|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  ĐỒ ÁN  **TỐT NGHIỆP ĐẠI HỌC**  **Đề tài:**  **THIẾT KẾ HỆ THỐNG BẢO VỆ CĂN HỘ QUA MẠNG GSM VÀ MODULE RF**  Sinh viên thực hiện: TRẦN TIẾN THÀNH  Lớp ĐT8 – K55  Giảng viên hướng dẫn: ThS. ĐÀO LÊ THU THẢO  Hà Nội, 5-2015 | |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  ĐỒ ÁN  **TỐT NGHIỆP ĐẠI HỌC**  **Đề tài:**  **THIẾT KẾ HỆ THỐNG BẢO VỆ CĂN HỘ QUA MẠNG GSM VÀ MODULE RF**  Sinh viên thực hiện: TRẦN TIẾN THÀNH  Lớp ĐT8 – K55  Giảng viên hướng dẫn: ThS. ĐÀO LÊ THU THẢO  Cán bộ phản biện:  Hà Nội, 5-2006 | |

**Đánh giá quyển đồ án tốt nghiệp**

**(Dùng cho giảng viên hướng dẫn)**

Giảng viên đánh giá:......................................................

Họ và tên Sinh viên:................................................ MSSV:…………………

Tên đồ án: ................................... ................................... ...................................

…………………………………………………………………………………..

***Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:***

***Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Có sự kết hợp giữa lý thuyết và thực hành (20)** | | | | | | | |
| 1 | Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án | | 1 | 2 | 3 | 4 | 5 |
| 2 | Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế) | | 1 | 2 | 3 | 4 | 5 |
| 3 | Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề | | 1 | 2 | 3 | 4 | 5 |
| 4 | Có kết quả mô phỏng/thưc nghiệm và trình bày rõ ràng kết quả đạt được | | 1 | 2 | 3 | 4 | 5 |
| **Có khả năng phân tích và đánh giá kết quả (15)** | | | | | | | |
| 5 | Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống | | 1 | 2 | 3 | 4 | 5 |
| 6 | Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng. | | 1 | 2 | 3 | 4 | 5 |
| 7 | Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. | | 1 | 2 | 3 | 4 | 5 |
| **Kỹ năng viết (10)** | | | | | | | |
| 8 | Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định | | 1 | 2 | 3 | 4 | 5 |
| 9 | Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.) | | 1 | 2 | 3 | 4 | 5 |
| **Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)** | | | | | | | |
| 10a | | Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế | 5 | | | | |
| 10b | | Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest. | 2 | | | | |
| 10c | | Không có thành tích về nghiên cứu khoa học | 0 | | | | |
| **Điểm tổng** | | | **/50** | | | | |
| **Điểm tổng quy đổi về thang 10** | | |  | | | | |

***3. Nhận xét thêm của Thầy/Cô (****giảng viên hướng dẫn nhận xét về thái độ và tinh thần làm việc của sinh viên****)***

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

Ngày: / /201

Người nhận xét

(Ký và ghi rõ họ tên)

**Đánh giá quyển đồ án tốt nghiệp**

**(Dùng cho cán bộ phản biện)**

Giảng viên đánh giá:......................................................

Họ và tên Sinh viên:................................................ MSSV:…………………

Tên đồ án: ................................... ................................... ...................................

…………………………………………………………………………………..

***Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:***

***Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Có sự kết hợp giữa lý thuyết và thực hành (20)** | | | | | | | |
| 1 | Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án | | 1 | 2 | 3 | 4 | 5 |
| 2 | Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế) | | 1 | 2 | 3 | 4 | 5 |
| 3 | Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề | | 1 | 2 | 3 | 4 | 5 |
| 4 | Có kết quả mô phỏng/thưc nghiệm và trình bày rõ ràng kết quả đạt được | | 1 | 2 | 3 | 4 | 5 |
| **Có khả năng phân tích và đánh giá kết quả (15)** | | | | | | | |
| 5 | Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống | | 1 | 2 | 3 | 4 | 5 |
| 6 | Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng. | | 1 | 2 | 3 | 4 | 5 |
| 7 | Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. | | 1 | 2 | 3 | 4 | 5 |
| **Kỹ năng viết (10)** | | | | | | | |
| 8 | Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định | | 1 | 2 | 3 | 4 | 5 |
| 9 | Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.) | | 1 | 2 | 3 | 4 | 5 |
| **Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)** | | | | | | | |
| 10a | | Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế | 5 | | | | |
| 10b | | Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest. | 2 | | | | |
| 10c | | Không có thành tích về nghiên cứu khoa học | 0 | | | | |
| **Điểm tổng** | | | **/50** | | | | |
| **Điểm tổng quy đổi về thang 10** | | |  | | | | |

***3. Nhận xét thêm của Thầy/Cô***

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

Ngày: / /201

Người nhận xét

(Ký và ghi rõ họ tên)

# LỜI NÓI ĐẦU

Hiện nay, với những ứng dụng của khoa học kỹ thuật và công nghệ tiên tiến, thế giới của chúng ta đã và đang ngày càng thay đổi, phát triên, văn minh và hiện đại hơn rất nhiều.Và khái niệm nhà thông minh được ra đời. Với hàng loạt những giải phát, thiết bị thông minh được tích hợp và đưa vào ứng dụng ngay trong các căn hộ của chúng ta đã ra đời. Tuy nhiên giá thành của các thiết bị này là tương đối cao cũng như các bước triển khai lắp đặt để tích hợp vào căn hộ của chúng có thể gặp nhiều khó khăn, đặc biệt với những căn hộ tầm trung hay những căn hộ cũ, khó thay đổi kiến trúc để phù hợp cho việc tích hợp các thiết bị này.

Nhận thức được tầm thực tiễn của việc mang sự thông minh, tiện nghi đến bất kỳ căn hộ nào với phương thức lắp đắt đơn giản, dễ sử dụng là cơ sở để em chọn đề tài đồ án tốt nghiệp “Thiết kế hệ thống bảo vệ cho căn hộ qua sóng RF và mạng GSM”. Trong thời gian thực hiện đồ án, em đã tìm hiểu về vi điều khiển, các chuẩn truyền thông khác nhau, một vài loại cảm biến, cách điều khiển hệ thống qua sóng RF và tin nhắn SMS, đồng thời tìm hiểu về các tiêu chí của một ngôi nhà thông minh.

Để hoàn thành tốt đề tài này,trước tiên, em xin chân thành cảm ơn các thầy cô

và ban giám hiệu trường Đại học Bách Khoa Hà Nội nói chung và các thầy cô trong khoa Điện tử viễn thông nói riêng đã tạo mọi điều kiện giúp đỡ chúng em trong quá trình học tập và nghiên cứu. Đặc biệt, em xin chân thành gửi lời cảm ơn đến cô Đào Lê Thu Thảo, là giáo viên hướng dẫn và cũng là người đã nhiệt tình chỉ bảo cho em trong suốt thời gian qua.

Cuối cùng xin gửi lời cảm ơn đặc biệt đến gia đình luôn là một chỗ dựa vững

chắc cũng như luôn tạo mọi điều kiện tốt nhất cho em được học tập.

Với những điều kiện thuận lợi đó, em đã hoàn thành đồ án trong đúng thời hạn cho phép. Tuy nhiên do hạn chế về thời gian cũng như giới hạn về kiến thức và kinh nghiệm nên đề tài không thể tránh khỏi còn có những thiếu xót hay hạn chế nhất định. Vì vậy em rất mong sẽ nhận được nhiều ý kiến đánh giá, góp ý của các thầy cô giáo trong hội động để em có thể phát triển và hoàn thiện đề tài này.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Trần Tiến Thành

Mục lục

[LỜI NÓI ĐẦU 9](#_Toc420457279)

[DANH SÁCH HÌNH VẼ 13](#_Toc420457280)

[DANH SÁCH CÁC BẢNG BIỂU 14](#_Toc420457281)

[CHƯƠNG 1: MỞ ĐẦU 15](#_Toc420457282)

[1.1 Đặt vấn đề 15](#_Toc420457283)

[1.2 Mục tiêu nghiên cứu của đề tài 16](#_Toc420457284)

[1.3 Kết quả nghiên cứu của đề tài 16](#_Toc420457285)

[CHƯƠNG 2: LÝ THUYẾT TỔNG QUAN 18](#_Toc420457286)

[2.1 Tổng quan về tin nhắn SMS và Module SIM900 18](#_Toc420457287)

[2.1.1 Tổng quan về tin nhắn SMS 18](#_Toc420457288)

[2.1.2 Giới thiệu về Module SIM900 18](#_Toc420457289)

[2.2 Tổng quan về AVR 20](#_Toc420457290)

[2.2.1 Tổng quan về AVR 20](#_Toc420457291)

[2.2.2 Đặc điểm cơ bản của AVR 20](#_Toc420457292)

[2.2.3 Kiến trúc và tổ chức bộ nhớ họ vi điều khiển AVR 22](#_Toc420457293)

[2.3 Tìm hiểu về truyền thông UART 34](#_Toc420457294)

[2.3.1 Truyền thông nối tiếp không đồng bộ 34](#_Toc420457295)

[2.3.2 Truyền thông nối tiếp không đồng bộ với AVR (UART) 38](#_Toc420457296)

[2.3.3 Sử dụng UART 43](#_Toc420457297)

[2.4 Giao tiếp SPI 43](#_Toc420457298)

[2.4.1 Chuẩn truyền thông SPI 43](#_Toc420457299)

[2.4.2 Truyền thông SPI trên AVR 46](#_Toc420457300)

[CHƯƠNG 3: THIẾT KẾ VÀ THI CÔNG 51](#_Toc420457301)

[3.1. Phân tích yêu cầu thiết kế 51](#_Toc420457302)

[3.2 Sơ đồ tổng quan của hệ thống 51](#_Toc420457303)

[3.3 Sơ đồ khối của hệ thống 52](#_Toc420457304)

[3.4 Thiết kế phần cứng mạch chính 53](#_Toc420457305)

[3.4.1 Sơ đồ nguyên lý 53](#_Toc420457306)

[3.4.2 Sơ đồ layout mạch chính trung tâm 54](#_Toc420457307)

[3.5 Thiết kế và thi công Module SIM900 55](#_Toc420457308)

[3.5.1 Giới thiệu về tập lệnh AT Command 55](#_Toc420457309)

[3.5.2 Một số tập lệnh cơ bản được sử dụng 55](#_Toc420457310)

[3.6 Khối cảm biến (Em đang layout phần này) 58](#_Toc420457311)

[3.6.1 Sơ đồ nguyên lý 58](#_Toc420457312)

[3.6.2 Sơ đồ layout 58](#_Toc420457313)

[3.7 Một số hình ảnh thực tế của mạch 58](#_Toc420457314)

[CHƯƠNG 4: KẾT LUẬN 59](#_Toc420457315)

[4.1 Kết quả đạt được 59](#_Toc420457316)

[4.2 Ưu điểm và khuyết điểm của hệ thống 59](#_Toc420457317)

[4.3 Hướng phát triển trong tương lai 60](#_Toc420457318)

[Tài liệu tham khảo 61](#_Toc420457319)

[PHỤ LỤC 1: VI ĐIỀU KHIỂN ATMEGA 8 62](#_Toc420457320)

[PHỤ LỤC 2: 68](#_Toc420457321)

[GIỚI THIỆU VỀ TRÌNH BIÊN DỊCH AVRSTUDIO 68](#_Toc420457322)

[2.2 Giới thiệu AVR Studio 68](#_Toc420457323)

[2.3 Giới thiệu chương trình nạp AvrProg 68](#_Toc420457324)

[PHỤ LỤC 3: THÔNG SỐ KỸ THUẬT SIM900 69](#_Toc420457325)

[PHỤ LỤC 4: THÔNG SỐ KỸ THUẬT CÁC CẢM BIẾN ĐƯỢC SỬ DỤNG 72](#_Toc420457326)

# DANH SÁCH HÌNH VẼ

[Hình 2.1: Module SIM900 18](#_Toc420322710)

[Hình 2.2: Sơ đồ các chân module SIM900 19](#_Toc420322711)

[Hình 2.3: Tổ chức bộ nhớ 22](#_Toc420322712)

[Hình 2.4: Tập thanh ghi 24](#_Toc420322713)

[Hình 2.5: Ngắt trên AVR 27](#_Toc420322714)

[Hình 2.6: Sơ đồ cấu trúc của bộ định thời 30](#_Toc420322715)

[Hình 2.7: Giản đồ xung hoạt đông của bộ định thời 31](#_Toc420322716)

[Hình 2.8: Giá trị các thanh ghi 31](#_Toc420322717)

[Hình 2.9: Sơ đồ khối Timer1 32](#_Toc420322718)

[Hình 2.10: Tín hiệu tương đương của UART và RS232 34](#_Toc420322719)

[Hình 2.11: Truyền 8 bit theo phương pháp song song và nối tiếp 35](#_Toc420322720)

[Hình 2.12: Giao diện SPI 45](#_Toc420322721)

[Hình 2.13: Truyền dữ liệu SPI 45](#_Toc420322722)

[Hình 2.14: Chế độ hoạt động của SPI (CPHA=0) 48](#_Toc420322723)

[Hình 2.15: Các chế độ hoạt động của SPI (CHPA=1) 48](#_Toc420322724)

[Hình 3.1: Sơ đồ tổng quan của hệ thống 52](#_Toc420322725)

[Hình 3.2: Sơ đồ tổng quan của hệ thống 52](#_Toc420322726)

[Hình 3.3: Sơ đồ nguyên lý các khối 54](#_Toc420322727)

[Hình 3.4: Sơ đồ layout mạch xử lý trung tâm 54](#_Toc420322728)

# DANH SÁCH CÁC BẢNG BIỂU

[Bảng 2.1: Chọn kiểm tra parity 41](#_Toc420322315)

[Bảng 2.2: Độ dài dữ liệu truyền 42](#_Toc420322316)

[Bảng 2.3: Tính tốc độ baud 43](#_Toc420322317)

[Bảng 2.4: Tốc độ mà SPI trong AVR đạt được 49](#_Toc420322318)

# CHƯƠNG 1: MỞ ĐẦU

## 1.1 Đặt vấn đề

Ngày nay, con người cùng với những ứng dụng của khoa học kỹ thuật tiên tiến của thế giới, chúng ta đã và đang ngày một thay đổi, văn minh và hiện đại hơn. Sự phát triển của kỹ thuật điện tử đã tạo ra hàng loạt những thiết bị với các đặc điểm nổi bật như sự chính xác cao, tốc độ nhanh, gọn nhẹ…là những yếu tố rất cần thiết góp phần cho hoạt động của con người đạt hiệu quả ngày càng cao hơn. Điện tử đang trở thành một ngành khoa học đa nhiệm vụ. Điện tử đã đáp ứng được những đòi hỏi không ngừng của các ngành, lĩnh vực khác nhau cho đến nhu cầu thiết yếu của con người trong cuộc sống hàng ngày. Một trong những ứng dụng của rất quan trọng của ngành công nghệ điện tử là các kỹ thuật điều khiển thiết bị từ xa. Có rất nhiều kỹ thuật điều khiển thiết bị từ xa khác nhau, có thể kể đến đó là điều khiển từ xa thiết bị qua Wifi, các module Zigbee, module RF, sóng vô tuyến hay hồng ngoại. Sử dụng mạng di động GSM được ứng dụng rất nhiều trong công nghiệp và các lĩnh vực khác trong cuộc sống với những thiết bị điều khiển từ xa rất tinh vi và đạt được năng suất, kinh tế thật cao.

Như chúng ta cũng đã biết, hiện nay tình hình tội phạm đang ngày càng gia tăng, đặc biệt là ở các thành phố lớn. Số lượng các vụ trộm cắp tài sản gia đình tăng cao với mức độ tinh vi và nguy hiểm. Do đó việc mỗi gia đình nên lắp đặt cho mình hệ thống chống trộm hay các thiết bị chống trộm gia đình phù hợp là điều hết sức cần thiết. Bên cạnh vấn đề về an ninh do tác động từ người ngoài can thiệp vào thì việc bảo vệ căn hộ ngay từ trong cũng rất đáng được quan tâm. Mọi sự cố là điều chúng ta không mong muốn và không thể nào lường trước được nhưng lại có thể phòng chống từ xa được.

Nắm bắt được nhu cầu bảo vệ tài sản cũng như sự an toàn cho nhưng người thân. Em đa chọn đề tài “Thiết kế hệ thống bảo vệ căn hộ qua mạng GSM và sóng RF” để đắp ứng được nhu cầu ngày càng cao của con người.

Với đề tài trên hệ thống hoạt động với hai hệ thống con sau:

* Khi có sự can thiệp trái phép vào căn hộ, hệ thống chống trộm sẽ bật chuông báo động đồng thời gửi tin nhắn qua mạng điện thoại (GSM) báo cho chủ nhà ngay lập tức để nhanh chóng tìm ra cách xử lý
* Hệ thống báo cháy sẽ phát tín hiệu cảnh báo khi nhiệt độ phòng tăng cao vượt quá giới hạn quy định hoặc có sự dò dỉ khí gas. Hệ thống cũng sẽ gửi tin nhắn thông báo cho chủ nhà
* Điều khiển hệ thống bật/tắt báo động qua tin nhắn SMS từ những số điện thoại được cho phép hoặc qua điều khiển cầm tay hoặc ngay phím cảm ứng trên thiết bị.

## 1.2 Mục tiêu nghiên cứu của đề tài

Đồ án được nghiên cứu và thực hiên với mục đích áp dụng những kiến thức đã được học trong nhà trường đề thiết kế tạo ra một hệ thống bảo vệ căn hộ từ xa qua mạng GSM, và có thể ứng dụng vào sử dụng thực tế

Tìm hiểu về hệ thống điện thoại di động GSM hiện nay, nghiên cứu và sử dụng các module GSM/GPRS phổ biến hiện nay. Module được thực hiện và sử dựng là module SIM900 của hang SIMCom

Nghiên cứu và sự dụng chip vi điều khiển AVR của Atmel (Hoa Kỳ). Ở đây em đã tiên hành nghiên cứu và sử dụng là chip ATmega8

Tìm hiểu về sóng RF, thực hiện điều khiển thiết bị từ xa qua module RF

Tìm hiểu về các loại cảm biến dung để phát triển trong nhà thông minh

## 1.3 Kết quả nghiên cứu của đề tài

Với những mục tiêu và kế hoạch đề ra, cùng với sự nỗ lực nghiên cứu tìm tòi đề tài đã đạt được những kết quả nhất định sau:

Thiết kế thành công mạch điều khiển giao tiếp qua mạng GSM và giao tiếp với module RF với các tính năng sau:

Giao tiếp thành công với module SIM900, nhận và gửi tinh nhắn thành cong, thực hiệu điều khiển hệ thống qua tin nhắn SMS

Cảnh báo tới người dung và phát tín hiệu khi xảy ra sự cố, với mỗi sự cố khác nhau (chống trộm & chống cháy) sẽ cảnh báo cho người sử dụng khác nhau

Hệ thống được điều khiển trực tiếp qua các nút cảm ứng đặt trên thiết bị hoặc qua điều khiển cầm tay RF

Ngoài ra, mạch điều khiển được thiết kế sẵn sàng cho việc tích hợp them các ngoại vi trong tương lai

# CHƯƠNG 2: LÝ THUYẾT TỔNG QUAN

## 2.1 Tổng quan về tin nhắn SMS và Module SIM900

### 2.1.1 Tổng quan về tin nhắn SMS

SMS là từ viết tắc của Short Message Service. Đó là một công nghệ cho phép gửi và nhận các tin nhắn giữa các điện thoại với nhau. Dữ liệu có thể được lưu giữ bởi một tin nhắn SMS là rất giới hạn. Một tin nhắn SMS có thể chưa tối đa là 140 byte (1120 bit) dữ liệu. Vì vậy, một tin nhắn SMS chỉ có thể chứa:

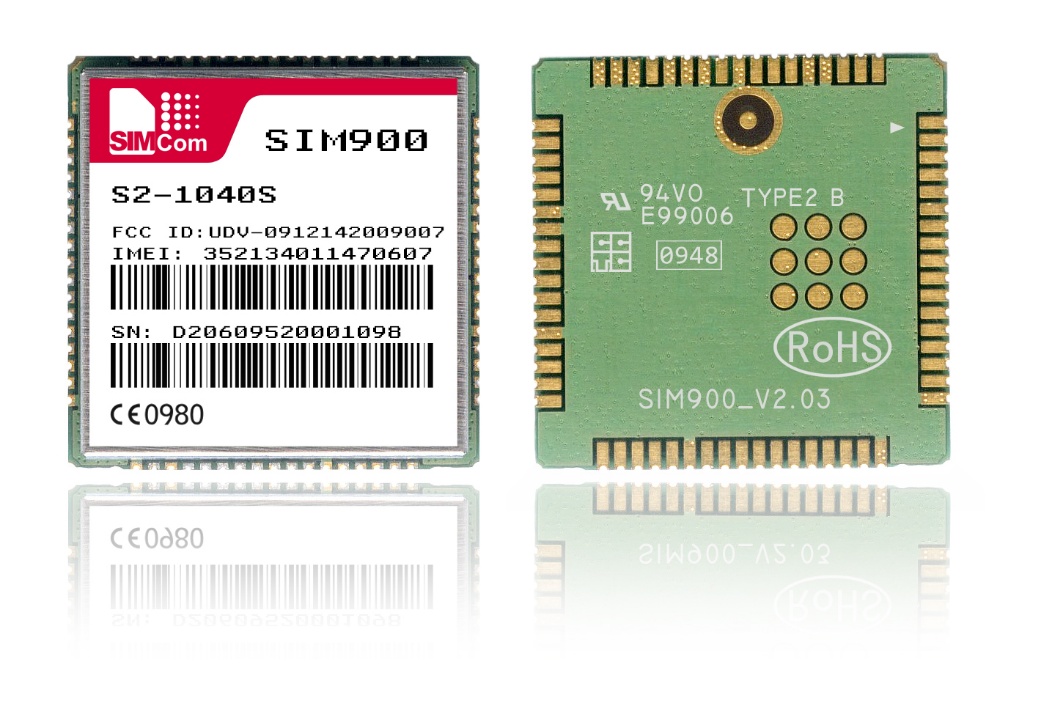
+ 160 kí tự nếu như mã hóa kí tự 7 bit được sử dụng

+ 70 kí tự nếu như mã hóa kí tự 16 bit Unicode UCS2 được sử dụng.

Tín nhắn SMS dạng text hỗ trợ nhiều ngôn ngữ khác nhau. Nó có thể hoạt động tốt với nhiều ngôn ngữ mà có hỗ trợ mã Unicode, bao gồm cả Arabic, Trung Quốc, Nhật Bản và Hàn Quốc.

### 2.1.2 Giới thiệu về Module SIM900

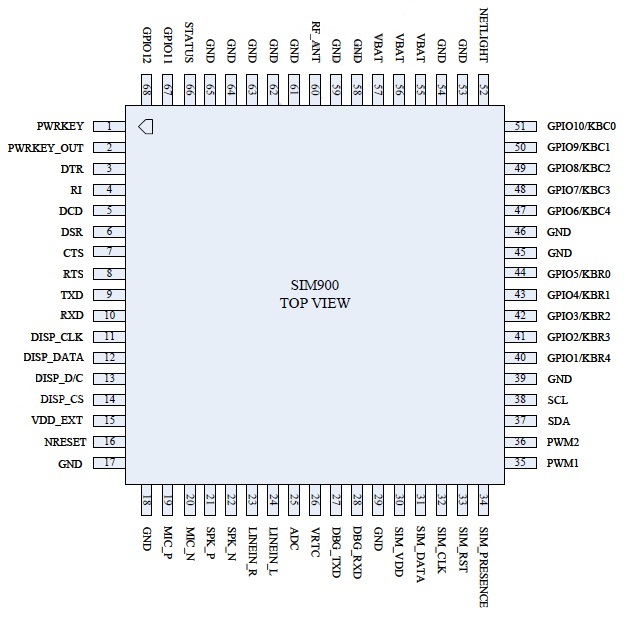
#### 2.1.2.1 Tổng quan về Module SIM900



Hình 2.: Module SIM900

Sim900 là một module GSM/GPRS cực kỳ nhỏ gọn, được thiết kế cho thị trường toàn cầu. Sim900 hoạt động được ở 4 băng tần GSM 850Hz, EGSM 900MHz, DCS 1800MHz và PCS 1900MHz như là một loạt thiết bị đầu cuối với một Chip xử lý đơn nhân đầy sức mạnh, tăng cường các tính năng quan trọng dựa trên nền vi xử lý ARM926EJ-S, mang lại nhiều lợi ích từ kích thước nhỏ gọn (24x24 mm), đáp ứng những yêu cầu về không gian trong các ứng dụng M2M

#### 2.1.2.2 Khảo sát sơ đồ chân và chức năng từng chân



Hình 2.: Sơ đồ các chân module SIM900

## 2.2 Tổng quan về AVR

### 2.2.1 Tổng quan về AVR

Vi điều khiển AVR là sản phẩm của công ty Atmel( Hoa Kỳ), đây là một bộ xử lý có khiến trúc Harvard, nghĩa là đơn vị xử lý trung tâm của bộ nhớ chương trình và một bộ nhớ dữ liệu tách biệt. Bộ vi điều khiển AVR có nhiều khả năng để giảm năng lượng tiêu thụ. Đây là một trong những ưu điểm lớn của họ vi điều khiển này.

Gần như chúng ta không cần mắc them bất kỳ một linh kiện phụ nà khi sử dụng AVR, thậm chí không cần nguồn tạo xung clock cho chip (thường là khối thạch anh). Thiết bị lập trình (mạch nạp) cho AVR rất đơn giản, có loại mạch chỉ cần vài con điện trỏ là có thể làm được. Một số chip còn hỗ trợ lập trình on-chip bằng bootloader không cần mạch nạp. Bên cạch lập trình bằng ASM, cấu trúc AVR được thiết kế tương thích với ngôn ngữ C.

Họ vi điều khiển AVR gồm nhiều bộ điều khiển với các tài nguyên khác nhau về bộ phận ngoại vi, bộ nhớ chương trình và kiểu đóng vỏ.

AVR có rất nhiều dòng khác nhau bao gồm dòng Tiny AVR (như AT tiny 12, AT tiny 22 … ) có kích thước bộ nhớ nhỏ, ít bộ phận ngoại vi, rồi đến dòng AVR (chẳng hạn AT90S8535, AT90S8515) có kích thước bộ nhớ vào loại trung bình và mạnh hơn là dòng Mega (như Atmega32, Atmega128, …) với bộ nhớ có kích thước vài Kbyte đến vài Kb cngf với các bộ ngoại vi đa dạng được tích hợp trên chip, cũng có cả dòng tích hợp cả LCD trên chip (dòng LCD AVR). Tốc đọ của dòng mega cũng cao hơn các dòng khác, sự khác nhau cơ bản giữa các dòng chính là cấu trúc ngoại vi, còn nhân thì vẫn như nhau.

### 2.2.2 Đặc điểm cơ bản của AVR

Những đặc điểm cở bản của họ vi điều khiển AVR

Bộ nhớ flash được tích hợp ngay trên chip có khả năng lập trình ngay trên hệ thống được sử dụng làm bộ nhớ chương trình. Điều này có nghĩa là ta không cần phải dung đến các bộ nhớ EPROM hoặc ROM bên ngoài để chứa mã chương trình. Hơn nữa bộ nhớ chương trình có thể nạp được chương trình trong khi bộ vi xử lý vẫn ở nguyên trên bản mạch, không cần nhấc ra ngoài để nạp.

Các thanh ghi làm việc đa năng 32-X-8. Một tập hợp bao gồm rất nhiều thanh ghi có nghĩa là các biến có thể được lưu trữ bên trong CPU chứ không phải lưu trữ cá biến trong bộ nhớ, vì việc truy cập lên bộ nhớ thường tốn nhiều thời gian hơn. Như vậy chương trình sẽ chạy nhanh hơn.

Bộ nhớ dữ liệu ngay trên chip loại EEPROM và RAM có trong hầu hết các thành viên của họ AVR. Đơn vị CPU có kiến trúc Harvard, còn các bộ nhớ EEPROM và RAM được nhìn nhận như là bộ nhớ dữ liệu và được dung để cất giữ các hằng và biến.

Hoạt động với xung nhịp giữ nhịp có tần số từ 0 đến 10MHz. Hầu hết các lệnh được thực hiện trong một chu kỳ đồng hồ làm cho tốc độ xử lý lớn hơn khoảng 10 lần so với 8051 ở cùng tần số của đồng hồ giữ nhịp. Có mạch đặt lại trạng thái mỗi khi cấp lại điện nguồn cho hệ thống.

Có bộ định thời ngay trên chip và lập trình được với mạch chia tần số tách biệt. Bộ định thời này được sử dụng cho các ứng dụng cần có sự phân định thời gian của các sự kiện.

Có các nguồn ngắt bên trong và bên ngoài. Có bộ định thời watchdog ngay trên chip và lập trình được với bộ dao động độc lập. Bộ phận này được sử dụng để khôi phục lại trạng thái hoạt động của hệ thống trong trường hợp xảy ra lỗi treo khi chạy phần mềm, ngoài ra còn được sử dụng cho một vài ứng dụng khác.

Có các chế độ hoạt động như SLEEP và POWER DOWN (nghỉ hay giảm dòng tiêu thụ khi cần thiết). Đặc điểm này cho phép tiết kiệm năng lượng khi bộ xử lý nhàn rỗi.

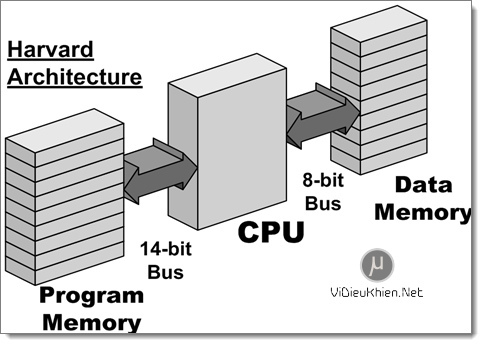
Nhiều chip có mạch dao động đồng hồ RC ngay trên chip. Khi sử dụng bộ dao động RC trên chip số lượng các linh kiện phụ trợ sẽ giảm đi.

Hệ thống chip đa dạng từ cỡ nhỏ đến lớn thuận tiện cho từng ứng dụng riêng biệt

### 2.2.3 Kiến trúc và tổ chức bộ nhớ họ vi điều khiển AVR

* **Kiến trúc họ vi điều khiển AVR**

Các bộ xử lý AVR có kiến trúc Harvard, nghĩa là có bộ nhớ dữ liệu bộ nhớ chương trình tách riêng biệt nhau: Bộ nhớ chương trình là loại bộ nhớ flash. Dung lượng của bộ nhớ thay đổi khác nhau giữa các bộ xử lý trong cùng họ. Bộ nhớ này được truy nhập theo từng chu kì đồng hồ, và một lệnh được nạp vào thanh ghi lệnh. Thanh ghi lệnh nối với tệp thanh ghi bằng cách lựa chọn xem thanh ghi nào được ALU sử dụng để thực thi lệnh. Lối ra của thanh ghi lệnh được giải mã bằng bộ giải mã lệnh để quyết định chọn tín hiệu điều khiển nào sẽ đựơc kích hoạt để hoàn thành lệnh hiện tại. Bộ nhớ chương trình, bên cạnh các lệnh lưu trữ cũng chứa các vectơ ngắt bắt đầu ở địa chỉ$0000



Hình 2.: Tổ chức bộ nhớ

* **Tổ chức bộ nhớ AVR**

Bộ nhớ dữ liệu gồm 5 thành phần khác nhau:

1. Một tệp thanh ghi (register file) với 32 thanh ghi có độ rộng 8 bit.

2. 64 thanh ghi vào/ra mỗi thanh 8 bit. Các thanh ghi này thực chất là một phần của bộ nhớ SRAM trên chip và có thể được truy nhập hoặc như bộ nhớ SRAM với các địa chỉgiữa $20 và $5F hoặc như các thanh ghi I/O với các địa chỉ giữa $00 và $3F. Số lượng 64 thanh ghi này có thể thay đổi tuỳ theo từng loại chip trong họ AVR.

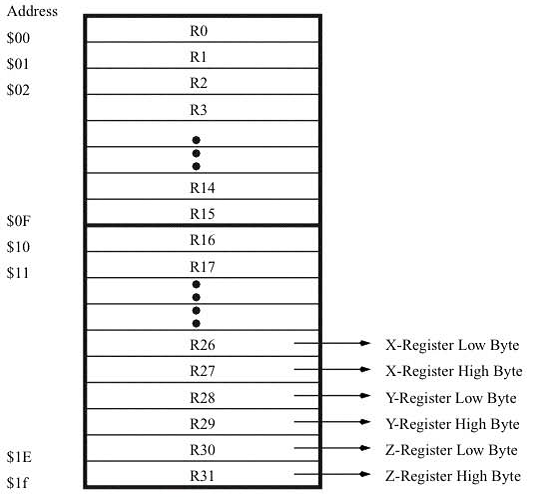
3. Bộ nhớ SRAM bên trong: bộ nhớ này được sử dụng cho ngăn xếp cũng như đểlưu trữ các biến. Trong thời gian có ngắt và gọi đoạn chương trình, giá trị hiện tại của bộ đếm chương trình được lưu trữ trong ngăn xếp. Kích thước của ngăn xếp bị giới hạn bởi bộ nhớ SRAM trên chip. Vị trí của ngăn xếp được chỉ thị bởi con trỏ ngăn xếp.

4. Bộ nhớ SRAM bên ngoài: Đặc tính này chỉ có ở các bộ xử lý cỡ lớn trong họ viđiều khiển AVR.

5. EEPROM: bộ nhớ EEPROM có sẵn trên hầu hết các bộ vi điều khiển AVR và được truy nhập theo một bản đồ bộ nhớ tách biệt. Địa chỉ bắt đầu của bộ nhớ EEPROM luôn là $0000. Các bộ xử lý khác nhau có từ 64 byte đến 4 Kbyte bộ nhớ EEPROM. Bộ nhớ EEPROM có thể ghi vào được khoảg 100000 lần.

* **Tệp các thanh ghi**

Tất cả các bộ điều khiển AVR đều có 32 thanh ghi đa năng. Một số trong cácthanh ghi này còn có các chức năng riêng, bổ sung. Các thanh ghi được đặt tên từ R0 đếnR31. Tệp thanh ghi được tách thành 2 phần, mỗi phần có 16 thanh ghi, đánh số từ R0 đếnR15 và R16 đến R31. Tất cả các lệnh thao tác trên các thanh ghi đều có thể truy nhập trựctiếp và truy nhập trong chu trình đơn đến tất cả các thanh ghi. Nhưng có một ngoại lệ làcác lệnh SBCI, SUBI, CPI, ANDI và ORI cũng như lệnh WI, các lệnh này chỉ tác động đếncác thanh ghi R16 đến R31



Hình 2.: Tập thanh ghi

Các thanh ghi R0 và R26 đến R31 có các chức năng bổ sung. Thanh ghi R0 được sử dụng trong các lệnh nạp bộ nhớ chương trình LPM (Load Program Memory) , trong khi các thanh ghi R26 đến R31 được sử dụng làm các thanh ghi con trỏ như được minh hoạ trên hình 2. Các thanh ghi con trỏ nay được sử dụng trong nhiều lệnh gián tiếp dùng cho thanh ghi (register indirect instruction)

* **Khối logic số học**

Khối số học lôgic(ALU) thực hiện các thao tác như thao tác bit; phép tính số học và lôgic trên nội dung của các thanh ghi đã được chỉ định. Các thao tác này được thực hiện trong một chu kỳ đồng hồ đơn lẻ. Mỗi một thao tác ALU đều làm ảnh hưởng đến các cờ trong thanh ghi trạng thái (STATUS), tuỳ thuộc vào lệnh

* **Bộ nhớ EEPROM**

Tất cả các bộ vi điều khiển AVR đều có một bộ nhớ EEPROM trên chip. Dunglượng bộ nhớ EEPROM thay đổi từ 64 byte trên bộ điều khiển AT90S1200, Tiny10/12đến 4 Kbyte trên Mega103. Các thanh ghi dùng để truy nhập EEPROM gồm: thanh ghi địa chỉ EEPROM (EEAR), thanh ghi dữ liệu EEPROM(EEDR), và thanh ghi điều khiển EEPROM (EECR). Bộ nhớ EEPROM là một bộ nhớ không bị mất dữ liệu khi nguồn điện cung cấp bịngắt. Dữ liệu trong nó có thể được ghi và xóa bằng điện và vì vậy việc ghi và đọc bộ nhớ này có thể làm trực tiếp. Bộ nhớ này được xem như một bộ nhớ dữ liệu nhưng chúngkhông được truy nhập như một bộ nhớ SRAM mà được truy nhập như một thiết bị ngoại vi. Thời gian truy cập để viết mất khoảng 2.5 đến 4 ms, và phụ thuộc vào nguồn điện cung cấp cho vi điều khiển (Vcc). Để điều khiển vào ra dữ liệu với EEPROM chúng ta có thể sử dụng 3 thanh ghi đó là: EEPROM address, EEDR và EECR. Để ghi dữ liệu vào EEPROM ta làm các bước sau:

- Bước 1: chờ đợi bit EEWE đã bị xóa chưa ?

- Bước 2: Ghi dữ liệu mới vào thanh ghi dữ liệu (EEDR)

- Bước 3: Set bit EEWE rồi đến bit EEMWE để bắt đầu ghi dữ liệu.

Chú ý:

Nếu như dang ghi dữ liệu ở EEPROM mà xuất hiện ngắt thì dữ liệu đó sẽ không được ghi một cách an toàn vào EEPOM. Để đọc dữ liệu vào EEPROM thì đơn giản hơn.

- Bước 1: kiểm tra bit EEWE nếu như có quá trình ghi EEPROM thì chờ đợi.

- Bước 2: Đưa địa chỉ cần đọc vào thanh ghi địa chỉ EEAR.

- Bước 3: Set bit EERE lên 1 bắt đầu quá trình đọc.

- Bước 4: Chờ đợi đoc xong bằng cách kiểm tra bit EERE nếu đã được xóa thì có dữ liệu ở thanh ghi dữ liệu. Sau đó đọc dữ liệu.

* **Bộ nhớ SRAM**

Bộ nhớ SRAM có trên hầu hết các bộ xử lý có giá thành cao trong họ AVR. Dung lượng của bộ nhớ SRAM thay đổi từ 128 byte đến 4 Kbyte. Bộ nhớ SRAM được truy nhập bằng cách sử dụng nhiều lệnh truy nhập dữ liệu trực tiếp hoặc gián tiếp. Bộ nhớ này cũng được sử dụng cho ngăn xếp. Thời gian truy nhập bộ hớ SRAM bằng 2 chu kì đồng hồ.Trên các bộ điều khiển AVR cỡ lớn đều có khả năng kết nối với bộ nhớ SRAM ở  bên ngoài. Để cho phép truy nhập bộ nhớ SRAM ngoài trên PORTA và PORTC của các bộ điều khiển cũng như tín hiệu ALE dùng cho việc phân kênh địa chỉ/dữ liệu, bit SRE(bit 7) trong thanh ghi MCUCR được đặt thành “1”.

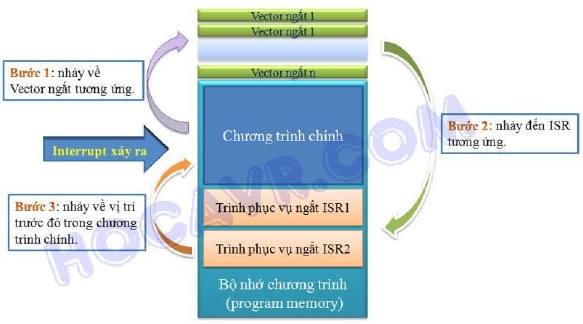
* **Các cổng vào ra**

Tất cả các bộ điều khiển AVR đều có một lượng lớn các cổng vào/ra nằm trong khoảng từ 3 bit trên AT90S2323 đến 48 bit trên Mega103. Tất cả các cổng lối ra của các bộ điều khiển AVR có thể chịu dòng điện đến 20 mA nên rất thích hợp đối với việc điều khiển trực tiếp các LED vì không cần đến mạch bổ sung. Các cổng vào ra đều có 3 địa chỉ vào ra đi kèm với chúng. Ba địa chỉ vào/ra được cần đến để đặt cấu hình cho các bit riêng biệt thành lối vào hoặc lối ra, địa chỉ khác cần đến để xuất ra dữ liệu tới các bit đó được đặt cấu hình thành lối ra, và địa chỉ thứ 3 được cần đến để đọc dữ liệu từ các chân đó được cấu hình thành lối vào

* **Truy nhập bộ nhớ và thực thi lệnh**

Bộ xử lý AVR được điều khiển bởi đồng hồ hệ thống, đồng hồ này có thể ở bên ngoài hoặc, nếu có tồn tại và được phép, một đồng hồ RC bên trong có thể được sử dụng. Đồng hồ hệ thống này không qua bất kỳ bộ chia nào và được sử dụng trực tiếp cho tất cả các thao tác truy nhập bên trong bộ xử lý. Bộ xử lý có một đường ống hai tầng, và lệnh tìm nạp/giải mã (fetch/decode) được thực hiện đồng thời với việc thực thi lệnh.Cứ mỗi lần lệnh được tìm nạp (fetch), nếu đây là một lệnh liên quan đến ALU, nó có thể được thực thi bởi khối ALU cho một chu trình đơn lẻ. Mặt khác, việc truy nhập bộ nhớ SRAM chiếm mất 2 chu kỳ. Nguyên nhân là việc truy nhập bộ nhớ SRAM sử dụng một thanh ghi con trỏ dùng cho địa chỉ bộ nhớ RAM.Thanh ghi con trỏ này chỉ là một trong các thanh ghi con trỏ (các cặp thanh ghi X, Y, hoặc Z) có trên chip. Chu trình đồng hồ thứ nhất được cần đến để truy nhập tệp thanh ghi và để thao tác trên thanh ghi con trỏ. Ở thời điểm kết thúc của chu kỳ đồng hồ thứ nhất, khối ALU thực hiện phép tính này, và sau đó địa chỉ này (hoặc đọc ra từ đó vào thanh ghi đích)

* **Ngắt trên AVR**



Hình 2.: Ngắt trên AVR

Interrupts, thường được gọi là ngắt, là một tín hiệu khẩn cấp gởi đến bộ xử lí, yêu cầu bộ xử lí tạm ngừng tức khắc các hoạt động hiện tại để “nhảy” đến một nơi khác thực hiện một nhiệm vụ khẩn cấp nào đó, nhiệm vụ này gọi là trình phục vụ ngắt – isr (interrupt service routine ). Sau khi kết thúc nhiệm vụ trong isr, bộ đếm chương trình sẽ được trả về giá trị trước đó để bộ xử lí quay về thực hiện tiếp các nhiệm vụ còn dang dở. Như vậy,ngắt có mức độ ưu tiên xử lí cao nhất, ngắt thường được dùng để xử lí các sự kiện bất ngờ nhưng không tốn quá nhiều thời gian. Các tín hiệu dẫn đến ngắt có thể xuất phát từ các thiết bị bên trong chip (ngắt báo bộ đếm timer/counter tràn, ngắt báo quá trình gởi dữ liệu bằng RS232 kết thúc…) hay do các tác nhân bên ngoài (ngắt báo có 1 button được nhấn,ngắt báo có 1 gói dữ liệu đã được nhận…). Ngắt là một cơ cấu điều khiển dòng lệnh, cơ cấu này đựơc thiết kế trên hầu hết các bộđiều khiển. Trong quá trình giao tiếp của hệ thống bộ xử lý với thế giới bên ngoài, nhiều sự việc xảy ra theo cách không đồng bộ, chắng hạn người dùng có thể nhấn một công tắc đểthực hiện một công việc nào đó, trong khi một byte dữ liệu đã đi đến cổng nối tiếp. Điều này gây khó khăn cho hoạt động của bộ xử lý khi mà nó phải kiểm tra tất cả các thiết bị đểgiám sát sự di chuyển của dữ liệu. Ngược lại mọi việc sẽ tốt hơn nếu các thiết bị này có thể loan báo sự đến nơi của dữ liệu. Đây là tất cả những gì mà cơ chế ngắt phải thực hiện.Thiết bị ngoại vi sẽ ngắt việc thực thi của chương trình chính, và bộ xử lý tạm ngưng việc thực thi chương trình bình thường để thẩm tra nguồn ngắt và để thực hiện những thao tác đáp ứng cần thiết. Sau khi hoàn thành những thao tác đáp ứng cầm thiết việc thực thi chương trình bị ngắt lại tiếp tục. Chương trình ngắt chỉ đơn giản giống hệt một chương trình con bình thường ngoại trừ một đặc điểm là việc thực thi của chương trình này không bị bộ xử lý chặn trước là sẽ xuất hiện ở một thời điểm cụ thể nào.Thứ tự các sự việc xuất hiện khi xảy ra ngắt:

1. Thiết bị ngoại vi ngắt bộ xử lý.

2. Thực hiện nốt lệnh hiện tại

3. Địa chỉ lệnh tiếp theo được lưu trữ vào ngăn xếp

4. Địa chỉ của chương trình con phục vụ ngắt được nạp vào bộ đếm chương trình.

5. Bộ xử lý thực thi chương trình con phục vụ ngắt

6. Trở lại chương trình bị ngắt bằng lệnh RETI.

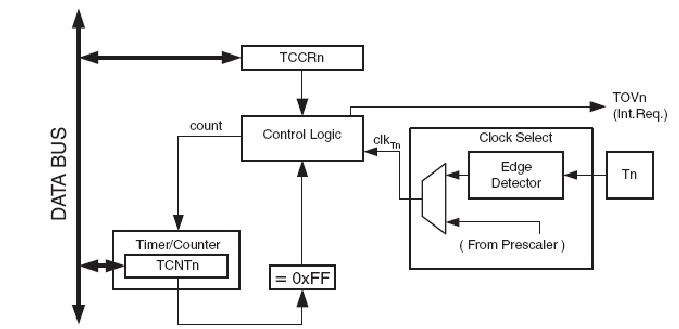
7. Bộ xử lý nạp bộ đếm chương trình với giá trị được lưu trữ trên ngăn xếp và việc thực thi chương trình bình thường lại tiếp tục

Bởi vì ngắt có thể xuất hiện bất cứ lúc nào nên trạng thái bộ xử lý cần phải được lưu trữ sao cho việc thực thi chương trình bình thường có thể tiếp tục ngay sau khi đoạn chương trình ISR được hoàn tất. Trạng thái của bộ xử lý được chứa trong thanh ghi SREG. Thanh ghi ISR cần phải lưu trữ SREG trước khi thực thi bất kỳ một lệnh nào khác, và trước khi trở lại với việc điều khiển chương trình chính cần phải khôi phục lại thanh ghi SREG. Yêu cầu này có thể được thực hiện theo hai cách: hoặc là SREG được sao chép sang một thanh ghi khác ký hiệu là RI, thanh ghi này cần phải không được phục vụ cho bất kỳ một mục đích nào khác và trước khi ISR thực thi lệnh RETI, RI được sao chép ngược trở lại SREG. Một cách khác để lưu trữ SREG là lưu nó trên ngăn xếp (bằng cách dùng lệnh PUSH SREG) và sau đó trước khi thực hiện lệnh RETI, giá trị SREG được sao chép ngược trở lại từ ngăn xếp(lệnh POP SREG). Phương pháp này chỉ có thể áp dụng cho những bộ xử lý có ngăn xếp tổ chức bằng phần mềm.Cũng có khả năng ngắt một ISR nếu một ngắt khác xuất hiện và cờ ngắt toàn cục đã được đặt thành “1” bên trong ISR dùng cho ngắt 1 (bằng cách sử dụng lệnh SEI). Trong trường hợp này, ngắt ISR1 bị ngắt và ISR khác, ISR2 thực thi. Việc thực thi ISR1 lại tiếp tục sau khi ISR2 kết thúc, và sau khi thực thi xong ISR1 chương trình chính lại tiếp tục. Bình thường sau khi một ngắt xuất hiện và đang đựoc phục vụ bởi thanh ghi ISR tương ứng, các ngắt toàn cục bị cấm (tương đương việc thực hiện lênh CLI), tuy nhiên vẫn có khả năng cho phép các ngắt trong khi một ISR đang thực thi bằng việc thực hiện lệnh SEI trong ISR.. Nếu như các ngắt khác xuất hiện trong khoảng thời gian khi một ISR đang hoạt động thì nó sẽ được thực hiện bằng việc ngắt đoạn chương trình ISR ban đầu

Tính ưu tiên của ngắt: đựoc quy định bởi cách gán các vectơ ngắt. Một vectơ ngắt ở địa chỉ thấp hơn trong bộ nhớ chương trình có mức ưu tiên cao hơn. Mức độ ưu tiên của ngắt dung để quyết định xem ngắt nào được phục vụ trước nếu như có nhiều ngắt đang chờ xử lý ở cùng một thời điểm.Một điều rất đáng quan tâm khi sử dụng các ngắt là bộ vi xử lý có thể đáp ứng ngắt nhanh đến mức nào. Câu trả lời phụ thuộc rất nhiều vào kiến trúc của bộ vi xử lý. Đối với các bộ điều khiển AVR, việc thực thi ngắt đáp ứng đối với tất cả các ngắt AVR đã được cho phép ít nhất là 4 chu kỳ đồng hồ. Bốn chu kỳ đồng hồ sau khi cờ ngắt đã được đặt, chương trình vectơ địa chỉ dùng cho đoạn chương trình xử lý ngắt hiện tại được thực thi. Trong khoảng thờigian 4 chu kỳ máy bộ đếm chương trình(2 byte) được đẩy lên ngăn xếp, và con trỏ ngăn xếp thì giảm đi 2. Véctơ thường là một lệnh nhảy tương đối đến hưởng trình ngắt, và thao tác nhảy này chiếm 2 chu kỳ đồng hồ. Nếu như một ngắt xuất hiện trong khi đang thực thi một lệnh chiếm hiều chu kỳ, lệnh này được hoàn thành trước khi ngắt được phục vụ. Việc quay trở lại từ một chương trình xử lý ngắt chiếm 4 chu kỳ đồng hồ. Trong bốn chu kỳ đồng hồ này, bộ đếm chương trình (2 byte) được đẩy trở lại ngăn xếp, con trỏ ngăn xếp được tăng thêm 2 và cờ I trong SREG đựơc đặt. Khi vi điều khiển AVR thoát ra khỏi một ngắt nó sẽ luôn trở về với chương trình chính và chấp hành lệnh kế tiếp trước khi có ngắt mới được xử lý.

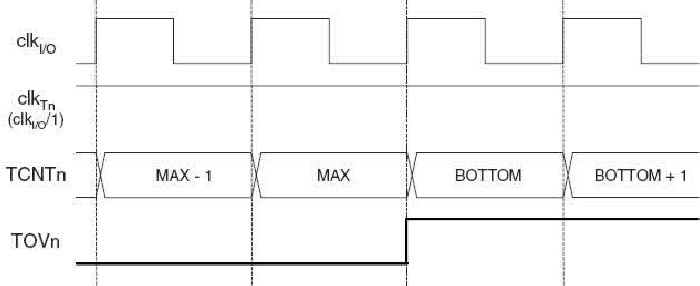
* **Timer – Counter**

Timer/Counter là các module độc lập với CPU. Chức năng chính của các bộ Timer/Counter, như tên gọi của chúng, là định thì (tạo ra một khoảng thời gian, đếm thờigian…) và đếm sự kiện. Trên các chip AVR, các bộ Timer/Counter còn có thêm chức năng tạo ra các xung điều rộng PWM (Pulse Width Modulation), ở một số dòng AVR, một số Timer/Counter còn được dùng như các bộ canh chỉnh thời gian (calibration) trong các ứng dụng thời gian thực. Các bộ Timer/Counter được chia theo độ rộng thanh ghi chứa giá trị định thời hay giá trị đếm của chúng, cụ thể trên chip Atmega8 có 2 bộ Timer 8 bit(Timer/Counter0 và Timer/Counter2) và 1 bộ 16 bit (Timer/Counter1). Chế độ hoạt động và phương pháp điều khiển của từng Timer/Counter cũng không hoàn toàn giống nhau



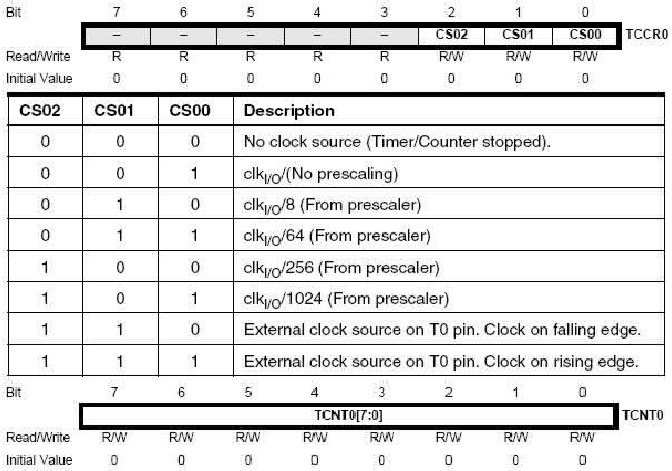
Hình 2.: Sơ đồ cấu trúc của bộ định thời

Hoạt động của bộ Timer/Couter + Mạch đếm lên làm thanh ghi TCNTn tăng 1 đơn vị mỗi khi có xung clkTn, khi đạt giá trịlớn nhất (8bit=255), cờ TOVn được set (logic 1) và bộ đếm tràn, giá trị bộ đến TCNTn trở về 00 và tiếp tục đếm.+ Xung clkTn có thể được lựa chọn từ nhiều nguồn khác nhau. Khi chọn xung nội (systemclock), Timer/Counter là một Timer. Khi chọn xung ngoài (thông qua chân Tn)Timer/Counter là Counter. Hoạt động này có thể diễn tả bằng giản đồ xung sau



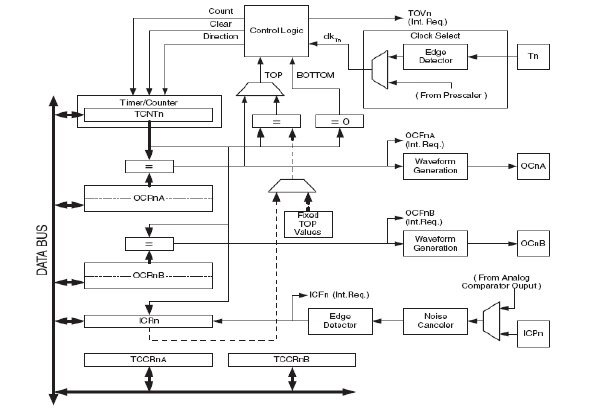
Hình 2.: Giản đồ xung hoạt đông của bộ định thời

Cũng giống như bộ timer/counter trong các vi điều khiển khác, chúng ta quan tâm đến 2 thanh ghi: Timer/Counter Control và Timer/Counter Value. Trong AVR, đó là thanh ghi TCCRn và TCNTn



Hình 2.: Giá trị các thanh ghi

TCNT0 - Timer/CTCNT0 và OCR0 là các thanh ghi 8 bit. Các tín hiệu yêu cầu ngắt đều nằm trong thanhghi TIFR. Các ngắt có thể được che bởi thanh ghi TIMSK.Bộ định thời có thể sử dụng xung clock nội thông qua bộ chia hoặc xung clock ngoàitrên chân T0. Khối chọn xung clock điều khiển việc bộ định thời/bộ đếm sẽ dùng nguồnxung nào để tăng giá trị của nó. Ngõ ra của khối chọn xung clock được xem là xung clock của bộ định thời (clk T0).Thanh ghi OCR0 luôn được so sánh với giá trị của bộ định thời/bộ đếm. Kết quả so sánh có thể được sử dụng để tạo ra PWM hoặc biến đổi tần số ngõ ra tại chân OC0



Hình 2.: Sơ đồ khối Timer1

Bộ định thời (timer/counter1) là một module định thời/đếm 16 bit, có các đặc điểm sau:

- True 16-bit Design (i.e., allows 16-bit PWM)

- 2 đơn vị ngõ vào so sánh độc lập (Two Independent Output Compare Units)

- Đôi thanh ghi so sánh ngõ ra đệm (Double Buffered Output Compare Registers)

- 1 đơn vị chốt ngõ vào (One Input Capture Unit)

- Bộ chống nhiễu lối vào (Input Capture Noise Canceler)

- Xóa timer trong Compare Match (Clear Timer on Compare Match (Auto Reload))

- Chống nhiễu sọc ngang Glitch-free, Phase Correct Pulse Width Modulator (PWM)

- Giá trị chu kỳ PWM

- Bộ phát tần số chung

- Bộ đếm sự kiện ngoài

* **Các chế độ truy nhập địa chỉ AVR**

Địa chỉ thanh ghi đơn trực tiếp: ở chế độ này địa chỉ các thanh ghi được lấy trực tiếp từ vùng các thanh ghi (từ vùng 0 đến 31)

Ví dụ: COM Rd NEG Rd…- Địa chỉ hai thanh ghi trực tiếp: đây là chế độ mà trong một lệnh ALU truy nhập trực tiếp vào hai thanh ghi. Chế độ này hoàn toàn tương tự như chế độ trên.

Ví dụ: ADD Rd, Rr …- Địa chỉ trực tiếp cổng vào ra: trong đó địa chỉ toán hạng được chứa trong 6 bit cảu một từ lệnh là địa chỉ củathanh ghi nguồn hoặc đích.

Ví dụ: Out DDRB, R16In R12, DDRB - Địa chỉ trực tiếp dữ liệu - Địa chỉ của dữ liệu trong RAM được đưa trực tiếp vào lệnh

Ví dụ: RDS R12,0x0fff STS 0x0fff,R11- Địa chỉ dữ liệu gián tiếp cùng với dịch chuyển Địa chỉ của toán hạng nguồn hoặc đích được trỏ bởi thanh ghi Y hoặc Z cộngthêm 1 chỉ số nào đó

Ví dụ: LDD,Y + 10- Địa chỉ gián tiếp dữ liệu

Đây là cách mà CPU nhập dữ liệu trong RAM thông thanh ghi X,Y,Z. Địa chỉ của dữ liệu được lưu trong thanh ghi này.

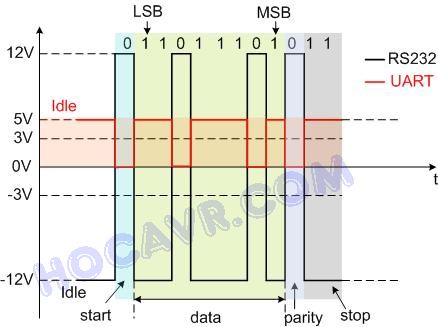
Ví dụ:ST X,R11LD R13,Y- Địa chỉ dữ liệu gián tiếp cùng với tăng hoặc giảm con trỏ

Ví dụ: LD R17, X+LD -Y,R14…- Địa chỉ của hằng số trong bộ nhớ chương trình, cách này chỉ sử dụng cho lệnh LPM, địa chỉ của hằng số được lưu trong thanh ghi Z.

## 2.3 Tìm hiểu về truyền thông UART

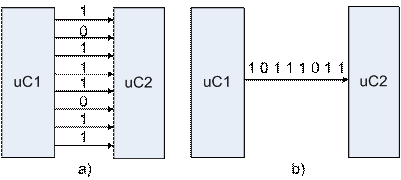
### ****2.3.1 Truyền thông nối tiếp không đồng bộ****

       Thuật ngữ USART trong tiếng anh là viết tắt của cụm từ: Universal Synchronous & Asynchronous serial Reveiver and Transmitter, nghĩa là bộ truyền nhận nối tiếp đồng bộ và không đồng bộ. Cần chú ý rằng khái niệm USART (hay UART nếu chỉ nói đến bộ truyền nhận không đồng bộ) thường để chỉ thiết bị phần cứng (device, hardware), không phải chỉ một chuẩn giao tiếp. USART hay UART cần phải kết hợp với một thiết bị chuyển đổi mức điện áp để tạo ra một chuẩn giao tiếp nào đó. Ví dụ, chuẩn RS232 (hay COM) trên các máy tính cá nhân là sự kết hợp của chip UART và chip chuyển đổi mức điện áp. Tín hiệu từ chip UART thường theo mức TTL: mức logic high là 5, mức low là 0V. Trong khi đó, tín hiệu theo chuẩn RS232 trên máy tính cá nhân thường là -12V cho mức logic high và +12 cho mức low (tham khảo hình 1). Chú ý là các giải thích trong tài liệu này theo mức logic TTL của USART, không theo RS232.



Hình 2.: Tín hiệu tương đương của UART và RS232

**Truyền thông nối tiếp:** giả sử bạn đang xây dựng một ứng dụng phức tạp cần sử dụng nhiều vi điều khiển (hoặc vi điều khiển và máy tính) kết nối với nhau. Trong quá trình làm việc các vi điều khiển cần trao đổi dữ liệu cho nhau, ví dụ tình huống Master truyền lệnh cho Slaver hoặc Slaver gởi tín hiệu thu thập được về Master xử lí…Giả sử dữ liệu cần trao đổi là các mã có chiều dài 8 bits, bạn có thể sẽ nghĩ đến cách kết nối đơn giản nhất là kết nối 1 PORT (8 bit) của mỗi  vi điều khiển với nhau, mỗi line trên PORT sẽ chịu trách nhiệm truyền/nhận 1 bit dữ liệu. Đây gọi là cách giao tiếp song song, cách này là cách đơn giản nhất vì dữ liệu được xuất và nhận trực tiếp không thông qua bất kỳ một giải thuật biến đổi nào và vì thế tốc độ truyền cũng rất nhanh. Tuy nhiên, như bạn thấy, nhược điểm của cách truyền này là số đường truyền quá nhiều, bạn hãy tưởng tượng nếu dữ liệu của bạn có giá trị càng lớn thì số đường truyền cũng sẽ nhiều thêm. Hệ thống truyền thông song song thường rất cồng kềnh và vì thế kém hiệu quả. Truyền thông nối tiếp sẽ giải quyết vần đề này, trong tuyền thông nối tiếp dữ liệu được truyền từng bit trên 1 (hoặc một ít) đường truyền. Vì lý do này, cho dù dữ liệu của bạn có lớn đến đâu bạn cũng chỉ dùng rất ít đường truyền. Hình 2 mô tả sự so sánh giữa 2 cách truyền song song và nối tiếp trong việc truyền con số 187 thập phân (tức 10111011 nhị phân).



Hình 2.: Truyền 8 bit theo phương pháp song song và nối tiếp

       Một hạn chế rất dễ nhận thấy khi truyền nối tiếp so với song song là tốc độ truyền và độ chính xác của dữ liệu khi truyền và nhận. Vì dữ liệu cần được “chia nhỏ” thành từng bit khi truyền/nhận, tốc độ truyền sẽ bị giảm. Mặt khác, để đảm bảo tính chính xác của dữ liệu, bộ truyền và bộ nhận cần có những “thỏa hiệp” hay những tiêu chuẩn nhất định. Phần tiếp theo trong chương này giới thiệu các tiêu chuẩn trong truyền thông nối tiếp không đồng bộ.  
       Khái niệm “đồng bộ” để chỉ sự “báo trước” trong quá trình truyền. Lấy ví dụ thiết bị 1 (tb1) kết với với thiết bị 2 (tb2) bởi 2 đường, một đường dữ liệu và 1 đường xung nhịp. Cứ mỗi lần tb1 muốn send 1 bit dữ liệu, tb1 điều khiển đường xung nhịp chuyển từ mức thấp lên mức cao báo cho tb2 sẵn sàng nhận một bit. Bằng cách “báo trước” này tất cả các bit dữ liệu có thể truyền/nhận dễ dàng với ít “rủi ro” trong quá trình truyền. Tuy nhiên, cách truyền này đòi hỏi ít nhất 2 đường truyền cho 1 quá trình (send or receive). Giao tiếp giữa máy tính và các bàn phím (trừ bàn phím kết nối theo chuẩn USB) là một ví dụ của cách truyền thông nối tiếp đồng bộ.  
       Khác với cách truyền đồng bộ, truyền thông “không đồng bộ” chỉ cần một đường truyền cho một quá trình. “Khung dữ liệu” đã được chuẩn hóa bởi các thiết bị nên không cần đường xung nhịp báo trước dữ liệu đến. Ví dụ 2 thiết bị đang giao tiếp với nhau theo phương pháp này, chúng đã được thỏa thuận với nhau rằng cứ 1ms thì sẽ có 1 bit dữ liệu truyền đến, như thế thiết bị nhận chỉ cần kiểm tra và đọc đường truyền mỗi mili-giây để đọc các bit dữ liệu và sau đó kết hợp chúng lại thành dữ liệu có ý nghĩa. Truyền thông nối tiếp không đồng bộ vì thế hiệu quả hơn truyền thông đồng bộ (không cần nhiều lines truyền). Tuy nhiên, để quá trình truyền thành công thì việc tuân thủ các tiêu chuẩn truyền là hết sức quan trọng. Chúng ta sẽ bắt đầu tìm hiểu các khái niệm quan trọng trong phương pháp truyền thông này.  
       **Baud rate (tốc độ Baud):** như trong ví dụ trên về việc truyền 1 bit trong 1ms, bạn thấy rằng để việc truyền và nhận không đồng bộ xảy ra thành công thì các thiết bị tham gia phải “thống nhất” nhau về khoảng thời dành cho 1 bit truyền, hay nói cách khác tốc độ truyền phải được cài đặt như nhau trước, tốc độ này gọi là tốc độ Baud. Theo định nghĩa, tốc độ baud là số bit truyền trong 1 giây. Ví dụ nếu tốc độ baud được đặt là 19200 thì thời gian dành cho 1 bit truyền là 1/19200 ~ 52.083us.     
       **Frame (khung truyền):** do truyền thông nối tiếp mà nhất là nối tiếp không đồng bộ rất dễ mất hoặc sai lệch dữ liệu, quá trình truyền thông theo kiểu này phải tuân theo một số quy cách nhất định. Bên cạnh tốc độ baud, khung truyền là một yếu tốc quan trọng tạo nên sự thành công khi truyền và nhận. Khung truyền bao gồm các quy định về số bit trong mỗi lần truyền, các bit “báo” như bit Start và bit Stop, các bit kiểm tra như Parity, ngoài ra số lượng các bit trong một data  cũng được quy định bởi khung truyền. Hình 1 là một ví dụ của một khung truyền theo UART, khung truyền này được bắt đầu bằng một start bit,  tiếp theo là 8 bit data, sau đó là 1 bit parity dùng kiểm tra dữ liệu và cuối cùng là 2 bits stop.   
       **Start bit:**start là bit đầu tiên được truyền trong một frame truyền, bit này có chức năng báo cho thiết bị nhận biết rằng có một gói dữ liệu sắp được truyền tới. Ở module USART trong AVR, đường truyền luôn ở trạng thái cao khi nghỉ (Idle), nếu một chip AVR muốn thực hiện việc truyền dữ liệu nó sẽ gởi một bit start bằng cách “kéo” đường truyền xuống mức 0. Như vậy, với AVR bit start là mang giá trị 0 và có giá trị điện áp 0V (với chuẩn RS232 giá trị điện áp của bit start là ngược lại). start là bit bắt buộc phải có trong khung truyền.  
       **Data:** data hay dữ liệu cần truyền là thông tin chính mà chúng ta cần gởi và nhận. Data không nhất thiết phải là gói 8 bit, với AVR bạn có thể quy định số lượng bit của data là 5, 6, 7, 8 hoặc 9 (tương tự cho hầu hết các thiết bị hỗ trợ UART khác). Trong truyền thông nối tiếp UART, bit có ảnh hưởng nhỏ nhất (LSB – Least Significant Bit, bit bên phải) của data sẽ được truyền trước và cuối cùng là bit có ảnh hưởng lớn nhất (MSB – Most Significant Bit, bit bên trái).   
       **Parity bit:**parity là bit dùng kiểm tra dữ liệu truyền đúng không (một cách tương đối). Có 2 loại parity là parity chẵn (even parity) và parity lẻ (odd parity). Parity chẵn  nghĩa là số lượng số 1 trong dữ liệu bao gồm bit parity luôn là số chẵn. Ngược lại tổng số lượng các số 1 trong parity lẻ luôn là số lẻ. Ví dụ, nếu dữ liệu của bạn là 10111011 nhị phân, có tất cả 6 số 1 trong dữ liệu này, nếu parity chẵn được dùng, bit parity sẽ mang giá trị 0 để đảm bảo tổng các số 1 là số chẵn (6 số 1). Nếu parity lẻ được yêu cầu thì giá trị của parity bit là 1. Hình 1 mô tả ví dụ này với parity chẵn được sử dụng. Parity bit không phải là bit bắt buộc và vì thế chúng ta có thể loại bit này khỏi khung truyền (các ví dụ trong bài này tôi không dùng bit parity).  
       **Stop bits:** stop bits là một hoặc các bit báo cho thiết bị nhận rằng một gói dữ liệu đã được gởi xong. Sau khi nhận được stop bits, thiết bị nhận sẽ tiến hành kiểm tra khung truyền để đảm bảo tính chính xác của dữ liệu. Stop bits là các bits bắt buộc xuất hiện trong khung truyền, trong AVR USART có thể là 1 hoặc 2 bits (Trong các thiết bị khác Stop bits có thể là 2.5 bits).  Trong ví dụ ở hình 1, có 2 stop bits được dùng cho khung truyền.Giá trị của stop bit luôn là giá trị nghỉ (Idle) và là ngược với giá trị của start bit, giá trị stop bit trong AVR luôn là mức cao (5V).  
       **(Chú ý và gợi ý: khung truyền phổ biến nhất là : start bit+ 8 bit data+1 stop bit)**  
       Sau khi nắm bắt các khái niệm về truyền thông nối tiếp, phần tiếp theo chúng ta sẽ khảo sát cách thực hiện phương pháp truyền thông này trên chip AVR (cụ thể là chip Atmega32).

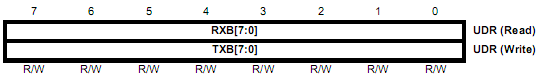
### ****2.3.2 Truyền thông nối tiếp không đồng bộ với AVR (UART)****

Vi điều khiển Atmega32 có 1 module truyền thông nối tiếp USART. Có 3 chân chính liên quan đến module này đó là chân xung nhịp - XCK (chân số 1), chân truyền dữ liệu – TxD (Transmitted Data) và chân nhận dữ liệu – RxD (Reveived Data). Trong đó chân XCK chỉ được sử dụng như là chân phát hoặc nhận xung giữ nhịp trong chế độ truyền động bộ. Tuy nhiên bài  này chúng ta không khảo sát chế độ truyền thông đồng bộ, vì thế bạn chỉ cần quan tâm đến 2 chân TxD và RxD. Vì các chân truyền/nhận dữ liệu chỉ đảm nhiệm 1 chức năng độc lập (hoặc là truyền, hoặc là nhận), để kết nối các chip AVR với nhau (hoặc kết nối AVR với thiết bị hỗ trợ UART khác) bạn phải đấu “chéo” 2 chân này. TxD của thiết bị thứ nhất kết nối với RxD của thiết bị 2 và ngược lại. Module USART trên chip Atmega32 hoạt động “song công” (Full Duplex Operation), nghĩa là quá trình truyền và nhận dữ liệu có thể xảy ra đồng thời.

**\* Thanh ghi:**

Cũng như các thiết bị khác trên AVR, tất cả hoạt động và tráng thái của module USART được điều khiển và quan sát thông qua các thanh ghi trong vùng nhớ I/O. Có 5 thanh ghi được thiết kế riêng cho hoạt động và điều khiển của USART, đó là:

**UDR:** hay thanh ghi dữ liệu, là 1 thanh ghi 8 bit chứa giá trị nhận được và phát đi của USART. Thực chất thanh ghi này có thể coi như 2 thanh ghi TXB (Transmit data Buffer) và RXB (Reveive data Buffer) có chung địa chỉ. Đọc UDR thu được giá trị thanh ghi đệm dữ liệu nhận, viết giá trị vào UDR tương đương đặt giá trị vào thanh ghi đệm phát, chuẩn bị để gởi đi. Chú ý trong các khung truyền sử dụng 5, 6 hoặc 7 bit dữ liệu, các bit cao của thanh ghi UDR sẽ không được sử dụng



**UCSRA** (USART Control and Status Register A): là 1 trong 3 thanh ghi điều khiển hoạt động của module USART.

UCSRA Register

       Thanh ghi UCSRA chủ yếu chứa các bit trạng thái như bit báo quá trình nhận kết thúc (RXC), truyền kết thúc (TXC), báo thanh ghi dữ liệu trống (UDRE), khung truyền có lỗi (FE), dữ liệu tràn (DOR), kiểm tra parity có lỗi (PE)…Bạn chú ý một số bit quan trọng của thanh ghi này:

\* UDRE (USART Data Register Empty) khi bit bày bằng 1 nghĩa là thanh ghi dữ liệu UDR đang trống và sẵn sàng cho một nhiệm vụ truyền hay nhận tiếp theo. Vì thế nếu bạn muốn truyền dữ liệu đầu tiên bạn phải kiểm tra xem bit UDRE có bằng 1 hay không, sau khi chắc chắn rằng UDRE=1 hãy viết dữ liệu vào thanh ghi UDR để truyền đi.

\* U2X là bit chỉ định gấp đôi tốc độ truyền, khi bit này được set lên 1, tốc độ truyền so cao gấp 2 lần so với khi bit này mang giá trị 0.

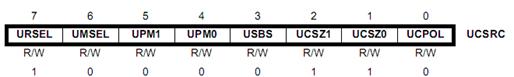
\* MPCM là bit chọn chế độ hoạt động đa xử lí (multi-processor).

**UCSRB** (USART Control and Status Register B): đây là thanh ghi quan trọng điều khiển USART. Vì thế chúng ta sẽ khảo sát chi tiết từng bit của thanh ghi này.

UCSRB Register

\* RXCIE (Receive Complete Interrupt Enable) là bit cho phép ngắt khi quá trình nhận kết thúc. Việc nhận dữ liệu truyền bằng phương pháp nối tiếp không đồng bộ thường được thực hiện thông qua ngắt, vì thế bit này thường được set bằng 1 khi USART được dung nhận dữ liệu.  
\* TXCIE (Transmit Complete Interrupt Enable) bit cho phép ngắt khi quá trình truyền kết thúc.  
\* UDRIE (USART Data Register Empty Interrupt Enable) là bit cho phép ngắt khi thanh ghi dữ liệu UDR trống.  
\* RXEN (Receiver Enable) là một bit quan trọng điều khiển bộ nhận của USART, đề kích hoạt chức năng nhận dữ liệu bạn phải set bit này lên 1.  
\* TXEN (Transmitter Enable) là bit điều khiển bộ phát. Set bit này lên 1 bạn sẽ khởi động bộ phát của USART.  
\* UCSZ2 (Chracter size) bit này kết hợp với 2 bit khác trong thanh ghi UCSRC quy định độ dài của dữ liệu truyền/nhận. Chúng ta sẽ khảo sát chi tiết khi tìm hiểu thanh ghi UCSRC.  
\* RXB8 (Receive Data Bit 8) gọi là bit dữ liệu 8. Bạn nhớ lại rằng USART trong AVR có hỗ trợ truyền dữ liệu có độ dài tối đa 9 bit, trong khi thanh ghi dữ liệu là thanh ghi 8 bit. Do đó, khi có gói dữ liệu 9 bit được nhận, 8 bit đầu sẽ chứa trong thanh ghi UDR, cần có 1 bit khác đóng vai trò bit thứ chín, RXD8 là bit thứ chín này. Bạn chú ý là các bit được đánh số từ 0, vì thế bit thứ chín sẽ có chỉ số là 8, vì lẽ đó mà bit này có tên là RXD8 (không phải RXD9).  
\* TXB8 (Transmit Data Bit 8), tương tự như bit RXD8, bit TXB8 cũng đóng vai trò bit thứ 9 truyền thông, nhưng bit này được dung trong lúc truyền dữ liệu.

**UCSRC** (USART Control and Status Register C): thanh ghi này chủ yếu quy định khung truyền và chế độ truyền. Tuy nhiên, có một rắc rối nho nhỏ là thanh ghi này lại có cùng địa chỉ với thanh ghi UBRRH (thanh ghi chứa byte cao dùng để xác lập tốc độ baud), nói một cách khác 2 thanh ghi này là 1. Vì thế bit 7 trong thanh ghi này, tức bit URSEL là bit chọn thanh ghi. Khi URSEL=1, thanh ghi này được chip AVR hiểu là thanh ghi điều khiển UCSRC, nhưng nếu bit URSEL=0 thì thanh ghi UBRRH sẽ được sử dụng.



       Các bit còn lại trong thanh ghi UCSRC được mô tả như sau:

\* UMSEL (USART Mode Select) là bit lựa chọn giữa 2 chế độ truyền thông đồng bộ và không đồng bộ. Nếu  UMSEL=0, chế độ không đồng bộ được chọn, ngược lại nếu UMSEL=1, chế độ đồng bộ được kích hoạt.

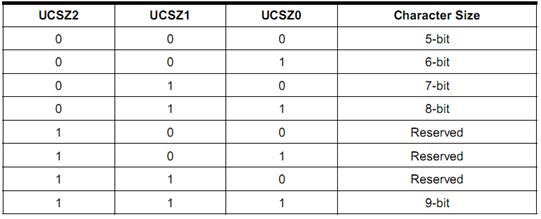
\* Hai bit UPM1 và UPM0( Parity Mode) được dùng để quy định kiểm tra pariry. Nếu UPM1:0=00, parity không được sử dụng (mode này khá thông dụng), UPM1:0=01 không được sử dụng, UPM1:0=10 thì parity chẵn được dùng, UPM1:0=11 parity lẻ được sử dụng (xem thêm bảng 1).



Bảng 2.: Chọn kiểm tra parity

\* USBS (Stop bit Select), bit Stop trong khung truyền bằng AVR USART có thể là 1 hoặc 2 bit, nếu USBS=0 thì Stop bit chỉ là 1 bit trong khi USBS=1 sẽ có 2 Stop bit được dùng.

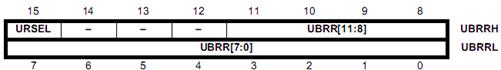
\* Hai bit UCSZ1 và UCSZ2 (Character Size) kết hợp với bit UCSZ2 trong thanh ghi UCSRB tạo thành 3 bit quy định độ dài dữ liệu truyền. Bảng 2 tóm tắt các giá trị có thcó của tổ hợp 3 bit này và độ dài dữ liệu truyền tương ứng.



Bảng 2.: Độ dài dữ liệu truyền

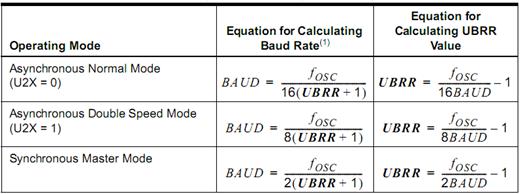
\* UCPOL (Clock Pority) là bit chỉ cực của xung kích trong chế độ truyền thông đồng bộ. nếu UCPOL=0, dữ liệu sẽ thay đổi thay đổi ở cạnh lên của xung nhịp, nếu UCPOL=1, dữ liệu thay đổi ở cạnh xuống xung nhịp. Nếu bạn sử dụng chế độ truyền thông không đồng bộ, hãy set bit này bằng 0..

UBRRL và UBRRH (USART Baud Rate Register): 2 thanh ghi thấp và cao quy định tốc độ baud.



       Nhắc lại là thanh ghi UBRRH dùng chung địa chỉ thanh ghi UCSRC, bạn phải set bit này bằng 0 nếu muốn sử dụng thanh ghi UBRRH. Như bạn quan sát trong hình trên, chỉ có 4 bit thấp của UBRRH được dùng, 4 bit này kết hợp với 8 bit trong thanh ghi UBRRL tạo thành thanh ghi 12 bit quy định tốc độ baud. Chú ý là nếu bạn viết giá trị vào thanh ghi UBRRL, tốc độ baud sẽ tức thì được cập nhật, vì thế bạn phải viết giá trị vào thanh ghi UBRRH trước khi viết vào thanh ghi UBRRL.

       Giá trị gán cho thanh ghi UBRR không phải là tốc độ baud, nó chỉ được USART dùng để tính tốc độ baud. Bảng 3 hướng dẫn cách tính tốc độ baud dựa vào giá trị của thanh ghi UBRR và ngược lại, cách tính giá trị cần thiết gán cho thanh ghi UBRR khi đã biết tốc độ baud.



Bảng 2.: Tính tốc độ baud

       Trong các công thức trong bảng 3, fOSC là tốc tần số xung nhịp của hệ thống (thạch anh hay nguồn xung nội…). Để tiện cho bạn theo dõi, tôi đính kèm bảng ví dụ cách đặt giá trị cho UBRR theo tốc độ baud mẫu.

### 2.3.3 Sử dụng UART

       Thông thường, để sử dụng module USART trên AVR bạn phải thực hiện 3 việc quan trọng, đó là: cài đặt tốc độ baud (thanh ghi UBRR), định dạng khung truyền (UCSRB, UCSRC) và cuối cùng kích hoạt bộ truyền, bộ nhận, ngắt…Như đã đề cập, trong tài liệu này tôi chủ yếu đề cập đến phương pháp truyền thông không đồng bộ, việc xác lập các thông số hoạt động chủ yếu dựa trên chế độ này. Trong hầu hết các ứng dụng, tốc độ baud và khung truyền thường không đổi, trong trường hợp này chúng ta có thể khởi tạo trực tiếp USART ở phần đầu trong main và sau đó chỉ cần truyền hoặc nhận dữ liệu mà không cần thay đổi các cài đặt. Tuy nhiên, nếu trường hợp giao tiếp “linh hoạt” ví dụ bạn đang chế tạo một thiết bị có khả năng giao tiếp với một thiết bị đầu cuối khác (như máy tính chẳng hạn), lúc này bạn nên cho phép người dùng thay đổi tốc độ baud hoặc các thông số khác để phù hợp với thiết bị đầu cuối

## 2.4 Giao tiếp SPI

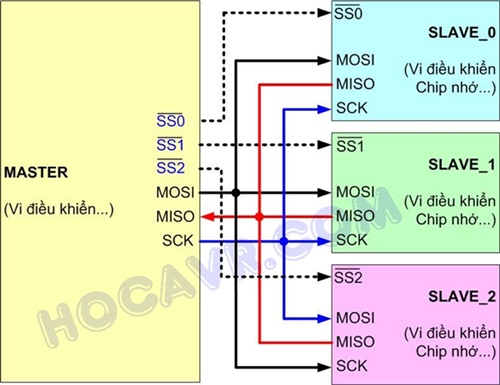
### 2.4.1 Chuẩn truyền thông SPI

       SPI (Serial Peripheral Bus) là một chuẩn truyền thông nối tiếp tốc độ cao do hang Motorola đề xuất. Đây là kiểu truyền thông Master-Slave, trong đó có 1 chip Master điều phối quá trình tuyền thông và các chip Slaves được điều khiển bởi Master vì thế truyền thông chỉ xảy ra giữa Master và Slave. SPI là một cách truyền song công (full duplex) nghĩa là tại cùng một thời điểm quá trình truyền và nhận có thể xảy ra đồng thời. SPI đôi khi được gọi là chuẩn truyền thông “4 dây” vì có 4 đường giao tiếp trong chuẩn này đó là SCK (Serial Clock), MISO (Master Input Slave Output), MOSI (Master Ouput Slave Input) và SS (Slave Select). Hình 1 thể hiện một kết SPI giữa một chip Master và 3 chip Slave thông qua 4 đường.

**SCK:** Xung giữ nhịp cho giao tiếp SPI, vì SPI là chuẩn truyền đồng bộ nên cần 1 đường giữ nhịp, mỗi nhịp trên chân SCK báo 1 bit dữ liệu đến hoặc đi. Đây là điểm khác biệt với truyền thông không đồng bộ mà chúng ta đã biết trong chuẩn UART. Sự tồn tại của chân SCK giúp quá trình tuyền ít bị lỗi và vì thế tốc độ truyền của SPI có thể đạt rất cao. Xung nhịp chỉ được tạo ra bởi chip Master.

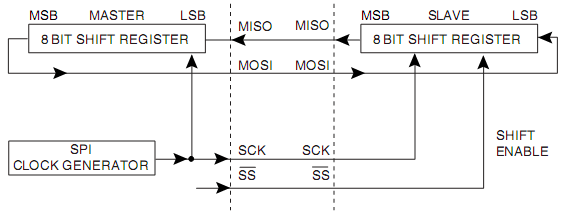
**MISO– Master Input / Slave Output:** nếu là chip Master thì đây là đường Input còn nếu là chip Slave thì MISO lại là Output. MISO của Master và các Slaves được nối trực tiếp với nhau..     
       **MOSI – Master Output / Slave Input:** nếu là chip Master thì đây là đường Output còn nếu là chip Slave thì MOSI là Input. MOSI của Master và các Slaves được nối trực tiếp với nhau.

**SS – Slave Select:**SS là đường chọn Slave cần giap tiếp, trên các chip Slave đường SS sẽ ở mức cao khi không làm việc. Nếu chip Master kéo đường SS của một Slave nào đó xuống mức thấp thì việc giao tiếp sẽ xảy ra giữa Master và Slave đó. Chỉ có 1 đường SS trên mỗi Slave nhưng có thể có nhiều đường điều khiển SS trên Master, tùy thuộc vào thiết kế của người dùng.

.

Hình : Giao diện SPI

**Hoạt động:** mỗi chip Master hay Slave có một thanh ghi dữ liệu 8 bits. Cứ mỗi xung nhịp do Master tạo ra trên đường giữ nhịp SCK, một bit trong thanh ghi dữ liệu của Master được truyền qua Slave trên đường MOSI, đồng thời một bit trong thanh ghi dữ liệu của chip Slave cũng được truyền qua Master trên đường MISO. Do 2 gói dữ liệu trên 2 chip được gởi qua lại đồng thời nên quá trình truyền dữ liệu này được gọi là “song công”. Hình 2 mô tả quá trình truyền 1 gói dữ liệu thực hiện bởi module SPI trong AVR, bên trái là chip Master và bên phải là Slave.



Hình : Truyền dữ liệu SPI

       Cực của xung giữ nhịp, phase và các chế độ hoạt động: cực của xung giữ nhịp (Clock Polarity) được gọi tắt là CPOL là khái niệm dùng chỉ trạng thái của chân SCK ở trạng thái nghỉ. Ở trạng thái nghỉ (Idle), chân SCK có thể được giữ ở mức cao (CPOL=1) hoặc thấp (CPOL=0). Phase (CPHA) dùng để chỉ cách mà dữ liệu được lấy mẫu (sample) theo xung giữ nhịp. Dữ liệu có thể được lấy mẫu ở cạnh lên của SCK (CPHA=0) hoặc cạnh xuống (CPHA=1). Sự kết hợp của SPOL và CPHA làm nên 4 chế độ hoạt động của SPI. Nhìn chung việc chọn 1 trong 4 chế độ này không ảnh hưởng đến chất lượng truyền thông mà chỉ cốt sao cho có sự tương thích giữa Master và Slave.

### 2.4.2 Truyền thông SPI trên AVR

      Module SPI trong các chip AVR hầu như hoàn toàn giống với chuẩn SPI mô tả trong phần trên. Vì thế, nếu đã hiểu cách truyền thông SPI thì sẽ khống quá khó để thực hiện việc truyền thông này với AVR. Phần bên dưới tôi trình bày một số điểm quan trọng khi điều khiển SPI trên AVR.

Các chân SPI: Các chân giao tiếp SPI cũng chính là các chân PORT thông thường, vì thế nếu muốn sử dụng SPI chúng ta cần xác lập hướng cho các chân này. Trên chip ATmega32, các chân SPI như sau:

**SCK**    – PB7 (chân 8)  
**MISO**  – PB6 (chân 7)  
**MOSI** – PB5 (chân 6)  
**SS**      – PB4 (chân 5)

      Khi chip AVR được sử dụng làm Slave, bạn cần set các chân SCK input, MOSI input, MISO output và SS input. Nếu là Master thì SCK output, MISO output, MOSI input và khi này chân SS không quan trọng, chúng ta có thể dùng chân này để điều khiển SS của Slaves hoặc bất kỳ chân PORT thông thường nào.

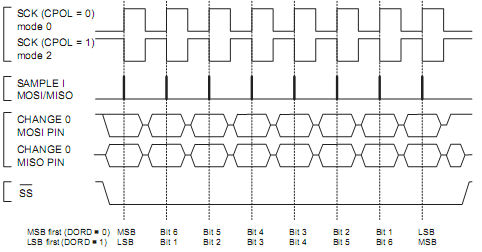
**Thanh ghi:** SPI trên AVR được vận hành bởi 3 thanh ghi bao gồm thanh ghi điều khiển SPCR , thanh ghi trạng thái SPSR và thanh ghi dữ liệu SPDR.

**SPCR** (SPI Control Register): là 1 thanh ghi 8 bit điều khiển tất cả hoạt động của SPI.

UCSRB Register

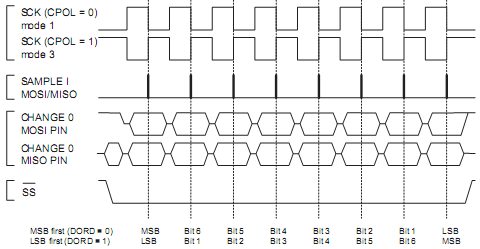
      \* Bit 7- SPIE (SPI Interrupt Enable) bit cho phép ngắt SPI. Nếu bit này được set bằng 1 và bit I trong thanh ghi trạng thái được set bằng 1 (sei), 1 ngắt sẽ xảy ra sau khi một gói dữ liệu được truyền hoặc nhận. Chúng ta nên dùng ngắt (nhất là đối với chip Slave) khi truyền nhận dữ liệu với SPI.  
     \* Bit 6 – SPE (SPI Enable). set bit này lên 1 để cho phép bộ SPI hoạt động. Nếu SPIE=0 thì module SPI dừng hoạt động.  
     \* Bit 5 – DORD (Data Order)  bit này chỉ định thứ tự dữ liệu các bit được truyền và nhận trên các đường MISO và MOSI, khi DORD=0 bit có trọng số lớn nhất của dữ liệu được truyền trước (MSB) ngược lại khi DORD=1, bit LSB được truyền trước. Thật ra khi giao tiếp giữa 2 AVR với nhau, thứ tự này không quan trọng nhưng phải đảm bảo các bit DORD giống nhau trên cả Master và Slaves.  
     \* Bit 4 – MSTR (Master/Slave Select) nếu MSTR =1 thì chip được nhận diện là Master, ngược lại MSTR=0 thì chip là Slave..  
     \* Bit 3 và 2 – CPOL và CPHA đây chính là 2 bit xác lập cực của xung giữ nhịp và cạnh sample dữ liệu mà chúng ta đã khảo sát trong phần đầu. Sự kết hợp 2 bit này tạo thành 4 chế độ hoạt động của SPI. Một lần nữa, chọn chế độ nào không quan trọng nhưng phải đảm bảo Master và Slave cùng chế độ hoạt động. Vì thế có thể để 2 bit này bằng 0 trong tất cả các chip.

**CPHA=0**



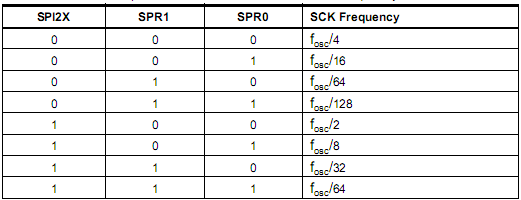
Hình : Chế độ hoạt động của SPI (CPHA=0)

**CPHA=1**



Hình : Các chế độ hoạt động của SPI (CHPA=1)

        \* Bit 1:0 – CPR1:0 hai bit này kết hợp với bit SPI2X trong thanh ghi SPSR cho phép chọn tốc độ giao tiếp SPI, tốc độ này được xác lập dựa trên tốc độ nguồn xung clock chia cho một hệ số chia. Bảng 4 tóm tắt các tốc độ mà SPI trong AVR có thể đạt. Thông thường, tốc bộ này không được lớn hơn 1/4 tốc độ xung nhịp cho chip.



Bảng 2.: Tốc độ mà SPI trong AVR đạt được

**SPSR** (SPI Status Register): là 1 thanh ghi trạng thái của module SPI. Trong thanh ghi này  chỉ có 3 bit được sử dụng. Bit 7 – SPIF là cờ báo SPI, khi một gói dữ liệu đã được truyền hoặc nhận từ SPI, bit SPIF sẽ tự động được set len 1. Bit 6 – WCOL là bít báo va chạm dữ liệu (Write Colision), bit này được AVR set lên 1 nếu chúng ta cố tình viết 1 gói dữ liệu mới vào thanh ghi dữ liệu SPDR trong khi quá trình truyền nhận trước chưa kết thúc. Bit 0 – SPI2X gọi là bit nhân đôi tốc độ truyền, bit này kết hợp với 2 bit SPR1:0 trong thanh ghi điều khiển SPCR xác lập tốc độ cho SPI.

UCSRC Register  
       **SPDR** (SPI Data Register):  là thanh ghi dữ liệu của SPI. Trên chip Master, ghi giá trị vào thanh ghi SPDR sẽ kích quá trình tuyền thông SPI. Trên chip Slave, dữ liệu nhận được từ Master sẽ lưu trong thanh ghi SPDR, dữ liệu được lưu sẵn trong SPDR sẽ được truyền cho Master.

**Sử dụng SPI trên AVR:** SPI trên AVR hoạt động không khác nguyên lý chung của chuẩn SPI là mấy. Vận hành SPI trên AVR được thực hiện dựa trên việc ghi và đọc 3 các thanh ghi SPCR, SPSR và SPDR. Trước khi truyền nhận bằng SPI chúng ta cần khởi động SPI, quá trình khởi động thường bao gồm chọn hướng giao tiếp cho các chân SPI, chọn loại giao tiếp: Master hay Slave, chọn chế độ SPI (SPOL, SPHA) và chọn tốc độ giao tiếp. Truyền thông SPI luôn được khởi xướng bởi chip Master, khi Master muốn giao tiếp với 1 Slave nào đó, nó sẽ kéo chân SS của Slave xuống mức thấp (gọi là chọn địa chỉ) và sau đó viết dữ liệu cần truyền vào thanh ghi dữ liệu SPDR, khi dữ liệu vừa được viết vào SPDR xung giữ nhịp sẽ được tự động tạo ra trên SCK và quá trình truyền nhận bắt đầu. Đối với các chip Slave, khi chân SS bị kéo xuống nó sẽ sẵn sàng cho quá trình truyền nhận. Khi phát hiện xung giữ nhịp trên SCK, Slave sẽ bắt đầu sample dữ liệu đến trên đường MOSI và gởi dữ liệu di trên MISO.

# CHƯƠNG 3: THIẾT KẾ VÀ THI CÔNG

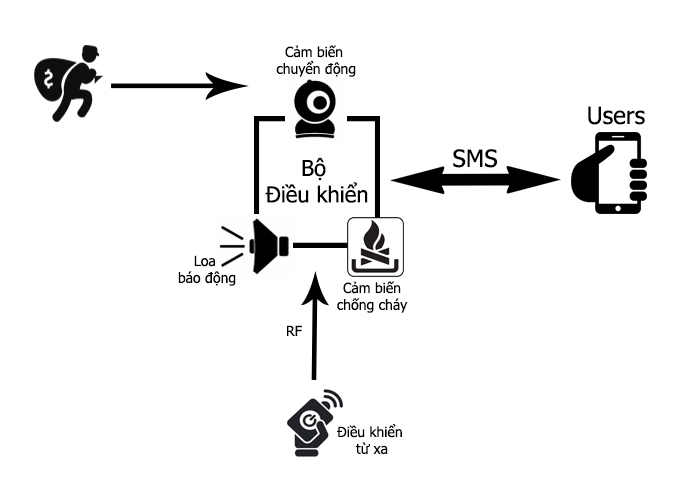
## 3.1. Phân tích yêu cầu thiết kế

Các hệ thống hoặc thiết bị điện tử thông minh được tích hợp mang vào triển khai lắp ráp trong các căn hộ hiện nay thường có những yêu cầu thi công khá phức tạp, đôi khi căn hộ phải thay đổi chính bản thân kiến trúc của chúng để phù hợp với các hệ thống và thiết bị thông minh đó. Điều này là hoàn toàn gây mất thời gian và bất tiện cho người sử dụng, đồng thời việc thay đổi kiến trúc ngôi nhà đôi khi sẽ đẩy chi phí lắp đắt lên cao, nhằm khắc phục tình trạng này em đã đưa ra ý tưởng thiết kế một thiết bị đồng nhất, không cần lắp đặt, chúng ta có thể đặt bất cứ đâu trong căn hộ và khởi động chế độ bảo vệ lên.

Yêu cầu điều khiển được đặt ra với thiết bị qua tin nhắn SMS là người sử dụng có thể dẽ dàng sử dụng điện thoại của mình để nhắn tin khởi động thiết bị, bật tắt thông báo khi ở xa. Bên canh đó với thuê bao điểu khiển chính được thiết lập sẵn trên module SIM900, người dung thuê bao này có thể thiết lập quyền quản trị cho 2 số thuê bao khác, 2 thuê bao này sẽ nhận được thông báo khi có sự cố và có thể khởi đông, bật tắt thông báo thiết bị từ xa qua tin nhắn SMS như thuê bao chính. Tuy nhiên chỉ có thuê bao chính có khả năng thay đổi quyền quản trị của các thuê bao khác hay nạp tiền cho module SIM900.

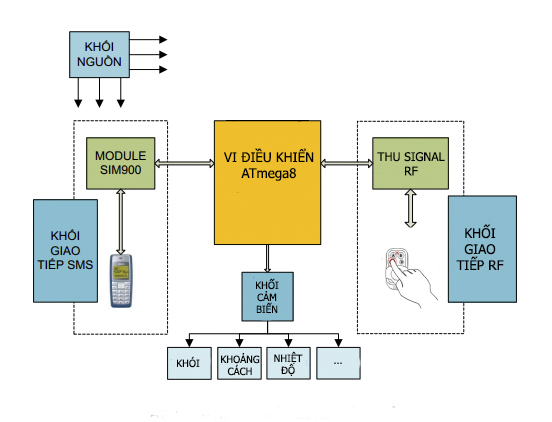
Về hướng đi trong tương lai, em sẽ cố gắng tích hợp nhiều cảm biến cho thiết bị, nâng cao khả năng giao tiếp của thiết bị hơn, tăng tính tương tác với người sử dụng cũng như tăng cao độ an toan cho căn hộ thông minh hơn.

## 3.2 Sơ đồ tổng quan của hệ thống



Hình 3.1: Sơ đồ tổng quan của hệ thống

## 3.3 Sơ đồ khối của hệ thống



Hình 3.2: Sơ đồ tổng quan của hệ thống

**Chức năng từng khối**

* Khối giao tiếp SMS

Gồm Module SIM900, có chức năng gửi nhận tin nhắn SMS cho việc điều khiển các thiết bị. Module SIM900 được kết nối với vi điều khiển ATmega8 qua giao thức UART, có nhiệm vụ nhận SMS hoặc cuộc gọi gửi tới vi điều khiển khi người sử dụng yêu cầu

* Khối giao tiếp RF

Gồm 2 module phát RF (1 ở remote điều khiển cầm tay, 2 ở trong mạch cảm biến) và bộ thu sóng RF ở tần số hoạt động 315MHz với 4 kênh. Module thu sóng RF được kết nối với vi điều khiển ATmega8 truyền nhận qua giao thức SPI. Khối này có nhiệm vụ nhân lệnh điều khiển qua Remote cầm tay, và nhận tín hiệu từ mạch cảm biến đặt bên ngoài mạch xử lý trung tâm. Vi điều khiển nhận lệnh, thực thi xong và gửi trả lại kết quả

* Khối vi điều khiển ATmega8

Khối trung tâm trong việc xử lý và điều khiển. Vi điều khiển nhận tín hiệu từ khối giao tiếp SMS, khối giao tiếp RF và mạch cảm biến sau đó đưa ra tín hiệu điều khiển

* Khối cảm biến

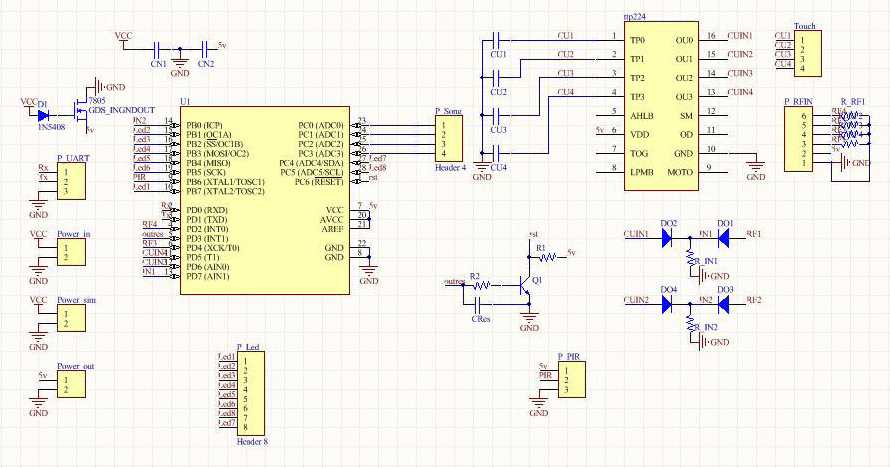
Khối này bao gồm cảm biến khói/ khí Gas… 1 module phát RF và một nguồn pin 5v, ngoài ra có thể chứa các cảm biến khác, được tích hợp sau. Khối này có nhiệm vụ phát hiện xâm nhập và các sự cố rồi gửi tín hiệu đến vi điều khiển thông qua module phát RF,

* Khối nguồn

Tạo nguồn với hai mức điện áp 5V cấp cho vi điều khiển và 3V3 cấp cho module RF. Ngoài ra module SIM900 được cấp nguồn với điện áp 4V

## 3.4 Thiết kế phần cứng mạch chính

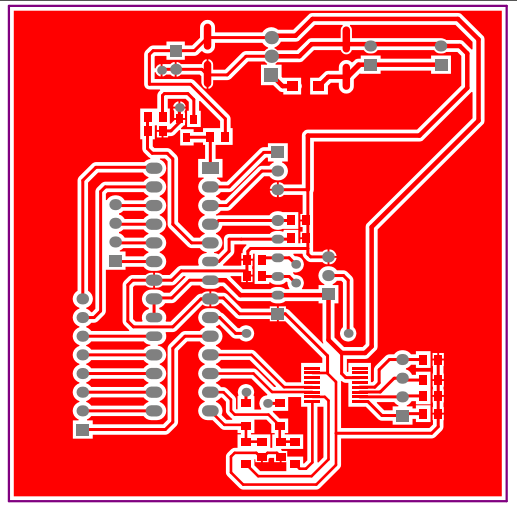
### 3.4.1 Sơ đồ nguyên lý



Hình 3.3: Sơ đồ nguyên lý các khối

* Mạch gồm các khối chính sau:
* Khối xử lý điều khiển trung tâm: vi điều khiển ATmega 8
* Khối giao tiếp với module Module SIM900 qua UART
* Khối thu RF, thu tín hiệu từ remote phát RF và tín hiệu từ khối các cảm biến đặt bên ngoài khối xử lý trung tâm
* Khối giao tiếp với module phím cảm ứng

### 3.4.2 Sơ đồ layout mạch chính trung tâm



Hình 3.4: Sơ đồ layout mạch xử lý trung tâm

## 3.5 Thiết kế và thi công Module SIM900

### 3.5.1 Giới thiệu về tập lệnh AT Command

* AT Command là gì?

Các lệnh AT là các hướng dẫn được sử dụng để điều khiển một modem. AT là một cách viết gọn của chữ Attention. Mỗi dòng lệnh của nó bắt đầu với “AT” hay “at”. Đó là lý do tại sao các lệnh modem được gọi là các lệnh AT. Nhiều lệnh của nó được sử dụng để điều khiển các modem quay số sử dụng dây mối (wired dial-up modems), chẳng hạn như ATD (Dial), ATA (Answer), ATH (Hool control) và ATO (return to online datastate), cũng được hỗ trợ bởi các modem GSM/GPRS và cá điện thoại di động. Bên cạnh bộ lệnh AT thông dụng này, các modem GSM/GPRS và các điện thoại di đọng còn được hỗ trợ bởi một tập lệnh AT đặc biệt đối với công nghệ GSM. Nó bao gồm các lệnh liên quan tới SMS như AT+CMGS (gửi tin nhắn SMS), AT+CMSS (gửi tin nhắn SMS từ một vùng lưu trữ, AT+CMGL (chuỗi liệt kê các tin nhắn SMS) và AT+CMGR (đọc tin nhắn SMS).

Ngoài ra, các modem GSM còn hỗ trợ một bộ lệnh AT mở rộng.Những lệnh AT mở rộng này được định nghĩa trong các chuẩn của GSM. Với các lệnh AT mở rộng này,bạn có thể làm một số thứ như sau:

- Đọc,viết, xóa tin nhắn.   
- Gửi tin nhắn SMS.   
- Kiểm tra chiều dài tín hiệu.   
- Đọc, viết và tìm kiếm về các mục danh bạ.

Số tin nhắn SMS có thể được thực thi bởi một modem SMS trên một phút thì rất thấp, nó chỉ khoảng từ 6 đến 10 tin nhắn SMS trên 1 phút.

### 3.5.2 Một số tập lệnh cơ bản được sử dụng

* Một số tập lệnh cơ bản

Trong khuôn khổ của đồ án em chỉ tìm hiểu 1 số tập lệnh cơ bản phục vụ cho công việc của mình. Sau đây em xin giới thiệu 1 số tập lệnh cơ bản để thao tác dùng cho dịch  vụ SMS,bao gồm:   
- Khởi tạo.   
- Nhận cuộc gọi.   
- Thiết lập cuộc gọi.   
- Nhận tin nhắn.   
- Gửi tin nhắn.

* Các thuật ngữ được sử dụng   
  - <CR> : carriage return (0x0D).   
  - <LF> : Line Feed (0x0A).   
  - MT : Mobile Terminal :Thiết bị đầu cuối mạng (chính là module).

- TE : Terminal Equipment: Thiết bị đầu cuối(chính là vi điều khiển)

* Khởi tạo cấu hình mặc định cho modem   
  (1) ATZ<CR> reset modem, kiểm tra modem đã hoạt động bình thường chưa. Gửi nhiều lần cho chắc ăn, cho đến khi nhận được chuỗi   
  ATZ<CR><CR><LF>OK<C><LF>   
  (2) ATE0<CR> tắt chế độ echo lệnh. Chuỗi trả về có dạng ATE0<CR><CR><LF>OK<CR><LF>.

(3) AT+CLIP=1<CR> định dạng chuỗi trả về khi nhân cuộc gọi. Thông thường, ở chế độ mặc định, khi có cuộc gọi đến, chuỗi trả về sẽ có dạng:   
<CR><LF>RING<CR><LF>   
Sau khi lệnh AT+CLIP=1<CR> đã được thực thi, chuỗi trả về sẽ có dạng:

<CR><LF>RING<CR><LF>

<CR><LF>+CLIP:”0939100892”,129,””,,””,0,<CR><LF>

Chuỗi tra về có chứa thông tin về số điện thoại gọi đến. Thông tin này cho phép xác định việc có nên nhận cuộc gọi hay từ chối cuộc gọi. Kết thúc các thao tác khỏi tạo cho quá trình nhận cuộc gọi đến. Các bước khỏi tạo tiếp theo liên quan đến các thao tác truyền nhận tin nhắn.   
(4) AT&W<CR> Lưu cấu hình cài đặt được thiết bị bởi các tập lệnh AET0 và   
AT+CLIP vào bộ nhớ.   
(5) AT+CMGF=1<CR> Thiết lập quá trình truyền nhận tin nhắn được thực hiện ở   
chế độ text (mặc định là ở chế độ PDU) Chuỗi trả về sẽ có dạng: <CR><LF>OK<CR><LF>   
(6) AT+CNMI=2,0,0,0,0<CR> Thiết lập chế độ thông báo cho TE khi MT nhận được tin nhắn mới. Chuỗi trả về sẽ có dạng: <CR><LF>OK<CR><LF> Sau khi lệnh trên được thiết lập, tin nhắn mới nhận được sẽ được lưu trong SIM, và MT không truyền tở về TE bất cứ thông báo nào. TE sẽ đọc tin nhắn được lưu trong sim trong trường hợp cần thiết.   
(7) AT+CSAS<CR>   
(8) Lưu cấu hình cài đặt được thiết lập bởi các lệnh AT+CMGF và AT+CNMI.

* Delete tin nhắn trong SIM.   
  (1) AT+CMGD=1   
  Xóa tin nhắn ở vùng nhớ 1 trong SIM.   
  Chuỗi trả về sẽ có dạng: <CR><LF>OK<CR><LF>   
  (2) AT+CMGD=2

Lệnh này được dùng để xóa tin nhắn được lưu trong ngăn số 2.

Có thể hình dung bộ nhớ lưu tin nhắn trong SIM bao gồm nhiều. Mỗi ngăn được đại diện bằng một số thứ tự. Khi nhận được tin nhắn mới, nội dung tin nhắn sẽ được lưu trong một ngăn trống có số thứ tự nhỏ nhất có thể. Việc xóa nội dung tin nhắn ở hai ngăn 1 và 2 cho phép tin nhắn nhận được luôn được lưu vào trong hai ô nhớ này, giúp dễ dàng xác định vị trí lưu tin nhắn vừa nhận được, và giúp cho việc thao tác với tin nhắn mới nhận được trở nên dễ dàng và đơn giản hơn, giảm khả năng việc tin nhắn mới nhận được bị thất lạc ở vùng nhớ nào đó mà ta không kiểm soát được.

Ngoài ra, khi bộ nhớ chứa tin nhắn đầy, MT sẽ không được phép nhận thêm tin nhắn mới nào nữa. Những tin nhắn được gửi đến MT trong trường hợp bộ nhớ chứa tin nhắn được gửi đến MT trong trường hợp bộ nhớ chứa tin nhắn của MT đã được đầy sẽ được lưu trên tổng đài, và sẽ được gửi đến MT sau khi bộ nhớ chứa tin nhắn của MT có xuất hiện những ngăn trống dùng để chứa tin nhắn. Việc xóa nội dung tin nhắn trong các ngăn 1 và 2 sẽ giúp đảm bảo khả năng nhận thêm tin nhắn mới của MT.

## 3.6 Khối cảm biến (Em đang layout phần này)

Khối nằm bên ngoài khối xử lý trung tâm, giao tiếp với khối xử lý trung tâm qua sóng RF

Khối này có chứa các loại cảm biến (khói, khí gas …) và một module phát RF giúp truyền tín hiệu không dây từ khối này đến khối xử lý trung tâm

### 3.6.1 Sơ đồ nguyên lý

### 3.6.2 Sơ đồ layout

## 3.7 Một số hình ảnh thực tế của mạch

# CHƯƠNG 4: KẾT LUẬN

## 4.1 Kết quả đạt được

Sau hơn 3 tháng thực hiện đề tài, em đã hoàn thành việc thiết kế và thi công mô hình hệ thống bảo vệ căn hộ từ xa qua mạng GSM. Hệ thống đã được test thử và chạy rất ổn định

* Về phần cứng: gồm 4 board
* Board Module SIM900 giao tiếp mạng GSM nhận và phản hồi tin nhắn điều khiểu 2 chiều thành công
* Board vi điều khiển ATmega 8 xử lý tin nhắn, nhận tín hiệu từ các cảm biến và xuất tín hiệu điều khiển, nhận biết trạng thái của thiết bị.
* Board Module RF giao tiếp với vi điều khiển ATmega 8 hoạt động thu phát tín hiệu ổn định
* Board nút bấm cảm ứng, giúp bật/tắt thiết bị trực tiếp
* Về phần mềm:
* Khởi tạo các thông số ban đầu cho Module SIM900 bằng cách kết nối với máy tính thông qua cổng COM
* Phân quyền quản trị với người sử dụng từ module SIM900
* Nhận và phẩn hồi tin nhắn sau khi thực hiện lệnh điều khiển

## 4.2 Ưu điểm và khuyết điểm của hệ thống

* Ưu điểm
* Sử dụng mạng viễn thông GSM trong truyền thông giúp hệ thống hoạt động ở những ví trí khó khan mà đường dây điện thoại không có, chỉ cần có mạng di động phủ sóng
* Hệ thống có thể sử dụng tất cả các sim của các nhà cung cấp khác nhau tại Việt Nam như: Mobiphone, Viettel, Vinaphone …
* Hệ thống được bảo mật thông qua số điện thoại được quy định quyền quản trị cao nhất
* Nếu ở gần thiết bị có thể dễ dàng điều khiển công tắc thông qua điều khiển cầm tay mà không phải gửi tin nhắn
* Khuyết điểm
* Hệ thống mới dừng lại ở 2 loại cảm biến
* Hệ thống chỉ hoạt động được ở vùng có phủ sóng điện thoại di động
* Phát sinh chi phí trong quá trính gửi SMS

## 4.3 Hướng phát triển trong tương lai

* Mở rộng thêm các cảm biến, thiết bị ngoại vi giúp hệ thống có nhiều chức năng bảo vệ hơn nữa
* Phát triển kết nối thông báo và điều khiển lên thiết bị di động qua 3G hoặc wifi thông qua ứng dụng cho Smartphone

# Tài liệu tham khảo

[1]. Văn Thế Minh, Kỹ thuật vi xử lý, Nhà Xuất Bản Giáo Dục, 1997

[2]. Dương Minh Trí, Sơ đồ chân linh kiện, Nhà xuất bản Khoa học kỹ thuật, 1997

[3]. Nguyễn Ngọc Khánh, Nguyễn Hiếu Nghĩa, Hệ thống thu thập và xử lý dữ liệu cho ngôi nhà thông minh, ĐH Sư Phạm Kỹ Thuật Tp HCM, 2004

[4]. SIMCom, AT Commands Set, Hardward Design Module SIM900

Các trang web tham khảo:

1. [www.atmel.com](http://www.atmel.com)

2. [www.datasheet.com](http://www.datasheet.com)

4. [www.dientuvietnam.net](http://www.dientuvietnam.net)

5. [www.hocavr.com](http://www.hocavr.com)

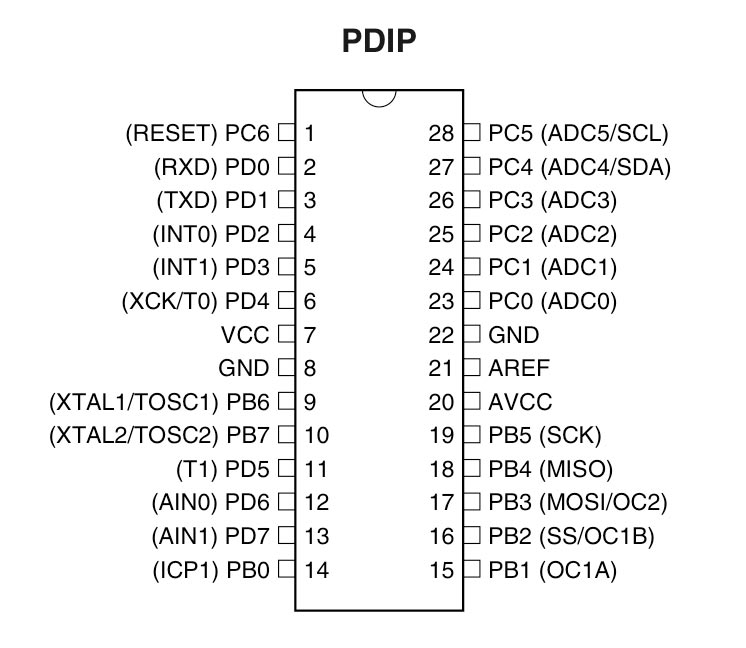
# PHỤ LỤC 1: VI ĐIỀU KHIỂN ATMEGA 8

**1.1 Các thông số cơ bản của vi điều khiển**

- Tốc độ tối đa: 16MHz.  
- Dung lượng bộ nhớ chương trình: 8 KB.  
- Bộ nhớ EEPROM: 512 Byte.  
- Dung lượng bộ nhớ RAM: 1 KB.

Bộ nhớ chương trình có khả năng ghi 10.000 lần, bộ nhớ EEPROM có thể ghi 100.000 lần. Hỗ trợ bootloader, có khả năng tự ghi vào bộ nhớ chương trình, cập nhật chương trình cho chip mà không cần mạch nạp.  
- Timer 8 bit: 2.  
- Timer 16 bit: 1.  
- ADC: 6 kênh, 10 bit.  
- Giao tiếp: TWI (I2C), UART, SPI  
- Điện áp hoạt động:  
Atmega8L: 2.7V – 5.5V.  
Atmega8: 4.5V – 5.5V.

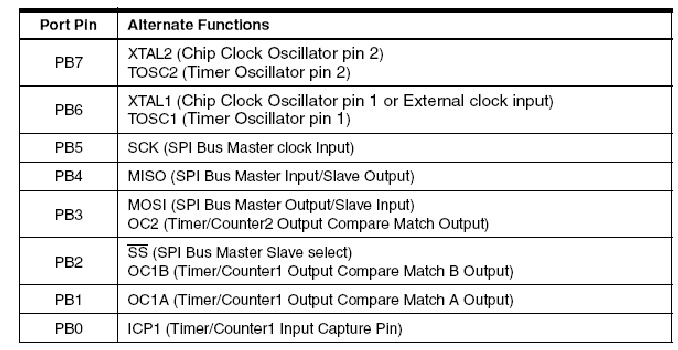
**1.2 Sơ đồ các chân và chức năng**



Hình P1.1: Sơ đồ các chân ATmega8

ATMega 8 đóng gói dạng PDIP 28 chân.Trong đó có 23 chânI/O. (in put/ out put) chia thành 3 Port: B,C,D. Mỗi một chân có thể đảm nhận nhiều vai trò. Cụ thể như sau: VCC (chân 7): Chân điện áp.GND (chân 8): Chần tiếp đất.Các cổng vào ra của Atmega 8 gồm có 3 PortB, PortC, PortD,

* PortB (PB7…PB0) XTAL2/TOSC2 – Port B, Bit 7XTAL2: Chân 2 dao động tạo clock. Sử dụng chân clock thạch anh,hoặc dao động thạch anh tần số thấp. Khi dùng chân làm dao động thì không thể làm chân nhập xuất được nữa. TOSC2: Chân 2 là dao dộng Timer. Nếu PB7 được dùng làm clock pin, DDB7,PORTB7 and PINB7 sẽ sẽ hiều là mức 0XTAL1/TOSC1 – Port B, Bit 6XTAL1: Chip clock Oscillator pin 1.TOSC1: Timer Oscillator pin 1. Nếu PB6 dùng làm chân clock, DDB6, PORTB6 and PINB6 sẽ hiều là mức 0

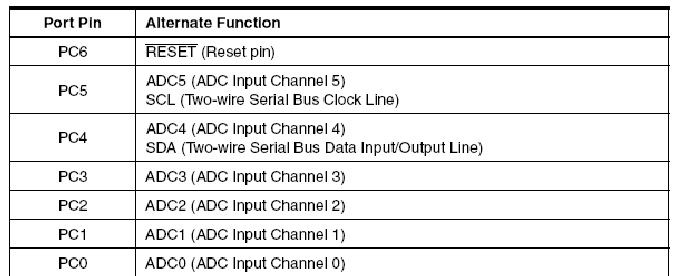


Hình P1.2: Chức năng PortB

SCK – Port B, Bit 5SCK: Master Clock output, Slave Clock input pin for SPI channel. Khi SPI được kích hoạtlà Slave, chân này được cấu hình là 1 chân ngõ vào bất chấp sự điều chỉnh từ DDB5.MISO – Port B, Bit 4MISO: Master Data input, Slave Data output pin for SPI channel. Khi SPI được kích hoạtlà Master, chân này được cấu hình là 1 chân ngõ vào bất chấp sự điều chỉnh từ DDB4.MOSI/OC2 – Port B, Bit 3MOSI: SPI Master Data output, Slave Data input for SPI channel. Khi SPI được kích hoạt là Slave, chân này được cấu hình là 1 chân ngõ vào bất chấp sự điều chỉnh từ DDB3. KhiSPI được kích hoạt là Master, dữ liệu trực tiếp của chân này được điều khiển bởi DDB3.SS/OC1B Port B, Bit 2SS: Slave Select ngõ vào. Khi SPI được kích hoạt là Slave, chân này được cấu hình là 1chân ngõ vào bất chấp sự điều chỉnh từ DDB2.OC1A – Port B, Bit 1

OC1A, Output Compare Match output: Chân PB1 có thể xử lý như 1 ngõ ra bên ngoàiTimer/Counter1 Compare Match A.ICP1 – Port B, Bit 0ICP1 –chân giữ(chốt) ngõ vào : Chân PB0 có thể tác động làm 1 chân giữ choTimer/Counter1.

* PortC ( PC6…PC0) RESET – Port C, Bit 6 RESET, Reset pin: Khi cầu chì RSTDISBL đã lập trình, chức năng của chân này là vào ra binh thường, và 1 phần sẽ phải dựa vào Power-on Reset và Brown-out Reset như là nguồnreset của nó. Nếu chân PC6 dùng là chân reset , DDC6, PORTC6 và PINC6 sẽ hiều là mức 0.

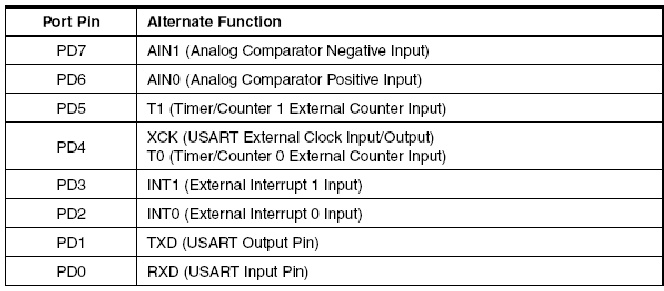


Hình P1.3: Chức năng PortC

SCL/ADC5 – Port C, Bit 5SCL, giao diện nối tiếp hai dây Xung nhịp: Khi bit TWEN trong TWCR set (one) để bậtgiao diện nối tiếp hai dây, pin PC5 bị ngắt từ port và trở thành chân Serial Clock I/O choTwo-wire Serial Interface.SDA/ADC4 – Port C, Bit 4

SDA, Two-wire Serial Interface Data: When the TWEN bit in TWCR is set (one) to enablethe Two-wire Serial Interface, pin PC4 is disconnected from the port and becomes theSerial Data I/O pin for the Two-wire Serial Interface.ADC3 – Port C, Bit 3PC3 cũng có thể dùng là ADC input Channel 3. Chú ý là ADC input channel 3 dùng nguồnxoay chiều.ADC2 – Port C, Bit 2PC2 cũng có thể dùng là ADC input Channel 2. Chú ý là ADC input channel 2 dùng nguồnxoay chiều.ADC1 – Port C, Bit 1PC1 cũng có thể dùng là ADC input Channel 1. Chú ý là ADC input channel 1 dùng nguồnxoay chiều.ADC0 – Port C, Bit 0PC0 cũng có thể dùng là ADC input Channel 0. Chú ý là ADC input channel 0 dùng nguồnxoay chiều

* PortD (PD7-PD0)

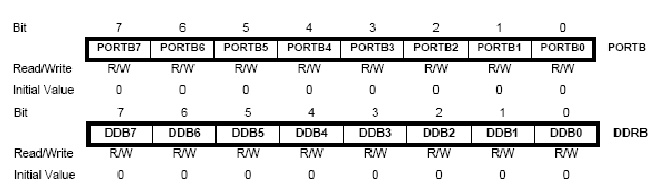
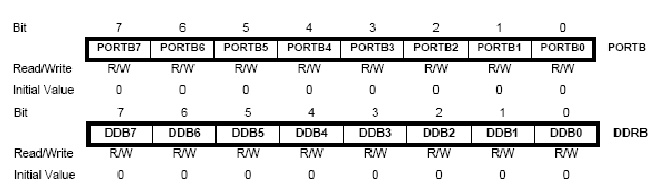


Hình P1.4: Chức năng PortD

AIN1 – Port D, Bit 7AIN1,bộ so sánh tương tự thụ động ngõ vào. Cấu hình chân của port là nhập vào với ngắt pull-up bên trong để tránh nhiễu từ port số với chức năng của bộ so sánh tương tự AIN0 – Port D, Bit 6AIN0,Bộ so sánh tương tự ngõ vao tích cực. Cấu hình chân của port là nhập vào với ngắt pull-up bên trong để tránh nhiễu từ port số với chức năng của bộ so sánh tương tự.T1 – Port D, Bit 5T1, số lượng mã nguồn Timer/Counter1.XCK/T0 – Port D, Bit 4XCK, USART xung nhịp ngoài. T0, số lượng mã nguồn Timer/Counter0.INT1 – Port D, Bit 3INT1, Ngắt nguồn bên ngoài 1: Chân PD3 có thể làm chức năng như 1 nguồn ngắt ngoài.INT0 – Port D, Bit 2INT0, Ngắt nguồn bên ngoài 0: Chân PD2 có thể làm chức năng như 1 nguồn ngắt ngoài.TXD – Port D, Bit 1TXD, Truyền tải dữ liệu (chân dữ liệu ra của USART). Khi bộ truyền USART được kíchhoạt ,chân này được cấu hình như là một ngõ ra bất kể giá trị của DDD1.RXD Port D, Bit 0RXD, Nhận dữ liệu (chân dữ liệu vào của USART). Khi bộ nhận USART được kích hoạt,chân này được cấu hình như là một ngõ vào bất kể giá trị của DDD0

Mô tả thanh ghi của port I/O

The Port B Data Register – PORTB

Hình P1.5: Mô tả thanh ghi của port I/O

**\* Tóm lại**

Để đọc dữ liệu từ ngoài thì ta phải thực hiện các bước sau:

- Đưa dữ liệu ra thanh ghi điều khiển DDRxn để đặt cho PORTx (hoặc bit n trong port) đó là đầu vào (xóa thanh ghi DDRx hoặc bit).

- Sau đó kích hoạt điện trở pull-up bằng cách set thanh ghi PORTx ( bit).

- Cuối cùng đọc dữ liệu từ địa chỉ PINxn (trong đó x: là cổng và n là bit).2. Để đưa dữ liệu từ vi điều khiển ra các cổng cũng có các bước hoàn toàn tương tự. Banđầu ta cũng phải định nghĩa đó là cổng ra bằng cách set bit tương ứng của cổng đó….vàsau đó là ghi dữ liệu ra bit tương ứng của thanh ghi PORTx

# PHỤ LỤC 2:

# GIỚI THIỆU VỀ TRÌNH BIÊN DỊCH AVRSTUDIO

**2.1 Giới thiệu WinAVR**

WinAVR là một gói các công cụ phục vụ cho việc biên dịch chương trình viết

cho AVR bằng ngôn ngữ C, tạo ra file .hex ñể nạp vào chip AVR.

WinAVR hoàn toàn tương thích với AVR Studio. Sau khi cài ñặt thành công

WinAVR bạn không cần quan tâm ñến nó nữa, mỗi khi biên dịch chương trình, AVR Studio sẽ tự động gọi WinAVR thực thi và trả về kết quả cho bạn.

WinAVR là một ứng dụng nguồn mở, hoàn toàn miễn phí, có thể dễ dàng tải về

tại <http://winavr.sourceforge.net/>.

2.2 Giới thiệu AVR Studio

AVR Studio là một phần mềm phục vụ việc lập trình (soạn thảo, biên dịch, gỡ

rối, …) các ứng dụng viết cho AVR.

AVR Studio chạy trên môi trường Windows, giao diện tương ñối dễ dùng. AVR

Studio có ñầy ñủ các công cụ quản lý project, soạn thảo source files, tích hợp sẵn trình

biên dịch hợp ngữ, chương trình nạp, hỗ trợ chạy mô phỏng và debug chương trình.

AVR Studio ñược hỗ trợ miễn phí bởi chính ATMEL, hãng sản xuất AVR và có

thể dễ dàng tải về tại <http://www.atmel.com/>.

2.3 Giới thiệu chương trình nạp AvrProg

AvrProg là một chương trình dùng ñể nạp file \*.hex từ máy tính vào vi ñiều

khiển AVR .AvrProg là một Tools của AVR Studio, vì thế khi cài ñặt AVR Studio thì có cả AvrProg, không cần phải cài ñặt thêm.

Ngoài ra, AvrProg còn có thể chạy như một phần mềm độc lập, không phụ thuộc AVR Studio.

# PHỤ LỤC 3: THÔNG SỐ KỸ THUẬT SIM900

**Các thông số kỹ thuật**  
1. Nguồn cung cấp khoảng 3,2 – 4,8V  
2. Nguồn lưu trữ  
3. Băng tần

• EGSM 900Mhz, DCS 1800 Mhz và PCS 1900Mhz  
• Phù hợp với GSM Pha 2/2+

4. Loại GSM là loại MS nhỏ  
5. Kết nối GPRS

• GPRS có nhiều rãnh loại 8 ( lựa chọn )  
• GPRS có nhiều ránh loại 10 ( tự động )

6. Giới hạn nhiêt độ

• Bình thường -900 C tới +700C

• Hạn chế : - 350C tới -900 C và +700C tới +800C

• Nhiệt độ bảo quản: -450C tới 850C

7. Dữ liệu GPRS

• GPRS dữ liệu tải xuống: Max 85.6 kbps

• GPRS dữ liệu tải lên: Max 42.8 kbps

• Sơ đồ mã hóa: CS-1, CS-2, CS-3 và CS-4

• Sim 900 hổ trợ giao thức PAP ,kiểu sử dụng kết nối PPP

• Sim 900 tích hợp giao thức TCP/IP

• Chấp nhận thông tin được điều chỉnh rộng rãi  
8. CSD

• Tốc độ truyền dẫn CSD: 2; 4; 8; 9; 6; 14 KPPS  
• Hỗ trợ USSD

9. SMS

• MT, MO, CB, Text and PDU mode

• Bộ nhớ SMS: Sim, card  
10. FAX

• Nhóm 3 loại 1  
11.Sim card

• Hỗ trợ sim card: 1,8v ; 3v  
12. Anten ngoài

• Kết nối thông qua anten ngoài 500km hoặc đế anten  
13. Âm thanh

• Dạng mã hòa âm thanh.  
• Mức chế độ (ETS 06.20)  
• Toàn bộ chế độ (ETS 06.10)  
• Toàn bộ chế độ tăng cường (ETS 06.50/ 06.06/ 06.80)  
• Loại bỏ tiếng dội

14. Giao tiếp nối tiếp và sự ghép nối

• Cổng nối tiếp: 7 Cổng nối tiếp( ghép nối)

• Cổng kết nối có thể Sd với CSD Fax, GPRS và gửi lệnh ATCommand tới mudule điều khiển

• Cổng nối tiếp có thể Sd chức năng giao tiếp

• Hỗ trợ tốc độ truyền 1200 BPS tới 115200 BPS

• Cổng hiệu chỉnh lỗi: 2 cổng nối tiếp TXD và RXD

• Cổng hiệu chỉnh lỗi chỉ sử dụng sữa lỗi  
15. Quản lý danh sách

• Hỗ trợ mẫu danh sách: SM, FD, LD, RC,ON, MC  
16. Sim Application toolkit

• Hỗ trợ SAT loại GSM 11,14 bản 99  
17. Đồng hồ thời gian thực

• Người cài đặt  
18. Times function

• Lập trình thông qua AT Command  
19. Đặc tính vật lý (đặc điểm)

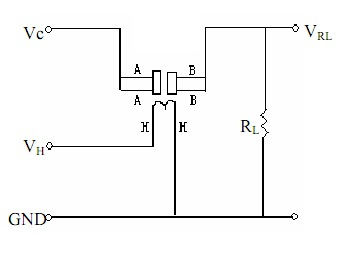
• Kích thướt 50±0.15 x 33±0.15 x7.7±0.3mm

• Nặng 13.8 kg

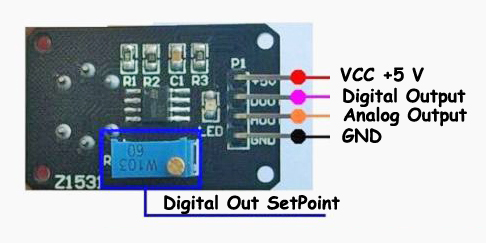
# PHỤ LỤC 4: THÔNG SỐ KỸ THUẬT CÁC CẢM BIẾN ĐƯỢC SỬ DỤNG

**4.1 Cảm biến khí Gas MQ2**

MQ2 là [cảm biến khí](http://mcu.banlinhkien.vn/tags/c%E1%BA%A3m+bi%E1%BA%BFn+kh%C3%AD/), dùng để phát hiện các khí có thể gây cháy. Nó được cấu tạo từ chất bán dẫn SnO2. Chất này có độ nhạy cảm thấp với không khí sạch. Nhưng khi trong môi trường có chất ngây cháy, độ dẫn của nó thay đổi ngay. Chính nhờ đặc điểm này người ta thêm vào mạch đơn gian để biến đổi từ độ nhạy này sang điện áp.  
- Khi môi trường sạch điện áp đầu ra của cảm biến thấp, giá trị điện áp đầu ra càng tăng khi nồng độ khí gây cháy xung quang MQ2 càng cao.  
- MQ2 hoạt động rất tốt trong môi trường khí hóa lỏng LPG, H2, và các chất khí gây cháy khác. Nó được sử dụng rộng rãi trong công nghiệp và dân dụng do mạch đơn giản và chi phí thấp.



Hình P4.1: Sơ đồ nguyên lý cảm biến MQ2



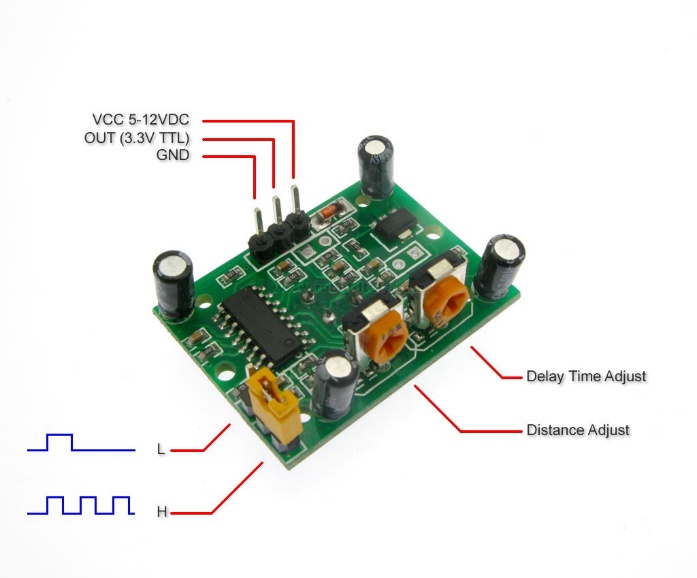
Hình P4.2: Sơ đồ các chân của MQ2

**4.2 Cảm biến chuyển động**



Hình P4.3: Cảm biến chuyển động SC-HR 501

 Sử dụng điện áp: 4.5-20V  
- Đầu ra: 0-3.3V  
- Thứ tự chân: **Vcc OUT GND**- 2 chế độ hoạt động:  
        + (L) không lặp lại kích hoạt   
        + (H) lặp lại kích hoạt.   
- Thời gian trễ: điều chỉnh trong khoảng 0.5-200S.  
- Kích thước PCB:32mmx24mm  
- Góc quét <100 độ.  
- Sử dụng cảm biến: 500BP  
- Khoảng các phát hiện: 2-4.5m



Hình P4.4: Sơ đồ chân của HC-SR501