

## LỜI MỞ ĐẦU

Ngày nay kỹ thuật vi điều khiển đã trở nên quen thuộc trong các ngành kỹ thuật và trong dân dụng. Các bộ vi điều khiển có khả năng xử lý nhiều hoạt động phức tạp mà chỉ cần một chip vi mạch nhỏ, nó đã dần thay thế các tủ điều khiển lớn và phức tạp bằng những mạch điện gọn nhẹ, dễ dàng thao tác sử dụng.

Vi điều khiển không những góp phần vào kỹ thuật điều khiển mà còn góp phần to lớn vào việc phát triển thông tin. Chính vì các lý do trên, việc tìm hiểu, khảo sát vi điều khiển là điều mà các sinh viên ngành điện mà đặc biệt là chuyên ngành kỹ thuật điện-điện tử phải hết sức quan tâm. Đó chính là một nhu cầu cần thiết và cấp bách đối với mỗi sinh viên, đề tài này được thực hiện chính là đáp ứng nhu cầu đó.

Để góp phần đáp ứng nhu cầu trên và đóng góp thêm giải pháp thay thế các tủ điều khiển lớn và phức tạp, sau một thời gian dưới sự giảng dạy của các thầy cô trường Đại học Bà Rịa - Vũng Tàu, đồng thời được sự giúp đỡ nhiệt tình của thầy Nguyễn Lương Thanh Tùng và các bạn cùng khoa, tôi đã thiết kế, chế tạo "***Mô hình phân loại sản phẩm theo chiều cao dùng Arduino.***"

Do thời gian, kiến thức và kinh nghiệm của tôi còn có hạn nên sẽ không thể tránh khỏi những sai sót. Tôi rất mong được sự giúp đỡ và tham khảo ý kiến của thầy cô và các bạn nhằm đóng góp phát triển thêm đề tài.

## MỤC LỤC

Đề mục	Trang
<b>PHIẾU GIAO ĐỀ TÀI ĐỒ ÁN TỐT NGHIỆP</b>	
<b>LỜI CAM ĐOAN</b>	
<b>LỜI CẢM ƠN</b>	
<b>NHẬN XÉT</b>	
<b>Chương 1: TỔNG QUAN ĐỀ TÀI</b> .....	9
1.1 Lý do chọn đề tài: .....	9
1.2 Giới hạn đề tài: .....	9
1.3 Nguồn tư liệu: .....	9
1.4 Nhiệm vụ nghiên cứu: .....	10
<b>Chương 2: TỔNG QUAN CÁC LINH KIỆN CHÍNH</b> .....	11
<b>2.1 ArduinoUno R3</b> .....	11
2.1.1 Giới thiệu .....	11
2.1.2 Uno .....	13
2.1.3 Cấu trúc, thông số .....	14
<b>2.2 IC 74HC595</b> .....	20
2.2.1 Thế nào là dịch và chốt? .....	20
2.2.2 Sơ đồ & chức năng các chân 74HC595 .....	21
2.2.3 Giảm đồ thời gian về cách hoạt động của IC .....	23
<b>2.3 E18-D80NK</b> .....	24
2.3.1 Giới thiệu .....	24
2.3.2 Thông số kỹ thuật .....	25
2.3.3 Sơ đồ dây .....	25
<b>2.4 LED 7 đoạn</b> .....	26
2.4.1 Giới thiệu .....	26
2.4.2 Cấu tạo & Nguyên lý hoạt động .....	26
2.4.3 Trở hạn dòng: .....	27
<b>2.5 Motor DC</b> .....	28

2.5.1 Giới thiệu .....	28
2.5.2 Thông số kỹ thuật .....	28
2.5.3 Cấu tạo & Hoạt động .....	29
<b>2.6 Module LM2596 .....</b>	<b>30</b>
2.6.1 Giới thiệu .....	30
2.6.2 Thông số kỹ thuật .....	30
2.6.3 Nguyên lý hoạt động .....	31
<b>2.7 Motor Servo.....</b>	<b>32</b>
2.7.1 Giới thiệu .....	32
2.7.2 Hoạt động & Cấu tạo .....	33
2.7.3 Thông số kỹ thuật .....	34
2.7.4 Điều biến độ rộng xung.....	35
2.7.5 Giới hạn quay .....	36
2.7.6 Phân loại và các kích thước đặc biệt .....	36
<b>Chương 3: MÔ HÌNH ĐẾM &amp; PHÂN LOẠI SẢN PHẨM .....</b>	<b>37</b>
<b>3.1 Giới thiệu .....</b>	<b>37</b>
<b>3.2 Các phần mềm thiết kế .....</b>	<b>38</b>
3.2.1 Arduino .....	38
3.2.2 Proteus 8 Professional .....	39
<b>3.3 Sơ đồ khối .....</b>	<b>40</b>
<b>3.4 Nguyên lý hoạt động .....</b>	<b>42</b>
<b>3.5 Mạch in.....</b>	<b>44</b>
<b>3.6 Lưu đồ thuật toán .....</b>	<b>45</b>
<b>3.7 Code – chương trình.....</b>	<b>46</b>
<b>Chương 4: KẾT LUẬN .....</b>	<b>53</b>
<b>4.1 Ưu/Nhược điểm .....</b>	<b>53</b>
<b>4.2 Hướng phát triển.....</b>	<b>53</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>54</b>

## **Chương 1: TỔNG QUAN ĐỀ TÀI**

### **1.1 Lý do chọn đề tài:**

Với sự phát triển của xã hội, khoa học kỹ thuật nói chung và vi điều khiển nói riêng ngày càng được ứng dụng ở hầu hết các lĩnh vực.

Trong nhiều lĩnh vực được quan tâm, có một lĩnh vực về vi điều khiển được quan tâm rất nhiều hiện nay đó là vi điều khiển AVR. Một trong số những biến thể phổ biến của AVR là Arduino.

Việc tìm hiểu và ứng dụng hết khả năng của nhiều loại Arduino là cả một quá trình dài lý thú và hữu ích, vì sự thuận tiện, tinh gọn, khả năng phát triển cũng như sự đa dạng các dòng sản phẩm phù hợp nhiều quy mô ứng dụng của nó.

Một ý tưởng khác được quan tâm đông đảo trên các diễn đàn học tập ngành điện tử và tự động hóa, nhưng chưa có một tài liệu chính thống phổ biến hướng dẫn hay cung cấp thông tin về nó, cũng như chưa được giảng dạy ở nhiều trung tâm đó là ứng dụng Arduino trong sản xuất.

Trước thực tiễn ấy, tôi đã quyết định chọn đề tài này nhằm tìm hiểu về vấn đề đếm và phân loại sản phẩm qua ứng dụng của Arduino.

### **1.2 Giới hạn đề tài:**

Trong phạm vi đồ án này, tôi xin trình bày sơ lược về cấu tạo cũng như nguyên lý hoạt động của Mô hình phân loại và đếm sản phẩm theo chiều cao sử dụng Arduino Uno R3, 74HC595, led 7 đoạn...

### **1.3 Nguồn tư liệu:**

Dựa vào mục đích tìm hiểu, phạm vi giới hạn và đối tượng nghiên cứu; trong quá trình thực hiện, đề tài sử dụng nguồn tư liệu như sau:

- Các tài liệu kỹ thuật về cấu trúc, nguyên lý hoạt động.
- Các tài liệu về trang thiết bị điện tử.
- Tài liệu về lập trình lập trình, mã hóa vi điều khiển.

#### **1.4 Nhiệm vụ nghiên cứu:**

Mục đích nghiên cứu thông qua đề tài là tìm hiểu về ứng dụng đếm và phân loại sản phẩm nên nhiệm vụ nghiên cứu gồm:

- Tìm hiểu cơ chế hoạt động.
- Phân tích sơ đồ nguyên lý.
- Nâng cao kỹ năng lập trình vi điều khiển, hàn mạch, làm sản phẩm điện tử.
- Phát triển khả năng tư duy cho sinh viên trong quá trình nghiên cứu.

## **Chương 2: TỔNG QUAN CÁC LINH KIỆN CHÍNH**

### **2.1 ArduinoUno R3**

#### **2.1.1 Giới thiệu**

Arduino được khởi động vào năm 2005 như là một dự án dành cho sinh viên tại Interaction Design Institute Ivrea (Viện thiết kế tương tác Ivrea) tại Ivrea, Italy. Cái tên "Arduino" đến từ một quán bar tại Ivrea, nơi một vài nhà sáng lập của dự án này thường xuyên gặp mặt.

Các thiết bị dựa trên nền tảng Arduino được lập trình bằng ngôn ngữ riêng. Ngôn ngữ này dựa trên ngôn ngữ Wiring được viết cho phần cứng nói chung trên một môi trường phát triển tích hợp (IDE) chạy trên các máy tính cá nhân. Và Wiring lại là một biến thể của C/C++. Một số người gọi nó là Wiring, một số khác thì gọi là C hay C/C++.

Sau khi nền tảng Wiring hoàn thành, các nhà nghiên cứu đã làm việc với nhau để giúp nó nhẹ hơn, rẻ hơn, và khả dụng đối với cộng đồng mã nguồn mở. Một trong số các nhà nghiên cứu là David Cuarlierles, đã phổ biến ý tưởng này.

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 thiết bị có khả năng tương tác với môi trường thông qua các cảm biến và các cơ cấu chấp hành.

Thông tin thiết kế phần cứng được cung cấp công khai để những ai muốn tự làm một mạch Arduino bằng tay có thể tự mình thực hiện được (mã nguồn mở). Người ta ước tính khoảng giữa năm 2011 có trên 300 ngàn mạch Arduino chính thức đã được sản xuất thương mại, và vào năm 2013 có khoảng 700 ngàn mạch chính thức đã được đưa tới tay người dùng.

Phần cứng Arduino gốc được sản xuất bởi công ty Italy tên là Smart Projects. Một vài board dẫn xuất từ Arduino cũng được thiết kế bởi công ty của Mỹ tên là

SparkFun Electronics. Nhiều phiên bản của Arduino cũng đã được sản xuất phù hợp cho nhiều mục đích sử dụng:



Arduino Uno



Arduino Leonardo



Arduino Due



Arduino Robot



Arduino Esplora



Arduino Mega ADK



LilyPad Arduino USB



LilyPad Arduino  
Simple



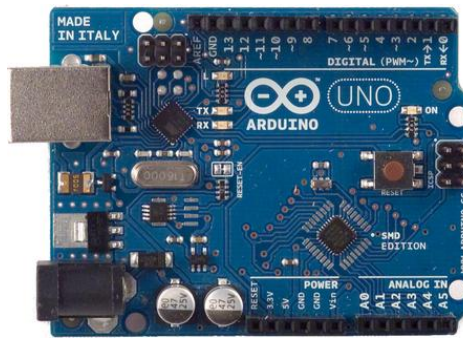
LilyPad Arduino  
SimpleSnap

Hình 2.1: Những phiên bản của Arduino

### 2.1.2 Uno

"Uno" có nghĩa là một bằng tiếng Ý và được đặt tên để đánh dấu việc phát hành sắp tới của Arduino 1.0. Uno và phiên bản 1.0 sẽ là phiên bản tài liệu tham khảo của Arduino. Uno là mới nhất trong các loại board Arduino, và các mô hình tham chiếu cho các nền tảng Arduino.

Arduino Uno là một “hội đồng quản trị” dựa trên ATmega328. Nó có 14 số chân đầu vào / đầu ra, 6 đầu vào analog, 16 MHz cộng hưởng gốm, kết nối USB, một jack cắm điện, một tiêu đề ICSP, và một nút reset. Nó chứa tất cả mọi thứ cần thiết để hỗ trợ các vi điều khiển; chỉ cần kết nối nó với máy tính bằng cáp USB hoặc cấp điện cho nó để bắt đầu.



Hình 2.2: Arduino Uno

Uno khác với tất cả các phiên bản trước ở chỗ nó không sử dụng các FTDI chip điều khiển USB-to-serial. Thay vào đó, nó có tính năng Atmega 16U2 lập trình như là một công cụ chuyển đổi USB-to-serial.

Phiên bản 2 (R2) của Uno sử dụng Atmega8U2 có một điện trở kéo dòng 8U2 HWB xuống đất, làm cho nó dễ dàng hơn để đưa vào chế độ DFU.

Phiên bản 3 (R3) của Uno có các tính năng mới sau đây:

- Thêm SDA và SCL gần với pin Aref và hai chân mới được đặt gần với pin RESET, các IOREF cho phép thích ứng với điện áp cung cấp.
- Đặt lại mạch khỏe mạnh hơn.
- Atmega 16U2 thay thế 8U2.



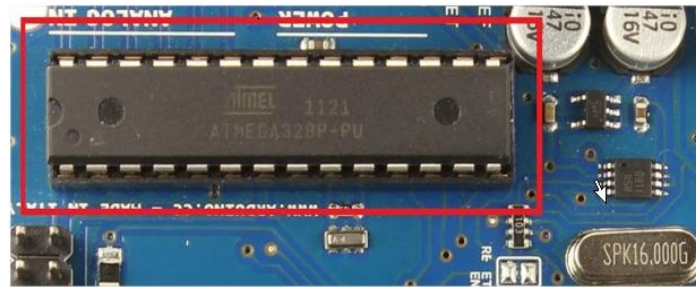
## 2.1.3 Cấu trúc, thông số

Bảng 2.1: Một vài thông số của 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 khuyến 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 (5 V)	500 mA
Dòng ra tối đa (3.3 V)	50 mA
Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

a. Vi điều khiển & bộ nhớ

Arduino UNO có thể sử dụng 3 vi điều khiển họ 8bit AVR là ATmega8, ATmega168, ATmega328. Bộ não này có thể xử lý những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lý tín hiệu cho xe điều khiển từ xa, làm một trạm đo nhiệt độ - độ ẩm và hiển thị lên màn hình LCD,...



Hình 2.3: Vi điều khiển Atmega328 tiêu chuẩn

**32KB bộ nhớ Flash:** những đoạn lệnh lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader nhưng đừng lo, hiếm khi nào cần quá 20KB bộ nhớ này.

**2KB cho SRAM (Static Random Access Memory):** giá trị các biến khai báo khi lập trình sẽ lưu ở đây. Khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Tuy vậy, thực sự thì cũng hiếm khi nào bộ nhớ RAM lại trở thành thứ phải bận tâm. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.

**1Kb cho EEPROM (Electrically Erasable Programmable Read Only Memory):** đây giống như một chiếc ổ cứng mini – nơi có thể đọc và ghi dữ liệu vào mà không phải lo bị mất khi cúp điện giống như dữ liệu trên SRAM.

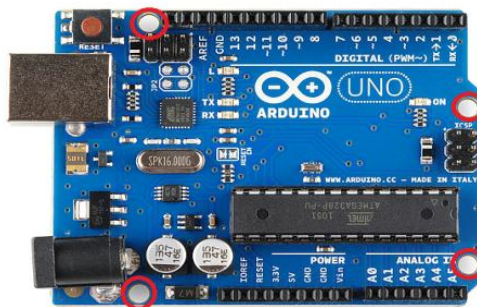
## b. Cấu tạo

*Hình 2.4: Arduino đời đầu*

Một board Arduino đời đầu gồm một cổng giao tiếp RS-232 (góc phía trên-bên trái) và một chip Atmel ATmega8 (màu đen, nằm góc phải-phía dưới); 14 chân I/O số nằm ở phía trên và 6 chân analog đầu vào ở phía đáy.

Board Arduino sẽ đưa ra hầu hết các chân I/O của vi điều khiển để sử dụng cho những mạch ngoài. Diecimila, Duemilanove, và bây giờ là Uno đưa ra 14 chân I/O kỹ thuật số, 6 trong số đó có thể tạo xung PWM (điều chế độ rộng xung) và 6 chân input analog, có thể được sử dụng như là 6 chân I/O số. Những chân này được thiết kế nằm phía trên mặt board, thông qua các header cái 0.10-inch (2.5 mm). Các board Arduino Nano, và Arduino-compatible Bare Bones Board và Boarduino có thể cung cấp các chân header đực ở mặt trên của board dùng để cắm vào các breadboard.

Chiều dài tối đa và chiều rộng của Uno PCB là 2,7 và 2,1 inch tương ứng, với kết nối USB và jack điện mở rộng vượt ra ngoài không gian cũ. Bốn lỗ vít cho phép được gắn vào một bề mặt khác:

*Hình 2.5: Các lỗ vít giúp cố định vị trí Arduino*

c. Vị trí & chức năng các chân

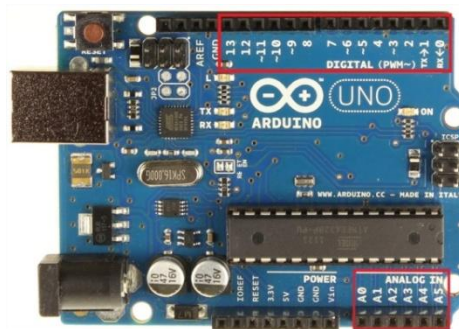
Nếu không có sẵn nguồn từ cổng USB, có thể cấp nguồn cho Arduino UNO từ một bộ chuyển đổi AC→DC hoặc pin. Các bộ chuyển đổi có thể được kết nối bằng một plug-2.1mm trung tâm tích cực vào jack cắm điện.

Trường hợp cấp nguồn quá ngưỡng trên sẽ làm hỏng Arduino UNO.

**Các chân năng lượng:**

- **GND (Ground):** cực âm của nguồn điện cấp cho Arduino UNO. Khi dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.
- **5V:** cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.
- **3.3V:** cấp điện áp 3.3V đầu ra. Dòng tối đa ở chân này là 50mA.
- **Vin (Voltage Input):** để cấp nguồn ngoài cho Arduino UNO, ta nối cực dương của nguồn với chân này và cực âm với chân GND.
- **IOREF:** điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.
- **RESET:** việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

**Các chân Input/Output:**



Hình 2.6: Các ngõ vào/ngõ ra của Arduino

Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng tối đa trên mỗi chân là 40mA.

Một số chân digital có các chức năng đặc biệt như sau:

- **2 chân Serial:** 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối Serial không dây. Nếu không cần giao tiếp Serial, không nên sử dụng 2 chân này nếu không cần thiết
- **Chân PWM (~): 3, 5, 6, 9, 10, và 11:** cho phép bạn xuất ra xung PWM với độ phân giải 8bit (giá trị từ  $0 \rightarrow 2^8-1$  tương ứng với  $0V \rightarrow 5V$ ) bằng hàm `analogWrite()`. Nói một cách đơn giản, có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.
- **Chân giao tiếp SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.
- **LED 13:** trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, ta sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.

Arduino UNO có 6 chân analog (A0  $\rightarrow$  A5) cung cấp độ phân giải tín hiệu 10bit ( $0 \rightarrow 2^{10}-1$ ) để đọc giá trị điện áp trong khoảng  $0V \rightarrow 5V$ . Với chân **AREF** trên board, ta có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu cấp điện áp 2.5V vào chân này thì ta có thể dùng các chân analog để đo điện áp trong khoảng từ  $0V \rightarrow 2.5V$  với độ phân giải vẫn là 10bit.

Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

## d. Extension shield

*Hình 2.7: Các shield xếp chồng lên Arduino*

Arduino cũng sử dụng chip AVR của Atmel làm nền tảng, thế nên hầu hết cái gì PIC/AVR làm được thì Arduino làm được. Nếu muốn điều khiển động cơ, sẽ có các mạch công suất tương thích hoàn toàn với Arduino. Nếu muốn điều khiển qua mạng Internet, cũng có một mạch Ethernet/Wifi tương thích hoàn toàn với Arduino. Và còn rất nhiều thứ khác nữa.

Những mạch được đề cập như trên được gọi là các extension shield (mạch mở rộng). Các shield này giúp tăng tính linh hoạt của Arduino.



Arduino GSM Shield



Arduino WiFi Shield



Arduino Wireless Proto Shield



Arduino Ethernet Shield



Arduino Wireless SD Shield



Arduino Proto Shield

*Hình 2.8: Một số shield thông dụng*



## 2.2 IC 74HC595

IC 74HC595 còn được gọi là IC dịch chốt với mối quan hệ "vào nối tiếp và ra song song 8 bit".



Hình 2.9: IC dịch chốt

Để thực sự hiểu rõ IC này chúng ta sẽ phải trả lời một số câu hỏi cơ bản nhất của vấn đề. Trước tiên, **thế nào là dịch và chốt?** Nói một chút lý thuyết kỹ thuật số về 2 thuật ngữ dịch và chốt để ta nắm chắc vấn đề hơn.

### 2.2.1 Thế nào là dịch và chốt?

Dịch là gì?

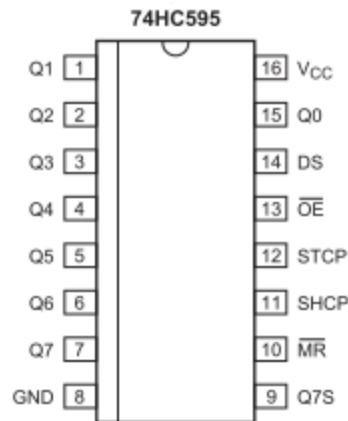
- Thuật ngữ "Dịch" được dùng cho IC này thực chất là cách nói của dân kỹ thuật về đặc tính ngõ vào nối tiếp của IC. Vậy "đặc tính ngõ vào nối tiếp của IC" là như thế nào? Có thể nói 1 cách đơn giản là ta có thể đưa lần lượt nhiều giá trị logic vào 1 ngõ vào của IC đó để nó lưu vào bộ nhớ chờ đến khi có lệnh xử lý. Có thể hiểu ngõ vào nối tiếp tức là các giá trị được nạp vào IC một cách từ từ ở từng thời điểm khác nhau trên cùng 1 ngõ vào, còn nếu nói vào song song thì có nghĩa là sẽ có nhiều ngõ vào và các giá trị sẽ được nạp vào IC cùng 1 lúc. Thuật ngữ này đối với ngõ ra cũng tương tự.

Chốt là gì?

- Chốt là khi thỏa mãn 1 điều kiện nào đó nó sẽ cho phép IC giữ nguyên giá trị ngõ ra, không cho nó thay đổi mặc dù tín hiệu ngõ vào có thay đổi thế nào.

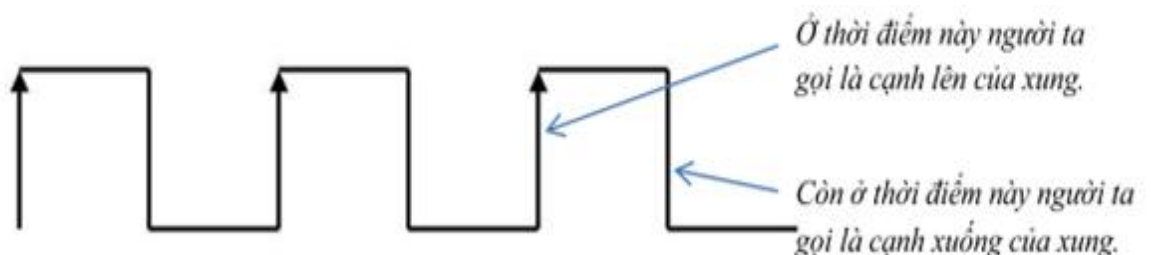
### 2.2.2 Sơ đồ & chức năng các chân 74HC595

Để hiểu rõ về IC 74HC595 ta cần có trong tay datasheet của nó



Hình 2.10: Sơ đồ chân của IC

- Các chân từ 1 tới 7 và chân số 15 là ngõ ra của IC. Chân DS (14) là ngõ vào của IC (đây là IC vào nối tiếp nên ta chỉ cần 1 ngõ vào là đủ).
- Chân 16 - VCC là chân cấp nguồn dương (từ 2V đến 6V)
- Chân số 8 GND là chân cấp Ground – cực (-) của nguồn
- **Chân SHCP:** là chân đưa xung clock (xung nhịp) vào IC và khi có cạnh lên của xung thì IC đưa tín hiệu ở ngõ vào vào bộ nhớ của IC để chờ xử lý
- (Chú thích: Thế nào là xung clock? Thực ra xung clock là 1 chuỗi tín hiệu logic 0 và 1 có thể là 1 xen kẽ với 0 cũng có thể là 0,1 ngẫu nhiên, nhưng nói chung nó là 1 chuỗi tín hiệu logic.

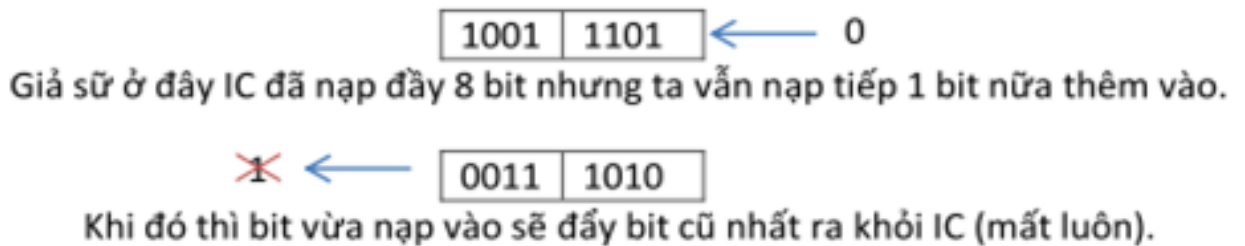


Hình 2.11: VD về xung clock

Còn cạnh lên và cạnh xuống của xung thì ta có thể thấy trên hình, cạnh lên là khi xung clock chuyển trạng thái từ 0 lên 1, còn cạnh xuống là thời điểm khi chuyển từ 1 xuống 0).



Vậy khi có cạnh lên của xung tại chân **SHCP** thì 1 tín hiệu logic từ ngõ vào của IC sẽ được lưu trữ vào trong IC để chờ tín hiệu cho phép xử lý. Bộ nhớ tối đa của IC là 8 bit, nếu vượt quá ngưỡng này thì giá trị mới sẽ được đưa vào IC và đồng thời giá trị cũ nhất của IC sẽ được xoá đi.

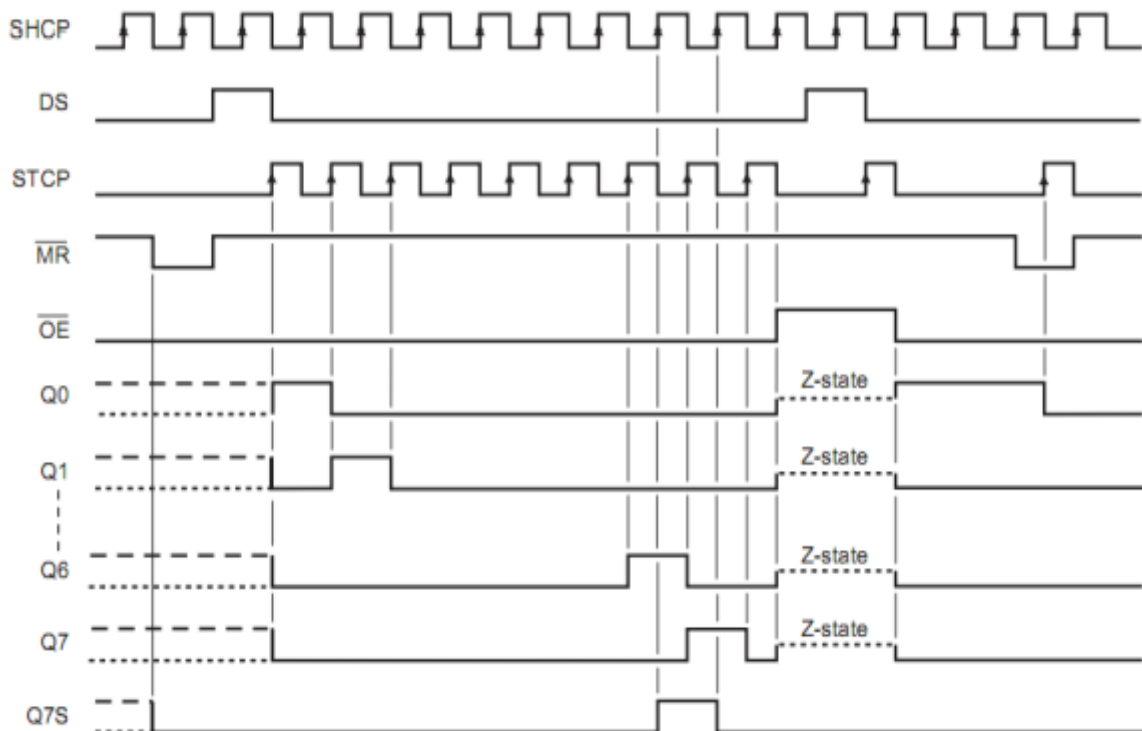


Hình 2.12: Cách nạp dữ liệu của bộ nhớ IC ghi dịch

- **Chân STCP:** là chân đưa xung clock vào IC để khi có cạnh lên của xung thì IC đưa toàn bộ 8bit data đã được lưu (đã nói ở chân SHCP) ra ngõ ra của IC.
- **Chân MR:** là chân reset IC (tức là trả IC về trạng thái ban đầu – giống như khi ta ghost máy tính vậy – khi chân này tích cực thì toàn bộ bộ nhớ của IC sẽ bị xoá tất cả bằng 0, tuy nhiên lưu ý là lúc này tín hiệu ở ngõ ra không bị xoá mà vẫn giữ nguyên giá trị trước đó) và chân này tích cực mức thấp (LOW active) có nghĩa là muốn reset IC thì phải đưa 0V vào chân này.
- **Chân EO:** là chân Output Enable chân khi được tích cực thì mới cho phép ta điều chỉnh được giá trị ngõ ra. Khi tên chân IC mà có dấu gạch trên đầu tức là nó tích cực thấp (LOW active) tức là muốn tích cực chân này thì ta phải đưa 0v (GND) vào chân này. Còn nếu khi chân này không được tích cực (tức là đưa mức logic 1 vào chân này thì ngõ ra bị đưa lên trạng thái trở kháng cao).
- **Chân số 9 chân Q7S:** Chữ S ở đây là viết tắt cho từ Serial (nối tiếp) chân này thường được dùng khi ta nối tiếp các IC 74HC595 với nhau (chân Q7S của con trước nối vào chân DS của con sau) chân này sẽ có giá trị của bit trọng số cao của bộ nhớ IC (Bit mới được đưa vào sẽ nằm ở vị trí LSB – trong số thấp) nếu mắc nối tiếp các IC 74HC595 lại với nhau theo cách như vậy thì khi bit

MSB bị đẩy ra khỏi bộ nhớ của IC sẽ không mất đi mà trước đó nó đã được sao chép qua IC phía sau.

### 2.2.3 Giải đồ thời gian về cách hoạt động của IC



Hình 2.13: Giải đồ hoạt động của IC

## **2.3 E18-D80NK**

### **2.3.1 Giới thiệu**

Lâu nay, chúng ta đã quen với việc sử dụng cảm biến siêu âm để phát hiện vật cản, tuy nhiên điểm yếu của nó là dễ bị nhiễu. Để khắc phục điểm yếu trên, đồ án đã sử dụng một phương pháp phát hiện vật cản khác. Đó chính là sử dụng hồng ngoại, mà cụ thể hơn là sử dụng cảm biến E18-D80NK thường ứng dụng cho các đặc tính Robot tránh vật cản, trên các dây chuyền phát hiện sản phẩm, các bộ reminder đa chức năng..v.v..

Cảm biến vật cản hồng ngoại E18-D80NK dùng ánh sáng hồng ngoại để xác định vật cản cho độ phản hồi nhanh và rất ít nhiễu do sử dụng mắt nhận và phát tia hồng ngoại theo tần số riêng biệt. Cảm biến có thể chỉnh khoảng cách hoạt động thông qua biến trở ở phần cuối thân cảm biến.



*Hình 2.14: Cảm biến E18-D80NK*

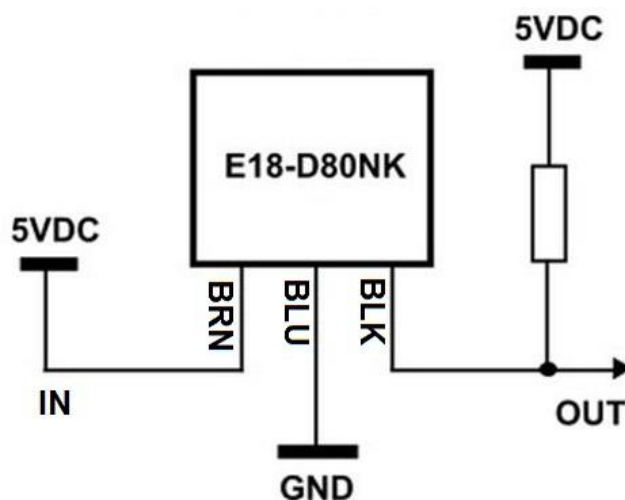
### 2.3.2 Thông số kỹ thuật

- Điện áp hoạt động: 5VDC
- Khoảng cách hoạt động tối đa: ~80cm
- Dòng kích ngõ ra: 300mA
- Góc điểm:  $\sim 15^\circ$
- Thời gian hồi đáp:  $\sim 2\text{ms}$
- Nhiệt độ môi trường làm việc:  $-25^\circ\text{C} \sim 50^\circ\text{C}$
- Ngõ ra dạng NPN cực thu hở giúp tùy biến được điện áp ngõ ra, trở treo lên áp bao nhiêu sẽ tạo thành điện áp ngõ ra bấy nhiêu.
- Chất liệu vỏ cảm biến: Nhựa
- Hiển thị ngõ ra bằng Led
- Kích thước: 1.8cm (D) x 7.0cm (L)

### 2.3.3 Sơ đồ dây

E18-D80NK có cách nối dây tương đối đơn giản:

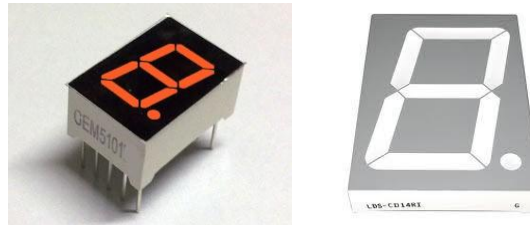
- **Màu nâu:** VCC, nguồn dương 5VDC
- **Màu xanh dương:** GND, nguồn âm 0VDC
- **Màu đen:** tín hiệu ngõ ra cực thu hở NPN, cần trở treo để tạo mức cao



Hình 2.15: Sơ đồ chân của E18-D80NK

## 2.4 LED 7 đoạn

### 2.4.1 Giới thiệu

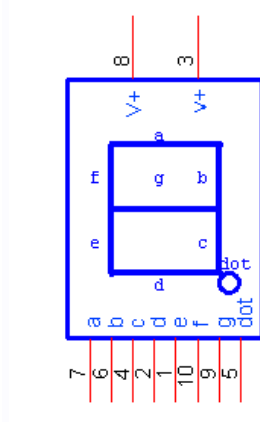


Hình 2.16: LED 7 đoạn

LED 7 đoạn hay LED 7 đoạn (Seven Segment display) dùng nhiều trong các mạch hiển thị thông báo, hiển thị số, kí tự đơn giản... LED 7 đoạn được cấu tạo từ các LED đơn sắp xếp theo các đoạn nét để có thể biểu diễn các chữ số hoặc các kí tự đơn. Tùy vào kích thước của số và kí tự mà mỗi đoạn được cấu tạo bởi một hay nhiều LED đơn. Qua đó người ta chỉ cần các bit tương ứng với các LED đơn để điều khiển, hiển thị số từ 0 đến 9 và các kí tự.

### 2.4.2 Cấu tạo & Nguyên lý hoạt động

Trong LED 7 đoạn bao gồm ít nhất là 7 con LED mắc lại với nhau, vì vậy mà có tên là LED 7 đoạn là vậy, 7 LED đơn được mắc sao cho nó có thể hiển thị được các số từ 0 - 9, và 1 vài chữ cái thông dụng, để phân cách thì người ta còn dùng thêm 1 led để hiển thị dấu chấm (dot).

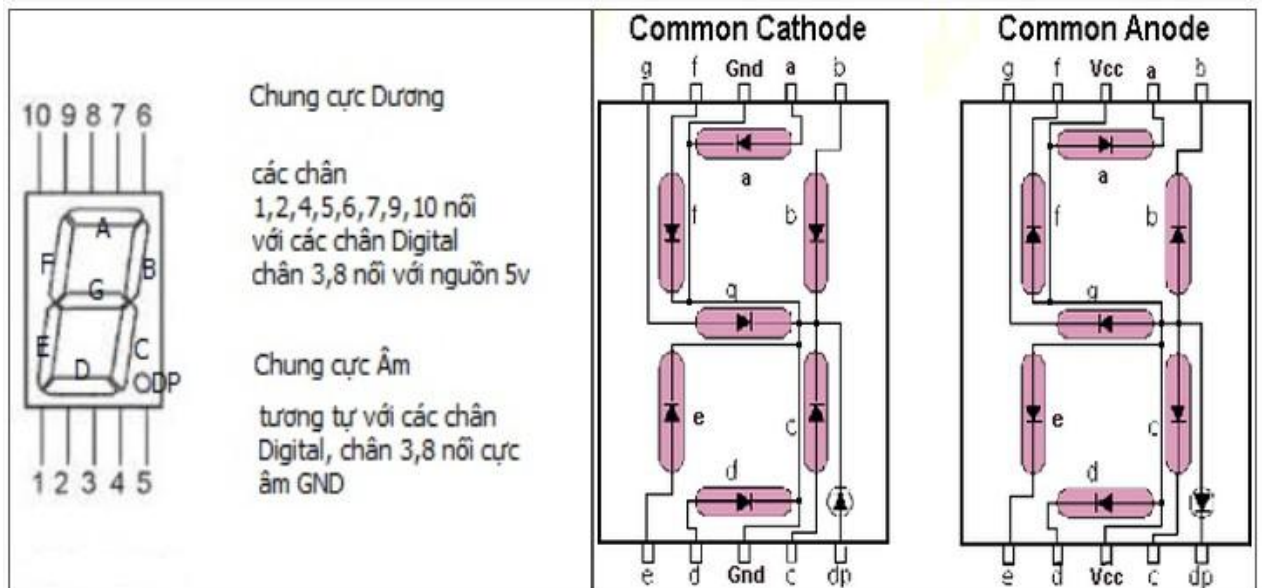


Hình 2.17: Cấu tạo LED 7 đoạn

Như vậy nếu như muốn hiển thị ký tự nào thì ta chỉ cần cấp nguồn vào chân đó là led sẽ sáng như mong muốn.

LED 7 đoạn dù có nhiều biến thể nhưng cũng chỉ vẫn có 2 loại:

- + Chân Anode chung (Chân + các led mắc chung lại với nhau .)
- + Chân Catode chung (Chân - các led được mắc chung với nhau .)



Hình 2.18: Cấu tạo 2 loại LED 7 đoạn

Điện áp giữa Vcc và Gnd phải lớn hơn 1.3 V mới cung cấp đủ led sáng, tuy nhiên không được cao quá 3V.

#### 2.4.3 Trở hạn dòng:

Trong các mạch thì thường dùng nguồn 5V nên để tránh việc đốt cháy led thì cách đơn giản nhất là mắc thêm trở hạn dòng.

Thông số làm việc của LED:

- Điện áp = 2V.
- Dòng = 20mA.

⇒ Vậy nếu dùng nguồn 5V , thì áp rơi trên trở = 5 - 2 = 3 V.

$$R = U / I = 3 / (20 \times 10^{-3}) = 150 \Omega .$$

## 2.5 Motor DC

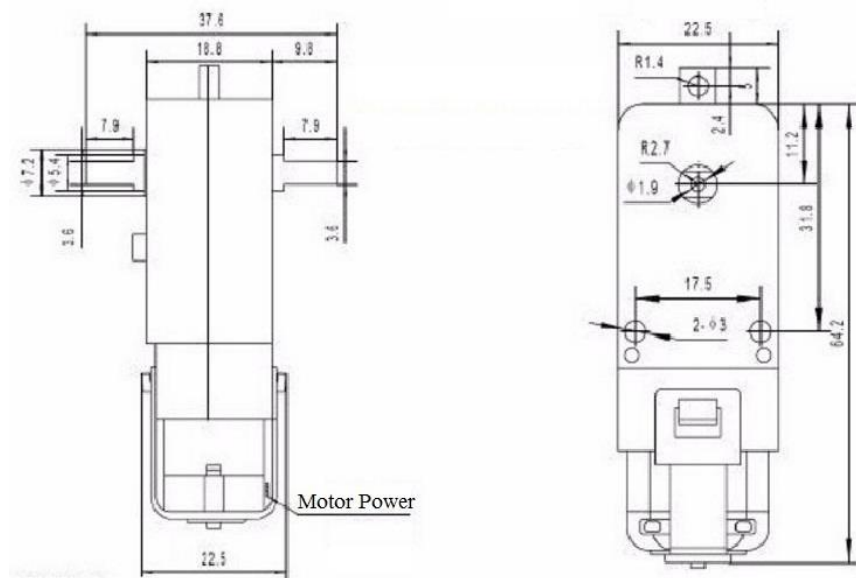
### 2.5.1 Giới thiệu

Động cơ điện một chiều là máy điện chuyển đổi năng lượng điện một chiều sang năng lượng cơ. (Máy điện chuyển đổi từ năng lượng cơ sang năng lượng điện là máy phát điện).

Động cơ DC giảm tốc V1 là loại được lựa chọn và sử dụng nhiều nhất hiện nay cho các mô hình, thiết kế Robot đơn giản... Động cơ DC giảm tốc V1 có chất lượng tương đối cùng với khả năng dễ lắp ráp đem lại sự tiện dụng, thích hợp cho mô hình đồ án này.

### 2.5.2 Thông số kỹ thuật

- Điện áp hoạt động:  $3 \rightarrow 9\text{VDC}$
- Dòng điện tiêu thụ:  $110 \rightarrow 140\text{mA}$
- Tỷ số truyền: 1:120
- Số vòng/phút:
  - \* 50 vòng/phút – 3VDC
  - \* 83 vòng/phút – 5VDC
- Momen:  $\sim 1\text{ Kg.Cm}$

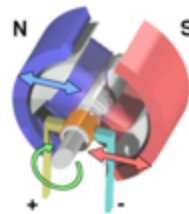


Hình 2.19: Kích thước động cơ

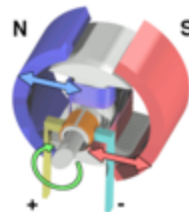
### 2.5.3 Cấu tạo & Hoạt động

Gồm có 3 phần chính stator (phần cảm), rotor (phần ứng), và phần chỉnh lưu (chổi than và cổ góp).

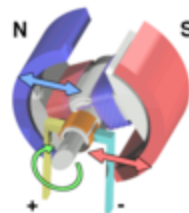
- Stator của động cơ điện 1 chiều thường là 1 hay nhiều cặp nam châm vĩnh cửu, hay nam châm điện.
- Rotor có các cuộn dây quấn và được nối với nguồn điện một chiều.
- Bộ phận chỉnh lưu, nó có nhiệm vụ là đổi chiều dòng điện trong khi chuyển động quay của rotor là liên tục.



Pha 1: Từ trường của rotor cùng cực với stator, sẽ đẩy nhau tạo ra chuyển động quay của rotor.



Pha 2: Rotor tiếp tục quay



Pha 3: Bộ phận chỉnh điện sẽ đổi cực sao cho từ trường giữa stator và rotor cùng dấu, trở lại pha 1

*Hình 2.20: Nguyên lý hoạt động phần cảm và phần ứng.*



## 2.6 Module LM2596

### 2.6.1 Giới thiệu

**Mạch Giảm Áp LM2596** là module giảm áp có khả năng điều chỉnh được dòng ra đến 3A. Tức là khi cấp nguồn 9v vào module, sau khi giảm áp ta có thể lắp được nguồn  $3A < 9v...$  như 5V hay 3.3V.

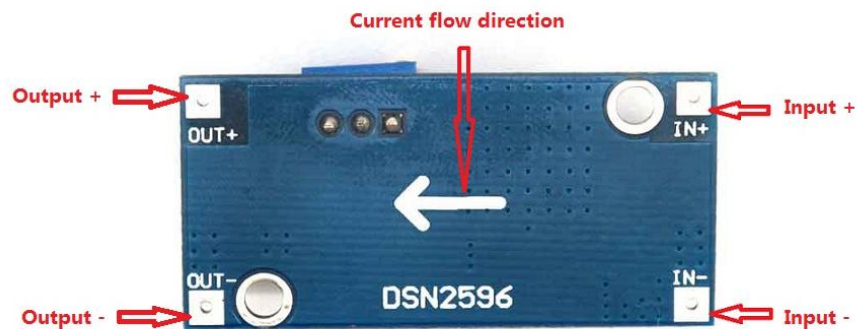


Hình 2.21: module giảm áp

### 2.6.2 Thông số kỹ thuật

- Module nguồn không sử dụng cách ly
- Nguồn đầu vào từ 4V - 35V.
- Nguồn đầu ra: 1V - 30V.
- Dòng ra Max: 3A
- Kích thước mạch: 53mm x 26mm
- Đầu vào: INPUT+, INPUT-
- Đầu ra: OUTPUT+, OUTPUT-

### 2.6.3 Nguyên lý hoạt động

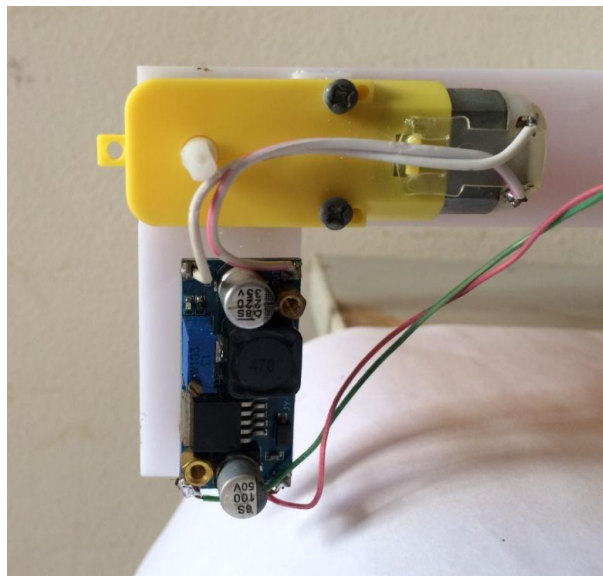


Hình 2.22: Hướng đi của dòng điện trong LM2596

Khi cấp nguồn thô vào chân INPUT+, INPUT- ta sẽ nhận được nguồn ra từ chân OUTPUT+, OUTPUT-

Điện áp đầu ra được tùy chỉnh bằng cách vặn biến trở trên module... Biến trở trên module này hỗ trợ vặn 14 vòng.

⇒ Dựa trên nguyên lý trên, Module LM2596 được áp dụng vào mô hình nhằm tùy chỉnh tốc độ bằng tải hoạt động thông qua động cơ DC:



Hình 2.23: Điều chỉnh tốc độ Motor thông qua module LM2596

## 2.7 Motor Servo

### 2.7.1 Giới thiệu

Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển với góc quay nằm trong khoảng giới hạn. Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô hình), có loại thì sở hữu một momen lực tương đối (vài chục Newton/m), hoặc có loại thì khỏe và không sắc chắc chắn...



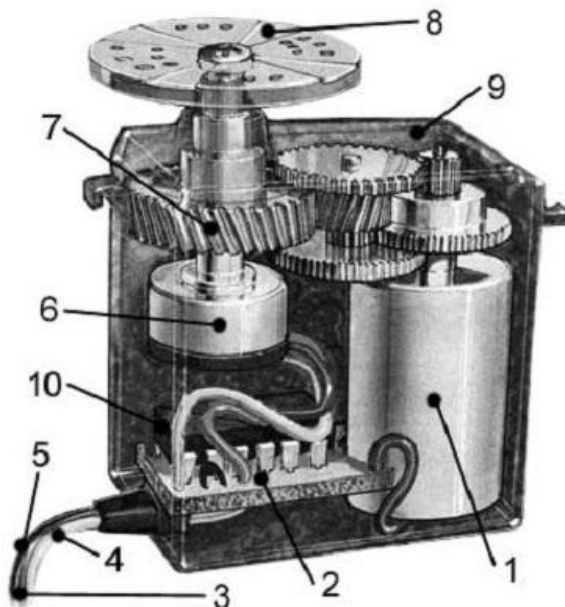
*Hình 2.24: Micro Servo 9g*

Các động cơ servo điều khiển bằng liên lạc vô tuyến được gọi là động cơ servo RC (radio-controlled). Trong thực tế, bản thân động cơ servo không phải được điều khiển bằng vô tuyến, nó chỉ nối với máy thu vô tuyến trên máy bay hay xe hơi. Động cơ servo nhận tín hiệu từ máy thu này.

### 2.7.2 Hoạt động & Cấu tạo

Động cơ DC và động cơ bước vốn là những hệ hồi tiếp vòng hở. Việc thiết lập một hệ thống điều khiển để xác định những gì ngăn cản chuyển động quay của động cơ hoặc làm động cơ không quay cũng không dễ dàng.

Mặt khác, động cơ servo được thiết kế cho những hệ thống hồi tiếp vòng kín. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về một mạch điều khiển. Nếu có bất kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác.



1. Motor
2. Electronics Board
3. Positive Power Wire (Red)
4. Signal Wire (Yellow or White)
5. Negative or Ground Wire (Black)
6. Potentiometer
7. Output Shaft/Gear
8. Servo Attachment Horn/Wheel/Arm
9. Servo Case
10. Integrated Control Chip

*Hình 2.25: Cấu tạo cơ bản của một động cơ (motor) servo*

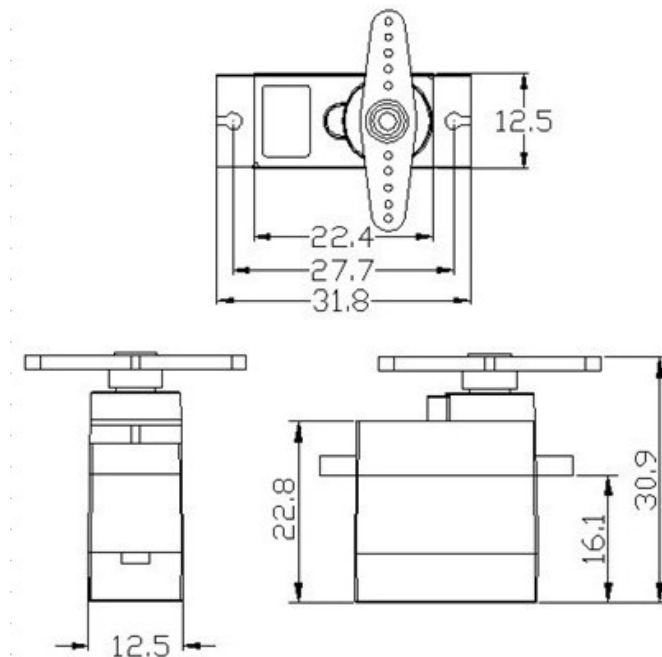
Để quay động cơ, tín hiệu số được gởi tới mạch điều khiển. Tín hiệu này khởi động động cơ, thông qua chuỗi bánh răng, nối với vận kế. Vị trí của trục vận kế cho biết vị trí trục ra của servo. Khi vận kế đạt được vị trí mong muốn, mạch điều khiển sẽ tắt động cơ. Mặc dù ta có thể chỉnh quay liên tục nhưng công dụng chính của động cơ servo là đạt được góc quay chính xác trong khoảng giới hạn.

## 2.7.3 Thông số kỹ thuật



Hình 2.26: Sơ đồ nối dây của Micro Servo

- Khối lượng: 9g
- Tín hiệu: Analog
- Mô men xoắn: 1.6kg/cm
- Tốc độ hoạt động: 0,12sec/60degree
- Điện áp hoạt động: 4.8VDC~5VDC
- Nhiệt độ hoạt động: 0 °C – 55 °C
- Delay: 10us

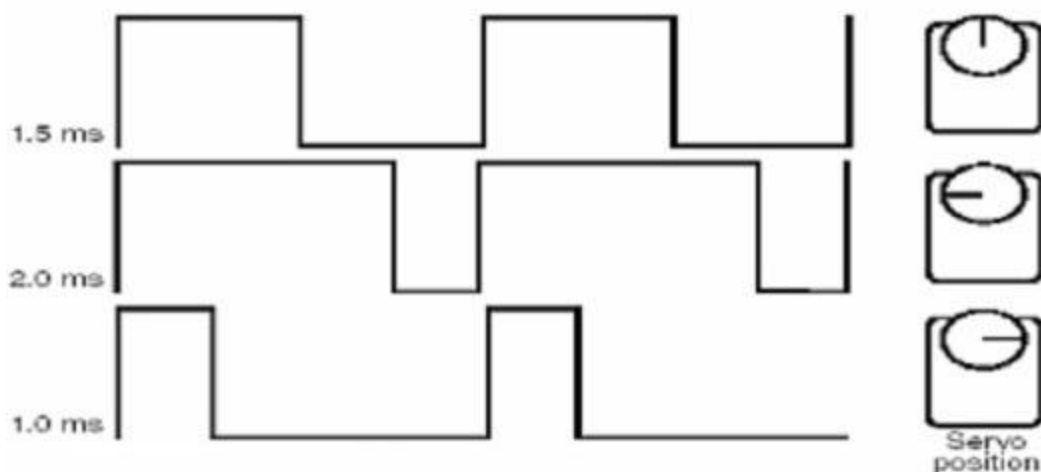


Hình 2.27: Kích thước Micro Servo

#### 2.7.4 Điều biến độ rộng xung

Trục của động cơ servo R/C được định vị nhờ vào kỹ thuật gọi là điều biến độ rộng xung (PWM). Trong hệ thống này, servo là đáp ứng của một dãy các xung số ổn định. Cụ thể hơn, mạch điều khiển là đáp ứng của một tín hiệu số có các xung biến đổi từ 1 – 2 ms. Các xung này được gửi đi 50 lần/giây. Chú ý rằng không phải số xung trong một giây điều khiển servo mà là chiều dài của các xung. Servo đòi hỏi khoảng 30 – 60 xung/giây. Nếu số này quá thấp, độ chính xác và công suất để duy trì servo sẽ giảm.

Với độ dài xung 1 ms, servo được điều khiển quay theo một chiều (giả sử là chiều kim đồng hồ):



Hình 2.28: Điều khiển trục ra của động cơ bằng cách điều chế độ rộng xung

Với độ dài xung 2 ms, servo quay theo chiều ngược lại. Kỹ thuật này được gọi là tỉ lệ số – chuyển động của servo tỉ lệ với tín hiệu số điều khiển.

### 2.7.5 Giới hạn quay

Các servo khác nhau ở góc quay được với cùng tín hiệu 1 – 2 ms (hoặc bất kỳ) được cung cấp. Các servo chuẩn được thiết kế để quay tới và lui từ 90 độ – 180 độ khi được cung cấp toàn bộ chiều dài xung. Nếu ta cố điều khiển servo vượt quá những giới hạn cơ học của nó, hiện tượng này kéo dài hơn vài giây sẽ làm bánh răng của động cơ bị phá hủy

### 2.7.6 Phân loại và các kích thước đặc biệt

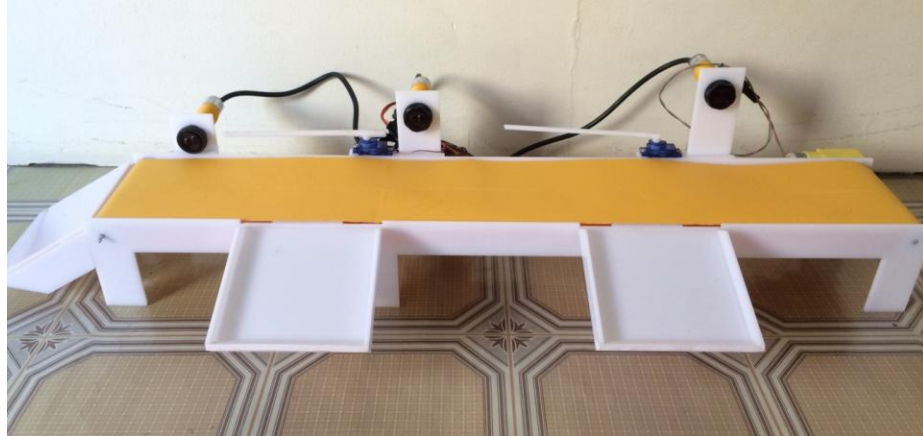
Ngoài servo kích thước chuẩn dùng trong robot và mô hình điều khiển vô tuyến còn có các loại servo R/C khác:

- Servo tỉ lệ  $\frac{1}{4}$  / tỉ lệ lớn (quarter-scale / large-scale servo): kích thước gấp khoảng 2 lần servo chuẩn, công suất lớn hơn rõ, được dùng trong các mô hình máy bay lớn, làm động cơ công suất cho robot.
- Servo nhỏ (mini-micro servo): kích thước nhỏ hơn khoảng 2 lần so với servo chuẩn, không mạnh bằng servo chuẩn, dùng ở những không gian hẹp trong mô hình máy bay hay xe hơi.
- Servo tời buồm (sail minch servo): mạnh nhất, dùng để điều khiển các dây thừng của buồm nhỏ và buồm chính trong mô hình thuyền buồm.
- Servo thu bộ phận hạ cánh(landing-gear retraction servo): dùng để thu bộ phận hạ cánh trong mô hình máy bay vừa và lớn. Thiết kế bộ phận hạ cánh thường đòi hỏi servo phải đảm bảo góc quay ít nhất là 170 độ. Các servo này thường nhỏ hơn kích thước chuẩn vì không gian giới hạn trong mô hình máy bay



## Chương 3: MÔ HÌNH ĐẾM & PHÂN LOẠI SẢN PHẨM

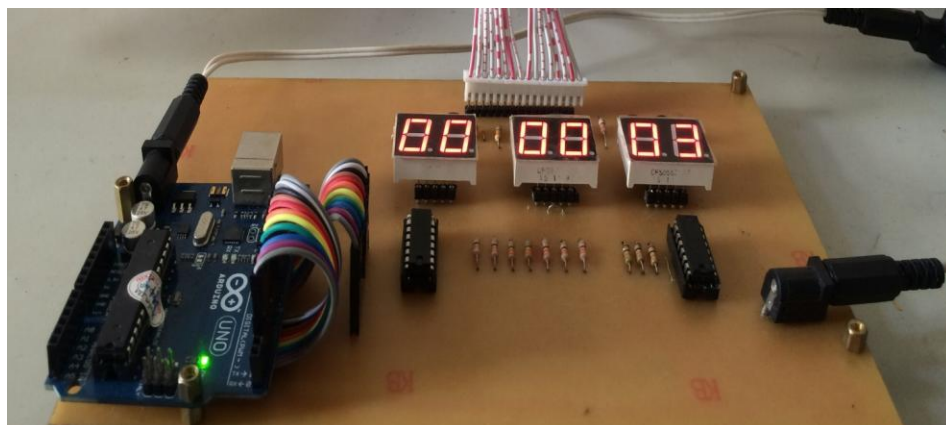
### 3.1 Giới thiệu



Hình 3.1: Mô hình hệ thống

Đồ án “Mô hình phân loại và đếm sản phẩm theo chiều cao sử dụng Arduino” bằng việc thiết kế mô hình phân loại sản phẩm và lập trình Arduino ý tưởng áp dụng Arduino trong công nghiệp sản xuất càng sáng tỏ, khả thi hơn.

Khi sản phẩm trên băng tải đi qua cảm biến (E18-D80NK) được thiết kế với động cơ Servo để phân loại sản phẩm kết hợp với LED 7 đoạn để hiển thị số lượng sản phẩm, giúp chúng ta một phần nào hiểu được dây chuyền phân loại sản phẩm trong thực tế.



Hình 3.2: Board mạch chính của hệ thống



### 3.2 Các phần mềm thiết kế

#### 3.2.1 Arduino

Arduino là môi trường phát triển tích hợp mã nguồn mở, cho phép người dùng dễ dàng viết code và tải nó lên board mạch, được viết bằng Java dựa trên ngôn ngữ lập trình và phần mềm mã nguồn mở khác.



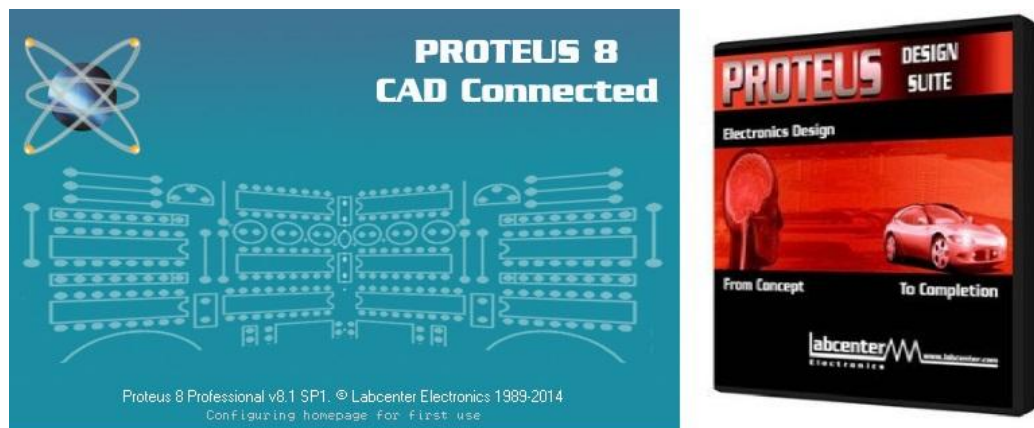
*Hình 3.3: phần mềm Arduino*

Kể từ tháng 3 năm 2015, Arduino IDE (Intergrated Devalopment Editor – môi trường phát triển thích hợp) đã được phổ biến tại rất nhiều nơi với giao diện trực quan.

Ngôn ngữ phổ quát cho Arduino là C và C++. Do đó phần mềm phù hợp với những người dùng quen thuộc các ngôn ngữ này.

Phần mềm gồm những mảng thư viện phong phú như: EEPROM, Firmata, GSM, Servo, TFT, Wifi,... Và các mảng thư viện ngày càng đa dạng nhờ sự đóng góp của cộng đồng Arduino trên toàn thế giới.

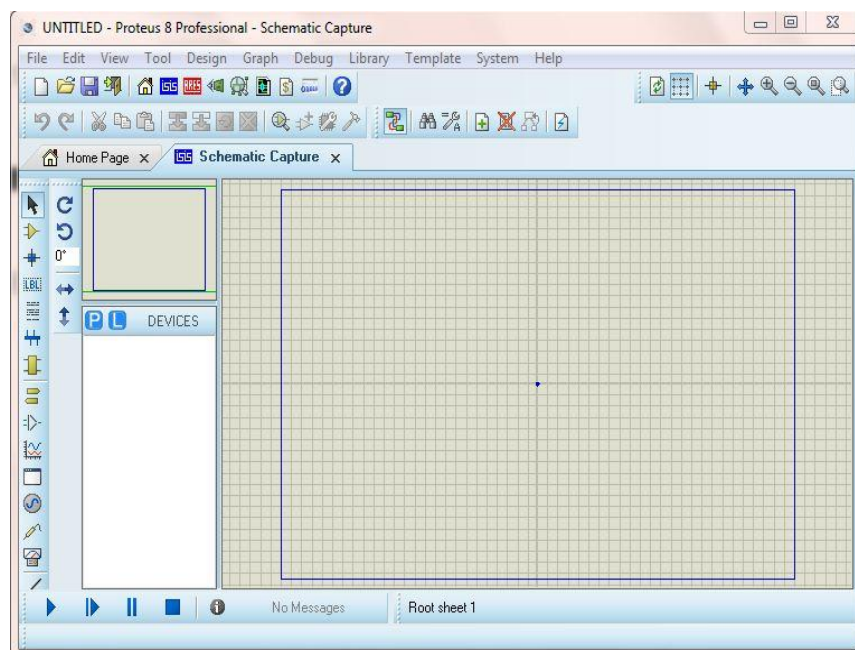
### 3.2.2 Proteus 8 Professional



*Hình 3.4: Proteus Labcenter Electronics*

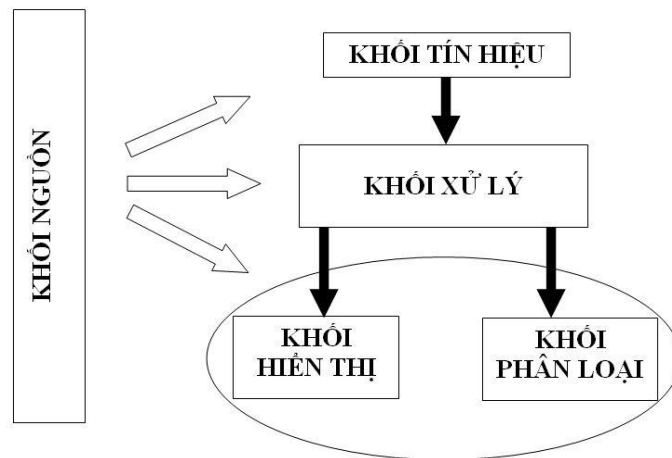
Proteus là phần mềm mô phỏng mạch điện tử của Labcenter Electronics, mô phỏng cho hầu hết các linh kiện điện tử thông dụng, đặc biệt hỗ trợ cho cả các MCU như PIC, 8051, AVR, ...

Phần mềm bao gồm 2 mảng chính là ISIS cho phép mô phỏng mạch điện tử và ARES dùng để vẽ mạch in.



*Hình 3.5: Giao diện chính của Proteus*

### 3.3 Sơ đồ khối



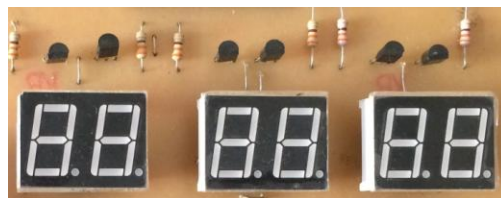
Hình 3.6: Sơ đồ khối mô hình đếm & phân loại sản phẩm

- Khối nguồn gồm các linh kiện tác động đến công suất, dòng điện. (adapter, module nguồn...) cung cấp năng lượng thích hợp cho mô hình hệ thống.



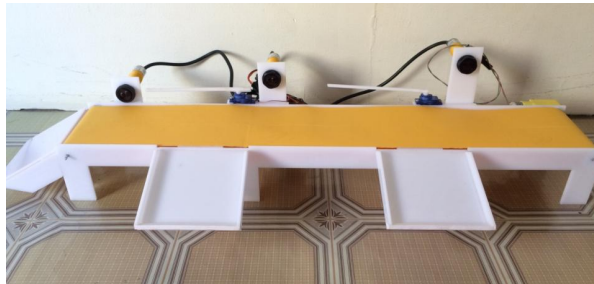
Hình 3.7: Adapter AC→DC cho hệ thống

- Khối hiển thị (Led 7 đoạn): hiển thị số lượng đếm được từ cảm biến



Hình 3.8: các cặp Led 7 đoạn

- Khối phân loại (Băng tải, Servo): phân các sản phẩm thành nhiều loại theo yêu cầu của mô hình đề tài.
- Khối tín hiệu là các cảm biến E18D80NK: phát hiện vật thể và truyền tín hiệu về khối xử lý để mã hóa dữ liệu.



Hình 3.9: Các Servo, cảm biến và băng tải

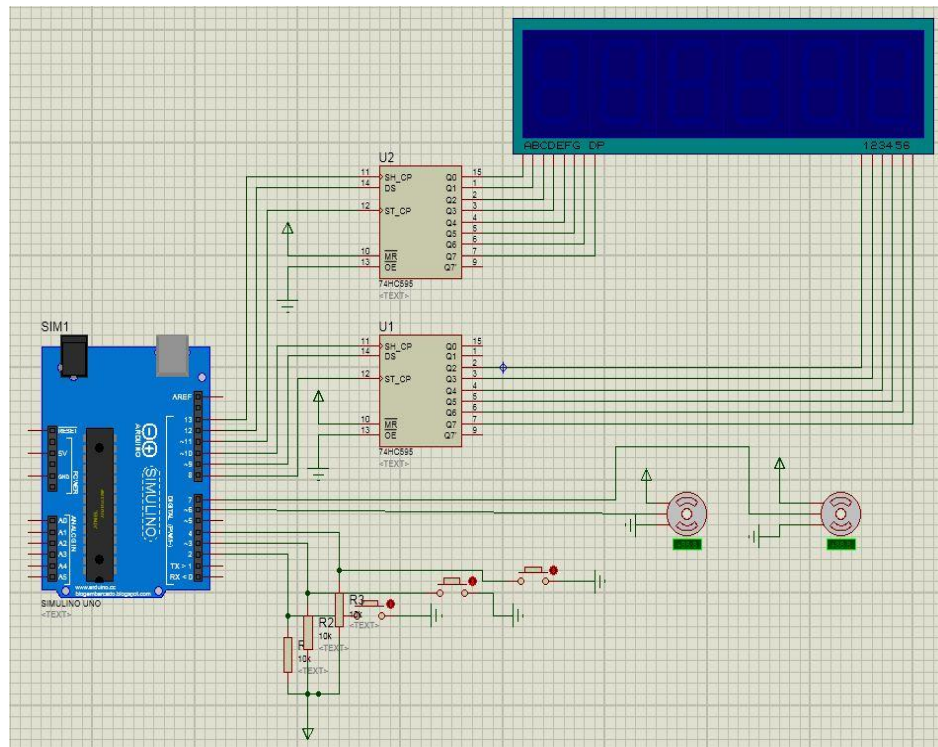
- Khối xử lý (Arduino Uno R3, IC 74HC595...): xử lý tín hiệu từ cảm biến và xuất dữ liệu được mã hóa đến các khối hiển thị, khối phân loại.



Hình 3.10: Board mạch cùng các linh kiện

### 3.4 Nguyên lý hoạt động

Khi được cấp nguồn, mạch sẽ hoạt động theo chương trình được thiết lập trên Arduino



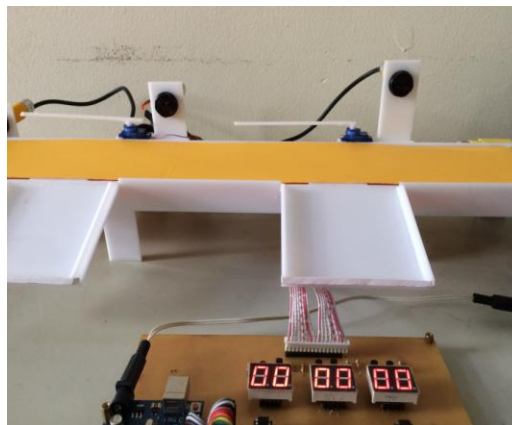
Hình 3.11: Sơ đồ nguyên lý hoạt động

Chu trình hoạt động của mô hình hệ thống này được bắt đầu từ sự thu nhận tín hiệu của các cảm biến.

Các tín hiệu được trung tâm xử lý và xuất các lệnh tương thích cho Servo cùng với chuỗi Led 7 đoạn để thực hiện theo đúng yêu cầu đề tài.

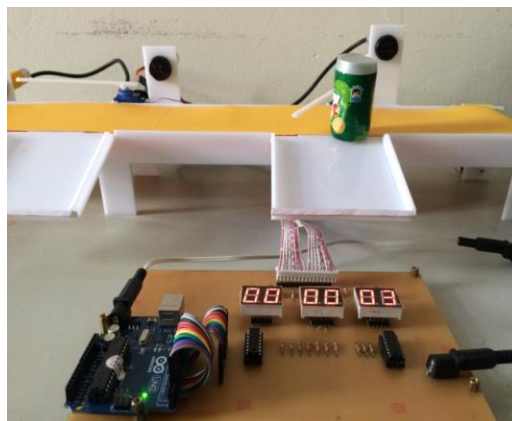


Ở trạng thái bình thường, cảm biến E18-D80NK cấp xung mức cao cho bộ xử lý trung tâm. Với xung mức cao, thông qua code đã được lập trình, Arduino Uno R3 duy trì hiện trạng. 2 IC “dịch chốt” 74HC595 tiếp tục chu trình quét các led 7 đoạn với khối dữ liệu đang có (1 IC sẽ phụ trách những con số mà led 7 đoạn hiển thị, IC còn lại sẽ phụ trách vị trí hiển thị những con số đó). Động cơ Servo giữ nguyên thanh gạt phân loại ở góc  $0^\circ$ .



*Hình 3.12: Mô hình khi chưa có tác động từ cảm biến.*

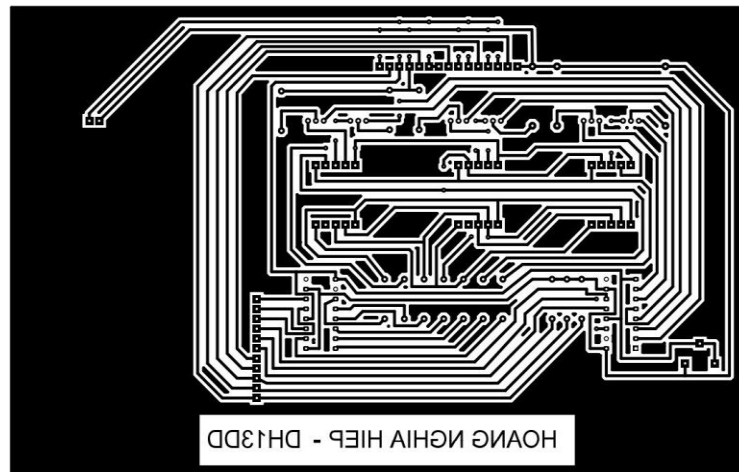
Khi được tác động bởi đối tượng, dòng điện qua trở treo thông qua cảm biến E18-D80NK tới GND. Lúc này, xung mức thấp được cấp cho Arduino Uno R3. Dữ liệu được xử lý và đưa vào 2 IC 74HC595 làm thay đổi sự hiển thị ở led 7 đoạn. Đồng thời, động cơ Servo quay thanh gạt phân loại một góc  $50^\circ$ , đối tượng được đưa ra khỏi băng tải.



*Hình 3.13: Mô hình hoạt động khi có tín hiệu từ cảm biến.*

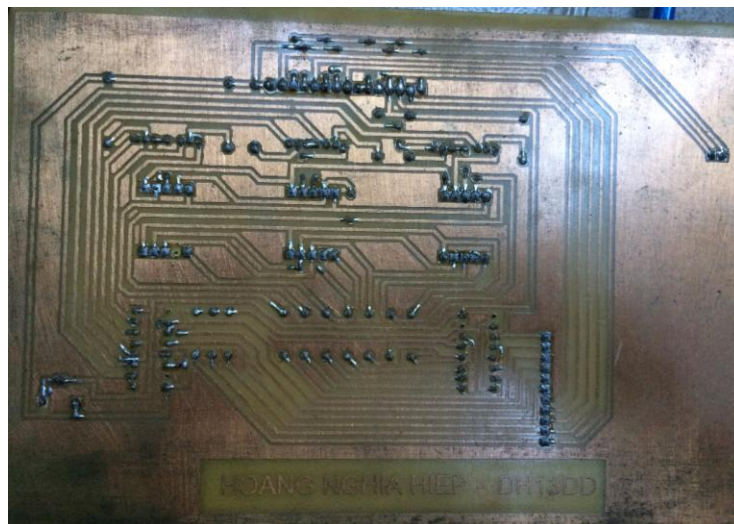
### 3.5 Mạch in

Mạch in có thể được thiết kế từ các phần mềm vẽ mạch chuyên dụng như OrCad, Proteus, EsysEDA, KitCad.... (Đối với đồ án này thì mạch in được vẽ bằng phần mềm Proteus để tiện lợi hơn khi tích hợp mô phỏng.)



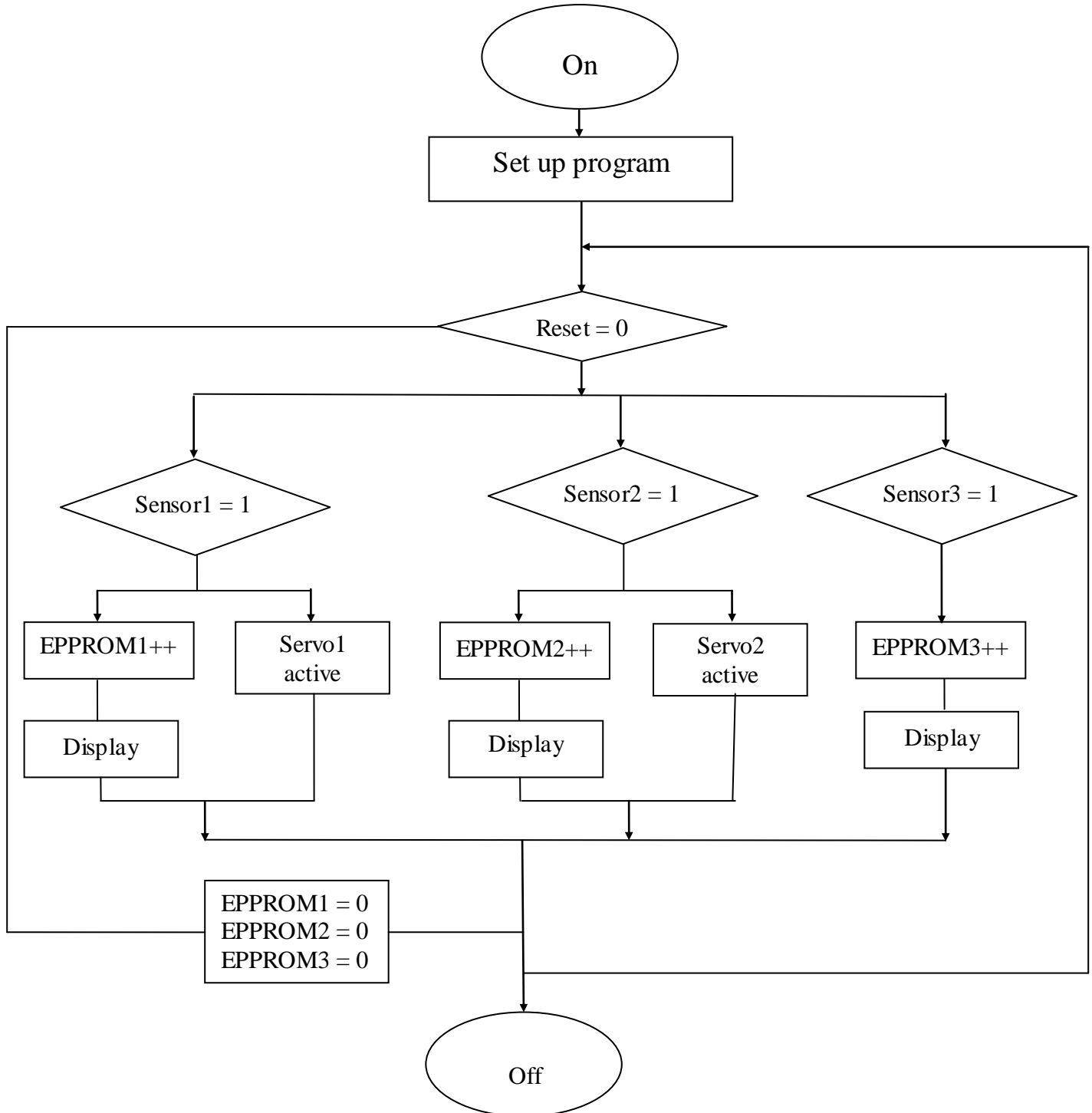
Hình 3.14: Bản thiết kế mạch in.

Trong quá trình làm mạch, do tay nghề còn non trẻ, thiếu kinh nghiệm nên mạch thiếu sự thẩm mỹ cũng như tính chuyên nghiệp.



Hình 3.15: Mạch thực tế

### 3.6 Lưu đồ thuật toán





### **3.7 Code – chương trình**

```
#include <Servo.h>
#include <EEPROM.h>

Servo servo1;
Servo servo2;

int latchPin1 = 8;
int clockPin1 = 10;
int dataPin1 = 9;
int latchPin2 = 11;
int clockPin2 = 13;
int dataPin2 = 12;

int resetPin = 2;
int Sensor1 = 3;
int Sensor2 = 4;
int Sensor3 = 5;

int SensorState1;
int SensorState2;
int SensorState3;

int LastSensorState1;
int LastSensorState2;
int LastSensorState3;

int b1,b2,b3,c1,c2,c3;
```

```
const int Seg[10] = {  
    0b11000000, //0  
    0b11111001, //1  
    0b10100100, //2  
    0b10110000, //3  
    0b10011001, //4  
    0b10010010, //5  
    0b10000010, //6  
    0b11111000, //7  
    0b10000000, //8  
    0b10010000, //9  
};
```

```
const int led [9] = {  
    0,  
    0b100000,  
    0b010000,  
    0b001000,  
    0b000100,  
    0b000010,  
    0b000001,  
};
```

```
void setup()
{
    pinMode (latchPin1, OUTPUT);
    pinMode (clockPin1, OUTPUT);
    pinMode (dataPin1, OUTPUT);
    pinMode (latchPin2, OUTPUT);
    pinMode (clockPin2, OUTPUT);
    pinMode (dataPin2, OUTPUT);

    pinMode(resetPin, INPUT);
    pinMode(Sensor1, INPUT);
    pinMode(Sensor2, INPUT);
    pinMode(Sensor3, INPUT);

    servo1.attach(7);
    servo2.attach(6);
}

void Delay(float delayTime, void (func)()){
    unsigned long endTime = millis() + delayTime;
    while(millis() < endTime)
    {
        func();
    }
}
```

```
void Display(){
    led7segPlus (c1,1);
    delay(1);
    led7segPlus (b1,2);
    delay(1);
    led7segPlus (c2,3);
    delay(1);
    led7segPlus (b2,4);
    delay(1);
    led7segPlus (c3,5);
    delay(1);
    led7segPlus (b3,6);
    delay(1);}
```

```
void reset()
{
    EEPROM.write(1,0);
    EEPROM.write(2,0);
    EEPROM.write(3,0);
}
```

```
void Type1()
{
    Display();
    Type3();
    SensorState1 = digitalRead(Sensor1);
    if ((SensorState1 != LastSensorState1)&(SensorState1 == 0))
    {
        EEPROM.write(1,EEPROM.read(1)+1);
        if (EEPROM.read(1) > 99) { EEPROM.write(1,0);}
    }
```

```
    c1 = EEPROM.read(1) % 10;
    b1 = EEPROM.read(1) / 10;

    servo1.write(50);
    Delay(2300,Type2);
    servo1.write(0);}
    LastSensorState1 = SensorState1;}

void Type2()
{ Display();
  Type3();
  SensorState2 = digitalRead(Sensor2);
  if ((SensorState2 != LastSensorState2)&(SensorState2 == 0))
  {
    EEPROM.write(2,EEPROM.read(2)+1);
    if (EEPROM.read(2) > 99) { EEPROM.write(2,0);}
    c2 = EEPROM.read(2) % 10;
    b2 = EEPROM.read(2) / 10;

    servo2.write(50);
    Delay(2300,Type1);
    servo2.write(0);
  }
  LastSensorState2 = SensorState2;
}

void Type3()
{
  SensorState3 = digitalRead(Sensor3);
```

```
if ((SensorState3 != LastSensorState3)&(SensorState3 == 0))
{
    EEPROM.write(3,EEPROM.read(3)+1);
    if (EEPROM.read(3) > 99) { EEPROM.write(3,0);}
    c3 = EEPROM.read(3) % 10;
    b3 = EEPROM.read(3) / 10;
}
LastSensorState3 = SensorState3;
}
void loop()
{ servo1.write(0);
  servo2.write(0);

  if (digitalRead(resetPin)!=0) { reset();}

  Type1();
  Type2();
  Type3();
  Display();
}
```

```
void led7segPlus (int number, int pin)
{
    digitalWrite(latchPin1, 0);
    shiftOut(dataPin1, clockPin1, MSBFIRST, led[pin]);
    digitalWrite(latchPin2, 0);
    shiftOut(dataPin2, clockPin2, MSBFIRST, Seg[number]);
    digitalWrite(latchPin2, 1);
    digitalWrite(latchPin1, 1);

    }
```

## **Chương 4: KẾT LUẬN**

Sau một thời gian nghiên cứu và tìm hiểu đề tài này; đến nay, “mô hình phân loại và đếm sản phẩm theo chiều cao sử dụng Arduino” đã được thiết kế, chế tạo thành công.

### **4.1 Ưu/Nhược điểm**

#### **Ưu điểm:**

- Mạch nhỏ gọn.
- Đáp ứng được yêu cầu của đề tài.
- Hiển thị rõ ràng.

#### **Nhược điểm:**

- Chưa có giao diện giám sát, mô phỏng trên máy tính.
- Độ ổn định chưa tối ưu.

### **4.2 Hướng phát triển**

- Áp dụng cho các dây chuyền sản xuất nhỏ.
- Thay đổi cảm biến để tạo ra dây chuyền phân loại dựa trên các tiêu chí khác nhau của sản phẩm.
- Khắc phục nhược điểm đề tài được hoàn thiện hơn.

Do thời gian và kiến thức còn hạn hẹp nên không thể tránh khỏi những thiếu sót trong quá trình thực hiện đề tài. Rất mong nhận được những góp ý, những đánh giá quý báu của quý thầy cô và các bạn.



## TÀI LIỆU THAM KHẢO

1. **Kỹ thuật điện tử.** (1999)  
Đỗ Xuân Thụ. – NXB giáo dục.
2. **Giáo trình cảm biến.** (2000)  
Phan Quốc Phô, Nguyễn Đức Chiến. – NXB Khoa học và kỹ thuật.
3. **Vi điều khiển cấu trúc lập trình và ứng dụng.** (2008)  
Kiều Xuân Thục, Vũ Thị Hương, Vũ Trung Kiên – NXB Giáo Dục.
4. **Website** <http://alldatasheet.com/>
5. **Website** <http://arduino.vn/>
6. **Website** <http://codientu.org/>
7. **Website** <http://webdien.com/>
8. **Website** <http://www.tailieu.vn/>
9. **Website** <http://wikipedia.com/>