**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO**

**MÔN HỌC: CÔNG NGHỆ INTERNET OF THINGS HIỆN ĐẠI**

***ĐỀ TÀI:***

**Máy cho thú cưng ăn tự động**

**LỚP NT532.O21- NHÓM 06**

**Giảng viên hướng dẫn: Nguyễn Khánh Thuật**

**THÀNH VIÊN NHÓM:**

|  |  |
| --- | --- |
| Trần Đình Khôi | 21521017 |
| Nông Mạnh Hùng | 21522122 |
| Nguyễn Đặng Thanh Huy | 21522149 |

**Thành phố Hồ Chí Minh, tháng 04 năm 2024**

**MỤC LỤC**

**Phụ lục hình ảnh** 3

[**1.** **Giới thiệu đồ án** 4](#_Toc167989314)

[**2.** **Thiết bị sử dụng** 4](#_Toc167989315)

[2.1. NodeMCU ESP8266[1] 4](#_Toc167989316)

[2.2. Servo 996R[2] 4](#_Toc167989317)

[2.3. DHT11[3] 5](#_Toc167989318)

[2.3. HC-SR04[4] 5](#_Toc167989319)

[2.4. Load cell + HX711[5] 6](#_Toc167989320)

[**3.** **Thiết kế hệ thống** 6](#_Toc167989321)

[3.1. Lưu đồ hệ thống 6](#_Toc167989322)

[3.2. Mô hình hệ thống 7](#_Toc167989323)

[3.3. Xử lý thông tin ở ESP8266 8](#_Toc167989324)

[3.4. Xây dựng MQTT server[2] 12](#_Toc167989325)

[3.4.1. Mã nguồn ThingsBoard 12](#_Toc167989326)

[3.4.2. Thêm thiết bị mới 13](#_Toc167989327)

[3.4.3. Tạo màn hình Dashboard 15](#_Toc167989328)

[3.4.4. Xây dựng rule tạo cảnh bảo 16](#_Toc167989329)

[**4.** **Kết luận** 19](#_Toc167989330)

[**5.** **Hướng phát triển** 20](#_Toc167989331)

[**6.** **Tài liệu tham khảo** 20](#_Toc167989332)

**Phụ lục hình ảnh**

[Hình 1. ESP8266 4](#_Toc167989352)

[Hình 2. Động cơ servo 5](#_Toc167989353)

[Hình 3. Cảm biến DHT11 5](#_Toc167989354)

[Hình 4. Cảm biến khoảng cách DC-SR04 5](#_Toc167989355)

[Hình 5. Cảm biến loadcell và mạch HX711 6](#_Toc167989356)

[Hình 6. Lưu đồ của hệ thống 6](#_Toc167989357)

[Hình 7. Mô hình của hệ thống 7](#_Toc167989358)

[Hình 8. Một số hình ảnh thực tế của hệ thống 7](#_Toc167989359)

[Hình 9. Màn hình dashboard của server 8](#_Toc167989360)

[Hình 10. Thêm thiết bị mới trên nền tảng ThingsBoard 13](#_Toc167989361)

[Hình 11. Thêm thiết bị mới trên nền tảng ThingsBoard 13](#_Toc167989362)

[Hình 12. Đặt tên cho thiết bị mới 14](#_Toc167989363)

[Hình 13. Nhập các thông tin của MQTT client 14](#_Toc167989364)

[Hình 14. Tạo dashboard mới 15](#_Toc167989365)

[Hình 15. Nhập tên dashboard mới 15](#_Toc167989366)

[Hình 16. Thêm các widget cho dashboard 16](#_Toc167989367)

[Hình 17. Tạo Rule chain mới 16](#_Toc167989368)

[Hình 18. Nhập tên rule chain 17](#_Toc167989369)

[Hình 19. Các rule chain đã tạo để cảnh báo cho người dùng 17](#_Toc167989370)

[Hình 20. Các thành phần của rule **check hum** 17](#_Toc167989371)

[Hình 21. Cấu hình cho script của rule **check hum** 18](#_Toc167989372)

[Hình 22. Cấu hình cho script của rule **check tem** 18](#_Toc167989373)

[Hình 23. Thêm các rule mới vào Root Rule Chain 19](#_Toc167989374)

[Hình 24. Màn hình dashboard khi các cảnh báo được kích hoạt 19](#_Toc167989375)

**BÁO CÁO CHI TIẾT**

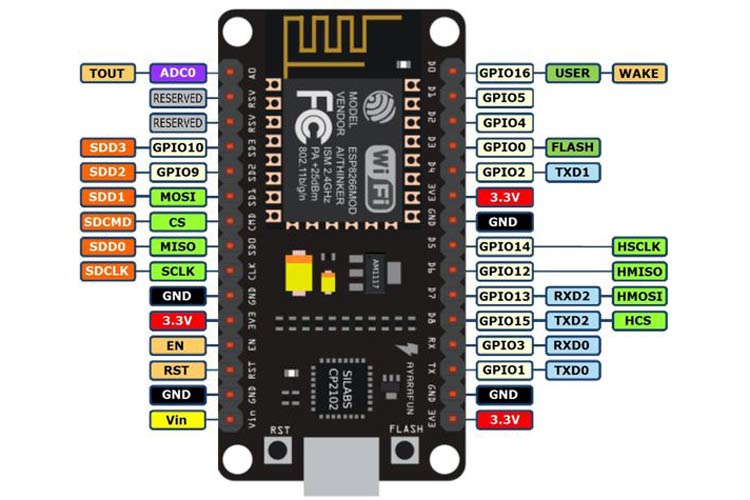
1. **Giới thiệu đồ án**

Trong thời đại hiện nay, nhu cầu sử dụng các thiết bị tự động hóa trong đời sống hàng ngày ngày càng cao. Bên cạnh đó sở thích nuôi thú cưng ngày càng phát triển. Nắm bắt được các nhu cầu đó, nhóm chúng em đã quyết định thực hiện đồ án máy cho thú cưng ăn tự động. Mục đích nhắm đến là giúp cho mọi người có thể chăm sóc thú cưng của mình tốt hơn mà không cần phải luôn có mặt bên cạnh chúng. Đồ án này trình bày việc thiết kế và chế tạo máy cho thú cưng ăn tự động sử dụng module ESP8266.

1. **Thiết bị sử dụng**

## 2.1. NodeMCU ESP8266[1]

ESP8266 là một module Wi-Fi với khả năng kết nối Internet và được tích hợp sẵn trên một số board nhúng NodeMCU (MCU). ESP8266 có thể hoạt động như một điểm truy cập (access point), một client kết nối đến một điểm truy cập khác, hoặc cả hai đều được. Nó được sử dụng rộng rãi trong các ứng dụng IoT (Internet of Things) như cảm biến thông minh, hệ thống kiểm soát thiết bị, hoặc các ứng dụng điều khiển từ xa. Trong đồ án này module ESP8266 được sử dụng để điều khiển các thiết bị của máy và kết nối mạng.



Hình 1. ESP8266

## 2.2. Servo 996R[2]

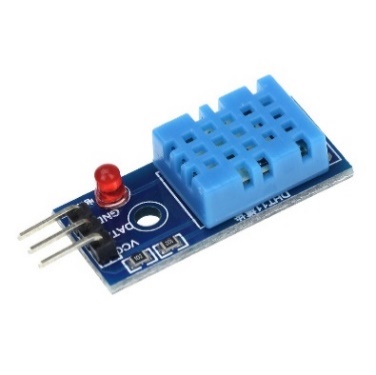
Servo 996 là một loại servo motor thông dụng trong các dự án điện tử và robotica. Nó cung cấp chức năng điều khiển chính xác vị trí và góc quay của một cơ cấu hoặc bộ phận cơ khí. Servo 996 có khả năng quay từ 0 đến 360 độ (trong đồ án này sử dụng 01 servo quay được 180 độ và 01 servo quay được 360 độ) và được điều khiển bằng các tín hiệu PWM (Pulse Width Modulation). Nó có động cơ mạnh mẽ và chất lượng ổn định, giúp nó thích hợp cho nhiều ứng dụng như robot cánh tay, xe tự động, máy in 3D và các dự án điều khiển chính xác vị trí. Servo 996 cũng được sử dụng phổ biến trong lĩnh vực mô hình hóa và điều khiển RC (Remote Control) do tính linh hoạt và dễ sử dụng của nó. Trong hệ thống này động cơ servo có chức năng đóng mở van xả thức ăn từ ngăn chứa thức ăn dự trữ đến khay đựng thức ăn và điều khiển thanh trượt để đưa khay chứa thức ăn ra ngoài



Hình 2. Động cơ servo

## 2.3. DHT11[3]

DHT11 là một cảm biến đo nhiệt độ và độ ẩm có giá thành rẻ và dễ sử dụng. Nó có thể cung cấp dữ liệu chính xác về nhiệt độ và độ ẩm của môi trường xung quanh. Cảm biến này thích hợp cho các dự án điều khiển môi trường như cảm biến thời tiết, quản lý điều hòa không khí trong nhà, và giám sát môi trường nông nghiệp. Trong đồ án này sử dụng DHT11 để theo dõi nhiệt độ và độ ẩm của thức ăn trong ngăn dự trữ.



Hình 3. Cảm biến DHT11

## 2.3. HC-SR04[4]

HC-SR04 là một cảm biến siêu âm đo khoảng cách. Nó sử dụng sóng siêu âm để đo khoảng cách từ chính nó đến vật thể trong phạm vi của nó. Với khả năng đo khoảng cách từ một vài centimet tới vài mét, HC-SR04 rất phổ biến trong các ứng dụng như xe tự lái, robot tự hành, và các dự án IoT liên quan đến đo khoảng cách. Trong đồ án này sử dụng cảm biến HC-SR-04 để đo khoảng cách từ nắp ngăn chứa đến bề mặt thức ăn, qua đó xác định được lượng thức ăn dự trữ còn lại.



Hình 4. Cảm biến khoảng cách DC-SR04

## 2.4. Load cell + HX711[5]

Load cell và HX711 là một cặp thiết bị thường được sử dụng để đo lực và trọng lượng. Load cell là thiết bị chuyển đổi lực thành tín hiệu điện, trong khi HX711 là một bộ chuyển đổi tín hiệu analog sang kỹ thuật số với độ chính xác cao. Khi kết hợp với nhau, chúng có thể đo được trọng lượng với độ chính xác cao và phổ biến trong các ứng dụng như cân điện tử, cân đóng gói và cân kiểm tra lực. Trong hệ thống này cảm biến loadcell và mạch HX711 dùng để giám sát lượng thức ăn thừa của thú cưng.



Hình 5. Cảm biến loadcell và mạch HX711

1. **Thiết kế hệ thống**

## 3.1. Lưu đồ hệ thống

**A diagram of a diagram

Description automatically generated**

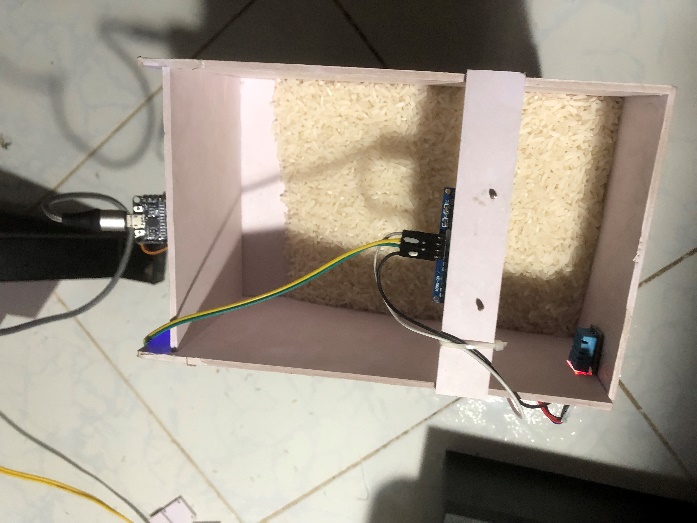
Hình 6. Lưu đồ của hệ thống

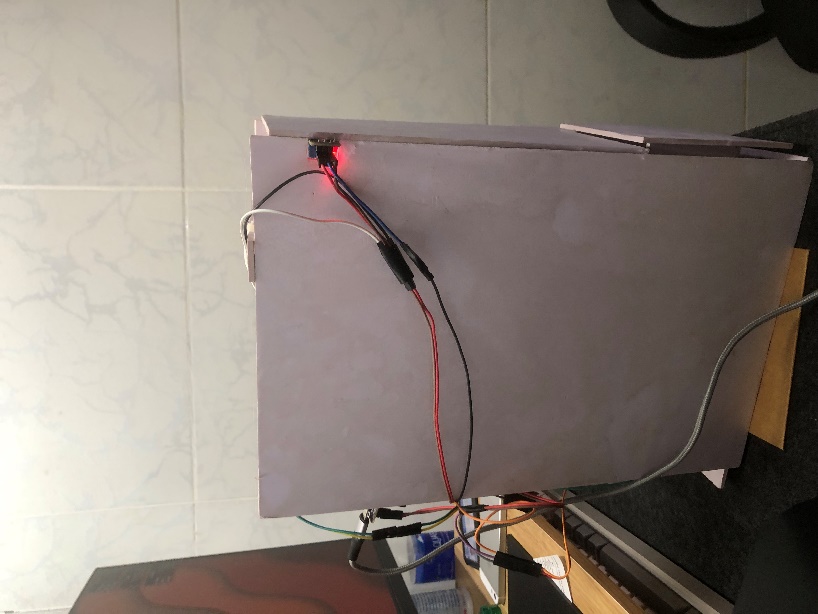
Khi khởi động máy, các cảm biến sẽ gửi dữ liệu đến ESP8266, sau đó ESP8266 sẽ tổng hợp và gửi các dữ liệu này đến server. Server sẽ hiển thị các dữ liệu này lên dashboard.

## 3.2. Mô hình hệ thống

**A circuit board with wires

Description automatically generated**

A white box with wires and wires

Description automatically generatedHình 7. Mô hình của hệ thống

Hình 8. Một số hình ảnh thực tế của hệ thống

A screenshot of a computer

Description automatically generated  
Hình 9. Màn hình dashboard của server

Khi nhấn các nút điều khiển “Cho ăn” hoặc “Thêm thức ăn” thì server sẽ gửi tín hiệu điều khiển có dạng {"method":"<tên metho>","params":<giá trị boolean>} đến ESP8266. Sau đó ESP8266 sẽ phân tích thông tin nhận được và đưa ra hành động cụ thể:

Nếu tín hiệu nhận được có tên metho là “setValue” thì tín hiệu nhận được sẽ điều khiển khay trượt chứa thức ăn, tùy vào giá trị của params mà đóng hoặc mở khay.

Nếu tín hiệu nhận được có tên metho là “setValue1” thì tín hiệu nhận được sẽ dùng để điều khiển van xả thức ăn, nếu giá trị params bằng 1 thì sẽ mở van xả thức ăn 0.5s để thêm thức ăn vào khay chứa.

## 3.3. Xử lý thông tin ở ESP8266

Thêm các thư viện, định nghĩa các hằng số như SSID và mật khẩu của wifi muốn kết nối tới, các thông tin của MQTT server, các chân kết nối với cản biến cũng như động cơ servo.

Hàm callback được gọi khi nhận được tin nhắn từ MQTT server đã đăng ký. Hàm còn có chức năng chuyển đổi payload thành chuỗi và phân tích thành dữ liệu dạng JSON, cùng với đó là phân tích dữ liệu JSON nhận được để điều khiển các động cơ servo.

Hàm reconnect để kết nối lại với MQTT server nếu xảy ra mất kết nối.

Hàm loop để duy trì kết nối với MQTT, đọc dữ liệu từ cảm biến và gửi đến MQTT server dưới dạng JSON. Do nhóm chưa hoàn thành được chức năng đo lượng thức ăn thừa nên giá trị **net1** đang để mặc định là 100.

## 3.4. Xây dựng MQTT server[2]

### 3.4.1. Mã nguồn ThingsBoard

ThingsBoard là một nền tảng mã nguồn mở dùng để quản lý và giám sát các thiết bị IoT, cung cấp các tính năng quan trọng như quản lý thiết bị, thu thập và phân tích dữ liệu, hình ảnh hóa dữ liệu qua các dashboard tùy chỉnh, và thiết lập cảnh báo sự kiện. Nền tảng này hỗ trợ nhiều giao thức IoT phổ biến như MQTT, CoAP và HTTP, có khả năng mở rộng để xử lý hàng triệu thiết bị và hàng tỷ tin nhắn mỗi ngày, đồng thời cung cấp các tính năng bảo mật như xác thực thiết bị, mã hóa dữ liệu và kiểm soát truy cập. ThingsBoard thường được ứng dụng trong giám sát từ xa, tự động hóa công nghiệp và năng lượng thông minh, giúp các nhà phát triển và doanh nghiệp triển khai các giải pháp IoT một cách hiệu quả và tiết kiệm chi phí.

### 3.4.2. Thêm thiết bị mới

Trên màn hình làm việc chính của ThingsBoard, chọn **Entities → Device → Add device**

**A screenshot of a computer

Description automatically generated**

Hình 10. Thêm thiết bị mới trên nền tảng ThingsBoard

Sau đó chọn **Add new divice**

**A screenshot of a computer

Description automatically generated**

Hình 11. Thêm thiết bị mới trên nền tảng ThingsBoard

Nhập tên của thiết bị mới rồi chọn **Netxt: Credentials**

**A screenshot of a computer

Description automatically generated**

Hình 12. Đặt tên cho thiết bị mới

Chọn **MQTT Basic**, nhập các thông tin của MQTT client rồi chọn **ADD**

**A screenshot of a login screen

Description automatically generated**

Hình 13. Nhập các thông tin của MQTT client

### 3.4.3. Tạo màn hình Dashboard

Trên màn hình chính chọn **Dashboard → Add dashboard → Create new dashboard**

A screenshot of a computer

Description automatically generated

Hình 14. Tạo dashboard mới

Tại phần Title nhập tên dashboard mới sau đó chọn **Add**

**A screenshot of a computer

Description automatically generated**

Hình 15. Nhập tên dashboard mới

Chọn **Add widget** rồi thêm các widget mong muốn, sau đó chọn **Save** để hoàn thành

A screenshot of a computer

Description automatically generated

Hình 16. Thêm các widget cho dashboard

**3.4.4. Xây dựng rule tạo cảnh bảo**

Trên màn hình chính chọn **Rule chain** rồi chọn **Create new rule chain**:

**A screenshot of a computer

Description automatically generated**

Hình 17. Tạo Rule chain mới

Nhập tên rule chain mới rồi chọn **ADD**

A screenshot of a facebook messenger

Description automatically generated

Hình 18. Nhập tên rule chain

A screenshot of a phone

Description automatically generated

Hình 19. Các rule chain đã tạo để cảnh báo cho người dùng

Sau đó thêm và cấu hình cho các thành phần của rule chain

A screenshot of a computer

Description automatically generated

Hình 20. Các thành phần của rule **check hum** (các rule còn lại tương tự)

A screenshot of a chat

Description automatically generated

Hình 21. Cấu hình cho script của rule **check hum**

A screenshot of a computer

Description automatically generated

Hình 22. Cấu hình cho script của rule **check tem**

\**Các rule chain còn lại cũng cấu hình script tương tự để tạo cảnh báo*

Sau khi tạo rule thành công cần thêm các rule chain mới vào Root Rule Chain để lấy dữ liệu input

A screenshot of a computer

Description automatically generated

Hình 23. Thêm các rule mới vào Root Rule Chain

A screenshot of a computer

Description automatically generated

Hình 24. Màn hình dashboard khi các cảnh báo được kích hoạt

1. **Kết luận**

Trong đồ án này, chúng em đã triển khai một hệ thống IoT sử dụng nền tảng ThingsBoard để giám sát và điều khiển các thiết bị từ xa. Hệ thống bao gồm các cảm biến đo nhiệt độ, độ ẩm, và khoảng cách, cũng như các động cơ servo điều khiển khay tự động. Dữ liệu từ các cảm biến được thu thập và gửi lên ThingsBoard thông qua giao thức MQTT, giúp người dùng có thể giám sát các thông số môi trường theo thời gian thực và điều khiển các thiết bị từ xa thông qua các dashboard trực quan.

Việc sử dụng ThingsBoard đã giúp đơn giản hóa quá trình quản lý thiết bị và xử lý dữ liệu, đồng thời cung cấp một nền tảng mạnh mẽ và linh hoạt để mở rộng hệ thống trong tương lai. Hệ thống này có thể được áp dụng trong nhiều lĩnh vực như tự động hóa nhà ở, quản lý nông trại thông minh, và các ứng dụng công nghiệp khác.

Kết quả đạt được từ đồ án đã chứng minh tính khả thi và hiệu quả của việc tích hợp các thiết bị IoT với nền tảng ThingsBoard, đồng thời mở ra nhiều hướng nghiên cứu và phát triển thêm, như cải thiện độ chính xác của cảm biến, tăng cường tính năng bảo mật, và mở rộng quy mô hệ thống để xử lý lượng dữ liệu lớn hơn. Tổng kết lại, đồ án không chỉ đạt được hầu hết các mục tiêu đề ra (chức năng cân thức ăn thừa chưa hoàn thành) mà còn mang lại những kiến thức và kinh nghiệm quý báu trong lĩnh vực IoT cho các thành viên trong nhóm.

1. **Hướng phát triển**

Dựa trên kết quả đạt được và những kinh nghiệm thu thập từ đồ án này, có một số hướng phát triển tiềm năng để nâng cao chất lượng của hệ thống hiện tại. Trước hết, việc tăng cường bảo mật là cần thiết, bao gồm sử dụng các giao thức xác thực mạnh hơn và mã hóa dữ liệu để bảo vệ thông tin truyền tải, cùng với cơ chế kiểm soát truy cập chi tiết hơn để đảm bảo chỉ những người dùng được ủy quyền mới có thể điều khiển các thiết bị và truy cập dữ liệu. Cải thiện độ chính xác và hiệu suất của cảm biến cũng là một hướng phát triển quan trọng, thông qua việc thử nghiệm và tích hợp các cảm biến có độ chính xác cao hơn và áp dụng các phương pháp hiệu chuẩn để giảm sai số. Mở rộng phạm vi ứng dụng của hệ thống vào các lĩnh vực như công nghiệp và nông nghiệp thông minh có thể giúp tăng cường hiệu quả giám sát và quản lý môi trường trồng trọt, tưới tiêu, và sản xuất. Bên cạnh đó, tích hợp các thuật toán học máy và trí tuệ nhân tạo để phân tích dữ liệu, dự báo xu hướng và tối ưu hóa hoạt động của hệ thống cũng là một hướng đi tiềm năng. Tối ưu hóa hiệu suất và khả năng mở rộng thông qua kiến trúc phân tán và cải tiến phần mềm sẽ giúp đảm bảo độ tin cậy và hiệu quả khi số lượng thiết bị và dữ liệu tăng lên. Cuối cùng, việc kết nối hệ thống với các dịch vụ đám mây khác và phát triển các API mở để dễ dàng tích hợp với các hệ thống khác sẽ mở rộng khả năng ứng dụng và tạo ra nhiều giá trị hơn cho người dùng và các bên liên quan. Những hướng phát triển này không chỉ nâng cao hiệu quả và tính năng của hệ thống hiện tại mà còn mở rộng khả năng ứng dụng, tạo ra nhiều giá trị hơn cho người dùng và các bên liên quan.

1. **Tài liệu tham khảo**

*[1] Arduinokit. (2023, June 19). Sơ đồ chân ESP8266 NodeMCU. ARDUINO KIT.* [*https://arduinokit.vn/so-do-chan-esp8266-nodemcu/*](https://arduinokit.vn/so-do-chan-esp8266-nodemcu/)

*[2] Arduino - MG996R | Arduino getting started. (n.d.). Arduino Getting Started.* [*https://arduinogetstarted.com/tutorials/arduino-mg996r*](https://arduinogetstarted.com/tutorials/arduino-mg996r)

*[3] Santos, R., & Santos, R. (2024b, March 6). ESP8266 DHT11/DHT22 Web Server Arduino IDE | Random Nerd Tutorials. Random Nerd Tutorials. https://randomnerdtutorials.com/esp8266-dht11dht22-temperature-and-humidity-web-server-with-arduino-ide/*

*[4] Santos, R., & Santos, R. (2021, July 22). ESP8266 NodeMCU HC-SR04 Ultrasonic Sensor with Arduino IDE | Random Nerd Tutorials. Random Nerd Tutorials.* [*https://randomnerdtutorials.com/esp8266-nodemcu-hc-sr04-ultrasonic-arduino/*](https://randomnerdtutorials.com/esp8266-nodemcu-hc-sr04-ultrasonic-arduino/)

*[5] Santos, S., & Santos, S. (2022, April 23). ESP8266 NodeMCU with Load Cell and HX711 Amplifier (Digital Scale) | Random Nerd Tutorials. Random Nerd Tutorials. https://randomnerdtutorials.com/esp8266-load-cell-hx711/*

*[6] Thingsboard. (n.d.). ThingsBoard Community Edition. ThingsBoard.* [*https://thingsboard.io/docs/*](https://thingsboard.io/docs/)

**HẾT.**