**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TPHCM**

**KHOA CÔNG NGHỆ ĐIỆN-ĐIỆN TỬ**



**ĐỒ ÁN 1**

**CÂN ĐIỆN TỬ DÙNG LOADCELL**

GIẢNG VIÊN HƯỚNG DẪN: **DƯƠNG VĂN KHẢI**

VIÊN THỰC HIỆN: **PHẠM QUANG KHANH**

**MSSV: 2013110245**

**THÁNG 05/2015MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc453128698)

[CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ ĐỀ TÀI CÂN ĐIỆN TỬ 2](#_Toc453128699)

[1.1. Giới thiệu sơ lược về đề tài cân điện tử 2](#_Toc453128700)

[1.2. Giới thiệu các thiết bị phân cứng sử dụng cho cân điện tử 3](#_Toc453128701)

[1.2.1. Máy vi tính 3](#_Toc453128702)

[1.2.2. Mạch Arduino Uno R3 3](#_Toc453128703)

[1.2.3. Loadcell và mạch khuếch đại tín hiệu cân Hx711 5](#_Toc453128704)

[1.2.4. Mạch hiển thị -nút nhấn 7](#_Toc453128705)

[1.2.5. Động cơ servo 7](#_Toc453128706)

[CHƯƠNG 2: VIẾT GIAO DIỆN MÁY TÍNH CHO CÂN 9](#_Toc453128707)

[2.1. Giới thiệu chung về giao diện máy tính cho cân điện tử 9](#_Toc453128708)

[2.1.1. Giới thiệu phần mềm Visual studio 2012 9](#_Toc453128709)

[2.1.2. Giới thiệu chung về giao diện cho cân điện tử 9](#_Toc453128710)

[2.2. Lập trình chung cho form 10](#_Toc453128711)

[2.3. Nhận và sử lý dữ liệu đến và xuất ra màn hình 11](#_Toc453128712)

[2.4. Gửi tín hiệu điều khiển xuống Arduino 12](#_Toc453128713)

[2.5. Hiển thị trạng thái của cân bằng hình ảnh 13](#_Toc453128714)

[2.6. Tạo menuscrip và form thông tin 14](#_Toc453128715)

[CHƯƠNG 3: THIẾT KẾ, THỰC HIỆN PHẦN CỨNG VÀ LẬP TRÌNH CHO ARDUINO 16](#_Toc453128716)

[3.1. Thiết kế phần cứng 16](#_Toc453128717)

[3.1.1. Mạch hiển thị - nút nhấn 16](#_Toc453128718)

[3.1.2. Lắp ráp loadcell 20](#_Toc453128719)

[3.1.3. Lắp động cơ servo để điều khiển van 21](#_Toc453128720)

[3.2. Lập trình cho arduino 21](#_Toc453128721)

[3.2.1. Khái quát về lập trình cho mạch arduino 21](#_Toc453128722)

[3.2.2. Sơ đồ khối của chương trình 23](#_Toc453128723)

[3.2.3. Khai báo các hằng, biến và tạo lập các giá trị ban đầu 24](#_Toc453128724)

[3.2.4. Chương trình chính 25](#_Toc453128725)

[3.2.5. Kiểm tra thời gian nhấn nút và xử lý các trường hợp 27](#_Toc453128726)

[3.2.6. Nhận giá trị cân nặng, đặt góc cho động cơ servo 28](#_Toc453128727)

[3.2.7. Tính trung bình và hiển thị giá trị cân nặng, gửi dữ liệu quan cổng serial 29](#_Toc453128728)

[3.2.8. Xử lý tín hiệu điến 31](#_Toc453128729)

[CHƯƠNG 4: BÀI HỌC KINH NGHIỆM VÀ KẾT LUẬN 34](#_Toc453128730)

[4.1. Những kinh nghiệm thu thập được trong quá trình thực hiện đề tài 34](#_Toc453128731)

[4.1.1. Những kinh nghiệm về sử dụng arduino 34](#_Toc453128732)

[4.1.2. Kinh nghiệm về truyền dữ liệu 34](#_Toc453128733)

[4.1.3. Những kinh nghiệm về viết giao diện 35](#_Toc453128734)

[4.2. Kết luận 35](#_Toc453128735)

Tài liệu tham khảo

# LỜI MỞ ĐẦU

Để tạo cơ hội thực hành nắm rõ hơn những kiến thức về ngành học, tác giả đã chọn nghiên cứu và thực hiện đề tài: **Cân điện tử sử dụng mạch arduino có kết nối với giao diện máy tính**. Trong khuôn khổ đề tài, tác giả sẽ thực hiện làm mạch phần cứng, phần mềm và giao diện như sau:

**Mục tiêu của đề tài:** tạo cơ hội tìm hiểu và thực hành một cách thiết thực những nội dung lý thuyết trong các môn đã học, đặc biệt là môn Cấu trúc máy tính và giao diện môn Đo lường va điều khiển bằng máy tính. Và kế đó, áp dụng những nghiên cứu này vào một bài thực hành cụ thể để hiểu rõ hơn lý thuyết và thu góp kinh nghiệm thực hành thực tế.

**Đối tượng và phạm vi đề tài:** Sử dụng mạch arduino, loadcell cùng với các thiết bị điện khác để làm một cái cân có thể cân được một số vật dụng thông thường có những đặc điểm sau:

* Cân được vật nặng tối đa 5kg
* Sử dụng led đơn và led 7 đoạn hiển thị cân nặng đơn vị lạng (100 gam)
* Có nút nhấn dùng để lấy lại mức không (reset zero) khi sử dụng thêm đĩa cân hoặc vật chứa những gì cần cân trọng lượng
* Có thể kết nối với máy tính để theo dõi cân nặng, lấy lại mức không cho cân và gửi tín hiệu điều khiển xuống cơ cấu chấp hành của cân

**Phương pháp nghiên cứu:** Dựa trên những kiến thức đã học kết hợp với những kiến thức trong môn học cấu trúc máy tính và giao diện về mạch arduino và giao diện máy tính cùng với sự tìm tòi, tổng hợp kiến thức từ các nguồn khác để phân tích và chọn lựa những cách thức ứng dụng phù hợp phục vụ cho thực hiện đề tài.

**Ý nghĩa của đề tài:** Đề tài là một cây cầu gắn kết giữa lý thuyết học được và với việc thực hiện, tạo ra các sản phẩm thực tế để tăng kiến thức và tay nghề của học viên. Mặt khác, đề tài cũng là tài liệu tham khảo cho những nghiên cứu co liên quan hoặc áp dụng cho thực tế.

Bố cục đề tài được chia làm 4 phần như sau:

Chương 1: Giới thiệu chung về đề tài cân điện tử

Chương 2: Viết giao diện máy tính cho cân

Chương 3: Thiết kế, thực hiện phần cứng và lập trình cho arduino

Chương 4: Bài học kinh nghiệm và kết luận

## GIỚI THIỆU CHUNG VỀ ĐỀ TÀI CÂN ĐIỆN TỬ

### Giới thiệu sơ lược về đề tài cân điện tử

Đề tài cân điện tử được thực hiện dựa trên một số tiêu chí như sau:

**Phần cứng:**

* Tận dụng những linh kiện, chất liệu có sẵn để giảm thiểu chi phí
* Sản phẩm cần gọn, nhẹ, dễ kết nối và sử dụng
* Thiết kế đơn giản và bắt mắt

**Phần mềm:**

* Dùng những kiến thức về lập trình đã được học và kế thừa những dữ liệu, đoạn lệnh của những người đi trước, kết hợp và phát triển thành một phần mềm phù hợp cho phần cứng
* Dễ hiểu và dễ nghiên cứu
* Thời gian đáp ứng nhỏ, có tín hiệu thông báo khi reset zero

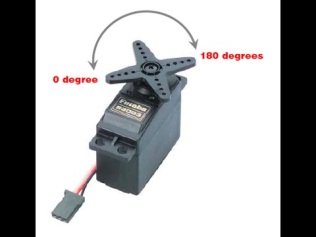
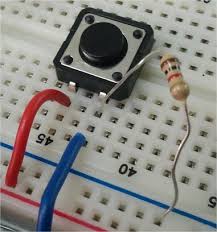
**Giao diện:**

* Đơn giản, bắt mắt, dễ quan sát các số liệu
* Có nút reset zero

Với những tiêu chí đó, cân được kết nối như hình sau:

(5)

Hình 1: Sơ đồ kế nối cân điện tử



Mô hình cân điện tử được tạo ra bởi sự kết nối giữa mạch arduino (2) với máy tính (1), mạch cân loadcell (3), mạch nút nhấn - hiển thị (4) và động cơ servo (5). Trong đó thì mạch ardruino làm nhân tố trung tâm để liên lạc giữa các thiết bị khác với nhau. Máy tính cùng với giao diện có khả năng đọc được những thông tin chỉ số mà arduino gửi lên và gửi các tín hiệu điều khiển đến arduino. Mạch cân loadcell có chức năng đọc giá trị cân nặng và gửi cho arduino. Mạch hiển thị- nút nhấn có chức năng hiển thị số bằng tín hiệu từ arduino và gửi tín hiệu trạng thái nút nhấn cho arduino. Động cơ servo có chức năng đóng mở van theo tín hiệu điều khiển từ arduino. Arduino có chức năng xử lý tín các tín hiệu đến từ máy tính, nút nhấn và loadcell cùng gửi tín hiệu đi và tín hiệu cho máy tính và tín hiệu điều khiển đến mạch hiển thị và động cơ servo.

(4)

(2)

(1)

(3)

### Giới thiệu các thiết bị phân cứng sử dụng cho cân điện tử

#### Máy vi tính

Máy vi tính là một thiết bị không thể thiếu trong cuộc sống hiện đại và cả trong nghiên cứu, nhất là trong ngành điện tử có lập trình.

Trong nội dung đề tài này, tác giả sử dụng một máy vi tính có cấu hình trung bình (chíp sử lý core 2 duo- ram 2GB) có cài sẵn một số phần mềm cần thiết:

* Phần mêm Arduino IDE dùng để rà soát lỗi và biên dịch lệnh cho arduino.
* Phần mềm hỗ trợ lập trình giao diện.
* Phần mềm Proteus: dùng để vẽ mạch nguyên lý và mạch in để làm mạch hiển thị-nút nhấn.

Internet cũng là một yếu tố quan trọng hỗ trợ cho việc viết chương trình nạp cho arduino thông qua các thư viện, các đoạn code mẫu và hướng dẫn khác.

Máy vi tính được kết nối với arduino qua cổng usb bằng cáp phụ kiện của arduino.

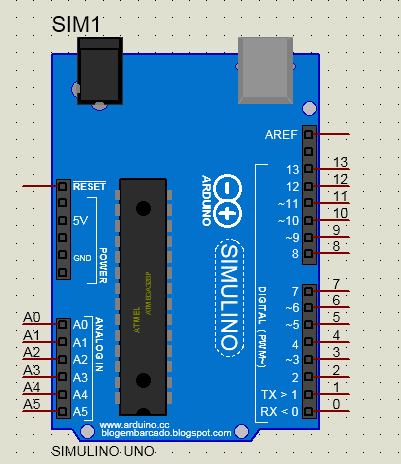
#### Mạch Arduino Uno R3

##### Hình dáng và công dụng của mạch Arduino Uno R3

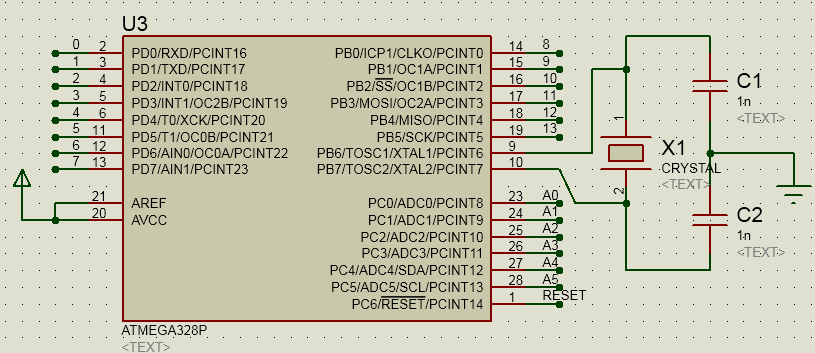
Arduino là một board mạch vi xử lý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau.

Được giới thiệu vào năm 2005, Những nhà thiết kế của Arduino cố gắng mang đến một phương thức dễ dàng, không tốn kém cho những người yêu thích, sinh viên và giới chuyên nghiệp để tạo ra những nhiết bị có khả năng tương tác với môi trường thông qua các cảm biến và các cơ cấu chấp hành. Những ví dụ phổ biến cho những người yêu thích mới bắt đầu bao gồm các robot đơn giản, điều khiển nhiệt độ và phát hiện chuyển động. Đi cùng với nó là một môi trường phát triển tích hợp (IDE) chạy trên các máy tính cá nhân thông thường và cho phép người dùng viết các chương trình cho Aduino bằng ngôn ngữ C hoặc C++.

Hình 1: Mạch Arduino Uno R3



Hình 2: Các chân kết nối của arduino R3



Hình 3: Các chân kết nối của chíp Atmega328 và các chân tương ứng trong arduino

Nhắc tới dòng mạch Arduino dùng để lập trình, cái đầu tiên mà người ta thường nói tới chính là dòng Arduino UNO. Hiện dòng mạch này đã phát triển tới thế hệ thứ 3 (R3). Dựa vào hình 2 và hình 3, chúng ta hoàn toàn có thể thiết kế một bo mạch rời với arduino và chạy bằng chíp Atmega328 đã nạp code bằng Arduino. Trong đề tài này, tác giả thực hiện thi công một bo mạch rời sử dụng chíp của arduino. Mạch này có thể hoạt động độc lập mà không cần sử dụng đến bo arduino để tiết kiệm chi phí. Khi cần kết nối với máy tính, cần sử dụng mạch chuyển đổi serial – Usb để kết nối với máy tính. Trong trường hợp giao tiếp với các chíp vi điều khiển khác, có thể giao tiếp trực tiếp bằng cổng serial.

##### Những thông số kỹ thuật và đặc điểm của mạch Arduino Uno R3

Vi điều khiển ATmega328 họ 8bit

Điện áp hoạt động 5V DC (chỉ được cấp qua cổng USB)

Tần số hoạt động 16 MHz

Dòng tiêu thụ khoảng 30mA

Điện áp vào khuyên dùng 7-12V DC

Điện áp vào giới hạn 6-20V DC

Số chân Digital I/O 14 (6 chân hardware PWM)

Số chân Analog 6 (độ phân giải 10bit)

Dòng tối đa trên mỗi chân I/O 30 mA

Dòng ra tối đa (5V) 500 mA

Dòng ra tối đa (3.3V) 50 mA

Bộ nhớ flash 32 KB (ATmega328) với 0.5KB dùng bởi bootloader

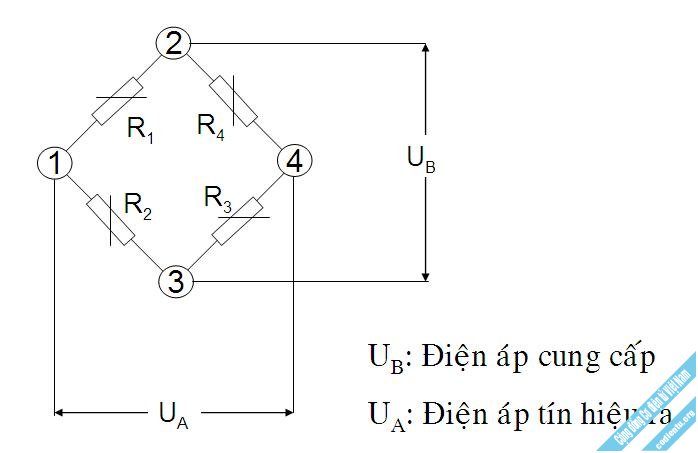
SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

#### Loadcell và mạch khuếch đại tín hiệu cân Hx711

##### Loadcell 5kg

Cấu tạo chính của loadcell gồm các điện trở strain gauges R1, R2, R3, R4 kết nối thành 1 cầu điện trở Wheatstone như hình dưới và được dán vào bề mặt của thân loadcell.



Hình 4: Sơ đồc ấu tạo loadcell

Một điện áp kích thích được cung cấp cho ngõ vào loadcell (2 góc (1) và (4) của cầu điện trở Wheatstone) và điện áp tín hiệu ra được đo giữa hai góc.

Tại trạng thái cân bằng (trạng thái không tải), điện áp tín hiệu ra là số không hoặc gần bằng không khi bốn điện trở được gắn phù hợp về giá trị.

Đó là lý do tại sao cầu điện trở Wheatstone còn được gọi là một mạch cầu cân bằng.

Khi có tải trọng hoặc lực tác động lên thân loadcell làm cho thân loadcell bị biến dạng (giãn hoặc nén), điều đó dẫn tới sự thay đổi chiều dài và tiết diện của các sợi kim loại của điện trở strain gauges dán trên thân loadcell dẫn đến một sự thay đổi giá trị của các điện trở strain gauges. Sự thay đổi này dẫn tới sự thay đổi trong điện áp đầu ra.

Sự thay đổi điện áp này là rất nhỏ, do đó nó chỉ có thể được đo và chuyển thành số sau khi đi qua bộ khuếch đại của các bộ chỉ thị cân điện tử (đầu cân).

Trong khuôn khổ đề tài, tác giả đã chọn sử dụng loại loadcell có khả năng cân nặng 5kg có những thông số như sau:

Hình 5: Loadcell 5Kg

Điện áp điều khiển: 5-10v

Tín hiệu đầu ra: tín hiệu điện áp

Kích thước: 12,7 x 12.7 x 75mm

Loadcell có 4 dây có màu lần lượt là đen, đỏ, trắng và xanh, trong đó có 2 dây để cấp nguồn và 2 dây cấp tín hiệu ra. Tùy loại loadcell và nhà sản xuất mà các dây này có chức năng khác nhau nhưng thường thì cặp dây đỏ - đen là dây cấp nguồn, dây đỏ cấp nguồn dương và dây đen nối mass, 2 dây còn lại là dây tín hiệu (có thể phát hiện chính xác 2 dây còn lại dây nào là dây tín hiệu dương và dây nào là dây tín hiệu âm bằng cách mắc thử mạch và nếu tín hiệu cân ra là âm thì đảo 2 dây này lại thì sẽ không còn hiện tượng này nữa.

Loadcell thực tế nhóm sử dụng có dây đỏ là dây nguồn dương, dây đen là dây trung tính, dây trắng là dây tín hiệu dương và dây xanh lục là dây tín hiệu âm.

Hình 6: Mạch khuếch đại cân nặng Hx711

##### Mạch khuếch đại tín hiệu cân nặng Hx711

Hx711 là mạch đọc giá trị cảm biến loadcell với độ phân giải 24bit và chuyển sang giao tiếp 2 dây (clock và data) để gửi dữ liệu cho vi điều khiển /arduino.

Thông số kỹ thuật:

* Điện áp hoạt động: 2.7 – 5V
* Dòng điện tiêu thụ: <1.5mA
* Tốc độ lấy mẫu: 10-80 mẫu trên một giây (Samples Per Second)
* Độ phân giải mẫu: 24 bit ADC
* Độ phân giải điện áp: 40mV
* Kích thước: 38x21x10 mm

#### Mạch hiển thị -nút nhấn

Mạch hiển thị và nút nhấn được tạo ra từ các linh kiện điện tử thông thường như led 7 đoạn, led đơn, nút nhấn, biến trở và điện trở. Boar được thiết kế phù hợp để gắn trực tiếp vào board arduino và có tạo các cổng phụ để dễ dàng kết nối với nguồn điện hay làm trung gian để mạch loadcell và động cơ servo kết nối với mạch arduino.

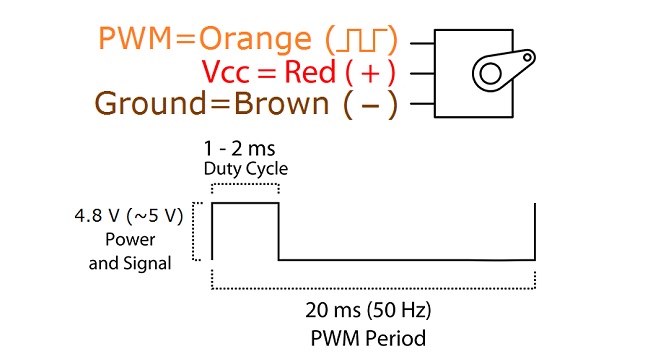
Ngoài ra, để tiết kiệm chân kết nối với arduino, mạch có dùng thêm chíp 74595 (chíp ghi dịch) kết hợp trong mạch hiển thị.

#### Động cơ servo

Động cơ servo là một loại động cơ có thể điều chỉnh được góc quay bằng tín hiệu điều khiển thông qua dây điều khiển.

Động cơ servo được chia làm nhiều loại, chủ yếu dựa vào góc có thể quay được (180 độ hoặc 360 độ) hay dựa vào chất lượng (loại thường: sử dụng bánh răng nhựa, loại tốt: dùng bánh răng kim loại). Trong đề tài này, tác giả sử dụng động cơ servo SG90 có thể quay được góc 180 độ và sử dụng bánh răng nhựa để tiết kiệm chi phí.

Để kết nối động cơ servo với arduino, chỉ cần phân biệt được 3 dây của động cơ servo gồm một dây nguồn (màu đỏ), dây trung tính (màu nâu) và dây nhận tín hiệu điều khiển (màu vàng) như hình dưới.



Hình 7: Chức năng các dây kết nối động cơ servo

Như vậy, dây đỏ sẽ được kết nối với chân 5v của arduino, dây nâu nối với chân GND của arduino và dây cam sẽ nối với một chân tín hiệu tương tự (PWM) của arduino.

## VIẾT GIAO DIỆN MÁY TÍNH CHO CÂN

### Giới thiệu chung về giao diện máy tính cho cân điện tử

#### Giới thiệu phần mềm Visual studio 2012

Phần mềm Visual studio 2012 là một phiên bản phần mềm Visual studio khá mới do công ty phần mềm Microsoft cung cấp. Phần mềm có thể giúp để lập trình tạo ra những giao diện (phần mềm máy tính) một cách khá dễ dàng và tiện lợi. Trong đề tài này, giao diện cân điện tử sẽ được viết bằng phần mềm visual studio bằng ngôn ngữ Visual basic.



Hình 8: Logo của phần mềm visual studio 2012

#### Giới thiệu chung về giao diện cho cân điện tử

Giao diện cân điện tử được viết ra có các thành phần như trong giao diện sau:



Hình 9: Giao diện cân điện tử

Vì giao diện mang tính hỗ trợ cho việc nghiên cứu nên trong giao diện vẫn giữ lại phần nội dung truyền nhận để dễ theo dõi và chỉnh sửa trong thời gian nghiên cứu.

Ngoài những gì có thể thấy trực tiếp trên giao diện, phần mềm còn sử dụng 2 timer, 1 cổng serial và 1 menu scrip. Việc lập trình cho từng phần sẽ được trình bày trong các mục tiếp sau đây.

### Lập trình chung cho form

Phần lập trình chung cho form có mục đích để khai báo biến và hằng dùng chung cho cả phần mềm, khai báo cổng com và quy định cho cổng com hoạt động. Phần lập trình này được viết như sau:

Public Class Form1

;khai báo biến và hằng

Private readBuffer As String = String.Empty

Private comOpen As Boolean

Private nhangiatridat As String = String.Empty

Private cannang As String = String.Empty

Private dem As Integer = 0

;khai báo cổng com cùng những thông số để cổng hoạt động

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

' read avaiable COM Ports:

Dim Portnames As String() = System.IO.Ports.SerialPort.GetPortNames

If Portnames Is Nothing Then

MsgBox("There are no Com Ports detected!")

Me.Close()

End If

' device params

With SerialPort1

.ParityReplace = &H3B ' replace ";" when parity error occurs

.PortName = **"COM3"** 'cboComPort.Text ;Tên cổng com rất quan trọng

.BaudRate = CInt("**9600**") 'CInt(cboBaudRate.Text)

.Parity = IO.Ports.Parity.None

.DataBits = 8

.StopBits = IO.Ports.StopBits.One

.Handshake = IO.Ports.Handshake.None

.RtsEnable = False

.ReceivedBytesThreshold = 1 'threshold: one byte in buffer > event is fired

.NewLine = vbCr ' CR must be the last char in frame. This terminates the SerialPort.readLine

.ReadTimeout = 10000

End With

' check whether device is avaiable:

Try

SerialPort1.Open()

comOpen = SerialPort1.IsOpen

Catch ex As Exception

comOpen = False

MsgBox("Error Open: " & ex.Message)

End Try

End Sub

Khi đã có cổng com, ta viết chương trình để phần mềm nhận tín hiệu từ arduino gửi lên máy tính.

### Nhận và sử lý dữ liệu đến và xuất ra màn hình

Vì dữ liệu từ arduino gửi lên sẽ có trình tự là <giá trị cân nặng > ;<góc quay servo> ;<giá trị đặt > vì thế, việc nhận dữ liệu và tách các dữ liệu này ra là cần thiết.

Private Sub SerialPort1\_DataReceived(sender As Object, e As IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived

If comOpen Then

Try

readBuffer = SerialPort1.ReadLine()

'data to UI thread

Me.Invoke(New EventHandler(AddressOf DoUpdate))

Catch ex As Exception

MsgBox("read " & ex.Message)

End Try

End If

End Sub

''' <summary>

''' update received string in UI

''' </summary>

''' <remarks></remarks>

Public Sub DoUpdate(ByVal sender As Object, ByVal e As System.EventArgs)

readBuffer = Trim(readBuffer)

Txt\_nhan.Text = readBuffer

; lưu dữ liệu nhận được vào Txt\_nhan.Text

cannang = Mid(readBuffer, 1, InStr(readBuffer, ";") - 1)

; gán giá trị từ ký tự 1 đến trước dấu ; của Txt\_nhan.Text vào cannang

hienthicannang.Text= Mid(cannang, 2)

; xuất giá trị cân ra hienthicannang

readBuffer = Mid(readBuffer, InStr(readBuffer, ";") + 1)

; xóa từ đầu đến dấu ; trong readBuffer

trangthaivan.Text = Mid(readBuffer, 1, InStr(readBuffer, ";") - 1)

; xuất giá trị chuỗi trước dấu ; trong readBuffer ra trangthaivan

nhangiatridat = Mid(readBuffer, InStr(readBuffer, ";") + 1)

; gán giá trị đặt bằng nội dung sau dấu ; trong readBuffer

sailech.Text = cannang – nhangiatridat

; gán giá trị sailech bằng sự chênh lệch giữa cân nặng và giá trị đặt

Timer1.Enabled = True ;khởi động timer 1

End Sub

Ngoài đoạn code này, còn có 1 đoạn code nhỏ của timer 1 có liên quan đến hoạt động nhận dữ liệu để báo khi hết thời gian nhận tín hiệu như sau:

Private Sub Timer1\_Tick(ByVal sender As System.Object, \_

ByVal e As System.EventArgs) Handles Timer1.Tick

Timer1.Enabled = False

End Sub

Như vậy, tới đây thì khi có dữ liệu gửi lên, giao diện đã sử lý được và hiện lên màn hình. Kế đến, cần lập trình cho giao diện có khả năng gửi những yêu cầu xuống arduino để điều khiển.

### Gửi tín hiệu điều khiển xuống Arduino

Để điều khiển được arduino, cần phải quy ước một số ký hiệu khi gửi xuống để khi lập trình arduino, cần phải có sự đồng nhất này để tín hiệu gửi đi không bị hiểu nhầm.

|  |  |  |
| --- | --- | --- |
| Nút nhấn | Chuỗi gửi xuống arduino | Ý nghĩa |
| Reset | r | Reset zero cho cân |
| Mở | m | Mở van 100% |
| Đóng | d | Đóng van |
| Tự động | t | Bật chế độ tự động |
| Đặt | s# | Set (đặt lại) giá trị đặt bằng con số sau ký tự s. (# có thể là 1 hay nhiều chữ số) |

Mỗi lệnh gửi đi sẽ tương ứng với 1 nút nhấn trên giao diện và cần được lập trình riêng. Tuy nhiên, cấu trúc các câu lệnh là tương tự nhau. Nút reset được lập trình như sau:

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Reset.Click

Txt\_truyen.Text = "r"

If comOpen Then SerialPort1.WriteLine(Txt\_truyen.Text)

End Sub

Đoạn code trên thực hiện 2 công việc khi nhấn nút là gán nội dung của Txt\_truyen là ký tự r. sau đó gửi nội dung đó bằng cổng nối tiếp. Như vậy, đoạn code sẽ được rút gọn đi nếu không sử dụng hộp Txt\_truyen.Text để hiển thị nội dung truyền như sau:

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Reset.Click

If comOpen Then SerialPort1.WriteLine(“r”)

End Sub

Tuy nhiên, để tiện cho việc theo dõi, tác giả sẽ chọn cách đầu là cho thể hiện ở Txt\_truyen.Text để theo dõi.

Các nút khác cũng được lập trình tương tự:

Nút nhấn mở:

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Mo.Click

Txt\_truyen.Text = "m"

If comOpen Then SerialPort1.WriteLine(Txt\_truyen.Text)

tudong.Enabled = True

End Sub

Nút nhấn đóng:

Private Sub Button5\_Click(sender As Object, e As EventArgs) Handles dong.Click

Txt\_truyen.Text = "d"

If comOpen Then SerialPort1.WriteLine(Txt\_truyen.Text)

tudong.Enabled = True

End Sub

Nút nhấn mở và đóng có thêm một dòng lệnh đổi chế độ enabled của nút nhấn tudong thành True để khi nhấn 1 trong 2 nút này thì nút tudong se nổi lên, cho phép nút này hoạt động.

Nút nhấn tudong:

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles tudong.Click

Txt\_truyen.Text = "t"

If comOpen Then SerialPort1.WriteLine(Txt\_truyen.Text)

tudong.Enabled = False

End Sub

Khi nút nhấn tudong được nhấn thì giao diện sẽ gửi đi chữ t và đồng thời vô hiệu hóa chính nó.

Nút nhấn dat:

Private Sub Button6\_Click(sender As Object, e As EventArgs) Handles Dat.Click

If comOpen Then

Txt\_truyen.Text = "s"

Txt\_truyen.Text += giatridat.Text

SerialPort1.Write("s")

SerialPort1.Write(giatridat.Text)

End If

End Sub

Nút nhấn dat (đặt) có chức năng gửi đi chuỗi gồm có chữ s và chuỗi số trong textbox giatri dat. Vì thế, cần dùng 2 lệnh gửi đi (lệnh SerialPort1.Write), lần đầu gửi chữ s, lần sau gửi giá trị trong hộp thoại giá trị đặt. Đồng thời, cũng phải hiển thị ra trong Txt\_truyen chuỗi gửi đi, vì thế ta dùng lệnh gán cộng thêm: += để nối chuỗi s với nội dung trong giatridat.

Ngoài ra, nút gửi và nút xóa cũng được lập trình:

Private Sub Btn\_gui\_Click(sender As Object, e As EventArgs) Handles Btn\_gui.Click

If comOpen Then SerialPort1.WriteLine(Txt\_truyen.Text)

End Sub

Private Sub Btn\_xoa\_Click(sender As Object, e As EventArgs) Handles Btn\_xoa.Click

Txt\_nhan.Text = " "

Txt\_truyen.Text = " "

End Sub

### Hiển thị trạng thái của cân bằng hình ảnh

Trong giao diện còn có một khung hình ảnh được điều khiển bằng timer phụ thuộc vào giá trị của biến trangthaivan.text. Khi trangthaivan.text =0 thì hiển thị hình d không thay đổi. Nhưng nếu trangthaivan.text khác 0 thì sẽ hiển thị luân phiên hình a, b và c để tạo hiệu ứng ảnh động. Để làm được việc này, cần dùng thêm một biến dem và một timer (timer2). Đoạn code được viết trong timer2 như sau:

Private Sub Timer2\_Tick(sender As Object, e As EventArgs) Handles Timer2.Tick

If (trangthaivan.Text = 0) Then dem = 0 Else dem = dem + 1

If (dem = 0) Then hinhhienthi.Image = System.Drawing.Image.FromFile \_

("E:\Hinh can\d.jpg")

If (dem = 1) Then hinhhienthi.Image = System.Drawing.Image.FromFile \_

("E:\Hinh can\a.jpg")

If (dem = 2) Then hinhhienthi.Image = System.Drawing.Image.FromFile \_

("E:\Hinh can\b.jpg")

If (dem = 3) Then

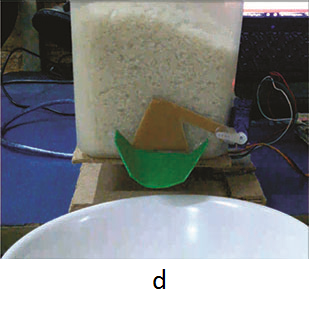
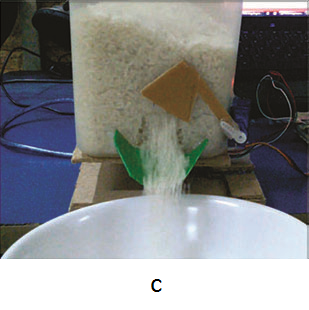
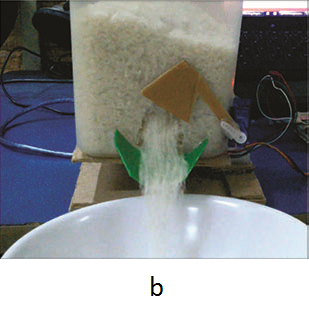
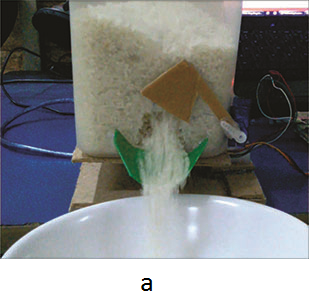
hinhhienthi.Image = System.Drawing.Image.FromFile \_

("E:\Hinh can\c.jpg")

dem = 0

End If

End Sub

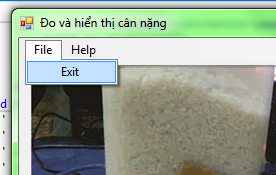
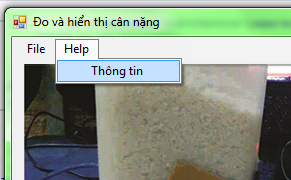


Hình 10: Hình ảnh nguồn cho giao diện

Đoạn code này có ý nghĩa như sau: khi timer tác động thì kiểm tra trạng thái van, nếu trangthaivan bằng 0 thì cho biến dem =0, nếu trangthaivan khác 0 thì tăng biến đếm lên 1 đơn vị. Sau đó xét biến dem, nếu biến dem =0 thì cho hiện lên hình d, nếu là 1 cho hiện hình a, nếu là 2 thì cho hiện hình b, nếu là 3 thì cho hiện hình c và đặt lại dem=0.

### Tạo menuscrip và form thông tin

Có thể tạo thêm menuscrip như trong 2 hình sau và lập trình cho mỗi phần như biên dưới:



Hình 11: Menu scrip cho giao diện

Lập trình cho nút Exit:

Private Sub ExitToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles ExitToolStripMenuItem.Click

Close()

End Sub

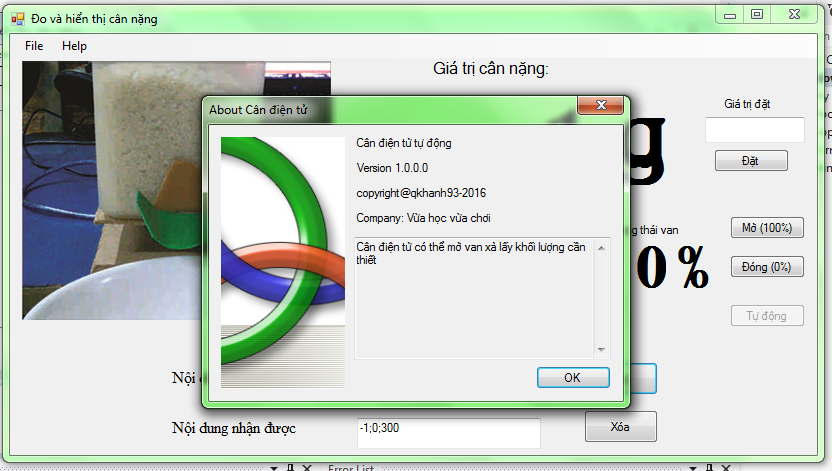
Lập trình cho nút thông tin:

Private Sub ThôngTinToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles ThôngTinToolStripMenuItem.Click

AboutBox1.Show()

End Sub

Trong đoạn code lập trình cho nút thông tin có sử dụng một giao diện phụ khác có tên là AboutBox1. Ta có thể tạo giao diện phụ bằng cách vào mục Project chọn mục Add window form… để khi bấm vào thông tin, sẽ có một hộp thoại mới mở ra, hiển thị những thông tin cần thiết như hình:



Hình 12: Cửa sổ phụ hiển thị thông tin

Tiếp sau đây, phần lập trình arduino và thiết kế phần cứng cũng rất quan trọng để có một sản phẩm hoàn chỉnh.

## THIẾT KẾ, THỰC HIỆN PHẦN CỨNG VÀ LẬP TRÌNH CHO ARDUINO

### Thiết kế phần cứng

#### Mạch hiển thị - nút nhấn

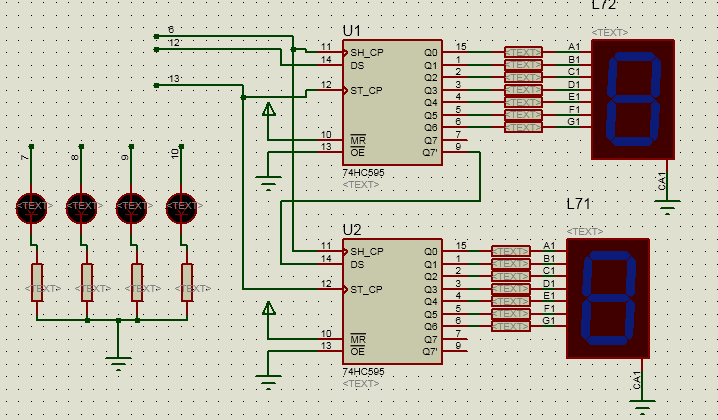
##### Mạch hiển thị

Để có thể hiển thị kết quả cân nặng ra, cần lắp một mạch hiển thị. Chúng ta có thể sử dụng các led đơn, led 7 đoạn hay màn hình để biểu diễn kết quả. Mạch arduino có 14 chân in/out (từ chân 0 đến chân 13), trong đó có hai chân 1 và 2 dùng để truyền tín hiệu serial, như vậy chúng ta có thể sử dụng 11 chân còn lại (chân số 2 đến chân 13) để điều khiển 11 led đơn. Tuy nhiên, trong thực tế, người ta cần điều khiển được nhiều led hơn hay nhiều thiết bị hiển thị hơn để có thể truyền tải được nhiều nội dung hơn. Trong trường hợp này, cần dùng thêm chip giải mã để có thể thực hiện việc này mà sử dụng ít chân của ic hơn hoặc sử dụng những chân analog có thể đưa ra nhiều mức tín hiệu để điều khiển nhiều led hơn.

Trong khuôn khổ đề tài, tác giả chọn sử dụng 4 con led đơn và 1 con led 7 đoạn (loại Anot chung) để tận dụng những thiết bị sẵn có và đồng thời tận dụng tối đa số chân ra cua arduino. Sơ đồ nối dây được thực hiện giống như hình 6.

Các led đơn đều được nối với một điện trở hạn dòng, một đầu nối với chân điều khiển của arduino, chân còn lại được nối với nguồn (tín hiệu điều khiển tích cực mức thấp) hoặc với đất (tín hiệu điều khiển tích cực mức cao).

Các led trong led 7 đoạn cũng được mắc tương tự, nhưng phụ thuộc vào led 7 đoạn thuộc loại anot chung hay katot chung mà phải nối xuống đất hay lên nguồn, nếu mắc sai, led 7 đoạn sẽ không hoạt động. Cụ thể trong trường hợp này dùng led 7 đoạn anot chung, vì thế chúng ta bắt buộc phải nối chân chung này lên nguồn và nối đầu còn lại mỗi led với một con điện trở hạn dòng và nối vào chân điều khiển. Người ta cũng có thể nối chân chung với một chân điều khiển để điều khiển led 7 đoạn trong trạng thái hoạt động hay không. Trong trường hợp này, nếu chân chung được nối với chân điều khiển thì khi chân điều khiển ở mức 5v thì led 7 đoạn hoạt động và nếu mức điện áp ở chân này là 0v thì led 7 đoạn sẽ không hoạt động.



Hình 2: Sơ đồ nối dây mạch hiển thị (tên dây được đặt trùng với cổng kết nối với arduino)

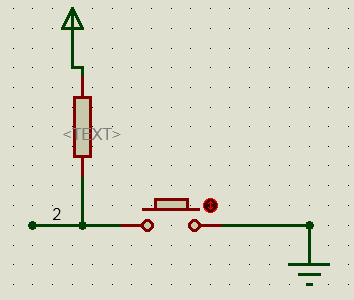
Như vậy, dựa vào sơ đồ kết nối dây này, có thể làm một board mạch để hiển thị chỉ số cân nặng.

##### Nút nhấn

Khi thiết kế các hệ thống, đôi khi cần dùng đến một vài thiết bị để điều khiển. Đối với cân điện tử, điều đó là cần thiết khi muốn đưa chỉ số cân hiện tại về mức không khi thêm đĩa cân hoặc vật dụng chứa đồ vật cần cân.

Để arduino nhận được tín hiệu này chúng ta cần mắc nút nhấn như hình 7. Nút nhấn có thể kết nối với bất kỳ chân in nào của arduino. Trong đề tài này, tác giả chọn kết nối với chân số 2 để có thể sử dụng tính năng ngắt ngoài của arduino.

Quy tắc hoạt động của nút nhấn như sau: khi nút nhấn đang ở trạng thái bình thường, chân nhận tín hiệu (ở đây là chân số 2) được nối với nguồn thông qua một điện trở, vì dòng vào các chân input rất nhỏ nên điện áp ở chân này coi như ở mức 5v. Khi nhấn nút, điện áp ở chân nhận tín hiệu này được đưa xuống mức thấp (0v). Điện trở đóng vai trò cản trở bớt dòng điện từ nguồn xuống mass để tránh tình trạng ngắn mạch xảy ra khi nhấn nút. Nếu thiếu điện trở này, khi nhấn nút sẽ xảy ra ngắn mạch.



Hình 13: Sơ đồ nối dây nút nhấn (nối với chân số 2 arduino)

##### Biến trở

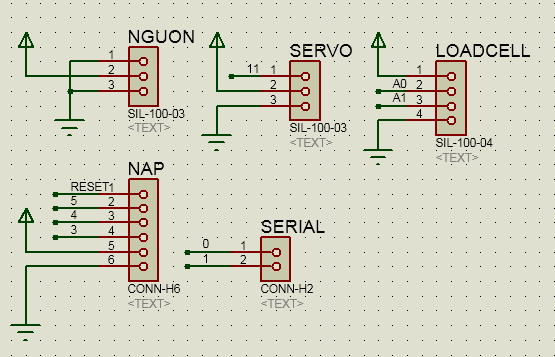
Hình 14: Sơ đồ kết nối biến trở

Khi hoạt động, cần có một thiết bị để có thể điều chỉnh được giá trị đặt cho cân. Có thể dùng một biến trở được mắc một cách đơn giản như hình 8 và kết nối chân tín hiệu vào một chân nhận tín hiệu tương tự. Ở đây, tác giả chọn kết nối với chân A5.

Như vậy, trạng thái của biến trở sẽ được truyền trực tiếp vào chân A5 của arduino. Trạng thái này chúng ta có thể thay đổi được, dẫn đến ta có thể đặt được giá trị mong muốn cho arduino.

##### Các cổng kết nối

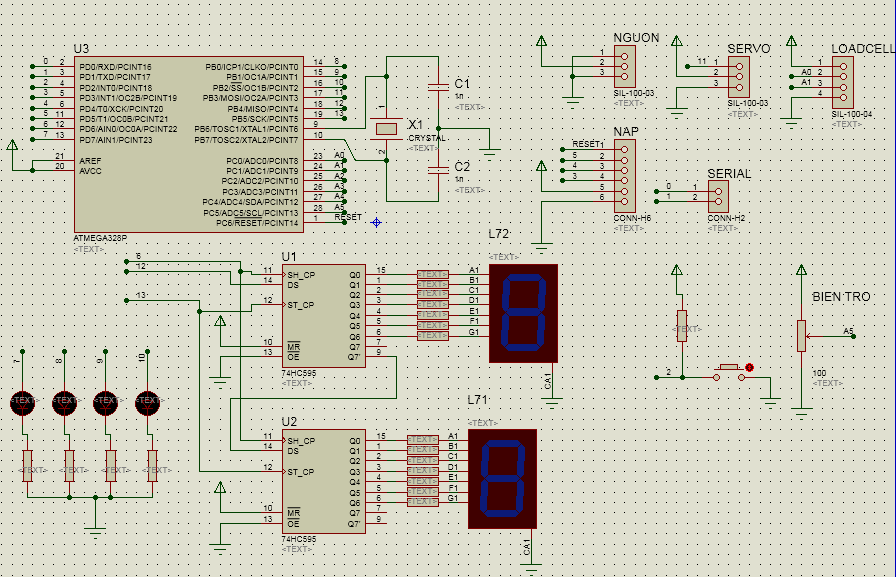
Khi làm mạch, cần chú ý tạo các cổng kết nối để thuận tiện cho việc liên kết với các bộ phận khác một cách dễ dàng. Trong đề tài này, cần làm 5 cổng kết nối như trong hình bao gồm cổng cấp nguồn, cổng serial để truyền nhận tín hiệu nối tiếp, cổng kết nối với động cơ servo, cổng kết nối với mạch HX711 để nhận tín hiệu từ loadcell và một cổng để nạp chương trình cho chíp bằng mạch arduino (sử dụng khi nghiên cứu viết chương trình hoặc muốn nạp chương trình cho chíp mà không muốn gỡ chíp ra khỏi mạch).



Hình 15: Các cổng kết nối

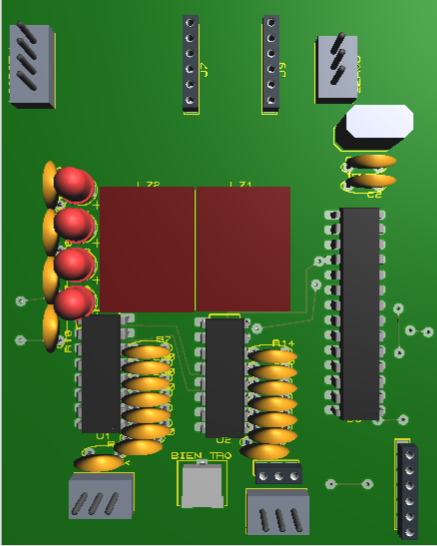
##### Làm mạch và gắn linh kiện

Sau khi đã kết hợp tất cả các thành phần của mạch điện thì ta được mạch nguyên lý như sau:



Hình 16: Mạch nguyên lý của cân điện tử

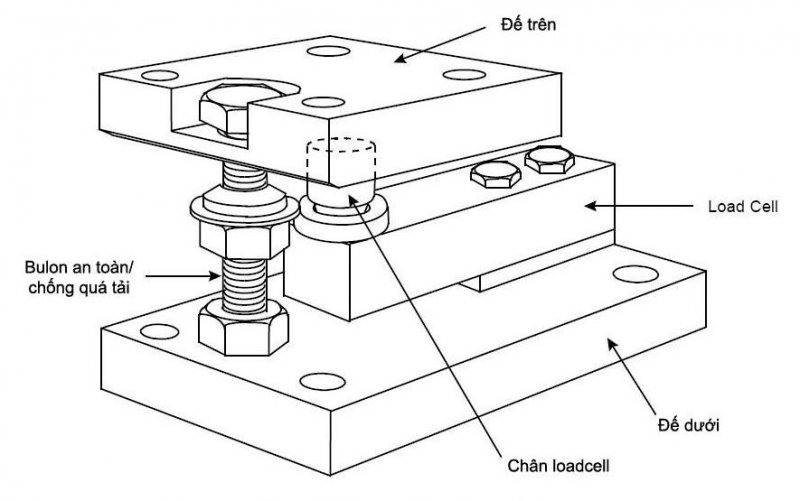
Sau đó, thiết kế và làm mạch in, sau đây là kết quả mô phỏng 3d sử dụng phần mềm proteus.



Hình 17: Mô phỏng bo mạch sản phẩm 3 chiều trên Proteus

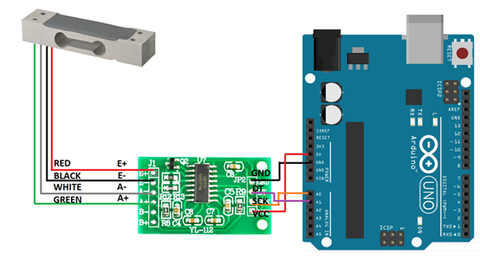
#### Lắp ráp loadcell

Để loadcell có thể hoạt động an toàn và chính xác được, loadcell cần được lắp ráp đúng cách. Cách lắp ráp loadcell được biểu diễn như hình sau. Theo đó, loadcell loại thanh cần được lắp với đế dưới gắn với một đầu của loadcell và đế trên gắn với đầu còn lại, cả 2 đế đều đặt song song với chiều dài của loadcell và có khoảng trống để loadcell có thể biến dạng vừa phải. Một bulon an toàn cần được lắp ráp để bảo đảm không làm hỏng loadcell khi có quá tải xảy ra.



Hình 18: Sơ đồ lắp ráp loadcell

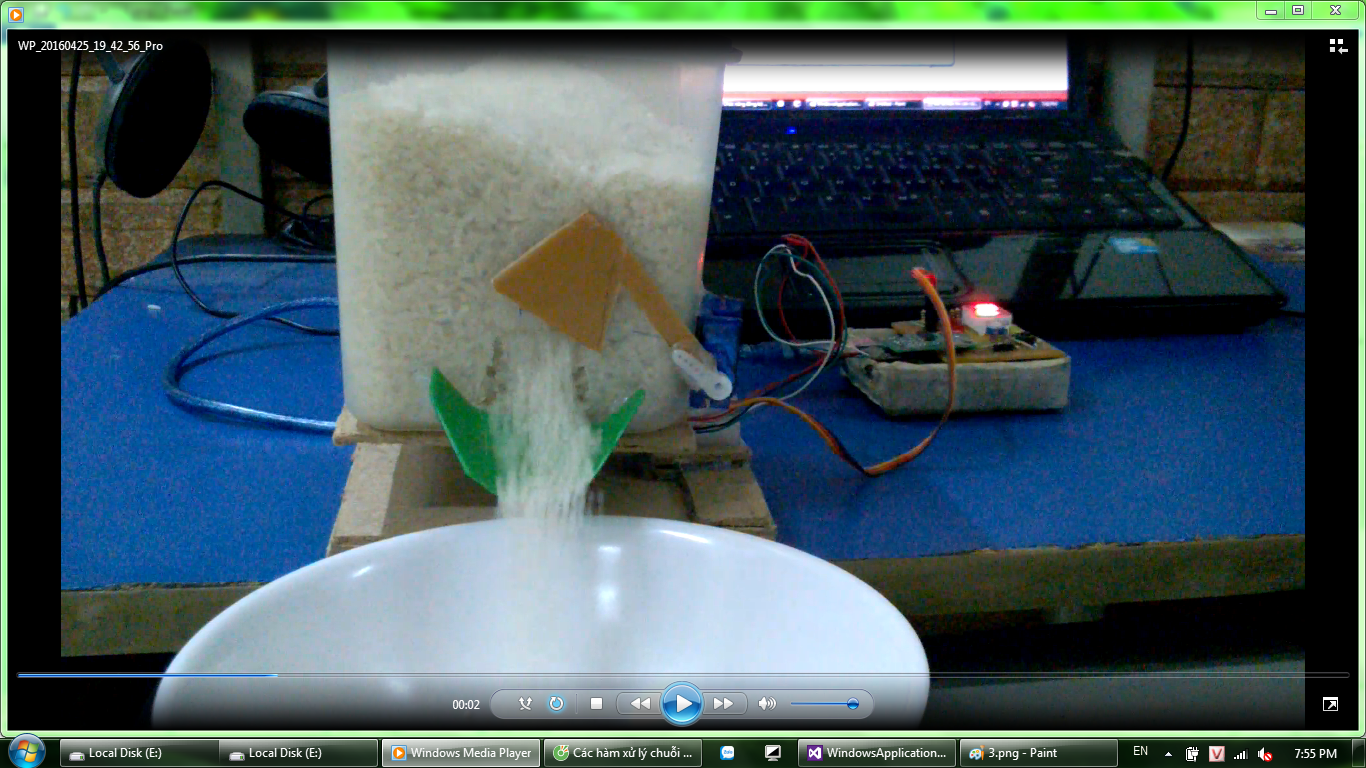
Để kết nối loadcell với mạch arduino, chúng ta kết nối như hình sau: (tùy chức năng dây của loadcell mà có thể nối dây hơi khác so với hình miễn là đúng chức năng của dây thì mạch mới hoạt động chính xác).



Hình 3: Sơ đồ kết nối loadcell- Hx711- Arduino  
(thực tế sản phẩm dây trắng và xanh lá được đổi chỗ cho nhau)

#### Lắp động cơ servo để điều khiển van

Động cơ servo được gắn dính liền với thùng cân (trong đề tài này là thùng cân gạo) tại góc vuông của thùng để cơ cấu tác động của động cơ có thể điều khiển được một cái nắp van gắn liền với nó như hình dưới.



Hình 19: Động cơ servo được gắn vào thùng cân để điều khiển van

### Lập trình cho arduino

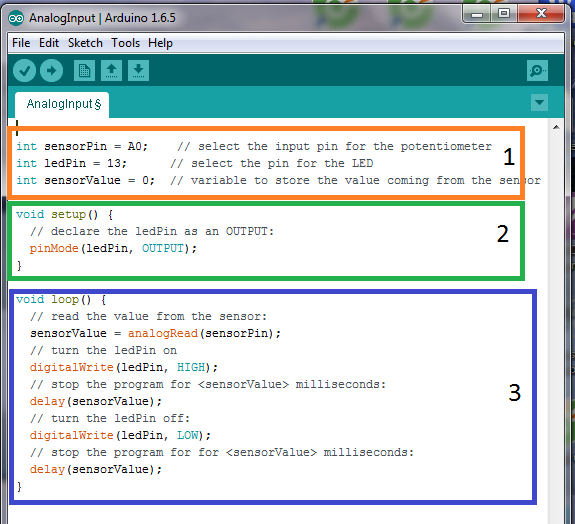
#### Khái quát về lập trình cho mạch arduino

Việc lập trình cho arduino được cho là có nhiều nét tương đồng với c++ và được tạo ra nhằm giúp những người không có nhiều kiến thức về vi điều khiển cũng có thể tiếp cận và ứng dụng chúng cách dễ dàng.

Để lập trình cho arduino chỉ cần cài phần mềm Arduino IDE, driver kèm theo và kết nối mạch arduino với máy tính thông qua cổng usb là đã có thể viết code và nạp trực tiếp vào arduino.

Phần mềm arduino được chia làm 3 phần (xem hình 8):

* Phần 1: Khai báo chung
* Phần 2: Đoạn lệnh chỉ chạy 1 lần (được đặt trong void setup())
* Phần 3: Đoạn lệnh chạy lập lại (được đặt trong void lop())



Hình 4: Các phần của code arduino

Để dễ theo dõi, tác giả sẽ trình bày việc viết code này theo từng nhóm lệnh xử lý những công việc riêng. Sau đó mới kết nối vào với nhau thành một chương trình hoàn chỉnh.

#### Sơ đồ khối của chương trình

0

>500

<500

Khai báo các chân vào, ra và khởi động các chương trình đi kèm (2)

Kiểm tra thời gian nhấn nút (3)

Reset zero (5)

Nhập giá trị đặt (4)

Nhận giá trị cân nặng và đặt góc cho động cơ servo (6)

Tính trung bình và hiển thị giá trị cân nặng, gửi dữ liệu quan cổng serial (7)

Khai báo thư viện, biến và hằng (1)

Xử lý tín hiệu đến (8)

Sự kiện nối tiếp (9)

Ngắt ngoài  
(nhấn nút) (10)

Nhận tín hiệu đến

Đếm thời gian nhấn nút

Void setup

Void loop

Hình 20: Sơ đồ khối chương trình trên aruidno

Từng phần của chương trình sẽ được trình bày những điểm cốt yếu và đồng thời, những chương trình con hoặc các hàm sử dụng trong đó cũng sẽ được trình bày trong các mục sau.

#### Khai báo các hằng, biến và tạo lập các giá trị ban đầu

Như thông thường, tất cả các biến và hằng cùng với các thư viện và thủ tục đều phải được thông báo và trong mục void setup thì những thủ tục ban đầu cần được thực hiện. Phần này tương ứng với mục (1) và (2) trong sơ đồ khối chương trình. Sau đây là đoạn chương trình cho phần này.

#include <HX711.h> // Khai báo thư viện của mạch hx711

#include <EEPROM.h> //Khai báo thư viện EEPROM

#include <Servo.h> //Khai bao thu vien cua dong co servo

Servo myServo; // Đặt tên động cơ servo

HX711 scale(A0, A1); // khai bao 2 chan tin hieu cua loadcell ket noi voi 2 chan analog 0 va analog 1

int ledPins[] = {6}; // khai bao cac chan ket noi led 7 doan theo thu tu abcdefg

int ledle[]={10,9,8,7}; // Khai bao cac chan ket noi voi katot cua 4 led don chi thi

int nutnhan =2; // Nút nhấn được kết nối với chân số 2

int bientro = A5;// Biến trở- chân A5

int chot = 13; // Chân chốt tín hiệu 7495- 13

int led7doan = 12; //Chân nhận tín hiệu nối tiếp 7495- 12

int ck =6;// Xung ck 7495 - 6

boolean macc[14][8]= //mã led 7 đoạn catot chung

{

{1,1,1,1,1,0,1,0},//0- ABCDEFGHDi

{0,1,1,0,0,0,0,0},//1

{1,1,0,1,1,1,0,0},//2

{1,1,1,1,0,1,0,0},//3

{0,1,1,0,0,1,1,0},//4

{1,0,1,1,0,1,1,0},//5

{1,0,1,1,1,1,1,0},//6

{1,1,1,0,0,0,0,0},//7

{1,1,1,1,1,1,1,0},//8

{1,1,1,1,0,1,1,0},//9

{1,0,0,1,0,1,0,0},//10 ngang

{0,1,1,0,1,0,1,0},//11 doc

{0,0,0,0,0,1,0,0},//12 vài kiểu khác

{0,0,0,0,0,0,0,0},//13

};

const int gocmin=167; //đặt góc khi động cơ đóng van

const int gocmax=140;// Đặt góc khi động cơ mở hoàn toàn van

int save = 0; //Một số biến sử dụng và giá trị khởi tạo

int zero=0;

int giatridat;

float cannang;

int lamtron;

float tong=0;

int tonggoc=0;

int dem=0;

float trungbinh;

int trungbinhgoc;

String inputString = "";

boolean stringComplete = false;

boolean tudong =1;

boolean nhangiatridat =0;

int gocmoservo;

int giatrinhan=0;

char kytudau;

int demthoigian=0;

**void setup()** // Chương trình chỉ chạy 1 lần khởi động

**{**

for(int index = 3; index < 14; index++) //khai báo chân tín hiệu ra

{pinMode(index, OUTPUT);}

pinMode(2, INPUT); // khai báo chân tín hiệu vào

scale.set\_scale(2280.f); //thủ tục cho loadcell

scale.tare();

Serial.begin(9600); //khai báo cổng nối tiếp tốc độ truyền 9600bit/s

inputString.reserve(200); //khai báo dung lượng dự trữ cho biến inputString

myServo.attach(11); //Khai báo chân điều khiển servo

myServo.write(gocmin); //Đóng van

for (int i=0; i<6; i++) // Nhay led de bao reset

{

songuyen(5); tled7doan(11, 11); delay (50);

songuyen(6); tled7doan(10, 10); delay (50);

}

giatridat=EEPROM.read(save)\*100; //lấy lại giá trị đặt trong epprom

serialEvent(); //gọi chức năng sự kiện nối tiếp

attachInterrupt(0, ngat, LOW);//Khai báo ngắt ngoài 0

}

#### Chương trình chính

Chương trình chính của cân điện tử được lập trình bao gồm các phần từ (3) đến (8). Trong đoạn trương trình này sử dụng nhiều chương trình con sẽ được trình bày ở mục tiếp theo. Chương trình chính của cân điện tử được viết như sau :

**void loop()**

{

while(1)//Tạo vòng lặp vô hạn

{

serialEvent(); //Gọi chức năng sự kiện nối tiếp

if(demthoigian !=0) //Kiểm tra thời gian nhấn nút (3)

{

if (demthoigian<500){

demthoigian=0;

resetzero (zero); //(5)

}

else {

demthoigian=0;

nhapgiatridat(giatridat); //(4)

}

}

cannang=scale.get\_units()\*5-zero; //Nhận giá trị và xử lý (6)

lamtron= round(cannang/10)\*10;

if(tudong==1)gocmoservo=goc(giatridat, cannang);

myServo.write(gocmoservo); //Tính góc quay cho servo

tonggoc+=gocmoservo;

tong+=cannang;

dem+=1;

if(dem>=5)

{

trungbinh=tong\*1.0/dem; //tính trung bình (7)

trungbinhgoc=tonggoc/dem;

trungbinhgoc=(gocmin-trungbinhgoc)\*100/(gocmin-gocmax);

Serial.print(trungbinh,0); //Gửi dữ liệu qua cổng serial

Serial.print(";");

Serial.print(trungbinhgoc);

Serial.print(";");

Serial.println(giatridat);

hienthi(trungbinh); //Hiển thị ra cơ cấu hiển thị

dem=0;

tong=0;

tonggoc=0;

}

if (trungbinh<-50 and dem==4) //phần sử lý hiển thị khi chỉ số âm

tled7doan(12, 12);

if(stringComplete == true){ // Xử lý tín hiệu đến (8)

xulydulieuden(inputString);

stringComplete = false;

inputString="";

}

}

}

Cần tạo một vòng lặp vô hạn bao bọc tất cả các lệnh trong chương trình chính để các biến sử dụng trong chương trình không bị mất giá trị.

#### Kiểm tra thời gian nhấn nút và xử lý các trường hợp

Trong phần kiểm tra thời gian nhấn nút (3), biến demthoigian sẽ được kiểm tra để phân trường hợp sẽ thực hiện việc nhập giá trị đặt (4) hay resetzero (5). Biến demthoigian là giá trị đếm được tương ứng với thời gian nhấn nút được thực hiện thông qua chương trình ngắt ngoài điều khiển bằng nút nhấn. Việc nhập giá trị đặt hay resetzero được viết trong các chương trình con riêng. Sau khi phân trường hợp xong, nếu biến demthoigian khác 0 sẽ được đưa về 0 ngay vì trong chương trình con nhập giá trị đặt cần phải sử dụng lại biến này. Nếu không, chương trình con thực hiện nhập giá trị đặt sẽ không hoạt động như mong muốn.

if(demthoigian !=0) //Kiểm tra thời gian nhấn nút (3)

{

if (demthoigian<500){

demthoigian=0;

resetzero (zero); //(5)

}

else {

demthoigian=0;

nhapgiatridat(giatridat); //(4)

}

}

Trong thời gian nhấn nút thì cứ 50ms thì biến thời gian sẽ được tăng lên 1 đơn vị. Đoạn chương trình ngắt sau sẽ thể hiện công việc của chíp điều khiển khi nhấn nút:

void ngat()

{

demthoigian+=1;

delay(50);

}

Chương trình con resetzero được viết như sau:

void resetzero (int &zero) // lay lai muc zero

{

zero=scale.get\_units()\*5; // gan zero bang can nang hien tai

for (int i=0; i<7; i++){ // Nhay led de bao reset

songuyen(5); tled7doan(10,10); delay (50);

songuyen(6); tled7doan(11,11); delay (50);

}

}

Chương trình con nhập giá trị đặt được viết như sau:

void nhapgiatridat (int &giatridat)

{

while (1) //tao vong lap mai mai de bien zero khong bi xoa

{

hienthi(round(analogRead(bientro)\*5/100)\*100);

if(demthoigian!=0)

{

demthoigian=0;

giatridat=(round(analogRead(bientro)\*5/100)\*100); //gan giatrihienthi cho giatridat, Lam tron toi hang tram

EEPROM.write(save, giatridat/100); // Luu giatri dat vao eeprom

for(int n=0; n<3; n++) //hiển thị báo hiệu nhận giá trị

{

hienthi(round(analogRead(bientro)\*5/100)\*100); delay (200);

songuyen(0); tled7doan(13,13); delay(100);

}

break;

}

}

}

(xem chương trình con hienthi trong mục 3.2.7)

#### Nhận giá trị cân nặng, đặt góc cho động cơ servo

Code :

cannang=scale.get\_units()\*5-zero; //Nhận giá trị (6)

lamtron= round(cannang/10)\*10;

if(tudong==1)gocmoservo=goc(giatridat, cannang);

myServo.write(gocmoservo);

tonggoc+=gocmoservo;

tong+=cannang;

dem+=1;

Công việc nhận giá trị cân nặng và đặt góc cho động cơ servo thực hiện nhờ một số câu lệnh được đánh dấu (6) trong đoạn chương trình chính. Trong đó, câu lệnh nhận giá trị cân nặng từ loadcell là cầu lệnh :

cannang=scale.get\_units()\*5-zero;

Trong đó, giá trị nhận được được gọi bằng hàm scale.get\_units(), sau đó được nhân 5 vì loadcell thuộc loại 5kg và trừ đi biến zero là biến căn chỉnh cho việc resetzero tương ứng với giá trị cân nặng của đĩa cân. Mặc định ban đầu giá trị của biến zero bằng 0 vì mặc định, thư viện hx711 sẽ trả về giá trị cân nặng khi khởi động hệ thống (dù có vật nặng trên cân hay không) là 0. Biến này sẽ được thay đổi trong chương trình con resetzero.

Việc điều khiển động cơ servo phụ thuộc vào chế độ hoạt động của van là tự động hay bằng tay. Mặc định là chế độ tự động, chế độ này chỉ được thay đổi bằng tín hiệu gửi từ cổng nối tiếp. Có thể tham khảo phần xử lý tín hiệu đến để hiểu rõ hơn về chế độ này.

Trong khi xử lý góc quay cho động cơ servo, có sử dụng một hàm là hàm góc được viết như sau :

int goc (int giatridat, int lamtron)

{

int goccanquay;

int doquay=gocmin-gocmax;

if (lamtron <= giatridat + 5) return gocmin;

else if (lamtron >= giatridat + 60) return gocmax;

else goccanquay= (lamtron-giatridat)\*(lamtron-giatridat)/150+8.5;

return gocmin-goccanquay;

}

Hàm góc có kiểu trả về là kiểu interger với giá trị phụ thuộc vào 2 biến là biến giá trị cân nặng được làm tròn : lamtron và giá trị đặt : giatridat. Ngoài ra, còn sử dụng 2 hằng số quy định góc quay tối thiểu và tối đa cho động cơ. Hàm sẽ giới hạn và tính toán góc quay cho động cơ bắt đầu tăng goc mở khi có sai lệch bắt đầu từ lớn hơn 5g và sẽ mở góc tối đa khi khối lượng cân được lớn hơn giá trị đặt 60g. Hàm tính toán là hàm bậc hai, được viết dựa vào thử nghiệm thực tiễn.

#### Tính trung bình và hiển thị giá trị cân nặng, gửi dữ liệu quan cổng serial

Code :

if(dem>=5)

{

trungbinh=tong\*1.0/dem; //tính trung bình (7)

trungbinhgoc=tonggoc/dem;

trungbinhgoc=(gocmin-trungbinhgoc)\*100/(gocmin-gocmax);

Serial.print(trungbinh,0); //Gửi dữ liệu qua cổng serial

Serial.print(";");

Serial.print(trungbinhgoc);

Serial.print(";");

Serial.println(giatridat);

hienthi(trungbinh); //Hiển thị ra cơ cấu hiển thị

dem=0; //xóa các biến liên quan để chuẩn bị cho đợt lấy mẫu mới

tong=0;

tonggoc=0;

}

Việc xử lý dữ liệu đến bao gồm việc làm tròn dữ liệu, cân chỉnh dữ liệu, lấy mẫu, và tính trung bình cho các đại lượng giá trị cân nặng và giá trị góc quay của động cơ servo. Trong đoạn dữ liệu này cũng tính toán và điều khiển góc quay cho động cơ servo để đóng mở van. Việc lấy mẫu ở đây được thực hiện 5 lần và sau đó tính trung bình rồi hiển thị và gửi đi.

Dữ liệu được gửi đi bao gồm giá trị cân nặng đơn vị gam, góc của động cơ servo đơn vị phần trăm và giá trị đặt đơn vị gam. Các con số này được gửi dưới dạng chuỗi và gửi chung với nhau trong một lần gửi cách nhau bằng dấu chấm phẩy ( ;).

Trong phần này, quan trọng nhất là chương trình con hiển thị có chức năng xuất dữ liệu ra cơ cấu hiển thị. Đoạn chương trình con hiển thị được viết như sau :

**void hienthi(float n)**

{

if (n<0) n=n\*-1;

int t=round(n/10); //lấy 3 số đầu của dãy 5 số thể hiện cân nặng

int donvi=t%10; //tách lấy số thứ 3

t=t/10;

int chuc=t%10; // tách lấy số thứ 2

t=t/10;

int songuyen1=t%10; // tách lấy số đầu tiên

songuyen(songuyen1);

tled7doan(chuc,donvi);

}

Chương trình con hienthi có chức năng tách lấy ra trong tham số n 3 con số đầu tiên để hiển thị ra cơ cấu hiển thị, cụ thể lấy số đầu tiên sẽ được thể hiện bằng 4 led đơn (sử dụng chương trình con songuyen), số thứ 2 và số thứ 3 được thể hiện ra 2 con led 7 đoạn (sử dụng chương trình con tled7doan). Hai đoạn chương trình con này được viết như sau :

**void songuyen(int b)** //led chi thi 0-4

{

if(b==5) // b=5 va 6 la hai cach hien thi de the hien khi reset muc zero

{ digitalWrite(ledle[0], HIGH);

digitalWrite(ledle[1], LOW);

digitalWrite(ledle[2], HIGH);

digitalWrite(ledle[3], LOW);}

else if (b==6)

{digitalWrite(ledle[0], LOW);

digitalWrite(ledle[1], HIGH);

digitalWrite(ledle[2], LOW);

digitalWrite(ledle[3], HIGH);}

else

for (int i=0; i<5; i++) // khi b = 0 ko co led sang, b =1 led 1 sang...

{

if(i<b) digitalWrite(ledle[i], HIGH);

else digitalWrite(ledle[i], LOW);

}

}

**void tled7doan(int chuc, int donvi)**

{

for (int i=7; i>=0; i--){ //Truyen 7 bit don vi

digitalWrite(ck, LOW);

digitalWrite(led7doan, macc[donvi][i]);

digitalWrite(ck, HIGH);

}

for (int i=7; i>=0; i--){ //Truyen 7 bit hang chuc

digitalWrite(ck, LOW);

digitalWrite(led7doan, macc[chuc][i]);

digitalWrite(ck, HIGH);

}

digitalWrite(chot, LOW); // cho phep hien thi

digitalWrite(chot, HIGH);

}

Trong chương trình con **songuyen** ngoài việc hiển thị các con số từ 0 đến 4 tương ứng với 0 led sáng đến 4 led sáng thì 4 led còn hiển thị giá trị 5 và 6 để phục vụ cho các trường hợp thông báo, nhấp náy.

Hàm **tled7doan** dùng các lệnh digitalWrite để điều khiển 3 chân vào cho ic74595 : ic ghi dịch có chức năng chuyển đổi tín hiệu nối tiếp thành tính hiệu song song, rất có ích trong việc mở rộng các chân tín hiệu số đầu ra. Ở đây, tác giả sử dụng 2 con ic 74595 mắc kiểu nối tiếp nhau nên chỉ cần điều khiển 3 chân là chân tín hiệu, chân chốt và chân xung ck. Như vậy, mỗi ic 74595 điều khiển được 7 chân tương ứng với 1 led 7 đoạn.

#### Xử lý tín hiệu điến

Code :

if(stringComplete == true) // Xử lý tín hiệu đến (8)

{

xulydulieuden(inputString); //gọi chương trình con xulydulieuden

stringComplete = false; //xóa các biến liên quan để nhận đợt dữ liệu mới

inputString="";

}

Việc xử lý tín hiệu đến chỉ được thực hiện khi biến stringComplete có giá trị là true và công việc chủ yếu nằm ở chương trình con xulydulieuden, chương trình con này được viết như sau :

void xulydulieuden (String x)

{

char kytudautien = x.charAt(0); //tách lấy ký tự đầu tiên

int giatrinhan=0;

if (kytudautien=='r') resetzero (zero);

if (kytudautien=='m') {

tudong=0;

gocmoservo=gocmax;

}

if (kytudautien=='d'){

tudong=0;

gocmoservo=gocmin;

}

if (kytudautien=='t') tudong=1;

if (kytudautien=='s'){

x.remove(0,1);

giatrinhan= x.toInt();

giatridat=giatrinhan;

EEPROM.write(save, giatridat/100); // Luu giatri dat vao eeprom

for(int n=0; n<3; n++)

{

hienthi(round(giatridat\*1.0/100)\*100); delay (200);

songuyen(0); tled7doan(13,13);; delay(100);

}

}

}

Trong đó, x là tham trị của chương trình con được đặt vào bằng chuỗi nhận được từ cổng nối tiếp. sau đó công việc so sánh ký tự đầu tiên được thực hiện và thực hiện các công việc tương ứng. Riêng với lệnh đặt giá trị đặt mới (khi nhận được chữ s đầu tiên) thì cần tiếp tục nhận chuỗi số sau chữ s để lấy giá trị đặt. Hàm resetzero đã được trình bày trong mục 2.3.5

Phải kể đến là một đoạn code nhỏ phục vụ cho việc nhận dữ liệu nối tiếp đến, đó là chương trình con serial event :

void serialEvent() {

while (Serial.available())

{

// get the new byte:

char inChar = (char)Serial.read();

// add it to the inputString:

inputString += inChar;

} stringComplete = true;

}

Như vậy, mỗi khi có tín hiệu nối tiếp đến thì chíp sẽ thực hiện lưu chuỗi nhận được vào biến inputString và đặt cho biến stringComplete = true để chuẩn bị cho việc xử lý tín hiệu đến được trình bày ở phần trên.

## BÀI HỌC KINH NGHIỆM VÀ KẾT LUẬN

### Những kinh nghiệm thu thập được trong quá trình thực hiện đề tài

Trong quá trình thực hiện đề tài, có nhiều khó khăn và những bất cập xảy ra. Chính những lúc đó mới làm cho nhóm tác giả học hỏi thêm được nhiều kinh nghiệm và lối giải quyết vấn đề cũng như nhận thức được sự yếu kém của chính mình. Vì thế cũng xin được nêu ra những kinh nghiệm học được trước hết là để lưu giữ lại, sau là để làm tài liệu tham khảo cho những người muốn tìm hiểu về những vấn đề tương tự.

#### Những kinh nghiệm về sử dụng arduino

###### Cảm nhận được khả năng ứng dụng rất lớn từ việc áp dụng arduino

Arduino là một môi trường mới có khả năng phát triển rất mạnh nhờ sự đơn giản trong cách lập trình của nó. Những thư viện có sẵn và những đoạn code mẫu giúp cho những người thực hiện đề tài cảm thấy họ không thực hiện đề tài một mình, mà luôn được hỗ trợ bởi một cộng đồng đông đảo. Lập trình trên arduino chủ yếu sử dụng ngôn ngữ C++ vì thế rất dễ hiểu và sử dụng biến đổi code mẫu làm việc theo ý của mình.

###### Có thể mở rộng số chân của arduino bằng những con chíp rẻ tiền

Các bạn hoàn toàn có thể mở rộng số chân của arduino chỉ bằng 2 chân sẵn có của nó bằng một hay nhiều con chíp ghi dịch 74595. Việc lập trình để điều khiển những chân mở rộng này không khó khăn. Bạn chỉ cần tạo ra một mảng chứa các biến logic ứng với các chân đã mở rộng, viết một chương trình con để xuất nội dung các chân ra và bạn có thể sử dụng được các chân đó bằng cách điều chỉnh trạng thái chân đó bằng cách thay đổi bit tương ứng của chân đó và gọi hàm xuất.

###### Arduino có các thư viện chuyên dụng cho mỗi chức năng cá biệt

Khi cần làm việc gì đó hơi cá biệt thì các bạn nên tìm hiểu các thư viện của arduino xem có ai đã phát triển thư viện cho việc đó chưa. Nếu có rồi thì công việc của bạn đơn giản chỉ là thêm thư viện đó vào Arduino IDE và tìm hiểu các hàm của thư viện đó và sử dụng.

#### Kinh nghiệm về truyền dữ liệu

Hiểu hơn về việc truyền thông tin thông qua cổng com và việc mã hóa cũng như tách đoạn code để lấy thông tin từ chuỗi nhận được.

###### Nối tắt chân TX và RX

Có kinh nghiệm trong việc thiết lập trình truyền nhận bằng cách nối chân truyền và chân nhận để biết được tình trạng làm việc của cổng. Sau đó mới phát triển liên kết với đối tượng khác.

###### Mạch USB to COM miễn phí

Có thể dùng mạch arduino làm một mạch biến đổi cổng USB thành cổng COM với giá 0 đồng bằng cách gỡ chíp vi điều khiển của arduino ra và sử dụng chân 0 và 1 của mạch làm chân truyền và nhận nối tiếp.

#### Những kinh nghiệm về viết giao diện

Trước khi học môn Cấu trúc máy tính và giao diện thì việc lập trình một giao diện trên máy tính là một việc xa vời. Tuy nhiên, khi học môn này thì việc này lại trở nên dễ dàng và có thể thực hiện được trong tầm tay những dự án nhỏ theo ý mình. Tuy rằng tài liệu tìm hiểu về các hàm của ngôn ngữ VB (Visual Basic) hơi khó khăn nhưng cũng không quá khó để viết lệnh trong ngôn ngữ này vì trên phần mềm có hỗ trợ sẵn.

###### Nên đặt tên cho các chi tiết để khỏi phải lục lọi lại

Một kinh nghiệm nho nhỏ khi làm việc trên phần mềm Visual Studio 2012 là bạn nên đặt tên cho các chi tiết trong giao diện mới của bạn để giảm thời gian tìm lại chúng khi cần thiết.

###### Gửi project cho người khác phải đầy đủ

Khi đóng gói và gửi project của mình đến cho người khác để xem, bạn cần phải gửi cả file có đuôi là .sln và folder chứa dữ liệu của project đó. Folder này được đặt tên bằng tên của project của bạn khi bạn tạo nó.

###### Label canh lề phải

Khi làm việc với label để thể hiện con số thì bạn cần phải chú ý điều này một chút là bình thường khi để cho label hiện số, nó sẽ canh lề trái, đây là điều không mong muốn. Khi gặp vấn đề này, bạn cần quan tâm đến thuộc tính AutoSize của label và chỉnh giá trị của nó về false thì label sẽ hiển thị cố định trong phạm vi của nó và canh lề phải.

### Kết luận

Với những gì đã tìm hiểu và thực hiện được, tác giả cảm thấy rất phấn chấn vì đề tài này là lần đầu tiên mở đường cho những nghiên cứu áp dụng thực tiễn trong ngành điện- điện tử. Xin chân thành cảm ơn sự đồng hành và hỗ trợ rất tâm huyết và thân thiện của các thầy trong khoa điện-điện tử. Tuy nhiên, bài báo cáo còn rất sơ sài và chắc chắn còn nhiều thiếu xót, mong quý thầy cô và bạn đọc góp ý chỉnh sửa để bài báo cáo này thiết thực hơn. Cuối cùng, xin cảm ơn quý thầy cô và các bạn đã đọc bài báo cáo này.

# TÀI LIỆU THAM KHẢO

1. Giáo trình cấu trúc máy tính và giao diện của Tiến sỹ Nguyễn Đức Thành
2. Arduino cookbook 2nd edition
3. Trang web cộng đồng arduino: arduino.vn