

# MỤC LỤC

|  |    |
|--|----|
| MỤC LỤC .....                                      | 1  |
| DANH MỤC CÁC HÌNH.....                             | 3  |
| DANH MỤC CÁC THUẬT NGỮ TIẾNG ANH VÀ VIẾT TẮT ..... | 4  |
| LỜI MỞ ĐẦU.....                                    | 6  |
| CHƯƠNG 1: TỔNG QUAN VỀ BLE VÀ ESP32.....           | 7  |
| 1.1. Bluetooth Low Energy (BLE) .....              | 7  |
| 1.1.1. Định nghĩa .....                            | 7  |
| 1.1.2. Đặc điểm của BLE .....                      | 7  |
| 1.1.3. Kiến trúc giao thức BLE .....               | 8  |
| 1.1.4. Ứng dụng của BLE .....                      | 9  |
| 1.1.5. So sánh với Bluetooth truyền thống.....     | 9  |
| 1.1.6. Thông số kỹ thuật .....                     | 10 |
| 1.2. ESP32 .....                                   | 10 |
| 1.2.1. Định nghĩa .....                            | 10 |
| 1.2.2. Thông số kỹ thuật .....                     | 11 |
| 1.2.3. Đặc điểm .....                              | 11 |
| 1.3. Kết nối BLE với ESP32.....                    | 12 |
| 1.4. Kết luận chương .....                         | 12 |
| CHƯƠNG 2: MÔ HÌNH CÂN ĐIỆN TỬ.....                 | 13 |
| 2.1. Tổng quan về mô hình .....                    | 13 |
| 2.2. Module Loadcell .....                         | 13 |
| 2.2.1. Khái niệm .....                             | 13 |
| 2.2.2. Nguyên lý hoạt động.....                    | 13 |
| 2.2.3. Thông số kỹ thuật của loadcell .....        | 14 |
| 2.2.4. Các loại loadcell phổ biến hiện nay .....   | 15 |
| 2.2.5. Ứng dụng .....                              | 16 |
| 2.3. Module HX711.....                             | 17 |

|  |           |
|--|-----------|
| 2.3.1. Khái niệm .....   | 17        |
| 2.3.2. Chức năng .....   | 17        |
| 2.3.3. Ứng dụng .....  | 17        |
| 2.4. Module I2C LCD.....   | 18        |
| 2.4.1. Cơ bản về module LCD và I2C .....   | 18        |
| 2.4.1.1. Module LCD.....   | 18        |
| 2.4.1.2. Module I2C (PCF8574) .....  | 18        |
| 2.4.1.3. Kết nối giữa Module I2C PCF8574 và LCD.....   | 19        |
| 2.4.2. Thông số kỹ thuật .....   | 20        |
| 2.4.2.1. Thông số kỹ thuật Module I2C PCF7854 .....  | 20        |
| 2.4.2.2. Thông số kỹ thuật module LCD (16x2 hoặc 20x4): .....  | 21        |
| 2.4.3. Nguyên lý hoạt động.....  | 22        |
| 2.5. Chức năng hoạt động của cân điện tử.....  | 22        |
| 2.6. Kết luận chương .....   | 22        |
| <b>CHƯƠNG 3: THIẾT KẾ MÔ HÌNH CÂN ĐIỆN TỬ HIỂN THỊ GIÁ TRỊ KHỐI LƯỢNG TRÊN MÀN HÌNH MÁY TÍNH SỬ DỤNG KẾT NỐI BLE CỦA ESP32. ....</b> | <b>23</b> |
| 3.1. Sơ đồ khối.....   | 23        |
| 3.1.1. Khối dữ liệu vào.....   | 23        |
| 3.1.2. Khối xử lý .....  | 24        |
| 3.1.3. Khối hiển thị.....  | 24        |
| 3.2. Mô hình hệ thống .....  | 25        |
| 3.3. Các bước thực hiện .....  | 25        |
| 3.3.1. Code hệ thống .....   | 25        |
| 3.3.1.1. Giao diện Web.....  | 25        |
| 3.3.1.2. Vi điều khiển ESP32 .....   | 27        |
| 3.3.2. Mô phỏng.....   | 31        |
| 3.3.3. Kết quả đạt được .....  | 36        |
| 3.3.4. Nhận xét.....   | 36        |
| 3.4. Kết luận chương .....   | 37        |
| <b>CHƯƠNG 4: KẾT LUẬN CHUNG .....</b>  | <b>37</b> |
| <b>CHƯƠNG 5: TÀI LIỆU THAM KHẢO .....</b>  | <b>38</b> |

## DANH MỤC CÁC HÌNH

|   |    |
|---|----|
| Hình 1.1: BLE .....   | 7  |
| Hình 1.2: Kiến trúc giao thức BLE .....                           | 8  |
| Hình 1.3: Bảng so sánh BLE và Bluetooth truyền thống .....        | 10 |
| Hình 1.4: ESP32 .....   | 11 |
| Hình 1.5: Kết nối BLE và ESP32.....                               | 12 |
| Hình 2.1: Nguyên lý làm việc của loadcell .....                   | 14 |
| Hình 2.2: Loadcell phân loại theo phương hướng lực tác động ..... | 15 |
| Hình 2.3: Loadcell phân loại hình dạng.....                       | 15 |
| Hình 2.4: Loadcell phân loại theo tín hiệu truyền và nhận.....    | 16 |
| Hình 2.5: Module HX711 .....                                      | 17 |
| Hình 2.6: Module I2C(PCF8574) .....                               | 19 |
| Hình 2.7: Kết nối giữa Module I2C PCF8574 và LCD.....             | 20 |
| Hình 2.8: LCD16x2 .....   | 21 |
| Hình 3.1: Sơ đồ khối của mô hình cân điện tử.....                 | 23 |
| Hình 3.2: Mô hình phần cứng.....                                  | 25 |
| Hình 3.3: Lập trình xây dựng các khối cho trang web.....          | 26 |
| Hình 3.4: Thiết lập kiểu dáng cho các khối của trang web.....     | 26 |
| Hình 3.5: Thiết lập chức năng cho trang web .....                 | 27 |
| Hình 3.6: Code xác định thông số hiệu chuẩn của cân .....         | 28 |
| Hình 3.7: Lập trình cho HX711 và Loadcell .....                   | 29 |
| Hình 3.8: Lập trình chức năng cho nút nhấn.....                   | 29 |
| Hình 3.9: Lập trình cho màn hình LCD .....                        | 30 |
| Hình 3.10: Lập trình thiết lập kết nối BLE cho ESP32.....         | 30 |
| Hình 3.11: File chính chạy cả chương trình cho ESP32.....         | 31 |
| Hình 3.12: Giao diện Web .....                                    | 32 |
| Hình 3.13: Xác định thông số hiệu chuẩn.....                      | 32 |
| Hình 3.14: Kết nối Web với ESP32 bằng kết nối BLE .....           | 33 |
| Hình 3.15: Đặt quả cân lên đĩa cân và xem kết quả hiển thị .....  | 33 |
| Hình 3.16: Nhấn nút CALIB và xem kết quả hiển thị.....            | 34 |
| Hình 3.17: Nhấn nút HOLD và xem kết quả hiển thị.....             | 35 |
| Hình 3.18: Nhấn nút UNHOLD và xem kết quả hiển thị.....           | 35 |
| Hình 3.19: Tính giá tiền thực phẩm theo khối lượng .....          | 36 |

## DANH MỤC CÁC THUẬT NGỮ TIẾNG ANH VÀ VIẾT TẮT

| Từ viết tắt | Thuật ngữ tiếng Anh/Giải thích     | Thuật ngữ tiếng Việt/Giải thích                         |
|-------------|------------------------------------|---|
| AES         | Advanced Encryption Standard       | Mã hóa AES 128-bit                                      |
| ATT         | Attribute Protocol                 | Giao thức thuộc tính                                    |
| BLE         | Bluetooth Low Energy               | Chuẩn kết nối không dây tiết kiệm năng lượng            |
| CPU         | Central Processing Unit            | Bộ xử lý trung tâm                                      |
| CSS         | Cascading Style Sheets             | Ngôn ngữ định kiểu trang web                            |
| ESP32       | Embedded System Processor 32-bit   | Bộ vi điều khiển tích hợp Bluetooth và Wi-Fi chế độ kép |
| GAP         | Generic Access Profile             | Hồ sơ truy cập chung                                    |
| GATT        | Generic Attribute Profile          | Hồ sơ thuộc tính chung                                  |
| GPIO        | General Purpose Input/Output       | Ngõ vào/ra đa năng                                      |
| HCI         | Host Controller Interface          | Giao diện bộ điều khiển chủ                             |
| HX711       | IC for load cell signal conversion | Bộ chuyển đổi tín hiệu cho loadcell                     |

| <b>Từ viết tắt</b> | <b>Thuật ngữ tiếng Anh/Giải thích</b>        | <b>Thuật ngữ tiếng Việt/Giải thích</b>            |
|--------------------|--|---|
| I2C                | Inter-Integrated Circuit                     | Giao thức truyền thông giữa các vi mạch           |
| IoT                | Internet of Things                           | Mạng lưới vạn vật kết nối                         |
| JS                 | JavaScript                                   | Ngôn ngữ lập trình JavaScript                     |
| LCD                | Liquid Crystal Display                       | Màn hình hiển thị tinh thể lỏng                   |
| L2CAP              | Logical Link Control and Adaptation Protocol | Giao thức điều khiển liên kết logic và thích nghi |
| LL                 | Link Layer                                   | Lớp liên kết                                      |
| PCF8574            | I2C to GPIO interface chip                   | IC mở rộng GPIO thông qua giao tiếp I2C           |
| PHY                | Physical Layer                               | Lớp vật lý  |
| RTC                | Real-Time Clock                              | Đồng hồ thời gian thực                            |
| SM                 | Security Manager                             | Quản lý bảo mật                                   |
| ULP                | Ultra Low Power                              | Công suất cực thấp                                |

## LỜI MỞ ĐẦU

Trong thời đại công nghệ phát triển nhanh chóng, Internet of Things (IoT) đã trở thành một xu hướng chủ đạo, mang đến những ứng dụng vượt trội trong nhiều lĩnh vực, từ y tế, nông nghiệp cho đến sản xuất và đời sống hàng ngày. Một trong những công nghệ cốt lõi hỗ trợ cho IoT là kết nối không dây Bluetooth Low Energy (BLE). Với khả năng tiết kiệm năng lượng và hiệu quả cao, BLE ngày càng được tích hợp vào các thiết bị thông minh, giúp tăng cường khả năng kết nối và tương tác.

Báo cáo này tập trung vào việc xây dựng một mô hình cân điện tử hiển thị giá trị khối lượng trên màn hình máy tính, sử dụng kết nối BLE và vi điều khiển ESP32. Thông qua đề tài, nhóm mong muốn không chỉ nghiên cứu và ứng dụng BLE trong thực tế mà còn nâng cao hiểu biết về cách tích hợp các thành phần phần cứng và phần mềm để tạo nên một hệ thống IoT hoàn chỉnh và hiệu quả.

Báo cáo được chia thành các chương với nội dung từ giới thiệu công nghệ BLE và ESP32, cho đến thiết kế, triển khai và thử nghiệm mô hình. Qua quá trình thực hiện, nhóm hy vọng sẽ mang đến một cái nhìn tổng quan và chi tiết về tiềm năng ứng dụng của BLE trong các hệ thống IoT hiện đại, góp phần giải quyết các vấn đề thực tiễn và thúc đẩy sự phát triển của các thiết bị thông minh trong tương lai.

# CHƯƠNG 1: TỔNG QUAN VỀ BLE VÀ ESP32

## 1.1. Bluetooth Low Energy (BLE)

### 1.1.1. Định nghĩa

Bluetooth Low Energy (BLE) là một chuẩn kết nối không dây tiết kiệm năng lượng, được phát triển bởi tổ chức Bluetooth SIG (Special Interest Group). BLE được giới thiệu lần đầu tiên vào năm 2010 và đã nhanh chóng trở thành một trong những chuẩn kết nối không dây phổ biến nhất trên thế giới. BLE được thiết kế để hoạt động trong các ứng dụng IoT, như thiết bị y tế thông minh, thiết bị đeo tay, thiết bị định vị GPS và các thiết bị gia đình thông minh. Với khả năng tiết kiệm năng lượng, BLE cho phép các thiết bị hoạt động trong thời gian dài hơn, giúp tiết kiệm pin và tăng tuổi thọ của thiết bị.



*Hình 1.1: BLE*

### 1.1.2. Đặc điểm của BLE

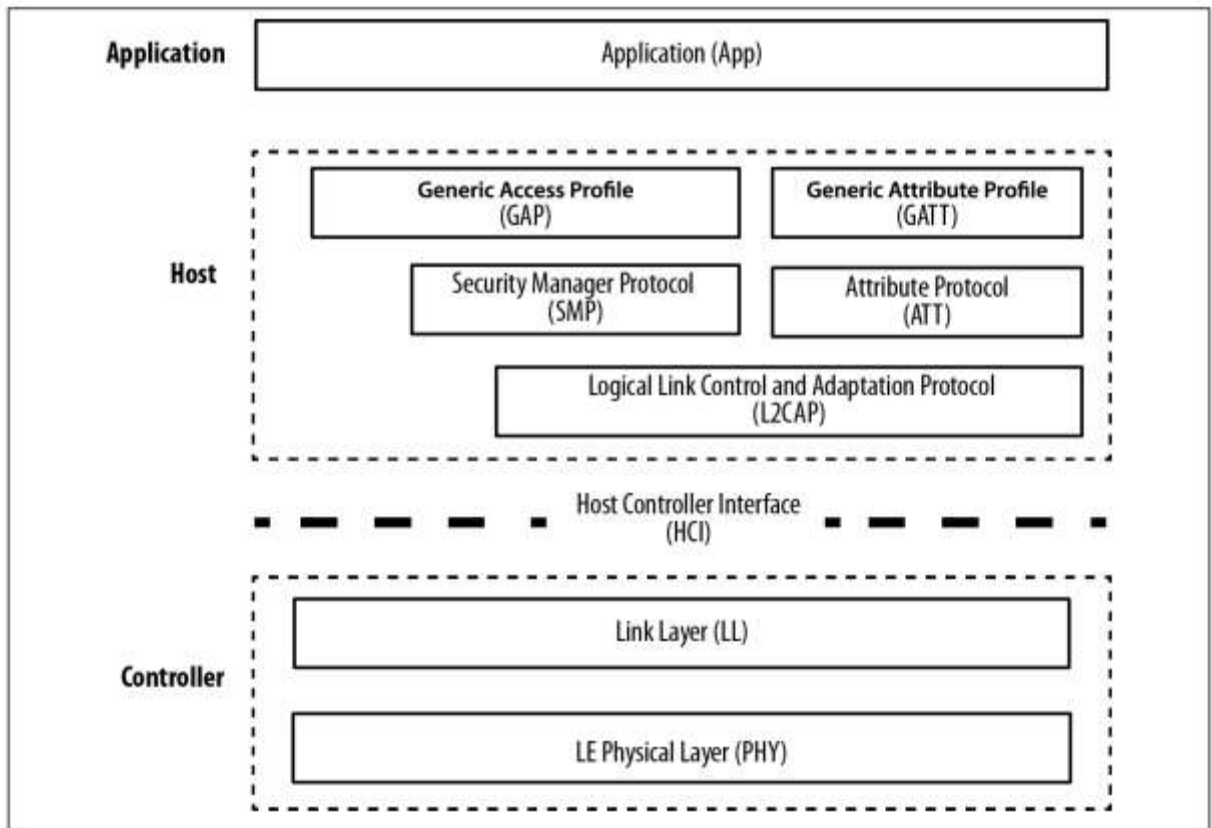
**Tiêu thụ năng lượng thấp:** BLE tối ưu hóa để tiêu thụ ít năng lượng hơn so với Bluetooth Classic, phù hợp cho các thiết bị nhỏ gọn với dung lượng pin hạn chế như đồng hồ thông minh, cảm biến sức khỏe, và các thiết bị IoT.

**Phạm vi giao tiếp:** Phạm vi của BLE tương tự như Bluetooth Classic, thường là khoảng từ 10 đến 100 mét, tùy thuộc vào môi trường và loại thiết bị.

**Tốc độ truyền dữ liệu:** BLE không được thiết kế để truyền tải dữ liệu lớn, nhưng đủ để truyền các gói dữ liệu nhỏ trong các ứng dụng băng thông thấp.

**Hỗ trợ đa nền tảng:** BLE được hỗ trợ trên nhiều hệ điều hành như iOS, Android, Windows Phone, macOS, Linux, Windows 8 và 10.

### 1.1.3. Kiến trúc giao thức BLE



**Hình 1.2: Kiến trúc giao thức BLE**

Bộ giao thức cho thiết bị BLE được chia thành 3 phần chính: controller, host và application. Mỗi phần bao gồm một hoặc nhiều lớp (layer) theo chức năng:

- **Application:** Là lớp cao nhất của bộ giao thức, cung cấp giao diện người dùng, xử lý logic, và điều khiển dữ liệu của mọi thứ liên quan đến các trường hợp hoạt động của ứng dụng. Kiến trúc của application phụ thuộc nhiều vào từng bài toán cụ thể.
- **Host:** bao gồm các lớp sau
  - Generic Access Profile (GAP)
  - Generic Attribute Profile (GATT)
  - Attribute Protocol (ATT)
  - Security Manager (SM)



- Logical Link Control and Adaptation Protocol (L2CAP)
- Host Controller Interface (HCI), Host side
- **Controller:** bao gồm các lớp sau
  - Host Controller Interface (HCI), Controller side
  - Link Layer (LL)
  - Physical Layer (PHY)

#### 1.1.4. Ứng dụng của BLE

- **Chăm sóc sức khỏe :** Kết nối các thiết bị y tế như máy đo nhịp tim, máy đo huyết áp.
- **Thể dục :** Ứng dụng trong các thiết bị theo dõi hoạt động và đồng hồ thông minh.
- **Beacon:** Sử dụng trong các hệ thống định vị trong nhà hoặc quảng cáo dựa trên vị trí.
- **An ninh:** Tích hợp vào các hệ thống khóa thông minh, cảm biến an ninh.
- **Giải trí gia đình:** Điều khiển các thiết bị âm thanh không dây, điều khiển từ xa.

#### 1.1.5. So sánh với Bluetooth truyền thống

Bluetooth truyền thống thích hợp để xử lý, truyền và trao đổi một lượng lớn dữ liệu (ví dụ như âm thanh, video). Tuy nhiên, nó tiêu tốn nhiều điện năng hơn và có chi phí cao hơn. Ngược lại, BLE phù hợp với các ứng dụng không yêu cầu chuyển một lượng lớn dữ liệu, do đó tiết kiệm năng lượng và cho phép thiết bị hoạt động nhiều năm chỉ với một lượng pin nhỏ, với chi phí thấp hơn vì BLE không cần kết nối liên tục.

Bảng so sánh dưới đây nêu bật các điểm khác biệt chính:

|                            | Classic Bluetooth            | Bluetooth BLE          |
|----------------------------|------------------------------|------------------------|
| Tần số vô tuyến            | 2.4G                         | 2.4G                   |
| Khoảng cách                | 10m                          | >60m                   |
| Tốc độ truyền dữ liệu      | 1-3Mbps                      | 1Mbps                  |
| Thông lượng                | 0.7-2.1Mbps                  | 305kbps                |
| Bảo mật                    | 64bit, 128bit                | 128-bit AES            |
| Mức độ tiêu thụ năng lượng | Low                          | Very Low               |
| Nhà sản xuất               | Bluetooth SIG                | Bluetooth SIG          |
| Cấu trúc liên kết mạng     | Point-to-Point<br>Scatternet | Point-to-Point<br>Star |

*Hình 1.3: Bảng so sánh BLE và Bluetooth truyền thống*

#### 1.1.6. Thông số kỹ thuật

- **Tần số vô tuyến:** 2.4 GHz.
- **Khoảng cách:** Hơn 60 mét.
- **Tốc độ truyền dữ liệu:** 1 Mbps.
- **Thông lượng:** 305 kbps.
- **Bảo mật:** Mã hóa AES 128-bit.
- **Mức độ tiêu thụ năng lượng:** Rất thấp.
- **Cấu trúc liên kết mạng:** Topology Star, kết nối điểm-điểm.

### 1.2. ESP32

#### 1.2.1. Định nghĩa

ESP32 là một vi điều khiển chi phí thấp và tiêu thụ điện năng thấp, tích hợp Bluetooth và Wi-Fi chế độ kép. Điều này giúp nó trở nên cực kỳ linh hoạt, mạnh mẽ và đáng tin cậy cho nhiều ứng dụng như IoT, robot và tự động hóa.



*Hình 1.4: ESP32*

### 1.2.2. Thông số kỹ thuật

- **Bộ xử lý:**
  - CPU: Bộ vi xử lý Xtensa lõi kép (hoặc lõi đơn) 32-bit LX6, hoạt động ở tần số 240 MHz (160 MHz cho ESP32-S0WD và ESP32-U4WDH) và hoạt động ở tối đa 600 MIPS (200 MIPS với ESP32-S0WD/ESP32-U4WDH)
  - Bộ đồng xử lý (co-processor) công suất cực thấp (Ultra low power, viết tắt: ULP)
  - Hệ thống xung nhịp: CPU Clock, RTC Clock và Audio PLL Clock
  - Bộ nhớ nội: 448 KB bộ nhớ ROM và 520 KB bộ nhớ SRAM
  - Kết nối không dây: Wi-Fi: 802.11 b/g/n và Bluetooth: v4.2 BR/EDR và BLE
  - 34 GPIO pad vật lý
- **Bảo mật:**
  - Hỗ trợ tất cả các tính năng bảo mật chuẩn IEEE 802.11, bao gồm WPA, WPA/WPA2 và WAPI.
  - Secure boot (tạm dịch: khởi động an toàn)
- **Mã hóa flash:**
  - 1024-bit OTP, lên đến 768-bit cho khách hàng
  - Tăng tốc mã hóa phần cứng: AES, SHA-2, RSA, elliptic curve cryptography

### 1.2.3. Đặc điểm

- Tích hợp Bluetooth và Wi-Fi chế độ kép.
- Tiêu thụ năng lượng thấp, lý tưởng cho các ứng dụng chạy bằng pin.

- Có nhiều chân I/O để hỗ trợ nhiều chức năng.
- Hệ thống quản lý năng lượng tiên tiến.

### 1.3. Kết nối BLE với ESP32

Kết nối BLE với ESP32 cho phép giao tiếp liền mạch giữa các thiết bị IoT. Bằng cách lập trình ESP32 thông qua các nền tảng như Arduino IDE, các nhà phát triển có thể tích hợp giao tiếp BLE vào các hệ thống IoT khác nhau. Các bước dưới đây mô tả quy trình này:

- Cài đặt các trình điều khiển và thư viện cần thiết (ví dụ: trình điều khiển USB CP210x UART để truyền thông).
- Sử dụng Arduino IDE để lập trình ESP32.
- Tải các thư viện BLE cho ESP32 trong Arduino IDE.
- Thiết lập kết nối BLE bằng cách sử dụng ESP32 để truyền dữ liệu.



*Hình 1.5: Kết nối BLE và ESP32*

### 1.4. Kết luận chương

Trong chương này, chúng ta đã tìm hiểu về các khái niệm cơ bản của Bluetooth Low Energy (BLE) và so sánh nó với công nghệ Bluetooth truyền thống. Chúng ta cũng đã thảo luận về ESP32, các đặc điểm của nó và cách kết nối ESP32 với BLE cho các ứng dụng IoT. Với khả năng tiết kiệm năng lượng của BLE và tính linh hoạt của ESP32, đây là giải pháp lý tưởng cho các hệ thống IoT hiện đại.

## **CHƯƠNG 2: MÔ HÌNH CÂN ĐIỆN TỬ**

### **2.1. Tổng quan về mô hình**

Mô hình cân điện tử sử dụng cảm biến loadcell để đo lường trọng lượng của vật thể và hiển thị kết quả thông qua module LCD. Loadcell, kết hợp với IC chuyển đổi tín hiệu HX711, cho phép chuyển đổi tín hiệu cơ học thành tín hiệu điện và truyền thông qua vi điều khiển ESP32. Thông tin trọng lượng sau đó được hiển thị qua màn hình LCD thông qua giao thức I2C và trên ứng dụng di động hoặc trang web thông qua kết nối Bluetooth Low Energy, tạo ra hệ thống cân điện tử đơn giản nhưng hiệu quả.

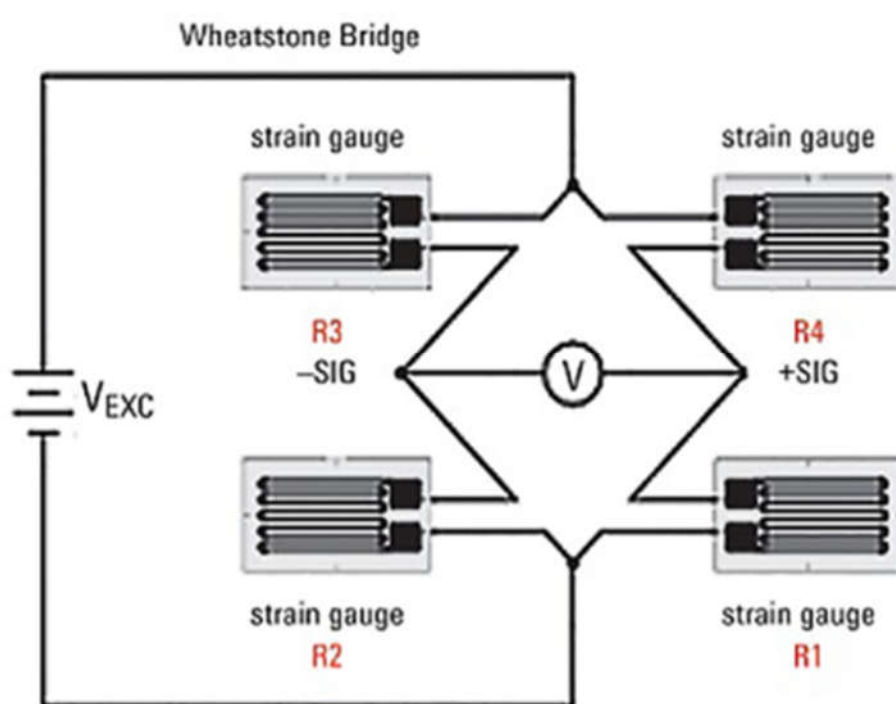
### **2.2. Module Loadcell**

#### **2.2.1. Khái niệm**

Loadcell là một loại cảm biến dùng để chuyển đổi lực hoặc trọng lượng thành tín hiệu điện. Tín hiệu này tỉ lệ với sự thay đổi điện trở trong cầu điện trở của cảm biến. Loadcell hoạt động dựa trên nguyên lý áp lực – trở kháng, khi lực tác động lên cảm biến làm thay đổi điện trở, dẫn đến thay đổi điện áp đầu ra.

#### **2.2.2. Nguyên lý hoạt động**

Nguyên lý hoạt động của loadcell dựa trên cầu điện trở Wheatstone. Khi có một lực tác động lên loadcell, nó làm thay đổi chiều dài của các sợi kim loại strain gauge dán trên thân loadcell, khiến giá trị điện trở thay đổi. Tín hiệu điện áp đầu ra thay đổi tương ứng, và được chuyển thành dữ liệu số thông qua bộ khuếch đại tín hiệu, thường là HX711.



*Hình 2.1: Nguyên lý làm việc của loadcell*

### 2.2.3. Thông số kỹ thuật của loadcell

Một số thông số kỹ thuật quan trọng của loadcell bao gồm:

- **Độ chính xác:** Thể hiện phần trăm sai số của phép đo, phụ thuộc vào độ phi tuyến tính, độ trễ và độ lặp lại.
- **Công suất định mức:** Trọng lượng lớn nhất mà loadcell có thể đo được.
- **Dải bù nhiệt độ:** Dải nhiệt độ mà loadcell có thể hoạt động mà không ảnh hưởng đến độ chính xác.
- **Cấp bảo vệ:** Đánh giá khả năng chống nước và bụi (thường được chỉ định bằng chỉ số IP).
- **Điện áp hoạt động:** Thường từ 5 đến 15V.
- **Trở kháng đầu vào và đầu ra:** Được xác định thông qua các cổng kết nối của loadcell.
- **Quá tải an toàn:** Công suất mà loadcell có thể chịu đựng mà không bị hư hỏng.

- **Hệ số tác động của nhiệt độ:** Ảnh hưởng của nhiệt độ lên công suất và kết quả đo của loadcell.

#### 2.2.4. Các loại loadcell phổ biến hiện nay

Có nhiều cách phân loại loadcell:

- **Phân loại theo phương hướng lực tác động:** Loadcell dạng nén, dạng uốn, dạng kéo, dạng xoắn.



*Hình 2.2: Loadcell phân loại theo phương hướng lực tác động*

- **Phân loại theo hình dáng:** Loadcell hình trụ, hình thanh, hình cầu, hình chữ Z, hình xoắn.



*Hình 2.3: Loadcell phân loại hình dáng*

- **Phân loại theo tín hiệu truyền và nhận:** Loadcell analog và loadcell digital.



*Hình 2.4: Loadcell phân loại theo tín hiệu truyền và nhận*

### 2.2.5. Ứng dụng

Loadcell được ứng dụng rộng rãi trong nhiều lĩnh vực như:

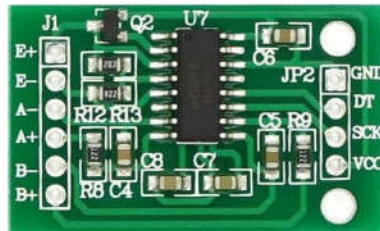
- **Cân điện tử:** Là thành phần chính trong các loại cân bàn, cân sàn, cân xe tải.
- **Dây chuyền sản xuất:** Giám sát trọng lượng của sản phẩm trong quá trình sản xuất tự động.
- **Hệ thống tự động hóa:** Được tích hợp để đo lường lực và trọng lượng trong quá trình sản xuất.
- **Kiểm tra vật liệu:** Đo độ bền và khả năng chịu tải của vật liệu trong ngành xây dựng và cơ khí.
- **Ứng dụng y tế:** Được sử dụng trong giường bệnh hoặc thiết bị đo lường lực cầm nắm trong quá trình phục hồi.
- **Ngành vận tải:** Đo lường trọng lượng hàng hóa trên xe tải để đảm bảo tuân thủ quy định về tải trọng.
- **Nghiên cứu khoa học:** Sử dụng trong các thí nghiệm vật lý và cơ học để đo lực và trọng lượng.



## 2.3. Module HX711

### 2.3.1. Khái niệm

HX711 là một IC chuyên đổi tín hiệu tương tự (analog) sang tín hiệu số (digital) với độ phân giải 24-bit. Nó được thiết kế để khuếch đại và chuyển đổi tín hiệu từ các cảm biến loadcell trong các ứng dụng cân điện tử và điều khiển công nghiệp.



*Hình 2.5: Module HX711*

### 2.3.2. Chức năng

HX711 có các chức năng chính sau:

- **Chuyển đổi tín hiệu:** Chuyển đổi tín hiệu điện áp tương tự từ loadcell thành tín hiệu số với độ phân giải cao.
- **Khuếch đại tín hiệu:** Khuếch đại tín hiệu điện áp thấp từ loadcell để dễ dàng xử lý bởi vi điều khiển.
- **Giao tiếp đơn giản:** Sử dụng giao thức hai dây (Clock và Data) để truyền dữ liệu đến vi điều khiển.
- **Điều chỉnh độ nhạy:** HX711 có thể điều chỉnh mức khuếch đại (gain) ở các mức 32, 64 và 128.
- **Tiết kiệm năng lượng:** HX711 tiêu thụ ít điện năng, phù hợp cho các ứng dụng cần tiết kiệm năng lượng.

### 2.3.3. Ứng dụng

HX711 được sử dụng trong nhiều ứng dụng khác nhau như:

- **Cân điện tử:** Đo lường trọng lượng trong các loại cân từ cân bàn đến cân xe tải.
- **Điều khiển công nghiệp:** Giám sát và điều khiển lực trong các quy trình sản xuất.
- **Thiết bị y tế:** Đo trọng lượng bệnh nhân hoặc lực cầm nắm trong quá trình phục hồi chức năng.

## 2.4. Module I2C LCD

### 2.4.1. Cơ bản về module LCD và I2C

#### 2.4.1.1. Module LCD

Module LCD (Liquid Crystal Display) là một màn hình hiển thị dựa trên tinh thể lỏng. Nó thường được sử dụng trong các dự án IoT, điện tử, và hệ thống nhúng để hiển thị thông tin như dữ liệu cảm biến, trạng thái hệ thống, hoặc các thông báo từ vi điều khiển.

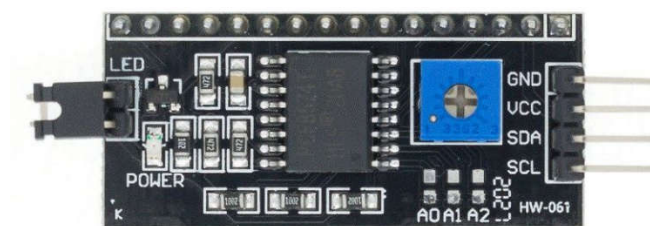
**Loại phổ biến:** LCD 16x2 (16 cột, 2 dòng) hoặc 20x4 (20 cột, 4 dòng).

**Chân kết nối:** Một module LCD truyền thống có nhiều chân, thường là 16 chân, cần kết nối trực tiếp với vi điều khiển. Điều này làm tốn nhiều chân GPIO (General Purpose Input/Output) của vi điều khiển.

Một số chân quan trọng của module LCD:

- VSS, VDD: Nguồn nuôi.
- RS (Register Select): Chọn thanh ghi.
- E (Enable): Kích hoạt LCD.
- D0-D7: Đường dữ liệu 8 bit.
- RW: Chọn chế độ đọc/ghi.
- LED+/-: Chân cấp nguồn cho đèn nền (nếu có).

#### 2.4.1.2. Module I2C (PCF8574)

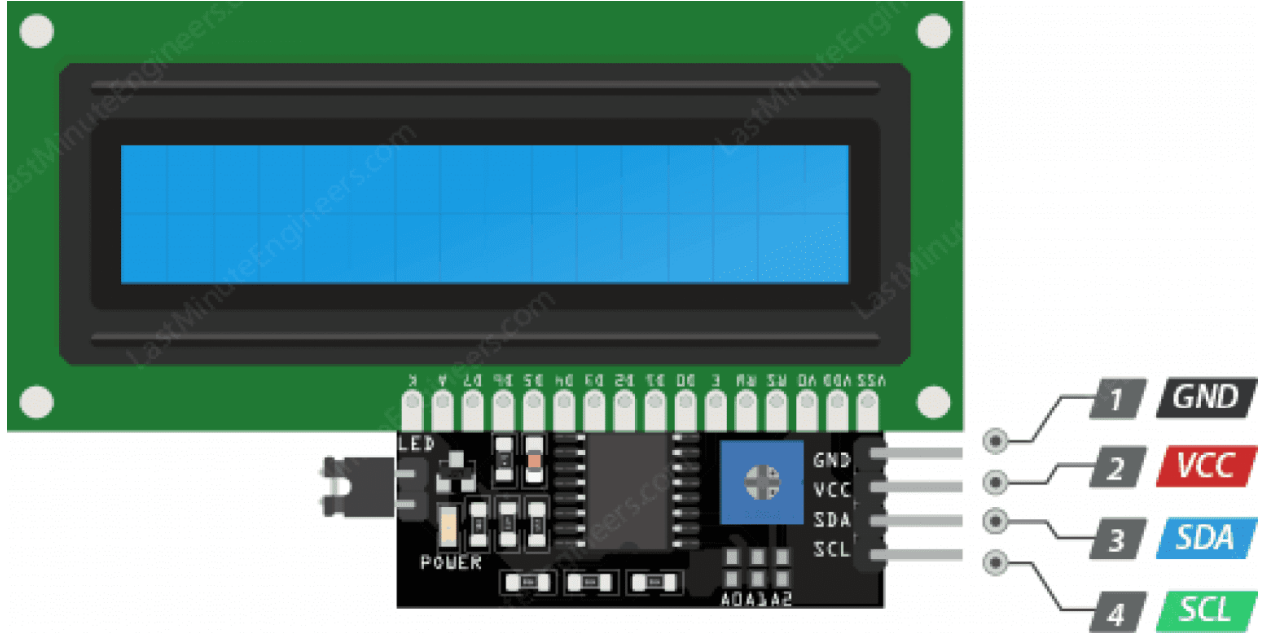


**Hình 2.6: Module I2C(PCF8574)**

Để giải quyết vấn đề cần nhiều chân kết nối giữa vi điều khiển và LCD, module I2C PCF8574 thường được sử dụng. Đây là một IC mở rộng GPIO thông qua giao tiếp I2C, giúp giảm số lượng dây kết nối.

I2C PCF8574 là một IC đóng vai trò chuyển đổi giữa giao tiếp I2C và các chân GPIO tương tự như trên module LCD. Nhờ có PCF8574, bạn chỉ cần sử dụng 2 chân (SDA và SCL) để giao tiếp với LCD thay vì 8 chân dữ liệu và nhiều chân điều khiển.

#### **2.4.1.3. Kết nối giữa Module I2C PCF8574 và LCD**



**Hình 2.7: Kết nối giữa Module I2C PCF8574 và LCD**

Việc kết nối module I2C PCF8574 với LCD rất đơn giản, IC này sẽ điều khiển các chân của LCD thông qua giao tiếp I2C.

- SDA của PCF8574 sẽ nối với SDA của vi điều khiển.
- SCL của PCF8574 sẽ nối với SCL của vi điều khiển.
- Các chân còn lại của PCF8574 sẽ kết nối với các chân điều khiển của LCD như RS, E, và D4-D7 (kết nối ở chế độ 4-bit).

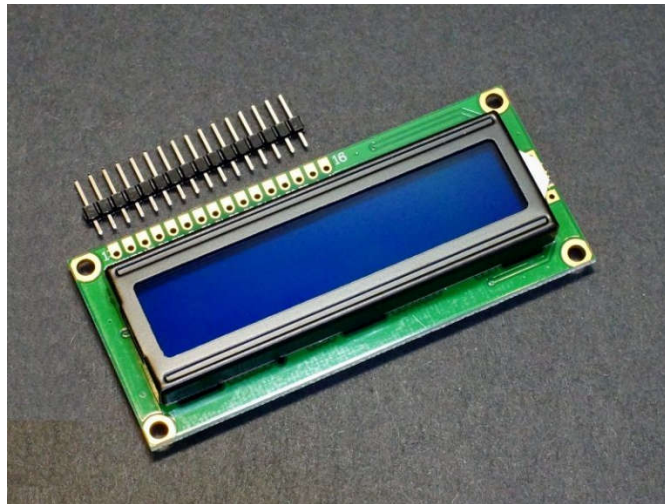
Kết quả là, thay vì sử dụng nhiều chân điều khiển trên vi điều khiển, bạn chỉ cần sử dụng hai chân I2C, giúp tiết kiệm tài nguyên và dễ dàng phát triển các ứng dụng phức tạp hơn.

## 2.4.2. Thông số kỹ thuật

### 2.4.2.1. Thông số kỹ thuật Module I2C PCF7854

- **Giao thức:** I2C (SDA, SCL).
- **Địa chỉ I2C:** 0x20-0x27 (có thể thay đổi).
- **Chân I/O:** 8 chân, có thể cấu hình.
- **Điện áp hoạt động:** 2.5V - 6V.
- **Dòng tiêu thụ:** 100  $\mu$ A (chờ), tối đa 1 mA (hoạt động).
- **Tốc độ I2C:** 100kHz (chuẩn), 400kHz (nhẹ).

#### 2.4.2.2. Thông số kỹ thuật module LCD (16x2 hoặc 20x4):



**Hình 2.8: LCD16x2**

- **Kích thước:** 16x2 hoặc 20x4 ký tự.
- **Điện áp hoạt động:** 5V
- **Dòng điện cấp nguồn:** 350 $\mu$ A - 600 $\mu$ A
- **Kích thước của màn hình LCD 1602:** 8 x 3.6 x 0.8 cm
- **Màu nền:** xanh lá hoặc xanh dương
- **Màu chữ:** Màu đen
- **Khoảng cách giữa hai chân kết nối:** 0.1 inch
- **Nhiệt độ hoạt động:** - 30 - 75 độ C

### **2.4.3. Nguyên lý hoạt động**

Module I2C LCD hoạt động bằng cách nhận tín hiệu từ vi điều khiển qua giao thức I2C. Vi điều khiển gửi lệnh và dữ liệu thông qua SDA và SCL, và module LCD hiển thị dữ liệu trên màn hình. Giao thức I2C cho phép dễ dàng kết nối và điều khiển nhiều thiết bị chỉ với hai chân tín hiệu.

### **2.5. Chức năng hoạt động của cân điện tử**

Hệ thống cân điện tử hoạt động dựa trên nguyên tắc đo lường trọng lượng thông qua cảm biến loadcell và hiển thị giá trị trọng lượng trên màn hình LCD. Quá trình bao gồm các bước:

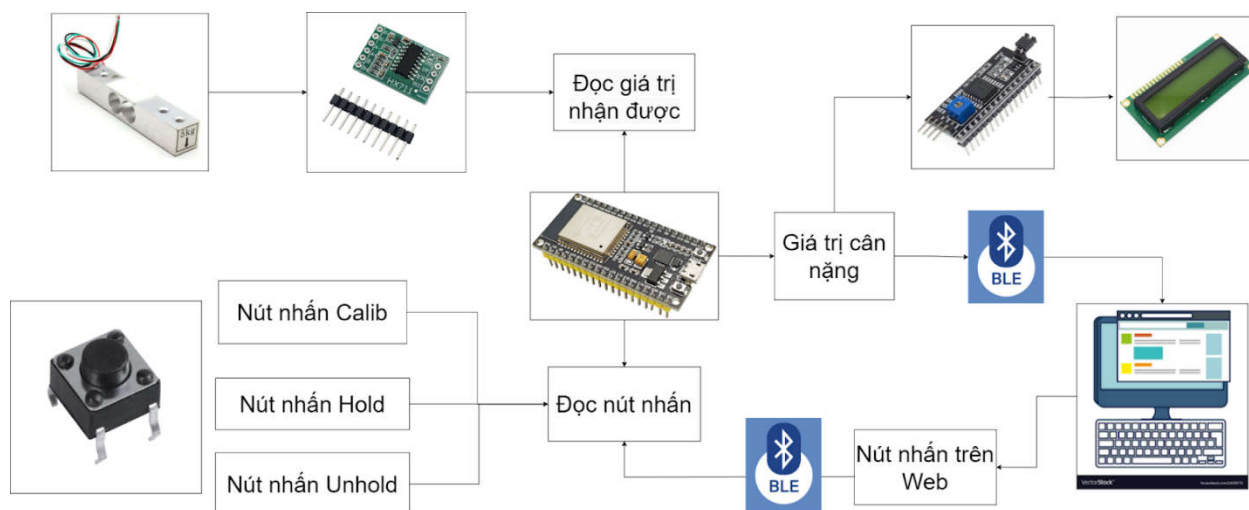
- Loadcell đo trọng lượng vật thể và chuyển đổi thành tín hiệu điện.
- Tín hiệu từ loadcell được HX711 khuếch đại và chuyển đổi thành dữ liệu số.
- ESP32 nhận dữ liệu từ HX711, tính toán trọng lượng và gửi lệnh hiển thị tới module LCD qua giao thức I2C.
- Module LCD hiển thị kết quả trọng lượng trên màn hình.

### **2.6. Kết luận chương**

Chương này đã giới thiệu chi tiết về mô hình cân điện tử, bao gồm nguyên lý hoạt động của cảm biến loadcell, module HX711 và module I2C LCD. Hệ thống cho phép đo lường trọng lượng chính xác và hiển thị dữ liệu lên màn hình, tạo nên một mô hình cân điện tử đơn giản và hiệu quả cho các ứng dụng thực tế.

## CHƯƠNG 3: THIẾT KẾ MÔ HÌNH CÂN ĐIỆN TỬ HIỂN THỊ GIÁ TRỊ KHỐI LƯỢNG TRÊN MÀN HÌNH MÁY TÍNH SỬ DỤNG KẾT NỐI BLE CỦA ESP32.

### 3.1. Sơ đồ khối



**Hình 3.1: Sơ đồ khối của mô hình cân điện tử**

Sơ đồ khối trên gồm 3 thành phần chính:

- Khối dữ liệu đầu vào gồm: Loadcell, HX711 và nút nhấn
- Khối xử lý: Vi điều khiển ESP32
- Khối hiển thị: LCD I2C và giao diện web

#### 3.1.1. Khối dữ liệu vào

Cảm biến cân nặng Loadcell để tạo ra giá trị thay đổi khi có lực tác động hoặc đo vật nặng, giá trị đó là giá trị tương tự. Sau đó, nó được đi qua module HX711 để chuyển đổi sang giá trị số.

Nút nhấn Calib có chức năng hiệu chuẩn lại giá trị cân nặng của mô hình. Ví dụ: khi ta cân lượng gạo cần lấy, ta sẽ đo cái cốc để chứa gạo, sau đó ta nhấn nút Calib để

hiệu chỉnh lại giá trị cân nặng và giá trị cân của nó là 0g, khi đó ta đổ gạo vào cốc và xem giá trị cân nặng phù hợp với lượng gạo cần lấy.

Nút nhấn Hold có chức năng giữ giá trị cân hiện tại đang hiển thị trên màn hình. Khi nó đang trạng thái giữ, giá trị cân hiện tại bị đứng yên khi hiển thị trên màn hình, kể cả khi ta tác động vào cân hoặc bỏ vật ra khỏi cân thì giá trị hiển thị không đổi.

Nút nhấn Unhold có chức năng là thoát khỏi trạng thái giữ giá trị cân đang hiển thị trên màn hình, trở về trạng thái bình thường.

### **3.1.2. Khối xử lý**

ESP32 được sử dụng để xử lý các chức năng từ hai khối còn lại.

Đối với khối tạo dữ liệu đầu vào, ESP32 sẽ đọc giá trị nhận được từ HX711 và xử lý con số để tạo ra giá trị cân nặng với đơn vị là gram. Ngoài ra, ESP32 sử dụng lệnh ngắt theo thời gian để đọc nút nhấn và thực hiện chức năng theo yêu cầu từ nút nhấn. Tương tự với nút nhấn trên giao diện web, ESP32 nhận được bản tin gửi về từ nút nhấn và thực hiện chức năng.

Đối với khối hiển thị, ESP32 dùng giao thức I2C để điều khiển màn hình theo ý muốn. Ngoài ra, ESP32 có chức năng Bluetooth Low Energy để giao tiếp với trang web hiển thị. Nó có thể gửi dữ liệu lên trang web và nhận dữ liệu từ trang web đó.

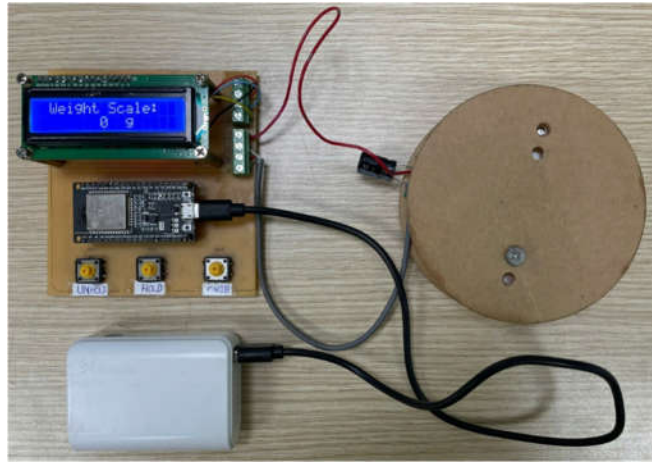
### **3.1.3. Khối hiển thị**

Sử dụng module I2C PCF8574 để giao tiếp với màn hình LCD và hiển thị quá trình và cân nặng theo mong muốn.

Còn giao diện web thì nhóm thiết kế phần hiển thị giá trị cân nặng và giá tiền khi ta chọn mua thực phẩm được tính theo cân. Bằng việc sử dụng kết nối BLE, giao diện web sẽ nhận được giá trị cân nặng và hiển thị.



### 3.2. Mô hình hệ thống



*Hình 3.2: Mô hình phần cứng*

### 3.3. Các bước thực hiện

#### 3.3.1. Code hệ thống

##### 3.3.1.1. Giao diện Web

Nhóm sử dụng HTML, CSS và JS để xây dựng trang web.

**File index.html**



```

// Cài đặt các biến và tham số
const connectButton = document.getElementById('connectBleButton');
const disconnectButton = document.getElementById('disconnectBleButton');
const received_weight_value = document.getElementById('weight_value_label');
const bleStateContainer = document.getElementById('bleState');
const Calib_Button = document.getElementById('button_calib');
const Hold_Button = document.getElementById('button_hold');
const Unhold_Button = document.getElementById('button_unhold');

var deviceName = 'ESP32';
var bleService = '19b10000-e8f2-537e-4f6c-d104768a1214';
var buttonCharacteristic = '19b10002-e8f2-537e-4f6c-d104768a1214';
var LoadCell_Characteristic = '19b10001-e8f2-537e-4f6c-d104768a1214';

var Calib_Command = 0;
var Hold_Command = 1;
var Unhold_Command = 2;

var bleServer;
var bleServiceFound;
var sensorCharacteristicFound;

var getWeighingResults = 0; //Biến lưu giá trị cân theo gram
var oldWeightResults = 1; // Biến lưu giá trị cân cũ theo gram
var Circular_Progress_Bar_Val = 0; // Biến lưu giá trị cho biểu đồ tròn

received_weight_value.innerHTML = getWeighingResults + " kg";
document.getElementById("price").innerHTML = "Giá: " + getWeighingResults + " VND";

// Nút nhấn kết nối BLE
connectButton.addEventListener('click', (event) => {
    if (isWebBluetoothEnabled()){
        connectToDevice();
    }
});

```

*Hình 3.5: Thiết lập chức năng cho trang web*

### 3.3.1.2. Vi điều khiển ESP32

Vi điều khiển ESP32 là khối xử lý cho hệ thống, nhóm lần lượt lập trình ESP32 với các thành phần phần cứng được sử dụng và thiết lập kết nối BLE với trang web.

**File Loadcell\_HX711\_Calib.ino**

```

// Calibrating the load cell
#include <Arduino.h>
#include "HX711.h"

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 16;
const int LOADCELL_SCK_PIN = 4;

HX711 scale;

void setup() {
  Serial.begin(115200);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
}

void loop() {

  if (scale.is_ready()) {
    scale.set_scale();
    Serial.println("Tare... remove any weights from the scale.");
    delay(5000);
    scale.tare();
    Serial.println("Tare done...");
    Serial.print("Place a known weight on the scale...");
    delay(5000);
    long reading = scale.get_units();
    Serial.print("Result: ");
    Serial.println(reading);

  }
  else {
    Serial.println("HX711 not found.");
  }
  delay(1000);
}

//calibration factor will be the (reading)/(known weight)

```

**Hình 3.6: Code xác định thông số hiệu chuẩn của cân**

**File HX711\_Loadcell.cpp**

```

#include <Arduino.h>
#include <HX711.h>
#include "LCD.h"

// Kết nối HX711 với ESP32
#define LOADCELL_DOUT_PIN 16 // Chân Data kết nối GPIO16
#define LOADCELL_SCK_PIN 4 // Chân CLK kết nối GPIO4

HX711 scale; // Đặt tên cho lớp HX711 là scale

// Thông số hiệu chuẩn cân
float CALIBRATION_FACTOR = 390.0171;

void Setup_HX711(){
  LCD_Print_Setup();
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN); // Khởi tạo cảm biến lực hoạt động
  scale.set_scale(CALIBRATION_FACTOR); //thêm thông số hiệu chuẩn cân
  scale.tare(); //Hiệu chuẩn cân về 0
  delay(2000);
  clearRow(0);
  clearRow(1);
}

int HX711_Get_Value(){
  if (scale.wait_ready_timeout(100)){
    double weight = scale.get_units();
    return _round(weight);
  }
  else {
    Serial.println("HX711 not found.");
    return 0;
  }
}

void Calibrating(){
  scale.tare();
}

```

**Hình 3.7: Lập trình cho HX711 và Loadcell**

## File button.cpp

```

#include <Arduino.h>
#include "HX711_Loadcell.h"

// nút nhấn
#define BUTTON_CALIB_PIN 12
#define BUTTON_HOLD_PIN 14
#define BUTTON_UNHOLD_PIN 27

bool calib_flag = 0;
bool hold_flag = 0;
bool print_flag = 1;

hw_timer_t * timer = NULL;
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;

void Setup_Btn(){
  pinMode(BUTTON_CALIB_PIN, INPUT_PULLUP);
  pinMode(BUTTON_HOLD_PIN, INPUT_PULLUP);
  pinMode(BUTTON_UNHOLD_PIN, INPUT_PULLUP);
}

void IRAM_ATTR onTimer(){
  portENTER_CRITICAL_ISR(&timerMux); //vào chế độ tránh xung đột

  // Đọc trạng thái của các nút nhấn
  int currentButtonCalibState = digitalRead(BUTTON_CALIB_PIN);
  int currentButtonHoldState = digitalRead(BUTTON_HOLD_PIN);
  int currentButtonUnholdState = digitalRead(BUTTON_UNHOLD_PIN);
  if(currentButtonCalibState == LOW) calib_flag = 1;
  else if(currentButtonHoldState == LOW) hold_flag = 1;
  else if(currentButtonUnholdState == LOW) hold_flag = 0;

  portEXIT_CRITICAL_ISR(&timerMux); // thoát
}

void Set_Timer_Interrupt(){
  // khởi tạo timer với chu kì 1us vì thạch anh của ESP chạy 80MHz
  timer = timerBegin(0,80,true);
}

```

**Hình 3.8: Lập trình chức năng cho nút nhấn**

## File LCD.cpp

```
#include <Arduino.h>
#include <LiquidCrystal_I2C.h> // thư viện cho module I2C và LCD

/*
Kết nối LCD I2C với ESP32
VCC      VIN
GND      GND
SCL      D22
SDA      D21
*/

// Thiết lập địa chỉ của I2C cho LCD là 0x27 với 16 cột và 2 hàng
LiquidCrystal_I2C lcd(0x27,16,2);

void Setup_LCD(){
    lcd.clear();
    lcd.init();
    lcd.noBacklight();
    delay(1000);
    lcd.backlight(); // Make sure backlight is on
    lcd.home();
}

void LCD_Print_Weight(int weight)
{
    clearRow(1);
    lcd.setCursor(1, 0);
    lcd.print("Weight Scale:");

    if((weight >= 1000 && weight <= 5000) or (weight >= -5000 && weight <= -1000)){
        lcd.setCursor(4,1);
        lcd.print(weight / 1000);
        lcd.setCursor(5,1);
        lcd.print(".");
        lcd.setCursor(6,1);
        lcd.print("kg");

        if(weight % 1000 < 10){

```

**Hình 3.9: Lập trình cho màn hình LCD**

## File BLE.cpp

```
#include <Arduino.h>
#include <BLEDevice.h>
#include <BLEServer.h>
#include <BLEUtils.h>
#include <BLE2902.h>
#include "button.h"

#define SERVICE_UUID "19b10000-e8f2-537e-4f6c-d104768a1214"
#define WEIGHT_SCALE_CHARACTERISTIC_UUID "19b10001-e8f2-537e-4f6c-d104768a1214"
#define BUTTON_CHARACTERISTIC_UUID "19b10002-e8f2-537e-4f6c-d104768a1214"

#define CALIB_COMMAND 0
#define HOLD_COMMAND 1
#define UNHOLD_COMMAND 2

// Thiết lập thành phần cho Bluetooth low energy
BLEServer* pServer = NULL;
BLECharacteristic* pWeight_Scale_Characteristic = NULL;
BLECharacteristic* pButton_Characteristic = NULL;
bool deviceConnected = false;
bool oldDeviceConnected = false;

class MyServerCallbacks: public BLEServerCallbacks {
    void onConnect(BLEServer* pServer) {
        deviceConnected = true;
    };

    void onDisconnect(BLEServer* pServer) {
        deviceConnected = false;
    }
};

class MyCharacteristicCallbacks : public BLECharacteristicCallbacks {
    void onWrite(BLECharacteristic* pButton_Characteristic) {
        std::string value = pButton_Characteristic->getValue();
        if(value.length() > 0) {
            Serial.print("Characteristic event, written: ");
            Serial.println(static_cast<int>(value[0]));

```

**Hình 3.10: Lập trình thiết lập kết nối BLE cho ESP32**

## File main.cpp

```

#include <Arduino.h>
#include "BLE.h"
#include "HX711_Loadcell.h"
#include "LCD.h"
#include "button.h"

// Biến lưu giá trị cũ của cân.
int old_weight = 1;
void Show_Weight();

void setup() {
  Serial.begin(9600);
  Setup_Btn();
  Setup_LCD();
  Setup_BLE_Connect();
  Setup_HX711();
  Set_Timer_Interrupt();
}

void loop() {
  Disconnect_To_Web();
  Connect_To_Web();
  Show_Weight();
}

void Show_Weight(){
  Hold_Action();
  Calib_Action();
  int weight = HX711_Get_Value();
  if(weight != old_weight and is_print()){
    old_weight = weight;
    LCD_Print_Weight(weight);
    WEB_Print_Weight(weight);
  }
}

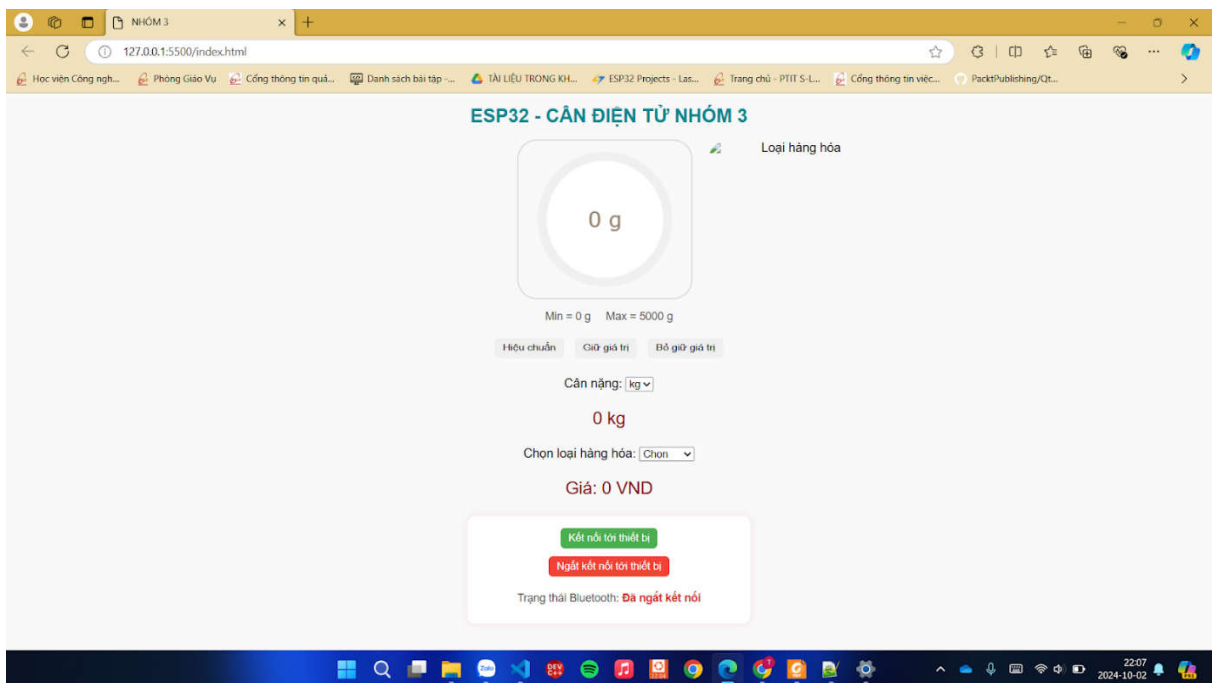
```

***Hình 3.11: File chính chạy cả chương trình cho ESP32***

### **3.3.2. Mô phỏng**

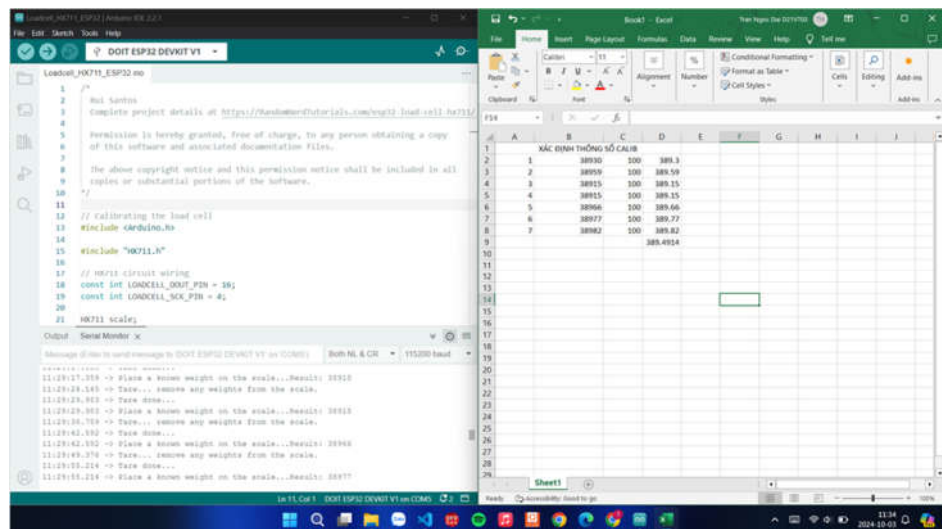
Chạy trang web





**Hình 3.12: Giao diện Web**

Xác định thông số hiệu chuẩn của cân

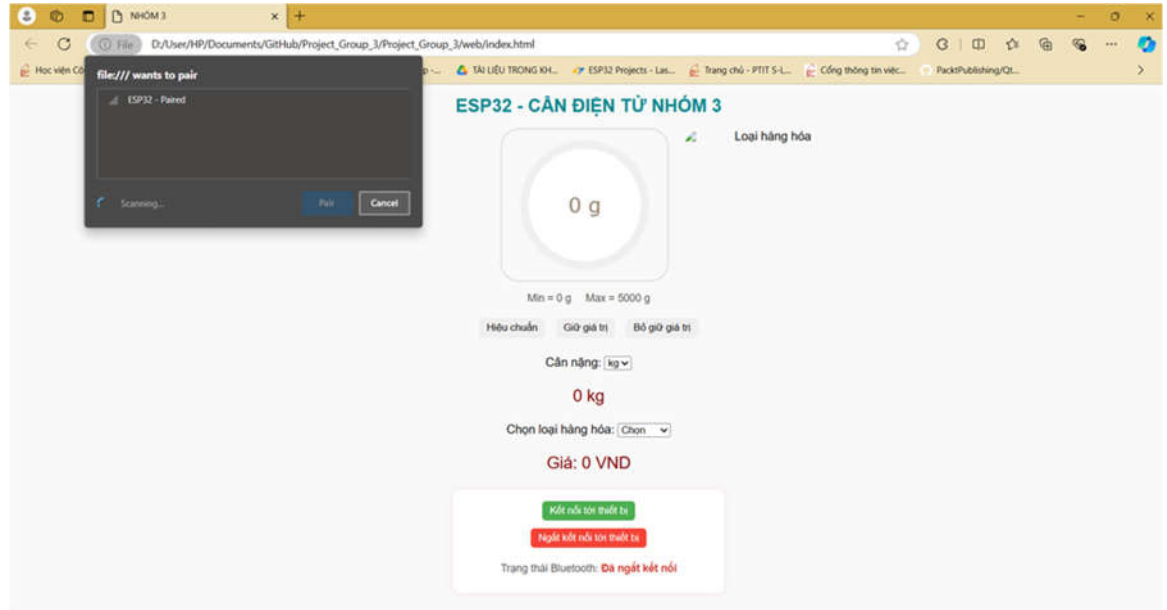


**Hình 3.13: Xác định thông số hiệu chuẩn**

Ta cắm nguồn cho mô hình, nạp code chính cho ESP32 và nó bắt đầu cấu hình cho phần cứng, cấu hình kết nối và bắt đầu việc hiển thị khối lượng của vật.

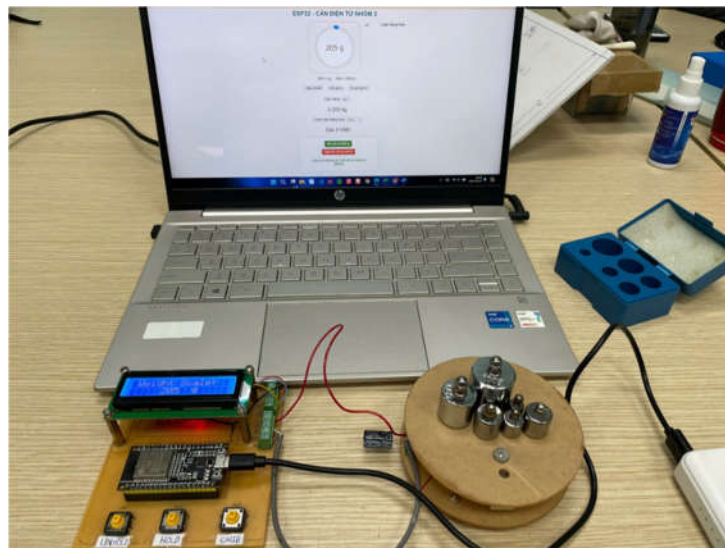


Ta mở trang web, nhấn nút kết nối tới thiết bị để quét tìm thiết bị phát kết nối BLE, ta thấy thiết bị tên ESP32 đang phát và ta chọn để kết nối.



**Hình 3.14: Kết nối Web với ESP32 bằng kết nối BLE**

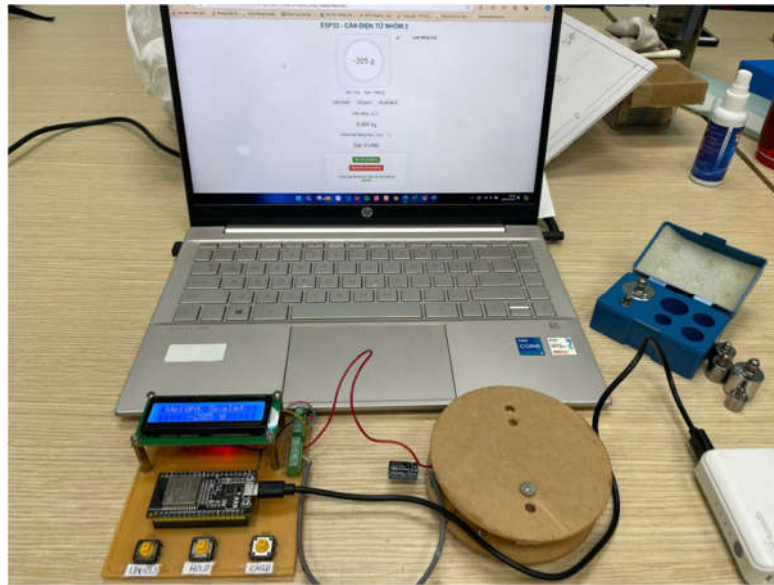
Đặt quả cân lên mặt cân, quả cân đó đã được biết trước khối lượng, nên ta kiểm tra xem khi đặt lên thì nó có hiện ra kết quả giống với khối lượng cho sẵn.



**Hình 3.15: Đặt quả cân lên đĩa cân và xem kết quả hiển thị**

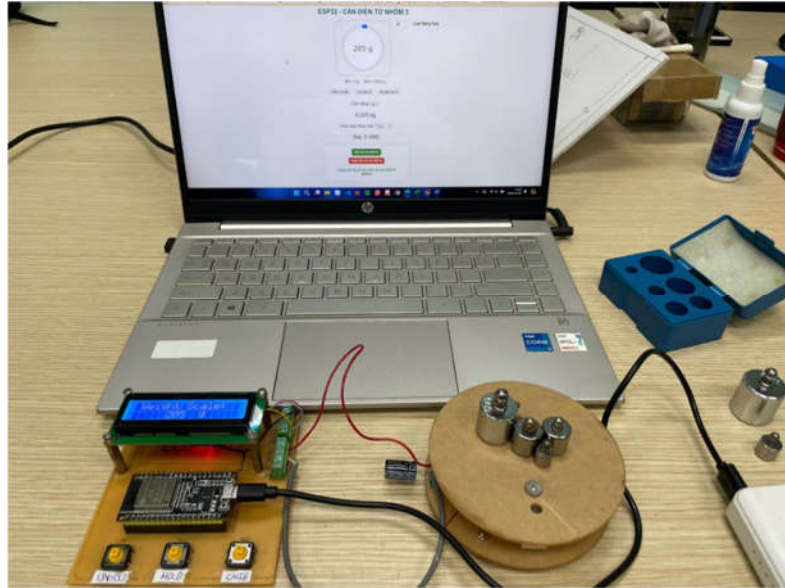
Như vậy màn hình web và LCD đã thể hiện đúng giá trị cân của quả cân khi đặt lên đĩa. Bây giờ ta thử tính năng của nút nhấn.

Ta nhấn nút CALIB, ESP32 sẽ hiệu chuẩn lại giá trị cân và sẽ đưa nó về giá trị 0 trên màn hình, ta bỏ các quả cân ra khỏi đĩa lúc đó và nó trở về giá trị -205g.



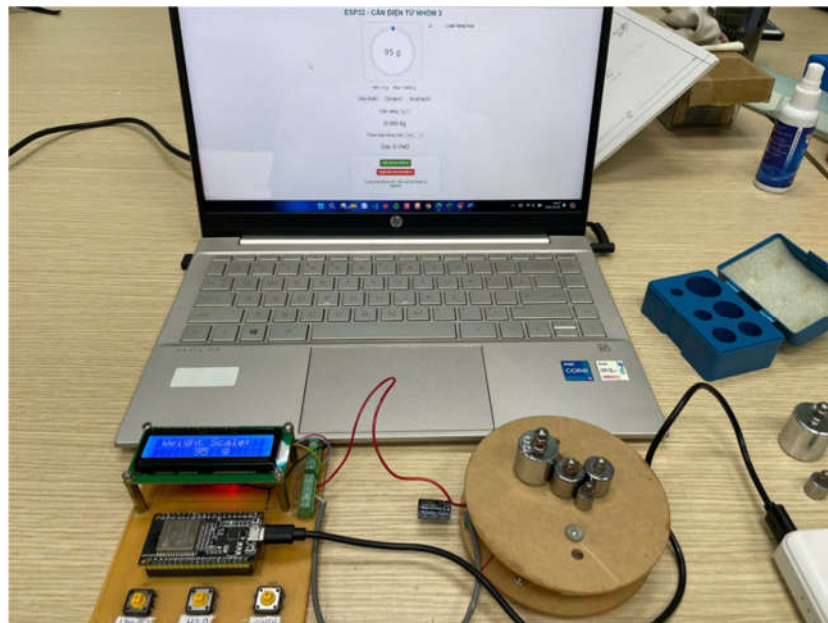
**Hình 3.16: Nhấn nút CALIB và xem kết quả hiển thị**

Tiếp đến, ta hiệu chuẩn cân và không đặt gì lên cân để cho màn hình giá trị 0g. Ta đặt các quả cân lên đĩa để hiển thị khối lượng. Sau đó, ta nhấn nút HOLD và bỏ một ít quả cân ra khỏi đĩa, lúc đó giá trị cân hiển thị không đổi.



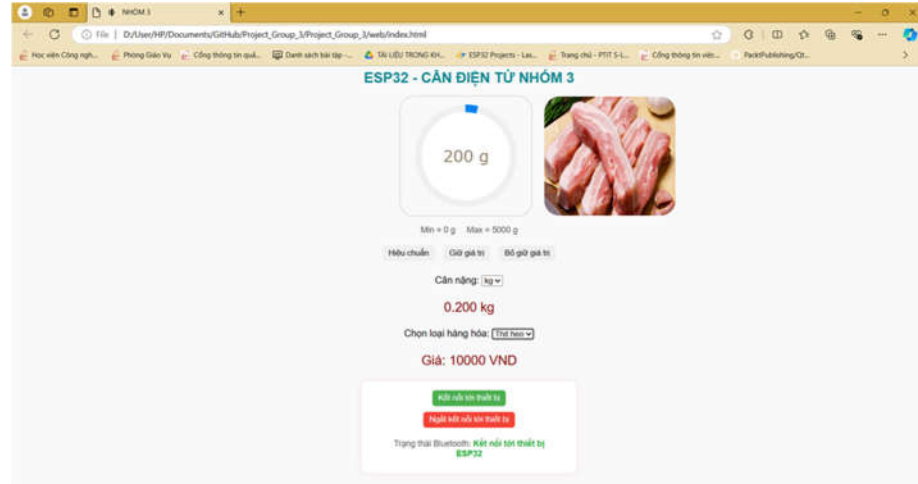
**Hình 3.17: Nhấn nút *HOLD* và xem kết quả hiển thị**

Để thoát khỏi trạng thái giữ giá trị của cân, ta nhấn nút UNHOLD để thoát khỏi trạng thái giữ giá trị, lúc đó giá trị cân hiển thị thay đổi và ta cân khối lượng các vật khác như bình thường.



**Hình 3.18: Nhấn nút *UNHOLD* và xem kết quả hiển thị**

Ở trên trang web, ta có thể tính giá tiền của các loại thực phẩm tính theo khối lượng.



*Hình 3.19: Tính giá tiền thực phẩm theo khối lượng*

### 3.3.3. Kết quả đạt được

Như vậy, mô hình đã chạy hoàn toàn ổn định. Giá trị cân nặng của vật mẫu được hiển thị chính xác trên màn hình LCD và giao diện web. Các nút nhấn chức năng ở ngoài và trên trang web hoạt động đúng theo chức năng được nhóm đề ra. Trang web được xây dựng với giao diện dễ nhìn và cụ thể, giúp cho ta xem được giá trị cân nặng, giá tiền của sản phẩm và có thể kết nối và ngắt kết nối BLE khi cần. Trang web và ESP32 trao đổi với nhau thông qua kết nối BLE hoạt động ổn định kể cả khi ta đặt hai phần ở khoảng cách xa.

### 3.3.4. Nhận xét

Qua các bước thực hiện phía trên, mô hình của nhóm đã chạy thành công. Có thể thấy, phần lập trình cho từng khối được viết một cách cụ thể và tường minh, quá trình mô phỏng hệ thống đã thể hiện rõ quy trình hoạt động của mô hình và các chức năng sử dụng trong đấy. Ngoài ra, quy trình thực hiện đã thể hiện rõ cho quá trình ghép nối các khối với nhau để tạo thành một hệ thống hoàn chỉnh, giúp người đọc người làm hiểu rõ các bước thiết lập phần cứng, thiết lập kết nối và chạy phần mềm.

### 3.4. Kết luận chương

Trong chương này, nhóm đã xây dựng thành công mô hình cân điện tử hiển thị giá trị khối lượng trên màn hình máy tính sử dụng kết nối BLE của ESP32. Mô hình này có thể sử dụng thực tế trong gia đình hoặc các sạp hàng bán thực phẩm, vì nó phù hợp với nhu cầu với việc đo đạc khối lượng của thực phẩm. Ngoài ra, việc sử dụng thêm kết nối BLE là giải pháp thay thế cho bluetooth truyền thông hoặc wifi trong việc kết nối trong phạm vi hẹp, nó giúp tiết kiệm năng lượng cho hệ thống, phù hợp với dung lượng dữ liệu thấp và khả năng kết nối ổn định.

## CHƯƠNG 4: KẾT LUẬN CHUNG

Nhóm đã thành công trong việc xây dựng hệ thống cân điện tử hiển thị giá trị khối lượng trên màn hình máy tính sử dụng kết nối BLE của ESP32. Quá trình làm việc của nhóm bắt đầu từ việc tìm hiểu về kết nối BLE, tìm hiểu về các thành phần cứng được sử dụng, sau đó thiết kế sơ đồ khối tổng quát của hệ thống, thiết kế mô hình phần cứng cho chúng. Tiếp đó, nhóm lập trình từng khối của hệ thống, lập trình ESP32 với các module phần cứng, thiết lập kết nối BLE cho ESP32, xây dựng giao diện web để hiển thị và kết nối đến ESP32. Quá trình mô phỏng đã thể hiện rõ kết quả của sản phẩm đúng theo yêu cầu, chức năng mà nhóm đề ra và nó hoạt động ổn định trong thời gian dài.

Trong quá trình làm việc, nhóm đã được tiếp cận và tìm hiểu về kết nối Bluetooth Low Energy, về lập trình ESP32 cho các thành phần phần cứng và cách thiết lập kết nối giữa nó với một trang web thông qua kết nối BLE để chúng giao tiếp với nhau.

Mô hình sản phẩm này có thể được ứng dụng sử dụng trong gia đình hoặc trong các tiệm bán thực phẩm với chức năng đo khối lượng vật và hiển thị khối lượng trên giao diện máy tính. Sử dụng kết nối BLE như một giải pháp hữu hiệu khi không sử dụng kết nối có dây hoặc không có mạng wifi. Từ đó cho thấy lợi ích tiềm năng của kết nối BLE trong hệ thống IoT hiện đại, góp phần giải quyết các vấn đề thực tiễn và thúc đẩy sự phát triển của các thiết bị thông minh trong tương lai.

## CHƯƠNG 5: TÀI LIỆU THAM KHẢO

### 1. Hướng dẫn sử dụng ESP32:

- "Getting Started with ESP32 Development Board" Espressif Systems,(năm 2024,[online](#)).
- "ESP32 Technical Reference Manual," Espressif Systems,(năm 2024,[online](#)).

### 2. Kết nối Bluetooth Low Energy

- Công nghệ kết nối BLE – Rạng Đông,(năm 2024,[online](#)).
- BLE với ESP32: ESP32 Bluetooth Low Energy (BLE) on Arduino IDE | Random Nerd Tutorials, (năm 2024, [online](#)).

### 3. Cảm biến Loadcell và HX711

- ESP32 with Load Cell and HX711 Amplifier (Digital Scale),(năm 2024,[online](#)).

### 4. Module I2C và LCD

- "ESP32 LCD Display Tutorial" on Random Nerd Tutorials,(năm 2024,[online](#)).

### 5. Thiết kế hệ thống - Mô hình tham khảo:

- Arduino IDE + ESP32 + 5kg Load Cell Sensor + HX711 Module | ESP32 Digital Scales,(năm 2024,[online](#)).

### 6. Tài liệu tham khảo khác

- Thiết kế giao diện: ESP32 Web Bluetooth (BLE): Getting Started Guide | Random Nerd Tutorials(năm 2024,[online](#)).
- Tham khảo đề tài: Building Smart Weight ,(năm 2024,[online](#)).