**MỤC LỤC**

**TRANG DANH MỤC HÌNH ẢNH**

**TRANG DANH MỤC CÁC BẢNG BIỂU**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Tên đầy đủ** | **Giải thích** |
| 1 | API | Application Programming Interface | Giao diện lập trình ứng dụng |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TRANG DANH MỤC CÁC TỪ VIẾT TẮT**

CHƯƠNG 1: GIỚI THIỆU

* 1. Tổng quan

Sự phát triển của CNTT và internet đã làm cho cuộc sống con người thay đổi đáng kể. Chỉ một chiếc máy tính kết nối mạng, chúng ta có thể điều khiển mọi thứ thông qua smartphone hay máy tính.

Ngay từ tên gọi của nó, Internet Of Things viết tắt là IOT chính là internet trong mọi thứ. Và theo định nghĩa thì IOT chính là mạng lưới vạn vật kết nối Internet hoặc mạng lưới kết nối thiết bị Internet . Là một kịch bản của thế giới, khi mà mỗi đồ vật, con người được cung cấp một định danh riêng của nó và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính.

Khi mà vạn vật đều có chung một mạng kết nối thì việc liên lạc và làm việc trở nên rất dễ dàng. Con người có thể hiện thực hóa mục đích của mình trong tương lai. Chúng ta hoàn toàn có thể kiểm soát mọi thứ. Giả sử 1 chiếc ví mà các bạn đang sử dụng có tích hợp công nghệ IOT. Chúng có nhiệm vụ kiểm tra số lượng tiền trong ví, kiểm tra ngày hết hạn của các giấy tờ mà các bạn để trong đó như: bảo hiểm y tế, hạn nộp học phí,.. và thông báo tình trạng của nó đến cho chúng ta biết thông qua các ứng dụng tin nhắn SMS, facebook, skype, zalo,…

Hay như một hệ thống tưới nước tự động cây cối trong gia đình bạn được tích hợp công nghệ IOT. Giúp bạn điều khiển qui trình chăm sóc cây, tưới nước cây, thậm chí là bắt sâu bọ,…khi bạn có chuyến đi công tác xa vài ngày hay vài tháng mà không thể thực hiện được các chức năng đó. Điều đó sẽ trở nên rất đơn giải khi giả sử mà hệ thống tưới cây tự động và điện thoại hoặc laptop, PC,.. của bạn được kết nối và mạng lưới Internet và qua đó có thể trao đổi thông tin cũng như thực thi các câu lệnh mà bạn mong muốn.

Điều kì diệu ở đây đó chính là cảm biến. Các thiết bị cần kết nối phải được tích hợp một chip cảm biến để có thể chuyển đổi, phát hiện các hiện tượng trong môi trường tự nhiên và biến nó thành dữ liệu trong môi trường Internet để xử lý dữ liệu và tiến hành thực thi các điều hướng trong mạng Internet đó theo cách mà người dùng mong muốn.

Lấy ví dụ hệ thống tưới nước cây tự động như ở trên thì hệ thống sinh thái của chúng ta phải được gắn 1 bộ cảm biến dùng để nhận biết các yếu tố như: nhiệt độ, lượng nước, độ ẩm, thời tiết,… Sau đó được chuyển thành dữ liệu và các dữ liệu này được sử dụng và được thiết lập các thiết lập các chế độ theo mục đích sử dụng.  Và qui trình này sẽ kết nối và hoạt động trong môi trường Internet để thông báo và tạo giao diện đến người dùng.

Thực ra khái niệm Internet Of Things được đưa ra vào năm 1999. Khi mà công nghệ Internet đang từng bước phát triển. Và cho đến ngày nay, Internet Of Things không còn là 1 vấn đề quá viễn vông, cụ thể là rất nhiều các phát mình, ứng dụng được trình làng. Khái niệm thiết bị thông minh, tủ lạnh thông minh, tivi thông minh, … được sử dụng rỗng rãi trong thể giới công nghệ ngày nay. Các bạn có thể điều khiển 1 chiếc tivi bằng điều hướng bàn tay, giọng nói,…bằng công nghệ smart tivi của hãng Samsung, máy lạnh tự động điều chỉnh nhiệt độ theo thời tiết,…Hay như xe ô tô tích hợp chức năng chống sốc tự dộng, tự động báo cho người sử dụng khi lốp xe bị xẹp hay gặp trúng vật cản phía trước khoảng bao nhiêu mét chẳng hạn. Thậm chí là ngôi nhà – nơi chúng ta đang sống cũng có định nghĩa là ngôi nhà thông minh với rất nhiều ứng dụng công nghệ hiện đại. Rất và rất nhiều những ứng dụng trong Internet Of Things đã được các công ty công nghệ khai thác vấn đề này.

Còn với quốc phòng, chi tiêu cho các thiết bị bay không người lái dự kiến sẽ đạt 8,7 tỉ USD vào năm 2020. Ngoài ra, theo dự báo của Frost & Sullivan, sẽ có khoảng 126.000 robot quân sự sẽ được triển khai vào năm 2020. Về nhà kết nối, tới năm 2030, phần lớn các thiết bị trong gia đình sẽ được kết nối Internet. Các nhà sản xuất cũng sẽ trang bị tính năng kết nối cho bất cứ vật dụng nào mà họ sản xuất ra.

Lĩnh vực nông nghiệp cũng không nằm ngoài vòng xoáy IoT. Dự kiến sẽ có 75 triệu thiết bị IoT được triển khai trong lĩnh vực này vào năm 2020, với tỉ lệ tăng hàng năm đạt 20%. Chủ yếu đó sẽ là những bộ cảm biến đặt trong lòng đất để theo dõi độ axit, nhiệt độ và các thông số giúp canh tác vụ mùa hiệu quả hơn.

Trong lĩnh vực dịch vụ ăn uống, sẽ có 310 triệu thiết bị IoT được các công ty cung cấp dịch vụ ăn uống sử dụng vào năm 2020. Phần lớn các thiết bị này sẽ kết nối thông tin tới quầy bán thực phẩm và các công ty sản xuất đồ ăn nhanh. Vì thế, Internet of Thing đang là khóa của thành công trong tương lai.

* 1. Mục tiêu đề tài

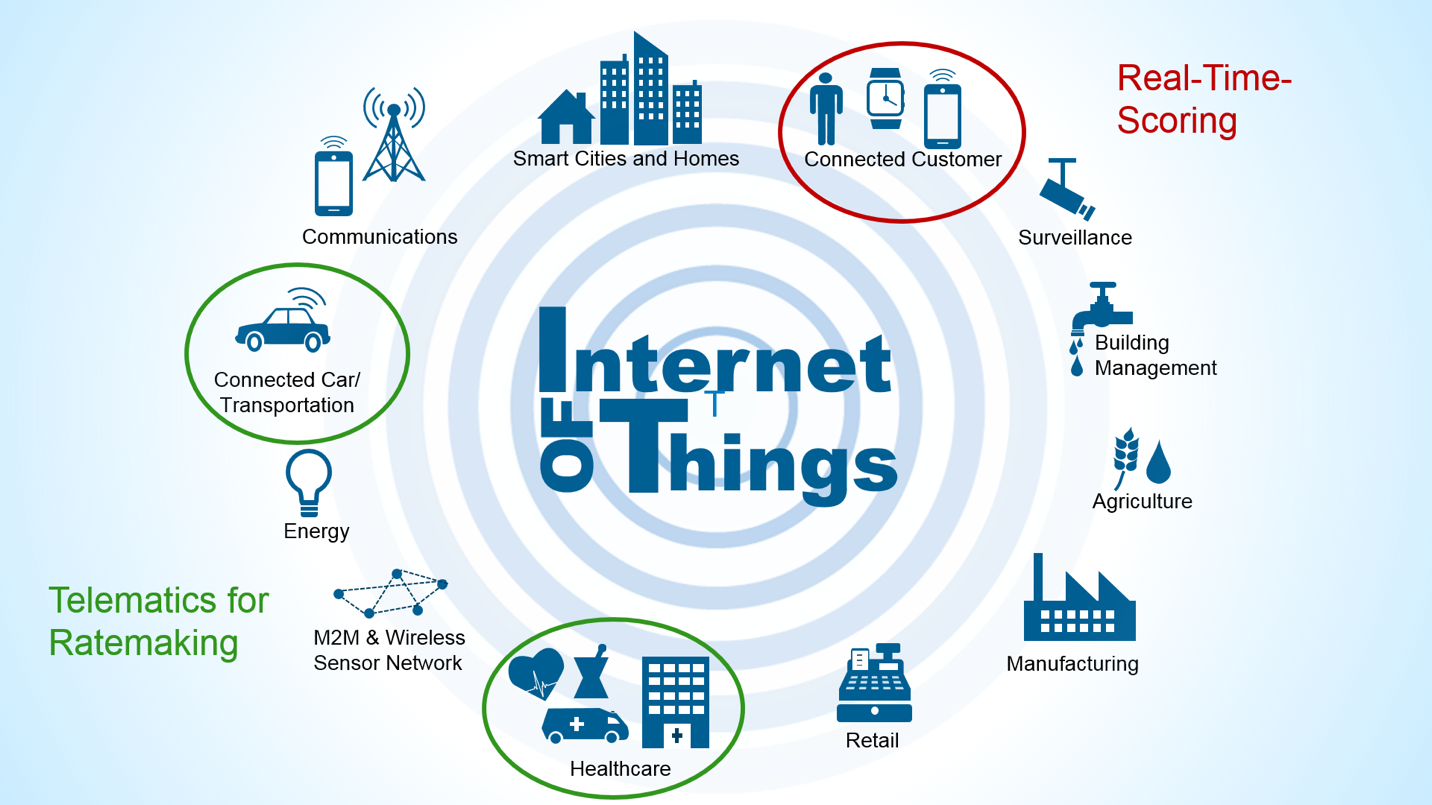
Xây dựng mô hình cơ bản về hệ thống giám sát giấc ngủ của trẻ, nhằm đáp ứng nhu cầu của những bậc cha mẹ bận rộn trong thời đại công nghệ hiên nay. Sản phẩm có chức năng theo dõi hoạt động của trẻ khi đang ngủ.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

* 1. **Tìm hiểu IOT**
     1. **Định nghĩa**

Thiết bị (devices): Đối với Internet Of Things, đây là một phần của cả hệ thống với chức năng bắt buộc là communication và chức năng không bắt buộc là: cảm biến, thực thi, thu thập dữ liệu, lưu trữ và xử lý dữ liệu.

Internet Of Things: là một cơ sở hạ tầng mang tính toàn cầu cho xã hội thông tin, mang đến những dịch vụ tiên tiến bằng cách kết nối các “Things” (cả physical lẫn virtual) dựa trên sự tồn tại của thông tin, dựa trên khả năng tương tác của các thông tin đó, và dựa trên các công nghệ truyền thông.

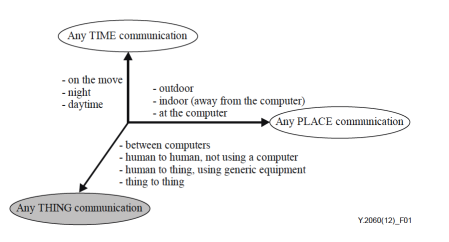
Things đối với Internet Of Things, “Thing” là một biểu tượng của thế giới vật chất (physical things) hay thế giới thông tin ảo (virtual things). “Thing” có khả năng được nhận diện, và “Thing” có thể được tích hợp vào trong mạng lưới thông tin liên lạc.

Hình 2.1

* + 1. **Khái niệm của IoT**

IoT có thể được coi là một tầm nhìn sâu rộng của công nghệ và cuộc sống. Từ quan điểm của tiêu chuần kỹ thuật, IoT có thể được xem như là một cơ sở hạ tầng mang tính toàn cầu cho xã hội thông tin, tạo điều kiện cho các dịch vụ tiên tiến thông qua sự liên kết các “Things”. IoT dự kiến sẽ tích hợp rất nhiều công nghệ mới, chằng hạn như các công nghệ thông tin machine-to-machine, mạng tự trị, khai thác dữ liệu và ra quyết định, bảo vệ sự an ninh và sự riêng tư, điện toán đám mây.

Như hình dưới, một hệ thống thông tin trước đây đã mang đến 2 chiều “Any Time” và “Any Place” communication. Giờ IoT đã tạo thêm một chiều mới trong hệ thống thông tin đó là “Any Thing” Communication.



Hình 2.2

* + 1. **IoT từ góc nhìn kỹ thuật**

Như đề cập ở mục 1, “Things” trong IoT có thể là đối tượng vật lý (Physical) hoặc là đối tượng thông tin (hay còn gọi là đối tượng ảo – Virtual). Hai loại đối tượng này có thể ánh xạ (mapping) qua lại lẫn nhau. Một đối tượng vật lý có thể được trình bày hay đại diện bởi một đối tượng thông tin, tuy nhiên một đối tượng thông tin có thể tồn tại mà không nhất thiết phải được ánh xạ từ một đối tượng vật lý nào.

Yêu cầu tối thiểu của các “divice” trong IoT là khả năng giao tiếp. Divices sẽ được phân loại vào các dạng như device mang thông tin, device thu thập dữ liệu, device cảm nhận (sensor), device thực thi, hay general device.

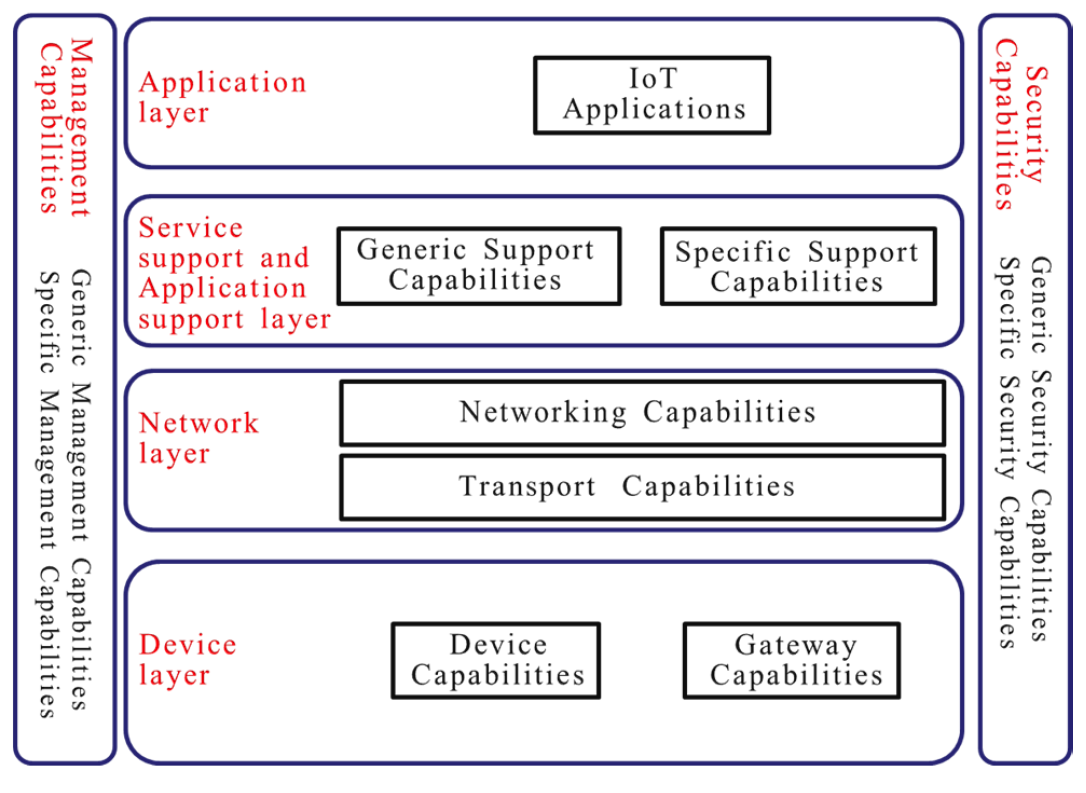
* + 1. **Đặc điểm cơ bản và yêu cầu ở mức high-level của một hệ thống IoT**
       1. **Đặc điểm cơ bản**
* Tính kết nối liên thông (interconnectivity).
* Những dịch vụ liên quan đến “Things”.
* Tính không đồng nhất.
* Thay đổi linh hoạt.
* Quy mô lớn.
  + - 1. **Yêu cầu ở mức high-level đối với một hệ thống IOT**

Một hệ thống IoT phải thoả mãn các yêu cầu sau:

* Kết nối dựa trên nhận diện.
* Khả năng cộng tác.
* Khả năng tự quản của network.
* Dịch vụ thoả thuận.
* Các khả năng dựa vào vị trí (location-based capabilities).
* Bảo mật.
* Bảo vệ tính riêng tư.
* Plug and play.
* Khả năng quản lý.
  + 1. **Mô hình của hệ thống IoT**

Bất kỳ một hệ thống IoT nào cũng được xây dựng lên từ sự kết hợp của 4 layer sau:

* Lớp ứng dụng (Application Layer)
* Lớp hỗ trợ dịch vụ và hỗ trợ ứng dụng (Service support and appliacation support layer).
* Lớp mạng (Network Layer).
* Lớp thiết bị (Device Layer).



Hình 2.3

* + 1. **Lớp ứng dụng**

Lớp ứng dụng (Application Layer) cũng như tương tự nhue trong mô hình OSI 7 lớp, lớp này tương tác trực tiếp với người dùng để cung cấp một chức năng hay một dịch vụ vụ thể của một hệ thống IOT.

* + 1. **Lớp dịch vụ và hỗ trợ ứng dụng**

Các dịch vụ hỗ trợ (Service support and application support layer) gồm hai nhóm có khả năng sau:

Nhóm dịch vụ chung: Các dịch vụ hỗ trợ chung, phổ biến mà hầu hết các ứng dụng IoT cần, ví dụ như xử lý dữ liệu hoặc lưu trữ dữ liệu.

Nhóm dịch vụ cụ thể, riêng biệt: Những ứng dụng IoT khác nhau sẽ có nhóm dịch phụ hỗ trợ khác nhau và đặc thù. Trong thực tế, ví dụ trong SmartFraming, nhóm dịch vụ cụ thể riêng biệt là tính toán độ tăng trưởng của cây mà đưa ra quyết định tưới nước hoặc bón phân.

* + 1. **Lớp mạng**

Lớp mạng (Network Layer) có 2 chức năng:

Chức năng Networking: cung cấp chức năng điều khiển các kết nối kết nối mạng, chẳng hạn như tiếp cận được nguồn tài nguyên thông tin và chuyển tài nguyên đó đến nơi cần thiết, hay chứng thực, uỷ quyền…

Chức năng Transporting: Tập trung vào việc cung cấp kết nối cho việc truyền thông tin của dịch vụ ứng dụng IoT.

* + 1. **Lớp thiết bị**

Lớp thiết bị (Divice Layer) chính là các phần cứng vật lý trong hệ thống IoT.

Divice có thể phân thành hai loại như sau:

Thiết bị thông thường: Divice này sẽ tương tác trực tiếp với network. Các thiết bị có khả năng thu thập và tải lên thông tin trực tiếp (nghĩa là không phải sử dụng gateway) và có thể trực tiếp nhận thông tin (ví dụ, lệnh) từ các network. Divice này cũng có thể tương tác gián tiếp với network. Các thiết bị có thể thu thập và tải network gián tiếp thông qua khả năng gateway. Ngược lại, các thiết bị có thể gián tiếp nhận thông tin (ví dụ, lệnh) từ network. Trong thực tế, các thiết bị thông thường bao gồm các cảm biến, các phần cứng điều khiển motor, đèn,…

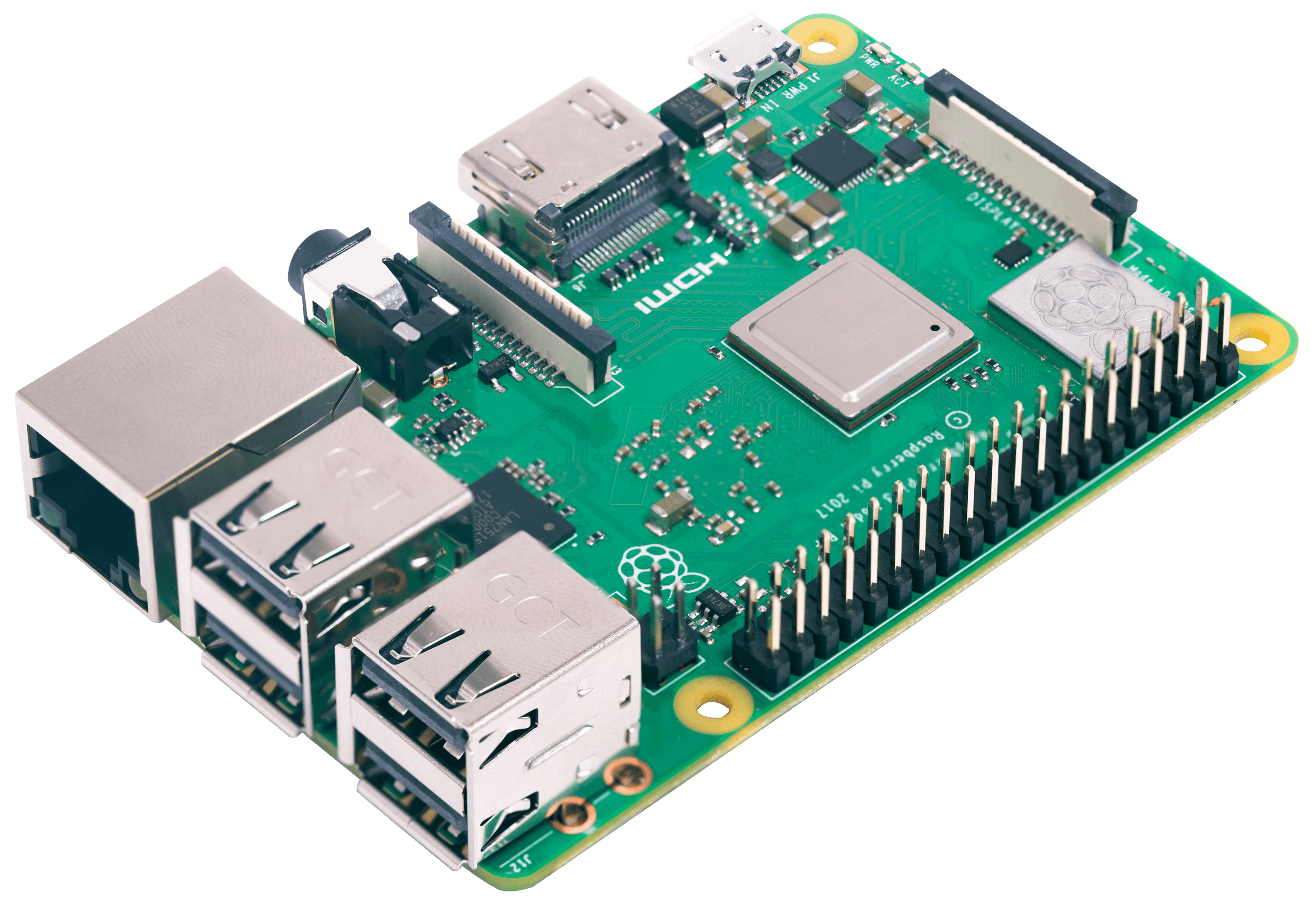
Thiết bị Gateway: Gateway là cổng liên lạc giữa divice và network. Một Gateway hỗ trợ 2 chức năng sau:

Có nhiều chuẩn giao tiếp: Vì các Things khác nhau có kiểu kết nối khác nhau, nên Gateway phải hỗ trợ đa dạng từ có dây đến không dây, chẳng hạn CAN bus, Zigbee, Bluetooth hoặc Wi-Fi. Tại Network Layer, gateway có thể giao tiếp thông qua các công nghệ khác nhau như PSTN, mạng 2G và 3G, LTE, Ethernet hay DSL.

Chức năng chuyển đổi giao thức: Chức năng này cần thiết trong hai tình huống là khi truyền thông ở lớp Divice, nhiều divice khác nhau sử dụng giao thức khác nhau, ví dụ, ZigBee với Bluetooth, và là khi truyền thông giữa các divice và Network, device dùng giao thức khác, network dùng giao thức khác, ví dụ, device dùng ZigBee còn tầng network thì lại dùng công nghệ 3G. Trong thực tế, gateway có thể được buid từ các board như Raspberry Pi hoặc Arduino, hoặc Gateway được sản xuất công nghiệp bởi các tập đoàn lớn như Intel hay Texas Instrument.

* 1. **Một số thiết bị phần cứng hỗ trợ**
     1. **Tìm hiểu về thiết bị Raspberry Pi**
        1. **Giới thiệu chung**

Raspberry Pi là một chiếc máy tính tí hon chạy hệ điều hành Linux ra mắt vào tháng 2 năm 2012 với giá chỉ $25. Ban đầu Raspberry Pi được phát triển dựa trên ý tưởng tiến sĩ Eben Upton tại đại học Cambridge muốn tạo ra một chiếc máy tính giá rẻ để học sinh có thể dễ dàng tiếp cận và khám phá thế giới tin học. Dự định khiêm tốn của ông đến cuối đời là có thể bán được tổng cộng 1000 bo mạch cho các trường học. Tuy nhiên với những ưu điểm nổi bật, hơn một triệu board Raspberry Pi đã được bán ra chỉ trong vòng chưa đầy một năm.



Hình 2.4

Chỉ cần 1 bàn phím, 1 tivi hoặc 1 màn hình có cổng HDMI/DVI, 1 nguồn USB 5V và 1 dây micro USB là đã có thể sử dụng Raspberry Pi như 1 máy tính bình thường. Với Raspberry Pi, ta có thể sử dụng các ứng dụng văn phòng, nghe nhạc, xem phim độ nét cao... Một điều quan trọng là nó rất tiết kiệm điện và khả năng chạy liên tục 24/24.

Phần cứng

Thông số kỹ thuật chi tiết:

|  |  |
| --- | --- |
| Vi xử lý | Broadcom BCM2837B0, quad-core A53 (ARMv8) 64-bit SoC @1.4GHz |
| RAM | 1GB LPDDR2 SDRAM |
| Kết nối | 2.4GHz and 5GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 4.2, BLE, Gigabit Ethernet over USB 2.0 (Tối đa 300Mbps) |
| Cổng USB | 4 x 2.0 |
| Mở rộng | 40-pin GPIO |
| Video và âm thanh | 1 cổng full-sized HDMI, Cổng MIPI DSI Display, cổng MIPI CSI Camera, cổng stereo output và composite video 4 chân |
| Multimedia | H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics |
| Lưu trữ | MicroSD |
| Nguồn điện sử dụng | 5V/2.5A DC cổng microUSB, 5V DC trên chân GPIO, Power over Ethernet (PoE)  (yêu cầu thêm PoE HAT) |

Raspberry Pi 3 Model B+ là sản phẩm mới nhất trong gia đình Raspberry Pi, nổi bật với chip 4 nhân 64-bit có tốc độ 1.4GHz – nhanh nhất từ trước đến nay! Phiên bản mới còn hỗ trợ Wifi Dual-band 2.4GHz và 5GHz, Bluetooth 4.2/Bluetooth Low Energy, cổng Ethernet tốc độ cao (300Mbps) và Power over Ethernet (PoE) thông qua PoE HAT.

Low Energy, cổng Ethernet tốc độ cao (300Mbps) và Power over Ethernet (PoE) thông qua PoE HAT.

Trái tim của Raspberry Pi là chip SoC (System-On-Chip) Broadcom BCM2835 chạy ở tốc độ 700mHz. Chip này tương đương với nhiều loại được sử dụng trong smartphone phổ thông hiện nay, và có thể chạy hệ điều hành Linux.

Tích hợp trên chip này là nhân đồ hoạ (GPU) Broadcom VideoCore IV. CPU này đủ mạnh để có thể chơi một số game phổ thông và phát video chuẩn full HD.

Hệ thống GPIO (General Purpose Input Output): gồm 26 chân chia làm hai hàng. Đúng như tên gọi của nó, từ đây chúng ta có thể kết nối và điều khiển rất nhiều thiết bị điện tử/cơ khí khác.

Ngõ HDMI: dùng để kết nối Pi với màn hình máy tính hay tivi có hỗ trợ cổng HDMI.

Ngõ RCA Video (analog): Khi thiết kế Pi người ta cũng tính đến trường hợp người sử dụng ở các nước đang phát triển không có điều kiện sắm một chiếc tivi đời mới tích hợp cổng HDMI. Vì vậy cổng video analog này được thêm vào, giúp Raspberry Pi có thể kết nối với tivi đời cũ.

Ngõ audio 3.5mm: Kết nối dễ dàng với loa ngoài hay headphone. Đối với tivi có cổng HDMI, ngõ âm thanh được tích hợp theo đường tín hiệu HDMI nên không cẩn sử dụng ngõ audio này.

Cổng CSI: Khe cắm này để cắm module camera vào Raspberry Pi. Khi sản xuất Raspberry Pi thì nhà sản xuất còn sản xuất thêm một module camera 5MP nhưng người mua không được hỗ trợ mà phải mua thêm. Chúng ta có thể chụp hình, quay phim,… làm việc tất cả các tác vụ như trên một camera bình thường.

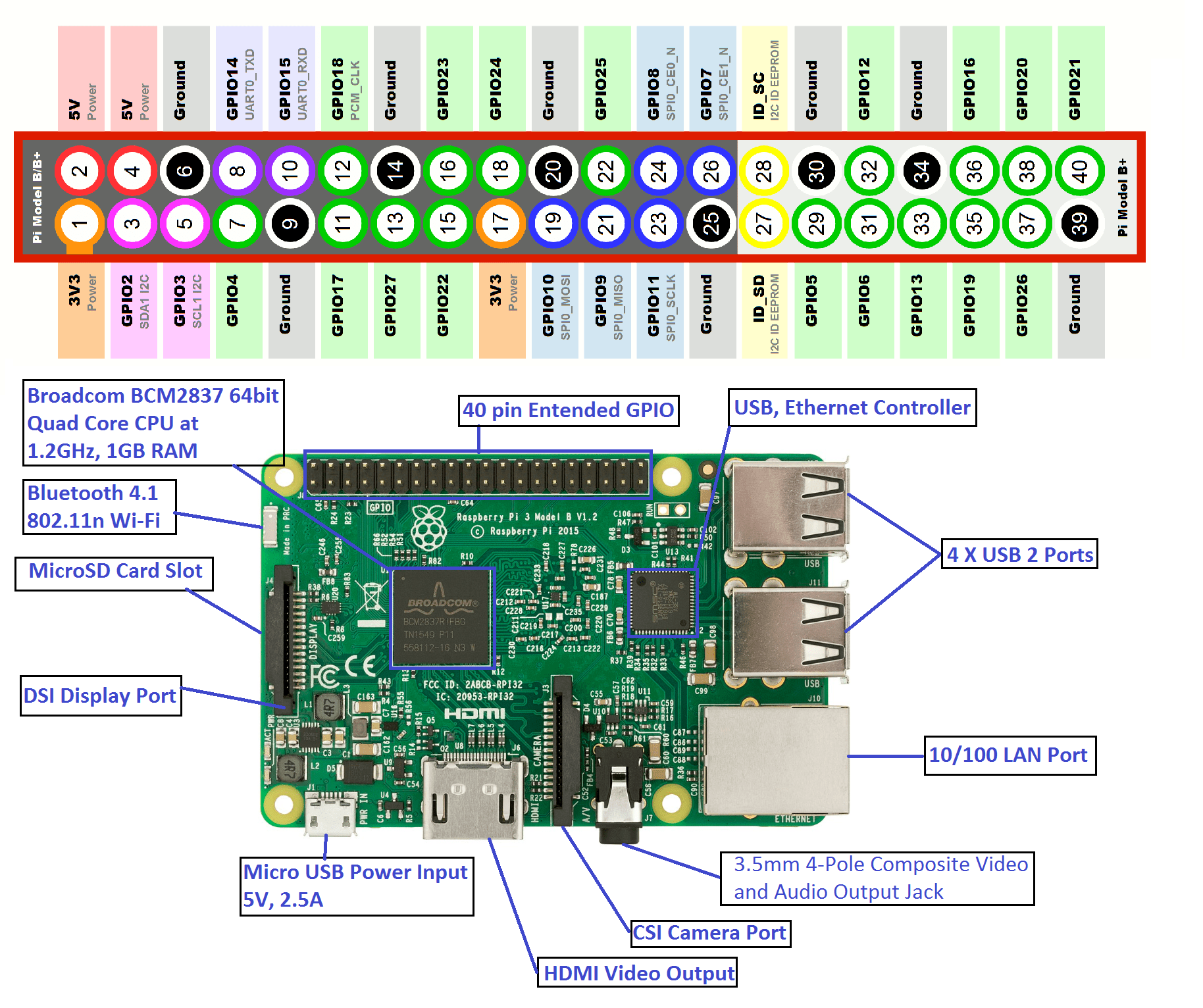
Cổng DSI: Nơi đây sẽ giúp ta có thể kết nối Raspberry Pi với màn hình cảm ứng để hiển thị và sử dụng Raspberry một cách trực quan nhất. Chúng ta có thể thực hiện các tác vụ tương đương như khi sử dụng chuột và bàn phím.

Cổng USB: Một điểm mạnh nữa của Raspberry Pi là tích hợp 2 cổng USB 2.0. Ta có thể kết nối tới bàn phím, chuột hay webcam, bộ thu GPS… qua đó có thể mở rộng phạm vi ứng dụng. Vì Raspberry Pi chạy Linux nên hầu hết các thiết bị chỉ cần cắm và chạy (Plug-and-Play) mà không cần cài driver phức tạp.

Cổng Ethernet: cho phép kết nối Internet dễ dàng. Cắm dây mạng vào Pi, kết nối với màn hình máy tính hay tivi và bàn phím, chuột là có thể lướt web dễ dàng.

Khe cắm thẻ SD: Raspberry Pi không tích hợp ổ cứng. Thay vào đó nó dùng thẻ SD để lưu trữ dữ liệu. Toàn bộ hệ điều hành Linux sẽ hoạt động trên thẻ SD này vì vậy nó cần kích thước thẻ nhớ tối thiểu 4 GB và dung lượng hỗ trợ tối đa là 32 GB.

Sơ đồ chân của thiết bị Raspberry Pi 3:



Hình 2.5

Cấu trúc phần mềm

Các Raspberry Pi sử dụng hệ điều hành dựa trên nền tàng Linux. Phần cứng CPU được truy cập thông qua Image Firmware được nạp vào CPU vào lúc khởi động từ thẻ SD. Image Firmware được gọi là đốm màu nhị phân (Binary Blob), trong khi ARM liên kết với mã trình điều khiển Linux ban đầu được dựa vào nguồn đóng. Một phần của mã điều khiển đã được giải phóng, tuy nhiên nhiều chương trình điều khiển thực tế được thực hiện bằng cách sử dụng mã nguồn đóng GPU. Phần mềm ứng dụng sử dụng các cuộc gọi đến thư viện thời gian chạy nguồn đóng (OpenMax, OpenGL ES hay OpenVG). Nó sẽ gọi một trình điều khiển nguồn mở bên trong lõi Linux, sau đó gọi mã điều khiển nguồn đóng GPU VideoCore IV. Các API của trình điều khiển lõi là cụ thể cho những thư viện đóng. Các ứng dụng Video sử dụng OpenMax, ứng dụng 3D sử dụng OpenGL ES và ứng dụng 2D sử dụng OpenVG và cả hai lần lượt sử dụng EGL. OpenMax và EGL sử dụng trình điều khiển nền tảng mã nguồn mở.



Hình 2.6

Nhà sản xuất Raspberry sẽ cung cấp một tập hợp các thư viện mã nguồn đóng cho phép chúng ta truy cập vào các tính năng tăng tốc GPU. Các thư viện sẽ có sẵn là:

⮚ OpenGL ES 2.0 (opengl) là một thư viện 3D, rất thường được sử dụng trên máy tính để bàn và các hệ thống nhúng. Nó được định nghĩa bởi Khronos Group.

⮚ OpenVG là một thư viện bản vẽ véc tơ 2D, cũng thường được sử dụng trên máy tính để bàn và các hệ thống nhúng. Một lần nữa, được định nghĩa bởi Khronos Group.

⮚ EGL là một giao diện lập trình ứng dụng giữa Khronos và API như OpenGL ES hay OpenVG và hệ thống cửa sổ nền tảng nguồn gốc cơ bản.

⮚ Openmax cung cấp một tập hợp các API với khái niệm trừu tượng của người dùng cho những thói quen sử dụng trong âm thanh, video, và xử lý hình ảnh tĩnh. OpenMax định nghĩa ba lớp, đây là lớp IL, cung cấp một giao diện giữa các khuôn khổ đa phương tiện như Gstreamer và một tập hợp các thành phần đa phương tiện (như bảng mã).

⮚ Openmax IL không có một API chuẩn ở giai đoạn này, vì vậy đó là một cài đặt tùy chỉnh. Tất cả các thư viện này được cung cấp bởi chip SoC Broadcom.

* + - 1. Tìm hiểu hệ điều hành

Raspberry Pi là một máy tính, để máy tính này hoạt động cần cài đặt hệ điều hành. Trong thế giới nguồn mở linux, có rất nhiều phiên bản hệ điều hành tùy biến (distro) khác nhau. Tùy theo nhu cầu và mục đích, cũng như khả năng học hỏi mà ta sẽ sử dụng distro phù hợp với mình.

Có 5 phiên bản hệ điều hành được cung cấp chính thức cho Raspberry Pi:

⮚ Raspian "wheezy": đây là distro dựa trên Debian wheezy, sử dụng hard-float ABI (tính toán dấu chấm động bằng phần cứng) cho thời gian chạy các ứng dụng nhanh hơn. Có sẵn giao diện đồ họa. Phù hợp với người mới bắt đầu tiếp cận Linux vì tính dễ sử dụng và trực quan.

⮚ Soft-float "wheezy": vẫn được xây dựng dựa trên Debian wheezy nhưng việc xử lý dấu chấm động được thực hiện bằng phần mềm. Việc này giúp có thể sử dụng máy ảo Java (Oracle JVM) trên Raspberry.

⮚ Arch Linux: phiên bản giành cho ARM. Đảm bảo thời gian khởi động trong vòng 10 giây. Chỉ khởi động và load các gói cần thiết. Để sử dụng được Arch Linux cần có kiến thức cơ bản về Linux.

⮚ Pidora: là phiên bản của Fedora được tối ưu cho Raspberry Pi, có sẵn giao diện đồ họa. Giành cho những ai đã quen xài Fedora.

⮚ RISC OS: là hệ điều hành do nhóm phát triển ARM thiết kế riêng. Đây

không phải là một phiên bản Linux, do vậy cần làm quen với cấu trúc và câu lệnh đặc trưng cho hệ điều hành này.

Xây dựng mạng cảm biến giám sát các thông số môi trường qua việc sử dụng thiết bị Raspberry Pi và các Sensor. Chúng ta sẽ xây dựng một ứng dụng để giám sát các thông số môi trường tiêu biểu đối với việc giám sát quá trình chuyển động.

* + 1. Các loại cảm biến
       1. Cảm biến chuyển động

Ứng dụng:

* Dùng trong cho an ninh: phát hiện người trong khoảng cách cho phép.
* Dùng để điều khiển tự động các thiết bị trong nhà khi có người: đèn điện, cửa…

Giới thiệu về HC – SR501

Điện áp hoạt động : DC  4.5V – 20V.

Đầu ra : High 3.3V/ Low 0V.

Có 2 chế độ hoạt động:

* L không lặp lại kích hoạt.
* H lặp lại kích hoạt.

Thời gian trễ 5 – 200s có thể điều chỉnh từ 0,xx đến hàng chục giây.

Thời gian khoá: 2.5s (mặc định).

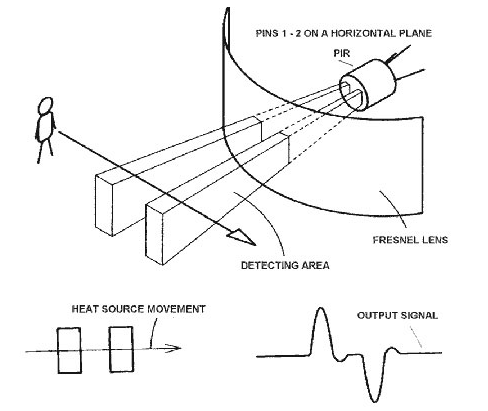
Nhiệt độ hoạt động: -15oC – 70oC.

Kích thước broad: 32mm \* 24mm.

Khối lượng: 6g.

Nguyên tắc hoạt động

Cơ chế hoạt động của cảm biến hồng ngoại PIR: là cảm biến thu tia hồng ngoại được phát ra từ các vật thể phát ra tia hồng ngoại như thân thể con người (hay nguồn nhiệt bất kì).



Hình 2.7

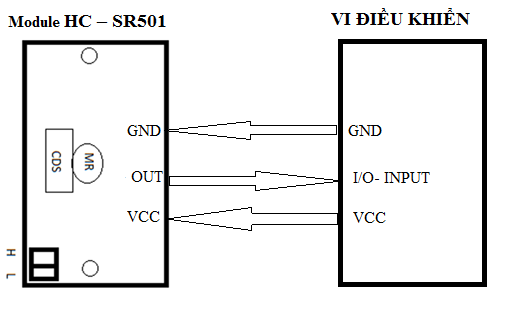
Các cảm biến PIR luôn có sensor (mắt cảm biến) với 2 đơn vị (element). Chắn trước mắt sensor là một lăng kính (thường làm bằng plastic), chế tạo theo kiểu lăng kính fresnel.

Lăng kính fresnel này có tác dụng chặn lại và phân thành nhiều vùng (zone) cho phép tia hồng ngoại đi vào mắt sensor. 2 đơn vị của mắt sensor có tác dụng phân thành 2 điện cực.

Một cái là điện cực dương (+) và cái kia là âm (-). Khi 2 đơn vị này được tuần tự kích hoạt (cái này xong rồi mới đến cái kia) thì sẽ sinh ra một xung điện, xung điện này kích hoạt sensor.

Kiểm soát ánh sáng (tùy chọn): Bạn có thể lắp thêm quang trở, khi có quang trở, sẽ thiết lặp module hoạt động ban ngày hoặc ban đêm.

Sơ đồ kết nối



Hình 2.8

Chân VCC: nguồn hoạt động của cảm biến cấp vào từ 4.5V đến 20V.

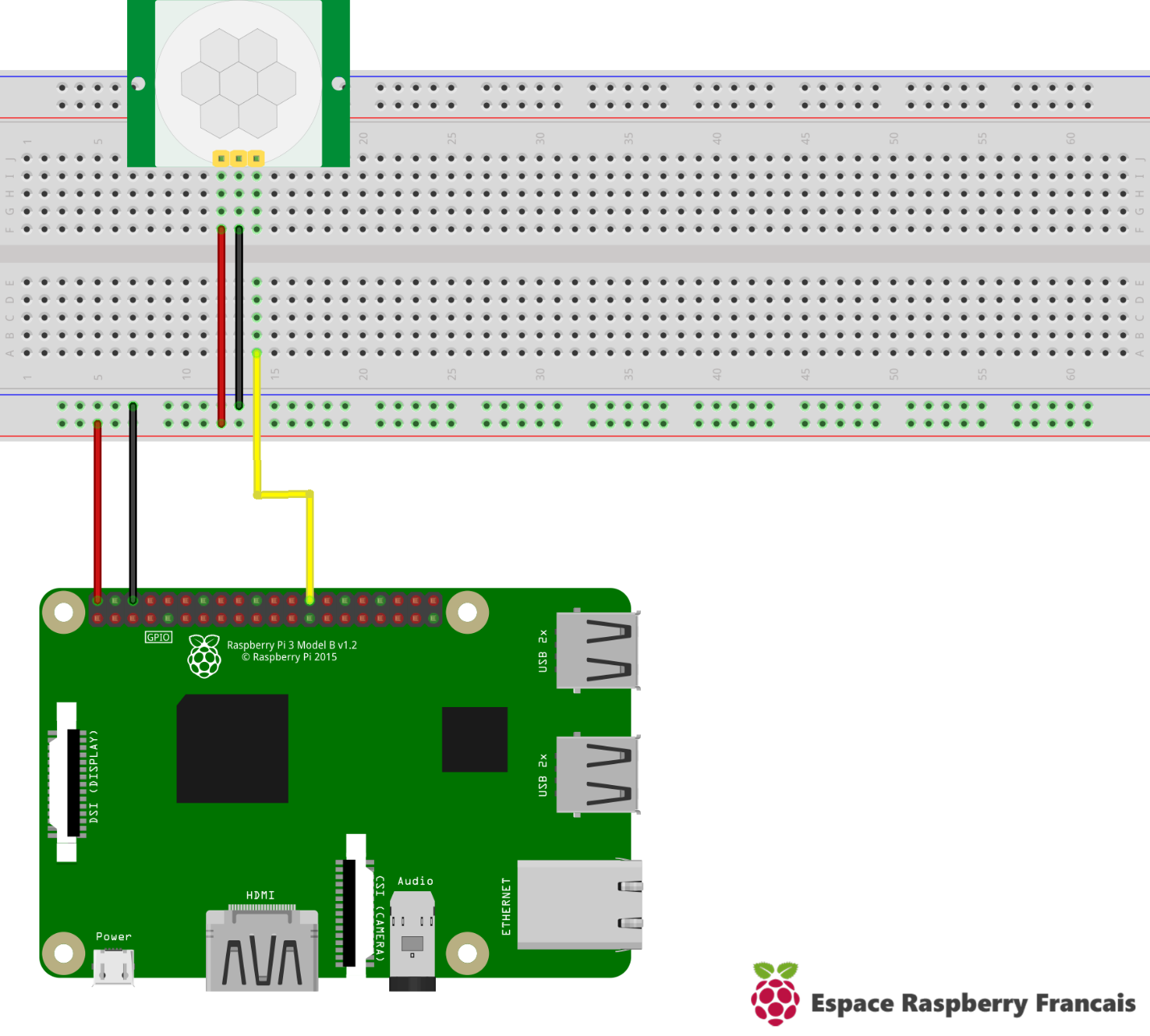
Chân OUT: Output kết nối với chân I/O của vi điều khiển hoặc relay. Khi cho tín hiệu:

* 3.3V có vật thể chuyển động qua.
* 0V không có vật thể qua.

Chân GND: chân đất nối GND.

Chế độ H: Điện áp ra V\_out tự động giữ nguyên 3.3V cho đến khi không còn chuyển động.

Chế độ L: Điện áp ra V\_out tự động chuyển về 0 khi hết thời gian trễ.



Hình 2.9

Lưu ý sử dụng:

Cài đặt: Khi khởi tạo, module cần thời gian khởi tạo khoảng 1 phút. Trong thời gian này, moudle tạo ra điện áp cao từ 1-3 lần sau đó vào chế độ chờ.

Điện áp ra 1.5-3.3V, nếu sử dụng I/O 4.5-5.5V bạn nên lắp thêm chân.

Nên cố gắng tránh ánh sáng trực tiếp và nguồn nhiễu gần với bề mặt lăng kính của các module, để tránh đưa ra tín hiệu nhiễu, tránh sử dụng môi trường nhiều gió.

* + - 1. **Cảm biến âm thanh**

Cảm biến âm thanh giúp nhận biết và phát hiện cường độ âm thanh của môi trường xung quanh. Thích hợp để làm các ứng dụng cơ bản về nhận biết, giám sát các tiếng động của môi trường xung quanh, và nhiều ứng dụng thú vị khác. Độ nhạy có thể được điều chỉnh bằng chiết áp trên module để phù hợp với từng nhu cầu ứng dụng.

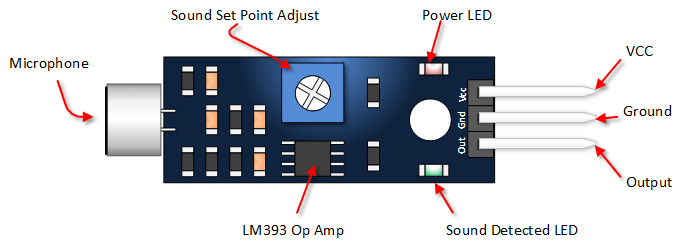
Nó sử dụng một microphone, mạch khuếch đại tín hiệu, máy dò biên độ và bộ đệm. Khi phát hiện âm thanh nó sẽ qua mạch khuếch đại để khuếch đại tín hiệu rồi gửi tín hiệu digital qua chân “OUT”.

Thông số kỹ thuật

Điện áp hoạt động: DC 4-6V

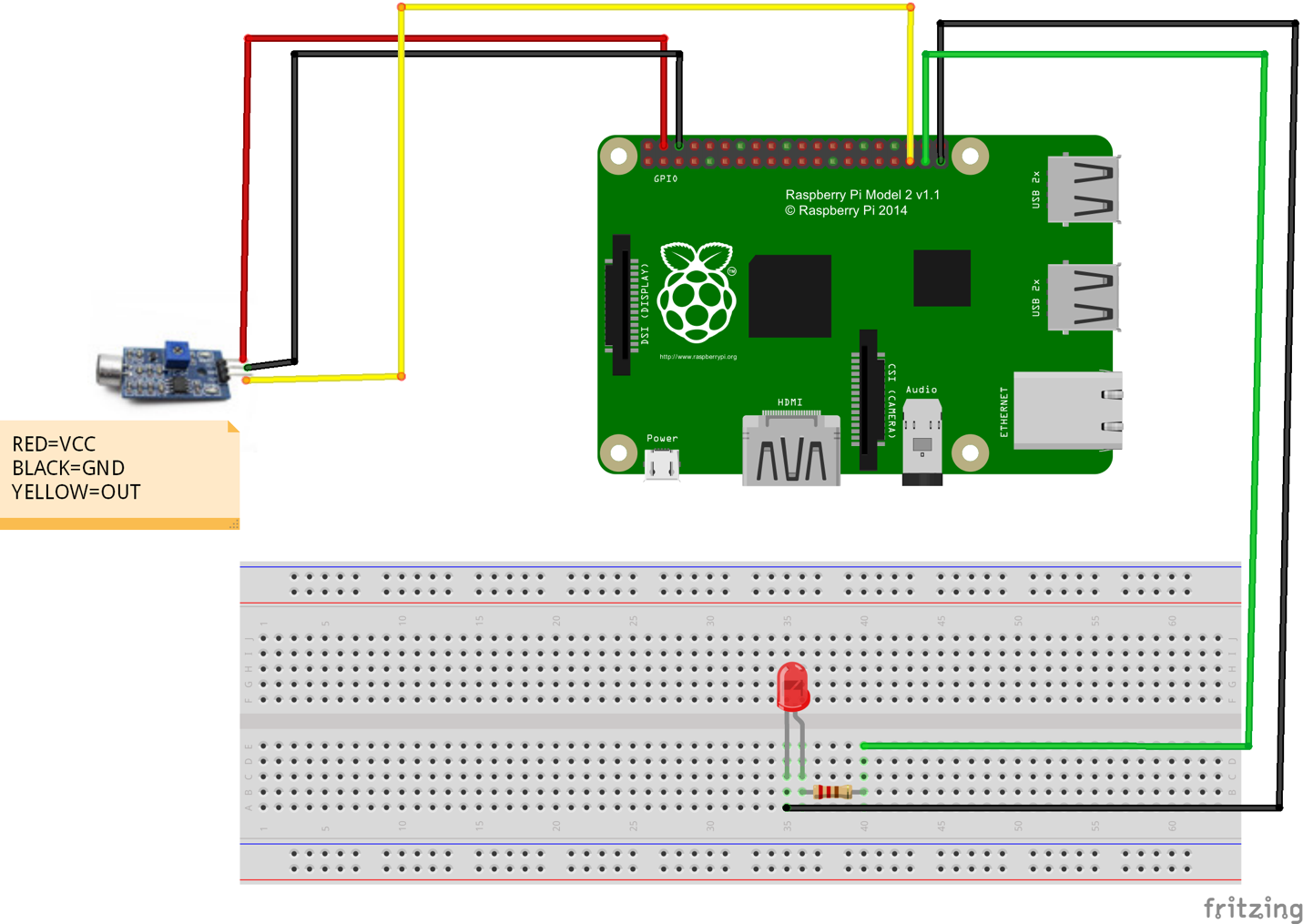
• Xuất ra kiểu tín hiệu: digital (0 hoặc 1)

• Kích thước: 38mm \* 16mm \* 9mm



Hình 2.10

|  |  |
| --- | --- |
| VCC | Nguồn 4V đến 6V |
| GND | Mass |
| OUT | Đầu ra tín hiệu số (mức cao hoặc mức thấp) |



Hình 2.11

* + - 1. **Thiết bị chuyển động điều khiển đồ chơi trẻ em**

Động cơ RC Servo 9G có kích thước nhỏ, là loại được sử dụng nhiều nhất để làm các mô hình nhỏ hoặc các cơ cấu kéo không cần đến lực nặng. Động cơ RC Servo 9G có tốc độ phản ứng nhanh, các bánh răng được làm bằng nhựa nên cần lưu ý khi nâng tải nặng vì có thể làm hư bánh răng, động cơ RC Servo 9G có tích hợp sẵn Driver điều khiển động cơ bên trong nên có thể dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.

Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kì từ 0o -180o. Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô mình), có loại thì sở hữu một momen lực bá đạo (vài chục Newton/m), hoặc có loại thì khỏe và nhông sắc chắc chắn,…

Động cơ servo được thiết kế những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bầt kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác. Các động cơ servo điều khiển bằng liên lạc vô tuyến được gọi là động cơ servo RC (radio-controlled). Trong thực tế, bản thân động cơ servo không phải được điều khiển bằng vô tuyến, nó chỉ nối với máy thu vô tuyến trên máy bay hay xe hơi. Động cơ servo nhận tín hiệu từ máy thu này.

**Thông số kỹ thuật:**

• Điện áp hoạt động: 4.8 – 5VDC.

• Tốc độ: 0.12 sec/60 degrees (4.8VDC).

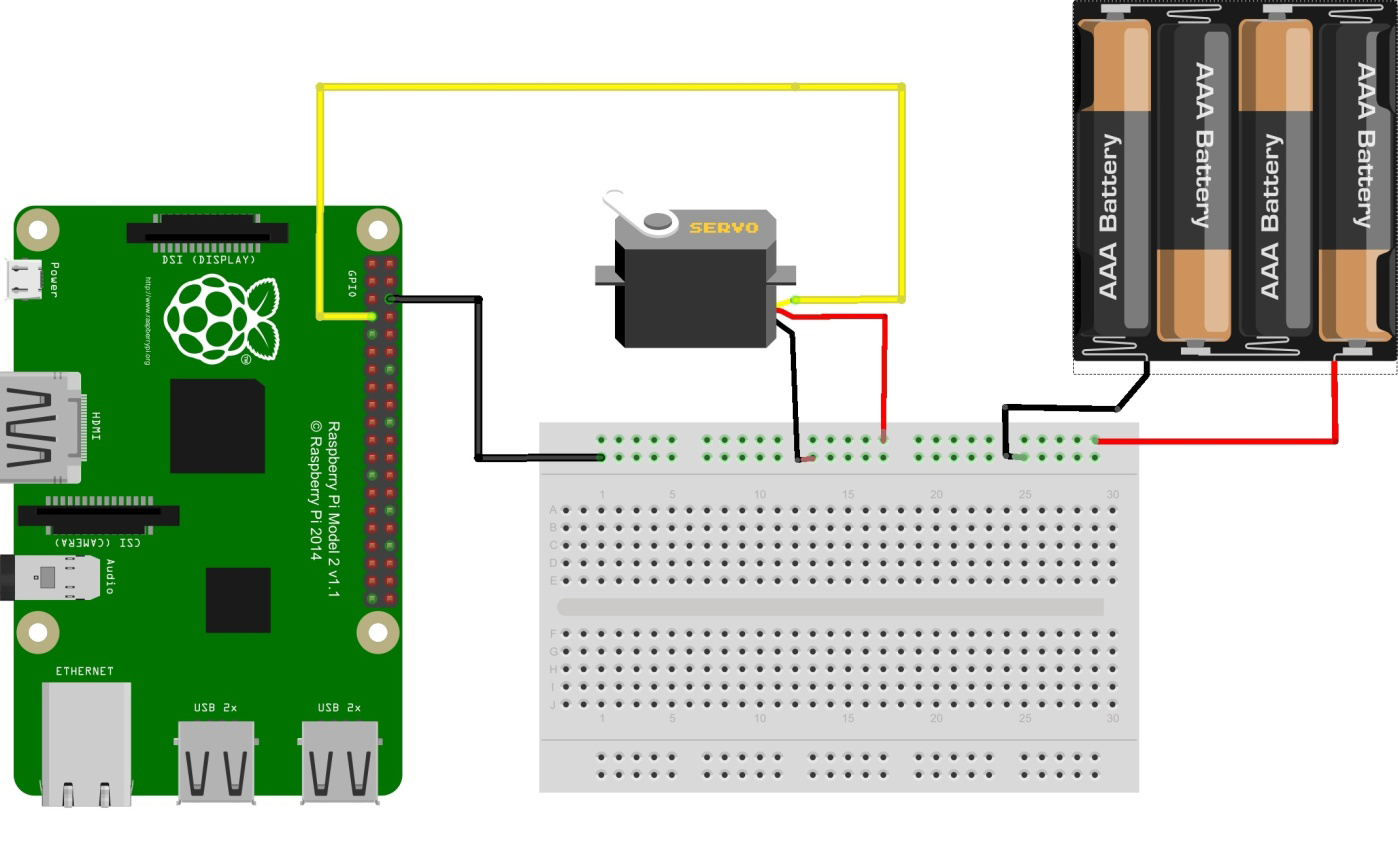
• Lực kéo: 1.6KG.CM.

• Kích thước: 21x12x22mm.

• Trọng lượng: 9g.



Hình 2.12



Hình 2.13

* + - 1. **Camera Raspberry**

Camera Raspberry Pi V1 5MP là Version đầu tiên của module camera cho Raspberry Pi với cảm biến OV5647 độ phân giải 5MP, sử dụng tương thích với tất cả các dòng Raspberry Pi từ trước đến nay, chất lượng hình ảnh tốt, độ phân giải cao và có khả năng quay phim ở chất lượng HD.

Để sử dụng Module Camera cho Raspberry Pi tốt và bền Hshop.vn có tặng cho quý khách khi mua hàng vỏ Case bằng Mica để chống tĩnh điện từ tay người làm hư cảm và bảo vệ mạch.

**Thông số kỹ thuật:**

• Module Camera V1 cho Raspberry Pi.

• Cảm biến: OV5647.

• Độ phân giải: 5MP.

• Độ phân giải màn hình: 2592 x 1944 pixel.

• Quay phim HD 1080P 30, 720P 60, VGA 640x480P 60.

• Lens: Fixed conector.

• Kích thước: 25x24x9mm



Hình 2.14

* + - 1. **Loa phát nhạc**

Lựa chọn loa có jack cắm chuẩn 3.5mm có thể tương thích với các loại desktop, laptop, smartphone, máy tính bảng, máy nghe nhạc…

Loa được tích hợp sẵn volume thuận tiện sử dụng, lấy nguồn điện trực tiếp từ cổng usb máy tính hoặc các bộ adapter sạc usb. Cách sử dụng vô cùng đơn giản, chỉ cần cắm cáp vào là sử dụng ngay mà không cần cài đặt gì.

* Tích hợp với máy tính để bàn, laptop, điện thoại, tai nghe, usb
* Giao tiếp qua cổng USB phổ biến và jack cắm tiêu chuẩn 3.5mm giúp bạn dễ dàng kết nối với nhiều dòng máy nghe nhạc có cổng tương thích.



Hình 2.15

* 1. **Tìm hiểu về các ngôn ngữ lập trình**
     1. **Ngôn ngữ lập trình ASP.NET**
        1. **Giới thiệu về ASP.NET**

Ta phải công nhận một điều là .NET Framework và các ứng dụng của nó đã và đang tạo một cuộc cách mạng kỹ thuật trong các công nghệ tin học (Information Technology), thay đổi tận gốc rễ các kiểu mẫu lập trình hay phét triển và triển khai mạng trên thế giới và do đó tạo một vận hội mới đáp ứng mọi yêu cầu khẩn thiết cho các ngành nghề kỹ thuật và thương mại hiện nay cũng như vạch một hướng đi vững chắc và lâu dài cho tương lai tin học. ASP.NET chính là một trong những ứng dụng quan trọng nhất để phát triển và triển khai mạng một cách dễ dàng chưa từng thấy từ xưa đến nay. Hãy lắng nghe chính Microsoft nói về ASP.NET như thế nào:

“ASP.NET” là một framework lập trình mang tính cách mạng cho phép phát triển nhanh chóng các ứng dụng và dịch vụ web mạnh mẽ. Là một phần của Microsoft .NET Platform, nó cung cấp cách dễ nhất và dễ mở rộng nhất để phát triển, triển khai và chạy các ứng dụng web phân tán có thể nhắm mục tiêu bất kỳ trình duyệt hoặc ứng dụng nào.

* Easy Programming Model: ASP.NET giúp chúng ta phát triển và triển khai các ứng dụng về mạng trong một thời gian kỉ lục vì nó cung cấp cho ta một kiểu mẫu lập trình dễ dàng và gọn gàng nhất. Ngoài ra, các trang ASP.NET còn làm việc với mọi trình duyệt hiện nay như Chrome, CocCoc, Firefox, Internet Explorer (IE), Netscape, Opera, AOL…mà không cần đổi tới đổi lui các nguồn mã vật vã như trước.
* Flexible Language Options: Không như ASP kiểu cổ điển chỉ giới hạn với VBScripts and Java Scripts, ASP.NET yểm trợ trên 25 ngôn ngữ lập trình (dĩ nhiên ngoài các ngôn ngữ mới thiết lập đã cài sẵn yểm trợ .NET framework như là VB.NET, C# và JScript.NET, ngoài ra còn có Perl.NET, Component Pascal.NET, Mercury.NET, Oberon.NET, Python.NET…).
* Great Tool Support: Mặc dù ta có thể chỉ dùng đến Notepad để triển khai các trang ASP.NET nhưng Visual Studio.NET giúp năng suất triển khai mạng thêm phần hiệu quả vì ta có thể quan sát các kế hoạch của ta dễ dàng hơn khi phác họa các thành phần của ASP.NET bằng hình ảnh với ASP.NET Web Form Shay Services theo phương pháp “drag-drop-double-click” quen thuộc của nền Windows. Hơn nữa, còn yểm trợ ta trong việc phát hiện và loại bỏ các lỗi sai một cách rất thuận lợi trong khi phát triển các ứng dụng về mạng (support for debugging and deploying ASP.NET Web applications).
* Rich Class Framework: Nhờ nền tảng vững vàng và tài nguyên phong phú .NET Framework với hơn 5000 classes bao gồm đủ thứ như XML, data access, file upload, regular expressions, transactions, message queuing, SMTP mail, … nên việc thiết kế các đặc tính trong một ứng dụng trở nên nhẹ nhàng và thoải mái hơn xưa rất nhiều.
* Compile execution: ASP.NET không những chạy nhanh hơn ASP cổ điển gấp 5 lần mà còn có thể duy trì kiểu mẫu cập nhật gọi là kiểu mẫu “just hit save”, nghĩa là ASP.NET tự động dò tìm mọi sự thay đổi và biên dịch khi cần thiết cũng như lưu trữ kết quả compile đó để cung ứng dịch vụ cho những yêu cầu tiếp theo sau, nhờ vậy ứng dụng của bạn luôn được cập nhật hóa và chạy với tốc độ nhanh.
* Rich output caching: ASP.NET có khả năng lưu trữ một kết quả chung trong bộ phận memory của trang để gửi giải đáp cho cùng một yêu cầu từ nhiều khách hàng khác nhau và nhờ đó không những tiết kiệm được sự lặp đi lặp lại công tác thi hành của một trang web mà còn gia tăng hiệu suất một cách ngoạn mục do giới hạn tối đa việc chất vấn các cơ sở dữ liệu (eliminating the need to query the database on every request) rất tốn nhiều thời gian.
* .NET Outperforms J2EE: Trong việc đối đầu nhau về hiệu suất (performance) và scalability với cùng một ứng dụng phát triển giữa Sun's Java Pet Store J2EE và ASP.NET thì ASP.NET không những nhanh vượt trội hơn J2EE đến 28 lần.
* Memory Leak, Dead Lock và Crash protection: ASP.NET cũng có khả năng dò tìm và phục hồi (detects and recovers) những trở ngại nghiêm trọng như deadlocks hay bộ nhớ bị rỉ để đảm bảo ứng dụng của bạn luôn sẵn sàng khi dùng mà không làm cản trở việc cung ứng dịch vụ cần thiết thường lệ.
* Simple application deployment: ASP.NET đơn giản hóa việc triển khai ứng dụng mạng, do đó biến việc triển khai toàn bộ ứng dụng trở nên dễ dàng và thuận tiện hơn hẳn trước kia vì bây giờ ta chỉ cần sao (với XCOPY) và lưu trữ ở server chứ không cần phải chạy chương trình “regsrv32” để đăng ký bất cứ thành phần nào cả, và thêm nữa khi cần lưu trữ những yếu tố phụ cần thiết cho việc thiết lập hay bố trí các ứng dụng, ta chỉ cần lưu trữ nó vào trong một tệp dưới dạng XML là đủ.
* Dynamic update of running application: ASP.NET cho phép ta tự động cập nhật hoá các thành phần đã compiled (compiled components) mà không cần phải khởi động lại các Web server.
* Easy Migration Path: Ta không cần phải du nhập những ứng dụng được phát triển và triển khai bằng ASP cổ điển hiện có vào ASP.NET vì ASP.NET có thể chạy song song với ASP ở cùng một Internet Information Server (IIS) trong nền windows 2000 hay windows XP. Các ứng dụng cũ vẫn được chạy hết sức thoải mái với ASP.DLL trong khi ASP.NET sẽ xử lý các ứng dụng mới. Ngoài ra, ASP.NET còn cho phép bạn dùng lại những thành phần thương mại hiện nay kiểu COM cổ điển trong các ứng dụng của nó.
* XML Web Services: Dịch vụ về mạng với XML cũng cho phép bạn truyền đạt (communicate) và chia sẻ các dữ kiện xuyên qua mạng Internet dễ dàng hơn các SOAP client mà không hề phân biệt đối xử các hệ điều hành hay ngôn ngữ lập trình khác nhau (regardless of OS or programming language). Nhờ đó, ta không cần phải học thêm hay đào sâu các kiến thức về Networking, XML hay SOAP, ...
* Mobile Web Service Support: Thêm nữa, ASP.NET Mobile Controls còn giúp ta phát triển và triển khai mạng nhắm vào thị trường những cell phone hay PDA với gần hơn 80 Mobile Web Services được cung cấp trong .NET framework. Bạn chỉ cần lập trình cho ứng dụng của bạn rồi phó mặc cho Mobile Controls đó tự động phát sinh ra những nguồn mã WAP/WML, HTML hay iMode thích hợp với từng loại thiết bị riêng biệt.
  + - 1. **ASP là gì**

Trước hết, họ tên của ASP.NET là Active Server Pages .NET (.NET dây là .NET framework). Nói đơn giản và ngắn gọn thì ASP.NET là một công nghệ có tính cách mạng dùng để phát triển các ứng dụng về mạng hiện nay cũng như trong tương lai (ASP.NET is a revolutionary technology for developing web applications). Lưu ý ở chỗ ASP.NET là một phương pháp tổ chức hay khung tổ chức (framework) để thiết lập các ứng dụng hết sức hùng mạnh cho mạng dựa trên CLR (Common Language Runtime) chứ không phải là một ngôn ngữ lập trình. Ngôn ngữ lập trình được dùng để diễn đạt ASP.NET là VB.NET (Visual Basic .NET) và VB.NET chỉ là một trong 25 ngôn ngữ .NET hiện nay được dùng để phát triển các trang ASP.NET mà thôi.

Tuy mang họ tên gần giống như ASP cổ điển nhưng ASP.NET không phải là ASP. Ta sơ lược ở đây một vài khác biệt giữa ASP.NET và ASP để ta có một khái niệm tổng quát.

Sự khác nhau giữa ASP.NET và ASP

ASP.NET được phác thảo lại từ số không, nó được thay đổi tận gốc rễ và phát triển phù hợp với yêu cầu hiện nay cũng như vạch một hướng đi vững chắc cho tương lai tin học. Lý do chính là vì Microsoft đã quá chán nản trong việc thêm thắt và kết hợp những công cụ mới vào các kiểu mẫu lập trình hay thiết kế mạng theo kiểu cổ điển nên Microsoft nghĩ rằng tốt nhất là nên làm một kiểu mẫu hoàn toàn mới thay vì vá víu chỗ này chỗ nọ vào ASP. Đó là chưa kể tới nhiều phát minh mới ra đời sau này dựa trên những khái niệm mới mẻ theo xu hướng phát triển hiện nay của công nghệ tin học (Information Technology) cần được đưa vào kiểu mẫu phát triển mới đó. ASP.NET cung cấp một phương pháp hoàn toàn khác biệt với ASP.

* + 1. **Restful API sử dụng python flask**

Thiết kế REST hay RESTful API (State Representational State Transfer) được thiết kế để tận dụng các giao thức hiện có. Mặc dù REST có thể được sử dụng trên hầu hết các giao thức, nhưng thường sử dụng HTTP khi được sử dụng cho các API Web. Điều này có nghĩa là các nhà phát triển không cần cài đặt thư viện hoặc phần mềm bổ sung để tận dụng thiết kế REST API. Thiết kế API REST được định nghĩa bởi Tiến sĩ Roy Fielding trong luận án tiến sĩ năm 2000 của ông. Điều đáng chú ý là lớp linh hoạt không thể tin được của nó. Vì dữ liệu không gắn với các phương pháp và tài nguyên, REST có khả năng xử lý nhiều loại cuộc gọi, trả lại các định dạng dữ liệu khác nhau và thậm chí thay đổi cấu trúc với việc thực hiện đúng của hypermedia.

Sự tự do và sự linh hoạt của nó trong thiết kế API REST cho phép bạn xây dựng một API đáp ứng nhu cầu của bạn đồng thời đáp ứng được nhu cầu của rất nhiều khách hàng khác nhau. Không giống như SOAP, REST không bị giới hạn bởi XML, mà có thể trả về XML, JSON, YAML hoặc bất kỳ định dạng nào tùy thuộc vào yêu cầu của khách hàng. Và không giống như RPC, người dùng không bắt buộc phải biết tên thủ tục hoặc các thông số cụ thể theo thứ tự cụ thể.

Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các api nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.

Flask là một micro-framework. Điều này có nghĩa Flask là một môi trường độc lập, ít sử dụng các thư viện khác bên ngoài. Do vậy, Flask có ưu điểm là nhẹ, có rất ít lỗi do ít bị phụ thuộc cũng như dễ dàng phát hiện và xử lý các lỗi bảo mật.

Giờ để tạo một REST API đơn giản sử dụng python, chúng ta cần những thứ sau:

* Python
* Flask
* Flask-SQLAlchemy
* Flask-Restful
* SQLite3
* Jsonify
  + 1. **Ngôn ngữ lập trình Android**
       1. **Giới thiệu hệ điều hành android**

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005.

Android ra mắt năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào tháng 10 năm 2008.

Android có mã nguồn mở và Google phát hành theo giấy phép Apache. Chính mã nguồn mở cũng với một số giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đồng đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị, bằng một loại ngôn ngữ lập trình Java có sửa đổi. Vào tháng 10 năm 2012, có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 25 tỷ lượt.

Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới, vượt qua Symbian vào quý 4 năm 2010, và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu. Kết quả là mặc dù được thiết kế để chay trên điện thoại và máy tính bảng, Android đã xuất hiện trên TV, máy chơi game và các thiết bị điện tử khác. Bản chất mở của Android cũng khích lệ một đội ngũ đông đảo lập trình viên và những người đam mê sử dụng mã nguồn mở để tạo ra những dự án do cộng đồng quản lý. Những dự án này bổ sung các tính năng cao cấp cho những người dùng thích tìm tòi hoặc đưa Android vào các thiết bị ban đầu chạy hệ điều hành khác.

Android chiếm 75% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Sự thành công của hệ điều hành cũng khiến nó trở thành mục tiêu trong các vụ kiện liên quan đến bằng phát minh, góp mặt trong cái gọi là “cuộc chiến điện thoại thông minh” giữa các công ty công nghệ.

* + - 1. **Sơ lược về Android Studio**

Google cung cấp một công cụ phát triển ứng dụng Android trên Website chính thức dựa trên nền tảng IntelliJ IDEA gọi là Android Studio. Android Studio dựa vào InteliJ IDEA, là một IDE tốt nhất Java hiện nay. Do đó Android Studio sẽ là môi trường phát triển ứng dụng tốt nhất trong Android.

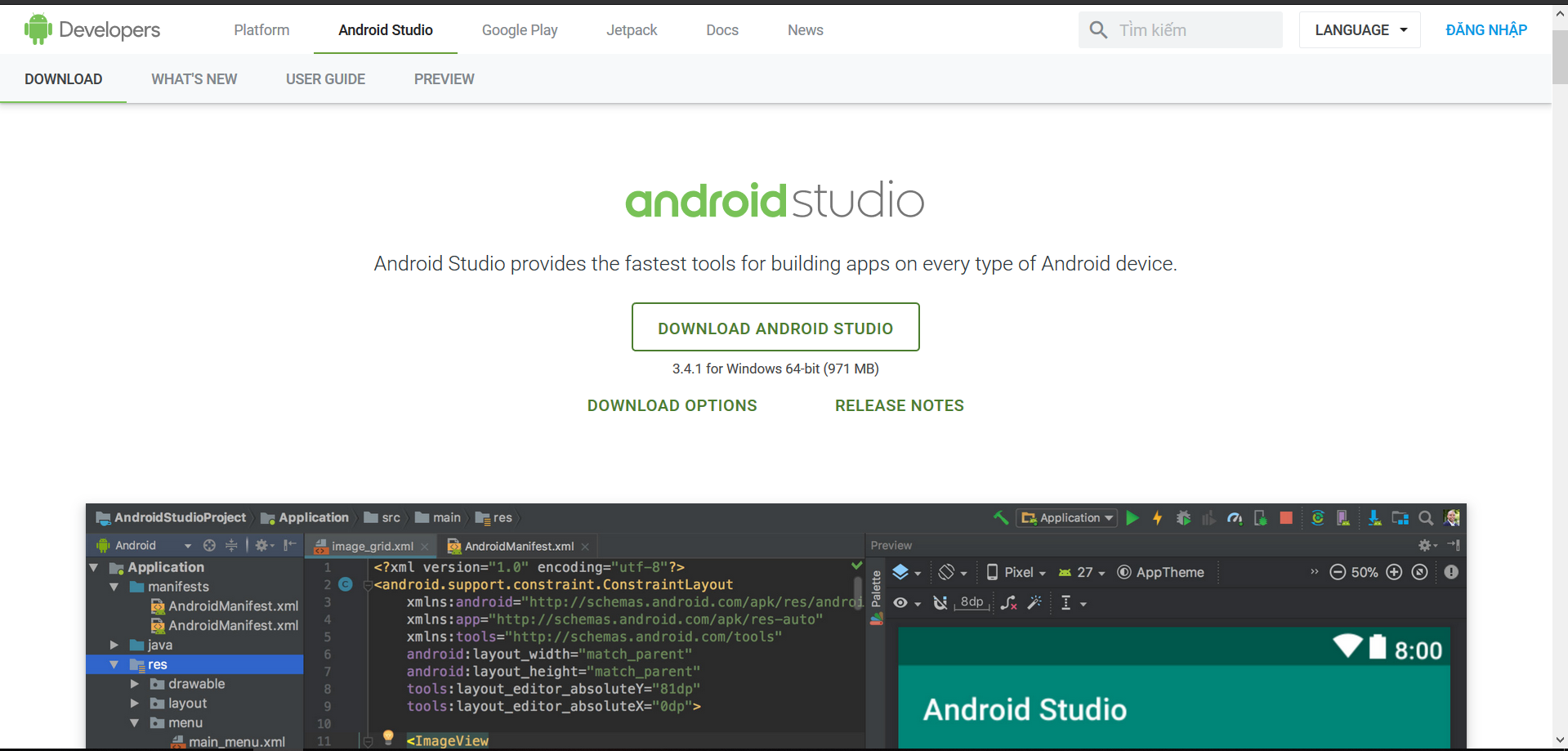
* + - 1. **Cài đặt Android Studio**

1. **Yêu cầu phần cứng máy tính**

* Microsoft Windows 10/8/7 (32 bit hoặc 64 bit).
* 4 GB RAM. (Tốt nhất là 8GB).
* 400 MB hard disk space + ít nhất 1GB cho Android SDK, emulator system images và caches.
* Độ phân giải tối thiểu 1280 x 800.
* Java Development Kit().

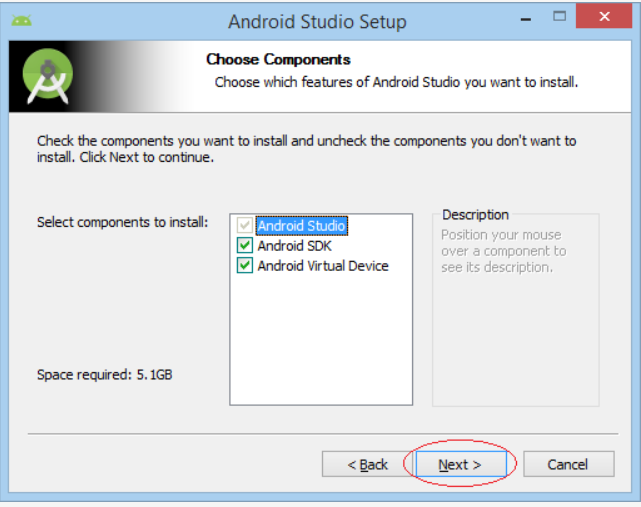
1. **Phần mềm Android Studio**

* Vào đường dẫn [**https://developer.android.com/studio**](https://developer.android.com/studio)**.**
* Để download bản mới nhất và tiến hành cài đặt click như hình



Hình 2.16

* Khi chú ý chọn cả SDK và trình giả lập thiết bị android ảo như hình:



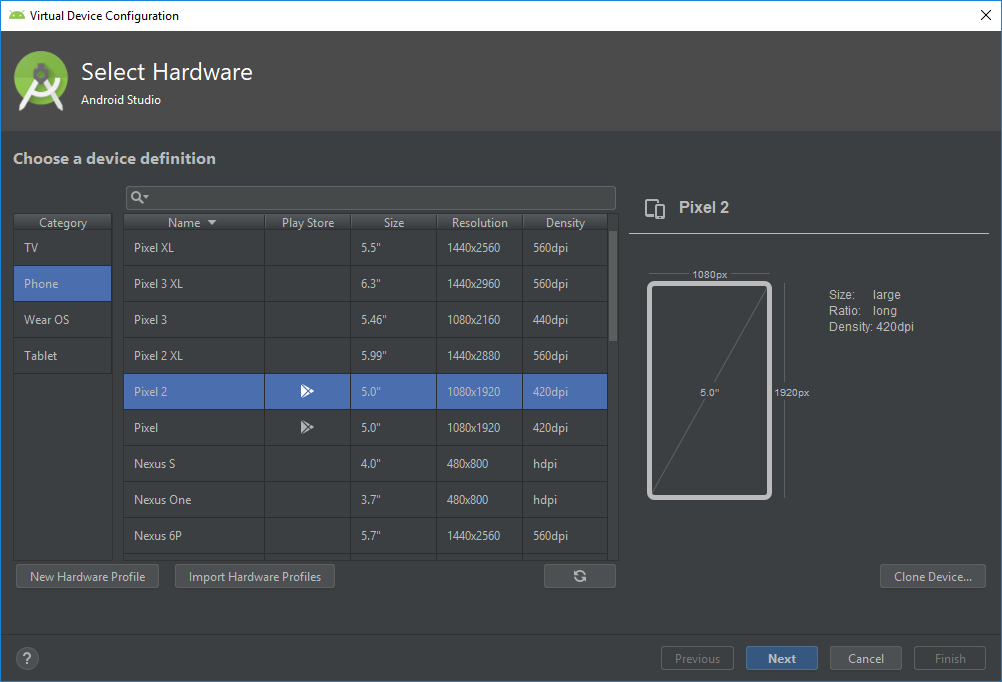
Hình 2.17

* Tiếp tục chọn Next và argee cho đến khi hoàn tất.
* Đây là màn hình khởi động.



Hình 2.18

* Các bước cài đặt thiết bị ảo trong android studio
* Máy ảo Android là một phần không thể thiếu khi chúng ta lập trình ứng dụng cho hệ điều hành Android, nó giúp chúng ta chạy thử ứng dụng ngay trên máy tính. Trong Android
* Studio có cung cấp cho chúng ta một máy ảo Android mặc định gọi là Android Virtual Device viết tắt là AVD.
* Để cài đặt máy ảo mở Android Studio lên và click vào nút AVD Manager. Cửa sổ AVD Manager sẽ xuất hiện:



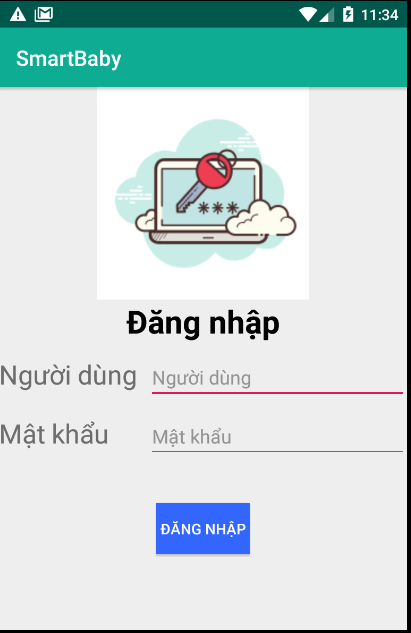
Hình 2.19

**CHƯƠNG 3: PHÂN TÍCH HOẠT ĐỘNG CỦA DỰ ÁN**

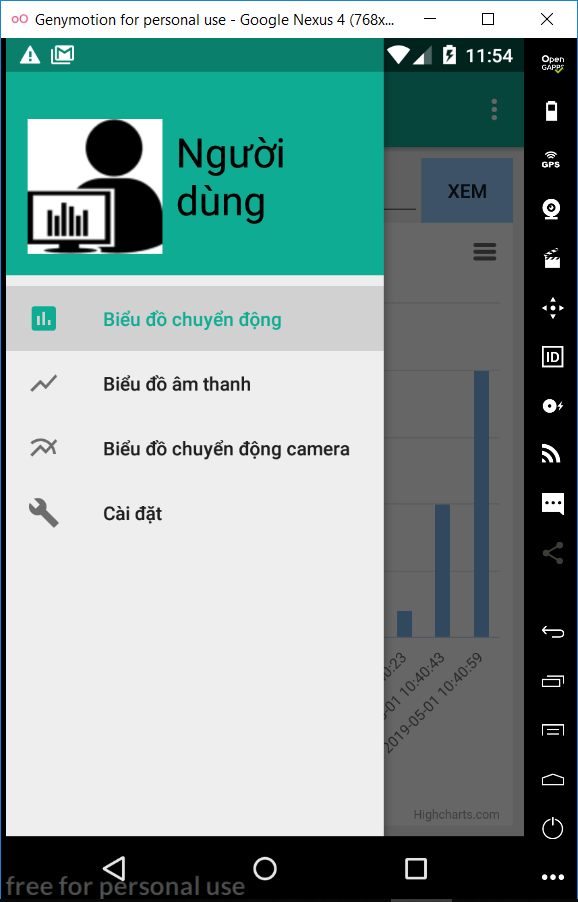
* 1. **Mô tả chức năng**
  2. **Các yêu cầu đặt ra**

**CHƯƠNG 4: THIẾT KẾ VÀ HIỆN THỰC**

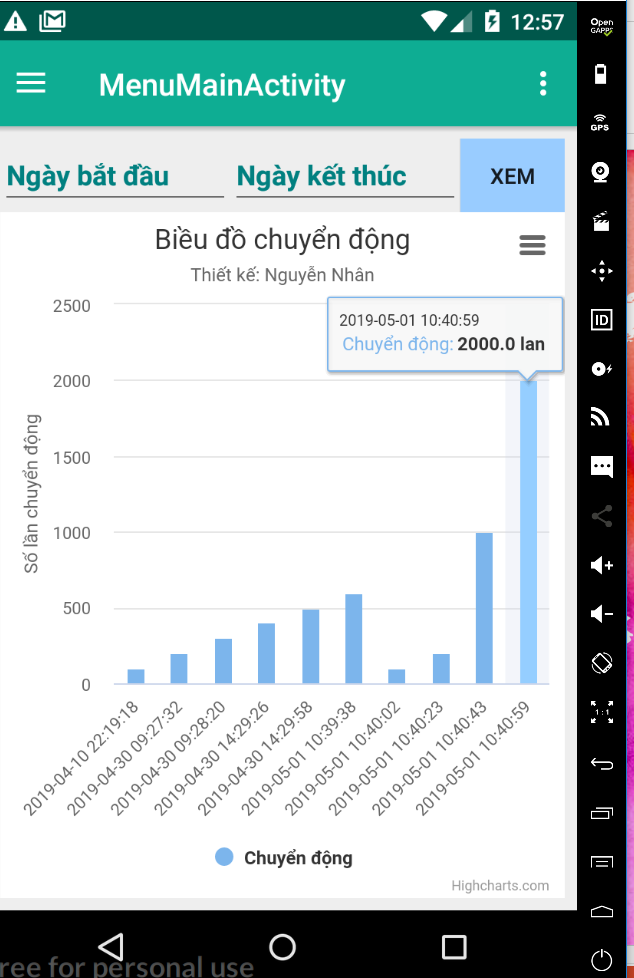
* 1. **Hình ảnh sản phẩm**
  2. **Giao diện web**
  3. **Giao diện android**
     1. **Giao diện đăng nhập**



* Chức năng người dùng có thể đăng nhập vào hệ thống.
* Khi người dùng đăng nhập sai, hệ thống sẽ thông báo lên màn hình.
* Khi người dùng đăng nhập đúng, hệ thống sẽ hiển thị sang trang người dùng.
  + 1. **Giao diện menu chọn chức năng**



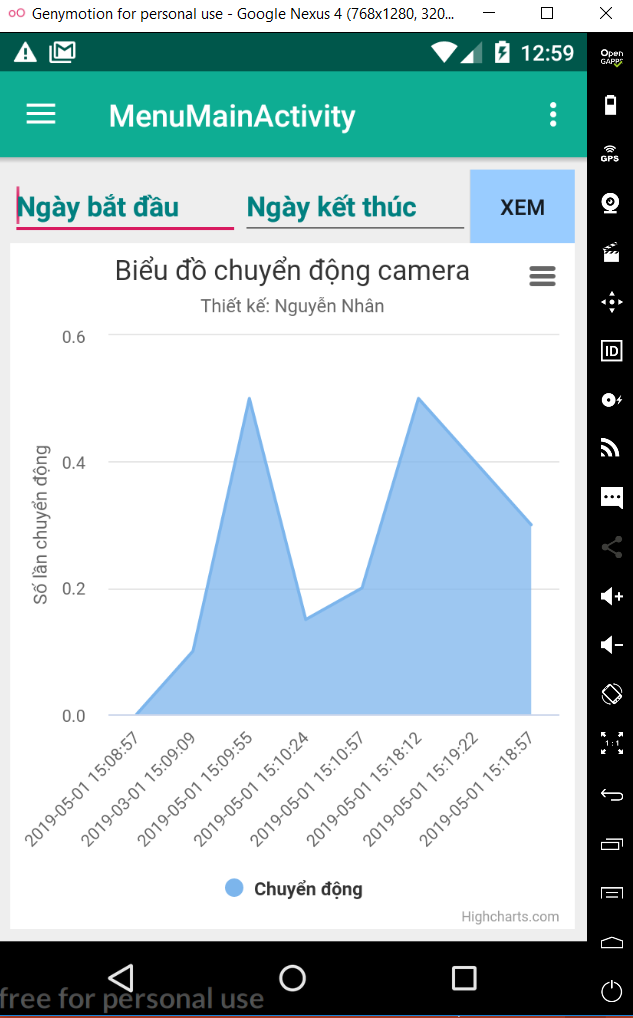
* Màn hình menu có thể xem dữ liệu biểu đồ.
  + 1. **Giao diện màn hình biểu đồ chuyển động**

****

* Màn hình sẽ hiển thị số lần chuyển động và thời gian chuyển động sẽ được gửi lên cơ dữ liệu thông qua webservice. Android có chức năng đọc dữ liệu hiển thị lên biều đồ.
* Ngày bắt đầu với ngày kết thúc thì người dùng có thể chọn ngày xem quá trình chuyển động của trẻ.
  + 1. **Giao diện màn hình biểu đồ âm thanh**

****

* Chức năng hiển thị đứa trẻ khóc sẽ hiển thị lên biểu đồ.
* Người dùng có thể xem tìm kiếm theo ngày.
  + 1. **Giao diện màn hình biểu đồ camera nhận chuyển động**

****

* Hiển thị biều đồ chuyển động từ camera đọc dữ liệu hiển thị lên biểu đồ.

**KẾT LUẬN**

Công nghệ IoT nói chung và mạng cảm biến không dây nói riêng hữa hẹn tạo ra những ứng dụng đầy tiềm năng, có thể áp dụng trong nhiều lĩnh vực khác nhau, mà đối với các công nghệ khác còn nhiều hạn chế. Tuy nhiên để triển khai mạng, người thiết kế hệ thống yêu cầu phải nắm bắt được những nhân tố tác động đến mạng, những nhược điểm của mạng cẩn phải được khắc phục. Tức là, người thiết kế cần phải quan tâm đến các tham số mạng, ví dụ như tập các chất lượng dich vụ QoS. Nhờ quá trình mô phỏng, người thiết kế hệ thống có thể đánh giá được chất lượng dich vụ mạng cung cấp, để từ đó có thể thiết kế hệ thống theo cách tối ưu nhất.

Trên cơ sở nghiên cứu tổng quan về IoT, công nghệ cảm biến không dây và các ứng dụng trong thực tiễn, em đã xây dựng thành công một chương trình thực nghiệm có tính khả thi cao (như đã trình bày trong phần đánh giá kết quả chạy thử nghiệm). Tuy nhiên, do thời gian và số lượng các cảm biến hạn chế, chương trình chưa chạy thử nghiệm với một số lượng lớn các thiết bị cảm biến, vì vậy chưa đánh giá hết được một số vấn đề như: việc chuyền nhận dữ liệu từ thiết bị cảm biến đến node mạng cảm biến, vấn đề xung đột dữ liệu… Đây cũng là một trong những hướng nghiên cứu, phát triển tiếp theo của luận văn.

**TÀI LIỆU THAM KHẢO**

1. Luigi Atzori, Antonio Iera, Giacomo Morabito, Internet of Things: Asurvey, Computer Networks 54 (2010) 2787–2805.  
2. Dr. Ovidiu Vermesan, Dr. Peter Friess, Patrick Guillemin, Internet ofThings Strategic Research Roadmap, 2009 Strategic Research Agenda, The IoT  
European Research Cluster - European Research Cluster on the  
Internet of Things (IERC).  
3. Everton Cavalcante, Marcelo Pitanga Alves, An Analysis of ReferenceArchitectures for the Internet of Things, Corba 2015.  
4. Anna Ha’c, Wireless Sensor Network Designs, University of Hawaii at  
Manoa, Honolulu, USA, John Wiley & Sons Ltd, Copyright 2003.  
5. Edgar H.Callaway, Jr. Wireless Sensor Networks: Architectures andProtocols, A CRC Press Company, Copyright © 2004 CRC Press LLC.  
6. Ovidiu Vermesan, Peter Friess, Internet of Things – ConvergingTechnologies For Smart Environments and Integrated Ecosystems, River  
Publishers Series in Communications.  
7. Kiran Maraiya, Kamal Kant, Nitin Gupta, Application based Study onWireless Sensor Network, International Journal of Computer Application (0975-  
8887), Volume 21, No.8, May 2011.  
8. I.F. Akyildiz, W. Su\*, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor  
networks: A survey, Broadband and Wireless Networking Laboratory, School  
of Electrical and Computer Engineering, Georgia Institute of Technology,  
Atlanta, GA 30332, Received 12 December 2001; accepted 20 December  
2001.<http://www.ece.gatech.edu/research/labs/bwn/sensornets.pdf>.9. Mainwaring, Polastre, et al. Wireless Sensor Networks For HabitatMonitoring, online posting. 2002 ACM International Workshop on Wireless  
Sensor Networks and Applications September 28, 2002. Atlanta, GA. (also Intel Research, IRB-TR-02-006, June 2002.) 12 Dec 2002.  
<http://www.cs.berkeley.edu/~polastre/papers/wsna02.pdf>.  
10. Matt Richardson & Shawn Wallace, Getting Started with Raspberry Pi.11. Maik Schmidt, Raspberry Pi, A Quick-Start Guide