

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**ĐỒ ÁN MÔN HỌC**

**Cân điện tử**

**Nguyễn Văn Minh – 20215092**

**Ngành CNTT Việt Nhật (HEDSPI)**

**Giảng viên hướng dẫn:** Nguyễn Đức Tiến

**Bộ môn:** Kỹ thuật máy tính

*Hà Nội, tháng 7 năm 2024*

# MỤC LỤC

<b>CHƯƠNG 1: Giới thiệu sản phẩm .....</b>	<b>4</b>
1. Tính năng sản phẩm .....	4
2. Thiết bị sử dụng .....	4
3. Môi trường phát triển.....	4
<b>CHƯƠNG 2: Xây dựng chương trình.....</b>	<b>5</b>
1. Tìm hiểu cách hoạt động của phần cứng .....	5
1.1 Load cell 50kg .....	5
1.2 HX711.....	7
2. Xây dựng phần mềm.....	8
2.1 Cấu trúc mã nguồn.....	8
2.2 Tính toán khối lượng.....	9
2.3 Xử lý ngắt .....	9
2.4 Chế độ ngủ.....	10
<b>Chương 3: Đánh giá tổng quan.....</b>	<b>10</b>
1. Kết quả sau khi hoàn thành đồ án môn học .....	10
2. Các vấn đề gặp phải.....	10
3. Hướng phát triển .....	10
<b>Phụ lục.....</b>	<b>11</b>
Mã nguồn sản phẩm .....	11
Mô phỏng sản phẩm .....	11

## LỜI NÓI ĐẦU

Công nghệ thông tin ngày càng phát triển và có vai trò hết sức quan trọng không thể thiếu trong cuộc sống hiện đại, nó đã thay đổi mọi khía cạnh của cuộc sống, từ việc giao tiếp, giải trí, đến học tập và công việc. Chúng ta sống trong một thế giới với hàng tỷ thiết bị kết nối với Internet, tạo nên mạng lưới thông tin phức tạp. Nhưng không chỉ có những thiết bị lớn như máy tính cá nhân và điện thoại thông minh, công nghệ thông tin còn hiện diện ẩn mình trong những hệ thống nhỏ gọn và tích hợp, được gọi là hệ thống nhúng.

Hệ thống nhúng là những thiết bị và hệ thống tích hợp các vi xử lý, cảm biến, và giao tiếp mạng. Điển hình cho các ứng dụng của hệ thống nhúng là các thiết bị di động, các thiết bị y tế thông minh, đồ gia dụng kết nối Internet (Internet of Things - IoT) và nhiều ứng dụng khác.

Sự phát triển của công nghệ đã thúc đẩy sự tiến bộ của hệ nhúng. Các bộ vi xử lý ngày càng mạnh mẽ và nhỏ gọn, công nghệ bộ nhớ và lưu trữ phát triển nhanh chóng, đồng thời với đó là xu hướng phần mềm mã nguồn mở và các công cụ phát triển hệ thống nhúng. Tất cả những tiến bộ này đã tạo điều kiện thuận lợi cho việc phát triển các ứng dụng hệ thống nhúng phức tạp và đa dạng.

Với sự hấp dẫn của lĩnh vực và những thách thức còn đang ở phía trước, với niềm đam mê, mong muốn được học hỏi các công nghệ, tiếp xúc với các bài toán liên quan đến hệ nhúng, em đã quyết định lựa chọn đề tài “Cân điện tử thông minh” cho Đồ án môn học của mình.

Mặc dù đã cố gắng hoàn thiện sản phẩm nhưng không thể tránh khỏi những thiếu hụt về kiến thức và sai sót trong kiểm thử. Em rất mong nhận được những nhận xét thẳng thắn, chi tiết đến từ thầy để tiếp tục hoàn thiện hơn nữa. Cuối cùng, em xin được gửi lời cảm ơn đến thầy Nguyễn Đức Tiến đã hướng dẫn em trong suốt quá trình hoàn thiện Đồ án môn học.

# CHƯƠNG 1: Giới thiệu sản phẩm

## 1. Tính năng sản phẩm

- Cân khối lượng: hiển thị kết quả tới 2 chữ số thập phân, cho phép hiển thị kết quả âm.
- Hiển thị nhiệt độ môi trường.
- Tinh chỉnh kết quả nếu cân chưa đúng.
- Thay đổi đơn vị cân từ kilogram sang pound, đơn vị nhiệt độ từ celsius sang fahrenheit.
- Điều chỉnh cân về 0 kg.
- Lưu kết quả của các lượt cân gần nhất.
- Tự động vào chế độ ngủ (tắt đèn nền, giảm tần suất đọc dữ liệu từ cảm biến).

## 2. Thiết bị sử dụng

STT	Tên linh kiện	Số lượng
1.	ESP32 devkit module	1
2.	Bộ khuếch đại 24 bit HX711	1
3.	Load cell 50kg	4
4.	LCD1602 I2C hoặc OLED 128x64	1
5.	Cảm biến nhiệt độ	1
6.	Công tắc bấm	5

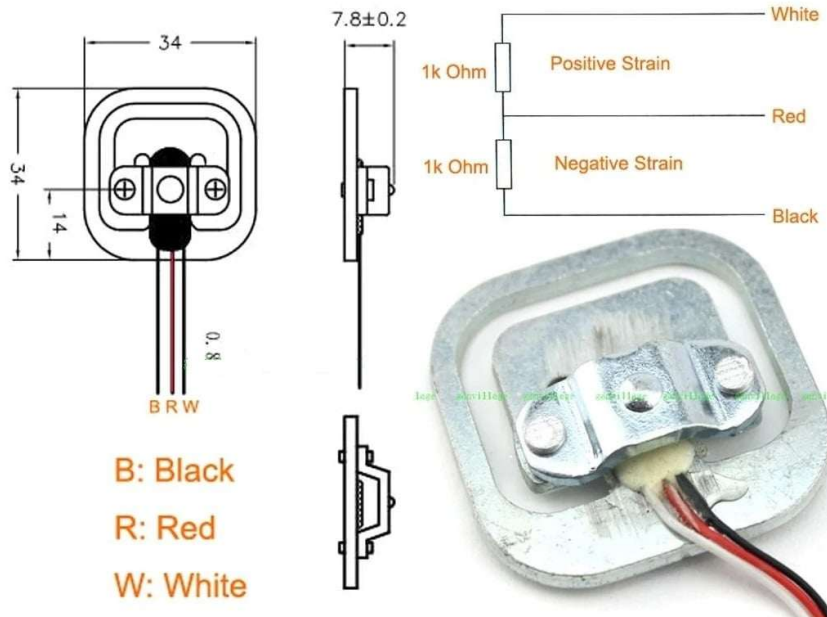
## 3. Môi trường phát triển

- Visual Studio Code, Arduino IDE: biên soạn mã nguồn.
- Wokwi: chạy mô phỏng sản phẩm.

## CHƯƠNG 2: Xây dựng chương trình

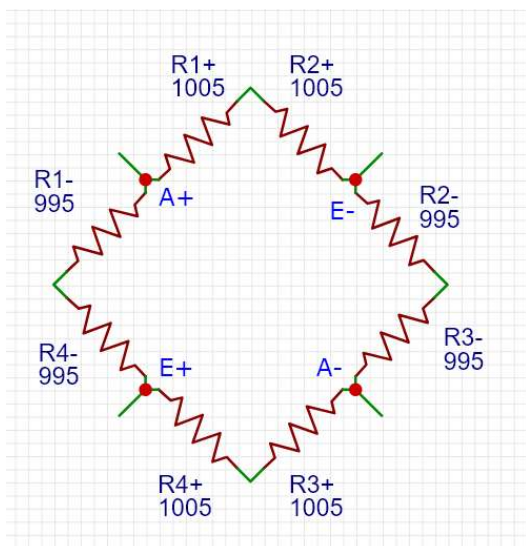
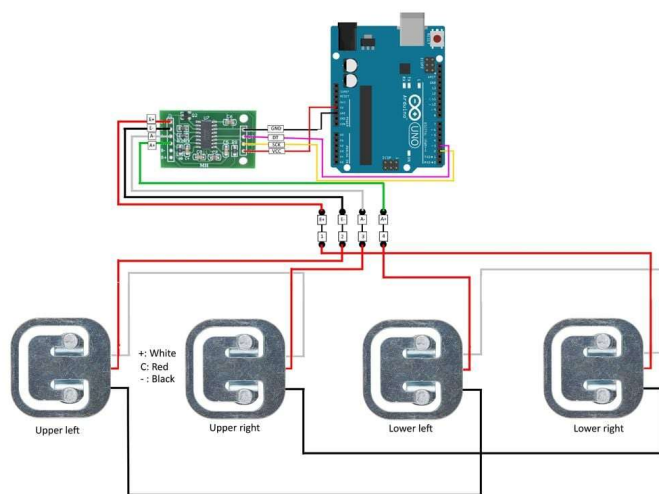
### 1. Tìm hiểu cách hoạt động của phần cứng

#### 1.1 Load cell 50kg



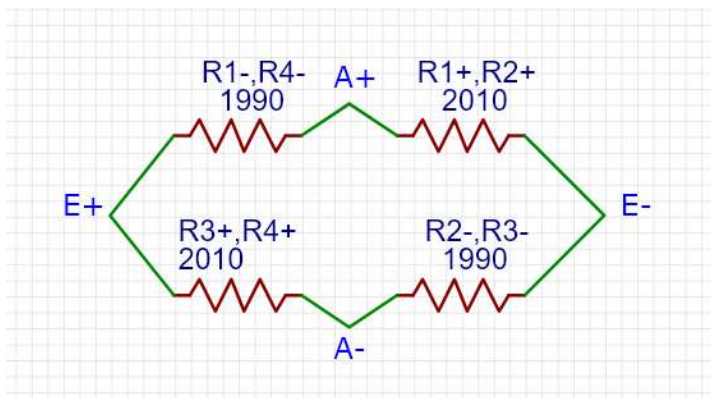
Cấu tạo của load cell 50kg

Load cell 50kg thông dụng trên thị trường có cấu tạo như hình trên. Gồm 2 điện trở có thể thay đổi giá trị dựa vào độ nén, giãn khi tác động lực lên load cell.



Cách đấu nối 4 load cell 50kg

4 load cell 50kg được nối dây để tạo thành một mạch cầu. R1+ và R1- lần lượt là điện trở có giá trị tăng và giảm khi tác động lực lên load cell 1. E+, E-, A+, A- là tên chân được đấu với HX711. Xét ví dụ khi tác động lực đều lên 4 load cell, các điện trở bình thường ở giá trị 1000Ω sẽ thay đổi lên 1005Ω hoặc xuống 995Ω. Gộp các điện trở ở giữa các điểm E+, E-, A+, A-. Ta có mạch cầu như sau:



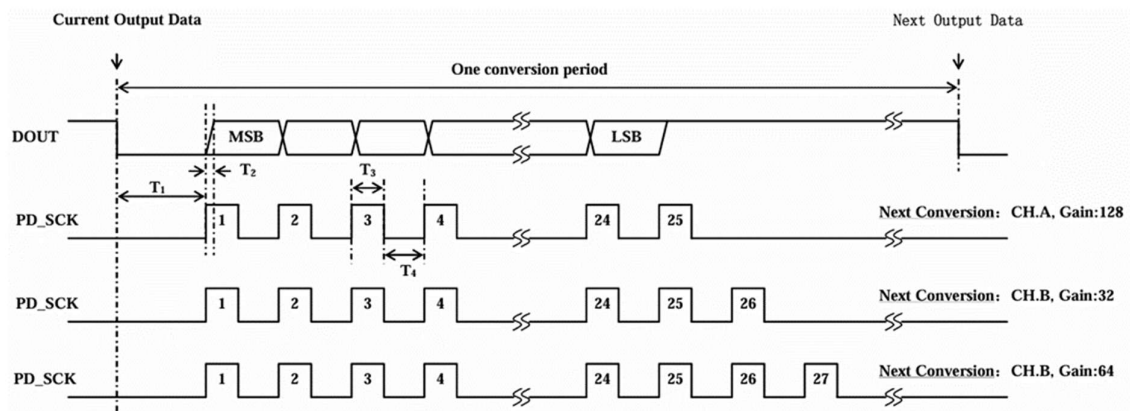
## Mạch cầu đơn giản hoá

Dựa vào công thức tính hiệu điện thế qua mạch điện gồm các điện trở nối tiếp. Ta tính được hiệu điện thế giữa A+ và A- như sau:

$$V = 2 \times \left[ (V_{E+} - V_{E-}) \left( \frac{1}{1 + \frac{R_-}{R_+}} - \frac{1}{2} \right) \right]$$

Với giá trị điện trở như trên sơ đồ bên trên. Ta tính được hiệu điện thế giữa A+ và A- là 25mV. Vì giá trị này quá nhỏ để vi điều khiển xử lý được nên cần một bộ khuếch đại để đọc được chính xác khối lượng trên các load cell.

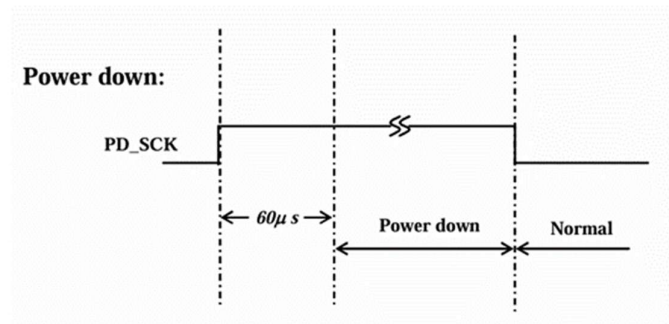
### 1.2 HX711



Cách thức truyền gửi dữ liệu trên HX711([HX711 datasheet](#))

Có thể tóm tắt thứ tự các bước thực hiện đọc dữ liệu từ HX711 như sau:

1. Chờ tới khi (DOUT → LOW) thì bắt đầu đọc dữ liệu
2. Đọc 24 bit giá trị khuếch đại. Mỗi khi phát một xung (LOW → HIGH) thì đọc 1 bit từ chân DOUT
3. Phát thêm (Gain – 24) xung để set up giá trị khuếch đại của HX711 cho lần đọc tiếp theo. Sau quá trình đọc dữ liệu, nếu DOUT không lên mức HIGH thì trả về một giá trị đã được định nghĩa (HX711\_FAIL) để thông báo lỗi



Power down control

- Ngoài ra có thể đưa PD\_SCK lên HIGH trong tối thiểu  $60\mu s$  để đưa HX711 vào chế độ Power down.
- Sau khi hiểu được cách đọc dữ liệu từ HX711 và áp dụng vào chương trình em thấy kết quả mỗi lần đọc chênh lệch nhau khá nhiều. Vì thế để giảm sai số, em đã thiết kế giải thuật để làm tròn kết quả của các lần đo qua hàm `getData_Avg()`.

Nguyên lý hoạt động của hàm này như sau:

1. Sử dụng mảng  $d[N]$  lấy N dữ liệu (không lấy các giá trị fail và các giá trị quy đổi ra khối lượng lớn hơn tải trọng tối đa).
2. Lấy giá trị trung bình của N phần tử trong mảng  $d$ . Sắp xếp mảng  $d$  theo thứ tự các phần tử cách xa dần giá trị trung bình.
3. Thực hiện tối đa K lần lấy dữ liệu nữa, đồng thời vẫn duy trì mảng  $d$  có N phần tử (phần tử nào có khoảng cách tới giá trị trung bình xa nhất sẽ bị thay thế). Tới khi hiệu giữa giá trị trung bình và giá trị cách xa nó nhất nhỏ hơn sai số tuyệt đối cho trước thì dừng lại.
4. Trả về kết quả, đồng thời lưu lại giá trị và sai số của lần đo đó.

## 2. Xây dựng phần mềm

### 2.1 Cấu trúc mã nguồn

File	Nhiệm vụ
config.h	Khai báo các thành phần phần cứng, các thông số phần cứng (các chân kết nối với board, sai số của load cell,...) và các thông số của chương trình (thời gian chờ để vào chế độ ngủ,...)
screen.cpp, screen.h	Khai báo và định nghĩa các lớp LCD_I2C, OLED_SSD1306 cùng thực thi interface Screen. In
HEDSPI_HX711.cpp HEDSPI_HX711.h	Khai báo và định nghĩa lớp HX711. Cho phép đọc dữ liệu từ bộ khuếch đại 24bit HX711, thiết lập chế độ khuếch đại,...
utility.cpp, utility.h	Định nghĩa các hàm tiện ích, các hàm xử lý ngắt.



main.h	Khai báo các biến toàn cục có thể sử dụng cho các file khác nhau trong chương trình
main.ino	Chương trình chính. Khai báo các biến chính được sử dụng trong chương trình.

## 2.2 Tính toán khối lượng

Hiện thị kết quả dưới dạng số thập phân 2 chữ số sau dấu phẩy, cho phép hiển thị số âm. Có 2 lựa chọn hiển thị đơn vị là kilogram-kg hoặc pound-lb. Giá trị hiển thị bằng dữ liệu lấy được từ bộ khuếch đại trừ cho giá trị dữ liệu tương ứng với 0kg, sau đó chia cho hệ số giữa dữ liệu thu được từ bộ khuếch đại và khối lượng thực tế.

data: Giá trị đọc được từ HX711

m: Khối lượng thực tế

w: Giá trị hiển thị

Tare: Giá trị tương đương với khối lượng của vỏ cân

$\Rightarrow \text{Scale} = \text{data} / m$

$\Rightarrow w = (\text{data} - \text{Tare}) / \text{Scale}$

Để tính toán được 2 tham số cần thiết là Scale và Tare. Em xử lý như sau đối với tham số:

- + Scale: Lấy một vật có khối lượng chuẩn (m kg) đặt lên cân sau đó xem giá trị data tương ứng và tính toán thủ công  $\text{Scale} = \text{data} / m$  rồi chèn vào mã nguồn. Ngoài ra cân còn có 2 nút bấm hiệu chỉnh khối lượng, khi sử dụng 2 nút này sẽ làm thay đổi trực tiếp giá trị của tham số Scale.
- + Tare: Khi cân mới khởi động hoặc khi bấm nút Tare, chương trình sẽ đọc giá trị data từ HX711 và gán vào Tare.

## 2.3 Xử lý ngắt

Hàm xử lý ngắt cho mỗi nút bấm có cấu trúc tương tự nhau. Để tránh hiện tượng tín hiệu không ổn định trong thời gian bấm nút, chương trình chỉ ghi nhận mỗi ngắt cách nhau một khoảng thời gian cố định DEBOUNCE\_TIME (200ms). Khi ghi nhận ngắt của nút nào thì gán biến tương ứng bằng 1, tăng biến ngắt tổng của chương trình (interrupt) lên 1 đơn vị.

- Xử lý ngắt cho nút UP (tương tự cho nút DOWN): nút UP được dùng để điều chỉnh kết quả hiển thị tăng lên nếu kết quả hiển thị đang bị lệch bằng cách giảm hệ số Scale xuống. Nếu người dùng nhấn nút 1 lần, khối lượng hiển thị sẽ chỉ tăng một ít. Nếu người dùng bấm giữ nút UP thì khối lượng hiển thị sẽ tăng nhanh hơn.
- Xử lý ngắt cho nút TARE: đọc dữ liệu từ bộ khuếch đại và gán cho biến Tare.

- Xử lý ngắt cho nút MODE: thay đổi biến Mode từ (kg, °C) sang (lb, °F) bằng cách thay đổi giá trị biến Mode.
- Xử lý ngắt cho nút RECORD: hiển thị lần lượt khối lượng của các lượt cân gần nhất

## 2.4 Chế độ ngủ

Nếu sau một khoảng thời gian cố định, giá trị đọc được từ bộ khuếch đại không thay đổi thì chương trình sẽ tự động vào chế độ ngủ. Trong chế độ ngủ, chương trình vẫn lắng nghe có ngắt xảy ra hay không, khối lượng có thay đổi hay không. Nếu phát hiện có ngắt hoặc thay đổi khối lượng, chương trình sẽ trở về luồng hoạt động chính. Ngoài ra khi vào chế độ ngủ, chương trình sẽ chuyển sang dùng GAIN\_64 thay vì GAIN\_128 để giảm thời gian đọc dữ liệu từ bộ khuếch đại.

# Chương 3: Đánh giá tổng quan

## 1. Kết quả sau khi hoàn thành đồ án môn học

- Nắm vững và áp dụng thành công nguyên lý hoạt động và kỹ thuật sử dụng load cell, bộ khuếch đại HX711.
- Phát triển kỹ năng lập trình vi điều khiển và giao tiếp giữa các thiết bị phần cứng.

## 2. Các vấn đề gặp phải

- Đọc dữ liệu từ bộ khuếch đại HX711 mất khá nhiều thời gian (với GAIN\_128 thì trung bình mất 100ms cho mỗi lần đọc). Mà chương trình cần đọc nhiều lần liên tiếp để đảm bảo lấy được giá trị có độ chính xác cao. Vì thế khi chương trình vừa cần delay để hiển thị kết quả, vừa cần lắng nghe xem khối lượng có thay đổi không để hiển thị lại ngay thì thời gian mỗi lần delay chưa được chính xác.

- Không thể đưa toàn bộ phần xử ngắt vào hàm xử lý ngắt vì trong hàm xử lý ngắt không thực hiện được hàm delay() và các hàm giao tiếp với màn hình lcd, oled. Có thể là do trong khi thực hiện một hàm ngắt, chương trình tạm thời tắt tất cả các ngắt khác của hệ thống (bao gồm timer) nên trong hàm xử lý ngắt không thực hiện được những hàm sử dụng timer. Đồng thời có thể có sự tối ưu code trong quá trình biên dịch dẫn đến sai logic của người lập trình.

## 3. Hướng phát triển

- Kết nối Bluetooth để gửi dữ liệu ra thiết bị khác.

- Kết nối nối máy đọc thẻ NFC, RFID.
- Kết hợp ESP32-Cam để chụp hình người cân, nhận dạng ảnh.
- Kết nối WiFi, gửi dữ liệu về datacenter.
- Cảm biến đo mạch máu và nồng độ oxy.

## Phụ lục

### Mã nguồn sản phẩm

Download mã nguồn tại: [VanMinh153/HX711\\_Digital\\_Scale \(github.com\)](https://github.com/VanMinh153/HX711_Digital_Scale)

### Mô phỏng sản phẩm

Cân với 1 bộ HX711: <https://wokwi.com/projects/403644391219010561>

Cân với 4 bộ HX711: <https://wokwi.com/projects/403649602321315841>