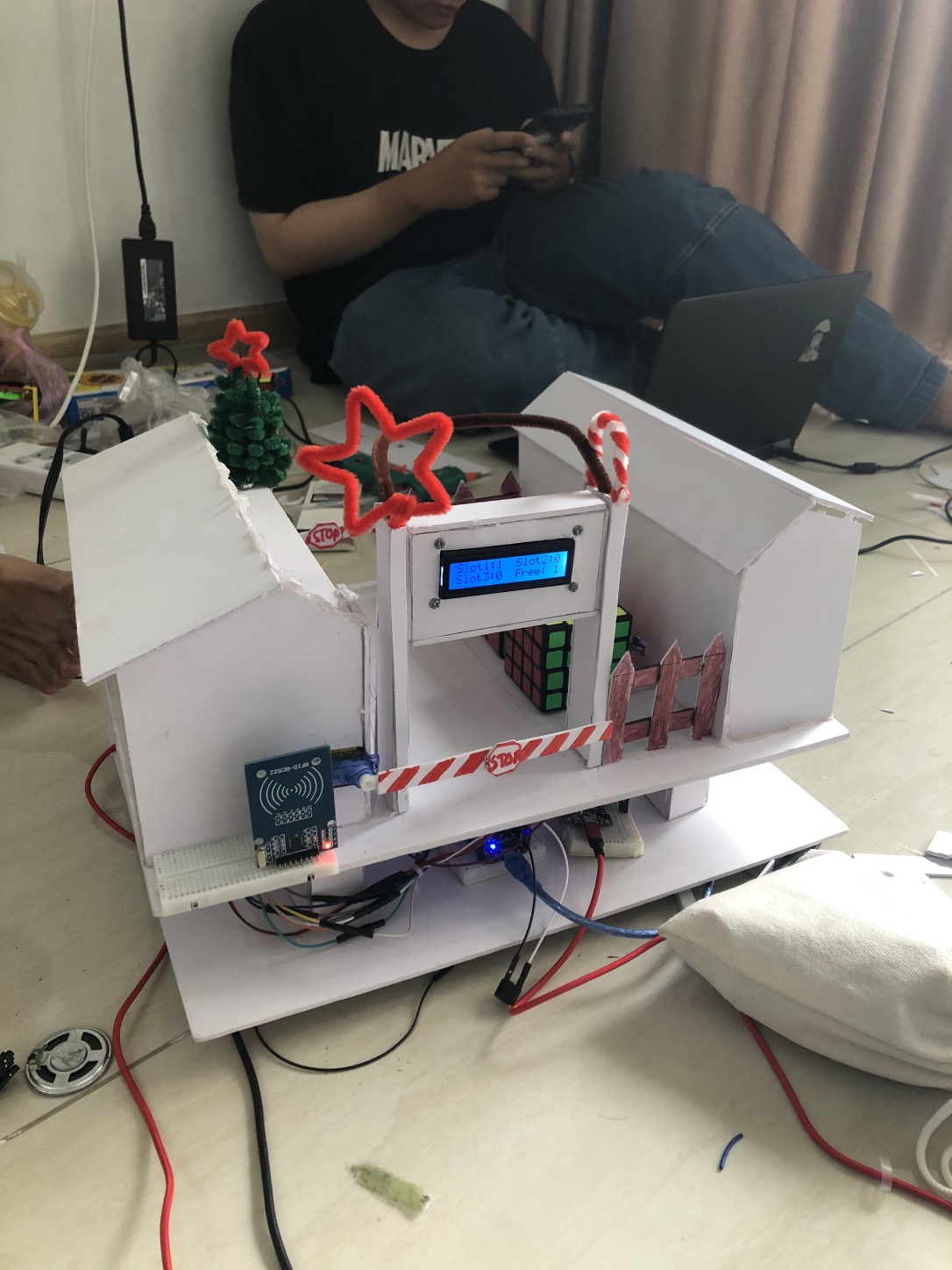
eBãi gửi xe tự động-report(demo)

Intro:

Thống kê và báo cáo từ bãi gửi xe của những địa điểm công cộng như chung cư , bệnh viện hoặc trung tâm thương mại, việc kiêm xoát só lượng phương tiện ra vào bằng cách đếm là bất khả thi vì độ nhiễu của nõ, hiện tượng mất kiểm soát trong bãi gửi xe sẽ gây ra un tắc hoặc quá sức chứa dẫn tới kẹt bãi gửi xe, từ đó gây ra nhiều hệ quả nghiêm trọng ảnh hưởng đến lợi ích của người sử dụng dịch vụ tại nhiều địa điểm. A parking lot full of cars

Description automatically generated

Như là 1 giải pháp của việc mất kiểm soát ở bãi gửi xe, dự án bãi gửi xe thông minh đã ra đời nhằm giải quyết vấn đề gây nhức nhối này với nhiều công nghệ tiên tiến và hiện đại.

Methodology:

1. Nguyên lý hoạt động:

Bãi gửi xe thông minh được chia làm 3 phần và vận hành dựa trên 4 hệ thống khác nhau, được liên kết qua 1 sever, trong đó sẽ có 1 hệ thống quản lý số lượng xe trong bãi, 2 hệ thống quản lý cổng ra và cổng vào, và 1 hệ thống quản lý âm thanh và tương tác với người sử dụng.

Đầu tiên hệ thống quản lý số lượng xe hoạt động bằng mạch esp8266 sẽ luôn cập nhật dữ liệu từ 3 cảm biến hồng ngoại (IR Sensor) , từ đó lưu vào các biến sensorValue1, sensorValue2, sensorValue3 nhằm mục đích hiển thị ra màn hình LCD và tính tổng, cái mà cập nhật dữ liệu số xe còn trống trong bãi gửi xe.

Sau đó hệ thống quản lý cổng vào hoạt động bằng mạch esp8266, nơi luôn cập nhật dữ liệu tổng số xe trong bãi từ sever về biến Sum, kế bên biến lưu trữ số lượng xe có 1 danh sách ID thẻ đã được đã ký, từ 2 dữ liệu trên hệ thống sẽ quyết định có cho động cơ servo mở cổng 1 gốc 90 độ hay không. Nếu được mở cổng hệ thống sẽ gửi ID của thẻ đầu vô lên sever nhằm mục đích lưu trữ ID có trong bãi và lưu lại thời gian xe vào để tiện cho mục đích bảo mật. Cuối cùng hệ thống sẽ gửi tín hiệu cho hệ thống quản lý âm thanh để đưa ra lời commend:” please come in” như một hệ thống chuyên nghiệp. Ngoài ra nếu số lượng xe bằng 3 (số lượng tối đa mà bãi chứa được) thì hệ thống sẽ gửi tín hiệu cho hệ thống quản lý âm thanh để phát ra âm thanh “sorry the parking’s slot is full”

Bên cạnh hệ thống cổng vào hoạt động bằng mạch esp8266, hệ thống cổng ra luôn lấy số ID từ sever về, xét các điều kiện đặt biệt để lưu vào 1 mảng chứa các phần tử ID thẻ xe đã vào bãi. Sau đó nếu có tín hiệu từ thẻ từ, hệ thống sẽ so sánh ID của thẻ từ và ID trong mảng nếu chỉ số ID trên trùng khớp, hệ thống sẽ kích hoạt động cơ servo cho xe ra và đồng thời xóa mã số ID đã đọc được từ thẻ từ trước đó. Cuối cùng hệ thống sẽ gửi tín hiệu cho hệ thống quản lý âm thanh để đưa ra lời commend:” please come out” như một hệ thống chuyên nghiệp.

Cuối cùng hệ thống quản lý âm thanh hoạt động bằng Python, sẽ đóng vai trò khởi tạo sever, sau đó liên tục lấy dữ liệu từ sever về để sử lý và phát ra âm thanh phù hợp theo từng tình huống khác nhau.

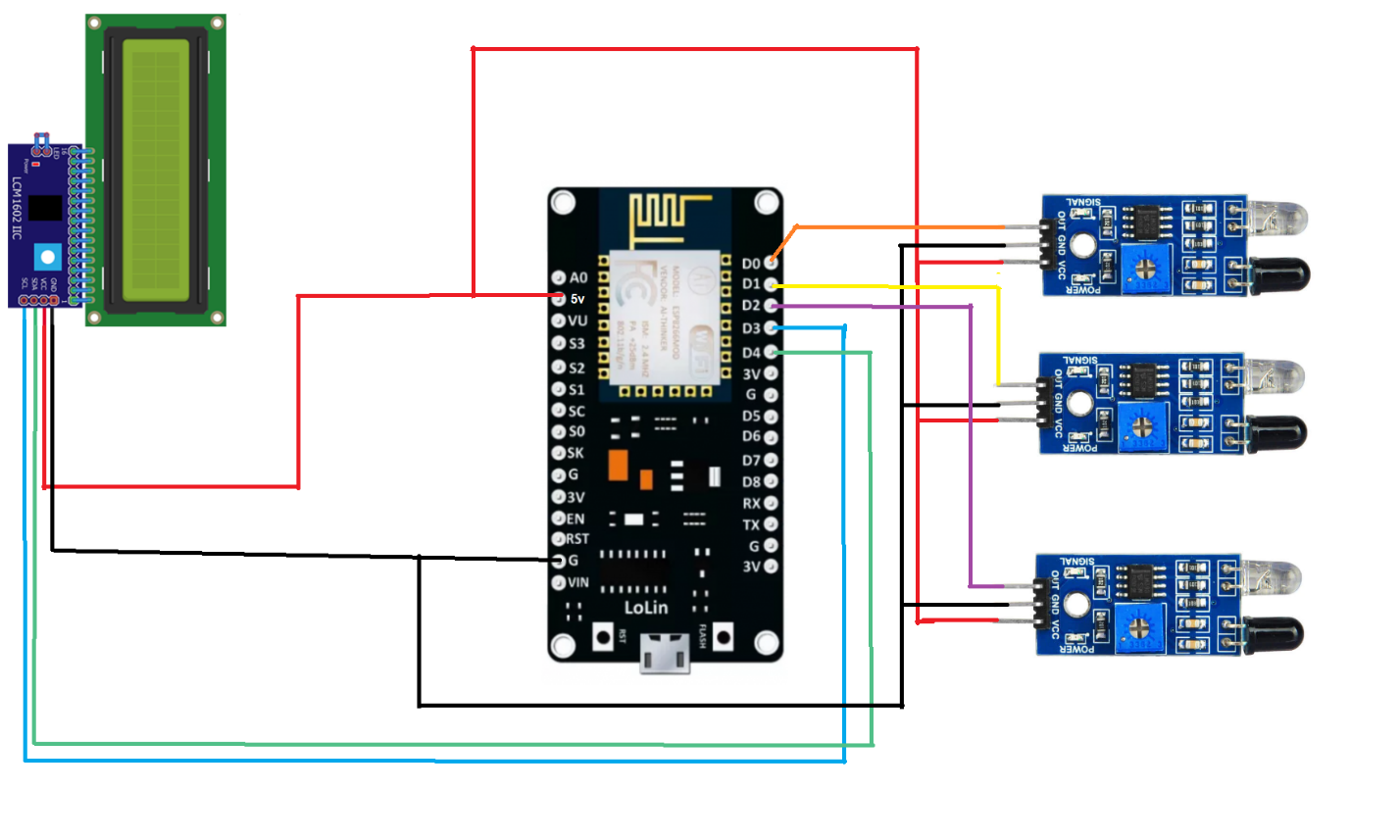
2. Hệ thống:

a) Hệ thống quản lý số lượng xe

- Cấu tạo

Được cấu tạo từ 3 IR sensor liên kết với 1 mạch esp8266 và 1 màng hinh lcd

- Sơ đồ mạch điện, đi dây



Nối cổng gnd mạch esp8266 với cổng gnd của màng hình led có i2c, và cổng gnd của 3 cảm biến IR (cảm biến hồng ngoại).

Nối cổng 5v mạch esp8266 với cổng vcc của màng hình led có i2c, và cổng vcc của 3 cảm biến IR (cảm biến hồng ngoại).

Với màn hình led, nối cổng SCL với chân D3 của mạch esp8266, và cổng SDA với chân D4.

Với cảm biến IR nối cổng OUTPUT của cảm biến trên, giữa, và dưới lần lượt với cổng D0, D1,D2.

- Nguyên lý hoạt động

Khi bắt đầu khởi tạo sever, 3 cảm biến hồng ngoại sẽ liên tục nhận tín hiệu từ môi trường để đếm số lượng xe trong bãi, với 0 là có xe và 1 là trống. Từ những tín hiệu này chương trình arduino sẽ được dùng để tính toán và gửi tính hiệu lên server Blynk để lưu trữ và sử lý cho việc vận hành hệ thống. Kế bên những tín hiệu được gửi đi chúng tôi còn sử dụng những tín hiệu này để hiển thị lên màng hình LED nhằm mục đích thông báo cho người sử dụng dịch vụ về tình hình của bãi xe.

A screenshot of a computer

Description automatically generated

Hình 2( màn hình LED)

- Kỹ thuật (technical)

Đầu tiên chúng tôi sẽ đọc giá trị từ cảm biến và lưu vào các biến sensorValue1, sensorValue2, sensorValue3. Vì giá trị đọc từ cảm biến bị ngược, tức 0 cho có và 1 cho không có tín hiệu, nên chúng tôi sẽ trừ 1 và nhân với (-1) giá trị từ cảm biến để giá trị đúng và thuận tiện hơn cho việc sử lý của hệ thống.

A screen shot of a computer

Description automatically generated

Sau đó những dữ liệu này sẽ được in ra màng hình bằng hàm lcd.setCursor để chọn vị trí với vị trí đầu là hàng, vị trí sau là cột, kế bên đó hàm lcd.print được sử dụng để in ra mà ng hình

A screen shot of a computer code

Description automatically generated

Và 1 phần khác của hệ thống sẽ tính tổng số xe trong bãi bằng hàm sum và gửi lên sever bằng hàm Blynk.virtualWrite, với sum luôn được trả về giá trị bằng 0 trước khi tính để tránh nhiễu, và V1, V2, V3, V4 là địa chỉ lưu của các dữ liệu lấy được từ cảm biến ở sever, và sensorValue là giá trị tương ứng được lưu tại địa chỉ V(number). Sau đó để tránh hiện tượng lag do cập nhật liên tục chúng tôi sẽ đợi 100ms để cập nhật lại giá trị lên hệ thống (vì 100ms = 1/10s là 1 khoảng thời gian rất nhỏ, không đáng kể nên sẽ không ảnh hưởng đến cách hệ thống vận hành)

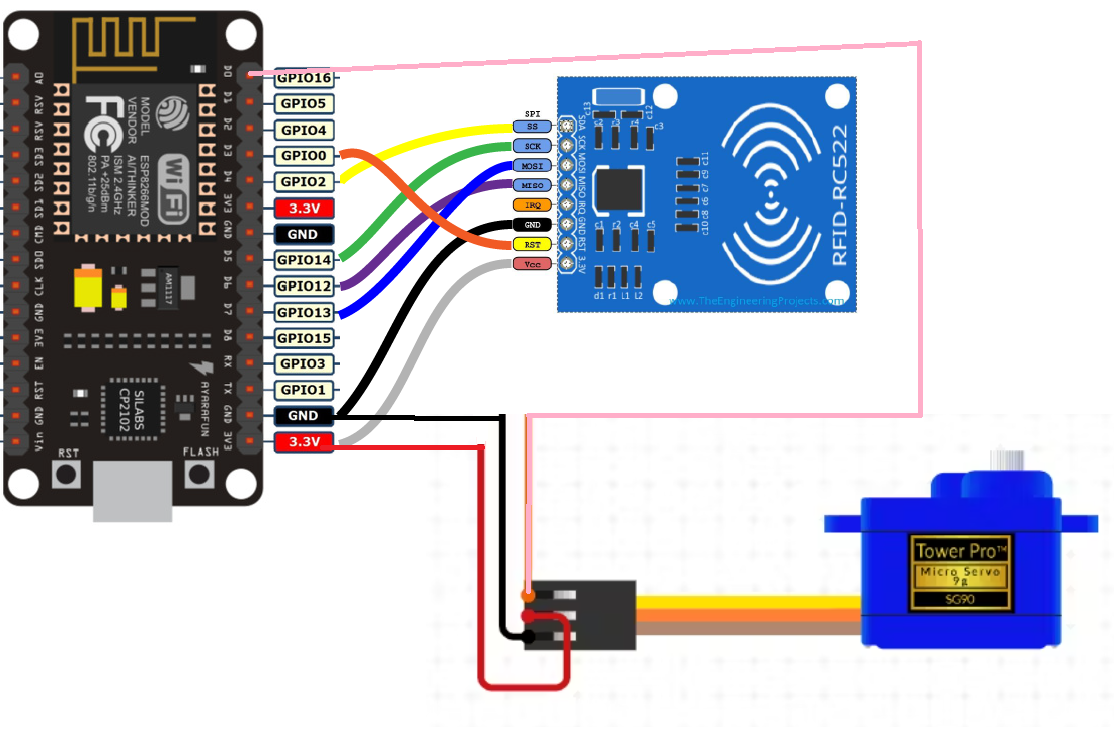
A screenshot of a computer program

Description automatically generated

b) Hệ thống quản lý cổng vào

- Cấu tạo: gồm 1 mạch esp8266, 1 mạch RFID NFC 13.56MHz RC522, 4 thẻ từ đã được đăng ký trong danh sách, 1 thẻ từ ngoài, 1 động cơ servo.

- Sơ đồ mạch điện, đi dây



Nối cổng gnd mạch esp8266 với cổng gnd của Rfid Rc522 Interfacing, và cổng gnd của động cơ servo

Nối cổng 5v mạch esp8266 với cổng vcc của Rfid Rc522 Interfacing, và cổng vcc của động cơ servo

Với Rfid Rc522 Interfacing, nối cổng SDA với chân D4, cổng SCK với chân D5, cổng MISO với chân D6, MOSI với chân D7, và chân RST với chân D3. (Cổng SDA hay cổng RX của Rfid Rc522 Interfacing có thể cắm vào chân D4 tức TXD1 hoặc D6 tức TXD2 của mạch esp8266, cổng RST hay reset có thể cắm vào bất kỳ chân digital nào khác).

Với động cơ servo nối dây màu cam với 1 cổng digital bất kỳ ở đây là D8.

- Nguyên lý hoạt động

Hệ thống liên tục quét cảm biến Rfid Rc522 Interfacing, khi có tính hiệu từ, tính hiệu sẽ được so sánh với các phần tử trong danh sách đã đăng ký thông qua câu điều kiện if or, nếu khớp, hệ thống tiếp tục so sánh với biến Sum, nếu biến Sum nhỏ hơn 3( tức số lượng xe tối đa trong bãi) động cơ servo nhận được tín hiệu quay gốc 90 độ (mở cửa ra để xe vào), sau đó đóng lại sau 1 giây. Dãy số ID của thẻ từ sẽ được truyền dưới dạng tín hiệu cho hệ thống quản lý cổng sau thông qua sever Blynk. Đồng thời câu nói :” Please come in” cũng được phát thông qua hệ thống quản lý âm thanh sau khi nhận được tín hiệu thông qua Blynk. Ngược lại nếu số lượng trong bãi đã đầy, câu nói “sorry the parking slot is full” sẽ được phát lên nhằm thông báo cho khách hàng.

- Kỹ thuật (technical)

Đầu tiên hàm PICC\_IsNewCardPresent() kiểm tra xem có thẻ mới nào được đặt trên đầu đọc thẻ không. Nếu không có thẻ mới, chương trình kết thúc tại đây (lệnh return), và không có thêm hành động nào được thực hiện. Sau đó hàm PICC\_ReadCardSerial() được gọi để đọc thông tin chi tiết từ thẻ, chẳng hạn như UID (Unique Identifier). Nếu không đọc được thông tin từ thẻ, chương trình kết thúc tại đây và không có thêm hành động nào được thực hiện. Cuối cùng sử dụng vòng lặp for để duyệt qua 4 byte của UID của thẻ. Các byte này sau đó được ghép lại để tạo ra một giá trị duy nhất, được lưu vào biến tag.

A screenshot of a computer code

Description automatically generated

Chương trình kiểm tra giá trị của thẻ RFID, được lưu trong biến `tag`, để xác định quyền truy cập. Nếu giá trị của `tag` trùng khớp với các giá trị được quy định trước, hệ thống sẽ thực hiện việc lấy dữ liệu của `signal4` thông qua `Blynk.syncVirtual(V4)`. Sau đó, chương trình kiểm tra sự thay đổi của dữ liệu, giá trị của `sum` và `current\_sum` được cập nhật dựa trên dữ liệu từ Blynk Server. Tiếp theo, chương trình kiểm tra giá trị của `sum` để xác định hành động tiếp theo. Nếu `sum` nhỏ hơn 3, hệ thống thực hiện loạt hành động, bao gồm xuất âm thanh "Please come in", phát ra âm thanh “tít” từ còi, điều khiển servo motor để mở cổng và cập nhật trạng thái trên Blynk Server dựa trên giá trị của `tag` (gửi ID của thẻ để lưu vào danh sách xe đang trong bãi). Nếu `sum` không đạt điều kiện, hệ thống xuất âm thanh "Sorry, the parking slot is full". Ngoài ra nếu thẻ không nằm trong danh sách cổng sẽ không được mở.

A screenshot of a computer program

Description automatically generated

c) Hệ thống quản lý cổng ra

- Cấu tạo: gồm 1 mạch esp8266, 1 mạch RFID NFC 13.56MHz RC522, 4 thẻ từ đã được đăng ký trong danh sách, 1 thẻ từ ngoài, 1 động cơ servo.

- Sơ đồ mạch điện, đi dây

A circuit board with wires and wires

Description automatically generated

Nối cổng gnd mạch esp8266 với cổng gnd của Rfid Rc522 Interfacing, và cổng gnd của động cơ servo

Nối cổng 5v mạch esp8266 với cổng vcc của Rfid Rc522 Interfacing, và cổng vcc của động cơ servo

Với Rfid Rc522 Interfacing, nối cổng SDA với chân D4, cổng SCK với chân D5, cổng MISO với chân D6, MOSI với chân D7, và chân RST với chân D3. (Cổng SDA hay cổng RX của Rfid Rc522 Interfacing có thể cắm vào chân D4 tức TXD1 hoặc D6 tức TXD2 của mạch esp8266, cổng RST hay reset có thể cắm vào bất kỳ chân digital nào khác).

Với động cơ servo nối dây màu cam với 1 cổng digital bất kỳ ở đây là D8.

- Nguyên lý hoạt động

Đầu tiên hệ thông sẽ liên tục quét dữ liệu từ Blynk sever nếu có dữ liệu từ hệ thống quản lý cổng vào, dãy số ID Rfid sẽ được lưu vào danh sách xe trong bãi. Đồng thời hệ thống liên tục quét cảm biến Rfid Rc522 Interfacing, khi có tính hiệu từ, tính hiệu sẽ được so sánh với các phần tử trong danh sách xe trong bãi, nếu khớp động cơ servo nhận được tín hiệu quay gốc 90 độ (mở cửa ra để xe vào), sau đó đóng lại sau 1 giây. Sau đó dãy số ID của thẻ từ sẽ được xóa khỏi danh sách. Đồng thời câu nói :” Please come Out” cũng được phát thông qua hệ thống quản lý âm thanh sau khi nhận được tín hiệu thông qua Blynk. Nếu thẻ không khớp động cơ servo không quay, cổng không mở.

- Kỹ thuật (technical)

Đầu tiên tạo ra biến curentSignal = 0, sau đó liên tục kiểm tra sự thay đổi giữa giá trị hiện tại và giá trị mới của signal5. Nếu có sự thay đổi, chương trình kiểm tra giá trị của signal5 và thêm một chuỗi ID thẻ vào mảng carID tương ứng với giá trị của signal5. Ví dụ nếu signal5 bằng 1 thì ID thẻ sẽ là 124142183137, và tương tự sẽ là 3100127247 với signal5 bằng 2. Các giá trị này tương ứng với một chuỗi cụ thể được thêm vào mảng carID. Sau khi thực hiện các bước kiểm tra và thêm chuỗi, giá trị hiện tại của signal5 được ghi nhớ để so sánh ở lần kiểm tra tiếp theo.

A screenshot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

Đồng thời hàm PICC\_IsNewCardPresent() kiểm tra xem có thẻ mới nào được đặt trên đầu đọc thẻ không. Nếu không có thẻ mới, chương trình kết thúc tại đây (lệnh return), và không có thêm hành động nào được thực hiện. Sau đó hàm PICC\_ReadCardSerial() được gọi để đọc thông tin chi tiết từ thẻ, chẳng hạn như UID (Unique Identifier). Nếu không đọc được thông tin từ thẻ, chương trình kết thúc tại đây và không có thêm hành động nào được thực hiện. Cuối cùng sử dụng vòng lặp for để duyệt qua 4 byte của UID của thẻ. Các byte này sau đó được ghép lại để tạo ra một giá trị duy nhất, được lưu vào biến tag.

A screenshot of a computer code

Description automatically generated

Sau đó vòng lập for được tạo ra để duyệt các phần tử trong mảng, so sánh chúng với biến tag( tức ID thẻ từ đã được quẹt), nếu ID có trong mảng CarID động cơ sevor sẽ được quay, cổng sẽ mở sau đó phần từ [i] của mảng sẽ được thay thế bằng “” tức không có gì để phục vụ cho mục đích bảo mật ( trách trường hợp 1 thẻ quẹt 2 lần). Kế bên đó tiếng “tit” từ còi và âm thanh “ please come out “ cũng sẽ đồng thời được phát ra.

A screenshot of a computer program

Description automatically generated

Top of Formd) Hệ thống quản lý âm thanh

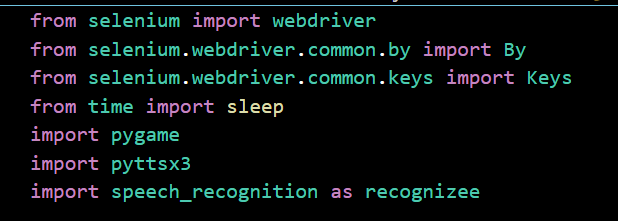
Hệ thống quản lý âm thanh được viết bằng ngôn ngữ Python dưới dạng 1 ứng dụng, nên sẽ không có sơ đồ mạch điện, và cấu tạo. Vì nhiều khó khăn khác nhau (cái sẽ được trình bày ở phần sau) nhóm chúng tôi quyết định sử dụng python để quản lý âm thanh đồng thời quản lý và vận hành hệ thống bằng Python, chương trình sẽ tích hợp của nhiều công nghệ để tạo ra một công cụ tự động hóa đa chức năng. Đoạn mã sử dụng Selenium để web scraping, Pygame để phát âm thanh, pyttsx3 để tổng hợp văn bản thành tiếng nói và speech\_recognition để nhập giọng nói. Các chức năng chính bao gồm tương tác dựa trên giọng nói, trích xuất dữ liệu web và phản hồi âm thanh động.

- Nguyên lý hoạt động

Python tạo nên 1 sever thông qua tài khoảng Blynk từ đó giúp các mạch esp8266 có thể kết nối, gửi tín hiệu, và liên kết với nhau. Sau đó hệ thống liên tục lấy dữ liệu từ websever thông qua Blynk về, phân tích và sử lý để đưa ra âm thanh phù hợp.

- Kỹ thuật (technical)

Chương trình sử dụng modules webdriver từ thư viện Selenium, modules chính trong việc điều khiển trình duyệt web tự động hóa web,. Kế bên đó để phục vụ trong việc phát âm thanh chúng tôi quyết định sử dụng thư viện pygame. Để phục vụ cho việc phát triển và mở rộng chức năng sau này, chương trình sử thư viện pyttsx3 và speech\_reconition, cho mục đích tạo nên 1 chương trình tích hợp trí tuệ nhân tạo có thể giao tiếp với con người.



Nhận Dạng Giọng Nói và Phản Hồi: Hàm Q\_A sử dụng thư viện speech\_recognition để bắt giọng nói. Nó nhận diện các từ khóa nhất định như "name" hoặc "we" và phản hồi tương ứng sử dụng pyttsx3 để tổng hợp văn bản thành tiếng nói.

A screen shot of a computer code

Description automatically generated

Tự động hóa Web: Chương trình sử dụng WebDriver của Selenium (cụ thể cho Chrome) để di chuyển đến trang web cụ thể ở đây là “https://blynk.cloud/dashboard/login”. Sau đó, nó tự động hóa quá trình đăng nhập bằng cách điền thông tin vào các trường tên đăng nhập và mật khẩu theo tên và mật khẩu đã được cung cấp trước đó, mô phỏng các phím bấm và đợi tải trang.

A screenshot of a computer screen

Description automatically generated

Web Scraping và Phản Hồi Âm Thanh Động: Sau đó, một vòng lặp vô hạn được tạo ra, chương trình liên tục trích xuất dữ liệu từ một XPath cụ thể trên trang web(tức tín hiệu được gửi lên từ hệ thống cổng vào, ra). Dữ liệu được trích xuất được so sánh với dữ liệu trước đó (curent\_data) để tránh hiện tượng nhiễu do vòng lập luôn lấy dữ liệu từ Blynk, nếu phát hiện sự thay đổi, nó tự động chọn và phát một tệp MP3 dựa trên dữ liệu đã cập nhật. Vd nếu tín hiệu là 5 âm thanh “Please come in” sẽ được phát ra. Nếu không có tín hiệu sẽ có bài nhạc nền được phát ra.

A screenshot of a computer program

Description automatically generated

Result & Discussion:

Result: Sau tất cả, nhóm chúng tôi mang lại 1 bãi xe có tính đột phá mang tính ứng dụng cao trong đời sống nhằm giảm thiểu sừ phiền nhiễu đến từ việc thiếu quản lý số lượng xe mang lại.



Discussion:

Tiện ích

Bãi gửi xe thông minh giúp theo dõi và quản lý hiệu suất sử dụng không gian đỗ xe, giảm tình trạng nhầm lẫn và lãng phí không gian. Bên cạnh đó, với công nghệ IoT, người lái xe có thể dễ dàng tìm kiếm và đặt chỗ đỗ xe trước qua ứng dụng di động, tiết kiệm thời gian và năng lượng. Về mặt kỹ thuật, hệ thống này tạo ra dữ liệu về lưu lượng xe và mô hình sử dụng không gian đỗ xe, cung cấp thông tin hữu ích cho quy hoạch đô thị và quản lý giao thông. Tăng cường an ninh: Các bãi gửi xe thông minh thường tích hợp các hệ thống an ninh như camera giám sát, giúp nâng cao mức độ an toàn cho phương tiện và người sử dụng.

Hạn chế:

Đầu tiên là Chi phí Vận Hành và Bảo Dưỡng, Ngoài chi phí đầu tư ban đầu, cần xem xét chi phí vận hành và bảo dưỡng trong thời gian dài. Việc này có thể đặt ra thách thức nếu không có kế hoạch bảo trì đặc biệt. Tuy nhiên nhóm chúng em đã tối đa hóa hiệu năng, cũng như chi phí sản xuất nhằm mục địch giảm giá thành sản phẩm đầu vào. Bên cạn đó là Hạn chế Cơ sở Hạ tầng Sự khác biệt về cơ sở hạ tầng giữa các vùng miền có thể là một thách thức, đặc biệt là ở những nơi có cơ sở hạ tầng kém. Đồng thời thách thức quản lý tăng cường dân số, đặc biệt là khi hệ thống bãi gửi xe cần mở rộng để đáp ứng nhu cầu ngày càng tăng cũng là 1 vấn đề cần phải đối mặt Thống Bãi Gửi Xe Cần Mở Rộng Để Đáp Ứng Nhu Cầu Ngày Càng Tăng Cũng Là 1 Vấn Đề Cần Phải Đối Mặt

Top of Form

Những khó khăn trong quá trình xây dựng hệ thống:

Đầu tiên là quá trình lên ý tưởng, , chúng tôi tiến hành phân chia hệ thống thành từng phần nhỏ và xác định những nguyên liệu cần thiết để xây dựng dự án. Trong những ngày đầu tiên của dự án, việc tưởng tượng ra sản phẩm cuối cùng và phân chia nó thành các phần nhỏ luôn là một thách thức đối với nhóm của chúng tôi.

Để giải quyết vấn đề này, chúng tôi quyết định tiếp cận từng phần một, theo phong cách làm từng bước một và chỉ sửa đến khi đạt được kết quả mong muốn. Sau khi chia sản phẩm thành 4 phần, chúng tôi tiếp tục đối mặt với thách thức liên quan đến giao tiếp giữa cổng ra và cổng vào. Cụ thể, đây là quá trình truyền mã số ID của một chiếc xe vào bãi đỗ để cổng ra có thể xử lý khi xe được đưa ra.

Sau khi chia sản phẩm thành  4 phần, thì  vấn đề kế tiếp là giao tiếp giữa cổng ra và cổng vào, đặc biệt là truyền mã số ID của xe vào bãi đỗ để cổng ra có thể xử lý khi đưa xe ra. Ban đầu, chúng tôi quyết định sử dụng giao tiếp phần cứng, tức là kết nối hai mạch esp8266 thông qua việc tạo ra hai cổng ảo RX và TX trên cả hai mạch esp8266.

Tuy nhiên, trong quá trình thử nghiệm, chúng tôi phát hiện rằng việc cắm cảm biến RFID vào mạch esp8266 gây nhiễu tần số quét của các cổng ảo, làm cho hệ thống phụ thuộc vào dây truyền tín hiệu và dẫn đến sự không ổn định trong quá trình truyền dữ liệu giữa hai hệ thống.

Để giải quyết vấn đề này, chúng tôi quyết định sử dụng biến ảo của Blynk để lưu trữ dữ liệu đầu ra và đầu vào của hệ thống cổng ra và cổng vào. Chúng tôi chuyển đổi các tín hiệu thành số nguyên và gửi chúng qua Blynk. Điều này giúp hệ thống nhận tín hiệu một cách dễ dàng và gần như ngay lập tức để thực hiện các hành động khác nhau, như lưu trữ dữ liệu vào cổng ra hoặc kích hoạt âm thanh tại hệ thống quản lý âm thanh.

Top of Form

Sau đó, chúng tôi phát hiện ra rằng độ delay của Blynk ảnh hưởng khá lớn đến hoạt động của hệ thống. Để khắc phục điều này, chúng tôi quyết định thường xuyên gửi dữ liệu hai lần hoặc double dữ liệu nhận về. Điều này được thực hiện để đảm bảo rằng không có trường hợp nào dữ liệu bị nhiễu xảy ra, đồng thời cải thiện độ ổn định của hệ thống.

Sau khi hoàn thành các bước trên, chúng tôi gặp vấn đề mới liên quan đến động cơ servo và lỏng dây. Một tuần sau khi sản phẩm được triển khai, chúng tôi nhận thấy động cơ servo có dấu hiệu lỏng ở các bánh răng, làm cho động cơ không hoạt động đúng cách. Để giải quyết vấn đề này, chúng tôi đã thử nghiệm nhiều phương pháp như vặn lỏng răng, thực hiện tác động vật lý để đặt các bánh răng về đúng vị trí, và nhiều cách khác nhau, nhưng không có kết quả khả quan.

Cuối cùng, một thành viên trong nhóm phát hiện ra rằng nguyên nhân của vấn đề là do sợi dây bị ép quá mức, dẫn đến nguồn điện không chạy mượt mà. Để khắc phục vấn đề này, chúng tôi quyết định đơn giản là mở rộng chỗ đặt servo để giảm áp lực lên dây và đảm bảo nguồn điện chạy một cách trơn tru.

Conclusion

Bên cạnh những hạn chế về mặt chi phí và tốn diện tích. Bãi gửi xe thông minh có thể quản lý và hộ trợ người dùng bằng các công nghệ tiến tiến của riêng nó, nhằm mục đích hướng tới 1 thế giới tự động hoá với nhiều chức năng khác nhau. Nếu trong tương lai công nghệ này được đưa vào vận hành và đồng bộ hoá thì người dùng có thể dễ dàng tìm được một nơi còn xe ở một khu vực nào đoá