

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ ĐIỆN TỬ



KHOÁ LUẬN TỐT NGHIỆP

**ĐIỀU KHIỂN THIẾT BỊ BẰNG GIỌNG NÓI ÁP
DỤNG THƯ VIỆN GOOGLE SPEECH
RECOGNITION API**

GVHD: THS CAO VĂN KIÊN

SVTH: LÊ HUY PHÁT

MSSV: 16052431

LỚP: DHDTMT12A

Chuyên ngành: Hệ Thống Nhúng

Mã chuyên ngành: 52510311

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

THÔNG TIN CHUNG

Họ và tên sinh viên: Lê Huy Phát MSHV: 16052431

Lớp : DHDTMT12A Khóa: 12

Chuyên ngành : Hệ Thống Nhúng Mã chuyên ngành: 52510311....

SĐT : 0349055566

Email : phatlh2005002@fpt.edu.vn

Địa chỉ liên hệ : 124/67/18 Xô Viết Nghệ Tĩnh, P21, Bình Thạnh, HCM

Tên đề tài : Điều khiển thiết bị bằng giọng nói áp dụng thư viện Google
speechrecognition API.

Người hướng dẫn : Th.S Cao Văn Kiên

SĐT : 0979896649

Email : caovankien@iuh.edu.vn

Cơ quan công tác : Trường Đại Học Công Nghiệp Tp.HCM

Tp. Hồ Chí Minh, ngày tháng năm 2021

Người hướng dẫn
(Ký và ghi rõ họ tên)

Sinh viên
(Ký và ghi rõ họ tên)

LỜI MỞ ĐẦU

Trong bối cảnh xu thế 4.0 hiện nay, AIOT (Artificial Intelligence of Things) là 1 xu thế, là sự kết hợp giữa hạ tầng Internet of Things (IoT) với công nghệ trí tuệ nhân tạo (AI) để đạt được các hoạt động IoT hiệu quả hơn, giúp cải thiện tương tác giữa máy và người, tăng cường phân tích dữ liệu, khả năng quản lý. Để chuyển đổi dữ liệu IoT thành thông tin hữu ích để cải thiện quá trình ra quyết định hiện nay người ta đã áp dụng AI vào những công việc như vậy, Tạo ra tiền đề và nền tảng cho ngành công nghệ mới hơn như IoT Data as a Service (IoTDataS).

Vào vấn đề chính, xã hội chúng ta hiện nay đang công nghệ hóa mọi thứ, việc làm cho nơi ở hay nơi làm việc của bạn trở nên thông minh và tối ưu hóa công việc giúp bạn nâng cao năng suất làm việc trong ngày là điều mà tất cả chúng ta đều đang quan tâm. Vì lý do đó nên việc áp dụng AIOT vào xây dựng lên một hệ thống có khả năng nhận dạng giọng nói của bạn và phân loại được ý định của bạn muốn làm gì để đưa ra hành động thực thi ngay lập tức dù bạn đang đứng ở đâu chỉ cần trong phạm vi nhận dạng giọng nói của thiết bị thì mọi thứ xung quanh bạn chỉ chờ khẩu lệnh.

Sinh viên thực hiện

Lê Huy Phát

LỜI CẢM ƠN

Lời đầu tiên em xin gửi lời cảm ơn sâu sắc đến toàn thể giảng viên trường *Đại học Công Nghiệp Thành phố Hồ Chí Minh*. Em xin gửi lời cảm ơn chân thành đến Thầy, Cô trong Khoa Điện Tử đã trang bị đầy đủ kiến thức và truyền đạt một cách tâm huyết giúp cho em để có thể hoàn thành tốt khóa luận tốt nghiệp này. Đặc biệt, em xin cảm ơn thầy ThS.Cao Văn Kiên, người hướng dẫn nhiệt tình giúp đỡ, chỉ dẫn tận tình trong suốt quá trình thực hiện khóa luận tốt nghiệp.

Em xin chân thành gửi lời chúc sức khỏe đến tất cả quý thầy cô để có thể tiếp tục sự nghiệp trồng người của mình.

Cảm ơn đến cha mẹ đã tạo điều kiện tốt nhất về kinh tế và tinh thần để con hoàn thành tốt đề tài này.

Đồng cảm ơn đến các anh, các bạn phòng Lab trường Đại học Bách Khoa Thành phố Hồ Chí Minh đã chia sẻ trao đổi kiến thức cũng như những kinh nghiệm quý báu trong thời gian em thực hiện đề tài, và đặc biệt gửi lời cảm ơn đến lab Câu Lạc Bộ Nghiên Cứu Khoa Học – CEE –FET – IUH đã luôn tạo điều kiện để em nghiên cứu và thực hành trong suốt quãng thời gian em hoạt động ở lab.

Em xin chân thành cảm ơn!

TP. Hồ Chí Minh, Ngày tháng Năm 2021

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày tháng năm 2021

Giáo viên hướng dẫn
(Ký tên)

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày tháng năm 2021

Giáo viên phản biện

(Ký tên)

MỤC LỤC

LỜI MỞ ĐẦU	2
LỜI CẢM ƠN	3
NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN	4
NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN	5
DANH MỤC HÌNH ẢNH	8
DANH MỤC BẢNG BIỂU	10
DANH MỤC TỪ VIẾT TẮT.....	11
CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI.....	12
1.1 Đặt vấn đề.....	12
1.2 Mục tiêu của đề tài	12
1.3 Nhiệm vụ đề tài	13
1.4 Đối tượng và phạm vi nghiên cứu.....	13
1.5 Điểm mới của đề tài	13
1.5.1 Các hệ thống đã được thực hiện.....	13
1.5.2 Điểm mới của mô hình.....	15
1.6 Ý nghĩa thực tiễn của đề tài.....	16
CHƯƠNG 2 THIẾT KẾ HỆ THỐNG	17
2.1 Giới thiệu phần cứng sử dụng	17
2.1.1 Vi điều khiển	17
2.1.2 Máy tính nhúng Raspberry Pi3	17
2.1.3 Module thu phát RF Lora SX1278 433MHZ Uart (E32-433T20DT). ..	18
2.2 Giới thiệu phần mềm sử dụng	19
2.2.1 Phần lập trình cho vi điều khiển.....	19
2.2.2 Phần lập trình cho bộ nhận dạng giọng nói, model phân loại văn bản và điều khiển, mô hình website-status	21
2.3 Tổng quan mô hình hệ thống	22
2.4 Thiết kế hệ thống.....	27
2.4.1 Sơ đồ khối thiết bị “Node device” và “Control device”	27
2.4.2 Sơ đồ nguyên lý thiết bị “Node device”.....	29

2.4.3	Sơ đồ mạch in thiết bị “Node device”	33
2.4.4	Sơ đồ nối chân thiết bị “Control device”	35
2.5	Lưu đồ giải thuật	36
2.5.1	Lưu đồ giải thuật của hệ thống	36
2.5.2	Lưu đồ giải thuật của thiết bị “Node device”	38
2.5.3	Lưu đồ giải thuật của thiết bị “Control device”	42
CHƯƠNG 3	KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ	54
3.1	Kết quả đạt được	54
3.1.1	Phân cứng	54
3.1.2	Model Text Classification tích hợp với Control device	56
3.1.3	Webapp	56
3.2	Kết quả thực nghiệm	57
3.3	Nhận xét và đánh giá	63
CHƯƠNG 4	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	72
4.1	Kết luận	72
4.2	Hướng phát triển	72
TÀI LIỆU THAM KHẢO	73

DANH MỤC HÌNH ẢNH

Hình 1 Thiết kế điều khiển thiết bị bằng giọng nói với google assistant (google home) – Tác giả Mai Thanh Tuấn [3]	14
Hình 2. 1 STM32F103C8T6 48pin	17
Hình 2. 2 Raspberry Pi 3 Model B+.....	17
Hình 2. 3 Module thu phát RF LoRa SX1278	19
Hình 2. 4: Giao diện phần mềm STM32 CubeMX	19
Hình 2. 5: Giao diện IDE KeilC5.....	20
Hình 2. 6 Giao diện phần mềm Visual Studio Code	21
Hình 2. 7 Mô tả tổng quát hệ thống	22
Hình 2. 8 Sơ đồ mô tả kết nối của hệ thống.....	23
Hình 2. 9 Cơ chế tổng quát của MQTT	24
Hình 2. 10 Cơ chế hoạt động khi thư viện Recognition gọi API xử lý ngôn ngữ tự nhiên của google.	25
Hình 2. 11 Sơ đồ khối toàn hệ thống	27
Hình 2. 12 Sơ đồ nguyên lý khối nguồn sử dụng.....	29
Hình 2. 13 Sơ đồ nguyên lý khối vi điều khiển	30
Hình 2. 14 Sơ đồ nguyên lý khối module thu phát rf lora và báo hiệu	31
Hình 2. 15 Sơ đồ nguyên lý khối relay	32
Hình 2. 16 Bố trí và sắp xếp linh kiện 3D mặt trên của thiết bị Node device	33
Hình 2. 17 Sơ đồ mạch in thiết bị Node device	34
Hình 2. 18 Lưu đồ giải thuật của hệ thống.....	36
Hình 2. 19 Lưu đồ giải thuật Node device.....	38
Hình 2. 20 Lưu đồ giải thuật hàm con xử lý truyền nhận dữ liệu và thực thi lệnh điều khiển.	40
Hình 2. 21 Lưu đồ giải thuật Control device	42
Hình 2. 22 Lưu đồ giải thuật hàm con model text classification.	44
Hình 2. 23 Quá trình xử lý phân loại văn bản.....	49
Hình 2. 24 Lưu đồ giải thuật hàm con xử lý chương trình chính của thiết bị Control device	50
Hình 2. 25 Lưu đồ giải thuật webapp localhost.	52
Hình 3. 1 Mạch in thực tế mặt trên thiết bị Node device	54
Hình 3. 2 Mạch in thực tế mặt dưới thiết bị Node device.....	54

Hình 3. 3 Hình ảnh thực tế mặt trên thiết bị Node device sau khi hàn linh kiện	55
Hình 3. 4 Hình ảnh thực tế thiết bị Control device	55
Hình 3. 5 Model Text Classification tích hợp với Control device	56
Hình 3. 6 Màn hình giám sát thiết bị trên website loaclhost.....	56
Hình 3. 7 Cấp nguồn cho 2 thiết bị Node device và Control device.....	57
Hình 3. 8 Giao diện website trạng thái thiết bị sau khi khởi động hệ thống.....	57
Hình 3. 9 Đang ra lệnh cho trợ lý bật đèn.	58
Hình 3. 10 Trạng thái đèn được mở trên website.....	59
Hình 3. 11 Relay1 (đèn) được đóng.	59
Hình 3. 12 Truyền nhận giữa Control device với Node device khi bật đèn	60
Hình 3. 13 Truyền nhận giữa Control device với Node device khi tắt đèn	60
Hình 3. 14 Trạng thái đèn được tắt trên website.....	61
Hình 3. 15 Relay1 (đèn) khi được mở.....	61
Hình 3. 16 Đoạn hội thoại hỏi thời tiết ở Hồ Chí Minh với trợ lý.	62
Hình 3. 17 Thí nghiệm đo tại nhà riêng	63
Hình 3. 18 Kết quả phân loại văn bản cho thí nghiệm “Bật đèn lên”.....	65
Hình 3. 19 Kết quả tách từ và lập từ điển bằng máy tính.	67
Hình 3. 20 Mô tả quá trình huấn luyện với 2 lớp đầu ra.....	69
Hình 3. 21 Mô tả quá trình phân loại với đầu vào là các từ của văn bản test	70

DANH MỤC BẢNG BIỂU

Bảng 1 Sơ đồ nối chân thiết bị “control device”.....	35
Bảng 2 Thứ tự các bước hoạt động của hệ thống.....	37
Bảng 3 Bảng vị trí thực nghiệm thí nghiệm đo khoảng cách truyền nhận.....	63
Bảng 4 Điều khiển giám sát trên tổng 20 lần.....	64
Bảng 5 Điều khiển giám sát dự đoán nhận trên tổng 20 lần.	64
Bảng 6 Tập văn bản huấn luyện cho thí nghiệm test model	65
Bảng 7 Tách từ với tệp dữ liệu huấn luyện	66
Bảng 8 Minh họa quá trình huấn luyện khi sử dụng Multinomial Naive Bayes cho nhãn ‘bat_den’ thí nghiệm test model.....	68
Bảng 9 Minh họa quá trình huấn luyện khi sử dụng Multinomial Naive Bayes cho nhãn ‘bat_bom’ thí nghiệm test model.	69

DANH MỤC TỪ VIẾT TẮT

IoT: Internet of Things

API: Application Programming Interface

NLP: Natural Language Processing

NBC: Naïve Bayes Classification

MQTT: Message Queuing Telemetry Transport

CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI

1.1 Đặt vấn đề

Hiện tại chúng ta đang ở trong thời đại khi con người tin dùng các thiết bị công nghệ hiện đại để giảm lượng công việc, thì thế nhu cầu sử dụng các sản phẩm công nghệ cũng được tăng cao. Các thiết bị luôn hoạt động để phục vụ nhu cầu đời sống của con người nên thường không thể tách rời nhau và chưa có một sự đồng bộ nào với nhau về mặt dữ liệu. Chúng ta đã khá quen thuộc với việc sử dụng công tắc để điều khiển các thiết bị khác nhau, mỗi thiết bị gắn với mỗi công tắc khác nhau nên việc ra khỏi nhà lại quên tắt đèn, điều hòa và các thiết bị khác là điều không tránh khỏi.

Việc làm cho nơi bạn đang ở hay nơi bạn làm việc trở nên thông minh và tối ưu hóa công việc hằng ngày giúp nâng cao đời sống và tăng năng suất làm việc là điều mà tất cả chúng ta đều đang quan tâm. Đó lý do để tôi nảy ra ý tưởng trong việc áp dụng AIOT vào xây dựng lên một hệ thống có khả năng nhận dạng được giọng nói của bạn và phân loại được ý định của bạn muốn làm gì để đưa ra hành động thực thi ngay lập tức dù bạn đang đứng ở đâu chỉ cần trong phạm vi nhận dạng giọng nói của thiết bị thì mọi thứ xung quanh bạn chỉ chờ khẩu lệnh của bạn nói ra mà thôi.

- Đối với hộ gia đình: thuận tiện cho người dân thời đại công nghệ 4.0 trong việc điều khiển một thiết bị nào đó chỉ qua thao tác ra lệnh bằng giọng nói.
- Đối với doanh nghiệp: Giảm được chi phí để mướn nhân viên giám sát, thuận tiện cho nhân viên làm việc, đôi lúc có thể xả stress.

1.2 Mục tiêu của đề tài

Phần Cứng: Có cứng board mạch cho hệ thống điều khiển các thiết bị thông qua giọng nói.

Phần mềm:

- + Giao diện người dùng trên raspberry pi
- + Mô hình nhận dạng giọng nói, phán đoán được ý định của người nói trên raspberry pi. Có giọng nói từ LOA xác nhận thực hiện tác vụ.

1.3 Nhiệm vụ đề tài

Phần cứng: Xây dựng các Node Device gồm: bộ vi điều khiển để điều khiển thiết bị điện 220VAC, giao tiếp với raspberry pi3 model b+ bằng giao thức truyền thông không dây LoRa.

Phần mềm:

- **Firmware:** Viết chương trình giao tiếp dữ liệu với raspberry pi 3b+ qua giao thức truyền thông không dây LoRa để các thiết bị điện 220VAC được điều khiển theo lệnh.
- **Software:** Xây dựng giao diện cho người dùng trên Webserver. Có thể xem được các trạng thái của các thiết bị điện được kết nối, xử lý nhận dạng giọng nói bằng thư viện Recognition Api của Google, xây dựng mô hình phân loại văn bản áp dụng thuật toán Naive Bayes Classifier (NBC).

1.4 Đối tượng và phạm vi nghiên cứu

Ngôn ngữ lập trình python

Các thư viện xử lý ngôn ngữ tự nhiên (NLP).

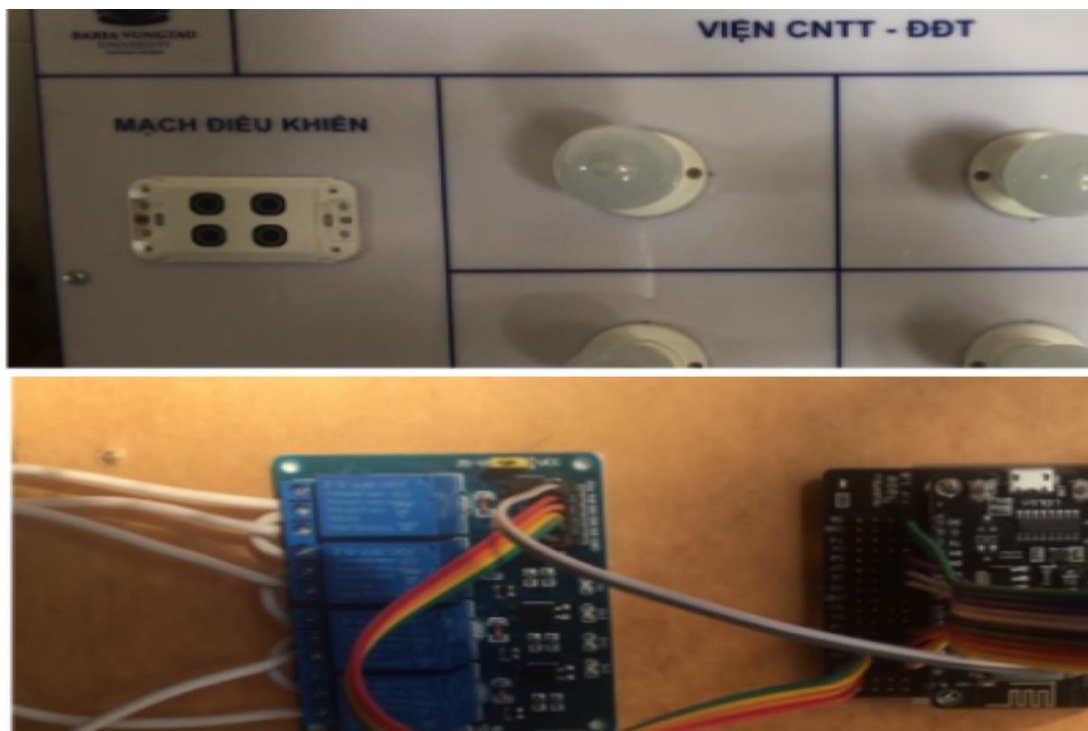
Naive bayes classifier (NBC).

Raspberry pi, Recognition, Flask-python, Socket, Mqtt, Lora, API, STM32F103c8t6, Arduino uno, STM32 CubeMX.

1.5 Điểm mới của đề tài

1.5.1 Các hệ thống đã được thực hiện

Trên thực tế, hệ thống các thiết bị được điều khiển bằng giọng nói và giám sát trạng thái thiết bị được nhiều nhóm sinh viên nghiên cứu và chọn làm khóa. Một vài hệ thống được tạo ra trong số đó:



Hình 1 Thiết kế điều khiển thiết bị bằng giọng nói với google assistant (google home) – Tác giả Mai Thanh Tuấn [3]

Hệ thống được thực hiện bởi tác giả Mai Thanh Tuấn (Hình 1) gồm giao diện bảng điều khiển và bộ board mạch điều khiển bao gồm board Node MCU ESP8266 có tích hợp chip wifi esp8266 và bộ board module 4 relay cho 4 thiết bị điện. Hệ thống này sử dụng 3 nền tảng có sẵn để thiết lập mô hình tiếp nhận và xử lý giọng nói điều khiển thiết bị:

1. Google Assistant: Đây là trợ lý ảo thông minh do google phát triển ứng dụng cho thiết bị smartphone và các ngôi nhà thông minh, được ra mắt lần đầu tại hội nghị nhà phát triển của google vào tháng 5 năm 2016.[5]
2. IFTTT: Đây là một dạng dịch vụ web trung gian giữa hai dịch vụ khác nhau, để thực hiện các tác vụ được thiết lập từ trước khi có điều kiện được xảy ra, IFTTT là viết tắt của If This Then That.

Trong đó, If This (nếu việc này xảy ra) sẽ dẫn đến Then That (thì làm việc kia), Là khi có sự kiện nào đó trên ứng dụng này, thì thông qua IFTTT ứng dụng kia cũng sẽ hoạt động và thay đổi tương tự.[4]

3. Blynk: Blynk được phát triển sau quá trình kêu gọi đầu tư trên Kickstarter. Blynk là nền tảng đám mây, tổ hợp gồm nhiều thành phần giúp tạo ra một dự án Internet of things (IoT) hoàn chỉnh, bao gồm: code, app, server,... Nói một cách dễ hiểu là khi chúng ta cần theo dõi, điều khiển các loại thiết bị khác nhau ngay từ xa, thông qua Internet, 4G.
- Quá trình điều khiển sẽ hoạt động như sau: Khi người dùng nói, một câu lệnh trên Google Assistant sẽ nhận lệnh sau đó thông qua IFTTT truyền xuống Blynk, từ đó Blynk truyền dữ liệu xuống board NodeMCU ESP8266. Khi này NodeMCU ESP8266 sẽ xử lý dữ liệu để kích mở/đóng relay tương ứng với các mức logic để bật/tắt các thiết bị điện theo lệnh cài đặt trên google assistant (Tức là phải nói đúng câu được thiết lập để bật hoặc tắt thiết bị nào đó thì mới điều khiển được, và người dùng phải ghi nhớ câu đó).

1.5.2 Điểm mới của mô hình

Mô hình sử dụng chip ARM STM32 làm bộ xử lý (vi điều khiển) chính cho thiết bị Node device thay vì sử dụng module hay board kit như bình thường.

Sử dụng máy tính nhúng Raspberry Pi3 làm bộ điều khiển trung tâm (Control device) quản lý tất cả các Node device để dễ dàng trong việc điều khiển các thiết bị.

Xây dựng một webserver riêng trên raspberry pi3 b+ để chạy ứng dụng website hiển thị trạng thái các thiết bị.

Xây dựng model xử lý ngôn ngữ tự nhiên riêng, tự chủ trong việc điều khiển thiết bị với tệp dữ liệu riêng.

Chỉ cần một bộ điều khiển trung tâm chính (Control device) được kết nối với 4G hoặc wifi, và tương tác trực tiếp với người dùng. Còn các khối điều khiển phụ (Node device) sẽ được người dùng điều khiển thông qua chuẩn giao tiếp LoRa do đó không cần phụ thuộc internet. Đây là tính năng mà người dùng có thể giám sát và điều khiển khi đặt bộ thiết bị Node device ở khu vực không có internet.

1.6 Ý nghĩa thực tiễn của đề tài

Sản phẩm sẽ được sử dụng trong các nhà máy, xí nghiệp, trong nhà, trong trường học để điều khiển và giám sát các thiết bị điện.

CHƯƠNG 2 THIẾT KẾ HỆ THỐNG

2.1 Giới thiệu phần cứng sử dụng

2.1.1 Vi điều khiển

STM32F103C8T6 MCU ARM 48-LQFP

STM32F103 (STM32: nền tảng STMicroelectronics, F1: thuộc họ F1, 03: phân khu của họ thiết bị, C8T6: xác định dung lượng bộ nhớ và tốc độ xung) với lõi là ARM Cortex-M3. STM32F103 là vi điều khiển 32 bit, tốc độ tối đa là 72Mhz (Hình 2.1). Mạch nạp - STlink cũng như công cụ lập trình STM32CubeMx.



Hình 2. 1 STM32F103C8T6 48pin

Ưu điểm :

- Hoạt động ổn định, tốc độ xử lý cao.
- Phù hợp với nhiều ứng dụng khác nhau.
- Khả năng tiết kiệm năng lượng cao. Có số lượng tài nguyên lớn.

2.1.2 Máy tính nhúng Raspberry Pi3



Hình 2. 2 Raspberry Pi 3 Model B+

Máy tính nhúng raspberry pi 3 model b+ (Hình 2.2) là board mạch mini computer được sử dụng nhiều, ngoài việc sử dụng như một máy tính bình thường chạy hệ điều hành Linux hoặc Windows 10 IoT, máy còn có khả năng xuất tín hiệu ra 40 chân GPIO với mục đích giao tiếp và điều khiển với các ngoại vi bên ngoài để thực hiện vô số các dự án khác nhau.

Ưu điểm:

- Là một nền tảng nguồn mở, có thể tùy chỉnh hệ thống tùy theo nhu cầu.
- Ứng dụng làm Box HD giống như Android Box.
- Có mạng LAN và Bluetooth không dây, có thể thiết lập wifi hotspot để kết nối internet. Đối với **Internet of Things**, tính năng này là phù hợp nhất.
- Có cổng chuyên dụng để kết nối màn hình LCD cảm ứng, đây là một tính năng hoàn toàn không cần đến màn hình. Cũng như có cổng camera chuyên dụng để kết nối với camera mà không gặp rắc rối.
- Nhỏ gọn, tiêu thụ năng lượng thấp, có khả năng hoạt động nhiều giờ...

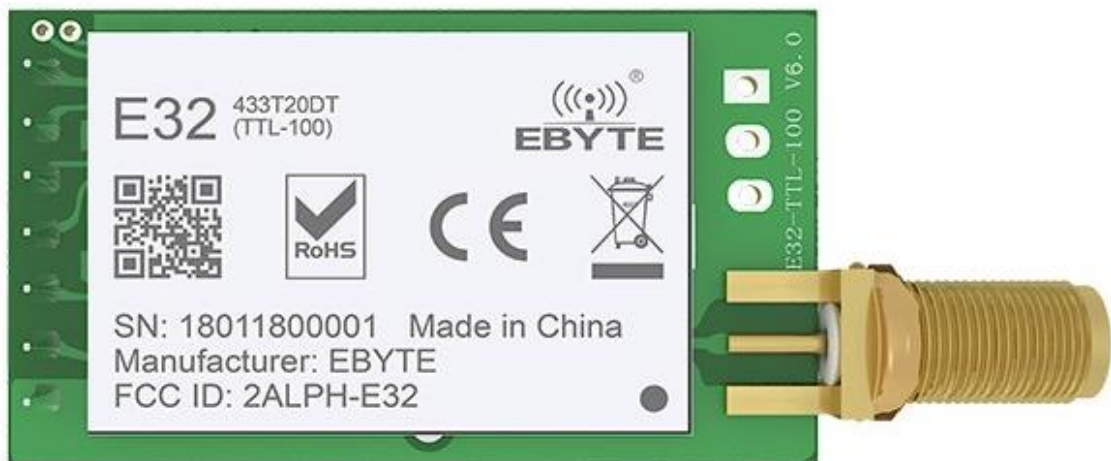
Nhược điểm:

- CPU cấu hình thấp.
- Để có thể sử dụng được phải có kiến thức cơ bản về Linux.

2.1.3 Module thu phát RF Lora SX1278 433MHZ Uart (E32-433T20DT).

Module Thu Phát RF Lora SX1278 433Mhz UART 3Km (E32-433T20DT) sử dụng chip SX1278 do nhà sản xuất SEMTECH sản xuất theo chuẩn giao tiếp LORA (LoRa spread spectrum), chuẩn giao tiếp này mang đến hai yếu tố quan trọng bậc nhất, một là tiết kiệm năng lượng và hai là khoảng cách phát siêu xa (Ultimate long range wireless solution). (Hình 2.3)

Ngoài những ưu điểm kể trên, nó còn có khả năng cấu hình để tạo thành mạng (lora mesh) nên hiện tại được phát triển và sử dụng rộng rãi trong các nghiên cứu về IoT.

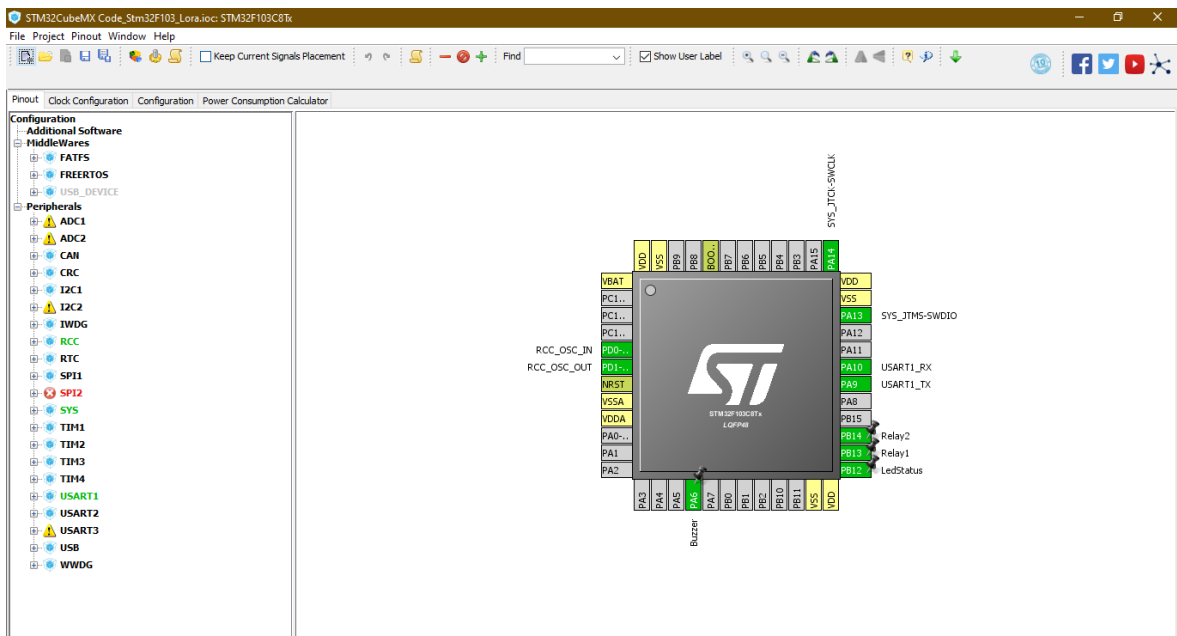


Hình 2. 3 Module thu phát RF LoRa SX1278

2.2 Giới thiệu phần mềm sử dụng

2.2.1 Phần lập trình cho vi điều khiển

Giới thiệu về STM32 CubeMX, phần mềm KeilC5 và thư viện HAL STM32 CubeMx.

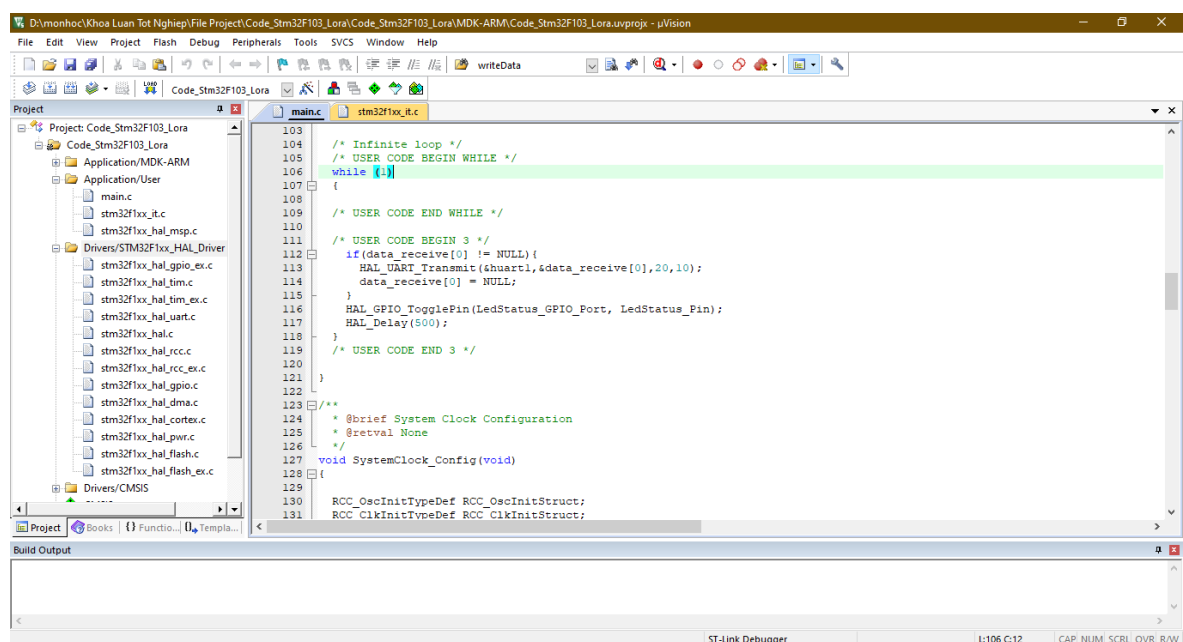


Hình 2. 4: Giao diện phần mềm STM32 CubeMX

STM32Cube là sự kết hợp của các công cụ phần mềm và các thư viện phần mềm nhúng. (Hình 2.4)

- Đầy đủ các công cụ phần mềm hỗ trợ chạy trên máy tính giúp giải quyết tất cả những nhu cầu trong một dự án hoàn chỉnh.
- Các phần mềm nhúng được thiết kế để chạy trên các dòng vi điều khiển STM32 khác nhau và các vi xử lý tương ứng với nhiều chức năng khác nhau từ các driver cho từng ngoại vi của vi điều khiển đến những tính năng định hướng ứng dụng nâng cao.

KEILC5: KeilC5 là một phần mềm hỗ trợ cho người dùng trong việc lập trình cho các dòng vi điều khiển khác nhau như Atmel, AVR,... Keil C giúp cho lập trình viên soạn thảo code và biên dịch chương trình C hay cả ASM sang ngôn ngữ máy để đưa vào vi điều khiển, từ đó giúp các lập trình viên tương tác giữa vi điều khiển và người lập trình. (Hình 2.5)

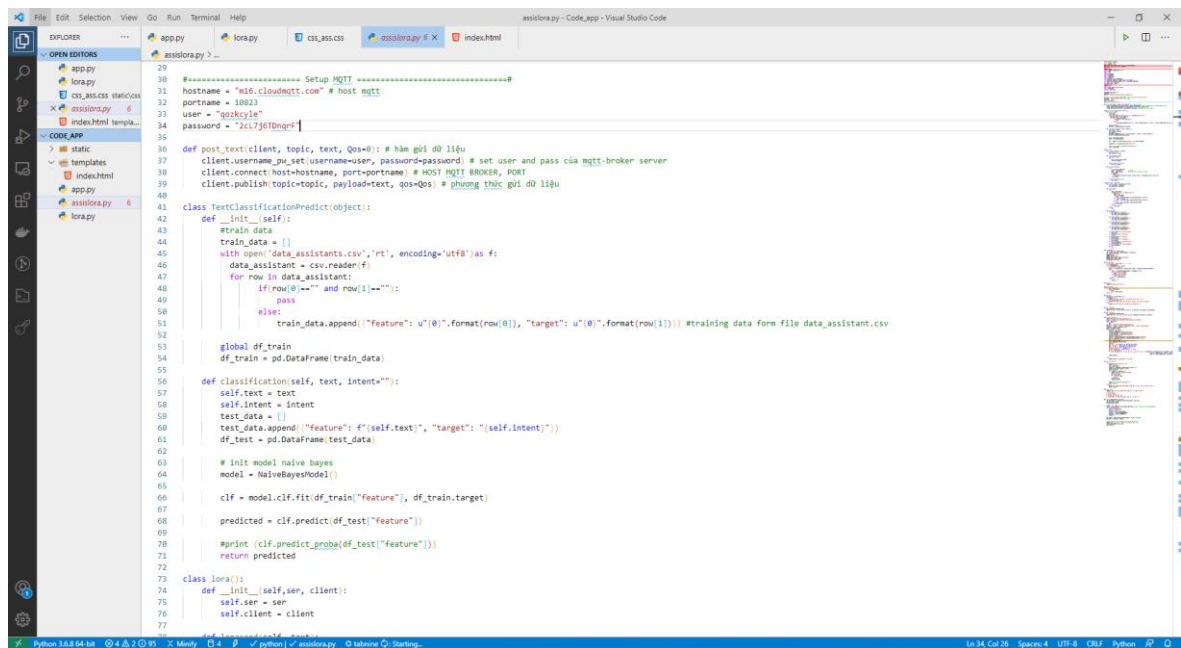


Hình 2. 5: Giao diện IDE KeilC5

Thư viện HAL

Thư viện đi kèm với SW4STM32 là STM32Cube HAL, gọi tắt là thư viện HAL, bộ thư viện này được chuẩn hóa, giúp đồng nhất giữa các dòng F0, F1, F2, F3, F4,...

2.2.2 Phân lập trình cho bộ nhận dạng giọng nói, model phân loại văn bản và điều khiển, mô hình website-status

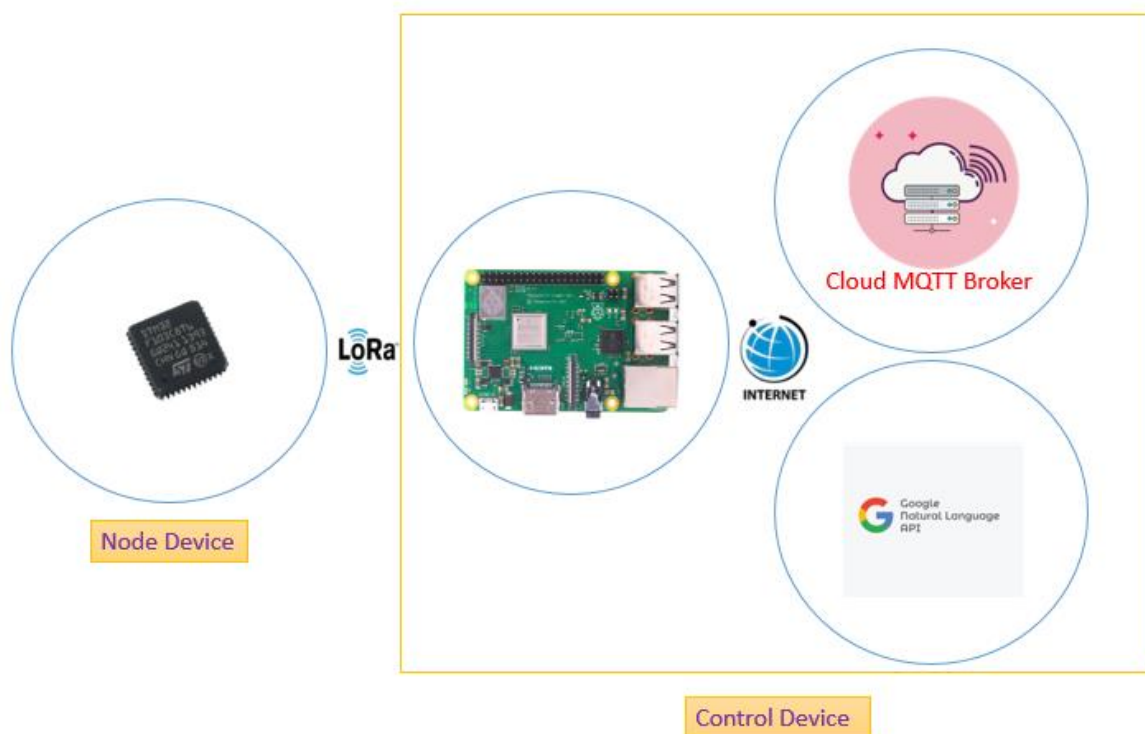


Hình 2. 6 Giao diện phần mềm Visual Studio Code

Visual Studio Code được phát triển bởi Microsoft (Hình 2.6). Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và cả macOS.

Visual Studio Code hỗ trợ chức năng debug, có syntax highlighting, đi kèm với Git và tự hoàn thành mã thông minh, snippets, cải tiến mã nguồn.

2.3 Tổng quan mô hình hệ thống



Hình 2. 7 Mô tả tổng quát hệ thống

Mô hình hệ thống (Hình 2.7) hoạt động gồm 2 phần chính:

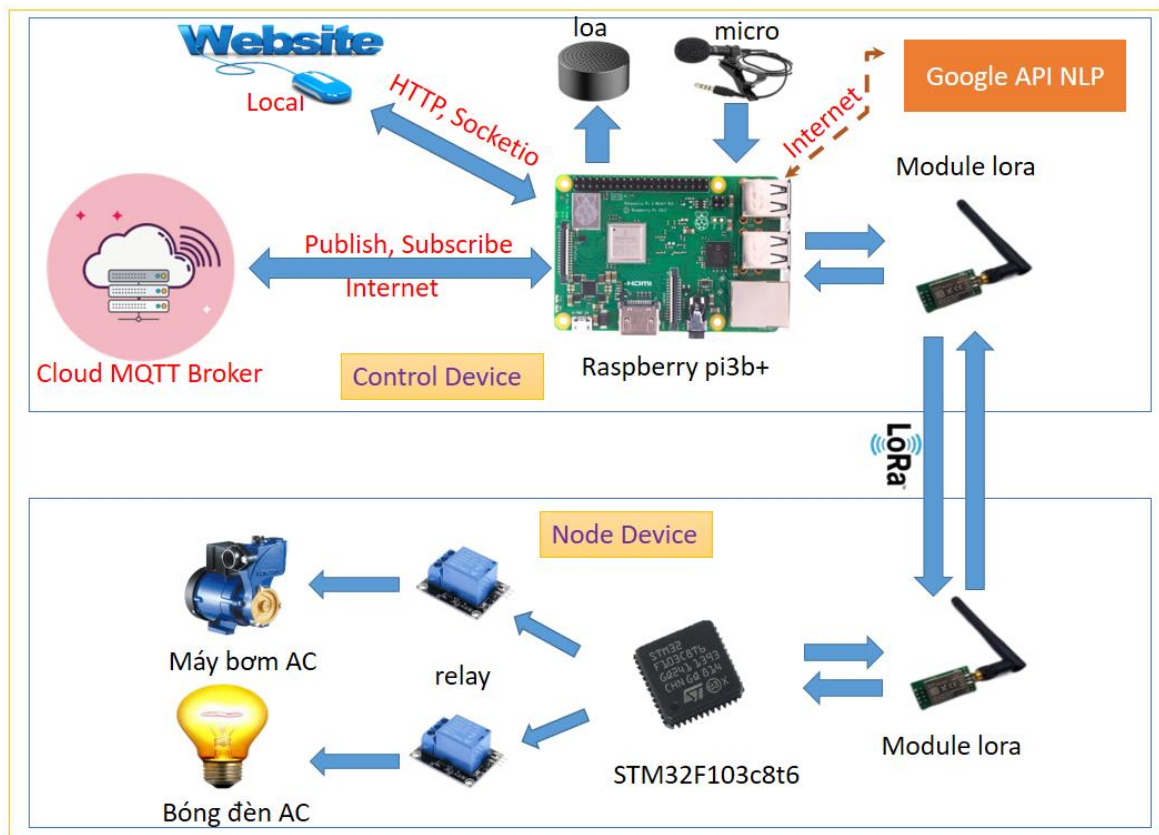
Node device: Sử dụng chip ARM STM32F103C8T6 làm vi điều khiển chính:

- Giao tiếp UART với module RF Lora SX1278-E32 để nhận lệnh điều khiển thiết bị từ Control device xuống.
- Điều khiển các thiết bị chấp hành bằng cách đóng mở relay.

Control device: Sử dụng máy tính nhúng Raspberry pi 3 b+ làm vi điều khiển chính:

- Truyền nhận dữ liệu với thiết bị Node device thông qua module RF Lora SX1278-E32.
- Thực hiện request đến Recognition API để phân tích giọng nói sau đó thực hiện chuyển đổi giọng nói thu được sang văn bản.
- Dự đoán ý định của người nói trong đoạn văn bản với mô hình máy học phân loại văn bản.
- Thực hiện quá trình chuyển đổi văn bản sang giọng nói.

- Truyền nhận dữ liệu với web-server localhost thông qua giao thức trung gian MQTT.



Hình 2. 8 Sơ đồ mô tả kết nối của hệ thống

Cơ cấu hoạt động của hệ thống (Hình 2.8):

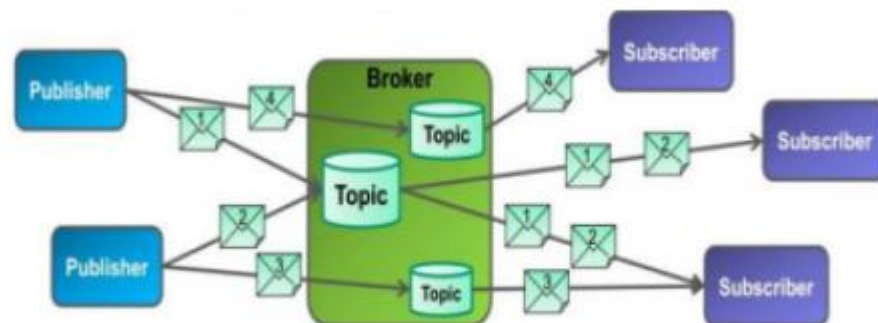
Node Device: Sử dụng chip ARM STM32F103C8T6 làm vi điều khiển chính:

- Điều khiển các thiết bị chấp hành bằng cách đóng mở relay.
- Xử lý các thông số điện năng và đọc trạng thái thiết bị gửi lên Raspberry pi
- Nhận lệnh điều khiển thiết bị từ Raspberry pi xuống thông qua module thu phát RF LoRa E32-TTL-100 SX1278.

Control Device: Bao gồm raspberry đóng vai trò như một gateway, Cloud MQTT Broker, và Google API NLP.

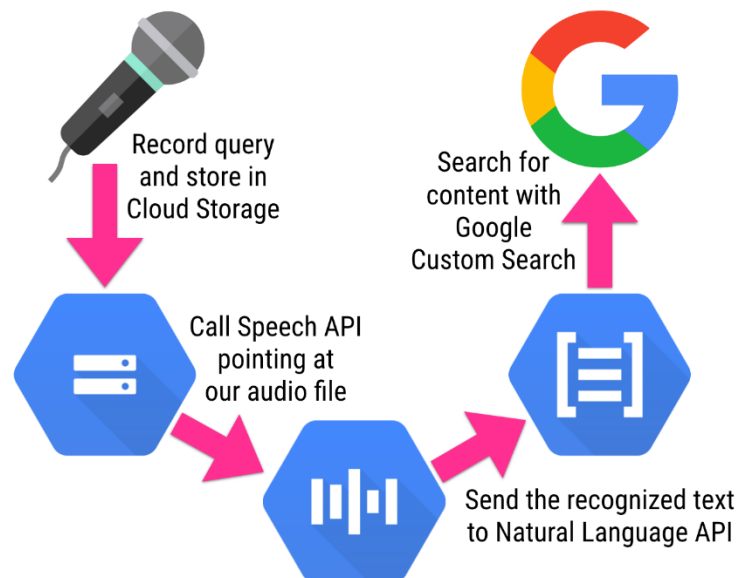
- **Raspberry pi (gateway) :**

- + Nhận giọng nói từ micro sau đó phân tích qua bộ tiền xử lý giọng nói là “Speech to Text” của thư viện Recognition API, khi đã có được đoạn văn bản thu được sau khi nhận voice sẽ đưa vào “model phân loại văn bản bằng máy học” phân tích và phán đoán đưa ra được nội dung chính của câu nói. Cuối cùng sẽ xử lý lệnh và truyền lệnh điều khiển xuống “Node Device” để thực thi.
- + Nhận các dữ liệu trả về từ “Node Device” thông qua module thu phát RF LoRa E32-TTL-100 SX1278 gửi lên, xử lý dữ liệu nhận được và encode dữ liệu sau đó gửi lên cho tầng application là Website thông qua giao thức HTTP, Socketio và dữ liệu chính được gửi qua giao thức MQTT.
- **Cloud MQTT Broker:** Là một máy chủ có thiết lập MQTT Broker, Broker là một lớp trung gian, giúp luân chuyển dữ liệu giữa các client với nhau thông qua cơ chế Publisher và Subscriber. Nhiệm vụ chính của Broker là nhận thông điệp (message) từ Publisher, xếp vào hàng đợi rồi chuyển đến một địa điểm cụ thể.(Hình 2.9)



Hình 2. 9 Cơ chế tổng quát của MQTT

- **Google API NLP:** Là một API AI của Google xử lý ngôn ngữ tự nhiên, chuyển giọng nói sang văn bản. (Hình 2.10)



Hình 2. 10 Cơ chế hoạt động khi thư viện Recognition gọi API xử lý ngôn ngữ tự nhiên của google.

- **Machine Learning NLP - Text Classification sử dụng scikit-learn:**

Mô hình phân loại văn bản (text classification) giúp giải quyết bài toán phân loại cảm xúc hay thái độ của người dùng qua bình luận (comment) trên các trang phim, mv ca nhạc, đánh giá về sản phẩm. Hay như trong ứng dụng chatbot được sử dụng để phát hiện mục đích của người dùng.

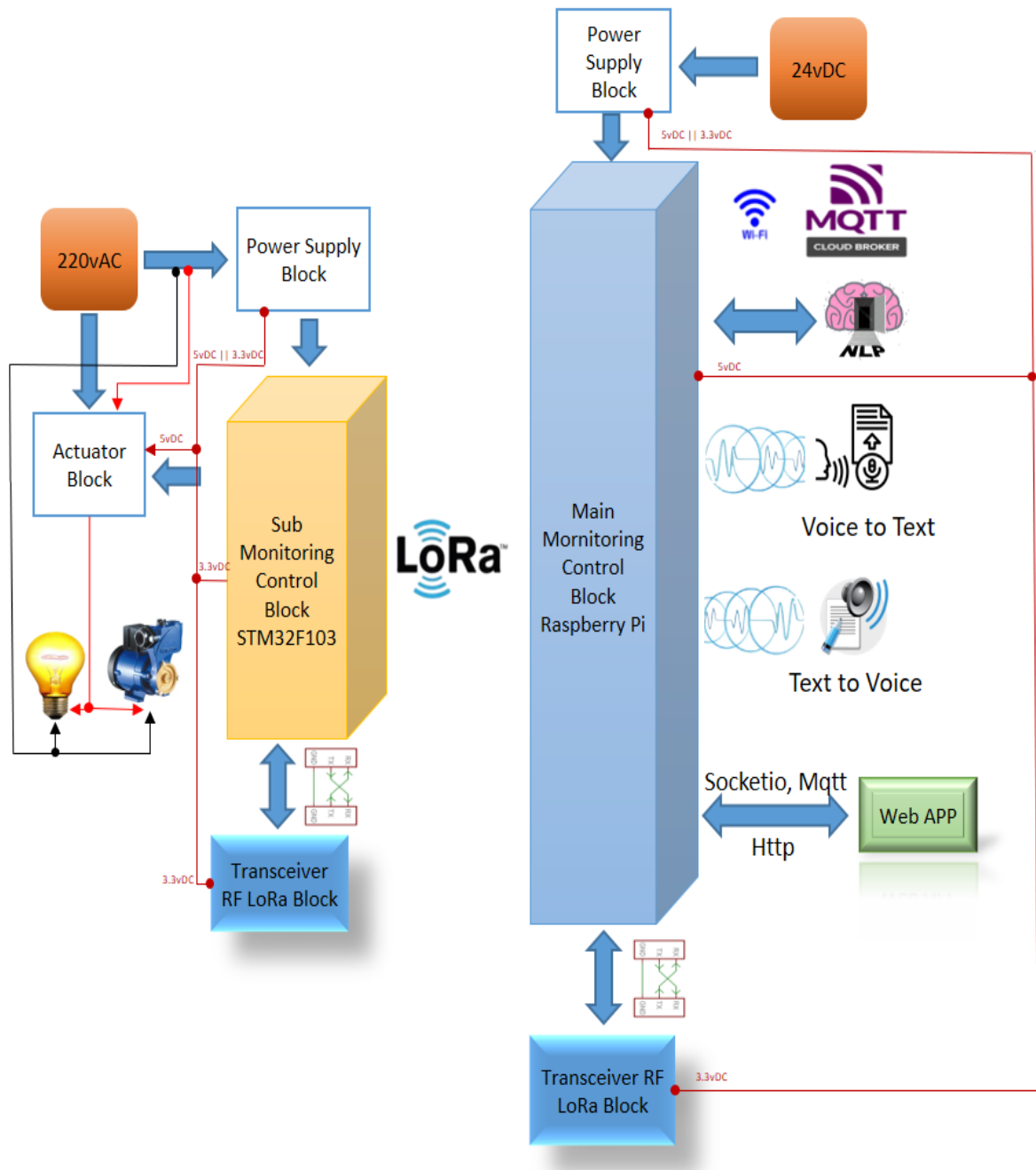
Ví dụ chúng ta muốn hỏi chatbot về điều khiển bật máy bơm trong nhà?

- Máy bơm có mở không nhỉ.
 - Bật máy bơm.
 - Mở máy bơm.
 - Bật bơm.
 - ...
- Cùng một nội dung là điều khiển bật máy bơm nhưng với mỗi người và ở trong các hoàn cảnh khác nhau, cảm xúc và tâm trạng khác nhau, mỗi người sẽ có những

câu hỏi, câu nói khác nhau. Trong trường hợp như vậy, lập trình viên không thể viết code xử lý theo phương pháp truyền thống là sử dụng if else, switch case được. Và lúc này là lúc ứng dụng bài toán **Text Classification** để giải quyết vấn đề trên. Công việc của lập trình viên là làm cách nào đó, để cũng với những câu hỏi như trên sẽ phân loại về 1 class có tên là “**bat_bom**” chẳng hạn.

2.4 Thiết kế hệ thống

2.4.1 Sơ đồ khối thiết bị “Node device” và “Control device”



Hình 2. 11 Sơ đồ khối toàn hệ thống

Chức năng từng khối chi tiết (Hình 2.11):

Khối xử lý và điều khiển trung tâm: Có nhiệm vụ kết nối với khối xử lý điều khiển phụ trong hệ thống để gửi dữ liệu điều khiển (voice) từ người dùng đến các thiết bị, đồng thời nhận dữ liệu trạng thái của các thiết bị được khối “xử lý điều khiển phụ” trả về.

Khối xử lý điều khiển phụ: Khối này có nhiệm vụ điều khiển thiết bị điện theo lệnh nhận được từ khối xử lý trung tâm. Đồng thời, cập nhật trạng thái của các thiết bị và trả về cho “khối xử lý và điều khiển trung tâm”.

Khối nguồn: Có nhiệm vụ chuyển đổi điện áp từ 220VAC về 5VDC và từ 5VDC về 3.3VDC để cấp nguồn cho toàn board mạch hoạt động.

Khối thu phát RF Lora: Khối này sử dụng module thu phát sóng RF Lora, có nhiệm vụ nhận dữ liệu từ vi điều khiển, từ đó chuyển đổi thành sóng RF để truyền đi và ngược lại.

Khối cơ cấu cấu chấp hành: Khối này có nhiệm vụ nhận lệnh điều khiển từ vi điều khiển sau đó sẽ đóng/ngắt relay để điều khiển thiết bị điện 220VAC.

Khối giao diện (Web app): Là ứng dụng web được viết bằng Javascript, flask-python để người dùng điều khiển và giám sát các thiết bị, các giá trị điện năng trong hệ thống.

Khối Voice to Text: có nhiệm vụ chuyển đổi giọng nói thành đoạn văn bản.

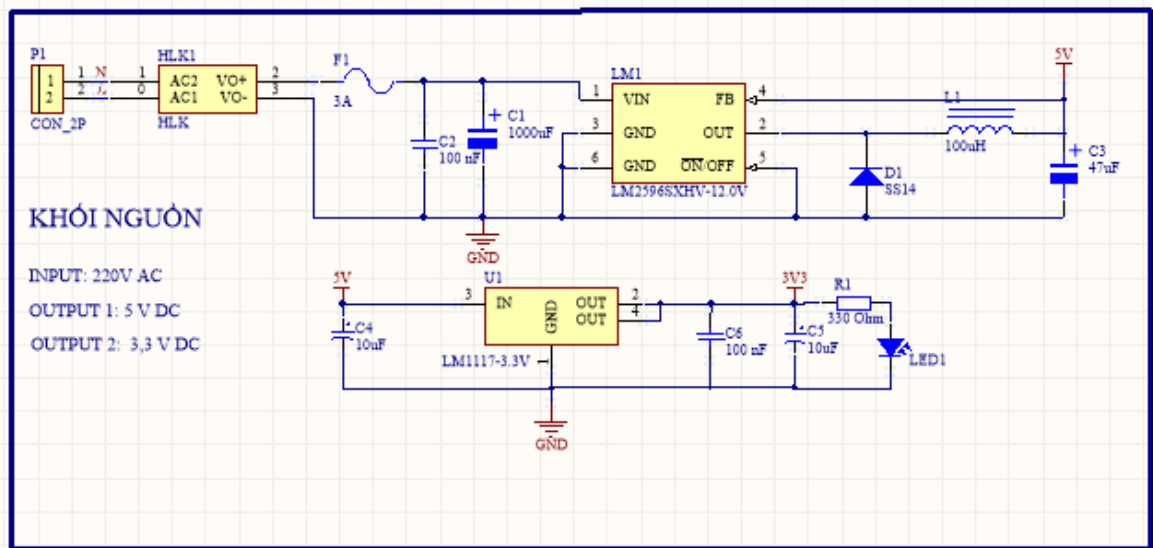
Khối model xử lý ngôn ngữ tự nhiên: Khối này sẽ nhận các đoạn văn bản sau khi qua khối “Speech to Text” sau đó đưa vào các layer mạng nơ ron nhân tạo của model để phân tích và phán đoán ra nội dung chính của đoạn văn bản.

Khối Text to Voice: có nhiệm vụ chuyển đoạn văn bản thành giọng nói.

Khối cloud mqtt broker: Khối này đóng vai trò là người vận chuyển dữ liệu trung gian giữa Raspberry với tầng Application (Website).

2.4.2 Sơ đồ nguyên lý thiết bị “Node device”

❖ Khối nguồn

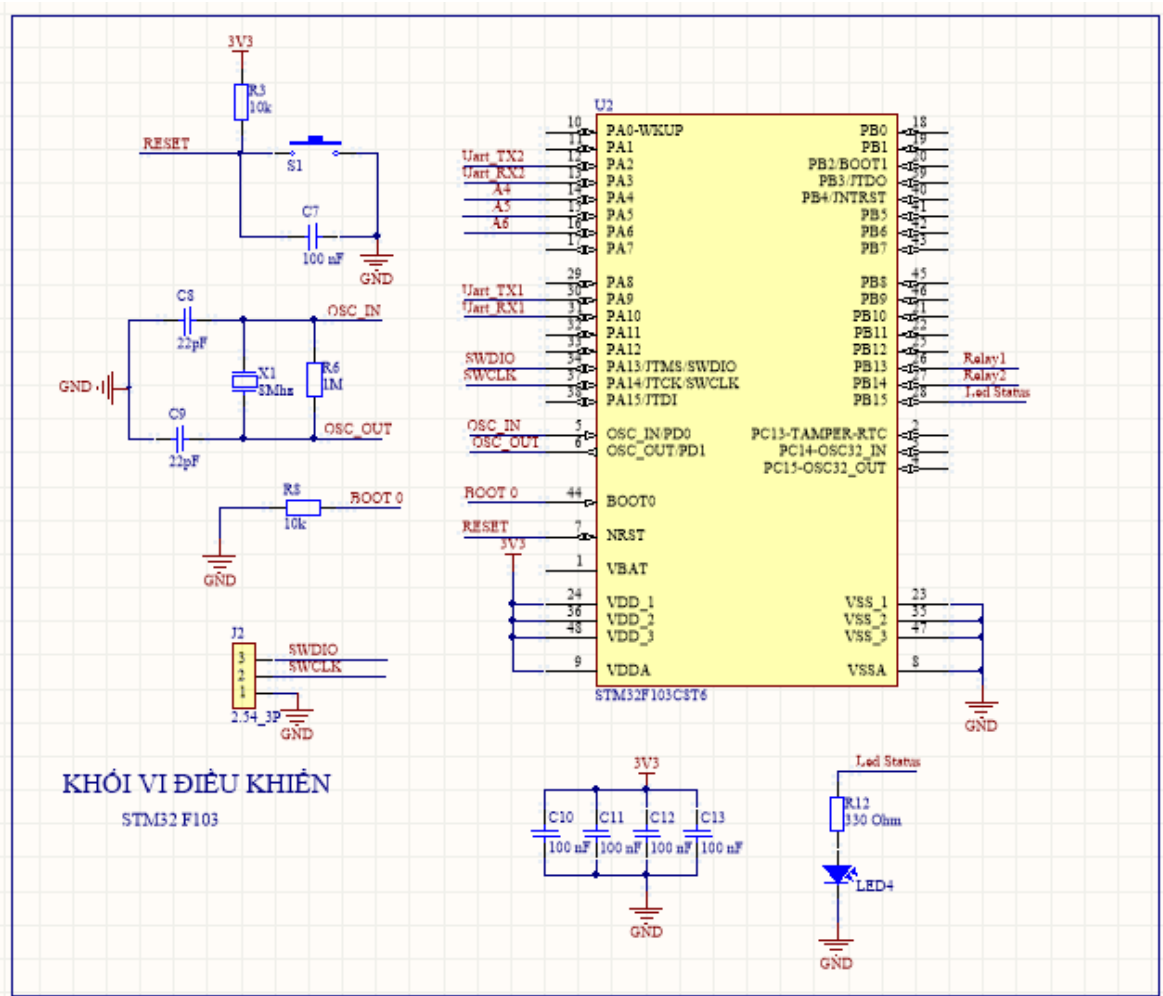


Hình 2. 12 Sơ đồ nguyên lý khối nguồn sử dụng

Khối nguồn (Hình 2.12), sử dụng ic nguồn LM2596sxhv, dòng LM2596 là mạch tích hợp có chi phí thấp mà cung cấp tất cả các chức năng hoạt động cho một mạch giảm áp (buck) hoàn chỉnh. Cho điện áp đầu ra ở OUTPUT1 là 5v DC.

Ngoài ra mạch có sử dụng thêm ic LM1173 để cho điện áp đầu ra OUTPUT2 là 3.3v DC.

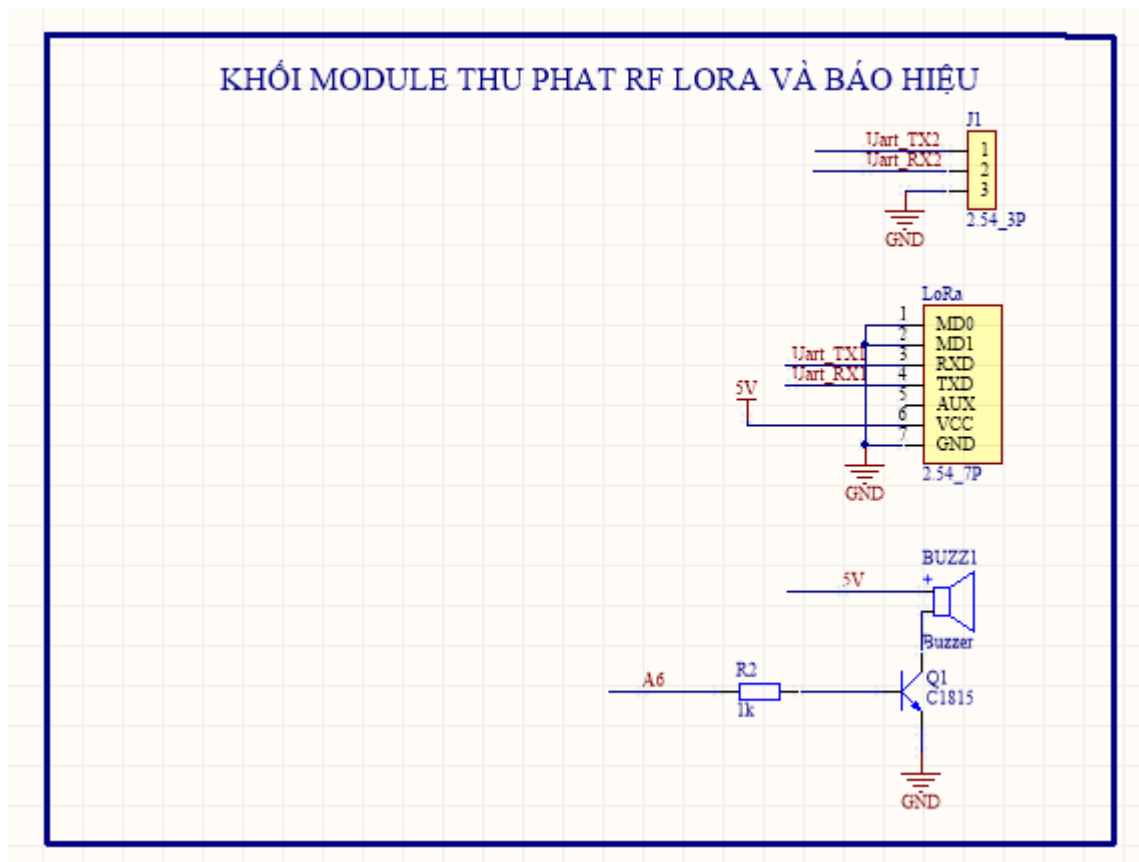
❖ Khối vi điều khiển



Hình 2. 13 Sơ đồ nguyên lý khối vi điều khiển

Khối vi điều khiển xử lý (Hình 2.13) truyền nhận dữ liệu với module lora thông qua chuẩn giao tiếp uart1 của vi điều khiển STM32F103C8T6 tại chân PA9_TX và PA10_RX. Tốc độ baudrate giữa 2 thiết bị là 9600, parity: none, data bits:8, stop bits: one. Xử lý tín hiệu đầu ra relay1 tại chân PB13 và relay2 tại chân PB14.

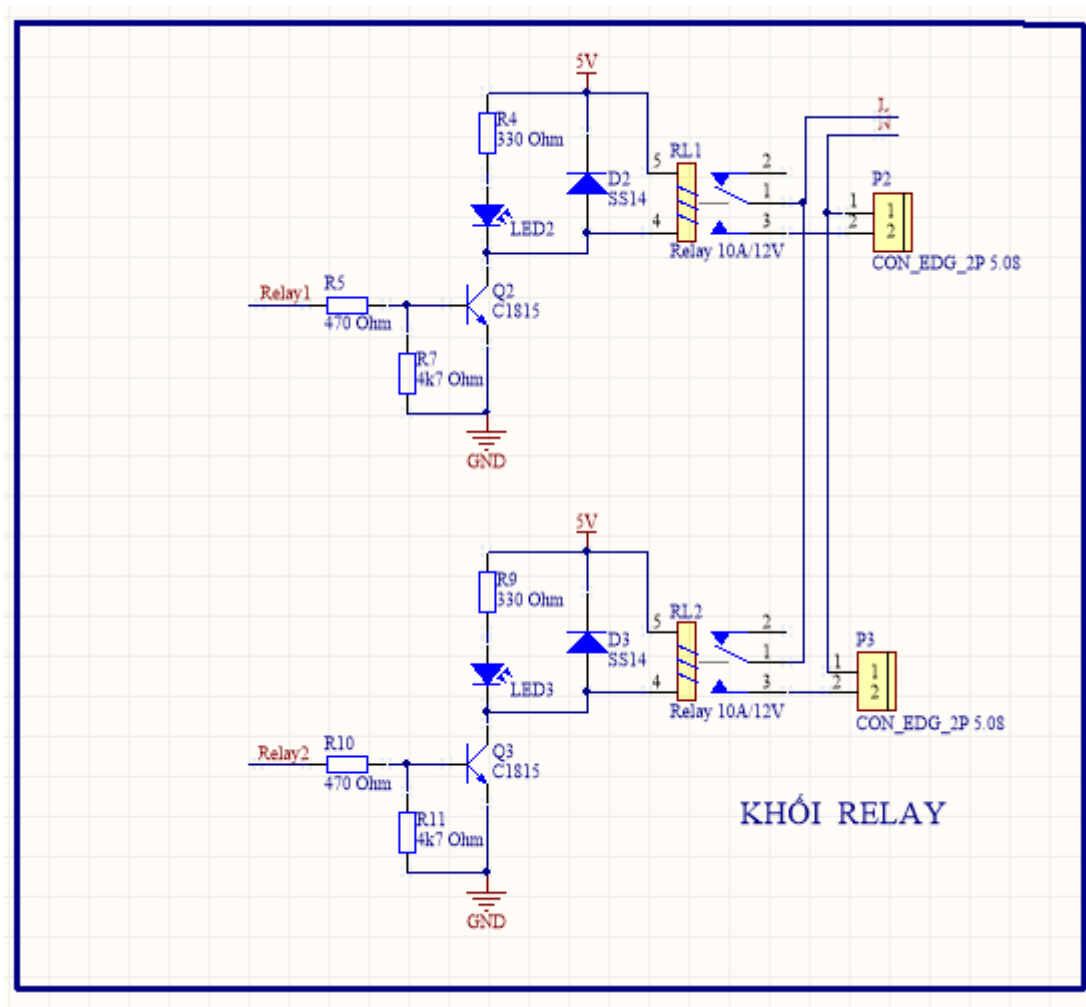
❖ Khối module thu phát RF Lora và báo hiệu



Hình 2. 14 Sơ đồ nguyên lý khối module thu phát rf lora và báo hiệu

Khối module thu phát rf lora (Hình 2.14) và báo hiệu xử lý truyền nhận dữ liệu giữa khối vi điều khiển và bộ **Control device** (Raspberry pi3 & rf lora)

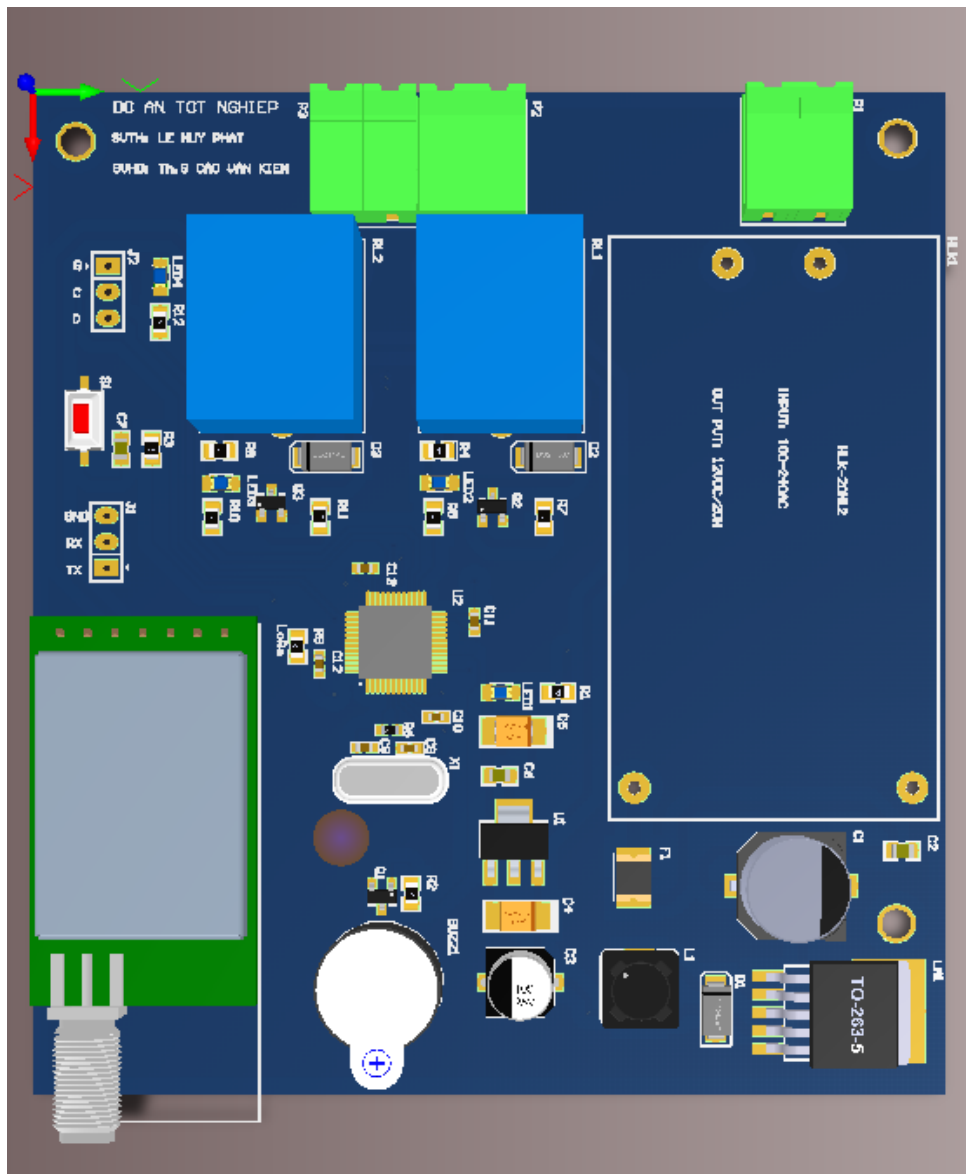
❖ Khối relay



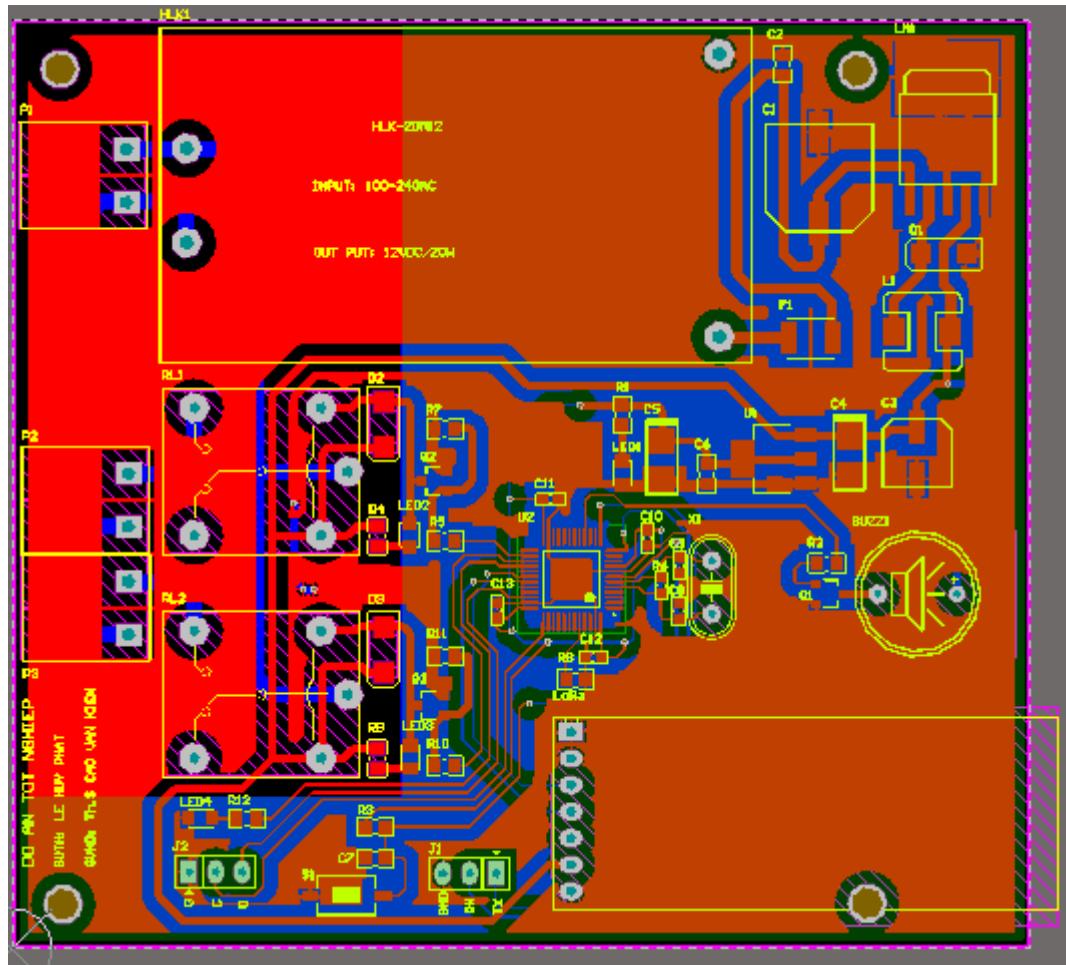
Hình 2. 15 Sơ đồ nguyên lý khối relay

Khối relay (Hình 2.15) thực thi các lệnh nhận được từ bộ vi điều khiển xử lý điều khiển các thiết bị.

2.4.3 Sơ đồ mạch in thiết bị “Node device”



Hình 2. 16 Bố trí và sắp xếp linh kiện 3D mặt trên của thiết bị Node device



Hình 2. 17 Sơ đồ mạch in thiết bị Node device

Sau khi sắp xếp 3D các linh kiện mặt trên và mặt dưới của thiết bị Node device (Hình 2.16). Tiến hành đi dây và kiểm tra khoảng cách các linh kiện cho thiết bị Node device (Hình 2.17).

2.4.4 Sơ đồ nối chân thiết bị “Control device”

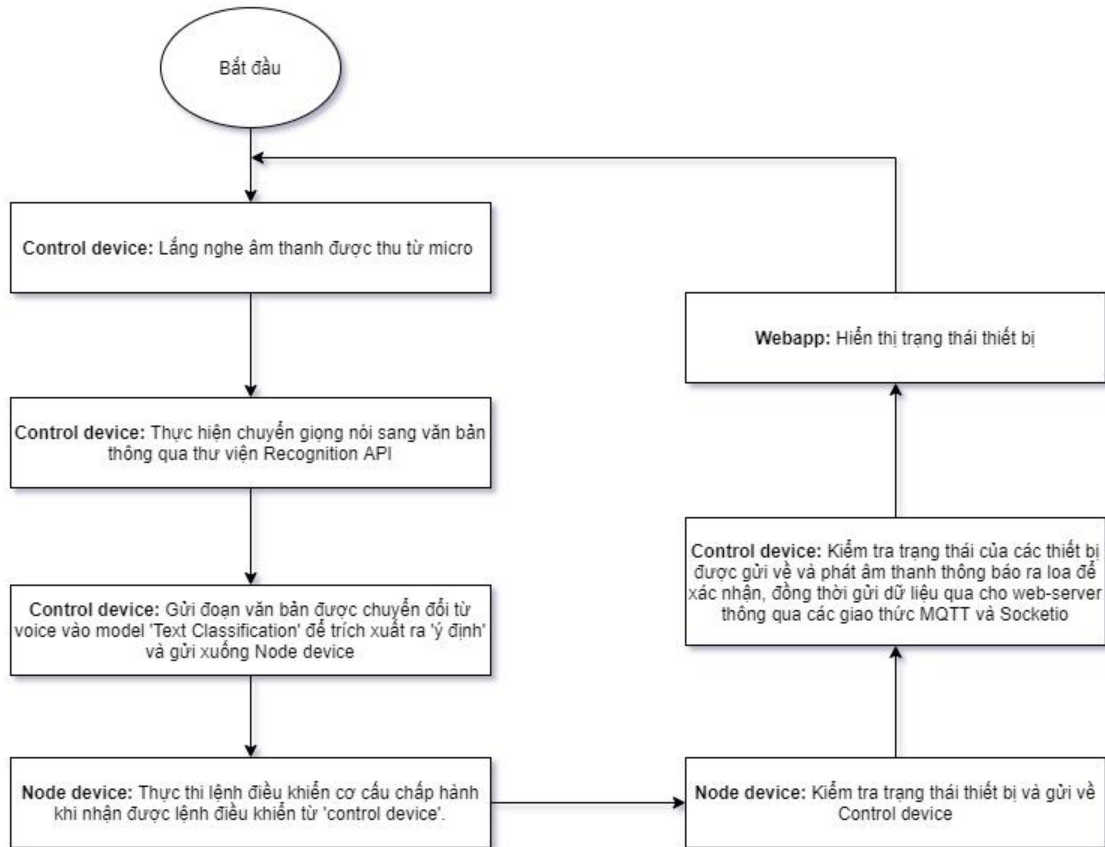
LORA E32-433T20DT	RASPBERRY PI 3 MODEL B+
M0	GND
M1	GND
RX	GPIO 14 (TX)
TX	GPIO 15 (RX)
AUX	-
VCC	3V3
GND	GND

Bảng 1 Sơ đồ nối chân thiết bị “control device”

Bộ thiết bị Control device bao gồm module Lora SX1278-E32 được đấu nối với máy tính nhúng Raspberry pi3 b+ theo Bảng 1.

2.5 Lưu đồ giải thuật

2.5.1 Lưu đồ giải thuật của hệ thống



Hình 2. 18 Lưu đồ giải thuật của hệ thống

Giải thích lưu đồ (Hình 2.18):

Sau khi micro thu được giọng nói, thiết bị Control device sẽ thực hiện chuyển giọng nói sang văn bản (Speech to text) bằng thư viện recognition API, ở đây chương trình sẽ thực hiện thao tác request đến API xử lý giọng nói của google với dữ liệu là đoạn voice và nhận về là đoạn văn bản (Tiếng Việt). Sau khi quá trình speech to text diễn ra xong thì Control device sẽ đưa đoạn văn bản đó vào model Text Classification (phân loại văn bản) để trích xuất ra ý định có trong đoạn văn bản. Quá trình Text Classification diễn ra xong thì thiết bị Control device tiếp tục gửi lệnh điều khiển (ý định trong đoạn văn mới được trích xuất) xuống thiết bị Node device.

Thiết bị Node device nhận được lệnh điều khiển sẽ thực thi cơ cấu chấp hành, điều khiển các thiết bị, đồng thời kiểm tra trạng thái của thiết bị có được thực thi hay chưa và gửi lại thông tin về trạng thái thiết bị cho Control device. Khi nhận được thông tin về trạng thái của các thiết bị được Node device gửi lên thì Control device sẽ phát thông báo âm thanh ra loa về trạng thái thiết bị đồng thời gửi thông tin trạng thái các thiết bị lên web-server để các browser-client request xem trạng thái thiết bị trên website localhost.

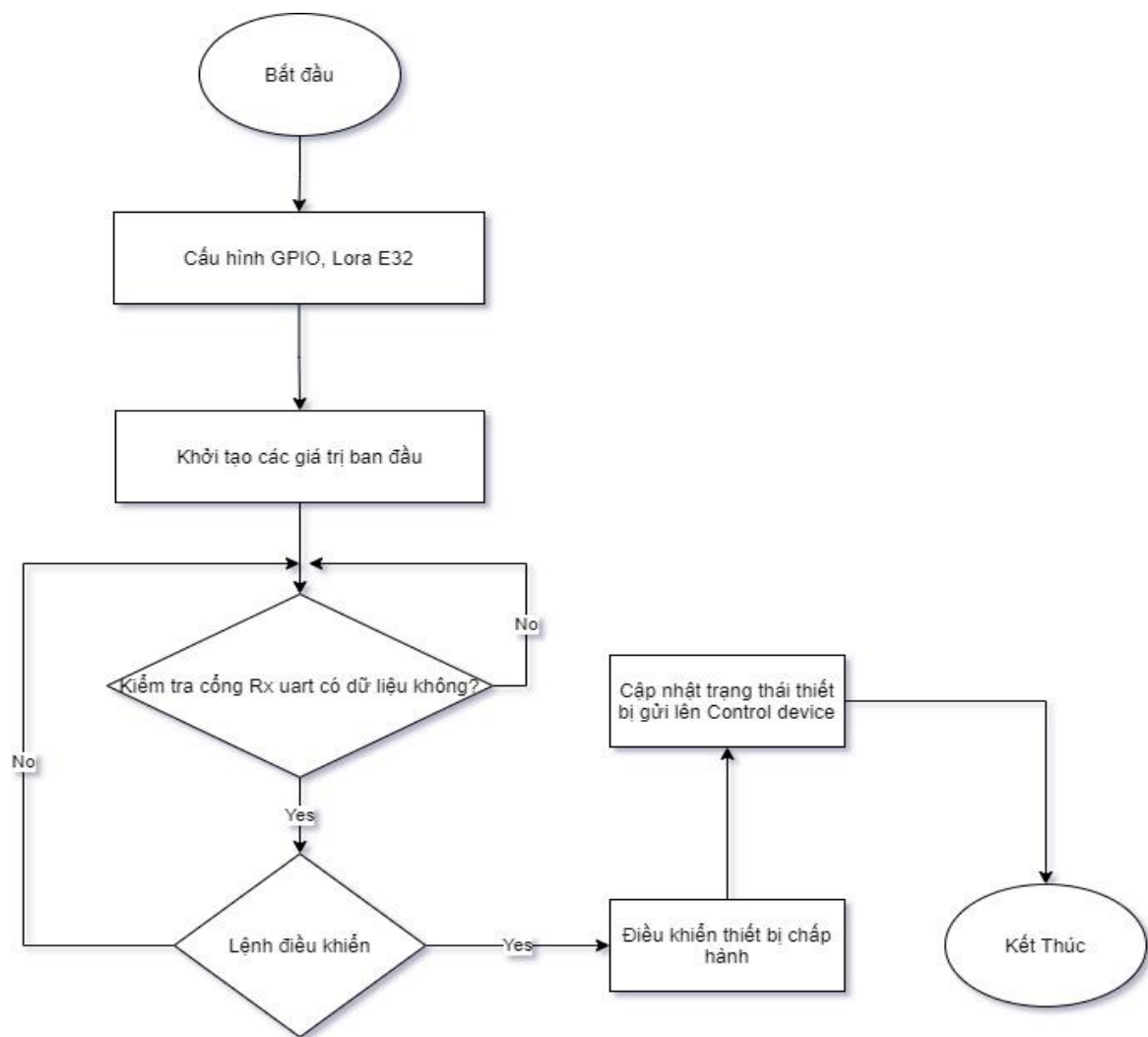
Nguyên tắc hoạt động của hệ thống được mô tả theo Bảng 2.

Bảng 2 Thứ tự các bước hoạt động của hệ thống

Thứ tự các bước xử lý	Control device	Node device
	Khởi động thiết bị Control device	Khởi động thiết bị Node device
1	Lắng nghe micro	Chờ lệnh điều khiển
2	Request API xử lý giọng nói của google với data là voice thu từ micro	Chờ lệnh điều khiển
3	Nhận lại văn bản được trả về từ API xử lý giọng nói của google và đưa vào model Text Classification	Chờ lệnh điều khiển
4	Trích xuất ra đoạn ý định trong văn bản và gửi xuống cho Node device	Nhận lệnh điều khiển
5	Chờ dữ liệu	Xử lý điều khiển các thiết bị

6	Chờ dữ liệu	Kiểm tra trạng thái hoạt động của các thiết bị điện
7	Đọc và kiểm tra dữ liệu	Chờ lệnh điều khiển tiếp theo
8	Phát thông báo trạng thái của các thiết bị ra loa	Chờ lệnh điều khiển tiếp theo
9	Gửi thông tin trạng thái thiết bị qua web-server	Chờ lệnh điều khiển tiếp theo

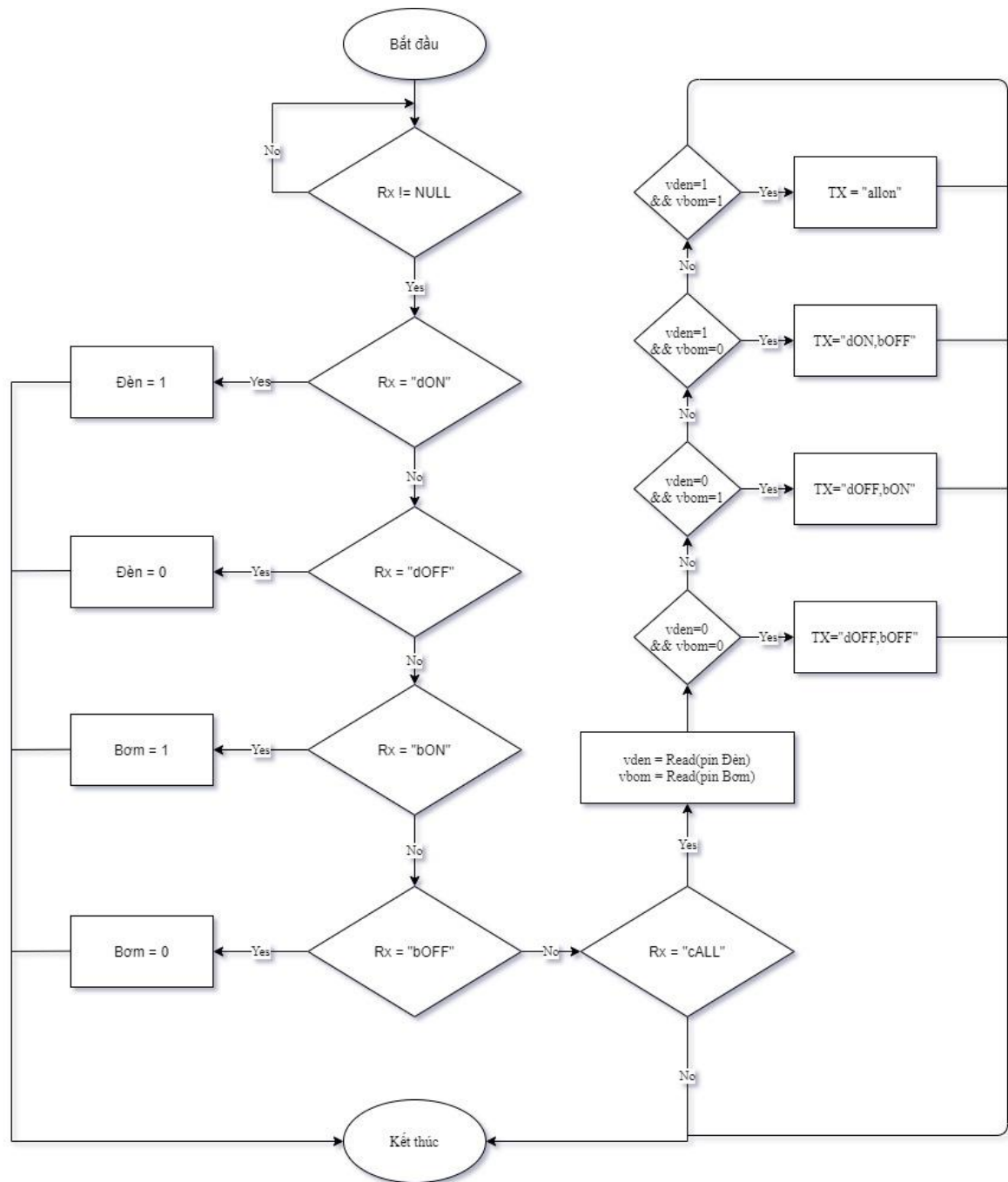
2.5.2 Lưu đồ giải thuật của thiết bị “Node device”



Hình 2. 19 Lưu đồ giải thuật Node device.

Giải thích lưu đồ (Hình 2.19):

Cấu hình các GPIO, module lora e32 . Kiểm tra cổng receive (RX-UART) có dữ liệu hay không, nếu không thì quay lại bước kiểm tra dữ liệu ở cổng receive, có dữ liệu thì thực hiện bước kiểm tra xem dữ liệu nhận về có phải dữ liệu nằm trong tệp dữ liệu lệnh được quy định hay không. Nếu dữ liệu nhận về không phải là lệnh điều khiển thì quay lại bước kiểm tra dữ liệu, nếu dữ liệu nhận về là lệnh điều khiển thì tiếp tục thực thi điều khiển các thiết bị điện, đồng thời gửi trạng thái các thiết bị điện lên cho Control device.



Hình 2. 20 Lưu đồ giải thuật hàm con xử lý truyền nhận dữ liệu và thực thi lệnh điều khiển.

Giải thích lưu đồ (Hình 2.20):

Kiểm tra cổng Rx-UART có dữ liệu hay không, nếu không có thì quay lại bước kiểm tra, có dữ liệu thì tiếp tục kiểm tra dữ liệu đó thuộc data receive frame:

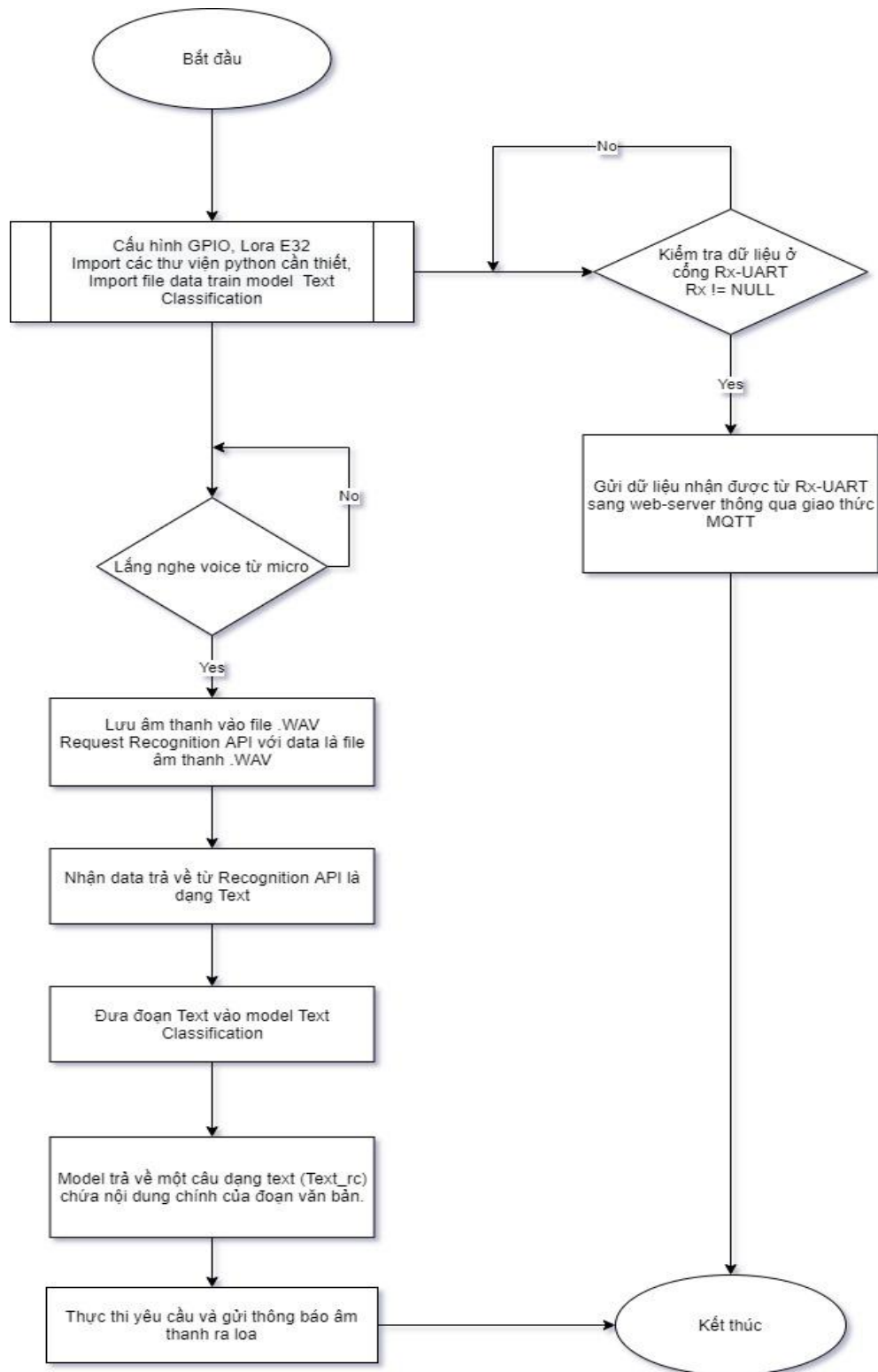
- dON : lệnh bật đèn
- dOFF: lệnh tắt đèn
- bON: lệnh bật máy bơm
- bOFF: lệnh tắt máy bơm
- cALL: lệnh kiểm tra trạng thái các thiết bị khi các browser-client website mới request lần đầu tiên.

Nếu dữ liệu nhận về từ cổng Rx-UART thuộc data receive frame thì tiếp tục thực thi lệnh điều khiển các thiết bị điện, đồng thời gửi thông báo xác nhận hoàn thành tác vụ cho Control device thông qua Tx-UART.

Dữ liệu nhận về là “cALL” thì chương trình con sẽ thực thi đọc trạng thái của các thiết bị và gửi dữ liệu thông báo theo data transmit frame:

- allon: đèn và máy bơm đều đang bật
- dON,bOFF: đèn bật, máy bơm tắt
- dOFF,bON: đèn tắt, máy bơm bật
- dOFF,bOFF: đèn tắt, máy bơm tắt

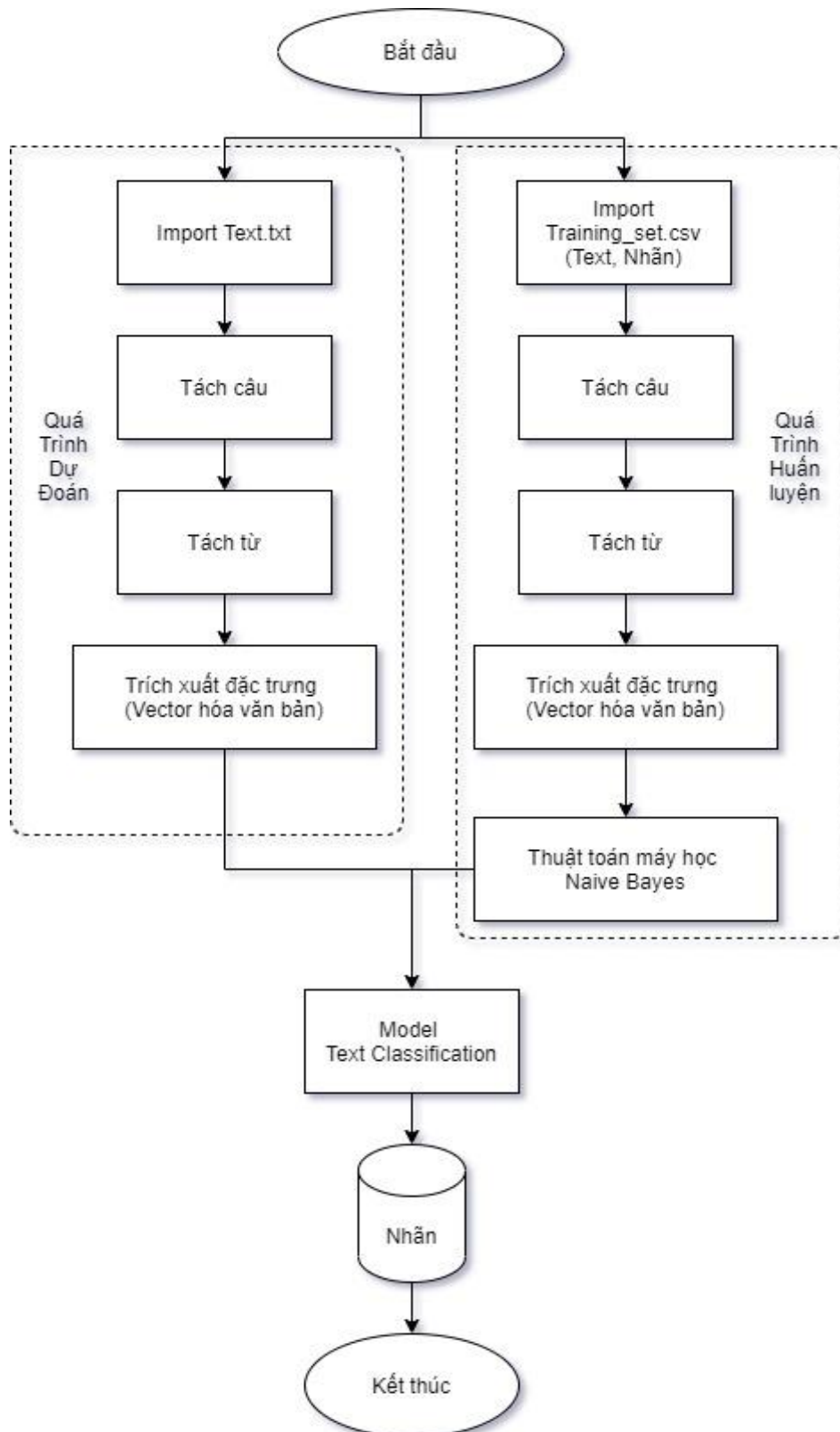
2.5.3 Lưu đồ giải thuật của thiết bị “Control device”



Hình 2. 21 Lưu đồ giải thuật Control device

Giải thích lưu đồ (Hình 2.21)

Sau khi khởi động thiết bị Control device, chương trình sẽ lắng nghe âm thanh thu từ micro được kết nối với raspberry pi 3 b+, khi có âm thanh được thu từ micro sẽ được lưu thành file âm thanh có phần mở rộng là '.WAV' và tiếp tục thực hiện request Recognition API với data là file âm thanh '.WAV', API sẽ trả về dữ liệu là đoạn văn bản được chuyển đổi từ file âm thanh theo ngôn ngữ là Tiếng Việt. Khi đã có đoạn văn bản chương trình sẽ đưa đoạn văn bản vào model Text Classification thực hiện quá trình dự đoán và đưa ra nhãn ứng với nội dung chính của đoạn văn bản đó. Khi đã có nhãn, chương trình sẽ thực thi các yêu cầu ứng với các nhãn đó và thông báo sẽ được phát ra loa với giọng người đọc là giọng của Google. Chạy song song với hàm xử lý chính đó thì Control device còn xử lý cùng lúc một hàm con khác là hàm kiểm tra dữ liệu nhận về từ cổng Rx-UART raspberry pi



Hình 2. 22 Lưu đồ giải thuật hàm con model text classification.

Giải thích lưu đồ (Hình 2.22)

Khi khởi động hệ thống sẽ xảy ra quá trình training dữ liệu, đầu tiên là bước tiền xử lý dữ liệu đầu vào, sau đó là quá trình trích xuất đặc trưng (Extracting features), cuối cùng là huấn luyện sử dụng thuật toán máy học Naïve Bayes. Đầu ra của quá trình huấn luyện là mô hình được xây dựng và các tham số tối ưu tương ứng cho mô hình.

Quá trình huấn luyện sẽ xảy ra với các bước như sau:

- Chương trình bắt đầu nạp file dataset (Training_set.csv) và bắt đầu phân tích từng câu, đầu tiên sẽ xảy ra quá trình tách câu, tiếp đến là tách từ. Do đặc thù của Tiếng Việt khá phức tạp, dấu cách (space) không được sử dụng như 1 ký hiệu phân tách từ, nó chỉ có ý nghĩa phân tách các âm tiết với nhau. Vì thế, để xử lý Tiếng Việt, công đoạn tách từ (word segmentation) là 1 trong những bài toán cơ bản và quan trọng bậc nhất.[2]

Ví dụ: Từ “đất nước” được tạo ra từ 2 âm tiết “đất” và “nước”, cả 2 âm tiết này đều có nghĩa riêng khi đứng độc lập, nhưng khi ghép lại sẽ mang một nghĩa khác.

Để giải quyết bài toán này cũng cần một lượng lớn dữ liệu. Dữ liệu đó đòi hỏi sự chính xác cao và tốn rất nhiều công sức. Tuy nhiên hiện nay cộng đồng xử lý ngôn ngữ tự nhiên Tiếng Việt khá mạnh nên có rất nhiều mã nguồn chất lượng. Với độ chính xác cao, dễ dàng sử dụng nên ở dự nán này tôi đã sử dụng thư viện “pyvi” có sẵn trên python.

- Khi đã giải quyết xong bài toán tách từ bằng thư viện “pyvi” của python, chương trình tiếp tục xử lý quá trình trích xuất văn bản (vector hóa văn bản), đoạn text đầu vào đã được tách từ. Ở đây tôi sử dụng thuật toán “Bag of words”.

Bag of words (Bow) là một thuật toán xử lý ngôn ngữ tự nhiên và mục đích của Bow là phân loại text hay văn bản. Ý tưởng của Bow là phân tích và phân nhóm dựa theo “Bag of words” (corpus). Với test data mới, tiến hành tìm ra số lần từng từ của test data xuất hiện trong “bag”. Tuy nhiên Bow vẫn tồn tại khuyết điểm, nên TF-IDF (Term Frequency – Inverse Document Frequency) là phương pháp

khắc phục. TF-IDF là một kỹ thuật sử dụng trong khai phá dữ liệu văn bản. Trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản. Nhưng TF-IDF còn có thể làm được nhiều hơn và tôi đã sử dụng tính năng lọc những từ stopwords (loại bỏ những từ không quan trọng) trong dự án này của tôi, khắc phục sự thiếu sót của Bow.[8]

Nguyên lý của Bag of words là: đầu tiên xây dựng từ điển và tạo vector số cho các văn bản theo phương pháp túi đựng từ. Tất cả các từ trong văn bản cần được chuyển thành dạng biểu diễn số. Cách đơn giản nhất là xây dựng một bộ từ điển, sau đó thay thế từ đó bằng thứ tự xuất hiện trong từ điển. Mỗi vector có độ dài chính bằng số từ trong từ điển.

Khi này mỗi văn bản được biểu diễn bởi một vectors có độ dài \mathbf{d} , chính là giá trị thành phần thứ i xuất hiện trong văn bản đó.

Khi đó, $P(x_i|c)$ tỉ lệ với tần suất từ thứ i (hay feature thứ i cho trường hợp tổng quát) xuất hiện trong các văn bản của class c . Giá trị này có thể tính bằng cách:[2]

$$\lambda_{ci} = p(x_i|c) = \frac{N_{ci}}{N_c} \quad (1)$$

Trong đó :

- N_{ci} là tổng số lần thứ i xuất hiện trong các văn bản của class c , nó được tính là tổng của tất cả các thành phần thứ i của feature vectors ứng với class c .
- N_c là tổng số từ (kể cả lặp) xuất hiện trong class c . Nói cách khác nó bằng tổng độ dài của toàn bộ các văn bản thuộc vào class c .

Tuy nhiên cách tính này có hạn chế là nếu có một từ mới chưa bao giờ xuất hiện trong class c thì biểu thức (1) sẽ bằng 0. Để giải quyết vấn đề này, tôi đã áp dụng một kỹ thuật có tên là smoothing:

$$\widehat{\lambda_{ci}} = \frac{N_{ci} + \alpha}{N_c + d\alpha} \quad (2)$$

Với α là một số dương, thường là 1, để tránh trường hợp tử số bằng 0. Mẫu số được cộng với $d\alpha$ để đảm bảo tổng xác suất $\sum_{i=1}^d \widehat{\lambda_{ci}} = 1$.

- Bước tiếp theo là xây dựng mô hình các thuật toán học máy: Ở đây tôi sử dụng thuật toán Naive Bayes. Naive Bayes Classifier (NBC) là một thuật toán phân loại dựa trên tính toán xác suất áp dụng định lý Bayes. Thuật toán này thuộc nhóm Supervised Learning (Học có giám sát).[2]

Theo định lý Bayes, ta có công thức tính xác suất như sau:[1]

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \quad (3)$$

Trong đó:

- $P(y|X)$ Xác suất của mục tiêu y với điều kiện có đặc trưng X
- $P(X|y)$ Xác suất của đặc trưng X khi đã biết mục tiêu y
- $P(y)$ Xác suất của mục tiêu y
- $P(X)$ Xác suất của đặc trưng X

Ở đây, X là vector các đặc trưng, có thể viết dưới dạng:

$$X = (x_1, x_2, x_3, \dots, x_n) \quad (4)$$

Khi đó, đẳng thức Bayes trở thành:

$$P(y|x_1, \dots, x_n) = \frac{P(x_1|y) * P(x_2|y) * \dots * P(y)}{P(x_1) * P(x_2) * \dots * P(x_n)} \quad (5)$$

Do đó ta có công thức cho bài toán Text Classification áp dụng định lý bayes như sau:

$$P(class | DATA) = \frac{P(DATA | class) * P(class)}{P(DATA)} \quad (6)$$

Với DATA là vector các đặc trưng.

$$DATA = (data_1, data_2, data_3, \dots, data_n)(7)$$

Trong mô hình Naive Bayes, có hai giả thiết được đặt ra:

1. Các đặc trưng đưa vào mô hình là độc lập với nhau. Tức là sự thay đổi giá trị của một đặc trưng không ảnh hưởng đến các đặc trưng còn lại.
2. Các đặc trưng đưa vào mô hình có ảnh hưởng ngang nhau đối với đầu ra mục tiêu.

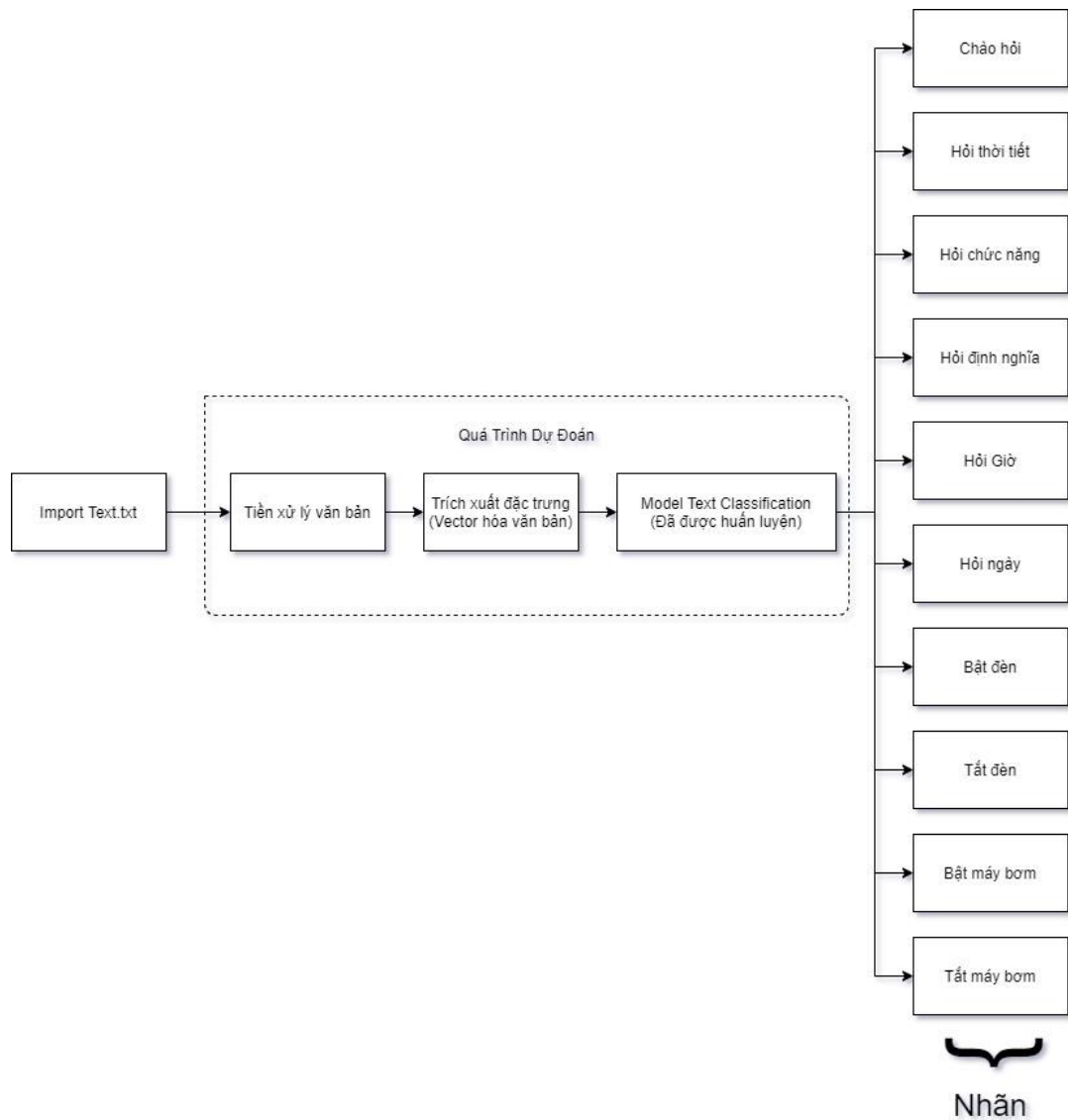
Khi đó, kết quả mục tiêu class để $P(class|DATA)$ đạt cực đại trở thành:[2]

$$class = \underset{class}{\operatorname{argmax}} * P(class) \prod_{i=1}^n P(data_i | class) \quad (8)$$

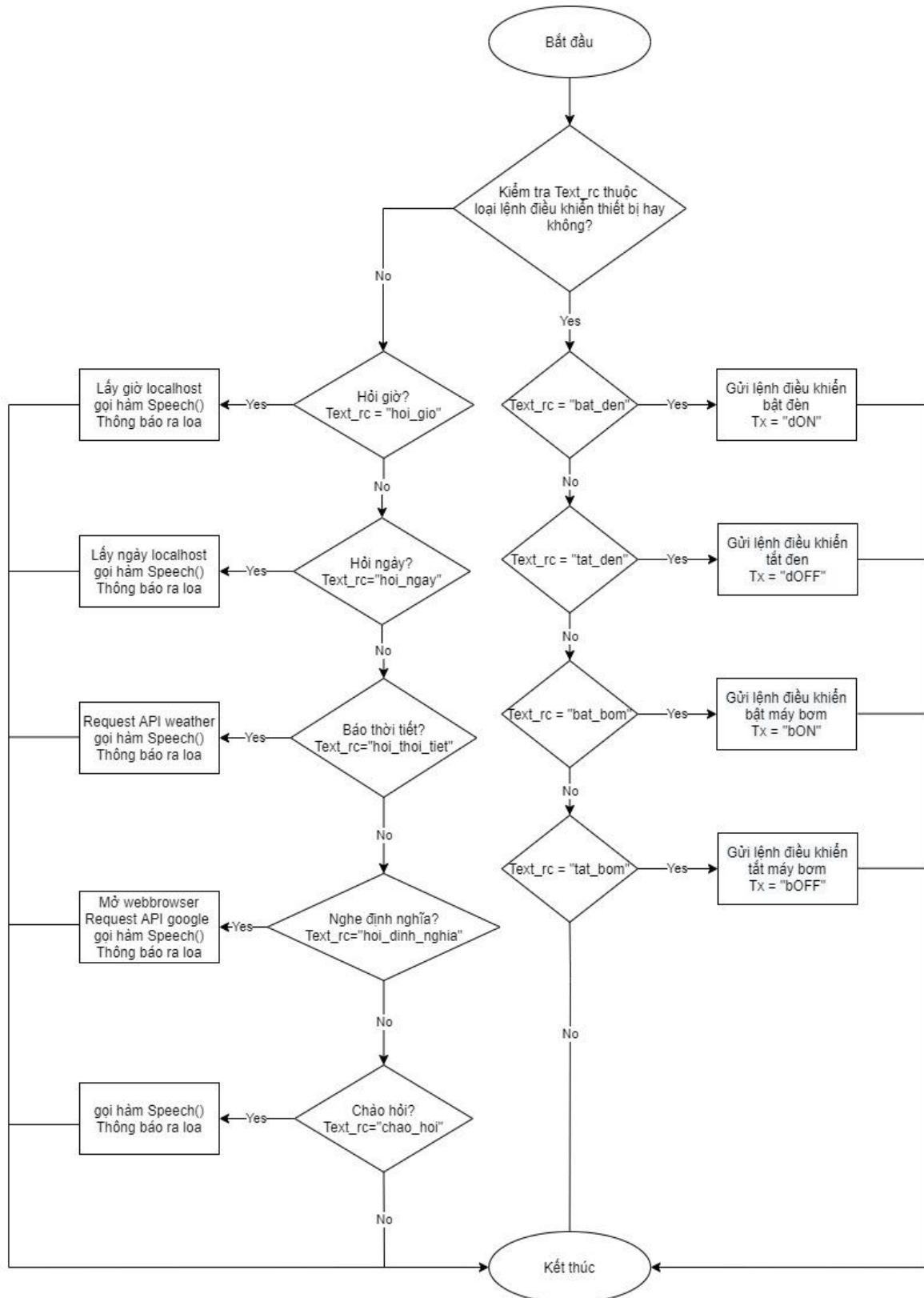
Quá trình học thuật toán diễn ra xong chúng ta sẽ có model Text Classification ứng dụng xử lý phân loại văn bản.

Quá trình phân loại văn bản:

- Các dữ liệu đầu vào được thực hiện tiền xử lý và trích xuất đặc trưng dữ liệu (tương tự như bước 1 và 2 trong quá trình huấn luyện). Sau đó, các vector thuộc tính dữ liệu được đưa vào đầu vào của mô hình đã huấn luyện, đầu ra sẽ là các nhãn (các nhãn là các chủ đề của mô hình huấn luyện, ở dự án này model của tôi có 10 nhãn ngõ ra) tương ứng với các file văn bản đầu vào. (Hình 2.23)



Hình 2. 23 Quá trình xử lý phân loại văn bản



Hình 2. 24 Lưu đồ giải thuật hàm con xử lý chương trình chính của thiết bị Control device

Giải thích lưu đồ (Hình 2.24)

Sau khi nhận được dữ liệu từ model Text Classification trả về (Text_rc), chương trình sẽ tiến hành kiểm tra dữ liệu đó có phải là loại lệnh điều khiển thiết bị hay không.

Nếu là lệnh điều khiển thiết bị thì kiểm tra dữ liệu đó có nằm trong data frame là loại dữ liệu nào và gửi lệnh điều khiển thông qua module lora xuống Node device để cơ cấu chấp hành thực hiện các tác vụ. Với data frame:

- Text_rc = “bat_den” : Gửi lệnh bật đèn xuống Node device, Tx=”dON”
- Text_rc = “tat_den” : Gửi lệnh tắt đèn xuống Node device, Tx=”dOFF”
- Text_rc = “bat_bom” : Gửi lệnh bật máy bơm xuống Node device, Tx=”bON”
- Text_rc = “tat_bom” : Gửi lệnh tắt máy bơm xuống Node device, Tx=”bOFF”

Nếu Text_rc không phải là các lệnh điều khiển thiết bị thì chương trình tiếp tục kiểm tra tiếp Text_rc có phải là nhãn hỏi về giờ hay không (hoi_gio), nếu là nhãn hỏi giờ thì thực hiện lấy giờ hiện tại trên máy localhost và thực hiện thông báo bằng âm thanh ra loa.

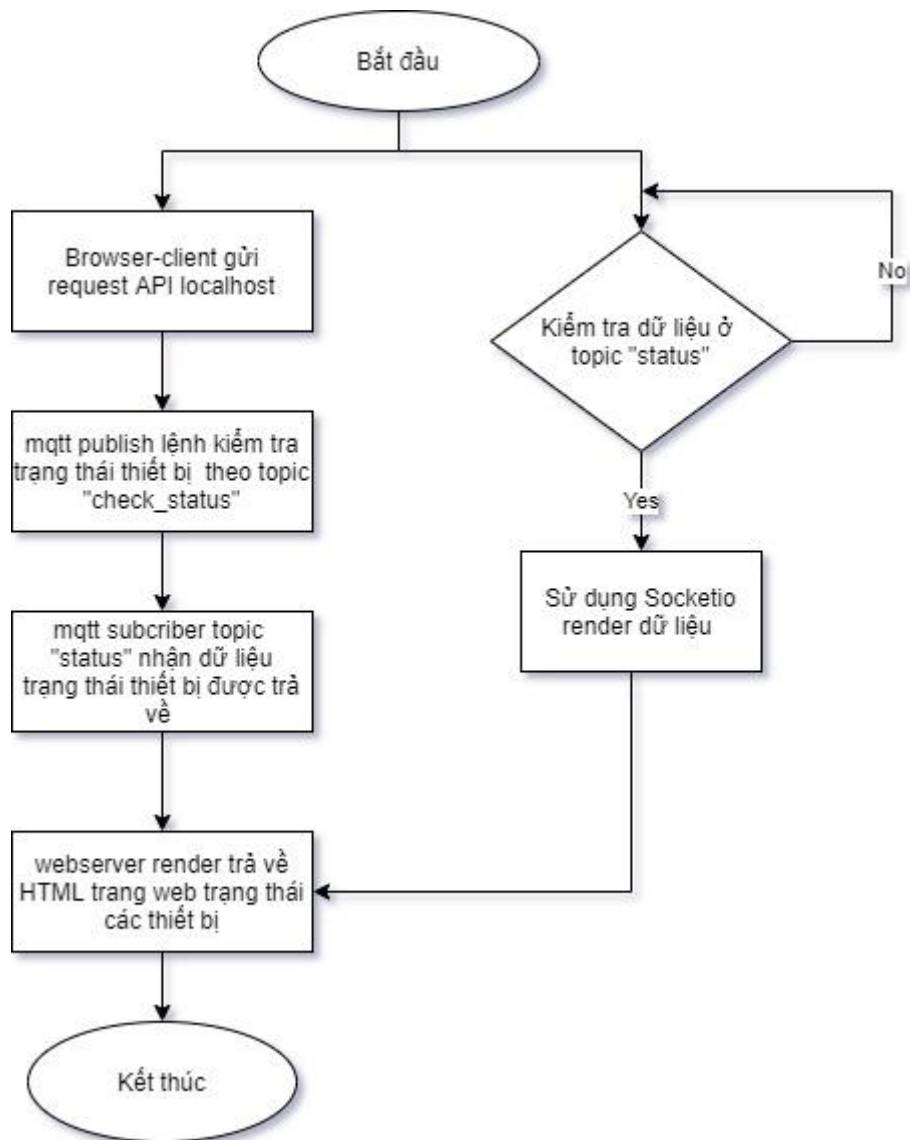
Không phải nhãn “hoi_gio” thì kiểm tra tiếp Text_rc có phải nhãn hỏi về ngày (hoi_ngay) hay không, nếu phải thì lấy ngày hiện tại trên máy localhost và thực hiện thông báo bằng âm thanh ra loa.

Không phải nhãn “hoi_ngay” thì kiểm tra tiếp có phải nhãn hỏi về thời tiết (hoi_thoi_tiet) hay không, phải thì thực hiện gửi một request tới API weather để lấy dữ liệu về thời tiết tại vị trí hiện tại của thiết bị và thực hiện thông báo bằng âm thanh ra loa.

Không phải nhãn “hoi_thoi_tiet” sẽ kiểm tra tiếp có phải nhãn hỏi về định nghĩa

(hoi_dinh_nghia) không, phải thì thực hiện mở webbrowser sau đó gửi request tới API google với thông tin được yêu cầu và trả về kết quả, cuối cùng thực hiện thông báo âm thanh kết quả ra loa.

Không phải nhãn “hoi_dinh_nghia” thì kiểm tra tiếp có phải nhãn chào hỏi (chao_hoi) hay không, phải thực hiện thông báo âm thanh ra loa. Không phải nhãn “chao_hoi” sẽ kết thúc tiến trình kiểm tra.



Hình 2. 25 Lưu đồ giải thuật webapp localhost.

Giải thích lưu đồ (Hình 2.25)

Khi có request tới API localhost của raspberry, API sẽ thực hiện gửi lệnh kiểm tra trạng thái các thiết bị hiện tại qua topic “check_status” với giao thức MQTT, sau đó đợi dữ liệu trả về qua topic “status”, khi đã có dữ liệu trả về thì tiếp tục kiểm tra dữ liệu có thuộc trong data frame hay không:

- “allon” : đèn và máy bơm đang bật
- “dOFF,bOFF” : đèn và máy bơm đang tắt
- “dON,bOFF” : đèn đang bật, máy bơm đang tắt
- “dOFF,bON” : đèn đang tắt, máy bơm đang bật

Tiếp theo API dựa trên dữ liệu đã trả về và thực hiện render HTML trả về cho các browser-client đang request.

Chạy song song với tiến trình trên là hàm kiểm tra dữ liệu ở topic “status” theo data frame:

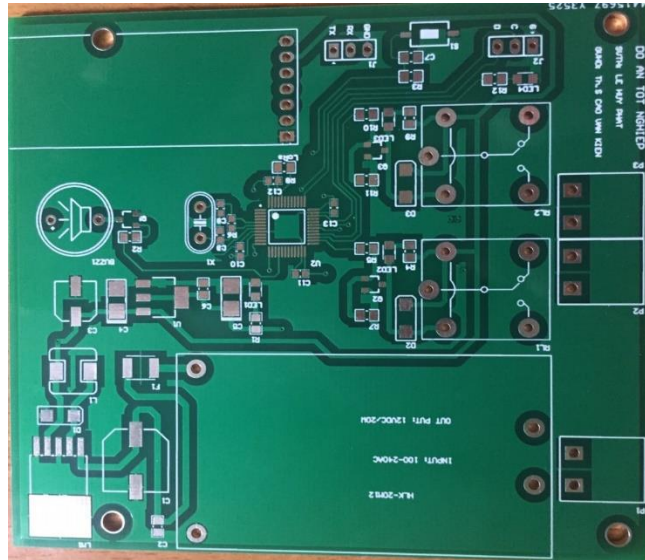
- “dON” : đèn đang bật
- “dOFF”: đèn đang tắt
- “bON” : máy bơm đang bật
- “bOFF” : máy bơm đang tắt

Dựa trên dữ liệu trả về API sử dụng giao thức socketio để realtime trạng thái các thiết bị được hiển thị trên các browser-client đang request trước đó.

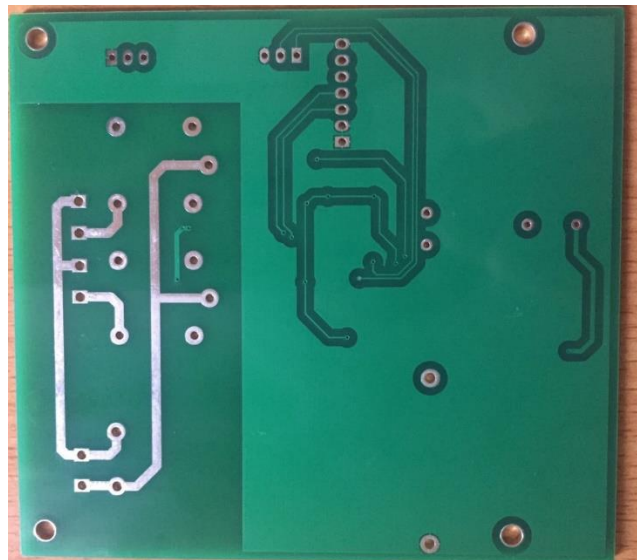
CHƯƠNG 3 KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

3.1 Kết quả đạt được

3.1.1 Phần cứng



Hình 3. 1 Mạch in thực tế mặt trên thiết bị Node device



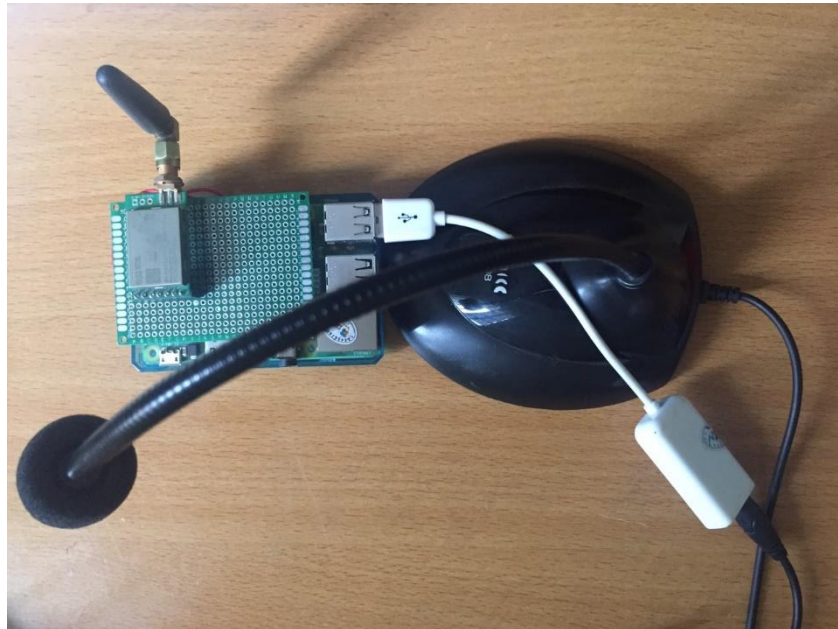
Hình 3. 2 Mạch in thực tế mặt dưới thiết bị Node device

Tiến hành hàn linh kiện sau khi đã gia công mạch in thực tế (Hình 3.1, Hình 3.2).



Hình 3. 3 Hình ảnh thực tế mặt trên thiết bị Node device sau khi hàn linh kiện

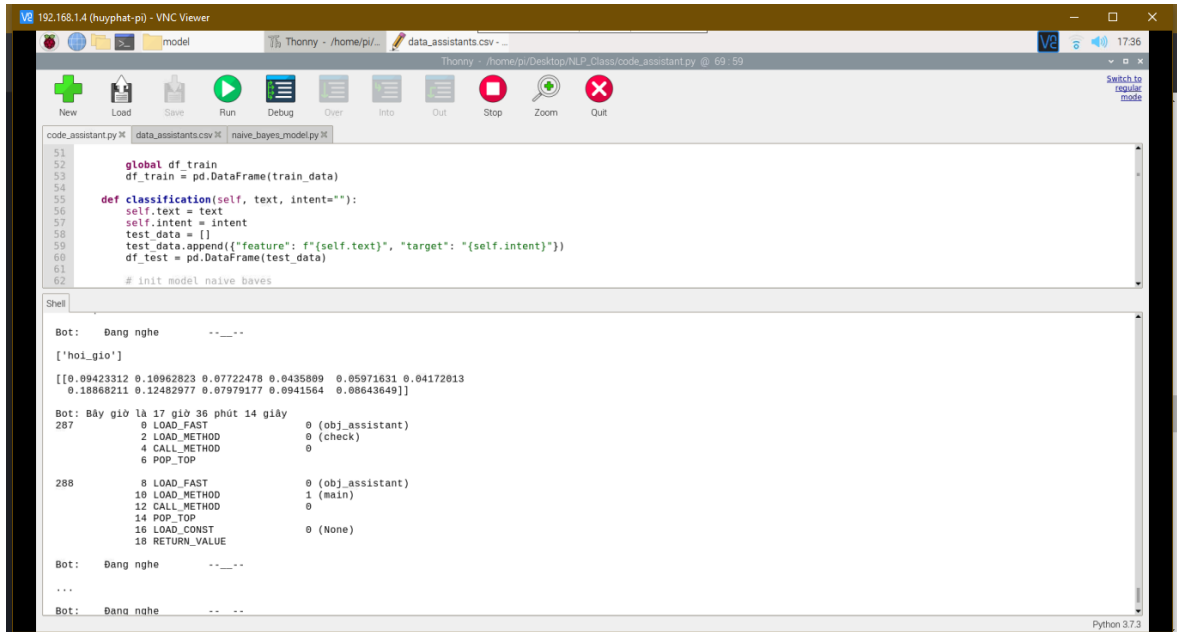
Sau khi thiết kế và hàn linh kiện (Hình 3.3). Dùng VOM kiểm tra board có bị đứt hay chập mạch hay không bằng cách kiểm tra các chân nguồn và chân điều khiển.



Hình 3. 4 Hình ảnh thực tế thiết bị Control device

Thiết kế hàn mạch kết nối module lora sx1278-e32 với raspberry pi 3 b+ và đi dây trên thiết bị Control device (Hình 3.4)

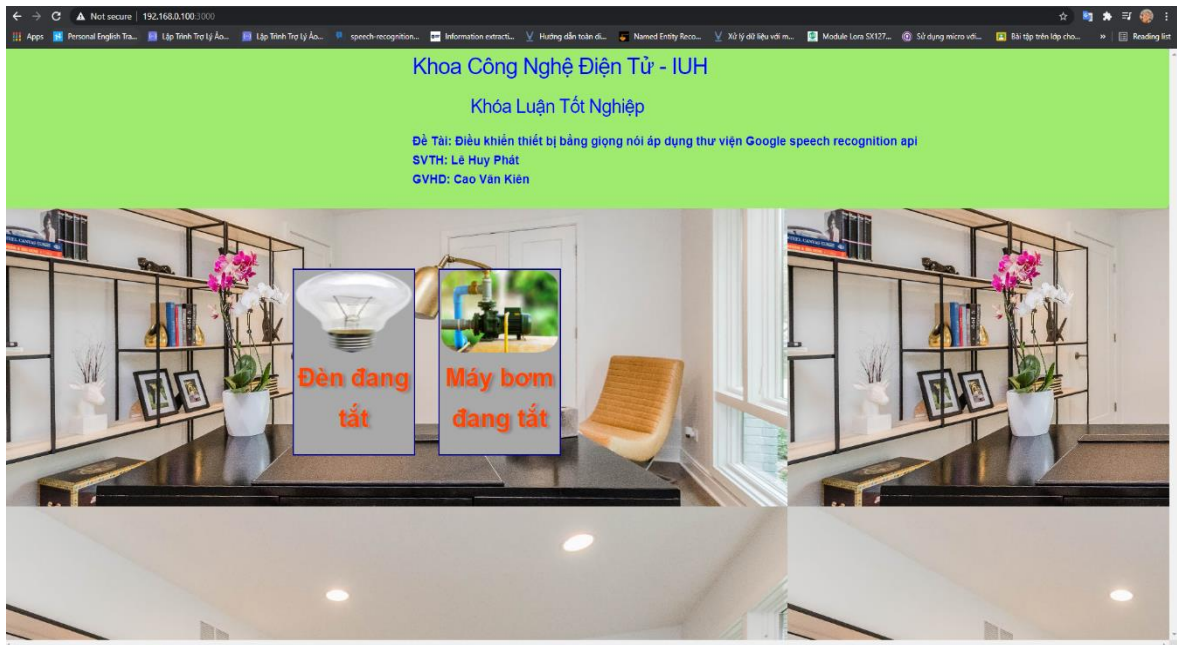
3.1.2 Model Text Classification tích hợp với Control device



Hình 3. 5 Model Text Classification tích hợp với Control device

Tích hợp model Text Classification vào chương trình chính của thiết bị Control device và kiểm tra. (Hình 3.5)

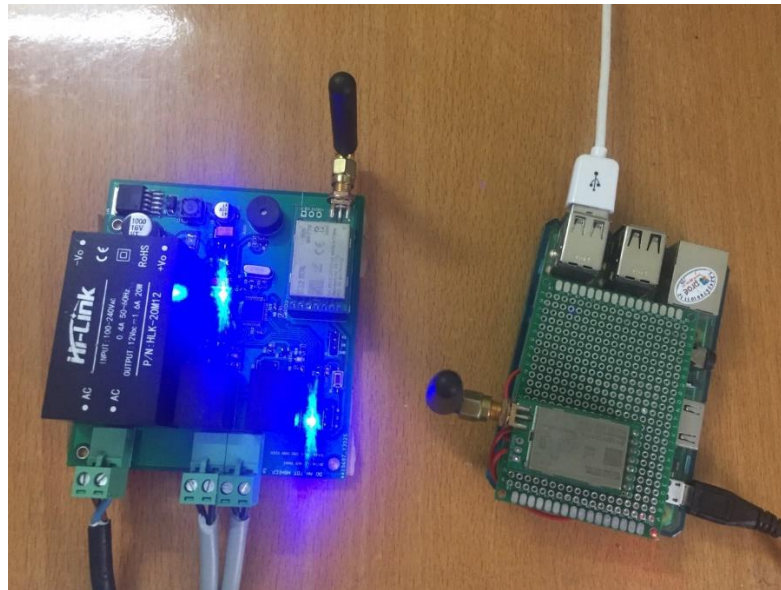
3.1.3 Webapp



Hình 3. 6 Màn hình giám sát thiết bị trên website localhost

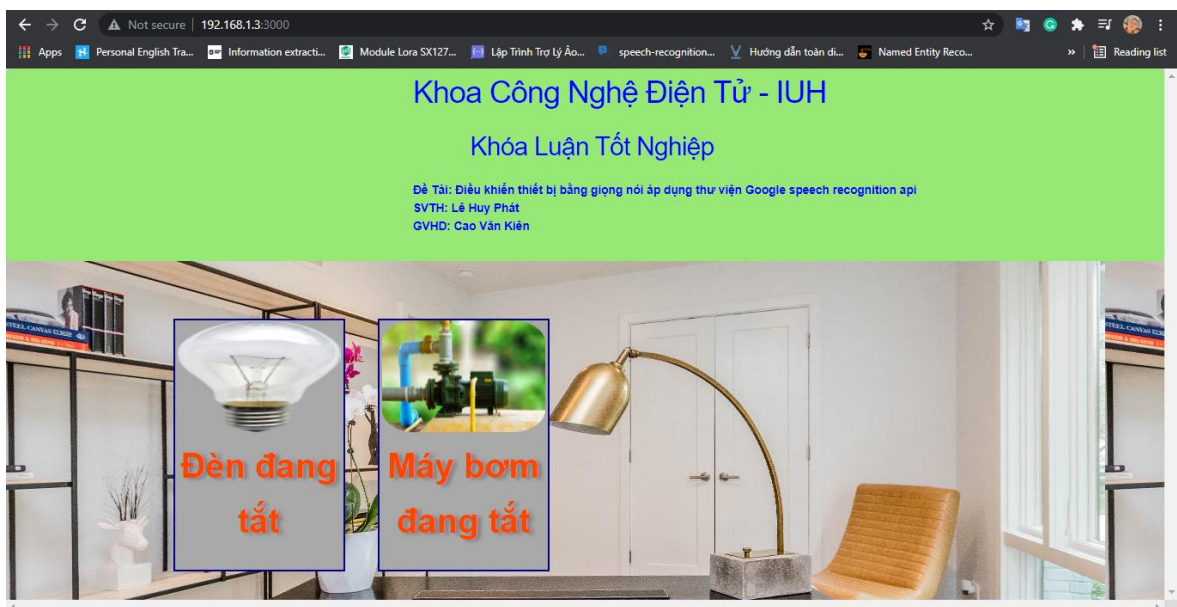
Màn hình giám sát thiết bị điện của hệ thống. (Hình 3.6). Sử dụng thư viện socketio của javascript (một thư viện áp dụng giao thức socket) để realtime dữ liệu.

3.2 Kết quả thực nghiệm



Hình 3. 7 Cấp nguồn cho 2 thiết bị Node device và Control device

Cấp nguồn cho các board trong hệ thống (Hình 3.7) và kết nối các ngõ ra điều khiển với các thiết bị điện: Nguồn cấp cho các board mạch in là 220VAC trực tiếp từ ổ cắm điện.



Hình 3. 8 Giao diện website trạng thái thiết bị sau khi khởi động hệ thống.

- Sau khi hệ thống được cấp nguồn, tất cả các chương trình đều được tự động cài đặt và chạy, khi đó ta truy cập vào địa chỉ ip mà router cấp cho raspberry với port 3000 sẽ hiển thị ra website kiểm tra trạng thái của các thiết bị điện (Hình 3.8).
- Thực hiện ra lệnh cho “Trợ lý” bằng giọng nói qua micro được kết nối với raspberry:

Ví dụ 1: Trường hợp tôi đang nói “bật đèn đi”:

```

Shell
Bot: Dạ

Bot:   Đang nghe   --__--

['bat_den']

[[[0.12029361 0.33363037 0.06843911 0.05380416 0.03633833 0.05567373
0.06465657 0.07263252 0.09288455 0.10164704]]]

dON
Bot: Đèn đã bật

287      0 LOAD_FAST          0 (obj_assistant)
          2 LOAD_METHOD        0 (check)
          4 CALL_METHOD         0
          6 POP_TOP

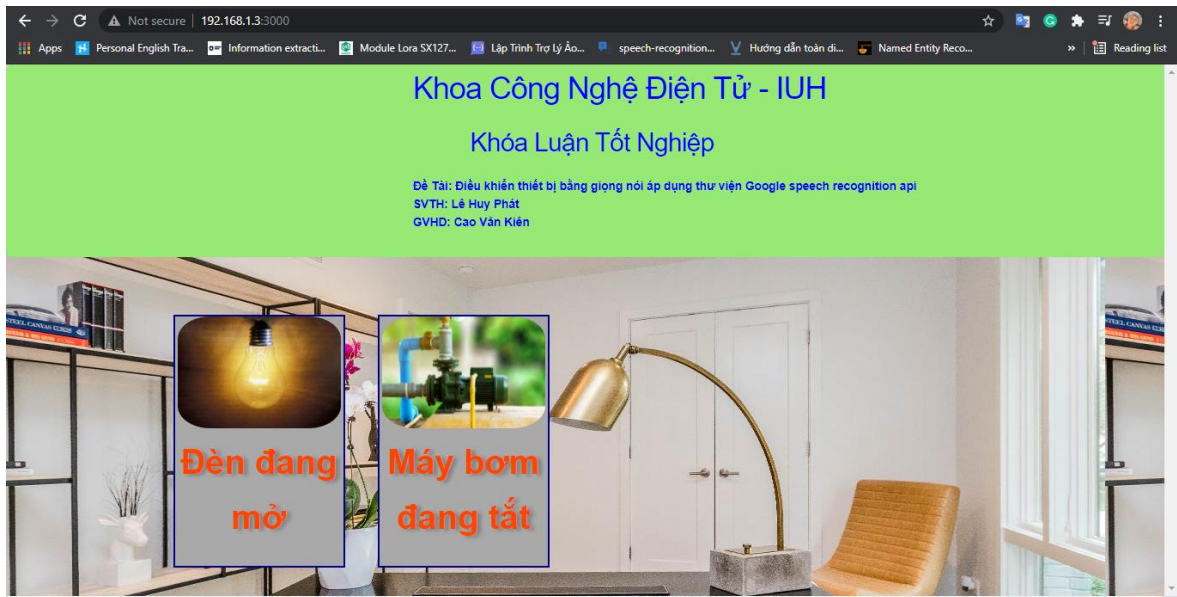
288      8 LOAD_FAST          0 (obj_assistant)
         10 LOAD_METHOD        1 (main)
         12 CALL_METHOD         0
         14 POP_TOP
         16 LOAD_CONST         0 (None)
         18 RETURN_VALUE

Bot:   Đang nghe   --__--

```

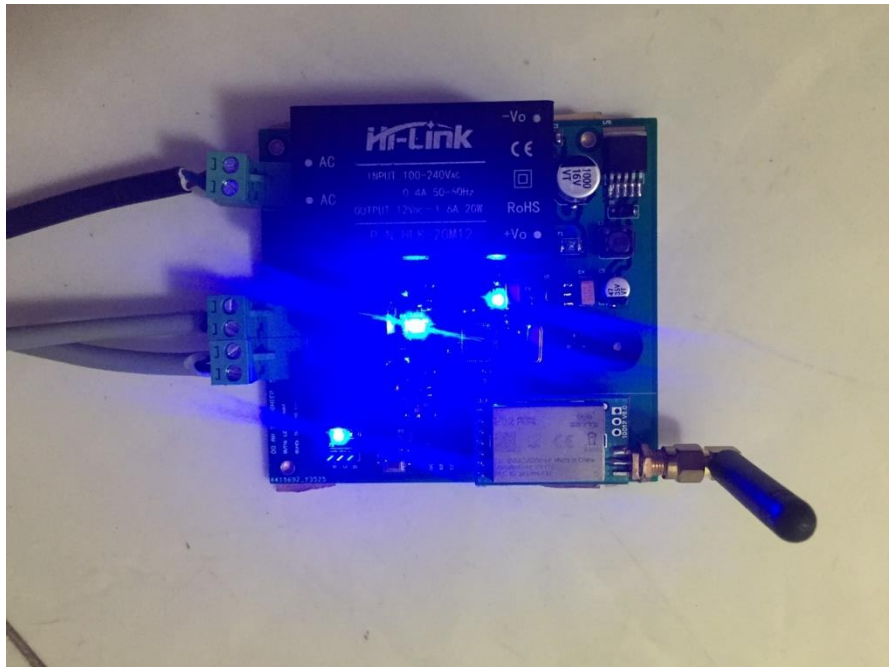
Hình 3. 9 Đang ra lệnh cho trợ lý bật đèn.

Sau khi ra lệnh bật đèn bằng giọng nói thiết bị sẽ thực hiện xử lý dữ liệu để đưa ra kết quả cuối cùng hiển thị ra màn hình terminal để lập trình viên kiểm tra trạng thái dữ liệu truyền, nhận và kết quả trả ra đã chính xác chưa. (Hình 3.9)



Hình 3. 10 Trạng thái đèn được mở trên website.

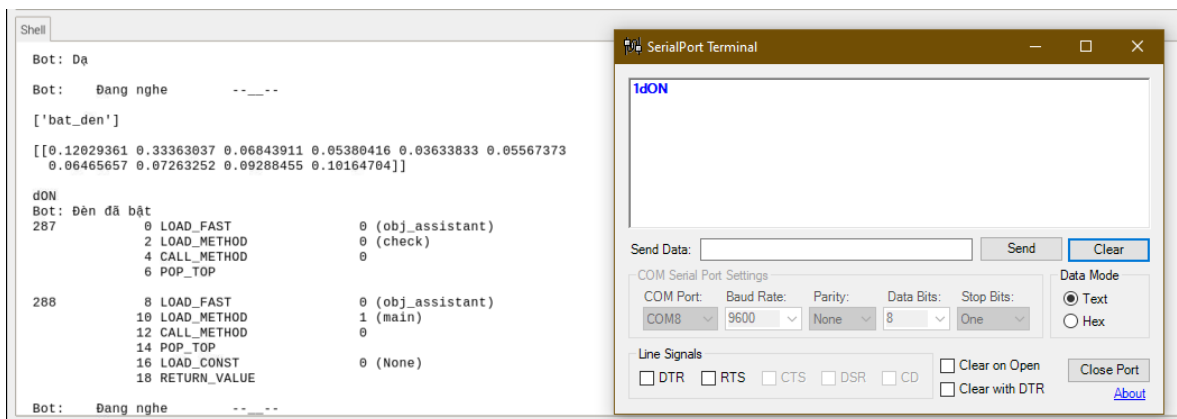
Trạng thái đèn thay đổi từ tắt sang bật trên website khi nhận được dữ liệu trả về từ thiết bị Node device. (Hình 3.10)



Hình 3. 11 Relay1 (đèn) được đóng.

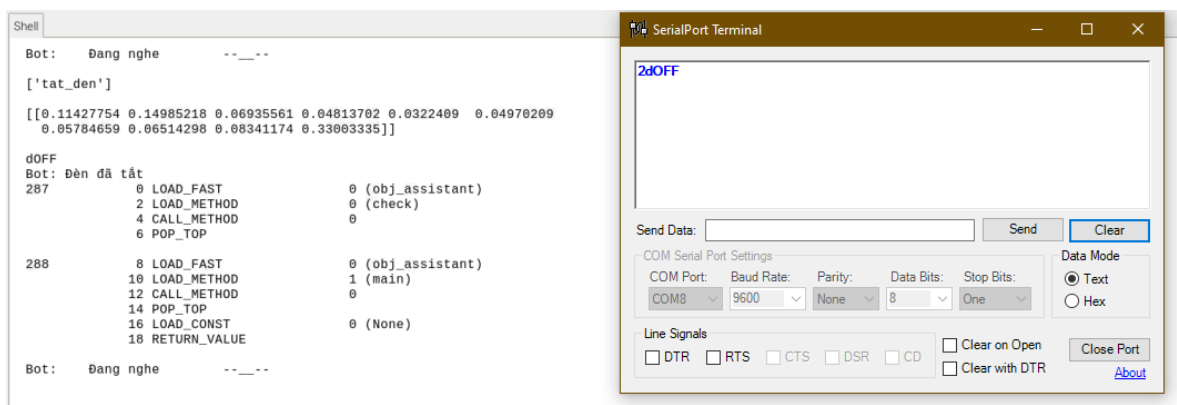
Trạng thái relay1 (đèn) thay đổi từ mở sang đóng khi nhận được lệnh điều khiển (Hình 3.11)

Dùng phần mềm “SerialPort Terminal” để bắt trạng thái truyền nhận giữa Control device với Node device khi được ra lệnh bật đèn. (Hình 3.12)



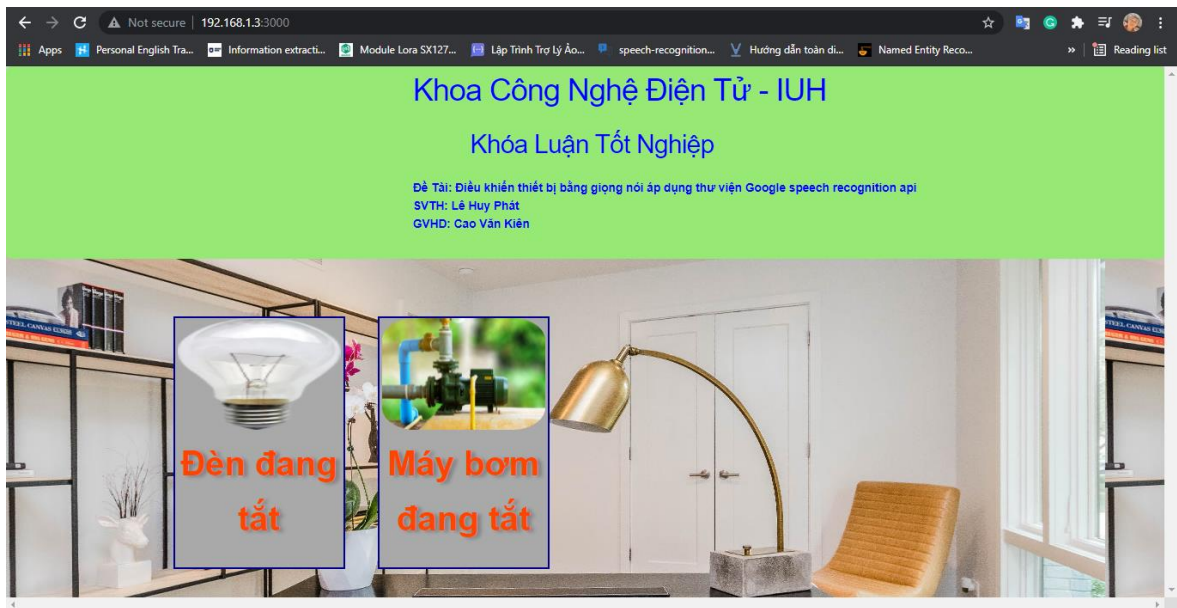
Hình 3. 12 Truyền nhận giữa Control device với Node device khi bật đèn

Ví dụ 2: Trường hợp tôi đang nói “Chối mắt quá” (ý nghĩa của câu này là đang muốn tắt đèn) hoặc chúng ta cũng có thể 1 câu khác với ý nghĩa tương tự.



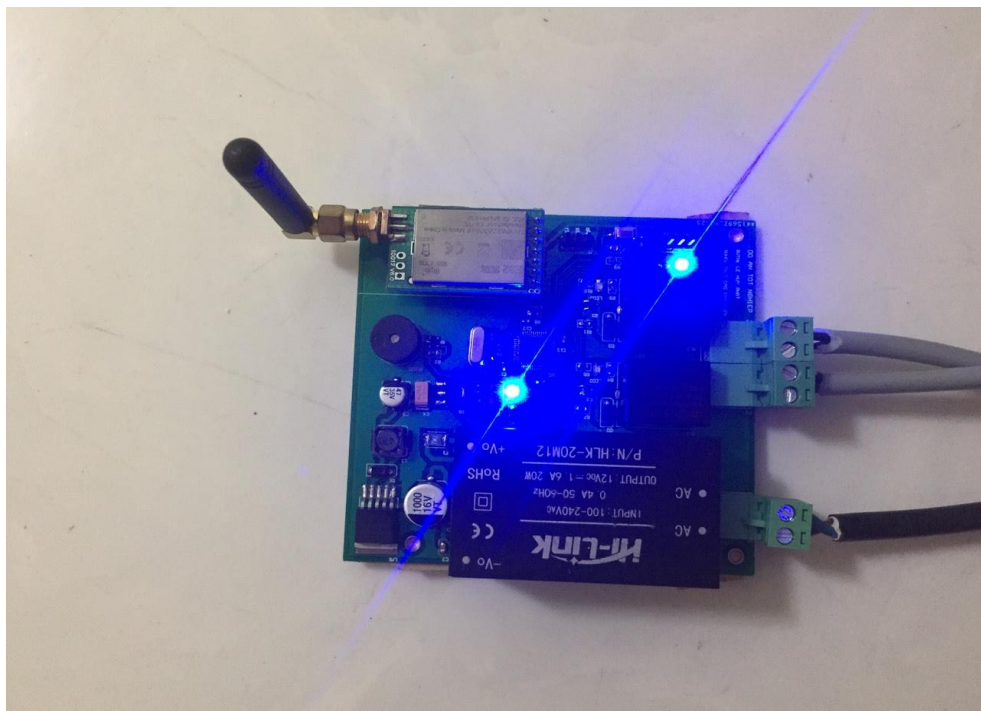
Hình 3. 13 Truyền nhận giữa Control device với Node device khi tắt đèn

Bắt dữ liệu truyền nhận giữa Control device với Node device khi điều khiển tắt đèn. (Hình 3.13)



Hình 3. 14 Trạng thái đèn được tắt trên website.

Trạng thái đèn thay đổi từ bật sang tắt trên website khi nhận được dữ liệu trả về từ thiết bị Node device. (Hình 3.14)



Hình 3. 15 Relay1 (đèn) khi được mở.

Trạng thái relay1 (đèn) thay đổi từ đóng sang mở khi nhận được lệnh điều khiển (Hình 3.15)

Ví dụ 3: Trường hợp tôi đang nói “Hôm nay trời nóng nhỉ” (ý của tôi là đang muốn nghe thời tiết hôm nay) hoặc bạn cũng có thể thử với một câu khác với ý nghĩa tương tự. (Hình 3.16)

```

Shell

Bot: Dạ

Bot:   Đang nghe           --__--

['hoi_thoi_tiet']

[[0.1282058  0.12056521 0.07169059 0.05036537 0.03220084 0.05130439
  0.11457242 0.23662784 0.08868784 0.10577971]]

Bot: Bạn muốn xem thời tiết ở đâu ạ.

Bot:   Đang nghe           --__--

Bot:
    Hôm nay là ngày 11 tháng 7 năm 2021
    Mặt trời mọc vào 5 giờ 37 phút
    Mặt trời lặn vào 18 giờ 19 phút
    Nhiệt độ trung bình là 28.01 độ C
    Áp suất không khí là 1009 héc tơ Pascal
    Độ ẩm là 89%
    Trời hôm nay quang mây. Dự báo mưa rải rác ở một số nơi.
287      0 LOAD_FAST          0 (obj_assistant)
          2 LOAD_METHOD      0 (check)
          4 CALL_METHOD      0
  
```

Hình 3. 16 Đoạn hội thoại hỏi thời tiết ở Hồ Chí Minh với trợ lý.

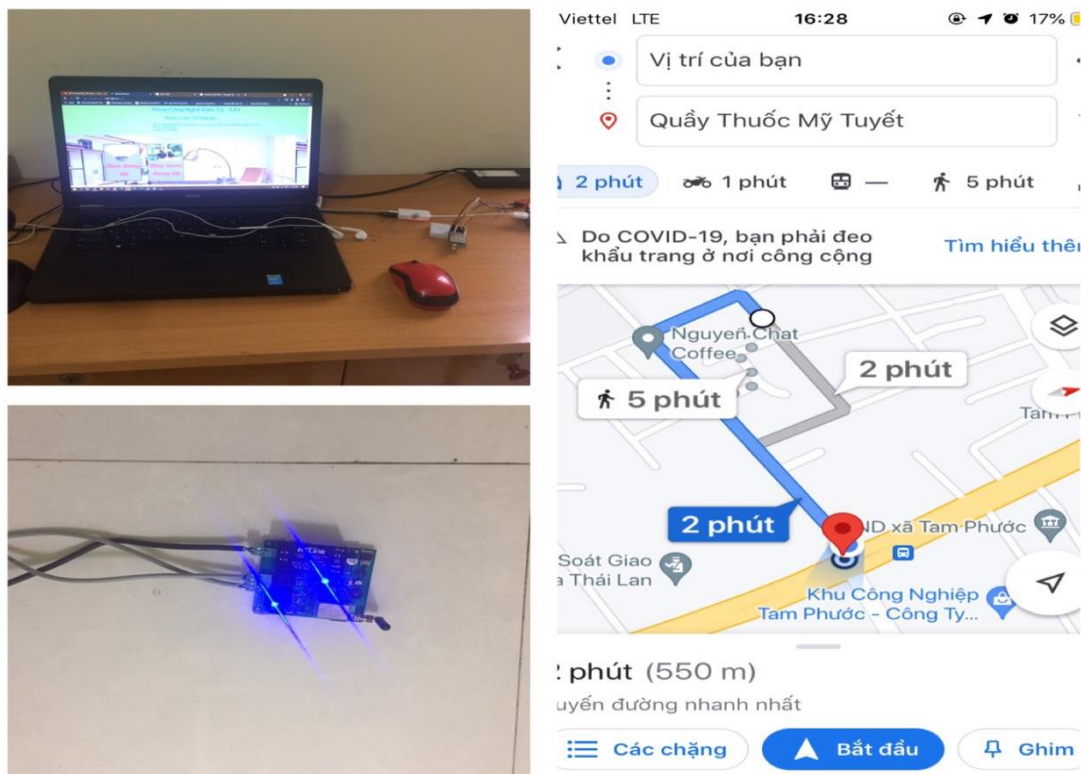
3.3 Nhận xét và đánh giá

Khoảng cách truyền nhận: Sử dụng anten 443MHz, khoảng cách truyền nhận giữa Node device và Control device. Khoảng cách truyền nhận chưa được xa, dựa vào bảng thực hiện thí nghiệm (Bảng 3).

Bảng 3 Bảng vị trí thực nghiệm thí nghiệm đo khoảng cách truyền nhận

Vị Trí thí nghiệm	Môi trường có vật cản	Môi trường không vật cản
Nhà riêng (228 Đường số 3, Tam Phước, Biên Hòa, Đồng Nai)	100m	500m

Thực hành 2 thí nghiệm tại địa điểm nhà riêng gồm có vật cản và không vật cản. (Hình 3.17)



Hình 3. 17 Thí nghiệm đo tại nhà riêng

Quá trình điều khiển, giám sát:

Thực hiện quá trình điều khiển và giám sát 20 lần thí nghiệm (Bảng 4).

Bảng 4 Điều khiển giám sát trên tổng 20 lần

Tác Vụ	Control device	Node device
Nhận dữ liệu thành công	20/20	20/20
Điều khiển thành công	20/20	20/20

Thực hiện dự đoán nhãn trên tổng 20 lần điều khiển (Bảng 5).

Bảng 5 Điều khiển giám sát dự đoán nhãn trên tổng 20 lần.

STT	Đoạn văn bản thí nghiệm	Nhãn mong muốn	Nhãn dự đoán được
1	Bật đèn lên	bat_den	bat_den
2	Sáng đèn	bat_den	bat_den
3	Sao tối thế	bat_den	bat_den
4	Mở đèn	bat_den	bat_den
5	Tối quá	bat_den	bat_den
6	Tắt đèn	tat_den	tat_den
7	Chói mắt quá	tat_den	tat_den
8	Turn off	tat_den	tat_den
9	Tắt đèn nào	tat_den	tat_den
10	Chói mắt	tat_den	tat_den
11	Bật máy bơm	bat_bom	bat_bom
12	Bật bơm	bat_bom	bat_bom
13	Bơm nước	bat_bom	bat_bom
14	Chạy máy bơm	bat_bom	bat_bom
15	Mở máy bơm	bat_bom	bat_bom
16	Tắt máy bơm	tat_bom	tat_bom
17	Dừng máy bơm	tat_bom	tat_bom

18	Off máy bơm	tat_bom	tat_bom
19	Stop máy bơm	tat_bom	tat_bom
20	Nước đầy rồi	tat_bom	tat_bom

Để mô tả cho quá trình huấn luyện và phân loại của model, tôi sẽ thực hành thí nghiệm test model với việc cho 2 nhãn (2 đầu ra) là ‘bat_den’ và ‘bat_bom’ như sau:

Với văn bản thí nghiệm là “Bật đèn lên”, khi qua model Text Classification thì cho ra 2 nhãn (2 ngõ ra của model), model sẽ lựa chọn xác suất của ngõ ra nào là lớn nhất để đưa ra kết quả cuối cùng.

```

PS D:\monhoc\Khoa Luan Tot Nghiep\File Project\Code and manual\NLP_Class> & C:/Users/Dell/AppData/Local/Programs/Python/Python36/python.exe "d:/monhoc/Khoa Luan Tot Nghiep/File Project/Code and manual/NLP_Class/Classification.py"
Nhập text: Bật đèn lên
C:\Users\Dell\AppData\Roaming\Python\Python36\site-packages\sklearn\linear_model\stochastic_gradient.py:183: FutureWarning: max_iter and tol parameters have been added in SGDClassifier in 0.19. If max_iter is set but tol is left unset, the default value for tol in 0.19 and 0.20 will be None (which is equivalent to -infinity, so it has no effect) but will change in 0.21 to 1e-3. Specify tol to silence this warning.
  FutureWarning)
['bat_den']
[[18.08468625 81.91531375]]
PS D:\monhoc\Khoa Luan Tot Nghiep\File Project\Code and manual\NLP_Class>

```

Hình 3. 18 Kết quả phân loại văn bản cho thí nghiệm “Bật đèn lên”.

Và ở đây ta thấy model đã xác định được nhãn có xác suất cao nhất là nhãn “bat_den” với tỉ lệ dự đoán chính xác lên đến 81,91%. (Hình 3.18)

Bây giờ hãy cùng tính lại xác suất để có đầu ra là nhãn “bat_den” theo công thức Naive Bayes đã phân tích ở phần “Giải thích lưu đồ (Hình 2.22)” mục **2.5.3 Lưu đồ giải thuật của thiết bị “Control device”**:

Bảng 6 Tập văn bản huấn luyện cho thí nghiệm test model

Ký hiệu	Văn bản huấn luyện	Nhãn
D1	Mở đèn lên	Bat_den
D2	Bật đèn đi	Bat_den
D3	Bật đèn nhanh đi	Bat_den
D4	mở đèn	Bat_den
D5	Bật máy bơm	Bat_bom

D6	Bật máy bơm đi	Bat_bom
D7	Bật bơm	Bat_bom
D8	Khởi động máy bơm	Bat_bom
D9	Bơm nước	Bat_bom

Để tính xác suất dựa theo công thức naive bayes classifier thì chúng ta cần thực hiện đúng các bước sau:

Bước 1: Tiền xử lý văn bản tập dữ liệu huấn luyện.

- Từ tập dữ liệu huấn luyện ở bảng 6 ta thực hiện tách như mô tả ở bảng 7.

Bảng 7 Tách từ với tập dữ liệu huấn luyện

Nhãn	Từ được tách trong tập văn bản huấn luyện	Số câu đã huấn luyện
Bat_den	['mở', 'đèn', 'lên', 'bật_đèn', 'đi', 'bật_đèn', 'nhANH', 'đi', 'mở', 'đèn']	4
Bat_bom	['bật', 'máy_bơm', 'bật', 'máy_bơm', 'đi', 'bật_bơm', 'khởi_động', 'máy_bơm', 'bơm', 'nước']	5

- Lập từ điển với tập dữ liệu đã được tách từ (đảm bảo mỗi từ chỉ xuất hiện 1 lần):
['mở', 'đèn', 'lên', 'bật_đèn', 'đi', 'nhANH', 'bật', 'máy_bơm', 'bật_bơm', 'khởi_động', 'bơm', 'nước']
- Từ điển của chúng ta có 12 từ, với tổng số câu huấn luyện là 9 câu.

Các kết quả tách từ và lập từ điển trên được tính toán bằng máy tính (Hình 3.19). Sử dụng thư viện ‘pyvi’ của python.

```

TERMINAL  PROBLEMS 121  OUTPUT  DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\monhoc\Khoa Luan Tot Nghiep\File Project\Code and manual\NLP_Class> & C:/Users/Dell/AppData/Local/Programs/Python/Python36/python.exe "d:/monhoc/Khoa Luan Tot Nghiep/File Project/Code and manual/NLP_Class/tach_tu_pyvi.py"
Tách từ:
[['mở', 'đèn', 'lên'], ['bật_đèn', 'đi'], ['bật_đèn', 'nhANH', 'đi'], ['mở', 'đèn'], ['bật', 'máy_bom'], ['bật', 'máy_bom', 'đi'], ['bật_bom'], ['khởi_động', 'máy_bom'], ['bom', 'nước']]

Danh sách các từ được tách của tệp huấn luyện:
['mở', 'đèn', 'lên', 'bật_đèn', 'đi', 'bật_đèn', 'nhANH', 'đi', 'mở', 'đèn', 'bật', 'máy_bom', 'bật', 'máy_bom', 'đi', 'bật_bom', 'khởi_động', 'máy_bom', 'bom', 'nước']

Từ Điển của tệp huấn luyện:
['mở', 'đèn', 'lên', 'bật_đèn', 'đi', 'nhANH', 'bật', 'máy_bom', 'bật_bom', 'khởi_động', 'bom', 'nước']

Số từ có trong Từ Điển: 12

Tách từ cho văn bản test: Bật_đèn lên
PS D:\monhoc\Khoa Luan Tot Nghiep\File Project\Code and manual\NLP_Class>

```

Hình 3. 19 Kết quả tách từ và lập từ điển bằng máy tính.

Bước 2: Lập từ điển cho văn bản huấn luyện và tính tần suất xuất hiện của nhãn $P(c_j)$ trên toàn bộ tập dữ liệu.

- Nhận thấy rằng có 2 class lần lượt là: ‘bat_den’, ‘bat_bom’.
- Dựa vào dữ liệu ở bảng 7 và kết quả tính toán bằng máy tính ở hình 3.19 ta tính được xác suất của các class trên:

$P(class) = \text{tổng số câu thuộc class trong tệp huấn luyện} / \text{tổng số câu trong tệp huấn luyện}.$

$P(bat_den)=4/9$, $P(bat_bom)=5/9$

Từ điển $V=\{ 'mở', 'đèn', 'lên', 'bật_đèn', 'đi', 'nhANH', 'bật', 'máy_bom', 'bật_bom', 'khởi_động', 'bom', 'nước' \}$

- Tổng cộng số phần tử trong từ điển là $|V| = 12$

Bước 3: Huấn luyện và phân loại.

- 🚦 Minh họa quá trình huấn luyện và dự đoán khi sử dụng Multinomial Naive Bayes theo bảng 8, bảng 9. Trong đó có sử dụng Laplace smoothing với

$\alpha = 1$. Áp dụng thuật toán Bag of words (Trích xuất đặc trưng) ở biểu thức số (2) phần “Giải thích lưu đồ (Hình 2.22)” mục **2.5.3 Lưu đồ giải thuật của thiết bị “Control device”**.

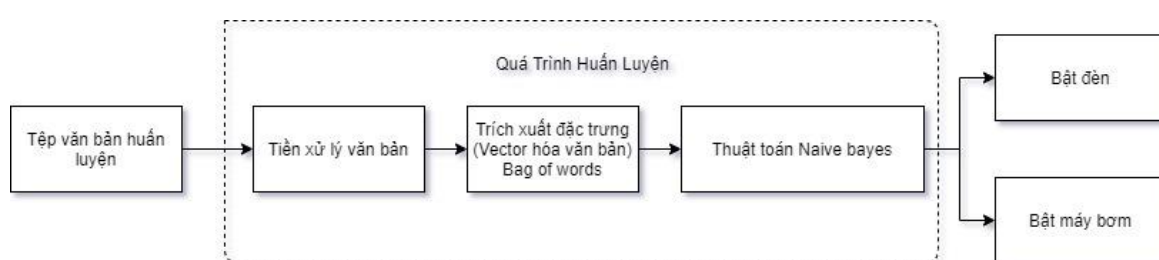
Bảng 8 Minh họa quá trình huấn luyện khi sử dụng Multinomial Naive Bayes cho nhãn ‘bat_den’ thí nghiệm test model.

STT	Từ trong từ điển	D1: x_1	D2: x_2	D3: x_3	D4: x_4	Total	$\Rightarrow \lambda_{bd}$
1	Mở	1	0	0	1	2	3/22
2	đèn	1	0	0	1	2	3/22
3	Lên	1	0	0	0	1	2/22
4	Bật_đèn	0	1	1	0	2	3/22
5	Đi	0	1	1	0	2	3/22
6	Nhanh	0	0	1	0	1	2/22
7	Bật	0	0	0	0	0	1/22
8	Máy_bom	0	0	0	0	0	1/22
9	Bật_bom	0	0	0	0	0	1/22
10	Khởi_động	0	0	0	0	0	1/22
11	Bom	0	0	0	0	0	1/22
12	Nước	0	0	0	0	0	1/22
$ V $ =12						N_{bd} = 10	$N_{bd} + V $ = 22

Bảng 9 Minh họa quá trình huấn luyện khi sử dụng Multinomial Naive Bayes cho nhãn ‘bat_bom’ thí nghiệm test model.

STT	Từ trong từ điển	D5: x_5	D6: x_6	D7: x_7	D8: x_8	D9: x_9	Total	$\Rightarrow \lambda_{bb}$
1	Mở	0	0	0	0	0	0	1/22
2	đèn	0	0	0	0	0	0	1/22
3	Lên	0	0	0	0	0	0	1/22
4	Bật_đèn	0	0	0	0	0	0	1/22
5	Đi	0	1	0	0	0	1	2/22
6	Nhanh	0	0	0	0	0	0	1/22
7	Bật	1	1	0	0	0	2	3/22
8	Máy_bom	1	1	0	1	0	3	4/22
9	Bật_bom	0	0	1	1	0	2	3/22
10	Khởi_động	0	0	0	0	0	0	1/22
11	Bom	0	0	0	0	1	1	2/22
12	Nước	0	0	0	0	1	1	2/22
$ V $ =12							N_{bd} = 10	$N_{bb} + V $ = 22

Mô tả quá trình huấn luyện với 2 lớp đầu ra ‘bat_den’ và ‘bat_bom’ tại hình 3.20



Hình 3. 20 Mô tả quá trình huấn luyện với 2 lớp đầu ra.

🚦 Quá trình dự đoán văn bản:

Văn bản cần dự đoán của chúng ta là : **“Bật đèn lên”** ký hiệu là D10, sau khi tách từ cho câu này ta được 2 từ: ‘Bật_đèn’ và ‘lên’.

⇒ Vector đặc trưng của văn bản D10: $x_{10} = [0,0,1,1,0,0,0,0,0,0,0,0]$, có 12 chiều được biểu diễn theo 12 từ trong từ điển của model được tính ở bước 2.

Dựa theo biểu thức tổng quát (8) và kết quả ở bảng 8, bảng 9 ta tính xác suất cho các nhãn:

- Tính xác suất cho đầu ra nhãn ‘bat_den’:

$$P(\text{bat_den} | D10) \propto P(\text{bat_den}) \prod_{i=1}^D P(x_i | \text{bat_den}) = 4/9 * 2/22 * 3/22 = 2/363$$

- Tính xác suất cho đầu ra nhãn ‘bat_bom’:

$$P(\text{bat_bom} | D10) \propto P(\text{bat_bom}) \prod_{i=1}^D P(x_i | \text{bat_bom}) = 5/9 * 1/22 * 1/22 = 5/3872$$

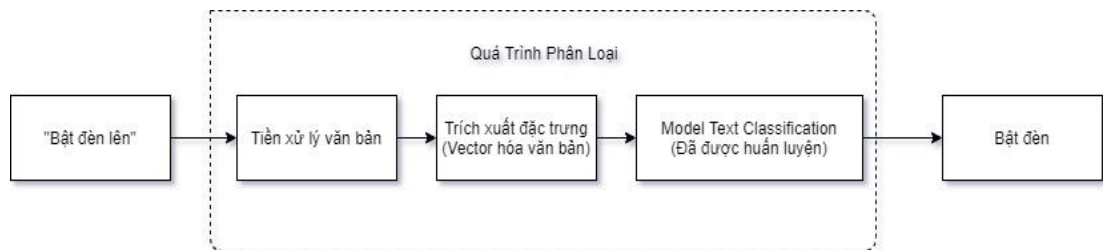
Ta thấy: $P(\text{bat_den} | D10) > P(\text{bat_bom} | D10) \Rightarrow D10 \in \text{class}(\text{bat_den})$

Xác suất cần tìm:

$$P(\text{bat_den} | D10) = \frac{\frac{2}{363}}{\left(\frac{2}{363} + \frac{5}{3872}\right)} = 0,8101$$

$$P(\text{bat_bom} | D10) = \frac{\frac{5}{3872}}{\left(\frac{2}{363} + \frac{5}{3872}\right)} = 0,1899$$

Mô tả quá trình phân loại văn bản với đầu vào là các từ trong văn bản đó [‘Bật_đèn’, ‘lên’] (2 đầu vào) của văn bản test “**Bật đèn lên**”: (Hình 3.21)



Hình 3. 21 Mô tả quá trình phân loại với đầu vào là các từ của văn bản test

Vậy với câu “Bật đèn đi” được đưa vào model sẽ có kết quả dự đoán cho nhãn Bật đèn (bat_den) là : 81,01% , và cho nhãn bật bom (bat_bom) là: 18,99%. Kết quả tính tay đúng với kết quả dự đoán của máy tính ở Hình 3.18.

- Trong quá trình chuyển đổi giọng nói sang văn bản cần nói chính xác các từ cần nói, sai từ có thể dẫn đến sai nghĩa của câu và điều đó dẫn đến model Text Classification dự đoán sai.
- Thời gian chuyển giọng nói sang văn bản khá chậm, mất tới 1 đến 2 giây mới nhận được đoạn văn bản trả về, do đang sử dụng API chuyển đổi giọng nói của google cần thời gian request đáp ứng.
- Tỷ lệ dự đoán chính xác còn khá thấp, do dữ liệu huấn luyện khá là ít.
- Hệ thống website hiển thị trạng thái chỉ hoạt động khi thiết bị request cùng trong một lớp mạng với thiết bị Control device.

CHƯƠNG 4 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

Dựa vào các số liệu trong quá trình chạy thực nghiệm kiểm tra khả năng truyền nhận tín hiệu của lora sx1278 ở các trường hợp có vật cản 100m và không vật cản 500m, hệ thống hoạt động tốt và đã đạt được những mục tiêu đề ra ban đầu và khá ổn định. Dựa vào quá trình thí nghiệm test model với mô hình 2 lớp đầu ra, mô hình máy học của hệ thống dự đoán tốt và tỉ lệ chính xác ở mức khá. Và qua quá trình thực nghiệm điều khiển giám sát 20 lần, đưa ra được kết luận hệ thống hoạt động ổn định. Mô hình nhỏ gọn, an toàn và dễ sử dụng. Tuy nhiên thời gian đáp ứng điều khiển thiết bị vẫn chưa được nhanh (thời gian chuyển đổi giọng nói sang văn bản).

Đề tài luận văn này là một giải pháp học tập hiệu quả cho sinh viên có cơ hội được áp dụng các kiến thức đã học vào thực tiễn. Việc thực hiện mô hình giúp tác giả hệ thống, củng cố lại các kiến thức đã học, đồng thời phát triển kỹ năng tự học, kỹ năng làm việc theo kế hoạch.

4.2 Hướng phát triển

- Thêm nhiều thiết bị cho hệ thống
- Mở rộng thêm phần điều khiển bằng giọng nói trực tiếp trên website
- Đưa hệ thống ra ngoài internet
- Thêm lượng lớn dữ liệu cho việc huấn luyện model Text Classification để dự đoán chính xác với xác suất cao hơn.
- Phát triển thêm phần xem trạng thái và điều khiển thiết bị trên app mobile.

TÀI LIỆU THAM KHẢO

- [1] McCallum Andrew, and Kamal Nigam, "A comparison of event models for naive bayes text classification," *AAAI-98 workshop on learning for text categorization*, pp. Vol. 752, No. 1, pp. 41-48, 1998, July.
- [2] H. SHIMODAIRA, "Text classification using naive bayes," *Learning and Data Note*, pp. 7, 1-9, 2014.
- [3] T. T. Mai, "THIẾT KẾ ĐIỀU KHIỂN THIẾT BỊ BẰNG GIỌNG NÓI VỚI GOOGLE ASSISTANT (GOOGLE HOME)," Trường Đại Học Bà Rịa-Vũng Tàu, Bà Rịa-Vũng Tàu, Tháng 06 Năm 2019.
- [4] S. Ovadia, "Automate the internet with “if this then that”(IFTTT)," *Behavioral & social sciences librarian*, Vols. 208-211, p. 33.4, 2014.
- [5] Tulshan, Amrita S., and Sudhir Namdeorao Dhage, "Survey on virtual assistant: Google assistant, siri, cortana, alexa.," *International symposium on signal processing and intelligent recognition systems*, pp. 190-201, 2018.