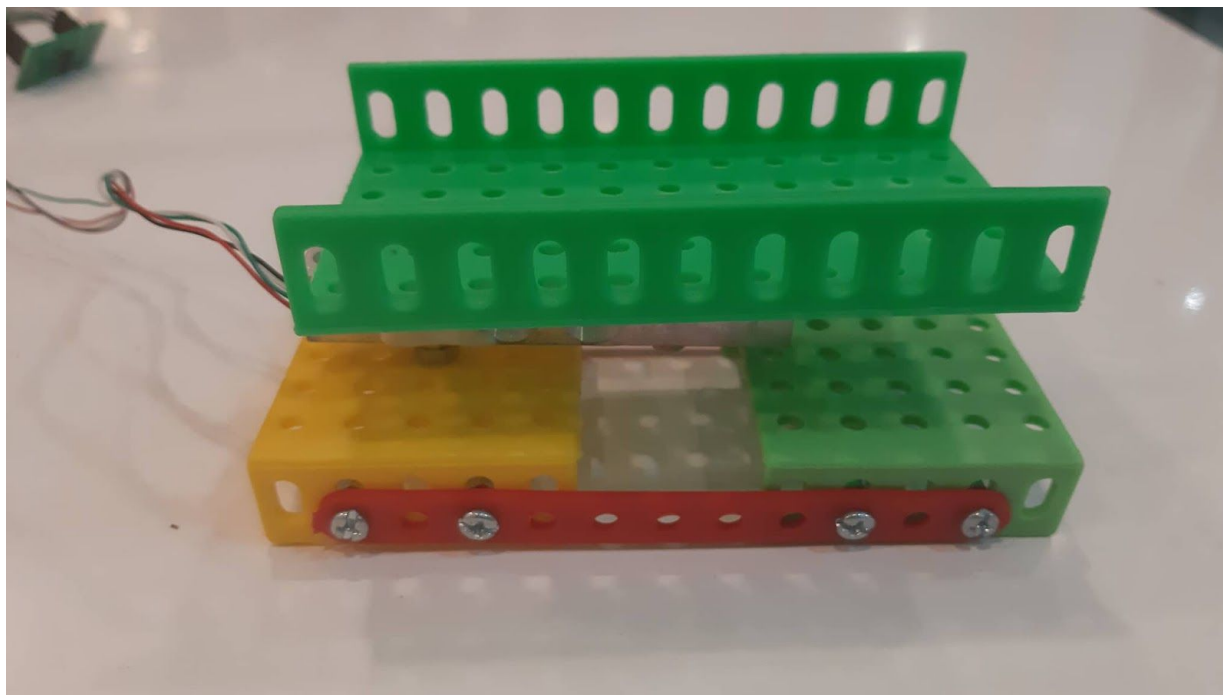




TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA TP.HCM



Cân điện tử

Đồ án cuối kỳ

Lập trình nhúng

1612899 Hoàng Xuân Trường

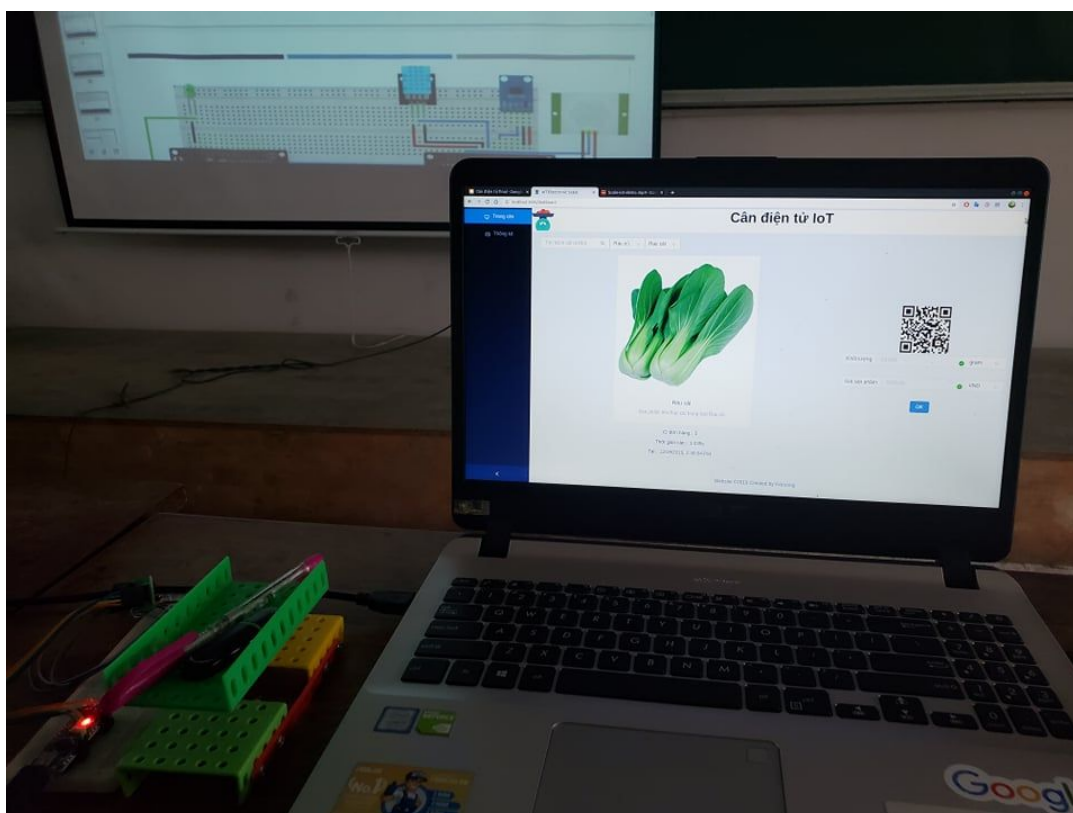
1612847 Đỗ Hoàng Anh Duy

1. Giới thiệu	2
2. Sản phẩm	3
2.1 Mô hình cân	3
2.2 Website	5
3. Nội dung chi tiết	8
I. Cảm biến LoadCell	8
II. Module HX711	9
III. Thiết kế phần cứng	9
IV. Mạch điện	10
V. Web server	10
VI. Xử lý dữ liệu từ Loadcell	10
a. Vấn đề	11
b. Ý tưởng	11
c. Thuật toán	11
VII. Giao tiếp với server	11
VIII. State machine	12
a. Sơ đồ 1: Tính toán khối lượng	12
b. Sơ đồ 2: Trạng thái cân điện tử	13
IX. Web UI	13
4. Source Code	15
5. Tham khảo	15

1. Giới thiệu

Với công nghệ đang phát triển hiện nay, cân điện tử đang dần thay thế các loại cân truyền thống thông thường. Nhờ vào ưu thế cân nhanh, chính xác của cân điện tử nên nó được đưa vào ứng dụng thực tiễn trong các siêu thị, chợ, hay các cửa hàng tiện lợi. Cân điện tử có thể cân được những vật rất nhẹ hoặc rất nặng, đồng thời dễ dàng xem khối lượng trên màn hình hiển thị càng giúp người dùng dễ dàng hơn trong công việc. Cân còn có nhiều hình dạng, khối lượng hay kích thước khác nhau tùy vào mục đích của người sử dụng.

Trong khuôn khổ đề án này, sản phẩm là **cân điện tử** có trọng tải **1kg** có khả năng cân đồ vật nhanh chóng và độ chính xác cao. Cơ bản, cân điện tử trong đề án gồm 3 thành phần chính: cảm biến loadcell, bo mạch điện và màn hình hiển thị (web). Sản phẩm như hình sau:



Hình 1: Cân điện tử với màn hình hiển thị

Hầu hết các cân điện tử hiện nay đều chỉ có hiển thị khối lượng cân và không có các tiện ích khác như tính tiền khi cân đồ hoặc bảng thống kê khi cân. Vì để đáp ứng nhu cầu này, sản phẩm cân điện tử được làm ngoài chức năng chính là cân được khối lượng đồ vật mà còn có thể chọn loại sản phẩm nào được cân để tính tiền luôn cho sản phẩm đó. Ngoài ra, có một trang thống kê các đồ vật vừa được cân để dễ dàng trong việc kiểm tra.

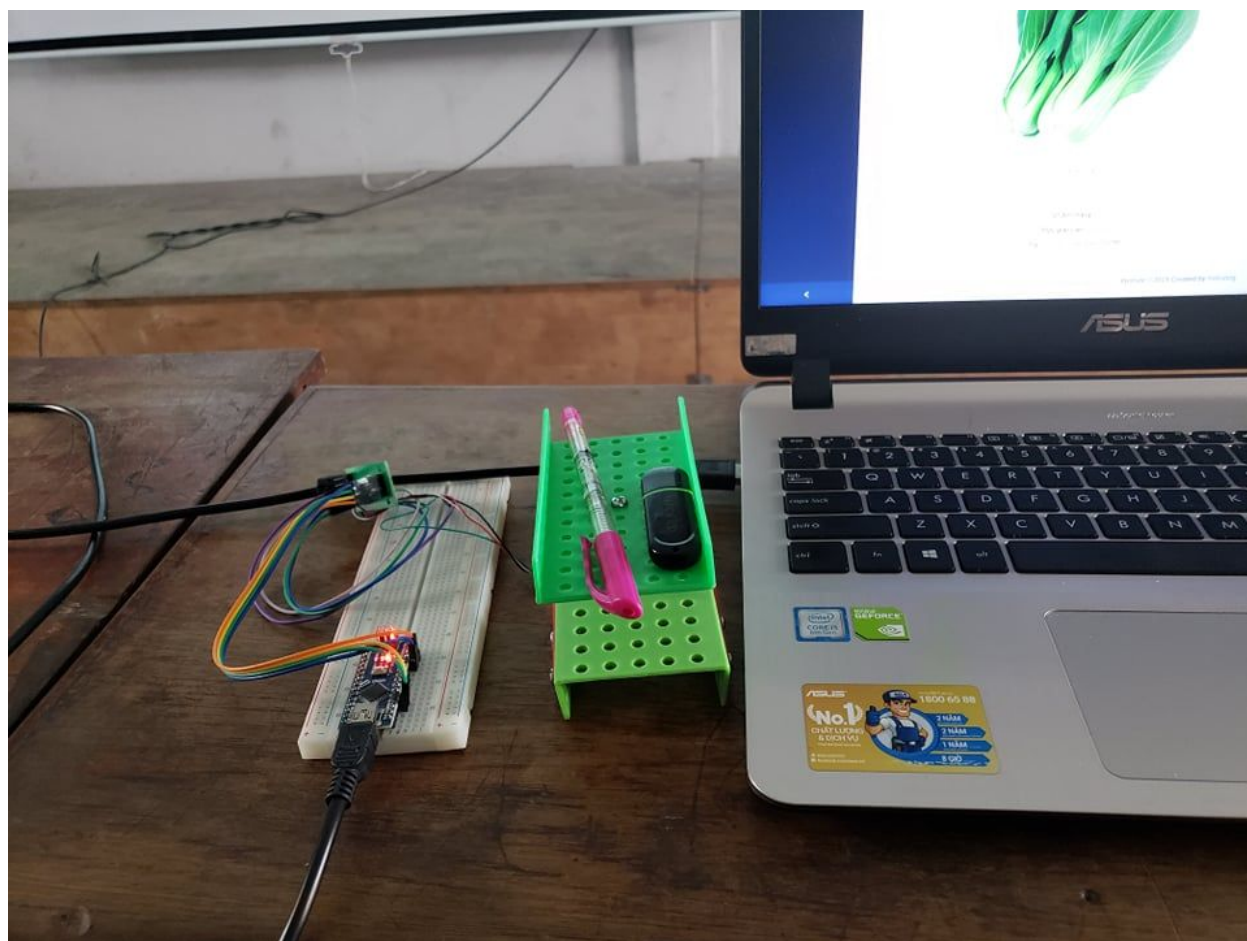
Sản phẩm trong đề án này là cân điện tử trong siêu thị hoặc các cửa hàng cần phải cân như thịt, rau, củ, quả.

2. Sản phẩm

Cân điện tử bao gồm các **chức năng** chính sau:

1. Đo khối lượng của vật
2. Cho phép chọn loại đồ vật cân để tính tiền sản phẩm ngay sau khi khối lượng hiển thị
3. Mã QR code để truy vết sản phẩm (tên, khối lượng, giá tiền)
4. Lưu trữ các lần cân của cân điện tử
5. Bảng thống kê cho phép người dùng xem lại các đồ vật mới cân

2.1 Mô hình cân



Hình 2: Cân điện tử đang cân

Hình trên là hình ảnh của cân điện tử đang cân 1 cây bút bi và 1 USB. Chiếc cân gồm các thành phần chính:

- Arduino nano
- Cảm biến lực loadcell 1kg
- Cảm biến HX711
- Dây điện, dây cáp
- Bàn cân và đế cân bằng nhựa

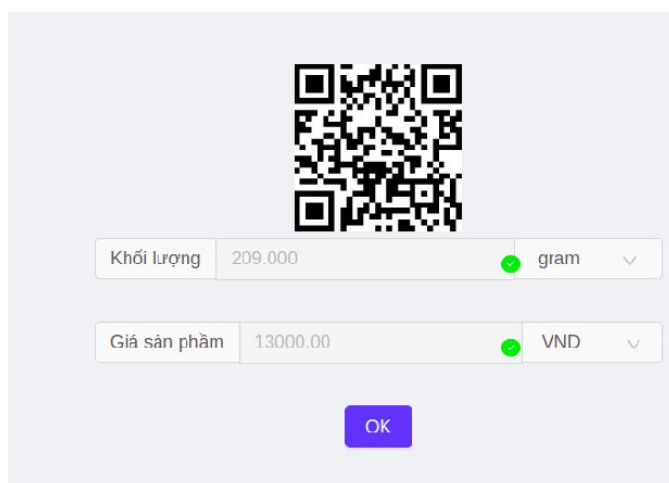
❖ Cách sử dụng

Ở trạng thái ban đầu, trên bàn cân không có bất kì vật dụng nào và trên website sẽ hiển thị khối lượng bằng 0

Ở trạng thái cân, có đồ vật trên bàn cân và website hiển thị khối lượng

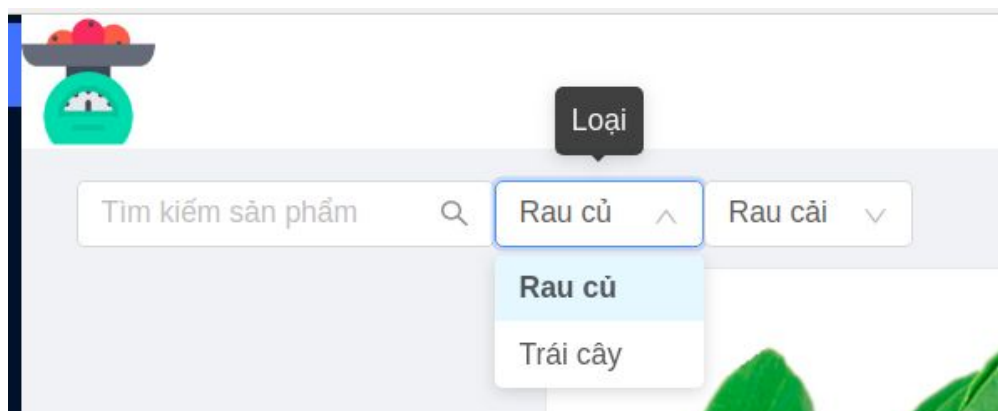
2.2 Website

Website dùng để hiển thị **khối lượng, giá tiền** và **QR code** của đồ vật vừa được cân

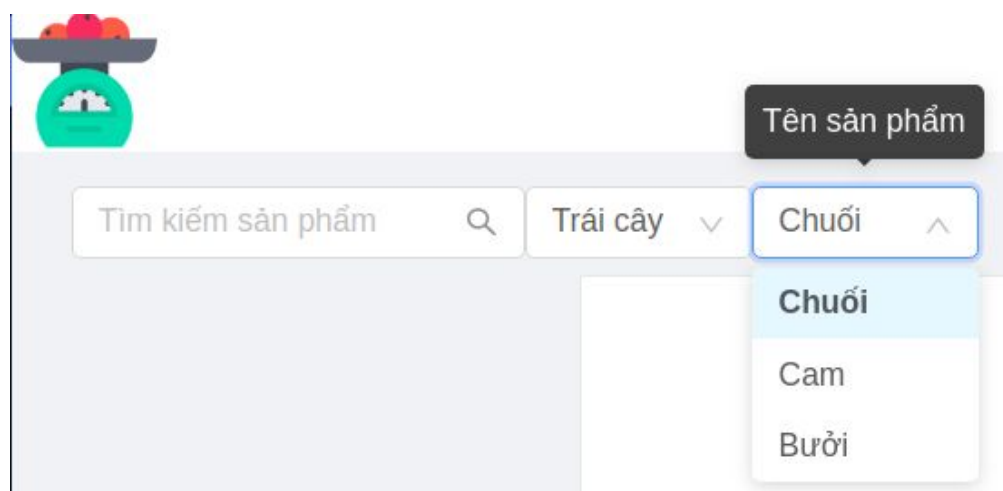


The screenshot shows a website interface with a QR code at the top. Below the QR code, there are two input fields. The first field is labeled "Khối lượng" (Weight) and contains the value "209.000". To its right is a green checkmark and a dropdown menu showing "gram". The second field is labeled "Giá sản phẩm" (Product price) and contains the value "13000.00". To its right is a green checkmark and a dropdown menu showing "VND". Below these fields is a purple button labeled "OK".

Cho phép người dùng chọn loại sản phẩm cân và chọn tên sản phẩm



The screenshot shows a website interface with a search bar labeled "Tìm kiếm sản phẩm" (Search for product). To the right of the search bar is a dropdown menu labeled "Loại" (Type). The dropdown menu is open, showing a list of product types: "Rau củ" (Vegetables), "Rau cải" (Leafy greens), and "Trái cây" (Fruits). The "Rau củ" option is currently selected and highlighted in blue.



Hiển thị ID, thời gian cân để thống kê



Bảng thống kê

IoT Electronic Scale

Cân điện tử IoT

ID	Tên	khối lượng(gam)	Phân loại	Thời gian cân	Tại	Hành động
1	Rau cải	65	Rau củ	2085	12/19/2019, 2:54:01 AM	
2	Cam	115	Trái cây	2085	12/19/2019, 2:54:13 AM	
3	Bưởi	210	Trái cây	1419	12/19/2019, 2:54:24 AM	
4	Cà rốt	14	Rau củ	1135	12/19/2019, 2:54:34 AM	

Xóa tất cả

Website ©2019 Created by hxtuong


Giao diện hoàn chỉnh:

IoT Electronic Scale

Cân điện tử IoT

Tìm kiếm sản phẩm

Trái cây Chuối



Chuối

Sản phẩm tên Chuối trong loại Trái cây

ID đơn hàng: 5

Thời gian cân: 1.419s

Tại: 12/19/2019, 2:55:36 AM

Khối lượng 65.000 gram

Giá sản phẩm 12000.00 VND

OK

Website ©2019 Created by hxtuong

3. Nội dung chi tiết

I. Cảm biến LoadCell

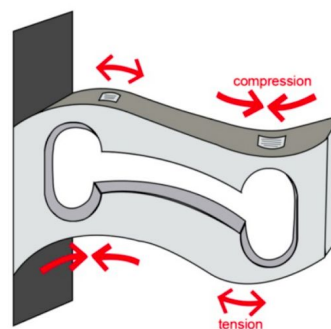
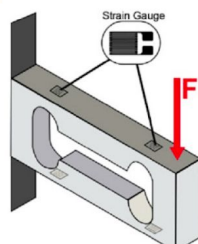
- Các Loadcell là các **cảm biến lực**
- Khi có một lực **tác động** lên loadcell, loadcell sẽ **chuyển đổi lực** tác dụng thành **tín hiệu điện** (truyền về là tín hiệu mV/V).
- Nguyên nhân là các loadcell sử dụng điện trở (loại strain gauge-đây là loại điện trở **thay đổi điện trở** khi có tác dụng của lực lên nó)



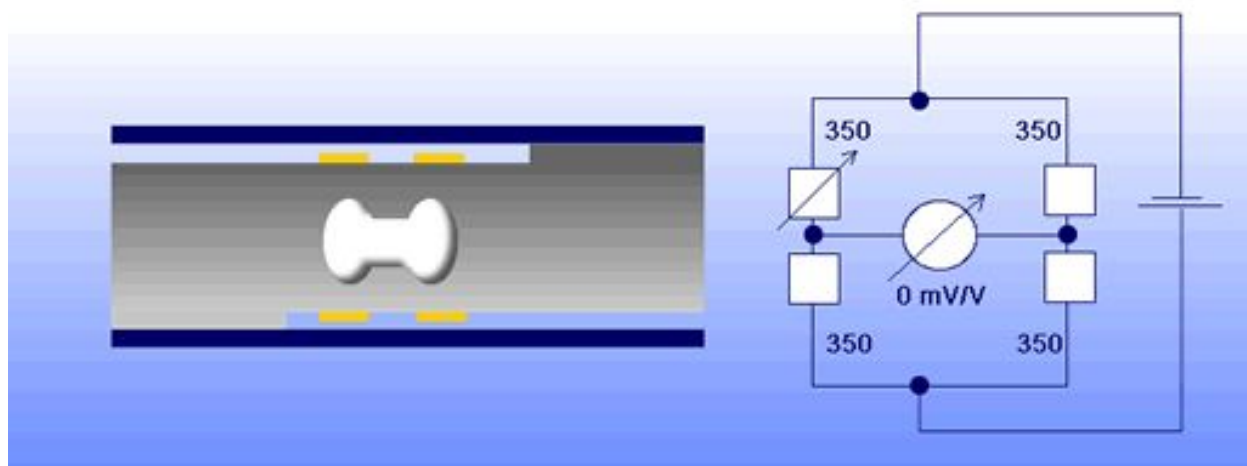
Về nguyên lý hoạt động:

Cảm biến loadcell

Resistive Load cell Principle



Khi có một lực **F** tác động vào cảm biến (như hình trên) thì cảm biến bị biến dạng tại 4 điểm được đặt điện trở tại đó. Khi có thay đổi điện trở, loadcell sẽ trả về tín hiệu điện biểu diễn khối lượng của đồ vật được cân.



Tín hiệu điện này khá phức tạp cho việc xử lý và sử dụng. Vì vậy để đơn giản việc này, module HX711 đã được sinh ra.

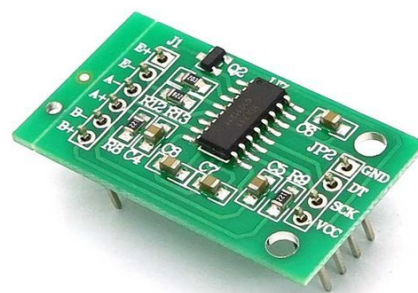
II. Module HX711

Là module chuyển đổi tương tự-số ADC (Analog Digital Convert)

Độ phân giải 24bit và giao tiếp 2 dây với vi điều khiển: 2 chân SCK (Clock) và DT (Data).

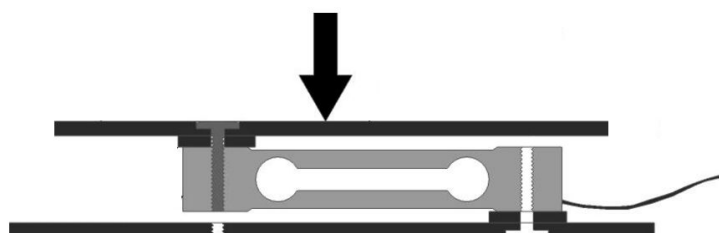
Thông số kỹ thuật:

- Điện áp hoạt động: 2.7 V - 5V
- Dòng tiêu thụ <1.5mA
- Tốc độ lấy mẫu: 10-80 SPS (Có thể tùy chỉnh)
- Độ phân giải điện áp: 40mV



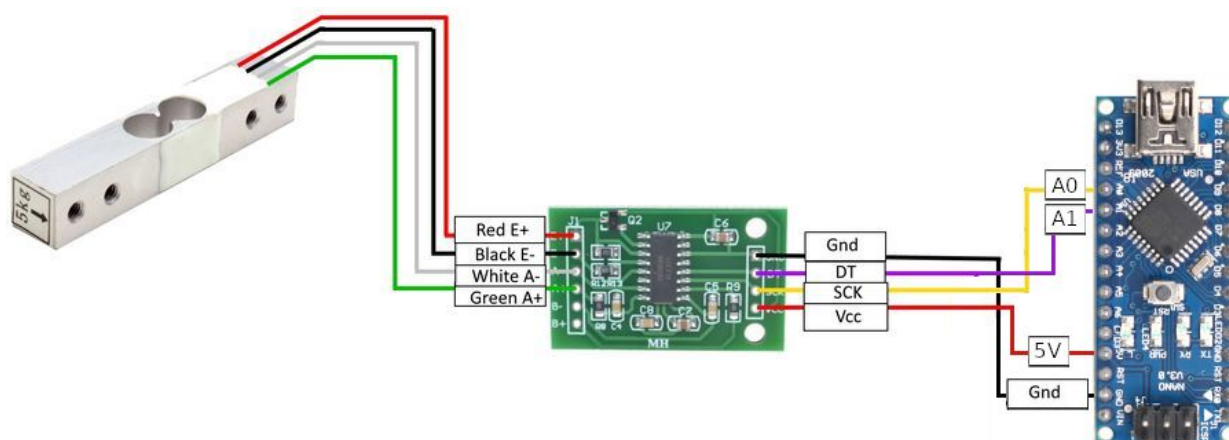
III. Thiết kế phần cứng

Bàn cân cần phải được thiết kế làm sao nằm ở giữa cảm biến loadcell. Việc này để đảm bảo tính cân bằng khi cân và không làm khối lượng vị sai lệch



IV. Mạch điện

Mạch điện được thiết kế như trong hình sau.



Điểm cần lưu ý là mỗi dây cần phải cắm đúng vị trí của nó.

V. Web server

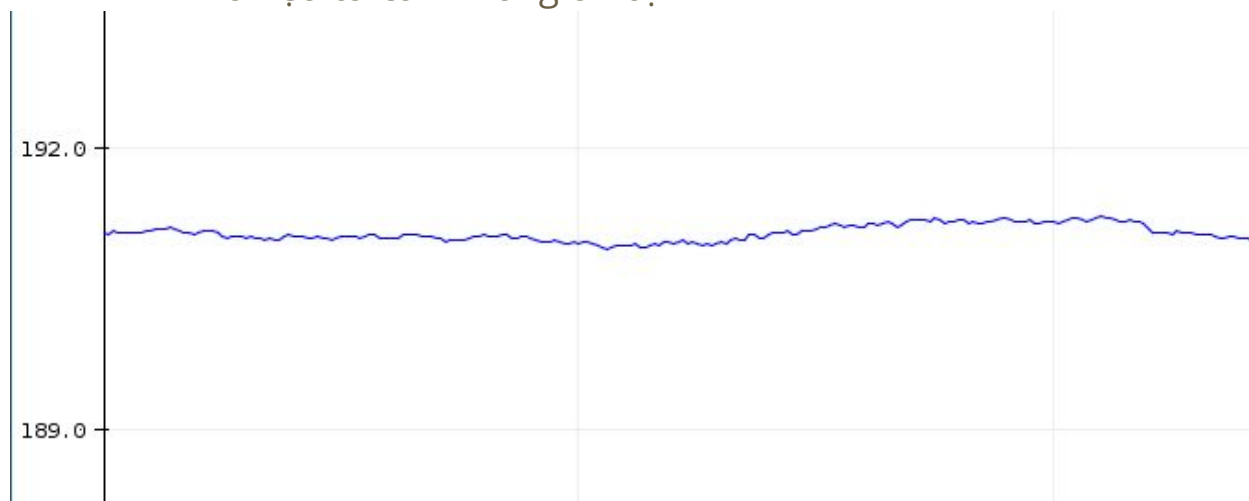
Web server sử dụng web socket.

API	Kết quả trả về	Ý nghĩa
GET /dev	["dev/ttyUSB0"]	Liệt kê danh sách các device
GET /dev/<tên device>	<ul style="list-style-type: none"> • "changing" • 0 • 200 	Trạng thái của cân (a) đang thay đổi và (b) khối lượng đã xác định . Trả về 0 là đang không có vật nào cân.

VI. Xử lý dữ liệu từ Loadcell

a. Vấn đề

- Cần xác định khi nào sản phẩm được đặt xuống/lấy lên
- Dữ liệu từ cân không ổn định



b. Ý tưởng

Khi vật được đặt xuống (lấy lên), giá trị cân sẽ tăng (giảm) một lượng lớn

c. Thuật toán

1. Dồn các giá trị tăng (giảm) liên tiếp thành một "gap".
2. Độ lớn $T = |\text{giá trị đầu} - \text{giá trị cuối}|$
3. Phân loại:
 - 3.1. Gap lớn: $T \geq 1$
 - 3.2. Gap nhỏ: $T < 1$
4. Gặp gap lớn \Rightarrow Đang thay đổi.
5. 5 gap nhỏ liên tiếp \Rightarrow Đã ổn định, khối lượng = trung bình 5 đỉnh gần nhất.

VII. Giao tiếp với server

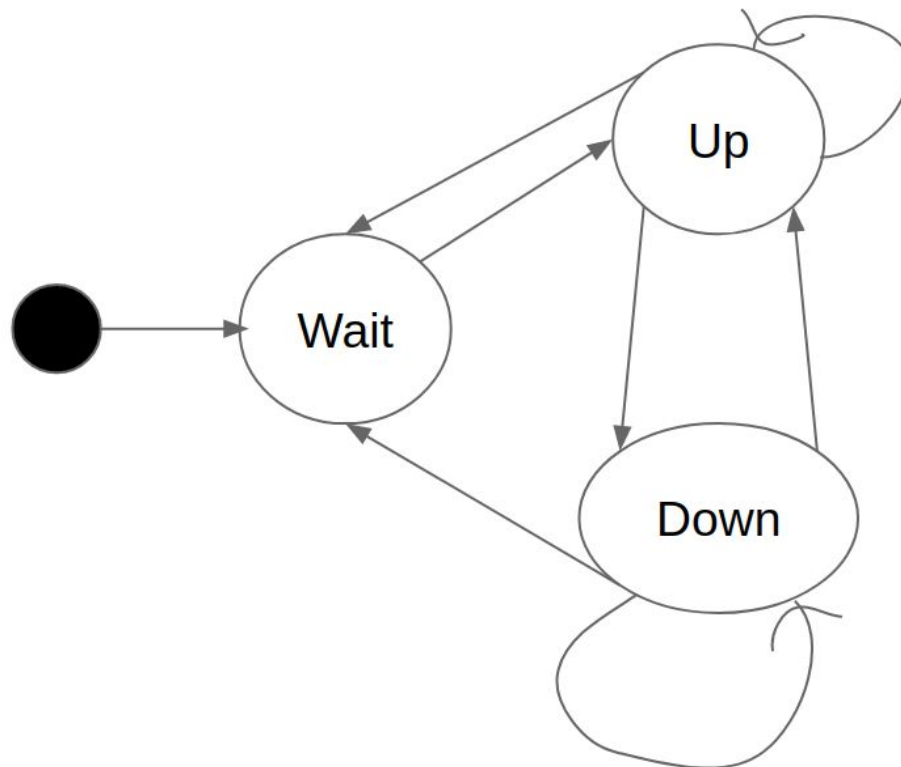
Sau khi cân một đồ vật, website cần phải biết được khối lượng của vật vừa được cân một các **realtime** để đảm bảo được tính hữu dụng của cân điện tử.

Cách thức giao tiếp với server như sau:

1. Ghi "started" khi xong hàm setup()
2. Chờ handshake có dạng "hi ..."
3. Echo lại handshake của server
4. Bắt đầu quá trình cân
5. Nếu nhận được "close" echo lại "close" và quay lại bước 2

VIII. State machine

a. Sơ đồ 1: Tính toán khối lượng



Mục đích:

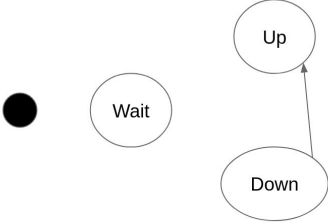
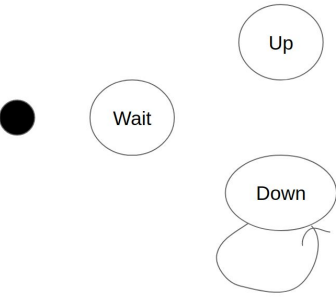
- Đợi handshake từ Server
- Dồn dữ liệu lại thành các gáp.

Trạng thái:

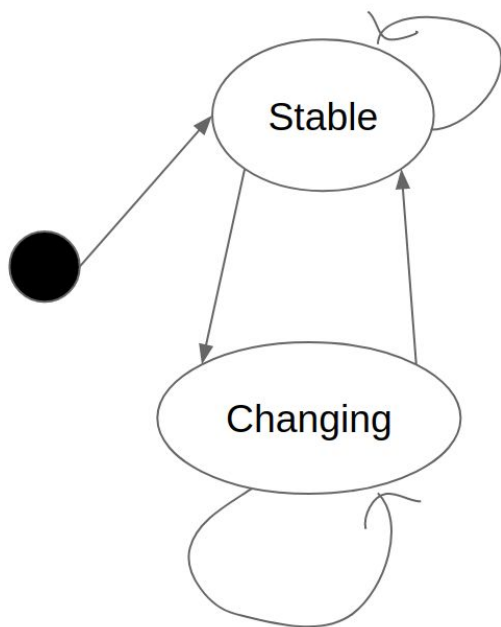
- Wait: Đang đợi

- Up: Đăng tăng
- Down: Đăng giảm

Trạng thái	Event	Hành động
<pre> stateDiagram-v2 [*] --> Wait Wait --> Up Up --> Down Down --> Wait </pre>	Nhận được "hi ..." từ server	<ul style="list-style-type: none"> - Reset cân về 0 - Khởi tạo các giá trị dùng trong việc tính toán
<pre> stateDiagram-v2 [*] --> Wait Wait --> Up Wait --> Down Up --> Down Down --> Wait </pre>	Nhận được "close" từ server	
<pre> stateDiagram-v2 [*] --> Wait Up --> Down Down --> Wait </pre>	$x(t) < x(t-1)$	<ul style="list-style-type: none"> - Tính giá trị gap và báo event có gap mới
<pre> stateDiagram-v2 [*] --> Wait Up --> Up Up --> Down Down --> Wait </pre>	$x(t) \geq x(t-1)$	

	$x(t) > x(t-1)$	Tính giá trị gap và báo event có gap mới
	$x(t) \leq x(t-1)$	

b. Sơ đồ 2: Trạng thái cân điện tử

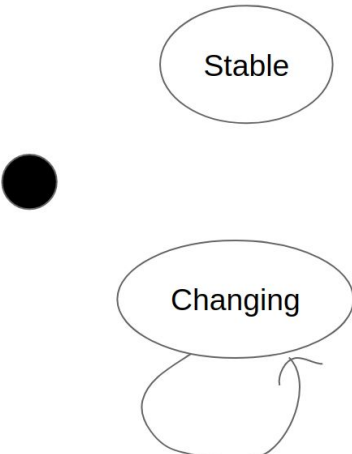
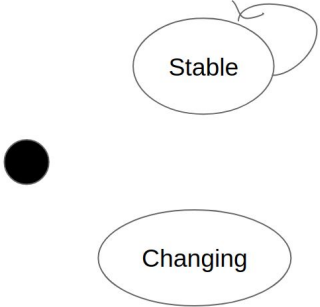
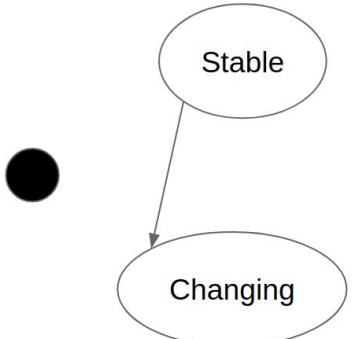


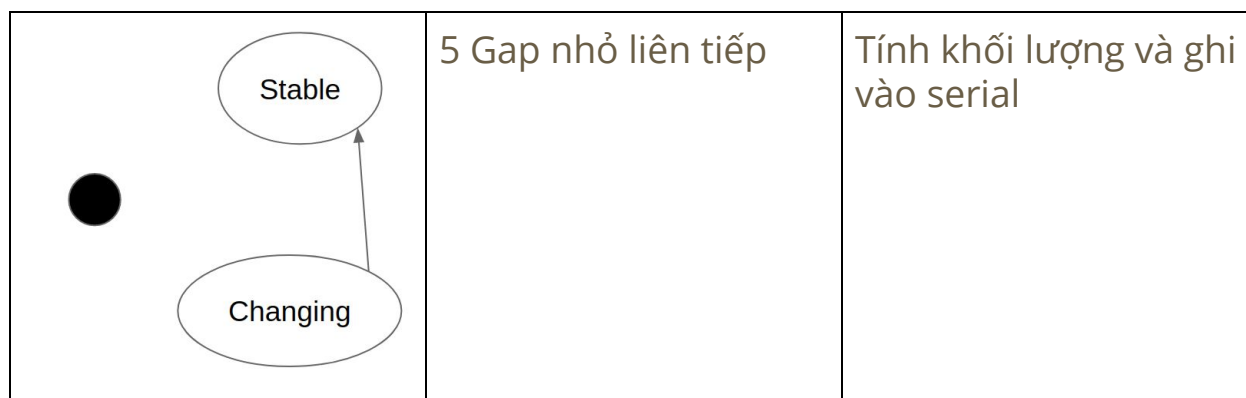
Mục đích:

- Đếm gap

Trạng thái:

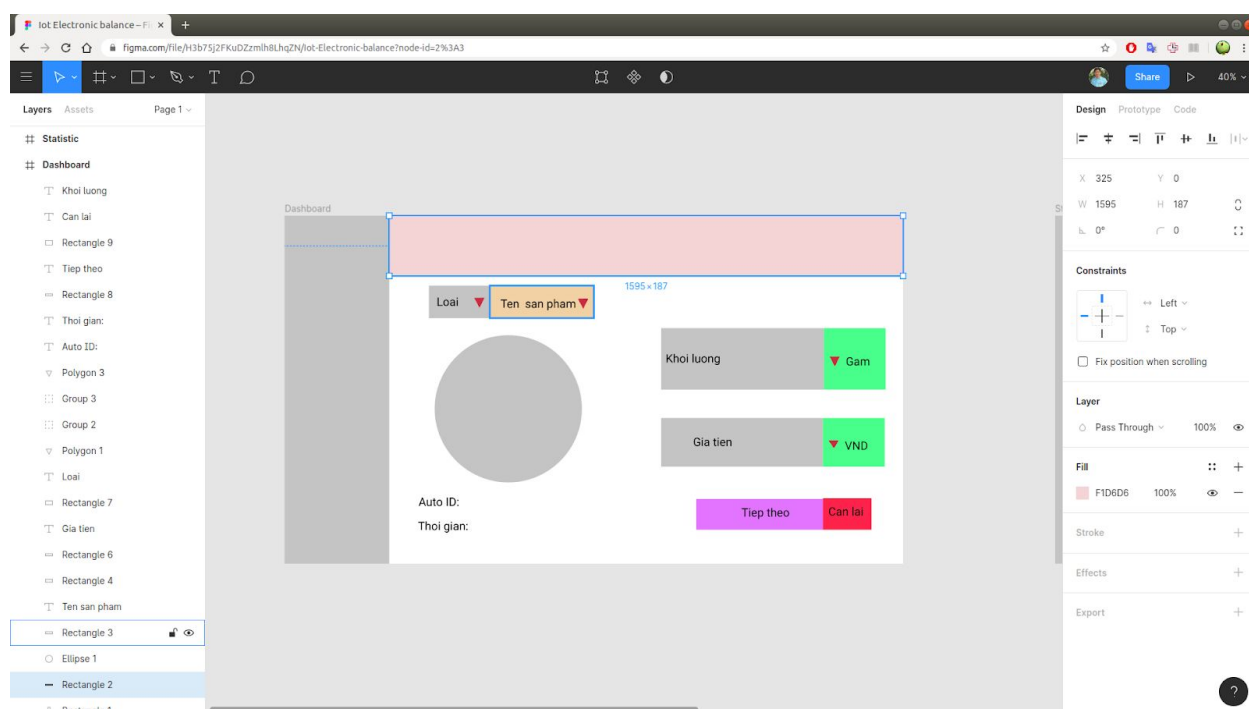
- Stable
- Changing

Trạng thái	Event	Hành động
	Gap lớn	
	Gap nhỏ	
	Gap lớn	Ghi "changing" vào serial

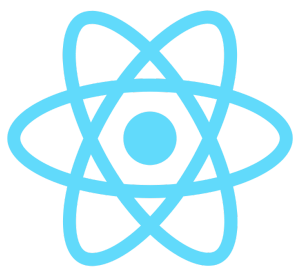


IX. Web UI

Trước khi bắt đầu cài đặt web, **Figma** được dùng để phác thảo UI.



Sau khi có UI, thư viện **ReactJs** và **AntD** được dùng để build web.



ReactJs

Ant Design

ReactJs là một thư viện của Javascript dùng để xây dựng giao diện web Single page application (SPA). Một cách đơn giản là cho phép thay đổi hiển thị trên web mà không cần phải tải lại (reload). Vì vậy nó phù hợp với việc cần hiển thị *realtime* kết quả của cân điện tử. **AntD** là một thư viện dễ dàng cho việc thiết kế website thân thiện với web trên laptop, mobile hay tablet.

4. Source Code

Github: <https://github.com/juchiast/iot-loadcell>

Slide:

https://docs.google.com/presentation/d/19xJtTPpbmGPfG5fr72lbFN2O2x8pX5Goul_OqmlxzQ/edit?usp=sharing

5. Tham khảo

1. <https://www.how2electronics.com/weighing-machine-using-arduino-load-cell-hx711-module/>
2. <http://cogetechcorp.blogspot.com/2017/11/can-ien-tu-voi-hx711-va-arduino.html>
3. <https://sites.google.com/site/chiasekienthucchuyensau/chuyendoitinhieu/strain-gauge-la-gi-cam-bien-loadcell-la-gi>
4. <https://candientubinhduong.com/tin-tuc/cam-bien-luc-co-cau-va-nguyen-ly-hoat-dong-61.html>