LỜI NÓI ĐẦU 4

CHƯƠNG I. GIỚI THIỆU VỀ RASPBERRY PI 5

* 1. Giới thiệu về Raspberry Pi 5
  2. Phần cứng 6
     1. Bộ vi xử lý 7
     2. Nguồn 8
     3. Cổng HDMI 8
     4. Cổng Ethernet và USB 8
     5. Jack audio và RCA video 8
     6. Các chân GPIO 9
  3. Ngôn ngữ lập trình 9
  4. Ưu điểm và nhược điểm 9
     1. Ưu điểm 9
     2. Nhược điểm 9

CHƯƠNG II. CÀI ĐẶT HỆ ĐIỀU HÀNH VÀ CẤU HÌNH RASPBERRY PI 11

* 1. Các hệ điều hành hỗ trợ 11
     1. Raspbian 11
     2. Unbutu Mate 12
     3. Snappy Unbutu Mate 13
     4. Windows 10 IoT Core 14
     5. OSMC và OpenELEC 15
  2. Cài đặt hệ điều hành 16
  3. Điều khiển Raspberry Pi thông qua SSH 18

CHƯƠNG III. NGÔN NGỮ LẬP TRÌNH PYTHON VÀ TRÌNH BIÊN DỊCH TRÊN LINUX 20

* 1. Ngôn ngữ Python 20
  2. LXTerminal 21
  3. Cách thức biên dịch và thực thi chương trình 22
     1. Trình biên dịch 22
     2. Biên dịch với thư viện 22

CHƯƠNG IV. TIN NHẮN SMS, MODULE SIM900 VÀ TẬP LỆNH AT 24

* 1. Tin nhắn SMS 24
  2. Module SIM900 24
     1. Tổng quan về Module SIM900 24
     2. Sơ đồ chân 25
     3. Module SIM Icomsat v1.1 25
     4. Kết nối giữa module SIM và vi điều khiển 26
  3. Khảo sát tập lệnh AT 26
     1. Lệnh xóa tin nhắn 26
     2. Lệnh chọn định dạng tin nhắn 26
     3. Lệnh liệt kê tin nhắn từ vùng nhớ ưu tiên 26
     4. Lệnh đọc tin nhắn 26
     5. Lệnh gửi tin nhắn 27
     6. Lệnh nhận cuộc gọi 27
     7. Lệnh thực hiện cuộc gọi 26

CHƯƠNG V. ETHERNET VÀ HỌ GIAO THỨC TCP/IP 28

* 1. Tổng quan về Ethernet 28
     1. Cấu trúc khung tin Ethernet 28
     2. Cấu trúc địa chỉ Ethernet 28
     3. Các loại khung Ethernet 29
        1. Khung unicast 29
        2. Khung broadcast 29
        3. Khung multicast 29
        4. Truy cập bus sử dụng phương pháp CSMA/CD 29
  2. Hoạt động của mạng Internet 30
     1. Các giao thức 30
     2. Tên miền và địa chỉ IP 31
     3. Mô hình khách chủ 31
  3. Họ giao thức TCP/IP 32

CHƯƠNG VI. THIẾT KẾ VÀ THI CÔNG 34

* 1. Sơ đồ khối 34
  2. Sản phẩm 35
  3. Code 40
     1. Gửi sms 40
     2. Đọc sms 41
     3. Xóa sms 42

CHƯƠNG VII. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 43

* 1. Kết luận 43
  2. Hướng phát triển 43

CHƯƠNG VIII. TÀI LIỆU THAM KHẢO 44

**LỜI NÓI ĐẦU**

Ngày nay, với những ứng dụng của khoa học kỹ thuật tiên tiến, thế giới của chúng ta đã và đang thay đổi ngày một văn minh và hiện đại hơn. Để thông báo thông tin về một nội dung, sự kiện cho học sinh, nhà trường có thể đăng trực tiếp lên website thay vì phải ghi ra và dán tại bảng thông báo của trường như trước đây. Học sinh, sinh viên có thể dễ dàng biết được những thông tin đó bằng cách truy cập vào địa chỉ website của trường. Nhưng vì sự cố mạng hoặc một vài lý do khác thì việc liên lạc này sẽ không diễn ra được như mong muốn, sinh viên sẽ không nắm được những thông tin từ phía nhà trường. Chỉ với Raspberry Pi và module SIM900 em đã thiết kế được một hệ thống có thể giúp thông tin từ phía nhà trường đến được với sinh viên và ngược lại mà không cần phải phụ thuộc quá nhiều vào internet.

**CHƯƠNG I. GIỚI THIỆU VỀ RASPBERRY PI**

* 1. **Giới thiệu về Raspberry Pi**

Raspberry Pi là một chiếc máy tính mini có kích thước cỡ như một thẻ ATM hoặc chiếc iphone chạy hệ điều hành Linux. Nó được phát triển bởi Raspberry Pi Foundation – là một tổ chức phi lợi nhuận với tiêu chí xây dựng hệ thống mà nhiều người dùng có thể sử dụng được trong những công việc tùy biến khác nhau.

Ban đầu, Raspberry Pi là một thẻ card được cắm trên board mạch máy tính, sau đó được phát triển thành một board mạch đơn có chức năng như một máy tính mini dùng để giảng dạy trong môn khoa học và máy tính ở các trường trung học.



*Hình 1. Kích thước Raspberry Pi model B ngang bằng chiếc Iphone 4.*

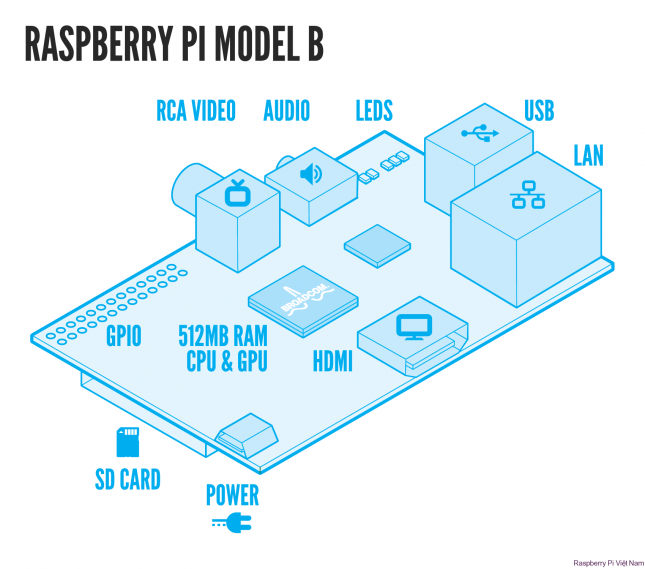
Nhiệm vụ ban đầu của dự án Raspberry Pi là tạo ra máy tính có giá thành thấp có khả năng lập trình cho những đối tượng là học sinh và sinh viên, nhưng nó đã được nhận được nhiều sự quan tâm từ nhiều đối tượng khác nhau.

Raspberry Pi được sản xuất bởi 3 OEM: sony, Qsida, Egooman và được phân phối chính bởi Element14, RS Components, Egoman. Tính tới thời điểm hiện tại, Raspberry Pi Foundation đã tung ra 5 thế hệ Raspberry có mặt trên thị trường (thế hệ 1, 2, 3, nano và zero Pi) và thế hệ mới nhất là thế hệ thứ 3 được ra mắt vào ngày 29/1/2016 với nhiều điểm cải tiến so với các phiên bản trước nó, đặc biệt là hỗ trợ Wifi và Bluetooth sẵn trên board mạch.

Raspberry Pi là một máy tính mini nhưng nó không thay thế được hoàn toàn hệ thống máy tính để bàn hay máy tính xách tay. Trước đây, Raspberry Pi không thể chạy hệ điều hành Windows trên nó vì BCM2835 dựa trên cấu trúc ARM nên không hỗ trợ mã x86/x64, chỉ có thể chạy bằng Linux với các tiện ích như lướt web, xem phim, các thao tác bằng Desktop, lập trình điều khiển thiết bị,… và nhiều chức năng khác. Tuy nhiên, kể từ tháng 2 năm ngoái, Raspberry Pi có thể cài đặt và chạy được Windows 10 IoT trên hệ thống của mình, điều này làm tăng tính ngạc nhiên về chiếc máy tính nhỏ có giá thành rẻ này, nó rất hoàn hảo cho những hệ thống điện tử và những dự án DIY.

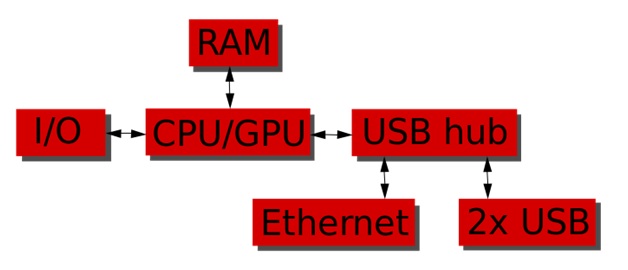
* 1. **Phần Cứng**

Các thế hệ Pi (trừ phiên bản nano và zero) có hai phiên bản chủ yếu là model A (A và A+) và model B (B và B+), nhưng model B thông dụng hơn cả do nó được trang bị sẵn cổng Ethernet, nhiều cổng USB và cấu hình mạnh hơn so với model A.



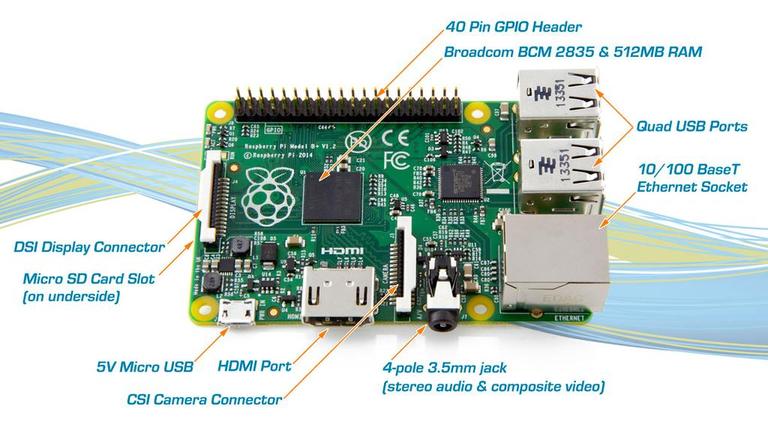
*Hình 2. Sơ đồ cấu tạo của Raspberry Pi (model B).*

Cấu tạo của Raspberry gồm các khối: Thẻ nhớ, nguồn, cổng HDMI, cổng Ethernet và USB, jack Audio và RCA video, các chân GPIO và chip xử lý.



*Hình 3. Sơ đồ khối của raspberry Pi.*

Phần cứng của Raspberry Pi qua nhiều phiên bản được trang bị và nâng cấp về cấu hình điều này làm tăng hiệu năng của board, hỗ trợ được nhiều hơn. Điển hình là Raspberry Pi 3 model B, ngoài việc sử dụng như một máy tính bình thường (chạy hệ điều hành Linux hoặc Windows 10 IoT) máy còn trang bị 40 chân GPIO, CPU phiên bản mới BCM2837 từ Broadcom với tốc độ 1.2GHz 4 nhân với kiến trúc ARM Cortex-A53 64-bit.



*Hình 4. Các khối chức năng trên board Raspberry Pi*

* + 1. ***Bộ vi xử lý***

Đặc tính của Raspberry Pi xây dựng xoay quanh bộ xử lý SoC Broadcom BCM2835 (là chip xử lý mạnh mẽ và có kích thước nhỏ, hay được dùng trên điện thoại di động), trong đó bao gồm một bộ vi xử lý ARM1176JZF-S 700 MHZ, GPU Video Core IV, RAM và các tính năng khác được tích hợp bên trong chip có điện năng tiêu thụ thấp này.

SoC có một bộ nhớ cache cấp 1 16KB và một bộ nhớ cache cấp 2 128KB. Cache cấp 2 này được sử dụng chủ yếu bởi GPU. Nó được xếp chồng lên nhau phía dưới Chip và Ram.

* + 1. ***Nguồn***

Tương tự như các dòng điện thoại hoặc tablets, nguồn cung cấp cho Raspberry là 5v và dòng điện phải lớn hơn 1.5A, được kết nối bằng cổng micro-USB. Tuy nhiên, trên thị trường cũng có nhiều loại Pin, sạc dự phòng có thể cung cấp nguồn cho Raspberry hoạt động bình thường nên tăng thêm tính năng di động hơn cho hệ thống có sử dụng board này.

* + 1. ***Cổng HDMI***

Trên board Raspberry Pi được trang bị một cổng HDMI (High Definition Multimedia Interface) cho phép xuất tín hiệu chất lượng hình ảnh cao full hd 1080p. GPU trên board có thể cho chất lượng cao và có thể sử dụng các thư viện hỗ trợ được cung cấp bởi nhà sản xuất.

* + 1. ***Cổng Ethernet và USB***

Cổng Ethernet chỉ được trang bị trên các model B của Raspberry Pi giúp cho các model này có thể kết nối với mạn LAN/Internet dễ dàng. Các model A không được trang bị cổng Ethernet nhưng cũng có thể kết nối với mạng LAN/Internet thông qua usb wifi. Tùy theo phiên bản mà sẽ có số cổng USB khác nhau, nhiều nhất là 4 cổng USB và có thể mở rộng bằng hub USB rời bên ngoài. Các cổng USB này có thể giao tiếp với các thiết bị ngoại vi thông thường như : chuột, bàn phím, usb wifi,…

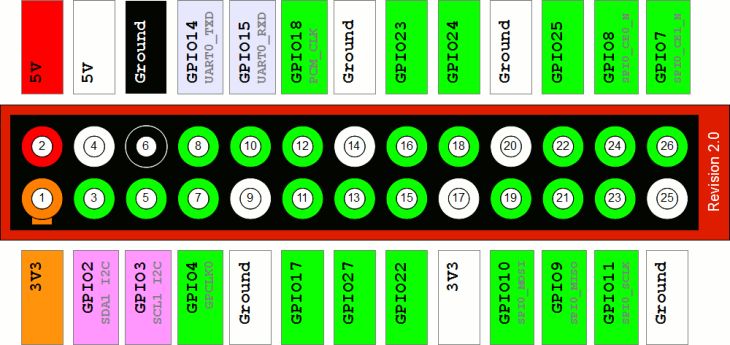
Cả hai cổng Ethernet và USB được kết nối với chip LAN9512. Theo datasheet của chip này, nó đáp ứng chuẩn USB 2.0 tốc độ cao và hỗ trợ Ethernet tốc độ 10/100Mbps.

* + 1. ***Jack audio và RCA video***

Cổng âm thanh và video RCA được thiết kế nằm đối diện với công HDMI. Raspberry Pi không hỗ trợ âm thanh qua cổng HDMI, nhưng chúng ta cũng có thể lấy được tín hiệu âm thanh bằng jack âm thanh 3.5 mm. Ngoài ra cổng âm thanh 3.5 mm thì raspberry còn hỗ trợ một công RCA video giúp cho nó có thể gửi dữ liệu video đến bất cứ thiết bị hiện thị nào có cổng RCA.

* + 1. ***Các chân GPIO***

Raspberry có 26 đến 40 chân tùy phiên bản, đối với phiên bản Raspberry Pi 1 model B có 26 chân, các chân của nó thường dùng cho mục đích gpio, từ điều khiển led và động cơ servo cho đến điều khiển động cơ và có thể kết nối với các board mở rộng khác.



*Hình 5. Sơ đồ chân GPIO của Raspberry Pi.*

Ngoài chức năng gpio cở bản, Raspberry Pi còn được tích hợp các module như : UART, SPI, I2C, …nhờ được tích hợp các module này, board có thể giao tiếp dễ dàng với các board mở rộng, các module khác để thực hiện những năng mong muốn.

* + 1. ***Ngôn ngữ lập trình***

Ngoài ngôn ngữ lập trình được cài đặt sẵn trong hệ điều hành là Python, Rasperry Pi còn hỗ trợ rất nhiều ngôn ngữ lập trình khác như: Scratch, Html5, Javascript, Jquery, Java, C, C++, Perl, Erlang, Wolfram, … và nhiều ngôn ngữ khác.

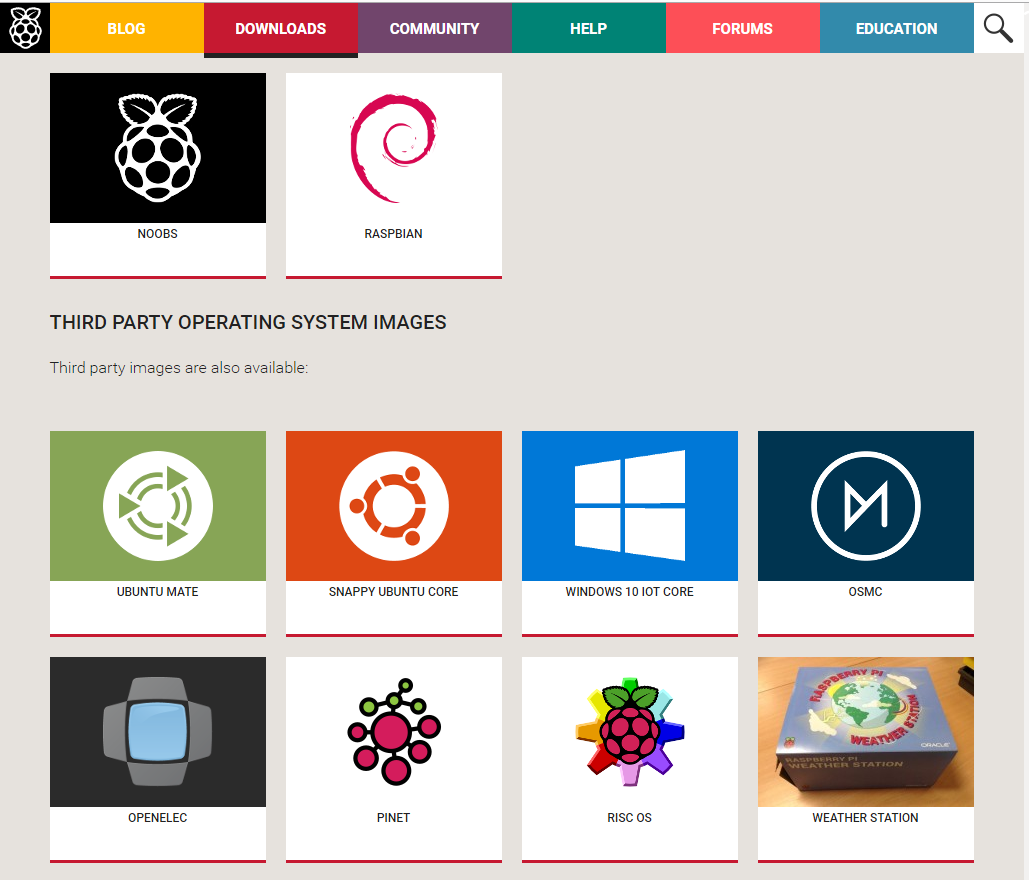
* 1. **Ưu điểm và nhược điểm**
     1. ***Ưu điểm***
* Thiết kế nhỏ gọn, sử dụng điện năng thấp, giá thành rẻ.
* Chức năng như một máy tính, có các kết nối HDMI, âm thanh và video.
* Nhiều chân gpio và các module được tích hợp.
* Cộng đồng người trên thế giới sử dụng nhiều.
* Hệ điều hành dễ sử dụng và được cung cấp miễn phí.
* Khả năng hoạt động liên tục 24/7.
  + 1. ***Nhược điểm***
* Không có nút on/off, reset.
* Không tích hợp module thời gian thực.
* Nhiều phần mềm hỗ trợ còn thu phí.
* Không được quyền tự do sản xuất lại board.

**CHƯƠNG II: CÀI ĐẶT HỆ ĐIỀU HÀNH VÀ CẤU HÌNH RASPBERRY PI**

* 1. **Các hệ điều hành hỗ trợ**

Raspberry Pi được hỗ trợ rất nhiều hệ điều hành, trong đó Raspbian là hệ điều hành chính thức của Raspberry Foundation, ngoài ra còn có 7 hệ điều hành khác được xác nhận hỗ trợ chính thức và nhiều hệ điều hành do các nhà phát triển tối ưu khác.

Tất cả hệ điều hành này có thể download trực tiếp tại website: <https://www.raspberrypi.org/downloads/> và các hệ điều hành này đều được cung cấp, hỗ trợ miễn phí.

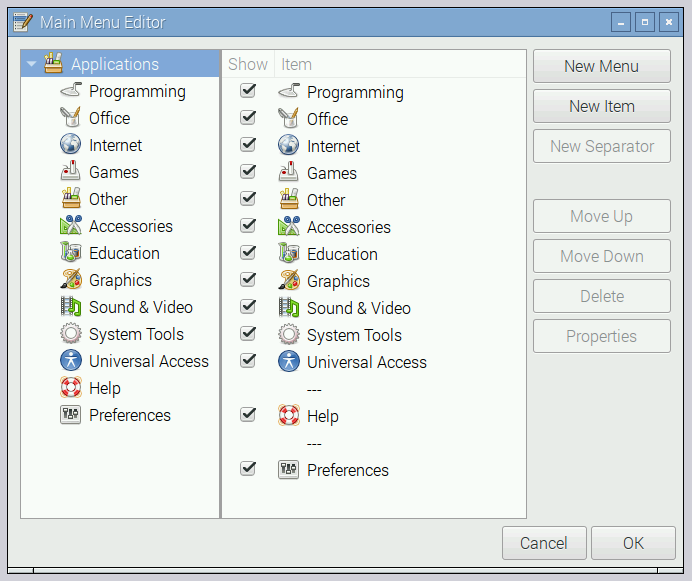


*Hình 6. Giao diện trên trang download hệ điều hành.*

Dưới đây là một số hệ điều hành phổ biến dành cho Raspberry Pi:

* + 1. ***Raspbian***

Đây là hệ điều hành cơ bản, phổ biến nhất do chính Raspberry Foundation cung cấp. Nó cũng được hãng khuyến cáo sử dụng, nhất là cho người mới làm quen với board này.



*Hình 7. Các ứng dụng sẵn có của hệ điều hành Raspbian.*

Raspbian có dung lượng sau giải nén là khoảng gần 4GB, nên cần một thẻ nhớ tối thiểu là 4GB để cài đặt hệ và sử dụng điều hành này. Raspbian hướng đến người dùng với mục đích:

* Sử dụng Raspberryi như máy tính văn phòng để lướt web, soạn văn bản, check mail và thỉnh thoảng nghe nhạc, xem phim.
* Nghiên cứu và phát triển các thiết bị điều khiển tự động.
* Sử dụng như một máy chủ, cung cấp các dịch vụ như : webserver, file server, printer, ….

Sau một thời gian sử dụng, hệ điều hành này cho thấy sự ổn định và tốc độ nhanh. Giao diện đồ họa đẹp, hiệu năng tốt.

* + 1. ***Unbutu Mate***

Tương tự như Raspbian, Unbutu Mate cũng hướng đến người dùng sử dụng Raspberry Pi như máy tính văn phòng. Tuy nhiên Unbutu Mate có giao diện đẹp hơn so với Raspbian. Unbutu Mate được phát triển từ Unbutu – hệ điều hành được xem là đối đầu trực tiếp với Windows.



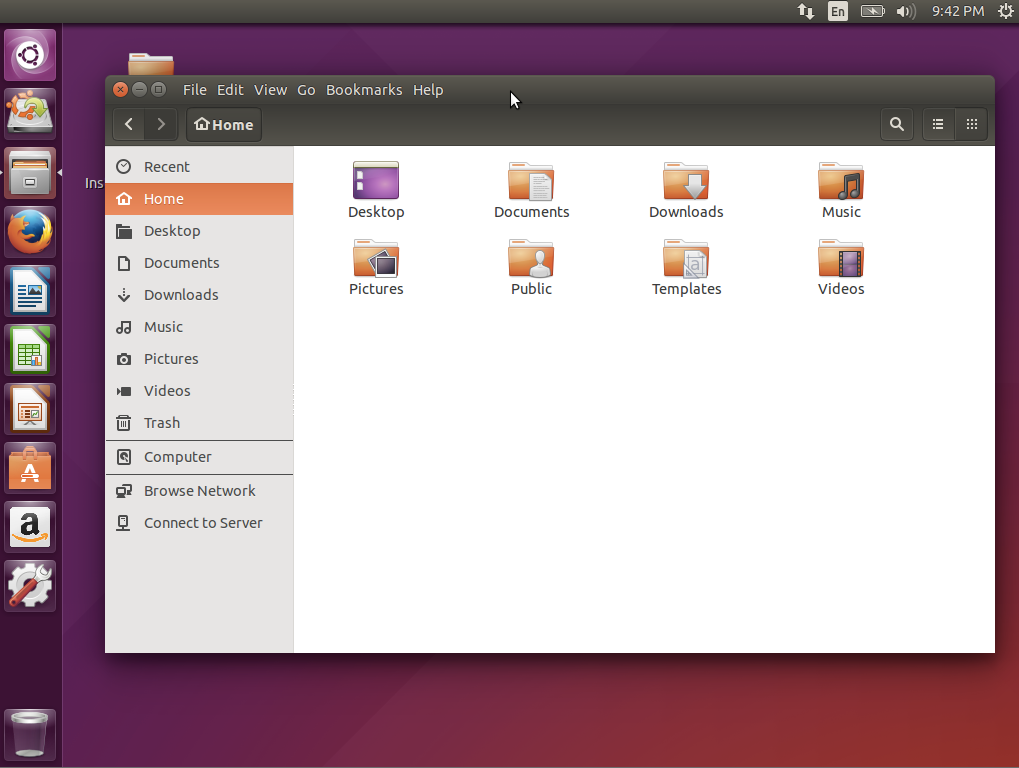
*Hình 8. Desktop của hệ điều hành Unbutu Mate.*

Martin Wimpress và Rohith Madhavan là cha đẻ của Unbutu Mate được phát triển từ Unbutu gốc. Theo tác giả, nó được tối ưu hơn đối với Raspberry Pi 2 và 3, tuy nhiên để đảm bảo tốc độ cao nhất thì nên sử dụng thẻ Micro SD từ class 6 trở lên. Theo đánh giá của nhiều người, Unbutu Mate có tốc độ rất nhanh, giao diện đẹp và hỗ trợ đầy đủ các phần mềm thông dụng cho nhu cầu văn phòng.

* + 1. ***Snappy Unbutu Core***

Những năm trước đây, Canoical đã làm việc không ngừng nghỉ để mở rộng hệ điều hành Unbutu tới nhiều sản phẩm khác nhau. Giờ đây, họ đã giới thiệu một hệ điều hành rút gọn của Unbutu, nó được tạo ra với nhiệm vụ là chạy các ứng dụng đám mây và trở thành một phần quan trọng trong IoT (Internet of Things) giúp các thiết bị (điện thoại, tivi, đèn, quạt, đồng hồ, ...) trong đời sống kết nối với nhau một cách hoàn hảo.

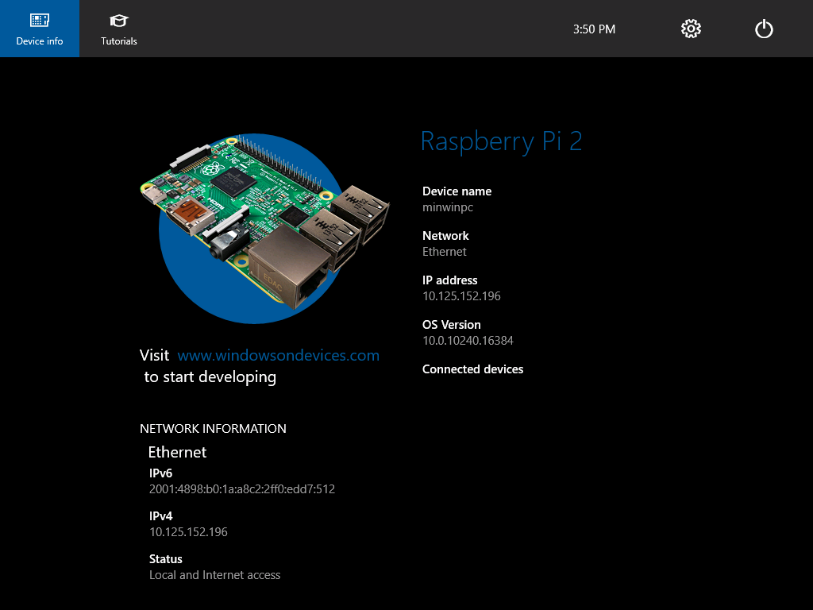
Snappy Unbutu Core được xây dựng trong dự án Unbutu Core. Unbutu Core là nhân của hệ điều hành Unbutu, khá trần trụi nhưng là một thành phần rất quan trọng của hệ điều hành Unbutu, Snappy Unbutu Core được thiết kế chạy trong một môi trường khá hạn chế. Thế mạnh của nó là chạy được nhiều ứng dụng mà không cần một hệ điều hành Unbutu đầy đủ.



*Hình 9. Màn hình làm việc của hệ điều hành Snappy Unbutu Core.*

* + 1. ***Windows 10 IoT Core***

Tương tự như Snappy Unbutu Core, Windows 10 IoT cũng chỉ có nhân của Windows, nó không có giao diện đồ họa hay các phần mềm thông dụng như office,.. Windows 10 IoT được sử dụng cho mục đích phát triển các ứng dụng IoT. Microsoft cho biết IoT Core được thiết kế để có thể hoạt động với một loạt các ngôn ngữ mã nguồn mở, khiến các nhà sản xuất dễ dàng cài đặt trên các thiết bị của mình cũng như phát triển ứng dụng cho riêng mình.



*Hình 10. Hệ điều hành windows 10 IoT trên Raspberry Pi.*

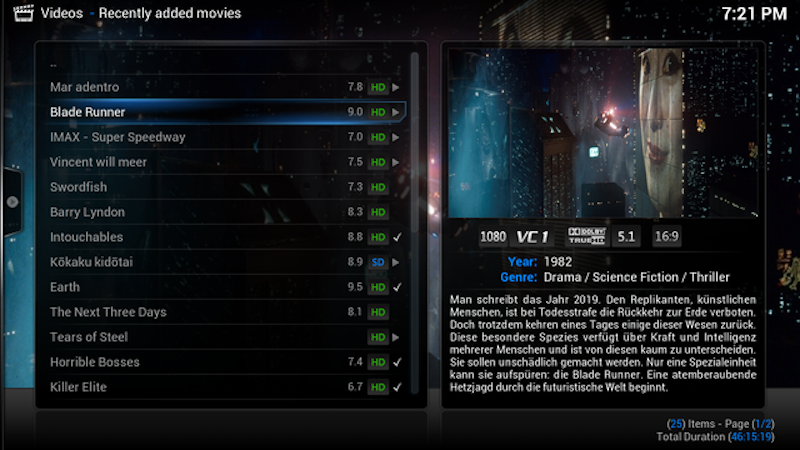
* + 1. ***OSMC và OpenELEC***

Đây là hai hệ điều hành phổ biến cho nhu cầu giải trí qua Raspberry Pi. OSMC được phát triển từ RaspBMC, còn OpenELEC thì đi lên từ Xbian. Cả OSMC và OpenELEC đều được phát triển để chạy trên KODI, tuy nhiên OSMC được phát triển với đầy đủ nền tảng của Debian, vì vậy OSMC có thể làm được nhiều hơn OpenELEC. Cả hai hệ điều hành này phù hợp cho nhu cầu biến Raspberry Pi thành một Media Center trong nhà của bạn hoặc làm một thiết bị chơi Video/Audio trên xe ô tô.



*Hình 11. Hệ điều hành OSMC.*

Về giao diện, nếu OpenELEC sử dụng nguyên giao diện đẹp đẽ của Koidi thì OSMC được thiết kế lại giao diện mới với các menu đơn giản hơn trên nền chữ trắng. Thoạt nhìn bạn sẽ thất OSMC có giao diện không bắt mắt. Mặc dù vậy, OSMC có nhiều tùy chon hơn OpenELEC.



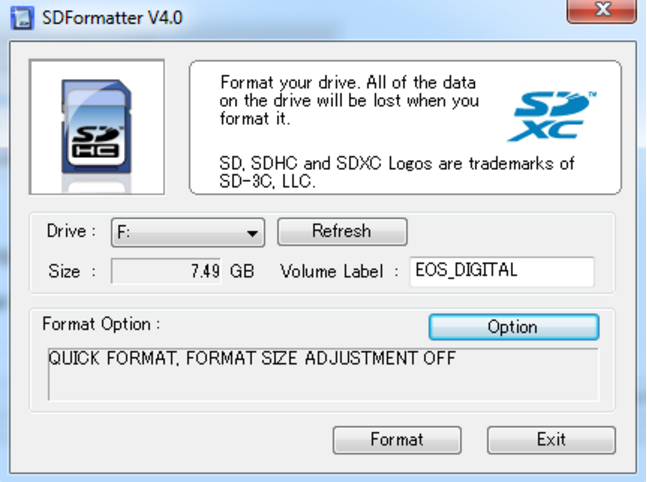
*Hình 11. Hệ điều hành OpenELEC*

* 1. **Cài đặt hệ điều hành**

Để cài đặt hệ điều hành cho Raspberry Pi cần chuẩn bị một thẻ nhớ micro SD có dung lượng lớn hơn 4GB và tiến hành theo các bước sau:

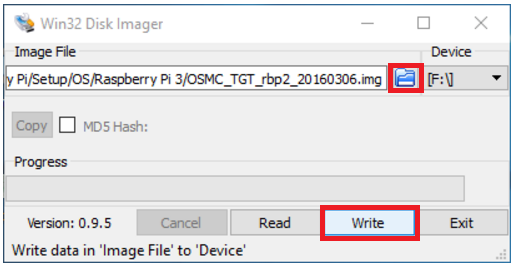
Bước 1: <https://www.raspberrypi.org/downloads/> tải file image hệ điều hành cần cài về máy tính và tiến hành giải nén. Chuẩn bị phần mềm SDFormatter dùng để format cho thẻ nhớ và phần mềm Win32 Disk Imager.

Bước 2: Cắm thẻ nhớ vào máy tính, mở phần mềm SDFormatter, tiến hành format thẻ nhớ như hình dưới:



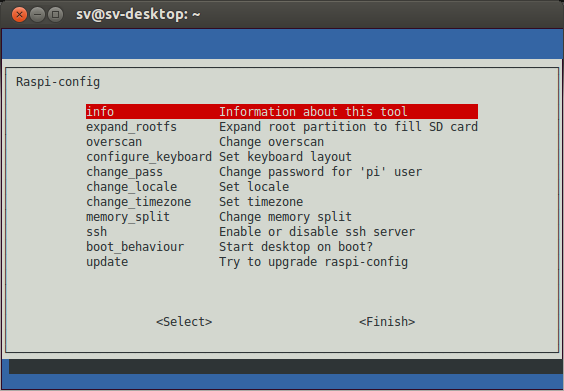
*Hình 12. Format thẻ nhớ bằng phần mềm SDFormatter.*

Bước 3: Mở phần mềm Win32 Disk Imager, tìm đến file image vừa tải ở bước một và tiến hành write vào thẻ nhớ như hình 13.



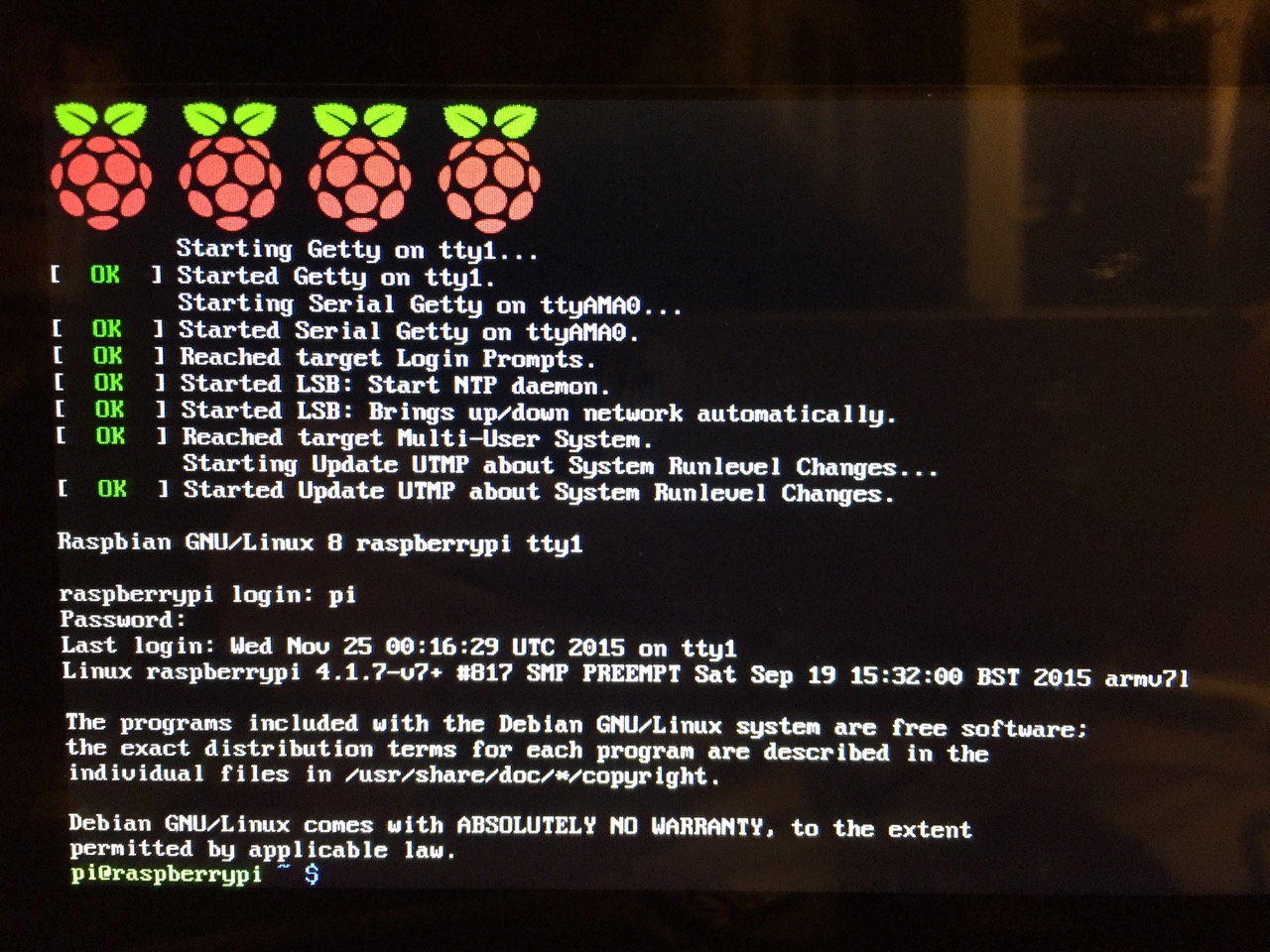
*Hình 13. Cài hệ điều hành vào thẻ nhớ.*

Bước 4: Sau khi hoàn tất quá trình cài hệ điều hành vào thẻ nhớ, lấy thẻ nhớ vừa cài hệ điều hành và cắm vào board Raspberry Pi (đã kết nối với LCD, Bàn phím).



*Hình 14. Cửa sổ cài đặt hệ thống của Pi ở lần cài đặt đầu tiên.*

Chọn “Finish” để kết thúc quá trình cài đặt, sau đó trên màn hình sẽ xuất hiện yêu cầu đăng nhập. Tài khoản mặc định là : “pi” và mật khậu là : “raspberry”.



*Hình 15. Sau khi đăng nhập thành công.*

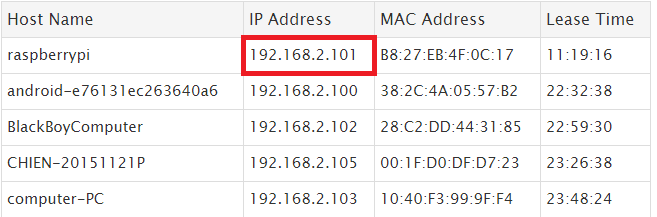
Bước 5: Sau khi đăng nhập thành công, nhập “startx” để vào màn hình làm việc Desktop của hệ điều hành. Đối với hệ điều hành Raspbian Lite không có màn hình đăng nhập, nên chỉ thao tác bằng dòng lệnh. Vậy là quá trình cài đặt thành công.

* 1. **Điều khiển Raspberry Pi thông qua SSH**

Phương pháp này cho phép lập trình, điều khiern Raspberry từ xa (thông qua mạng LAN/Internet) mà không cần sử dụng LCD, chuột, bàn phím. Để có thể điều khiển bằng phương pháp này thì Raspberry Pi phải được kích hoạt “Enable SSH server”. Thông thường thì server này được mở mặc định. Và làm theo các bước sau:

Bước 1: Kết nối Raspberry Pi với router thông qua cổng Ethernet hoặc module wifi.

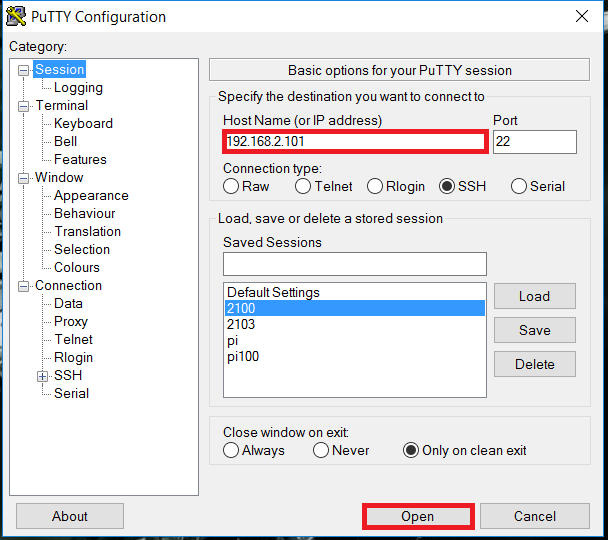
Bước 2: đăng nhập vào Router và tìm địa chỉ IP của Raspberry.



*Hình 16. DHCP Client List của router.*

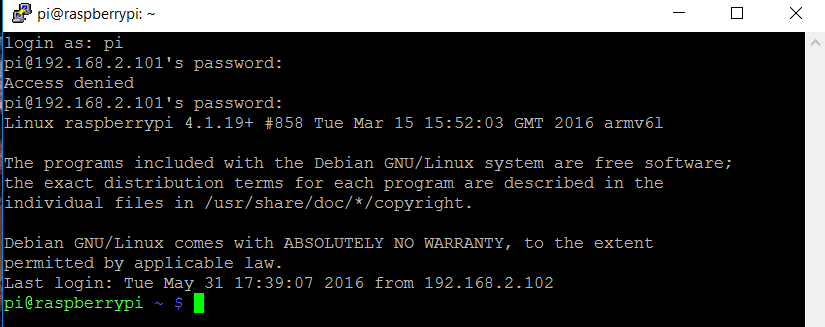
Bước 3: Download phần mềm PuTTY và tight VNC.

Bước 4: Mở phần mềm PuTTY, nhập địa chỉ IP vừa tìm được và nhấn open.



*Hình 17. Cài đặt PuTTY*

Sau khi nhấn open, một cửa sổ đăng nhập hiện ra, chúng ta tiến hành đăng nhập với tài khoản và mật khẩu mặc định của Raspberry Pi.



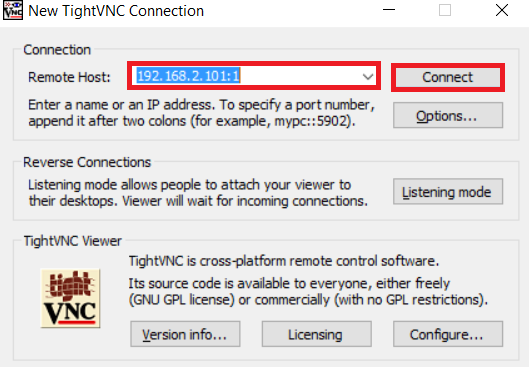
*Hình 18. Sau khi đăng nhập thành công.*

Bước 4: Với Phần mềm PuTTY ta chỉ có thể điều khiển Raspberry bằng dòng lệnh mà không thể nhìn thấy Desktop, GUI… của hệ điều hành được, nên muốn giám sát được Raspberry Pi ta cần tiếp tục cài Tight VNC bằng cách soạn hai dòng lệnh:

Sudo apt-get install tightvncserver

Vncserver :1 –name RasPi –depth 16 –geometry 1024x768.

Sau khi cài đặt xong, từ máy tính mở phần mềm tight VNC và nhập địa chỉ IP của board như hình 19.



*Hình 19. Nhập địa chỉ IP và kết nối đến server tight VNC.*

Bước 5: Đăng nhập bằng tài khoản mặc định và một mà hình Desktop được hiển thị như hình 7.

**CHƯƠNG III. NGÔN NGỮ LẬP TRÌNH PYTHON VÀ TRÌNH BIÊN DỊCH TRÊN LINUX**

* 1. **Ngôn ngữ Python**

Python là một ngôn ngữ lập trình thông dịch (interpreted language) do Guido Van Rosum tạo ra năm 1990. Python được phát triển trong một dự án mã nguồn mở, do tổ chức phi lợi nhuận Python software Foundation quản lý. Tên gọi Python được Van Rossum đặt bắt nguồn từ một series hài kịch “Monty Python’s Flying circus” của đài truyền hình BBC được phát trong những năm 1970.

Ban đầu, ngôn ngữ Python được phát triển để chạy trên nền Unix, nhưng sau đó nó đã mở rộng sang các hệ điều hành khác như MacOS, Windows ,… Đây là một ngôn ngữ dễ học, có những cấu trúc dữ liệu cấp cao hiệu quả và hướng lập trình đối tượng đơn giản. Ở chế độ tương tác, trình thông dịch Python tương tựu như shell trong các hệ điều hành học Unix, ở đó ra có thể nhập vào từng biểu thức rồi gõ Enter, các kết quả thực thi sẽ được hiển thị ngay lập tức (khác với các ngôn ngữ như C/C++ và Fortran đòi hỏi phải biên dịch chương trình). Đặc điểm này rất hữu ích cho người mới học, giúp họ tìm hiểu cách thức thực thi của chương trình từng bước một, hoặc để các lập trình viên chạy thử mã lệnh trong suốt quá trình phát triển phần mềm.

Một số đặc điểm chính của ngôn ngữ Python gồm có:

* *Ngôn ngữ thông dịch*: biên dịch chương trình nguồn theo từng phân đoạn hoặc từng câu lệnh, sau đó thực thi các đoạn mã đã được biên dịch, chương trình đích vừa tạo ra sẽ không được lưu lại. Do đó nó có ưu điểm tiết kiệm thời gian phát triển ứng dụng vì không cần phải thực hiện biên dịch và liển kết.
* *Ngôn ngữ kiểu động*: ngôn ngữ mà các kiểu chỉ được gán lên các dữ liệu trong thời gian chương trình được thực thi. Chương trình không kiểm tra các ràng buộc về kiểu dữ liệu tại thời điểm dịch, mà là tại thời điểm thực thi. Ngoài ra, khi sử dụng Python ta không cần phải khai báo biến, biến được xem là đã khai báo nếu nó được gán một giá trị lần đầu tiên. Căn cứ vào mỗi lần gán, Python sẽ tự động xác định kiểu dữ liệu của biến.
* *Cơ chế cấp phát bộ nhớ tự động*: không khởi tạo giá trị ban đầu cho biến.
* *Khả năng mở rộng*: trình thông dịch Python có thể được mở rộng dễ dàng với những chức năng và kiểu dữ liệu được viết trong C hoặc C++ (hoặc ngôn ngữ nào đó có thể gọi được từ C). Ta cũng có thể mở rộng chức năng của trình thông dịch, hoặc liên kết các chương trình Python với các thư viện chỉ ở dạng nhị phân.

Tuy nhiên, cũng như tất cả các ngôn ngữ lập trình khác, Python không phải là không có những khuyết điểm:

* Python vốn không phải là ngôn ngữ được xây dựng nên để phục vụ cho mục đích khoa học kỹ thuật do đó các thư viện của nó không so sánh được với các ngôn ngữ chuyên sau như Matlab/Octave, Mathematica, …
* Tốc độ thực thi chương trình không nhanh bằng những ngôn ngữ khác như C/C++, Fortan, …

Các nhánh chính của ngôn ngữ Python:

* *Cpython*: được xây dựng nên từ ngôn ngữ C bởi Guido va Rossum
* *Jpython*: Python cho Java.
* *IronPython*: Python cho Microsoft .NET.
* *PyMite*: Python cho các vi xử lý (microprocessor)

Ngoài ra, Python có các thư viện phục vụ cho khoa học kỹ thuật chẳng hạn như:

* *NumPy*: đây là thư viện được viết bằng ngôn ngữ lập trình cấp thấp như C và Fortran (chủ yếu phục vụ cho các tính toán ma trận), được xây dựng dựa trên nền tảng các thư viện như BLAS, LAPACK.
* *SciPy*: thư viện này phục vụ cho các tính toán tích phân số, giải hệ thống phương trình vi phân, tính toán tối ưu, …
* *Mathplotlib*: công cụ vẽ đồ thị tương đương với bộ công cụ trong Matlab.
* *Multi-processing*: gói tính toán song song.
  1. **LXTerminal**

Để tương tác được với shell chúng ta cần một chương trình giao tiếp, trong Linux có hai phương thức giao tiếp gồm có:

* Sử dụng giao diện đồ họa người dùng (Graphical User Interfaces - GUI) trong hầu hết các hệ điều hành Linux hiện nay đều có sẵn phần giao diện này, trong đó ta có thẻ thoải mái sử dụng chuột tương tự như trong hệ điều hành Windows.
* Sử dụng giao diện dòng lệnh (Command Line Interface - CLI) đây là giao diện truyền thống của Linux, tất cả các việc điều khiển đều phải thông qua việc nhập các dòng lệnh. Ưu điểm của phương thức này là nhanh gọn và xử lý được nhiều công việc, tuy nhiên khuyết điểm của nó là việc học và ghi nhớ các câu lệnh là khá khó khăn.

Để giao tiếp với shell qua dòng lệnh ta cần một chương trình giao tiếp, trong Raspberry Pi chương trình này là LxTerminal.

* 1. **Cách thức biên dịch và thực thi chương trình.**

Các chương trình thực thi trên Linux có thể được viết bởi nhiều ngôn ngữ như C, Fortran, Pascal, Assembly, Perl, … Các chương trình này tồn tại ở hai dạng : thực thi (executable) và dạng mã nguồn (source code). Để có thể thực thi được các chương trình, ta cần phải biên dịch các mã nguồn của chương trình thành các tập tin thực thi. Quá trình biên dịch này sẽ do các bộ biên dịch (compiler) đảm nhiệm.

* + 1. ***Trình biên dịch***

Trong hầu hết các phiên bản Linux, trình biên dịch được mặc định cài sẵn là GCC (GNU Compiler Collection), đây là một bộ các trình biên dịch có khả năng biên dịch nhiều ngôn ngữ khác nhau như C (gcc), C++ (g++), Fortran (gfortran),…

Tên gốc của GCC là GNU C Compiler do ban đầu nó chỉ hỗ trợ dịch ngôn ngữ lập trình C. Phiên bản đầu tiên GCC 1.0 được phát hành vào năm 1987, sau đó được mở rộng hỗ trợ dịch C++ vào tháng 12 cùng năm đó. Sau đóm GCC được phát triển cho các ngôn ngữ lập trình như Fortran, Pascal, Objective C, Java, Ada, …

* + 1. ***Biên dịch với thư viện***

Thư viện là các tập tin chứa các đoạn mã lệnh và dữ liệu (thường ở dạng mã nhị phân) được tổ chức thành các hàm hay các lớp nhằm cung cấp các dịch vụ, chức năng nào đó cho chương trình chạy trên Raspberry Pi. Có 3 loại thư viện trong Linux gồm : tĩnh (static), động (dynamic) và chia sẻ (shared).

Khi biên dịch một chương trình đang ở dạng mã nguồn sang dạng thực thi thì nhiều hàm chức năng của chương trình được liên kết từ các thư viện.

Một chương trình đã được biên dịch và thực thi một cách hoàn toàn độc lập được coi là được liên kết tĩnh (static linking) bởi vì nó không còn phụ thuộc vao sự tồn tại của thư viện chứa các đoạn mã nguồn tạo nên chương trình đó nữa. Các chương trình được liên kết tĩnh có một số điểm hạn chế như:

* Chương trình sẽ có kích thước lớn hơn, chiếm dụng nhiều bộ nhớ do phải bao gồm các đoạn mã của thư viện được liên kết trong chương trình.
* Gây ra sự lãng phí bộ nhớ RAM nếu chương trình đang chạy đồng thời chứa các đoạn mã giống nhau trong cùng một thư viện.

Để khắc phục 2 nhược điểm trên, thay vì liên kết tĩnh, các chương trình sẽ được liên kết động (dynanmic linking) tức là:

* Bản thân các chương trình này khi được lưu trữ ở bộ nhớ không chứa các đoạn mã trong thư viện mà chỉ chứa khai báo tham khảo tới đoạn mã đó. Điều này giúp giảm kích cỡ của chương trình.
* Khác với liên kết tĩnh có việc liên kết tới thư viện diễn ra tại thời điểm biên dịch, trong liên kết động việc liên kết giữa các tập tin thực thi của chương trình với thư viện chỉ diễn ra tại thời điểm chạy chương trình (runtime). Quá trình gắn kết này do bộ liên kế (linker) đảm nhiệm giúp cho phép nhiều chương trình có thể sử dụng chung thư viện trong bộ nhớ.

**CHƯƠNG IV. TIN NHẮN SMS, MODULE SIM900 VÀ TẬP LỆNH AT**

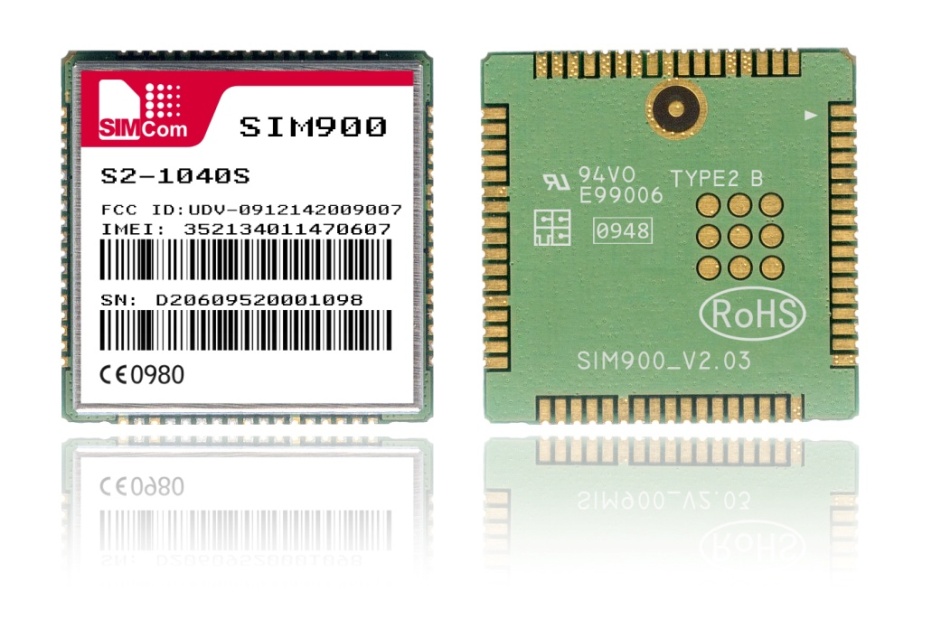
* 1. **Tin nhắn SMS**

SMS là từ viết tắt của Short Message Service. Đó là một công nghệ cho phép gửi và nhận các tín nhắn giữa các điện thoại với nhau. Dữ liệu có thể được lưu trữ bởi một tin nhắn SMS là rất giới hạn. Một tin nhắn SMS có thể chứa 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.

Tin 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.

* 1. **Module Sim900**
     1. ***Tổng quan về module sim900***

****

*Hình 20. Module sim900*

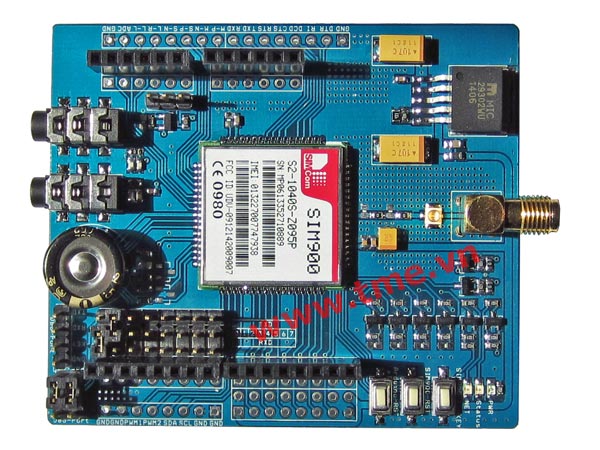
Sim900 là một module GSM/GPRS cực kỳ nhỏ gọn, được thiết kế cho thị trường toàn cầu. Sim900 hoạt động được ở 4 băng tần GSM 850MHz, EGSM 900MHz, DCS 1800MHz và PCS 1900MHz như là một loại thiết bị đầu cuối với một Chip xử lý đơn nhân đầy sức mạnh, tăng cường các tính năng quan trọng dựa trên nền vi xử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

* + 1. ***Sơ đồ chân***



*Hình 21. Sơ đồ chân của Module Sim900*

* + 1. ***Module sim icomsat v1.1 (Arduino Shield)***

**

*Hình 22. Module sim900 icomsat v1.1*

Icomset v1.1 là một board mở rộng GSM/GPRS cho arduino, sử dụng module Sim900 Quad-band 850/900/1800/1900MHz GSM/GPRS. Nó được điều khiển bằng tập lệnh AT và hoạt động với nguồn từ 9-20V.

* + 1. ***Kết nối giữa Module Sim900 và vi điều khiển***



*Hình 23. Kết nối giữa Module sim và vi điều khiển*

* 1. **Khảo sát tập lệnh AT**
     1. ***Lệnh xóa tin nhắn***

at + cmgd=<index><cr>

<index>: vị trí ngắn nhớ tin nhắn muốn xóa.

Nếu lệnh thực hiện được trả về: “OK” thì quá trình xóa tin nhắn diễn ra thành công.

Nếu lệnh không thực hiện, chuỗi gửi về sẽ có dạng:

+CMS ERROR <err>

* + 1. ***Lệnh chọn định dạng tin nhắn***

at + cmgf=[<mode>]

<mode> 0 : PDU mode

1. : TEXT mode
   * 1. ***Lệnh liệt kê tin nhắn từ vùng nhớ ưu tiên***

at + cmgl=<stat>[<mode>]

<mode> 0 : bình thường

1 : Không chuyển trạng thái của tin nhắn

* + 1. ***Lệnh đọc tin nhắn***

At + cmgr=<index>[<mode>] <cr>

<index>: số nguyên, đó là vị trí ngắn nhớ chứa tin nhắn cần đọc.

<mode>: 0 : dạng dữ liệu PDU

1 : dạng text

Sau khi gửi lệnh chuỗi sẽ trả về:

<cr><lf>+CMGR: “REC UNREAD”,” + số điện thoại”,” Thời gian” <cr><lf>

<cr><lf>OK<cr><lf>

* + 1. ***Lệnh gửi tin nhắn***

at + cmgs= “số điện thoại” <cr>

chuỗi trả về có dạng: <cr><lf>+CMGS:62<cr><lf>

<cr><lf>OK<cr><lf>

* + 1. ***Nhận cuộc gọi***

Khởi tạo bằng lệnh: at + clip=1 khi có cuộc gọi đến, chuỗi sẽ trả về có dạng:

<cr><lf>RING<cr><lf>

<cr><lf>+CLIP:” số điện thoại”,129,””,0<cr><lf>

Từ chối nhận cuộc gọi bằng lệnh ath và chuỗi sẽ trả về có dạng:

<cr><lf>OK<cr><lf> => cuộc gọi kết thúc.

Nhận cuộc gọi bằng lệnh: ata và chuỗi sẽ trả về có dạng:

<cr><lf>OK<cr><lf>

Khi nhận cuộc gọi thành công, muốn kết thúc cuộc gọi:

* Số gọi đến gác máy trước.
* Chủ động gác máy bằng lệnh ath.
  + 1. ***Thực hiện cuộc gọi***

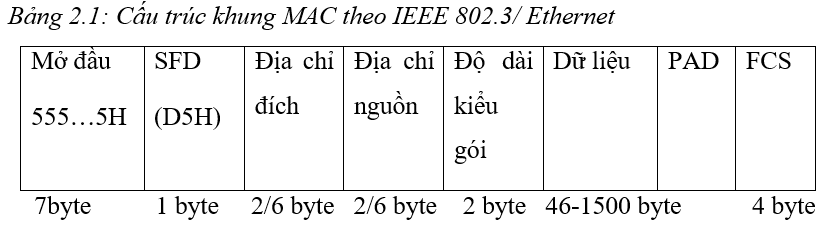
Thực hiện cuộc gọi bằng lệnh: athxxxx;<cr> xxxx là số điện thoại cần gọi.

Chuỗi trả về sẽ có dạng: <cr><lf>OK<cr><lf>

**CHƯƠNG V: ETHERNET VÀ HỌ GIAO THỨC TCP/IP**

* 1. **Tổng quan về Ethernet**
     1. ***Cấu trúc khung tin Ethernet***

Các chuẩn Ethernet đều hoạt động ở tầng Data Link trong mô hình 7 lớp OSI vì thế đơn vị dữ liệu mà các trạm trao đổi với nhau là các khung (frame). Cấu trúc khung Ethernet như sau:



* Preamble (mở đầu): trường này đánh dấu sự xuất hiện của khung bit, nó luôn mang giá trị 10101010. Từ nhóm bit này, phía nhận có thể tạo ra xung đồng hồ 10 Mhz.
* SFD (start frame delimiter): trường này mới thực sự xác định sự bắt đầu của 1 khung. Nó luôn mang giá trị 10101011.
* Các trường Destination và Source: mang địa chỉ vật lý của các trạm nhận và gửi khung, xác định khung được gửi từ đâu và sẽ được gửi tới đâu.
* LEN: giá trị của trường nói lên độ lớn của phần dữ liệu mà khung mang theo.
* FCS mang CRC (cyclic redundancy checksum): phía gửi sẽ tính toán trường này trước khi truyền khung. Phía nhận tính toán lại CRC này theo cách tương tự. Nếu hai kết quả trùng nhau, khung được xem là nhận đúng, ngược lại khung coi như là lỗi và bị loại bỏ.
  + 1. ***Cấu trúc địa chỉ Ethernet***

Mỗi giao tiếp mạng Ethernet được định danh duy nhất bởi 48 bit địa chỉ (6 octet). Đây là địa chỉ được  ấn  định khi sản xuất thiết bị, gọi là  địa chỉ MAC (Media Access Control Address ).  Địa chỉ MAC được biểu diễn bởi các chữ số hexa ( hệ cơ số 16 ). Ví dụ: 00:60:97:8F:4F:86 hoặc 00-60-97-8F-4F-86.Khuôn dạng địa chỉ MAC được chia làm 2 phần:

* 3 octet đầu xác định hãng sản xuất, chịu sự quản lý của tổ chức IEEE.
* 3 octet sau do nhà sản xuất ấn định.

Kết hợp ta lẽ có một địa chỉ MAC duy nhất cho một giao tiếp mạng Ethernet. Địa chỉ MAC được sử dụng làm địa chỉ nguồn và địa chỉ đích trong khung Ethernet.

* + 1. ***Các loại khung Ethernet***
       1. *Khung unicast*

Khung này được truyền tới một trạm xác định.  Tất cả các trạm trong phân đoạn mạng trên sẽ đều nhận được khung này nhưng:

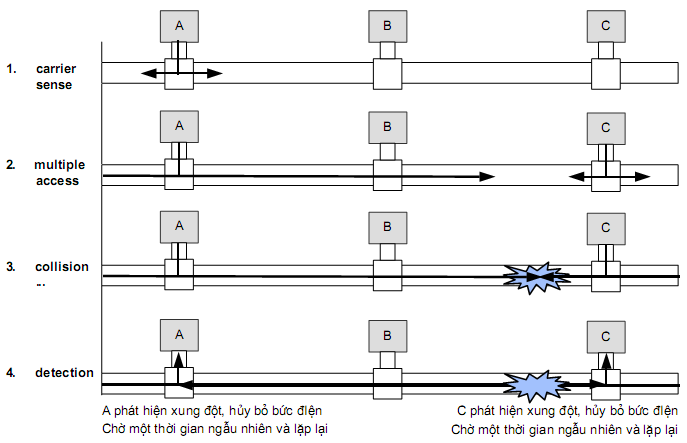
* Chỉ có trạm 2 thấy địa chỉ MAC đích của khung trùng với địa chỉ MAC của giao tiếp mạng của mình nên tiếp tục xử lý các thông tin khác trong khung.
* Các trạm khác sau khi so sánh địa chỉ sẽ bỏ qua không tiếp tục xử lý khung nữa.
  + - 1. *Khung broadcast*

Các khung broadcast có địa chỉ MAC đích là FF-FF-FF-FF-FF-FF. Khi nhận được các khung này, mặc dù không trùng với địa chỉ MAC của giao tiếp mạng của mình nhưng các trạm đều phải nhận khung và tiếp tục xử lý.

* + - 1. *Khung multicast*

Trạm nguồn gửi khung tới một số trạm nhất định chứ không phải là tất cả. Địa chỉ MAC đích của khung là địa chỉ đặc biệt mà chỉ các trạm trong cùng một nhóm mới chấp nhận các khung gửi tới địa chỉ này.

* + 1. *Truy cập bus sử dụng phương pháp CSMA/CD*



*Hình 24. Minh họa phương pháp CSMA/CD*

Theo phương pháp CSMA/CD, mỗi trạm đều có quyền truy nhập bus mà không cần một sự kiểm soát nào. Phương pháp được tiến hành như sau:

* Mỗi trạm đều phải tự nghe đường dẫn (carrier sense), nếu đường dẫn rỗi (không có tín hiệu ) thì mới được phát.
* Do việc lan truyền tín hiệu cần một thời gian nào đó, nên vẫn có khả năng hai trạm cùng phát tín hiệu lên đường dẫn. Chính vì vậy, trong khi phát thì mỗi trạm vẫn phải nghe đường dẫn để so sánh tín hiệu phát đi với tín hiệu nhận được xem có xảy ra xung đột hay không (collision detection).
* Trong trường hợp xảy ra xung đột, mỗi trạm đều phải hủy bỏ bức điện của mình, chờ một thời gian ngẫu nhiên và thử gửi lại.
  1. **Hoạt động của mạng Internet.**
     1. ***Các giao thức***

Các giao thức là tập các luật mà các máy tính phải tuân theo khi giao tiếp trên Internet.

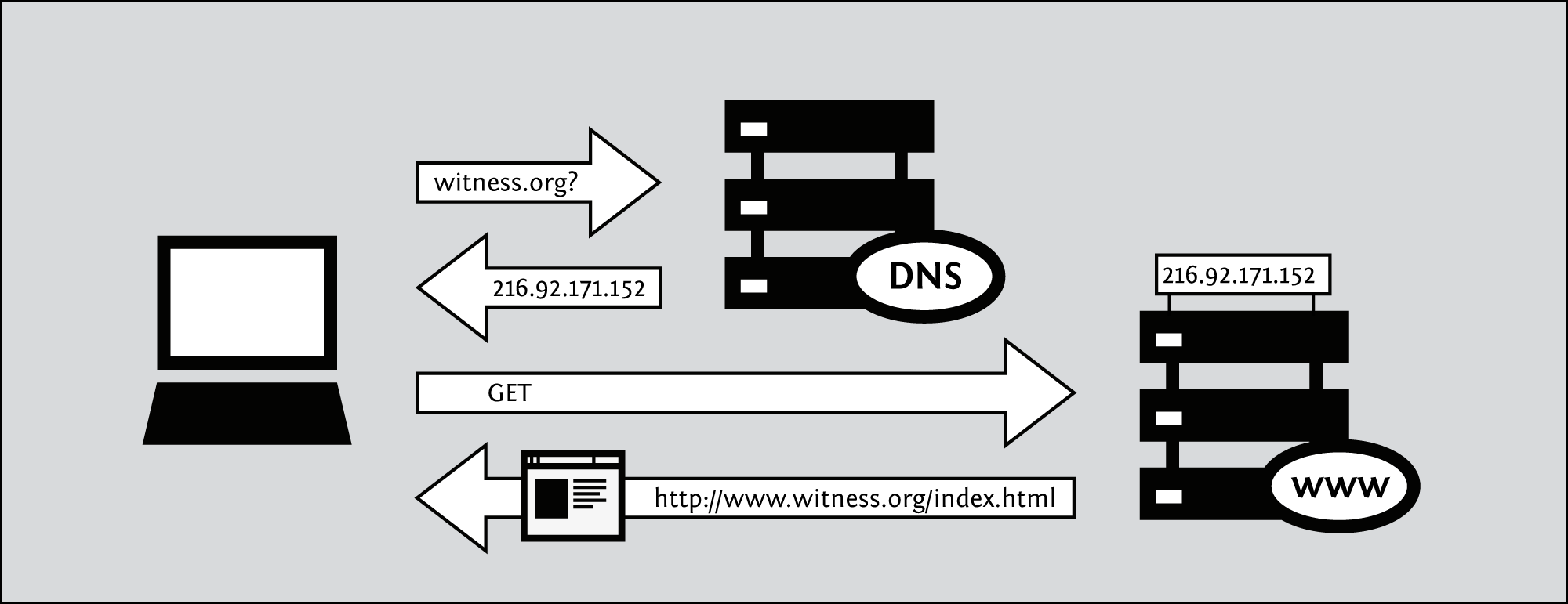
* Tranmission Control Protocol (TCP): thiết lập kết nối giữa hai máy tính để truyền tải dữ liệu, chia dữ liệu thành những gói nhỏ và đảm bảo việc truyền nhận dữ liệu. TCP là giao thức hướng kết nối (connection-oriented protocol)
* User Datagram Protocol (UDP): thiết lập kết nối nhanh nhưng không chắc chắn giữa các máy tính để truyền tải dữ liệu, cung cấp ít dịch vụ để khắc phục lỗi.
* Internet Protocol (IP):  điều chỉnh đường đi của những gói dữ liệu đường truyền nhận trên Internet. TCP là giao thức phi kết nối (connectionless  protocol).
* HTTP: cho phép trao đổi thông tin trên Internet.
* FTP: cho phép truyền nhận file trên Internet.
* SMTP: cho phép gửi thư điện tử trên Internet.
* POP3: cho phép nhận thư điện tử trên Internet.

TCP/IP được dùng làm giao thức chuẩn khi giao tiếp trên Internet vì nó   độc lập với nền của hệ thống (platform independent) và không có tổ chức nào có quyền sở hữu giao thức này.

* + 1. ***Tên miền và địa chỉ IP***

Internet là một mạng kết nối rộng lớn giữa các máy tính. Để xác định một máy tính trên mạng này, người ta dùng một con số gọi là địa chỉ IP. Địa chỉ IP gồm một tập 4  số nhỏ hơn 255 và được ngăn cách bởi các dấu ‘.’

Ví dụ: 41.212.196.197.



*Hình 25. Hệ thống DNS*

Tất cả các máy chủ Internet (server) như máy chủ các trang web, các thiết bị kết nối với internet cũng có địa chỉ IP riêng. Các địa chỉ IP này là dãy số dài và rất khó chọ việc nhớ chúng, đồng thời nó cũng thay đổi theo thời gian nên có các hệ thống được vận hành với mục đích giúp người dùng mạng có thể tới được nơi mình muốn trên Internet một cách dễ dàng. Những hệ thống đó được gọi là DNS (Domain Name System – hệ thống các tên miền), hệ thống này bao gồm các tập howpk các máy tính chuyên thực hiện cung cấp địa chỉ IP dưới hình thức là chữ viết mà con người có thể nhớ được.

* + 1. ***Mô hình khách chủ***

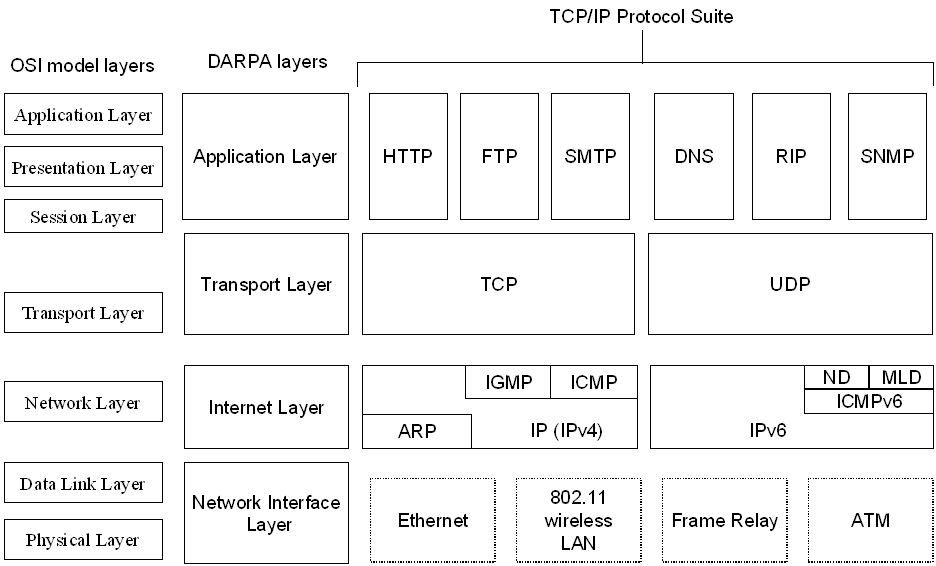
Internet dựa trên mô hình khách – chủ (client – server), trong đó dữ liệu được trao đổi thông qua các trang web. Trong mô hình client – server, mỗi máy tính được xác định bởi một địa chỉ Internet protocol (IP) và cả máy tính client, server cùng chấp nhận một giao thức chung để để giao tiếp với nhau.

Trong mô hình khách - chủ, máy khách (client computer) yêu cầu thông tin từ một máy chủ (server). Máy chủ chấp nhận yêu cầu và gửi thông tin về cho máy khách. Việc trao đổi thông tin này được diễn ra thông qua những trang web.

* 1. **Họ giao thức TCP/IP**

TCP/IP là viết tắt của Transmission Control Protocol / Internet Protocol (Giao thức điều khiển truyền thông / giao thức internet). Các tầng trong mô hình này là :

* Tầng ứng dụng (Application Layer).
* Tầng giao vận (Transport Layer).
* Tầng liên mạng (Internet Layer).
* Tầng giao tiếp mạng (Network Interface Layer).



*Hình 25. Cấu trúc họ giao thức TCP/IP*

Chức năng của các tầng:

* *Application Layer*: hỗ trợ các ứng dụng cho các giao thức tầng Host-to-host. Cung cấp giao diện cho người sử dụng mô hình TCP/IP. Các giao thức ứng dụng gồm: HTTP, TELNET, FTP, SMTP, …
* *Transprot Layer:* thực hiện những kết nối giữa hai máy chủ trên mạng bằng hai giao thức: TCP và UDP (User Datagramm Protocol).
* *Internet Layer:* giao thức IP cùng với các giao thức định tuyến RIP, OSPF tầng mạng cho phép kết nối một cách mềm dẻo và linh hoạt các loại mạng “vật lý” khác nhau như: Ethernet, Token Ring, X.25 … ánh xạ địa chỉ MAC-IP bằng giao thức ARP và RARP.
* *Network Interface Layer:*  cung cấp các phương tiện kết nối vật lý, bộ chuyển đổi cáp mạng, Card mạng, giao thức kết nối, giao thức truy cập đường truyền CSMA/CD, Tolen Ring, Tolen Bus .. Cung cấp các dịch vụ cho tầng internet.

**CHƯƠNG VI: THIẾT KẾ VÀ THI CÔNG**

* 1. **Sơ đồ khối**



Internet



**Raspberry Pi**



**Module Sim900**



Hệ thống tổng đài sinh viên

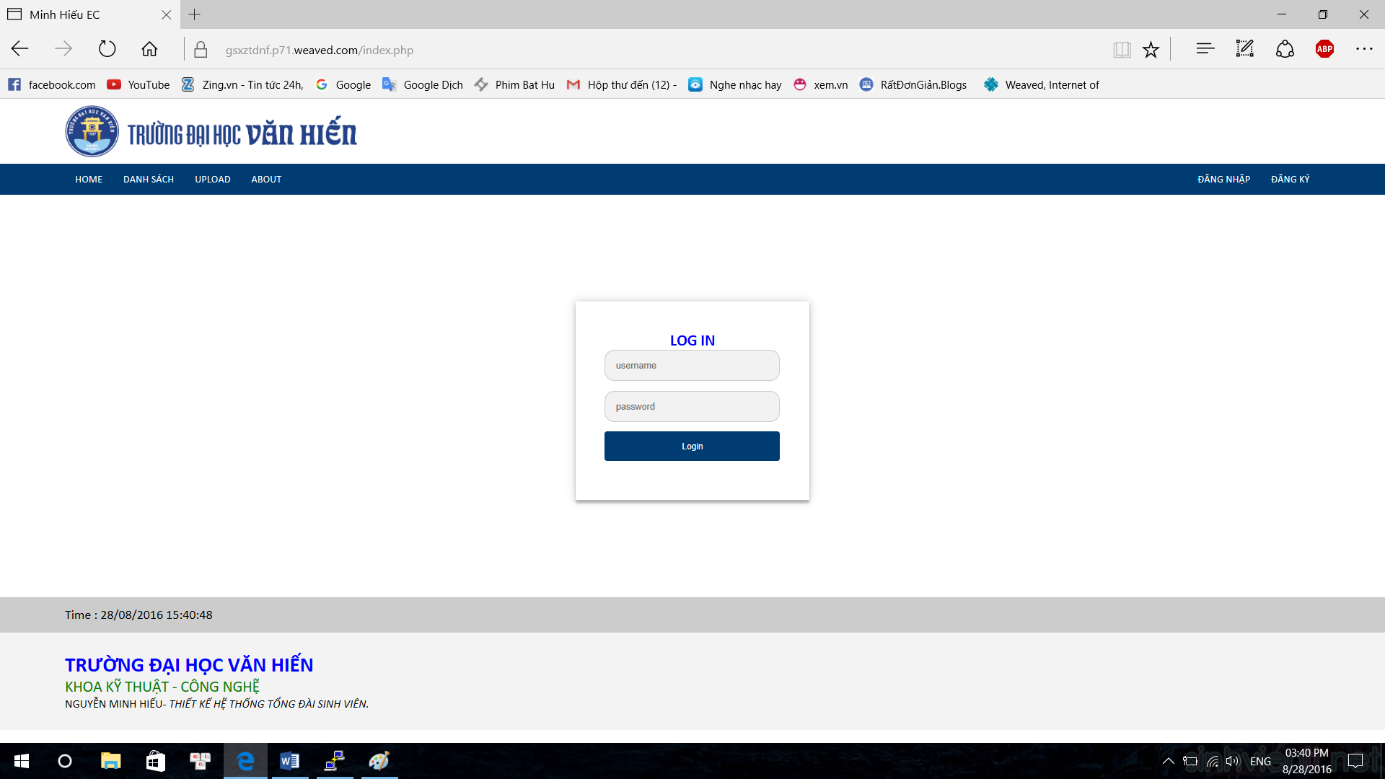
Sử dụng Raspberry Pi và module sim900



Hệ thống sử dụng:

* Một board Raspberry Pi model B được kết nối với Internet thông qua một router wifi: có chức năng xử lý dữ liệu, điều khiển module sim đồng thời đóng vai trò một webserver.
* Module sim900 icomsat v1.1: Tiếp nhận tin nhắn và gửi dữ liệu về cho Raspberry Pi xử lý, nhận tín hiệu điều khiển và gửi tin nhắn sms đến các số điện thoại được yêu cầu.
  1. **Sản phẩm**

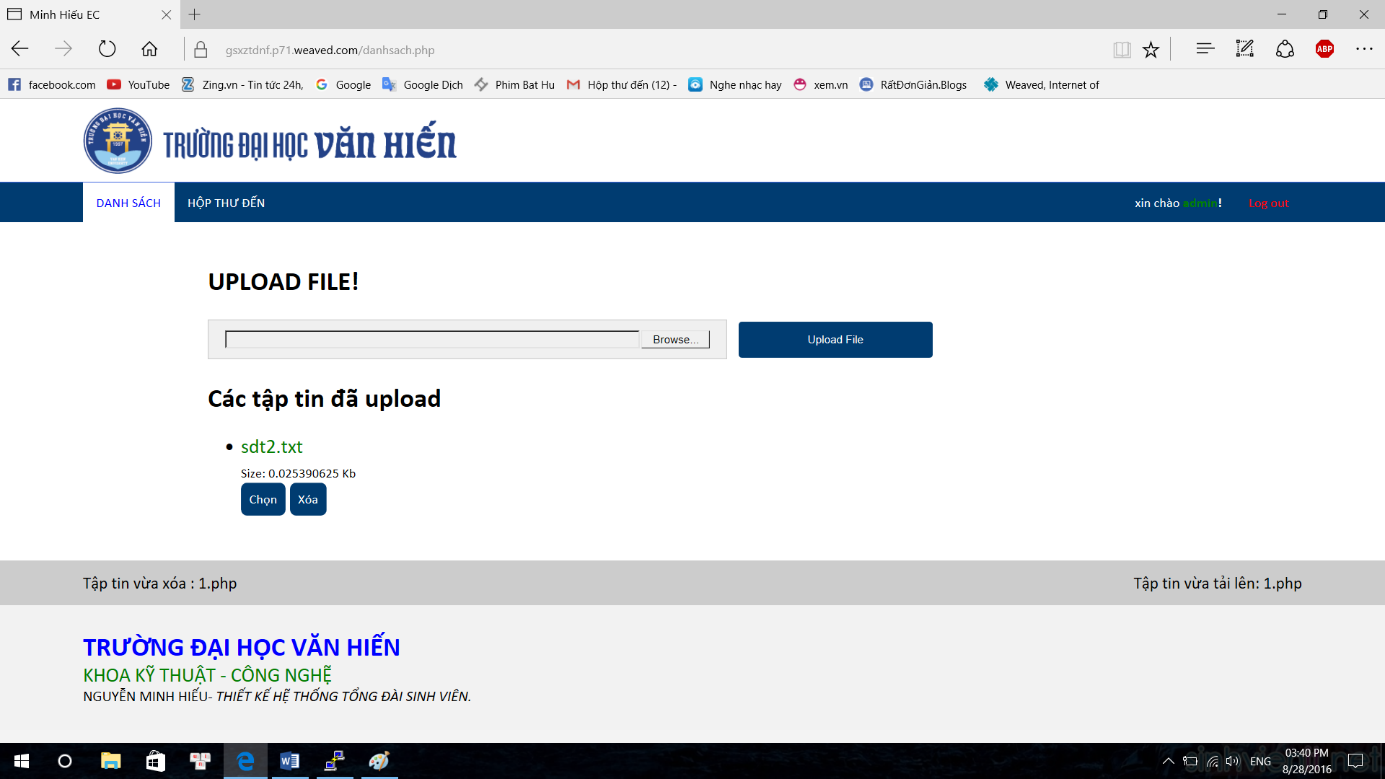
Hệ thống tổng đài sinh viên sử dụng Raspberry Pi và module sim900, sau khi khởi động sẽ có một địa chỉ DNS: <https://gsxztdnf.p71.weaved.com/> cho phép thao tác nhận tin nhắn và gửi tin nhắn trực tiếp trên giao diện web thông qua mạng Internet và LAN.



*Hình 26. Giao diện đăng nhập của hệ thống*

Sau khi truy cập vào địa chỉ web ở trên, hệ thống sẽ yêu cầu đăng nhập để tiếp tục sử dụng những chức năng như đọc tin nhắn và gửi tin nhắn. Nếu đăng nhập không đúng hoặc bỏ qua thì hệ thống sẽ báo lỗi trực tiếp và không truy cho phép truy cập sâu vào hệ thống. Điều này làm tăng tính bảo mật hơn và chống được ý định phá hoại của người khác đối với hệ thống.

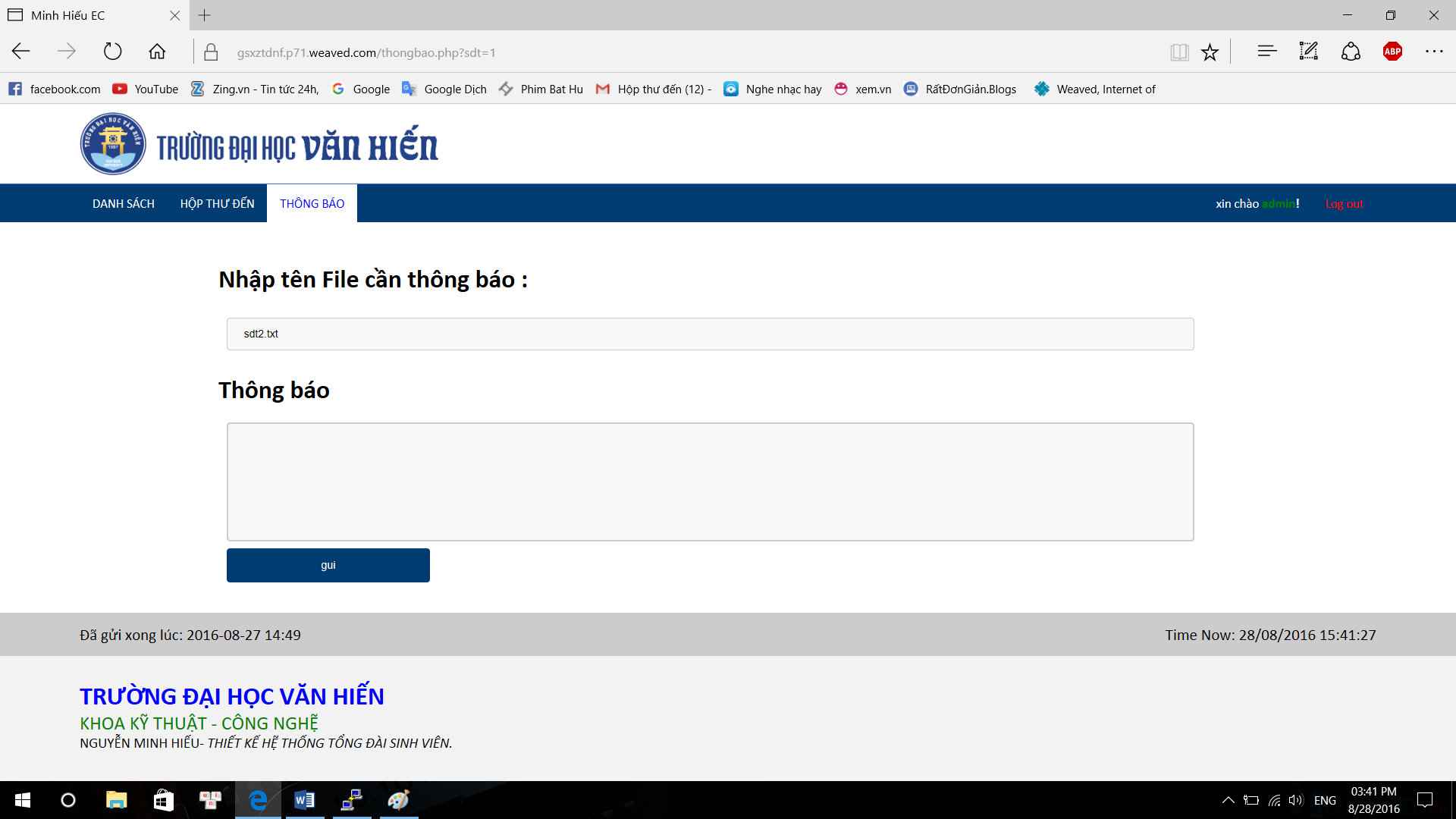
Sau khi đăng nhập thành công, một giao diện web mở ra cho phép chúng ta thao tác được những chức năng như đã được lập trình sẵn như hình 27. Nếu không đăng nhập và truy cập thẳng vào liên kết này thì hệ thống sẽ tự động trả về địa chỉ đăng nhập như hình 26.



*Hình 27. Giao diện web sau khi đăng nhập thành công*

Chức năng chính của hệ thống là cho phép gửi tin nhắn sms, thông báo đến nhiều số điện thoại có trong một file text (Tạo một file text chứa số những số điện thoại và chọn chức năng upload).

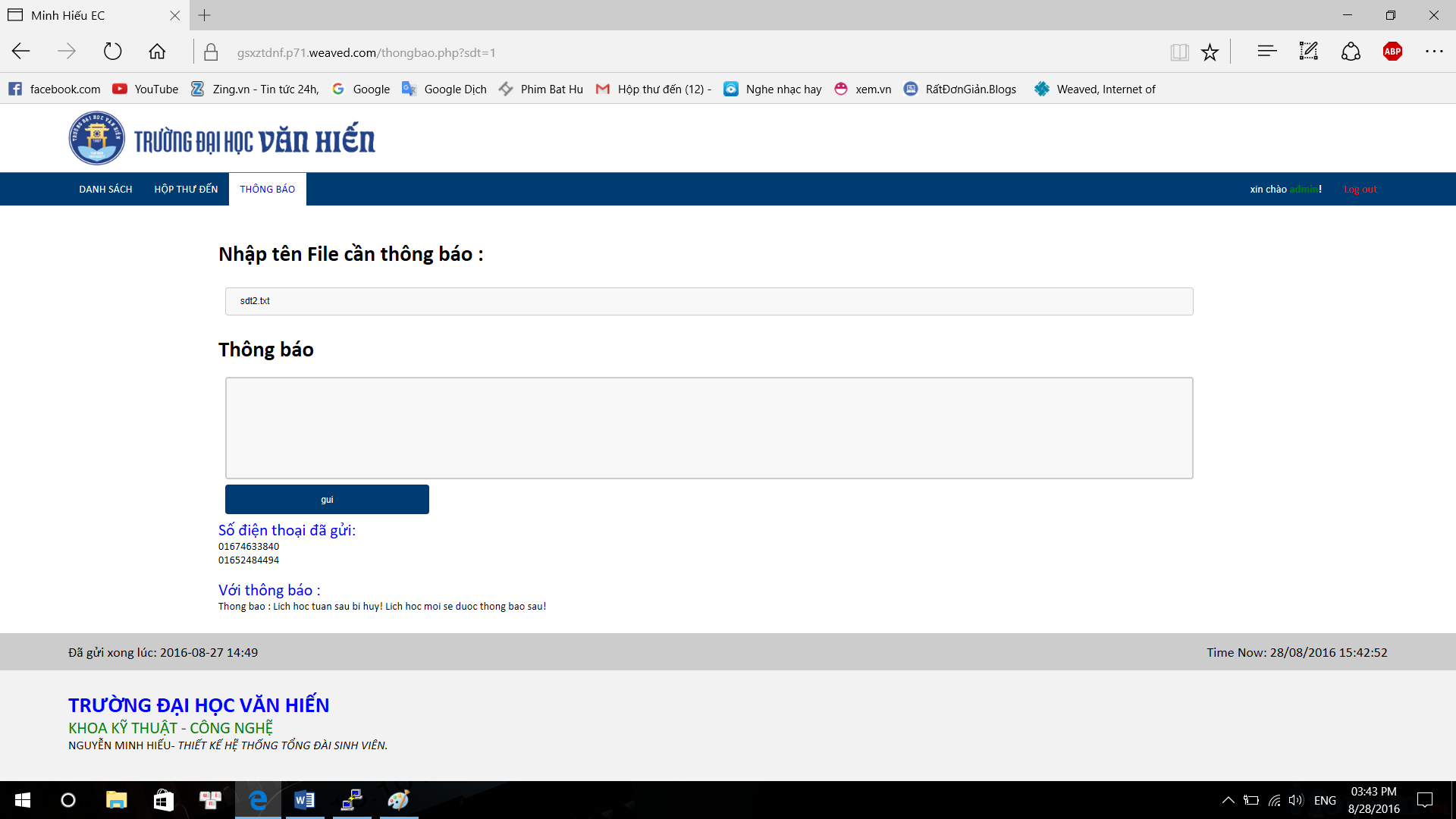
Để gửi tin nhắn cho nhiều số, chỉ cần chọn file chứa những số điện muốn thông báo bằng cách nhấn vào button “Chọn”. Hoặc xóa những tập tin không còn sử dụng nữa bằng cách nhấn vào buton “xóa”.



*Hình 28. Thao tác gửi tin nhắn cho nhiều số điện thoại.*

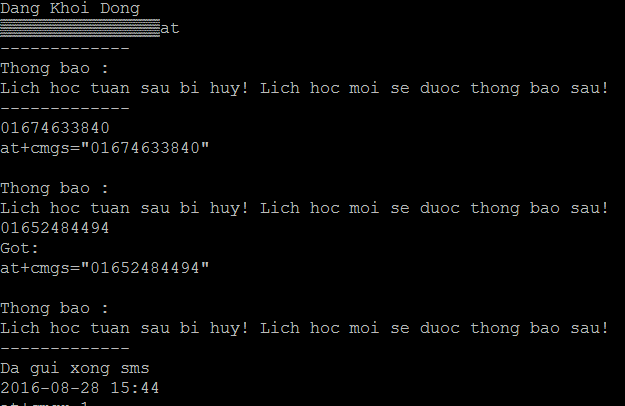
Sau khi chọn tệp tin muốn gửi thông báo, thì một giao diện sẽ hiện ra yêu cầu nhập thông báo để tiến hành gửi tin nhắn.

Sau khi nhập thông báo xong và nhấn nút gửi hệ thống sẽ tự động gửi tin nhắn đến những tin nhắn đến những số điện thoại có trong tệp như hình 30.



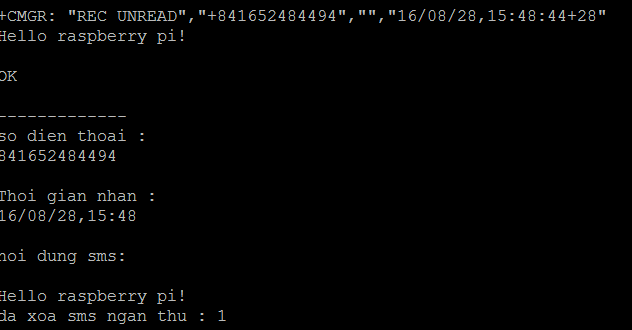
*Hình 30. Sau khi gửi tin nhắn cho tất cả số điện thoại trong tệp tin*

Raspberry Pi sẽ tự động nhận lệnh và gửi tin nhắn sau khi nhấn nút gửi (đã nhập thông báo) như hình 31.



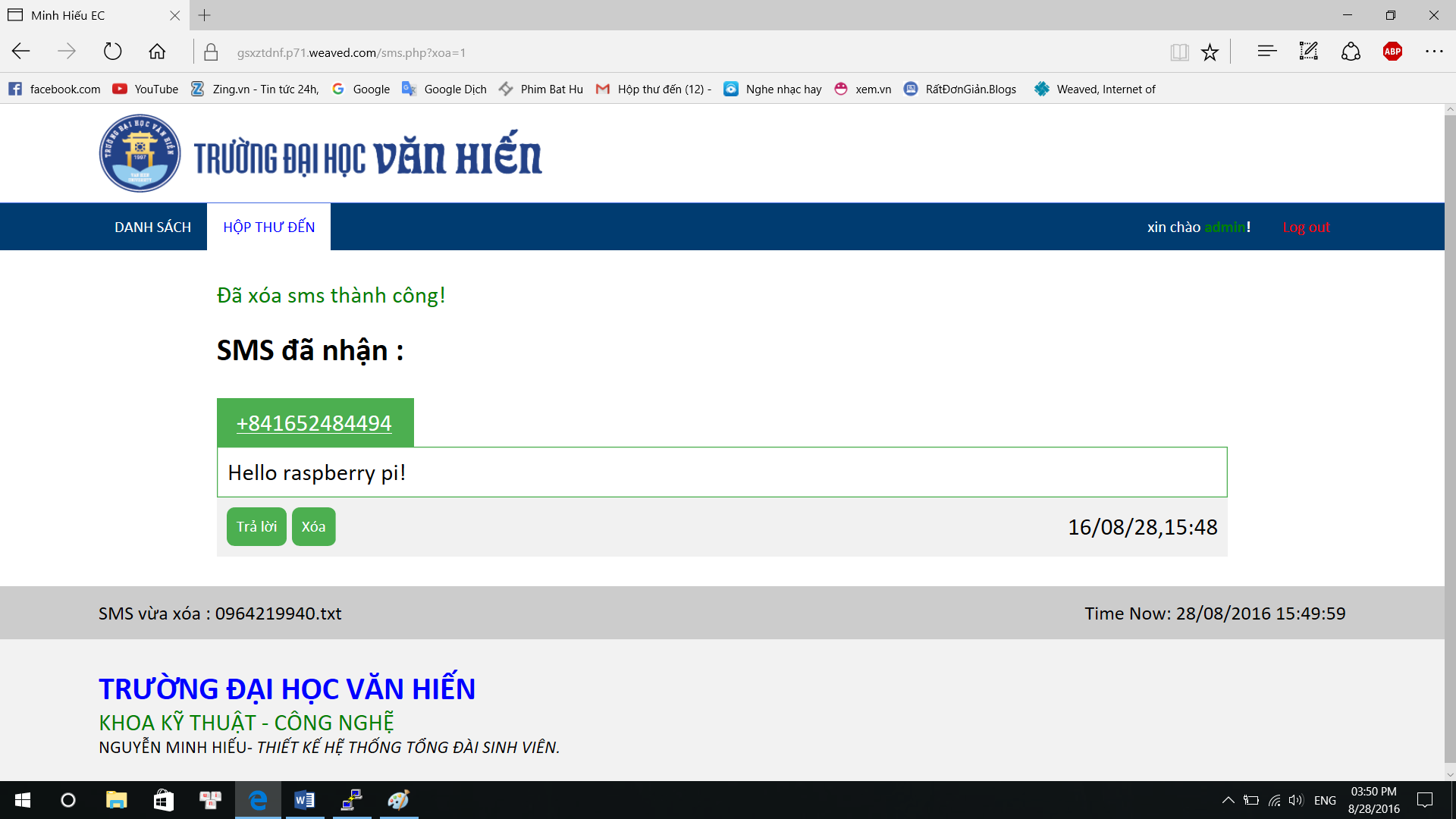
*Hình 31. Giao diện của Raspberry Pi khi nhận được lệnh gửi tin nhắn*

Ngoài chức năng gửi tin nhắn cho nhiều số điện thoại, hệ thống còn có chức năng đọc nội dung tin nhắn sms được gửi đến và hiển thị trực tiếp lên website, từ đó chúng ta có thể dễ dàng quan sát, và trả lời những tin nhắn được người khác gửi đến(hình 32).



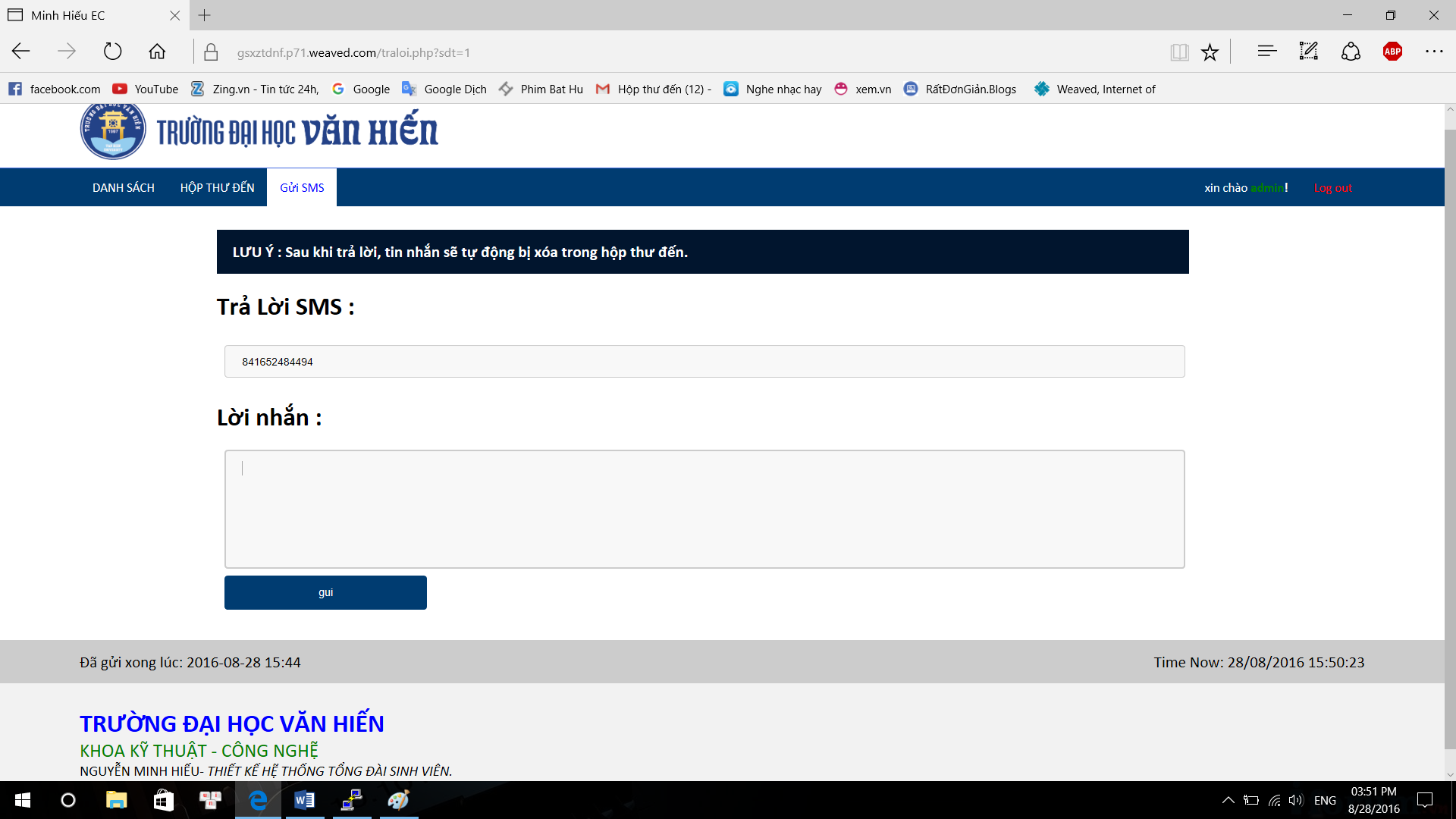
*Hình 32. Khi có tin nhắn gửi đến module sim900*

Sau khi có tin nhắn được gửi đến số điện thoại trong module sim900, Raspberry Pi sẽ đọc dữ liệu gửi đến từ module sim900 và tiến hành xử lý và hiện thị lên giao diện web.

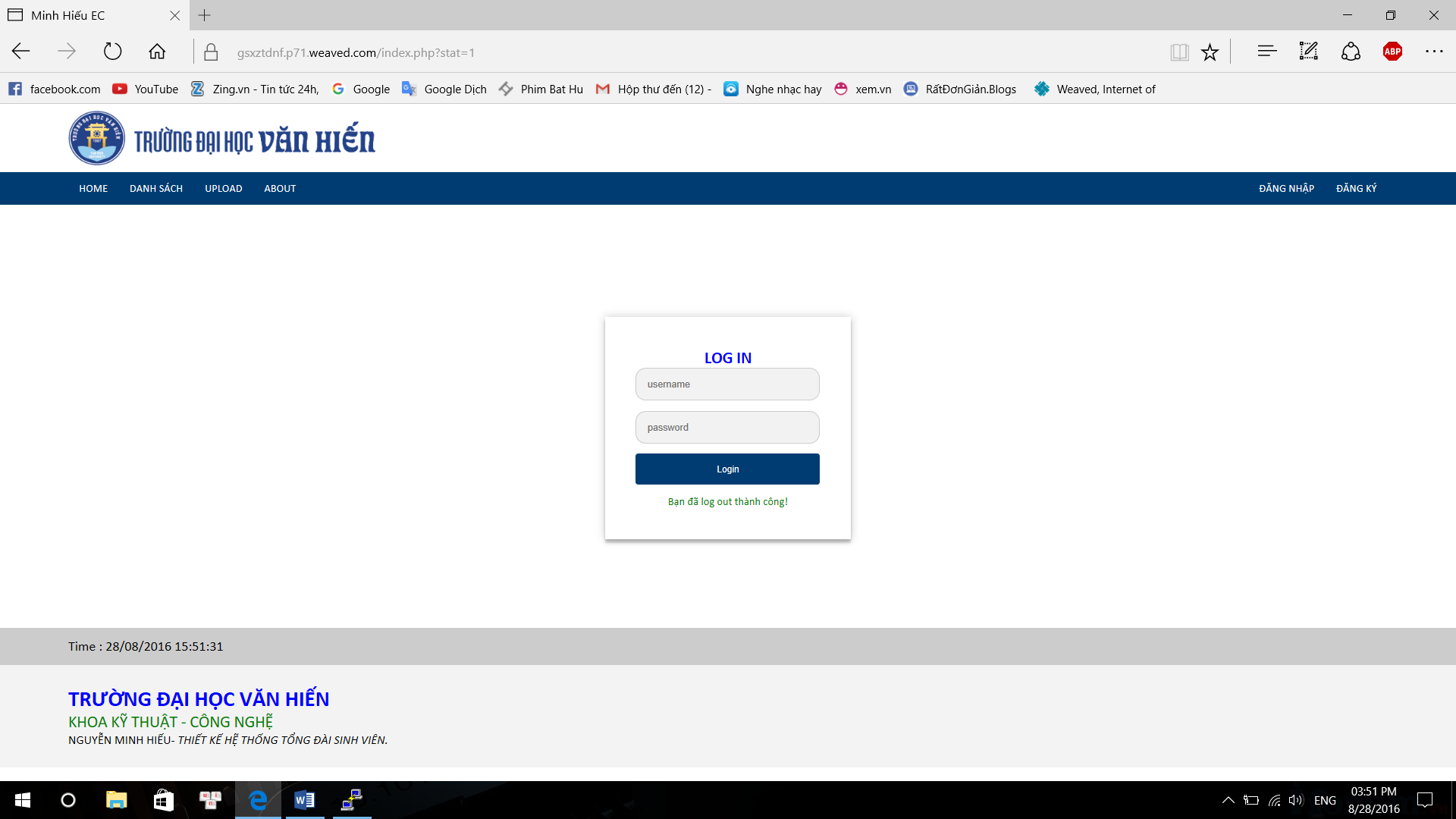


*Hình 33. Hộp thư đến của hệ thống.*

Trong hộp thư đến, có thể quản lý dễ dàng những tin nhắn được gửi đến, có thể trả lời những tin nhắn gửi đến và xóa những tin nhắn rác không cần thiết bằng cách nhấn vào button “xóa”.



*Hình 34. Giao diện hệ thống khi trả lời tin nhắn được gửi đến*



*Hình 35. Sau khi không dùng và tiến hành đăng xuất tài khoản*

* 1. **Code**
     1. ***Gửi SMS***

*import serial*

*from serial import Serial*

*from time import sleep*

*ser = Serial("/dev/ttyAMA0",2400, timeout=1)*

*data2 = ser.write("at\r")*

*data = ser.read(9999)*

*if len(data) > 0:*

*print 'Got:', data*

*sleep(0.5)*

*sdt = "01638800267"*

*ser.write('at+cmgs="01674633840"'+'\r\n')*

*rcv = ser.read(9999)*

*print rcv*

*sleep(1)*

*rcv = ser.read(9999)*

*print rcv*

*ser.write("\x1A")*

*sleep(2)*

*rcv = ser.read(9999)*

*print rcv*

*ser.close()*

* + 1. ***Đọc sms***

*import serial*

*from serial import Serial*

*from time import sleep*

*ser = Serial("/dev/ttyAMA0",2400, timeout=1)*

*ser.read(9999)*

*sleep(0.5)*

*ser.write('at+cmgr=1'+'\r\n')*

*rcv = ser.read(9999)*

*sleep(1)*

*print rcv*

*sleep(1)*

*kt = rcv[13:15]*

*if kt == "OK":*

*print "Khong co SMS"*

*else:*

*#print rcv[0:34]*

*print "-------------"*

*print "so dien thoai :"*

*print rcv[35:47] #cat so dien thoai can dung*

*print "Thoi gian nhan :"*

*print rcv[53:67]*

*print "noi dung sms:"*

*a = rcv[74:]*

*print a[:-8]*

*sleep(1)*

*ser.close()*

* + 1. ***Xóa SMS***

*import serial*

*from serial import Serial*

*from time import sleep*

*ser = Serial("/dev/ttyAMA0",2400, timeout=1)*

*data2 = ser.write("at\r")*

*data = ser.read(2)*

*sleep(1)*

*print 'Got:', data*

*ser.write('at+cmgda="DEL ALL"\r\n')*

*rcv = ser.read(9999)*

*ser.close()*

**CHƯƠNG VII. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

* 1. **Kết luận**

Sau một thời gian tìm hiểu và tiến hành xây dựng hệ thống tổng đài sinh viên dựa trên board Raspberry Pi và module sim900, về cơ bản em đã hoàn thành được một số mục tiêu đề ra trước khi thực hiện đồ án này. Hệ thống có thể gửi tin nhắn sms đến nhiều số điện thoại để thông báo về một sự kiện hay một lời nhắn nào đó. Hệ thống còn có thể đọc được tin nhắn và hiển thị lên Web để chúng ta có thể biết và trả lời những tin nhắn đó.

Ngoài ra hệ thống được kết nối với Internet nên có thể thao tác với nó ở bất kỳ nơi nào. Với một chiếc điện thoại, ipad, laptop hoặc PC có kết nối internet ta có thể đọc được những tin nhắn được gửi đến, trả lời tin nhắn một cách dễ dàng.

Hệ thống thích hợp sử dụng trong trường học. Học sinh có thể gửi tin nhắn để tra cứu những thông tin mình cần. Giảng viên có thể thông báo lịch học, lịch nghỉ hoặc bất cứ thông tin gì cho sinh viên. Sinh viên có thể biết những thông tin đó mà không cần túc trực trên trang web của trường như trước đây.

* 1. **Hướng Phát triển**
* Tối ưu hóa hơn thuật toán, để hệ thống có thể vận hành ổn định.
* Tăng thêm tính bảo mật, thêm tính năng nhiều tài khoản đăng nhập, lịch sử những lần thao tác trên hệ thống.
* Kết hợp với cơ sở dữ liệu để có thể tự động trả lời khi tin nhắn khi tin nhắn gửi đến đúng mẫu.
* Tạo, phát triển ứng dụng trên hệ điều hành android cho phép thao tác dễ và nhanh hơn.

**CHƯƠNG VIII. TÀI LIỆU THAM KHẢO**

* *Giới thiệu ngôn ngữ lập trình Python – Đặng Nguyên Phương.*
* *RaspberryPi.vn*
* *Raspberrypi.org*