## 

## ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG DỮ LIỆU





BÁO CÁO MÔN HỌC

CÔNG NGHỆ INTERNET OF THINGS HIỆN ĐẠI

ĐỀ TÀI: BÃI ĐỖ XE THÔNG MINH

*Giảng viên hướng dẫn: TS.****Lê Kim Hùng***

*Lớp:* ***NT532.L21.MMCL***

*Sinh viên thực hiện:*

1.Phạm Thùy Nhung MSSV: 17520860

2. Vũ Hà Anh MSSV: 17520258

2. Nguyễn Phương Thùy MSSV: 18521476

*TP HCM, Ngày 29 tháng 06 năm 2021*



BÁO CÁO CHI TIẾT

**MỤC LỤC**

[Chương 1: Đặt vấn đề 1](#_Toc60555174)

[Chương 2: Thiết bị và nguyên lý hoạt động](#_Toc60555175) 2

2.1 Thiết bị 2

2.1.1: IR Sensor 2

a. IR Sensor là gì? 2

b. IR Sensor được sử dụng trong mô hình 3

2.1.2: Servo SG90 4

a. Servo là gì? 4

b. Servo SG90. 5

2.1.3: Board ESP8266 6

2.1.4: Màn hình LSC I2C 7

a. LCD 7

b. I2C 7

2.2: Nguyên lý hoạt động. 8

[Chương 3: Kiến trúc hệ thống](#_Toc60555176) 8

## 3.1: Kiến trúc phần cứng 8

## 3.2: Kiến trúc phần mềm 9

## 3.2.1: Giới thiệu Thingsboard 9

## 3.2.2: Nguyên lý hoạt động của Dashboard 10

[Chương 4: Tổng kết và đánh giá](#_Toc60555177) 13

Link video demo: 13

Link source code: 13

[Tài liệu tham khảo: 14](#_Toc60555180)

MỤC LỤC HÌNH ẢNH

Hình 1: Tình trạng tắt nghẽn giao thông 1

Hình 2: Sơ đồ cảm biến PIR. 3

Hình 3: Cảm biến tiệm cận hồng ngoại Mô-đun cảm biến tránh chướng ngại vật

hồng ngoại IR Tương thích với Arduino 3

Hình 4: Servo SG90 5

Hình 5: ESP8266 6

Hình 6: Màn hình LCD I2C 7

Hình 7: Mô hình phần cứng 8

Hình 8: Mô hình phần cứng sau khi triển khai (ảnh thật) 9

Hình 9: Mô hình phần mềm 10

Hình 10: Nguyên lý hoạt động của Dashboard 11

Hình 11: Nguyên lý hoạt động của Dashboard 11

Hình 12: Nguyên lý hoạt động của Dashboard 12

Hình 13 : Giao diện của màn hình điều khiển 1 12

Hình 14: : Giao diện của màn hình điều khiển 2 13

## Chương 1: Đặt vấn đề

Trong những năm gần đây cùng với sự phát triển của nền kinh tế, thu nhập đời sống của người dân được nâng lên cùng với đó là sự gia tăng chóng mặt của số lượng phương tiện đặc biệt ở các thành phố lớn.[1]

Trong khi đó, theo thống kê, hiện nay số lượng bãi đỗ xe có giấy phép ở các thành phố chỉ đáp ứng được 8-10% nhu cầu người dân, dẫn tới tình trạng thiếu bãi đỗ xe là vô cùng nghiêm trọng.[1]

Giao thông tĩnh ở các thành phố lớn như Hà Nội, TP. Hồ Chí Minh, Đà Nẵng hiện nay đang là một bài toán khó và cần giải quyết ngay. Hàng ngày, hàng giờ chúng ta vẫn thường xuyên được nghe những thông tin về tình trạng tắc đường tại các tuyến phố, đặc biệt trong những giờ cao điểm.[1]

**Hình 1: Tình trạng tắc nghẽn giao thông

Và một trong những nguyên nhân gây ra tình trạng đó là việc đỗ, dừng xe không đúng nơi quy định. Các điểm đỗ, dừng xe được tận dụng ở mọi chỗ, mọi nơi: trên vỉa hè, lòng đường, công viên các nơi không phép… điều đó ảnh hưởng không nhỏ đến giao thông nội đô và mỹ quan đô thị. [1]

Chính vì vậy việc áp dụng các **mô hình bãi đỗ xe thông minh** công cộng để giải quyết tình trạng khan hiếm chỗ để xe là vô cùng cần thiết đối với các đô thị lớn của nước ta.[1]

Đồng thời cũng giải quyết được vấn đề cắt giảm được nhân công duy trì việc trông giữ xe, mọi công đoạn được máy tính tính toán và thực hiện không cần đến bàn tay con người.

Bãi đỗ xe thông minh sẽ hoạt động cho phép xe ra vào tự động, mọi quản lý tình trạng bãi đỗ xe sẽ thông qua Internet.

## Chương 2: Thiết bị và nguyên lý hoạt động.

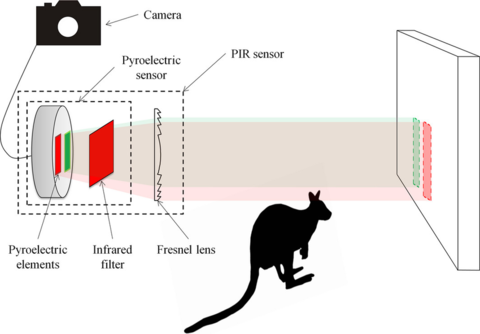
## 2.1: Thiết bị.

## 2.1.1: IR Sensor.

## a. IR Sensor là gì?

Cảm biến hồng ngoại (IR Sensor) là một thiết bị điện tử đo và phát hiện bức xạ hồng ngoại trong môi trường xung quanh của nó. Bức xạ hồng ngoại tình cờ được phát hiện bởi một nhà thiên văn học tên là William Herchel vào năm 1800. Trong khi đo nhiệt độ của từng màu ánh sáng (ngăn cách bằng lăng kính), ông nhận thấy rằng nhiệt độ ngay bên ngoài ánh sáng đỏ là cao nhất. IR là không thể nhìn thấy đối với mắt người, vì bước sóng của nó dài hơn bước sóng của ánh sáng nhìn thấy (mặc dù nó vẫn nằm trên cùng một phổ điện từ). Bất cứ thứ gì phát ra nhiệt (mọi thứ có nhiệt độ [**trên khoảng năm độ Kelvin**](https://www.livescience.com/50260-infrared-radiation.html)) đều phát ra bức xạ hồng ngoại.[2]

Có hai loại cảm biến hồng ngoại: chủ động và thụ động. Cảm biến hồng ngoại chủ động vừa phát ra vừa phát hiện bức xạ hồng ngoại. Cảm biến IR chủ động có hai phần: một điốt phát sáng (LED) và một bộ thu. Khi một đối tượng đến gần cảm biến, ánh sáng hồng ngoại từ đèn LED sẽ phản xạ khỏi đối tượng và được bộ thu phát hiện. Cảm biến IR chủ động hoạt động như [**cảm biến khoảng cách**](https://www.fierceelectronics.com/sensors/what-a-proximity-sensor) và chúng thường được sử dụng trong các hệ thống phát hiện chướng ngại vật (chẳng hạn như trong rô bốt).[2]



*Hình 2:* Sơ đồ cảm biến PIR. Nguồn: [***ResearchGate***](https://www.researchgate.net/figure/The-passive-infrared-PIR-trigger-consists-of-a-pyroelectric-sensor-and-Fresnel-lens_fig5_304608153)

## b. IR Sensor được sử dụng trong mô hình.



*Hình 3: Cảm biến tiệm cận hồng ngoại Mô-đun cảm biến tránh chướng ngại vật hồng ngoại IR Tương thích với Arduino*

Khi mô-đun phát hiện chướng ngại vật phía trước tín hiệu, đèn báo màu xanh lá cây trên bảng sẽ sáng ở mức. Và cổng OUT duy trì đầu ra tín hiệu thấp. Mô-đun phát hiện khoảng cách 20 ~ 300mm và góc 35 °. Khoảng cách thám sát có thể được điều chỉnh bằng chiết áp. Chiều kim đồng hồ điều chỉnh chiết áp, khoảng cách tăng dần; ngược chiều kim đồng hồ điều chỉnh chiết áp, khoảng cách. Hệ số phản xạ và hình dạng của mục tiêu là chìa khóa để phát hiện khoảng cách. Cổng đầu ra của mô-đun cảm biến OUT có thể được kết nối trực tiếp với cổng IO của bộ vi điều khiển hoặc trực tiếp điều khiển một rơ le 5V. Chế độ kết nối: VCC-VCC; GND-GND; RA-IO. Sử dụng bộ so sánh LM393, có hiệu suất ổn định. Có thể được cấp nguồn từ 3-5V DC. Khi bật nguồn, đèn báo nguồn màu đỏ sẽ sáng. Có lỗ bắt vít 3mm, thuận tiện cho việc cố định và lắp đặt. Điện áp so sánh ngưỡng đã được điều chỉnh thông qua chiết áp. Được sử dụng rộng rãi trong việc tránh chướng ngại vật bằng robot, theo dõi đường đen và trắng, v.v.

Khi mô-đun phát hiện chướng ngại vật phía trước tín hiệu, đèn báo màu xanh lá cây trên bảng sẽ sáng ở mức. Và cổng OUT duy trì đầu ra tín hiệu thấp.

Mô-đun phát hiện khoảng cách 20 ~ 300mm và góc 35.

Khoảng cách thám thính có thể được điều chỉnh bằng chiết áp. Chiều kim đồng hồ điều chỉnh chiết áp, khoảng cách tăng dần; ngược chiều kim đồng hồ điều chỉnh chiết áp, khoảng cách.

Hệ số phản xạ và hình dạng của mục tiêu là chìa khóa để phát hiện khoảng cách.

Cổng đầu ra của mô-đun cảm biến OUT có thể được kết nối trực tiếp với cổng IO của bộ vi điều khiển hoặc trực tiếp điều khiển một rơ le 5V. Chế độ kết nối: VCC-VCC; GND-GND; RA-IO.

Sử dụng bộ so sánh LM393, có hiệu suất ổn định.

Có thể được cấp nguồn từ 3-5V DC. Khi bật nguồn, đèn báo nguồn màu đỏ sẽ sáng.

Có lỗ bắt vít 3mm, thuận tiện cho việc cố định và lắp đặt.

Điện áp so sánh ngưỡng đã được điều chỉnh thông qua chiết áp.

Được sử dụng rộng rãi trong việc tránh chướng ngại vật bằng robot, theo dõi đường đen trắng, v.v.

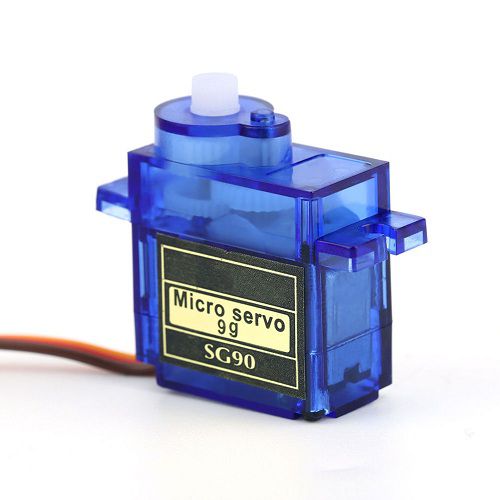
## 2.1.2: Servo SG90.

## a. Servo là gì?

Trong [kiểm soát kỹ thuật](https://en.wikipedia.org/wiki/Control_engineering) một động cơ servo , đôi khi rút ngắn xuống còn servo , là một thiết bị tự động sử dụng lỗi cảm biến [phản hồi tiêu cực](https://en.wikipedia.org/wiki/Negative_feedback) để khắc phục hành động của một cơ chế.  Trên các ứng dụng điều khiển dịch chuyển, nó thường bao gồm một [bộ mã hóa tích hợp](https://en.wikipedia.org/wiki/Encoder_(position)) hoặc cơ chế phản hồi vị trí khác để đảm bảo đầu ra đạt được hiệu quả mong muốn.

Thuật ngữ này chỉ áp dụng chính xác cho các hệ thống trong đó các tín hiệu [phản hồi](https://en.wikipedia.org/wiki/Feedback) hoặc sửa lỗi giúp kiểm soát vị trí cơ học, tốc độ, thái độ hoặc bất kỳ biến số có thể đo lường nào khác. [[](https://en.wikipedia.org/wiki/Servomechanism#cite_note-3) Ví dụ, điều khiển [cửa sổ điện trên](https://en.wikipedia.org/wiki/Power_window) ô tô không phải là cơ chế phục vụ, vì không có phản hồi tự động điều khiển vị trí — người vận hành thực hiện điều này bằng cách quan sát. Ngược lại, [hệ thống kiểm soát hành trình](https://en.wikipedia.org/wiki/Cruise_control) của ô tô sử dụng [phản hồi vòng kín](https://en.wikipedia.org/wiki/Control_theory) , phân loại nó như một cơ chế phục vụ.

## b. Servo SG90.



*Hình 4: Servo SG90*

Động cơ servo SG90 có thể xoay 180o

* Chủng loại : Động cơ servo SG90
* Kích thước : 22.2  x  11.8  x  32  mm
* Điện áp hoạt động : 5V

Công dụng chính của động cơ servo là đạt được góc quay chính xác trong khoảng từ 90o – 180o. Việc điều khiển này có thể ứng dụng để lái robot, di chuyển các tay máy lên xuống, quay một cảm biến để quét khắp phòng…

## 2.1.3: Board ESP8266.



*Hình 5: ESP8266*

ESP8266, hay gọi đầy đủ là ESP8266EX là một vi mạch [Wi-Fi](https://vi.wikipedia.org/wiki/Wi-Fi) [giá rẻ,](https://vi.wikipedia.org/wiki/Vi_%C4%91i%E1%BB%81u_khi%E1%BB%83n) có hỗ trợ [bộ giao thức TCP/IP](https://vi.wikipedia.org/wiki/TCP/IP) và có thể tích hợp vào thành phần của vi điều khiển, được sản xuất bởi hãng Espressif Systems ở Thượng Hải, Trung Quốc.

Chip ESP8226 lần đầu tiên được các nhà sản xuất phương Tây chú ý vào tháng 8 năm 2014 với module ESP-01, do nhà sản xuất bên thứ ba là Ai-Thinker sản xuất. Module này cho phép các vi điều khiển kết nối với mạng Wi-Fi và thực hiện các kết nối TCP/IP đơn giản bằng cách sử dụng các lệnh kiểu Hayes (tập lệnh AT). Tuy nhiên, ban đầu hầu như không có tài liệu tiếng Anh nào về chip và các tập lệnh của nó. Vì mức giá rất thấp với rất ít thành phần bên ngoài khác trên module, các module ESP8266 đã thu hút nhiều hacker khám phá nó và các phần mềm trên đó, cũng như việc dịch thuật các tài liệu tiếng Trung Quốc của chip.

ESP8285 là một chip ESP8266 với 1 MiB bộ nhớ flash được tích hợp, cho phép người dùng có thể sản xuất các thiết bị có khả năng kết nối với Wi-Fi chỉ với một chip đơn.

Dòng chip kế thừa ESP8266 là [ESP32](https://vi.wikipedia.org/wiki/ESP32), bao gồm cả vi điều khiển ESP32-C3tương thích chân.

## 2.1.4: Màn hình LCD I2C



*Hình 6: Màn hình LCD I2C*

## LCD.

**LCD 16×2** được sử dụng để hiển thị trạng thái hoặc các thông số.

LCD 16×2 có 16 chân trong đó 8 chân dữ liệu (D0 – D7) và 3 chân điều khiển (RS, RW, EN).

5 chân còn lại dùng để cấp nguồn và đèn nền cho LCD 16×2.

Các chân điều khiển giúp ta dễ dàng cấu hình LCD ở chế độ lệnh hoặc chế độ dữ liệu.

Chúng còn giúp ta cấu hình ở chế độ đọc hoặc ghi.

LCD 16×2 có thể sử dụng ở chế độ 4 bit hoặc 8 bit tùy theo ứng dụng ta đang làm.

## I2C.

LCD có quá nhiều nhiều chân gây khó khăn trong quá trình đấu nối và chiếm dụng nhiều chân trên vi điều khiển.

**Module I2C LCD** ra đời và giải quyết vấn để này cho bạn.

Thay vì phải mất 6 chân vi điều khiển để kết nối với LCD 16×2 (RS, EN, D7, D6, D5 và D4) thì module IC2 bạn chỉ cần tốn 2 chân (SCL, SDA) để kết nối.

Module I2C hỗ trợ các loại LCD sử dụng driver HD44780(LCD 16×2, LCD 20×4, …) và tương thích với hầu hết các vi điều khiển hiện nay.

## 2.2: Nguyên lý hoạt động.

Nguyên lý hoạt động của hệ thông bãi đỗ xe thông minh:

Khi xe vào cửa vào, cảm biến hồng ngoại sẽ phát hiện vật cản chặn phía trước, xét điều kiện nếu bãi xe còn chỗ đậu thì Servo sẽ hoạt động và mở cổng.

Tại mỗi vị trí đậu xe, có các cảm biến hồng ngoại được đặt tại đầu khu vực đậu xe để xác định tình trạng trống hay đã có xe đậu của các vị trí. Các vị trí trống sẽ được gửi lên server để xác nhận vị trí nào đang trống.

Cửa ra sẽ tự động mở khi xe lại gần tương tự như cổng vào sẽ có một cảm biến hồng ngoại và một servo để mở cổng ra, tình trạng chỗ đậu xe (còn trống) sẽ tăng lên một đơn vị.

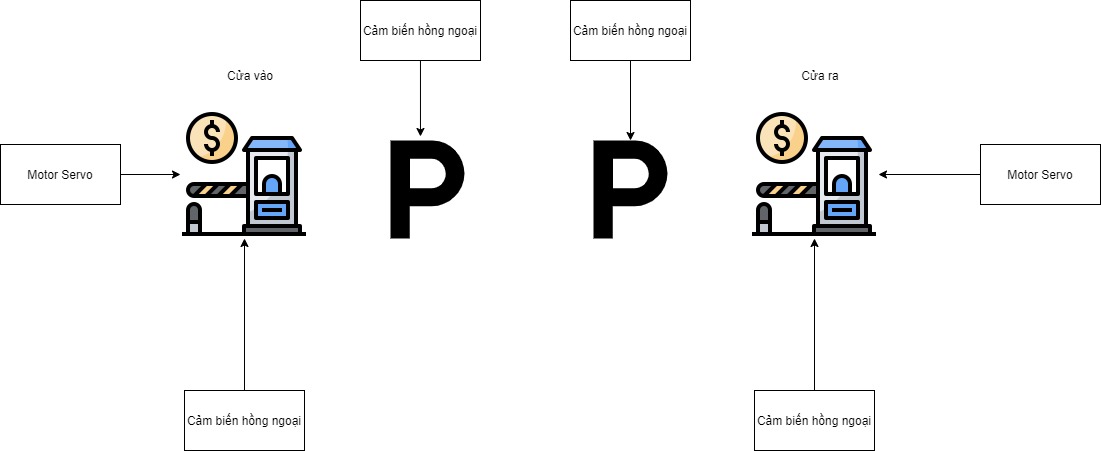
Các hoạt động của bãi xe tự động sẽ được gửi lên Web Server, các thông tin được gửi lên bao gồm: giờ xe vào, giờ xe ra, tình trạng của các vị trí đậu xe (còn trống hay đã đầy).

## Chương 3: Kiến trúc hệ thống.

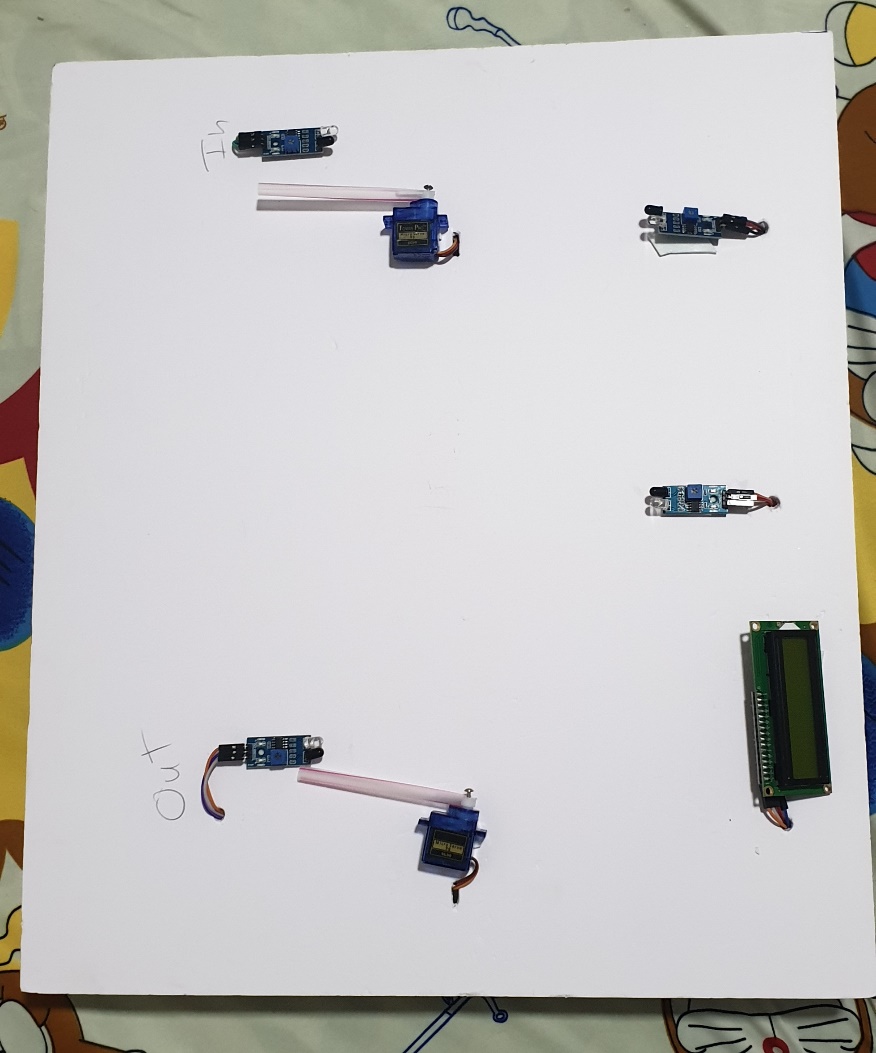
## 3.1: Kiến trúc phần cứng.

IR Sensor đóng vai trò chủ đạo trong mô hình này. Tại mỗi vị trí, công việc gửi/ nhận tín hiệu dựa vào kết quả phản hồi của các cảm biến.

* Cửa chỉ mở khi cảm biến phát hiện có xe di chuyển đến vị trí
* Bãi xe phản hồi tình trạng chỗ trống dựa vào cảm biến phát hiện có xe đầu vào hay chưa



*Hình 7: Mô hình phần cứng.*



*Hình 8: Mô hình phần cứng sau khi triển khai (ảnh thật)*

## 3.2: Kiến trúc phần mềm.

## 3.2.1 Giới thiệu Thingsboard

ThingsBoard là một nền tảng IoT mã nguồn mở. Nó cho phép phát triển nhanh chóng, quản lý và mở rộng các dự án IoT.  Với nền tảng Thingsboard người dùng có thể thu thập, xử lý, hiển thị trực quan và quản lý thiết bị.

Thingsboard cho phép kết nối thiết bị thông qua các giao thức IoT tiêu chuẩn công nghiệp – MQTT, CoAP và HTTP ,hỗ trợ cả triển khai đám mây và tại chỗ.

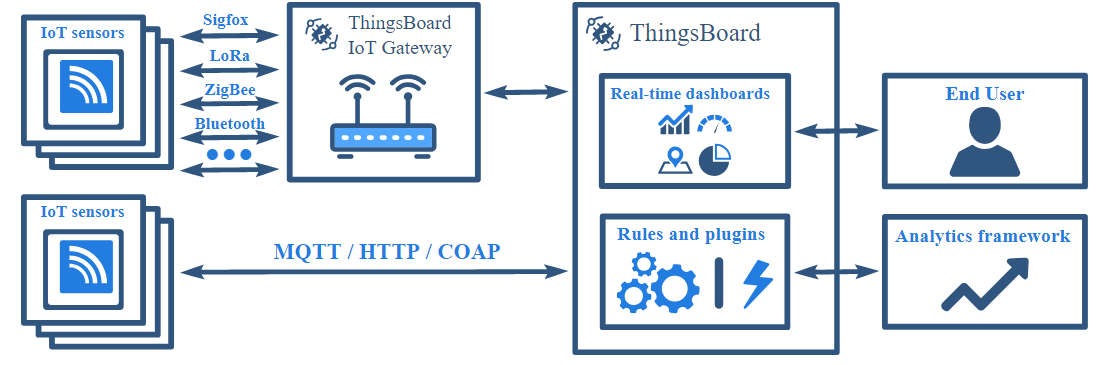
ThingsBoard cho phép người dùng tạo các Bảng điều khiển (Dashboard) IoT phong phú để hiển thị dữ liệu và điều khiển thiết bị từ xa trong thời gian thực. Ngoài ra, ThingsBoard còn cho phép tạo các chuỗi quy tắc phức tạp để xử lý dữ liệu từ thiết bị và phù hợp với các trường hợp sử dụng ứng dụng cụ thể.

Thingsboard hoạt động bằng cách thu thập các thông tin dữ liệu được gửi về từ các thiết bị cảm biến IoT. Có hai phương pháp chính trong việc gửi thông tin từ các cảm biến đến Thingsboard:

* Sử dụng các thiết bị có khả năng phát sóng truyền thông như Bluetooth, Zigbee, Lora để gửi tín hiệu về một Gateway ( ví dụ như Raspberry Pi ). Sau khi Gateway nhận được thông tin được gửi về, Gateway sẽ xử lý và hiển thị lên bảng điều khiển của Thingsboard.
* Sử dụng các giao thức truyền thông tin IoT như MQTT/HTTP/COAP. Thiết bị được khởi tạo trên Thingsboard sẽ cung cấp cho lập trình viên một mã token, tại phần mã nguồn lập trình, ta chỉ cần cài thêm các thư viện hỗ trợ các giao thức đó và sau đó sẽ có các hàm hỗ trợ gửi thông tin lên Thingsboard.

***Các tính năng của Thingsboard:***

* Quản lý các thiết bị từ xa
* Hiển thị trực quan dữ liệu trên bảng điều khiển.
* Sử dụng các chuỗi quy tắc để tạo nên ứng dụng trực quan
* Quản lý cảnh báo



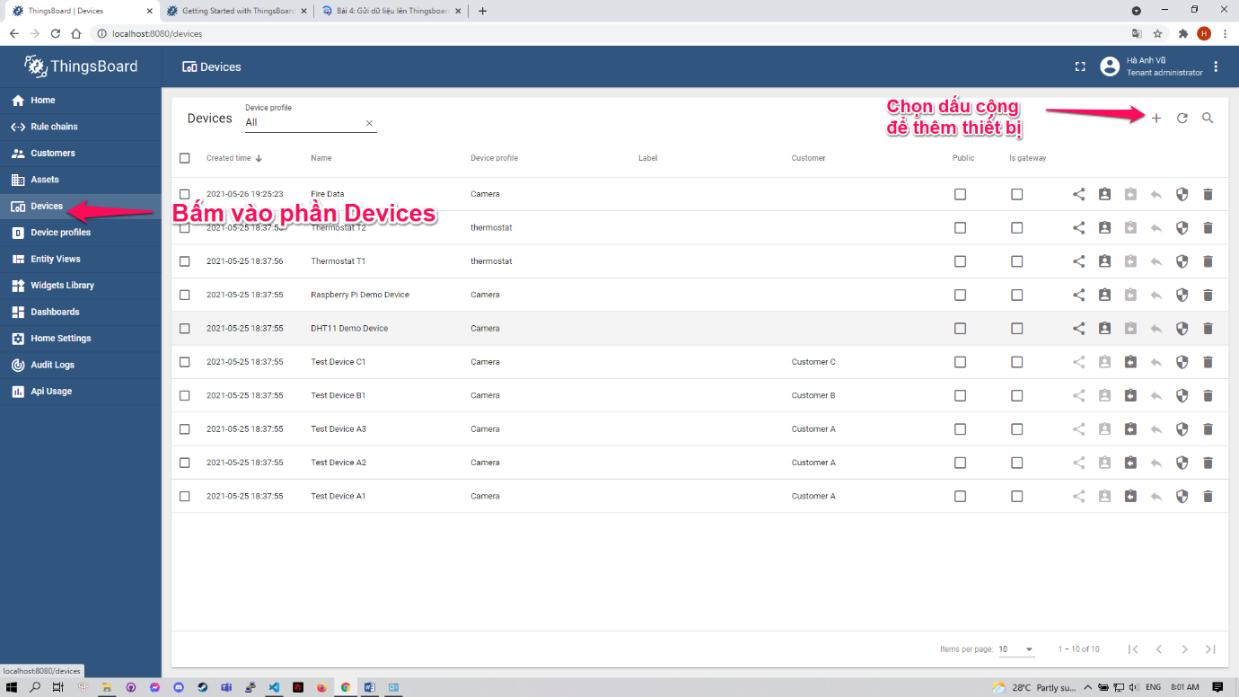
*Hình 9: Mô hình phần mềm*

## 3.2.2 Nguyên lý hoạt động của Dashboard

Dashboard lấy thông tin từ các cảm biến thiết lập trên mô hình. Ở đề tài này, nhóm đã sử dụng board ESP8266 được kết nối vào Wifi, sau đó sử dụng giao thức HTTP để gửi kết quả lên bảng quản lý tại địa chỉ demo.thingsboard.io, sau đó truy cập bằng tài khoản github.

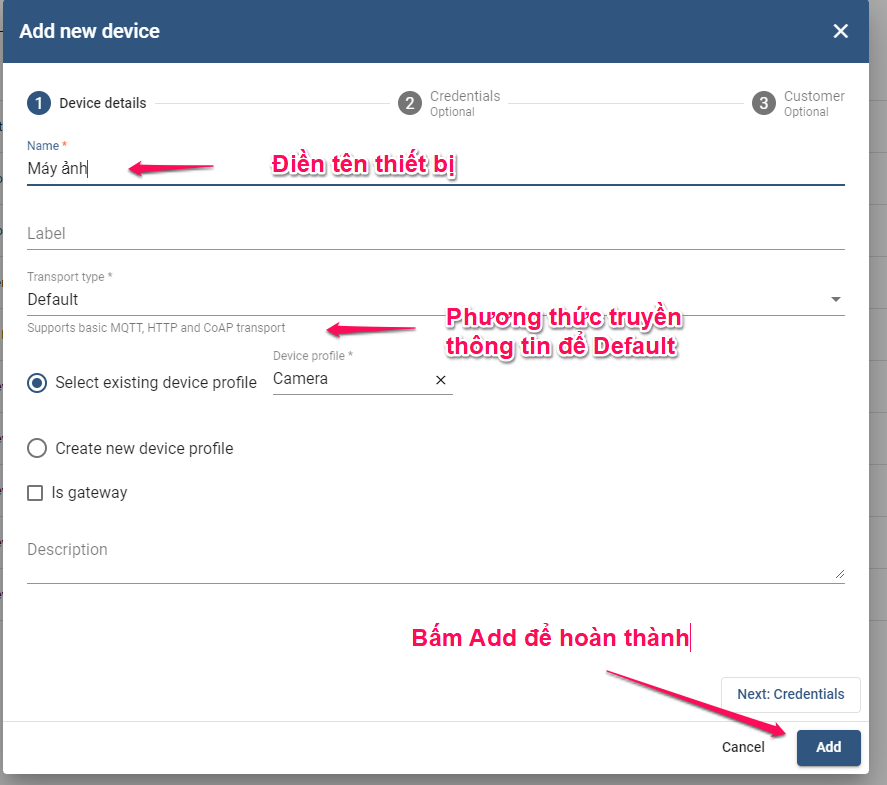
Để có thể giao tiếp với thingsboard, ta cần tạo thiết bị trên Thingsboard trước:

**Bước 1:** Tạo một thiết bị trên Thingsboard theo trình tự các bước trong hình sau:



*Hình 10: Nguyên lý hoạt động của Dashboard 1*

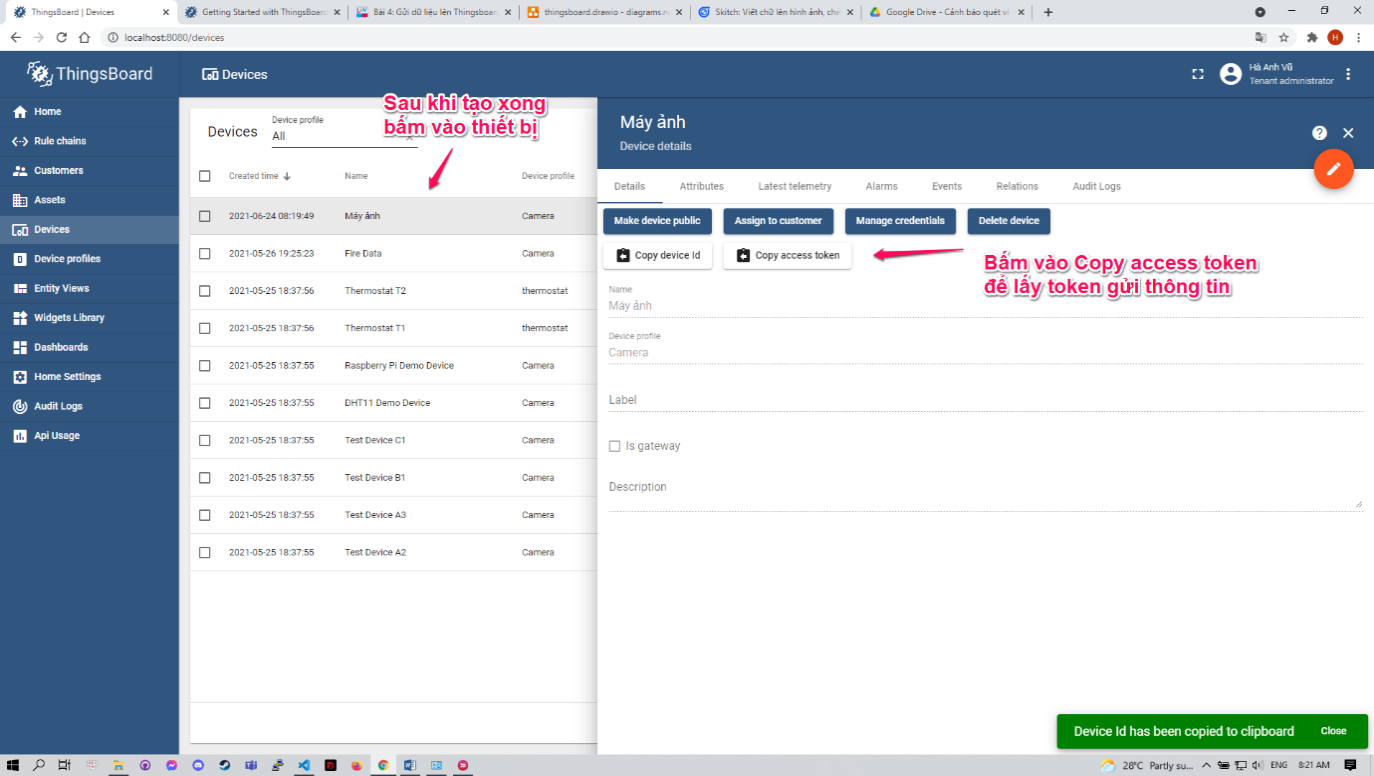
* Sau khi bấm vào dấu cộng, một bảng khởi tạo sẽ hiện lên:



*Hình 11: Nguyên lý hoạt động của Dashboard 2*

* Thực hiện các bước trên hình, sau đó bấm Add thì đã tạo xong một thiết bị mới

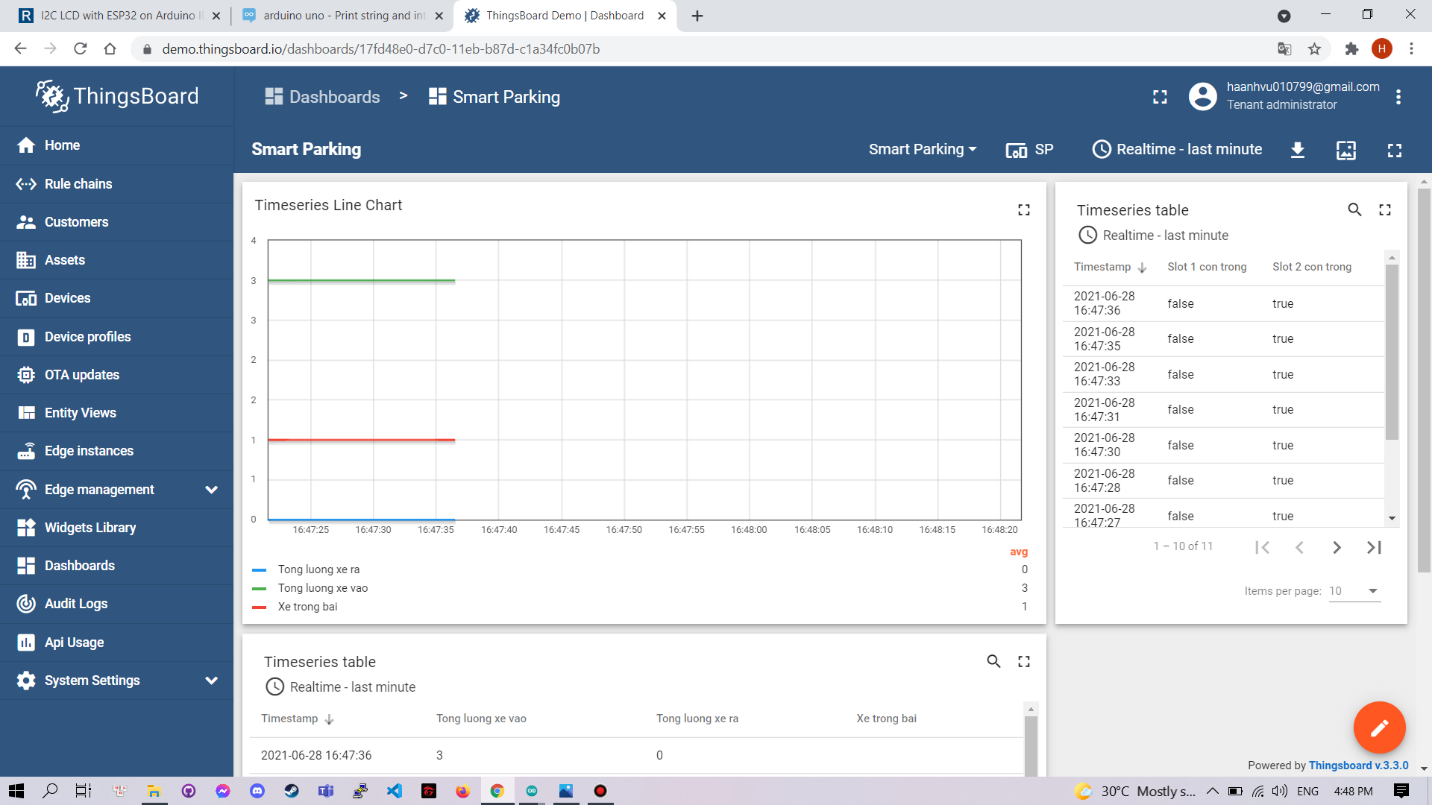
**Bước 2:** Lấy mã token để truy cập thiết bị



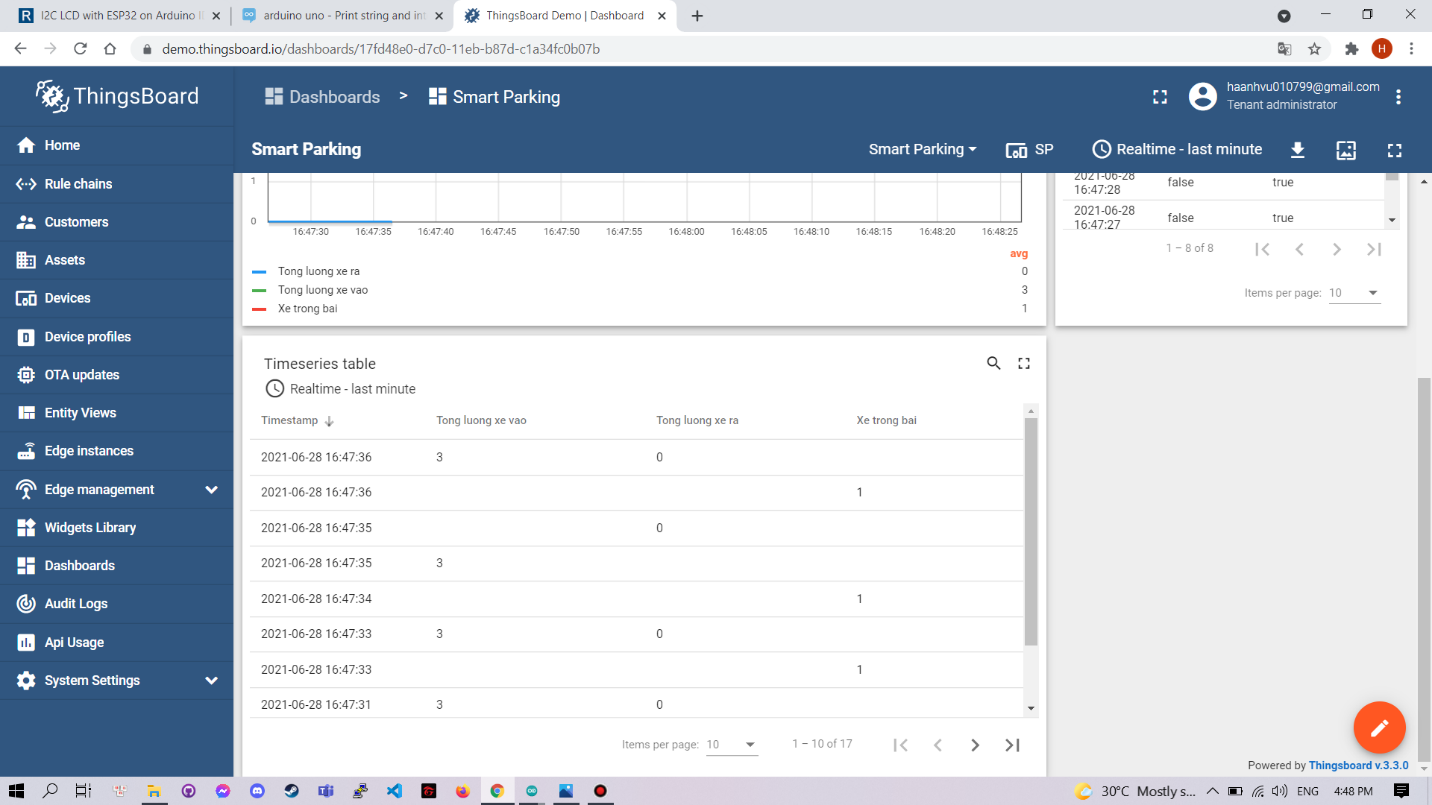
*Hình 12: Nguyên lý hoạt động của Dashboard 3*

Lấy mã Token trên Thingsboard

Sau khi cài đặt thành công, ta có giao diện của màn hình điều khiển hiển thị số xe hiện đang trong bãi, tình trạng chỗ trống của các Slot, tổng số xe ra/ vào.



*Hình 13: Giao diện của màn hình điều khiển 1.*

**

*Hình 14: Giao diện của màn hình điều khiển 2.*

## Chương 4: Tổng kết và đánh giá

## 4.1: Ưu điểm:

* Chi phí thấp, tất cả sensor và servo đều có giá rất rẻ.
* Dễ dàng triển khai lắp đặt.
* Board ESP8266 dễ dàng kết nối Wifi và gửi dữ liệu qua nhiều giao thức.
* Thời gian phản hồi nhanh, các cảm biến hoạt động ổn định, không bị nhiễu loạn.

## 4.2: Nhược điểm:

* Cảm biến hồng ngoại dễ bị nhầm lẫn khi có ánh sáng mặt trời mạnh.
* Không có khả năng cấp nguồn cho nhiều cảm biến.
* Chỉ có khả năng triển khai thử nghiệm, không áp dụng được vào thực tế.

***Link video demo:***

<https://drive.google.com/drive/u/0/folders/1_Th5B3eysd0WnB8fCLCB0BLdRrEnQZtn>

***Link source code:***

<https://github.com/haanhvu010799/Smart-Parking>

### Tài liệu tham khảo:

[1] <https://bilparking.com.vn/article/thuc-trang-bai-do-xe-thong-minh-thanh-pho#1.+Hi%E1%BB%87n+tr%E1%BA%A1ng+b%C3%A3i+%C4%91%E1%BB%97+xe+%E1%BB%9F+c%C3%A1c+th%C3%A0nh+ph%E1%BB%91+l%E1%BB%9Bn>

[2] <https://www.fierceelectronics.com/sensors/what-ir-sensor>

[3] <https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor>

[4] <https://nshopvn.com/product/cam-bien-sieu-am-hc-sr04/?gclid=Cj0KCQjw8vqGBhC_ARIsADMSd1DgFHcujbCef5Fe_Sk377ssUhp1b7doUBvTNN5fMpLhd6T-vwWJ3lsaAlSbEALw_wcB>