Khánh Hòa - 2021

****

BÁO CÁO HỆ THỐNG LẬP TRÌNH NHÚNG

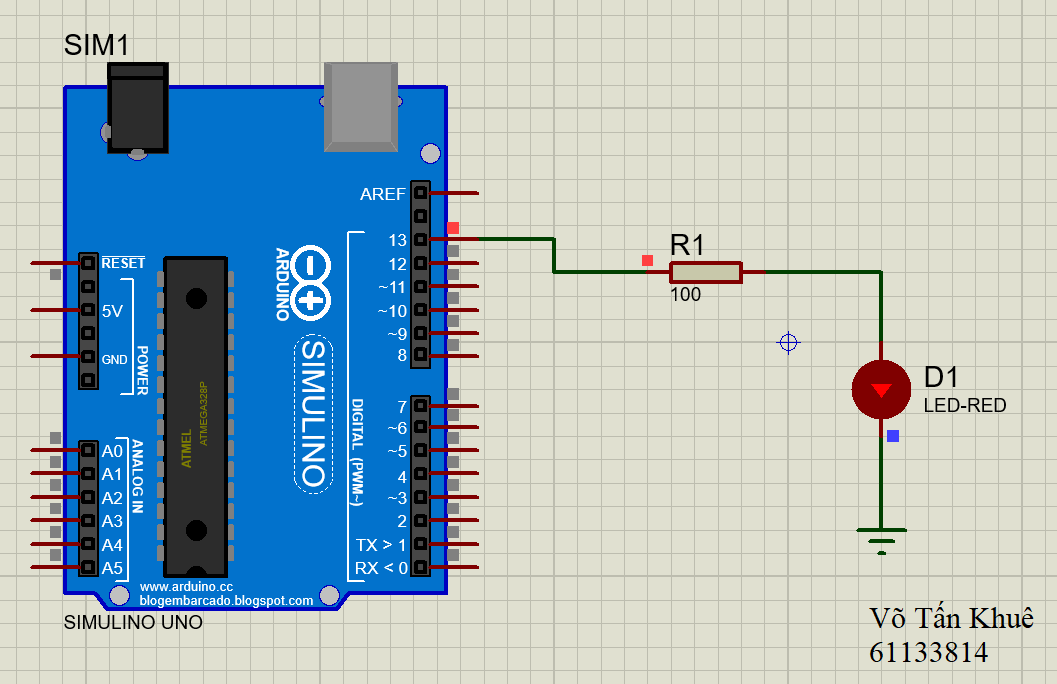
Sinh viên thực hiện: Võ Tấn Khuê Mã số sinh viên: 61133814

# Bài 1. Nháy đèn Led

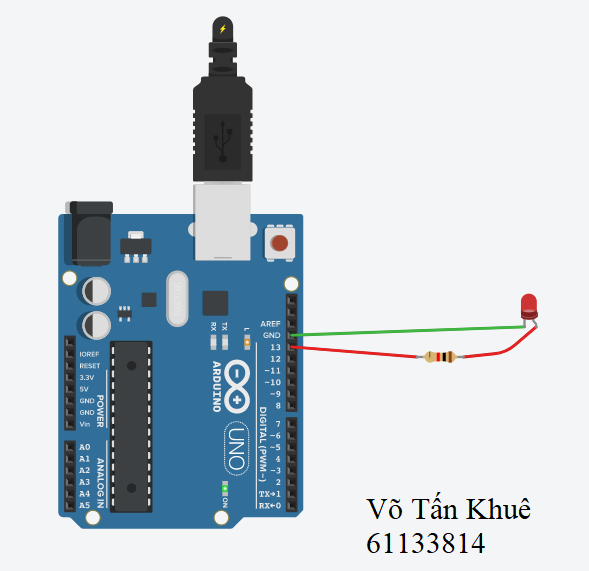
## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều kiển bật/tắt đèn Led trong thời gian 1 giây, đèn Led được kết nối vào cổng số 13 của board mạch Arduino.

## Sơ đồ thiết kế

****

Hình 1. Sơ đồ mạch Proteus



Hình 2. Sơ đồ mạch Tinkercad

## Đặc điểm của linh kiện

* 1 mạch Arduino Uno
* 1 đèn Led-RED
* 1 điện trở: 100 Ω

## Code chương trình

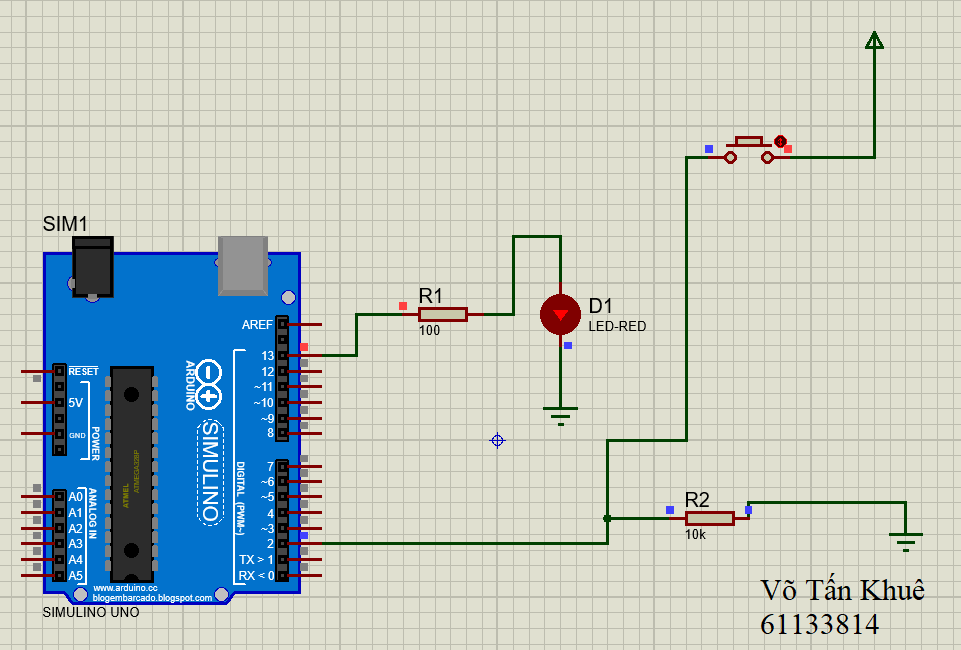
|  |
| --- |
| void **setup**() {  pinMode(13, OUTPUT);  }  void **loop**() {  digitalWrite(13, HIGH); // Bật Led  delay(1000); // Để Led sáng 1 giây  digitalWrite(13, LOW); // Tắt Led  delay(1000); // Trong 1 giây  } |

# Bài 2. Nút bấm đề bật đèn Led

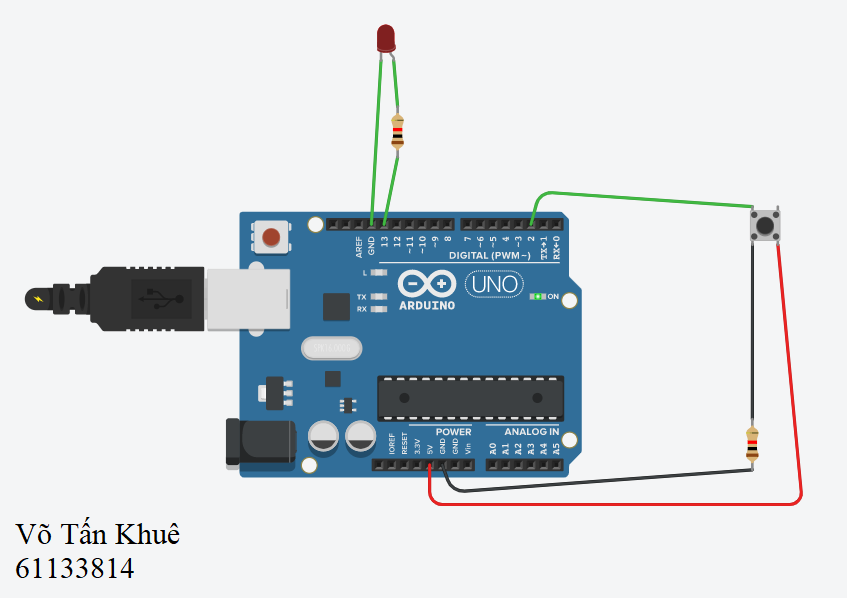
## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều kiển bật/tắt đèn Led thông qua nút bấm đề, đèn Led được kết nối vào cổng số 13 của board mạch Arduino.

## Sơ đồ thiết kế

****

Hình 3. Sơ đồ mạch Proteus

****

Hình 4. Sơ đồ mạch Tinkercad

## Đặc điểm của linh kiện

* 1 đèn Led
* 1 mạch Arduino Uno
* 2 điện trở: 100 Ωnhằm hạn chế cường độ dòng điện qua đèn, giúp đèn sáng an toàn hơn.
* 1 nút bấm (Push Button)

## Code chương trình

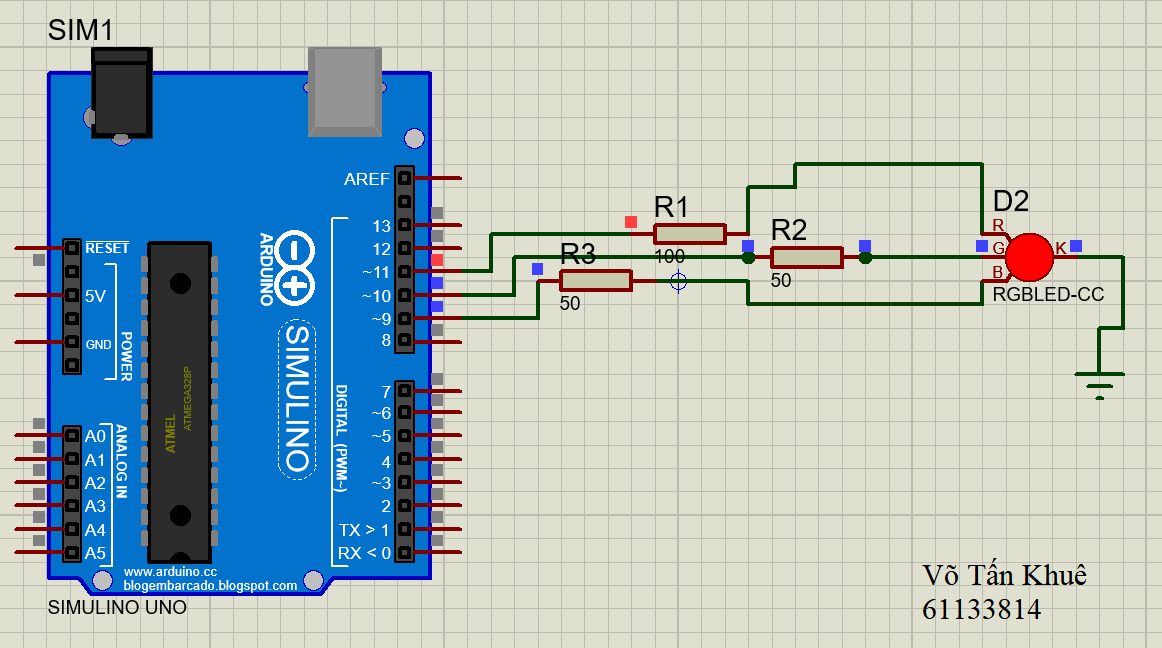
|  |
| --- |
| int x =0;  void **setup**() {  pinMode(2,INPUT);  pinMode(13,OUTPUT);  }  void **loop**() {  x = digitalRead(2); // Đọc tín hiêu ở đầu 2 và gán vào x  if(x == HIGH) {  digitalWrite(13, HIGH); // Bật đèn  } else {  digitalWrite(13, LOW); // Tắt đèn  }  delay(1000);  } |

# Bài 3. Nháy đèn LED RGB

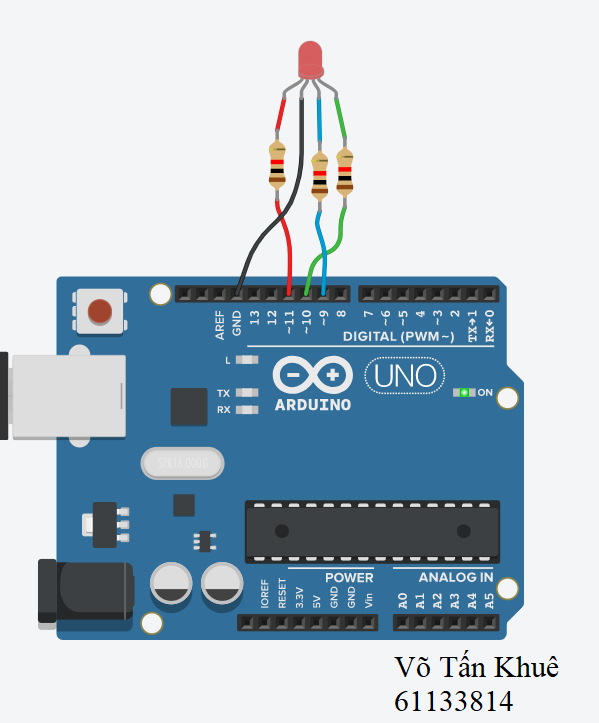
## Mô tả

Hệ thống được thiết kế điều khiển làm đèn RGB sáng ở màu đỏ, xanh lá cây, xanh dương, đèn được kết nối vào Arduino ở các cổng số 9, 10 ,11 (các chân PWM vì các cổng PWM cho phép bạn xuất ra xung PWM với độ phân giải 8bit, tức là tạo ra các giá trị từ 0 --> 28- l tương ứng với 0V --> 5V)

## Sơ đồ thiết kế



Hình 5. Sơ đồ mạch Proteus



Hình 6. Sơ đồ mạch Tinkercad

## Đặc điểm của linh kiện

* 3 điện trở: 100 Ω
* 1 Đèn RGB (Red, Green, Blue) trông giống như đèn LED thông thường, tuy nhiên, bên trong đèn thực sự có ba đèn LED, một màu đỏ, một màu xanh lá cây, một màu xanh lam. Bằng cách kiểm soát cường độ của từng đèn LED riêng lẻ, bạn có thể kết hợp khá nhiều màu sắc mà bạn muốn. Tổng số màu đèn RGB có thể sáng là 2563 màu.
* 1 mạch Arduino Uno

## Code chương trình

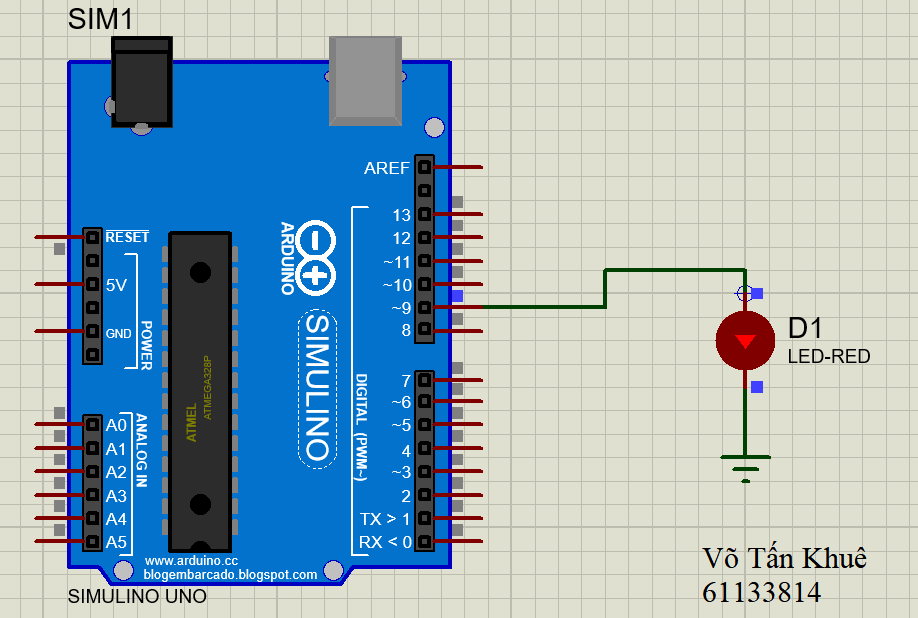
|  |
| --- |
| int denDo = 9; // Bạn có thể thay đổi các cổng khác sao cho phải là  int denXanhDuong = 10; // các cổng PWM (3, 5, 6, 9, 10, và 11)  int denXanhLa = 11;  void **setup**() {  pinMode(denDo, OUTPUT);  pinMode(denXanhDuong, OUTPUT);  pinMode(denXanhLa, OUTPUT);  }  void **loop**() {  // Bạn có thể thay đổi màu bằng cách đổi lại các giá trị cường độ dòng điện //qua các cổng  analogWrite(denDo, 0);  analogWrite(denXanhDuong, 255);  analogWrite(denXanhLa, 0);  delay(500);  analogWrite(denDo, 255);  analogWrite(denXanhDuong, 0);  analogWrite(denXanhLa, 0);  delay(500);  analogWrite(denDo, 0);  analogWrite(denXanhDuong, 0);  analogWrite(denXanhLa, 255) ;  delay(500);  } |

# Bài 4. Đèn LED sáng dần

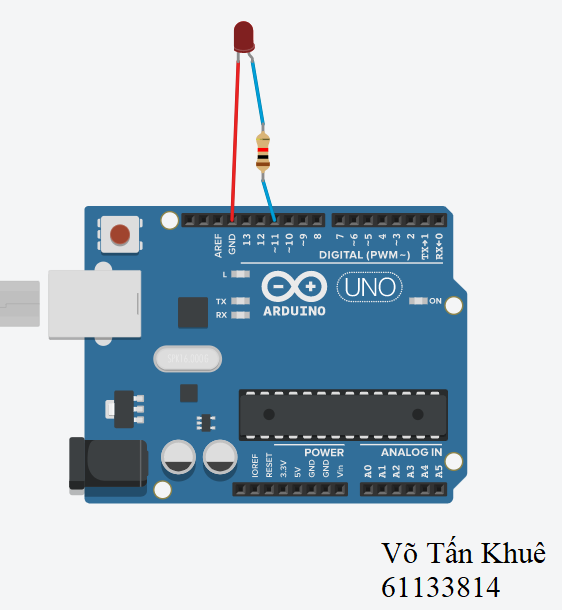
## Mô tả

Hệ thống được thiết kế đèn Led sáng dần sau đó giảm dần, đèn được kết nối vào Arduino ở cổng số 9.

## Sơ đồ thiết kế



Hình 7. Sơ đồ mạch Proteus



Hình 8. Sơ đồ mạch Tinkercad

## Đặc điểm của linh kiện

* 1 đèn Led
* 1 Điện trở: 100 Ω
* 1 mạch Arduino Uno

## Code chương trình

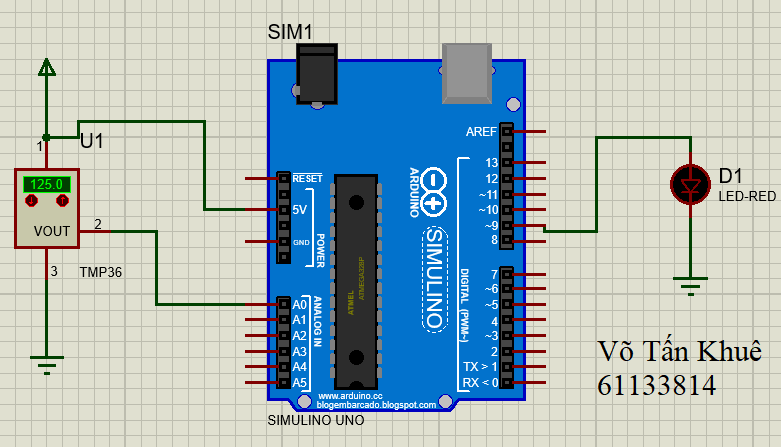
|  |
| --- |
| int i = 0;  void **setup**() {  pinMode(9,OUTPUT); // Bạn có thể sử dụng cổng PWM khác như 3, 5, // 6, 9, 10, và 11  }  void **loop**() {  for (i = 0; i<=255; i +=5) {  analogWrite(9,i);  delay(30);  }  For (i = 0; i<=255; i -=5) {  analogWrite(9,i);  delay(30);  }  } |

# Bài 5. Cảm biến nhiệt độ

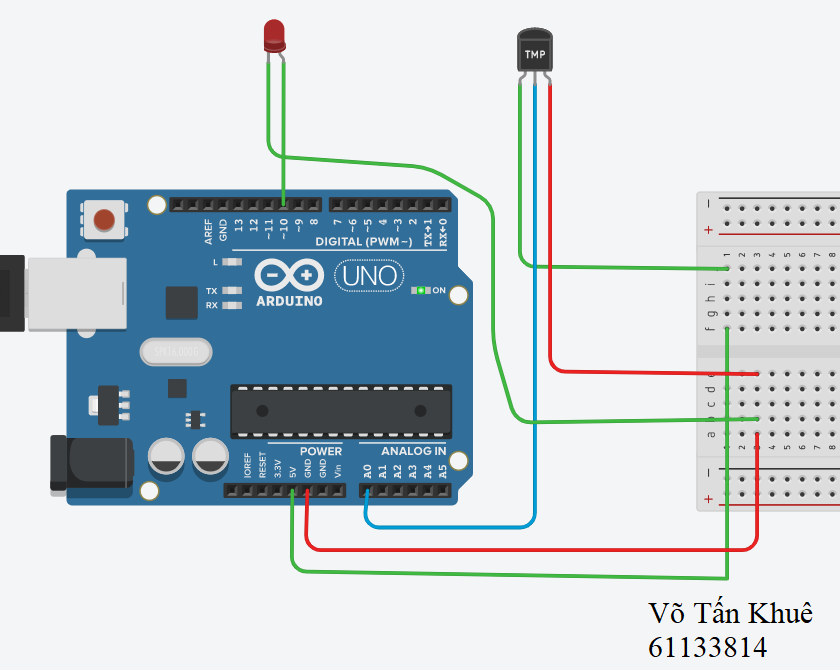
## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều kiển bật/tắt đèn Led thông qua cảm biến nhiệt độ Tmp36, đèn Led được kết nối vào cổng 9 của mạch Arduino.

## Sơ đồ thiết kế



Hình 9. Sơ đồ mạch Proteus



Hình 10. Sơ đồ mạch Tinkercad

## Đặc điểm của linh kiện

* 1 đèn Led
* 1 mạch Arduino Uno
* 1 Tmp36 là một cảm biến nhiệt độ độ chính xác, điện áp thấp do Analog Devices sản xuất. Nó là một con chip cung cấp đầu ra điện áp tỷ lệ tuyến tính với nhiệt độ tính bằng °C ( từ -40oC đến 125oC). Do đó rất dễ sử dụng với Arduino.

## Code chương trình

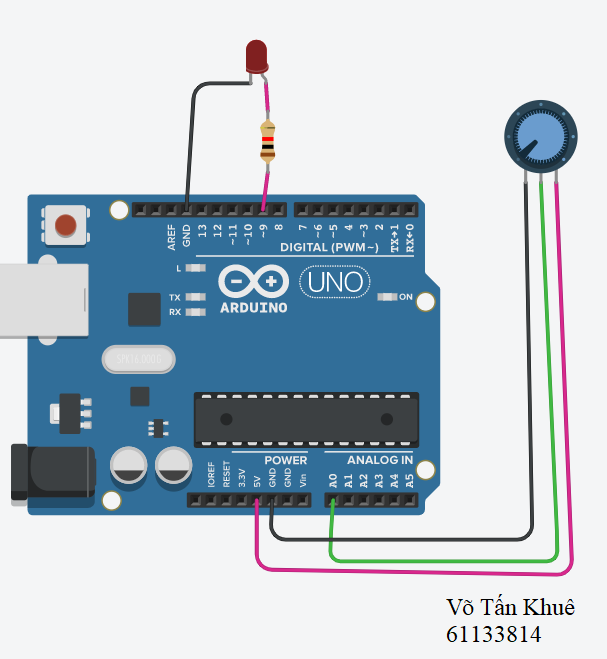
|  |
| --- |
| void **setup**()  { pinMode(9,OUTPUT); // bạn có thể thay đổi cổng khác }  void **loop**()  {  int giaTri = analogRead(A0);  int nhietDo = map (giaTri, 20,358,-40,125);  if(nhietDo > 25){  digitalWrite(9,HIGH); // Bật đèn  }  else {  digitalWrite(9,LOW); // Tắt đèn  }  delay(1000);  } |

# Bài 6. Điều khiển độ sáng của đèn bằng chiết áp

## Mô tả

Hệ thống được thiết kế điều chỉnh độ sáng của đèn bằng chiết áp, đèn được kết nối Arduno ở cổng số 9 còn chiếc áp ở cổng A0

## Sơ đồ thiết kế



Hình 11. Sơ đồ mạch Tinkercad

## Đặc điểm của linh kiện

* 1 đèn Led
* 1 điện trở (100 Ω)
* 1 chiết áp (Potentiometer)
* 1 mạch Arduino Uno

## Code chương trình

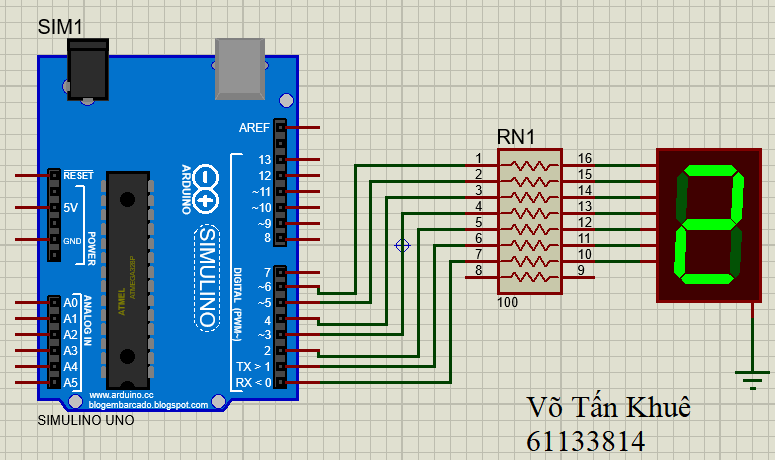
|  |
| --- |
| int x =0;  void **setup**()  {  pinMode(9, OUTPUT);  pinMode(A0, INPUT);  }  void **loop**()  {  x = analogRead(A0);  int doSang = map(x,0,1023,0,255); // Chuyển sang độ sáng  analogWrite(9,doSang);  } |

# Bài 7. Led 7 đoạn

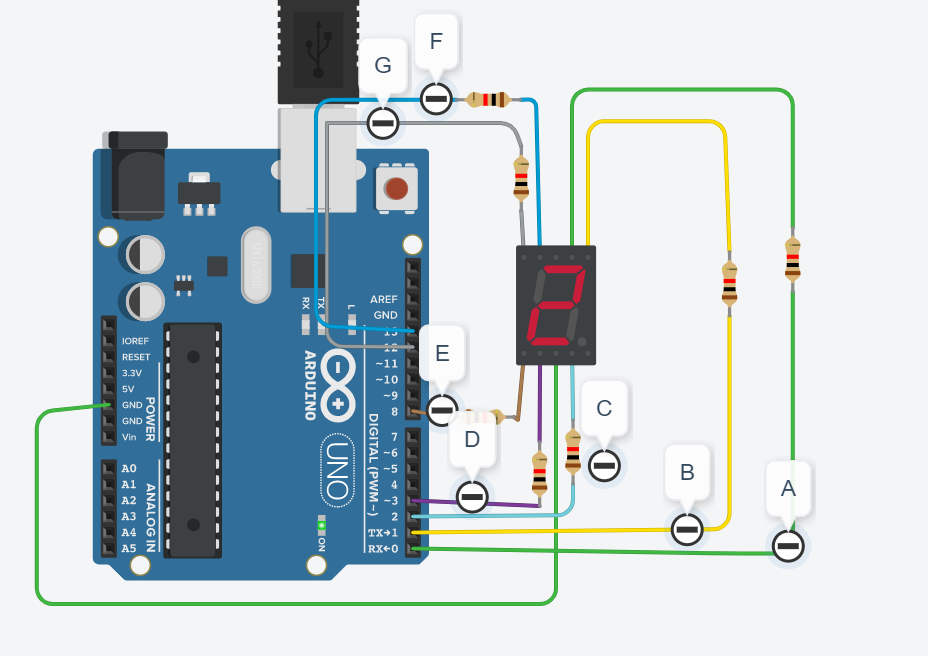
## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều kiển bật/tắt đèn Led 7 đoạn trong thời gian 1 giây, đèn Led được kết nối board mạch Arduino thông qua điện trở 16 cổng RES16DIPIS.

## Sơ đồ thiết kế



Hình 12. Sơ đồ mạch Proteus



Hình 13. Sơ đồ mạch Tinkercad

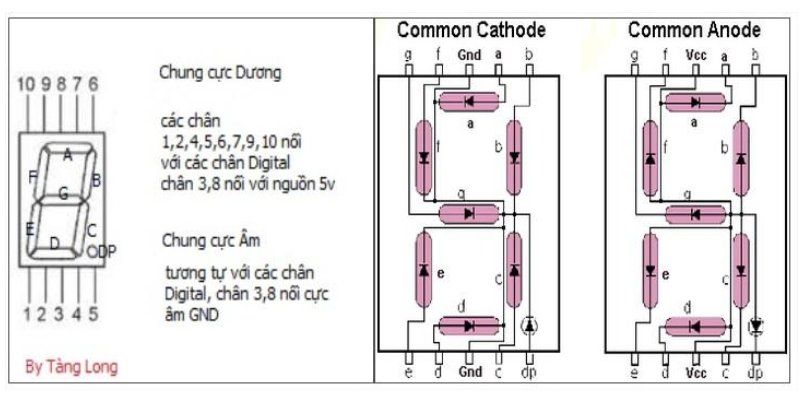
## Đặc điểm của linh kiện

* 1 đèn Led 7 đoạn

Led 7 đoạn thật chất chỉ là bảy đèn Led được kết nối song song. Mỗi đèn trong số bảy đèn Led được gọi là một đoạn vì khi được chiếu sáng, đoạn đó tạo thành một phần của chữ số (cả Thập phân và Hệ lục phân) sẽ được hiển thị. Đèn Led thứ 8 bổ sung bên cạnh vì đôi khi được sử dụng để chỉ báo dấu chấm thập phân. Mỗi đoạn led được đánh dấu từ a tới g. Đèn Led có bộ sung đèn thứ tám gọi là “chấm thập phân” (Decimal Point) ký hiệu DP được sử dụng khi hiển thị số không phải là số nguyên.

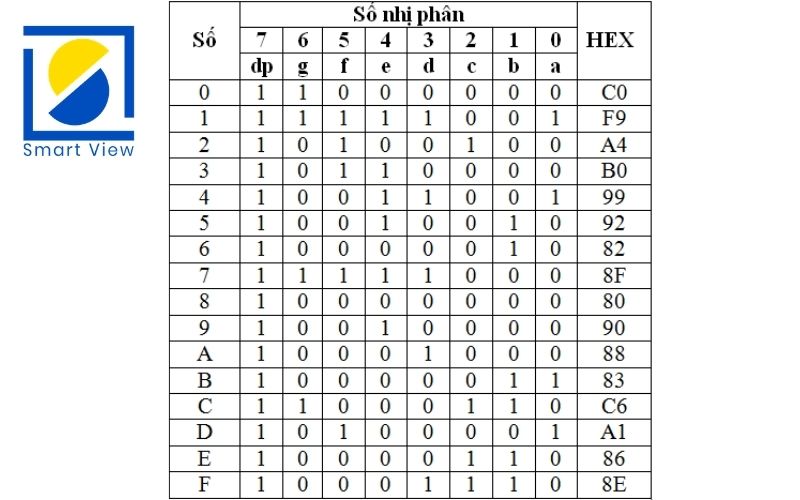
Dựa vào các cực được nối, có thể phân loại LED 7 đoạn như sau:

* Loại dương chung (Common Anode): nếu cực dương (anode) của tất cả 8 LED được nối với nhau và các cực âm (cathode) đứng riêng lẻ.
* Loại âm chung (Common Cathode): nếu cực âm (cathode) của tất cả 8 LED được nối với nhau và các cực dương (anode) đứng riêng lẻ.

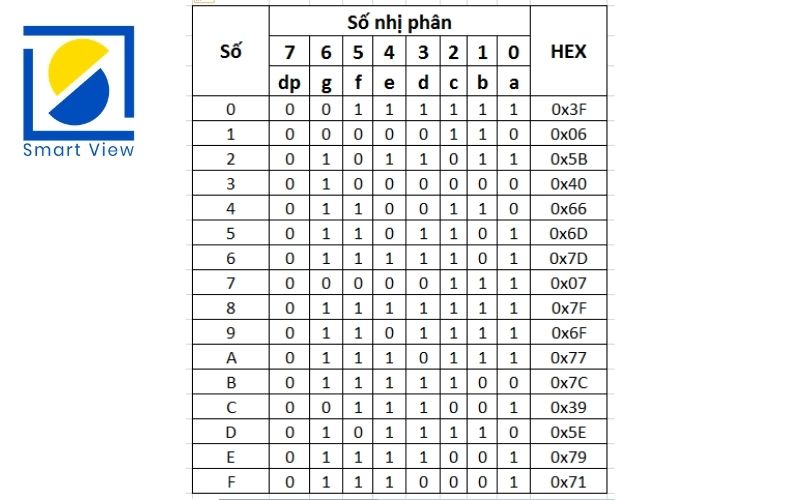


Nguyên lý hoạt động: muốn Led nào sáng thì Led đó phải được phân cực thuận. Do đó muốn tạo ra chữ số nào ta chỉ cần cho Led ở các vị trí tương ứng sáng lên. Bảng mô tả cách tạo ra các chữ số để hiển thị lên LED 7 đoạn:

+ Dương chung :



+ Âm chung:



* 1 mạch Arduino Uno
* 1 điện trở RES16DIPIS: 100 Ω, điện trở này có 16 cổng với các cặp tương ứng với nhau.

## Code chương trình

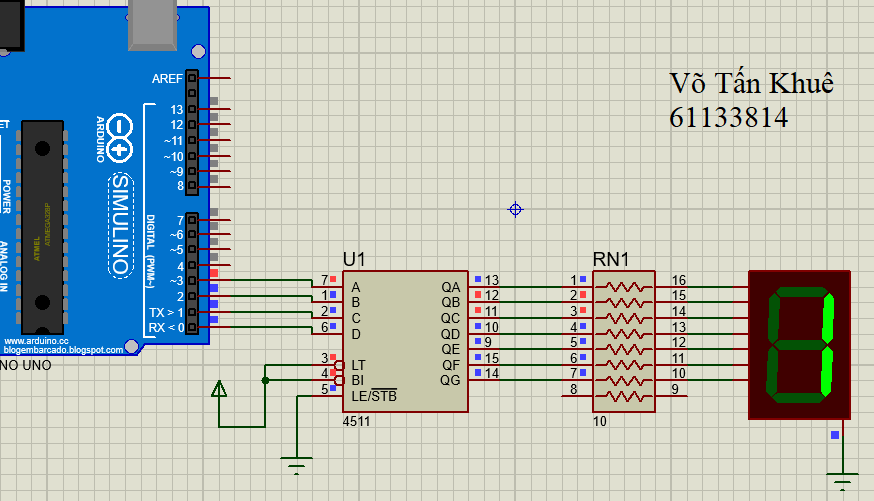
|  |
| --- |
| int a=6, b=5, c=4, d=3,e=2, f=1, g=0;  void **setup**()  {  pinMode(a, OUTPUT);  pinMode(b, OUTPUT);  pinMode(c, OUTPUT);  pinMode(d, OUTPUT);  pinMode(e, OUTPUT);  pinMode(f, OUTPUT);  pinMode(g, OUTPUT);  }  void **loop**()  { // Hiển thị số 0  digitalWrite(a, HIGH); digitalWrite(b, HIGH);  digitalWrite(c, HIGH); digitalWrite(d, HIGH);  digitalWrite(e, HIGH); digitalWrite(f, HIGH);  digitalWrite(g, LOW);  delay(1000); // Tạm ngưng 1s  // Hiển thị số 0  digitalWrite(a, LOW); digitalWrite(b, HIGH);  digitalWrite(c, HIGH); digitalWrite(d, LOW);  digitalWrite(e, LOW); digitalWrite(f, LOW);  digitalWrite(g, LOW);  delay(1000);  // Hiển thị số 0  digitalWrite(a, HIGH); digitalWrite(b, HIGH);  digitalWrite(c, LOW); digitalWrite(d, HIGH);  digitalWrite(e, HIGH); digitalWrite(f, LOW);  digitalWrite(g, HIGH);  delay(1000);  } |

# Bài 8. Led 7 đoạn BCD

## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều kiển bật/tắt đèn Led 7 đoạn trong thời gian 1 giây, đèn Led được kết nối vào board mạch Arduino thông qua RES16DIPIS và PCA9546APW

## Sơ đồ thiết kế



Hình 14. Sơ đồ mạch Proteus

## Đặc điểm của linh kiện

* 1 đèn led 7 đoạn
* 1 bộ điện trở 16 cổng 100 Ω
* 1 IC 4511

## Code chương trình

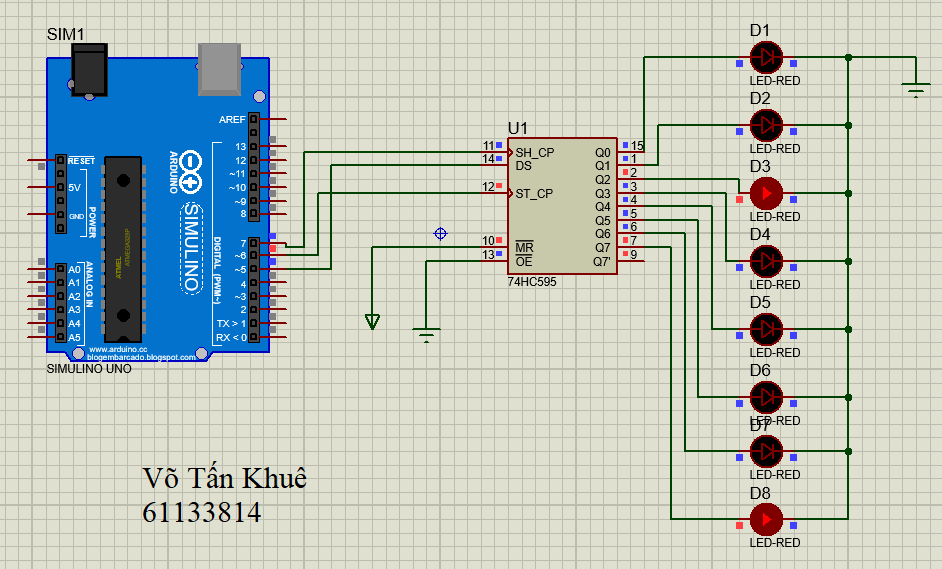
|  |
| --- |
| int a=0, b=1, c=2, d=3;  void **setup**() {  pinMode(a,OUTPUT);  pinMode(b,OUTPUT);  pinMode(c,OUTPUT);  pinMode(d,OUTPUT);  }  void **KHONG**(){  digitalWrite(a,LOW);  digitalWrite(b,LOW);  digitalWrite(c,LOW);  digitalWrite(d,LOW);  }  void **MOT**(){  digitalWrite(a,LOW);  digitalWrite(b,LOW);  digitalWrite(c,LOW);  digitalWrite(d,HIGH);  }  void **HAI**(){  digitalWrite(a,LOW);  digitalWrite(b,LOW);  digitalWrite(c,HIGH);  digitalWrite(d,LOW);  }  void **loop**() {  KHONG(); delay(500);  MOT(); delay(500);  HAI(); delay(500);  } |

# Bài 9. Sáng 8 Led bằng IC 74HC545

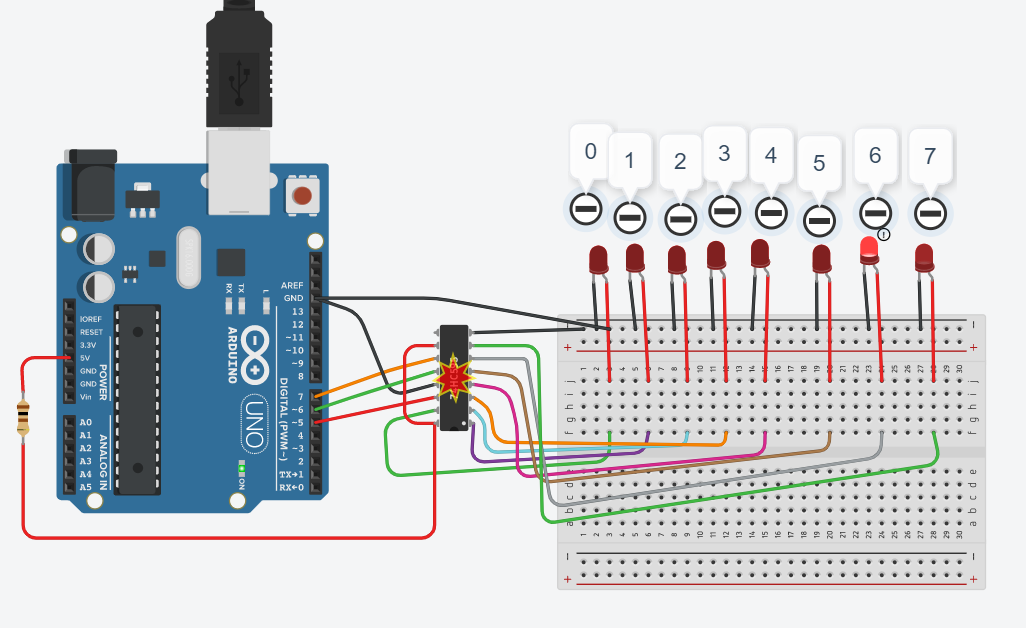
## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều kiển bật/tắt đèn Led trong thời gian 1 giây, đèn Led được kết nối vào cổng số 13 của board mạch Arduino.

## Sơ đồ thiết kế



Hình 15. Sơ đồ mạch Proteus



Hình 16. Sơ đồ mạch Tinkercad

## Đặc điểm của linh kiện

* 8 đèn LED
* 1 mạch IC 74HC5451
* 1 mạch Arduino Uno

## Code chương trình

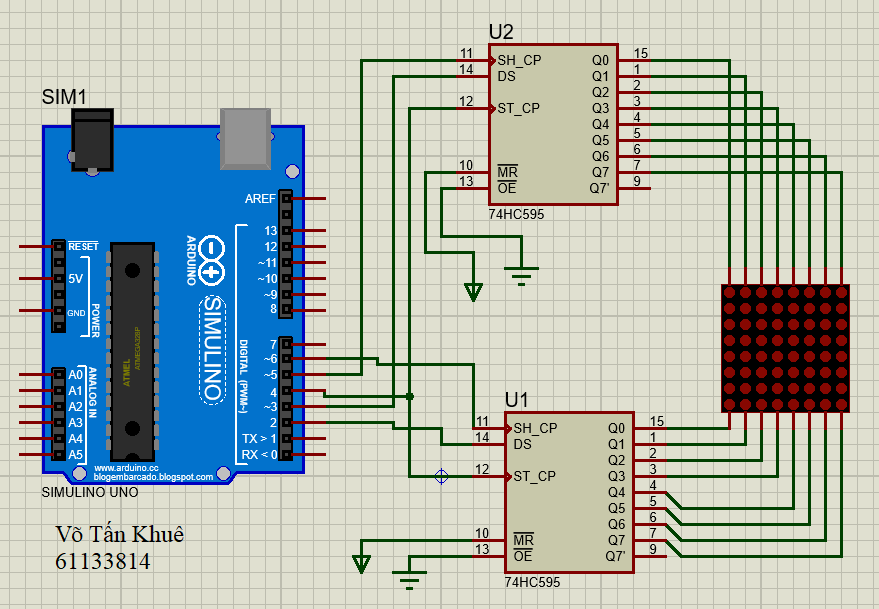
|  |
| --- |
| #define \_latch 6  #define \_clock 7  #define \_data 5  void **setup**()  {  pinMode(\_latch, OUTPUT); pinMode(\_clock, OUTPUT);  pinMode(\_data, OUTPUT);  }  void **loop**()  {  for (int i=0; i< 256;i++){  digitalWrite(\_latch,LOW);  shiftOut(\_data, \_clock,LSBFIRST, i);  digitalWrite( \_latch,HIGH);  delay(500);  }  } |

# Bài 10. Led Matrix 8x8

## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều đèn Led Matrix 8x8.

## Sơ đồ thiết kế



Hình 17. Sơ đồ mạch Proteus

## Đặc điểm của linh kiện

* 1 mạch Arduino Uno
* 2 mạch IC 74HC595
* 1 bảng mạch Led Matrix 8x8 Red

## Code chương trình

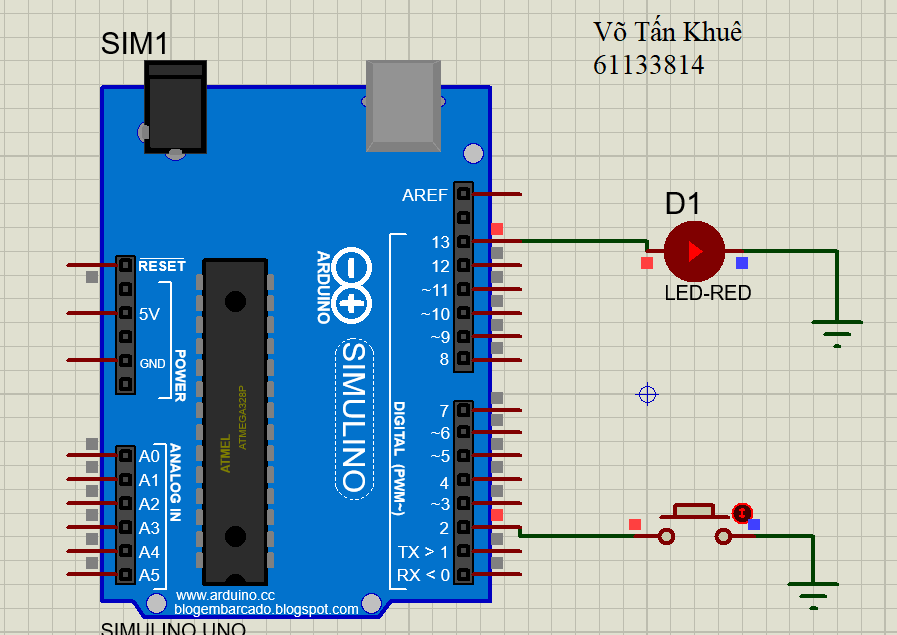
|  |
| --- |
|  |

# Bài 11. Sử dụng ngắt ngoài tắt đèn LED khi nhấn nút bấm đề

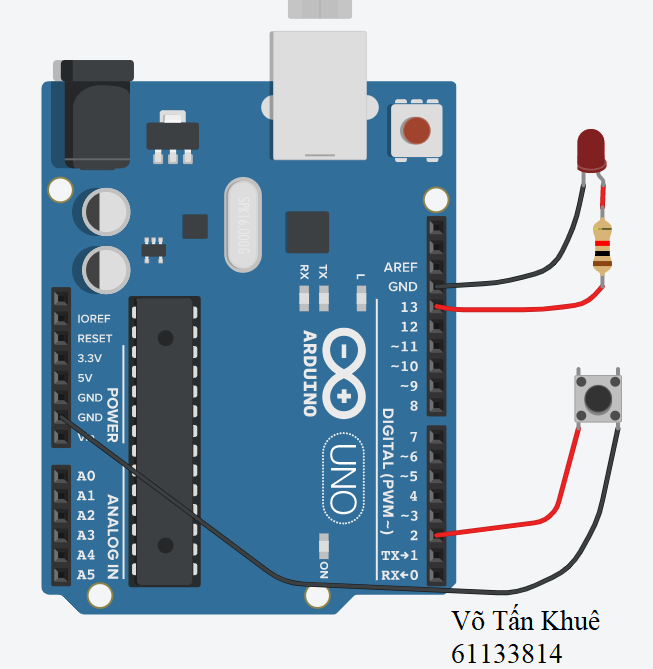
## Mô tả

Hệ thống được thiết kế thực hiện việc ngắt đèn Led đang sáng qua nút bấm đề.

## Sơ đồ thiết kế



Hình 18. Sơ đồ mạch Proteus



Hình 19. Sơ đồ mạch Tinkercad

## Đặc điểm của linh kiện

* 1 đèn Led
* 1 nút bấm

## Code chương trình

|  |
| --- |
| // Khai báo biến trangthai kiểu volatile để tránh sai số trong quá trình thực hiện ngắt.  volatile int trangthai = HIGH;  void **setup**()  {  pinMode(13, OUTPUT);  digitalWrite(2, HIGH); // Treo chân ngắt 0, chân số 2 lên mức cao.  attachInterrupt(0, daotrangthai, CHANGE);  }  void **loop**() // đoạn chương trình chính, vòng lặp vô tận.  {  digitalWrite(13, trangthai); // Gán trạng thái cho đen  }  void **daotrangthai**() // chương trình ngắt  {  trangthai = !trangthai; // đảo trang thái.  } |

# Bài 12. Điều khiển động cơ DC với IC L293D và Drduino

## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều kiển động cơ DC thông qua IC điều khiển động cơ L293D

Để có thể điều khiển hoàn toàn động cơ DC tức là chúng ta phải điều khiển tốc độ và chiều quay của nó. Để thực hiện diều này ta có thể kết hợp hai kỹ thuật sau. **Chế độ rộng xung (PWM)** để kiểm soát tốc độ và **Mạnh cầu H (H-Bridge)** để điều khiển hướng quay.

**PWM - Để kiểm soát tốc độ**

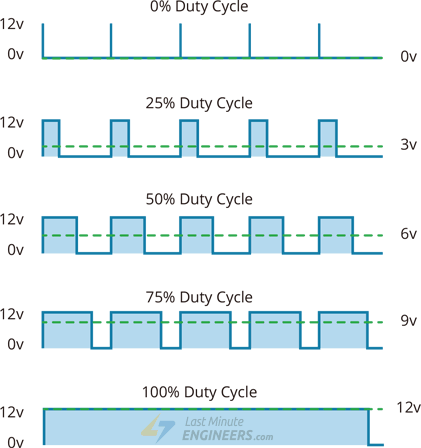
Tốc độ của động cơ DC có thể được điều khiển bằng cách thay đổi điện áp đầu vào của nó. Một kỹ thuật phổ biến để thực hiện việc này là sử dụng PWM (Pulse Width Modulation – Điều chế độ rộng xung).

PWM là một kỹ thuật trong đó giá trị trung bình của điện áp đầu vào được điều chỉnh bằng cách gửi một loạt các xung ON-OFF.

Điện áp trung bình tỷ lệ thuận với chiều rộng của các xung được gọi là chu kỳ làm việc (Duty Cycle).

Chu kỳ làm việc càng cao, điện áp trung bình cung cấp cho động cơ DC càng lớn (Động cơ quay nhanh) và chu kỳ làm việc càng thấp, điện áp trung bình cung cấp cho động cơ DC càng ít (Động cơ quay chậm).

Dưới đây hình ảnh minh họa kỹ thuật PWM với các chu kỳ làm việc khác nhau và điện áp trung bình.



Hình 20. Kỹ thuật điều chế độ rộng xung (PWM)

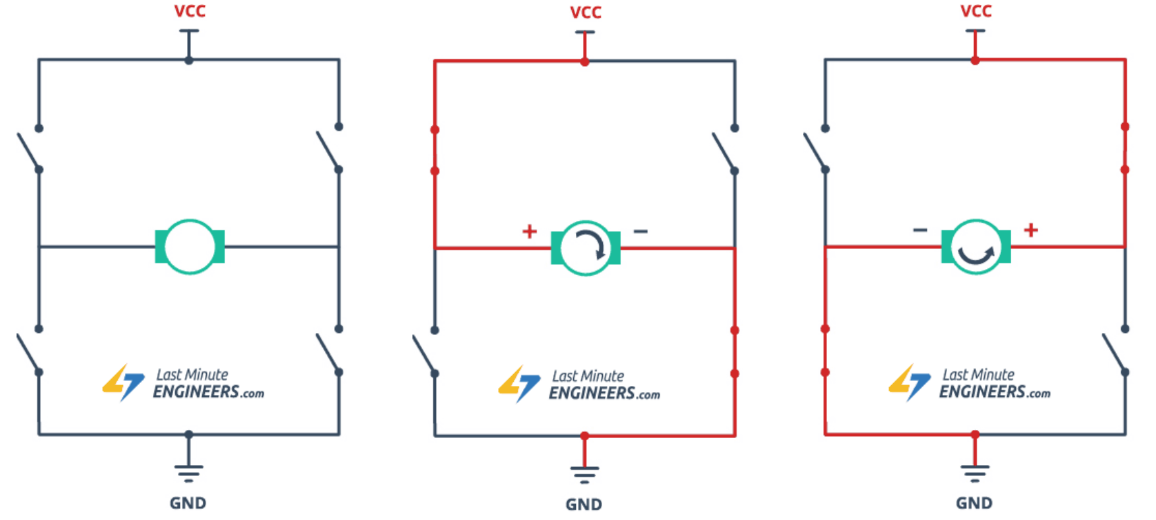
**H-Bridge - Để điều khiển hướng quay**

Chiều quay của động cơ DC có thể được điều khiển bằng cách thay đổi cực tính của điện áp đầu vào. Một kỹ thuật phổ biến để làm điều này là sử dụng mạch cầu H.

Một mạch cầu H gồm có bốn công tắc với động cơ ở trung tâm tạo thành một sắp xếp giống như hình chữ H.

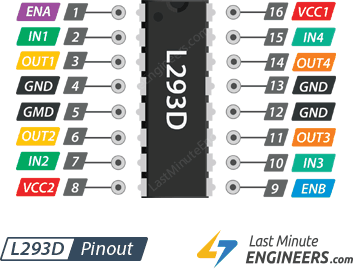
Đóng hai công tắc cụ thể cùng một lúc sẽ đảo ngược cực tính của điện áp đặt vào động cơ. Điều này gây ra sự thay đổi hướng quay của động cơ.

Hình ảnh động bên dưới minh họa mạch H-Bridge hoạt động.



Hình 21. H-Bridge hoạt động

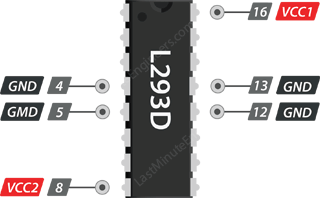
**IC điều khiển động cơ L293D**



Hình 22. Sơ đồ các chân của L293D

L293D là trình điều khiển động cơ H-Bridge có khả năng điều khiển một cặp động cơ DC cùng một lúc hoặc điều khiển hướng của hai động cơ này một cách độc lập. Điều đó có nghĩa là nó có thể truyền động riêng lẻ tối đa hai động cơ, lý tưởng cho việc xây dựng nền tảng robot hai bánh.

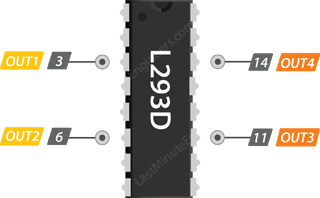
**Nguồn IC L293D**



Hình 23. Các chân nguồn và GND của L293D

IC L293D thực tế có chân nguồn đầu vào là **VCC1** và **VCC2**, trong đó **VCC1** dùng để kết nối với nguồn +5V để cho phép IC hoạt động còn **VCC2** dùng Kết nối với chân nguồn cho động cơ đang chạy (4.5V đến 36V). Cả 2 đầu này đều nối đến một điểm chung là **GND**.

**Các cổng kết nối đầu ra**



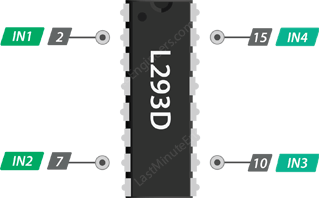
Hình 24. Các chân kết nối đầu ra của L293D

Các chân kết nối đầu ra của trình điều khiển động cơ L293D cho hai động cơ A và B được đưa ra qua các chân **OUT1**, **OUT2** và **OUT3**, **OUT4** nằm ở hai phía tương ứng. Bạn có thể kết nối hai động cơ DC có điện áp từ 4,5V đến 36V với các chân kết nối đầu ra này. Mỗi chân trên IC có thể cung cấp tới 600mA cho động cơ DC. Tuy nhiên, lượng dòng điện cug cấp cho động cơ còn phụ thuộc vào nguồn điện của hệ thống.

**Chân điều khiển**

Đối với các chân L293D, có hai loại chân điều khiển cho phép chúng ta điều khiển tốc độ và hướng quay của động cơ DC cùng một lúc. Direction control pins và Speed control pins.

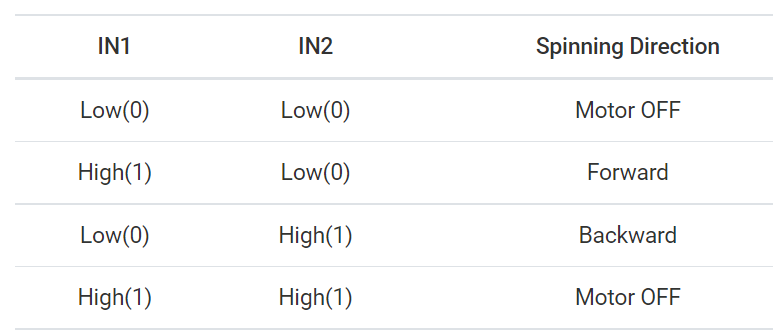
**Direction Control Pins**



Hình 25. Các chân điều khiển hướng của L293D

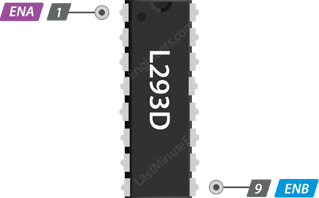
Sử dụng các chân điều khiển hướng, chúng ta có thể điều khiển động cơ quay tiến hoặc lùi. Các chân này điều khiển các công tắc của mạch H-Bridge nằm bên trong IC L293D. IC có hai chân điều khiển hướng cho mỗi phía. Các chân **IN1**, **IN2** điều khiển hướng quay của động cơ A và **IN3**, **IN4** điều khiển động cơ B.

Chiều quay của động cơ có thể được điều khiển bằng cách áp dụng logic HIGH (5 Volts) hoặc logic LOW (Ground) cho các chân này.



Hình 26. Biểu đồ minh họa cách thực hiện

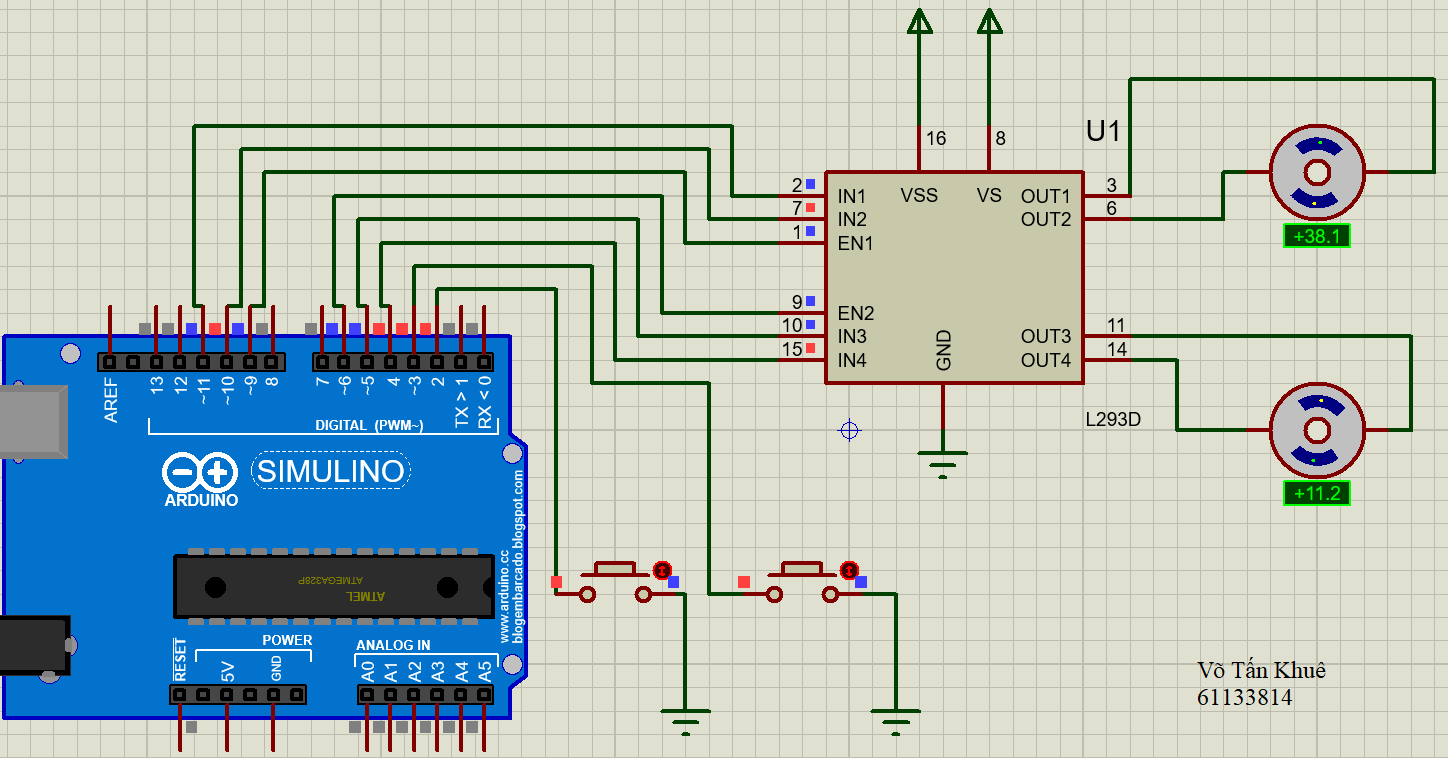
**Speed Control Pins**



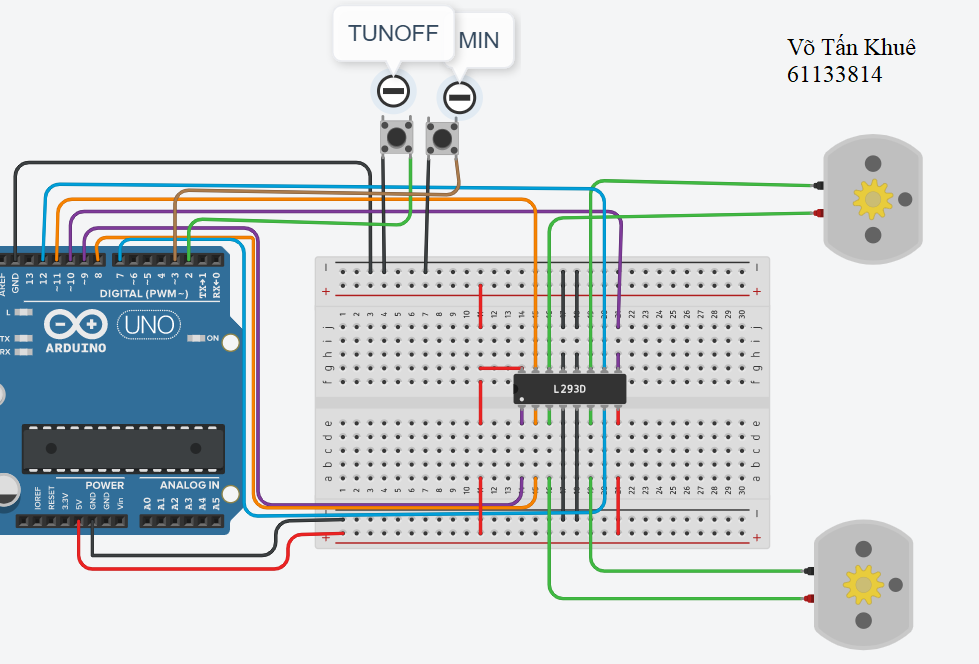
Hình 27. Các chân điều khiển tốc độ của L293D

Các chân điều khiển tốc độ **ENA** và **ENB** được sử dụng để bật, tắt và điều khiển tốc độ của động cơ A và động cơ B.

## Sơ đồ thiết kế



Hình 28. Sơ đồ mạch Proteus



Hình 29. Sơ đồ mạch Tinkercad

## Đặc điểm của linh kiện

## Code chương trình

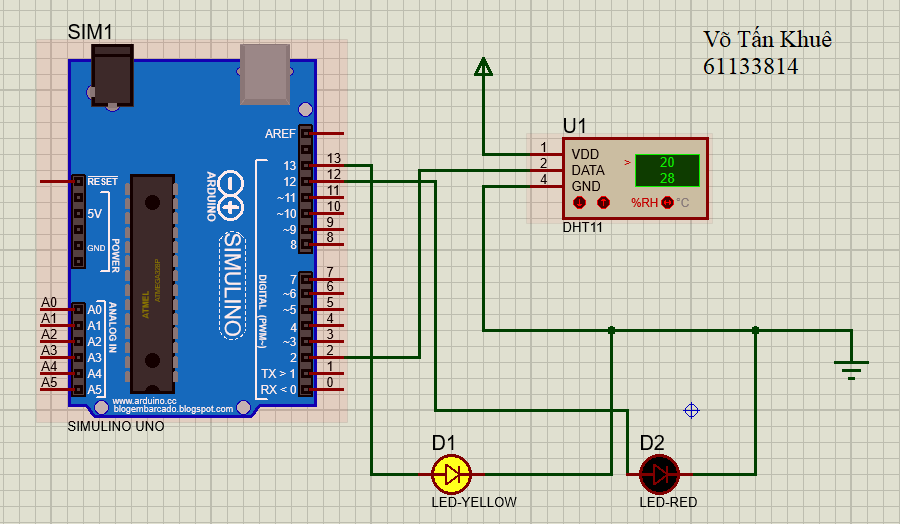
|  |
| --- |
| // Động cơ 1  int tocdo = 9; int inp1 = 11; int inp2 = 10;  // Động cơ 2  int tocdo2 = 6; int inp3 = 5; int inp4 = 4;  void setup()  {  pinMode(tocdo, OUTPUT); pinMode(inp1, OUTPUT);  pinMode(inp2, OUTPUT); pinMode(tocdo2, OUTPUT);  pinMode(inp3, OUTPUT); pinMode(inp4, OUTPUT);  digitalWrite(2, HIGH);  attachInterrupt(0, tatdongco, CHANGE);  digitalWrite(3, HIGH);  attachInterrupt(1,dieukientocdogiam, CHANGE);    //Tắt động cơ  digitalWrite(inp1,LOW); digitalWrite(inp2,LOW);  digitalWrite(inp3,LOW); digitalWrite(inp4,LOW);  }  void tatdongco()  {  digitalWrite(inp1,LOW); digitalWrite(inp2,LOW);  digitalWrite(inp3,LOW); digitalWrite(inp4,LOW);  }  void directionControl()  {  // Cho động cơ quay với tốc độ tối đa  // Giá trị của PWM thay đổi từ 0 đến 255  analogWrite(tocdo, 255);  analogWrite(tocdo2, 255);  // Tắt động cơ A và B  digitalWrite(inp1, HIGH); digitalWrite(inp2, LOW);  digitalWrite(inp3, HIGH); digitalWrite(inp4, LOW);  delay(2000);  // Thay đổi chiều quay của động cơ  digitalWrite(inp1, LOW); digitalWrite(inp2, HIGH);  digitalWrite(inp3, LOW); digitalWrite(inp4, HIGH);  delay(2000);    // Tắt tất cả các động cơ  digitalWrite(inp1, LOW); digitalWrite(inp2, LOW);  digitalWrite(inp3, LOW); digitalWrite(inp4, LOW);  }  void speedControl()  {  digitalWrite(inp1, LOW); digitalWrite(inp2, HIGH);  digitalWrite(inp3, LOW); digitalWrite(inp4, HIGH);    for(int i=0; i<256 ;i++){  analogWrite(tocdo,i);  analogWrite(tocdo2,i);  delay(10);  }    for(int i=255;i >= 0; --i){  analogWrite(tocdo,i);  analogWrite(tocdo2,i);  delay(10);  }  //tat dong co  digitalWrite(inp1,LOW); digitalWrite(inp2,LOW);  digitalWrite(inp3,LOW); digitalWrite(inp4,LOW);  }  void dieukientocdotang()  {  //bat cac dong co  digitalWrite(inp1,LOW); digitalWrite(inp2,HIGH);  digitalWrite(inp3,LOW); digitalWrite(inp4,HIGH);    //tang toc do den max  for(int i=0; i<256 ;i++){  analogWrite(tocdo,i);  analogWrite(tocdo2,i);  delay(10);  }  }  void dieukientocdogiam()  {  //bat cac dong co  digitalWrite(inp1,LOW); digitalWrite(inp2,HIGH);  digitalWrite(inp3,LOW); digitalWrite(inp4,HIGH);    //giam toc do ve 0  for(int i=255;i >= 0; --i){  analogWrite(tocdo,i);  analogWrite(tocdo2,i);  delay(10);  }  }  void loop()  {  directionControl();  delay(1000);  speedControl();  delay(1000);  } |

# Bài 13. Hệ thống theo dõi nhiệt độ và độ ẩm dựa trên cảm biến DHT11

## Mô tả

Hệ thống được thiết kế bật tắt đèn sáng bằng cảm biến theo dõi nhiệt độ và độ ẩm DHT11.

## Sơ đồ thiết kế



Hình 30. Sơ đồ mạch Proteus

## Đặc điểm của linh kiện

* 1 DHT11 Là cảm biến số nhiệt độ và độ ẩm rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1-wire (giao tiếp digital 1-wire truyền dữ liệu duy nhất). Cảm biến được tích hợp bộ tiền xử lý tín hiệu giúp dữ liệu nhận về được chính xác mà không cần phải qua bất kỳ tính toán nào. Có ba chân kết nối Vcc, GND, OUT

Thông số kỹ thuật

Điện áp hoạt động: 3V - 5V (DC)

Dãi độ ẩm hoạt động: 20% - 90% RH, sai số ±5%RH

Dãi nhiệt độ hoạt động: 0°C ~ 50°C, sai số ±2°C

Khoảng cách truyển tối đa: 20m

* 1 mạch Arduino
* 1 đèn led vàng, 1 led đỏ

## Code chương trình

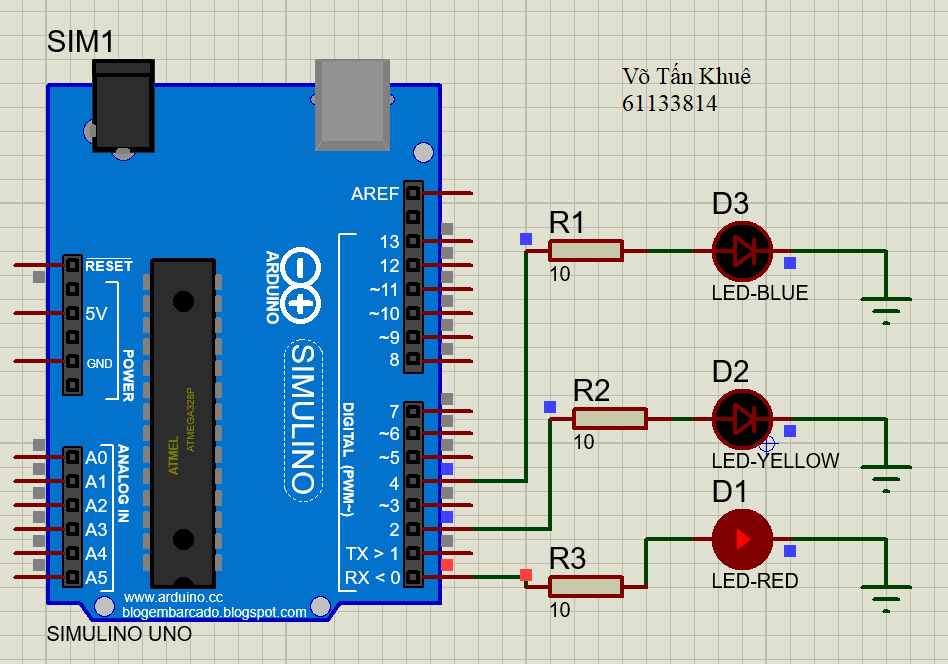
|  |
| --- |
| #include "DHT.h" // Khai báo thư viện DHT.h  #define dht\_1 2  #define DHTTYPE DHT11 //Khai báo DHT11  DHT dht(dht\_1, DHTTYPE); // Hàm DHT  void **setup**() {  dht.begin();  }  void **loop**() {  float doam = dht.readHumidity(); //Lấy chỉ số độ ẩm  float nhietdo = dht.readTemperature(); //Lấy chỉ số nhiệt độ  if(nhietdo > 30) { // Nếu nhiệt độ lớn hơn 30 thì sáng đèn đỏ  digitalWrite(12,HIGH);  }  Else digitalWrite(12,LOW);  if(doam <40 ) { // Nếu độ ẩm nhỏ hơn 30 thì sáng đèn vàng  digitalWrite(13,HIGH);  }  Else digitalWrite(13,LOW);  } |

# Bài 14. Đèn Giao Thông

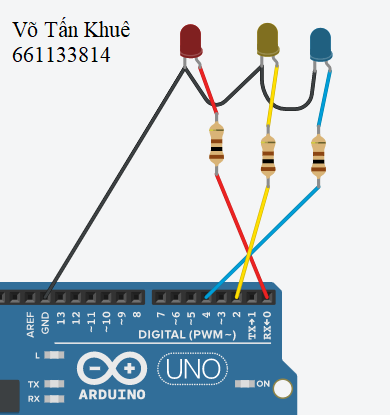
## Mô tả

Hệ thống được thiết kế tạo thành đèn giao thông qua việc bật tắt ba đèn đỏ, xanh, vàng trong một thời gian nhất định. Ở ví dụ này đèn đỏ sáng trong 45s gắng vào mạch Arduino ở cổng 0, đèn vàng sáng 10s ở cổng 2, đèn xanh sáng 30s ở cổng 4.

## Sơ đồ thiết kế



Hình 31. Sơ đồ mạch Proteus



Hình 32. Sơ đồ mạch Tinkercad

## Đặc điểm của linh kiện

* 1 Mạch Arduino
* 3 điện trở: 100 Ω
* 1 đèn Led màu đỏ, 1 Led màu vàng, 1 Led xanh

## Code chương trình

|  |
| --- |
| Int dem = 0;  void **setup**()  {  pinMode(0, OUTPUT);  pinMode(2, OUTPUT);  pinMode(4, OUTPUT);  }  void **loop**()  {  dem = dem + 1; delay(1000); //Biến dem để tính thời gian trong 1s  if(dem == 85){ //Tổng thời gian 3 đèn sáng là 85s  dem=0;  }  if(dem<=45){  digitalWrite(0, HIGH);  digitalWrite(2, LOW); //Bật đèn đỏ  digitalWrite(4, LOW);  }  if(dem>45&&dem<=55){  digitalWrite(0, LOW);  digitalWrite(2, HIGH); // Bật đèn vàng  digitalWrite(4, LOW);  }  if(dem>=56){  digitalWrite(0, LOW);  digitalWrite(2, LOW); //Bật đèn xanh  digitalWrite(4, HIGH);  }  } |

# Bài 15.

## Mô tả

## Sơ đồ thiết kế

## Đặc điểm của linh kiện

## Code chương trình