

**ĐẠI HỌC THÁI NGUYÊN**  
**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

---



**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN**

**ĐỀ TÀI**

**Robot iTNUT Module-02: XÂY DỰNG PHẦN MỀM NHẬN DẠNG  
TIẾNG NÓI TIẾNG VIỆT CHO MỘT TẬP CÂU HỎI ĐƠN GIẢN  
VÀ TỰ ĐỘNG TRẢ LỜI**

**Chủ nhiệm đề tài: SV. Dương Văn Huân – K185480106010**

**Giảng viên hướng dẫn: TS. Nguyễn Văn Huy**

**Thái nguyên - Ngày 27 Tháng 05 Năm 2022**

**ĐẠI HỌC THÁI NGUYÊN**  
**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

---

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN**

**ĐỀ TÀI**

**Robot iTNUT Module-02: XÂY DỰNG PHẦN MỀM NHẬN  
DẠNG TIẾNG NÓI TIẾNG VIỆT CHO MỘT TẬP CÂU HỎI  
ĐƠN GIẢN VÀ TỰ ĐỘNG TRẢ LỜI**

**Mã số: SV2022-02**

**Giảng viên hướng dẫn**

**Chủ nhiệm đề tài**

**TS. Nguyễn Văn Huy**

**Dương Văn Huân**

**Xác nhận của tổ chức chủ trì  
KT. HIỆU TRƯỞNG  
PHÓ HIỆU TRƯỞNG**

**PGS.TS. Vũ Ngọc Pi**

**Thái nguyên - Tháng 05 Năm 2022**

## **Danh sách thành viên tham gia đề tài**

### **CHỦ NHIỆM ĐỀ TÀI**

Họ và tên: Dương Văn Huân

Lớp: K54KMT

Mã số SV: K185480106010

Điện thoại di động: 0337674951

E-mail: K185480106010@tnut.edu.vn

### **CỘNG TÁC VIÊN**

Họ và tên: Hà Văn Khánh

Lớp: K56KMT

Mã số SV: K205480106041

Điện thoại di động: 0326091692

E-mail: K205480106041@tnut.edu.vn

### **GIÁO VIÊN HƯỚNG DẪN:**

Họ và tên: TS. Nguyễn Văn Huy

Điện thoại di động: 0968852824

E-mail: huynguyen@tnut.edu.vn

## Mục lục

ĐẠI HỌC THÁI NGUYÊN .....	1
CỘNG TÁC VIÊN.....	3
THÔNG TIN KẾT QUẢ NGHIÊN CỨU .....	6
LỜI NÓI ĐẦU.....	7
CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI .....	9
1.1. Tính cấp thiết.....	9
1.1.1 Tình hình nghiên cứu hiện nay.....	9
1.1.2. Tính cấp thiết .....	10
1.2. Phạm vi nghiên cứu .....	10
CHƯƠNG 2: TỔNG HỢP TIẾNG NÓI .....	11
2.1. Tổng quan.....	11
2.2. Các kỹ thuật tổng hợp cơ bản.....	12
2.3. Cách tiếp cận của đề tài .....	15
2.4. Giới thiệu google API.....	17
2.4.1 Google Speech to text API .....	17
2.4.2 Google Text to speech API.....	17
CHƯƠNG 3: XÂY DỰNG KỊCH BẢN HỎI ĐÁP .....	19
3.1. Tổng quan về trợ lý ảo.....	19
<i>Phương thức tương tác của trợ lý ảo: .....</i>	<i>19</i>
3.2 Tìm hiểu về trợ lý giọng nói.....	20
3.2.1 Lịch sử của trợ lý giọng nói.....	20
3.2.2 Công nghệ đằng sau Trợ lý giọng nói .....	22
3.2.3 Các phương pháp tiếp cận Trợ lý giọng nói .....	23
3.3. Giải pháp của đề tài .....	24
Chương 4. Các thành phần chính của đề tài .....	26
4.1. Tổng quan về Raspberry.....	26
4.1.1. Lịch sử .....	26
4.1.2. Cấu hình phần cứng.....	26
Thẻ SD Card .....	27
4.1.2. Các ngoại vi cần khi sử dụng Raspberry .....	30
4.1.3. Hệ điều hành.....	31
4.1.4. Cài đặt hệ điều hành .....	32
Bước 1. Format thẻ nhớ Boot Raspberry OS .....	32
Bước 3. Tiến hành cài đặt Raspberry Pi 4.....	35
Bước 4. Truy cập Raspberry Pi 4 bằng phần mềm PUTTY .....	37
Bước 5 Điều khiển từ xa Raspberry Pi 4 với VNC .....	38
4.2. Tổng quan về lập trình python.....	40
4.2.1. Biến trong Python.....	40
4.2.2. Kiểu dữ liệu số trong Python.....	40
4.2.3. Cấu trúc lệnh if trong Python .....	40
4.2.4. Vòng lặp while trong Python.....	41
4.2.5. Hàm trong Python.....	42
CHƯƠNG 5: THỬ NGHIỆM VÀ ĐÁNH GIÁ.....	44
5.1. Thử nghiệm tổng hợp tiếng nói .....	44

5.2. Thử nghiệm hỏi đáp.....	44
5.3. Đánh giá và Hướng phát triển .....	45
5.3.1 Đánh giá.....	45
5.3.2 Hướng phát triển của đề tài. ....	46
TÀI LIỆU THAM KHẢO .....	47
Phụ Lục: Code chương trình.....	48

## THÔNG TIN KẾT QUẢ NGHIÊN CỨU

### 1. Thông tin chung:

- TÊN ĐỀ TÀI: **Robot iTNUT. Module 02: Xây dựng phần mềm nhận dạng tiếng nói tiếng Việt cho một tập câu hỏi đơn giản và tự động trả lời.**
- Mã số: **SV2022-02**
- Chủ nhiệm đề tài: **Dương Văn Huân**
- Cộng tác viên: **Hà Văn Khánh**
- Giáo viên hướng dẫn: **TS. Nguyễn Văn Huy**
- Cơ quan chủ trì: **Khoa Điện tử**
- Thời gian thực hiện: **12 Tháng**

### 2. Mục tiêu:

- Xây dựng kịch bản tự động nhận dạng và trả lời những thắc mắc của sinh viên liên quan đến các vấn đề học tập, ngoại khóa, thông tin trường, lớp,... bằng các kỹ thuật nhận dạng và tổng hợp tiếng nói.

### 3. Kết quả nghiên cứu:

- Xây dựng được mô hình tổng hợp tiếng nói riêng sử dụng kỹ thuật truy vấn file audio.
- Trả lời được một số câu hỏi có trong tập dữ liệu.
- Tích hợp chương trình trên Raspberry pi 3.

Ngày 27 tháng 05 năm 2022

Cơ quan chủ trì  
**KT.HIỆU TRƯỞNG**  
**PHÓ HIỆU TRƯỞNG**

**Giáo viên hướng dẫn**

**Chủ nhiệm đề tài**

**PGS.TS. Vũ Ngọc Pi**

**TS. Nguyễn Văn Huy**

**Dương Văn Huân**

## LỜI NÓI ĐẦU

Mỗi sinh viên trong quá trình học tập, sinh hoạt tại trường của mình thường gặp rất nhiều khó khăn, thắc mắc cần được giải đáp. Thế nhưng do những kênh thông tin online lại hạn chế, thời gian thao tác lâu, phản hồi chậm sẽ không đáp ứng được yêu cầu của sinh viên.

Là một sinh viên năm 4 trường Đại học Kỹ thuật Công Nghiệp, em đã chứng kiến rất nhiều các bạn sinh viên năm nhất và cả bản thân em khi mới ngày đầu tiên bước chân đến trường nhưng lại không biết học tập ở nơi nào, đặc biệt là đối với môn thể chất. Sau một hồi chạy đi hỏi tới hỏi lui hết người này đến người khác thì các em mới biết sân thể chất nằm tách biệt với trường và cách trường khoảng 2km. Giá như lúc đó có một con bot trong trường nói rằng: “Đi thẳng 100 mét, sau đó rẽ trái đi thêm 2 km nữa là tới” khi hỏi về địa điểm sân thể dục thì có thể tiết kiệm rất nhiều thời gian công sức.

Những bạn học sinh cấp 3 chuẩn bị tốt nghiệp có cơ hội đến thăm TNUT chắc hẳn sẽ có rất nhiều câu hỏi muốn được giải đáp về trường trước khi đưa ra quyết định có nên nộp hồ sơ và chọn đây là nơi gắn bó suốt 5 năm sắp tới không. Thế nhưng do tâm lý một số bạn ngại hỏi, ngại tiếp xúc với người khác cũng như số người có thể trả lời được không nhiều.

Hay đơn giản như việc bạn đang đi trong trường và muốn biết xem bây giờ là mấy giờ để có thể thu xếp thời gian cho công việc tiếp theo. Nếu bạn không mang điện thoại, bạn sẽ phải chạy đi hỏi những người xung quanh hoặc tìm đến 1 chiếc đồng hồ công cộng. Và việc này có vẻ không khả thi lắm trong trường hợp bạn đang vội để bắt đầu một công việc nào đó.

Chính từ những khó khăn và yêu cầu thực tiễn đó mà em đã quyết định chọn đề tài “Xây dựng hệ thống trả lời tự động dựa trên kỹ thuật nhận diện và tổng hợp tiếng nói” làm đề tài nghiên cứu khoa học này.

Sau một thời gian nghiên cứu, từ nơi chưa có gì, em đã bước đầu hoàn thành hệ thống trả lời tự động, tiến tới mở rộng quy mô sản phẩm, phát triển ứng dụng thực tế tại Trường Đại học Kỹ thuật Công nghiệp.

Để đạt được những thành quả đó, trước hết phải kể đến sự cố vấn, định hướng, tận tình chỉ bảo của TS. *Nguyễn Văn Huy* – Giảng viên bộ môn Tin học Công nghiệp. Xin chân thành cảm ơn thầy!

Vì đề tài được triển khai trong khoảng thời gian tương đối ngắn, kết hợp vừa nghiên cứu đưa ra giải pháp, vừa sản xuất hệ thống, nên chắc chắn vẫn còn tồn tại những thiếu sót. Em hi vọng nhận được sự góp ý từ thầy cô và các bạn để sản phẩm được hoàn thiện hơn.

Thái Nguyên, ngày 24 tháng 05 năm 2022

Sinh viên

Dương Văn Huân



# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## 1.1. Tính cấp thiết

### 1.1.1 Tình hình nghiên cứu hiện nay

Robot thông minh được nghiên cứu và phát triển từ rất sớm đầu thế kỷ 19, robot thông minh là một hệ thống máy được cải thiện về khả năng nhận thức, ra quyết định và thực thi nhiệm vụ theo cách toàn diện hơn so với robot truyền thống. Chúng có thể mô phỏng ứng xử, cảm xúc và suy nghĩ giống người. Với một “bộ não” thông minh, robot có thể thực hiện theo chỉ dẫn của người vận hành, sau đó hoàn thành nhiều nhiệm vụ đã được lập trình trước, tự học và nâng cấp ứng xử của chúng trong lúc tương tác với con người. Xuất phát từ môi trường ứng dụng, có thể phân chia robot thông minh thành các loại robot sau:

Robot công nghiệp: thường là tay máy nhiều khớp (nhiều bậc tự do) ứng dụng trực tiếp cho công nghiệp và là robot chiếm hơn 60% giá trị thị trường robot toàn cầu. Đây là thiết bị máy luôn ứng dụng đầu tiên những kết quả nghiên cứu phát triển công nghệ mới nhất để có thể tự điều khiển và hoàn thành rất nhiều công việc khác nhau.

Robot dịch vụ trong nhà: ví dụ như robot quét và lau cửa sổ, có thể làm việc giống trợ lý giúp việc trong hộ gia đình. Chúng cũng có khả năng tìm kiếm thăm dò, tự động lập kế hoạch đường đi và tránh vật cản. Nhiều loại robot dịch vụ khác, như robot xã hội hóa gia đình, robot bầu bạn, robot trợ lý di động, robot huấn luyện thú cưng, đều có khả năng tương tác với người, cũng như hoàn thành các nhiệm vụ được giao, chăm nom người già và trẻ nhỏ, nhắc nhở sự kiện và tuần tra nhà. Bên cạnh đó, còn có robot giao tiếp cảm xúc, robot giáo dục trẻ nhỏ, robot nền học thông minh, UAV cá nhân, robot di động cá nhân... được hỗ trợ thêm công nghệ tương tác giọng nói cho phép tương tác giao tiếp và cảm nhận cảm xúc của con người. Nhiều robot dịch vụ thương mại, bao gồm cả robot nhận thức, robot hướng dẫn mua hàng, robot nấu ăn, robot văn phòng, robot an ninh... có thể tùy biến dịch vụ cá nhân theo tình huống ứng dụng cụ thể và hoàn thành nhiều nhiệm vụ như quảng cáo, cung cấp chỉ dẫn, tư vấn hỗ trợ, trợ lý công việc văn phòng, thực hiện tuần tra an ninh...

### **1.1.2. Tính cấp thiết**

AI hiện đang làm nên cuộc cách mạng giải phóng sức lao động trí não của con người. Một trong lĩnh vực ứng dụng công nghệ này là Robot thông minh. Với mục tiêu xây dựng robot có các hành vi tương tự con người phục vụ giải trí, nghiên cứu thực nghiệm AI tại TNUT, đề tài thực hiện nhằm xây dựng một phiên bản robot sử dụng trí tuệ nhân tạo thực hiện các công việc bước đầu đơn giản như vận chuyển hàng hóa, giao tiếp đa ngôn ngữ, marketing cho TNUT. Robot có khả năng nâng cấp các kỹ năng theo thời gian.

### **1.2. Phạm vi nghiên cứu**

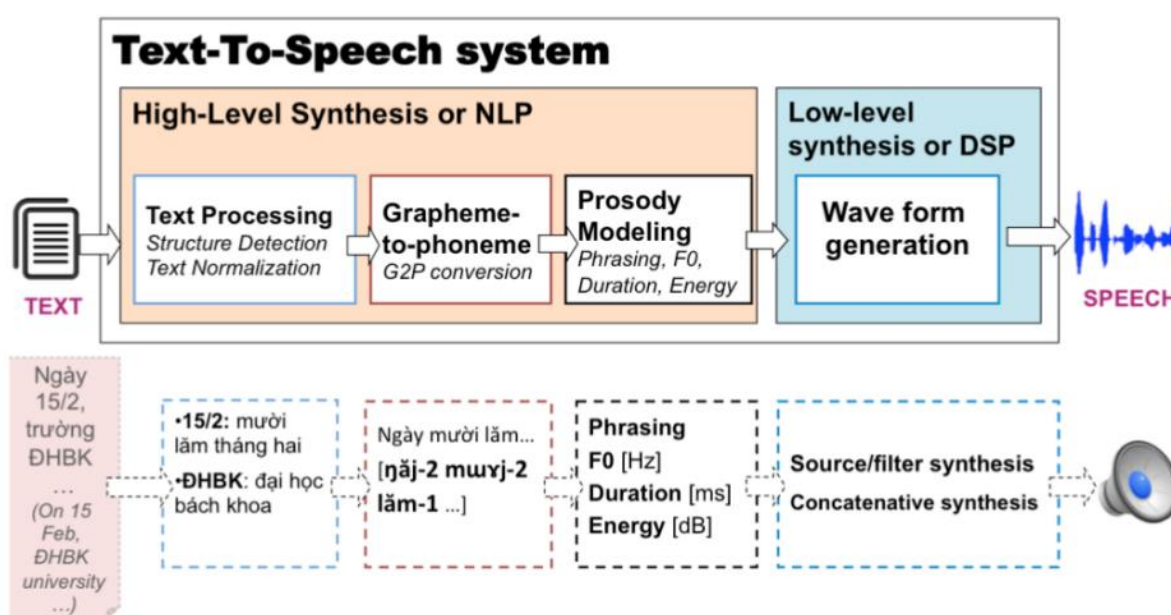
Đề tài này là một nhánh trong mục tiêu xây dựng Robot iTNUT có khả năng: Tự động nhận diện người đối diện; Tự động giao tiếp trả lời các câu hỏi đơn giản về TNUT bằng tiếng nói; Tự động di chuyển đến đích theo hành trình. Trong phạm vi đề tài này em sẽ xây dựng phần mềm nhận dạng được 1 tập các câu hỏi bằng tiếng nói và tự động trả lời bằng tiếng nói. Phần mềm được cài đặt và vận hành trên máy tính nhúng Raspberry Pi3.

# CHƯƠNG 2: TỔNG HỢP TIẾNG NÓI

## 2.1. Tổng quan

Tổng hợp tiếng nói là quá trình tạo ra tiếng nói của con người từ văn bản, hệ thống tổng hợp tiếng nói là hệ thống nhận đầu vào là một văn bản và tạo ra tín hiệu tiếng nói tương ứng ở đầu ra. Nghiên cứu về tổng hợp tiếng nói đã bắt đầu từ rất lâu, năm 1779 nhà khoa học người Đan Mạch Christian Kratzenstein đã xây dựng mô phỏng đơn giản hệ thống cấu âm của con người, mô hình này đã có thể phát ra được âm thanh của một số nguyên âm dài. Đến tận thế kỷ 19 các nghiên cứu tổng hợp tiếng nói vẫn còn ở mức đơn giản, phải sang thế kỷ 20 khi mà có sự lớn mạnh của hệ thống điện, điện tử thì mới thực sự xuất hiện những hệ thống tổng hợp tiếng nói chất lượng, có thể kể đến như hệ thống VODER lần đầu được giới thiệu năm 1939. Cho đến hiện nay, có rất nhiều các sản phẩm như sách nói, đồ chơi,.. sử dụng công nghệ tổng hợp tiếng nói. Đặc biệt các mô đun tổng hợp tiếng nói còn được tích hợp trong các trợ lý ảo trên điện thoại và máy tính như Siri hay Cortana.

Qua quá trình phát triển, hiện nay về cơ bản một hệ thống tổng hợp tiếng nói bao gồm hai thành phần chính: phần xử lý ngôn ngữ tự nhiên và phần xử lý tổng hợp tiếng nói. Phần xử lý ngôn ngữ tự nhiên: chuẩn hóa, xử lý các văn bản đầu vào thành các thành phần có thể phát âm được. Phần xử lý tổng hợp tiếng nói: Tạo ra tín hiệu tiếng nói từ các thành phần phát âm được nêu trên. Trên hình 1 mô tả một hệ thống tổng hợp tiếng nói gồm hai thành phần nêu trên.



Hình 2.1: Sơ đồ tổng quát một hệ thống tổng hợp tiếng nói

## 2.2. Các kỹ thuật tổng hợp cơ bản

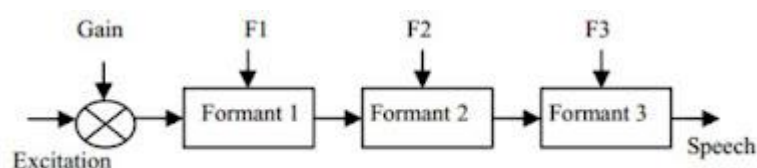
Hiện nay có hai hướng tiếp cận chính về tổng hợp tiếng nói: Hướng tiếp cận cổ điển, một hệ thống tổng hợp tiếng nói sẽ được phân ra làm hai phần chính gồm có phần xử lý ngôn ngữ tự nhiên và phần xử lý tổng hợp tiếng nói Hình 1. Hướng đi thứ hai là tổng hợp End to End, tức là toàn bộ quá trình từ huấn luyện hay tổng hợp chỉ cần chuẩn bị dữ liệu gồm có các tệp âm thanh và văn bản tương ứng, ngoài ra không cần nhiều kiến thức về ngôn ngữ học để xây dựng hệ thống.

### ***Tổng hợp mô phỏng hệ thống phát âm***

Tổng hợp mô phỏng hệ thống phát âm là các kỹ tổng hợp giọng nói dựa trên mô hình máy tính để mô phỏng các hệ thống phát âm của con người và mô phỏng quá trình tạo ra tiếng nói trên hệ thống đó. Tổng hợp dựa trên mô phỏng hệ thống phát âm hầu hết chỉ dành cho nghiên cứu. Lý do là vì mục tiêu của phương pháp này là mô phỏng quá trình tạo tiếng nói sao cho càng giống cơ chế của con người càng tốt, nên về lý thuyết có thể xem đây là phương pháp cơ bản nhất, nhưng cũng vì vậy mà phương pháp này khó thực hiện và chất lượng đầu ra thường không cao. Thời điểm hiện nay phương pháp này khó có thể được ứng dụng tại Việt Nam.

### ***Tổng hợp tần số Formant***

Tổng hợp tần số Formant là tổng hợp không sử dụng mẫu giọng thật nào khi chạy, thay vào đó tín hiệu được tạo ra bởi một mô hình tuyến âm. Mô hình này mô phỏng hiện tượng cộng hưởng của các cơ quan phát âm bằng một tập hợp các bộ lọc. Các bộ lọc này được gọi là các bộ lọc cộng hưởng Formant, chúng có thể được kết hợp song song hoặc nối tiếp hoặc cả hai. Hình hai biểu diễn mô hình tổng hợp Formant nối tiếp, trong đó đầu ra của bộ cộng hưởng này là đầu vào của bộ cộng hưởng kia:



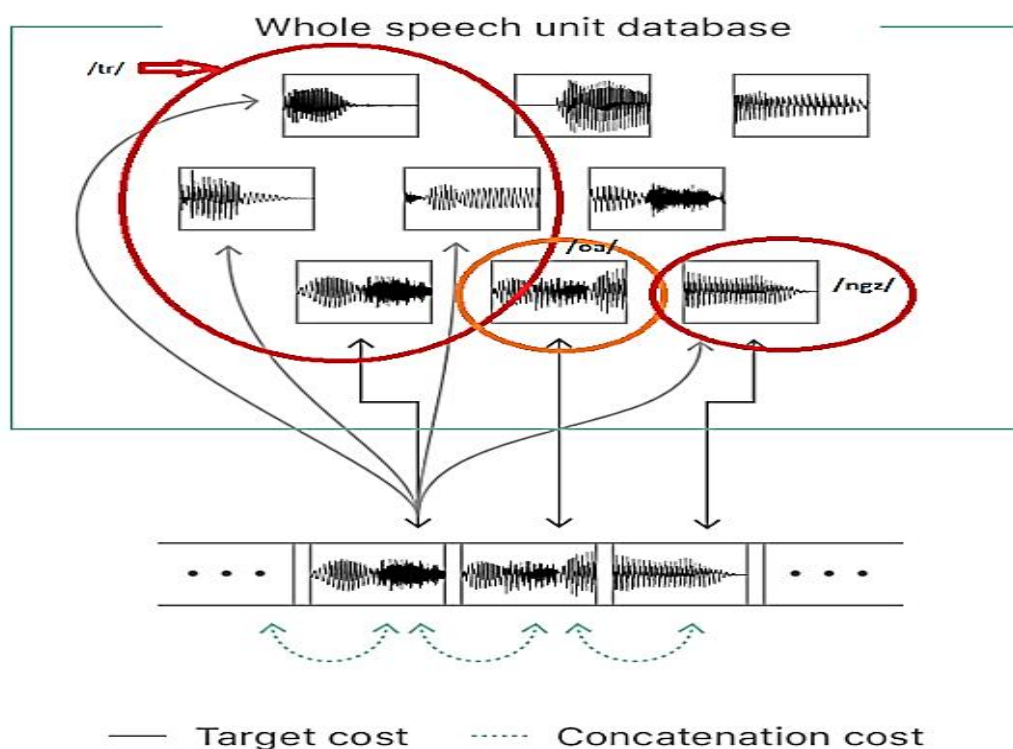
Hình 2.2: Hệ thống tổng hợp Formant nối tiếp.

Nhược điểm của phương pháp này là tạo ra giọng nói không tự nhiên, nghe rất "máy". Ưu điểm nhỏ gọn và chạy nhanh.

### ***Tổng hợp ghép nối***

Tổng hợp ghép nối là phương pháp dựa trên việc ghép nối các đoạn tín hiệu tiếng nói đã được ghi âm từ trước, xem mô tả cách hoạt động trên Hình 3. Đơn vị âm phổ biến là âm vị âm tiết, bán âm tiết, âm đôi, âm ba, từ cụm từ. Do đặc tính tự nhiên của tiếng nói

được ghi âm và lưu trữ trong các đơn vị âm, nên tổng hợp ghép nối là phương pháp có khả năng tổng hợp tiếng nói với mức độ dễ hiểu và tự nhiên cao. Tuy nhiên, do giọng nói tự nhiên được ghi âm thay đổi từ lần phát âm này sang lần phát âm khác, và công nghệ tự động hóa việc ghép nối các đoạn của tín hiệu thính thoảng tạo những tiếng cọt sạt không tự nhiên ở đoạn ghép.



Hình 2.3: Quá trình tổng hợp ghép nối

Có ba kiểu tổng hợp ghép nối chính:

- Tổng hợp chọn đơn vị (Unit selection)
- Tổng hợp âm kép (Diphone)
- Tổng hợp chuyên biệt (Domain-specific)

Tổng hợp chọn đơn vị dùng một cơ sở dữ liệu lớn các giọng nói ghi âm. Trong đó, mỗi câu được tách thành các đơn vị khác nhau như: các tiếng đơn lẻ, âm tiết, từ, nhóm từ hoặc câu văn. Một bảng tra các đơn vị được lập ra dựa trên các phần đã tách và các thông số âm học như tần số cơ bản, thời lượng, vị trí của âm tiết và các tiếng gần nó. Khi chạy các câu nói được tạo ra bằng cách xác định chuỗi đơn vị phù hợp nhất từ cơ sở dữ liệu. Quá trình này được gọi là chọn đơn vị và thường cần dùng đến cây quyết định được thực hiện. Thực tế, các hệ thống chọn đơn vị có thể tạo ra được giọng nói rất giống với người

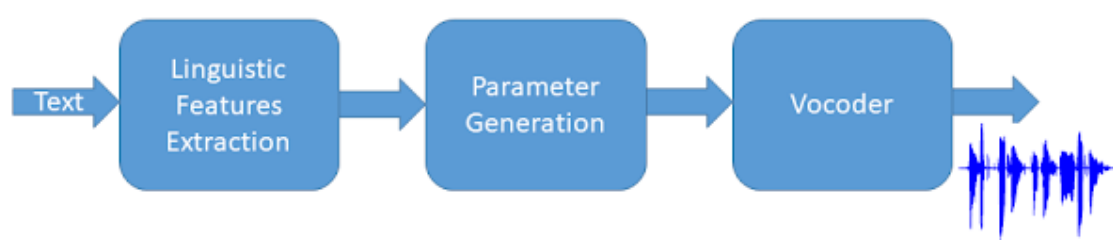
thật, tuy nhiên để đạt độ tự nhiên cao thường cần một cơ sở dữ liệu lớn chứa các đơn vị để lựa chọn.

Tổng hợp âm kép là dùng một cơ sở dữ liệu chứa tất cả các âm kép trong ngôn ngữ đang xét. Số lượng âm kép phụ thuộc vào đặc tính ghép âm học của ngôn ngữ. Trong tổng hợp âm kép chỉ có một mẫu của âm kép được chứa trong cơ sở dữ liệu, khi chạy thì lời văn được chồng lên các đơn vị này bằng kỹ thuật xử lý tín hiệu số nhờ mã tuyến đoán tuyến tính hay PSOLA. Chất lượng âm thanh tổng hợp theo cách này thường không cao bằng phương pháp chọn đơn vị nhưng tự nhiên hơn cộng hưởng tần số và ưu điểm của nó là có kích thước dữ liệu nhỏ.

Tổng hợp chuyên biệt (Domain-specific) là phương pháp ghép nối từ các đoạn văn bản đã được ghi âm để tạo ra lời nói. Phương pháp này thường được dùng cho các ứng dụng có văn bản chuyên biệt, cho một chuyên ngành, sử dụng từ vựng hạn chế như các thông báo chuyến bay hay dự báo thời tiết. Công nghệ này rất đơn giản và đã được thương mại hóa từ lâu. Mức độ tự nhiên của hệ thống này có thể rất cao vì số lượng các câu nói không nhiều và khớp với lời văn, âm điệu của giọng nói ghi âm. Tuy nhiên hệ thống kiểu này bị hạn chế bởi cơ sở dữ liệu chuyên biệt không áp dụng được cho miền dữ liệu mở.

### ***Tổng hợp dùng tham số thống kê***

Tiếp theo đây chúng ta sẽ xem xét về phương pháp tổng hợp tiếng nói đang được sử dụng rộng rãi, đó là tổng hợp dựa trên tham số thống kê. Trước đây tổng hợp thống kê thường dựa trên các mô hình Markov ẩn (HMMs), ngày nay mô hình này được thay thế bằng các mạng nơ ron học sâu để ước lượng các tham số đặc trưng âm học từ đầu vào là tham số đặc trưng ngôn ngữ học.



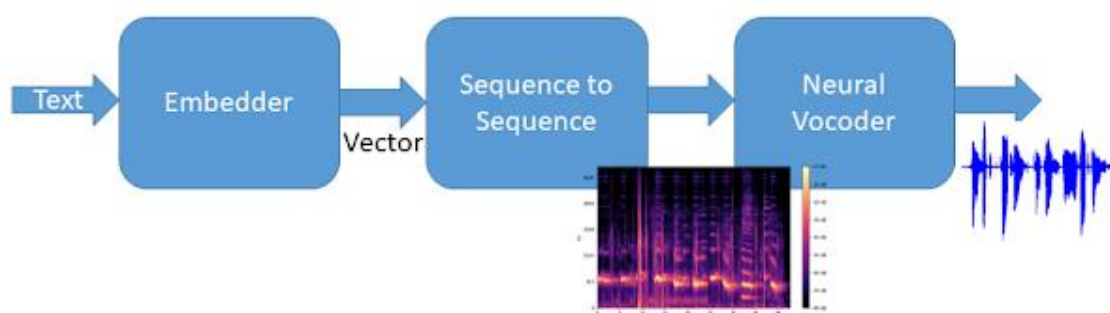
Hình 2.4: Tổng hợp tham số thống kê

Hình 4 mô tả kiến trúc phổ thông của một hệ thống tổng hợp tham số thống kê. Trong đó văn bản đầu vào sẽ được trích chọn thành các đặc trưng ngôn ngữ học bởi bộ Trích chọn đặc trưng ngôn ngữ (Linguistic Features Extraction). Sau đó các đặc trưng ngôn ngữ này đi qua bộ Parameter Generation và bộ này sẽ ước lượng được đặc trưng âm học ở đầu ra. Cuối cùng Vocoder tổng hợp tín hiệu tiếng nói từ những đặc trưng âm học này. Ngoài việc sử dụng riêng rẽ các phương pháp tổng hợp, thì trên thế giới cũng có những

ngiên cứu về tổng hợp lai ghép ví dụ như lai ghép giữa tổng hợp thống kê và tổng hợp ghép nối để tận dụng những ưu điểm của hai phương pháp này.

### ***Tổng hợp End to End***

Tổng hợp End to end là phương pháp mới được phát triển trong những năm gần đây. Mục tiêu của phương pháp này là tạo ra hệ thống tổng hợp có chất lượng cao nhất mà không cần dùng đến các kiến thức chuyên gia dựa trên các mạng nơ ron học sâu. Một kiến trúc nổi bật của phương pháp này có thể kể đến Tacotron2 của Google hay FastSpeech của Microsoft. Kiến trúc chung hay được sử dụng của một hệ thống End to End gồm có hai phần chính là: Phần tạo Mel spectrogram từ chuỗi ký tự đầu vào và phần chuyển hóa Mel spectrogram thành tín hiệu tiếng nói. Hình 5 mô tả kiến trúc một hệ thống tổng hợp End to End, trong đó phần tạo Mel spectrogram gồm hai mô đun là Embedder để chuyển hóa chuỗi ký tự thành chuỗi các véc tơ biểu diễn, sau đó mô đun Sequence to Sequence sẽ ước lượng Mel spectrogram từ chuỗi véc tơ này. Cuối cùng từ Mel spectrogram được chuyển hóa thành tín hiệu tiếng nói nhờ Neural Vocoder.


























Hình 2.5: Tổng hợp End to End

## **2.3. Cách tiếp cận của đề tài**

Có rất nhiều kỹ thuật tổng hợp tiếng đã được đề cập ở bên trên, tuy nhiên trong phạm vi đề tài nghiên cứu khoa học này, em sẽ tập trung vào nghiên cứu, tìm hiểu kỹ thuật tổng hợp tiếng nói ghép nối.

Chương trình sẽ sử dụng hơn 3000 file âm thanh dưới dạng mp3 tương ứng với hơn 3000 từ thông dụng được sử dụng trong tiếng Việt hàng ngày. Mỗi đoạn văn bản đầu vào sẽ được tách ra thành từng từ riêng biệt và với mỗi từ đó, chương trình sẽ truy xuất vào tập dữ liệu 3000 file mp3 để tìm các file âm thanh tương ứng. Và cuối cùng là gộp tất cả những file âm thanh đó thành 1 đoạn audio duy nhất theo đúng thứ tự ban đầu, xuất ra file mp3 để sử dụng.



 a.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 à.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ă.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 á.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ạ.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ác.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ách.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ai.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ài.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ải.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 am.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 àm.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ám.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 an.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 án.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ang.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 áng.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ảnh.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ánh.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ao.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ào.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ão.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9
 ớo.mp3	FPT Speech Synthesis	FPT Technology In...	banmai_ver1.9

Hình 2.6: Các file audio tương ứng với các tiếng trong tiếng Việt

Ưu điểm của kỹ thuật tổng hợp tiếng nói ghép nối là về tốc độ xử lý rất nhanh. Việc tổng hợp chỉ đơn giản là truy xuất, tìm kiếm vào các file và ghép chúng lại với nhau mà không cần phải mất thời gian để huấn luyện, không cần phải xử lý tính toán nhiều từ đó tiết kiệm tài nguyên CPU và GPU.

Phương pháp này có thể bắt chước được giọng của bất cứ ai miễn là có đủ các file âm thanh về tất cả các từ trong tiếng Việt do chính người đó đọc.

Tuy nhiên, bên cạnh đó kỹ thuật này vẫn còn tồn tại rất nhiều nhược điểm. Điểm lớn nhất phải kể đến đó là ngữ điệu của câu nói. File âm thanh được tổng hợp không thể hiện được âm điệu của câu văn ban đầu, không toát lên được cảm xúc, biểu lộ ý định của người nói.

Ngoài ra để có thể ghi âm đủ hơn 443 triệu âm tiết tiếng Việt và lưu chúng vào các file audio là điều rất khó và gây tốn kém bộ nhớ. Đối với đầu vào là các con số, các ký tự đặc biệt cũng rất khó để có thể tổng hợp được.



## 2.4. Giới thiệu google API

### 2.4.1 Google Speech to text API

Xét rằng Google về cơ bản là hệ thống thần kinh của Internet vào thời điểm này, không có gì ngạc nhiên khi API speech to text của họ là một trong những API phổ biến nhất - và mạnh mẽ nhất - dành cho các nhà phát triển.

Google Speech-To-Text đã được ra mắt vào năm 2018 , chỉ một tuần sau khi cập nhật tính năng chuyển văn bản thành giọng nói của họ. API Speech-To-Text của Google đưa ra một số tuyên bố táo bạo, giảm 54% lỗi từ trong bài kiểm tra này đến bài kiểm tra khác.

Một trong những lý do giải thích cho độ chính xác ấn tượng của API là khả năng lựa chọn giữa các mô hình học máy khác nhau , tùy thuộc vào ứng dụng của bạn đang được sử dụng để làm gì. Nó cũng có thể được cấu hình cho âm thanh từ cuộc gọi điện thoại hoặc video.

API Speech-To-Text cũng có một bản cập nhật ấn tượng cho các tùy chọn dấu câu mở rộng . Điều này được thiết kế để tạo ra các phiên âm hữu ích hơn, với ít câu chạy hoặc lỗi dấu câu hơn.

#### **Ưu điểm:**

- Nhận dạng hơn 120 ngôn ngữ
- Nhiều mô hình học máy để tăng độ chính xác
- Nhận dạng ngôn ngữ tự động
- Phiên âm văn bản
- Nhận dạng danh từ thích hợp
- Bảo mật dữ liệu
- Khử tiếng ồn cho âm thanh từ cuộc gọi điện thoại và video

#### **Nhược điểm:**

- Tốc độ phản hồi chậm

### 2.4.2 Google Text to speech API

Ngược lại với Speech to text, Google Text to speech là một API chuyển đổi từ dữ liệu dạng text sang dữ liệu giọng nói được phát triển bởi Google.

Công cụ tổng hợp tiếng nói này triển khai các công nghệ đột phá của Google để tạo ra giọng nói với ngữ điệu giống như con người. Được xây dựng dựa trên kiến thức chuyên môn về tổng hợp giọng nói của DeepMind, API mang đến những giọng nói gần giống với chất lượng của con người.

## CHƯƠNG 3: XÂY DỰNG KỊCH BẢN HỎI ĐÁP

### 3.1. Tổng quan về trợ lý ảo

Trợ lý ảo là các phần mềm được phát triển dựa trên trí thông minh nhân tạo (AI), hỗ trợ người dùng thực hiện các thao tác, hoặc tìm kiếm thông tin cho người dùng thông qua việc ra lệnh.

Trợ lý ảo đầu tiên được tạo ra là IBM Shoebox vào năm 1961, và đến nay thì trợ lý ảo đã được phổ cập khắp nơi, từ smartphone cho đến các thiết bị trong hệ sinh thái nhà thông minh.

#### *Phương thức tương tác của trợ lý ảo:*

##### **- Văn bản**

Đa số các trợ lý ảo hiện nay có hỗ trợ tương tác bằng văn bản. Ví dụ bạn có thể thấy đơn giản nhất đó là các chatbot bạn thường tương tác khi tiếp cận với các doanh nghiệp. Bạn chỉ cần nhắn tin, và chatbot sẽ trả lời bạn ngay lập tức dựa vào những gì được học từ trước.

##### **- Giọng nói**

Hiện nay, chỉ có vài công ty có tên tuổi trong ngành công nghệ như Google, Apple hay Amazon mới có các trợ lý ảo hỗ trợ nghe giọng nói ổn. Thậm chí, hiện nay các trợ lý ảo đã có khả năng phân biệt được giọng giữa các vùng miền nhờ vào lượng dữ liệu khổng lồ đến từ người dùng.

##### **- Hình ảnh**

Một số trợ lý ảo có thể xử lý thông tin dựa trên hình ảnh mà người dùng cung cấp. Tuy nhiên, phương thức này vẫn chưa được phổ biến ở thời điểm hiện tại.

Hiện nay thì các trợ lý ảo nổi tiếng như Siri, Google Assistant hay Samsung Bixby đều có hỗ trợ người dùng tương tác theo nhiều phương pháp khác nhau, mà chủ yếu nhất vẫn là thông qua văn bản và giọng nói.

#### *Các tính năng của trợ lý ảo:*

##### **- Tìm kiếm thông tin**

Nhiều trợ lý ảo hiện nay có khả năng tìm kiếm thông tin rất nhanh trên Internet, và nhờ đó chỉ trong vòng tích tắc, các trợ lý ảo như Siri hay Alexa có thể trả lời câu hỏi của bạn.

##### **- Điều khiển các tính năng nằm sâu trong thiết bị**

Một số trợ lý ảo sẽ được tích hợp sâu vào trong hệ điều hành của các thiết bị điện tử, vậy nên khả năng thao tác lệnh của các trợ lý ảo nhờ đây cũng được mở rộng ra.

#### **- Nhận diện giọng nói**

Nhờ khả năng nhận diện qua giọng nói, người dùng không còn phải tự tay gõ lệnh cho trợ lý ảo. Việc đơn giản bạn chỉ làm đó là nói “Hey! Siri” hay là “Ok Google”, sau đó nói ra câu lệnh của bạn.

#### **- Khả năng “học” từ người dùng**

Việc được phát triển dựa trên các mô hình Machine Learning giúp cho các trợ lý ảo này có thể không ngừng tiếp nhận dữ liệu từ người dùng, từ đó các thao tác thực hiện lệnh sau này sẽ nhanh, chính xác và giống con người hơn.

### **3.2 Tìm hiểu về trợ lý giọng nói**

Trợ lý giọng nói là trợ lý kỹ thuật số sử dụng nhận dạng giọng nói, thuật toán xử lý ngôn ngữ và tổng hợp giọng nói để nghe các lệnh thoại cụ thể và trả về thông tin liên quan hoặc thực hiện các chức năng cụ thể theo yêu cầu của người dùng.

Dựa trên các lệnh cụ thể, đôi khi được gọi là ý định, do người dùng nói, trợ lý giọng nói có thể trả lại thông tin liên quan bằng cách lắng nghe các từ khóa cụ thể và lọc ra tiếng ồn xung quanh.

Trong khi trợ lý giọng nói có thể hoàn toàn dựa trên phần mềm và có thể tích hợp vào hầu hết các thiết bị, một số trợ lý được thiết kế đặc biệt cho các ứng dụng thiết bị đơn lẻ, chẳng hạn như Đồng hồ treo tường Amazon Alexa.

Ngày nay, trợ lý giọng nói được tích hợp vào nhiều thiết bị mà chúng ta sử dụng hàng ngày, chẳng hạn như điện thoại di động, máy tính và loa thông minh. Do có nhiều mảng tích hợp, nên có một số trợ lý giọng nói cung cấp một bộ tính năng rất cụ thể, trong khi một số lại chọn kết thúc mở để giúp giải quyết hầu hết mọi tình huống trong tầm tay.

#### **3.2.1 Lịch sử của trợ lý giọng nói**

Trợ lý giọng nói có một lịch sử rất lâu đời thực sự đã có từ hơn 100 năm trước, điều này có vẻ đáng ngạc nhiên khi các ứng dụng như Siri chỉ mới được phát hành trong vòng mười năm trở lại đây.

Sản phẩm kích hoạt bằng giọng nói đầu tiên được phát hành vào năm 1922 với tên gọi Radio Rex. Đồ chơi này rất đơn giản, trong đó một con chó đồ chơi sẽ ở trong một ngôi nhà dành cho chó cho đến khi người dùng kêu lên tên của nó, “Rex” tại thời điểm đó nó sẽ nhảy ra khỏi nhà. Tất cả điều này được thực hiện bởi một nam châm điện được điều chỉnh đến tần số tương tự như nguyên âm được tìm thấy trong từ Rex, và có trước máy tính hiện đại hơn 20 năm.

Tại hội chợ Thế giới năm 1952, Audrey đã được Bell Labs công bố. Tuy nhiên, Bộ nhận dạng chữ số tự động không phải là một thiết bị nhỏ đơn giản, vỏ của nó cao sáu feet chỉ để chứa tất cả các vật liệu cần thiết để nhận dạng mười số!

IBM bắt đầu lịch sử lâu dài của họ về trợ lý giọng nói vào năm 1962 tại Hội chợ Thế giới ở Seattle khi IBM Shoebox được công bố. Thiết bị này có thể nhận dạng các chữ số 0-9 và sáu lệnh đơn giản như “cộng, trừ” để thiết bị có thể được sử dụng như một máy tính đơn giản. Tên của nó đề cập đến kích thước của nó, tương tự như hộp đựng giày trung bình và chứa một micro được kết nối với ba bộ lọc âm thanh để khớp với tần số điện của những gì đang được nói và khớp nó với các giá trị đã được chỉ định cho mỗi chữ số.

Darpa sau đó đã tài trợ cho 5 năm R & D về nhận dạng giọng nói vào năm 1971, được gọi là Chương trình Nghiên cứu Hiểu về Giọng nói (SUR) . Một trong những đổi mới lớn nhất sẽ xuất hiện nếu đây là Harpy của Carnegie Mellon , có khả năng hiểu hơn 1.000 từ.

Thập kỷ tiếp theo đã dẫn đến những tiến bộ và nghiên cứu đáng kinh ngạc trong lĩnh vực nhận dạng giọng nói, đưa hầu hết các thiết bị nhận dạng giọng nói từ việc hiểu vài trăm từ đến hiểu hàng nghìn từ và dần dần được đưa vào nhà người tiêu dùng.

Sau đó, vào năm 1990, Dragon Dictate được giới thiệu đến tận nhà người tiêu dùng với mức giá gây sốc 9.000 đô la! Đây là chương trình nhận dạng giọng nói theo định hướng người tiêu dùng đầu tiên được thiết kế cho PC gia đình. Người dùng có thể ra lệnh cho máy tính từng từ một, tạm dừng giữa mỗi từ để chờ máy tính xử lý trước khi họ có thể tiếp tục. Bảy năm sau, Dragon naturalSpeeches được phát hành và nó mang lại cuộc trò chuyện tự nhiên hơn, có thể hiểu được giọng nói liên tục với tốc độ tối đa 100 từ mỗi phút và mức giá thấp hơn nhiều là 695 đô la.

Năm 1994, Simon của IBM là trợ lý giọng nói thông minh đầu tiên. Simon là một PDA, và thực sự, là điện thoại thông minh đầu tiên trong lịch sử, coi như nó có trước Droid của HTC thực tế 25 năm!

Vào năm 2008, khi Android lần đầu tiên được phát hành, Google đã từ từ bắt đầu triển khai tính năng tìm kiếm bằng giọng nói cho các ứng dụng di động của Google trên nhiều nền tảng khác nhau, với Ứng dụng Tìm kiếm bằng giọng nói chuyên dụng của Google được phát hành vào năm 2011. Điều này dẫn đến ngày càng có nhiều tính năng nâng cao hơn, cuối cùng dẫn đến Google hiện tại và Trợ lý Google Voice .

Sau đó, Siri tiếp nối vào năm 2010. Được phát triển bởi SRI International với tính năng nhận dạng giọng nói do Nuance Communications cung cấp, ứng dụng gốc được phát hành vào năm 2010 trên iOS App Store và được Apple mua lại hai tháng sau đó. Sau đó, với việc phát hành iPhone 4s, Siri chính thức được phát hành như một trợ lý giọng nói tích

hợp trong iOS. Kể từ đó, Siri đã có mặt trên mọi thiết bị Apple có sẵn và liên kết tất cả các thiết bị với nhau trong một hệ sinh thái duy nhất.

Ngay sau khi Siri lần đầu tiên được phát triển, IBM Watson được công bố rộng rãi vào năm 2011. Watson được đặt theo tên của người sáng lập IBM, và ban đầu được hình thành vào năm 2006 để đánh bại con người trong trò chơi Jeopardy. Bây giờ, Watson là một trong những hệ thống máy tính thông minh nhất, nói một cách tự nhiên nhất hiện có.

Amazon Alexa sau đó được công bố vào năm 2015. Tên của nó được lấy cảm hứng từ Thư viện Alexandria và cũng là phụ âm cứng “X” trong tên, giúp nhận dạng giọng nói chính xác hơn. Với Alexa, dòng thiết bị thông minh Echo được công bố sẽ mang đến sự tích hợp thông minh cho ngôi nhà của người tiêu dùng với một lộ trình không tổn kém.

Alan cuối cùng đã được công bố rộng rãi vào năm 2017, làm khuynh đảo thế giới Ứng dụng Doanh nghiệp. Được sinh ra lần đầu tiên với tên gọi “Synqq”, Alan được tạo ra bởi những bộ óc đằng sau “Qik”, ứng dụng di động hội nghị và nhắn tin video đầu tiên. Alan là nền tảng AI bằng giọng nói đầu tiên nhắm đến các ứng dụng doanh nghiệp, vì vậy mặc dù nó có thể được tìm thấy trong nhiều ứng dụng tiêu dùng, nhưng nó được thiết kế để các doanh nghiệp có thể phát triển và tích hợp một cách nhanh chóng và hiệu quả!

### **3.2.2 Công nghệ đằng sau Trợ lý giọng nói**

Trợ lý giọng nói sử dụng Trí tuệ nhân tạo và Nhận dạng giọng nói để cung cấp chính xác và hiệu quả kết quả mà người dùng đang tìm kiếm. Mặc dù việc yêu cầu máy tính đặt hẹn giờ có vẻ đơn giản, nhưng công nghệ đằng sau nó thật hấp dẫn.

#### **Nhận diện giọng nói**

Nhận dạng giọng nói hoạt động bằng cách lấy tín hiệu tương tự từ giọng nói của người dùng và chuyển nó thành tín hiệu kỹ thuật số. Sau khi thực hiện việc này, máy tính sẽ lấy tín hiệu kỹ thuật số và cố gắng khớp nó với các từ và cụm từ để nhận ra ý định của người dùng. Để làm được điều này, máy tính yêu cầu một cơ sở dữ liệu gồm các từ và âm tiết có sẵn trong một ngôn ngữ nhất định để có thể khớp chặt chẽ với tín hiệu kỹ thuật số.

#### **Trí tuệ nhân tạo**

Trí tuệ nhân tạo hay trí thông minh nhân tạo (Artificial intelligence – viết tắt là AI) là một ngành thuộc lĩnh vực khoa học máy tính (Computer science). Là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người.

Trí tuệ nhân tạo khác với việc lập trình logic trong các ngôn ngữ lập trình là ở việc ứng dụng các hệ thống học máy (machine learning) để mô phỏng trí tuệ của con người trong các xử lý mà con người làm tốt hơn máy tính.

Cụ thể, trí tuệ nhân tạo giúp máy tính có được những trí tuệ của con người như: biết suy nghĩ và lập luận để giải quyết vấn đề, biết giao tiếp do hiểu ngôn ngữ, tiếng nói, biết học và tự thích nghi,...

AI hiện đại thường được coi là một hệ thống máy tính được thiết kế để hoàn thành các nhiệm vụ thường yêu cầu sự tương tác của con người. Các hệ thống này có thể tự cải thiện bằng cách sử dụng một quy trình được gọi là học máy.

### **Học máy**

Máy học đề cập đến tập hợp con của Trí tuệ nhân tạo trong đó các chương trình được tạo ra mà không cần sử dụng mã của con người để tạo chương trình theo cách thủ công. Thay vì tự viết ra chương trình hoàn chỉnh, các lập trình viên cung cấp cho AI các “mẫu” để nhận ra và học hỏi từ đó cung cấp cho AI một lượng lớn dữ liệu để sàng lọc và nghiên cứu. Vì vậy, thay vì phải tuân theo các quy tắc cụ thể, AI sẽ tìm kiếm các mẫu trong dữ liệu này và sử dụng nó để cải thiện các chức năng hiện có của nó. Một cách học máy có thể hữu ích cho Voice AI, đó là cung cấp cho thuật toán hàng giờ giọng nói từ các giọng và phương ngữ khác nhau.

Trong khi các chương trình truyền thống yêu cầu đầu vào và các quy tắc để phát triển đầu ra, thì các công cụ học máy được cung cấp đầu vào và đầu ra và sử dụng nó để tạo ra chính chương trình. Có hai cách tiếp cận để học máy, học có giám sát và học không giám sát. Trong học tập có giám sát, mô hình được cung cấp dữ liệu đã được gắn nhãn một phần, điều này có nghĩa là một số dữ liệu được cung cấp sẽ được gắn thẻ với câu trả lời đúng. Điều này giúp hướng dẫn mô hình phân loại phần còn lại của dữ liệu và phát triển một thuật toán chính xác. Trong học tập không giám sát, không có dữ liệu nào được gắn nhãn, vì vậy việc tìm đúng mẫu là tùy thuộc vào mô hình. Một trong những lý do khiến điều này rất hữu ích là vì nó cho phép mô hình tìm thấy các mẫu mà người sáng tạo có thể chưa bao giờ tự tìm thấy, nhưng dữ liệu khó dự đoán hơn nhiều.

### **3.2.3 Các phương pháp tiếp cận Trợ lý giọng nói**

Nhiều trợ lý đàm thoại ngày nay kết hợp cả quy trình làm việc theo định hướng nhiệm vụ và định hướng tri thức để thực hiện hầu hết mọi nhiệm vụ mà người dùng có thể thực hiện. Quy trình làm việc theo định hướng nhiệm vụ có thể bao gồm việc điền vào biểu mẫu, trong khi quy trình làm việc theo định hướng tri thức bao gồm việc trả lời vốn của trạng thái có thể là gì hoặc chỉ rõ các thông số kỹ thuật của sản phẩm.

### **Phương pháp tiếp cận theo định hướng nhiệm vụ**

Cách tiếp cận theo định hướng nhiệm vụ là sử dụng các mục tiêu cho các nhiệm vụ để đạt được những gì người dùng cần. Cách tiếp cận này thường tự tích hợp với các ứng dụng khác để giúp hoàn thành nhiệm vụ. Ví dụ: nếu bạn yêu cầu trợ lý giọng nói đặt báo thức lúc 3 giờ chiều, trợ lý này sẽ hiểu đây là một yêu cầu tác vụ và giao tiếp với ứng dụng Đồng hồ mặc định của bạn để mở và đặt báo thức cho 3 giờ chiều. Sau đó, nó sẽ giao tiếp với ứng dụng để xem có cần thiết gì khác không, chẳng hạn như tên cho báo thức, sau đó nó sẽ thông báo nhu cầu này lại cho bạn. Cách tiếp cận này không yêu cầu cơ sở dữ liệu trực tuyến rộng lớn, vì nó chủ yếu sử dụng kiến thức và kỹ năng đã có của các ứng dụng được cài đặt khác.

### **Phương pháp tiếp cận định hướng tri thức**

Phương pháp tiếp cận theo định hướng tri thức là việc sử dụng dữ liệu phân tích để giúp người dùng thực hiện các nhiệm vụ của họ. Cách tiếp cận này tập trung vào việc sử dụng cơ sở dữ liệu trực tuyến và kiến thức đã được ghi lại để giúp hoàn thành nhiệm vụ. Một ví dụ của cách tiếp cận này là bất cứ khi nào người dùng yêu cầu tìm kiếm trên internet, nó sẽ sử dụng cơ sở dữ liệu trực tuyến có sẵn để trả về kết quả phù hợp và đề xuất kết quả tìm kiếm cao nhất. Nếu ai đó đang tìm kiếm một câu hỏi đồ, điều này sẽ sử dụng cách tiếp cận theo định hướng kiến thức vì nó đang tìm kiếm dữ liệu thay vì làm việc với các ứng dụng khác để hoàn thành nhiệm vụ.

## **3.3. Giải pháp của đề tài**

Trong phạm vi đề tài nghiên cứu khoa học này, em sẽ đi xây dựng một chương trình hỏi đáp thông qua giọng nói tích hợp trên máy tính Raspberry Pi 3 phục vụ cho nhu cầu của sinh viên trường Đại học Kỹ thuật Công Nghiệp Thái Nguyên.

Đầu vào của bài toán là giọng nói tiếng Việt của người dùng, qua quá trình xử lý của chương trình sẽ cho ra câu trả lời tương ứng dưới dạng giọng nói được phát ra từ loa của robot.

Khi có người hỏi robot, chương trình sẽ gọi tới Google Speech to text API để chuyển giọng nói đó thành một chuỗi văn bản. Tiếp theo, chương trình sẽ kiểm tra xem đoạn text vừa rồi có phải tên của robot không để xác định xem có phải thực sự người dùng đang muốn giao tiếp với robot hay không.

Nếu đoạn text có chứa tên của robot thì chương trình sẽ lắng nghe câu hỏi hoặc yêu cầu từ phía người dùng. Người dùng có thể thực hiện một trong 2 hành vi đó là hỏi thông tin từ robot và thêm câu hỏi mới vào cơ sở dữ liệu.



Dữ liệu sẽ được lưu dưới dạng file csv, bao gồm 2 trường: cột keyword là các label xác định nội dung của câu hỏi và cột answer chứa câu trả lời cho câu hỏi tương ứng.

```
cauhoi.csv
1 question;answer
2 hiệu trưởng;tiến sĩ nguyên duy cương-ông nguyên duy cương-tôi không biết
3 thời gian đào tạo;bốn năm rưỡi
4 học ngành kỹ thuật máy tính ra làm gì;lập trình viên-phụ hồ
5 danh sách khoa;Khoa Điện, Khoa Cơ khí, Khoa Điện tử, Khoa Khoa học cơ bản, Khoa Kinh tế Công nghiệp, Khoa Kỹ thuật ô tô và
6 dễ không;không-có-dễ-không dễ-quá khó-bình thường
7 từ khóa;câu trả lời
8 tác giả;Đương Văn Huân
```

Hình 7: Cơ sở dữ liệu dưới dạng file csv.

Sau khi lắng nghe câu hỏi của người dùng và chuyển sang text, chương trình sẽ kiểm tra xem chuỗi văn bản đó có chứa keyword nào được lưu trong file csv không. Nếu có thì lấy ra câu trả lời và gọi API text to speech để phát câu trả lời ra loa.

Nếu người dùng yêu cầu thêm 1 câu hỏi và câu trả lời mới, hệ thống sẽ tiến hành kiểm tra xem câu hỏi đã tồn tại trong file csv chưa, nếu có thì có thể bổ sung câu trả lời cho câu hỏi đó hoặc bỏ qua. Còn nếu chưa có trong file csv thì tiến hành thêm mới một dòng dữ liệu.

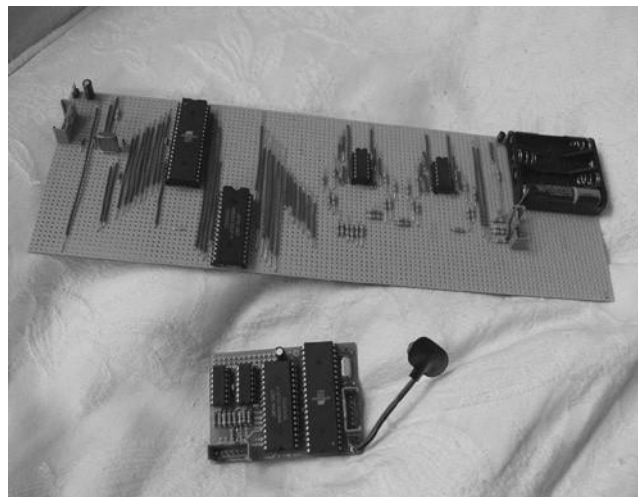
## Chương 4. Các thành phần chính của đề tài

### 4.1. Tổng quan về Raspberry

#### 4.1.1. Lịch sử

(Wikipedia, Raspberry Pi Foundation) Khi sự sụt giảm số lượng và kỹ năng của sinh viên đăng ký ngành Khoa học Máy tính trở thành mối lo ngại đối với một nhóm bao gồm Eben Upton, Rob Mullins, Jack Lang và Alan Mycroft tại Phòng thí nghiệm Máy tính của Đại học Cambridge vào năm 2006, nhu cầu về một chiếc máy tính nhỏ và giá cả phải chăng đã xuất hiện trong tâm trí họ. Một số phiên bản của nguyên mẫu Raspberry Pi ban đầu đã được thiết kế nhưng rất hạn chế bởi chi phí cao và bộ vi xử lý năng lượng thấp cho các thiết bị di động vào thời điểm đó.

Năm 2008, nhóm bắt đầu hợp tác với Pete Lomas, MD của Norcott Technologies và David Braben, đồng tác giả của BBC Micro game Elite, và thành lập Raspberry Pi Foundation. Ba năm sau, Raspberry Pi Model B ra đời và nó đã bán được hơn hai triệu chiếc trong vòng hai năm sản xuất hàng loạt.

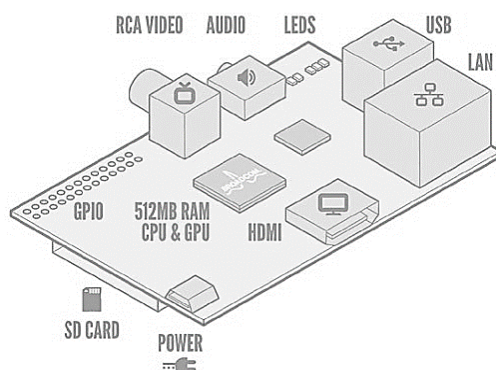


*Hình 4.1: Eben Upton's 2006 Raspberry Pi prototype (image ©Raspberry Pi Foundation)*

#### 4.1.2. Cấu hình phần cứng

Có hai mô hình Pi: mô hình A và mô hình B. Cả hai rất giống nhau, với mô hình B có thêm một vài tính năng (và giá cao hơn một chút) so với mô hình A. Model A có 256 MB RAM; mô hình B có 512 MB. Kiểu A có một cổng USB, trong khi kiểu B có hai. Mô hình A không có cổng Ethernet, trong khi mô hình B có một. Bạn vẫn có thể đặt hàng cái này hay cái kia; Sự khác biệt giữa một và hai cổng USB có thể rất lớn và khả năng cắm vào cáp Ethernet có dây cứng có thể làm cho mọi thứ như cập nhật và kết nối với bo mạch trở nên nhỏ gọn, mạng ad-hoc đơn giản hơn nhiều.

## RASPBERRY PI MODEL B



Hình 4.2: The Raspberry Pi Model B (image ©Raspberry Pi Foundation)

### Thẻ SD Card

Như bạn có thể thấy, rất nhiều thứ được dồn vào không gian nhỏ của Pi. Một trong những tính năng tiết kiệm dung lượng lớn nhất của Pi là không có ổ cứng thực như trong máy tính để bàn hoặc máy tính xách tay của; thẻ SD hoạt động giống như một ổ đĩa trạng thái rắn (SSD). Bạn có thể thay đổi kích thước của ổ đĩa đó chỉ bằng cách chuyển đổi thẻ SD. Bạn phải sử dụng ít nhất thẻ 2 GB và ít nhất là 4 GB được khuyến nghị nếu bạn muốn thêm bất kỳ phần mềm thực nào vào hình ảnh ổ đĩa của mình. Thẻ lên đến 32 GB đã được thử nghiệm và hoạt động, nhưng chúng không được đảm bảo, vì vậy hãy đảm bảo bạn sao lưu ổ đĩa của mình thường xuyên nếu bạn quyết định sử dụng thẻ bazillion-GB.

### Nguồn điện

Cổng nguồn chỉ là một đầu vào 5V micro-USB, tương tự như những gì bạn thấy trên nhiều điện thoại di động hoặc máy tính bảng. Trên thực tế, bộ sạc điện thoại di động là một trong những cách phổ biến nhất để cung cấp năng lượng cho Pi. Tuy nhiên, Raspberry Pi không có bộ điều chỉnh năng lượng tích hợp! Nếu đã quen với việc sử dụng Arduino, bạn biết rằng bạn có thể cấp nguồn cho Arduino một cách an toàn với tối đa 9V. Không vượt quá 5V—nếu bạn không chắc bộ sạc của mình hoạt động ra sao, hãy đo bằng đồng hồ vạn năng. Chúng ta cũng có thể cung cấp năng lượng cho Pi bằng pin, mặc dù hiệu suất có thể trở nên tồi tệ khi chúng phóng điện và mức năng lượng giảm xuống dưới 5V. Có lẽ cách dễ nhất để làm điều đó là sử dụng pin 9V hoặc một bộ 4 pin AA và đưa nó qua bộ điều chỉnh điện áp hoặc sử dụng bộ pin như loại được tìm thấy trong ô tô Điều khiển từ xa.

### Cổng HDMI Port

Pi được trang bị cổng đầu ra HDMI (Giao diện đa phương tiện độ nét cao) và nhiều người cho rằng đây thực sự là điểm xuất sắc của Pi vì nó có thể xuất ra đồ họa 1080p độ nét cao, với sức mạnh xử lý 1 gigapixel / giây. GPU tích hợp có thể phát lại chất lượng Blu-ray, sử dụng các thư viện OpenGL và OpenVG được cung cấp trên chip.

## Cổng Ethernet và USB

Cả hai cổng Ethernet và cổng USB (trên bo mạch kiểu B) đều được cung cấp thông qua chip LAN9512 trên bo mạch. Theo dữ liệu của 9512, đó là một USB 2.0 tốc độ cao với bộ điều khiển Ethernet 10/100. Con chip nhỏ này chỉ có cạnh 8 mm nhưng nó có khả năng hỗ trợ tốc độ USB 2.0 480 Mbps và hỗ trợ Ethernet 10BASE-T và 100-BASE-TX được tích hợp đầy đủ. Như vậy có nghĩa là hầu hết mọi thứ bạn có thể cắm vào máy tính để bàn của mình đều có thể được cắm vào Pi của bạn, từ bộ định tuyến đến webcam đến bộ chia USB đến ổ cứng ngoài (HDD .)

## Cổng Audio và Video

Giắc cắm âm thanh và video RCA cũng có trên bo mạch. Pi hỗ trợ âm thanh qua đầu ra HDMI, nhưng nếu bạn muốn cắm tai nghe, có jack cắm âm thanh 3,5 mm tiêu chuẩn. Nếu bạn muốn sử dụng micrô, hầu hết micrô USB sẽ hoạt động, giả sử chúng tương thích với Linux. Đối với video: Pi không hỗ trợ đầu ra VGA, nhưng jack cắm RCA sẽ gửi video đến bất kỳ thiết bị video RCA nào được kết nối — hữu ích nếu bạn có một cặp kính xem video độc lập như thiết bị MyVu.

## Các cổng vào ra đa dụng GPIO

Các chân này cho phép bạn kết nối Pi với bất kỳ số phần mở rộng vật lý nào, từ đèn LED và động cơ servo đến bộ điều khiển động cơ và bảng mở rộng như Gertboard lớn. Với một máy tính để bàn hoặc máy tính xách tay thông thường, điều này sẽ yêu cầu một số kiến thức chuyên môn và nhiều linh kiện, với trình điều khiển USB hoặc truy cập vào cổng nối tiếp (có lẽ không tồn tại) và thực hiện một số phép thuật lập trình cấp thấp. Nhưng Raspberry Pi đi kèm với các thư viện được cài đặt sẵn cho phép bạn truy cập các chân bằng Python, C hoặc C ++. Các thư viện bổ sung cũng tồn tại nếu bạn không thích các phiên bản chính thức, được tải trước. Điều này có nghĩa là bạn có thể kết nối tối đa tám servo với Pi ngay lập tức — đủ để điều khiển một rô bốt bốn chân chẳng hạn.

RPi A+, B+ và 2B GPIO J8 có 40-chân pinout.<sup>[21]</sup> Model A và B chỉ có 26 chân.

GPIO#	Chức năng thứ 2	pin#	pin#	Chức năng thứ 2	GPIO#
N/A	+3V3	1	2	+5V	N/A
GPIO2	SDA1 (I2C)	3	4	+5V	N/A
GPIO3	SCL1 (I2C)	5	6	GND	N/A
GPIO4	GCLK	7	8	TXD0 (UART)	GPIO14
N/A	GND	9	10	RXD0 (UART)	GPIO15
GPIO17	GEN0	11	12	GEN1	GPIO18

GPIO27	GEN2	13	14	GND	N/A
GPIO22	GEN3	15	16	GEN4	GPIO23
N/A	+3V3	17	18	GEN5	GPIO24
GPIO10	MOSI (SPI)	19	20	GND	N/A
GPIO9	MISO (SPI)	21	22	GEN6	GPIO25
GPIO11	SCLK (SPI)	23	24	CE0_N (SPI)	GPIO8
N/A	GND	25	26	CE1_N (SPI)	GPIO7
<i>(Models A and B stop here)</i>					
EEPROM	ID_SD	27	28	ID_SC	EEPROM
GPIO5	N/A	29	30	GND	N/A
GPIO6	N/A	31	32	-	GPIO12
GPIO13	N/A	33	34	GND	N/A
GPIO19	N/A	35	36	N/A	GPIO16
GPIO26	N/A	37	38	Digital IN	GPIO20
N/A	GND	39	40	Digital OUT	GPIO21

Model B rev 2 cũng có một pad (gọi là P5 trên board mạch và P6 trên sơ đồ) của 8 chân cung cấp truy cập đến một kết nối 4 GPIO bổ sung. [\[22\]](#)

Chức năng	chức năng thứ 2	chân#	chân#	Chức năng thứ 2	Chức năng
N/A	+5V	1	2	+3V3	N/A
GPIO28	GPIO_GEN7	3	4	GPIO_GEN8	GPIO29
GPIO30	GPIO_GEN9	5	6	GPIO_GEN10	GPIO31
N/A	GND	7	8	GND	N/A

Model A và B quy định GPIO truy cập vào LED trạng thái ACT sử dụng GPIO 16. Model A+ và B+ và quy định GPIO truy cập vào các LED trạng thái ACT sử dụng GPIO 47, và LED trạng thái nguồn sử dụng GPIO 35.

### Các Chip tích hợp

Phần quan trọng nhất trên bảng là chip tích hợp, còn được gọi là SoC. Chip của Pi là Broadcom BCM2835 với bộ xử lý ARM11 chạy ở tốc độ 700 MHz và GPU Videocore4. Con chip này có thể được ép xung lên ít nhất 800 MHz mà không gặp vấn đề gì.



*Hình 4.3: Pi-4 B*

Thông số kỹ thuật của Raspberry Pi 4 Model B:

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- Có 3 lựa chọn RAM: 2GB, 4GB hoặc 8GB LPDDR4-2400 SDRAM
- Wifi chuẩn 2.4 GHz và 5.0 GHz IEEE 802.11ac. Bluetooth 5.0, BLE
- Cổng mạng Gigabit Ethernet
- 2 cổng USB 3.0 và 2 cổng USB 2.0
- Chuẩn 40 chân GPIO, tương thích với các phiên bản trước
- Hỗ trợ 2 cổng ra màn hình chuẩn Micro HDMI với độ phân giải lên tới 4K
- Cổng MIPI DSI
- Cổng MIPI CSI
- Cổng AV 4 chân
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.0 graphics
- Khe cắm Micro-SD cho hệ điều hành và lưu trữ
- Nguồn điện DC 5V – 3A DC chuẩn USB-C
- 5V DC via GPIO header (minimum 3A\*)
- Hỗ trợ Power over Ethernet (PoE) (yêu cầu có PoE HAT)

#### **4.1.2. Các ngoại vi cần khi sử dụng Raspberry**

##### **Nguồn**

Do Pi không có bộ điều chỉnh điện áp trên bo mạch. Bạn không thể cắm pin 9V hoặc 200v trực tiếp. Nguồn được khuyến cáo là do nhà sản xuất với thông số là 5V-3A.

##### **Màn hình**

Thiết bị ngoại vi tiếp theo mà bạn cần là một màn hình có cổng HDMI hoặc DVI. Nếu đầu vào DVI thì cần sử dụng bộ chuyển đổi HDMI-to-DVI. Sau khi thiết lập xong và cài đặt

tất cả các phần mềm cần thiết, bạn có thể chạy Pi ở cấu hình không cần màn hình. Điều đó có nghĩa là bạn có thể đăng nhập vào nó từ một máy tính khác bằng SSH (Secure Shell) hoặc thậm chí là một máy khách VNC (Virtual Network Computing). Nhưng cho lần đầu sử dụng và cài đặt, bạn sẽ cần một màn hình để có thể thiết lập cho Pi.

### 4.1.3. Hệ điều hành

Về cơ bản Raspberry Pi có khá nhiều OS linux chạy được nhưng vẫn có sự thiếu vắng của Ubuntu (do CPU ARMv6) Điểm danh một số Distributions Linux (nhúng) chạy trên Raspberry Pi như Raspbian, Pidora, openSUSE, OpenWRT, OpenELEC,....

Raspberrypi.vn xin nêu một số HĐH nổi tiếng. Còn những OS nhỏ lẻ và cách cài đặt các bạn có thể tự tìm kiếm trên Google hoặc truy cập website chính thức của Raspberry Pi tại www.raspberrypi.org.

#### 1. Raspbian: <http://www.raspbian.org/>



Đây là bản build Linux dựa trên nền Debian (Gần giống ubuntu) với giao diện LXDE (thay vì GNOME). Có đầy đủ web browser, media player, tools, etc ... Nói chung HĐH này dành cho những người muốn dùng Raspberry Pi như một cái PC.



## 2. Raspbmc: <http://www.raspbmc.com/>

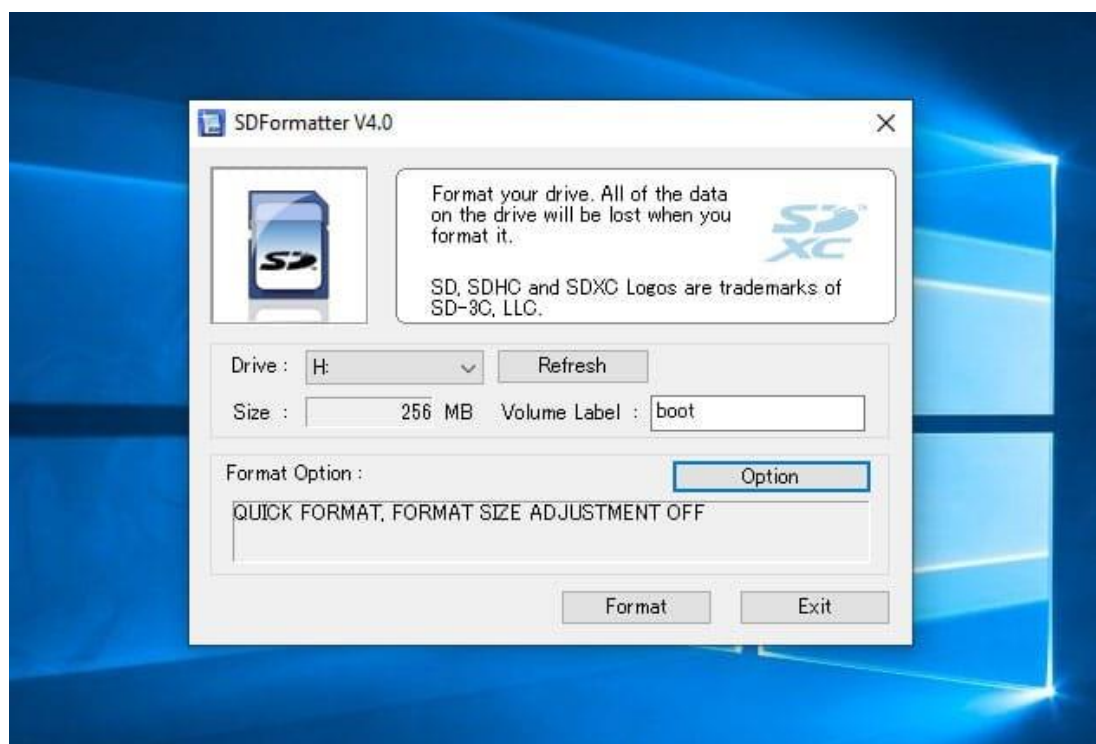


Có thể gọi đây là bản Raspbian lược bỏ đi LXDE và thay vào đó là XBMC. Nếu ở độ phân giải 720P (Chỉ là UI thôi nhé, phát phim vẫn 1080P), OC (CPU 1GHz, Ram 500, DSP 250, GPU 450, OverVolt 5), fps lúc nào cũng trên 60fps (nếu tắt VSync lên tới 80fps). Phim hỗ trợ đủ loại format, codec. Tuy nhiên vì codec VC-1 và MPEG2 là codec thu phí nên phải trả tiền mua code để unlock. Tuy nhiên các phim HD không bao giờ dùng codec này nên cũng không cần lo lắng mấy. Âm thanh hỗ trợ đầy đủ từ DTS-HD Master, DTS, Dolby, MP3, ACC,... Tuy nhiên muốn nghe âm thanh 5.1 cần phải qua receiver. Phim hỗ trợ đầy đủ nguồn từ NFS, samba, USB, HDD (3TB), UPNP,...

### 4.1.4. Cài đặt hệ điều hành

#### Bước 1. Format thẻ nhớ Boot Raspberry OS

Bạn cần Format thẻ nhớ của mình trước khi có thể tạo thành boot Raspberry OS.

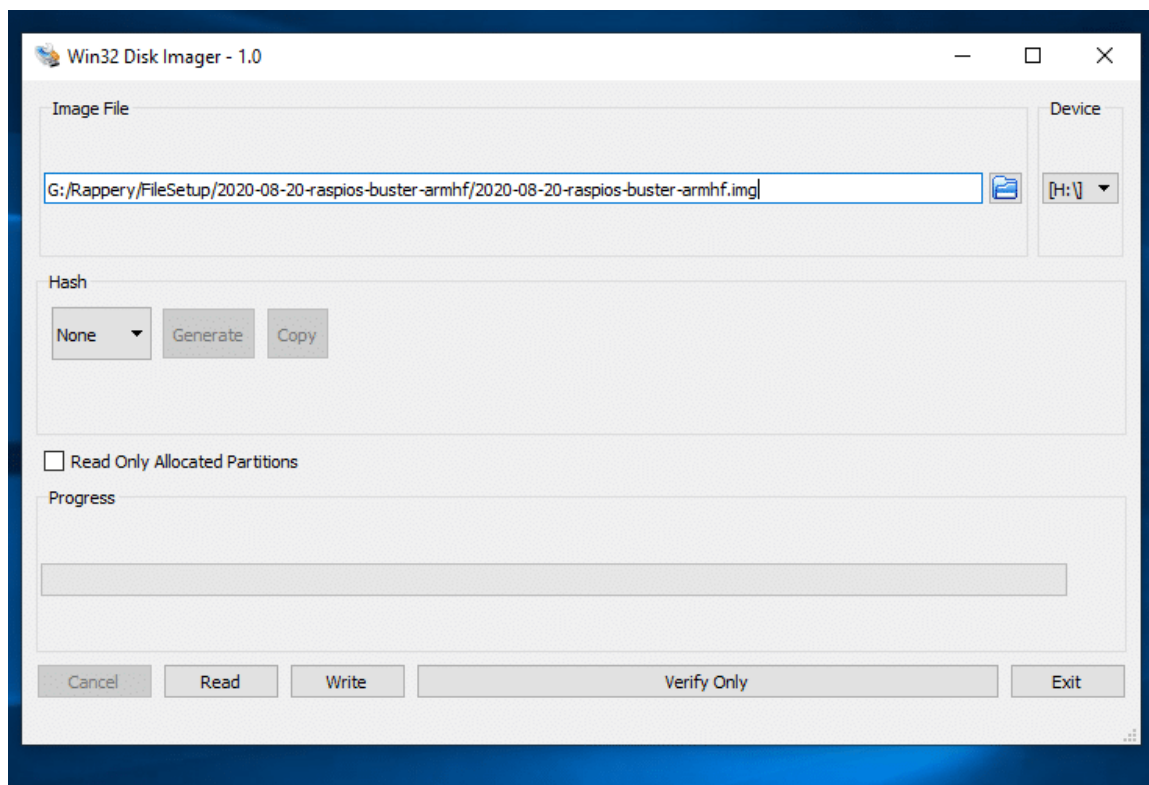




Dùng một đầu đọc thẻ nhớ cắm vào USB máy tính của bạn. Bật phần mềm SDFormatter vừa cài bên trên. Chọn đúng ổ đĩa mà máy tính nhận thẻ nhớ của bạn và Click Format.

## **Bước 2 : Tiến hành tạo file Boot Raspberry OS vào thẻ nhớ**

Các bạn cần download file Raspberry OS về trước nhé. Link RASSTEK Blog có đề bên trên.



Các bạn trở đến đường dẫn nên lưu file vừa mới Download Raspberry OS về.  
Sau đó chọn WRITE, quá trình ghi file sẽ mất từ 8-10 phút, nên chờ đợi và hạn chế thao tác để tránh ghi lỗi file.  
Sau khi hoàn thành bạn vào đúng ổ đĩa vừa Boot thành công bạn sẽ thấy các file sau

Name	Date modified	Type	Size
overlays	06/11/2020 5:59 SA	File folder	
bcm2708-rpi-b.dtb	06/11/2020 5:59 SA	DTB File	25 KB
bcm2708-rpi-b-plus.dtb	06/11/2020 5:59 SA	DTB File	25 KB
bcm2708-rpi-b-rev1.dtb	06/11/2020 5:59 SA	DTB File	25 KB
bcm2708-rpi-cm.dtb	06/11/2020 5:59 SA	DTB File	25 KB
bcm2708-rpi-zero.dtb	06/11/2020 5:59 SA	DTB File	25 KB
bcm2708-rpi-zero-w.dtb	06/11/2020 5:59 SA	DTB File	26 KB
bcm2709-rpi-2-b.dtb	06/11/2020 5:59 SA	DTB File	26 KB
bcm2710-rpi-2-b.dtb	06/11/2020 5:59 SA	DTB File	26 KB
bcm2710-rpi-3-b.dtb	06/11/2020 5:59 SA	DTB File	28 KB
bcm2710-rpi-3-b-plus.dtb	06/11/2020 5:59 SA	DTB File	28 KB
bcm2710-rpi-cm3.dtb	06/11/2020 5:59 SA	DTB File	26 KB
bcm2711-rpi-4-b.dtb	06/11/2020 5:59 SA	DTB File	47 KB
bcm2711-rpi-cm4.dtb	06/11/2020 5:59 SA	DTB File	47 KB
bootcode.bin	06/11/2020 6:00 SA	BIN File	52 KB
cmdline.txt		Text Document	1 KB
config.txt	06/11/2020 6:59 SA	Text Document	2 KB
COPYING.linux	06/11/2020 5:59 SA	LINUX File	19 KB
fixup.dat	06/11/2020 5:59 SA	DAT File	8 KB
fixup_cd.dat	06/11/2020 5:59 SA	DAT File	4 KB
fixup_db.dat	06/11/2020 6:00 SA	DAT File	11 KB
fixup_x.dat	06/11/2020 6:00 SA	DAT File	11 KB
fixup4.dat	06/11/2020 6:00 SA	DAT File	6 KB
fixup4cd.dat	06/11/2020 6:00 SA	DAT File	4 KB
fixup4db.dat	06/11/2020 6:00 SA	DAT File	9 KB
fixup4x.dat	06/11/2020 6:00 SA	DAT File	9 KB
issue.txt	20/08/2020 10:54 SA	Text Document	1 KB
kernel.img	06/11/2020 5:59 SA	Disc Image File	5.326 KB

Tiếp theo bạn cần tạo 1 file trống tên là SSH với không định dạng file.

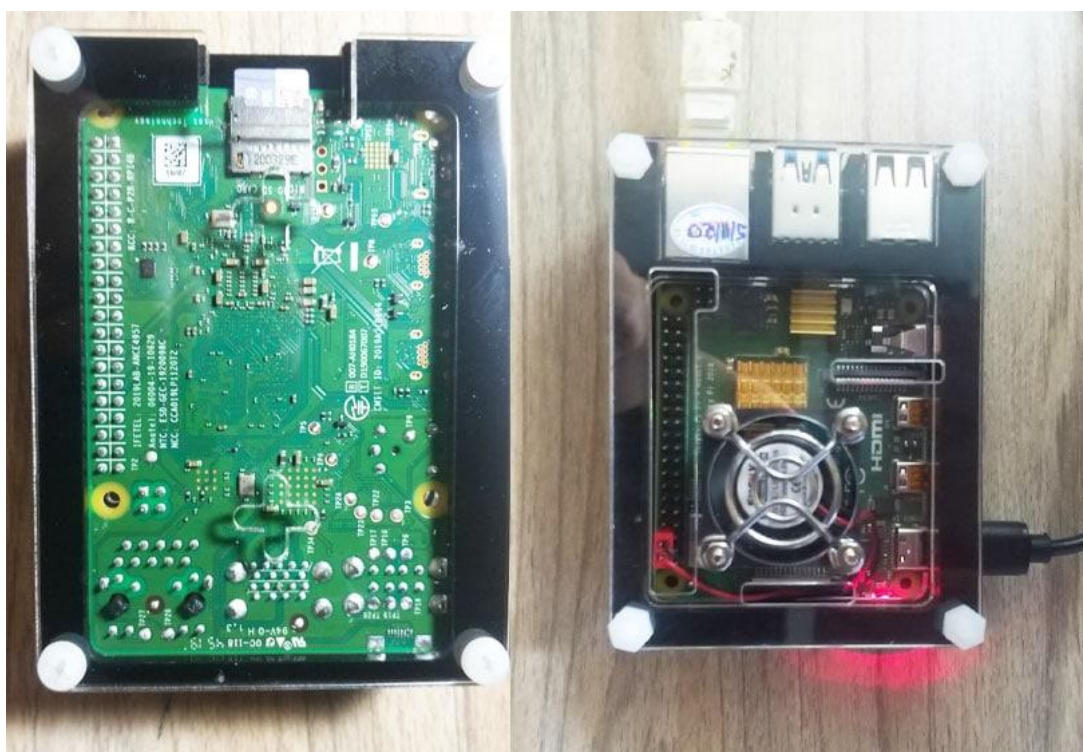
Chỉ cần click chuột phải chọn “New “ chọn “Text Document” đặt tên là “SSH” và bỏ luôn đuôi “.txt” .

Name	Date modified	Type	Size
bcm2711-rpi-4.dtb	06/11/2020 5:59 SA	DTB File	47 KB
bcm2711-rpi-cm4.dtb	06/11/2020 5:59 SA	DTB File	47 KB
bootcode.bin	06/11/2020 6:00 SA	BIN File	52 KB
cmdline.txt		Text Document	1 KB
config.txt	06/11/2020 6:59 SA	Text Document	2 KB
COPYING.linux	06/11/2020 5:59 SA	LINUX File	19 KB
fixup.dat	06/11/2020 5:59 SA	DAT File	8 KB
fixup_cd.dat	06/11/2020 5:59 SA	DAT File	4 KB
fixup_db.dat	06/11/2020 6:00 SA	DAT File	11 KB
fixup_x.dat	06/11/2020 6:00 SA	DAT File	11 KB
fixup4.dat	06/11/2020 6:00 SA	DAT File	6 KB
fixup4cd.dat	06/11/2020 6:00 SA	DAT File	4 KB
fixup4db.dat	06/11/2020 6:00 SA	DAT File	9 KB
fixup4x.dat	06/11/2020 6:00 SA	DAT File	9 KB
issue.txt	20/08/2020 10:54 SA	Text Document	1 KB
kernel.img	06/11/2020 5:59 SA	Disc Image File	5.326 KB
kernel7.img	06/11/2020 5:59 SA	Disc Image File	5.638 KB
kernel7l.img	06/11/2020 5:59 SA	Disc Image File	5.989 KB
kernel8.img	06/11/2020 5:59 SA	Disc Image File	15.661 KB
LICENCE.broadcom	06/11/2020 6:00 SA	BROADCOM File	2 KB
start.elf	06/11/2020 5:59 SA	ELF File	2.881 KB
start_cd.elf	06/11/2020 5:59 SA	ELF File	768 KB
start_db.elf	06/11/2020 5:59 SA	ELF File	4.688 KB
start_x.elf	06/11/2020 5:59 SA	ELF File	3.622 KB
start4.elf	06/11/2020 6:00 SA	ELF File	2.174 KB
start4cd.elf	06/11/2020 6:00 SA	ELF File	768 KB
start4db.elf	06/11/2020 6:00 SA	ELF File	3.641 KB
start4x.elf	06/11/2020 6:00 SA	ELF File	2.916 KB
SHH	08/11/2020 11:27 SA	File	0 KB

### Bước 3. Tiến hành cài đặt Raspberry Pi 4

Bạn cắm thẻ nhớ vừa mới tạo vào Raspberry Pi 4, chọn một cáp mạng và cắm vào trong modem wifi hoặc Switch nhé. Cấp nguồn cho máy tính Raspberry .

Lúc này Raspberry sẽ khởi động có đèn xanh nhấp nháy , cổng mạng sẽ sáng màu xanh.



Nhiệm vụ lúc này là mình cần biết địa chỉ IP chính xác mà Raspberry nhận là bao nhiêu. Có 2 cách bạn có thể dùng phần mềm quét tất cả các IP trong mạng, và thấy IP tên raspberry thì đó chính là IP bạn cần tìm. Mình thì quản lý được Modem wifi nhà mình nên sẽ truy cập trực tiếp và tìm IP của Raspberry luôn.

**TOTO LINK**  
The Smartest Network Device

Model No. N200RE (Firmware V3.1.2)

Status

Operation Mode

Network

WAN Settings  
 LAN Settings  
 IPTV Settings  
 DHCP&ARP Tables  
 Static Route  
 Routing Table

IPv6 Network

Wireless

QoS

Firewall

System

DHCP Lease Time  
 Static DHCP  
 Domain Name  
 802.1d Spanning Tree  
 For IP Routing  
 Second IP Address  
 Second Subnet Mask  
 Apply Refresh

480 (1 ~ 10080 minutes)  
 Set Static DHCP  
 TOTOLINK  
 Disabled  
☒ Disabled ☐ Enabled  
 0.0.0.0  
 0.0.0.0

DHCP Clients Table

Host Name	IP Address	MAC Address	Remaining Lease Time (in seconds)
OPPO-A3s	192.168.10.230	50:29:f5:cd:4b:c7	25828
DESKTOP-1BTSTQ9	192.168.10.231	6c:71:d9:99:9c:01	28735
raspberrypi	192.168.10.233	dc:a6:32:c1:b3:a7	28616
raspberrypi	192.168.10.234	dc:a6:32:c1:b3:a8	28618
Galaxy-A8-Star	192.168.10.237	d4:e6:b7:40:03:43	14985

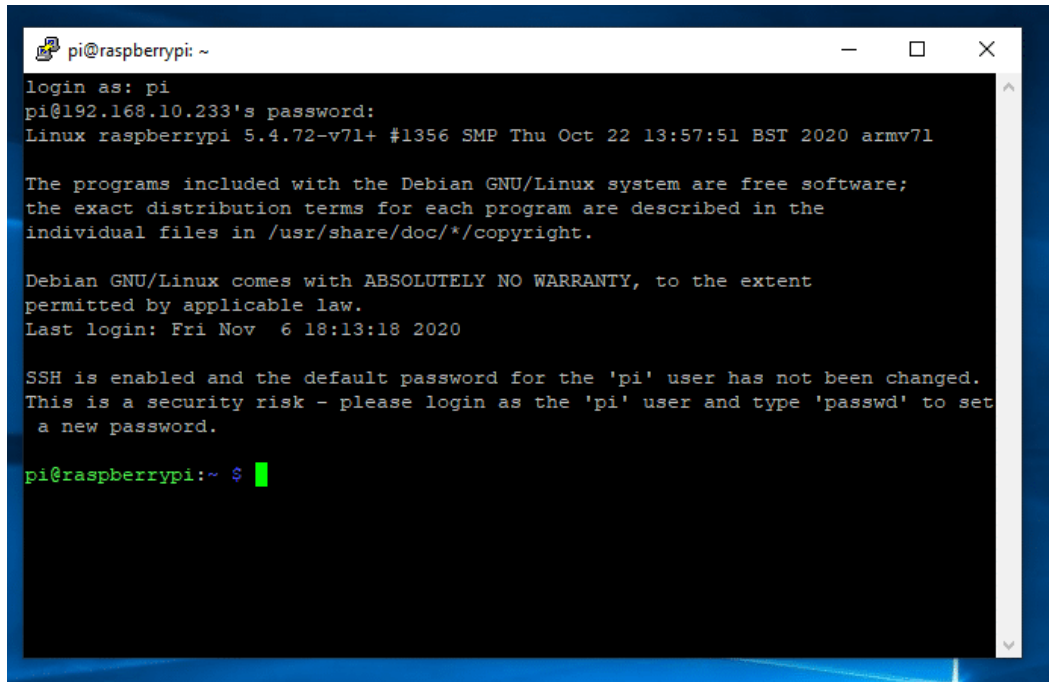
#### Bước 4. Truy cập Raspberry Pi 4 bằng phần mềm PUTTY

Mục đích của bước này là ta vào được cấu hình cơ bản của Raspberry để bật phần mềm điều khiển VNC truy cập thông qua địa chỉ IP mới tìm được bên trên.

Bật phần mềm **PUTTY** mới vừa Download bên trên.

Đăng Nhập: *pi*

Pass: *raspberrypi*



```
pi@raspberrypi: ~
login as: pi
pi@192.168.10.233's password:
Linux raspberrypi 5.4.72-v7l+ #1356 SMP Thu Oct 22 13:57:51 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov  6 18:13:18 2020

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

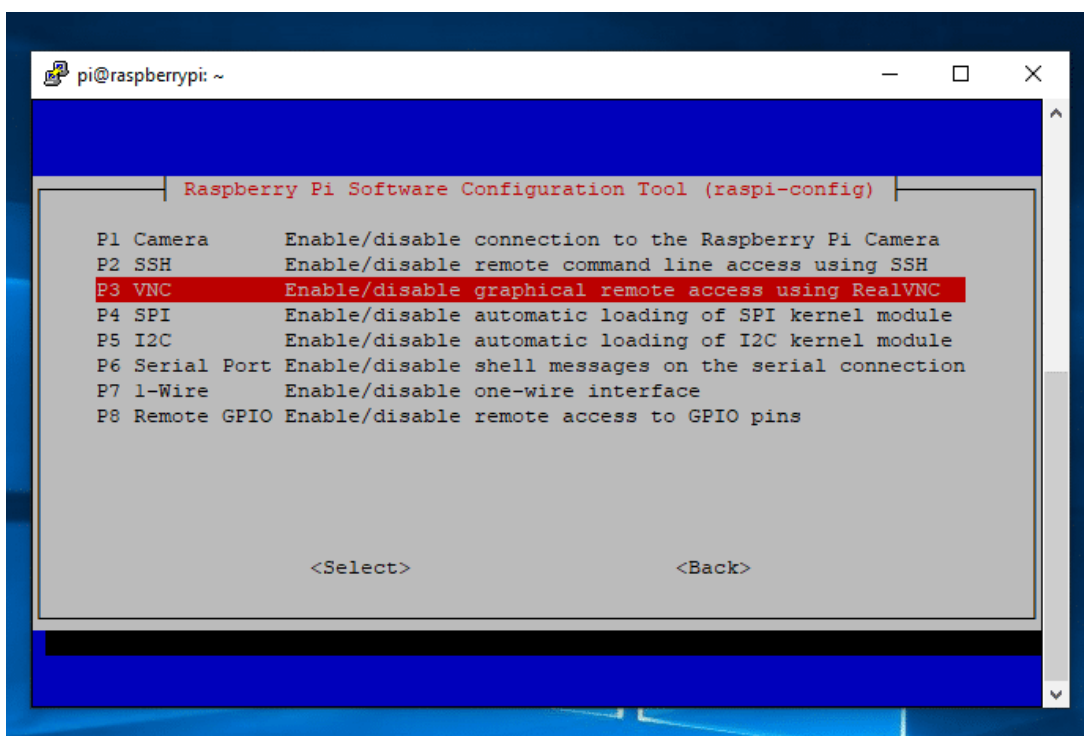
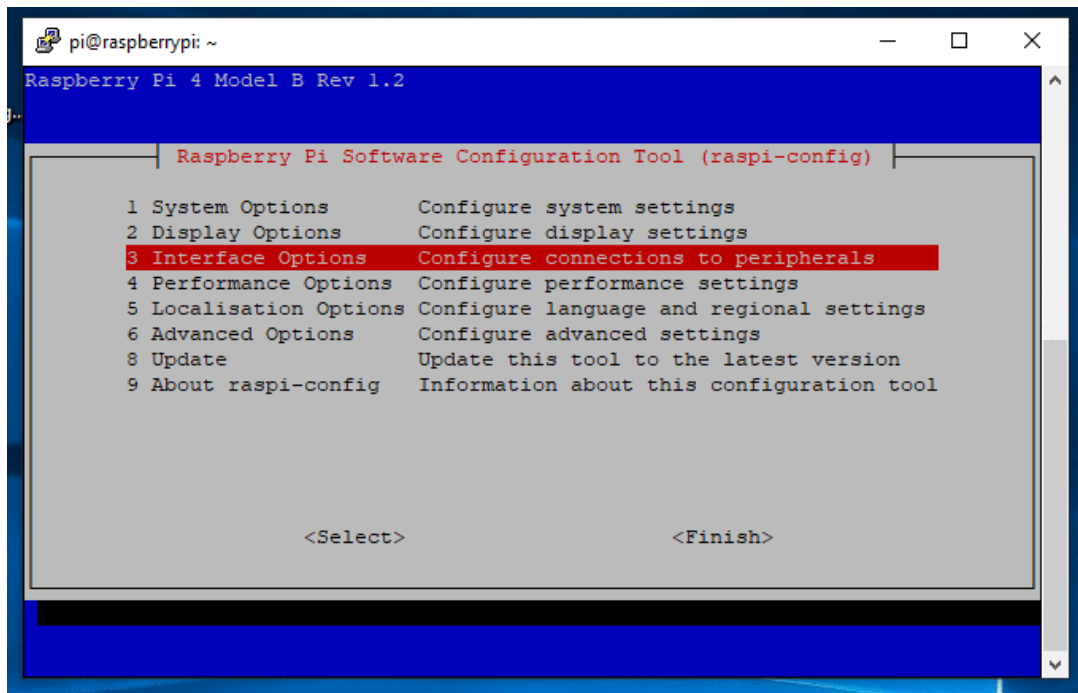
pi@raspberrypi:~ $
```

Đăng nhập thành công bạn sẽ có màn hình như trên.

Tiếp theo bạn gõ lệnh: *sudo raspi-config*.

Đến tiến hành vào giao diện thiết lập của Raspberry Pi.



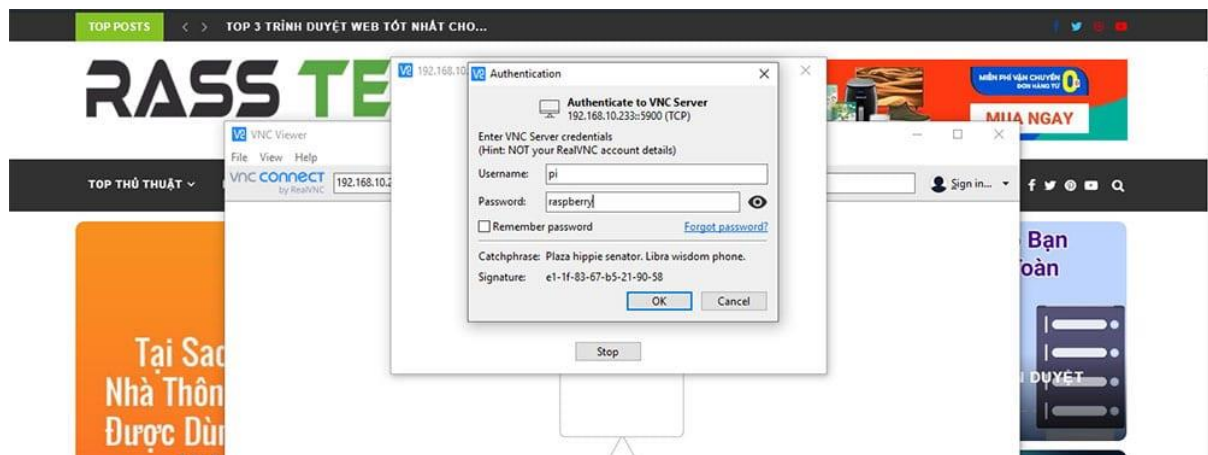


Các bạn chọn mục thứ 5: **Interface Options**.

Sau đó chọn mục P3: **VNC** và bật nó lên nhé. Sau đó, bạn đến bước cuối cùng và tận hưởng thành quả thôi nào.

### **Bước 5 Điều khiển từ xa Raspberry Pi 4 với VNC**

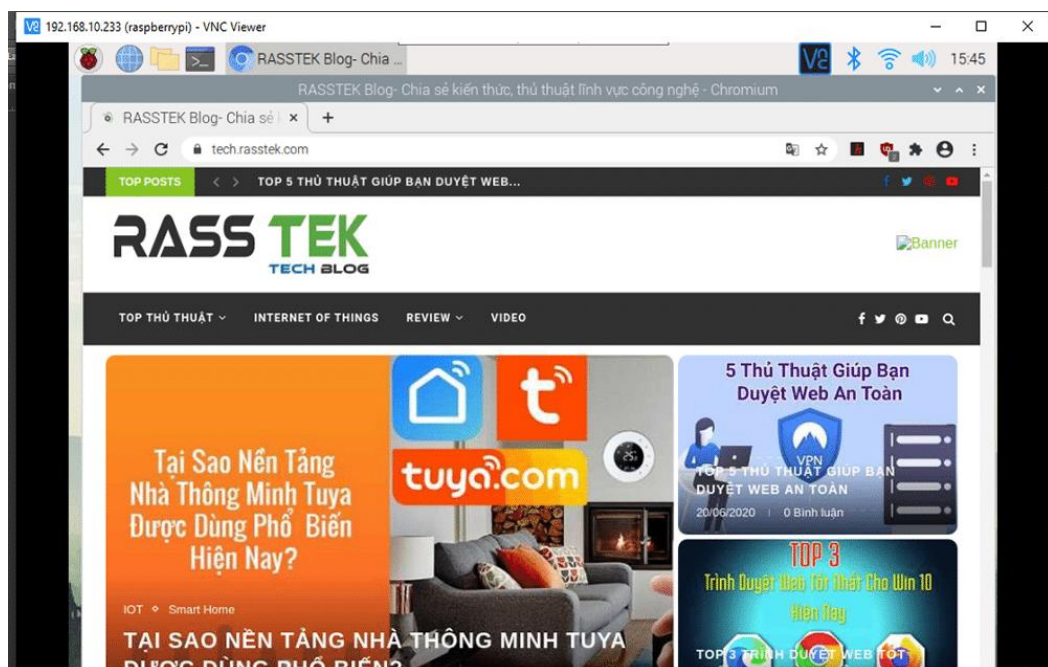
Bạn cài đặt phần mềm VNC vào máy nhé, và nhớ địa chỉ IP của máy tính nhúng Raspberry pi của mình.



Khi mở VNC thành công bạn cần đăng nhập bằng

Username: *pi*

Pass: *raspberrypi*



## 4.2. Tổng quan về lập trình python

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.

### 4.2.1. Biến trong Python

Biến là một vị trí trong bộ nhớ được sử dụng để lưu trữ dữ liệu (giá trị). Biến được đặt tên duy nhất để phân biệt giữa các vị trí bộ nhớ khác nhau. Các quy tắc để viết tên một biến giống như quy tắc viết các định danh trong Python. Trong Python, bạn không cần khai báo biến trước khi sử dụng, chỉ cần gán cho biến một giá trị và nó sẽ tồn tại. Cũng không cần phải khai báo kiểu biến, kiểu biến sẽ được nhận tự động dựa vào giá trị mà bạn đã gán cho biến.

#### Gán giá trị cho biến:

Để gán giá trị cho biến ta sử dụng toán tử =. Bất kỳ loại giá trị nào cũng có thể gán cho biến hợp lệ. VD:

```
hoa = "Hồng"  
la = 3  
canh = 5.5
```

### 4.2.2. Kiểu dữ liệu số trong Python

Python hỗ trợ số nguyên, số thập phân và số phức, chúng lần lượt được định nghĩa là các lớp int, float, complex trong Python. Số nguyên và số thập phân được phân biệt bằng sự có mặt hoặc vắng mặt của dấu thập phân. Ví dụ: 5 là số nguyên, 5.0 là số thập phân. Python cũng hỗ trợ số phức và sử dụng hậu tố j hoặc J để chỉ phần ảo. Ví dụ: 3+5j. Ngoài int và float, Python hỗ trợ thêm 2 loại số nữa là Decimal và Fraction. Ta sẽ dùng hàm type() để kiểm tra xem biến hoặc giá trị thuộc lớp số nào và hàm isinstance() để kiểm tra xem chúng có thuộc về một class cụ thể nào không.

### 4.2.3. Cấu trúc lệnh if trong Python

```
if điều kiện  
    khối lệnh
```



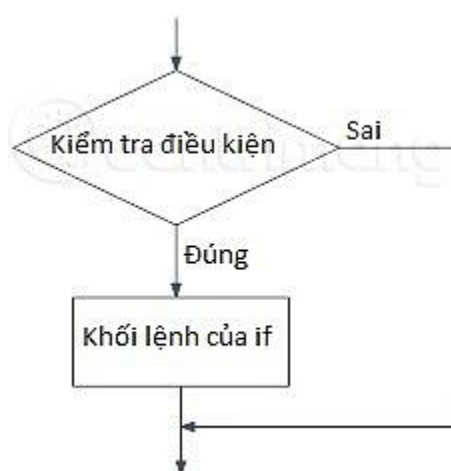
Ở đây, chương trình đánh giá điều kiện và sẽ thực hiện các lệnh khi điều kiện là True. Nếu điều kiện False thì lệnh sẽ không được thực hiện. Trong Python, khối lệnh của lệnh if được viết thụt lề vào trong. Khối lệnh của if bắt đầu với một khoảng thụt lề và dòng không thụt lề đầu tiên sẽ được hiểu là kết thúc lệnh if.

Sơ đồ lệnh if trong Python

Ở đây, chương trình đánh giá *điều kiện* và sẽ thực hiện các *lệnh* khi *điều kiện* là True. Nếu *điều kiện* False thì *lệnh* sẽ không được thực hiện.

Trong Python, khối lệnh của lệnh *if* được viết thụt lề vào trong. Khối lệnh của *if* bắt đầu với một khoảng thụt lề và dòng không thụt lề đầu tiên sẽ được hiểu là kết thúc lệnh *if*.

**Sơ đồ lệnh if trong Python**



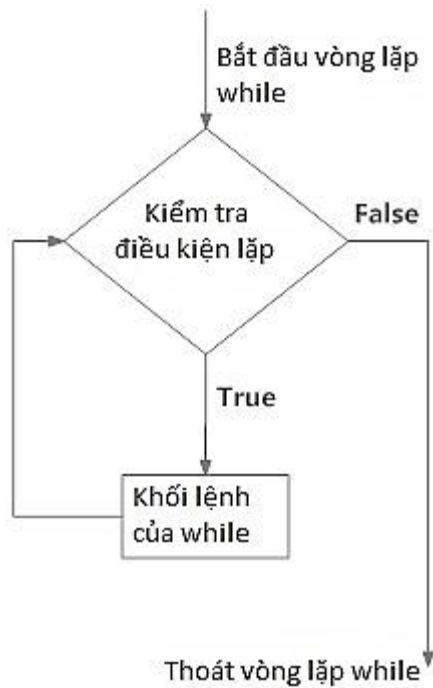
#### 4.2.4. Vòng lặp while trong Python

Vòng lặp được sử dụng trong lập trình để lặp lại một đoạn code cụ thể, while cũng là một trong số đó. Trong Python, while được dùng để lặp lại một khối lệnh, đoạn code khi điều kiện kiểm tra là đúng. while dùng trong những trường hợp mà chúng ta không thể dự đoán trước được số lần cần lặp là bao nhiêu.

Cú pháp của while trong Python:

```
while điều_kiện_kiểm_tra:
    Khối_lệnh_của_while
```

Trong vòng lặp while, điều\_kiện\_kiểm\_tra sẽ được kiểm tra đầu tiên. Khối lệnh của vòng lặp chỉ được nạp vào nếu điều\_kiện\_kiểm\_tra là True. Sau một lần lặp, điều\_kiện\_kiểm\_tra sẽ được kiểm tra lại. Quá trình này sẽ tiếp tục cho đến khi điều\_kiện\_kiểm\_tra là False. Trong Python mọi giá trị khác 0 đều là True, None và 0 được hiểu là False. Đặc điểm này của while có thể dẫn đến trường hợp là while có thể không chạy vì ngay lần lặp đầu tiên điều\_kiện\_kiểm\_tra đã False. Khi đó, khối lệnh của while sẽ bị bỏ qua và phần code ngay sau đó sẽ được thực thi.



#### 4.2.5. Hàm trong Python

Trong Python, hàm là một nhóm các lệnh có liên quan đến nhau được dùng để thực hiện một tác vụ, nhiệm vụ cụ thể nào đó. Hàm giúp chia chương trình Python thành những khối/phần/mô-đun nhỏ hơn. Khi chương trình Python quá lớn, hoặc cần mở rộng, thì các hàm giúp chương trình có tổ chức và dễ quản lý hơn.

Hàm còn có một tác dụng vô cùng quan trọng nữa là tránh việc phải lặp lại code để thực thi những tác vụ tương tự nhau, giúp code gọn hơn và có thể tái sử dụng. Ví dụ, nếu bạn cần viết một ứng dụng vẽ hàng trăm hình tam giác để tạo hiệu ứng vạn hoa, bạn có thể làm theo hai cách:

- Không sử dụng hàm: Bạn lặp lại mã để vẽ từng hình tam giác một
- Có sử dụng hàm: Bạn tạo ra một loạt tọa độ và đưa chúng vào hàm vẽ tam giác

Cách thứ 2 hiệu quả hơn, phải viết ít code hơn và thường là phương pháp ưa thích của các lập trình viên. Không chỉ vậy, với cách này nếu muốn thay đổi từ tam giác sang hình vuông, bạn chỉ cần đổi vài dòng code.

Cú pháp của hàm Python:

```
def ten_ham(các tham số/đối số):  
    Các câu lệnh
```

Về cơ bản, một định nghĩa hàm Python sẽ bao gồm các thành phần sau:

1. Từ khóa def: Đánh dấu sự bắt đầu của tiêu đề hàm.

2. `ten_ham`: Là định danh duy nhất dành cho hàm. Việc đặt tên hàm phải tuân thủ theo quy tắc viết tên và định danh trong Python.
3. Các tham số/đối số: Chúng ta truyền giá trị cho hàm thông qua các tham số này. Chúng là tùy chọn.
4. Dấu hai chấm (:): Đánh dấu sự kết thúc của tiêu đề hàm.
5. `docstring`: Chuỗi văn bản tùy chọn để mô tả chức năng của hàm.
6. Các câu lệnh: Một hoặc nhiều lệnh Python hợp lệ tạo thành khối lệnh. Các lệnh này phải có cùng một mức thụt đầu dòng (thường là 4 khoảng trắng).
7. Lệnh `return`: Lệnh này là tùy chọn, dùng khi cần trả về giá trị từ hàm.

## CHƯƠNG 5: THỬ NGHIỆM VÀ ĐÁNH GIÁ

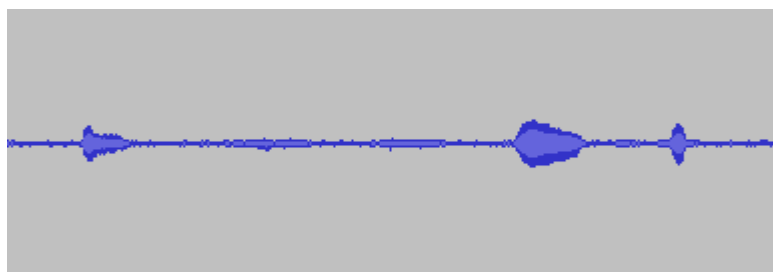
### 5.1. Thử nghiệm tổng hợp tiếng nói

Kết quả thử nghiệm với một số đoạn text mẫu:

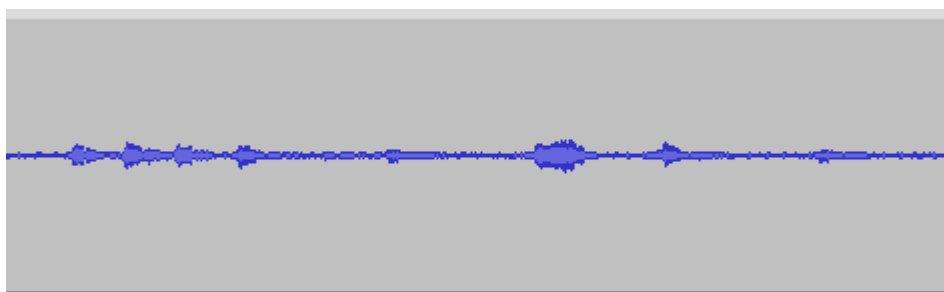
- “Xin chào tất cả mọi người”:



- “Đề tài nghiên cứu khoa học sinh viên”:



- “Bây giờ là mấy giờ”:



### 5.2. Thử nghiệm hỏi đáp

**Kết quả thử nghiệm với một kịch bản hỏi đáp đơn giản:**

Người dùng: hello

Robot: Xin chào, bạn cần tôi giúp gì?

Người dùng: Hiệu trưởng trường Đại học Kỹ thuật Công Nghiệp là ai?

Robot: Tiến sĩ Nguyễn Duy Cường.

Người dùng: hello

Robot: Xin chào, bạn cần tôi giúp gì?

Người dùng: Sân thể dục ở đâu?

Robot: Đi thẳng 2km sau đó rẽ phải là tới.

Người dùng: hello

Robot: Xin chào, bạn cần tôi giúp gì?

Người dùng: Học TNUT có khó không?

Robot: dễ trượt.

Người dùng: hello

Robot: Xin chào, bạn cần tôi giúp gì?

Người dùng: Đi chơi không?

Robot: Xin lỗi, câu hỏi của bạn không có trong danh sách.

Người dùng: hello

Robot: Xin chào, bạn cần tôi giúp gì?

Người dùng: Tôi muốn thêm câu hỏi.

Robot: Nhập vào từ khóa của câu hỏi

Người dùng: ngành kỹ thuật máy tính có bao nhiêu lớp.

Robot: Nhập câu trả lời của câu hỏi trên

Người dùng: 5 lớp

Robot: Có phải bạn muốn nhập câu trả lời 5 lớp cho câu hỏi ngành kỹ thuật máy tính có bao nhiêu lớp. Nếu đúng hãy nói có.

Người dùng: Có

Robot: Thêm dữ liệu thành công.

## 5.3. Đánh giá và Hướng phát triển

### 5.3.1 Đánh giá

#### Ưu điểm:

- Chương trình đã phần nào giải quyết được các yêu cầu thực tế của sinh viên như: tìm hiểu các thông tin về trường, ngành học, các hoạt động ngoại khóa, thể thao,...
- Vận dụng kỹ thuật tổng hợp tiếng nói truy vấn thành công, tạo được đầu ra audio với chất giọng riêng, tốc độ xử lý nhanh, tiêu tốn ít CPU.

#### Nhược điểm:

- Hệ thống chatbot chưa thông minh, chỉ hoạt động dựa theo việc bắt keyword của câu hỏi, không nắm được ý định của người dùng.
- Chương trình tổng hợp tiếng nói còn nhiều thiếu sót, giọng đọc chưa được tự nhiên, tốc độ đọc chậm, không biểu thị được ngữ điệu lời nói.

### **5.3.2 Hướng phát triển của đề tài.**

Trong thời gian tới, em sẽ khắc phục những nhược điểm của đề tài, cụ thể:

- Làm cho chatbot trở nên thông minh hơn bằng cách sử dụng machine learning để xác định nội dung câu hỏi và ý định người dùng, từ đó đưa ra câu trả lời một cách chính xác.
- Tiếp tục nghiên cứu, phát triển mô hình tổng hợp tiếng nói truy vấn để giọng nói từ robot tự nhiên hơn và đúng ngữ điệu mà người dùng mong muốn.

Bên cạnh đó, em sẽ nghiên cứu tích hợp chức năng chatbot thông minh với chức năng nhận dạng mặt người và các chức năng khác để tạo ra một robot iTNUT hoàn chỉnh, đa chức năng và đáp ứng nhiều hơn nhu cầu của sinh viên TNUT.

## TÀI LIỆU THAM KHẢO

- [1] Alan, *What is voice assistants*, 27/05/2022, <https://alan.app/blog/voiceassistant-2/>.
- [2] Wikipedia, *Tổng hợp giọng nói*, [https://vi.wikipedia.org/wiki/tổng\\_hợp\\_giọng\\_nói](https://vi.wikipedia.org/wiki/tổng_hợp_giọng_nói)

## Phụ Lục: Code chương trình

File **tong\_hop\_tiang\_noi.py**

```
#!/usr/bin/env python3
from pydub import AudioSegment
from playsound import playsound
from pydub import AudioSegment
from pydub.playback import play
import time
donvi = ["", "một ", "hai ", "ba ", "bốn ", "năm ", "sáu ", "bảy ", "tám ", "chín " ]
hangChuc = ["linh ", "một ", "hai ", "ba ", "bốn ", "năm ", "sáu ", "bảy ", "tám ", "chín "]
def num999c2(n):
    c = int(n % 10) # hang don vi
    b = int(((n % 100) - c) / 10) # hang chuc
    a = int(((n % 1000) - (b * 10) - c) / 100) # hang tram

    st=donvi[a]+"trăm " +hangChuc[b] +donvi[c]
    return st
#----- Xử lý chuỗi đầu vào-----
def tong_hop(x):
    x=x.split(" ")
    textArr=[]
    for e in x:
        try:
            tg=int(e)
            numberToArr=num999c2(tg)
            numberToArr=numberToArr.strip().split(" ")
            for n in numberToArr:
                textArr.append(n.lower())

        except:
            textArr.append(e.lower())

    strdir="banmai/" #dương dan tổ thu mục 3000 file audio

    arrHasdri=[]
    for ele in textArr:
        arrHasdri.append(strdir+ele+".mp3")

    # print(arrHasdri)

    #----- Load file theo đường dẫn rồi cộng
file -----
    S=0
    for ele1 in arrHasdri:
        S+=AudioSegment.from_mp3(ele1)
```



```

S.export("KQ.mp3", format="mp3")
#----- Hiển thị file kết quả ra màn hình-----
-----

tong_hop("xin chào tất cả mọi người")
def speed_change(sound, speed=1.0):
    # Manually override the frame_rate. This tells the computer how many
    # samples to play per second
    sound_with_altered_frame_rate = sound._spawn(sound.raw_data, overrides={
        "frame_rate": int(sound.frame_rate * speed)
    })
    # convert the sound with altered frame rate to a standard frame rate
    # so that regular playback programs will work right. They often only
    # know how to play audio at standard frame rate (like 44.1k)
    return sound_with_altered_frame_rate.set_frame_rate(sound.frame_rate)

```

### File **nhandang.py**:

```

# Import library
import speech_recognition as sr
import pyttsx3
from gtts import gTTS
from playsound import playsound
import pyaudio
import pygame
import pandas as pd
import random
import os
from tong_hop_tiang_noi import tong_hop

# Tìm keyword từ text, keyword from question column of dataframe
def findKeyword(text, listKW):
    for kw in listKW:
        if kw in text.lower():
            return kw

# Speech to text using google api
def speechToText(r):
    with sr.Microphone() as source2:
        audio = r.listen(source2)
        text = r.recognize_google(audio, language="vi-VI")
        return text

# Random answer from list answers where question data = keyword
def randomAnswer(keyword, data):
    dfByKeyword = data[data["question"].str.lower()==keyword]
    if dfByKeyword.empty==False:

```

```

        answers = dfByKeyword["answer"].tolist()
        an = answers[0].split("-")
        return random.choice(an)
    else: return "Câu hỏi của bạn không có trong danh sách"

md = 1
# Convert text to speech using gtts
def textToSpeech(text):
    if md != 1:
        tong_hop(text)
        playsound("KQ.mp3")
        os.remove("KQ.mp3")
        return 0
    tts = gTTS(text, lang='vi', slow=False)
    tts.save("Result.mp3")
    playsound("Result.mp3")
    os.remove("Result.mp3")

# Regconize and answer
def regconize(df, robotName, r):
    t = speechToText(r)
    if robotName.lower() in t.lower() :
        textToSpeech("xin chào bạn cần tôi giúp gì")
        text = speechToText(r)
        print (text)
        if "thêm câu hỏi" in text.lower():
            addQuestion(df, r)
        else:
            listKeyword = df["question"].values
            keyword = findKeyword(text, listKeyword)
            answer = randomAnswer(keyword, df)
            textToSpeech(answer)

# Add a row to csv file
def writeDataToCsv(keyword, answer):
    with open('cauhoi.csv', 'a') as fd:
        fd.write("\n"+keyword + ";" + answer)
        fd.close()

#Append data to answer
def appendAnswer(keyword, answer, df):
    df.loc[df['question'] == keyword, 'answer'] += "-" + answer
    df.to_csv("cauhoi.csv", sep=';', index=False, encoding='utf-8')

# Add a question
def addQuestion(df, r):
    textToSpeech("Nhập vào từ khóa của câu hỏi")
    txt_label = speechToText(r)

```

```

dfByKeyword = df[df["question"] == txt_label]
textToSpeech("Nhập câu trả lời của câu hỏi trên")
txt_answer = speechToText(r)
if dfByKeyword.empty:
    textToSpeech("có phải bạn muốn nhập câu trả lời "+txt_answer+" cho câu
hỏi "+txt_label+". Nếu đúng hãy nói có")
    txtConfirm = speechToText(r)
    if "có" in txtConfirm.lower():
        writeToCsv(txt_label, txt_answer)
        textToSpeech("Thêm câu hỏi thành công")
    else: return

else:
    if txt_answer in dfByKeyword["answer"].tolist()[0].split("-"):
        textToSpeech("Câu hỏi và câu trả lời của bạn đã tồn tại trên hệ
thống.")
    else:
        textToSpeech("có phải bạn muốn nhập câu trả lời "+txt_answer+" cho
câu hỏi "+txt_label+". Nếu đúng hãy nói có")
        txtConfirm = speechToText(r)
        if "có" in txtConfirm.lower():
            appendAnswer(txt_label, txt_answer, df)
            textToSpeech("Thêm trả lời thành công")
        else: return

# Main program
if __name__ == "__main__":
    # tong_hop("tiến sĩ nguyên duy cương")
    # playsound("KQ.mp3")
    ROBOT_NAME = "hello"
    r = sr.Recognizer() # Initialize the recognizer
    speechToText(r)
    while (1):
        df = pd.read_csv("cauhoi.csv", sep=";") # Read csv file
        try:
            regconize(df, ROBOT_NAME, r)
        except sr.RequestError as e:
            print("Lỗi request; {0}".format(e))
        except sr.UnknownValueError:
            print("Không nhận dạng được")

```

**THUYẾT MINH ĐỀ TÀI**  
**NGHIÊN CỨU KHOA HỌC SINH VIÊN NĂM 2022**

<b>1. TÊN ĐỀ TÀI: Robot iTNUT. Module 02: Xây dựng phần mềm nhận dạng tiếng nói tiếng Việt cho một tập câu hỏi đơn giản và tự động trả lời.</b>	<b>2. Mã số: SV2022-02</b>																
<b>3. THỜI GIAN THỰC HIỆN</b> (thực hiện theo năm học): <b>12 tháng</b> Từ tháng 03 năm 2022 đến tháng 04 năm 2023																	
<b>4. Đơn vị thực hiện: Khoa Điện tử</b>																	
<b>5. CHỦ NHIỆM ĐỀ TÀI</b> <table><tr><td>Họ và tên: Dương Văn Huân</td><td>Lớp: K54KMT</td></tr><tr><td>Mã số SV: K185480106010</td><td></td></tr><tr><td>Điện thoại di động: 0337674951</td><td>E-mail: K185480106010@tnut.edu.vn</td></tr></table> <b>CÔNG TÁC VIÊN</b> <table><tr><td>Họ và tên: Hà Văn Khánh</td><td>Lớp: K56KMT</td></tr><tr><td>Mã số SV: K205480106041</td><td></td></tr><tr><td>Điện thoại di động: 0326091692</td><td>E-mail: K205480106041@tnut.edu.vn</td></tr></table> <b>GIÁO VIÊN HƯỚNG DẪN:</b> <table><tr><td>Họ và tên: TS. Nguyễn Văn Huy</td><td></td></tr><tr><td>Điện thoại di động: 0968852824</td><td>E-mail: huynguyen@tnut.edu.vn</td></tr></table>		Họ và tên: Dương Văn Huân	Lớp: K54KMT	Mã số SV: K185480106010		Điện thoại di động: 0337674951	E-mail: K185480106010@tnut.edu.vn	Họ và tên: Hà Văn Khánh	Lớp: K56KMT	Mã số SV: K205480106041		Điện thoại di động: 0326091692	E-mail: K205480106041@tnut.edu.vn	Họ và tên: TS. Nguyễn Văn Huy		Điện thoại di động: 0968852824	E-mail: huynguyen@tnut.edu.vn
Họ và tên: Dương Văn Huân	Lớp: K54KMT																
Mã số SV: K185480106010																	
Điện thoại di động: 0337674951	E-mail: K185480106010@tnut.edu.vn																
Họ và tên: Hà Văn Khánh	Lớp: K56KMT																
Mã số SV: K205480106041																	
Điện thoại di động: 0326091692	E-mail: K205480106041@tnut.edu.vn																
Họ và tên: TS. Nguyễn Văn Huy																	
Điện thoại di động: 0968852824	E-mail: huynguyen@tnut.edu.vn																

## **6. TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU THUỘC LĨNH VỰC CỦA ĐỀ TÀI**

- Tổng quan tình hình nghiên cứu thuộc lĩnh vực của đề tài

Robot thông minh được nghiên cứu và phát triển từ rất sớm đầu thế kỷ 19, robot thông minh là một hệ thống máy được cải thiện về khả năng nhận thức, ra quyết định và thực thi nhiệm vụ theo cách toàn diện hơn so với robot truyền thống. Chúng có thể mô phỏng ứng xử, cảm xúc và suy nghĩ giống người. Với một “bộ não” thông minh, robot có thể thực hiện theo chỉ dẫn của người vận hành, sau đó hoàn thành nhiều nhiệm vụ đã được lập trình trước, tự học và nâng cấp ứng xử của chúng trong lúc tương tác với con người. Xuất phát từ môi trường ứng dụng, có thể phân chia robot thông minh thành các loại robot sau:

Robot công nghiệp: thường là tay máy nhiều khớp (nhiều bậc tự do) ứng dụng trực tiếp cho công nghiệp và là robot chiếm hơn 60% giá trị thị trường robot toàn cầu. Đây là thiết bị máy luôn ứng dụng đầu tiên những kết quả nghiên cứu phát triển công nghệ mới nhất để có thể tự điều khiển và hoàn thành rất nhiều công việc khác nhau.

Robot dịch vụ trong nhà: ví dụ như robot quét và lau cửa sổ, có thể làm việc giống trợ lý giúp việc trong hộ gia đình. Chúng cũng có khả năng tìm kiếm thăm dò, tự động lập kế hoạch đường đi và tránh vật cản. Nhiều loại robot dịch vụ khác, như robot xã hội hóa gia đình, robot bầu bạn, robot trợ lý di động, robot huấn luyện thú cưng, đều có khả năng tương tác với người, cũng như hoàn thành các nhiệm vụ được giao, chăm nom người già và trẻ nhỏ, nhắc nhở sự kiện và tuần tra nhà. Bên cạnh đó, còn có robot giao tiếp cảm xúc, robot giáo dục trẻ nhỏ, robot nền học thông minh, UAV cá nhân, robot di động cá nhân... được hỗ trợ thêm công nghệ tương tác giọng nói cho phép tương tác giao tiếp và cảm nhận cảm xúc của con người. Nhiều robot dịch vụ thương mại, bao gồm cả robot nhận thức, robot hướng dẫn mua hàng, robot nấu ăn, robot văn phòng, robot an ninh... có thể tùy biến dịch vụ cá nhân theo tình huống ứng dụng cụ thể và hoàn thành nhiều nhiệm vụ như quảng cáo, cung cấp chỉ dẫn, tư vấn hỗ trợ, trợ lý công việc văn phòng, thực hiện tuần tra an ninh...

- Tài liệu tham khảo

## 7. TÍNH CẤP THIẾT CỦA ĐỀ TÀI

AI hiện đang làm nên cuộc cách mạng giải phóng sức lao động trí não của con người. Một trong lĩnh vực ứng dụng công nghệ này là Robot thông minh. Với mục tiêu xây dựng robot có các hành vi tương tự con người phục vụ giải trí, nghiên cứu thực nghiệm AI tại TNUT, đề tài thực hiện nhằm xây dựng một phiên bản robot sử dụng trí tuệ nhân tạo thực hiện các công việc bước đầu đơn giản như vận chuyển hàng hóa, giao tiếp đa ngôn ngữ, marketing cho tnut. Robot có khả năng nâng cấp các kỹ năng theo thời gian.

## 8. MỤC TIÊU ĐỀ TÀI

Đề tài này là một nhánh trong mục tiêu xây dựng Robot iTNUT có khả năng: Tự động nhận diện người đối diện; Tự động giao tiếp trả lời các câu hỏi đơn giản về TNUT bằng tiếng nói; Tự động di chuyển đến đích theo hành trình.

Mục tiêu chính của đề tài này là: **Xây dựng phần mềm nhận dạng được 1 tập các câu hỏi bằng tiếng nói và tự động trả lời bằng tiếng nói. Phần mềm được cài đặt và vận hành trên máy tính nhúng Raspberry Pi3.**

## 9. NỘI DUNG NGHIÊN CỨU VÀ TIẾN ĐỘ THỰC HIỆN

STT	Các nội dung, công việc thực hiện chủ yếu	Sản phẩm phải đạt	Thời gian (bắt đầu-kết thúc)	Người thực hiện
1	Tìm hiểu và xây dựng phần mềm nhận dạng tiếng nói qua microphone sử dụng các pretrained model hoặc các API sẵn có cho một tập các câu hỏi chọn trước.	Phần mềm nhận dạng tiếng nói chạy trên raspberry	04/2022-08/2022	Dương Văn Huân
2	Tìm hiểu và xây dựng phần mềm tích hợp module nhận dạng và tự chọn câu trả lời bằng tiếng nói phát qua loa sử dụng raspberry	Phần mềm phát âm thanh qua loa trên raspberry	09/2022-12/2022	Hà Văn Khánh
3	Viết thuyết minh và báo cáo đề tài	Tài liệu báo cáo đề tài	01/2023-04/2023	Dương Văn Huân

<b>10. SẢN PHẨM</b>			
STT	Tên sản phẩm	Số lượng	Yêu cầu chất lượng sản phẩm
I	Sản phẩm khoa học	0	
II	Sản phẩm ứng dụng		
2.1	Phần mềm nhận và tự động trả lời một tập các câu hỏi chọn trước bằng tiếng nói chạy trên Raspberry	01	<ul style="list-style-type: none"> <li>- Tự động thu tín hiệu từ microphone của Raspberry.</li> <li>- Tự động nhận dạng các câu hỏi bằng tiếng nói được chọn trước trong cơ sở dữ liệu.</li> <li>- Tự động tổng hợp tiếng nói các câu trả lời và phát ra loa.</li> <li>- Phần mềm chạy trên môi trường phần cứng Raspberry.</li> </ul>
<b>11. KINH PHÍ THỰC HIỆN ĐỀ TÀI</b> <b>Tổng kinh phí: 3.000.000 VNĐ</b> <b>Bằng chữ: Ba triệu đồng chẵn.</b>			
<div> <div>Ngày 29 tháng 03 năm 2022</div> <div> <div>CHỦ NHIỆM ĐỀ TÀI</div> <div>GIÁO VIÊN HƯỚNG DẪN</div> <div>XÁC NHẬN CỦA ĐƠN VỊ</div> </div> </div> <div> <div> <div>Dương Văn Huân</div> <div>TS. Nguyễn Văn Huy</div> <div>PGS.TS. Đào Huy Du</div> </div> <div> <div>TRƯỞNG PHÒNG KHCN&amp;HTQT</div> <div>KT. HIỆU TRƯỞNG</div> <div>PHÓ HIỆU TRƯỞNG</div> </div> </div> <div> <div>PGS.TS. Phạm Thành Long</div> <div>PGS.TS. Vũ Ngọc Pi</div> </div>			

