

PHƯƠNG PHÁP ĐIỀU KHIỂN THIẾT BỊ BẰNG TIẾNG NÓI

Đỗ Văn Minh, Nguyễn Minh Sơn, Phan Thiện Phước

Khoa Công nghệ thông tin, Trường Đại học Lạc Hồng
minhit99@gmail.com, phanphuoc93pp@gmail.com, nmson@lhu.edu.vn

TÓM TẮT - Bài báo trình bày phương pháp nhận dạng tiếng nói tiếng Việt cho việc điều khiển thiết bị trong ngôi nhà thông minh. Thực hiện trên máy tính nhỏ nhằm tăng khả năng triển khai của hệ thống. Việc nhận dạng sử dụng CMU Sphinx, quá trình huấn luyện các từ điều khiển “Robot”, “Đèn”, “Quạt”,... Hệ thống sử dụng Raspberry Pi và dùng chip Atmega 328 làm bộ xử lý trung tâm (MCU-Micro Control Unit) để truyền, nhận tín hiệu và điều khiển thiết bị. Mặc dù phương pháp điều khiển thiết bị đã có rất nhiều như: smart phone, máy tính để bàn, laptop kết hợp với công nghệ khác, nhưng dùng tiếng nói để điều khiển thiết bị còn nhiều khía cạnh cần giải quyết. Phương pháp mà bài báo đưa ra tối ưu và triển khai dễ dàng hơn. Chi phí cho hệ thống được giảm đáng kể. Tuy nhiên vẫn còn một số hạn chế gặp phải chưa giải quyết được: xử lý nhiễu (noise), quá trình huấn luyện.

Từ khóa - CMU Sphinx, Raspberry Pi, GPIO, Arduino.

I. PHẦN MỞ ĐẦU

Công nghệ ngày càng phát triển, trong thời đại công nghệ thông tin, các chương trình giao tiếp người dùng ngày càng đòi hỏi sự thân thiện và hiệu năng mạnh mẽ. Có nhiều cách để con người giao tiếp với máy tính trong đó có giao tiếp bằng tiếng nói. Nhu cầu giao tiếp với thiết bị máy bằng tiếng nói trở nên cần thiết, đó là phương thức giao tiếp thông minh và tự nhiên nhất. Trong những năm gần đây, công nghệ này đã có mặt trên rất nhiều thiết bị và chủng loại, từ máy tính đến điện thoại di động, đồ chơi công nghệ và các thiết bị nhúng khác, các thiết bị ngày càng trở nên nhỏ hơn về kích thước [1]. Tuy nhiên các thiết bị càng nhỏ cũng làm giới hạn về chức năng. Tiếng nói có khả năng điều khiển và tương tác phức tạp với hệ thống nhúng [2].

Nhận diện tiếng nói được phân loại như nhận diện các từ đã được nối với với nhau và nhận biết từng từ một cách độc lập. Đối với hệ thống nhúng thì sử dụng nhận diện từng từ độc lập với nhau có hiệu quả hơn cả. Thông thường, nhận dạng tiếng nói là một loại mẫu nhận dạng dựa trên huấn luyện và nhận dạng [1]. Phương pháp sử dụng là HMM (Hidden Markov Model) và thư viện sử dụng là bộ nhận dạng HTK. Tín hiệu tiếng nói người dùng được lấy bởi micro USB đã được kết nối với hệ thống. Để xử lý tín hiệu nhận dạng bài báo sử dụng bộ thư viện nhận dạng tiếng nói CMU Sphinx nhận dạng các câu lệnh điều khiển thiết bị bằng tiếng Việt. Chương trình điều khiển được viết bằng ngôn ngữ Python và được biên dịch trên máy tính nhúng Raspberry Pi để điều khiển các thiết bị đèn, quạt, tivi trực tiếp qua hàng chân GPIO (General Purpose Input/ Output) trên Raspberry Pi. Sau khi chuỗi tín hiệu các từ được nhận diện. Các từ đã được nhận dạng chuyển thành dạng văn bản sẽ truyền đến Arduino sử dụng bộ phát/thu sóng vô tuyến RF SI4463. Tại Arduino sẽ thực thi lệnh để điều khiển các thiết bị điện.

Bài báo này giải quyết từng bước cơ bản trong mô hình Markov ẩn, mà một số đề tài trước còn bị giới hạn [1]. Bài báo có sử dụng công cụ Sphinx. Sphinx là một nền tảng mã nguồn mở, đang được rất nhiều chuyên gia sử dụng làm công cụ nhận dạng tiếng nói. Sphinx là một công cụ nhận dạng tiếng nói rất mạnh mẽ, có tính môđun hóa cao. Mỗi thành phần biểu diễn một môđun có thể dễ dàng được thay thế, cho phép các nhà nghiên cứu thực nghiệm một môđun khác mà không cần phải thay đổi các thành phần còn lại của hệ thống. Sphinx đã được sử dụng trong nhiều hệ thống nhận dạng như: cairo, jvoicexml,... các chương trình điều khiển như SpeechLion, VoiceKey...

Sự kết hợp giữa phần cứng và các thành phần đã tạo ra một hệ thống nhỏ gọn có thể điều khiển bật/tắt một số thiết bị điện gia dụng như: đèn, quạt và tivi. Đặc biệt là truyền tín hiệu điều khiển thiết bị không dây nhằm tạo sự linh động, dễ dàng lắp đặt.

II. CÁCH THỨC NHẬN DẠNG TIẾNG NÓI

A. Nhận dạng tiếng nói

Nhận dạng tiếng nói là một quá trình phức tạp bao gồm nhiều khâu biến đổi. Tín hiệu tiếng nói phát ra là tương tự. Từ quá trình lấy mẫu, lượng tử hóa và mã hóa để thu được tín hiệu số. Các mẫu tín hiệu này được trích chọn đặc trưng. Những đặc trưng này sẽ là đầu vào của quá trình nhận dạng. Hệ thống nhận dạng sẽ đưa ra kết quả nhận dạng. Tín hiệu tiếng nói đầu tiên được tiền xử lý và rút trích đặc trưng. Kết quả thu được sau quá trình này là tập các đặc trưng âm học (acoustic features). Để có thể thực hiện việc so sánh với các tham số đầu vào của hệ thống nhận dạng, trước hết hệ thống phải được huấn luyện và xây dựng các đặc trưng.

Trong quá trình huấn luyện, hệ thống dùng các vector đặc trưng được đưa vào để ước lượng, tính toán các tham số cho các mẫu tham khảo. Một mẫu tham khảo chính là bản mẫu dùng để so sánh và nhận dạng, các mẫu tham khảo này mô phỏng cho một từ, một âm tiết, hoặc một âm vị. Trong quá trình nhận dạng, dãy các vector đặc trưng được so sánh với các mẫu tham khảo. Sau đó, hệ thống tính toán độ tương đồng của dãy vector đặc trưng và mẫu tham khảo. Việc tính toán độ tương đồng được thực hiện bằng cách áp dụng các thuật toán đã được chứng minh hiệu quả như thuật toán Vitertbi (trong Markov ẩn). Mẫu có độ tương đồng cao nhất là kết quả của quá trình nhận dạng. Có các loại hệ

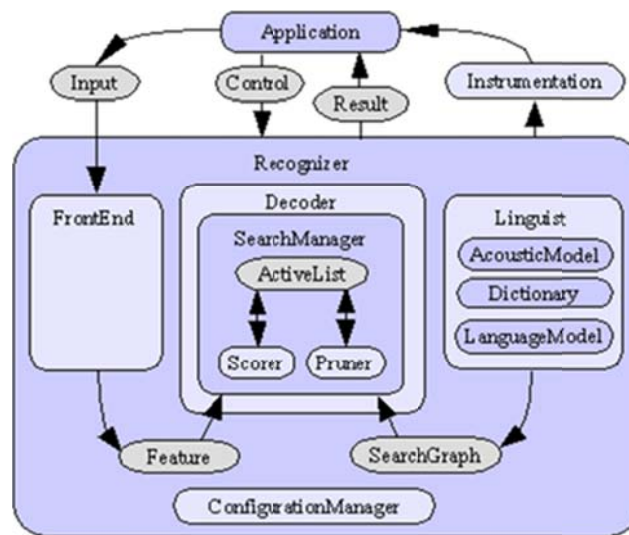
thông nhận dạng là nhận dạng từ liên tục, nhận dạng từ tách biệt, nhận dạng phụ thuộc người nói và độc lập người nói. Một số phương pháp nhận dạng như: phương pháp âm học - ngữ âm học, phương pháp nhận dạng mẫu, và phương pháp trí tuệ nhân tạo.

B. Nhận dạng tiếng nói bằng thư viện CMU Sphinx

Sphinx [3] là một bộ thư viện nhận dạng tiếng nói mạnh mẽ và được sử dụng rất nhiều trong các ứng dụng trong cuộc sống. Đó là nhờ các đặc điểm:

- Hỗ trợ nhận dạng tiếng nói ở chế độ trực tiếp hoặc chia lô, có khả năng nhận dạng tiếng nói rời rạc và liên tục.
- Là một hệ thống nhận dạng đồ sộ nhưng có khả năng tháo lắp rất linh động. Hỗ trợ sẵn đầy đủ các tính năng đáp ứng nhu cầu nhận dạng như xây dựng các bộ lọc, các hàm cửa sổ, các phép biến đổi,...
- Hỗ trợ nhiều mô hình ngôn ngữ dạng ASCII và các phiên bản nhị phân của unigram, bigram, trigram, Java Speech API Grammar Format (JSGF) và ARPAformat FST grammars.
- Xây dựng sẵn các thuật toán tìm kiếm tối ưu (breath first, word pruning), dễ dàng tinh chỉnh cho phù hợp với nhu cầu nhận dạng.

Kiến trúc bộ thư viện nhận dạng gồm các thành phần sau:



Hình 1. Kiến trúc tổng quát của Sphinx

Bộ ngoại vi (FrontEnd): Xử lý tín hiệu từ bên ngoài, thực hiện qua một số bộ lọc và xử lý dữ liệu cho ra kết quả là một tập các vector đặc trưng.

Bộ ngôn ngữ (Linguist): bằng các công cụ và phương pháp ngôn ngữ, đọc vào các tập tin cấu trúc của một ngôn ngữ rồi mô hình hóa chúng vào đồ thị. Ở bộ này cấu tạo khá phức tạp vì nó quy định hầu như toàn bộ phạm vi ngôn ngữ mà chúng ta cần nhận dạng, nó gồm các thành phần nhỏ sau: **Mô hình ngôn ngữ:** Đọc vào tập tin cấu trúc ngôn ngữ ở cấp độ là các từ. Thành phần này có vai trò quan trọng xác định những thứ hệ thống cần nhận dạng. Cấu trúc ngôn ngữ sẽ được mô hình hóa ở đây theo hai mô hình: graph-driven grammar và Stochastic N-Gram; **Mô hình graph-driven grammar:** Biểu diễn một đồ thị từ có hướng. Mỗi nút biểu diễn một từ đơn và mỗi cung là xác suất dịch chuyển sang một từ; **Mô hình stochastic N-Gram:** Mô hình này cung cấp các xác suất cho các từ được cho dựa vào việc quan sát N-1 từ đứng trước.

SimpleWordListGrammar: Định nghĩa một từ dựa trên danh sách các từ. Một tham số tùy chọn chỉ ra ngữ pháp có lặp hay không. Nếu không lặp, ngữ pháp sẽ được dùng cho một nhận dạng từ tách biệt. Nếu lặp, nó sẽ được dùng để hỗ trợ liên kết nhận dạng từ tầm thường.

JSGFGrammar: Hỗ trợ JavaTM Speech API Grammar Format (JSGF), định nghĩa một biểu diễn theo BNF, độc lập nền tảng, Unicode của các ngữ pháp.

LMGrammar: Định nghĩa một ngữ pháp dựa trên một mô hình ngôn ngữ thống kê. LMGrammar phát sinh một nút ngữ pháp mỗi từ và làm việc tốt với các unigram và bigram, xấp xỉ 1000 từ. FSTGrammar: Hỗ trợ một bộ chuyển đổi trạng thái giới hạn (finite-state transducer) trong định dạng ngữ pháp ARPA FST. SimpleNGramModel: Cung cấp hỗ trợ cho các mô hình ASCII N-Gram trong định dạng ARPA. SimpleNGramModel không cố làm tối ưu việc sử dụng bộ nhớ, do đó nó làm việc tốt với các mô hình ngôn ngữ nhỏ. LargeTrigramModel: Cung cấp hỗ trợ các mô hình N-Gram đúng được phát sinh bởi CMU-Cambridge Statistical Language Modeling Toolkit. LargeTrigramModel tối ưu việc lưu trữ bộ nhớ, cho phép nó làm việc với các tập tin rất lớn, trên 100MB. Bộ từ điển: Thành phần này cung cấp cách phát âm cho các từ ta đã xây dựng trong mô hình ngôn ngữ. Mô hình âm học: Cung cấp một ảnh xạ giữa một đơn

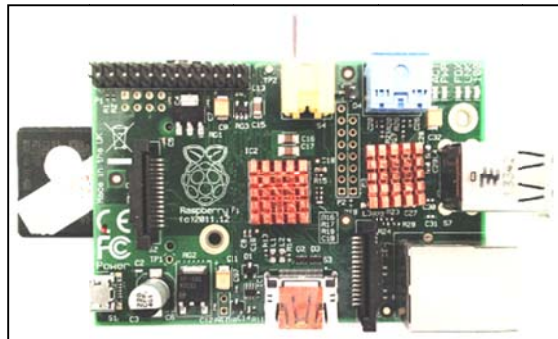
vị tiếng nói và một HMM có thể được đánh giá dựa vào các đặc trưng được cung cấp bởi bộ ngoại vi. Các ánh xạ có thể đưa thông tin vị trí của từ và ngữ cảnh từ thành phần mô hình ngôn ngữ. Định nghĩa ngữ cảnh này được xây dựng từ cấu trúc ngữ pháp của mô hình ngôn ngữ.

Đồ thị tìm kiếm (Search Graph): Là kết quả mà bộ ngôn ngữ phát sinh được cuối cùng để đưa vào sử dụng trong bộ giải mã. Đồ thị tìm kiếm này là một đồ thị có hướng, trong đó mỗi nút được gọi là một trạng thái tìm kiếm (SearchState), biểu diễn một trong hai trạng thái: phát hoặc không phát (emitting state hay non-emitting state). Và các đường cung biểu diễn các trạng thái biến đổi có thể, trên các cung này có các giá trị xác suất được tính toán từ mô hình âm học: biểu diễn khả năng chuyển từ trạng thái này đến trạng thái kia. Bộ giải mã (Decoder): Sử dụng các đặc trưng (Features) từ bộ ngoại vi kết hợp với đồ thị tìm kiếm được phát sinh từ bộ ngôn ngữ để tiến hành giải mã và áp dụng các thuật toán suy ra kết quả nhận dạng. Nhiệm vụ của thành phần quản lý tìm kiếm là nhận dạng các tập các vector đặc trưng để tìm ra ánh xạ tương ứng của nó trong đồ thị tìm kiếm. Để đáp ứng tìm ra kết quả chính xác trong đồ thị tìm kiếm khi xử lý kết quả, Sphinx cung cấp các tiện ích có khả năng phát sinh lưới và các đánh giá độ tin cậy từ kết quả. Và thêm đặc điểm nữa khác các hệ thống khác là không gian tìm kiếm trong Sphinx có thể được tinh chỉnh thay đổi trong quá trình tìm kiếm để tăng hiệu suất tìm kiếm. Ngoài ra để nâng cao hiệu suất của kết quả nhận dạng, Sphinx còn bổ sung thêm các công cụ hỗ trợ cho việc đánh giá kết quả nhận được, đó là thành phần đánh giá (Scorer) và thành phần cắt tỉa (Pruner). Nói về thành phần Scorer thì nó là một module dùng để ước lượng xác suất của trạng thái khi cung cấp các giá trị mật độ trạng thái xuất hiện. Khi thành phần quản lý tìm kiếm yêu cầu đánh giá điểm số cho một trạng thái, nó sẽ gọi đến thành phần Scorer, nó sẽ phân tích các thông tin đặc trưng của trạng thái đó rồi áp dụng các phép toán để tính điểm số.

III. HỆ THỐNG NHÚNG

Có thể chúng ta đã quen lập trình trên PC, với những ngôn ngữ như C, C++, C#, Java, Python, Ruby,... Nhưng phần mềm trên PC chỉ chiếm một phần nhỏ sản lượng phần mềm trên thị trường. Còn lại là code điều khiển tivi, máy giặt, điều hòa, tủ lạnh,... tóm lại là tất cả các thiết bị điện tử xung quanh chúng ta. Lập trình theo hướng này được gọi là embedded computing, hay physical computing, tức là lập trình tương tác với các thiết bị thực. Và 2 trong số các hệ thống nhúng được sử dụng phổ biến hiện nay là Arduino và Raspberry Pi.

A. Máy tính nhúng Raspberry Pi



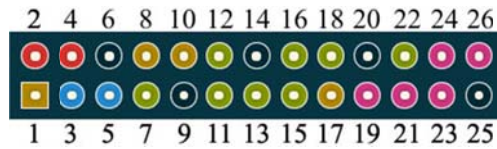
Hình 2. Máy tính nhúng Raspberry Pi thế hệ B

Raspberry Pi [4] là một máy tính nhúng kích cỡ nhỏ và chạy hệ điều hành Linux. Được phát triển bởi tổ chức phi lợi nhuận Raspberry Pi Foundation với tiêu chí xây dựng hệ thống mà nhiều người có thể sử dụng được trong những công việc tùy biến khác nhau. Đặc tính của Raspberry Pi xây dựng xoay quanh bộ xử lý SoC Broadcom BCM2835 (là chip xử lý mobile có kích thước nhỏ hay được dùng trong điện thoại di động) bao gồm CPU, GPU, bộ xử lý âm thanh/video, và các tính năng khác... Tất cả được tích hợp bên trong con chip này.

Bảng 1. Bảng cấu hình Raspberry Pi

CPU	ARM1176JZF-S 700MHz
GPU	Broadcom Video Core IV, hỗ trợ OpenGL, MPEG-2, VC-1; phát video Full HD (1080) H.264/MPEG-4
RAM	512MB SDRAM
USB	2 cổng
Ngõ ra Audio	Giắc 3.5mm, HDMI - Lưu trữ: khe cắm thẻ SD/MMC, SDIO - Kết nối: 10/100 Ethernet RJ45, USB hub...
Ngoại vi cấp thấp	8 GPIO, UART, I2C, SPI, +3.3V, +5V, GND
Nguồn nuôi	5V/700mA (3.5W), cấp nguồn qua USB hoặc chân GPIO
Kích thước	85.60 mm x 53.98 mm
Hệ điều hành	Debian GNU/Linux, Raspbian OS, Arch Linux ARM, RISC OS, FreeBSD, Plan 9

Raspberry Pi không thể chạy hệ điều hành Windows vì bộ xử lý BCM2835 dựa trên cấu trúc ARM nên không hỗ trợ mã x86/x64, nhưng vẫn có thể chạy bằng Linux với các tiện ích như lướt web, môi trường desktop và các nhiệm vụ khác. Raspberry Pi hỗ trợ lập trình C/C++, Java, Python, ... việc hỗ trợ nhiều ngôn ngữ lập trình nhằm tùy biến theo sở thích của người dùng. Khả năng lập trình ngay trên Raspberry Pi vẫn có thể được, tuy nhiên với một chiếc máy tính chuyên dụng chỉ xử lý 1 hay vài công việc thì việc lập trình trở nên khó khăn. Do đó bài báo đã dùng máy tính để lập trình cho việc nhận dạng tiếng nói thử nghiệm với độ ổn định cao. Bước tiếp theo đưa đoạn code vào hệ thống nhúng Raspberry Pi. Nhằm mục đích tối ưu trong quá trình xử lý tiếng nói. Máy tính nhúng này còn hỗ trợ điều khiển thiết bị ngoại vi qua hàng chân GPIO. Với tính năng này Raspberry Pi có thể được áp dụng vào nhiều công trình nghiên cứu và ứng dụng như: điều khiển robot, điều khiển thiết bị điện trong nhà, ...



Hình 3. Sơ đồ chân GPIO

Trong sơ đồ chân GPIO [5] có các chân được đánh thứ tự hàng trên là số chẵn hàng dưới là số lẻ. Tính từ trái qua phải khi đặt Raspberry Pi, hàng chân GPIO sẽ nằm cạnh bên cổng tín hiệu video trên bo Raspberry Pi. Trong đó: 1, 17: chân nguồn 3.3v, 2, 4: chân nguồn 5v, 6, 9, 14, 20, 25: chân nối đất Ground 0v, 8, 10: GPIO truyền nhận tín hiệu theo chuẩn UART, 7, 11, 12, 13, 15, 16, 18 và 22: chân GPIO, 19, 21, 23, 24, 26: GPIO giao tiếp chuẩn SPI, 3, 5: GPIO giao tiếp chuẩn I2C, 12: GPIO điều khiển PWM, tăng giảm cường độ.

B. Board mạch xử lý Arduino

Arduino là một bo mạch xử lý được dùng để lập trình tương tác các thiết bị phần cứng như cảm biến, động cơ, ... Bo mạch này sử dụng ngôn ngữ lập trình là ngôn ngữ gần giống với C/C++. Arduino là một nền tảng đã được chuẩn hóa, nên đã có rất nhiều các bo mạch mở rộng (gọi là shield) để cắm chồng lên bo mạch Arduino. Ví dụ, muốn kết nối Internet thì có Ethernet shield, muốn điều khiển động cơ thì có Motor shield, muốn kết nối nhận tín hiệu thì có GSM shield, ... Bài báo sử dụng trình biên dịch mà Arduino cung cấp để lập trình tại trang chủ <http://www.arduino.cc/>, tải về trình biên dịch Arduino 1.0.6 và sử dụng phiên bản dành cho hệ điều hành Windows.

IV. XÂY DỰNG MODULE ĐIỀU KHIỂN THIẾT BỊ TỪ XA

Để điều khiển thiết bị điện 220V AC, bài báo sử dụng một module điều khiển giao tiếp với bo điều khiển chính thông qua sóng vô tuyến. Module được xây dựng từ các thiết bị: bo mạch xử lý Arduino Pro Mini (ATmega 328), bộ thu/phát sóng vô tuyến RF SI4463, module bật/tắt điện 220V AC. Các thiết bị được kết nối với nhau như sau:

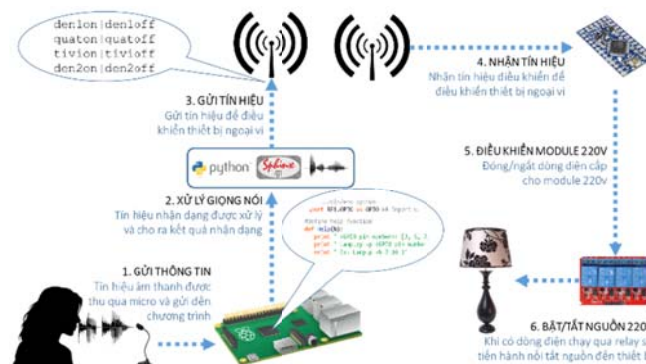


Hình 4. Mô hình kết nối module điều khiển thiết bị từ xa

Module này dùng để nhận lệnh điều khiển từ hệ thống trung tâm và đưa ra quyết định bật hay tắt nguồn điện cho thiết bị ngoại vi. Module này chạy cùng tần số 433 kHz với bộ điều khiển trung tâm và tốc độ truyền (baudrate) là 9600 bps. Và luôn luôn lắng nghe lệnh từ bộ trung tâm. Với mỗi lệnh được gửi đến thì module này sẽ điều khiển thiết bị tương ứng (gửi tín hiệu HIGH/LOW qua chân số 9 trên Arduino Pro Mini).

V. MÔ HÌNH ĐIỀU KHIỂN THIẾT BỊ BẰNG TIẾNG NÓI TIẾNG VIỆT

Hệ thống điều khiển trung tâm sẽ nhận tín hiệu tiếng nói và phản hồi điều khiển các thiết bị ngoại vi bên ngoài bằng sóng vô tuyến.



Hình 5. Sơ đồ hoạt động của hệ thống

Tín hiệu tiếng nói được thu từ micro gửi đến chương trình xử lý trong máy tính nhúng Raspberry Pi. Sau đó được khử nhiễu. Tại đây chương trình sẽ nhận dạng những câu lệnh mà đã được học và gửi tín hiệu điều khiển qua bộ phát RF đến module điều khiển 220V AC để bật/tắt thiết bị tương ứng với module đó. Để có thể xây dựng được hệ thống điều khiển trung tâm, bài báo sử dụng các phần mềm và thư viện hỗ trợ sau: Thư viện nhận dạng Pocketsphinx, CMUCLMTK, OpenFST, MIT Language Modeling Toolkit, m2m-aligner, Phonetisaurus, Python, subversion, autoconf, libtool, automake, gfortran, g++, jasper và RPi GPIO. Đây là các phần mềm và thư viện hỗ trợ điều khiển chân GPIO, các trình biên dịch để xây dựng chương trình và thư viện nhận dạng tiếng nói,... Ở đây bài báo sử dụng tên cho hệ thống là “ROBOT” - là từ ít xuất hiện khi người Việt nói chuyện với nhau nên sẽ không làm cho thiết bị hiểu nhầm là lệnh. Đầu tiên, cần xây dựng bộ từ điển được sử dụng. Các từ này được chứa trong một tập tin định dạng *.txt với nội dung là :”ROBOT” Sau khi xây dựng tập tin cho từ ROBOT.

```
ROBOT R OW B AA T
ROBOT(2) R OW B AH T
```

Và một tập tin mô hình ngôn ngữ của từ điển này cũng được cho ra với nội dung như sau:

```
\data\
ngram 1=3
ngram 2=2
ngram 3=1
\1-grams:
-0.7782 </s> -0.3010
-0.7782 <s> -0.2218
-0.7782 ROBOT -0.2218
\2-grams:
-0.3010 <s> ROBOT 0.0000
-0.3010 ROBOT </s> -0.3010
\3-grams:
-0.3010 <s> ROBOT </s>
\end\
```

Một số module không cần thiết đã được xóa đi là: Birthday.py, HN.py, Joke.py, Life.py, News.py, Time.py, Weather.py. Và để điều khiển được thiết bị qua sóng vô tuyến bài báo sử dụng 2 loại module với nội dung sau:

Module gửi tín hiệu là một chuỗi các ký tự điều khiển từ xa qua sóng vô tuyến truyền với tốc độ là 9600 bps và mỗi lần truyền tín hiệu mất 0,05 giây.

```
import serial
#...
def main(argv):
    try:
        opts, args = getopt.getopt(argv, "b:p:h:s:r", ['input=', 'params=', 'help'])
    except getopt.GetoptError:
        sys.exit(2)
    for opt, arg in opts:
        if opt in ("-r", "--radio"):
            port = serial.Serial("/dev/ttyAMA0", baud rate=9600, timeout=0.05)
            port.write(args[0])
            port.close()
```

Các module điều khiển thiết bị. Những module này gọi lại module gửi tín hiệu bên trên để gửi một câu lệnh cho bộ điều khiển ở xa. Đồng thời chạy tập tin âm thanh thông báo cho người dùng.

Ví dụ: Module mở tivi

```
import re
from tmp import minh
import os
WORDS = ["MO", "TIVI"]
def handle(text, mic, profile):
    print "======"
    print "ROBOT: DANG MO TIVI"
    print "======"
    os.system("aplay -D hw:1,0 /home/pi/jasper /static/audio/dangmotivi.wav")
    minh.main(['-r', 'tivion'])
def isValid(text):
    return bool(re.search(r'\b(mo|moat) tivi\b', text, re.IGNORECASE))
```

VI. KẾT QUẢ

Sau khi hoàn thành, hệ thống được đưa ra chạy thử nghiệm với micro sm-008, độ nhạy của micro là 60dB, tần số hoạt động trong khoảng 20 – 20000 Hz. Tiếng nói được thử nghiệm từ 3 người thanh niên, giới tính nam, 2 người quê ở Đồng Nai và 1 người quê ở Bắc Giang; mỗi người đọc 200 lần để ra lệnh (các lệnh của hệ thống gồm: [Đèn 1, Đèn 2, Tivi hoặc Quạt]); và đặt module nhận tín hiệu điều khiển từ xa ở các vị trí khác nhau trong phòng B304 (8x16 m²) được kết quả thử nghiệm như sau:

Bảng 2. Kết quả thử nghiệm

STT	Thực hiện	Số lần	Đúng	Sai
1	Phước	200	165	35
2	Son	200	181	19
3	Minh	200	188	12

Kết quả đúng 534/600 lần, đạt 89%, đây là một kết quả khả quan.

Bài báo đã sử dụng bộ thư viện có sẵn và cải tiến được việc xử lý điều khiển nhanh hơn. Thay vì việc điều khiển gồm 3 trạng thái gồm: từ khóa, công việc, thiết bị. Bài báo đã mạnh dạn áp dụng với 2 trạng thái: từ khóa, thiết bị. Mặc nhiên thiết bị chỉ ở trạng thái bật/ tắt nên việc kiểm tra chéo được thực hiện trong quá trình lập trình. Và bảng kết quả thử nghiệm trên đã cho ra một kết quả khả quan.

Bảng 3. Bảng giá thiết bị cho hệ thống

STT	Thiết bị	Số lượng	Giá (VNĐ)
1	Raspberry Pi B	1	900.000
2	Arduino Pro Mini	1	96.000
3	RF- SI4463	2	260.000
4	Relay_Opto	1	85.000

Bảng 3 là giá thiết bị dùng để điều khiển một hệ thống gồm 4 thiết bị 220V. Với việc điều khiển không dây, việc thi công dễ dàng. Một ngôi nhà thông minh thì chi phí cho một hệ thống này là đáp ứng tốt.

VII. KẾT LUẬN

Xây dựng thành công hệ thống nhận dạng tiếng trên hệ thống máy tính Raspberry Pi. Bài báo đã đạt được một số kết quả như: ứng dụng và cải tiến được công nghệ nhận dạng tiếng nói thông qua bộ thư viện nhận dạng CMU Sphinx; xây dựng hệ thống nhận dạng tiếng nói trên Raspberry Pi và điều khiển thiết bị bằng sóng. Hệ thống có thể điều khiển được một số thiết bị như: đèn, quạt và tivi, và khả năng thêm thiết bị được dễ dàng. Lập trình, thực thi trên hệ thống nhúng Raspberry Pi và thiết bị nhúng Arduino, nhỏ gọn nhưng vẫn linh động và hiệu quả về tốc độ xử lý và nâng cấp về sau. Có thể áp dụng để điều khiển ngôi nhà thông minh hiện đang là một xu thế mới của thế kỉ XXI.

Bài báo đã giải quyết một số vấn đề còn tồn tại trong các nghiên cứu trước đây: Có thể chỉ được áp dụng vào các hệ thống công kênh, kém linh hoạt [6] hoặc được làm từ các vi điều khiển nhỏ nhưng bị giới hạn về khả năng nhận biết, tốc độ xử lý cũng như số lượng từ vựng [1]. Bài báo này đã giải quyết được vấn đề về dung lượng bộ nhớ, tính linh hoạt của hệ thống.

VIII. LỜI CẢM ƠN

Xin chân thành cảm ơn tập thể quý thầy cô khoa Công nghệ thông tin, trường Đại học Lạc Hồng đã tạo điều kiện và đồng góp ý kiến để bài báo được hoàn thành.

IX. TÀI LIỆU THAM KHẢO

- [1] "Speech Recognition Based Wheelchair Using Device Switching" Bharati Vidyapeeth Deemed University, College Of Engineering (Electronics Engineering), Pune.
- [2] M. R. Alam, "A review of smart homes – Past, present and future," IEEE Trans. On Systems, Man and Cybernetics, vol. 42, no. 2, pp. 1190-1203, Nov. 2012. Author1_Name, Author2_Name, Web Caching and Replication, Addison-Wesley(Publication_Name), USA, 2014.
- [3] CMUSphinx Wiki, <http://cmusphinx.sourceforge.net/wiki/>.
- [4] Raspberry Pi models and revisions, <http://www.raspberrypi.org/documentation/hardware/raspberrypi/models/README.md>
- [5] Raspberry GPIO, <http://www.raspberrypi.org/documentation/usage/gpio/>
- [6] "A Computer Remote Control System Based on SpeechRecognition Technologies of Mobile Devices and Wireless Communication Technologies".

METHODS OF CONTROL DEVICE BY VOICE

Do Van Minh, Nguyen Minh Son, Phan Thien Phuoc

ABSTRACT - This paper presents methods Vietnamese voice recognition for controlling devices in smart home. Implement on small computers to increase the deployment of the system. The process to training the word: "Robot", "Lights", "Fan". The system uses Raspberry Pi and using Atmega 328 chip as the central processor (MCU-Micro Control Unit) for transmitting, receiving and signal control appliances. Although the device control method has a lot: smart phone, desktop, laptop combined with other technologies. But using voice to control appliances has many aspects need to be resolved. Method of paper offers a more optimal and easier deployment. The cost of the system is greatly reduced. But there are still some limitations encountered unresolved: handle interference (noise), the training process.