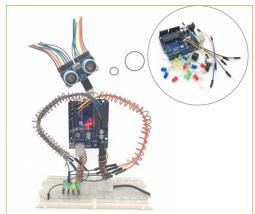


LẬP TRÌNH HỆ NHÚNG





Mai Cường Thọ

GIỚI THIỆU LẬP TRÌNH HỆ THỐNG NHÚNG

Chương 1

Mai Cường Thọ

Đánh giá

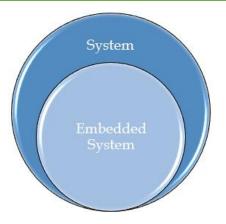


- Thi (50%)
 - Làm project, nhóm (5-7 sv), báo cáo tại buổi thi (quyển báo cáo + sp)
 - Hoặc thi lập trình trên máy, 75 phút
- Quá trình (50%)
 - Đánh giá qua các bài thực hành, trên các hệ thống giả lập: TinkerCad, App Inventer, Proteus (quyển báo cáo, và kho bài tập)



1.1.1 Khái niệm Hệ thống nhúng

- Là hệ thống có:
 - phần cứng và
 - phần mềm máy tính
 - nhúng "ẩn" trong nó



- Là thiết bị có một máy tính có thể lập trình được, nhưng không phải là máy tính đa năng
- HTN Phục vụ các bài toán chuyên dung trong nhiều lĩnh vực công nghiệp, tự động hoá điều khiển, quan trắc và truyền tin.

1.1.1. Hệ thống nhúng ...

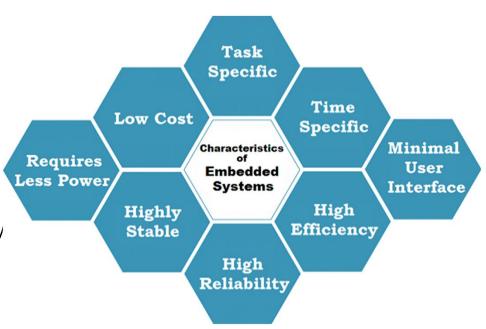
Một số hệ thống nhúng





1.1.1. Hệ thống nhúng ..

- Một số đặc điểm của các hệ thống nhúng:
 - Hoạt động ổn định
 - Tiết kiệm năng lượng
 - và có tính năng tự động hoá cao.
 - Thiết kế chuyên dụng
 - Thuật toán chuyên dụng
 - Giao diện người dùng
 - Một số hệ thống đòi hỏi ràng buộc v bảo độ an toàn và tính ứng dụng.
 - Hiệu năng/ Giá cả/ Kích thước
 - Tiêu hao năng lượng
 - Độ tin cây

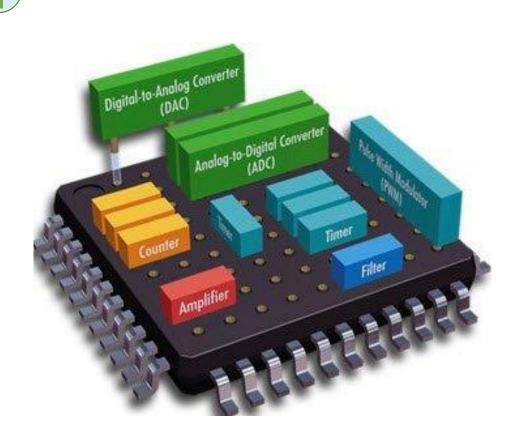


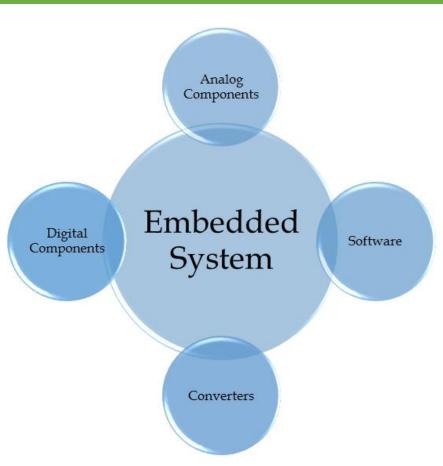


- 1.1.1. Hệ thống nhúng ..
- Một số ràng buộc thiết kế hệ thống nhúng:
 - Chi phí thấp
 - Một bộ xử lý phức tạp có thể làm tăng giá HTN
 - Tiêu hao năng lượng thấp
 - Nhiều HTN hoạt động dựa trên Pin
 - Bộ nhớ giới hạn
 - Đáp ứng thời gian thực
 - Do hầu hết các HTN được sử dụng để điều khiển thiết bị
 - Phải tạo ra các đáp ứng trong một khoản thời gian nhất định

1.1. Tổng quan Hệ thống nhúng

1.1.1. Hệ thống nhúng - Các thành phần cơ bản

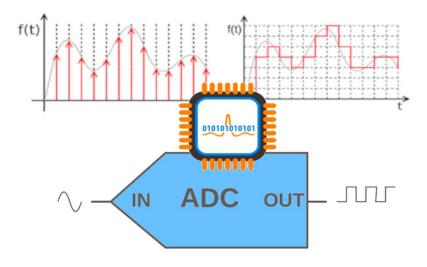




1.1.1. Hệ thống nhúng - Các thành phần cơ bản ..

- Analog Components: giúp tương tác với thế giới thực
 - Các bộ cảm biến (Sensors)
 - Các thiết bị truyền động (Actuators)
 - Các bộ điều khiển (Controllers)
- Converters: cho phép các thành phần Số và Tương tự làm việc với nhau
 - Bộ chuyển đổi Tương tự Số: Analog to Digital Converter (ADC).
 - Bộ chuyển đổi Số Tương tự: Digital to Analog Converter (DAC).





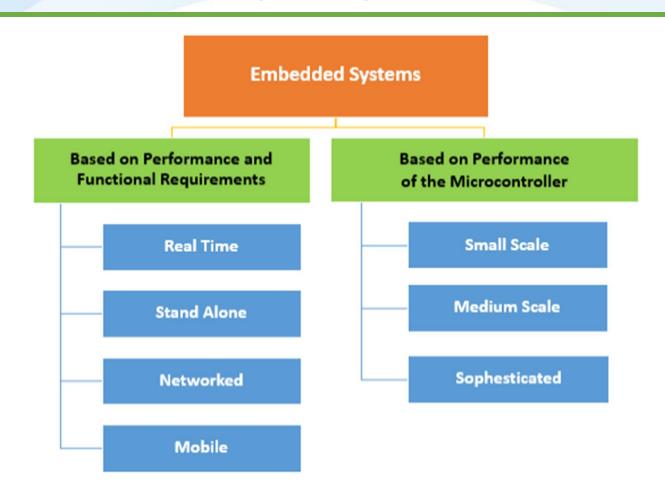
1.1.1. Hệ thống nhúng - Các thành phần cơ bản ..

- Digital Components: nằm trong chip, thực hiện các thao tác xử lý
 - Bộ xử lý: Processors.
 - Bộ đồng xử lý: Co-processors.
 - Bộ nhớ: Memory.
 - Bộ điều khiển: Controllers.
 - Buses.
 - Các mạch tích hợp ứng dụng cụ thể: Application Specific Integrated Circuits (ASIC).

Software

- Phần mềm điều khiển
- Phần mềm hệ thống

1.1.2. Phân loại hệ thống nhúng





1. Real Time

- The first type of embedded system according to functional requirement is real time embedded system.
- Such systems perform operations and give output at specific times.
- For completing the task assigned they follow time constraint.

2. Stand Alone

- Stand alone embedded systems are the most common in use.
- These are the systems which can work on their own without using an external host.
- These devices take input by its own input ports, process data and give output.

One example of stand alone system is a camera.

1.1.2. Phân loại HTN theo Yêu cầu về hiệu năng và chức năng..

Networked

- This is the fastest growing type of embedded systems.
- These systems are connected with network that could be LAN, WAN or internet.
- The connection can be wireless or wired.
- Such systems use network to access all the resources.

4. Mobile

- This is the class of embedded systems which is used in portable devices.
- The examples of devices are mobile phones, cameras, music players etc.

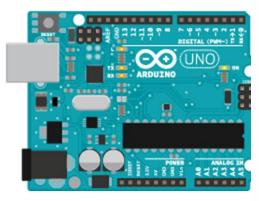
1.1.2. Phân loại HTN theo mức độ tích hợp

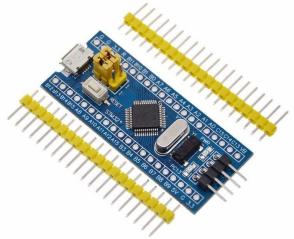
1. Small Scale Embedded Systems

- HTN có vi điều khiển được nhúng là vi điều khiển 8 bit hoặc 16bit
- Phần cứng có độ phức tạp nhỏ nhất
- Những HTN như vậy thường sử dụng vi điều khiển PIC, vđk 8051 hoặc các board mạch tích hợp vđk (Arduino Kit, STM32-F1x











2. Medium Scale Embedded Systems

- Sử dụng 1 hoặc hiều vđk 16, 32 bit
- Nó có thể sử dụng DSP (digital signal processor) hoặc máy tính RISC (reduced instruction set computer).
- Phần cứng và phần mềm hệ thống này phức tạp

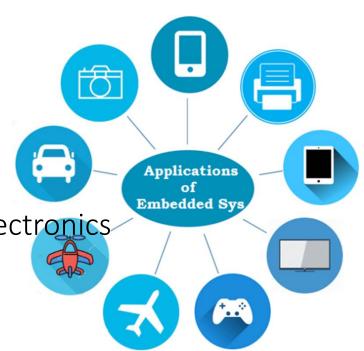
3. Sophisticated Embedded Systems

- The third class of embedded systems is sophisticated.
- Such systems have huge hardware and software complexity.
- So they need PLA (programmable logic array), scalable or configurable processors.
- These systems have speed constraints.

9/6/2021 15

1.1.3. Ứng dụng của HTN

- Embedded Systems in Automobiles
 - Motor Control System
 - Cruise Control System
 - Engine or Body Safety
 - Robotics in Assembly Line
 - Car Entertainment, Car multimedia
 - Mobile and E-Com Access
- Embedded Systems in Digital Consumer Electronics
 - DVDs
 - Set top Boxes
 - High Definition TVs
 - Digital Cameras

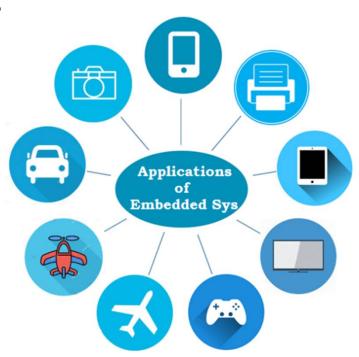


9/6/2021

16

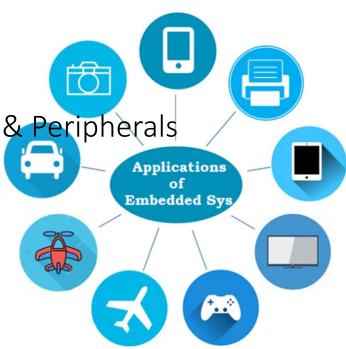
1.1.3. Ứng dụng của HTN ..

- Embedded systems in Telecommunications
 - Mobile computing
 - Networking
 - Wireless Communications
- Embedded Systems in Smart Cards
 - Banking
 - Telephone
 - Security Systems



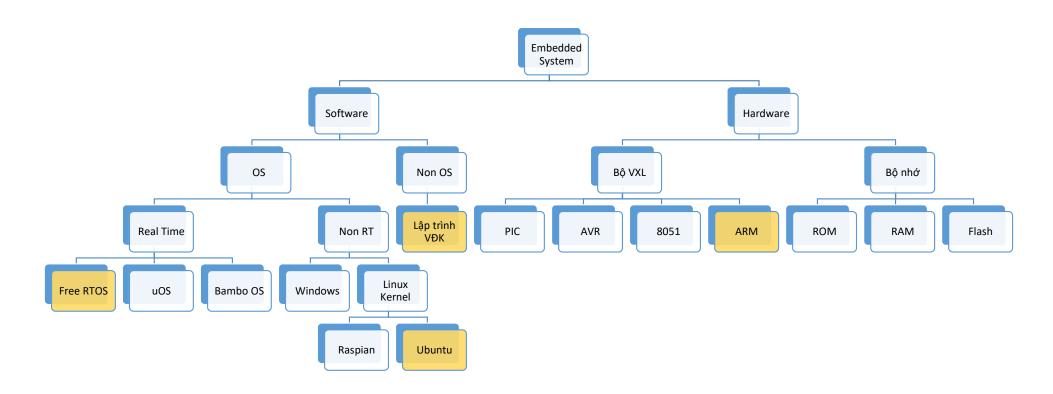
1.1.3. Ứng dụng của HTN

- Embedded Systems in Missiles and Satellites
 - Defense
 - Aerospace
 - Communication
- Embedded Systems in Computer Networking & Peripherals
 - Networking Systems
 - Image Processing
 - Printers
 - Networks Cards
 - Monitors and Displays



1.1. Tổng quan 1.1.4. Cây phát triển HTN





1.1.4. Lập trình hệ thống nhúng

- Lập trình ứng dụng trên hệ nhúng **phụ thuộc vào**
 - nền tảng (platform) phần cứng,
 - phần mềm của hệ nhúng đó.
- Hệ nhúng không có hệ điều hành:
 - Thường sử dụng các vi điều khiển hiệu năng tương đối thấp (8051, ATMega, PIC, ARM7, ...)
 - Lập trình bằng C, ASM
 - Môi trường, công cụ lập trình tùy theo từng dòng vi điều khiển (CodeVision, AVR Studio, Keil C...)
 - Phù hợp các ứng dụng điều khiển vào/ra cơ bản, các giao tiếp ngoại vi cơ bản.



1.1.4. Lập trình hệ thống nhúng..

Hệ nhúng có hệ điều hành:

- Dựa trên các vi điều khiển, vi xử lý (CPU) có hiệu năng cao (Ví dụ: AVR 32, ARM 9, ARM 11, ...)
- Nhiều nền tảng hệ điều hành nhúng : uCLinux, Embedded Linux,
 Windows CE, ...
- Môi trường, công cụ lập trình tùy thuộc nền tảng hệ điều hành:
 C/C++, QT SDK (Nokia), .Net Compact FrameWork (Microsoft), ...
- Úng dụng nhiều bài toán phức tạp: GPS Tracking/Navigator, Xử lý ảnh, ứng dụng Client/Server, ...

1.1.4. Lập trình hệ thống nhúng ...

- Các thiết bị di động thông minh:
 - Xu hướng công nghệ hiện nay
 - Nhiều nền tảng: iOS, Android, Windows Phone, Symbian OS/Memo
 - Môi trường, công cụ:
 - iOS: Xcode + iOS SDK (ngôn ngữ Object-C)
 - Android: C, Java + Android SDK, Eclipse/Netbean
 - Windows Phone: SDK + Visual Studio (C#)
 - Các ứng dụng phong phú: Google Play Store, Apple Store, Windows Market Place, ...

1.1.4. Lập trình hệ thống nhúng ..

Môn học này hướng tới 2 chủ đề chính:

- 1) Nhập môn LTN trên Arduino Uno Kit Lý do: đơn giản, dễ tiếp cận, phổ biến
- 2) Lập trình hệ nhúng nền tảng ARM + Linux Minh họa trên Kit STM32Fx--

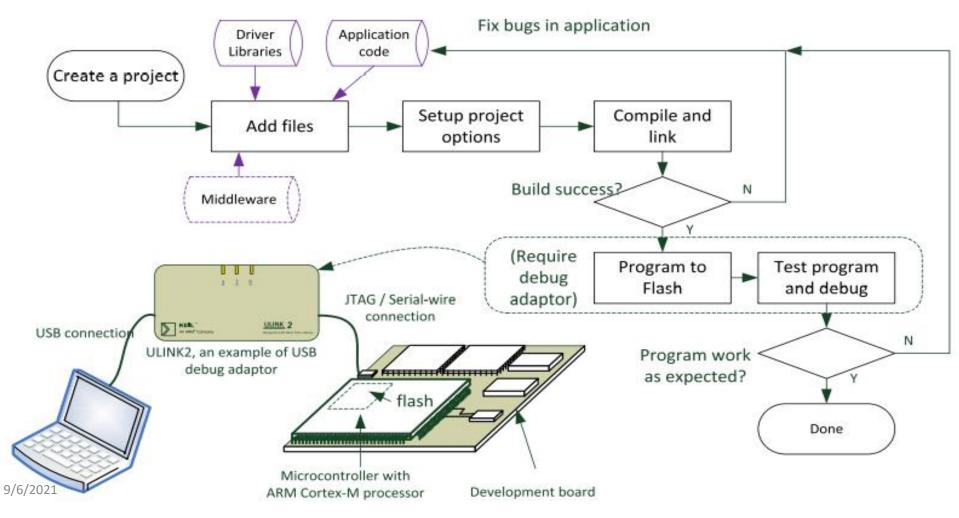
Lý do:

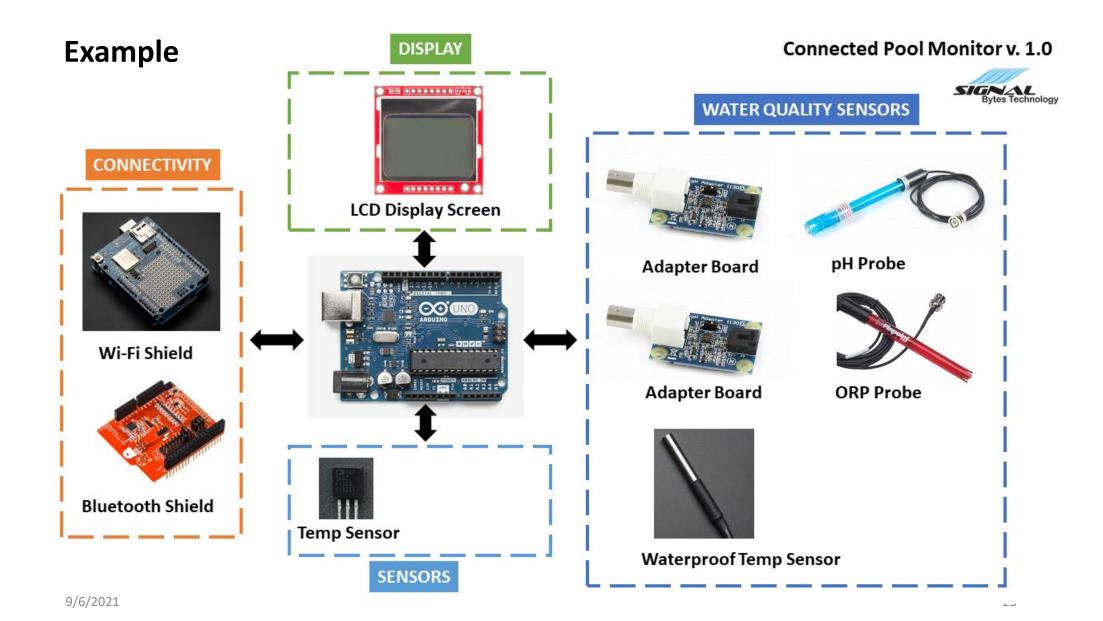
- ARM ? > 90% thị phần thiết bị nhúng, dòng vi điều khiển hiệu năng cao.
- Embedded Linux? Mã nguồn mở, khả năng can thiệp, hiểu sâu hệ thống. Nhiều OS khác (iOS, Android) dựa trên Linux Kernel



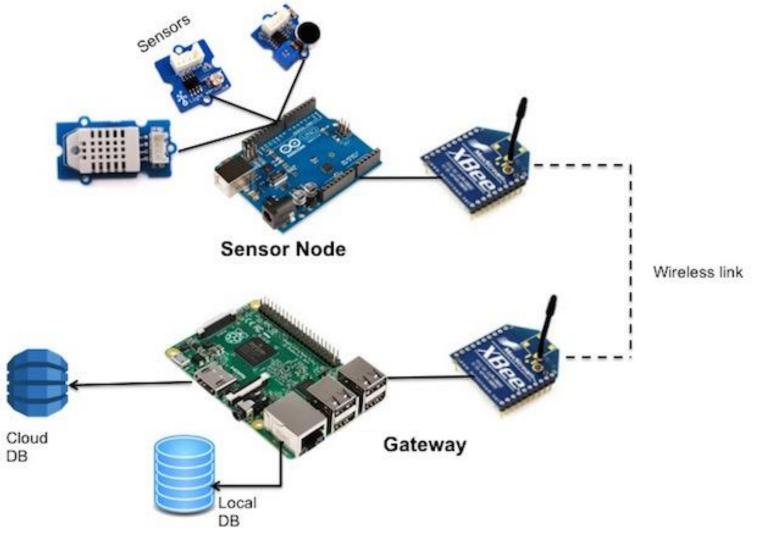
A simple Development flow





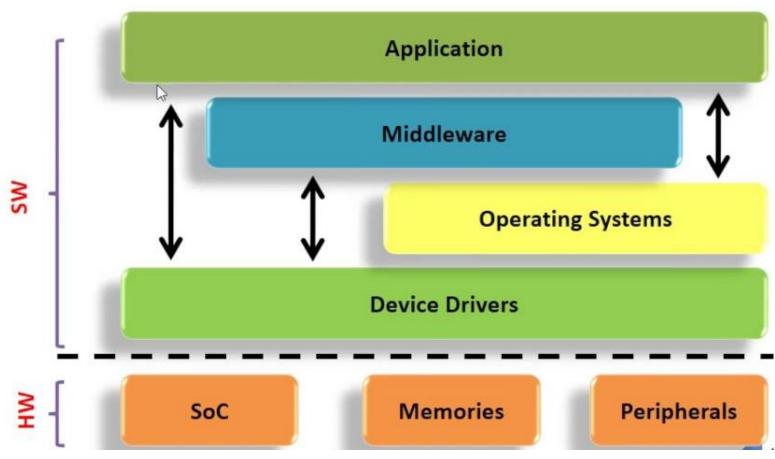


Example



1.2. Kiến trúc phân tầng HTN



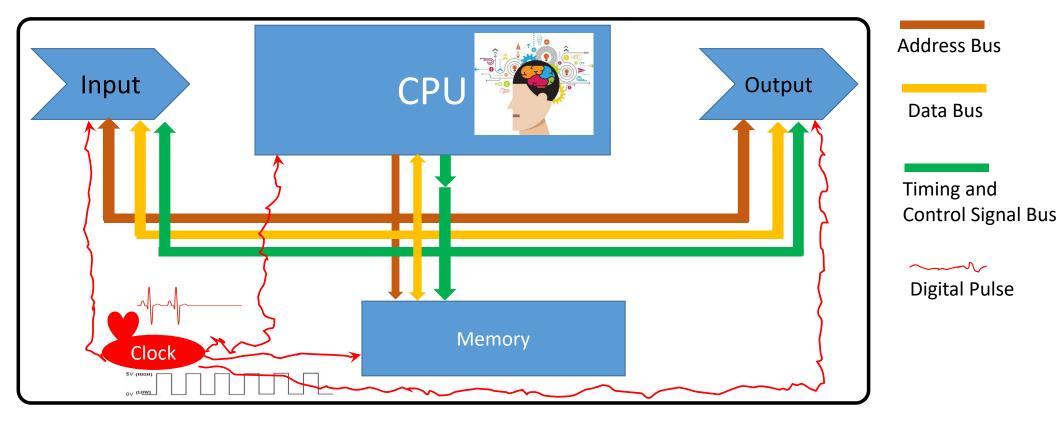


2. Phần cứng HTN

2.1. Tổng quan phần cứng HTN



Sơ đồ khối Phần cứng HTN



2.1. Tổng quan phần cứng HTN...



- CPU: Khối xử lý trung tâm

Memory: Bộ nhớ chứa chương trình, dữ liệu của hệ thống

Ngoại vi

 Input: Đọc thông tin từ các cảm biến nối tạ ngõ vào, nhận tín hiệu điều khiển từ người dung, ...

• Output: Điều khiện thiết bị kết nối tại ngõ ra, hiển thị thông tin, ...

BUS: Liên kết hệ thống, giúp gắn kết và truyền tín hiệu qua lại giữa các thành phần với nhau.

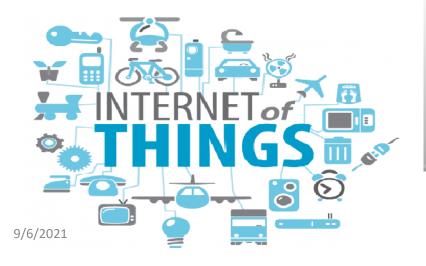
9/6/2021 30

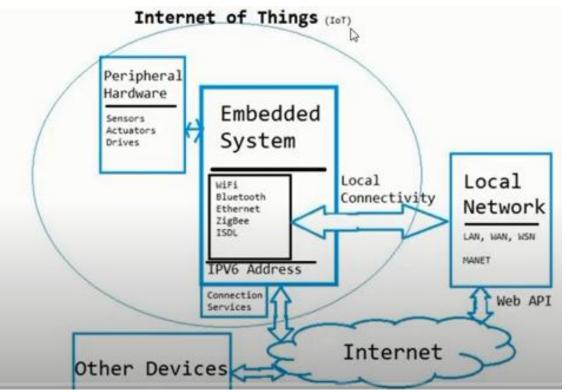
2.1. Tổng quan phần cứng HTN..



Hệ thống nhúng hiện đại

- Có thêm khối Kết nối mạng
- Nhằm liên kết HTN với các hệ thống khác, và với thế giới bên ngoài
- Để xây dựng lên một mạng lưới lớn hơn → IoT

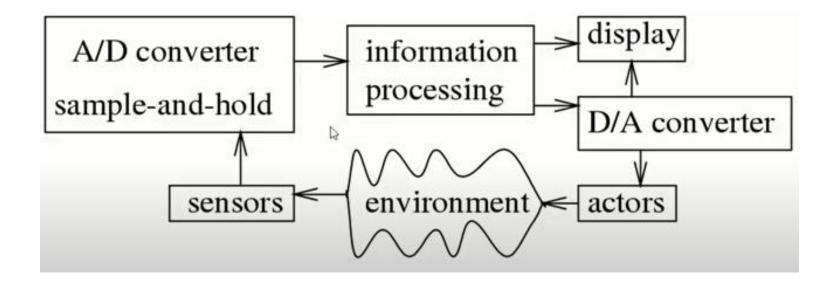




2.1. Tổng quan phần cứng HTN..



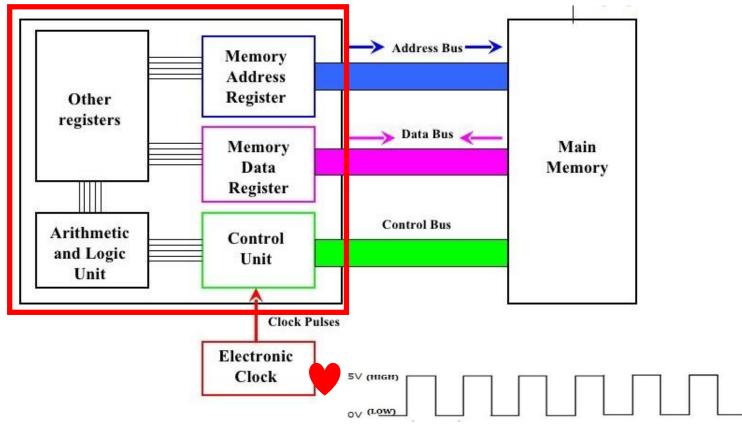
Phần cứng HTN hoạt động như một chu trình khép kín





2.2.1. Sơ đồ khối cơ bản của CPU





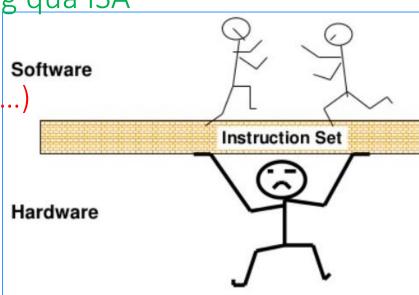
2.2.2. Kiến trúc tập lệnh (ISA – Instruction Set Architecture)

Phần mềm giao tiếp với Phần cứng thông qua ISA

■ ISA bao gồm

Instructions (type, encoding, operation |...)

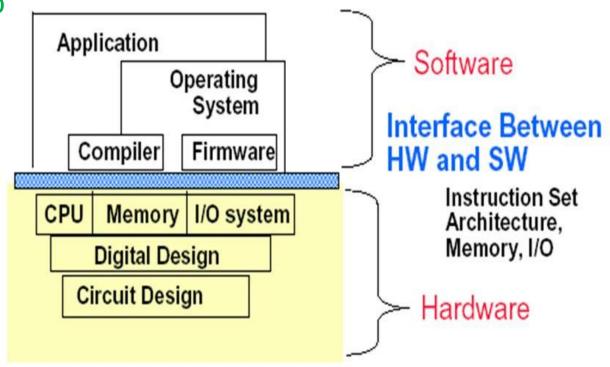
- Memory and I/O access
- Registers (Why do we need registers!)





2.2.2. Kiến trúc tập lệnh ...

- Mỗi bộ xử lý có kiến trúc tập lệnh khác nhau:
 - IA-32, MIPS, SPARC, ARM, DEC, HP, IBM, ...
- Cùng một ISA, lại có thể được thực theo các cách khác nhau để đạt được các cục tiêu khác nhau
 - single-cycle, multi-cycle, pipelining, ...





2.2.2. ISA (tiếp)

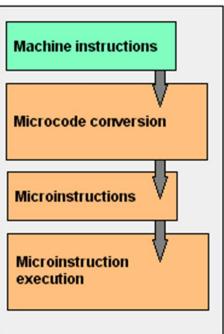
CISC: Complex Instruction Set Computing

RISC: Reduce Instruction Set Computing

Machine instructions Instruction execution

CIS Computer	RIS Computer
Many instructions and addressing modes	Few instructions and addressing modes
Instructions have different levels of complexity (different size and execution time)	Simple instructions of fixed size
Shorter programs	Longer programs
Relatively slow	Relatively fast
Expensive !!! (not really)	Cheaper !! (not really)

CISC

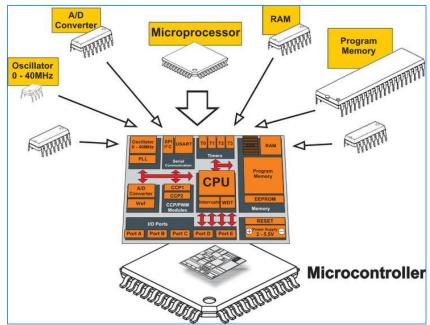


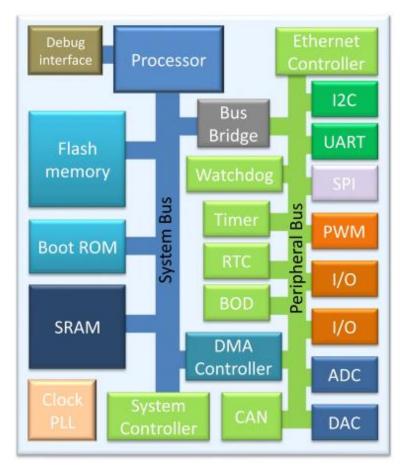
2.2. Bộ xử lý trung tâm (CPU)



2.2.3. Khác với Vi điều khiển?

- Mạch tích hợp đầy đủ các phần
 - Bộ xử lý + Bộ nhớ + Ngoại vi (GPIO, Timer,..)





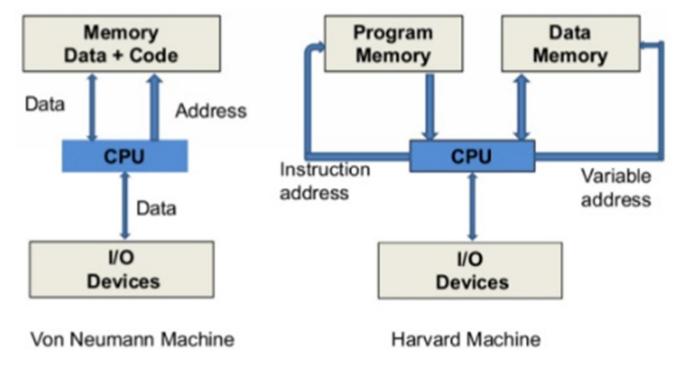
SO SÁNH MỘT SỐ VI ĐIỀU KHIỂN



	8051	PIC	AVR	ARM	
Chiều rộng bộ nhớ	8-bit cho lõi tiêu chuẩn	8/16/32-bit	8/32-bit	Chủ yếu 32-bit và cũng có 64-bit	
Giao tiếp	UART, USART, SPI, I2C	PIC, UART, USART, LIN, CAN, Ethernet, SPI, I2S	UART, USART, SPI, I2C, CAN, USB, Ethernet)	UART, USART, LIN, I2C, SPI, CAN, USB, Ethernet, I2S, DSP,	
Tốc độ	12 Clock / Instruction cycle	4 Clock / Instruction cycle	1 Clock / Instruction cycle	1 Clock / Instruction cycle	
Bộ nhớ	ROM, SRAM, FLASH	SRAM, FLASH	Flash, SRAM, EEPROM	Flash, SDRAM, EEPROM	
ISA	CLSC	RISC	RISC	RISC	
Kiến trúc bộ nhớ	Kiến trúc Von Neumann	Kiến trúc Harvard	Biến đổi	Kiến trúc Harvard đã biến đổi	
Các họ	Các biến thể 8051	PIC16,PIC17, PIC18, PIC24, PIC32	Tiny, Atmega, Xmega, AVR mục đích đặc biệt	ARMv4,5,6,7 và các dòng	
Cộng đồng	Rộng	Rất tốt	Rất tốt	Rộng	
Nhà sản xuất	NXP, Atmel, Silicon Labs, Dallas, Cyprus, Infineon,	Microchip Average	Atmel	Apple, Nvidia, Qualcomm, Samsung Electronics, STMicroElectronics	
Chi phí	Rất rẻ	Trung bình	Trung bình	Rẻ	
Các vi điều khiển phổ biến	AT89C51, P89v51,	PIC18fXX8, PIC16f88X, PIC32MXX	Atmega8, 16, 32, Cộng đồng <mark>Arduino</mark>	LPC2148, ARM Cortex-M0 đến ARM Cortex-M7,	

Kiến trúc HTN (Von Neumann và Havard)

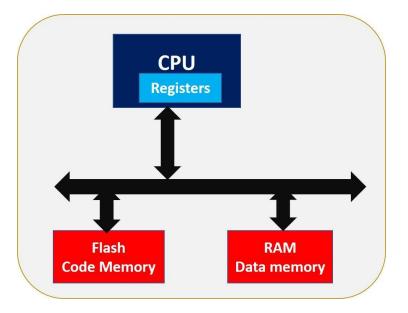
- Được chi làm 2 loại: Von Neumann và Havard
- Thể hiện sự liên kết giữa CPU với bộ nhớ hệ thống



2.3. Bộ nhớ



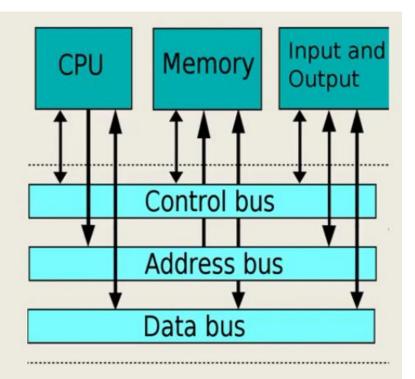
- Các thanh ghị (Registers): Lệnh, Đa năng, Cờ, Con trỏ...
- Bộ nhớ chính (Main Memory):
 - Bộ nhớ chương trình (bộ nhớ ROM/Flash ROM)
 - Lưu chương trình
 - Bộ nhớ dữ liệu (bộ nhớ RAM)
 - Dùng làm môi trường xử lý thông tin, lưu trữ các kết quả trung gian và kết quả cuối cùng của các phép toán, xử lí thông tin.
 - dùng để tổ chức các vùng đệm dữ liệu, trong các thao tác thu phát, chuyển đổi dữ liệu.



2.4. BUS liên kết hệ thống



- Bus địa chỉ: dùng để tham chiếu đến địa chỉ của bộ nhớ, của ngoại vi,...
 (1 chiều từ CPU)
- Bus dữ liệu: dùng để truyền các lệnh, dữ liệu giữa CPU và bộ nhớ/hoặc ngoại vi.
 (hai chiều)
- Bus điều khiển: dùng để gửi các tín hiệu điều khiển như WR, RD, Clock, RST,...
 (2 chiều, nhận t.hiệu từ slave.)



Activate Windows
Go to Settings to activate Windows.

2.5. Ngoại vi



Ngõ vào (Input)

- BXL tiến hành đọc các thông tin/dữ liệu (cảm biến, tác động ngoại cảnh) từ ngõ vào để xử lý,
- Ví dụ: CB nhiệt độ, CB vật cản, Nút nhấn, Chuột/bàn phím,...

Ngõ ra (Output)

- Sau khi xử lý dữ liệu, BXL gửi các lệnh để điều khiển các ngô ra (các cơ cấu chấp hành), hiển thị thông tin,...
- Ví dụ: Led, động cơ, LCD, đóng/mở rèm,...

Activate Win

Ngoại vi ...

- Hệ thống nhúng giao tiếp với bên ngoài thông qua các thiết bị ngoại vi, sử dụng giao tiếp 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: Controller Area Network, LonWorks...
 - Bộ định thời: PLL(s), Capture/Compare và Time Processing Units
 - Discrete IO: General Purpose Input/Output (GPIO)

2.6. Bộ định thời (Timer), bộ đếm (Counter)



- Đây là một trong những chức năng hữu ích của vi điều khiển.
- Một vi điều khiển có thể có nhiều bộ đếm thời gian và bộ đếm.
- Bộ đếm thời gian và bộ đếm có chức năng đếm thời gian và đếm bên trong vi điều khiển.
- Hoạt động chính của bộ phận này là làm chức năng đồng hồ, phát xung, đo tần số, tạo ra dao động...
- Nó cũng được sử dụng để đếm xung bên ngoài.

2.7. ADC, DAC

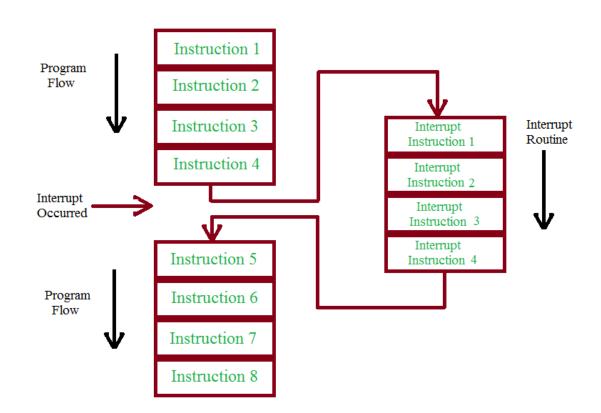


- Bộ chuyển đổi ADC được sử dụng để chuyển đổi tín hiệu analog sang dạng digital.
 - Tín hiệu đầu vào trong bộ chuyển đổi này phải ở dạng analog (ví dụ: đầu ra cảm biến) và đầu ra từ thiết bị này ở dạng digital.
 - Đầu ra digital có thể được sử dụng cho các ứng dụng kỹ thuật số (ví dụ: các thiết bị đo lường).
- Hoạt động của DAC là đảo ngược của ADC.
 - DAC chuyển đổi tín hiệu digital thành định dạng analog.
 - Nó thường được sử dụng để điều khiển các thiết bị analog như động cơ DC, các ổ đĩa...

2.8. Điều khiển ngắt (Interupt)



- Điều khiển ngắt được sử dụng để ngắt (trễ) một chương trình làm việc.
- Việc ngắt có thể
 - ở bên ngoài (được kích hoạt bằng cách sử dụng chân ngắt)
 - hoặc bên trong (bằng cách sử dụng lệnh ngắt trong khi lập trình).



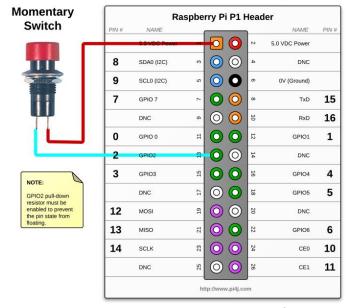
3. Các chuẩn giao tiếp có dây

Truyền thông nối tiếp (Digital GPIO, One Wire, URART, I2C, SPI, USB, CAN) Truyền thông song song (LPT, ATA)

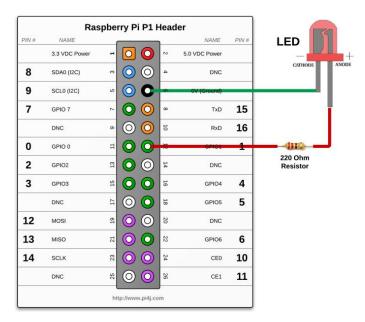
3.1. Cổng GPIO (- General Purpose Input Output)



- Là 1 chân tín hiệu số
- Có thể được cấu hình thành 1 trong 3 trạng thái
 - Ngõ vào, Ngõ ra, Ngõ cách ly (trạng thái tổng trở cao)



Ví dụ: Đọc trạng thái nút bấm

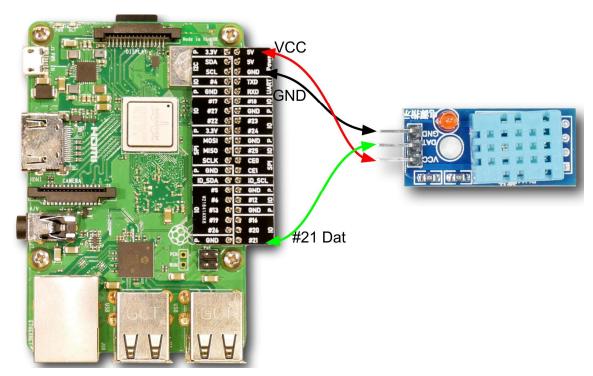


Ví dụ: Điều khiển đèn LED

3.2. One – Wire (1 dây)



- Là chuẩn giao tiếp đơn giản, sử dụng 1 dây để truyền/nhận dữ liệu
- Truyền/nhận dữ liệu dạng số, tốc độ truyền thấp: 16.3kbps
- Gồm 2 mức logic: 0, 1
- Ví dụ: Dùng đọc tín hiệu từ đầu ra của cảm biến nhiệt độ DTH11



3.3. UART



Universal Ansynchronous Receiver – Transmitter

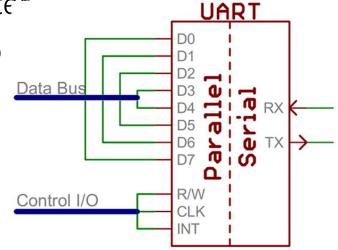
Chuẩn truyền-nhận tín hiệu số không đồng bộ

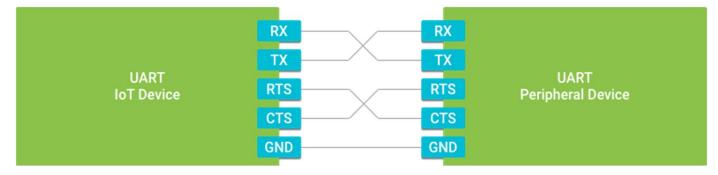
 Sử dụng 2 chân TX, RX để truyền thông giữa các thiết bị với nhau

TX: truyền đi

• RX: nhận về

Tốc độ truyền lên tới 115.2kbps

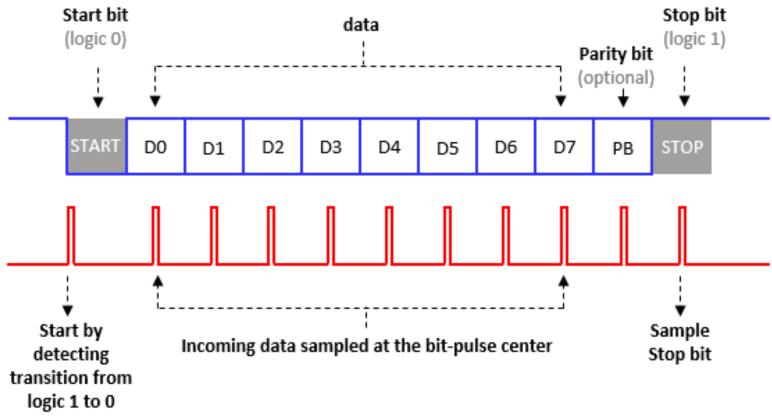




3.3. UART ..

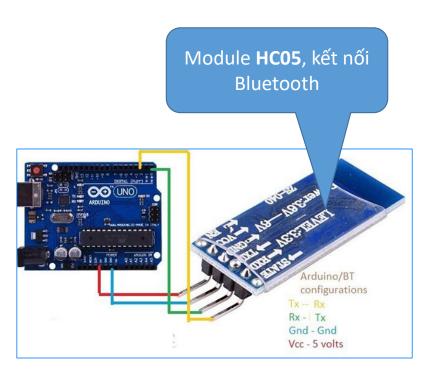


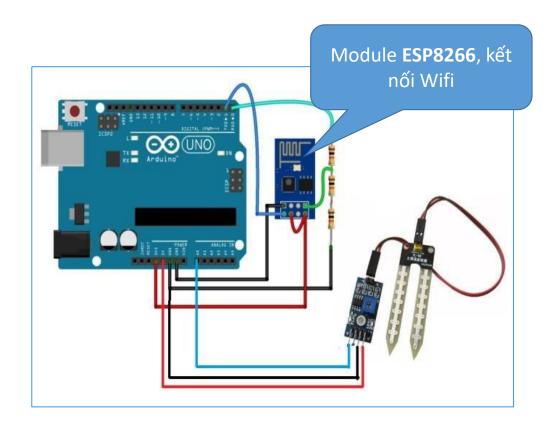
Cấu trúc khung dữ liệu



3.3. UART .. (ví dụ)





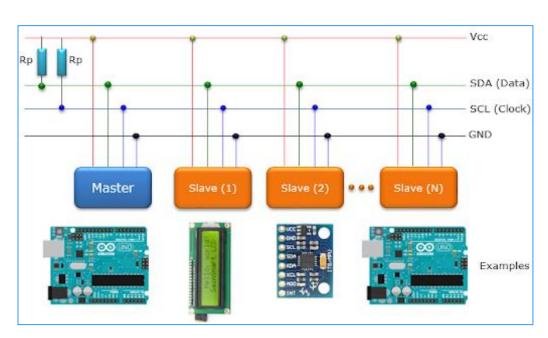


3.4. I2C (Inter – Intergrated Circuit)

Chuẩn giao tiếp Đồng bộ, dạng số. Sử dụng 2 dây

• SCL (Serial Clock): để gửi xung điều khiển/đồng bộ

• SDA (Serial Data): để truyền dữ liệu 2 chiều



Address: 0110000 SCL Slave 2 Master SDA SDA Address: 1001000 Slave 3 SDA Addres 010110

Slave 1

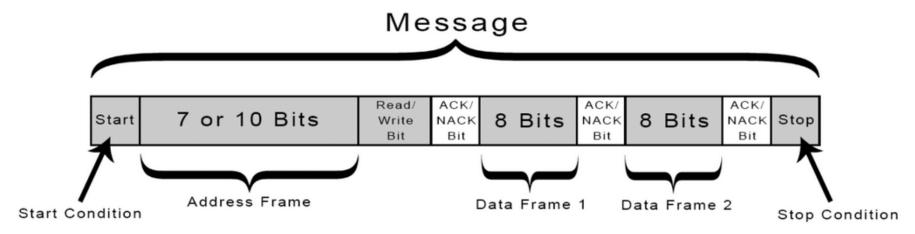
SDA

9/6/2021

3.4. **I2C** ...



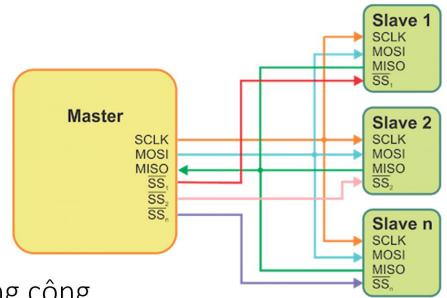
Cấu trúc khung dữ liệu



3.5. SPI (Serial Peripheral Interface)



- Truyền/Nhận tín hiệu số
- Đồng bộ dữ liệu bằng xung Clock
- Sử dụng 4 dây
 - SCLK: xung clock
 - MOSI: Dữ liệu từ Master → Slave
 - MISO: Dữ liệu từ Slave → Master
 - SS: chon chip
- Tốc độ: nhanh gấp đôi I2C, truyền song công
- Truyền bit-bit, không cần tạo frame như UART, I2C



4. Các chuẩn truyền thông không dây

Blutooth, Wifi, RF

4.1. Bluetooth



- Sóng không dây
- Tiêu thụ điện năng thấp
- Thông thường giao tiếp theo mô hình Client/Server
- Khoảng cách, tốc độ tùy thuộc phiên bản Chuẩn

Bluetooth Spec. Evolution										
Specifications	1.1	1.2	2.0 + EDR	2.1 + EDR	3.0 +HS	4.0	5.0			
Adopted	2002	2005	2004	2007	2009	2010	2016			
Transmission	723.1	723.1	2.1	3	24	25	50			
Rate	kbps	kbps	Mbps	Mbps	Mbps	Mbps	Mbps			
Standard PAN Range	10 m	10 m	10 m	10 m	10 m	50 m	300 m			
Improved Pairing (without a PIN)				Yes	Yes	Yes	Yes			
Improved Security		Yes	Yes	Yes	Yes	Yes	Yes			



4.2. Wifi



Sóng không dây, giúp giao tiếp giữa nhiều thiết bị với nhau

Hoạt động ở 2 băng tần thông dụng: 2.4GHz, 5GHz

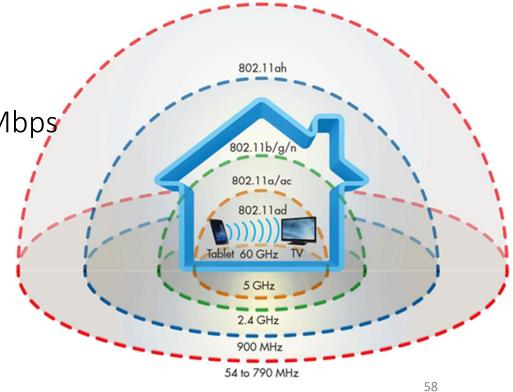
Tiêu thụ năng lượng lớn: 100mA

Khoảng cách: vài trăm mét

Tốc độ wifi có thể đạt: 54Mbps -600Mbps

3rd GEN 5th GEN 6th GEN r1axwifia 11a/11g 11n 11ac \boxtimes

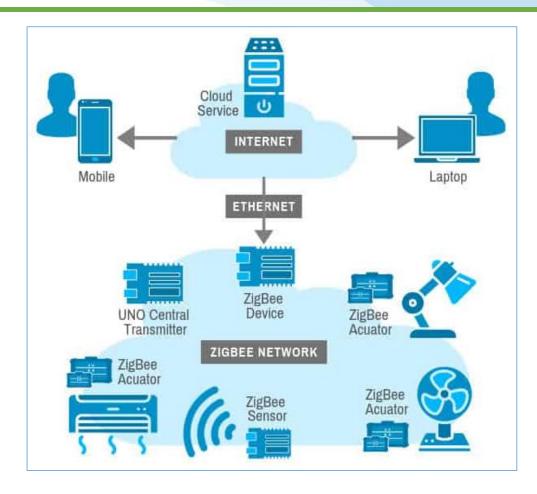
THE EVOLUTION OF WIFI



4.3. Zigbee



- Sử dụng tín hiệu vô tuyến (Radio Frequent) năng lượng thấp, cho mạng PAN (Private Area Network)
- ZigBee hoạt động dựa trên thông số kỹ thuật của IEEE 802.15.4



4.4. LoRA - Long Range Radio

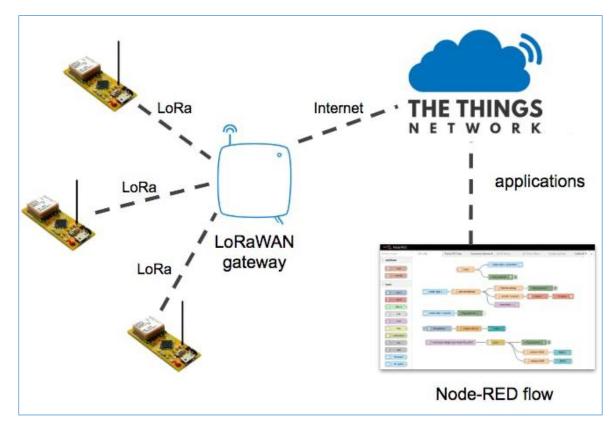
Băng tần làm việc của LoRa từ 430MHz đến 915MHz cho từng khu vực

khác nhau trên thế giới:

• 430MHz cho châu Á

780MHz cho Trung Quốc

- 433MHz hoặc 866MHz cho châu Âu
- 915MHz cho USA



5. Phần mềm HTN

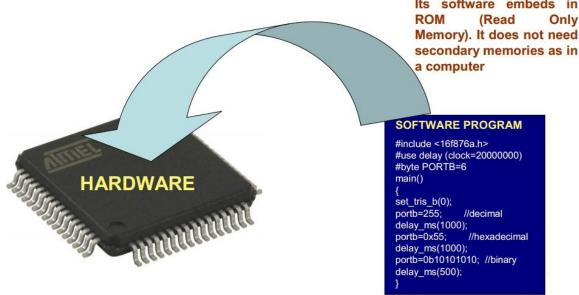
- 1) Tổng quan
- 2) Trình điều khiển thiết bị
- 3) Hệ điều hành nhúng
- 4) Tiến trình
- 5) Middleware, App

5.1. Tổng quan



- Phần mềm nhúng: là phần mềm với
 - Chức năng chuyên biệt
 - Được viết ra để điều khiển một loại hệ thống phần cứng cụ thể

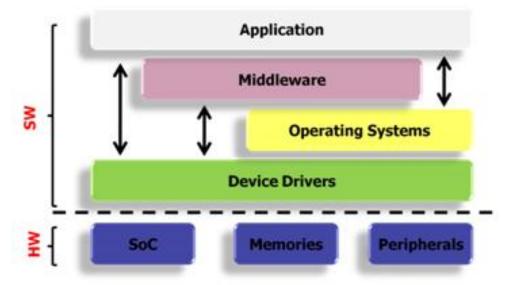
Da số HTN là thời gian thực, nên PMN cũng bị ràng buộc về mặt thời gian.
Its software embeds in



5.2. Trình điều khiển thiết bị



- Mối thành phần phần cứng bên trong HTN, đều có một phần mềm riêng biệt, để giao tiếp và điều khiển trực tiếp, gọi là Device Driver
- Các chương trình lớp trên muốn truy xuất đến phần cứng nào thì phải thông qua Driver của phần cứng đó



5.3. Hệ điều hành nhúng



 Chương trình phần mềm được viết cho HTN, giúp việc giao tiếp giữa người sử dụng với phần cứng được dễ dạng hơn

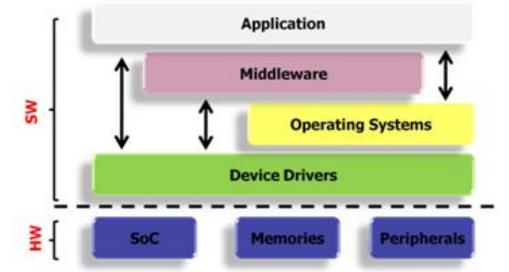
HĐH nhúng có vai trò phân phối, cấp phát và giải quyết tranh chấp tài nguyên hệ

thống, như:

• Thời Sử dụng CPU

- Vùng bộ nhớ, Vùng lưu trữ tập tin
- Ngoại vi
- Một số OS







5.4. Tiến trình



- HTN chứa hệ điều hành hoạt động bằng cách chia nhỏ chương trình lớn thành nhiều chương trình nhỏ hơn, cùng thực thi.
- Một chương trình khi hoạt động sẽ được gọi là tiến trình (mỗi tiến trình sẽ được cấp các tài nguyên riêng)

Hệ điều hành

Tiến trình 1

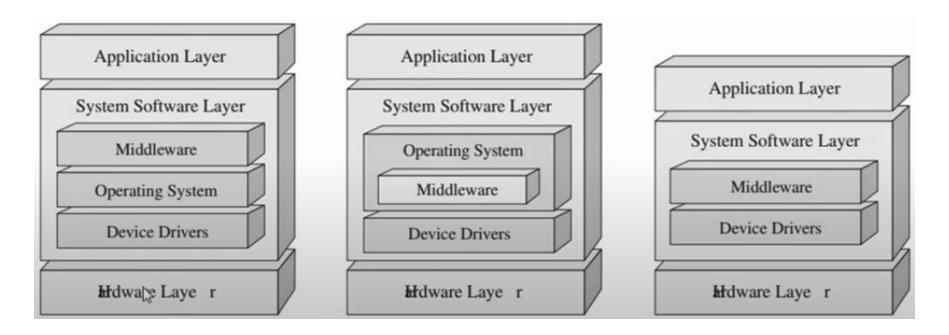
- Không gian địa chỉ A (bộ nhớ)
- Con trỏ lênh
- Thanh ghi stack
- Tài nguyên #

Tiến trình 2

- Không gian địa chỉ B (bộ nhớ)
- Con trỏ lênh
- Thanh ghi stack
- Tài nguyên #

5.5. Middleware, Aplication

- - Middleware: có thể nằm ở nhiều nơi, tách biệt với App, OS và Driver
 - Application: phần mềm nằm ở lớp trên cùng, giao tiếp với người dùng



5.5.1. Middleware



- Được xem là phần mềm trung gian, đảm bảo sự linh hoạt, an ninh và giúp liên kết các phần mềm ở các lớp trên và dưới kế cận.
- Thông thường chứa các hàm dịch vụ, có thể được gọi sử dụng nhiều lần bởi các phần mềm khác nhau. Nhằm làm giảm độ phức tạp của các phần mềm lớp trên

Một số câu hỏi ôn tập



- Câu 1. Hệ thống nhúng là gì, các thành phần cơ bản
- Câu 2. Phân loại hệ thống nhúng
- Câu 3. So sánh Vi xử lý với Vi điều khiển
- Câu 4, Kiến trúc tập lệnh và vai trò
- Câu 5. Một số chuẩn truyền thông nối tiếp
- Câu 6. Một số chuẩn truyền thông không dây
- Câu 7. Phân biệt một số phần mềm HTN