

**KHOA ĐIỆN TỬ - VIỄN THÔNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

\*\*\*\*\*



**XÂY DỰNG HỆ THỐNG IOT  
CHO BÃI ĐỖ XE THÔNG MINH**

*Hà Nội, ngày 2 tháng 5 năm 2022*

**KHOA ĐIỆN TỬ - VIỆN THÔNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

\*\*\*\*\*

**XÂY DỰNG HỆ THỐNG IOT  
CHO BÃI ĐỖ XE THÔNG MINH**

**Members:** Lê Đức

Nguyễn Huy Tú

Ngô Lê Xuân Phúc

Lê Phấn Nam

Hoàng Văn Kiên

**Memtor:** Nguyễn Tuấn Anh

*Hà Nội, ngày 2 tháng 5 năm 2022*

## **LỜI CẢM ƠN**

Trước hết nhóm Fò lo ren ti nô xin cảm ơn Ban tổ chức đến từ FPT Software đã hợp tác với Khoa Điện tử - Viễn thông trường Đại học Công nghệ - ĐHQGHN tạo ra một sân chơi bổ ích cho sinh viên, giúp chúng em được tiếp cận với IoT cũng như các công nghệ mới đến từ Silabs.

Nhóm em cũng xin chân thành cảm ơn sự giúp đỡ tận tình, chỉ dạy của anh mentor Nguyễn Tuấn Anh , các anh/chị trong FSoft đã giúp đỡ nhóm em rất nhiều trong quá trình hoàn thiện sản phẩm.

## MỤC LỤC

|   |           |
|---|-----------|
| <b>LỜI CẢM ƠN .....</b>                                   | <b>3</b>  |
| <b>MỤC LỤC .....</b>                                      | <b>4</b>  |
| <b>LỜI MỞ ĐẦU .....</b>                                   | <b>6</b>  |
| <b>CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI .....</b>        | <b>7</b>  |
| 1.1. Đặt vấn đề .....                                     | 7         |
| 1.2. Mục tiêu sản phẩm.....                               | 8         |
| 1.2.a. An Toàn .....                                      | 9         |
| 1.2.b. Thuận lợi.....                                     | 9         |
| 1.2.c. Tiết kiệm.....                                     | 10        |
| <b>CHƯƠNG 2. HỆ THỐNG VI ĐIỀU KHIỂN VÀ NGOẠI VI .....</b> | <b>11</b> |
| 2.1. Vi điều khiển .....                                  | 11        |
| 2.1.a. BGM220 .....                                       | 11        |
| 2.1.b. Raspberry Pi .....                                 | 13        |
| 2.2. Các thiết bị ngoại vi.....                           | 15        |
| <b>CHƯƠNG 3. GATEWAYS .....</b>                           | <b>19</b> |
| 3.1. Giao tiếp UART giữa Pi với BGM220:.....              | 19        |
| 3.2. Giao tiếp I2C giữa Pi với Arduino Uno: .....         | 20        |
| <b>CHƯƠNG 4. THINGSPEAK .....</b>                         | <b>22</b> |
| 4.1. Giới thiệu .....                                     | 22        |
| 4.2. Các tính năng .....                                  | 22        |
| 4.3. Thingspeak hoạt động như thế nào .....               | 23        |
| 4.4. Áp dụng vào dự án thực tế.....                       | 23        |
| <b>CHƯƠNG 5. APP MOBILE.....</b>                          | <b>26</b> |
| 5.1. Tổng quan về App Mobile.....                         | 26        |
| 5.2. Chức năng của App .....                              | 26        |
| 5.2.a. Giao diện ban đầu .....                            | 27        |
| 5.2.b. Các bãi đỗ được tích hợp hệ thống IoT .....        | 28        |
| 5.2.c. Bên trong bãi đỗ .....                             | 29        |
| <b>CHƯƠNG 6. MÔ HÌNH THỰC TẾ .....</b>                    | <b>30</b> |
| 6.1. Phác hoạ 3D.....                                     | 30        |
| 6.2. Mô hình thực tế.....                                 | 30        |
| <b>CHƯƠNG 7. PHÁT TRIỂN SẢN PHẨM.....</b>                 | <b>32</b> |
| 7.1. Cấu trúc mạng lưới .....                             | 32        |
| 7.1.a. Mạng Star.....                                     | 32        |

|                                |    |
|--------------------------------|----|
| 7.1.b. Mạng Mesh .....         | 32 |
| 7.1.c. Phương án .....         | 33 |
| 7.2. Xử lý ảnh .....           | 33 |
| 7.2.a. <i>OPENCV</i> .....     | 33 |
| 7.2.b. <i>TESSERACT</i> .....  | 34 |
| 7.2.c. <i>Nhược điểm</i> ..... | 34 |
| 7.2.d. Hướng tiếp cận .....    | 35 |
| 7.3. App Mobile.....           | 37 |

## LỜI MỞ ĐẦU

Trong những năm gần đây với sự phát triển kinh tế xã hội của Việt Nam, lượng phương tiện giao thông đã tăng một cách nhanh chóng. Phương tiện cá nhân tăng lên, đòi hỏi diện tích đất dành cho bãi đỗ xe cũng phải tăng theo. Tuy nhiên, tại các thành phố lớn như Hà Nội, Hồ Chí Minh, việc đáp ứng yêu cầu quỹ đất này ngày càng tỏ ra không khả thi do giá trị đất đang tăng nhanh, và nhu cầu đất cho các mục đích quan trọng khác cũng đang thiếu.

Hiện nay tại các khu vực trung tâm thành phố lớn, số ô tô phần lớn dừng đỗ trên vỉa hè, lòng đường gây cản trở giao thông. Để giải quyết vấn đề và đi theo xu thế phát triển của thế giới, các bãi xe tự động với nhiều loại hình khác nhau được ra đời. Với hệ thống đỗ xe tự động khi được đưa vào sử dụng sẽ mang lại rất nhiều lợi ích cho xã hội, giảm thiểu vấn đề cản trở giao thông, cảnh quan thành phố và đặc biệt là tiết kiệm thời gian cho người sử dụng ô tô.

Trong dự án này, nhóm em đã thực hiện nghiên cứu và phát triển đề tài “Xây dựng hệ thống IoT cho bãi đỗ xe thông minh”

Do khả năng và kiến thức còn hạn chế nên trong quá trình làm sản phẩm không thể tránh khỏi sai sót. Nhóm em rất mong nhận được những ý kiến đóng góp quý báu của các thầy, cô và các anh chị.

Một lần nữa nhóm Fò lo ren ti nô xin chân thành cảm ơn!

*Hà Nội, ngày 02 tháng 05 năm 2022*

*Nhóm sinh viên thực hiện*

*Trưởng nhóm*

Lê Đức

# CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

## 1.1. Đặt vấn đề

Trong những năm gần đây với sự phát triển kinh tế xã hội của Việt Nam, lượng phương tiện giao thông đã tăng một cách nhanh chóng. Phương tiện cá nhân tăng lên, đòi hỏi diện tích đất dành cho bãi đỗ xe cũng phải tăng theo. Tuy nhiên, tại các thành phố lớn như Hà Nội, Hồ Chí Minh, Đà Nẵng, Cần Thơ,.. việc đáp ứng yêu cầu quỹ đất này ngày càng tỏ ra không khả thi do giá trị đất đang tăng nhanh, và nhu cầu đất cho các mục đích quan trọng khác cũng đang thiếu.

Hiện nay tại các khu vực trung tâm thành phố lớn, số ô tô phần lớn dừng đỗ trên vỉa hè, lòng đường gây cản trở giao thông (Hình 1.1).



Hình 1.1. Tình trạng dừng đỗ xe trên vỉa hè và lòng đường

Đồng thời, việc các chủ xe dừng đỗ trên vỉa hè, không đúng nơi quy định sẽ bị xử phạt theo quy định của bộ GTVT. Bên cạnh đó, hiện hay xuất hiện nhiều kẻ gian phá hoại, tạt sơn, làm xước xe thậm chí là lấy đi những phụ tùng của xe (Hình 1.2).



Hình 1.2. Xe bị tạt sơn và kẻ gian có hành động bẻ gương

Để giải quyết vấn đề chỗ đỗ xe trong đô thị, nhiều nước trên thế giới sử dụng hệ thống nhà đỗ xe nhiều tầng tự động, phổ biến như Nhật Bản, Hàn Quốc, Ấn Độ, Singapore, Trung Quốc, Mỹ và các nước Châu Âu. Tại các nước này đã có nhiều công ty chủ yếu kinh doanh bãi đỗ ô tô nhiều loại, trong đó hệ thống đỗ nhiều tầng tự động được sử dụng rất phổ biến. Các công ty sản xuất hệ thống đỗ xe tự động là các nhà chế tạo, không trực tiếp kinh doanh bãi đỗ xe mà chỉ cung cấp và lắp đặt thiết bị cho các nhà đầu tư. Ngoài ra, còn các hệ thống các công ty sản xuất các thiết bị phụ trợ như: hệ thống lấy vé tự động đọc thẻ, trả tiền tự động.

So với các bãi đỗ xe kiểu truyền thống, những lợi ích của một bãi đỗ xe tự động không chỉ là nhanh chóng và tiện lợi. Bạn sẽ không còn phải lo lắng chiếc xe của mình bị xô xát, va đụng, trầy xước bởi nơi đỗ xe được cách ly hoàn toàn với bên ngoài.

Hệ thống đỗ xe tự động là loại thiết bị mang tính kỹ thuật cao, áp dụng hòa hợp các nguyên lý chung của hệ thống thang máy & hệ thống xếp dỡ tự động và hệ thống lưu kho tự động mà hàng hóa là ô tô và có độ chính xác nhất định. Trong hệ thống này xe được lưu giữ ở các ô (Block parkings) dưới mặt đất hoặc trên cao. Để thực hiện việc lưu giữ này hệ thống sử dụng các thiết bị nâng chuyển.

## **1.2. Mục tiêu sản phẩm**

Theo thống kê của Công an Thành phố Hà Nội, trung bình trên địa bàn TP xảy ra 50-100 vụ mất xe máy trong đó tỉ lệ tại các bệnh viện, siêu thị, trung tâm thương mại, chợ... chiếm 30%. Nguyên nhân chủ yếu là do kẻ gian dùng thủ đoạn làm vé giả, tráo biển số hay người giữ xe bị mất vé.

Hiện nay trên địa bàn thành phố nạn kẹt xe luôn là đề tài nóng, có rất nhiều kênh như báo chí, truyền thanh, truyền hình đăng tải về vấn đề này, và có một kênh truyền thanh chuyên cung cấp thông tin những điểm nóng giao thông trên địa bàn thành phố để thông báo cho mọi người dân, lưu lượng xe di chuyển quá nhiều trên các nẻo đường phố là rất nhiều mà khả năng sức chứa của đường phố hiện nay là có hạn cho nên việc xuất hiện những điểm giữ xe là điều tất yếu để giảm nạn kẹt xe, nhưng do lượng xe quá nhiều mà những điểm giữ xe hiện nay chưa thực sự đáp ứng được những yêu cầu mà khách hàng mong muốn, nguyên nhân là do :



- Mức độ an toàn của những điểm gửi xe không làm khách hàng hoàn toàn an tâm
- Thái độ phục vụ của những nhân viên gửi xe đối với khách hàng chưa được tể nhị, cũng như mức độ chuyên nghiệp
- Thời gian gửi xe làm cho khách hàng khó chịu vì phải đợi đến lâu mới tới lượt của mình

Nhận thấy được điều này, nhóm em đã nảy ra ý tưởng làm “*Nhà gửi xe thông minh theo tầng kết hợp hệ thống IOT*”. Do tính chất của cuộc thi, thời gian không cho phép và kiến thức của các thành viên không cho phép, đa số là sinh viên năm 2 nên nhóm em đã tập trung tìm hiểu, nghiên cứu về “*Hệ thống IOT cho bãi đỗ xe thông minh*”.

Nhóm em muốn đặt ra 3 mục tiêu cần giải quyết, đáp ứng được những nhu cầu được cho là cấp bách hiện nay.

#### ***1.2.a. An Toàn***

Một trong những điều kiện tiên quyết là “Sự an toàn”. Khách hàng khi gửi xe, dù là bãi đỗ xe truyền thống hay các bãi đỗ xe kiểu mới thì sự ưu tiên ban đầu luôn là an toàn. Nhưng như đã đề cập ở trên, các nhà gửi xe truyền thống vẫn xảy ra trường hợp mất – hỏng xe do các dịch vụ chưa đảm bảo như:

- Không có camera, gây ra không thể quan sát hết được nhà xe, tạo điều kiện cho kẻ xấu gây hư hại cho xe
- Mặc dù có sự xuất hiện của vé xe nhưng chỉ dừng lại ở vé xe bằng giấy, có mã số riêng biệt và ứng với các xe bằng phần viền lên đầu/yên xe gây ra hiện trạng dễ dàng thay đổi dễ dàng

Vì thế, nhóm em đặt ra mục tiêu giải quyết được vấn đề an ninh trong bãi đỗ xe, đặt việc bảo toàn tài sản của khách hàng, quản lí xe ra/ vào bằng cơ sở dữ liệu giúp tăng sự tin tưởng của khách hàng với nhà đỗ xe.

#### ***1.2.b. Thuận lợi***

Có một yếu tố không nhỏ ảnh hưởng tới việc xe đỗ không đúng nơi quy định đó là việc không thuận tiện khi gửi xe. Trái với việc quá tải ở các bãi gửi xe ở các khu trung tâm thương mại, bệnh viện, trường học. Các bãi gửi xe ở toà nhà, bãi gửi xe khu dân cư

lại tạo ra cảm giác không thuận lợi. Một phần là do bãi đỗ xe xa nơi cần đến, khoảng cách có thể lên tới 2-3km, một phần là do các bãi gửi xe tư nhân này chỉ nhận gửi xe theo tháng hoặc nằm sâu trong các toà nhà. Vì vậy, nếu khách hàng không gửi xe thường xuyên sẽ khó có thể chọn những nhà xe này.

Với dự án của chúng em, chúng em hướng tới một hệ thống nhiều bãi đỗ xe, với những khách hàng gửi xe theo ngày bãi đỗ xe vẫn luôn phục vụ. Đồng thời với những khách hàng gửi xe theo tháng thì sẽ có thể gửi xe trên toàn hệ thống bãi đỗ xe, tạo hiệu ứng tích cực và thuận tiện cho khách hàng.

### ***1.2.c. Tiết kiệm***

Mục tiêu cuối cùng nhóm em muốn hướng tới là tiết kiệm. Tiết kiệm không chỉ về tài chính mà còn về thời gian. Nhóm em hướng tới bãi đỗ xe thông minh tự động hoàn toàn, giải quyết được vấn đề thái độ phục vụ của những nhân viên gửi xe đối với khách hàng chưa được tế nhị, cũng như mức độ chuyên nghiệp chưa được tốt.

Đồng thời giảm thiểu được chi phí thuê nhân công, tiết kiệm cho chính khách hàng. Không những thế bãi đỗ xe thông minh sẽ sử dụng các công nghệ máy quét, camera xử lý biển số xe giúp cho việc quản lý xe ra vào bãi được tối ưu nhất, dự kiến trung bình mỗi xe chỉ mất 5-10 giây cho mỗi lượt gửi xe, giúp cho việc gửi xe tiết kiệm thời gian cho khách hàng.

## CHƯƠNG 2. HỆ THỐNG VI ĐIỀU KHIỂN VÀ NGOẠI VI

Với một hệ thống IoT chúng sẽ bao gồm 4 thành phần chính đó là thiết bị (Things), trạm kết nối (Gateways), hạ tầng mạng (Network and Cloud) và bộ phân tích và xử lý dữ liệu (Services-creation and Solution Layers).

### 2.1. Vi điều khiển

#### 2.1.a. BGM220

Máy móc, thiết bị, cảm biến hay còn gọi là “things” là thứ đầu tiên cần có trong một hệ thống IoT, chúng phải có tên hoặc địa chỉ IP duy nhất. Hệ thống IoT sẽ hỗ trợ kết nối giữa các “things” và các kết nối được thiết lập dựa trên định danh IP của things. Được sự hỗ trợ từ Ban tổ chức FPT Software (FSoft) và Silicon Labs nhóm chúng em đã sử dụng BGM220 Bluetooth Module Explorer Kit vì giá thành rẻ và có thể tích hợp công nghệ Bluetooth Low Energy (BLE).

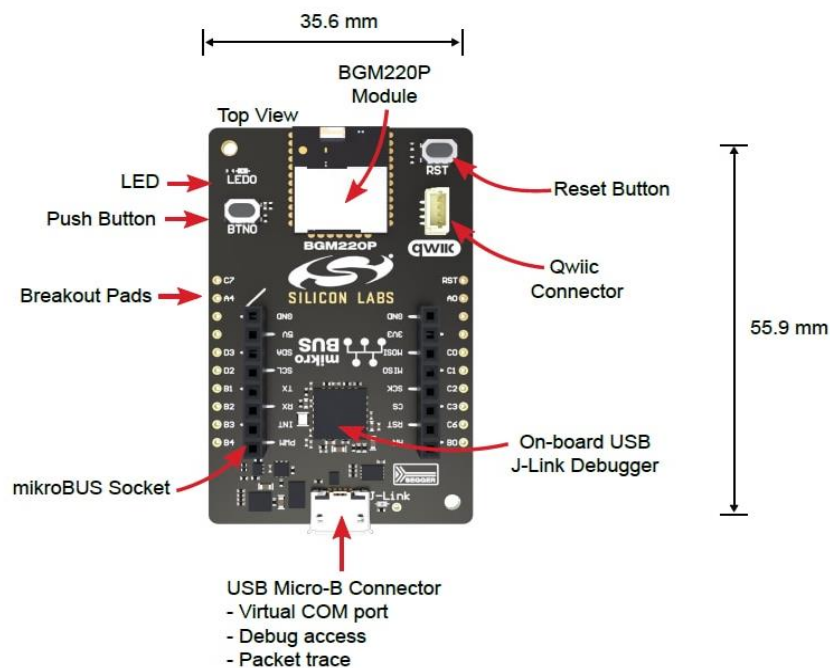


Hình 2.1. BGM220 Bluetooth Module Explorer Kit



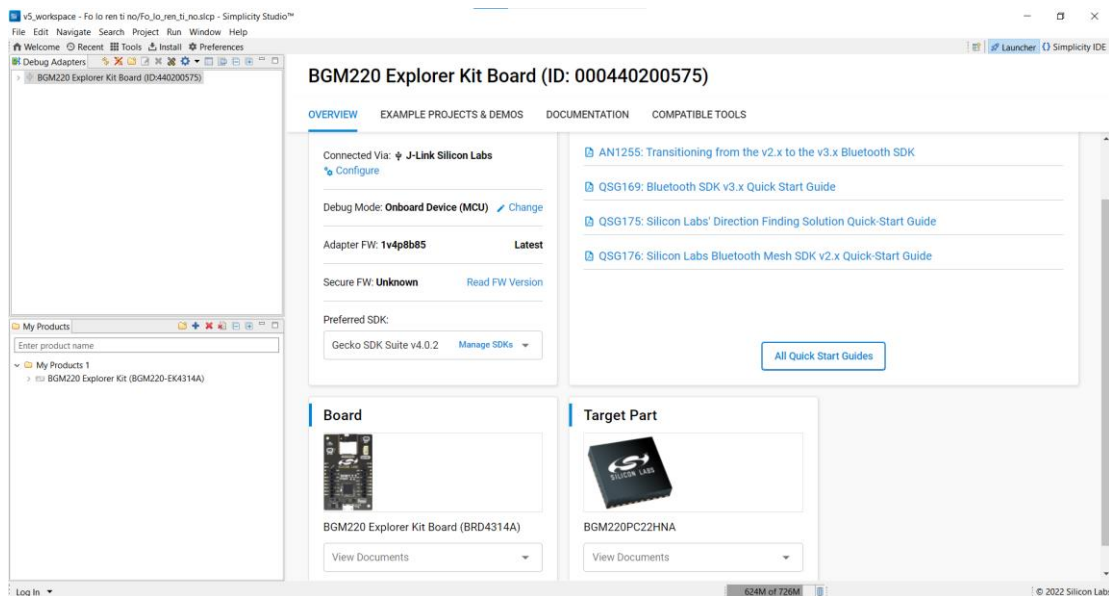
Hình 2.2. Silicon Labs

BGM220 Explorer Kit được thiết kế để khách hàng có thể tự tạo ra các thiết bị IoT cho mình với mô-đun không dây Gecko Bluetooth®. Việc lập trình BGM220 Explorer Kit được thực hiện dễ dàng bằng cáp USB Micro-B và debugger J-Link trên bo mạch. Ngoài các chân GPIO, BGM220 Explorer Kit còn hỗ trợ các giao thức cơ bản của vi điều khiển như I2C, SPI và UART.



Hình 2.3. Bố cục phần cứng BGM220 Explorer Kit

Lập trình cho BGM220 Explorer Kit thì không thể thiếu môi trường phát triển tích hợp (IDE) Simplicity Studio 5 được xây dựng bởi Silicon Labs. Simplicity Studio 5 có những công cụ giao diện người dùng hiện đại khá giống với web.



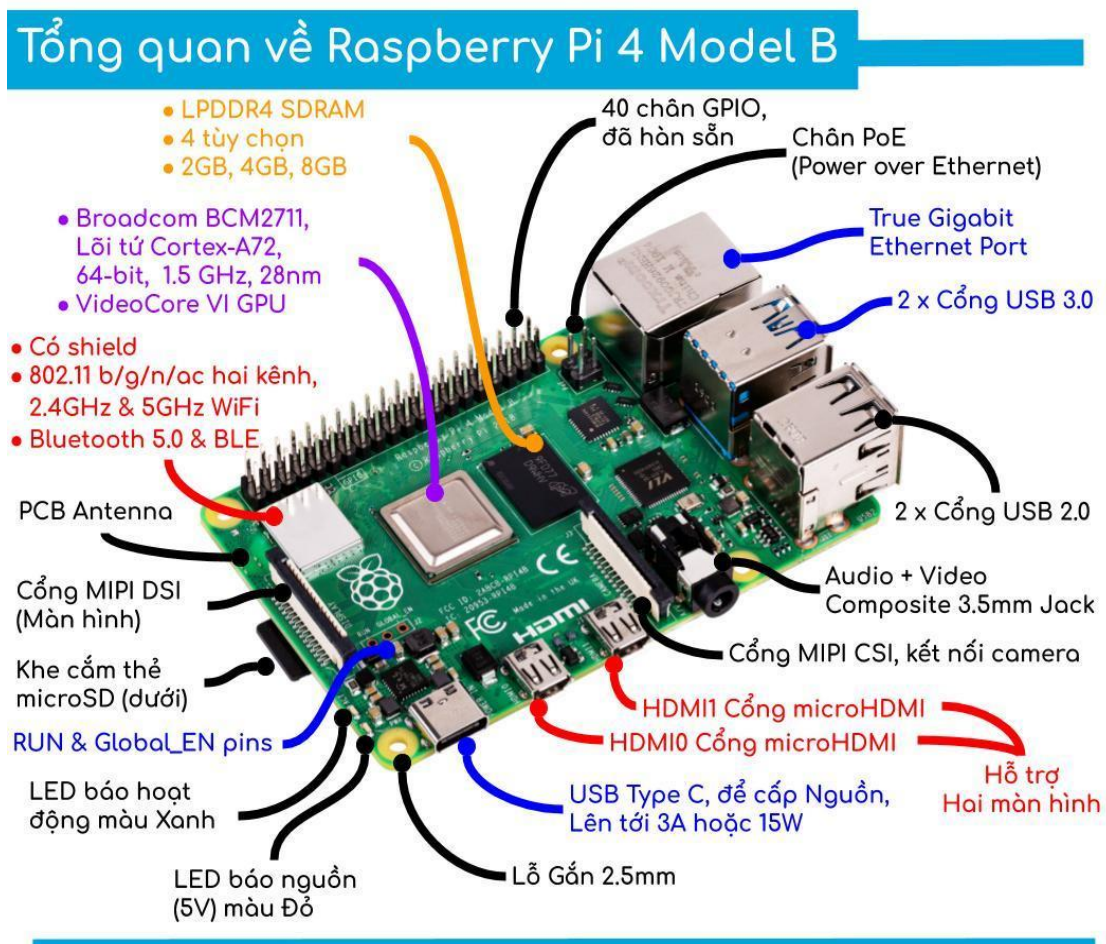
Hình 2.4. Giao diện Simplicity Studio 5 sau khi được kết nối với bo mạch BGM220

### 2.1.b. Raspberry Pi

#### - Tổng quan về Raspberry Pi 4

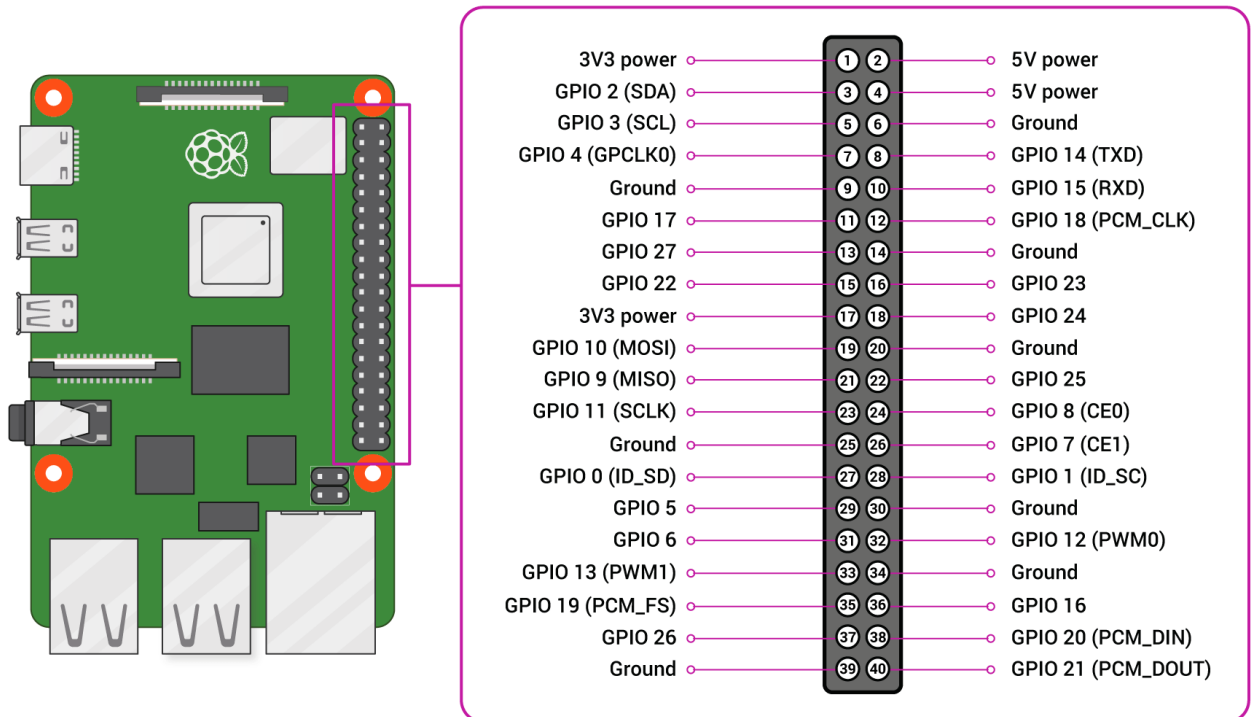
Raspberry Pi là một máy tính rất nhỏ gọn, kích thước hai cạnh như bằng khoảng một cái thẻ ATM và chạy hệ điều hành Linux. Raspberry Pi được phát triển bởi Raspberry Pi Foundation - một tổ chức phi lợi nhuận.

Raspberry Pi 4 Model B 4GB được ra mắt vào ngày 24 tháng 6, 2019. Raspberry Pi 4 Model B là sản phẩm mới nhất trong dòng máy tính Raspberry Pi phổ biến. Nó được gia tăng tốc độ vi xử lý, hiệu suất đa phương tiện, bộ nhớ và kết nối so với Raspberry Pi 3 Model B + của thế hệ trước trong khi vẫn giữ được khả năng tương thích ngược và mức tiêu thụ điện năng tương tự.



Hình 2.5. Tổng quan về Raspberry Pi 4B

- Sơ đồ chân GPIO:



Hình 2.6. Sơ đồ chân GPIO của Raspberry Pi

- Cài đặt hệ điều hành Raspberry Pi OS

+ Cài đặt Raspberry Pi Imager

Raspberry Pi đã phát triển một công cụ ghi thẻ SD hoạt động trên Mac OS, Ubuntu 18.04 và Windows, đây là sự lựa chọn dễ dàng cho người dùng vì mọi thứ sẽ tự động từ việc tải image và cài đặt vào thẻ nhớ SD.

Tải xuống phiên bản Raspberry Pi Imager mới nhất tại [“Download Pi Imager”](#) và cài đặt.

Tùy vào loại hệ điều hành khác nhau mà lựa chọn phiên bản thích hợp cho máy tính.

+ Raspberry Pi OS có 3 loại hệ điều hành.

Ở đây mình sử dụng Raspberry Pi OS (32-bit) Lite vì dung lượng nhỏ gọn mà vẫn có thể thực hiện đầy đủ những tác vụ cần thiết. Kết nối thẻ SD vào đầu đọc thẻ. Mở Raspberry Pi Imager và chọn hệ điều hành RASPBERRY PI OS LITE (32-BIT) từ danh sách.

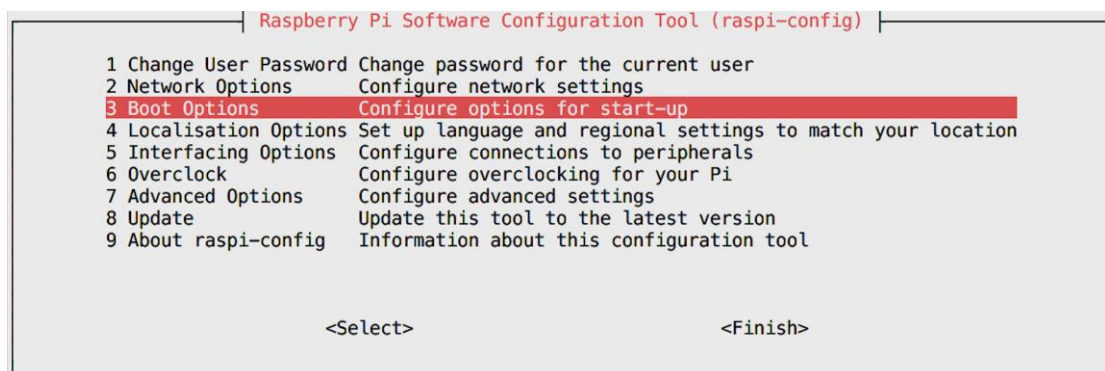
Chọn thẻ SD muốn ghi image vào. Nhấp vào WRITE để bắt đầu ghi dữ liệu vào thẻ nhớ SD.

+ Boot hệ điều hành mới

Lắp thẻ SD vào Raspberry Pi và khởi động, đối với hệ điều hành Raspberry Pi chính thức, tên người dùng mặc định là pi, mật khẩu raspberry.

Thiết lập cơ bản: thực thi câu lệnh:

```
sudo raspi-config
```



+ Wifi: Kết nối mạng cho pi

+ SSH: Cấu hình cho phép SSH vào Raspberry Pi, để cập nhật hoặc khởi động lại Pi từ xa, có thể sử dụng SSH sau khi bật SSH server thông qua raspi-conf.

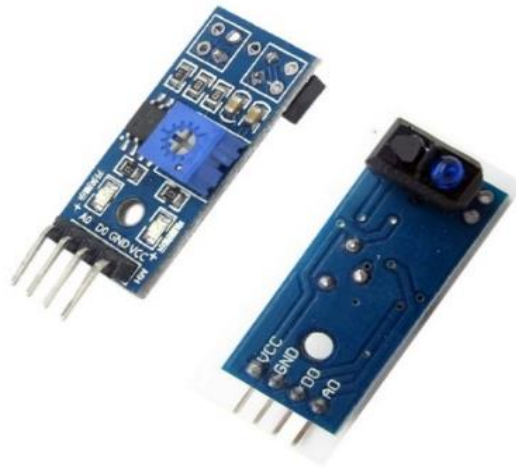
+ Serial: giao tiếp giữa pi với bgm

+ I2C: giao tiếp giữa pi với Arduino uno

## 2.2. Các thiết bị ngoại vi

Để có được sự chính xác cao, nhóm đã sử dụng cảm biến dò line (cảm biến hồng ngoại) để xác định xe đã vào vị trí đỗ.





Hình 2.5. Cảm biến dò line TCRT5000

Hệ thống vào – ra tích hợp quét thẻ RFID để quản lý xe ra vào nói chung cũng như là người dùng có thể quản lý phương tiện cá nhân nói riêng. Barrier sẽ đóng mở tự động thông qua sự điều khiển của Động cơ Servo SG90. Để thuận tiện cho việc tích hợp RFID và điều khiển servo trong thời gian ngắn nhất, Arduino sẽ là lựa chọn tối ưu hơn BGM220 nhờ cộng đồng lớn và nhiều thư viện hỗ trợ. Cụ thể tại hệ thống vào – ra của bãi đỗ xe, nhóm sử dụng Arduino UNO nhằm tiết kiệm chi phí.



Hình 2.6. Arduino UNO



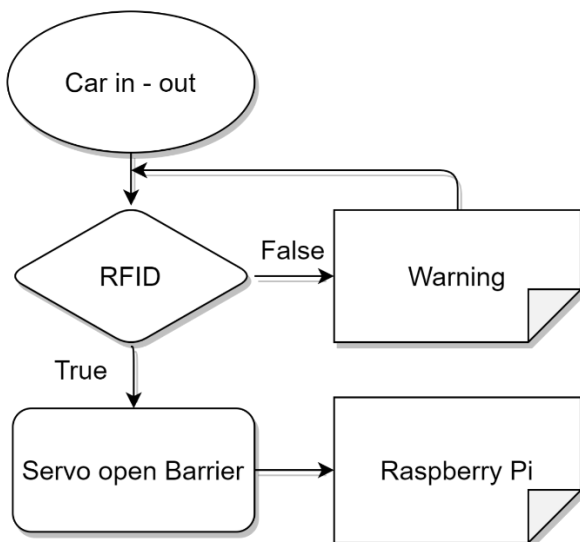
Hình 2.7. Mô đun RFID RC522



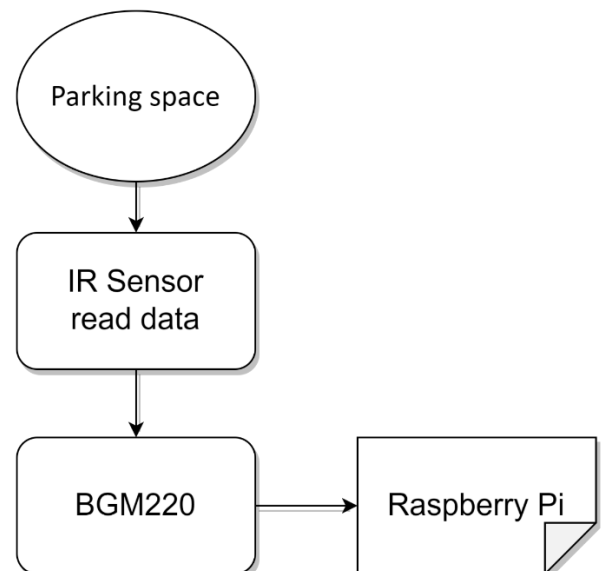


Hình 2.8. Động cơ Servo SG90

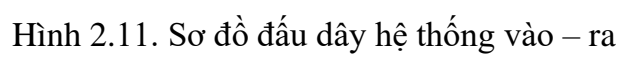
Tại đây, người điều khiển xe sẽ quét RFID trước khi vào – ra, hệ thống sẽ kiểm tra xem thẻ có hợp lệ hay không, nếu không sẽ có loa và đèn cảnh báo, nếu hợp lệ barrier sẽ mở cho đến khi người điều khiển phương tiện đi qua và việc này cũng sẽ được xác định bởi cảm biến dò line nói trên. Ngoài ra người quản lý có thể kiểm soát được đang có bao nhiêu phương tiện đã vào bãi mà chưa đỗ.



Hình 2.9. Sơ đồ thuật toán hệ thống quản lý vào - ra



Hình 2.10. Sơ đồ thuật toán xác định xe đỗ



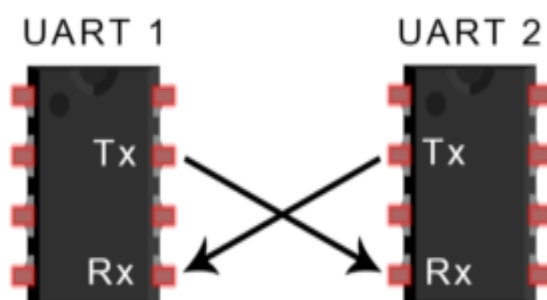
Hình 2.11. Sơ đồ đầu dây hệ thống vào – ra

## CHƯƠNG 3. GATEWAYS

### 3.1. Giao tiếp UART giữa Pi với BGM220:

UART là viết tắt của Universal Asynchronous Receiver / Transmitter. Đây là chuẩn giao tiếp nối tiếp với sự hỗ trợ của phần cứng (hardware). UART hoàn toàn khác biệt với chuẩn giao tiếp SPI hoặc I2C, những chuẩn này chỉ đơn thuần là giao tiếp phần mềm. Mục đích chính của UART là truyền và nhận dữ liệu nối tiếp.

Trong giao tiếp UART, hai UART giao tiếp trực tiếp với nhau. UART truyền chuyển đổi dữ liệu song song từ một thiết bị điều khiển như CPU thành dạng nối tiếp. Sau đó truyền nó nối tiếp đến UART nhận, rồi chuyển đổi dữ liệu nối tiếp trở lại thành dữ liệu song song cho thiết bị nhận. Chỉ cần hai dây để truyền dữ liệu giữa hai UART. Dữ liệu truyền từ chân Tx của UART truyền (Transmitter) đến chân Rx của UART nhận (Receiver):



Hình 3.1. Sơ đồ kết nối UART

Cả hai UART cũng phải được cấu hình để truyền và nhận cùng một cấu trúc gói dữ liệu. Nghĩa là khi thiết lập ở vi điều khiển, các cấu hình tốc độ baud phải giống nhau.

#### - Cách thực hiện:

Cắm dây theo đúng quy tắc, thiết lập Serial trên Pi, xác định cổng truyền và baud truyền trên BGM220

Thực hiện kết nối BGM với cảm biến hồng ngoại để xác định xe có ở vị trí hay không

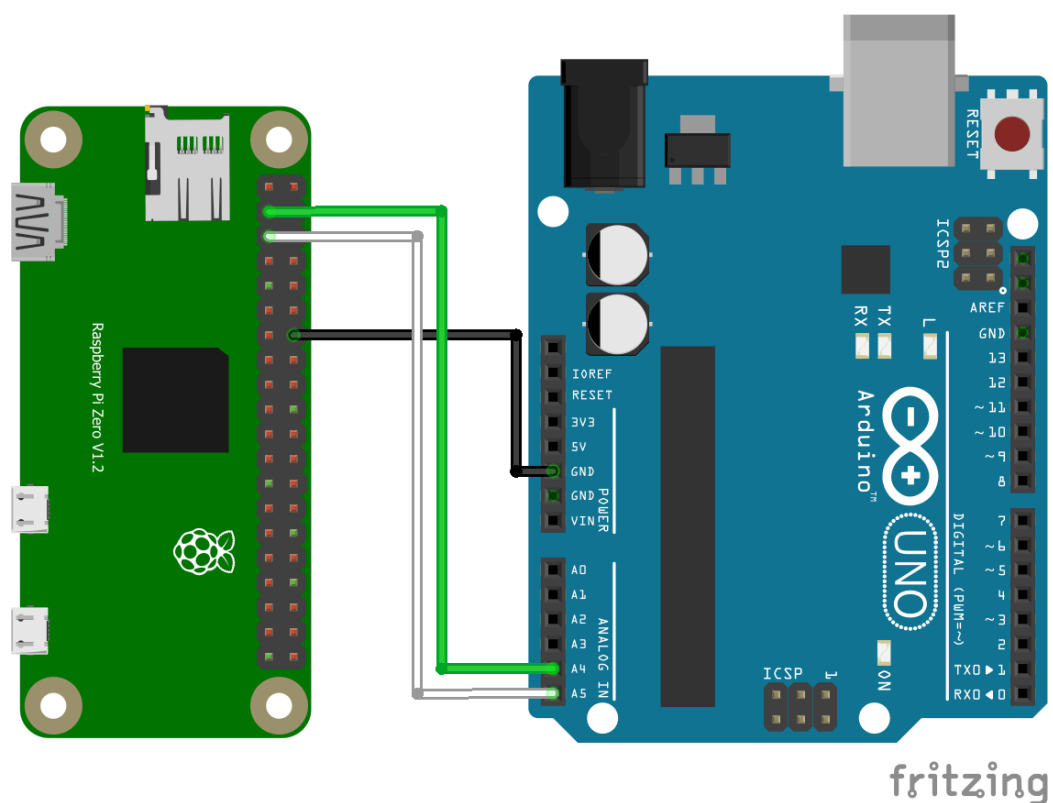
Trên Pi, gửi dữ liệu liên tục đến BGM, khi BGM nhận thấy dữ liệu được gửi đến thì lấy dữ liệu từ cảm biến in qua Serial. Pi thực hiện việc nhận và xử lý dữ liệu để gửi lên Thingspeak.

### 3.2. Giao tiếp I2C giữa Pi với Arduino Uno:

Do đã sử dụng giao thức UART để giao tiếp Pi với BGM nên để truyền nhận dữ liệu giữa Pi với Arduino ta lựa chọn giao tiếp I2C bởi vì nó khá đơn giản.

I2C sử dụng hai dòng - SDA (dữ liệu) và SCL (đồng hồ) - ngoài GND (mặt đất). SDA là hai chiều, vì vậy chúng tôi cần đảm bảo cách này hay cách khác đang gửi dữ liệu (chủ hoặc nô lệ). Với I2C, chỉ chủ mới có thể bắt đầu liên lạc. Các chủ cũng kiểm soát tín hiệu đồng hồ.

Kết nối:



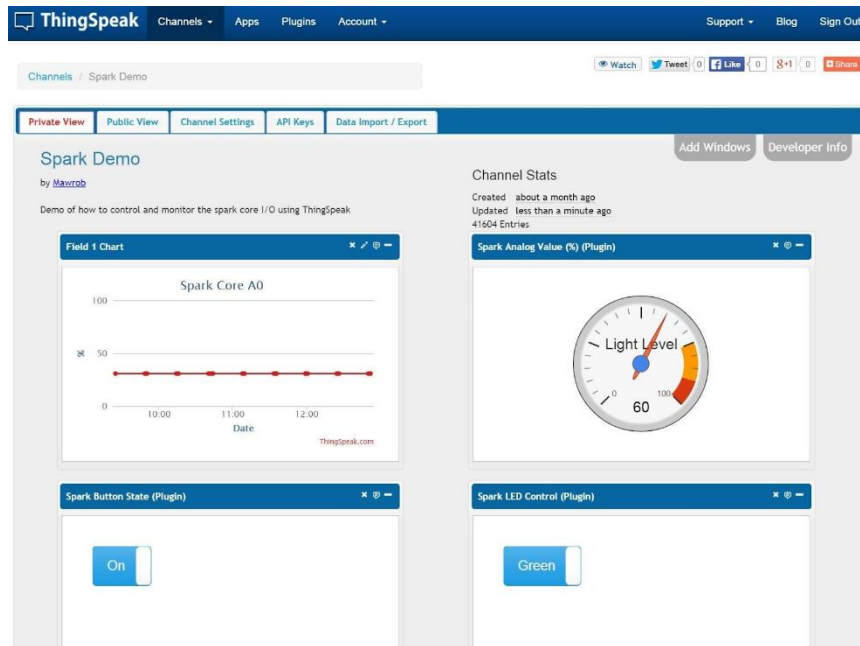
Hình 3.2. Sơ đồ kết nối I2C

| Raspberry   | Arduino      |
|-------------|--------------|
| SDA (GPIO2) | SDA (Pin A4) |
| SCL (GPIO3) | SCL (Pin A5) |
| GND         | GND          |

## CHƯƠNG 4. THINGSPEAK

### 4.1. Giới thiệu

Thingspeak là một nền tảng có thể trực quan hóa và phân tích dữ liệu trên đám mây. Nó là một sản phẩm Matlab và bạn có thể xử lý và phân tích dữ liệu trực tiếp từ đám mây. Chủ yếu nó được sử dụng trong các dự án IoT cần phân tích để theo dõi những thay đổi về giá trị cảm biến trên đám mây.



Hình 4.1. Mỗi kênh của nền tảng có thể nhận 1 dạng dữ liệu khác nhau

### 4.2. Các tính năng

Thingspeak cung cấp các khả năng khác nhau để thu thập, trực quan hóa và phân tích dữ liệu trên đám mây. Sau đây là các tính năng chính.

- Các thiết bị có thể dễ dàng định cấu hình và gửi dữ liệu tới Thingspeak bằng cách sử dụng các giao thức truyền thông.
- Có thể xem dữ liệu trong thời gian thực
- Có thể sử dụng với Matlab để phân tích dữ liệu
- Không cần server và phần mềm web để xây dựng prototype hệ thống IoT

### 4.3. Thingspeak hoạt động như thế nào

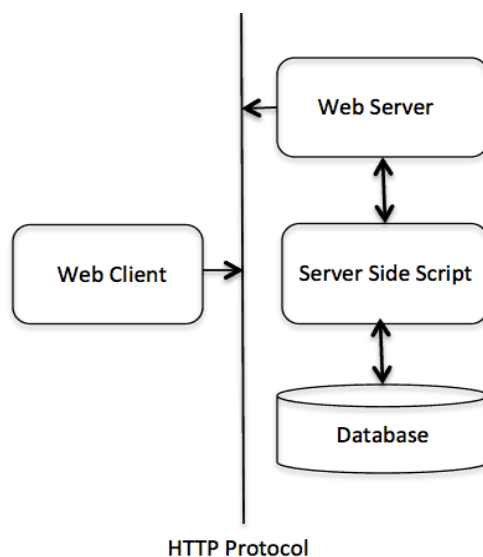
Ba bước chính mà Thingspeak yêu cầu để hoàn thành công việc đó là thu thập dữ liệu, phân tích và thực hiện một hành động.

Thingspeak hoạt động với điện toán đám mây, tất cả các thiết bị mà bạn muốn lấy dữ liệu phải nằm trong mạng với cơ sở dữ liệu đám mây. Thingspeak cũng kết nối với cơ sở dữ liệu đám mây và hiển thị luồng dữ liệu.

### 4.4. Áp dụng vào dự án thực tế

Được thiết kế lần đầu từ những năm 90, HTTP là 1 giao thức có thể mở rộng vốn đã phát triển dần theo thời gian. 1 giao thức lớp ứng dụng được gửi thông qua nền tảng TCP/IP, hay qua 1 kết nối TCP được mã hóa TLS. Mặc dù về mặt lý thuyết, bất kỳ giao thức truyền tải đáng tin cậy nào cũng có thể được sử dụng.

Nhờ vào khả năng mở rộng của nó, HTTP được sử dụng để không chỉ tìm nạp các tài liệu siêu văn bản mà còn cả hình ảnh và video hoặc để đăng tải nội dung lên server, giống như với các kết quả form HTML. HTTP cũng có thể được sử dụng để tìm nạp các phần của các doc nhằm cập nhật các trang web theo yêu cầu. HTTP không phải là một giao thức có tốc độ truyền và đọc dữ liệu nhanh. Nhưng trên thực tế, nó đáp ứng được nhu cầu của người dùng.



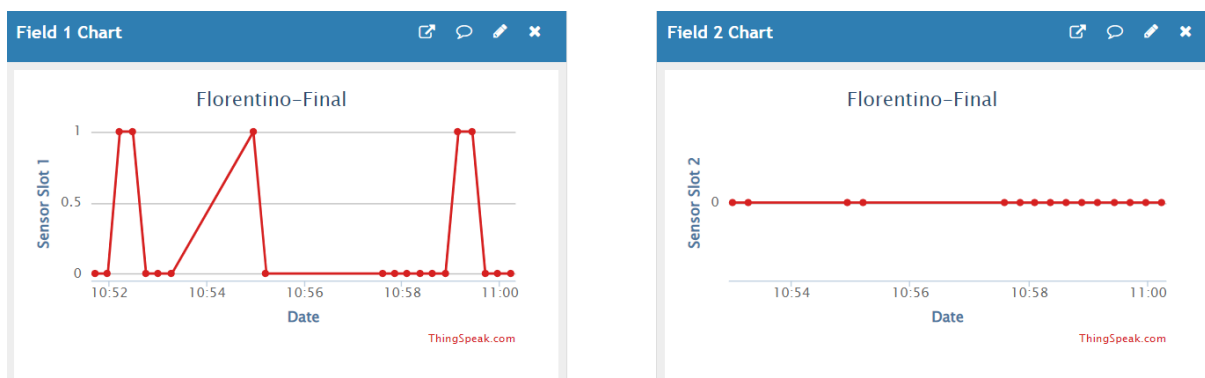
Hình 4.2. Mô hình hoạt động của giao thức

Dữ liệu do cảm biến truyền về sẽ được chuẩn hoá thành dạng số 0 và 1. Sau đó, Pi sẽ gửi dữ liệu này lên nền tảng Thingspeak, cụ thể là các channel đại diện cho các vị trí trong bãi gửi xe.

```
URL='http://api.thingspeak.com/update?api_key='
KEY = 'ABC'
HEADER = '&field1={}&field2={}'.format(val1, val2)
new_URL = URL+KEY+HEADER
v = urllib.request.urlopen(new_URL)
```

Hình 4.3. Dữ liệu được định dạng và gửi lên kênh

Mỗi kênh sẽ là một trung gian để nhận dữ liệu của cảm biến được Pi đưa lên. Từ đây, ứng dụng điện thoại sẽ lấy ra được các dữ liệu khi được kết nối với internet. Với độ trễ không quá cao so với yêu cầu thực tế. Sử dụng nền tảng này cho việc mô hình hoá và quản lý dữ liệu là điều khả thi.



Hình 4.4. Biểu đồ dữ liệu đại diện cho 2 vị trí trong bãi đỗ (0-Trống, 1-Đã có xe đỗ)

|   | A          | B        | C      | D      | E      | F        | G         | H         | I      |
|---|------------|----------|--------|--------|--------|----------|-----------|-----------|--------|
| 1 | created_at | entry_id | field1 | field2 | field3 | latitude | longitude | elevation | status |
| 2 | 2022-04-2  | 1        |        |        | -1     |          |           |           |        |
| 3 | 2022-04-2  | 2        |        |        | -1     |          |           |           |        |
| 4 | 2022-04-2  | 3        |        |        | -1     |          |           |           |        |
| 5 | 2022-04-2  | 4        |        |        | -1     |          |           |           |        |

Hình 4.5. Dữ liệu tại các kênh được export(Xuất) ra thành file dữ liệu



Từ đây ứng dụng của người dùng sẽ gửi các request lên hệ thống để nhận dữ liệu về điện thoại. Để có được dữ liệu này, ứng dụng cần API key của kênh, sau một thời gian nhất định, dữ liệu mới được cập nhật và cho người dùng biết số lượng xe, vị trí trống, ... thực tế trong bãi.

Read API Keys

Key

UCNJ3HCGA9NTRNNF

Note

Save Note

Delete API Key

Add New Read API Key

read key for the channel.

Note: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

GET https://api.thingspeak.com/update?api\_key=GLMF0VHK0FTYJF4Y&field

Read a Channel Feed

GET https://api.thingspeak.com/channels/1714971/feeds.json?api\_key=L

Read a Channel Field

GET https://api.thingspeak.com/channels/1714971/fields/1.json?api\_ke

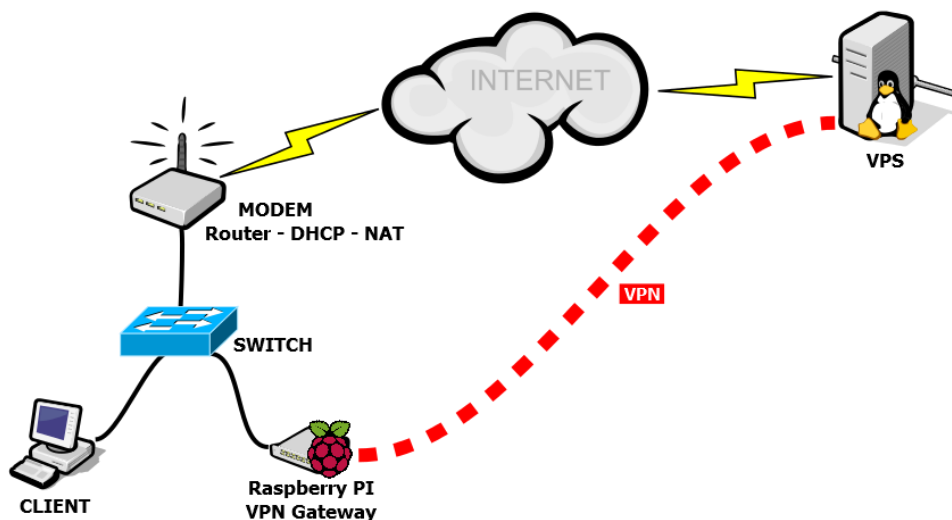
Read Channel Status Updates

GET https://api.thingspeak.com/channels/1714971/status.json?api\_key=

Learn More

Hình 4.6. Keys và các specific address để kết nối điện thoại và kênh thông tin

Tạo server từ Raspberry Pi đã được cộng đồng biết đến từ rất lâu. Với khả năng lưu trữ và truyền tải lượng thông tin khá tốt. Do đó, Raspberry Pi sẽ vừa là trung tâm điều khiển của hệ thống vừa đóng vai trò là một server cung cấp dịch vụ lưu trữ, truyền tải dữ liệu cho dự án.



Hình 4.7 Mô hình lưu trữ và quản lý dữ liệu qua Raspberry Pi

## **CHƯƠNG 5. APP MOBILE**

### **5.1. Tổng quan về App Mobile**

App mobile là một phần mềm điện tử được xây dựng bằng các ngôn ngữ lập trình và hoạt động trên nền tảng thiết bị di động. Hiện nay, android bao gồm thư viện các API giúp đơn giản hoá tối đa việc sử dụng phần cứng của thiết bị. Điều đó đảm bảo rằng bạn không cần phải bận tâm nhiều đến việc ứng dụng của mình có thể chạy như mong đợi trên nhiều thiết bị khác nhau hay không, miễn là thiết bị đó có hỗ trợ android.

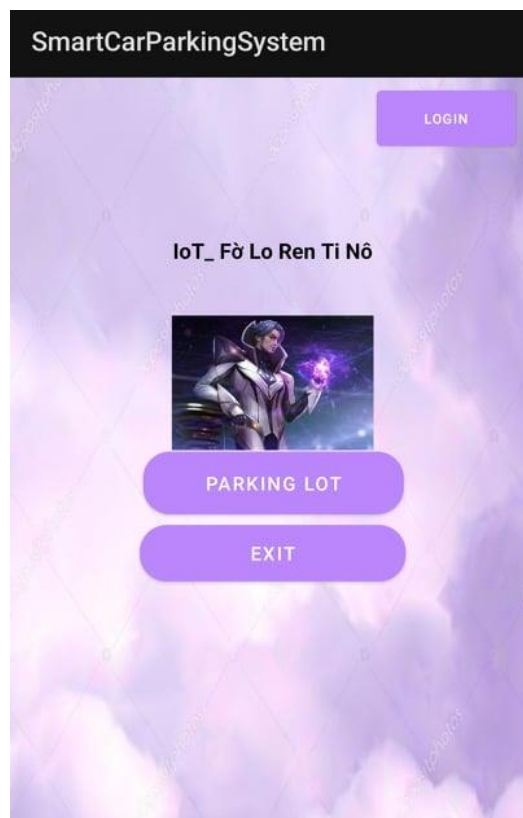
Hiện nay có rất nhiều ngôn ngữ để viết được một ứng dụng trên android như kotlin, react native, flutter, ... Trong đó java là ngôn ngữ thông dụng nhất, vì vậy ở đây nhóm sử dụng java làm ngôn ngữ cho app của mình.

### **5.2. Chức năng của App**

Để chủ xe dễ dàng tìm kiếm các bãi đỗ xe ở gần cũng như theo dõi tình trạng của chúng (còn hay đã hết chỗ) thì việc tích hợp thêm app mobile là tối cần thiết.

App sẽ được tích hợp API của google maps giúp người dùng dễ dàng tìm kiếm các bãi đỗ xe gần mình hoặc ở vị trí chỉ định sẵn. Đối với từng bãi đỗ xe, app sẽ lấy dữ liệu từ cảm biến và hiển thị xem bãi đỗ xe còn bao nhiêu chỗ, nếu còn ít chỗ để xe người dùng hoàn toàn có thể tìm kiếm 1 bãi đỗ xe khác với nhiều chỗ trống hơn.

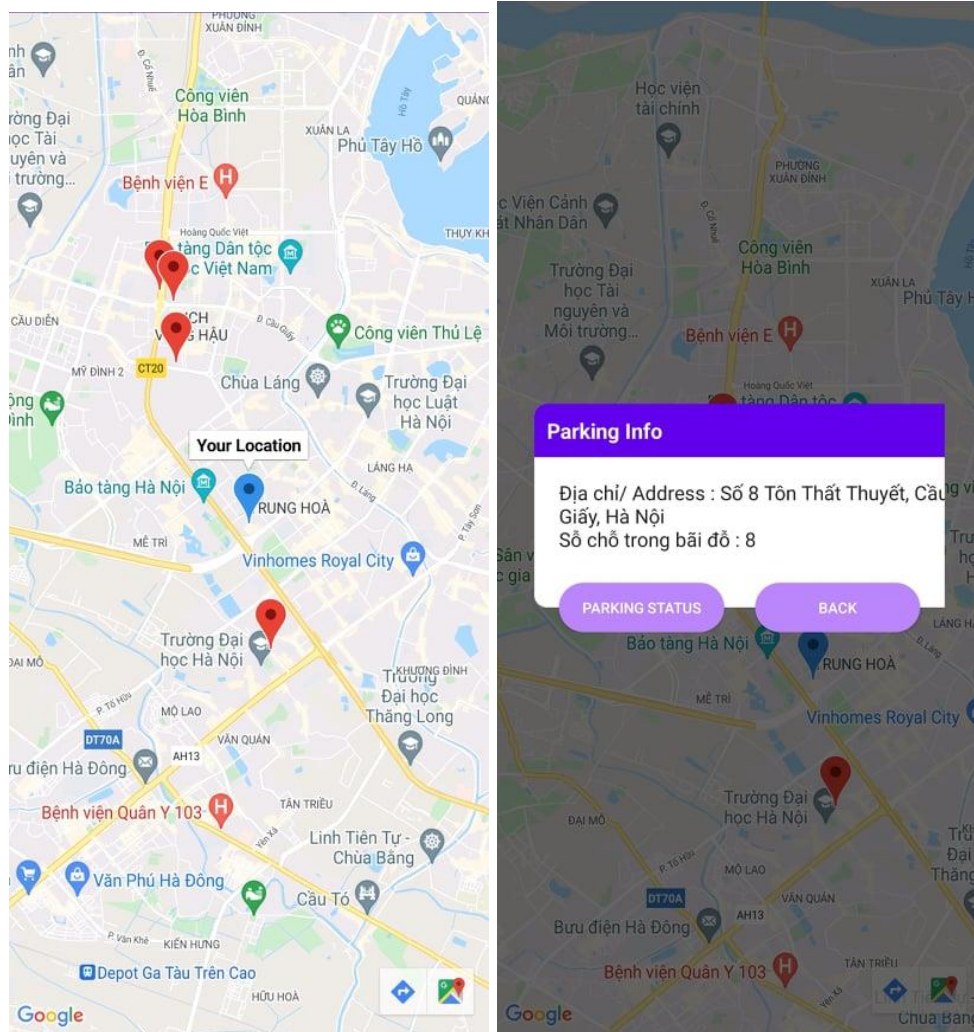
### 5.2.a. Giao diện ban đầu



Hình 5.1. Giao diện ban đầu

- LOGIN: Người dùng đăng nhập và tài khoản tại đây
- PARKING LOT: chọn bãi đỗ xe

### 5.2.b. Các bãi đỗ được tích hợp hệ thống IoT

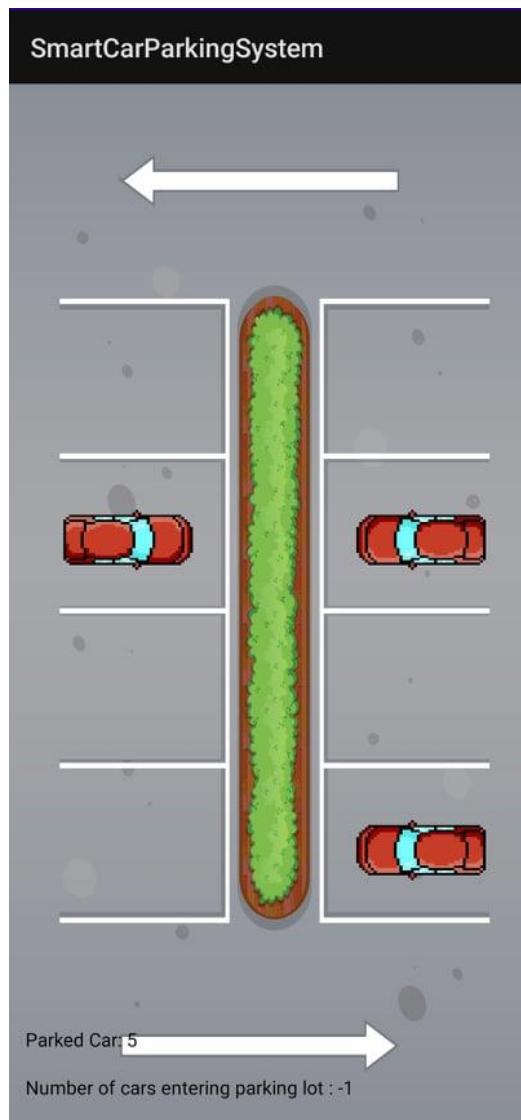


Hình 5.2. Giao diện các bãi đỗ

Tại đây, người dùng có thể thấy được vị trí hiện tại của mình và vị trí các bãi đỗ xe lân cận.

Khi chọn vào các điểm đỗ xe, thông tin bãi đỗ hiện lên thông báo địa chỉ bãi đỗ xe cũng như vị trí còn trống, thuận tiện cho việc lựa chọn bãi đỗ cho người dùng.

### 5.2.c. Bên trong bãi đỗ



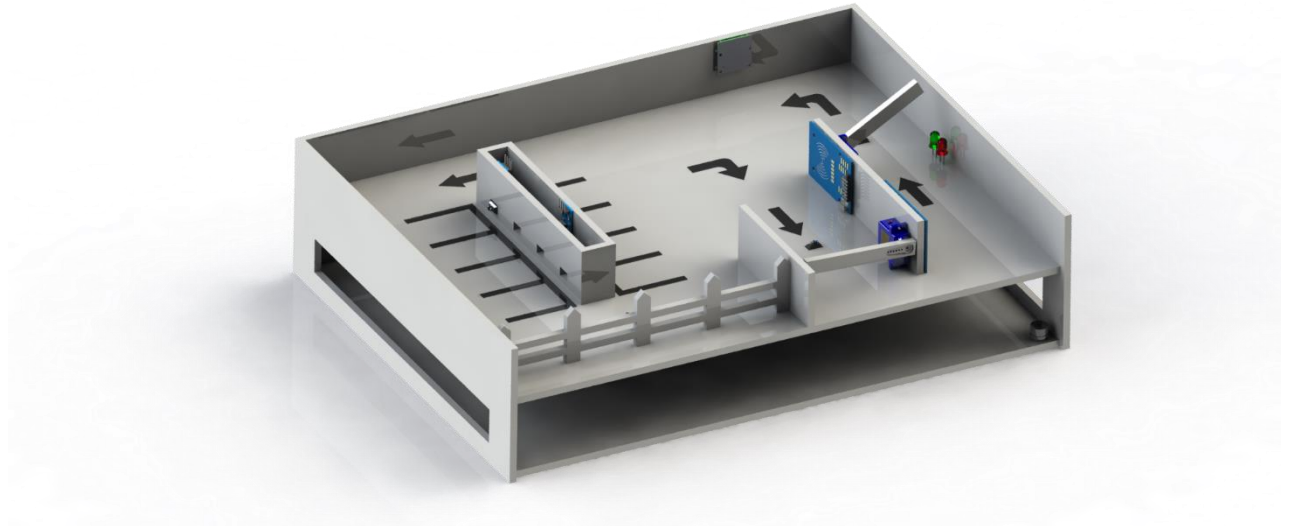
Hình 5.2. Giao diện bên trong bãi đỗ

Chức năng này hiển thị vị trí còn trống chính xác theo cách sắp xếp của bãi đỗ xe thực tế, giúp người dùng có góc nhìn tổng quan về vị trí trống bãi đỗ xe hiện tại giúp việc đỗ xe dễ dàng hơn

## CHƯƠNG 6. MÔ HÌNH THỰC TẾ

### 6.1. Phác hoạ 3D

Nhằm tăng tính thuyết phục cho sự khả thi của đề tài, nhóm đã quyết định thiết kế một nhà xe mini 3D và đầu tư kinh phí cho các cảm biến



Hình 6.1. Mô hình 3D bãi đỗ xe thông minh

### 6.2. Mô hình thực tế

Dưới đây là mô hình thực tế nhóm đã xây dựng với 2 tầng. Tầng dưới chứa, các bo mạch điều khiển, các cảm biến, loa cảnh báo,... Tầng trên được thiết kế giống như một bãi xe thật với barrier, vị trí đỗ, đèn báo hiệu và đặc biệt sử dụng công nghệ in 3D tại một số vị trí.



Hình 6.2. Tầng điều khiển tự động nhà xe



Hình 6.3. Mô hình thực tế nhà xe



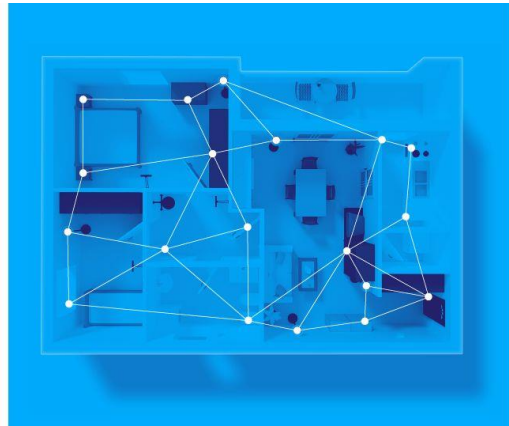
## CHƯƠNG 7. PHÁT TRIỂN SẢN PHẨM

### 7.1. Cấu trúc mạng lưới

Có 2 mạng lưới chủ yếu được sử dụng trong các hệ thống IoT là Star và Mesh.



Hình 7.1. Cấu trúc mạng Star



Hình 7.2. Cấu trúc mạng Mesh

#### 7.1.a. Mạng Star

Mạng Star có tất cả các trạm được kết nối với một thiết bị trung tâm có nhiệm vụ nhận tín hiệu từ các trạm và chuyển đến trạm đích. Vai trò của thiết bị trung tâm là thiết lập các liên kết Point to Point.

Việc thiết lập mạng đơn giản, dễ dàng cấu hình lại mạng (thêm, bớt các trạm) và có thể kiểm soát và khắc phục sự cố nhanh, đồng thời tận dụng được tối đa tốc độ truyền của đường truyền vật lý. Tuy nhiên, độ dài đường truyền nối một trạm với thiết bị trung tâm bị hạn chế (trong vòng 100m, với công nghệ hiện nay).

#### 7.1.b. Mạng Mesh

Mesh network hay mạng mesh là một cấu trúc liên kết mạng cục bộ trong đó các nút cơ sở hạ tầng (tức là cầu nối, thiết bị chuyển mạch và các thiết bị cơ sở hạ tầng khác) kết nối trực tiếp, động và không phân cấp với càng nhiều nút khác càng tốt và hợp tác với nhau để định tuyến dữ liệu đến client một cách hiệu quả.



### 7.1.c. Phương án

Nhóm sẽ chọn mạng Mesh để nhân rộng mô hình trong mỗi bãi xe do ít xảy ra sự cố hơn khi vì không cần thiết bị trung tâm, ngoài ra còn khắc phục được nhược điểm của BLE là khoảng cách đường truyền hạn chế.

## 7.2. Xử lý ảnh

### 7.2.a. OPENCV

#### - Giới Thiệu

OpenCV là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và máy học, và các tính năng tăng tốc GPU trong hoạt động thời gian thực.

OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn miễn phí cho cả học thuật và thương mại. Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực. Được viết bằng tối ưu hóa C/C++, thư viện có thể tận dụng lợi thế của xử lý đa lõi.

Như vậy, OpenCV là công nghệ phù hợp cho dự án vì là công nghệ mở, có tốc độ xử lý cao và có thể tùy biến theo nhu cầu của dự án.

#### - Các ứng dụng OpenCV

OpenCV đang được sử dụng rộng rãi trong các ứng dụng bao gồm:

- + Hình ảnh street view Kiểm tra và giám sát tự động Robot và xe hơi tự lái
- + Phân tích hình ảnh y tế Tìm kiếm và phục hồi hình ảnh/video phim - cấu trúc 3D từ chuyển động Nghệ thuật sắp đặt tương tác.

#### - Chức năng OpenCV

- + Image/video I/O, xử lý, hiển thị (core, imgproc, highgui) Phát hiện các vật thể (objdetect, features2d, nonfree) Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab) Computational photography (photo, video, superres) Machine learning & clustering (ml, flann) CUDA acceleration (gpu)

### **7.2.b. TESSERACT**

#### **- Giới thiệu**

Tesseract là một Công cụ nhận dạng văn bản (OCR) mã nguồn mở, có sẵn theo giấy phép Apache 2.0.

Các vấn đề mở có thể được tìm thấy trong trình theo dõi vấn đề và tài liệu lập kế hoạch. Tesseract có thể được sử dụng trực tiếp thông qua dòng lệnh hoặc (đối với lập trình viên) bằng cách sử dụng API để trích xuất văn bản in từ hình ảnh.

### **7.2.c. Nhược điểm**

- Thư viện Tesseract cần đầu vào chuẩn. Khi người dùng chưa tối ưu được hình ảnh trích xuất từ camera thì rất khó để lấy được thông tin



Hình 6.3. Ảnh chất lượng cao



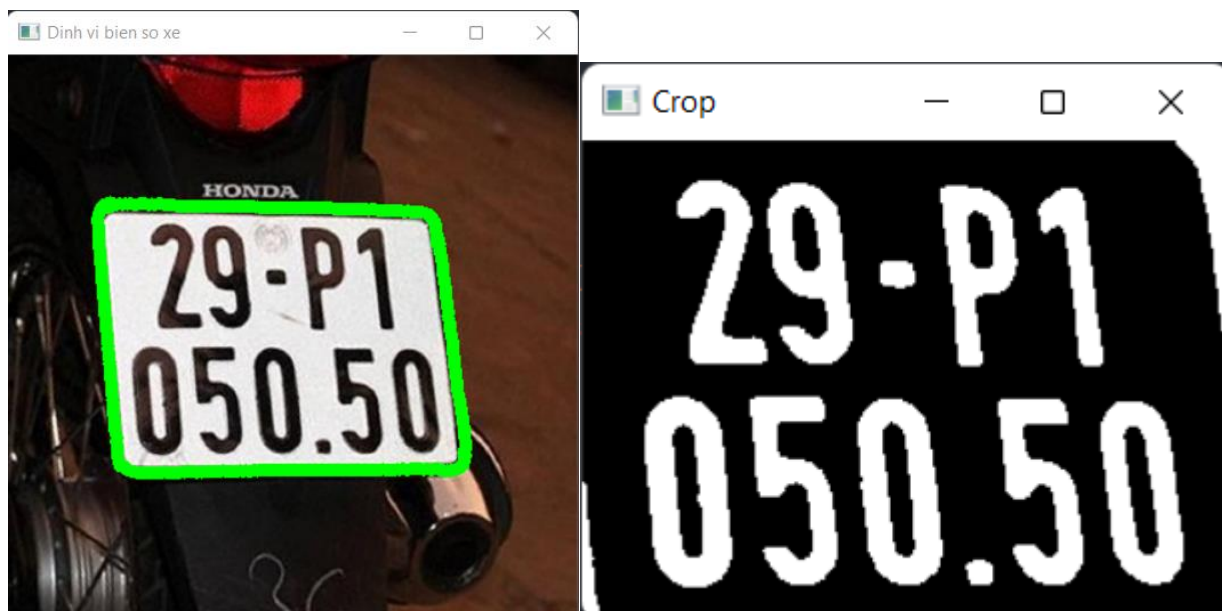
Hình 6.4. Ảnh đầu vào chất lượng thấp

#### 7.2.d. Hướng tiếp cận

Sử dụng hàm threshold để giảm nhiễu, đồng thời tìm ra các góc của biển số

```
thresh=cv2.adaptiveThreshold(gray,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY,11,2)
```

```
contours,h = cv2.findContours(thresh,1,2)
```



Hình 6.5. Quá trình xử lý ảnh và cắt góc ảnh

- Sử dụng Tess để nhận dạng văn bản.

```
pytesseract.pytesseract.tesseract_cmd='C:\ProgramFiles\Tesseract-OR\\tesseract.exe'
```

- Kết hợp cùng với cv2 để nhận hình ảnh live từ camera.

```
kernel=cv2.getStructuringElement(cv2.MORPH_RECT, (3,3))
```

```
opening=cv2.morphologyEx(thresh,cv2.MORPH_OPEN,kernel,iterations=1)
```

```
invert = 255 - opening
```

```
numberPlate_text=pytesseract.image_to_string(invert,lang='eng',config='--psm 6')
```



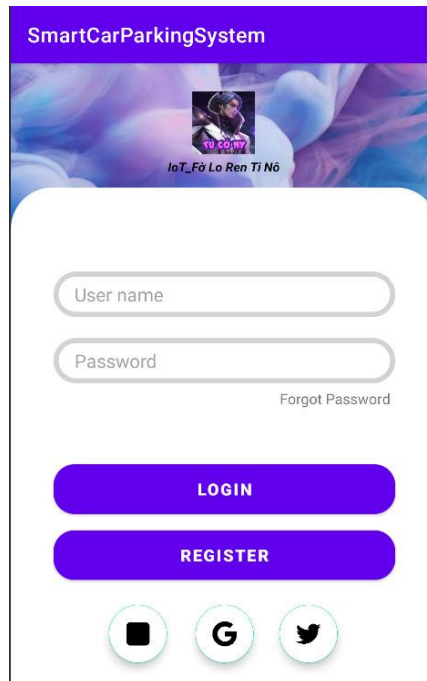
Hình 6.6 Hình ảnh thực tế từ camera

Hoạt động quét và lưu dữ liệu biển số sẽ được hoàn thành ngay sau khi người dùng quét thẻ RFID. Lần quét thẻ tiếp theo, hệ thống sẽ quét dữ liệu 1 lần nữa và so sánh với dữ liệu ban đầu nằm trong database. Tại đây nhóm sẽ sử dụng SQLite để lưu trữ dữ liệu của người dùng thông tin khách hàng.

### 7.3. App Mobile

Hiện tại app của nhóm vẫn đang phát triển giao diện người dùng cụ thể là quản lý phương tiện cá nhân, việc đăng nhập sẽ giúp người dùng quản lý được việc ra vào của xe mình và tính được chính xác số tiền gửi xe ( đối với mô hình gửi xe theo tháng ).

Ngoài ra, trong tương lai nhóm sẽ tích hợp định vị vào chủ phương tiện nhằm chỉ đường cho khách hàng tới bãi xe cũng như là vị trí đỗ gần nhất.



Hình 6.7. Giao diện đăng nhập