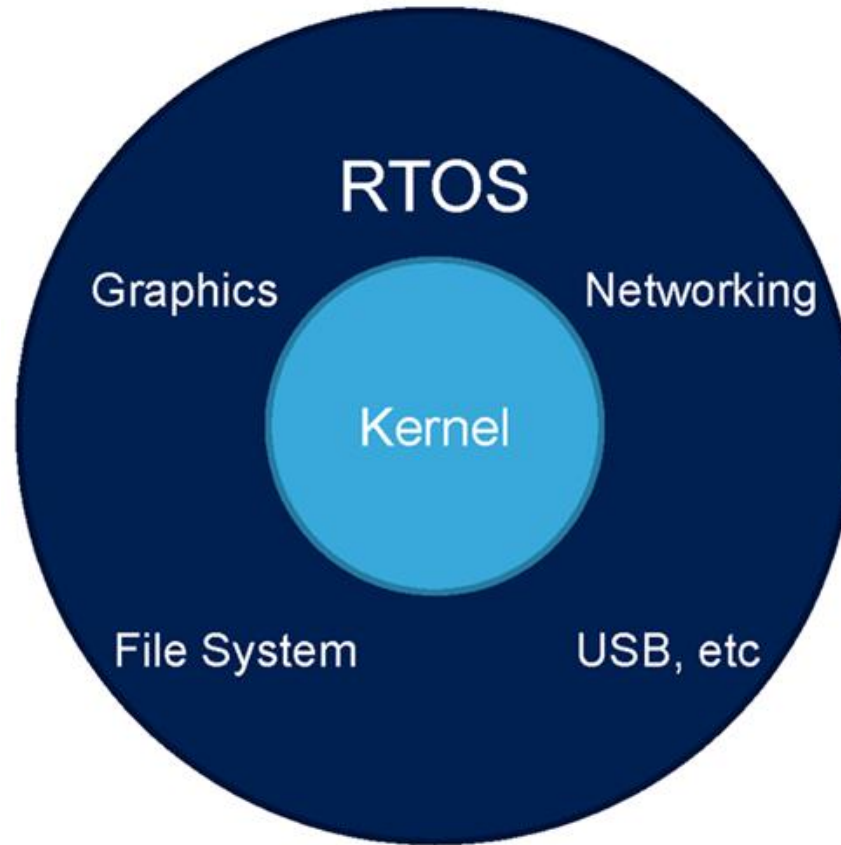


- Một hệ thống có khả năng thực hiện thời gian thực nghĩa là hệ thống đó phải có hệ điều hành thực hiện các chức năng của mình trong một khoảng thời gian xác định và nhỏ nhất có thể chấp nhận được. Khi đáp ứng được yêu cầu này, hệ thống đó được gọi là hệ thống thời gian thực (RTOS – Realtime Operating System).
- Có 2 loại thời gian thực : thời gian thực cứng và thời gian thực mềm.
 - ☐ Đối với hệ thống thời gian thực cứng, tất cả các chức năng của nó phải được thực thi chính xác trong một khoảng thời gian xác định, nếu không cả hệ thống sẽ bị lỗi nghiêm trọng. Ví dụ : hệ thống điều khiển không lưu, hệ thống dẫn đường tên lửa, thiết bị y tế, . . .
 - ☐ Đối với hệ thống thời gian thực mềm, các chức năng phải được thực hiện trong một khoảng thời gian xác định nhỏ nhất nhưng không bắt buộc.



C2: Hệ điều hành trên các hệ thống nhúng

3. Hệ điều hành thời gian thực



Hệ thống thời gian thực

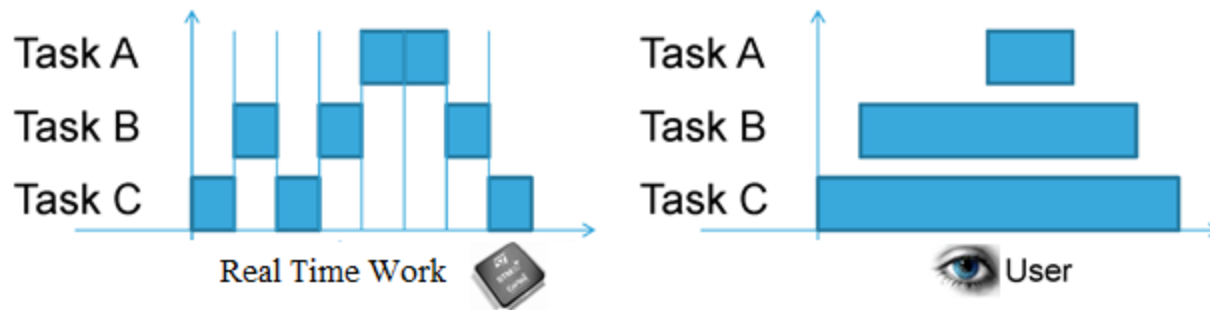


C2: Hệ điều hành trên các hệ thống nhúng

3. Hệ điều hành thời gian thực



- ❖ Kernel sẽ có nhiệm vụ quản lý nhiều task (công việc) cùng chạy 1 lúc, mỗi task thường chạy mất vài ms. Tại lúc kết thúc mỗi task - Kernel thường:
 - Lưu trạng thái task.
 - Thanh ghi sẽ load trạng thái của task tiếp theo.
 - Task tiếp theo cần khoảng vài ms để thực hiện.
- ❖ Vì bộ xử lý thực hiện tác vụ rất nhanh nên dưới góc nhìn người dùng thì hầu như các task là được thực hiện 1 cách liên tục.



Hình ảnh nhân hệ điều hành Kernel quản lý và điều hành các task theo thời gian thực

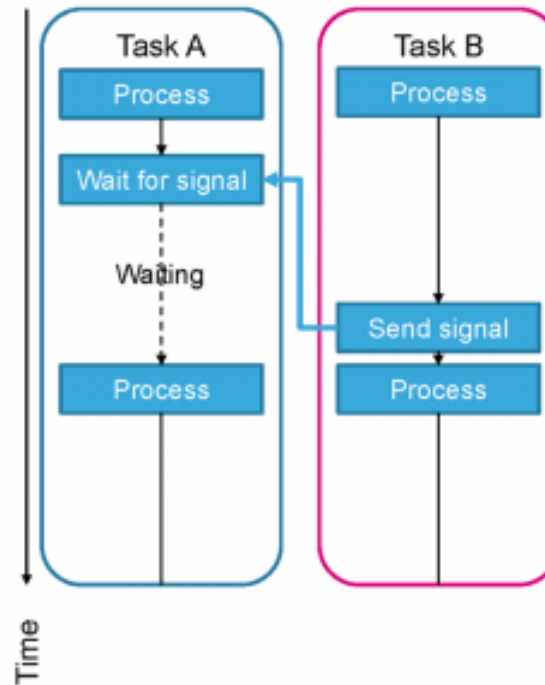


C2: Hệ điều hành trên các hệ thống nhúng

3. Hệ điều hành thời gian thực



❖ Task là một công việc chạy liên tục trong vòng lặp vô tận và không bao giờ dừng lại. Một chương trình thường sẽ có nhiều task khác nhau. Kernel sẽ quản lý việc chuyển đổi giữa các task, nó sẽ lưu lại trạng thái của task sắp bị hủy và khôi phục lại trạng thái của task tiếp theo.



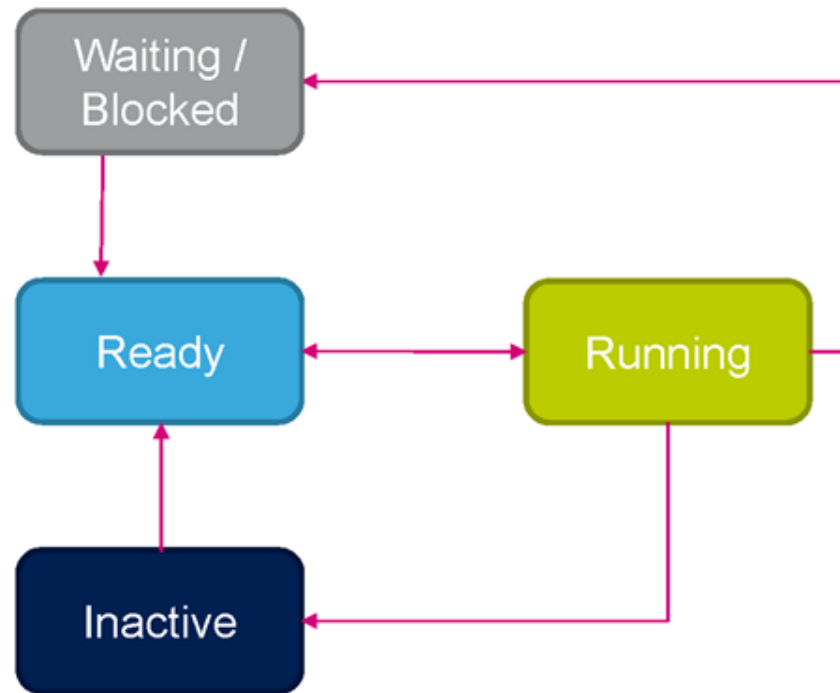
Hình ảnh một chương trình thực hiện nhiều task khác nhau.

C2: Hệ điều hành trên các hệ thống nhúng

3. Hệ điều hành thời gian thực



- Một task trong RTOS thường có các trạng thái như sau



- ❖ *RUNNING*: nhiệm vụ/công việc đang thực thi
- ❖ *READY*: nhiệm vụ/công việc đang sẵn sàng để thực hiện
- ❖ *WAITING*: nhiệm vụ/công việc đang chờ thực hiện
- ❖ *INACTIVE*: nhiệm vụ/công việc không được kích hoạt

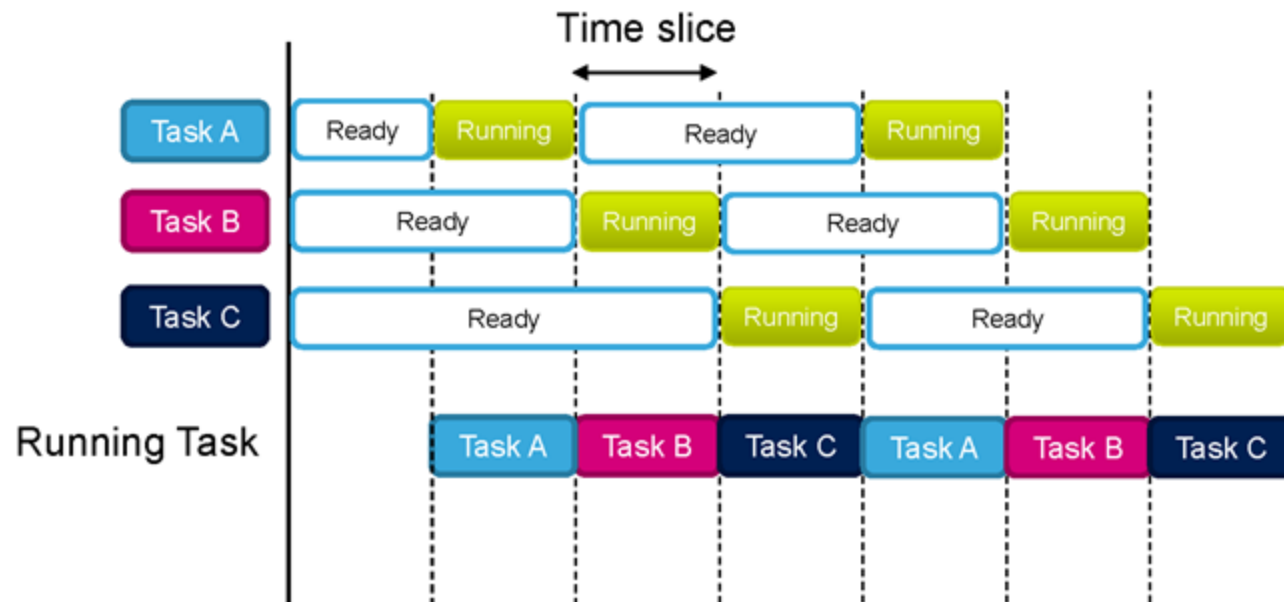
C2: Hệ điều hành trên các hệ thống nhúng

3. Hệ điều hành thời gian thực



❖ **Scheduler:** Đây là 1 thành phần của kernel quyết định task nào được thực thi. Có một số luật cho scheduling như:

- **Cooperative:** giống với lập trình thông thường, mỗi task chỉ có thể thực thi khi task đang chạy dừng lại, nhược điểm của nó là task này có thể dùng hết tất cả tài nguyên của CPU
- **Round-robin:** mỗi task được thực hiện trong thời gian định trước (time slice) và không có ưu tiên.



C2: Hệ điều hành trên các hệ thống nhúng

3. Hệ điều hành thời gian thực



❖ Đối với thuật giải RR, mỗi tiến trình trước khi bắt đầu được đưa vào danh sách hàng đợi (Ready Queue) để xử lý, sẽ được cấp phát cho một đơn vị thời gian nhất định để thực hiện. Ta gọi chung đơn vị thời gian nhất định này với cái tên là quantum.



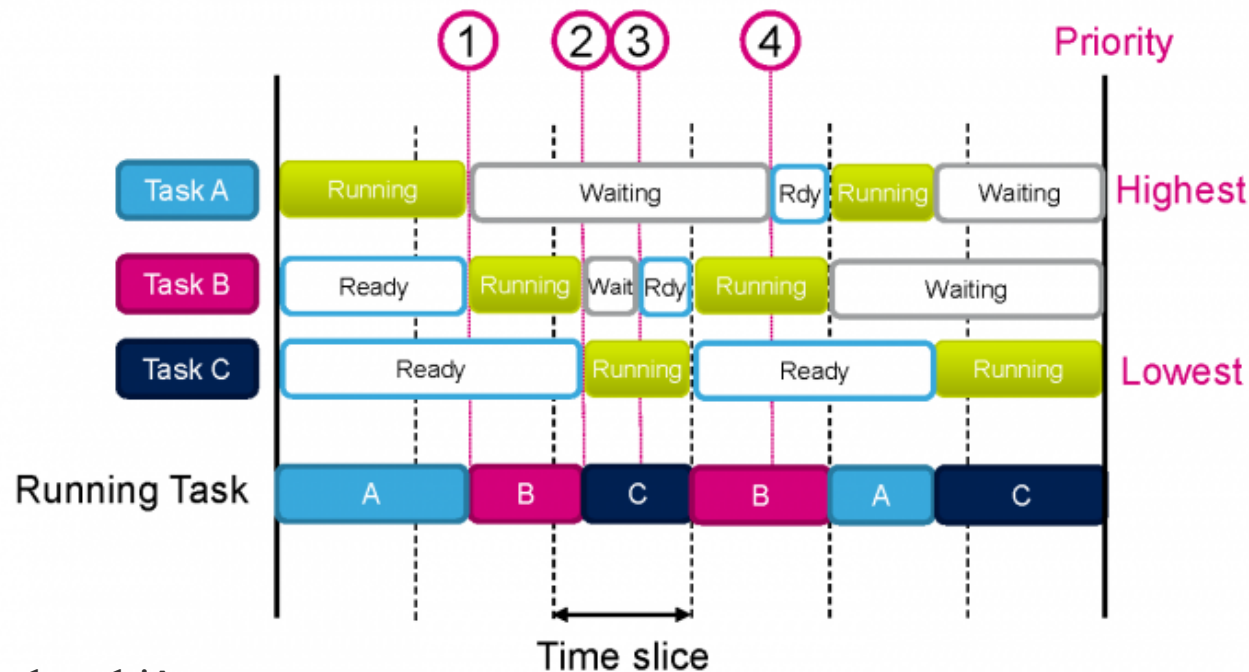
❖ Như vậy, khi một tiến trình sử dụng hết thời gian quantum mà nó được cấp phát, thì dù vẫn còn phải xử lý tiếp, phần dư của nó cũng sẽ được chuyển về phía sau trong danh sách hàng đợi. Sau đó, căn cứ vào danh sách Ready list đã nạp trước đó, Kernel sẽ lấy tiếp tiến trình kế cận để đưa vào xử lý, với mức quantum là như nhau cho tất cả các tiến trình.

C2: Hệ điều hành trên các hệ thống nhúng

3. Hệ điều hành thời gian thực



Priority: Task được phân quyền cao nhất sẽ được thực hiện trước, nếu các task có cùng quyền như nhau thì sẽ giống với round-robin, các task có mức ưu tiên thấp hơn sẽ được thực hiện cho đến cuối time slice.



1. Task A chờ thực hiện
2. Task B chờ thực hiện
3. Task B đang sẵn sàng thực hiện
4. Task A đang sẵn sàng thực hiện



C2: Hệ điều hành trên các hệ thống nhúng

3. Hệ điều hành thời gian thực



Ví dụ: Xét các tiến trình/công việc sau.

Process	Arrival Time	Burst Time
P ₁	0	24
P ₂	1	3
P ₃	2	3

❖ Với bảng dữ liệu trên, ta biết thêm được quantum time=4. Như vậy, để tính toán thuận tiện, ta cũng tiếp tục sử dụng giản đồ Gantt:



- Với giản đồ Gantt này, ta có thể tính được:
 - Thời gian xử lý: P₁=24, P₂=3 và P₃= 3.
 - Thời gian đợi lần lượt:
 - + P₁ đợi 0 + (10- 4) = 6 (ms).
 - + P₂ đợi 4 -1=3 (ms).
 - + P₃ đợi 7-2=5 (ms).
 - Thời gian hoàn tất tiến trình:
 - + P₁: 30 (ms).
 - + P₂: 6 (ms).
 - + P₃: 8 (ms).
 - Thời gian trung bình: $Avg_{WT} = (6+3+5)/3 = 4.66$



C2: Hệ điều hành trên các hệ thống nhúng

3. Hệ điều hành thời gian thực



Bài tập:

Hãy cho biết kết quả điều phối, tính thời gian chờ trung bình và thời gian hoàn thành trung bình của các tiến trình/công việc sau theo chiến lược điều phối Round Robin với thời lượng quantum = 4.

Tiến trình	Thời gian xử lý
P1	13
P2	8
P3	3
P4	5

Gợi ý: RR (Round Robin – xoay vòng)

- Tiến trình đến trước được xử lý trước
- Mỗi tiến trình thực hiện trong thời gian tối đa quantum
- Không có độ ưu tiên



C2: Hệ điều hành trên các hệ thống nhúng

4. Các hệ điều hành nhúng



- Các hệ điều hành dùng trong các hệ nhúng nổi trội hiện nay bao gồm:
 - Embedded Linux
 - VxWorks
 - Win CE
 - Lynxos
 - BSD
 - Green Hills,
 - QNX và DOS.



C2: Hệ điều hành trên các hệ thống nhúng

4. Các hệ điều hành nhúng



- Phần mềm nhúng là phần mềm tạo nên phần hồn, phần trí tuệ của các sản phẩm nhúng. Phần mềm nhúng ngày càng có tỉ lệ cao trong giá trị của các sản phẩm nhúng. Hiện nay phần lớn các phần mềm nhúng nằm trong các sản phẩm truyền thông và các sản phẩm điện tử gia dụng (consumer electronics) tiếp đến là trong các sản phẩm ô tô, phương tiện vận chuyển, máy móc thiết bị y tế, các thiết bị năng lượng các thiết bị cảnh báo bảo vệ và các sản phẩm đo và điều khiển. Để có thể tồn tại và phát triển, các sản phẩm công nghiệp và tiêu dùng cần phải thường xuyên đổi mới và ngày càng có nhiều chức năng tiện dụng và thông minh hơn. Các chức năng này phần lớn do các chương trình nhúng tạo nên.



C2: Hệ điều hành trên các hệ thống nhúng

4. Các hệ điều hành nhúng



- Phần mềm nhúng là một lĩnh vực công nghệ then chốt cho sự phát triển kinh tế của nhiều quốc gia trên thế giới như Nhật Bản, Hàn Quốc, Phần Lan và Trung Quốc. Tại Mỹ có nhiều chương trình hỗ trợ của nhà nước để phát triển các hệ thống và phần mềm nhúng. Hàn Quốc có những dự án lớn nhằm phát triển công nghệ phần mềm nhúng như các thiết bị gia dụng nối mạng Internet, hệ thống phần mềm nhúng cho phát triển thành phố thông minh, dự án phát triển ngành công nghiệp phần mềm nhúng, trung tâm hỗ trợ các ngành công nghiệp hậu PC v.v... Hàn Quốc cũng chấp nhận Embedded linux như một hệ điều hành chủ chốt trong việc phát triển các sản phẩm nhúng của mình. Thụy Điển coi phát triển các hệ nhúng có tầm quan trọng chiến lược cho sự phát triển của đất nước. Phần Lan có những chính sách quốc gia tích cực cho nghiên cứu phát triển các hệ nhúng đặc biệt là các phần mềm nhúng. Những quốc gia này còn thành lập nhiều viện nghiên cứu và trung tâm phát triển các hệ nhúng..

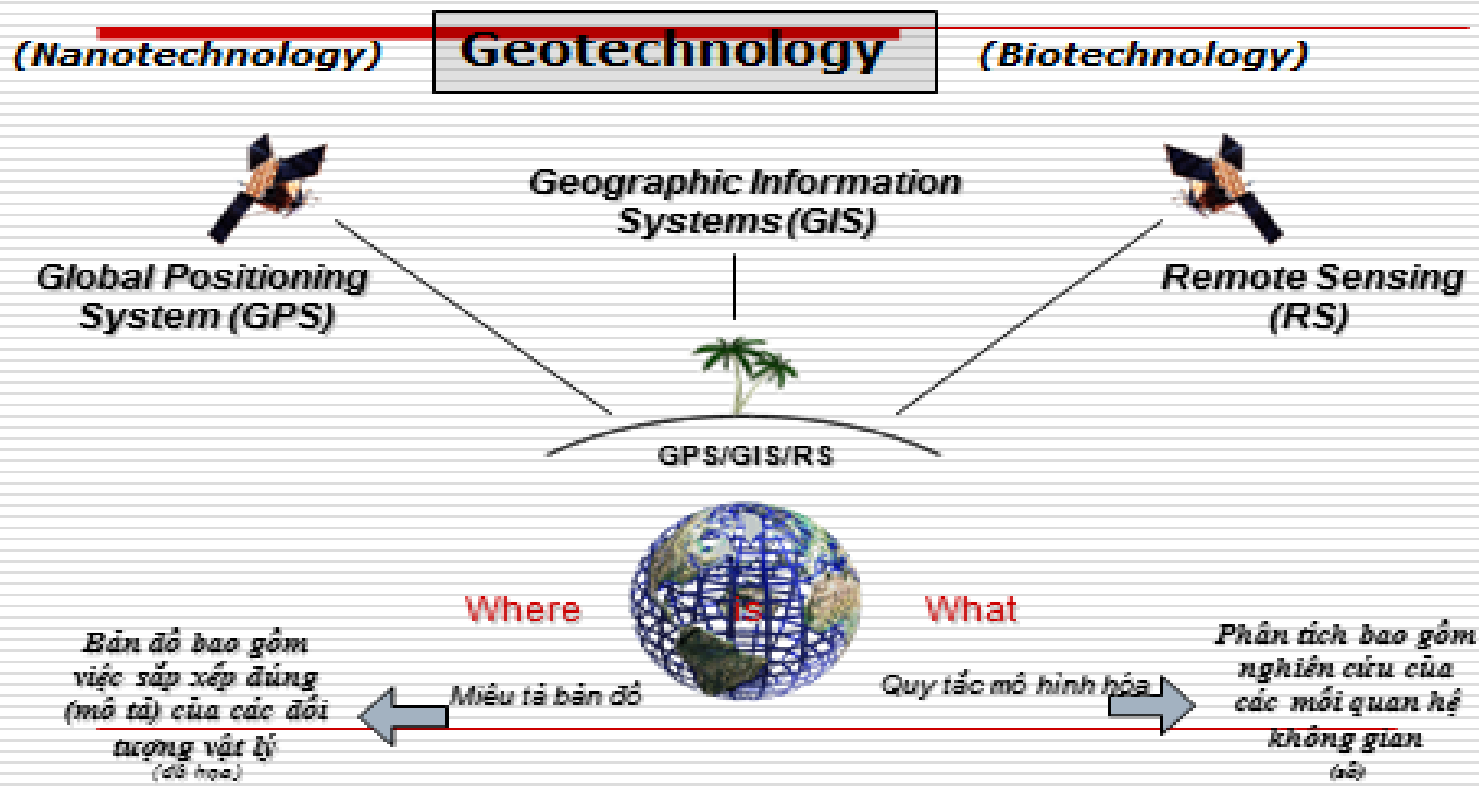


C2: Hệ điều hành trên các hệ thống nhúng

4. Các hệ điều hành nhúng



Bức tranh lớn về công nghệ của thế kỷ 21

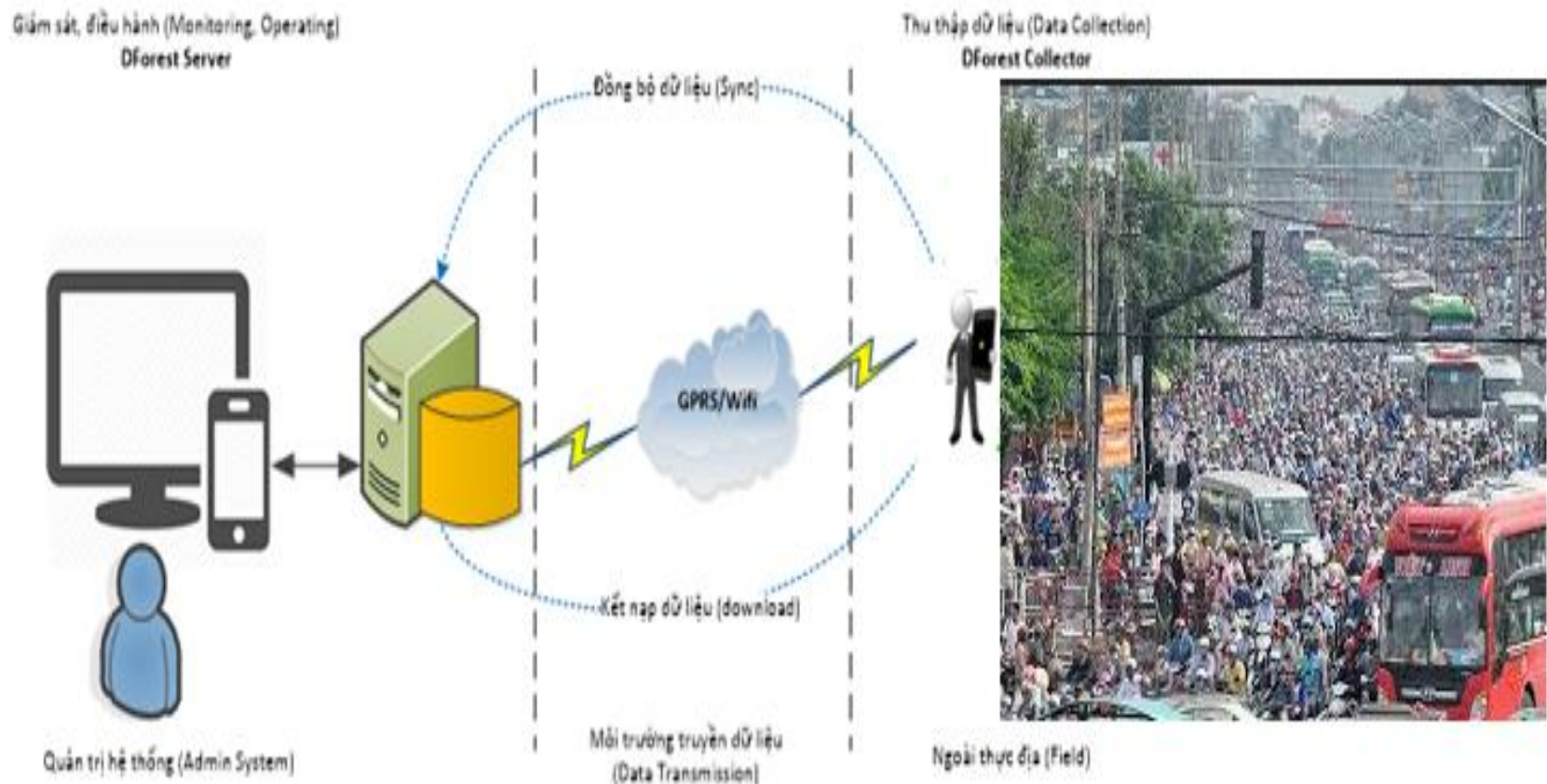


Hình ảnh ứng dụng phần mềm nhúng trong quản lý giao thông



C2: Hệ điều hành trên các hệ thống nhúng

4. Các hệ điều hành nhúng



Mô tả giải pháp của hệ thống giám sát giao thông



C2: Hệ điều hành trên các hệ thống nhúng

4. Các hệ điều hành nhúng



Hình ảnh Camera giám sát trên quốc lộ

C2: Hệ điều hành trên các hệ thống nhúng

5. Hệ điều hành Windows CE



❖ Hệ điều hành Windows CE

- Năm 1996, Microsoft cho ra đời phiên bản hệ điều hành đầu tiên CE 1.0 cho các thiết bị nhúng, không phải là máy tính cá nhân. Đến phiên bản 3.0 ra đời vào năm 2000 thì Windows CE đã có khả năng hỗ trợ các xử lý thời gian thực và các công nghệ đa phương tiện tiên tiến như DirectDraw, DirectShow và Windows Media Player.
- Năm 2005, Windows CE 5.0 xuất hiện. Lần này, nó cung cấp cho lập trình viên nhiều công nghệ mới lúc bấy giờ như USB 2.0, *SDIO*, *Windows Media 9* và *Microsoft Internet Explorer 6*. Bên cạnh đó là cơ chế thống nhất để "build" hệ thống cùng nhiều tùy biến khác để phát triển hệ thống một cách dễ dàng.
- Với phiên bản 6.0 ra đời vào năm 2006, kiến trúc hệ thống đã trải qua những thay đổi quan trọng. Dung lượng bộ nhớ ảo cho mỗi tiến trình tăng lên, khả năng nhiều tiến trình chạy đồng thời (đến 32,000 tiến trình). Nhân của hệ thống cũng được hợp nhất..
- Tháng 11 năm 2007, Microsoft cho ra đời bản nâng cấp Windows Embedded CE 6.0 R2, với nhiều thành phần và các gói *Board Support Package* (BSP) mới được thêm vào hệ điều hành. .



C2: Hệ điều hành trên các hệ thống nhúng

5. Hệ điều hành Windows CE



❖ Hệ điều hành Windows CE

Windows CE là một hệ điều hành mã nguồn mở 32 bit. Nó được sử dụng cho các hệ thống nhúng thông minh như các vi điều khiển dùng trong công nghiệp, các thiết bị đầu cuối trong tiêu dùng như camera, điện thoại di động, các thiết bị giải trí gia đình.

- **Các dòng vi xử lý được hỗ trợ bởi Windows CE**
- ARM của hãng Acorn.
- MIPS của MIPS Technologies.
- SH4 của Renesas.
- x86 của Intel.



C2: Hệ điều hành trên các hệ thống nhúng

6 Các hệ điều hành Embedded Linux, Android



❖ Hệ điều hành Embedded Linux.

- Embedded Linux hiện đang phát triển mạnh và chiếm vị trí số 1. Hiện nay 40% các nhà thiết kế các hệ nhúng cân nhắc đầu tiên sử dụng Embedded Linux cho các ứng dụng mới của mình và sau đó mới đến các hệ điều hành nhúng truyền thống như VxWorks, Win CE. Các đối thủ cạnh tranh của Embedded Linux hiện nay là các hệ điều hành nhúng tự tạo và Windows CE.
- Sở dĩ Embedded Linux có sự phát triển vượt bậc là do có sức hấp dẫn đối với các ứng dụng không đòi hỏi thời gian thực như set-top, các hệ server nhúng, các ứng dụng giá thành thấp và đòi hỏi thời gian đưa sản phẩm ra thị trường nhanh. Mặt khác Linux là phần mềm mã nguồn mở nên bất kỳ ai cũng có thể hiểu và thay đổi theo ý mình. Linux cũng là một hệ điều hành có cấu trúc module và chiếm ít bộ nhớ trong khi Windows không có các đặc tính ưu việt này.
- Bên cạnh các ưu điểm trên thì Embedded Linux cũng có các nhược điểm sau:
 - Embedded Linux không phải là hệ điều hành thời gian thực nên có thể không phù hợp với một số ứng dụng như điều khiển quá trình, các ứng dụng có các yêu cầu xử lý khẩn cấp.
 - Embedded Linux thiếu một chuẩn thống nhất và không phải là sản phẩm của một nhà cung cấp duy nhất nên khả năng hỗ trợ kỹ thuật ít.



C2: Hệ điều hành trên các hệ thống nhúng

6 Các hệ điều hành Embedded Linux, Android

❖ Hệ điều hành Android.

- Android là một hệ điều hành có mã nguồn mở dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng.
- Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google, sau này được chính Google mua lại vào năm 2005 và hệ điều hành Android đã ra mắt vào năm 2007. Chiếc điện thoại đầu tiên chạy Android là HTC Dream được bán vào ngày 22 tháng 10 năm 2008.



C2: Hệ điều hành trên các hệ thống nhúng

6 Các hệ điều hành Embedded Linux, Android



❖ Hệ điều hành Android.

- Hệ điều hành mã nguồn mở Android cùng với giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới. Android chiếm 75% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Tháng 10 năm 2012, đã có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play (cửa hàng ứng dụng chính của Android) ước tính khoảng 25 tỷ lượt. Hiện nay con số này đã giảm xuống do sự ảnh hưởng lớn của iOS từ Apple và một phần nhỏ của Windows Phone, tuy nhiên hiện nay Android vẫn dẫn đầu thị phần điện thoại thông minh trên thị trường.

C2: Hệ điều hành trên các hệ thống nhúng

6 Các hệ điều hành Embedded Linux, Android



❖ Ưu và nhược điểm của Android

• Ưu điểm:

- + Là hệ điều hành có mã nguồn mở nên khả năng tùy biến cao, có thể tùy ý chỉnh sửa mà không có sự can thiệp hay cấm cản từ Google.
- + Đa dạng sản phẩm, rất nhiều hãng điện thoại, thiết bị công nghệ đã ưu ái chọn Android cho thiết bị của họ, giá cả thì hợp lý từ bình dân đến cao cấp.
- + Kho ứng dụng Google Play Store đồ sộ.
- + Thân thiện và dễ sử dụng.
- + Khả năng đa nhiệm, chạy cùng lúc nhiều ứng dụng cao.



C2: Hệ điều hành trên các hệ thống nhúng

6 Các hệ điều hành Embedded Linux, Android



- **Nhược điểm:**

- + Dễ nhiễm phần mềm độc hại và virus. Do tính chất mã nguồn mở, nhiều phần mềm không được kiểm soát có chất lượng không tốt hoặc lỗi bảo mật vẫn được sử dụng.
- + Kho ứng dụng quá nhiều dẫn đến khó kiểm soát chất lượng, thiếu các ứng dụng thật sự tốt.
- + Sự phân mảnh lớn. Trong khi một số thiết bị Android xuất sắc đã trình làng như Galaxy S5, Galaxy Note 4, Xperia Z3..., vẫn còn rất nhiều sản phẩm giá rẻ bình thường khác.
- + Cập nhật không tự động với tất cả thiết bị. Khi một phiên bản hệ điều hành mới ra mắt, không phải tất cả sản phẩm đều được cập nhật, thậm chí nếu muốn trải nghiệm hệ điều hành mới đòi hỏi thường xuyên phải mua mới thiết bị.

Chương 3: Kỹ thuật lập trình nhúng



1 Tác vụ (task) và tiến trình (process)

2 Lập lịch (Scheduling)

3 Truyền thông và đồng bộ

4 Xử lý ngắt

C3: Kỹ thuật lập trình nhúng



- 1. Tác vụ (task) và tiến trình (process)
- 2. Lập lịch (Scheduling)
 - 2.1. Các khái niệm
 - 2.2. Các phương pháp lập lịch phổ biến
 - 2.3. Kỹ thuật lập lịch
- 3. Truyền thông và đồng bộ
 - 3.1. Semaphore
 - 3.2. Monitor
- 4. Xử lý ngắt.



C3: Kỹ thuật lập trình nhúng

1. Tác vụ (task) và tiến trình (process)



❖ Tác vụ (task)

- Task: bộ điều phối không cung cấp cơ chế, mà đưa ra các quyết định. Các hệ điều hành xây dựng nhiều chiến lược khác nhau để thực hiện việc điều phối, nhưng tựu chung lại cần đạt được các mục tiêu sau :
 - **Sự công bằng (Fairness)** : Các tiến trình chia sẻ CPU một cách công bằng, không có tiến trình nào phải chờ đợi vô hạn để được cấp phát CPU.
 - **Tính hiệu quả (Efficiency)** : Hệ thống phải tận dụng được CPU 100% thời gian.
 - **Thời gian đáp ứng hợp lý (Response time)** : Cực tiểu hoá thời gian hồi đáp cho các tương tác của người sử dụng
 - **Thời gian lưu lại trong hệ thống (Turnaround Time)** : Cực tiểu hóa thời gian hoàn tất các tác vụ xử lý theo lô.
 - **Thông lượng tối đa (Throughput)** : Cực đại hóa số công việc được xử lý trong một đơn vị thời gian. Tuy nhiên thường không thể thỏa mãn tất cả các mục tiêu kể trên vì bản thân chúng có sự mâu thuẫn với nhau mà chỉ có thể dung hòa chúng ở mức độ nào đó.

C3: Kỹ thuật lập trình nhúng

1. Tác vụ (task) và tiến trình (process)



❖ Tiến trình (process)

- Điều phối hoạt động của các tiến trình là một vấn đề rất phức tạp, đòi hỏi hệ điều hành khi giải quyết phải xem xét nhiều yếu tố khác nhau để có thể đạt được những mục tiêu đề ra. Một số đặc tính của tiến trình cần được quan tâm như là một tiêu chuẩn điều phối :
- **Tính hướng xuất / nhập của tiến trình (I/O-boundedness):** Khi một tiến trình nhận được CPU, chủ yếu nó chỉ sử dụng CPU đến khi phát sinh một yêu cầu nhập xuất ? Hoạt động của các tiến trình như thế thường bao gồm nhiều lượt sử dụng CPU , mỗi lượt trong một thời gian khá ngắn.
- **Tính hướng xử lý của tiến trình (CPU-boundedness):** Khi một tiến trình nhận được CPU, nó có khuynh hướng sử dụng CPU đến khi hết thời gian dành cho nó ? Hoạt động của các tiến trình như thế thường bao gồm một số ít lượt sử dụng CPU , nhưng mỗi lượt trong một thời gian đủ dài.
- **Tiến trình tương tác hay xử lý theo lô :** Người sử dụng theo kiểu tương tác thường yêu cầu được hồi đáp tức thời đối với các yêu cầu của họ, trong khi các tiến trình của tác vụ được xử lý theo lô nói chung có thể trì hoãn trong một thời gian chấp nhận được.
- **Độ ưu tiên của tiến trình :** Các tiến trình có thể được phân cấp theo một số tiêu chuẩn đánh giá nào đó, một cách hợp lý, các tiến trình quan trọng hơn (có độ ưu tiên cao hơn) cần được ưu tiên hơn.
- **Thời gian đã sử dụng CPU của tiến trình :** Một số quan điểm ưu tiên chọn những tiến trình đã sử dụng CPU nhiều thời gian nhất vì hy vọng chúng sẽ cần ít thời gian nhất để hoàn tất và rời khỏi hệ thống . Tuy nhiên cũng có quan điểm cho rằng các tiến trình nhận được CPU trong ít thời gian là những tiến trình đã phải chờ lâu nhất, do vậy ưu tiên chọn chúng.
- **Thời gian còn lại tiến trình cần để hoàn tất :** Có thể giảm thiểu thời gian chờ đợi trung bình của các tiến trình bằng cách cho các tiến trình cần ít thời gian nhất để hoàn tất được thực hiện trước. Tuy nhiên đáng tiếc là rất hiếm khi biết được tiến trình cần bao nhiêu thời gian nữa để kết thúc xử lý.

C3: Kỹ thuật lập trình nhúng

2. Lập lịch (Scheduling)



- Scheduling
 - Khái niệm: Có nhiều tiến trình nhưng tại một thời điểm chỉ có một tiến trình có thể được thực thi (trạng thái là running). Vấn đề chọn tiến trình nào để thực thi ở bước kế tiếp (từ trạng thái ready chuyển sang trạng thái running). Lập lịch (Scheduling) là thao tác quyết định tiến trình nào được quyền thực thi.
 - Các phương pháp lập lịch phổ biến:
 - Long-term scheduling: Xác định chương trình nào được chấp nhận nạp vào hệ thống để thực thi
 - Medium-term scheduling: Process nào được đưa vào (swapon), đưa ra khỏi (swapout) bộ nhớ chính. Được thực hiện bởi phần quản lý bộ nhớ và được thảo luận ở phần quản lý bộ nhớ
 - Short term scheduling: Xác định process nào trong ready queue sẽ được chiếm CPU để thực thi kế tiếp (còn được gọi là định thời CPU, CPU scheduling) . Short term scheduler còn được gọi với tên khác là dispatcher. Bộ định thời short-term được gọi mỗi khi có một trong các sự kiện/interrupt sau xảy ra:
 - Ngắt thời gian (clockinterrupt)
 - Ngắt ngoại vi (I/Ointerrupt)
 - Lỗi gọi hệ thống (operatingsystemcall)
 - Signal

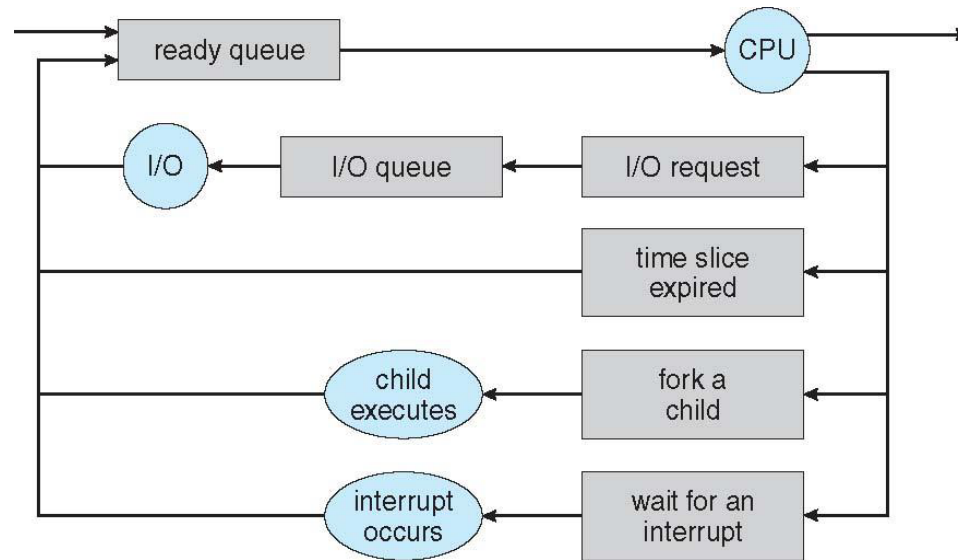


C3: Kỹ thuật lập trình nhúng

2. Lập lịch (Scheduling)



➤ Kỹ thuật lập lịch:



- ❑ Có nhiều hàng đợi:
- ❑ ready queue: hàng đợi chứa các tiến trình sẵn sàng chạy
- ❑ I/O queue: hàng đợi chứa các tiến trình sẵn sàng thi hành I/O
- ❑ Lựa chọn tiến trình nào → **điều phối tiến trình**



C3: Kỹ thuật lập trình nhúng

2. Lập lịch (Scheduling)



Tiến trình = Chương trình?

```
main ()
{
    ...;
}

A() {
    ...
}
```

Program

```
main ()
{
    ...;
}

A() {
    ...
}
```

Process

Heap

Stack

A

main

- Một chương trình có thể có nhiều tiến trình
 - Mở *Notepad.exe* xem file *a.txt* → 1 tiến trình.
 - Mở *Notepad.exe* xem file *b.txt* → 1 tiến trình.
- Chương trình nhìn từ góc độ mã lệnh chỉ là một phần của tiến trình.

C3: Kỹ thuật lập trình nhúng

3. Truyền thông và đồng bộ



- **Truyền thông:** là nhu cầu liên lạc giữa các tiến trình với mục đích.
 - Chia sẻ thông tin: nhiều tiến trình có thể cùng quan tâm đến những dữ liệu nào đó, do vậy hệ điều hành cần cung cấp một môi trường cho phép sự truy cập đồng thời đến các dữ liệu chung.
 - Hợp tác hoàn thành tác vụ: đôi khi để đạt được một sự xử lý nhanh chóng, người ta phân chia một tác vụ thành các công việc nhỏ có thể tiến hành song song.



C3: Kỹ thuật lập trình nhúng

3. Truyền thông và đồng bộ



❖ Các vấn đề nảy sinh trong việc liên lạc giữa các tiến trình

- Liên kết tường minh hay tiềm ẩn (explicit naming/implicit naming) : tiến trình có cần phải biết tiến trình nào đang trao đổi hay chia sẻ thông tin với nó? Mỗi liên kết được gọi là tường minh khi được thiết lập rõ ràng , trực tiếp giữa các tiến trình, và là tiềm ẩn khi các tiến trình liên lạc với nhau thông qua một qui ước ngầm nào đó.
- Liên lạc theo chế độ đồng bộ hay không đồng bộ (blocking / non-blocking): khi một tiến trình trao đổi thông tin với một tiến trình khác, các tiến trình có cần phải đợi cho thao tác liên lạc hoàn tất rồi mới tiếp tục các xử lý khác ?
- Liên lạc giữa các tiến trình trong hệ thống tập trung và hệ thống phân tán: cơ chế liên lạc giữa các tiến trình trong cùng một hệ thống có sự khác biệt với việc liên lạc giữa các tiến trình giữa những hệ thống khác nhau?.



C3: Kỹ thuật lập trình nhúng

3. Truyền thông và đồng bộ



- Đồng bộ: để giải quyết tranh chấp tài nguyên giữa các tiến trình, đảm bảo việc chia sẻ tài nguyên một cách nhất quán, đảm bảo cho tại mỗi thời điểm chỉ có một tiến trình được thao tác lên dữ liệu được chia sẻ. Do đó, cần có cơ chế đồng bộ hoạt động của các tiến trình này.
- Semaphore: được đề nghị bởi Dijkstra năm 1965
 - Các đặc tính: Semaphore s ;
 - Có một giá trị nguyên dương $\rightarrow$ Semaphore s ; // $s \geq 0$
 - Chỉ được thao tác bởi 2 primitives:
Down(s)
Up(s)
 - Các primitive Down và Up được thực hiện không thể phân chia
 - Nhận xét Semaphore
 - Là một cơ chế tốt để thực hiện đồng bộ (dễ dùng cho N tiến trình)
 - Nhưng ý nghĩa sử dụng không rõ ràng
 - Mutual Exclusion: Down & Up.
 - Render-vous: Down & Up
 - Chưa phân biệt qua mô hình.
 - Khó sử dụng đúng: dễ nhầm lẫn



C3: Kỹ thuật lập trình nhúng



Đoạn găng (Critical Section) hay miền găng là đoạn mã có tác động đến các tài nguyên găng, chỉ cho phép một tiến trình thi hành tại một thời điểm.

- Tổ chức “độc quyền truy xuất”

Semaphore $s = 1$

P_i

Down (s)
CS;
Up(s)

- Tổ chức “hồ hện”

Semaphore $s = 0$

$P_1 :$

Job1;
Up(s)

$P_2 :$

Down (s);
Job2;

C3: Kỹ thuật lập trình nhúng

3. Truyền thông và đồng bộ



- Monitor: đề xuất bởi Hoare(1974) & Brinch (1975)
 - Là cơ chế đồng bộ hóa do Ngôn ngữ lập trình cung cấp
 - Hỗ trợ các chức năng như Semaphore
 - Dễ sử dụng và kiểm soát hơn Semaphore
- Đảm bảo Mutual Exclusion một cách tự động
- Sử dụng biến điều kiện để thực hiện Synchronization



C3: Kỹ thuật lập trình nhúng



■ Tổ chức “độc quyền truy xuất”

```
Monitor    M
<resource type> RC;
Function   AccessMutual
           CS; // access RC
```

```
Pi
M.AccessMutual(); //CS
```

■ Tổ chức “hò hẹn”

```
Monitor    M
Condition  c;
Function   F1
           Job1;
           Signal(c);
Function   F2
           Wait(c);
           Job2;
```

```
P1 :
M.F1();
```

```
P2 :
M.F2();
```

C3: Kỹ thuật lập trình nhúng

4. Xử lý ngắt



- Ngắt là một cơ chế yêu cầu CPU tạm dừng công việc (task) đang thực hiện để thực hiện 1 công việc khác. Nói cụ thể hơn, ngắt yêu cầu CPU tạm dừng chương trình đang thực hiện để thực hiện một chương trình con phục vụ ngắt.
- Người ta tạm chia ngắt ra làm hai loại: ngắt cứng và ngắt mềm. Các ngắt mềm được kích hoạt bằng lệnh INT n trong đó n là số hiệu ngắt dưới dạng một số hexa. Ngắt cứng khác với ngắt mềm ở chỗ không được kích hoạt bằng một lệnh INT n trong chương trình mà được kích hoạt bằng các tác động của các tín hiệu linh kiện điện tử như bàn phím, ổ đĩa,...



C3: Kỹ thuật lập trình nhúng

4. Xử lý ngắt



NonCS;

Disable Interrupt;

CS;

Enable Interrupt;

NonCS;

- *Disable Interrupt*: Cấm mọi ngắt, kể cả ngắt đồng hồ
- *Enable Interrupt*: Cho phép ngắt

Chương 4: Lập trình trên ARM



1 Kiến trúc ARM

2 Các bộ KIT

3 Keil tools cho lập trình nhúng trên ARM

4 Lập trình C trên ARM



C4: Lập trình trên ARM

1. Kiến trúc ARM



- ❑ ARM (viết tắt từ tên gốc là Advanced RISC Machine) là một loại cấu trúc vi xử lý 32 bit và 64 bit kiểu RISC được sử dụng rộng rãi trong các thiết kế nhúng. Chúng có đặc điểm tiết kiệm năng lượng, vì vậy các bộ CPU ARM chiếm ưu thế trong các sản phẩm điện tử di động.
- ❑ Các đặc tính của ARM bao gồm:
 - Cấu trúc nạp / lưu trữ.
 - Hỗ trợ tập lệnh trực giao.
 - Thanh ghi lớn.
 - Hầu hết các lệnh được thực hiện trong 1 chu kỳ CPU.
 - Chiều dài mã máy cố định.



C4: Lập trình trên ARM

1. Kiến trúc ARM

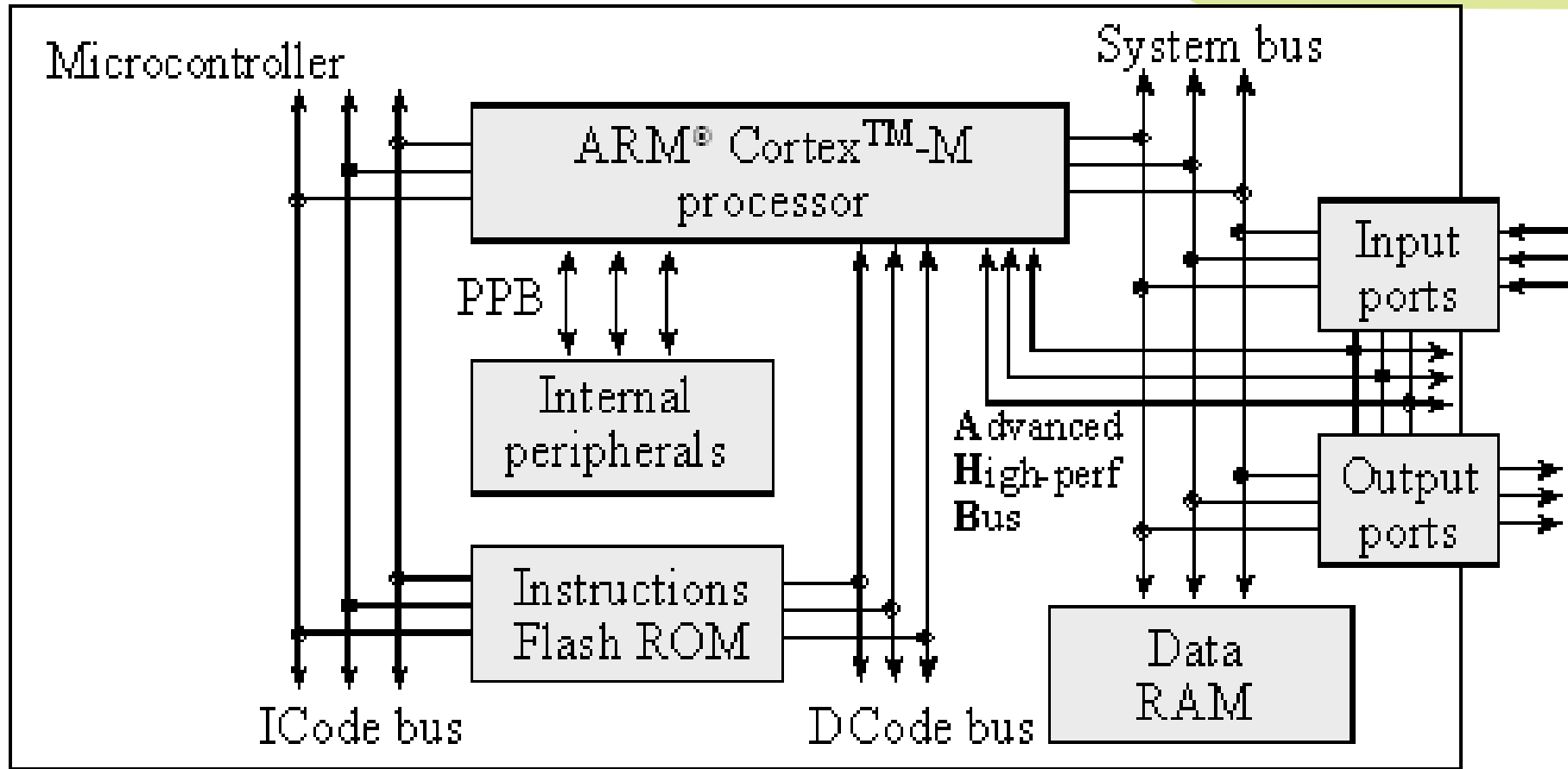


- ❑ Đối với người lập trình nhúng phần lớn thường hay sử dụng một dòng chip xử lý thông dụng như 8051, AVR, PIC,... Vì vậy mỗi khi muốn nâng cấp hay nghiên cứu một sản phẩm mới, ngoài việc phải coi lại kiến trúc phần cứng mà một điều quan trọng nữa là làm thế nào để chạy được chương trình ứng dụng đang hoạt động tốt ở bộ vi xử lý cũ. Nắm bắt được hạn chế này, các nhà sản xuất phần cứng đã cho ra đời dòng vi xử lý ARM với những chuẩn phần cứng đáp ứng khả năng tương thích với phần mềm.
- ❑ Lõi ARM Cortex được chia ra thành ba dòng cấu hình chính:
 - A: Application hỗ trợ các ứng dụng có độ phức tạp cao như: máy tính, điện thoại di động.
 - R: Realtime hỗ trợ các ứng dụng cần tính toán xử lý thời gian thực.
 - M: Microcontroller hỗ trợ cho các ứng dụng công nghiệp và điện tử dân dụng.



C4: Lập trình trên ARM

1. Kiến trúc ARM



Hình ảnh kiến trúc ARM



C4: Lập trình trên ARM

2. Các bộ KIT



- ❑ Để bắt đầu học về ARM, chúng ta nên có 1 bộ KIT ARM để thực hành ngay những gì học được (để nhớ và trực quan). Hiện nay trên thị trường có rất nhiều các loại KIT khác nhau:
 - STM32 (ST),
 - Nuvoton (Nuvoton),
 - Tiva C LaunchPad (TI)
 - ...

- ❑ Nếu khó khăn về tài chính thì có thể dùng Simulator (giả lập hệ thống)
 - Proteus.
 - Matlab.
 -



C4: Lập trình trên ARM

2. Các bộ KIT



- ❖ KIT Launchpad Tiva C của TI: Giá thành rẻ, chất lượng khá ổn và dễ dàng tìm mua ở VN).

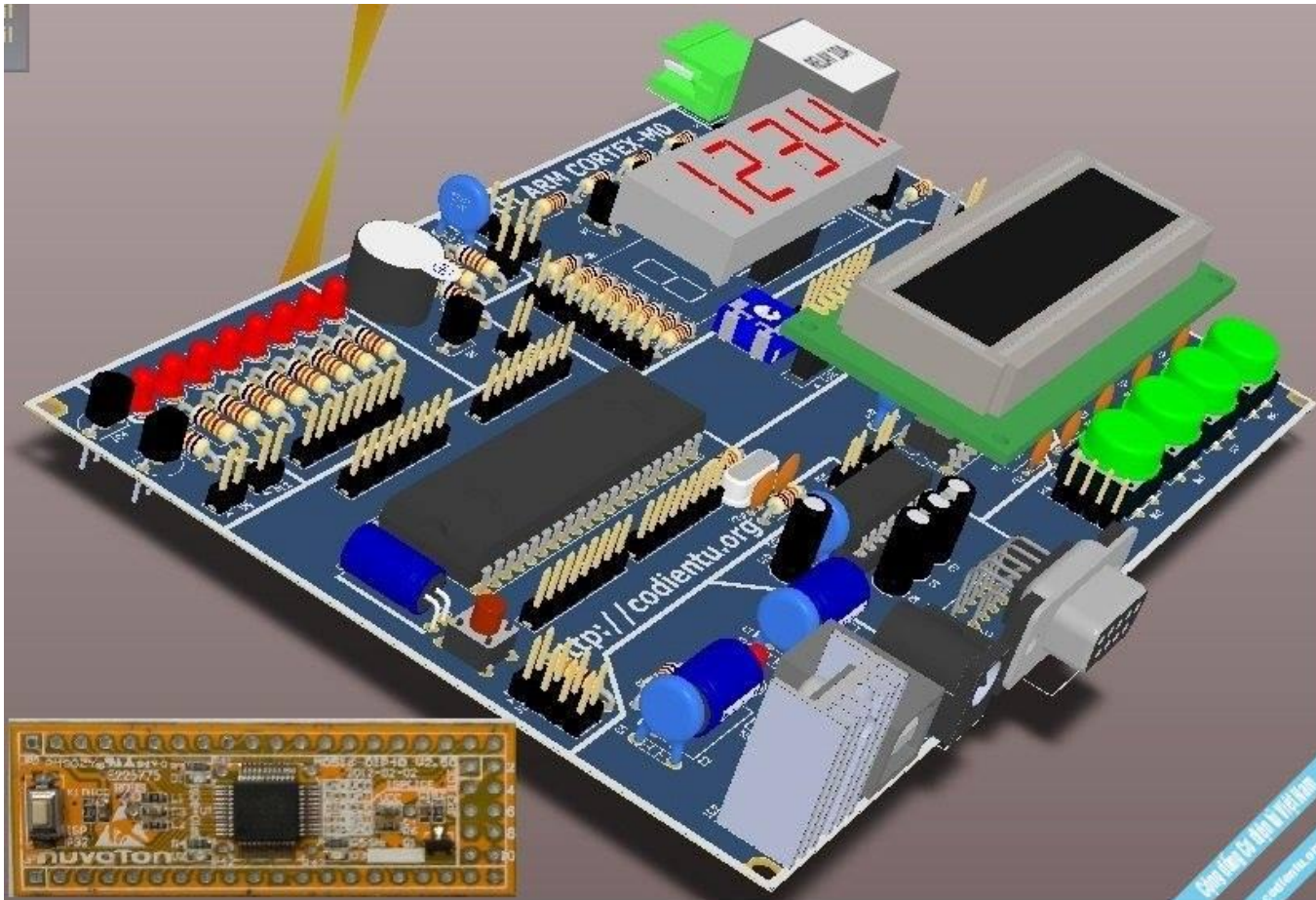


C4: Lập trình trên ARM

2. Các bộ KIT



- ❖ KIT thực hành ARM Cortex-M0: KIT này bao gồm sự thay thế tương đương giữa chip 8051 truyền thống và ARM Cortex-M0.

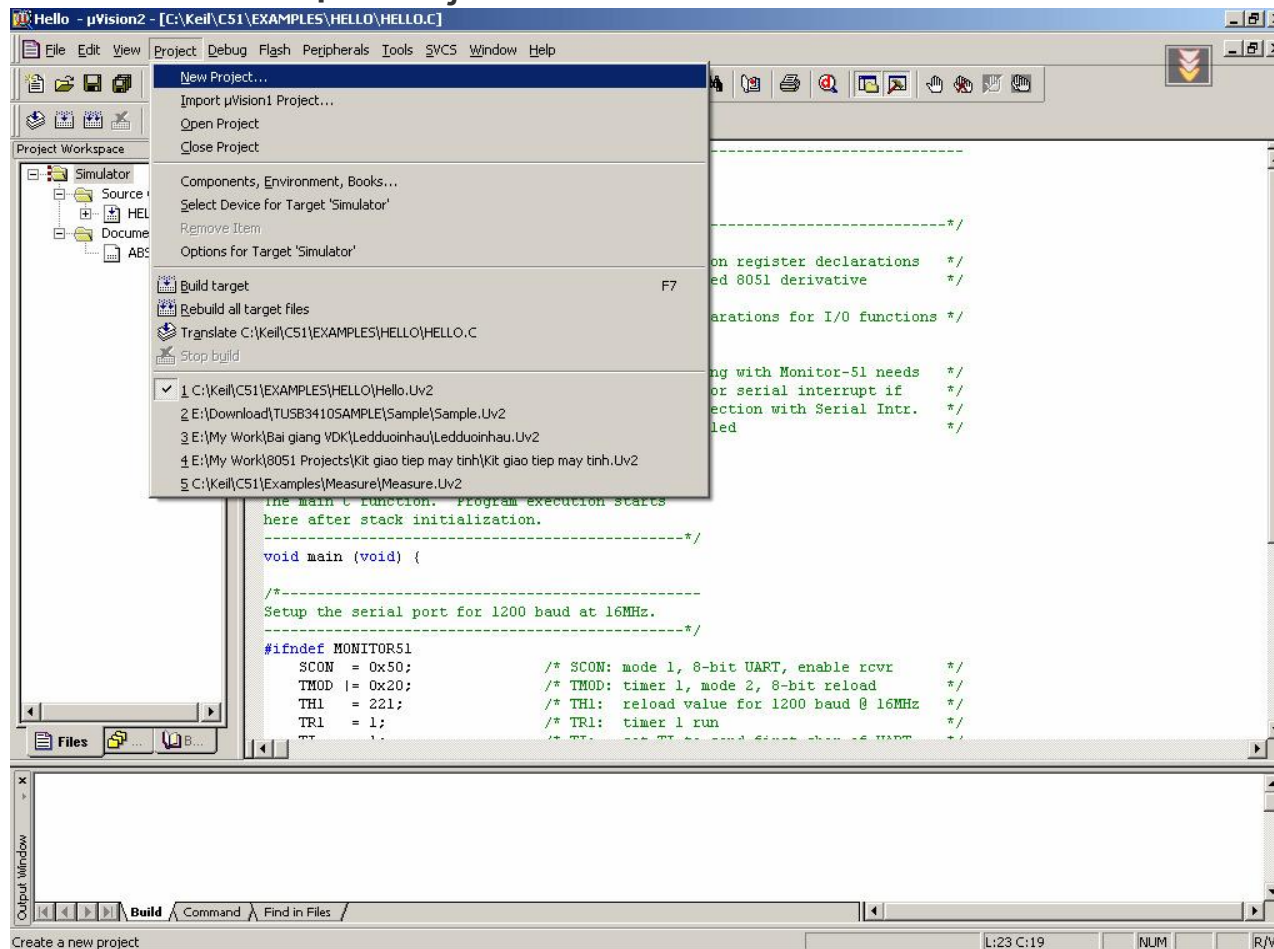


ARM

ARM

❖ Sử dụng ngôn ngữ lập trình Keil C.

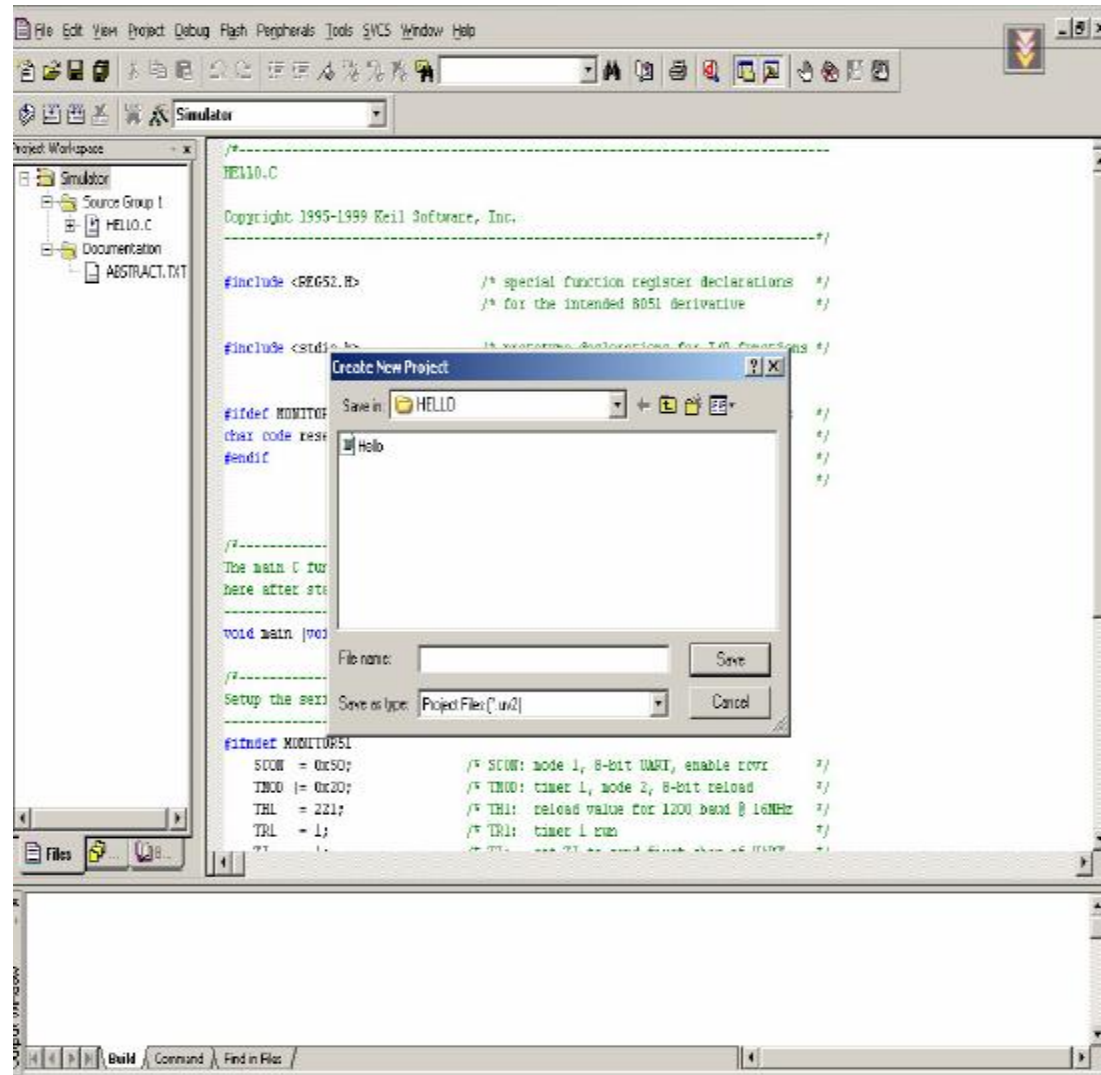
- Khởi tạo Project



C4: Lập trình trên ARM

3. Keil tools cho lập trình nhúng trên ARM

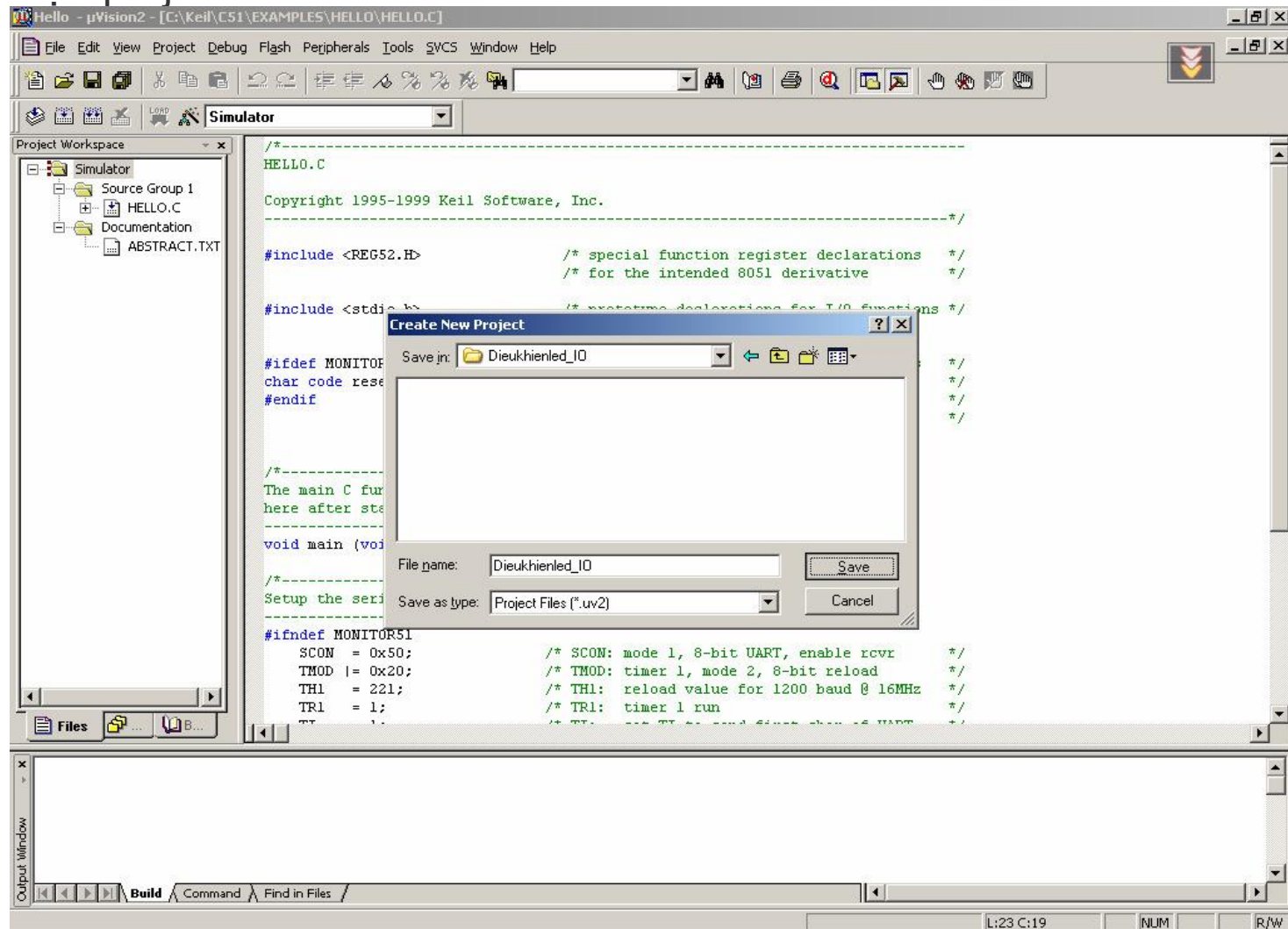
- Đặt tên cho Project



C4: Lập trình trên ARM

3. Keil tools cho lập trình nhúng trên ARM

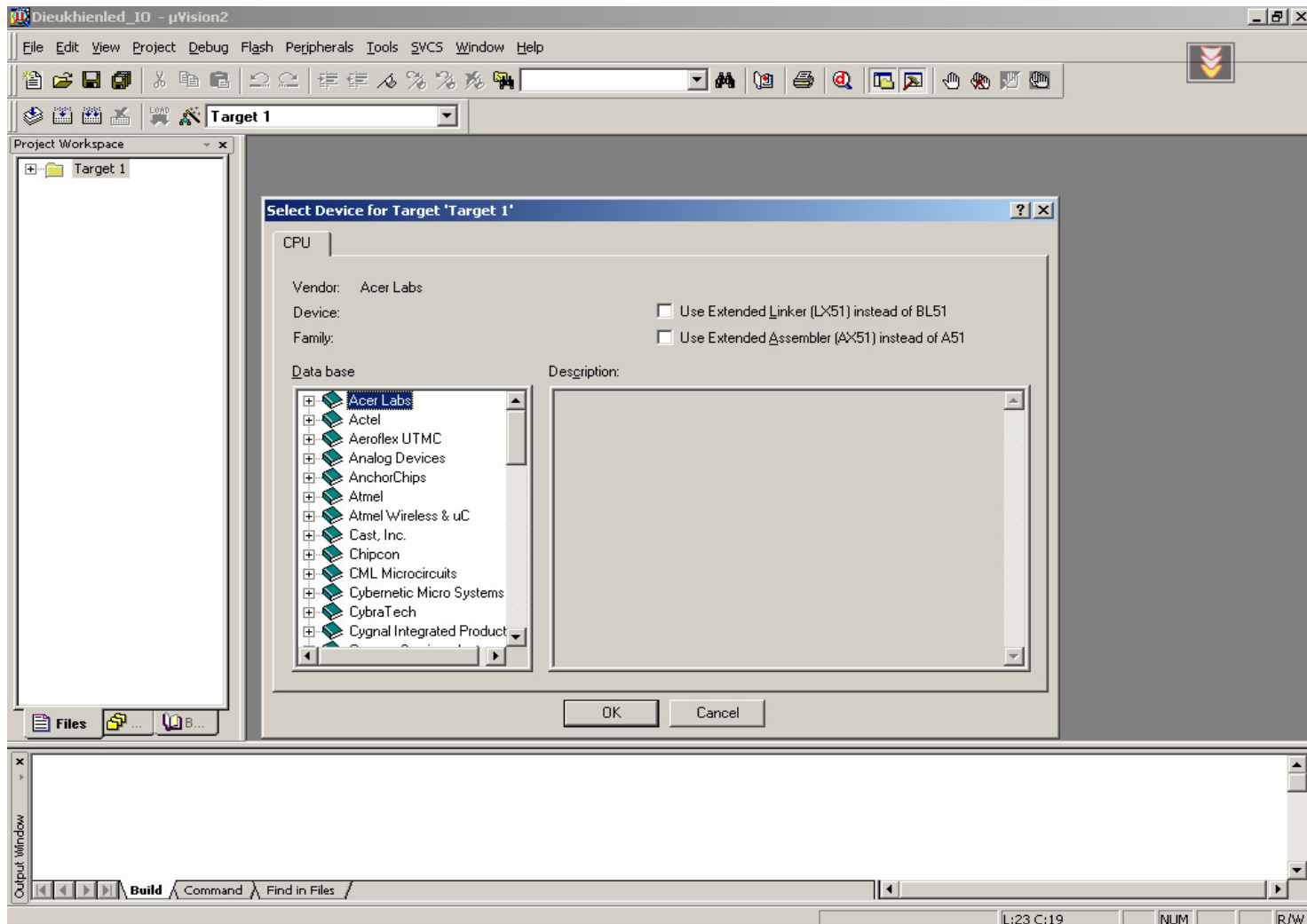
- Ta được project sau:



C4: Lập trình trên ARM

3. Keil tools cho lập trình nhúng trên ARM

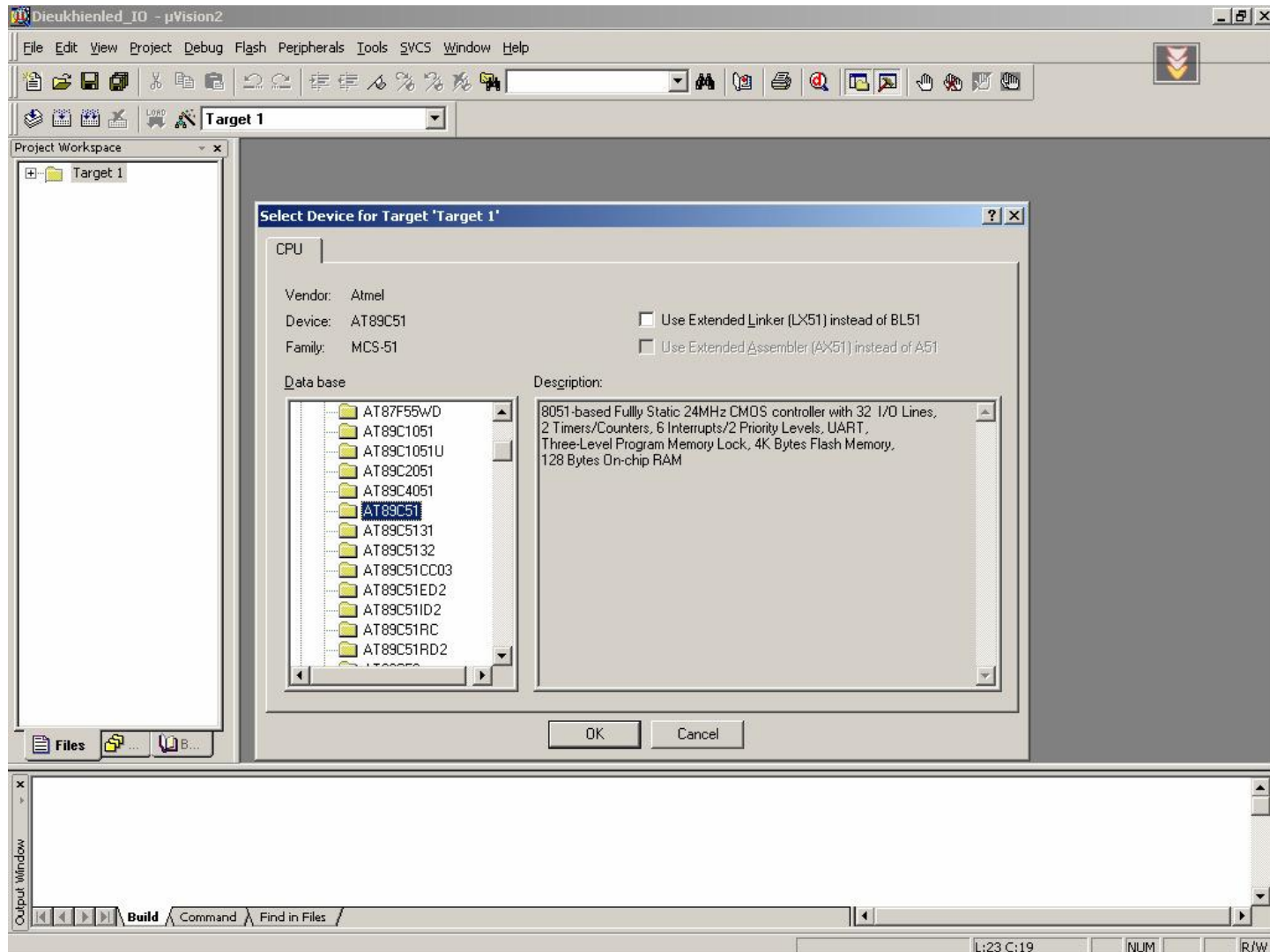
- Trong này có 1 loạt các hãng điện tử sản xuất 8051. Bạn lập trình cho con nào thì chọn con đó, kích chuột vào các dấu + để mở rộng các con IC của các hãng. Ở đây ta lập trình cho AT89C51 của hãng Atmel nên ta chọn như sau:



C4: Lập trình trên ARM

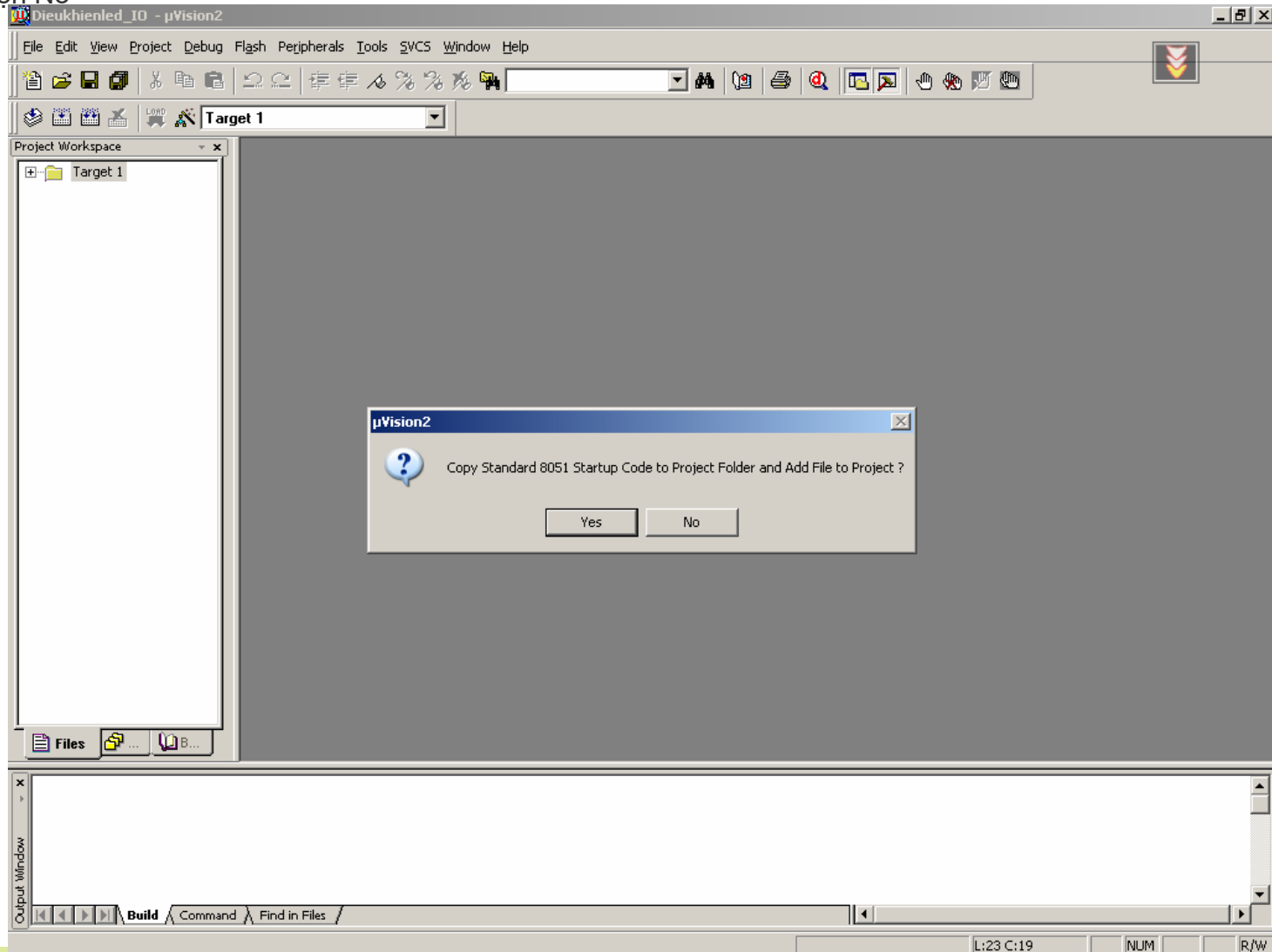
3. Keil tools cho lập trình nhúng trên ARM

- Chọn chip AT89C51



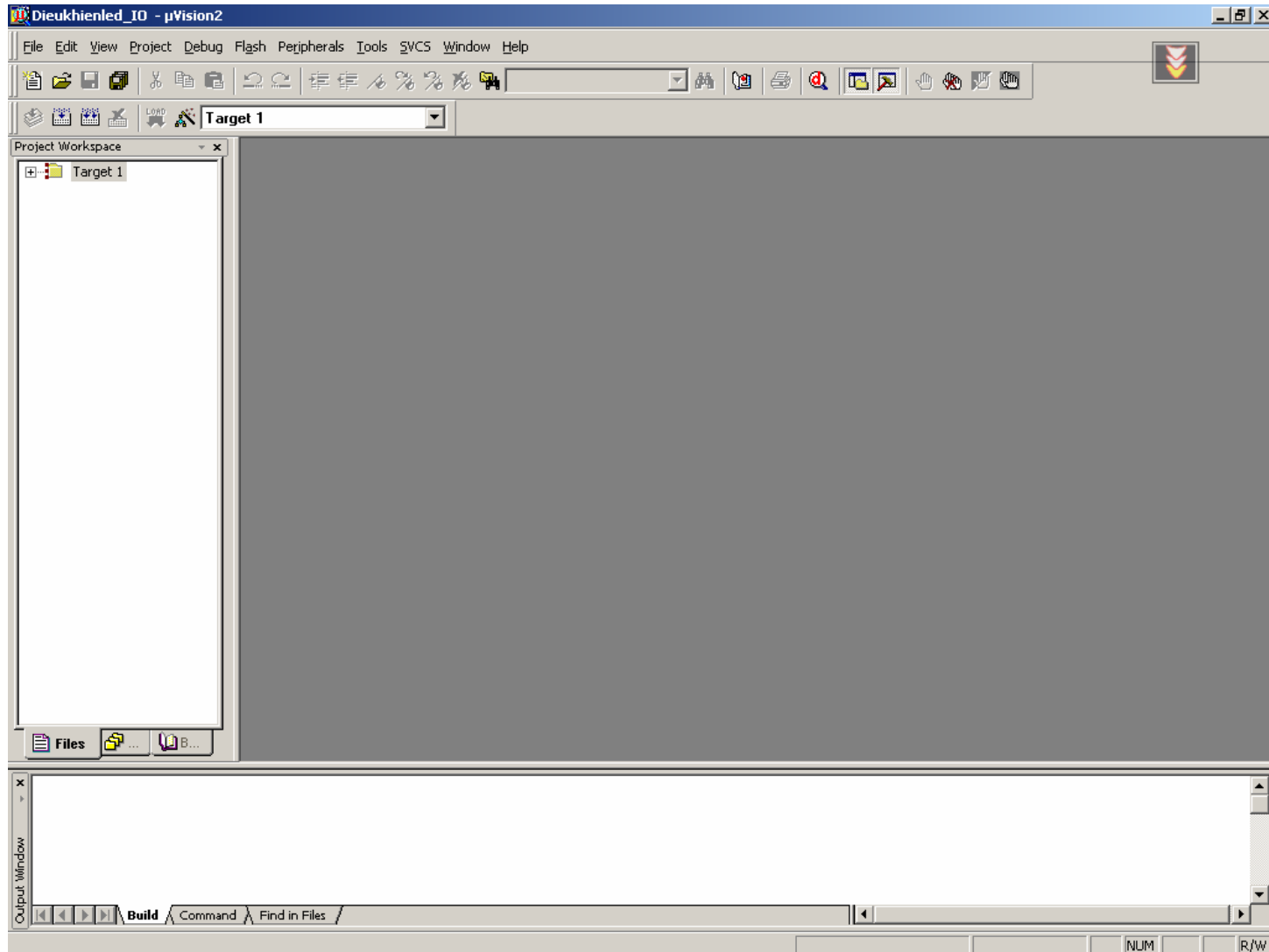
C4: Lập trình trên ARM

- Click chọn No



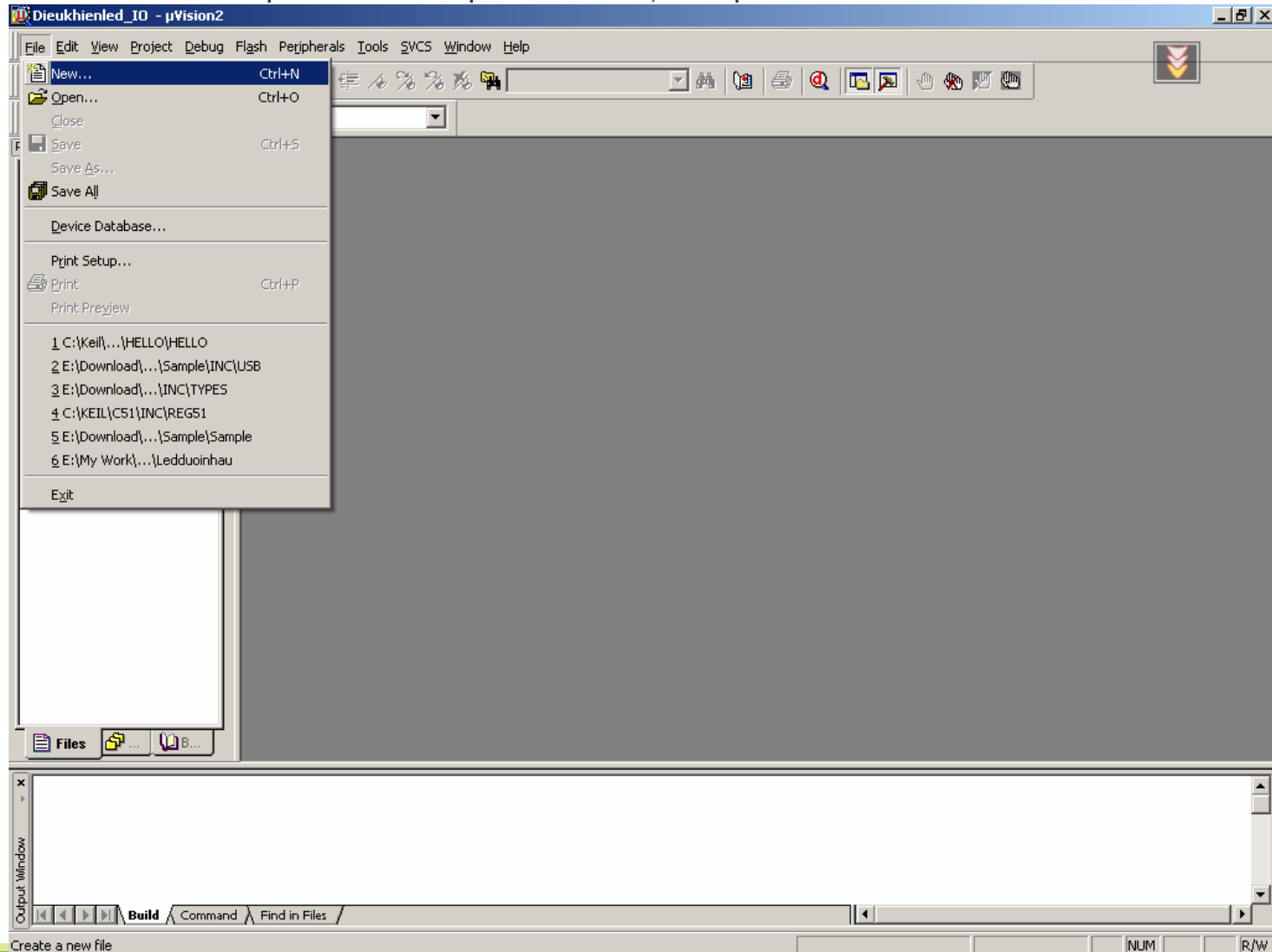
C4: Lập trình trên ARM

- Ta được cửa sổ như sau:



C4: Lập trình trên ARM

- Để tạo 1 file viết code ta chọn File - New hoặc nhấn Ctrl+N, ta được hình sau:



- E:\LapTrinhHeThongNhung\DeLTNhung_De2\thuchanh.uvproj - µVision4



Chương 5: Một số định hướng nghiên cứu



1 Hệ thống nhúng áp dụng điều khiển PID

2 Hệ thống nhúng áp dụng điều khiển mờ

3 Lập trình nhúng cho Robot di động

4 Lập trình nhúng cho xử lý tín hiệu số

5 Lập trình trên nền tảng Android

6 Lập trình trên .NET Micro Framework

C5: Một số định hướng nghiên cứu

1. Hệ thống nhúng áp dụng dk PID



- ❑ Một **bộ điều khiển vi tích phân tỉ lệ (bộ điều khiển PID)** là một cơ chế phản hồi vòng điều khiển (bộ điều khiển) tổng quát được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp – bộ điều khiển PID được sử dụng phổ biến nhất trong số các bộ điều khiển phản hồi.
Hiểu đơn giản nó là bộ điều khiển hệ thống. Gồm 3 khâu: Khuyếch đại, Tích phân, Vi phân.
- P:proportional(tỷ lệ): hệ thống luôn chấp hành và hoàn thành nhiệm vụ được giao ngay tức khắc, nhưng hậu quả mà nó để lại luôn là vấn đề nan giải: độ vọt và tần số dao động mang đi điều khiển những hệ thống có quán tính lớn thì không kiểm soát được.
- I:intergral(tích phân) :phương châm của hệ thống là "chậm mà chắc", cần cù, tỉ mỉ theo thời gian, hệ thống chạy độc lập triệt tiêu sai lệch.
- D:derivative(vi phân) : hệ thống có khả năng dự đoán tương lai, thích ứng tốt với sự thay đổi nhanh, tốc độ sai lệch thay đổi càng lớn thì đầu ra của D có tín hiệu càng lớn với mục đích chống lại sai lệch. Ưu điểm là giảm được đáng kể độ vọt.



C5: Một số định hướng nghiên cứu

1. Hệ thống nhúng áp dụng dk PID



- ❑ Đối với người lập trình nhúng phần lớn thường hay sử dụng một dòng chip xử lý thông dụng như 8051, AVR, PIC,... Vì vậy mỗi khi muốn nâng cấp hay nghiên cứu một sản phẩm mới, ngoài việc phải coi lại kiến trúc phần cứng mà một điều quan trọng nữa là làm thế nào để chạy được chương trình ứng dụng đang hoạt động tốt ở bộ vi xử lý cũ. Nắm bắt được hạn chế này, các nhà sản xuất phần cứng đã cho ra đời dòng vi xử lý ARM với những chuẩn phần cứng đáp ứng khả năng tương thích với phần mềm.
- ❑ Lõi ARM Cortex được chia ra thành ba dòng cấu hình chính:
 - A: Application hỗ trợ các ứng dụng có độ phức tạp cao như: máy tính, điện thoại di động.
 - R: Realtime hỗ trợ các ứng dụng cần tính toán xử lý thời gian thực.
 - M: Microcontroller hỗ trợ cho các ứng dụng công nghiệp và điện tử dân dụng.



C5: Một số định hướng nghiên cứu

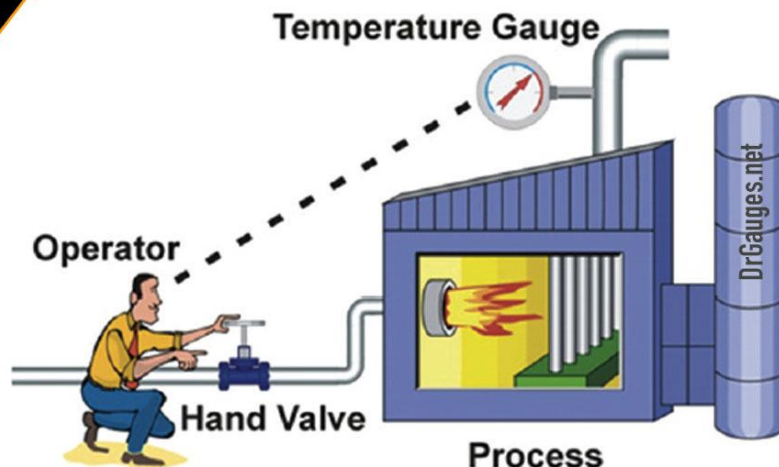
1. Hệ thống nhúng áp dụng dk PID



❖ Ví dụ điều khiển nhiệt độ lò hơi bằng tay

- Không có bộ điều khiển PID, điều khiển thủ công nhiệt độ nước là một quá trình tẻ nhạt. Ví dụ, để giữ nhiệt độ không đổi của nước xả ra từ lò sưởi đốt gas công nghiệp, người vận hành phải xem đồng hồ đo nhiệt độ và điều chỉnh van gas nhiên liệu phù hợp. Nếu nhiệt độ nước trở nên quá cao, người vận hành phải đóng van gas vừa đủ để đưa nhiệt độ trở về giá trị mong muốn. Nếu nước quá lạnh, anh ta phải mở van gas.

- Nhiệm vụ điều khiển được thực hiện bởi người vận hành được gọi là điều khiển phản hồi, bởi vì người vận hành thay đổi tốc độ đốt dựa trên phản hồi từ quá trình thông qua đồng hồ đo nhiệt độ. Người vận hành, van, lò sưởi (process) và đồng hồ đo nhiệt độ tạo thành một vòng điều khiển. Bất kỳ thay đổi nào của người vận hành đối với van gas đều ảnh hưởng đến nhiệt độ, được đưa trở lại cho người vận hành, nó là vòng lặp đóng.



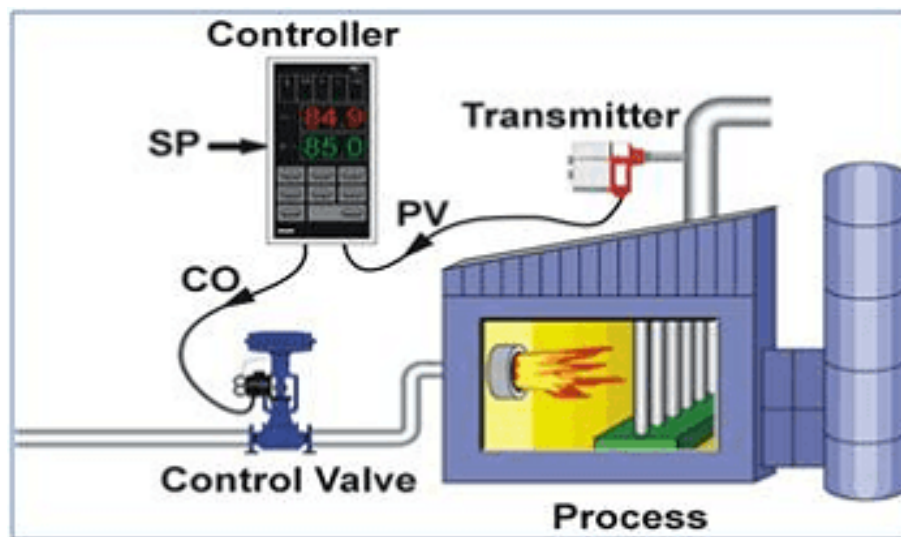
C5: Một số định hướng nghiên cứu

1. Hệ thống nhúng áp dụng dk PID



❖ Ví dụ điều khiển nhiệt độ lò hơi bộ điều khiển PID

- Một bộ điều khiển vi tích phân tỉ lệ (bộ điều khiển PID) là một cơ chế phản hồi vòng điều khiển (bộ điều khiển) tổng quát được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp – bộ điều khiển PID được sử dụng phổ biến nhất trong số các bộ điều khiển phản hồi. Một bộ điều khiển PID tính toán một giá trị "sai số" là hiệu số giữa giá trị đo thông số biến đổi và giá trị đặt mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào. Trong trường hợp không có kiến thức cơ bản về quá trình, bộ điều khiển PID là bộ điều khiển tốt nhất. Tuy nhiên, để đạt được kết quả tốt nhất, các thông số PID sử dụng trong tính toán phải điều chỉnh theo tính chất của hệ thống-trong khi kiểu điều khiển là giống nhau, các thông số phải phụ thuộc vào đặc thù của hệ thống.



C5: Một số định hướng nghiên cứu

2. Hệ thống nhúng áp dụng điều khiển mờ



❖ Ngày nay nhiều trang thiết bị được nhúng trong nó Logic mờ để cho việc sử dụng nó dễ dàng hơn, tiện lợi hơn. Chúng ta có thể tìm thấy logic mờ trong những camera, những nồi cơm điện, những máy hút bụi,....

-Như vậy có thể có một ý tưởng rằng chúng đã được làm như thế nào, chúng ta xem mô hình được đơn giản hóa cái này của một máy giặt ứng dụng logic mờ. Khi sử dụng một máy giặt, việc lựa chọn thời gian giặt dựa vào khối lượng quần áo, chất liệu vải và độ bẩn của mà quần áo có. Để tự động hóa quá trình này chúng ta sử dụng những phần tử sensor để phát hiện ra những tham số này (ví dụ thể tích quần áo, độ và kiểu chất bẩn, chất liệu vải).

- Thời gian giặt được xác định từ dữ liệu này. Không may, không dễ có cách công thức hóa một mối qua hệ toán học chính xác giữa thể tích quần áo và độ bẩn thời gian giặt cùng chất liệu vải. Chúng ta giải quyết vấn đề thiết kết bằng cách sử dụng logic mờ.



C5: Một số định hướng nghiên cứu

2. Hệ thống nhúng áp dụng điều khiển mờ



❖ Xây dựng bộ điều khiển mờ cho máy giặt

- Có 3 giá trị nhập vào (input) :

- Một cho độ bẩn trên quần áo.
- Một cho chất liệu vải.
- Một cho lượng quần áo cần giặt.

Ba đầu vào này thu được từ phần tử sensors quang học. Độ bẩn được xác định bởi sự trong suốt của nước. Mặt khác, loại chất bẩn được xác định từ sự bão hòa, thời gian nó dùng để đạt đến sự bão hòa. Quần áo dầu mỡ chẳng hạn cần lâu hơn cho sự trong suốt nước để đạt được sự bão hòa bởi vì mỡ là chất ít tan trong nước hơn những dạng khác của chất bẩn. Như vậy một hệ thống phần tử sensors khá tốt có thể cung cấp những giá trị vào (input) cần thiết được nhập vào cho bộ điều khiển mờ của chúng ta. Những giá trị cho độ bẩn và chất liệu vải là đã được chuẩn hóa được cho bởi giá trị phần tử sensors



C5: Một số định hướng nghiên cứu

2. Hệ thống những áp dụng điều khiển mờ



- ❖ Với biến ngôn ngữ Độ bền có các tập mờ
 - ☐ Bền ít (D.bền ít)
 - ☐ Bền vừa (D.banvừa)
 - ☐ Bền nhiều(D.quaban)
- ❖ Với biến ngôn ngữ khối lượng quần áo có các tập mờ
 - ☐ Nhẹ (K.nhẹ)
 - ☐ Trung bình (K.trung bình)
 - ☐ Nặng (K.nặng)
- ❖ Với biến ngôn ngữ chất liệu vải quần áo có các tập mờ
 - ☐ Mỏng (H.Mỏng)
 - ☐ Mỏng vừa (H.mỏngvua)
 - ☐ Dày (H.day)
- ❖ Với biến ngôn ngữ kết luận xác định lượng nước có các tập mờ
 - ☐ Ít (T.it)
 - ☐ Vừa (T.vua)
 - ☐ Cao (T.cao)
- ❖ Với biến ngôn ngữ kết luận lượng bột giặt có các tập mờ
 - ☐ Ít (G.it)
 - ☐ Vừa (G.vua)
 - ☐ Nhiều (G.nhiều)



C5: Một số định hướng nghiên cứu

2. Hệ thống nhúng áp dụng điều khiển mờ



❖ Với biến ngôn ngữ kết luận tốc độ quay có các tập mờ

- ☐ Chậm (N.chậm)
- ☐ Vừa (N.vua)
- ☐ Nhanh (N.Nhanh)

❖ Với biến ngôn ngữ kết luận xác định thời gian giặt có các tập mờ

- ☐ Rất nhanh (B.ratnhanh)
- ☐ Nhanh (B.Nhanh)
- ☐ Vừa (B.Vừa)
- ☐ Lâu (B.Lâu)
- ☐ Rất lâu (B.ratlau).



C5: Một số định hướng nghiên cứu

2. Hệ thống nhúng áp dụng điều khiển mờ



❖ Với biến ngôn ngữ kết luận tốc độ quay có các tập mờ

- ☐ Chậm (N.chậm)
- ☐ Vừa (N.vua)
- ☐ Nhanh (N.Nhanh)

❖ Với biến ngôn ngữ kết luận xác định thời gian giặt có các tập mờ

- ☐ Rất nhanh (B.ratnhanh)
- ☐ Nhanh (B.Nhanh)
- ☐ Vừa (B.Vừa)
- ☐ Lâu (B.Lâu)
- ☐ Rất lâu(B.ratlau).

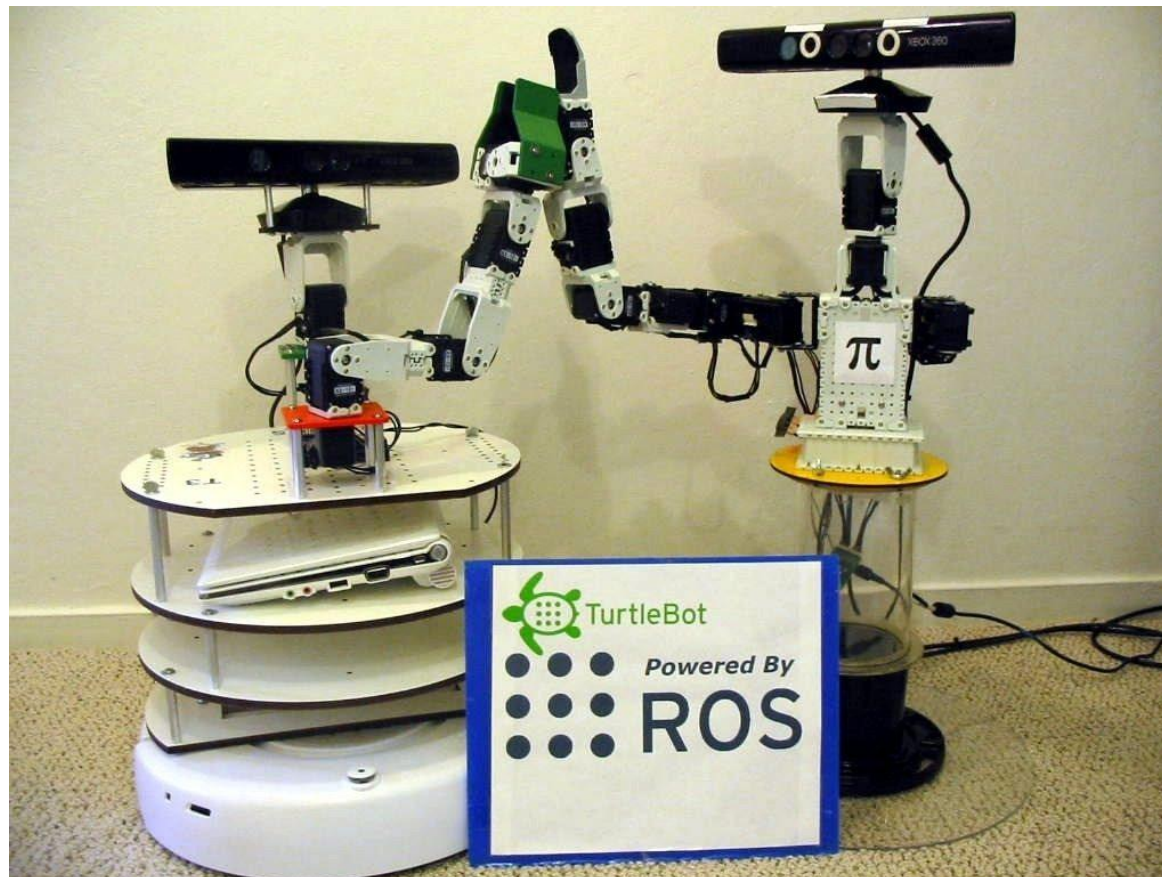


C5: Một số định hướng nghiên cứu

3. Lập trình nhúng cho Robot di động



- ❑ Robot là một cỗ máy được thiết kế làm một hoặc nhiều nhiệm vụ. Nói một cách đơn giản hơn Robot là một chiếc máy tính được lập trình có thêm động cơ, các chi tiết cơ khí để dàng di chuyển để thực hiện các nhiệm vụ của mình.



C5: Một số định hướng nghiên cứu

3. Lập trình nhúng cho Robot di động



- ❑ Một Robot gồm 3 phần là điện tử, cơ khí và lập trình: Điện tử giống như mạch máu trong cơ thể giúp kết nối các bộ phận, tín hiệu của robot tới bộ não của robot, trái tim của Robot chính là phần nguồn, tạo năng lượng giúp robot hoạt động. Cơ khí là các cơ cấu chấp hành giúp robot di chuyển, thực hiện các chuyển động theo thiết kế; cơ khí là các bộ phận giống như cơ bắp của con người, người làm cơ khí cần kỹ năng không gian, liên kết động tốt. Lập trình được ví như bộ não của Robot, là mảng công việc cần tư duy logic tốt kết hợp các tín hiệu điện tử cảm biến (sensor) để điều khiển các kết cấu cơ khí chuyển động nhịp nhàng theo nhiệm vụ đề ra..



C5: Một số định hướng nghiên cứu

3. Lập trình nhúng cho Robot di động



- ❑ Các công nghệ cần thiết để làm Robot:
 - Máy tính nhúng raspberry Pi 4: dòng máy tính nhúng, sản phẩm công nghệ ưu thích nhất thập niên này;
 - Node Red: Phần mềm mã nguồn mở của IBM giúp lập trình viên dễ dàng tiếp cận lập trình IoT/Robot;
 - Ngôn ngữ lập trình C/Python trên máy tính nhúng Pi;
 - K12 Maker: bộ mạch công suất, tín hiệu với 4 đầu ra động cơ một chiều, 4 động cơ servo, 10 cảm biến;
 - Nguồn 2600mAh Li-po 3S – 11.1V;
 - Động cơ DC và động cơ servo;
 - Cảm biến: cảm biến ánh sáng phân biệt màu sắc, cảm biến tiệm cận vật cản từ xa, công tắc hành trình – vật cản gần

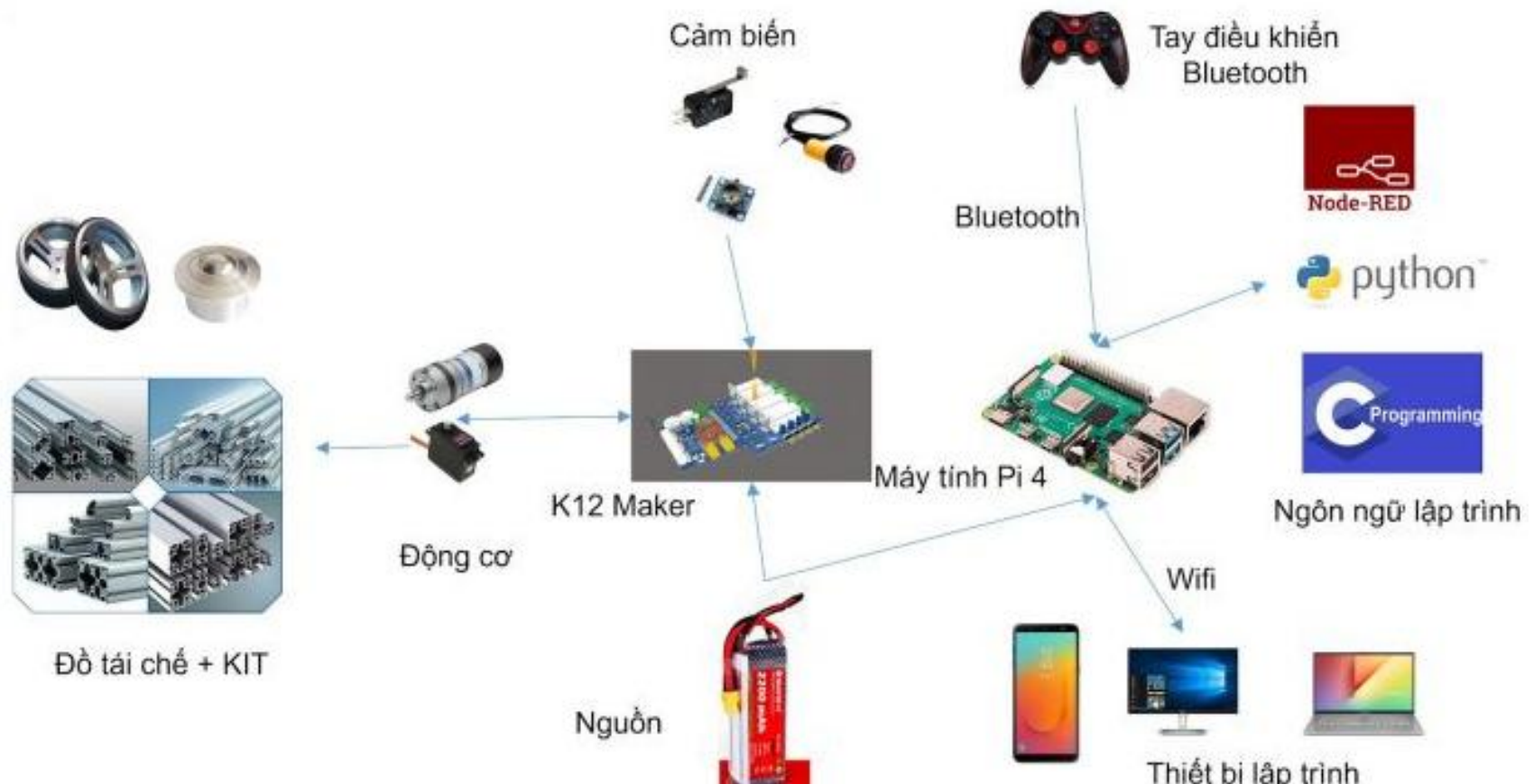


C5: Một số định hướng nghiên cứu

3. Lập trình nhúng cho Robot di động



❖ Các thiết bị cần thiết để lập trình Robot



C5: Một số định hướng nghiên cứu

3. Lập trình nhúng cho Robot di động



❖ Các công nghệ cần tìm hiểu để làm robot di động:

- Máy tính nhúng raspberry Pi 4: dòng máy tính nhúng, sản phẩm công nghệ ưa thích nhất hiện nay.
- Node Red: Phần mềm mã nguồn mở của IBM giúp lập trình viên dễ dàng tiếp cận lập trình IoT/Robot;
- Ngôn ngữ lập trình C/Python trên máy tính nhúng Pi.
- K12 Maker: bộ mạch công suất, tín hiệu với 4 đầu ra động cơ một chiều, 4 động cơ servo, 10 cảm biến.
- Nguồn 2600mAh Li-po 3S – 11.1V.
- Động cơ DC và động cơ servo.
- Cảm biến: cảm biến ánh sáng phân biệt màu sắc, cảm biến tiệm cận vật cản từ xa, công tắc hành trình – vật cản gần.



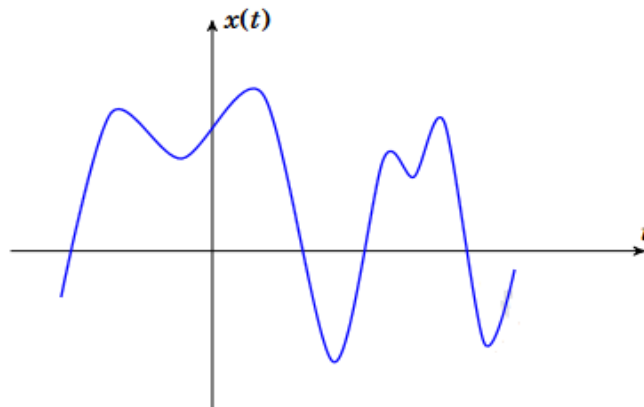
C5: Một số định hướng nghiên cứu

4. Lập trình nhúng cho xử lý tín hiệu số



❖ Tín hiệu là gì

Khi nghiên cứu một hiện tượng vật lý nào đó, người ta thường quan sát những đại lượng vật lý đặc trưng của hiện tượng này. Phương pháp quan sát chính thường dùng là đo lường. Các đại lượng vật lý được chuyển thành những dòng điện hay hiệu điện thế, được gọi là tín hiệu mà các máy đo có thể thu nhận được. Như vậy, thông tin đặc trưng của các đại lượng vật lý đang được quan tâm sẽ hoàn toàn được chứa đựng trong các tín hiệu này. Hình 1.1 biểu diễn một tín hiệu bằng một hàm toán học $x(t)$ biến thiên theo biến độc lập t .



Hình 1.1: Biểu diễn tín hiệu liên tục bằng hàm toán học.

Thông thường, t chỉ định thời gian, tuy nhiên tổng quan hơn t có thể có bất cứ dạng nào, và là một biến vô hướng hoặc là nhiều biến vô hướng độc lập (theo dạng véc-tơ).

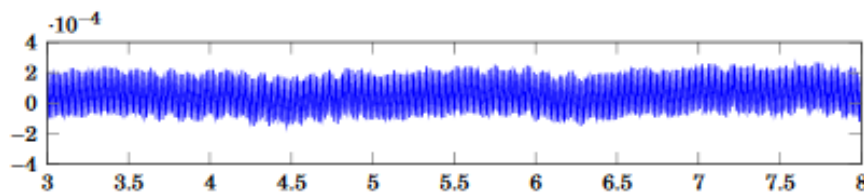
C5: Một số định hướng nghiên cứu

4. Lập trình nhúng cho xử lý tín hiệu số

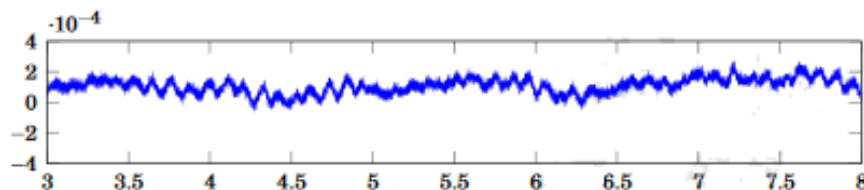


❖ Xử lý tín hiệu

Thiết kế một hệ thống để thu thập thông tin ta quan tâm chứa trong tín hiệu đầu vào $x(t)$ được gọi là xử lý tín hiệu. Như thế, nói một cách rất tổng quát, xử lý tín hiệu bao gồm tất cả những áp dụng mà chúng ta có thể hình dung, cần sử dụng tất cả các phương pháp luận hiện hữu trong điện tử, lọc tín hiệu, xử lý thông tin, lý thuyết nhận dạng, v.v...



(a) Dữ liệu EEG thô; 2048 mẫu



(b) EEG đã được loại bỏ tần số điện lưới 50 Hz; 2048 mẫu

Lọc một tín hiệu tức loại ra khỏi tín hiệu những thành phần được xem là nhiễu của tín hiệu.



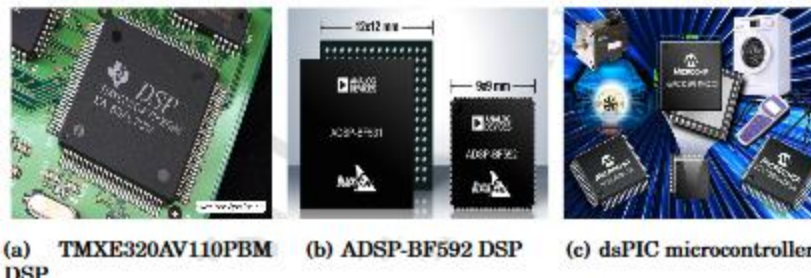
C5: Một số định hướng nghiên cứu

4. Lập trình nhúng cho xử lý tín hiệu số



❖ Công nghệ xử lý tín hiệu số

Hiện nay, người ta đã thiết kế những máy tính nhỏ đặc biệt để sử dụng cho xử lý tín hiệu có chất lượng tốt hơn rất nhiều và giá rẻ hơn rất nhiều so với các máy tính phổ cập. Những máy tính xử lý tín hiệu số này có tên là bộ vi xử lý tín hiệu(DSP)*. Bộ vi xử lý tín hiệu DSP là một bộ vi xử lý đặc biệt có cấu trúc được thiết kế một cách tối ưu để thực hiện nhanh chóng một số khối lượng tính toán lớn và phức tạp cần thiết cho các thuật toán xử lý tín hiệu số. Trong các thuật toán xử lý tín hiệu, phép tính cơ bản nhất là nhân rồi cộng và lưu giữ kết quả. Ngoài hoạt động tính toán, DSP cũng cần thường xuyên đọc dữ liệu đầu vào và viết dữ liệu đầu ra thật nhanh, vì hầu hết các áp dụng thực tế đều theo thời gian thực. Như thế nếu ta muốn bộ DSP có chất lượng cao, nó cần có một cấu trúc thích hợp, khác với một bộ vi xử lý bình thường.



Hình - Một số bộ DSP thông dụng của các hãng Texas Instruments, Analog Devices và Microchip.

C5: Một số định hướng nghiên cứu

5. Lập trình trên nền tảng Android



- ❑ Xu hướng phát triển các ứng dụng nhúng trên nền tảng Java (Android).

Triển khai và xác minh giải pháp cho phần mềm nhúng được sử dụng trong xe hơi như BSP (Gói hỗ trợ bảng), Phần mềm nhúng, HMI (Giao diện người máy), Cụm phần mềm Autosar.

Thiết kế, phát triển và xác minh hệ thống âm thanh, khởi động thụ động (điều khiển lái tự động), thông tin giải trí, viễn thông và an ninh mạng cho các đại gia trên toàn thế giới trong ngành công nghiệp ô tô.

Làm việc với nhiều loại vi điều khiển từ NXP, Renesas, Infineon, phát triển các giải pháp tiên tiến cho các mẫu ô tô khác nhau như BMW, Toyota, Honda, Nissan, Mitsubishi, GM, Volkswagen, Daimler, Geely, Hyundai, JLR



C5: Một số định hướng nghiên cứu

5. Lập trình trên nền tảng Android



- ❑ Các ứng dụng nhúng chạy trên nền tảng Java (Android) được sử dụng sau đây:
 - Trong các ngành máy tính, điện tử, viễn thông ngân hàng Hệ thống nhúng được sử dụng rộng rãi
 - Hệ thống nhúng được tích hợp cả phần mềm và phần cứng, tích hợp được 1 thiết bị lập trình như vi xử lý.
 - Hệ thống nhúng và PC, thiết bị cầm tay PDA có sự khác biệt là do sự thiết kế để chuyên biệt hóa chức năng cụ thể nhằm tăng chất lượng cho hệ thống giảm thiểu giá thành.



C5: Một số định hướng nghiên cứu

5. Lập trình trên nền tảng Android



❑ Hai hướng đi cho ứng dụng nhúng chạy trên nền tảng Java (Android

- **Embedded software**

Phát triển phát triển các sản phẩm phần mềm cho các sản phẩm nhúng, như là application (web, desktop hay mobile app), firmware, OS (hệ điều hành), driver, V...V.

- **Embedded hardware**

Thiết kế board mạch hay còn gọi là thiết kế PCB, test board mạch. Công việc này đòi hỏi phải rất giỏi về phần cứng và điện tử.



C5: Một số định hướng nghiên cứu

5. Lập trình trên nền tảng Android



- Một số phần mềm nhúng chạy trên hệ điều hành Android

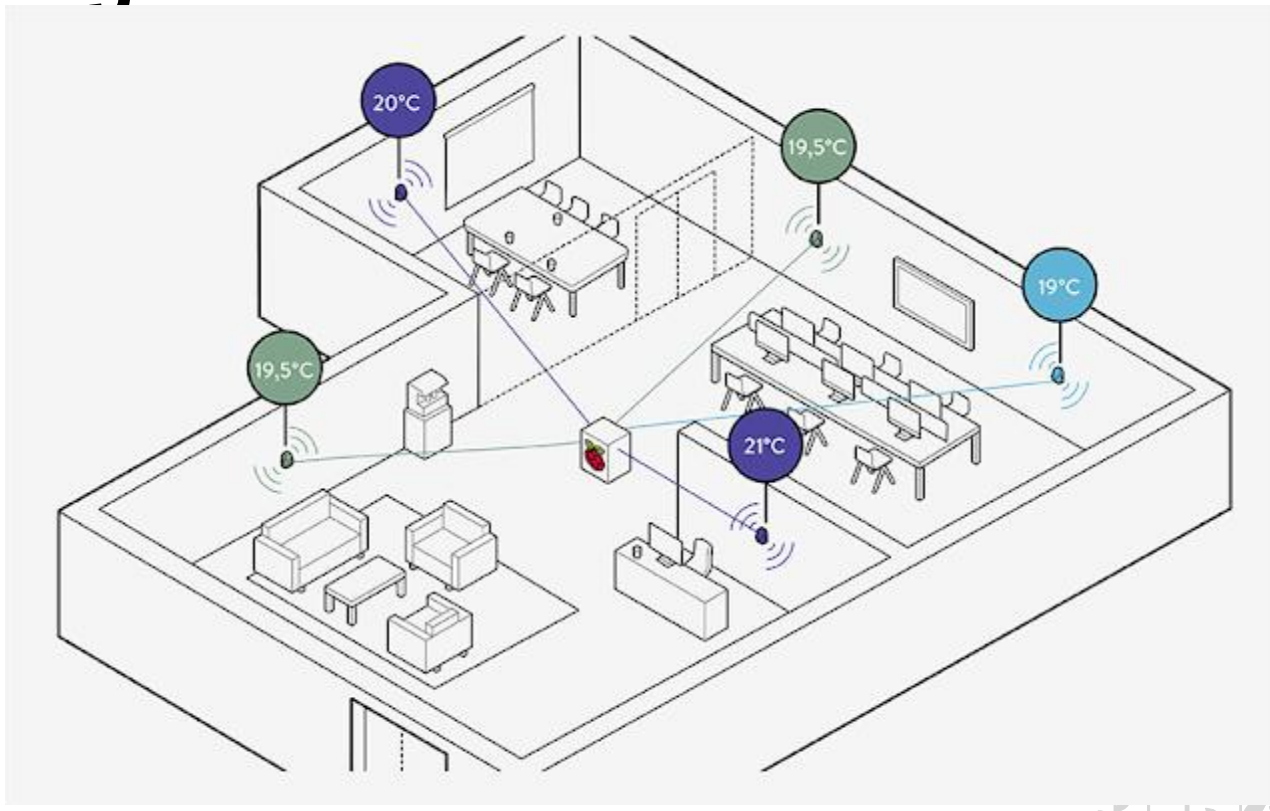


C5: Một số định hướng nghiên cứu

5. Lập trình trên nền tảng Android



- **Hệ thống định vị trong nhà thời gian thực với iBeacons + Raspberry PI + Android Things IoT**

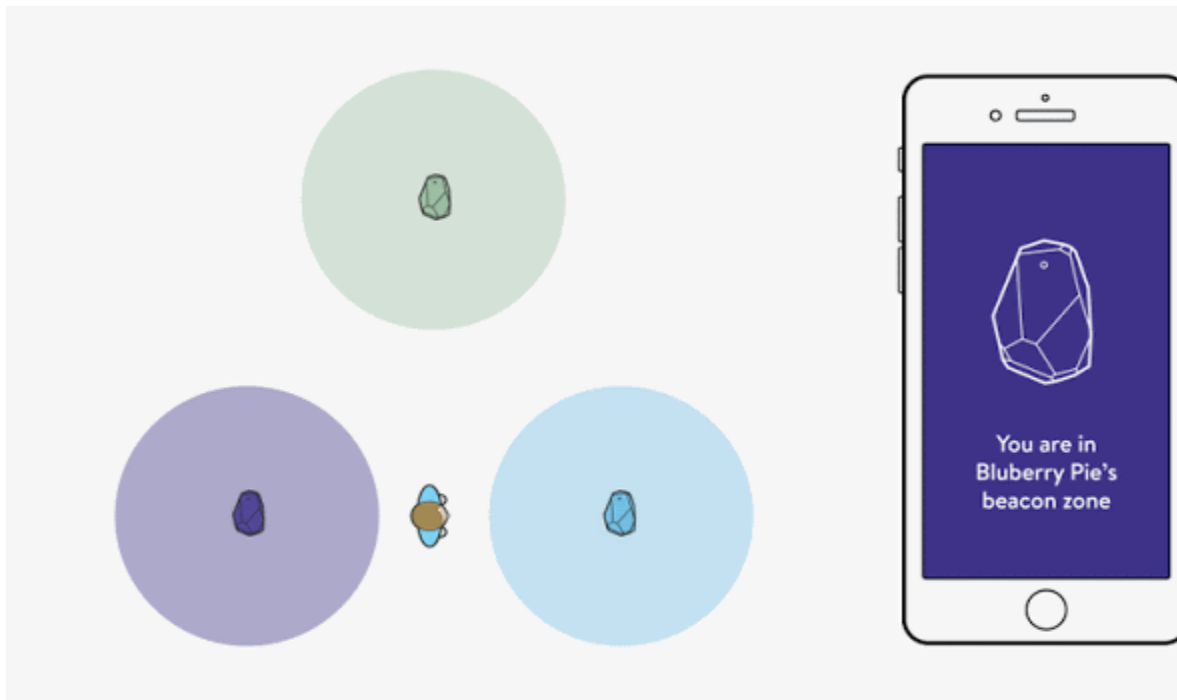


C5: Một số định hướng nghiên cứu

5. Lập trình trên nền tảng Android



- Ứng dụng quét tín hiệu BLE từ các thiết bị Beacons để phát hiện vị trí (proximity), hoặc là đọc các dữ liệu cảm biến. Sau đó bạn sẽ tổng hợp các thông tin về môi trường thực để xây dựng một dự án IoT hoàn chỉnh (như hệ thống định vị thời gian thực RTLS).



C5: Một số định hướng nghiên cứu

6. Lập trình trên .NET Micro Framework



- ❑ Từ rất sớm, các "ông lớn" như IBM, Microsoft, Intel đã chuyển hướng một số bộ phận nghiên cứu phát triển của mình sang làm hệ thống nhúng. Microsoft, tên tuổi gắn liền với các chương trình cho Desktop, cũng không thể đứng ngoài cuộc. Sản phẩm nhúng điển hình của hãng này là hệ điều hành Windows CE, máy chơi game Xbox...
- ❑ Ở các hệ nhúng của Microsoft, chức năng xử lý tính toán được ứng dụng cụ thể cho các thiết bị vật lý như điện thoại di động, đồ điện tử cầm tay, thiết bị y tế, các phương tiện vận tải thông minh, máy đo, thiết bị gia dụng thông minh...



C5: Một số định hướng nghiên cứu

6. Lập trình trên .NET Micro Framework



- ❑ Phần mềm do Microsoft sản xuất được cài sẵn vào các sản phẩm và chúng được sử dụng ngay cùng với đồ điện tử đó mà không cần có sự cài đặt của người sử dụng hay người thứ ba.
- ❑ Các phần mềm nhúng được cài đặt có trong các sản phẩm như: TV, tủ lạnh, máy giặt, điều hòa nhiệt độ, điện thoại... hay bất kỳ vật dụng, thiết bị điện tử hiện đại nào.



Tài liệu tham khảo



- [1] Embedded Systems Architecture, Tammy Noergaard, Newnes, Elsevier Inc, 2005.
- [2] Programming Embedded Systems I, II, Michael J. Pont, University of Leicester, 2006.
- [3] SPECIFICATION AND DESIGN OF EMBEDDED SYSTEMS, Daniel D - Gajsk - Frank Vahid - Sanjiv Narayan - Jie Gong, University of California at Irvine Department of Computer Science Irvine, CA 92715-3425, 1994.
- [4] Embedded Systems Design, Second edition, Steve Heath, Newnes An imprint of Elsevier Science Linacre House, Jordan Hill, Oxford OX2 8DP 200 Wheeler Road, Burlington MA 01803, 2003.
- [5] Android™ A Programmer's Guide, J.F. DiMarzio, McGraw-Hill Companies , 2008.
- [6] EMBEDDED SYSTEMS PROGRAMMING, Guillermo A Francia, MCIS Department Jacksonville State University 700 Pelham Rd North Jacksonville, AL 36265 Consortium for Computing in Small Colleges, 2001.
- [7] Lessons "Introduction to Programming Embedded Systems", Sebastian Fischmeister, Department of Computer and Information Science University of Pennsylvania.
- [8] <http://www.glib.hcmus.edu.vn> (Thư viện Đại học Khoa Học Tự Nhiên TP. HCM)