 ĐỒ ÁN TỐT NGHIỆP CƠ ĐIỆN tử

**CHƯƠNG 1. GIỚI THIỆU HỆ THỐNG TRONG NHÀ THÔNG MINH**

**1.1.      TỔNG QUAN**

-          Nhà thông minh là ngôi nhà được trang bị các hệ thống tự động thông minh cùng với cách bố trí hợp lý, các hệ thống này có khả năng tự điều phối các hoạt động trong ngôi nhà theo thói quen sinh hoạt và nhu cầu cá nhân của gia chủ. Chúng ta cũng có thể hiểu ngôi nhà thông minh là một hệ thống chỉnh thể mà trong đó, tất cả các thiết bị điện tử gia dụng đều được liên kết với thiết bị điều khiển trung tâm và có thế phối hợp với nhau để cùng thực hiện một chức năng. Các thiết bị này có thể tự đưa ra cách xử lý tình huống được lập trình trước, hoặc là được điều khiển và giám sát từ xa.

-          Giải pháp nhà thông minh sẽ biến những món đồ điện tử bình thường trong ngôi nhà trở nên thông minh và gần gũi với người dùng hơn, chúng được kiểm soát thông qua các thiết bị truyền thông như điều khiển từ xa, điện thoại di động… ngôi nhà thông minh đơn giản nhất có thể được hình dung bao gồm một mạng điều khiển liên kết một số lượng cố định các thiết bị điện, điện tử gia dụng trong ngôi nhà và chúng được điều khiển thông qua một chiếc điều khiển từ xa. Chỉ với kết nối đơn giản như trên cũng đủ để hài lòng một số lượng lớn các cá nhân có nhu cầu nhà thông minh ở mức trung bình.

-          Vậy liệu nhà thông minh có làm thay đổi các thói quen vốn đã rất gắn bó từ trước đến nay với hầu hết mọi người?.

-          Chúng ta đều biết phần lớn căn hộ từ trung bình đến cao cấp đều sử dụng các loại điều khiển từ xa để điều khiển máy lạnh, ti vi…còn lại phần lớn các thiết bị khác như hệ thống đèn, bình nước nóng lạnh…phải điều khiển bằng tay. Những việc như vậy đôi lúc sẽ đem lại sự bất tiện, khi mà chúng ta mong muốn có một sự tiện nghi và thoải mái hơn, vừa có thể tận hưởng nằm trên giường coi ti vi vừa có thể kiểm soát được hệ thống các thiết bị trong nhà chỉ với một chiếc smartphone hay máy tính bảng.

**1.2.      CÁC THÀNH PHẦN CƠ BẢN TRONG HỆ THỐNG NGÔI NHÀ THÔNG MINH**

-          Mô hình mô phỏng ngôi nhà trong thực tế và sự phân bố khá hợp lý các hệ thống đi kèm.

-          Việc bố trí rất quan trọng, những thiết bị không sử dụng nên sắp xếp vào chỗ hợp lý tránh gây bất tiện trong sinh hoạt. Vì vậy, khi thiết kế ngôi nhà thông minh cần quan tâm đến sự thay đổi mềm dẻo trong cách thức lắp đặt và cấu hình sử dụng.

**1.2.1.            Hệ thống quản lý chiếu sáng**

-          Các thiết bị chiếu sáng như bóng đèn sợi đốt, đèn neon, đèn ngủ, trang trí…được sử dụng rất nhiều. Vì vậy nếu phối hợp chiếu sáng không hợp lý sẽ dẫn tới bị “ô nhiễm” ánh sáng. Ngoài ra, việc chiếu sáng như vậy còn gây lãng phí điện, giảm tuổi thọ thiết bị. Bên cạnh đó số lượng đèn dùng để chiếu sáng là khá lớn, gia chủ sẽ gặp những bất tiện nhỏ trong việc bật tắt, điều chỉnh độ sáng cho phù hợp.

-          Hệ thống chiếu sáng sẽ được tích hợp chung với các hệ thống khác hoặc sẽ được tách riêng ra để điều khiển độc lập. Các giải pháp đều nhằm tối ưu hóa hệ thống và giúp gia chủ điều khiển dễ dàng hơn. Các giải pháp kết hợp sẽ được tính đến để tự động hóa tới mức tối đa.

**1.2.2.            Hệ thống kiểm soát vào ra**

-          Khi gia chủ đi vắng, việc kiểm soát các hệ thống vào ra trong ngôi nhà là rất quan trọng, giúp đề phòng trộm v.v… Ngôi nhà thông minh cung cấp hệ thống kiểm soát vào ra cho phép chủ nhà quản lý và cấp quyền “đăng nhập” cho các thành viên trong gia đình và người thân.

-          Hệ thống cửa ra vào ở các phòng sẽ được lắp đặt các khóa vân tay hoặc khóa phím v.v… nhằm nhận dạng người trong nhà hoặc khách để cấp quyền “đăng nhập”. Ngoài ra, còn có thể dùng hệ thống nhận diện khuôn mặt hay giọng nói tùy vào phòng riêng của mỗi người.

-          Mỗi khi có sự kiện mới, hệ thống kiểm soát ra vào này cũng sẽ kích hoạt các hệ thống khác để lưu giữ các thay đổi do người dùng tạo ra.

**1.2.3.            Hệ thống quan sát, thông tin liên lạc**

-          Một ngôi nhà bình thường sẽ có từ 4 đến 5 phòng kín, và do vậy sẽ có một vài vấn đề khó khăn khi giao tiếp từ phòng này sang phòng khác. Một hệ thống thông tin liên lạc nội bộ có thể giúp giải quyết vấn đề này.

-          Hệ thống liên lạc nội bộ đơn giản có thể là các điện thoại cố định hoặc điện thoại mẹ bồng con. Ngoài chức năng liên lạc trong nhà, hệ thống này cần được kết nối với mạng điện thoại để tiện cho việc giao tiếp và công việc hơn, để làm việc này cần đến một bộ chuyển kênh.

-          Hệ thống quan sát sẽ giúp việc kiểm soát an ninh, người vào/ra ngôi nhà…giúp cho gia chủ nhận diện khách nhanh chóng thông qua camera.

**1.2.4.            Hệ thống giải trí đa phương tiện**

-          Ngôi nhà là nơi sinh hoạt của một gia đình có thể gồm nhiều thế hệ và mỗi thế hệ lại có nhu cầu giải trí khác nhau. Do đó, một hệ thống giải trí đa phương tiện sẽ cung cấp cho các thành viên những hoạt động giải trí phù hợp…

**1.2.5.            Hệ thống quản lý cấp điện, nước, gas**

-          Đối với một ngôi nhà bình thường thì việc cung cấp và đo lường các chỉ số điện nước đều phải thông qua các cơ quan nhà nước.

-          Ngôi nhà thông minh cung cấp giải pháp nhằm đo lường và báo lại các thông số điện, nước thường xuyên, kết hợp với hệ thống quản lý chiếu sáng và hệ thống kiểm soát vào ra, từ đó tự động bật/tắt các thiết bị trong nhà nhằm tiết kiệm năng lượng. Ngoài ra, các cảm biến sẽ giúp hạn chế và cảnh báo các nguy cơ khác như rò rỉ gas, mực nước ở bể chứa thấp, bể đường ống nước, cháy chập điện…

**1.2.6.            Hệ thống cảm biến và báo động, báo cháy**

-          Hệ thống các cảm biến là thành phần quan trọng trong bất kì hệ thống nào của ngôi nhà, các cảm biến có nhiệm vụ gửi các thông số đo được về cho bộ xử lý trung tâm để có giải pháp phù hợp với từng gói dữ liệu và xử lý từng tình huống tương ứng.

-          Các cảm biến cơ bản như cảm biến nhiệt độ, độ ẩm, cảm biến gas, cảm biến áp suất, cảm biến hồng ngoại…

**1.2.7.            Hệ thống điều hòa không khí, kiểm soát môi trường**

-          Thông thường thì một ngôi nhà cần có những không gian xanh, nó không chỉ giúp các thành viên trong gia đình thư giãn mà còn giúp điều hòa không khí. Việc xây dựng và duy trì màu xanh trong ngôi nhà là cần thiết, do đó hệ thống điều hòa không khí và kiểm soát môi trường sẽ giúp ích rất nhiều trong việc chăm sóc cây như độ ẩm cần thiết, hay là ánh sáng phù hợp…

**1.2.8.            Hệ thống các công tắc điều khiển trạng thái**

-          Hệ thống các công tắc và bảng hiển thị sẽ cung cấp thông tin cũng như nhận lệnh điều khiển từ gia chủ. Đảm bảo sự tương tác hai chiều giữa các thành viên và hệ thống tự động. Hệ thống bao gồm: các điều khiển từ xa, các công tắc gắn tường, các bảng điều khiển tương tác HMI, điện thoại thông minh…

**1.2.9.            Hệ thống mạng, xử lý trung tâm và sự kết hợp hoạt động**

-          Ngôi nhà thông minh được đánh giá cao và khác hẳn những ngôi nhà bình thường là do nó được trang bị một hệ thống mạng điều khiển và toàn bộ các thay đổi và điều khiển tự động trong ngôi nhà được xử lý đồng nhất thông qua hệ thống mạng và xử lý trung tâm. Nó có vai trò quan trọng, làm nhiệm vụ liên kết các hệ thống khác trong ngôi nhà lại với nhau, điều phối của hệ thống chấp hành một cách nhịp nhàng theo các điều kiện tác động được lập trình từ trước. Chúng ta gọi đó là các hoạt cảnh – hay là các điều kiện môi trường trong ngôi nhà. Một vài sự kết hợp tiêu biểu:

* Hệ thống chiếu sáng với Hệ thống xử lý trung tâm có thể được lập trình theo thói quen của người sử dụng. Các thiết bị chiếu sáng sẽ hoạt động theo chu trình thời gian đặt trước.
* Hệ thống chiếu sáng kết hợp với Hệ thống cảm biến cung cấp khả năng tự động điều khiển ánh sáng như: đèn tự động tắt khi không có người trong phòng, một số khu vực tự sáng đèn khi qua 18h…
* Hệ thống cảnh báo kết hợp với hệ thống chiếu sáng: khi có vấn đề xảy ra như cháy nổ, phát hiện ăn trộm…các bóng đèn sẽ chớp sáng liên tục, đồng thời sẽ có tiếng còi báo hiệu.
* Hệ thống cảm biến kết hợp với hệ thống xử lý trung tâm báo cáo tình trạng lưu trữ điện trong các UPS, báo cáo mực nước trong bồn chứa…nhằm đảm bảo nguồn cung cho nhu cầu sinh hoạt hằng ngày.
* Hệ thống giải trí đa phương tiện kết hợp với hệ thống chiếu sáng nhằm đem lại những giây phút thư giãn cho thành viên trong gia đình.

**1.3.      TRIỂN KHAI MÔ HÌNH NGÔI NHÀ THÔNG MINH THỰC TẾ**

**1.3.1.            Mô tả yêu cầu**

-          Đảm bảo đầy đủ các yếu tố cơ bản nhất mô phỏng một ngôi nhà thông minh thu nhỏ.

-          Có tính khả thi và thực hiện được trong thời gian ngắn.

-          Đảm bảo phát triển theo mục tiêu của đề tài đặt ra: điều khiển và quản lý các thiết bị thông qua mạng internet.

**1.3.2.            Mục tiêu thực hiện**

-          Thiết lập mô hình mạng điều khiển từ internet đến board điều khiển trung tâm, rồi từ đó đi đến các nút mạng khác.

-          Giao tiếp dữ liệu bằng sóng radio thành công giữa board điều khiển trung tâm và các nút mạng.

-          Xây dựng chương trình điều khiển là một web server được tích hợp trên board, cho phép điều khiển thông qua trình duyệt web. Có thể dùng các thiết bị như laptop, máy tính bàn, điện thoại thông minh để truy cập.

-          Thiết kế và thi công một số cảm biến như: cảm biến ánh sáng, cảm biến nhiệt độ, độ ẩm…và các mạch công suất để điều khiển các thiết bị như đèn chiếu sáng, quạt…

**CHƯƠNG 2. LÝ THUYẾT**

**2.1.      MỘT SỐ KHÁI NIỆM CƠ BẢN**

-          Mạng máy tính là tập hợp các máy tính được nối với nhau bởi đường truyền theo cấu trúc nào đó và thông qua đó các máy tính trao đổi thông tin qua lại cho nhau.

-          Đường truyền là hệ thống các thiết bị truyền dẫn có dây hay không dây dùng để chuyển các tín hiệu điện tử từ máy tính này đến máy tính khác. Các tín hiệu điện tử đó biểu thị các giá trị dữ liệu dưới dạng các xung nhị phân (on - off). Tất cả các tín hiệu được truyền giữa các máy tính đều thuộc một dạng sóng điện từ. Tùy theo tần số của sóng điện từ có thể dùng các đường truyền vật lý khác nhau để truyền các tín hiệu. Ở đây đường truyền được kết nối có thể là dây cáp đồng trục, cáp xoắn, cáp quang, dây điện thoại, sóng vô tuyến… Các đường truyền dữ liệu tạo nên cấu trúc của mạng. Hai khái niệm đường truyền và cấu trúc là những đặc trưng cơ bản của mạng máy tính.

-          Cần phải phân biệt sự trao đổi qua lại giữa máy tính này với máy tính khác trong mạng máy tính với các hệ thống thu phát một chiều như truyền hình, phát thông tin từ vệ tinh xuống các trạm thu thụ động… Vì tại đây chỉ có thông tin một chiều từ nơi phát đến nơi thu mà không quan tâm đến có bao nhiêu nơi thu, có thu tốt hay không.

-          Đặc trưng cơ bản của đường truyền vật lý là giải thông. Giải thông của một đường chuyền chính là độ đo phạm vi tần số mà nó có thể đáp ứng được. Tốc độ truyền dữ liệu trên đường truyền còn được gọi là thông lượng của đường truyền - thường được tính bằng số lượng bit được truyền đi trong một giây (Bps). Thông lượng còn được đo bằng đơn vị khác là Baud (lấy từ tên nhà bác học - Emile Baudot). Baud biểu thị số lượng thay đổi tín hiệu trong một giây.

-          Ở đây Baud và Bps không phải bao giờ cũng đồng nhất. Ví dụ: nếu trên đường dây có 8 mức tín hiệu khác nhau thì mỗi mức tín hiệu tương ứng với 3 bit hay là 1 Baud tương ứng với 3 bit. Chỉ khi có 2 mức tín hiệu trong đó mỗi mức tín hiệu tương ứng với 1 bit thì 1 Baud mới tương ứng với 1 bit.

-          Hai máy tính không thể hiểu nhau nếu không có cách thức truyền/nhận dữ liệu giống nhau, ngoài ra còn phải đảm bảo dữ liệu an toàn, chính xác, nhanh chóng. Từ thực tế đó mô hình phân tầng ra đời, dựa trên ý tưởng phân chia công việc thành từng tầng, mỗi tầng thực hiện chức năng riêng như xử lý đóng gói, truyền, nhận dữ liệu v.v… Tuy mô hình phân tầng giải quyết vấn đề thông tin giữa hai máy tính nhưng lại nảy sinh vấn đề khác, đó là có quá nhiều chuẩn giao thức ra đời dẫn đến tình trạng mạng máy tính này không thể giao tiếp mạng máy tính khác và nhiều tác hại kèm theo. Và mô hình OSI ra đời.

-          Mô hình OSI là cơ sở dành cho việc chuẩn hóa các hệ thống truyền thông, nó được nghiên cứu và xây dựng bởi ISO. Việc nghiên cứu về mô hình OSI được bắt đầu tại ISO vào năm 1971 với mục tiêu nhắm tới việc kết nối các sản phẩm của các hãng sản xuất khác nhau và phối hợp các hoạt động chuẩn hóa trong các lĩnh vực viễn thông và hệ thống thông tin. Theo mô hình OSI chương trình truyền thông được chia ra thành 7 tầng với những chức năng phân biệt cho từng tầng:

1. Tầng ứng dụng (Application layer).

2. Tầng trình bày (Presentation layer).

3. Tầng giao dịch (Session layer).

4. Tầng vận chuyển (Transport layer).

5. Tầng mạng (Network layer).

6. Tầng liên kết dữ liệu (Data link layer).

7. Tầng vật lý (Physical layer).

Hình 2.1 7 lớp OSI

-          Mô hình TCP/IP được xây dựng và phát triển dựa trên mô hình OSI. Mô hình này về cơ bản là dạng rút gọn của mô hình OSI, nó làm đơn giản quá trình xử lý dữ liệu.

-          Hiện nay, mô hình TCP/IP được sử dụng rộng rãi. Nó là giao thức truyền/nhận trên mạng Internet.

-          So sánh sự tương đồng của 2 mô hình này

Hình 2.2 So sánh giữa hai mô hình TCP/IP và OSI

-          Mô hình TCP/IP gom tầng Application, Presentation, Session của mô hình OSI thành một tầng là Application, gom hai tầng Data Link và Physical thành tầng Network Access. Trong đó:

* Tầng Application: đây là tầng cao nhất trong cấu trúc phân lớp của TCP/IP. Tầng này bao gồm tất cả các chương trình ứng dụng sử dụng các dịch vụ sẵn có thông qua một chồng giao thức TCP/IP. Các chương trình ứng dụng tương tác với một trong các giao thức của tầng giao vận để truyền hoặc nhận dữ liệu. Mỗi chương trình ứng dụng lựa chọn một kiểu giao thức thích hợp cho công việc của nó. Chương trình ứng dụng chuyển dữ liệu theo mẫu mà tầng giao vận yêu cầu.

Một số giao thức thông dụng như:

            \_ DHCP (Dynamic Host Configuration Protocol)

            \_ DNS (Domain Name System)

            \_ SNMP (Simple Network Management Protocol)

            \_ FTP (File Transfer Protocol)

            \_ TFTP (Trivial File Transfer Protocol)

            \_ SMTP (Simple Mail Transfer Protocol)

            \_ TELNET

* Tầng Transport: nhiệm vụ trước tiên của tầng giao vận là cung cấp sự giao tiếp thông tin giữa các chương trình ứng dụng. Mỗi sự giao tiếp được gọi là end-to-end. Tầng giao vận cũng có thể điều chỉnh lưu lượng luồng thông tin. Nó cũng cung cấp một sự vận chuyển tin cậy, đảm bảo rằng dữ liệu đến mà không bị lỗi. Để làm như vậy, phần mềm giao thức hỗ trợ để bên nhận có thể gửi lại các thông báo xác nhận về việc thu dữ liệu và bên gửi có thể truyền lại các gói tin bị mất hoặc bị lỗi. Phần mềm giao thức chia dòng dữ liệu ra thành những đơn vị dữ liệu nhỏ hơn (thường được gọi là các Packet) và chuyển mỗi Packet cùng với địa chỉ đích tới tầng tiếp theo để tiếp tục quá trình truyền dẫn.

Có hai giao thức chính trong tầng này gồm:

\_ UDP (User Datagram Protocol): còn gọi là Giao Thức Gói Người Dùng. UDP cung cấp các kênh truyền thông phi kết nối nên nó không đảm bảo truyền dữ liệu 1 cách tin cậy. Các ứng dụng dùng UDP thường chỉ truyền những gói có kích thước nhỏ, độ tin cậy dữ liệu phụ thuộc vào từng ứng dụng.

\_ TCP (Transmission Control Protocol): ngược lại với UDP, TCP cung cấp các kênh truyền thông hướng kết nối và đảm bảo truyền dữ liệu 1 cách tin cậy. TCP thường truyền các gói tin có kích thước lớn và yêu cầu phía nhận xác nhận về các gói tin đã nhận.

* Tầng Internet: tầng mạng xử lý giao tiếp thông tin từ một máy này tới một máy khác. Nó chấp nhận một yêu cầu để gửi một gói từ tầng giao vận cùng với một định danh của máy đích mà gói tin sẽ được gửi tới. Ví dụ với giao thức TCP hay UDP của tầng giao vận, nó sẽ bọc gói tin trong một IP Datagram, điền đầy vào trong phần Header, sử dụng giải thuật chọn đường để quyết định là giao phát gói tin trực tiếp hay là gửi nó tới một Router, và chuyển Datagram tới giao diện phối ghép mạng thích hợp cho việc truyền dẫn. Tầng mạng cũng xử lý các Datagram đến, kiểm tra tính hợp lệ của chúng, và sử dụng giải thuật chọn đường để quyết định là Datagram sẽ được xử lý cục bộ hay là sẽ được chuyển đi tiếp. Đối với các Datagram có địa chỉ đích cục bộ, thì phần mềm tầng mạng sẽ xoá phần Header của các Datagram đó, và chọn trong số các giao thức tầng giao vận một giao thức thích hợp để xử lý Packet.

Bốn giao thức quan trọng nhất trong tầng này gồm:

\_ IP (Internet Protocol): có chức năng gán địa chỉ cho dữ liệu trước khi truyền và định tuyến chúng tới đích.

\_ ARP (Address Resolution Protocol): có chức năng biên dịch địa chỉ IP của máy đích thành địa chỉ MAC.

\_ ICMP (Internet Control Message Protocol): có chức năng thông báo lỗi trong trường hợp truyền dữ liệu bị hỏng.

\_ IGMP (Internet Group Management Protocol): có chức năng điều khiển truyền đa hướng (Multicast).

* Tầng Network Access: là tầng thấp nhất của bộ giao thức TCP/IP, chịu trách nhiệm về việc chấp nhận các Datagram của tầng trên (ví dụ IP datagram) và việc truyền phát chúng trên một mạng xác định. Theo quan điểm hiện nay mô hình TCP/IP không còn bao gồm các đặc tả vật lý, nói cách khác tầng Network Access cũng không còn bao gồm vấn đề về phần cứng hay việc truyền tín hiệu vật lý nữa.

Một số giao thức tiêu biểu trong tầng này gồm:

            \_ ATM (Asynchronous Transfer Mode)

            \_ Ethernet

            \_ Token Ring

**\_** FDDI (Fiber Distributed Data Interface)

**\_** Frame Relay

Hình 2.3 Mô hình TCP/IP

-          Lớp IP trong tầng Internet không thể không nhắc tới. Để một máy tính có thể phân biệt với các máy tính khác trong mạng thì mỗi máy tính phải có tên định danh, và đó là địa chỉ IP. Mỗi giao diện trong một máy có hỗ trợ giao thức IP đều phải được gán một địa chỉ IP (máy tính có thể gắn với nhiều mạng, do vậy có thể có nhiều địa chỉ IP). Địa chỉ IP gồm 2 phần: địa chỉ mạng (netID) và địa chỉ máy (hostID). Mỗi địa chỉ IP gồm 32 bit và được tách thành 4 vùng (mỗi vùng 1 byte), có thể biểu thị dưới dạng thập phân, bát phân, thập lục phân hay nhị phân. Cách viết phổ biến nhất là dùng ký tự thập phân có dấu chấm (dotted decimal notation) để tách các vùng.

-          Do tổ chức và độ lớn của các mạng con (Subnet) của liên mạng có thể khác nhau, người ta chia các địa chỉ IP thành 5 lớp, ký hiệu là A, B, C, D và E. Trong lớp A, B, C chứa địa chỉ có thể gán được. Lớp D dành riêng cho lớp kỹ thuật Multicasting. Lớp E được dành cho những ứng dụng trong tương lai.

-          Ngoài địa chỉ IP đã trình bày ở trên còn có Subnet Mask: khi truyền thông tin, một máy cần phải biết địa chỉ IP của máy nhận có trong cùng mạng với mình không, để thực hiện điều này, ngoài địa chỉ IP, một thông số khác gọi là Subnet Mask cần được xác định cho máy. Subnet Mask cũng gồm 4 số thập phân không dấu, mỗi số gồm 8 bit; giá trị của Subnet Mask gồm 32 bit được chia làm 2 phần: bên trái gồm những bit 1, bên phải gồm những bit 0, các bit 0 xác định những địa chỉ IP nào cùng nằm trên cùng một mạng con với nó. Thông thường, các client được cung cấp địa chỉ IP đi kèm với Subnet Mask khi kết nối vào một ISP.

Một số Subnet Mask thông dụng:

\_ Lớp A có Subnet Mask là 255.0.0.0

\_ Lớp B có Subnet Mask là 255.255.0.0

\_ Lớp C có Subnet Mask là 255.255.255.0

-          Một khái niệm khác chúng ta cần phải biết đó là Default gateway: trong trường hợp này máy tính đang cố gắng tiếp cận với máy tính khác (Website) và không thể gửi gói dữ liệu ra ngoài theo mạng cục bộ, vì Web Server không phải là một phần của mạng cục bộ. Thay vào đó, máy tính cần gửi gói dữ liệu sẽ xem xét đến địa chỉ Default gateway.

Cổng vào mặc định (Default gateway) là một phần của cấu hình TCP/IP trong một máy tính. Đó là cách cơ bản để nói với máy tính rằng nếu không biết chỗ gửi gói dữ liệu ở đâu thì hãy gửi nó tới địa chỉ cổng vào mặc định đã được chỉ định. Địa chỉ của cổng vào mặc định là địa chỉ IP của một Router. Để truy cập vào một trang Web, trình duyệt Web của bạn phải biết địa chỉ IP của trang Web đó. Sau đó trình duyệt cung cấp địa chỉ cho Router. Router sẽ xác định đường đi tới mạng khác và yêu cầu các gói dữ liệu tới máy đích phù hợp. Khi bạn mở một trình duyệt Web và nhập tên Website, trình duyệt sẽ đến thẳng Website mà không cần phải thông qua việc nhập địa chỉ IP. Đó là nhờ một quá trình “dịch” tên miền thành địa chỉ IP, là công việc của một Server DNS (trình chủ hệ thống tên miền).  
Tùy chọn Preferred DNS Server được định nghĩa như là một phần của cấu hình TCP/IP. Có nghĩa là máy tính sẽ luôn biết địa chỉ IP của DNS Server. Điều gì xảy ra khi bạn cố gắng ghé thăm một trang Web, Quá trình bắt đầu với việc bạn mở trình duyệt Web và nhập đường dẫn URL. Khi đó trình duyệt biết rằng nó không thể xác định được vị trí của Website nếu chỉ dựa vào một mình địa chỉ URL. Do đó nó truy vấn thông tin địa chỉ IP của DNS Server từ cấu hình TCP/IP của máy tính và đưa đường dẫn URL lên trình chủ DNS Server. DNS Server sau đó sẽ tra tìm đường dẫn URL trên bảng có danh sách địa chỉ IP của Website. Sau đó nó trả địa chỉ IP cho trình duyệt Web và trình duyệt có thể liên lạc với Website được yêu cầu.

**2.2.      MỘT SỐ DỊCH VỤ MẠNG PHỔ BIẾN**

**2.2.1.            Dịch vụ DNS (Domain Name Service)**

-          Mỗi máy tính trong mạng muốn liên lạc, trao đổi dữ liệu hay thông tin cần phải biết rõ địa chỉ IP của nhau. Nếu số lượng máy tính nhiều thì việc nhớ những địa chỉ IP này rất là khó khăn.

-          Mỗi máy tính ngoài địa chỉ IP còn có tên máy còn gọi là Computer Name. Đối với con người việc nhớ tên máy dù sao cũng dễ hơn vì chúng có tính trực quan và gợi nhớ hơn so với địa chỉ IP. Ban đầu người ta chỉ lưu tên máy dưới dạng tập in TXT nhằm ánh xạ tên máy và địa chỉ IP nhưng khi nhu cầu về máy nhiều nên tập tin như vậy không đáp ứng được. Do đó dịch vụ DNS ra đời nhằm khắc phục nhược điểm này.

-          Dịch vụ DNS thực sự là cơ sở dữ liệu phân tán, có nhiệm vụ phân giải địa chỉ IP thành tên máy và ngược lại. DNS hoạt động theo mô hình Server/Client. Phần Server gọi là máy phục vụ hay còn gọi là NameServer. Còn phần Client là chương trình yêu cầu phân giải gọi là Resolver.

-          Ví dụ: 172.0.0.1 là địa chỉ IP tương ứng của www.vieclam.com.vn

**2.2.2.      Dịch vụ FTP (File Transfer Protocol)**

-          Là giao thức hoạt động trên chuẩn TCP/IP, FTP cung cấp cơ chế truyền file qua lại giữa các máy tính. FTP là dịch vụ đặc biệt vì nó dùng đến hai cổng. Cổng 20 dùng để truyền dữ liệu và cổng 21 dùng để truyền lệnh.

-          FTP hoạt động ở 2 chế độ: Active FTP và Passive FTP. Điểm khác biệt cơ bản giữa hai chế độ này là cơ chế Active thì Server phải tạo kết nối tới Client, còn Passive giải quyết được vấn đề này nhưng gây ra vấn đề ở phía Server do cho phép kết nối ở xa với bất kỳ cổng nào lớn hơn 1024, điều này khá nguy hiểm vì dễ bị tấn công.

**2.2.3.      Dịch vụ WEB**

-          Web Server hoạt động trên cơ sơ giao thức HTTP. HTTP là giao thức cho phép trình duyệt Web và Server có thể giao tiếp với nhau. Nó giao thức đơn giản, thông tin điều khiển được truyền đi dưới dạng văn bản thô thông qua kết nối TCP, trên port 80.

Hình 2.4 Mô hình Web Server và các máy client

-          Giao thức này thực thi đơn giản 2 thao tác Request và Response. Các yêu cầu và đáp ứng thực chất là tải file (thường là HTML), còn được gọi là các trang Web. Các file này có vai trò quan trọng, nó chứa đựng thông tin về trang Web mà chúng ta truy cập. Ngoài ra, nó còn cung cấp các dịch vụ học tập, giải trí như học tiếng anh, nghe nhạc, chơi game, đọc truyện… Cấu trúc file này gồm các thẻ (còn gọi là Tab), khi có yêu cầu thì trình duyệt sẽ tải và hiển thị cho người dùng.

-          Ví dụ về file HTML

Trang chủ

……….

-          Web Server ban đầu chỉ đơn thuần là truyền các file HTML thông thường chứa thông tin và hình ảnh v.v… Tất cả các trang HTML này gọi chung là Web tĩnh, và không có khả năng thêm bớt nội dung một cách tự động. Do nhu cầu này cần hệ thống xử lý thông tin linh hoạt hơn, mô hình Web động ra đời cho phép chúng ta nhập dữ liệu lưu trữ lại trên máy chủ hay tìm thông tin hàng hóa trực tiếp hay gián tiếp. Cách cổ điển là dùng các trang CGI (Common Gateway Interface) – nó định nghĩa cách thức Web Server chạy chương trình cục bộ ở phía Server, sau đó nhận kết quả trả về cho trình duyệt Web.

-          Khi nhận được lệnh GET, POST của giao thức HTTP từ Client, Server có thể cho thực thi chương trình và tự động tạo ra nội dung dữ liệu, sau đó trả về file (thường là thẻ HTML) cho Client. Như vậy lúc này nội dung trả về cho Client sẽ không phải là file tĩnh tồn tại trong ổ cứng của Server. Nó được tạo ra từ chương trình mà chương trình này thường được gọi là CGI. Chương trình CGI của Server có thể sản sinh ra hàng ngàn trang Web theo yêu cầu của Client. Thật ra chương trình CGI đơn thuần chỉ là chương trình \*.exe (trên Windows) hoặc chương trình thực thi (trên Unix) bình thường. Chúng ta có thể xây dựng CGI bằng các ngôn ngữ như C, Pascal, Visual Basic…

-          Cùng với việc sản sinh các trang Web chương trình CGI còn tính toán xử lý, truy xuất số liệu từ cơ sở dữ liệu, cập nhật vào trang Web gởi về Web Server. Cuối cùng Web Server gởi trả về trình duyệt.

-          Cơ chế hoạt động của CGI rất hiệu quả đối với các trình duyệt hiện nay. Ngày nay CGI vẫn còn được dùng rộng rãi trên các Web Server chạy trên các hệ Unix. Nhược điểm duy nhất là tiêu tốn tài nguyên và phần giới hạn về tốc độ thực thi. Mỗi khi có request thì CGI tại Server thực hiện:

+ Nạp chương trình vào bộ nhớ.

+ Thực thi chương trình.

+ Trả kết quả về Client.

+ Giải phóng CGI khỏi bộ nhớ.

-         Để khắc phục những nhược điểm trên nhiều phương pháp khác nhau ra đời, trong những phương pháp đó là dùng một trình CGI duy nhất để thông dịch và thực thi các lệnh trong file HTML. Điển hình là sử dụng ngôn ngữ kịch bản Perl thông dịch CGI. Perl có cú pháp tựa C và được dùng khá rộng rãi trên hệ Unix.

-          Ví dụ khi trình duyệt gửi yêu cầu http://127.0.0.1/hello.pl, Web Server được cấu hình để gọi chương trình CGI Perl.exe. Chương trình Perl.exe sẽ đọc tập tin kịch bản hello.pl. Thực thi các lệnh chứa trong kịch bản và trả kết quả về cho Client. Dạng khác tương tự là trình thông dịch các trang PHP cũng được sử dụng rộng rãi trong hệ Unix.`

-          Để cải tiến thì sử dụng CGI ở dạng DLL cung cấp trong Window. Ứng dụng kiểu mới này gọi là ISAPI. Chương trình ISAPI được nạp vào bộ nhớ lần đầu tiên và các lần sau không cần phải nạp lại nên cải thiện được tốc độ. Đồng thời cũng tiêu tốn ít tài nguyên hơn ứng dụng CGI thông thường. Java cũng đưa ra khái niệm tương tự ISAPI đó là Servlet. Servlet là chương trình Java chỉ cần nạp một lần vào máy ảo. Servlet sẽ phục vụ mọi yêu cầu từ Client gởi đến tương tự như CGI hay ISAPI.

-          Đối với ngôn ngữ kịch bản thì MicroSoft đề xuất phương án thiết kế các trang ASP. ASP là các trang chứa mã lệnh viết bằng ngôn ngữ Visual Basic kết hợp các thẻ HTML. Nó sẽ được thông dịch từ các Client. Sau đó MicroSoft lại cải tiến bằng cách đưa ra ngôn ngữ ASP.NET. Về cơ bản nó cũng giống như ASP nhưng có nhiều cải tiến đáng kể, ví dụ các trang ASP là trang phải thông dịch thì các trang ASP.NET đã thực sự được thông dịch, điều này cải thiện tốc độ truy cập Web…

-          Tóm lại, dịch vụ Web ngày càng đáp ứng nhu cầu thực tiễn trong cuộc sống. Hiện nay hầu như nó có mặt khắp mọi nơi và đóng vai trò cực kỳ quan trọng trong việc truyền thông trên thế giới.

-          Đồ án này sử dụng board Arduino như 1 Server đơn giản, khi Client là các trình duyệt Web kết nối tới và gửi Request thì Server sẽ dùng các lệnh như ethernet.print(), lệnh này dùng để gửi cho các trình duyệt Web thẻ HTML. Các chương trình như Internet Explorer, Firefox hay Chrome sẽ biên dịch các thẻ và hiển thị trực quan, dễ dàng thao tác bật, tắt các đèn, đồng thời sẽ báo cho người điều khiển biết trạng thái của các đèn, nhiệt độ, độ ẩm…

**2.2.4.            Một số dịch vụ khác**

-          DHCP: dịch vụ cung cấp địa chỉ IP tự động, có thể thay thế địa chỉ IP trong khoảng thời gian cố định, được ứng dụng nhiều trong công nghiệp.

-          Mail: hiện nay rất phổ biến là dịch vụ thư điện tử nhanh chóng trên mạng trao đổi thông tin, chia sẻ dữ liệu v.v… Ví dụ như gmail, yahoo mail v.v…

-          Các dịch vụ chia sẻ tài nguyên như máy in, máy fax. Giúp tiết kiệm chi phí.

**2.3.      CÔNG NGHỆ AJAX**

**2.3.1.            Sơ lược về công nghệ AJAX**

-          Trở lại các vấn đề về dịch vụ Web đã đề cập ở mục trước, việc xử lý động các dữ liệu tức thời luôn là mong muốn của các nhà phát triển. Mặc dù các trang JSP và ASP đã làm được nhiều việc mà người ta chờ đợi như: cải thiện tốc độ hơn so với các trang CGI, làm các trang Web trông sinh động, đẹp mắt với các hiệu ứng, xử lý dữ liệu hiệu quả, truy cập cơ sở dữ liệu dễ dàng v.v… Tuy nhiên vấn đề thường gặp phải của các trang Web loại này là khi cần truy cập hay xử lý thì trình duyệt phải tải lại (reload) toàn bộ trang Web. Đôi khi điều này gây khó chịu đối với nhiều người dùng nếu như trang Web đó có nhiều hình ảnh, và nó thường dẫn đến thời gian tải lại lâu.

-          Năm 2005, công nghệ mới ra đời dựa trên sự kết hợp các công nghệ cũ, đó là Ajax. Kỹ thuật hay công nghệ Ajax (Asynchronous JavaScript And Xml) là sự kết hợp giữa JavaScript và XML giúp trình duyệt có thể gửi yêu cầu đến Server và nhận Response (đáp ứng) mà không cần tải lại toàn bộ trang Web. Dữ liệu truyền về chỉ là phần của trang Web mà chúng ta quan tâm. Ngoài ra, Request này còn có thể được thực hiện bất đồng bộ, nghĩa là trong thời gian truyền dữ liệu ta vẫn có khả năng tương tác với trang Web. Hiện tại Ajax đã làm được rất nhiều điều, dựa vào đó có thể xây dựng các ứng dụng Web giống với các ứng dụng thông thường trên Desktop – mà hiện nay đã xuất hiện hệ điều hành Web. Ajax gắn liền với khái niệm Web mới ra đời – Web 2.0, mở ra thời kì mới cho lập trình ứng dụng cũng như ứng Web.

Hình 2.5 Truyền dữ liệu giữa Server và trình duyệt Web thông thường

Hình 2.6 Truyền dữ liệu giữa Web Server và trình duyệt có sử dụng công nghệ Ajax

-          Nhìn vào hình 2.5 và 2.6 ở trên thì ta có thể so sánh tốc độ truyền tải dữ liệu trình duyệt khi có công nghệ Ajax hiệu quả hơn rất nhiều so với thông thường. Trong cùng thời gian làm việc nhưng số lần tương tác với Web Server theo cách thông thường ít hơn nhưng dữ liệu lại có kích thước lớn hơn so với sử dụng công nghê Ajax.

Hình 2.7 Minh họa hoạt động của Ajax

-          Để dễ hiểu, chúng ta có thể hình dung Ajax tạo ra giao diện xử lý trung gian và đứng trước trình duyệt Web như minh họa (Hình 2.7). Bộ máy Ajax sẽ đại diện cho trình duyệt Web thực hiện các thao tác gửi yêu cầu tới Web Server. Đồng thời có cơ chế giao tiếp với trình duyệt thực hiện các công việc nắm bắt sự kiện (Capture) do trình duyệt gởi tới và thực hiện việc cập nhật dữ liệu (Update) vào phần trang Web yêu cầu xử lý. Dữ liệu cập nhật được Server gửi tới Ajax.

-          Có thể mô tả lại từng bước quá trình gửi và nhận từ trình duyệt tới Server sử dụng công nghệ Ajax như sau:

* Bước 1: từ trang Web trên trình duyệt ta thực hiện yêu cầu tới Server. Yêu cầu này có thể là cập nhật dữ liệu lấy từ cơ sở dữ liệu hay là nhập/xóa dữ liệu mới/cũ. Cũng có thể yêu cầu này thực hiện việc xác định thông tin từ Server khác v.v…
* Bước 2: Ajax với vai trò trung gian nằm trên trình duyệt. Nó sẽ nắm bắt yêu cầu và chuẩn bị tiến hành gửi yêu cầu này tới Server.
* Bước 3: sau khi nhận yêu cầu nó tạo ra đối tượng Request cần gửi. Request này nằm trong hai dạng của giao thức HTTP, đó là lệnh GET và POST.

Ví dụ: GET /sams/testpage.php?a=1024 HTTP/1.0

* Bước 4: sau khi đã tạo ra Request, Ajax sử dụng hàm send() để gửi yêu cầu tới Server. Hàm này có đối số nếu bằng Null thì yêu cầu gửi sẽ là GET, nếu ngược lại là POST.
* Bước 5: sau khi Server nhận được yêu cầu. Nó thực hiện yêu cầu bằng cách tìm kiếm trong thư mục hiện hành file được yêu cầu (thường các file này là các Script được lập trình trước như PHP). Sau đó nó thực hiện các Script này.
* Bước 6: trong quá trình thực hiện yêu cầu, Server liên tục thông báo cho Ajax trạng thái hiện thời. Ajax đón bắt các trạng thái này để chờ đợi các gói dữ liệu trả về.
* Bước 7: sau khi hoàn tất việc thực thi các Script, Server sẽ trả về dữ liệu dạng Text hoặc XML.
* Bước 8: tại máy Client, Ajax sẽ đón bắt dữ liệu và thực hiện việc cập nhật dữ liệu lên trình duyệt.

-          Tất cả các công việc mà bộ máy Ajax thực hiện thông qua việc gửi/nhận đối tượng XMLHTTP.

**2.3.2.            Đối tượng XMLHTTP**

-          Đối tượng hay lớp XMLHTTP chứa nhiều phương thức cho phép tạo các Request tới Server cũng như cung cấp cơ chế linh hoạt nhận dữ liệu trả về từ Server. Ngoài ra, nó còn cung cấp các thuộc tính trạng thái dùng để nhận biết dữ liệu đường truyền.

-          Trước khi đi vào tìm hiểu lớp XMLHTTP ta cần tìm hiểu cách tạo ra lớp này bằng JavaScript. Đối với trình duyệt khác nhau thì cách tạo đối tượng này cũng khác.

* Đối với Internet Explorer (IE), đối tượng này được cài đặt dưới dạng ActiveXObject. Do đó để tạo ra được nó, cần phải thực hiện câu lệnh JavaScript sau:

Var oXmlHTTP=new ActiveXobject(“Microsoft.XMLHTTP”);

Trong đó Microsoft.XMLHTTP là loại đối tượng ActiveXObject cần tạo. Tuy nhiên vì có nhiều phiên bản IE khác nhau nên để đối tượng XMLHTTP tạo ra cũng có nhiều phiên bản khác nhau.

* Đối với các trình duyệt Mozilla Firefox, Safari và Opera thì câu lệnh JavaScript tạo đối tượng XMLHTTP giống nhau:

Var oXmlHTTP=new XMLHTTPRequest();

-          Các thuộc tính và phương thức của lớp XMLHTTP:

* Onreadystatechange: phát sinh khi có sự thay đổi thuộc tính readyState.
* readyState: thuộc tính này thay đổi khi mà Request được thực hiện và khi nhận được Response từ Server. Thuộc tính này có 5 giá trị.

0 = uninitialized (đối tượng mới tạo ra nhưng hàm open chưa gọi)

1 = loading (hàm open mới được tạo ra nhưng Request chưa được gọi)

2 = loaded (Request vừa mới được gởi)

3 = interactive (Client đã nhận được phần Response từ Server)

4 = completed (tất cả dữ liệu đã được Server gởi về Client và kết nối đã đóng lại)

* responseText: dữ liệu trả về từ Server dưới dạng Text, ta dùng thuộc tính này để lấy dữ liệu.
* responseXML: tương tự như thuộc tính responseText nhưng thay vì dữ liệu trả về dạng Text thì lại dưới dạng XML.
* Status: mã trạng thái HTTP trả về từ Server (nếu giá trị 200 là thành công, còn lại là thất bại và thông tin lỗi chứa trong statusText).
* statusText: chứa thông tin lỗi nhận được.
* abort(): ngưng Request hiện tại.
* getAllResponseHeaders(): trả về tất cả phần Header.
* getresponseHeader(x): trả về giá trị phần Header x như là chuỗi.
* open(‘method’, ’URL’, ’a’): chỉ định phương thức HTTP (GET hoặc POST), URL cho biết địa chỉ cần Request tới, a=‘True’ thì thực hiện cách bất đồng bộ, a=’false’ thì ngược lại.
* send(content): gởi Request đi, nếu là GET thì tham số là Null, POST thì tham số khác Null.
* setRequestHeader(‘x’,’y’): thiết lập tham số và giá trị vào phần Header gửi đi.

-          Sau khi tạo ra đối tượng XMLHTTP. Công việc tiếp theo là thực hiện lệnh Request tới Server bằng hàm open() và send().

-          Hàm open() có chức năng chuẩn bị giao tiếp với Server từ xa bằng cách tạo ra Request cần gửi. Cấu trúc hàm này như sau open(‘method’,’URL’,’a’). Method là GET hoặc POST. URL là địa chỉ file Script cần thực thi. Tham số a rất quan trọng, nếu nó bằng True thì quá trình hoạt động của Ajax là bất đồng bộ và ngược lại. Trong quá trình truyền và nhận các Request đối với cơ chế đồng bộ thì phải chờ dữ liệu truyền về ta mới có thể thực hiện các thao tác khác, còn với cơ chế bất đồng bộ nó cho phép ta thực hiện các thao tác khác trên Web browser mà không ảnh hưởng tới quá trình kết nối với Server. Thông thường nên để a=true.

-          Hàm send() như đã giới thiệu nó có đối số. Đối số này cho biết thực hiện Request theo cơ chế nào. Nếu lệnh GET thì nó là Null và ngược lại. Sau khi gọi hàm này Request chính thức được gởi tới Server.

Ví dụ: Lệnh GET objectname.send(null);

Lệnh POST objectname.send(var1=value1&var2=value2);

-          Để biết việc gởi yêu cầu hoàn thành hay không thì lúc này tại Ajax thiết lập cách thức đón bắt các trạng thái từ Server gởi về. Ajax sử dụng thuộc tính readyState để quản lý các trạng thái này. Cụ thể các trạng thái đã giới thiệu trong phần các lớp và thuộc tính của đối tượng XMLHTTP.

-          Nhưng quan trọng nhất là trạng thái thứ 4. Nó thông báo quá trình yêu cầu đáp ứng đã hoàn thành. Tuy nhiên việc yêu cầu và đáp ứng có thực hiện thành công hay không thì ta phải thông qua thuộc tính khác là Status. Nếu thuộc tính có giá trị là 200 thì thành công và ngược lại. Mã lỗi sẽ được lưu trữ trong thuộc tính statusText.

Ví dụ:

if (myRequest.readyState == 4)

            // if server HTTP response is “OK”

            if (myRequest.status == 200) {

                        … program execution statements …

            }

            else {

                        // issue an error message for any

// other HTTP response

alert(“An error has occurred:” + myRequest.statusText);

**}**

**}**

-          Hai thuộc tính quan trọng khác là responseText và responseXML. Hai đặc tính này quyết định dữ liệu nhận vào từ Server là dạng Text hoặc dạng XML. Nếu dữ liệu là dạng Text thì việc thể hiện thực hiện dễ dàng. Nếu dữ liệu là XML, ta cần tìm hiểu thêm về các thuộc tính và phương thức khác. JavaScript hỗ trợ về DOM và XML như các hàm getElementsByTagName()

Ví dụ:

if (myRequest.status == 200) {

            alert(“The server said: ” + myRequest.responseText);

}

else {

            // issue an error message for

            // any other HTTP response

            Alert(“An error has occurred: ” + myRequest.statusText);

}

-          Để hiểu rõ hơn ta có thể tham khảo thêm sách về Ajax. Cũng có thể tìm thấy trên mạng nhiều công cụ hay bộ khung Ajax xử lý sẵn. Chúng ta không cần làm lại tất cả từ đầu. Tuy nhiên tùy thuộc vào ý thích hoặc độ phức tạp của ứng dụng mà quyết định sử dụng hay không.

-          Chú ý thêm về phía Server là ta phải xây dựng các trang Script trả lời các yêu cầu từ trình duyệt. Ta có thể viết các Script bằng các ngôn ngữ khác như PHP, ASP, JSP…

**CHƯƠNG 3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG**

**3.1.      GIỚI THIỆU VỀ ARDUINO UNO – VI ĐIỀU KHIỂN ATMEGA328**

**3.1.1.            Tổng quan**

Hình 3.1 Sơ đồ chân ATmega 328

-          ATmega 328 thuộc dòng vi điều khiển AVR, là một vi điều khiển 8-bit được chế tạo bằng công nghệ CMOS tiêu thụ năng lượng thấp, dựa trên việc cải tiến cấu trúc RISC. Bằng cách tiêu tốn ít năng lượng hơn trong 1 một chu kỳ clock, ATmega 328 đạt được thông lượng xấp xỉ 1 MIPS trên 1 MHz cho phép các nhà thiết kế hệ thống tối ưu hoá điện năng tiêu thụ trong khi tốc độ xử lý cao hơn.

-          Nhân AVR tích hợp một cấu trúc phong phú với 32 thanh ghi đa dụng, được nối trực tiếp với Arithmetic Logic Unit (ALU). Với cấu trúc trên thì hiệu suất mã hoá cao hơn trong khi thông lượng nhanh hơn gấp 10 lần so với cấu trúc của vi điều khiển con-ventional CISC.

Hình 3.2 Board UNO thực tế và sơ đồ khối của vi điều khiển ATmega328

Hình 3.3 Sơ đồ khối cấu trúc AVR

Bảng 3.1 So sánh kích thước bộ nhớ, vector ngắt các dòng AVR

Hình 3.4 Sơ đồ bộ tạo xung clock

Bảng 3.2 Các chế độ clock

-          Cài đặt các bit điều khiển CKSEL3…0 để chọn bộ clock mong muốn.

1. **Bộ dao động thạch anh năng lượng thấp**

Hình 3.5 Bộ dao động thạch anh năng lượng thấp

-          Chúng ta cài đặt bit CKSEL3…1 để chọn tần số mong muốn, làm tương tự như vậy khi lựa chọn tần số mong muốn ở các bộ dao động thạch anh khác.

-          Riêng ở bộ dao động thạch anh tần số thấp và bộ dao động RC nội còn có các bit điều khiển khác là SUT1…0, cài đặt các bit này để chọn thời gian delay giữa lúc reset đến lúc hoạt động lại hoặc thời gian delay giữa lúc hạ thấp năng lượng (Power down) đến lúc tiết kiệm năng lượng (Power save).

Bảng 3.3 Tầm tần số của bộ dao động thạch anh năng lượng thấp

1. **Bộ dao động thạch anh full swing**

Bảng 3.4 Tầm tần số của bộ dao động thạch anh full swing

1. **Bộ dao động thạch anh tần số thấp**

Bảng 3.5 Bảng thời gian khởi động của bộ dao động thạch anh tần số thấp

1. **Bộ dao động RC nội**

Bảng 3.6 Tầm tần số của bộ dao động RC nội

Bảng 3.7 Thời gian khởi động của bộ RC nội

**3.1.2.            Các chức năng được dùng trong luận văn**

1. **Thanh ghi điều khiển ngắt ngoài**

-          ATmega 328 có 2 ngắt ngoài là INT0 và INT1 với 2 chân tương ứng số 2 và 3 trên board Arduino UNO.

Hình 3.6 Sơ đồ ngắt ngoài INT1

Hình 3.7 Thanh ghi điều khiển ngắt ngoài

Bảng 3.8 Chọn chế độ ngắt cho chân INT1

-          Cài đặt các bit ISC1…0 để chọn chế độ ngắt ngoài mong muốn.

-          Trong luận văn chỉ sử dụng ngắt ngoài ở chân INT1, chọn chế độ ngắt cạnh lên.

-          Khi có tín hiệu ngắt ngoài, cờ ngắt sẽ được gán bằng 1 và nhảy vào chương trình ngắt. Sau khi thực hiện chương trình ngắt xong, cờ ngắt sẽ được gán bằng 0 và chương trình quay lại dòng lệnh trước lúc ngắt.

1. **Timer**

-          Board Arduino UNO có 3 Timer là Timer0, Timer1 và Timer2. Cả 3 Timer đều có thể được dùng để xuất PWM. Riêng Timer1 16 bit, 2 Timer còn lại 8 bit.

-          Trong luận văn sử dụng Timer1 để đọc về giá trị cảm biến nhiệt độ và độ ẩm, sau đó tính nhiệt độ và độ ẩm trung bình, Timer0 để sử dụng cho các hàm delay()…

* Timer1:

-          Hoạt động với 16 bit.

-          Có 2 ngõ ra so sánh độc lập (Output Compare Unit).

-          2 thanh ghi đệm cho các ngõ ra so sánh.

-          Tự động nạp lại giá trị cho bộ đếm khi giá trị trong Timer1 bằng với giá trị đặt.

-          Giá trị đặt được lưu vào thanh ghi OC1x (với x là A hoặc B).

-          Có thể dùng để phát xung PWM. Ngoài ra, có nhiều chế độ hoạt động PWM khác nhau.

Hình 3.8 Sơ đồ khối Timer1

-          Thanh ghi điều khiển Timer1: gồm 3 thanh ghi là TCCR1A, TCCR1B và tCCR1c

-          Cài đặt các bit tương ứng để phù hợp với nhu cầu thực tế.

Hình 3.9 Thanh ghi điều khiển Timer1

Bảng 3.9 Chọn chế độ chạy cho Timer1

Bảng 3.10 Chọn prescaler cho bộ clock

-          Cài đặt các bit CS12…0 để chọn prescaler mong muốn. Luận văn này cài đặt prescaler là 1024.

-          Tiếp theo, cài đặt các bit WGM13…0 để chọn chế độ hoạt động mong muốn. Trong luận văn chọn chế độ CTC, khi giá trị trong thanh ghi TCNT1 bằng giá trị trong OCR1A/1B thì thực hiện chương trình ngắt. Tần số ngắt được tính bằng công thức:

-          Trong đó:

+ fclk\_I/O là tần số clock hệ thống (16MHz).

+ N là prescaler.

.......................................

-          Để điều khiển hệ thống đèn thì bấm chọn ON/OFF với các đèn tương ứng.

-          Để biết được giá trị nhiệt độ, độ ẩm là bao nhiêu thì bấm vào nút cập nhật. Quá trình cập nhật không tự động để người dùng chủ động hơn trong công việc và giảm tải bớt thông lượng đường truyền.

-          Khi cảm biến hồng ngoại phát hiện trộm thì sẽ báo động về cho người dùng biết và đồng thời sẽ kích hoạt hệ thống báo động trong nhà.

**CHƯƠNG 5. KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI**

**4.1.            Những kết quả đạt được**

-          Tìm hiểu về vi điều khiển ATmega328.

-          Tìm hiểu về giao tiếp Ethernet.

-          Tìm hiểu về giao tiếp bằng sóng radio.

-          Tìm hiểu về giao thức HTTP và 1 vài thẻ HTTP cơ bản.

-          Thiết kế và thi công mô hình ngôi nhà thông minh thực tế.

-          Điều khiển và vận hành ngôi nhà thông qua mạng ethernet.

**4.2.            Hạn chế**

-          Tầm hoạt động của sóng radio bị hạn chế, nếu triển khai thực tế sẽ có những vùng không phủ sóng được.

-          Giao diện điều khiển chưa thực sự mang tính chuyên nghiệp.

**4.3.      Hướng phát triển của đề tài**

-          Khắc phục những hạn chế ở trên.

-          Kết hợp camera để nhận diện người nhà, khách…

-          Có thể kết hợp với PLC để xây dựng mạng lưới tốt hơn.

-          Tiến hành public lên internet để có thể kiểm soát tình trạng ngôi nhà ngay cả khi ở công ty.

-          Cảnh báo qua điện thoại khi có sự cố như báo cháy, có trộm.

**CHƯƠNG 6. TÀI LIỆU THAM KHẢO**

[1] http://arduino.cc/

[2] http://forum.arduino.cc/

[3] http://tme.vn/

[4] http://startingelectronics.com/tutorials/arduino/ethernet-shield-web-server-tutorial/

[5] http://forum.arduino.cc/index.php?topic=158872.0

[6] http://www.w3schools.com/

[7] http://thegioiweb.vn/Story/vn/Html\_Css/thamkhaoHtml\_Css/2008/5/4542.html

[8] http://www.vietphotoshop.com/html/html\_chapter\_1.htm

[9] http://www.vietphotoshop.com/html/html\_advance.htm

[10] https://vi.wikipedia.org/wiki/Trang\_Ch%C3%ADnh

[11] https://www.google.com.vn/?gws\_rd=cr&ei=N8auUp-zFcq7kgWgj4DwBg

**KẾT LUẬN CHUNG**

Qua thời gian thực hiện luận văn, phần lớn các mục tiêu đặt ra ban đầu đều đạt được. Thiết kế và xây dựng phần cứng chạy ổn định và dựa trên mô hình này ta có thể xây dựng được nhiều ứng dụng có ý nghĩa thực tiễn cao, có khả năng phát triển và mở rộng.

Tuy nhiên, các ứng dụng đòi hỏi nhiều kiến thức về các lĩnh vực khác như phần cứng, phần mềm, mạng máy tính ở một trình độ cao hơn. Do đó, vẫn còn nhiều vấn đề tồn tại và cần có thời gian để hoàn thiện, nâng cấp.

Thông qua việc hoàn thành luận văn, bản thân em đã từng bước ứng dụng được những kiến thức đã học, tiếp thu thêm các kiến thức thực tế về phần cứng, phần mềm… Tìm hiểu các thiết bị liên quan đến ngôi nhà thông minh, từ đó có thể vận dụng và sử dụng vào một số mục đích tương tự như đo đạc nhiệt độ, độ ẩm ở nhà vườn…

Em mong muốn được đem những gì thực hiện trong luận văn này hiện thực hóa nhằm góp phần cống hiến cho xã hội, đem công nghệ tiên tiến của thế giới đến với các gia đình Việt Nam.

Cuối cùng, một lần nữa, em xin gởi lời cảm ơn sâu sắc đến gia đình, quý thầy cô, bạn bè đã quan tâm, giúp đỡ em hoàn thành khóa luận.