

## Mục lục

LỜI CẢM ƠN .....	3
BẢNG LIỆT KÊ CÁC TỪ VIẾT TẮT .....	4
LỜI MỞ ĐẦU .....	5
Chương 1: Tổng quan về mạng WSN, vi điều khiển CC1010 và mô hình triển khai của mạng WSN .....	7
1.1. Tổng quan về mạng cảm nhận không dây WSN.....	7
1.1.1 Giới thiệu.....	7
1.1.2 Những khó khăn trong vấn đề chọn đường và thiết kế trong mạng WSN .....	9
1.2. Vi điều khiển CC1010 và ứng dụng làm nút mạng WSN.....	12
1.2.1. Đặc điểm chính của CC1010 .....	13
1.2.2. Sơ đồ chân tín hiệu.....	14
1.2.3. Cổng.....	14
1.2.4. Ngắt.....	15
1.2.5. Bộ nhớ Flash .....	15
1.2.6. Bộ định thời.....	16
1.2.7. Bộ biến đổi ADC .....	16
1.2.8. Bộ thu phát không dây (RF Transceiver) .....	16
1.2.9. Bộ mã hoá DES.....	17
1.3. Giới thiệu mô hình triển khai của mạng WSN.....	18
1.4. Kết luận .....	19
Chương 2: Khung dữ liệu của mạng WSN và tìm hiểu phần mềm nhúng trên nút mạng WSN .....	20
2.1. Khung dữ liệu của mạng WSN.....	20
2.2. Phần mềm nhúng trên nút mạng WSN.....	22
2.2.1. Giới thiệu chung.....	22
2.2.2. Phần mềm nhúng viết cho CC1010 .....	23
2.3. Kết luận .....	25
Chương 3: Thu nhận dữ liệu từ nút mạng WSN truyền về PC và thực hiện lưu trữ thành tệp tin.....	26
3.1. Giới thiệu về ngôn ngữ Visual Basic 6.0 .....	26
3.2. Lập trình giao tiếp qua cổng COM bằng VB .....	28
3.2.1. Điều khiển Microsoft Comm Control 6.0.....	28
3.2.2. Thiết lập tham số.....	28
3.2.3. Nhận dữ liệu.....	31
3.2.4. Xuất dữ liệu.....	33
3.2.5. Thuộc tính khác.....	33
3.2.6. Sự kiện On_Comm().....	35
3.3. Làm thế nào để thu nhận dữ liệu từ cổng COM và lưu trữ kết quả thành tệp tin trên máy tính.....	37

Chương 4: Nghiên cứu xây dựng Website, tự động truyền tải thông tin của nút mạng WSN lên Website.....	40
4.1. Giới thiệu về công nghệ Website .....	40
4.2. Triển khai ứng dụng trên Website cho hệ thống mạng WSN .....	42
4.3. Thực hiện tự động quảng bá tệp tin trên mạng .....	43
4.4. Lựa chọn ngôn ngữ lập trình website .....	46
KẾT LUẬN.....	49
TÀI LIỆU THAM KHẢO .....	51

## **LỜI CẢM ƠN**

*Em xin bày tỏ lòng biết ơn sâu sắc nhất tới PGS.TS Vương Đạo Vy, thầy đã tận tình hướng dẫn và giúp đỡ em rất nhiều trong quá trình tìm hiểu và nghiên cứu chương trình để em có thể hoàn thành tốt đề tài tốt nghiệp của mình.*

*Em xin chân thành cảm ơn sự dạy bảo của các thầy giáo, cô giáo Khoa Công Nghệ Thông Tin - Trường Đại học Dân Lập Hải Phòng đã trang bị cho em những kiến thức quý báu để em có thể hoàn thành tốt đề tài tốt nghiệp.*

*Xin chân thành cảm ơn các bạn trong nhóm đã giúp đỡ và đóng góp ý kiến cho đề tài của tôi.*

Sinh viên thực hiện:

**Lê Thị Phương**

## **BẢNG LIỆT KÊ CÁC TỪ VIẾT TẮT**

<b>Từ Tiếng Việt</b>	<b>Từ viết tắt</b>	<b>Từ Tiếng Anh</b>
Mạng cảm nhận không dây	WSN	Wirless Sensor Network
Truyền nhận không dây	RF	Radio Frequency
Biến đổi tương tự - số	ADC	Analog to Digital Converter
Truyền nhận không đồng bộ vạn năng	UART	Universal Asynchronous Receiver Transmitter
Kiểm soát truy nhập môi trường	MAC	Medium Access Control
Chất lượng dịch vụ	QoS	Quality of Service
Yêu cầu dịch vụ ngắt	ISR	Interrupt Service Request
Bộ điều chế rộng xung	PWM	Pulse Width Modulation
Hệ điều hành	OS	Operating System

## **LỜI MỞ ĐẦU**

Sự phát triển đi lên vượt bậc của ngành kỹ thuật máy tính và điện tử hiện nay đã được minh chứng cụ thể qua cuộc sống hằng ngày của chúng ta trong tất cả các lĩnh vực. Việc ứng dụng máy tính vào kỹ thuật đo lường và điều khiển đã đem lại những kết quả đầy tính ưu việt. Các thiết bị, hệ thống đo lường và điều khiển ghép nối với máy tính có độ chính xác cao, thời gian thu thập số liệu ngắn, nhưng đáng quan tâm hơn là mức độ tự động hoá trong việc thu nhận, xử lý và quảng bá dữ liệu.

Một lĩnh vực nổi bật của mạng cảm nhận không dây ( WSN - Wireless Sensor Network ) là sự kết hợp việc cảm nhận, tính toán và truyền thông vào một thiết bị nhỏ. Và chúng ta hoàn toàn có khả năng triển khai các thiết bị nhỏ này thành một hệ thống có ứng dụng lớn và rộng rãi. Sử dụng những thiết bị này để theo dõi theo thời gian thực, để giám sát điều kiện môi trường, để theo dõi cấu trúc hoặc tình trạng thiết bị.

Hầu hết các ứng dụng của mạng WSN là dùng để giám sát, theo dõi một hệ thống hoặc cảm nhận thông tin dữ liệu từ môi trường rồi gửi về trung tâm. Khi đó cần phải có người theo dõi, trực tại trung tâm xử lý để quan sát và nắm bắt tình hình. Nhưng điều đó thật bất cập khi mà ta suốt ngày phải ngồi bên máy vi tính để theo dõi, điều này làm tăng chi phí để hoạt động hệ thống và thời gian phục vụ hệ thống.

Cùng với sự ra đời và phát triển của hệ thống mạng máy tính thì thông tin hiện nay được phổ cập trên toàn thế giới. Một hệ thống quan sát tối ưu khi mà hệ thống đó có thể thực hiện tự động quảng bá dữ liệu trên mạng. Khi đó ta hoàn toàn có thể quan sát hệ thống từ xa ( qua mạng Internet ) mà không cần phải ngồi tại trung tâm xử lý để theo dõi diễn biến tình hình.

Bản luận văn tốt nghiệp “ Xây dựng Website quảng bá thông tin mạng WSN” sẽ tổng quan về mạng WSN, đi sâu và phân tích cấu trúc khung dữ liệu truyền nhận của nút mạng về nút cơ sở; nghiên cứu kỹ thuật tách các thông tin từ khung dữ liệu, nghiên cứu xây dựng Website có khả năng tự động cập nhật thông tin.

Bài luận văn gồm có 4 chương nội dung, phần mở đầu, phần kết luận và phần phụ lục.

*Chương 1: Tổng quan về mạng WSN, vi điều khiển CC1010 và mô hình triển khai của mạng WSN*, chương này sẽ giới thiệu một cách tổng quát về mạng cảm nhận không dây, giới thiệu vi điều khiển CC1010 của hãng CHIPCON và mô hình triển khai mạng WSN.

*Chương 2: Khung dữ liệu của mạng WSN và tìm hiểu phần mềm nhúng trên nút mạng WSN*, chương này tìm hiểu khung dữ liệu của mạng WSN và phần mềm nhúng trên nút mạng WSN

*Chương 3: Thu nhận dữ liệu từ nút mạng WSN truyền về PC và thực hiện lưu trữ thành tệp tin*, chương này nói về việc thu nhận dữ liệu từ nút mạng WSN truyền về PC và thực hiện lưu trữ thành tệp tin

*Chương 4: Nghiên cứu xây dựng Website, tự động truyền tải thông tin của nút mạng WSN lên Website*, tìm hiểu và nghiên cứu việc xây dựng website có khả năng tự động cập nhật nội dung tệp tin

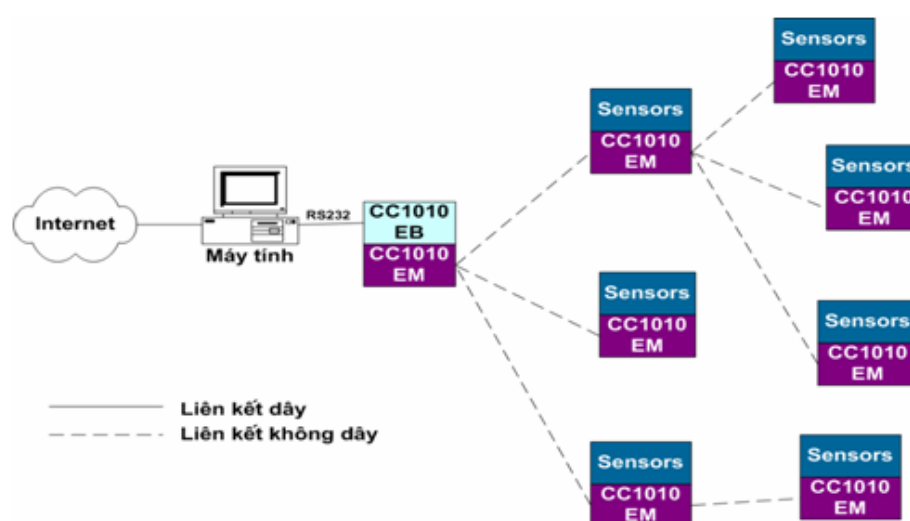
## Chương 1: Tổng quan về mạng WSN, vi điều khiển CC1010 và mô hình triển khai của mạng WSN.

### 1.1. Tổng quan về mạng cảm nhận không dây WSN

#### 1.1.1 Giới thiệu

Ngày nay, các vi điều khiển đã có một bước phát triển mạnh với mật độ tích hợp cao, khả năng xử lý mạnh, tiêu thụ năng lượng ít và giá thành thấp. Khi được nạp phần mềm nhúng, các vi điều khiển này sẽ hoạt động độc lập trong các loại môi trường và ở những vị trí địa lý khác nhau. Mỗi vi điều khiển khi được tích hợp với bộ thu phát sóng vô tuyến và bộ cảm biến sẽ tạo thành một nút mạng, tập hợp các nút mạng đó trong một phạm vi nhất định được gọi là mạng cảm nhận không dây (WSN-Wireless Sensor Network).

Mạng cảm nhận không dây là một mạng không dây mà các nút mạng là các vi điều khiển sau khi đã được cài đặt phần mềm nhúng kết hợp với các bộ phát sóng vô tuyến cùng với các cảm biến và nó có khả năng thu nhận, xử lý dữ liệu từ các nút mạng và môi trường xung quanh nút mạng.



Trong hệ thống WSN có các trạm gốc và trung tâm điều khiển. Trạm gốc đóng vai trò cổng kết nối giữa các nút mạng và trung tâm điều khiển, tiếp nhận thông tin của các nút mạng chuyển tới trung tâm điều khiển qua nhiều cách khác nhau. Các nút mạng truyền thông tin theo kiểu nhiều chặng từ nút mạng này sang nút mạng khác và về trạm gốc. Từ trạm gốc có thể gửi thông tin cho người dùng (trung tâm điều khiển) theo nhiều cách như trực tiếp qua hệ thống máy tính, qua mạng Internet, qua vệ tinh.... nhờ đó người giám sát có thể nhận được thông tin dù đang ở bất cứ đâu.

### **\*Ưu nhược điểm WSN, ứng dụng trong cuộc sống**

#### **\*Ưu điểm :**

- Tính linh hoạt: Trong phạm vi của bán kính truyền, sóng radio có thể đi xuyên qua được những bức tường ngăn cách nhau, do đó máy gửi và nhận tin có thể ở bất kỳ vị trí nào trong vùng phủ sóng.
- Tiết kiệm được chi phí xây lắp, thiết lập môi trường truyền tin cho mạng do không phải lắp đặt lại hệ thống cáp khi có biến động như ở mạng có dây. Với mạng không dây (WLAN) có thể dễ dàng di chuyển mạng từ nơi này đến nơi khác và giảm thiểu những chi phí không cần thiết.
- Dễ dàng sử dụng và cài đặt, tiết kiệm chi phí.
- Thuận tiện cho việc mở rộng hệ thống mạng.
- Hiệu suất làm việc tăng, có tính mềm dẻo.

#### **\*Nhược điểm :**

- Mạng cảm nhận không dây có thể cho mọi người truy cập ở bất kỳ đâu nhưng do thiết bị di động có màn hình hiển thị nhỏ nên khi hiển thị thông tin gặp khó khăn, điều khó khăn nữa là năng lượng tiêu thụ.
- Tốc độ truyền dữ liệu của mạng không dây chậm.



- An toàn bảo mật thông tin trên mạng phức tạp

**\* Ứng dụng:**

Với những ưu điểm vượt trội như đã nêu ở trên .Mạng cảm nhận không dây không ngừng được phát triển và được ứng dụng rất nhiều vào cuộc sống.

Ta có thể gói gọn các ứng dụng thành 3 dạng :

- Thu thập dữ liệu môi trường.
- Giám sát an ninh.
- Theo dõi đối tượng.

### **1.1.2 Những khó khăn trong vấn đề chọn đường và thiết kế trong mạng WSN**

Mặc dù các ứng dụng của mạng WSN là rất lớn, tuy nhiên những mạng này có một số hạn chế như giới hạn về nguồn công suất, khả năng tính toán và độ rộng băng thông. Một trong những mục tiêu thiết kế chính của WSN là kéo dài thời gian sống của mạng và tránh suy giảm kết nối nhờ các kỹ thuật quản lý năng lượng. Dưới đây sẽ tóm tắt một số khó khăn khi triển khai thiết kế mạng WSN:

**Phân bố nút:** Việc phân bố nút trong mạng WSN phụ thuộc vào ứng dụng và có thể được thực hiện bằng tay hoặc phân bố ngẫu nhiên. Khi phân bố bằng tay, số liệu được chọn đường thông qua các đường xác định trước. Tuy nhiên khi phân bố các nút ngẫu nhiên sẽ tạo ra một cấu trúc chọn đường đặc biệt (Ad-hoc). Liên lạc giữa các nút cảm biến thường có cự ly ngắn do hạn chế về năng lượng và băng thông. Do đó việc thực hiện chọn đường sẽ thực hiện qua nhiều bước ( Multi hop).

**Tiêu thụ năng lượng:** Các nút cảm biến có thể sử dụng quá giới hạn về công suất để thực hiện tính toán và truyền tin trong môi trường vô tuyến. Thời gian sống của các nút cảm biến phụ thuộc rất nhiều vào thời gian sử dụng của PIN. Trong WSN đa bước nhảy, mỗi nút đóng vai trò là truyền số liệu và chọn đường. Một số nút cảm biến hoạt động sai chức

năng do lỗi nguồn công suất có thể gây ra sự thay đổi cấu hình mạng nghiêm trọng và phải chọn đường lại các gói hoặc tổ chức lại mạng.

Trong hầu hết các ứng dụng, đặc điểm chính của các nút là tự cấp nguồn. Chúng sẽ có đủ năng lượng cho nhiều năm, hoặc có thể lấy năng lượng từ môi trường thông qua thiết bị khác, như năng lượng mặt trời hay nguồn áp điện. Cả hai sự lựa chọn đều yêu cầu năng lượng tiêu thụ trung bình của các nút càng ít càng tốt. Yếu tố quyết định thời gian sống là năng lượng tiêu thụ radio sẽ tiêu thụ năng lượng lớn. Năng lượng tiêu thụ này có thể giảm được bằng cách giảm năng lượng truyền, tức là giảm chu trình làm việc của Radio.

**Phương pháp báo cáo số liệu:** Việc báo cáo số liệu trong WSN phụ thuộc vào ứng dụng và có thể được chia thành báo cáo theo thời gian, theo sự kiện, theo yêu cầu hoặc lai ghép những phương pháp này. Phương pháp báo cáo theo thời gian phù hợp với các ứng dụng yêu cầu giám sát số liệu định kỳ. Khi đó, các nút cảm biến sẽ bật bộ phận cảm biến và bộ phận phát theo định kỳ, cảm nhận môi trường, phát số liệu yêu cầu theo chu kỳ thời gian xác định. Trong phương pháp báo cáo theo sự kiện và theo yêu cầu, các nút cảm biến sẽ phản ứng tức thì đối với những thay đổi giá trị của thuộc tính cảm biến do xuất hiện một sự kiện xác định nào đó hoặc để trả lời một yêu cầu được tạo ra bởi nút gốc hay các nút khác trong mạng. Do vậy những phương pháp này phù hợp với các ứng dụng phụ thuộc thời gian. Cũng có thể sử dụng kết hợp các phương pháp trên. Giao thức chọn đường chịu ảnh hưởng đáng kể từ phương pháp báo cáo số liệu về vấn đề sử dụng năng lượng và chọn đường.

**Tính không đồng nhất của nút/tuyến:** Trong nhiều nghiên cứu, tất cả các nút cảm biến được giả thiết là đồng nhất( nghĩa là có khả năng tính toán, khả năng truyền tin và có công suất như nhau). Tuy nhiên, tùy theo ứng dụng mà nút cảm biến có thể có vai trò hoặc khả năng khác nhau. Các nút cảm biến không đồng nhất tạo ra một số vấn đề kỹ thuật

liên quan đến chọn đường. Ví dụ, một số ứng dụng có thể cần kết hợp các bộ cảm biến để giám sát nhiệt độ, áp suất, độ ẩm của môi trường, phát hiện chuyển động nhờ âm thanh, chụp ảnh hoặc ghi hình các vật chuyển động. Ngoài ra, việc đọc hoặc báo cáo số liệu từ những bộ cảm biến này có thể có tốc độ khác nhau tùy theo chất lượng của dịch vụ ( QoS ) và có thể thuộc nhiều mô hình báo cáo số liệu khác nhau.

**Khả năng chống lỗi:** Một số nút cảm biến có thể bị lỗi hoặc bị ngắt do thiếu công suất, hỏng phần cứng hoặc bị nhiễu môi trường. Sự cố của các nút cảm biến không được ảnh hưởng tới nhiệm vụ của toàn mạng cảm biến. Nếu có nhiều nút bị lỗi, các giao thức chọn đường hoặc điều khiển truy nhập môi trường (MAC) phải thành lập các tuyến mới tới nút gốc. Việc này có thể cần thiết phải điều chỉnh công suất phát và tốc độ tín hiệu trên các tuyến hiện tại để giảm sự tiêu thụ năng lượng hoặc là các gói phải chọn đường lại qua các vùng mạng có công suất khả dụng lớn hơn.

**Khả năng định cỡ:** Số lượng nút cảm biến có thể hàng trăm, hàng nghìn hoặc nhiều hơn. Bất kỳ phương pháp chọn đường nào cũng phải có khả năng làm việc với một số lượng lớn các nút cảm biến như vậy.

**Tính động của mạng:** Trong nhiều nghiên cứu, các nút cảm biến được giả thiết là cố định. Tuy nhiên trong một số ứng dụng, cả nút gốc và các nút cảm biến có thể di chuyển. Khi đó các bản tin chọn đường từ hoặc tới các nút di chuyển sẽ gặp phải các vấn đề như đường liên lạc, cấu hình mạng, năng lượng, độ rộng băng thông... Tuy nhiên, đối tượng thì có thể di chuyển ( ví dụ ứng dụng dò / tìm theo dõi mục tiêu ). Các sự kiện cố định thì cho phép mạng làm việc ở chế độ phản ứng ( tạo lưu lượng khi cần báo cáo ) trong khi các sự kiện chuyển động thì trong hầu hết các ứng dụng đều yêu cầu phải báo cáo định kỳ cho nút gốc.

**Môi trường truyền dẫn:** Trong mạng cảm biến đa bước nhảy, các nút thông tin được kết nối qua môi trường vô tuyến. Các đặc tính của

kênh vô tuyến như tỷ lệ lỗi cũng có thể ảnh hưởng đến hoặc động của mạng cảm biến. Nói chung, độ rộng băng yêu cầu của số liệu cảm biến là thấp, khoảng từ 1-100kb/s. Liên quan đến môi trường truyền dẫn là việc thiết kế MAC. Một phương pháp thiết kế MAC cho các mạng cảm biến là sử dụng giao thức đa truy nhập phân chia theo thời gian (TDMA) sẽ tiết kiệm năng lượng hơn so với các giao thức đa truy cập khác như đa truy cập theo sóng mang (CSMA).

**Khả năng giám sát:** Trong WSN, mỗi nút cảm biến giám sát một vùng xác định. Vùng giám sát môi trường của nút cảm biến bị giới hạn bởi cự ly và độ chính xác, nó có thể chỉ giám sát một phạm vi rất nhỏ. Do đó, vùng giám sát cũng là một tham số thiết kế quan trọng trong WSN.

**Kết hợp số liệu:** Vì các nút cảm biến có thể tạo ra số liệu dư thừa nên các gói tương tự nhau từ nhiều nút có thể được kết hợp để giảm số lượng truyền dẫn. Việc kết hợp số liệu là từ nhiều nguồn khác nhau theo một hàm kết hợp xác định. Kỹ thuật này được sử dụng để đạt hiệu quả về năng lượng và tối ưu hoá việc truyền số liệu trong một số giao thức chọn đường.

**Chất lượng dịch vụ:** Trong một số ứng dụng, số liệu có thể được phân phối trong một khoảng thời gian xác định ngay khi nó cảm nhận được hiện tượng nếu không số liệu sẽ trở nên vô dụng. Vì vậy, giới hạn trễ của việc phân phối số liệu là một chỉ tiêu khác trong các ứng dụng phụ thuộc thời gian. Tuy nhiên trong một số ứng dụng khác thì việc tiêu thụ công suất ( ảnh hưởng trực tiếp đến thời gian sống của mạng ) lại quan trọng hơn. Khi năng lượng gần hết, mạng có thể yêu cầu giảm chất lượng các kết quả để giảm mức tiêu thụ năng lượng của nút và kéo dài thời gian sống của toàn mạng.

## **1.2. Vi điều khiển CC1010 và ứng dụng làm nút mạng WSN**

Vấn đề lựa chọn vi điều khiển để xây dựng nút mạng là một vấn đề quan trọng. Việc chọn vi điều khiển hợp lý sẽ làm cho quá trình xây dựng

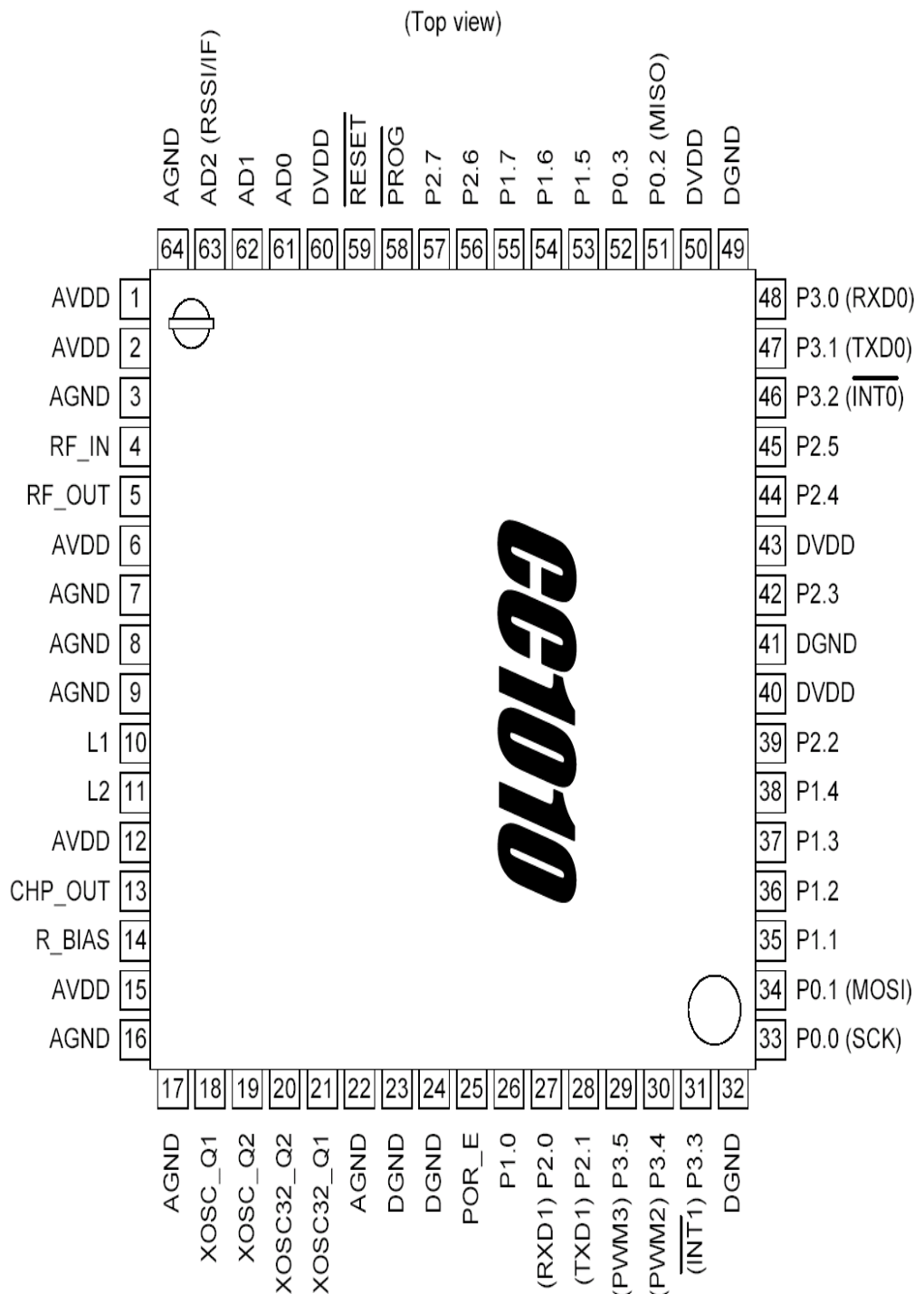
hệ thống được rút ngắn, hệ thống được hoạt động ổn định, tin cậy và đạt các chỉ tiêu đề ra.

Chip CC1010 là một bộ vi xử lý thích hợp cho các ứng dụng truyền nhận không dây. CC1010 được tích hợp nhiều các tính năng phục vụ cho các ứng dụng không dây như bộ truyền- nhận vô tuyến, bộ biến đổi ADC, bộ nhớ lập trình Flash, kích thước nhỏ, tiêu thụ năng lượng thấp... Vì vậy CC1010 chỉ cần đến ít các thành phần phụ trợ khác để có thể trở thành một nút mạng của mạng cảm nhận không dây.

### **1.2.1. Đặc điểm chính của CC1010**

- ✓ Bộ thu phát sóng vô tuyến 300-1000 MHz.
- ✓ Tiêu thụ dòng thấp (9.1 mA trong chế độ thu )
- ✓ Có thể lập trình cho công suất đầu ra tới +10dBm
- ✓ Tốc độ truyền RF có thể đạt tới 76.8 kbit/s
- ✓ Lõi là vi điều khiển họ 8051
- ✓ Tốc độ xử lý bằng 2.5 lần vi điều khiển 8051 chuẩn
- ✓ 32 kB Flash, 2048 + 128 Byte SRAM
- ✓ 3 kênh ADC 10 bit, 4 timers / 2PWMs, 2 UARTs, RTC, Watchdog, SPI, mã hoá DES tích hợp bên trong, 26 cổng I/O
- ✓ Nguồn cung cấp 2.7 – 3.6V
- ✓ Cần ít thành phần ngoài, có khả năng gỡ lỗi bằng cách sử dụng chương trình dịch Keil  $\mu$ Vision2 IDE qua cổng nối tiếp.

### 1.2.2. Sơ đồ chân tín hiệu



Hình 1.2.2 : Sơ đồ chân tín hiệu CC1010

### 1.2.3. Cổng

Có 4 cổng I/O P0, P1, P2, P3 với 26 chân cổng. Mỗi cổng có 2 thanh ghi tương ứng: thanh ghi cổng P0, P1, P2, P3 và thanh ghi hướng (

P0DIR, P1DIR, P2DIR, P3DIR). Mỗi bit trong thanh ghi Px có một bit hướng tương ứng trong thanh ghi PxDIR.y. Việc thiết lập PxDIR.y sẽ làm cho chân Px.y trở thành đầu nhập dữ liệu và đưa vào bit Px (y). Tất cả các chân đều là chân nhập dữ liệu khi mà reset lại chip. Việc xóa bit hướng PxDir.y sẽ làm cho chân Px.y trở thành chân xuất dữ liệu từ thanh ghi Px(y). Một số cổng có những hàm chức năng thêm vào (ví dụ như giao diện SPI). Các chức năng này có thể được dùng thông qua các thanh ghi khác (như SPCR.SPE). Các chức năng thêm vào này có thể ghi đè lên bit hướng được thiết lập trong PxDIR hoặc là không. Khi đọc thanh ghi Px, dữ liệu sẽ lấy từ bộ đệm (Pad). Khi sử dụng các lệnh ghi - đọc dữ liệu thì giá trị của thanh ghi xuất dữ liệu sẽ bị thay đổi bất chấp việc thiết lập các bit hướng trong PxDIR.

#### **1.2.4. Ngắt**

CC1010 có 15 nguồn ngắt, chia sẻ 12 đường ngắt. Mỗi ngắt có một mức ưu tiên, vector ngắt, cờ cho phép ngắt và cờ báo ngắt.

Việc xử lý ngắt diễn ra như sau: Khi một ngắt được cho phép xảy ra, CPU nhảy tới địa chỉ phục vụ ngắt tương ứng với ngắt đó (ISR), CC1010 thực hiện ISR để hoàn thành trừ khi một ngắt khác có mức ưu tiên cao hơn xảy ra. Mỗi ISR kết thúc với lệnh RETI.

#### **1.2.5. Bộ nhớ Flash**

CC1010 có tích hợp 32-kbyte bộ nhớ chương trình flash. Nó được chia thành 256 trang, mỗi trang dài 128 byte. Nó có thể được lập trình, xoá thông qua giao diện nối tiếp SPI hoặc thông qua vi nhân 8051. Tuổi thọ của bộ nhớ Flash thường là 20.000 lần ghi/xoá. Bộ nhớ Flash có thể được khoá để không đọc/ghi được bằng cách thiết lập bit tương ứng thông qua giao diện nối tiếp. Việc xoá chip phải được thực hiện trên bộ nhớ không bị khoá. Điều này cho phép ngăn chặn phần mềm không bị copy trái phép

### **1.2.6. Bộ định thời**

CC1010 có 4 bộ định thời Timer 0, Timer 1, Timer 2, Timer 3 hoạt động như là bộ định thời hay bộ đếm (Timer / Counter ) trong đó Timer2 và Timer3 còn có thể hoạt động như bộ điều chế rộng xung (PWM – Pulse Width Modulation ).

### **1.2.7. Bộ biến đổi ADC**

Bộ biến đổi ADC của CC1010 có độ phân giải 10 bit, được điều khiển bởi các thanh ghi **ADCON** và **ADCON2**. Có 3 kênh vào ADC, được chọn bởi **ADCON.ADADR**. Thanh ghi này cũng được sử dụng để chọn chân AD1 như là điện áp tham chiếu ngoài (khi sử dụng AD0). Khi chân AD1 được dùng như tham chiếu ngoài, chỉ có 2 lối vào ADC được sử dụng.

Bộ biến đổi ADC hoạt động 1 trong 4 chế độ được lựa chọn bởi bit **ADCON.ADCM**. Mỗi lần biến đổi thì mất 11 chu kì xung nhịp. Trong chế độ xung nhịp 1 khi **POWER.PMODE** được thiết lập thì đồng hồ 32 kHz được đưa trực tiếp vào bộ biến đổi ADC. Trong chế độ xung nhịp 0, xung nhịp đầu vào của ADC lấy từ bộ dao động chính bằng cách sử dụng bộ chia được lựa chọn bởi **ADCON2.ADCDIV**. Thanh ghi phải được thiết lập để tần số xung nhịp của ADC phải nhỏ hơn hoặc bằng 250 kHz.

### **1.2.8. Bộ thu phát không dây (RF Transceiver)**

Bộ thu phát CC1010 UHF RF được thiết kế cho những ứng dụng tiêu thụ năng lượng thấp và điện áp thấp. Mạch thu phát được dành cho ISM ( công nghiệp, khoa học và y học ) và SRD (Short Range Device ) dải tần 315, 433, 868 và 915 MHz, nhưng có thể dễ dàng lập trình để hoạt động trong dải tần 300 – 1000 MHz. Các thông số chính của CC1010 có thể lập trình thông qua các thanh ghi chức năng đặc biệt (SFRs – Special Function Registers ), làm cho CC1010 rất mềm dẻo và dễ sử dụng bộ thu phát vô tuyến. Rất ít các thành phần tích cực đòi hỏi cho hoạt động của bộ thu phát RF



### **1.2.9. Bộ mã hoá DES**

Phần cứng trong CC1010 có tích hợp sẵn bộ mã hoá DES. Một gói dữ liệu (từ 1 đến 256 byte) có thể được mã hóa trong một lần thực hiện. Gói dữ liệu lớn hơn phải được thực hiện trong nhiều lần. Việc mã hóa là quá trình chuyển mã dòng bit thông tin để bảo mật dữ liệu. Giải thuật DES thì thông dụng, đơn giản và dễ dàng cài đặt thủ tục mã hóa. Một khóa mã 56 bit được dùng để mã hóa thông tin. Thiết bị nhận phải dùng cùng một khóa để giải mã dữ liệu. Việc giải mã và mã hóa trong giải thuật DES là các thao tác ngược nhau. Các thao tác này sẽ sinh ra số lượng byte đầu ra bằng số lượng byte đầu vào. Độ tin cậy của giải thuật DES phụ thuộc vào số bit của khoá, càng dài càng tốt. Giải thuật mã hoá DES cho phép mức độ bảo mật từ thấp đến trung bình. Nếu yêu cầu độ bảo mật cao hơn thì giải thuật DES cấp 3 được sử dụng. DES cấp 3 được thực hiện bằng việc chạy giải thuật DES tuần tự 3 lần và sử dụng 3 khoá khác nhau. Các chế độ OFB và CFB là các chế độ hoạt động của giải thuật DES cho phép độ dài dữ liệu không phải là bội số của 8. Chế độ hoạt động được lựa chọn thông qua bit điều khiển CRPCON.CRPM D. Mã hoá và giải mã sử dụng cùng một chế độ CFB được dùng nhiều hơn bởi vì nó an toàn hơn OFB. CRPCON.ENCDEC được xoá khi mã hoá dữ liệu và được thiết lập khi giải mã dữ liệu.

Các khoá 56 bit phải được lưu trữ trong RAM ngoài. Vị trí được xác định trong thanh ghi CRPKEY, chứa 8 bit địa chỉ. Khoá mới chỉ được nạp vào khi bắt đầu giải mã hoặc mã hoá nếu bit CRPCON.LOADKEYS được thiết lập. Nếu không khoá cũ trong lần chạy trước sẽ được sử dụng lại. Các khoá không chứa bit chẵn lẻ, nếu có chứa các bit đó phải được loại bỏ trước khi mã hoá hoặc giải mã. Vì vậy khoá được lưu trữ trong 7 byte trong RAM.

Việc mã hoá giải mã được thực hiện tại chỗ: mỗi byte dữ liệu được đọc từ RAM để mã hoá/ giải mã sẽ được ghi lại đúng vị trí sau khi đã mã

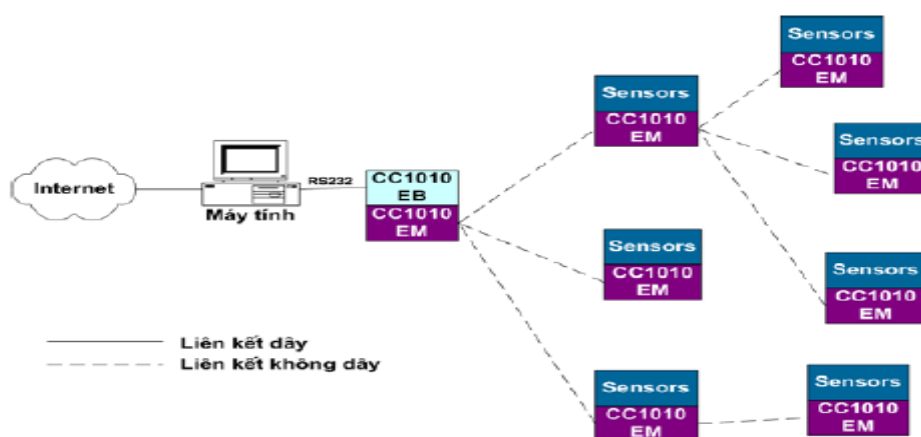
hoá/giải mã. Khối dữ liệu đầu vào và đầu ra phải bắt đầu tại địa chỉ mà chia hết cho 8. Các vector cài đặt cho việc mã hoá/giải mã phải được ghi vào các thanh ghi CRPINI0 đến CRPINI7. Các thanh ghi này phải được ghi lại mỗi khi mã hoá/giải mã một khối dữ liệu mới bởi vì chúng bị phần cứng làm thay đổi giá trị

### **1.3. Giới thiệu mô hình triển khai của mạng WSN**

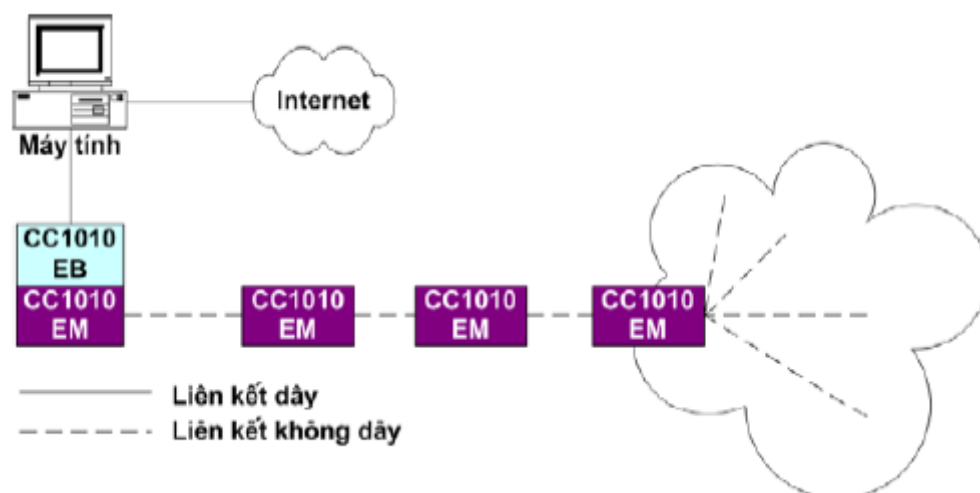
WSN gồm nhiều nút mạng CC1010 giao tiếp với nhau qua sóng vô tuyến tần số 300 – 1000 Mhz. Có 3 loại nút mạng: trạm gốc, nút cảm nhận và nút chuyển tiếp ( hay nút trung gian ).

Trạm gốc tiếp nhận dữ liệu từ mạng và chuyển trực tiếp vào máy tính. Trong mô hình mạng đã xây dựng, khối CC1010EB kết nối trực tiếp với máy tính và khối CC1010EM được đính kèm để thu, phát dữ liệu không dây.

Các nút cảm nhận có gắn các đầu đo vừa trực tiếp đo số liệu và truyền về trạm gốc vừa chuyển tiếp dữ liệu nhận được từ các nút con trong Topology dạng cây, gửi về cho nút cha.



**Hình 1.3. a– Mô hình triển khai của mạng cảm nhận sử dụng CC1010**



**Hình 1.3.b: Mô hình triển khai của mạng cảm nhận khi trạm gốc không gần trung tâm**

**Cơ chế hoạt động :** WSN thu thập dữ liệu môi trường phải thực hiện 3 chức năng : thu thập dữ liệu, phân tích dữ liệu và biểu diễn dữ liệu.

#### **1.4. Kết luận**

Trong chương 1 đã trình bày tổng quan về mạng WSN, giới thiệu về vi điều khiển CC1010 và mô hình triển khai của mạng WSN sử dụng CC1010.

Chương tiếp theo sẽ tìm hiểu về khung dữ liệu và phần mềm nhúng cho mạng WSN

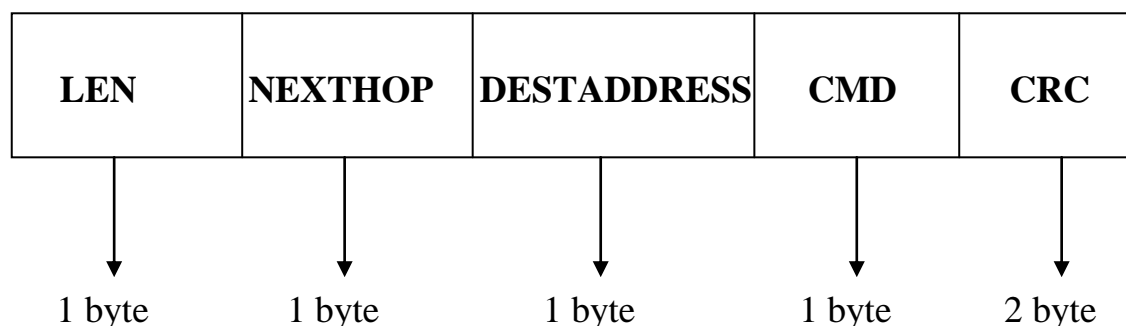
## Chương 2: Khung dữ liệu của mạng WSN và tìm hiểu phần mềm nhúng trên nút mạng WSN

### 2.1. Khung dữ liệu của mạng WSN

Trong mô hình mạng cảm nhận không dây WSN thì luôn có mối quan hệ truyền - nhận giữa nút cảm nhận và trung tâm, có thể cả nút trung gian (nếu truyền đa bước – Multi hop). Như vậy, việc xác định khuôn dạng truyền dữ liệu trên mạng là một đặc điểm rất quan trọng để giúp ta có thể viết ra phần mềm nhúng trên nó. Hoặc có thể dễ dàng lấy ra dữ liệu có ích để xử lý. Tại trung tâm xử lý có kết nối với máy tính qua cáp nối RS232, tuy nhiên việc truyền về máy tính là chỉ ở dạng Text và khi đó ta cũng phải có sử dụng kỹ thuật riêng để có thể thu nhận được dữ liệu.

✓ Truyền dữ liệu từ **Master** tới **EndPoint**:

Định dạng dữ liệu truyền từ **Master** qua các chặng trung gian cho **Endpoint** như sau:



**Hình 2.1.a : Định dạng dữ liệu truyền từ Master đến Endpoint**

Trong đó, ý nghĩa các trường như sau:

- + **LEN**: chỉ độ dài gói tin, 1 byte
- + **EndPoint**: là nút mạng cuối cùng mà Master muốn gửi gói tin đến.
- + **NEXTHOP**: Địa chỉ nút kế tiếp có thể tới đích ( nút trung gian ), 1 byte.

- + **DESTADDRESS**: Địa chỉ của Endpoint, 1 byte.
  - + **CMD**: Lệnh gửi cho Endpoint phải thi hành.
  - + **CRC**: Mã vòng dư, để kiểm tra lỗi trong khi truyền- nhận dữ liệu, 2 byte.
- ✓ Định dạng dữ liệu truyền về **Master**
- Định dạng dữ liệu của **Endpoint** truyền về trung tâm như sau:

<b>LEN</b>	<b>PARENTADDRESS</b>	<b>DESTADDRESS</b>	<b>CMD</b>	<b>DATA</b>	<b>CRC</b>
------------	----------------------	--------------------	------------	-------------	------------

### Hình 2.1.b: Định dạng dữ liệu truyền về Master

Trong đó, ý nghĩa các trường như sau:

- + Các trường : **LEN**, **DESTADDRESS**, **CMD** và **CRC** có ý nghĩa như trên
  - + **PARENTADDRESS**: Địa chỉ nút cha của nút hiện thời , 1 byte
  - + **DATA**: Dữ liệu gửi, 1 byte.
- ✓ Định dạng dữ liệu truyền trong phần mềm nhúng trên nút mạng với cơ chế phát quảng bá (Broadcast)

<b>Destination</b>	<b>Flags</b>	<b>DataLen</b>	<b>pDataBuffer</b>	<b>Status</b>
--------------------	--------------	----------------	--------------------	---------------

Trong đó,

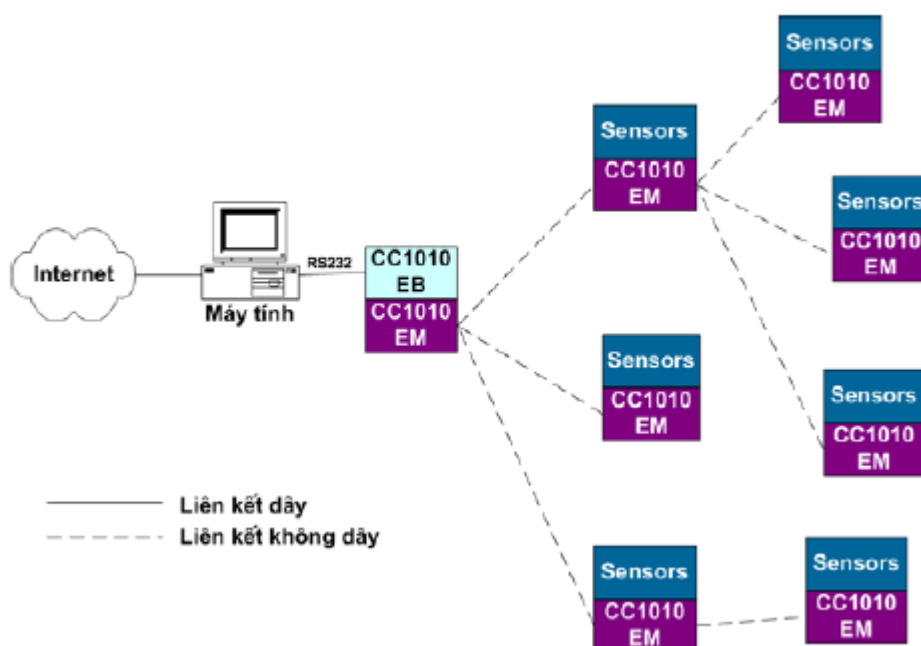
**Destination** : là trường chứa địa chỉ nút nhận, 1 byte.

**Flags** : cờ thiết lập cho nút truyền, 1 byte

**DataLen**: độ dài của **pDataBuffer**, 1 byte

**pDataBuffer**: địa chỉ của dữ liệu cần truyền, 1 byte. Độ dài của **pDataBuffer** chính là **DataLen**

**status:** là trạng thái của nút truyền (BUSY, IDLE, TRANSMITTING, TX\_STARTED)



**Hình 2.1.c: Mô hình mạng**

Thông tin dữ liệu từ khối CC1010EB được chuyển tiếp về máy tính qua cáp nối RS232, dữ liệu truyền này có dạng Text. Như vậy trên máy tính sẽ có chương trình để thu nhận dữ liệu từ cổng COM.

## **2.2. Phần mềm nhúng trên nút mạng WSN**

### **2.2.1. Giới thiệu chung**

Ngày nay, với sự phát triển nhanh chóng của ngành công nghiệp điện tử và công nghệ bán dẫn. Do đó, nhiều sản phẩm mang tính công nghệ cao đã ra đời trên cơ sở đó. Đặc biệt, các sản phẩm có kích cỡ nhỏ đã trở lên phổ biến và thông dụng hiện nay. Như vậy, vấn đề đặt ra là phải thiết kế một chương trình điều khiển trên thiết bị đó để nó có thể thực hiện được chức năng chuyên biệt. Chính vì vậy mà công nghệ lập trình nhúng ra đời. Đến nay nó đã phát triển vượt bậc và có mặt ở nhiều lĩnh vực như: công nghệ di động, điều khiển ROBOT...

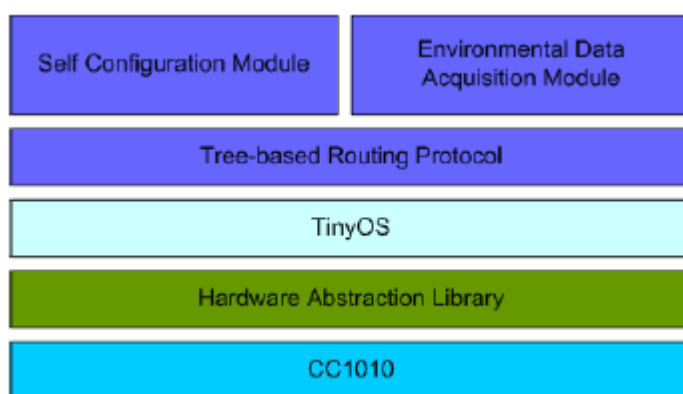
Mạng cảm nhận không dây WSN sử dụng vi điều khiển CC1010 làm nút mạng cũng có kích thước nhỏ, do đó ta cũng phải thiết kế được một chương trình nhúng trên nó, để nó có thể thu thập dữ liệu từ môi trường và truyền về trung tâm để xử lý.

### **2.2.2. Phần mềm nhúng viết cho CC1010**

Thành phần phức tạp nhất của hệ thống là phần mềm nhúng trên bộ vi điều khiển. Phần mềm được chấp nhận rộng rãi hiện nay và trở thành hệ điều hành riêng cho các vi điều khiển trong việc xây dựng WSN là Tiny OS 2.

Tiny OS có kích thước nhỏ, mã nguồn mở, dùng mô hình hướng sự kiện, với bộ lập lịch đơn giản, cho phép vi điều khiển xử lý nhiều tác vụ song song trong sự hạn chế về tài nguyên tính toán và không gian nhớ. Tiny OS sử dụng bộ lập lịch thao tác kiểu FIFO kết nối mềm dẻo giữa phần cứng và ứng dụng.

Tiny OS tạo ra khả năng giao tiếp mạnh cho các nút mạng trong WSN. Hiện tại, Tiny OS đang được nghiên cứu, chuyển đổi để làm việc với CC1010.



**Hình 2.2.2a: Kiến trúc chương trình nhúng sử dụng Tiny OS và vi điều khiển CC1010**

Dựa trên Tiny OS và các đặc trưng của CC1010, kiến trúc phần mềm nhúng cho WSN được đề xuất như biểu diễn ở hình 2.2.2 . Tầng

trung gian giữa Tiny OS và CC1010 là thư viện HAL (Hardware Abstraction Library), cho phép Tiny OS tương tác với phần cứng 4.

Tầng phía trên Tiny OS là giao thức dẫn đường trong WSN. Nó vừa cho phép truyền dữ liệu an toàn vừa hạn chế hiện tượng thất cổ chai. Tầng trên cùng là các ứng dụng đặc thù cho WSN bao gồm: module tự cấu hình mạng và tự cấu hình lại mạng, module thực hiện việc đo các thông số môi trường và chuyển về cho trạm gốc. Hai module này hoạt động theo chế độ định kỳ: sau một khoảng thời gian nhất định, nó được bộ định thời của CC1010 đánh thức và chuyển sang hoạt động: thực hiện xong nhiệm vụ, lại chuyển về chế độ nghỉ.

Thời gian cấu hình lại hệ thống và đo dữ liệu không giống nhau và phụ thuộc vào từng ứng dụng cụ thể.

Phần mềm nhúng viết cho CC1010 được viết bằng ngôn ngữ C, sử dụng các thư viện cho CC1010 do hãng CHIPCON cung cấp, chương trình được biên dịch bởi Keil  $\mu$ Vision 2.0.

Phần mềm viết cho nút mạng WSN cần thực hiện những chức năng cơ bản sau:

- Cảm nhận
- Tính toán
- Truyền thông

Một thách thức đặt ra là phải tích hợp tất cả các chức năng trên vào một vi điều khiển bị ràng buộc về mặt tài nguyên. Điều đó đòi hỏi chương trình viết càng ngắn và càng ít tốn bộ nhớ càng tốt, trong khi vẫn đảm bảo việc viết chương trình nhanh, bảo trì và nâng cấp dễ dàng.

#### **\*Các bước thuật toán cho phần mềm nhúng trên nút mạng**

**Bước 1:** Khởi tạo các tham số

- + Khởi tạo RF, ADC, TIMER
- + Khởi tạo cảm biến

**Bước 2:** Đọc thông tin dữ liệu từ kênh gắn cảm biến (AD1)

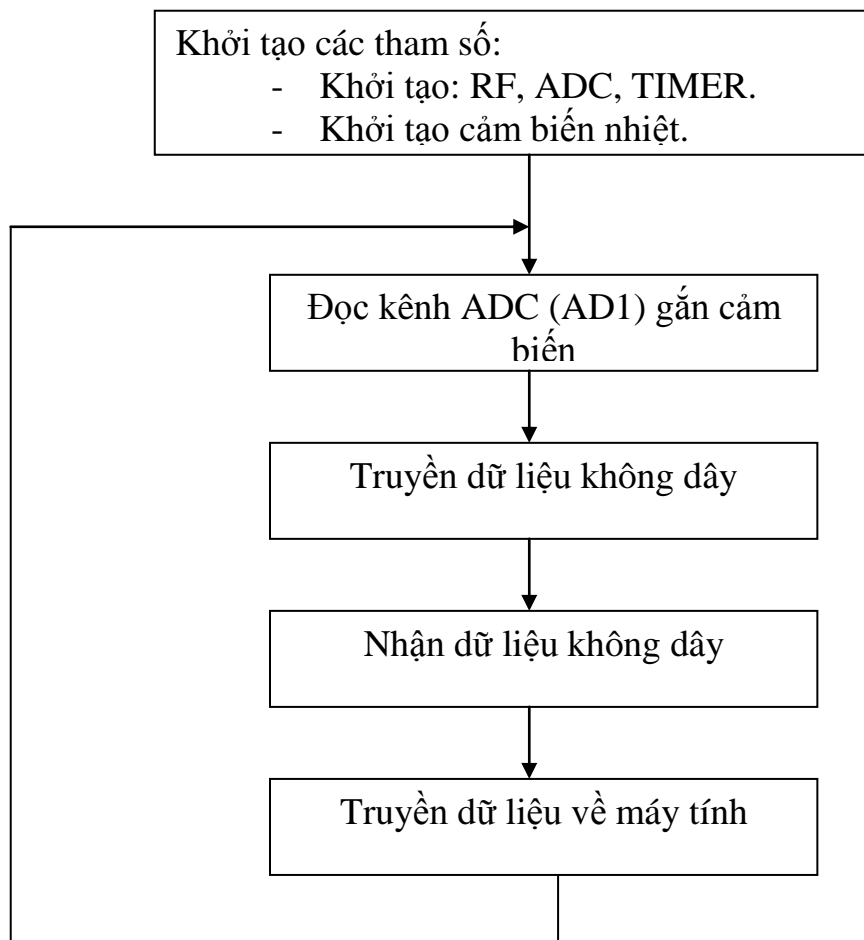


**Bước 3:** Truyền dữ liệu không dây

**Bước 4:** Nhận dữ liệu không dây

**Bước 5:** Truyền thông tin về máy tính

**\*Sơ đồ**



**Hình 2.2.2b: Sơ đồ giải thuật cho phần mềm nhúng trên CC1010EB**

### 2.3. Kết luận

Chương 2 đã đi sâu nghiên cứu về khung dữ liệu của mạng WSN và phần mềm nhúng CC1010.

Trong chương 3 sẽ nghiên cứu việc thu nhận dữ liệu từ nút mạng WSN truyền về PC và thực hiện lưu trữ thành tệp tin.

### **Chương 3: Thu nhận dữ liệu từ nút mạng WSN truyền về PC và thực hiện lưu trữ thành tệp tin**

Trong thực tế việc thu nhận dữ liệu về máy tính rất quan trọng, nó giúp chúng ta có thể phán đoán được tình hình biến đổi của môi trường dựa trên một loạt các thông tin dữ liệu thay đổi về môi trường mà ta nhận được. Tuy nhiên tùy từng theo hệ thống, tùy theo cách kết nối của hệ thống đó với trung tâm xử lý mà ta có cách thu nhận tương ứng.

Trong trong các nghiên cứu về nút mạng WSN, WSN kết nối với máy tính thông qua cáp nối RS232, do đó việc thu nhận dữ liệu sẽ từ nút mạng về máy tính sẽ thực hiện truyền qua giao tiếp RS232 (hay ta vẫn gọi là cổng COM ).

Qua quá trình nghiên cứu, phân tích bài toán của những người đi trước đã chọn ngôn ngữ VB (Visual Basic 6.0 ) làm ngôn ngữ thích hợp để viết phần mềm này.

Như vậy một cách tổng quát rằng, phần mềm trên phải giải quyết được các vấn đề sau:

- ✓ Thu nhận dữ liệu từ nút mạng WSN truyền về máy tính thông qua cáp nối RS232
  - ✓ Cho phép thay đổi khoảng thời gian để thu nhận
- Thực hiện lưu trữ thành tệp tin một cách tự động theo khoảng thời gian định trước (có thể thay đổi thông qua giao diện ).

#### **3.1. Giới thiệu về ngôn ngữ Visual Basic 6.0**

**Visual Basic (VB)** là sản phẩm phần mềm của hãng Microsoft. Phiên bản thường sử dụng là VB6.0 (năm 1998), nó là chương trình 32 bit chạy trên các hệ điều hành từ Windows 95 trở đi. Việc tiếp cận lập trình trên Microsoft Windows với **VB** tỏ ra rất đơn giản đối với cả những người chuyên nghiệp cũng như những người vừa bắt đầu.

**VB** là ngôn ngữ lập trình phát triển từ ngôn ngữ **Basic** (Beginner All-purpose Symbolic Instruction Code), một ngôn ngữ lập trình đơn giản dễ hiểu dễ học. Nó được viết ra cho những khoa học gia, những người không có thời gian học lập trình điện toán.

**VB** hỗ trợ phương pháp tạo giao diện đồ họa người dùng (Graphical User Interface viết tắt là GUI). **VB** có sẵn các bộ phận hình ảnh, gọi là *control*, chúng ta có thể tùy ý sắp đặt trên một khung hình, gọi là *form*. Rất dễ dàng để tạo ra một giao diện bắt mắt và trực quan cho người sử dụng. **VB** sử dụng phương pháp lập trình hướng đối tượng, người lập trình làm việc với các đối tượng cùng các thuộc tính, phương thức và sự kiện của nó.

***\*Khả năng lập trình với các cổng vào ra của Visual Basic.***

▪ Trong môi trường **Dos** việc truy cập trực tiếp đến cổng vào ra được thực hiện rất dễ dàng. Các ngôn ngữ từ Assembly, Pascal, C đều hỗ trợ đầy đủ các hàm truy cập trực tiếp cổng. Nhưng trong môi trường **Windows95/98** việc truy cập trực tiếp đến các cổng vào ra trở nên khó khăn. Để truy cập trực tiếp cổng có thể nhờ sự trợ giúp của thư viện liên kết động là các tệp DLL. VB và Delphi đều có thể gọi các tệp DLL. Hệ điều hành có những biện pháp bảo vệ và cần phải khai báo với hệ điều hành theo đúng cách trước khi truy cập tới phần cứng tương ứng. **Windows NT** hoạt động trong chế độ bảo vệ việc truy cập đến phần cứng là rất khó khăn. Chỉ có một khả năng hợp lệ là thông qua các tệp đệm nhưng cũng rất hạn chế chỉ ở cổng nối tiếp RS-232.

▪ **VB** không hỗ trợ các hàm truy cập trực tiếp các cổng nói chung nhưng nó có thể sử dụng các hàm được định nghĩa trong thư viện liên kết động DLL. Qua đó nó sẽ trực tiếp truy xuất đến các cổng.

▪ **VB** có hỗ trợ cho việc lập trình ghép nối máy tính qua cổng nối tiếp nhờ điều khiển truyền thông. Điều khiển truyền thông **Microsoft**

**Comm Control 6.0** được đưa vào để tham gia truyền nhận dữ liệu qua cổng nối tiếp.

### **3.2. Lập trình giao tiếp qua cổng COM bằng VB**

#### **3.2.1. Điều khiển Microsoft Comm Control 6.0**

- Để lập trình ghép nối qua cổng COM cần phải bổ xung một thành phần điều khiển truyền thông là Microsoft Comm Control 6.0. Để sử dụng được thành phần của Comm cần phải có tệp tin MSCOMM32.OCX trong thư mục hệ thống thường là C:\WINDOWS\SYSTEM ( SYSTEM32 ).

- Có 2 kiểu điều khiển và trao đổi thông tin qua cổng COM

- **Kiểu điều khiển sự kiện**

Đây là cách tốt nhất để điều khiển vào trao đổi thông tin nối tiếp, bởi vì máy tính có thể xử lý các công việc khác khi không có những sự kiện phát sinh ở cổng COM. Sự kiện **On\_Com** trả về hầu hết sự thay đổi và các lỗi truyền thông nối tiếp ví dụ như nó phát sinh khi có một kí tự được gửi đến.

- **Kiểu hỏi vòng**

Cách này cũng thường được sử dụng trong các ứng dụng nhỏ xử lý ít công việc. Trong mỗi chu kỳ của chương trình kiểm tra các giá trị của đặc tính ComEvent để xác định xem có sự kiện hoặc một lỗi nào đó xuất hiện không.

Điều khiển truyền thông nối tiếp có rất nhiều đặc tính, tìm hiểu kĩ các đặc tính đó là yếu tố quan trọng cho việc lập trình điều khiển, trao đổi thông tin qua cổng COM thành công.

#### **3.2.2. Thiết lập tham số**

Để thiết lập tham số cho cổng COM thì ta phải thiết lập một số tham số chính sau:

- ❖ **Thiết lập kết nối (Settings)**

Đặc tính **Settings** có kiểu xâu kí tự ( String ) cho phép đặt và trả lại các thông số cho cổng nối tiếp. Các thông số bao gồm: Tốc độ **baud**, **bit** chẵn lẻ (**parity**), **bit dữ liệu** và số **bit stop**.

Cú pháp câu lệnh:

**[form.]MSComm.Settings [ = paramString\$]**

Nếu **paramstring** không có thì khi tiến hành mở cổng sẽ gây lỗi.

Định dạng của xâu kí tự **paramstring** như sau:

**“BBB, P, D, S”**

Trong đó:

**BBBB** : Là tốc độ baud

**P** : Là số bit chẵn lẻ

**D** : Là số bit dữ liệu

**S** : Là số bit stop

Ví dụ : “9600,N,8,1” nghĩa là cổng COM được thiết lập với tốc độ 9600 baud, không kiểm tra chẵn lẻ, 8 bit dữ liệu, 1 bit stop.

Các thông số phải được thiết lập với các giá trị hợp lệ nếu không sẽ phát sinh lỗi:

○ **Các giá trị hợp lệ của tốc độ Baud**

5	○	7	○	1	○	1
		200		4400		
10	○	1	○	1	○	1
		800		9200		
34	○	1	○	2	○	3
		400		8400		
50	○	1	○	4	○	5
		800		7600		
	○	3	○	7	○	1

<b>00</b>	<b>200</b>	<b>15200</b>
○ <b>6</b>	○ <b>9</b>	○ <b>1</b>
<b>00</b>	<b>600</b>	<b>28000</b>

- **Giá trị chẵn lẻ (parity)**

<b>Giá trị</b>	<b>Ý nghĩa</b>
<b>E</b>	Chẵn (Even)
<b>M</b>	Dấu (Mark)
<b>N</b>	Không kiểm tra (mặc định)
<b>O</b>	Lẻ (Old)
<b>S</b>	Trống (Space)

- **Số bit dữ liệu truyền ( bit data )**

○ <b>4</b>	○ <b>7</b>
○ <b>5</b>	○ <b>8</b> (mặc
○ <b>6</b>	<b>định)</b>

- **Số bit dừng (bit stop)**

○ <b>1</b>	○ <b>2</b>
○ <b>1,5</b>	

- ❖ **Số hiệu cổng COM ( CommPort)**

Đặc tính Commport có kiểu số cho phép đặt và trả lại số hiệu của cổng truyền thông

Cú pháp câu lệnh:

**[form.]MSComm.CommPort [= portnumber%]**

Đặc tính Commport phải được thiết lập trước khi mở cổng bằng thuộc tính PortOpen

❖ **Mở cổng ( PortOpen)**

Đặc tính PortOpen có kiểu logic ( Boolean ) cho phép đặt và trả lại trạng thái của cổng truyền thông.

*Cú pháp của câu lệnh:*

**[form.]MSComm.PortOpen [ = {True|False}]**

Phải thiết lập giá trị True trước khi thao tác với cổng COM. Khi kết thúc và không sử dụng cổng COM phải được thiết lập giá trị False để đóng cổng, chương trình tự động xoá nội dung bộ đệm truyền và nhận, giải phóng cổng cho các ứng dụng khác. Trước khi dùng đặc tính này cần phải đặt thuộc tính Commport nếu không nó sẽ phát sinh lỗi.

*Ví dụ:*

**MSComm1.settings = " 57600, N, 8, 1 "**

**MSComm1.Commport=1**

**MSComm1.PortOpen=True**

Trong ví dụ này thì ta thực hiện mở cổng COM1 với tốc độ truyền là 57600 kbit/s, không kiểm tra chẵn lẻ, 1 bit stop và 8 bit dữ liệu. Và MSComm1 là tên một đối tượng truyền thông nối tiếp được tạo trên form.

### **3.2.3. Nhận dữ liệu**

Các đặc tính được sử dụng để nhập dữ liệu vào gồm: **Inputlen**, **Input**, **InBufferCount**, **InBufferSize**

○ **InputLen (Độ dài xâu kí tự đọc được )**

Đặc tính **InputLen** có kiểu số nguyên nó cho phép đặt và trả lại số kí tự mà đặc tính **Input** đọc từ bộ đệm nhận.

*Cú pháp câu lệnh:*

**[form.]MSComm1.InputLen [ = numchars%]**

Giá trị mặc định của đặc tính này bằng 0, tức là đặc tính Input sẽ đọc và xoá hết toàn bộ nội dung của bộ đệm.

- **Input (Nhập dữ liệu)**

Đặc tính Input có kiểu chuỗi ký tự (String) nó trả lại và xoá một chuỗi ký tự từ bộ đệm nhận.

*Cú pháp câu lệnh:*

**[form.]MSComm.Input**

Số ký tự đọc được bởi câu lệnh này được quy định trong đặc tính InputLen. Mặc định InputLen = 0, lệnh Input sẽ đọc và xoá cả bộ đệm nhận. Giả sử muốn đọc 5 ký tự từ bộ đệm ta có đoạn mã VB sau:

**MSComm1.InputLen = 5**

**ComData\$ = MSComm1.InPut**

- **InBufferSize (Kích thước bộ đệm nhận)**

Đặc tính InBufferSize có kiểu số nguyên (Integer) cho phép đặt và trả lại kích thước bộ đệm nhận tính theo đơn vị Byte.

*Cú pháp câu lệnh:*

**[form.]MSComm.InBufferSize [= numbyte%]**

Giá trị mặc định là 1024 Byte. Việc đặt kích thước của bộ đệm tùy thuộc vào từng ứng dụng, không nên để kích thước quá lớn (gây lãng phí bộ nhớ), cũng không để quá nhỏ (gây tràn bộ đệm)

- **InBufferCount (Số ký tự trong bộ đệm)**

Đặc tính InBufferCount có kiểu số nguyên (Integer) nó trả lại số ký tự có trong bộ đệm nhận.

*Cú pháp câu lệnh:*

**[form.]MSComm.InBufferCount [= count%]**

Có thể thực hiện xoá bộ đệm bằng cách đặt thuộc tính InputBufferCount =0



### **3.2.4. Xuất dữ liệu**

Đặc tính Output có kiểu xâu cho phép viết một xâu kí tự vào bộ đệm truyền

*Cú pháp câu lệnh:*

**[form.]MSComm.Output [ = outstring\$]**

Muốn gửi xâu kí tự “Hello” ra cổng nối tiếp sử dụng câu lệnh:

**MSComm1.Output = “Hello”**

- **OutBufferSize (Kích thước bộ đệm truyền )**

Đặc tính OutBufferSize có kiểu số nó cho phép đặt và trả lại kích thước của xâu kí tự tối đa trong bộ đệm truyền.

*Cú pháp câu lệnh:*

**[form.]MSComm.OutBufferSize [ = numbyte%]**

Giá trị mặc định của OutBufferSize là 512 byte. Việc đặt kích thước của bộ đệm tùy thuộc vào từng ứng dụng, không nên để kích thước quá lớn (lãng phí bộ nhớ) cũng không để quá nhỏ (gây tràn).

- **OutBufferCount ( Số ký tự trong bộ đệm )**

Đặc tính OutBufferCount có kiểu số nguyên nó trả lại số kí tự có trong bộ đệm truyền

*Cú pháp câu lệnh:*

**[form.]MSComm.OutBufferCout [ = 0]**

Có thể xoá nội dung bộ đệm truyền bằng cách đặt giá trị OutBufferCount bằng 0.

### **3.2.5. Thuộc tính khác**

Phần này trình bày thêm một số thuộc tính khác thường dùng của điều khiển truyền thông nối tiếp.

- **Break.**

Đặc tính này cho phép đặt hoặc xoá trạng thái tín hiệu BREAK.

*Cú pháp câu lệnh:*

**[form.]MSComm.Break [ = {True| False}]**

Giá trị True tính hiệu được gửi đi, nó làm dừng quá trình truyền dữ liệu và quá trình truyền chỉ tiếp tục khi giá trị bằng False.

- **CTSHolding, DSRHolding, CDHolding.**

Đặc tính này cho phép đặt và xác định trạng thái các đường tương ứng CTS, DSR, CD.

*Cú pháp câu lệnh:*

**[form.]MSComm.CTSHolding[ = {True|False}]**

**[form.]MSComm.DSRHolding[ = {True|False}]**

**[form.]MSComm.CDHolding[ = {True|False}]**

Giá trị True đưa ra các đường lên trạng thái tích cực CTS xuống mức thấp, DSR, CD lên mức cao.

- **DTREnable, RTSEnable.**

Đặc tính này cho phép xác định trạng thái tín hiệu tương ứng DTR và RTS

*Cú pháp câu lệnh:*

**[form.]MSComm.DTREnable[ = {True|False}]**

**[form.]MSComm.DTREnable[ = {True|False}]**

Thông số được đặt True đường dẫn tương ứng lên mức cao, giá trị False nó xuống mức thấp.

- **CDTimeout, CTSTimeout, DSRTIMEout.**

Các đặc tính này đặt và trả lại giá trị cực đại của thời gian (mili giây) đợi các tín hiệu tương ứng CD, CTS, DSR.

*Cú pháp câu lệnh:*

**[form.]MSComm.CDTimeout[ = milliseconds&]**

**[form.]MSComm.CTSTimeout[ = milliseconds&]**

**[form.]MSComm.DSRTIMEout[ = milliseconds&]**

Khi quá thời gian chương trình phát sinh sự kiện On\_Comm() mã lỗi chứa trong CommEvent.

Các đặc tính này được sử dụng kết hợp với các đặc tính CDHolding, CTSHolding, DSRHolding để bẫy các lỗi.

### **3.2.6. Sự kiện On\_Comm()**

#### **○ Đặc tính Ssthreshold.**

Đặc tính này cho phép đặt và trả lại số lượng tối thiểu kí tự trong bộ đệm truyền để không xuất hiện sự kiện On\_Comm. Điều đó có nghĩa là khi số kí tự hiện có trong bộ đệm ít hơn giá trị của đặc tính này thì phát sinh sự kiện On\_Comm.

*Cú pháp câu lệnh:*

**[form.]MSComm.Ssthreshold[ = numChars%]**

Đặc tính Ssthreshold được đặt tất cả bằng 0 sẽ không cho phép sự kiện On\_Comm. Còn nếu đặt bằng 1 thì khi bộ đệm truyền rỗng nó sẽ phát sinh sự kiện On\_Comm.

#### **○ Đặc tính Rthreshold.**

Đặc tính cho phép đặt trả lại số kí tự nhận được trong bộ đệm nhận khi phát sinh sự kiện On\_Comm

*Cú pháp câu lệnh:*

**[form.]MSComm.Rthreshold[ = numChars%]**

Giá trị bằng 0 làm mất khả năng phát sinh sự kiện On\_Comm. Nếu giá trị của Rthreshold là 1 sự kiện On\_Comm phát sinh khi có 2 kí tự vào bộ đệm.

#### **○ Đặc tính CommEvent:**

Đặc tính này trả hầu hết sự kiện hoặc lỗi truyền thông.

*Cú pháp câu lệnh:*

**[form.]MSComm.CommEvent**

Giá trị mặc định CommEvent với các lỗi truyền thông.

<b>Thiết lập</b>	<b>Giá trị</b>	<b>Mô tả</b>
ComBreak	1001	Nhận được tín hiệu Break.

ComCTSTO	1002	Chờ tín hiệu CTS quá giờ(Timeout).
ComDSRTO	1003	Chờ tín hiệu DSR quá giờ(Timeout).
ComFrame	1004	Sai số khung truyền.
ComOverrun	1006	Tràn cổng, bộ đệm bị tràn, kí tự nhận được tiếp theo viết đè lên kí tự trước đó trong bộ đệm.
ComCDTO	1007	Chờ tín hiệu CD quá giờ(Timeout).
ComRxOver	1008	Bộ đệm nhận bị tràn.
ComRxParity	1009	Lỗi chẵn lẻ.
ComTxFull	1010	Bộ đệm truyền đầy.

Giá trị đặc tính CommEvent với các sự kiện truyền thông.

Thiết lập	Giá trị	Mô tả
ComEvSend	1	Kí tự đã được gửi.
ComEvReceive	2	Kí tự đã được nhận.
ComEvCTS	3	Sự thay đổi trên đường CTS.
ComEvDSR	4	Sự thay đổi trên đường DSR từ cao sang thấp.
ComEvCD	5	Sự thay đổi trên đường CD.
ComEvRing	6	Tiếng chuông được phát hiện
ComEvEOF	7	EOF kí tự nhận.

○ **Sự kiện On\_Comm().**

Khi giá trị của đặc tính CommEvent thay đổi thì sự kiện On\_Comm lập tức phát sinh

*Sự kiện:*

**Private Sub On\_Comm()**

**End Sub**

Sự kiện On\_Comm phát sinh thì đặc tính CommEvent chứa mã số của lỗi hay sự kiện truyền thông nào đó.

### **3.3. Làm thế nào để thu nhận dữ liệu từ cổng COM và lưu trữ kết quả thành tập tin trên máy tính**

❖ Như ta tìm hiểu về ngôn ngữ **VB 6.0** như phần trên thì để có thể đọc thông tin từ cổng COM thì ta sử dụng điều khiển MSCOMM.

Mặt khác để có thể thu nhận thông tin một cách tự động thì ta có thể sử dụng điều khiển Timer (định thời). Điều khiển Timer được đưa vào ứng dụng để xác định khoảng thời gian lấy mẫu. Thuộc tính **Interval** của điều khiển Timer sẽ thực hiện việc đếm thời gian, đơn vị của thuộc tính là **Mili giây**. Giả sử như khi ta đặt thuộc tính Interval là 100 thì cứ sau 100 mili giây thì xuất hiện một sự kiện Timer. Việc kích hoạt hay không kích hoạt là nhờ vào thuộc tính Enable của Timer. Đây là thuộc tính kiểu Logic, tức là nó nhận giá trị True/ False tương ứng với kích hoạt / không kích hoạt.

❖ Để có thể đọc thông tin từ bộ đệm cổng COM thì ta thiết lập sự kiện On\_Comm

**Private Sub MSComm1\_OnComm()**

**Dim InBuff As String**

**Select Case MSComm1.CommEvent**

.....

**Case comEvReceive**

**InBuff = MSComm1.Input**

Call HandleInput(InBuff)

.....

End Select

End Sub

Như vậy khi thuộc tính `Comm_Event` có giá trị **comEvReceive** thì ta tiến hành đọc thông tin bộ đệm bằng câu lệnh **InBuff = MSComm1.Input** và gọi hàm **HandleInput(InBuff)** để hiển thị

❖ **Vấn đề** lưu trữ tập tin một cách tự động cũng dựa trên bộ định thời Timer.

Ý tưởng của việc lưu trữ tập tin một cách tự động này khá đơn giản. Ta chỉ cần đưa đoạn mã thực hiện việc lưu trữ tập tin vào trong phần điều khiển Timer.

**Private Sub Timer1\_Timer()**

‘ **đoạn mã thực hiện lưu trữ tập tin**

**End sub**

Như vậy ta chỉ cần chỉ định thời gian lưu trữ tập tin là được, điều này đã được giới thiệu ở phần Timer trên.

Một vấn đề đặt ra nữa là việc xây dựng đoạn mã nguồn để lưu trữ tập tin như thế nào ?

Phần này cũng khá đơn giản, trong VB hỗ trợ việc thao tác trên các tập tin văn bản dễ dàng.

Dim **tenfile, sofile, temmp** As Integer

**tenfile = App.Path**

If **Right(tenfile, 1) <> "\"** Then **tenfile = tenfile & "\"**

**tenfile = tenfile & txtfilename & ".txt"**

**sofile = FreeFile**

*Open tenfile For Output As sofile*

*Print #sofile, txtdata.Text*

*Close sofile*

Như vậy để mở file và ghi thông tin ta thực hiện câu lệnh

*Open tenfile For Output As sofile*

*Print #sofile, txtdata.Text*

❖ **Thuật toán chương trình**

**Bước 1:** Khởi tạo các tham số

+ Khởi tạo thiết lập các tham số kết nối tới cổng COM

+ Khởi tạo bộ định thời Timer.

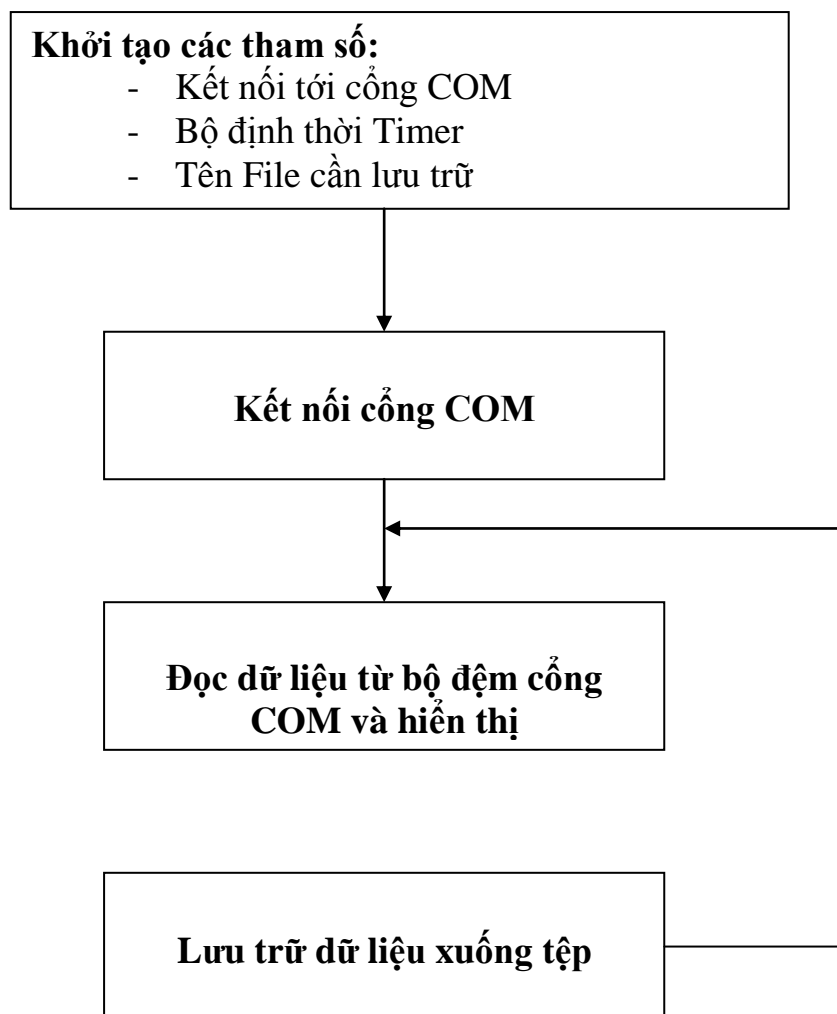
+ Chỉ thị tên File cần lưu trữ.

**Bước 2:** Thực hiện kết nối tới cổng COM

**Bước 3:** Đọc dữ liệu từ bộ đệm cổng COM và hiển thị.

**Bước 4:** Thực hiện lưu trữ dữ liệu xuống tập tin chỉ định

❖ **Sơ đồ**



**Hình SD1: Sơ đồ thuật toán lưu trữ dữ liệu từ mạng WSN**

Như vậy chương 3 đã tìm hiểu cách thu nhận dữ liệu từ mạng WSN. Ở chương 4 tiếp theo sẽ giới thiệu việc nghiên cứu xây dựng website, tự động truyền tải thông tin của nút mạng lên Website

**Chương 4: Nghiên cứu xây dựng Website, tự động truyền tải thông tin của nút mạng WSN lên Website.**

**4.1. Giới thiệu về công nghệ Website**

❖ Tiền thân của mạng Internet ngày nay là mạng ARPANET. Vào năm 1960 khi một cơ quan của Bộ Quốc phòng Mỹ, cơ quan quản lý dự án nghiên cứu phát triển (ARPA) đề nghị liên kết 4 địa điểm đầu tiên



vào tháng 7 năm 1968. Bốn địa điểm đầu tiên đó là Viện Nghiên cứu Stanford, Trường Đại học tổng hợp California ở Los Angeles, Úc - Santa Barbara và trường Đại học tổng hợp Utah. Đó là mạng liên khu vực (Wide area Network) hay mạng Wan đầu tiên được xây dựng (mặc dù nó nhỏ hơn nhiều so với các mạng WAN ngày nay). Bốn địa điểm trên được nối thành mạng vào năm 1969 đã đánh dấu sự ra đời của Internet ngày nay : mạng được biết đến dưới cái tên ARPANET đã hình thành.

❖ Nếu xét về thời gian thì thuật ngữ Internet xuất hiện lần đầu vào khoảng năm 1974. Lúc đó mạng vẫn được gọi là ARPANET. Năm 1983, giao thức TCP/IP chính thức được coi như một chuẩn đối với ngành quân sự Mỹ, và tất cả các máy tính nối với ARPANET phải sử dụng chuẩn mới này. Năm 1984, ARPANET đã được chia ra thành hai phần : phần thứ nhất vẫn được gọi là ARPANET, dành cho việc nghiên cứu và phát triển; phần thứ hai được gọi là MILNET, là mạng dùng cho các mục đích quân sự.

❖ Giao thức TCP/IP ngày càng thể hiện rõ các điểm mạnh của mình, quan trọng nhất là khả năng liên kết các mạng khác với nhau lại một cách dễ dàng. Chính điều này cùng với các chính sách mở cửa đã cho phép các mạng dùng cho nghiên cứu và thương mại kết nối được với ARPANET, thúc đẩy việc tạo ra một siêu mạng (SuperNetwork). Nhưng năm 1980 ARPANET được đánh giá là mạng trụ cột của mạng Internet. Mốc lịch sử quan trọng của Internet được chọn vào giữa thập kỷ 1980, khi tổ chức khoa học quốc gia Mỹ NSF thành lập mạng liên kết các trung tâm máy tính lớn với nhau gọi là NSFNET. Nhiều doanh nghiệp đã chuyển từ ARPANET sang NSFNET và do đó sau gần 20 năm hoạt động, ARPANET không còn hiệu quả nữa và đã ngừng hoạt động vào khoảng năm 1990

❖ Sự hình thành mạng backbone của NSFNET và những mạng vùng khác đã tạo ra một môi trường thuận lợi cho sự phát triển của

Internet. Tới năm 1995, NSFNET thu lại thành một mạng nghiên cứu còn Internet thì vẫn tiếp tục phát triển.

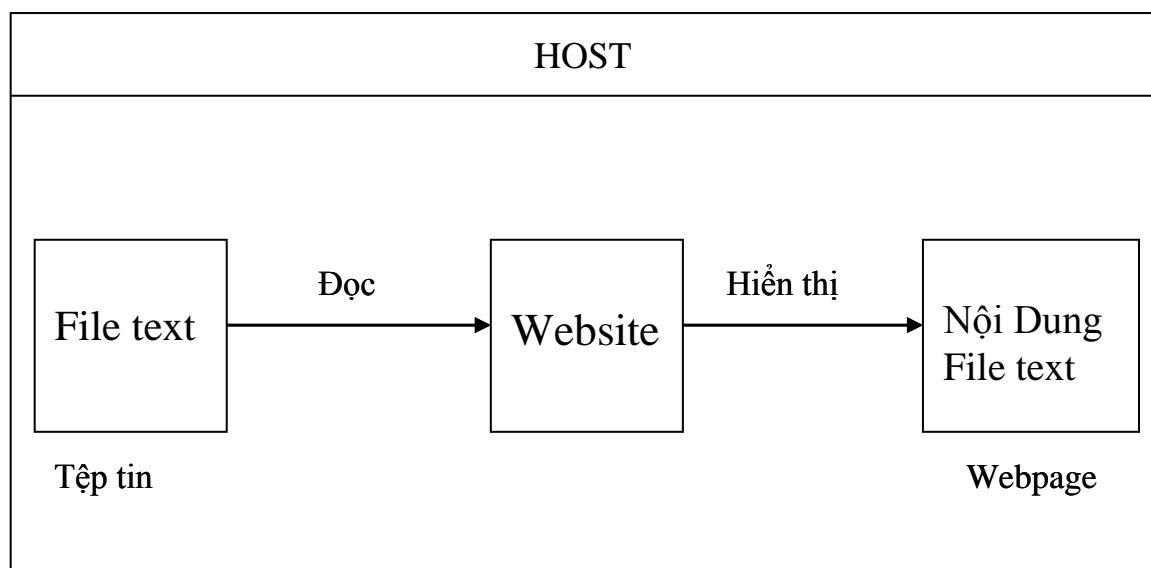
❖ Sự ra đời và phát triển nhanh chóng của Internet đã thúc đẩy sự phát triển của công nghệ Web. Website ra đời giúp cho các tổ chức chính phủ cũng như các thương nhân, thậm chí cả những cá nhân có khả năng quảng bá thông tin cho mọi người một cách nhanh chóng trên toàn thế giới. Điều đó góp phần đáng kể vào sự phát triển của kinh tế thế giới .

❖ Sự ra đời của Website không chỉ để quảng bá thông tin về các doanh nghiệp hay các tổ chức chính phủ. Nó còn cho thấy vai trò to lớn của nó trong các hệ thống giám sát, thu thập dữ liệu. Mạng cảm nhận không dây ứng dụng thu thập dữ liệu môi trường cũng không phải là một ngoại lệ. Việc giám sát kết quả thu thập được từ mạng WSN trên hệ thống Website sẽ làm cho hệ thống mang tính chất toàn cầu và có tính khả chuyển cao.

#### **4.2. Triển khai ứng dụng trên Website cho hệ thống mạng**

##### **WSN**

❖ **Bài toán đặt ra** là làm sao chương trình ứng dụng trên Website sẽ thực hiện việc đọc thông tin từ một file văn bản (File này chứa kết quả thu được từ mạng WSN trên máy tính và được đưa lên HOST một cách tự động) để hiển thị nội dung tập tin đó lên Website một cách tự động.



**Hình 4.2: Sơ đồ chương trình**

### **4.3. Thực hiện tự động quảng bá tệp tin trên mạng**

Ngày nay mạng Internet đã trở nên phổ biến khắp mọi nơi trên thế giới, mọi thông tin của chúng ta đều có thể chia sẻ lẫn nhau trên Internet. Với các ứng dụng mang tính chất quảng bá và cần yêu cầu quan sát thì việc kết hợp hệ thống đó với môi trường Internet là rất quan trọng và cần thiết.

Mạng cảm nhận không dây ứng dụng thu thập dữ liệu môi trường cũng thuộc lĩnh vực ứng dụng đó.

Việc thu nhận kết quả từ nút mạng về máy tính để xử lý đã ngày trở lên phổ biến. Thật vậy, trong thực tế việc đưa thông tin tự động lên mạng rất hữu ích cho việc quan sát từ xa:

- ✓ Giảm thiểu thời gian trực tại hệ thống để quan sát kết quả
- ✓ Hoàn toàn có thể quan sát từ xa, thậm chí ở bất kỳ đâu trên thế giới miễn là ở đó hỗ trợ Internet.
- ✓ Giảm thiểu nguồn nhân lực cho việc quan sát hệ thống.
- ✓ Hệ thống trở lên mang tính toàn cầu vì thông tin lúc này không chỉ đơn thuần ở trên máy tính.

**Bài toán** đặt ra là: thông tin dữ liệu từ nút mạng WSN truyền về máy tính qua cáp nối RS232 được lưu trữ trên máy tính thành tập tin, sau đó chương trình phải thực hiện tự động đưa tập tin đó lên mạng (HOST).



**Hình 4.3: Mô hình quảng bá tập tin trên mạng**

❖ **Thuật toán chương trình quảng bá dữ liệu**

**Bước 1:** Khởi tạo các tham số

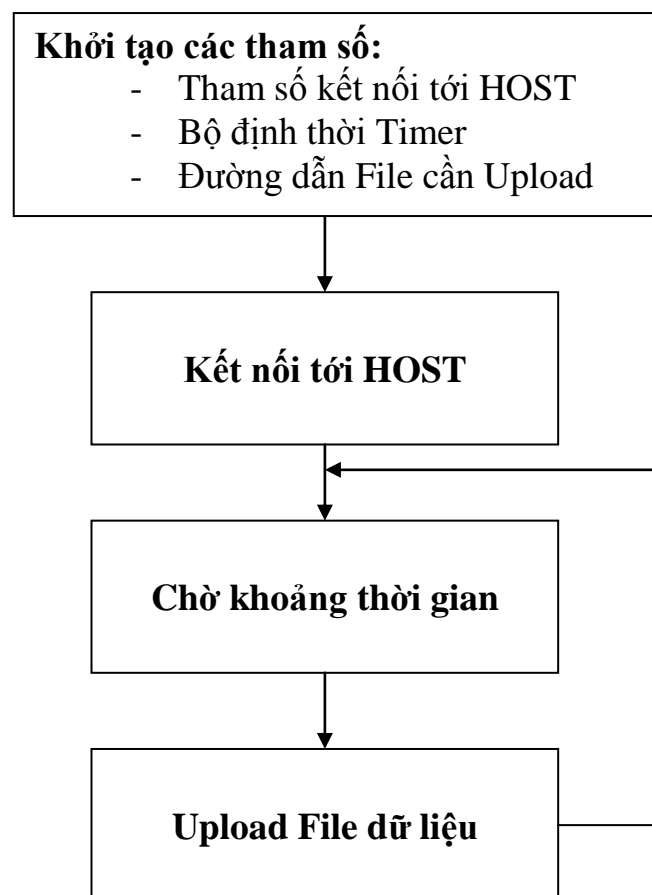
- + Thiết lập các tham số kết nối tới HOST.
- + Khởi tạo bộ định thời Timer.
- + Chỉ định đường dẫn cho File cần Upload

**Bước 2:** Thực hiện kết nối tới HOST

**Bước 3:** Chờ khoảng thời gian.

**Bước 4:** Upload file dữ liệu lên HOST.

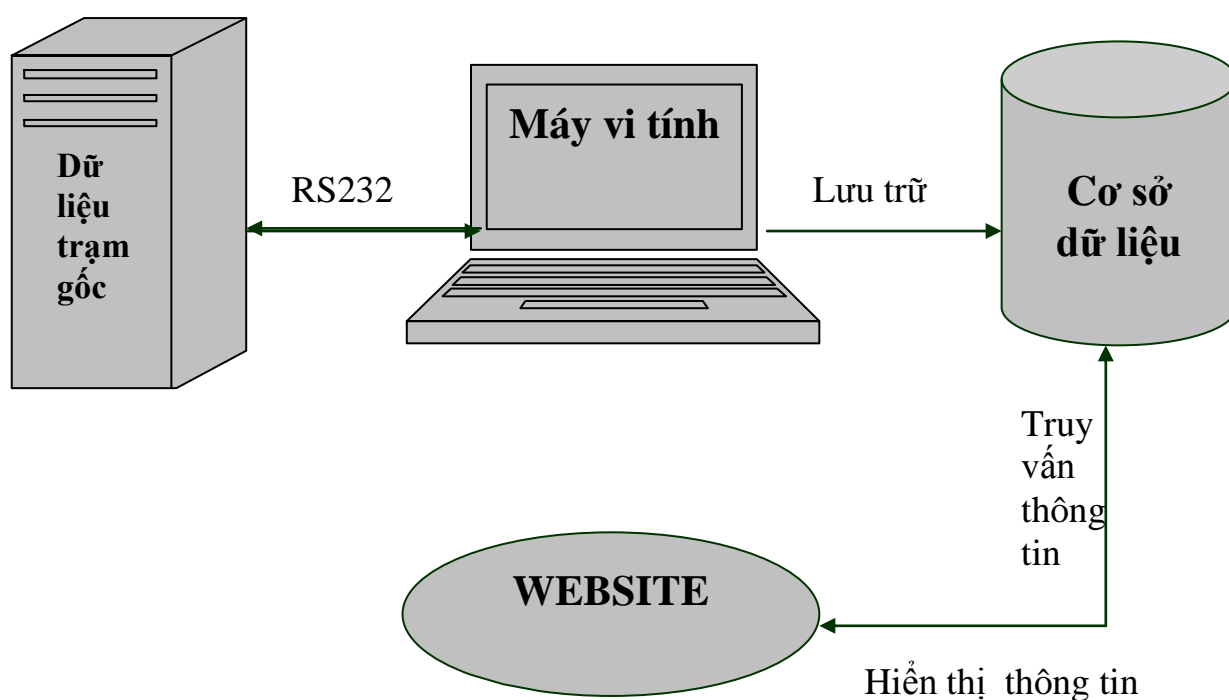
❖ **Sơ đồ**



### **Hình SD2:Sơ đồ thuật toán quảng bá tập tin tự động**

Khi đã upload file dữ liệu thành công thì chương trình ứng dụng trên Website sẽ thực hiện đọc thông tin từ một file văn bản( file chứa kết quả thu được từ mạng WSN trên PC và được đưa lên HOST một cách tự động) để hiển thị nội dung tập tin đó lên Website một cách tự động.

Sơ đồ



## MÔ HÌNH HỆ THỐNG

### 4.4. Lựa chọn ngôn ngữ lập trình website

Trong bài luận văn này em sử dụng ngôn ngữ lập trình asp để xây dựng website.

- ASP (Active Server Pages) là một môi trường lập trình cung cấp cho việc kết hợp HTML, ngôn ngữ kịch bản (Scripting) như VBScript, Javacript, và các thành phần được viết trong các ngôn ngữ nhằm tạo ra một ứng dụng Internet mạnh mẽ và hoàn chỉnh.

Như tên gọi của nó, ASP giúp người xây dựng ứng dụng web tạo ra các trang web có nội dung linh hoạt. Với các người dùng khác nhau khi truy cập vào những trang web này có thể sẽ nhận được các kết quả khác nhau. Nhờ những đối tượng có sẵn(Built\_in Object) và khả năng hỗ trợ các ngôn ngữ script như VBScript và Jscript, ASP giúp người xây dựng ứng dụng dễ dàng và nhanh chóng tạo ra các trang web chất lượng.

Những tính năng trên giúp người phát triển ứng dụng nhanh chóng tiếp cận ngôn ngữ mới, điều này là một ưu điểm không nhỏ của ASP.

- Cấu trúc của một trang ASP: Trang ASP đơn giản là một trang văn bản với phần mở rộng là .asp, gồm có 3 phần :

- + Văn bản (Text)

- + HTML tag (HTML : Hypertext Markup Language)

- + Các đoạn script asp

Khi thêm 1 đoạn script vào HTML , ASP dùng dấu phân cách (delimiters) để phân biệt giữa đoạn HTML và đoạn ASP. <% bắt đầu đoạn script và %> để kết thúc đoạn script. Có thể xem trang ASP như một trang HTML có bổ sung các ASP Script Command

- Các ưu điểm của asp:

- + ASP bổ sung cho các công nghệ đã có từ trước như CGI (Common Gateway Interface), Giúp người dùng xây dựng các ứng dụng web với những tính năng sinh động. Tuy nhiên, khác với CGI, các trang ASP không cần phải hợp dịch.

- + Dễ dàng tương thích với các công nghệ của MicroSoft. ASP sử dụng ActiveX Data Object (ADO) để thao tác với cơ sở dữ liệu hết sức tiện lợi.

- + Với những gì ASP cung cấp, các nhà phát triển ứng dụng web dễ dàng tiếp cận công nghệ này và nhanh chóng tạo ra các sản phẩm có giá trị. Điều này hết sức có ý nghĩa trong điều kiện phát triển như vũ bão của tin học ngày nay. Nó góp phần tạo nên một đội ngũ lập trình web lớn mạnh.

- + ASP có tính năng mở. Nó cho phép các nhà lập trình xây dựng các component và đăng ký sử dụng dễ dàng. Hay nói cách khác ASP có tính năng COM(Component Object Model).

Thật vậy, nếu bạn có một trang web, mỗi khi có người đến thăm trang web này thì sẽ có một thành phần thay đổi( cụ thể tăng lên một đơn

vi) khi đó, dùng HTML không thể đáp ứng được yêu cầu nêu ra. Bạn sẽ cần một ngôn ngữ lập trình nào đó. Javascript là một client script - nghĩa là nó tương tác trực tiếp ở mức độ client, không cần thông qua server. Còn asp là server script nghĩa là: các mã lệnh được người lập trình viết ra sẽ được server đọc, dịch ra thành các mã HTML và trả về cho trình duyệt (không như các ngôn ngữ lập trình khác - not web - là trả về mã máy). Dễ dàng nhận thấy được khi sử dụng ASP sẽ không bị lộ mã nguồn (asp code). Vì khi ai đó view source trang web của bạn, sẽ chỉ nhận được các mã HTML do server trả về sau khi thông dịch các mã asp.



## **KẾT LUẬN**

Trên cơ sở nghiên cứu tổng quan về mạng cảm nhận không dây, thu nhận dữ liệu từ nút mạng WSN truyền về PC và thực hiện lưu trữ thành tệp tin, đặc biệt đi sâu vào nghiên cứu về khai thác ứng dụng thu thập và quảng bá dữ liệu tự động, em đã thực hiện đề tài: “ *Xây dựng Website quảng bá thông tin mạng WSN* ”.

Trong quá trình làm khoá luận này, các vấn đề sau đây đã được em nghiên cứu và thực hiện:

**Về mặt lý thuyết**, tìm hiểu và nghiên cứu được kiến trúc tổng quát và chỉ ra các yêu cầu chủ yếu đạt được xây dựng hệ thống mạng WSN, tìm hiểu khung dữ liệu từ mạng WSN, ứng dụng viết chương trình thu nhận dữ liệu trên máy tính thành tệp tin và trên Website.

**Về mặt thực nghiệm**, em đã thiết kế thành công chương trình ứng dụng trong khuôn khổ của luận văn với những ưu điểm sau:

- + Giao diện thuận tiện, cập nhập thông tin nhanh chóng thuận lợi
- + Thực hiện được vấn đề cập nhập thông tin dưới dạng file text có khả năng quảng bá tự động trên website

**Hạn chế của đề tài**, chương trình ứng dụng chưa có khả năng tự động cập nhập thông tin dưới nhiều dạng phong phú, đa dạng, khả năng thu nhiều thông tin từ nhiều loại đầu đo vẫn chưa thực hiện được.

**Hướng ứng dụng của đề tài**, trong quá trình thực hiện đề tài em thấy rằng cần phải đi khai thác ứng dụng quảng bá dữ liệu tự động trên Website. Điều đó giúp cho một số hệ thống giám sát trở lên khả chuyên và phát triển rộng rãi hơn.

**Hướng phát triển của đề tài**, trong tương lai em mong rằng chương trình ứng dụng quảng bá thông tin mạng WSN sẽ được sử dụng nhiều trong các lĩnh vực hơn với tiện ích tự động cập nhập thông tin từ nhiều loại đầu đo với các thông tin đa dạng và phong phú không chỉ dừng lại ở file text. Ngoài ra ứng dụng thực tế của hệ thống mạng WSN không

chỉ được thực hiện trên Website mà sẽ còn được mở rộng và lập trình ứng dụng trên nhiều phương tiện thông tin khác.

## **TÀI LIỆU THAM KHẢO**

- [1]. Chipcon, *CC1010IDE Manual*, [www.chipcon.com](http://www.chipcon.com)
- [2]. Chipcon, *CC1010 Datasheet*, [www.chipcon.com](http://www.chipcon.com)
- [3]. PGS.TS Vương Đạo Vy - Mạng và truyền dữ liệu.
- [4]. Lập trình Visual Basic 6.0 – NXB Lao Động Xã Hội
- [5]. ASP database-ấn bản năm 2000- NXB Sài Gòn