Nha Trang - 2021

****

BÁO CÁO HỆ THỐNG LẬP TRÌNH NHÚNG

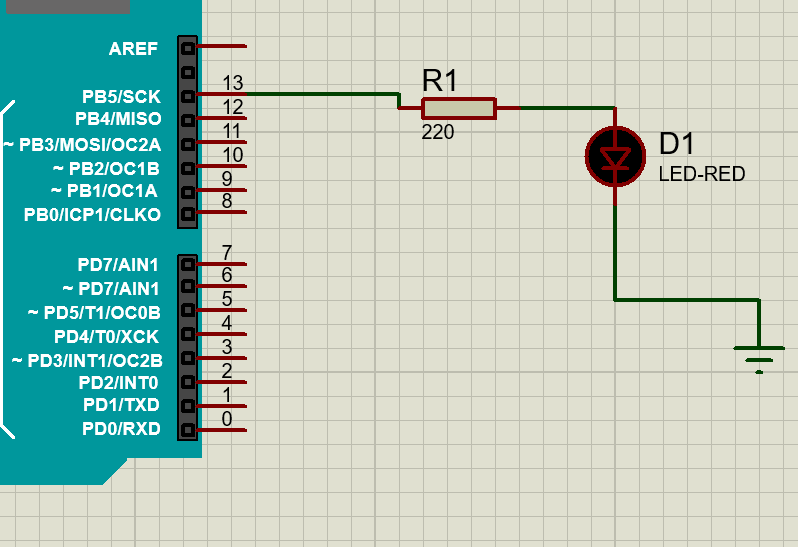
Sinh viên thực hiện: Mạnh Văn Hiệp Mã số sinh viên: 61133631

# Bài 1. Nháy đèn Led

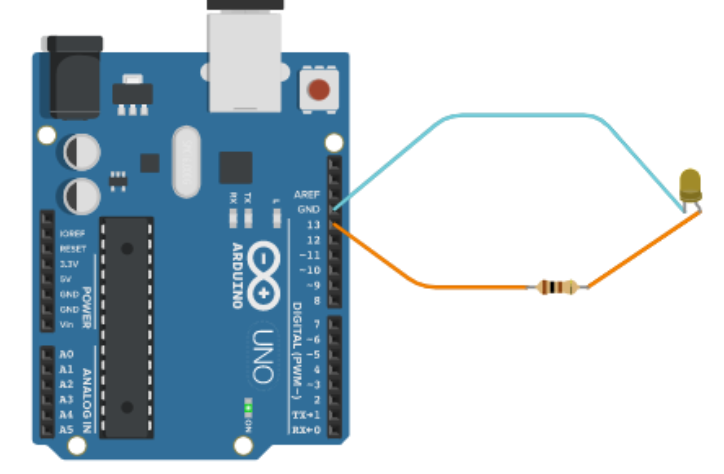
## Mô tả

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: nháy đèn led



Hình 2: nháy đèn led

## Đặc điểm của linh kiện

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

## Code chương trình

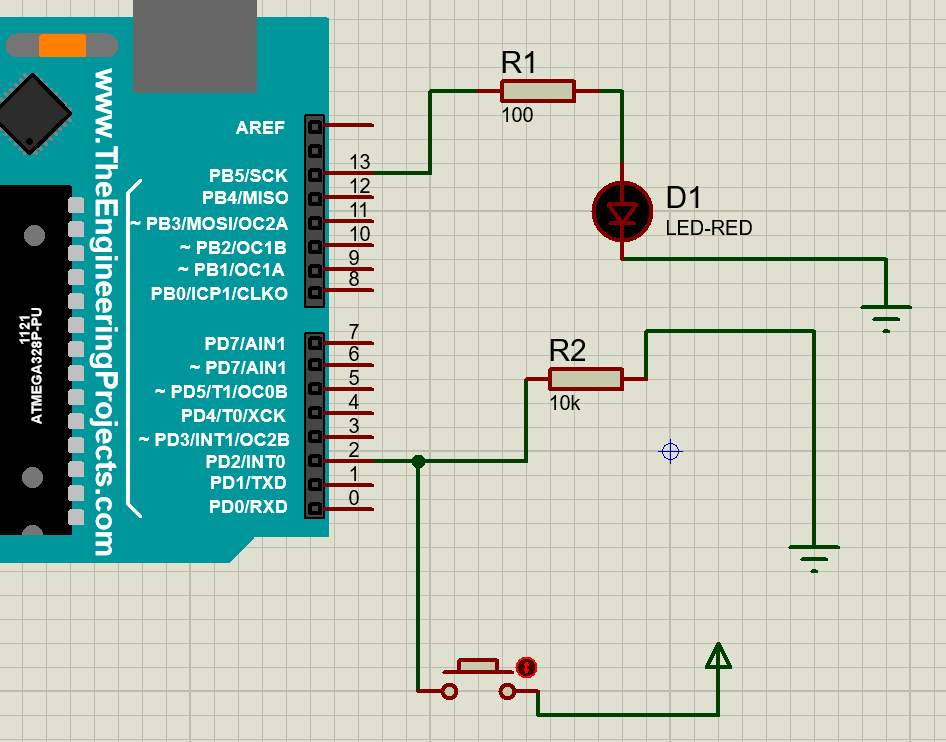
|  |
| --- |
| void **setup**() {  pinMode(13, OUTPUT);  }  void **loop**() {  digitalWrite(13, HIGH);  delay(1000);  digitalWrite(LED\_BUILTIN, LOW);  delay(1000);  } |

# Bài 2. Nút bấm đề bật đèn Led

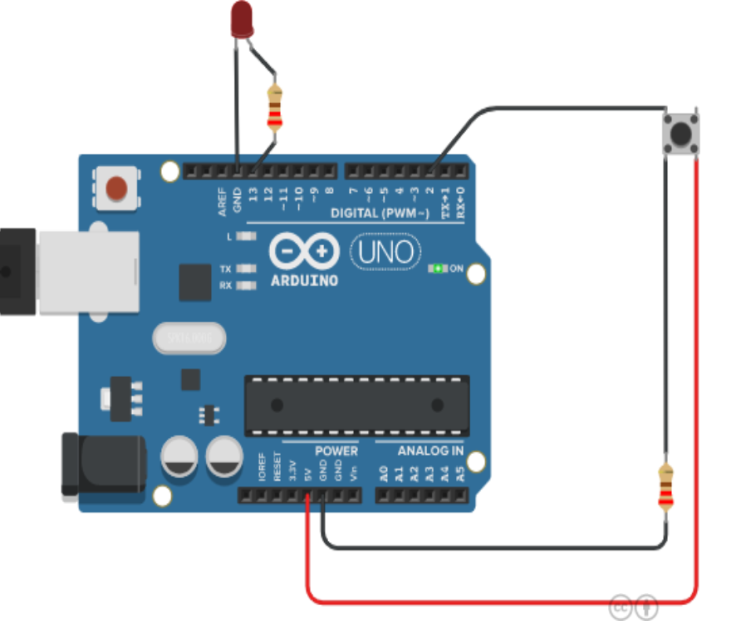
## Mô tả

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

## Sơ đồ thiết kế



Hình 3: Nút bấm led



Hình 4: Nút bấm led

## Đặc điểm của linh kiện

* 1 đèn Led
* 1 mạch Arduino Uno
* 2 điện trở: 100 Ω
* 1 nút bấm

## Code chương trình

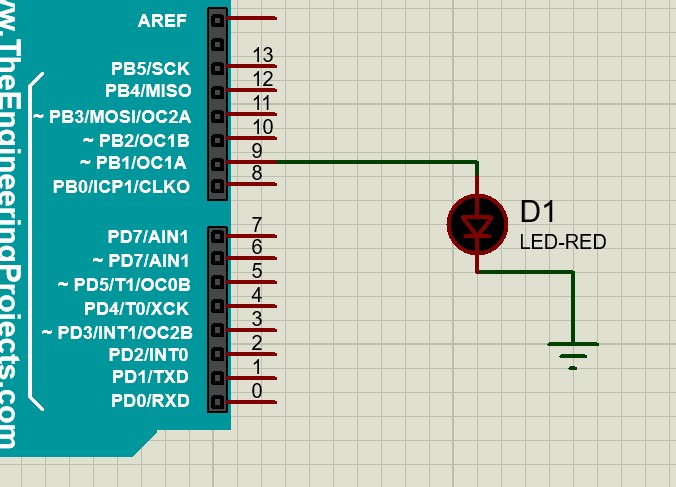
|  |
| --- |
| void **setup**() {  pinMode(ledPin, OUTPUT);  pinMode(buttonPin, INPUT);  }  void **loop**() {  buttonState = digitalRead(buttonPin);  if (buttonState == HIGH)  digitalWrite(ledPin, HIGH);  else  digitalWrite(ledPin, LOW);  } |

# Bài 3. Đèn LED sáng dần

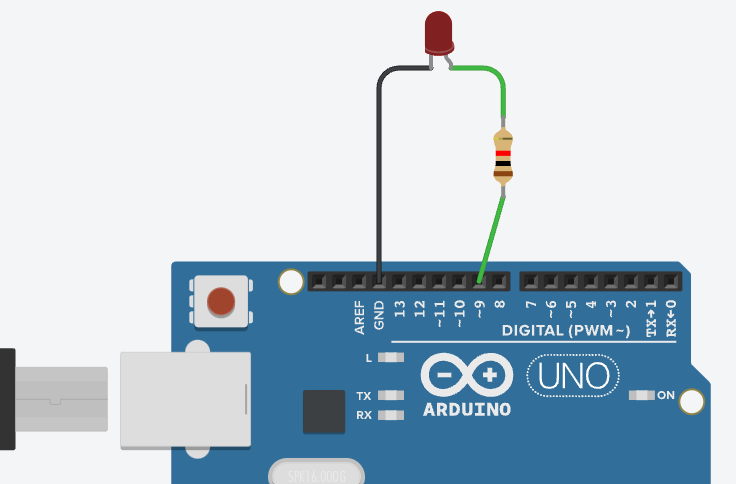
## Mô tả

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

## Sơ đồ thiết kế



Hình 5: Led sáng dần



Hình 6: Led sáng dần

## Đặc điểm của linh kiện

* 1 đèn LED
* 1 Điện trở
* 1 mạch Arduino Uno

## Code chương trình

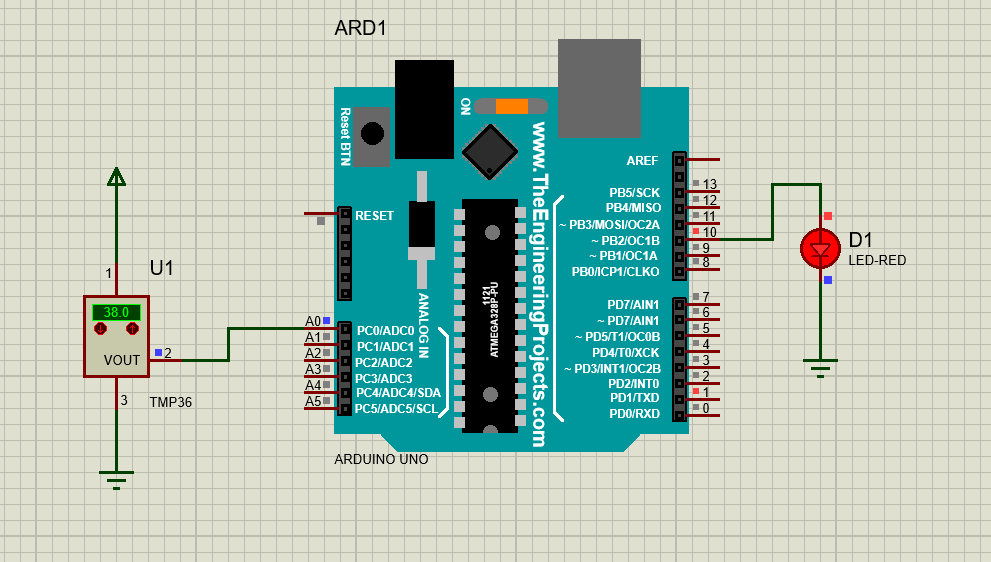
|  |
| --- |
| int i = 0;  void **setup**() {  pinMode(9,OUTPUT);  }  void **loop**() {  for (i = 0; I < = 255; i +=5) {  analogWrite(9,i);  delay(30);  }  For (i = 0; i <=255; i -= 5) {  analogWrite(9,i);  delay(30);  }  } |

# Bài 4. CẢM BIẾN NHIỆT ĐỘ

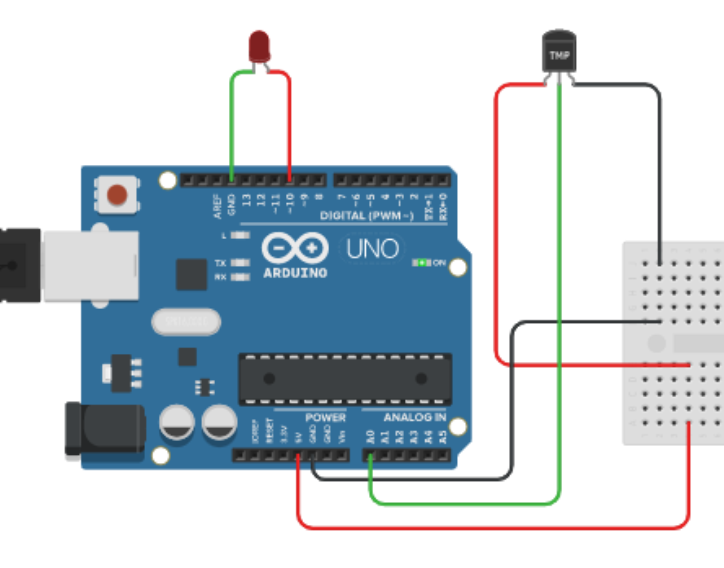
## Mô tả

Hệ thống được thiết kế thực hiện việc bật đèn led nếu bộ cảm biến > 380, đèn Led được kết nối vào cổng số 10 của board mạch Arduino.

## Sơ đồ thiết kế



Hình 7: Cảm biến nhiệt độ



Hình 8: Cảm biến nhiệt độ

## Đặc điểm của linh kiện

* Đèn LED
* TMP36
* Mạch Arduino Uno

## Code chương trình

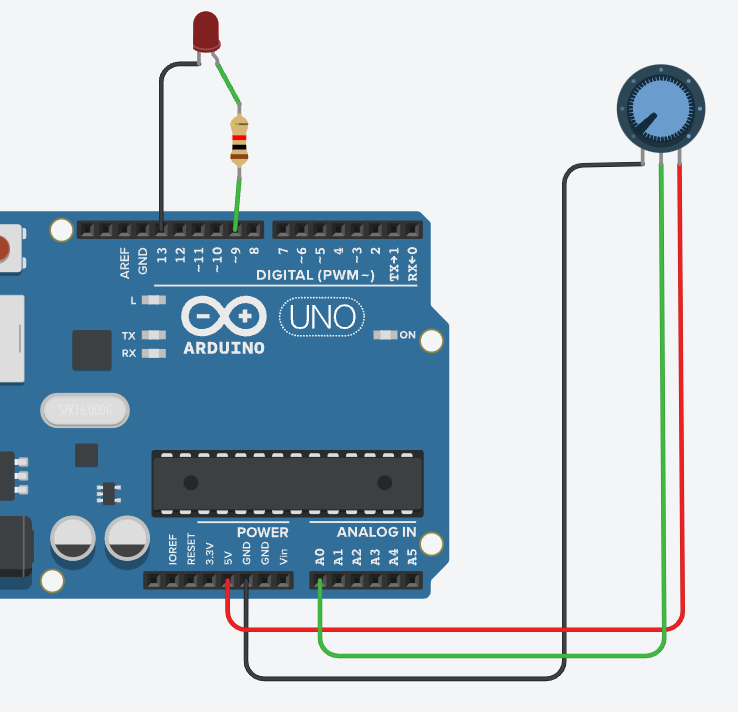
|  |
| --- |
| *void* ***setup****()*  *{*  *Serial.begin(9600);*  *}*  *void* ***loop****()*  *{*  *int giatri = analogRead(A0);*  *int nhietdo = map(giatri, 20, 358, -40, 125);*  *if(nhietdo > 37) digitalWrite(10, HIGH);*  *else digitalWrite(10, LOW);*  *delay(10);*  *}* |

# Bài 5. Điều khiển độ sáng của đèn bằng chiết áp

## Mô tả

## Hệ thống nhúng được thiết kế để cho phép điều khiển độ sáng của Led (pin 9) thông qua một biến trở (gắn ở chân A0)

## Sơ đồ thiết kế



Hình 9: Điều khiển độ sáng bằng chiết áp

## Đặc điểm của linh kiện

* Đèn LED
* Chiết Áp (Potentiometer)

## Code chương trình

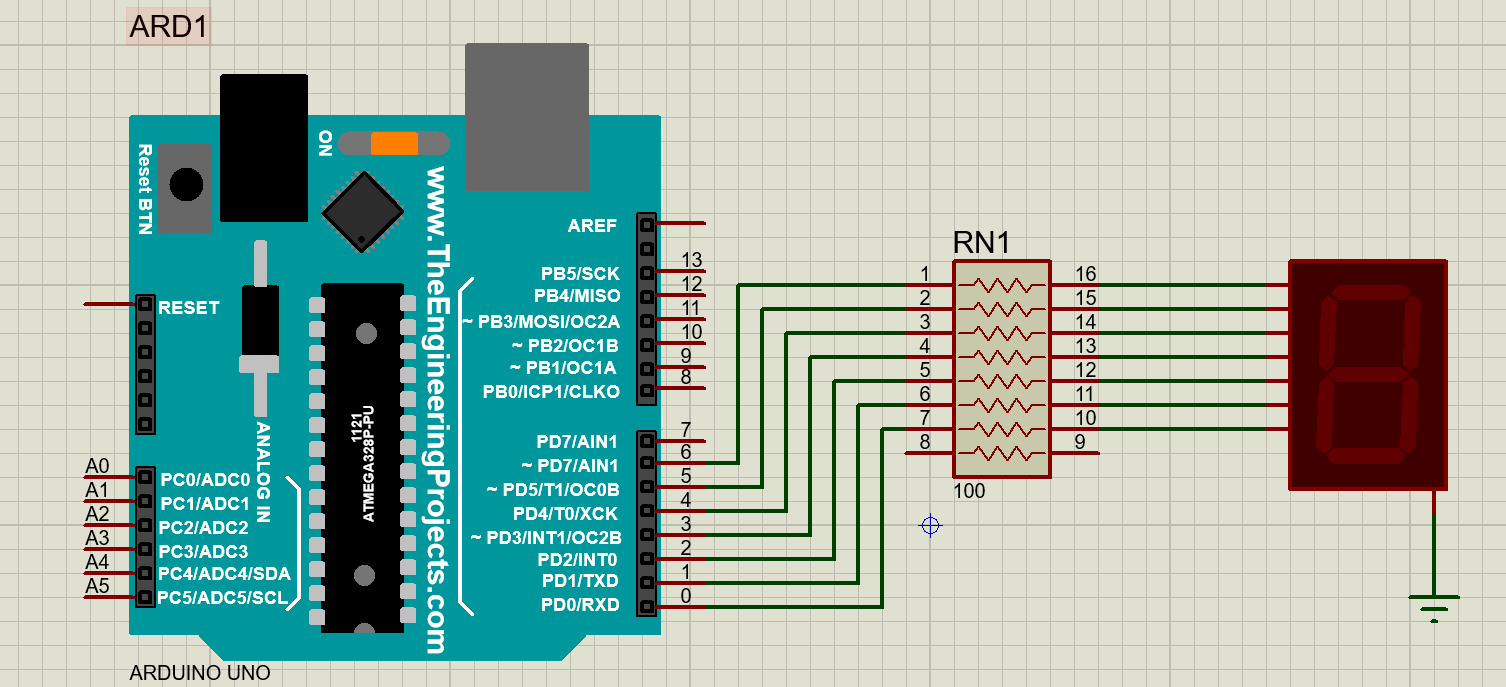
|  |
| --- |
| *int x = 0;*  *void* ***setup****()*  *{*  *pinMode(A0, INPUT);*  *pinMode(9, OUTPUT);*  *}*  *void* ***loop****()*  *{*  *x = analogRead(A0);*  *int do\_sang = map(x, 0, 1023, 0, 255);*  *analogWrite(9, do\_sang);*  *}* |

# Bài 6. Led 7 đoạn

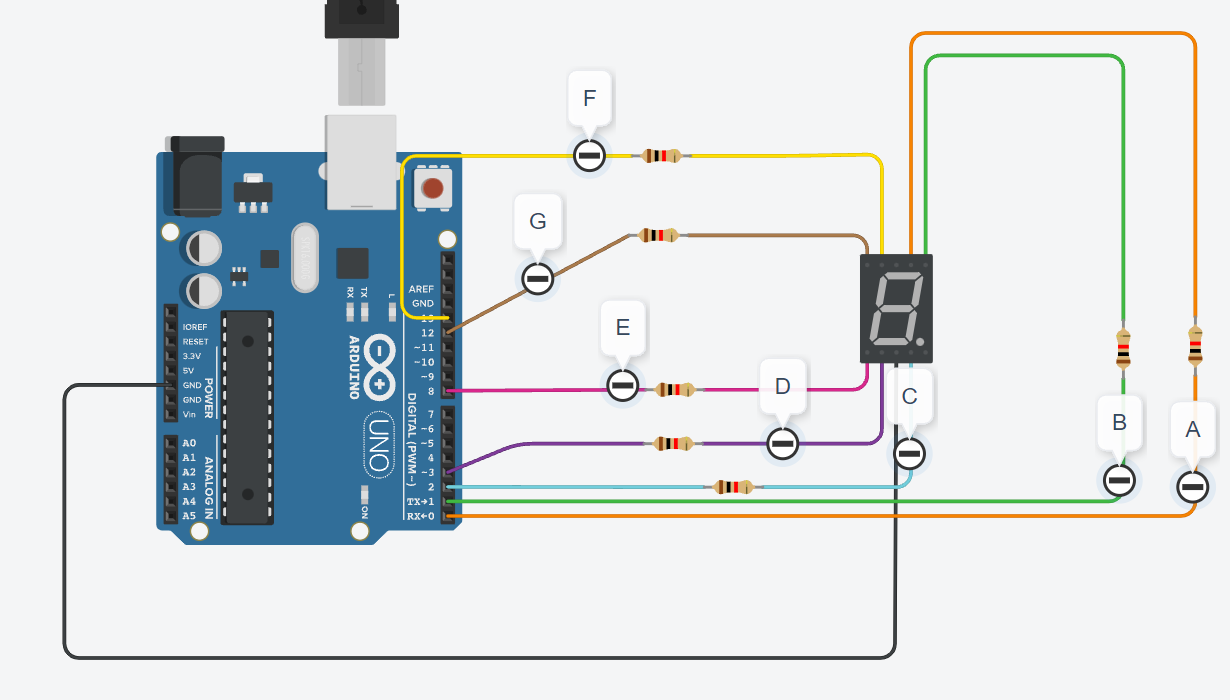
## Mô tả

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 vào cổng số 13 của board mạch Arduino.

## Sơ đồ thiết kế



Hình 10: Led 7 đoạn



Hình 11: Led 7 đoạn

## Đặc điểm của linh kiện

* 1 mạch Arduino Uno
* 7 Segment Display
* RES16DIPIS

## Code chương trình

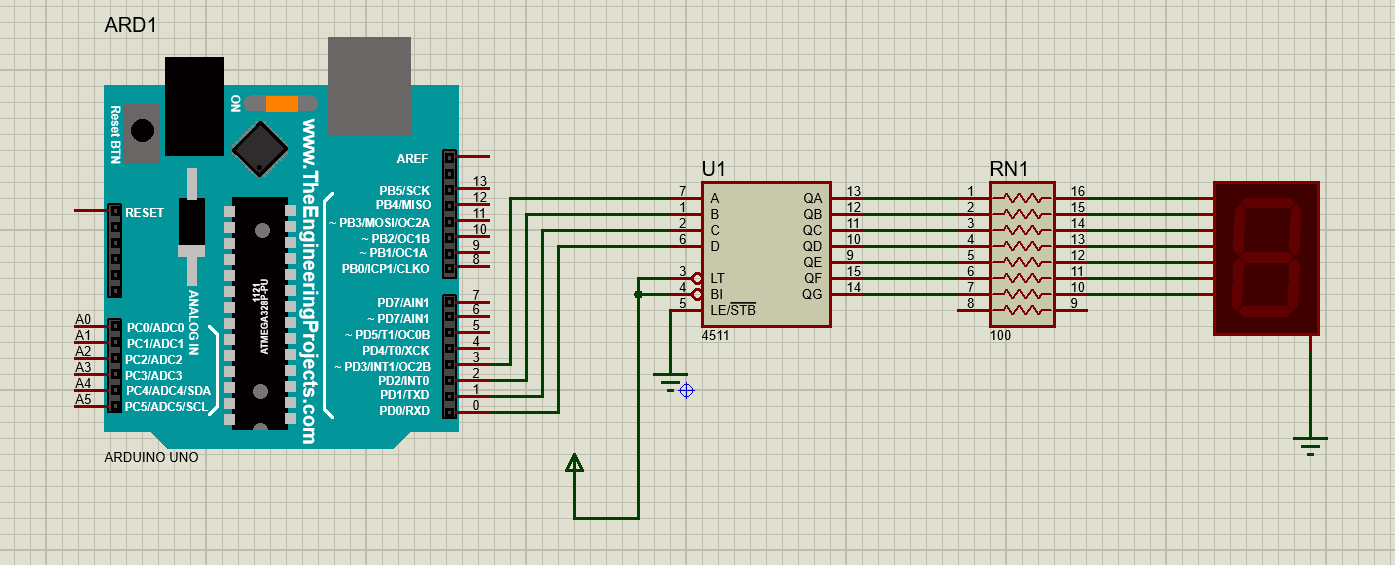
|  |
| --- |
| int a=6, b=5, c=4, d=3,e=2, f=1, g=0;  void **setup**()  {  pinMode(a, OUTPUT);  pinMode(b, OUTPUT);  pinMode(c, OUTPUT);  pinMode(d, OUTPUT);  pinMode(e, OUTPUT);  pinMode(f, OUTPUT);  pinMode(g, OUTPUT);  }  void **loop**()  {  digitalWrite(a, HIGH);  digitalWrite(b, HIGH);  digitalWrite(c, HIGH);  digitalWrite(d, HIGH);  digitalWrite(e, HIGH);  digitalWrite(f, HIGH);  digitalWrite(g, LOW);  delay(500);    digitalWrite(a, LOW);  digitalWrite(b, HIGH);  digitalWrite(c, HIGH);  digitalWrite(d, LOW);  digitalWrite(e, LOW);  digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  delay(500);    digitalWrite(a, HIGH);  digitalWrite(b, HIGH);  digitalWrite(c, HIGH);  digitalWrite(d, HIGH);  digitalWrite(e, HIGH);  digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  delay(500);  } |

# Bài 7. 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 cổng số 13 của board mạch Arduino.

## Sơ đồ thiết kế



Hình 12: Led 7 đoạn BCD

## Đặc điểm của linh kiện

* 1 mạch Arduino Uno
* 7 Segment Display
* RES16DIPIS
* Mạch 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);

delay(500);

}

void **BON**()

{

digitalWrite(a, LOW);

digitalWrite(b, HIGH);

digitalWrite(c, LOW);

digitalWrite(d, LOW);

}

void **TAM**(){

digitalWrite(a, HIGH);

digitalWrite(b, LOW);

digitalWrite(c, LOW);

digitalWrite(d, LOW);

}

void **loop**() {

**KHONG**(); delay(500);

**BON**(); delay(500);

**TAM**(); delay(500);

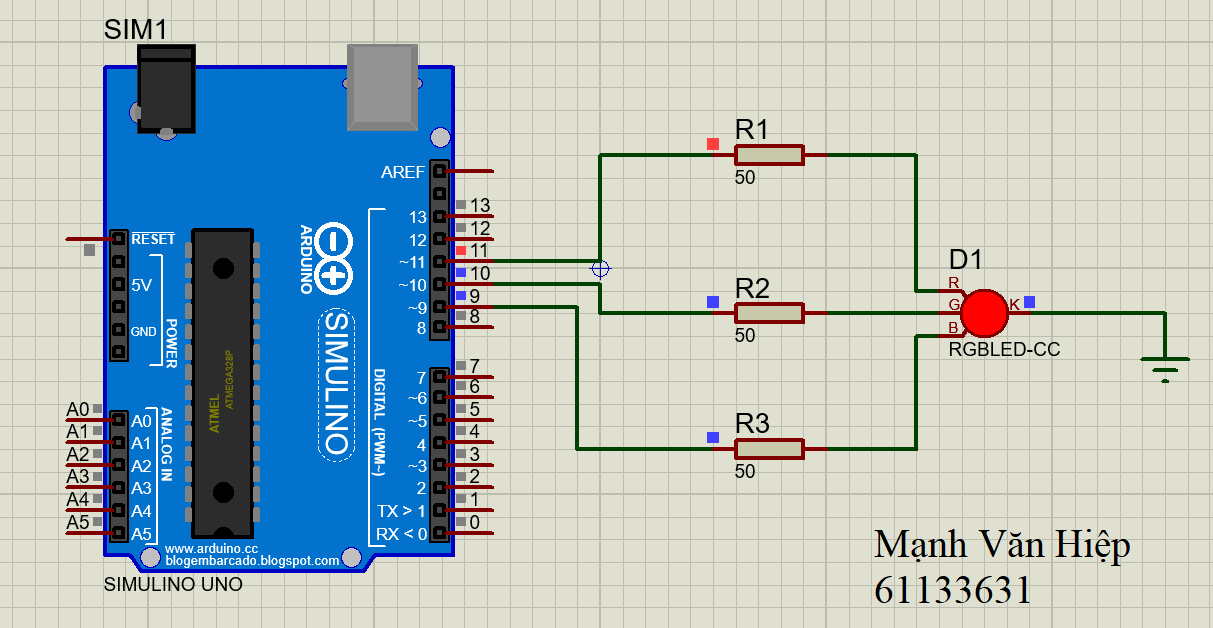
}

# Bài 8. Nháy đèn LED RGB

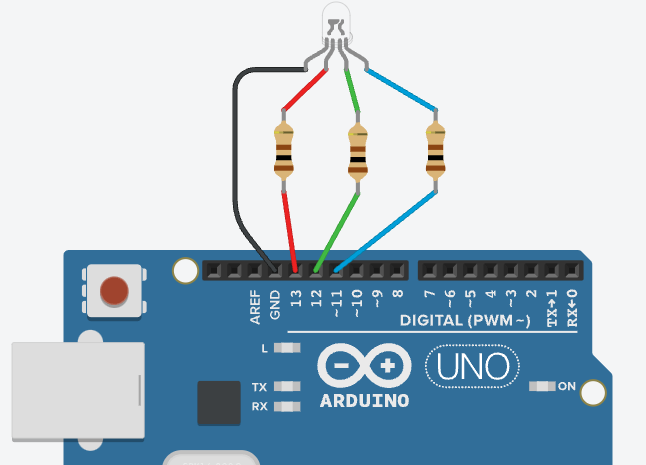
## Mô tả

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.

## Sơ đồ thiết kế

****

Hình 13: Nháy led RGB



Hình 14: Nháy led RGB

## Đặc điểm của linh kiện

* 3 điện trở
* 1 đèn RGB
* 1 mạch Arduino Uno

## Code chương trình

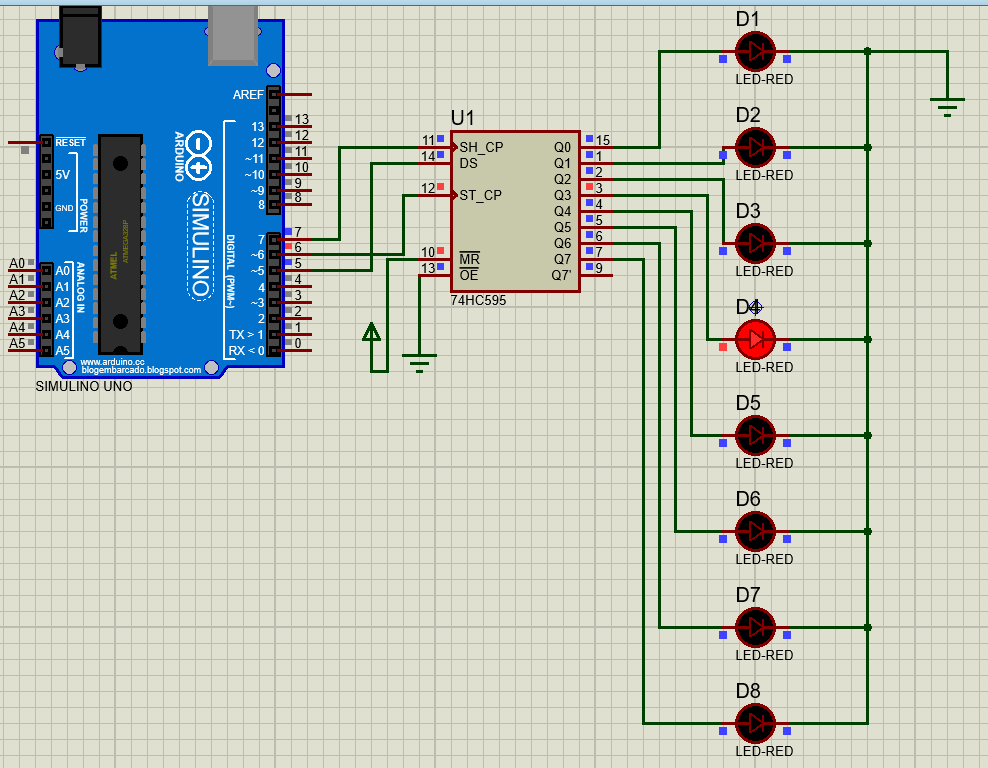
|  |
| --- |
| *int den\_do = 9;*  *int den\_xanh\_lam = 10;*  *int den\_xanh\_luc = 11;*  *void* ***setup****() {*  *pinMode(den\_do, OUTPUT);*  *pinMode(den\_xanh\_lam, OUTPUT);*  *pinMode(den\_xanh\_luc, OUTPUT);*  *}*  *void* ***loop****() {*  *analogWrite(den\_do, 255);*  *analogWrite(den\_xanh\_lam, 0);*  *analogWrite(den\_xanh\_luc, 0);*  *delay(500);*  *analogWrite(den\_do, 0);*  *analogWrite(den\_xanh\_lam, 255);*  *analogWrite(den\_xanh\_luc, 0);*  *delay(500);*  *analogWrite(den\_do, 0);*  *analogWrite(den\_xanh\_lam, 0);*  *analogWrite(den\_xanh\_luc, 255) ;*  *delay(500);*  *}* |

# Bài 9. Sáng 8 Led theo một trình tự, sử dụng 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: 8 Led IC 74HC545

## Đặc điểm của linh kiện

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

## Code chương trình

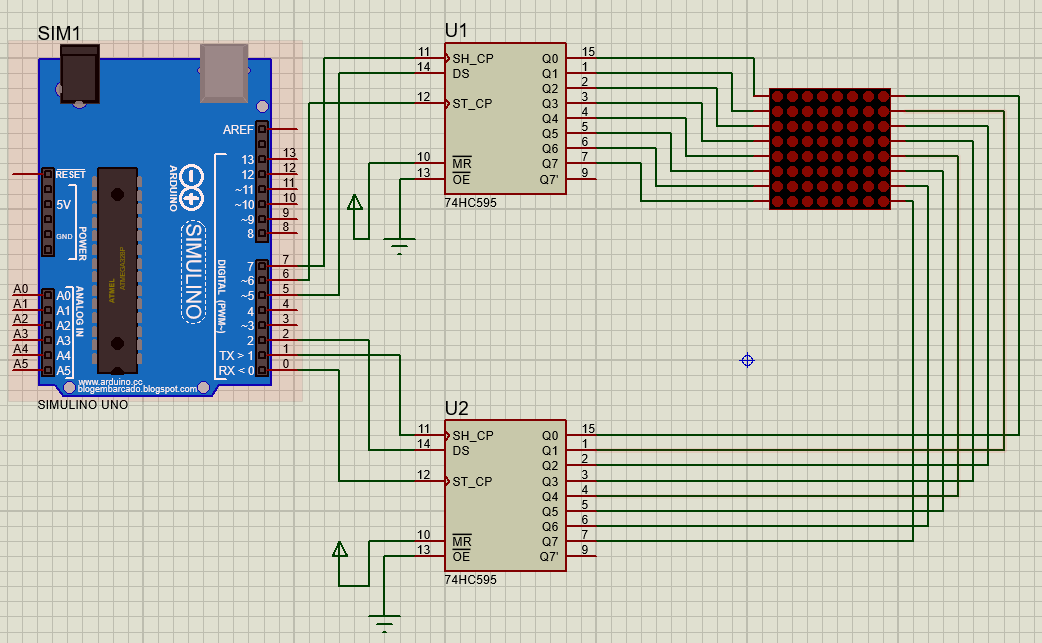
|  |
| --- |
| *#define \_clock 7*  *#define \_latch 6*  *#define \_data 5*  *void* ***setup****()*  *{*  *pinMode(\_data, OUTPUT);*  *pinMode(\_latch, OUTPUT);*  *pinMode(\_clock, OUTPUT);*  *}*  *void* ***loop****()*  *{*  *for(int i = 0; i < 256; i++)*  *{*  *digitalWrite(\_latch, LOW);*  *shiftOut(\_data,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

# Bài 10. Led Matrix 8x8

## Mô tả

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

## Sơ đồ thiết kế



Hình 16: Led Matrix 8x8

## Đặc điểm của linh kiện

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

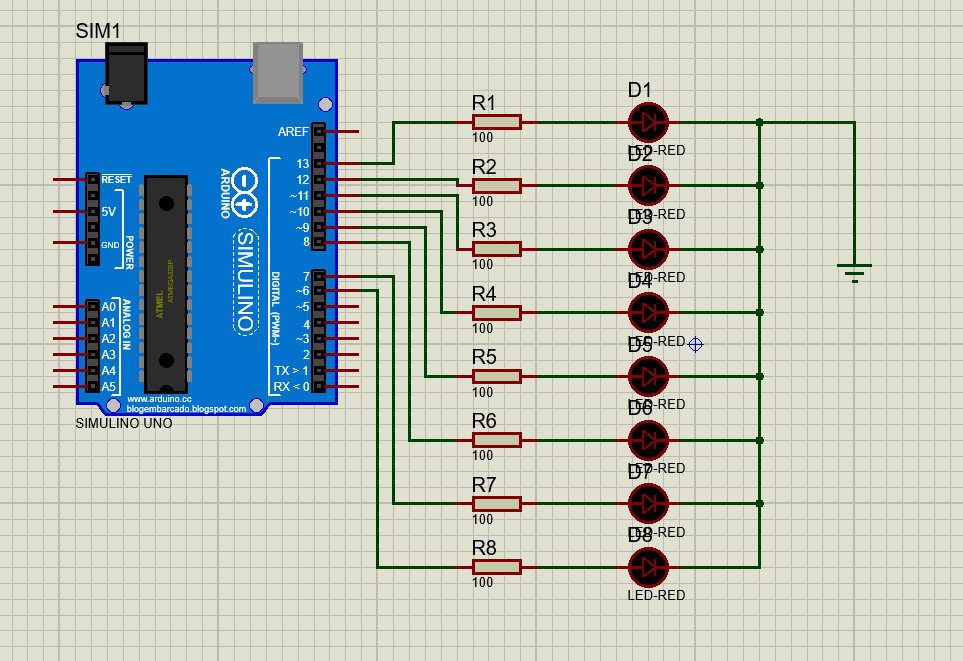
## Code chương trình

# Bài 11. Nháy Led theo yêu cầu

## Mô tả

Hệ thống sẽ thực hiện sáng tuần tự các cặp led (có 8 led); sau đó tắt đồng thời các led ở port lẻ và sáng đồng thời các led ở port chẵn; sau đó chớp 3 lần đồng thời 8 led.

## Sơ đồ thiết kế



Sáng 8 led theo yêu cầu

## Đặc điểm của linh kiện

* 1 mạch Arduino Uno
* 8 Led red
* 8 điện trở 100 Ω

## Code chương trình

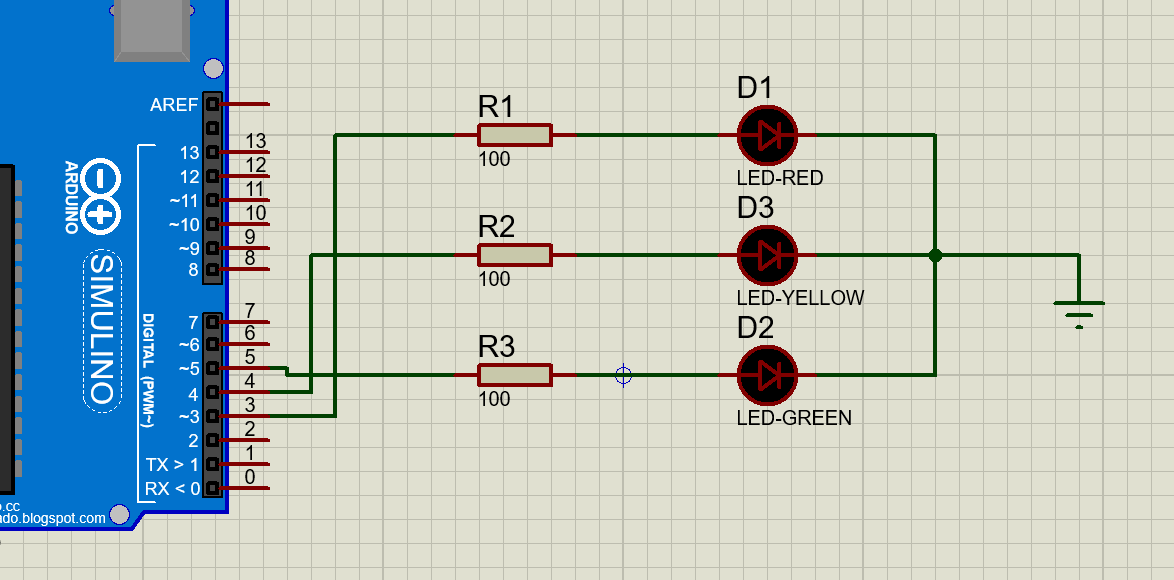
|  |
| --- |
| *byte ledPin[] = {6, 7, 8, 9, 10, 11, 12, 13};*  *void* ***setup****() {*  *for (int x = 0; x < 8; x++) {*  *pinMode(ledPin[x], OUTPUT);*  *}*  *}*  *void* ***loop****() {*  *digitalWrite(13, LOW);*  *digitalWrite(12, LOW);*  *digitalWrite(11, LOW);*  *digitalWrite(10, LOW);*  *digitalWrite(9, LOW);*  *digitalWrite(8, LOW);*  *digitalWrite(7, LOW);*  *digitalWrite(6, LOW);*  *//nhay 2 den sang tuan tu*  *for (int x = 0; x < 8; x += 2) {*  *digitalWrite(ledPin[x], HIGH);*  *digitalWrite(ledPin[x + 1], HIGH);*  *delay(500);*  *}*  *delay(1000);*  *//tat den port le*  *for (int x = 0; x < 8; x += 2) {*  *digitalWrite(ledPin[x], LOW);*  *}*  *delay(1000);*  *//sang tat 3 lan*  *for (int y = 0; y < 3; y++) {*  *digitalWrite(13, LOW);*  *digitalWrite(12, LOW);*  *digitalWrite(11, LOW);*  *digitalWrite(10, LOW);*  *digitalWrite(9, LOW);*  *digitalWrite(8, LOW);*  *digitalWrite(7, LOW);*  *digitalWrite(6, LOW);*  *delay(1000);*  *digitalWrite(13, HIGH);*  *digitalWrite(12, HIGH);*  *digitalWrite(11, HIGH);*  *digitalWrite(10, HIGH);*  *digitalWrite(9, HIGH);*  *digitalWrite(8, HIGH);*  *digitalWrite(7, HIGH);*  *digitalWrite(6, HIGH);*  *delay(1000);*  *}*  *}* |

# Bài 12. 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 25s gắng vào mạch Arduino ở cổng D3, đèn vàng sáng 10s ở cổng D4, đèn xanh sáng 30s ở cổng D5.

## Sơ đồ thiết kế



Nháy Led đèn giao thông

## Đặc điểm của linh kiện

* 1 mạch Arduino Uno
* 1 Led red, 1 Led yellow, 1 Led green
* 3 điện trở 100 Ω

## Code chương trình

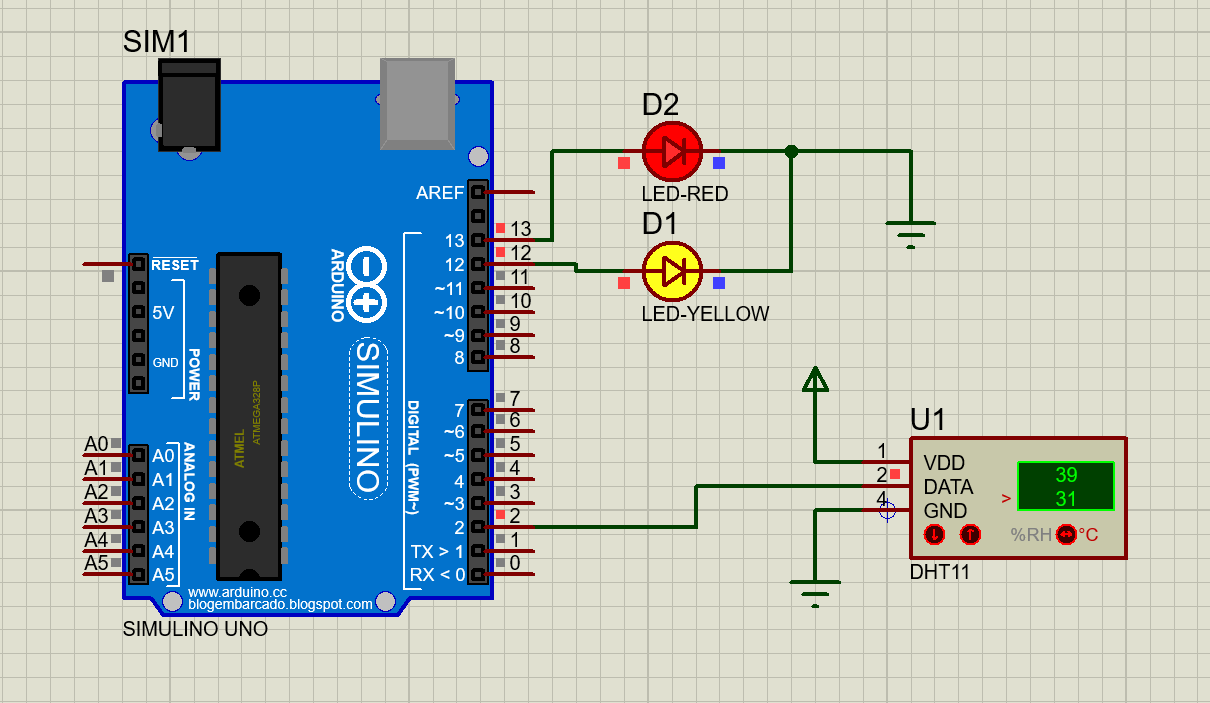
|  |
| --- |
| *void* ***setup****() {*  *pinMode(3, OUTPUT);*  *pinMode(4, OUTPUT);*  *pinMode(5, OUTPUT);*  *}*  *void* ***loop****() {*  *digitalWrite(3, HIGH);*  *delay(25000);*  *digitalWrite(3, LOW);*  *digitalWrite(4, HIGH);*  *delay(10000);*  *digitalWrite(4, LOW);*  *digitalWrite(5, HIGH);*  *delay(30000);*  *digitalWrite(5, LOW);*  *}* |

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

## Mô tả

Hệ thống thiết kế cảnh báo thông qua Led đỏ nếu nhiệt độ quá 300C và Led vàng nếu độ ẩm dưới 40%.

## Sơ đồ thiết kế



Cảnh báo nhiệt độ và độ ẩm thông qua bật/tắt Led

## Đặc điểm của linh kiện

* 1 mạch Arduino Uno
* 1 Led red, 1 Led Yellow
* 1 DHT11. DHT11 là một cảm biến kỹ thuật số giá rẻ để cảm nhận nhiệt độ và độ ẩm. Cảm biến này có thể dễ dàng giao tiếp với bất kỳ bộ vi điều khiển vi nào như Arduino, Raspberry Pi, ... để đo độ ẩm và nhiệt độ ngay lập tức. DHT11 là một cảm biến độ ẩm tương đối. Để đo không khí xung quanh, cảm biến này sử dụng một điện trở nhiệt và một cảm biến độ ẩm điện dung.

## Code chương trình

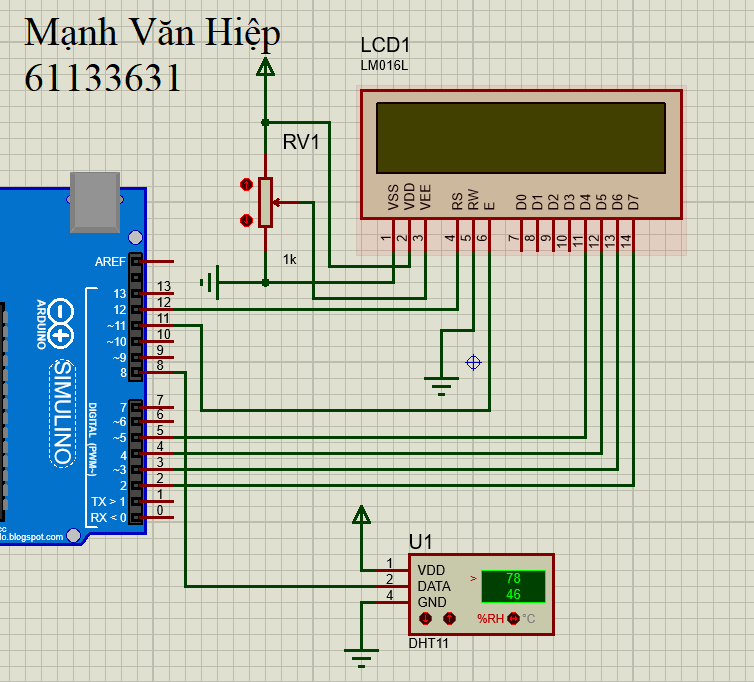
|  |
| --- |
| *#include "DHT.h"*  *const int DHTPIN = 2;*  *const int DHTTYPE = DHT11;*  *DHT dht(DHTPIN, DHTTYPE);*  *void* ***setup****() {*  *pinMode(13, OUTPUT);*  *pinMode(12, OUTPUT);*  *dht.begin();*  *}*  *void* ***loop****() {*  *float doam = dht.readHumidity();*  *float nhietdo = dht.readTemperature();*    *if (nhietdo > 30)*  *digitalWrite(13, HIGH);*  *else*  *digitalWrite(13, LOW);*    *if (doam < 40)*  *digitalWrite(12, HIGH);*    *else*  *digitalWrite(12, LOW);*  *}* |

# Bài 14. Liên tục đọc nhiệt độ môi trường và hiện ra LCD sau mỗi giây

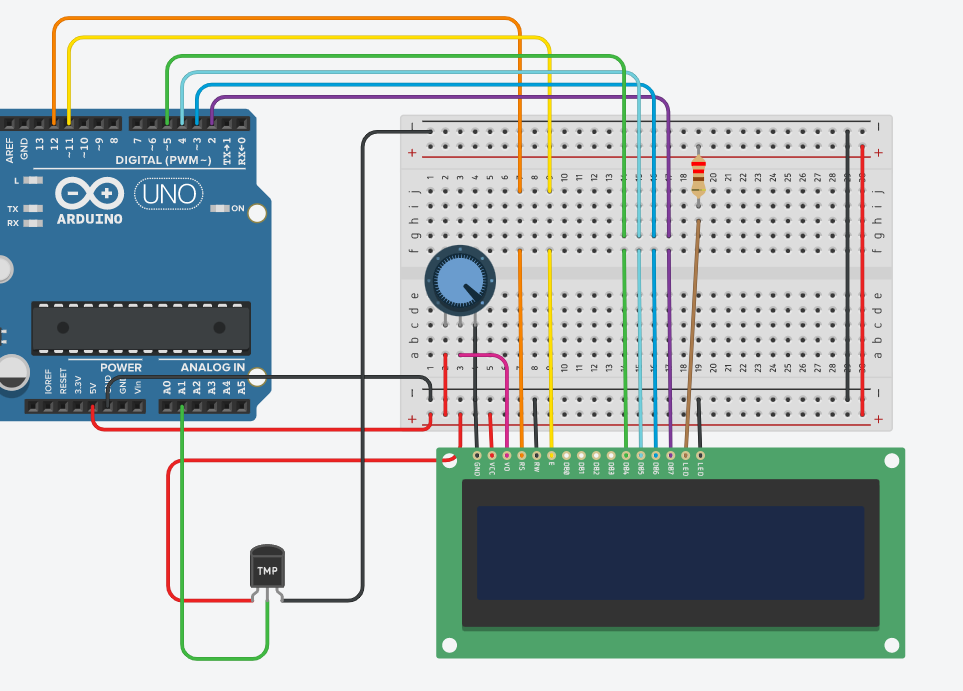
## Mô tả

Hệ thống được thiết kế thực hiện việc đọc nhiệt độ từ môi trường và hiện thị lên màn hình LCD.

## Sơ đồ thiết kế



Hình 1. Sơ đồ mạch trên proteus



Hình 2. Sơ đồ mạch trên tinkercard

## Đặc điểm của linh kiện

* 1 mạch Arduino Uno
* 1 chiết áp
* 1 DHT11

DHT11 Là cảm biến nhiệt độ, 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.

* 1 màn hình LCD 16 x 2

**LCD 16×2** được sử dụng để hiển thị trạng thái hoặc các thông số.

* LCD 16×2 có 16 chân trong đó 8 chân dữ liệu (D0 – D7) và 3 chân điều khiển (RS, RW, EN).
* 5 chân còn lại dùng để cấp nguồn và đèn nền cho LCD 16×2.
* Các chân điều khiển giúp ta dễ dàng cấu hình LCD ở chế độ lệnh hoặc chế độ dữ liệu.
* Chúng còn giúp ta cấu hình ở chế độ đọc hoặc ghi.

LCD 16×2 có thể sử dụng ở chế độ 4 bit hoặc 8 bit tùy theo ứng dụng ta đang làm.

## Code chương trình

* Code trong proteus

|  |
| --- |
| // Trước khi làm các bạn nhớ cài thư viện DHT.h và LiquidCrystal.h trong arduino  #include "DHT.h"  #include <LiquidCrystal.h>  const int DHTPIN = 8;  const int DHTTYPE = DHT11;  DHT dht(DHTPIN, DHTTYPE);  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  void **setup**() {  dht.begin();  }  void **loop**() {  float nhietdo = dht.readTemperature();  lcd.begin(16, 2);  lcd.print("Nhiet do: ");  lcd.print(nhietdo);  lcd.setCursor(0,1);  } |

* Code trong tinkercard

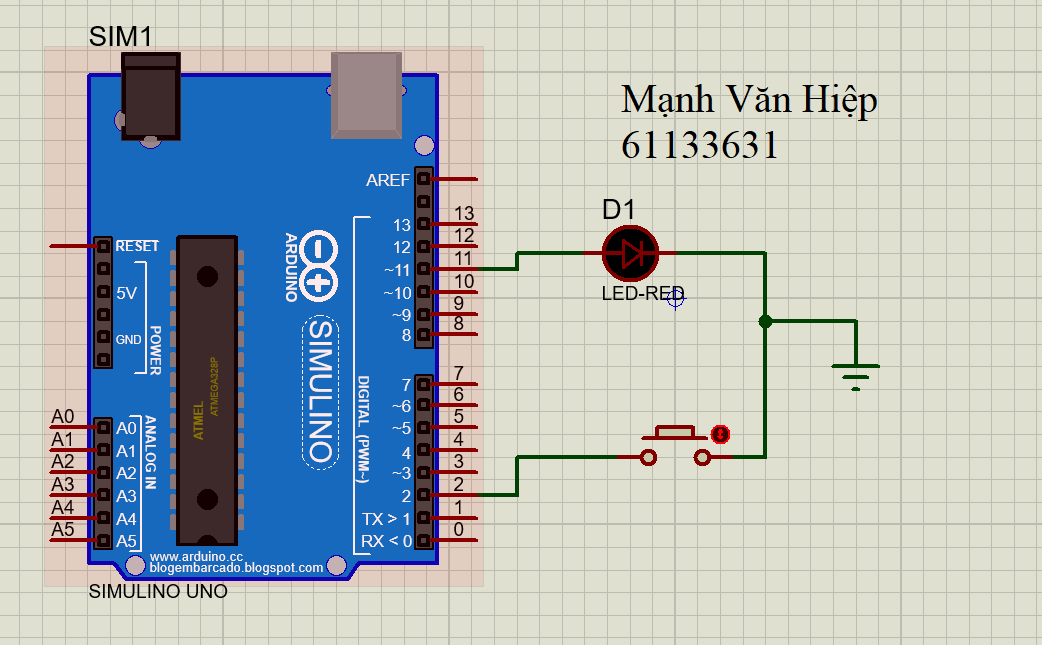
|  |
| --- |
| #include <LiquidCrystal.h>  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  void **setup**() {  lcd.begin(16, 2);  lcd.setCursor(3,0);  lcd.print("Nhiet do: ");  }  void **loop**() {  lcd.setCursor(0, 1);  int doC = analogRead(A1);  int doc = map(doC, 20, 358, -40, 125);  lcd.print(doc);  } |

# Bài 15. Lặp trình ngắt để có ngắt xảy ra (do một nút bấm) thì bật đèn sáng và ngược lại

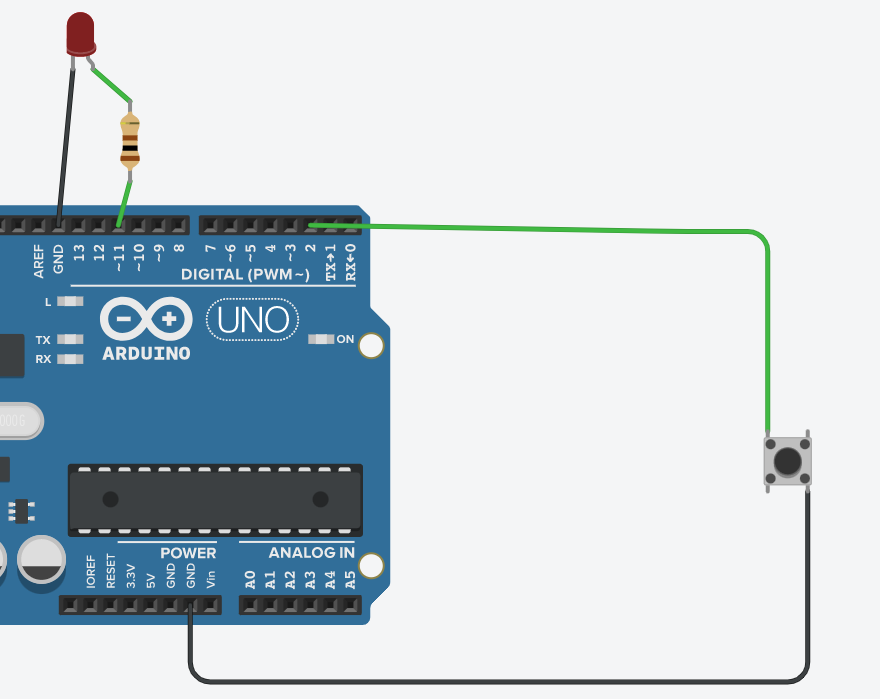
## Mô tả

Hệ thống được thiết kế thực hiện ngắt xảy ra do nút bấm thì đèn bật sáng và ngược lại.

## Sơ đồ thiết kế



Hình 3. Sơ đồ mạch trên Proteus



Hình 4. Sơ đồ mạch trên 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

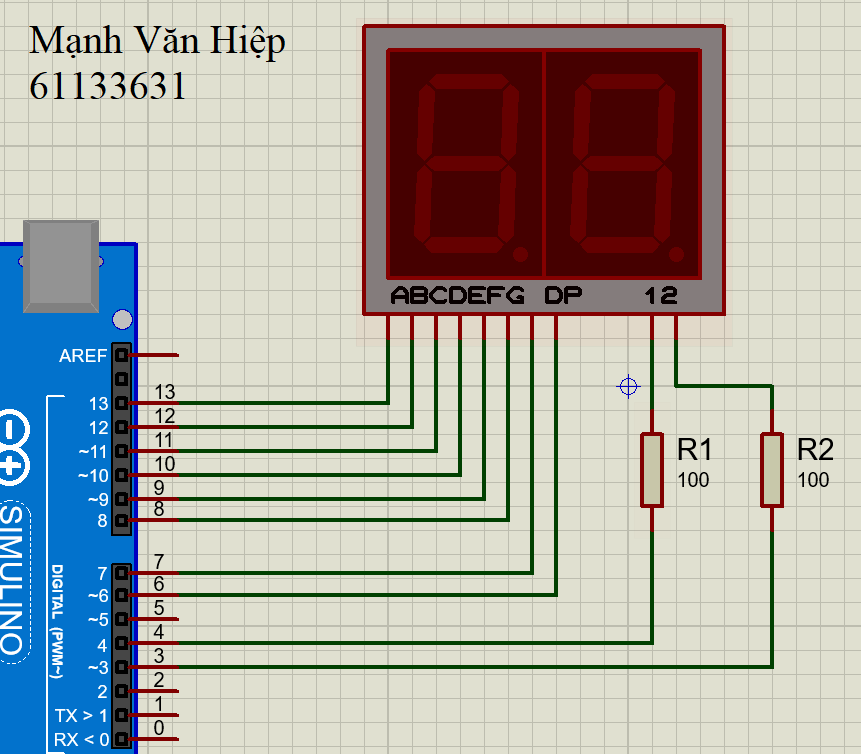
|  |
| --- |
| int led = 11;  int trangthai = LOW;  void **setup**()  {  pinMode(led, OUTPUT);  digitalWrite(2, HIGH);  attachInterrupt(0, tatled, CHANGE);    }  void **loop**()  {  digitalWrite(led, trangthai);  }  void **tatled**()  {  trangthai = !trangthai;  } |

# Bài 16. Led 7 đoạn hiện các số đếm từ 0 – 9 và từ 00 – 99

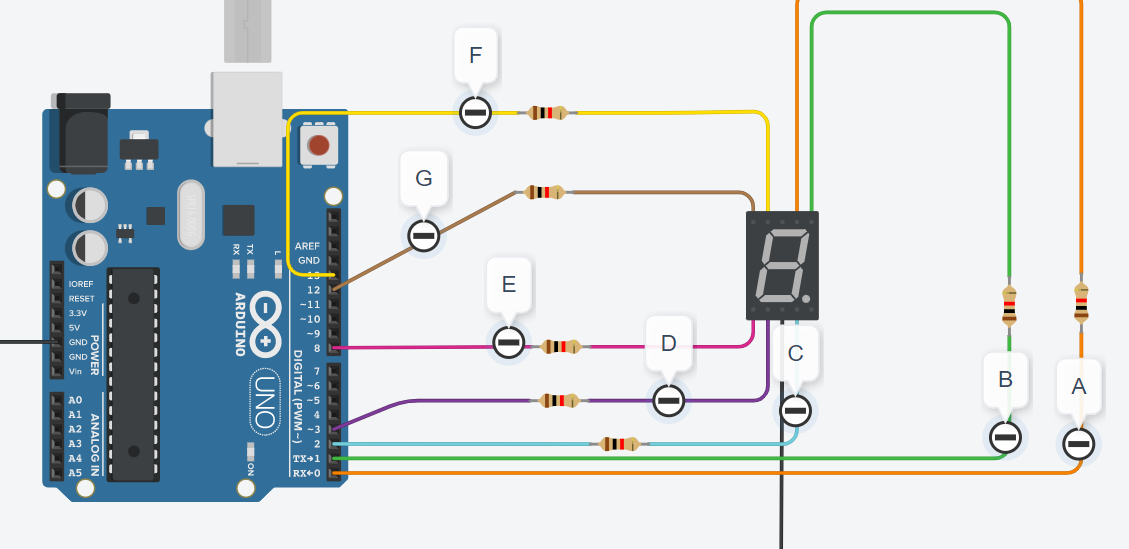
## Mô tả

Hệ thống được thiết kế thực hiện đếm số từ 0 – 9 và từ 00 – 99 bằng Led 7 đoạn .

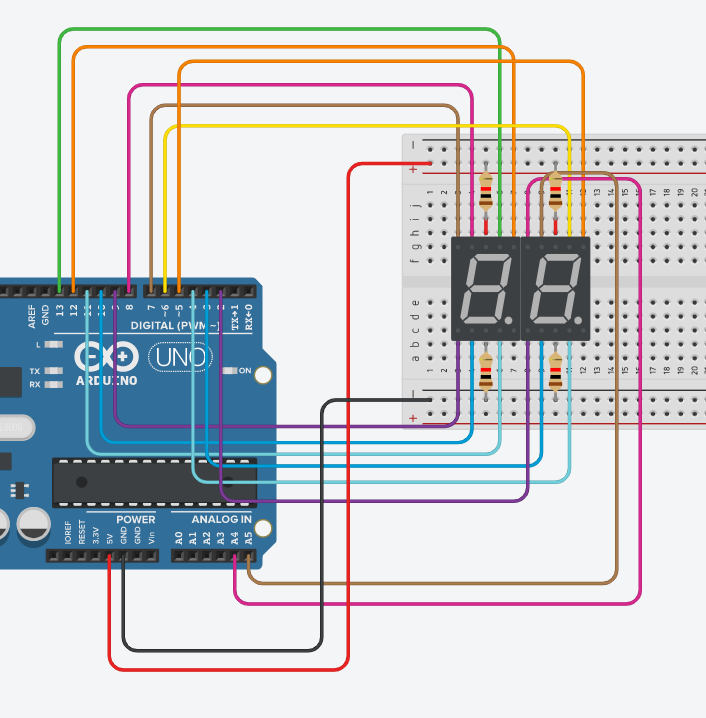
## Sơ đồ thiết kế



Hình 5. Sơ đồ mạch trên Proteus



Hình 6 Sơ đồ trên tinkercard 0 – 9



Hình 7. Sơ đồ mạch Tinkercad 00 – 99

## Đặc điểm của linh kiện

* Đối với proteus
* 1 mạch Arduino
* 7 SEG – MPX2 – CC (LED 7 thanh đôi cathode chung)
* 2 điện trở: 100 Ωnhằm hạn chế cường độ dòng điện qua đèn, giúp đèn sáng an toàn hơn.
* Đối với Arduino (0 – 9)
* 1 mạch Arduino
* 7 điện trở: 100 Ωnhằm hạn chế cường độ dòng điện qua đèn, giúp đèn sáng an toàn hơn.
* 1 7 Segment Display
* Đối với Arduino (00 – 99)
* 1 mạch Arduino
* 4 điện trở: 100 Ωnhằm hạn chế cường độ dòng điện qua đèn, giúp đèn sáng an toàn hơn.
* 2 7 Segment Display

## Code chương trình

* Code trong proteus

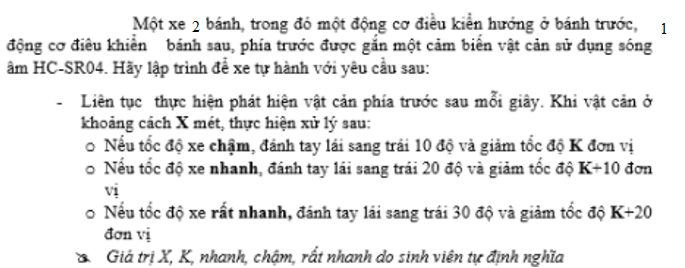
|  |
| --- |
| **// Hiển thị số từ 0 – 9**  int seg1 = 4;  int seg2 = 3;  int i = 0;  int j = 0;  int n = 0;  int den[7] = {7, 8, 9, 10, 11, 12, 13};  int matran[10][7] = {{0, 1, 1, 1, 1, 1, 1}, // 0  {0, 0, 0, 0, 1, 1, 0}, // 1  {1, 0, 1, 1, 0, 1, 1}, // 2  {1, 0, 0, 1, 1, 1, 1}, // 3  {1, 1, 0, 0, 1, 1, 0}, // 4  {1, 1, 0, 1, 1, 0, 1}, // 5  {1, 1, 1, 1, 1, 0, 1}, // 6  {0, 0, 0, 0, 1, 1, 1}, // 7  {1, 1, 1, 1, 1, 1, 1}, // 8  {1, 1, 0, 1, 1, 1, 1}, // 9  };  void **setup**() {  pinMode(13, OUTPUT);  pinMode(12, OUTPUT);  pinMode(11, OUTPUT);  pinMode(10, OUTPUT);  pinMode(9, OUTPUT);  pinMode(8, OUTPUT);  pinMode(7, OUTPUT);  pinMode(4, OUTPUT);  pinMode(3, OUTPUT);  }  void **loop**() {  for (int i = 0; i < 10; i++) {  digitalWrite(seg1, 1);  digitalWrite(seg2, 0);  for (j = 0; j < 7; j++)  digitalWrite(den[j], matran[i][j]);  delay(100);  }  }  **// Hiển thị số từ 00 – 99**  int seg1 = 4;  int seg2 = 3;  int i = 0;  int j = 0;  int n = 0;  int den[7] = {7, 8, 9, 10, 11, 12, 13};  int matran[10][7] = {{0, 1, 1, 1, 1, 1, 1}, // 0  {0, 0, 0, 0, 1, 1, 0}, // 1  {1, 0, 1, 1, 0, 1, 1}, // 2  {1, 0, 0, 1, 1, 1, 1}, // 3  {1, 1, 0, 0, 1, 1, 0}, // 4  {1, 1, 0, 1, 1, 0, 1}, // 5  {1, 1, 1, 1, 1, 0, 1}, // 6  {0, 0, 0, 0, 1, 1, 1}, // 7  {1, 1, 1, 1, 1, 1, 1}, // 8  {1, 1, 0, 1, 1, 1, 1}, // 9  };  void **setup**() {  pinMode(13, OUTPUT);  pinMode(12, OUTPUT);  pinMode(11, OUTPUT);  pinMode(10, OUTPUT);  pinMode(9, OUTPUT);  pinMode(8, OUTPUT);  pinMode(7, OUTPUT);  pinMode(4, OUTPUT);  pinMode(3, OUTPUT);  }  void **loop**() {  for (int i = 0; i < 10; i++) {  digitalWrite(seg1, 0);  digitalWrite(seg2, 1);  for (j = 0; j < 7; j++)  digitalWrite(den[j], matran[n][j]);  delay(50);  digitalWrite(seg1, 1);  digitalWrite(seg2, 0);  for (j = 0; j < 7; j++)  digitalWrite(den[j], matran[i][j]);  delay(50);  if (i == 9) {  n++;  if (n > 9)  n = 0;  }  }  } |

* Code trong tinkercard

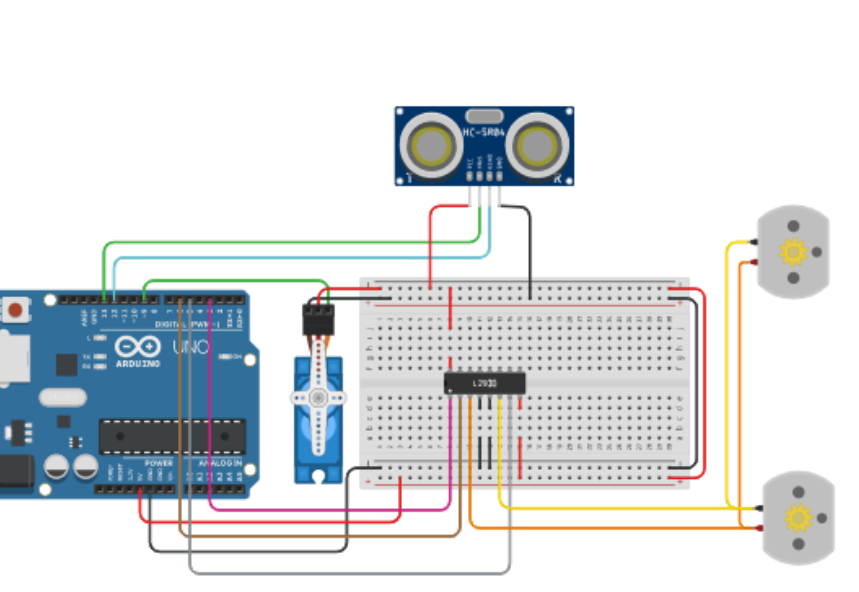
|  |
| --- |
| **// Hiển thị số từ 0 – 9**  int a=0, b=1, c=2, d=3,e=8, f=13, g=12;  void **setup**()  {  pinMode(a, OUTPUT);  pinMode(b, OUTPUT);  pinMode(c, OUTPUT);  pinMode(d, OUTPUT);  pinMode(e, OUTPUT);  pinMode(f, OUTPUT);  pinMode(g, OUTPUT);    }  void **loop**()  {  digitalWrite(a, HIGH);  digitalWrite(b, HIGH);  digitalWrite(c, HIGH);  digitalWrite(d, HIGH);  digitalWrite(e, HIGH);  digitalWrite(f, HIGH);  digitalWrite(g, LOW);  delay(500);    digitalWrite(a, LOW);  digitalWrite(b, HIGH);  digitalWrite(c, HIGH);  digitalWrite(d, LOW);  digitalWrite(e, LOW);  digitalWrite(f, LOW);  digitalWrite(g, LOW);  delay(500);    digitalWrite(a, HIGH);  digitalWrite(b, HIGH);  digitalWrite(c, LOW);  digitalWrite(d, HIGH);  digitalWrite(e, HIGH);  digitalWrite(f, LOW);  digitalWrite(g, HIGH);  delay(500);    digitalWrite(a, HIGH);  digitalWrite(b, HIGH);  digitalWrite(c, HIGH);  digitalWrite(d, HIGH);  digitalWrite(e, LOW);  digitalWrite(f, LOW);  digitalWrite(g, HIGH);  delay(500);    digitalWrite(a, LOW);  digitalWrite(b, HIGH);  digitalWrite(c, HIGH);  digitalWrite(d, LOW);  digitalWrite(e, LOW);  digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  delay(500);    digitalWrite(a, HIGH);  digitalWrite(b, LOW);  digitalWrite(c, HIGH);  digitalWrite(d, HIGH);  digitalWrite(e, LOW);  digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  delay(500);    digitalWrite(a, HIGH);  digitalWrite(b, LOW);  digitalWrite(c, HIGH);  digitalWrite(d, HIGH);  digitalWrite(e, HIGH);  digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  delay(500);    digitalWrite(a, HIGH);  digitalWrite(b, HIGH);  digitalWrite(c, HIGH);  digitalWrite(d, LOW);  digitalWrite(e, LOW);  digitalWrite(f, LOW);  digitalWrite(g, LOW);  delay(500);    digitalWrite(a, HIGH);  digitalWrite(b, HIGH);  digitalWrite(c, HIGH);  digitalWrite(d, HIGH);  digitalWrite(e, HIGH);  digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  delay(500);    digitalWrite(a, HIGH);  digitalWrite(b, HIGH);  digitalWrite(c, HIGH);  digitalWrite(d, HIGH);  digitalWrite(e, LOW);  digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  delay(500);  }  **// Hiện thị số từ 00 – 99**  int seg1[] = {7, 8, 9, 10, 11, 12, 13};  int seg2[] = {A4, A5, 2, 3, 4, 5, 6};  int n = 0;  int i = 0;  int j = 0;  // number matrix  int matran[10][7] = {  {0, 1, 1, 1, 1, 1, 1}, // 0  {0, 0, 0, 0, 1, 1, 0}, // 1  {1, 0, 1, 1, 0, 1, 1}, // 2  {1, 0, 0, 1, 1, 1, 1}, // 3  {1, 1, 0, 0, 1, 1, 0}, // 4  {1, 1, 0, 1, 1, 0, 1}, // 5  {1, 1, 1, 1, 1, 0, 1}, // 6  {0, 0, 0, 0, 1, 1, 1}, // 7  {1, 1, 1, 1, 1, 1, 1}, // 8  {1, 1, 0, 1, 1, 1, 1}, // 9  };  void **setup**() {  for(byte i = 0; i < 7; i++){  pinMode(seg1[i], OUTPUT);  pinMode(seg2[i], OUTPUT);  }  }  void **loop**() {    for (int i = 0; i < 10; i++) {  for (j = 0; j < 7; j++)  digitalWrite(seg1[j], matran[n][j]);  delay(100);  for (j = 0; j < 7; j++)  digitalWrite(seg2[j], matran[i][j]);  delay(100);  if (i == 9) {  n++;  if (n > 9)  n = 0;  }  }  } |

# Bài 17. Điều khiển động cơ 2 bánh

## Mô tả



## Sơ đồ thiết kế



Sơ đồ mạch trên tinkercard

## Đặc điểm của linh kiện

* 1 mạch Arduino Uno
* Ultrasonic Distance Sensor: thiết bị đo khoảng cách của các vật thể bằng cách phát ra một sóng siêu âm và chuyển đổi âm thanh phản xạ thành tín hiệu điện.
* 2 DC Motor: động cơ điện hoạt động với dòng điện một chiều.
* Micro Servo: là một thiết bị tự động có cảm biến phản hồi để điều chỉnh hành động. Servo là một bộ phận của hệ thống điều khiển, cung cấp lực chuyển động cần thiết cho các thiết bị máy móc khi vận hành.
* L293D: là một module mở rộng chuyên dụng cho các ứng dụng điều khiển động cơ, robot. Driver Motor Shield L293D có nhiều chức năng, dễ vận hành và có thư viện hỗ trợ. Đây là một driver phổ biến cho điều khiển động cơ DC, sử dụng IC điều khiển động cơ DC L293D.

## Code chương trình

* Code trong tinkercard

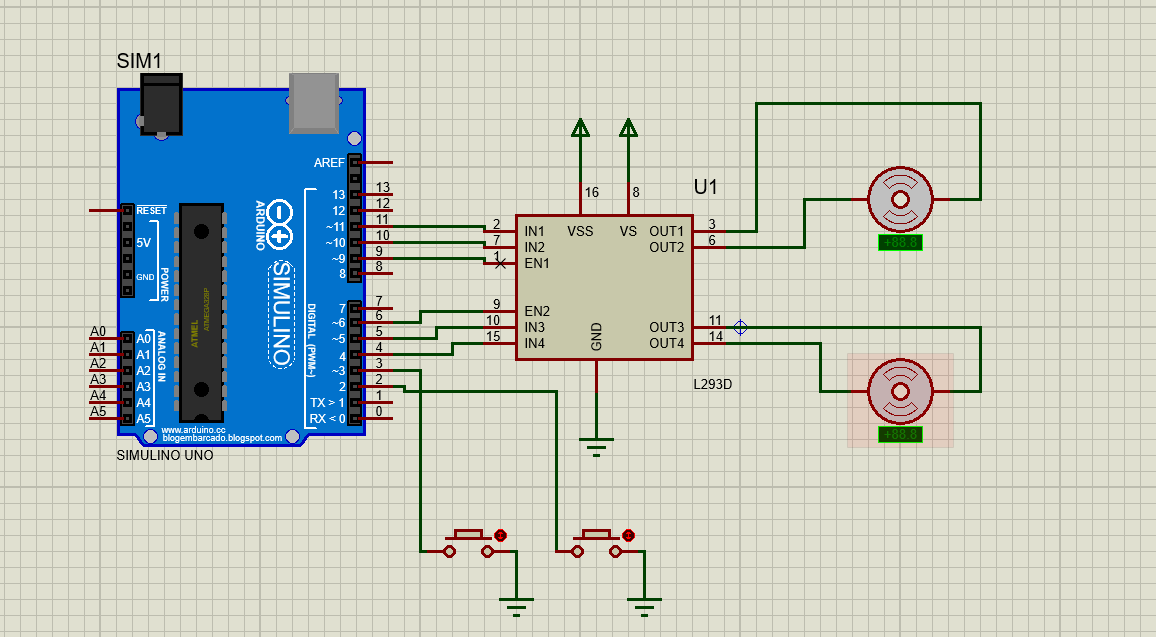
|  |
| --- |
| *#include <Servo.h>*  *int x = 50;//khoang cach*  *int k = 10;//giam k toc do*  *int tocdo = 25;//quy dinh toc do nhanh, cham, rat nhanh*  *Servo servo\_9;*  ***// động cơ***  *int enA = 3;//cham bâm xung*  *int in\_a1 = 6;*  *int in\_a2 = 5;*  ***//cảm biến***  *int trig = 13;*  *int echo = 12; //chân bâm xung*  *int khoangcach;*  *int time;*  *void* ***setup****()*  *{*  *Serial.begin(9600);*    *servo\_9.attach(9, 500, 2500);*    *pinMode(enA, OUTPUT); // chan dk xung*  *pinMode(in\_a1, OUTPUT);*  *pinMode(in\_a2, OUTPUT);*    *pinMode(trig,OUTPUT);*  *pinMode(echo,INPUT);*  *}*  *void* ***loop****()*  *{*  ***/\* Phát xung từ chân trig \*/***  *digitalWrite(trig, HIGH); // Bật chân trig*  *digitalWrite(trig, LOW); // tắt chân trig*  *delayMicroseconds(5);*    *time = pulseIn(echo, HIGH);*  *khoangcach = int(time/2/29.412);*    ***// forwork A***  *digitalWrite(in\_a1, HIGH);*  *digitalWrite(in\_a2, LOW);*  *Serial.println(khoangcach);*    *analogWrite(enA, tocdo);*  *if(tocdo == 25 && khoangcach == x){****//cham***  *servo\_9.write(10);*  *analogWrite(enA, tocdo - k);*  *delay(10);*  *} else if(tocdo == 60 && khoangcach == x){****//nhanh***  *servo\_9.write(20);*  *analogWrite(enA, tocdo -k - 10);*  *delay(10);*  *} else if(tocdo == 80 && khoangcach == x){****//rat nhanh***  *servo\_9.write(30);*  *analogWrite(enA, tocdo - k - 20);*  *delay(10);*  *}else {*  *servo\_9.write(0);*  *}*  *}* |

# Bài 18. Điều khiển động cơ DC với IC L293D

## 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

## Sơ đồ thiết kế



Sơ đồ mạch trên proteus

## Đặc điểm của linh kiện

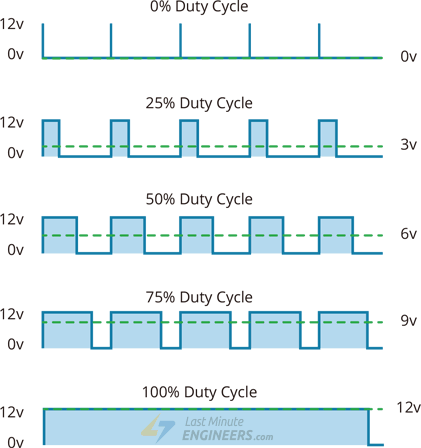
* DC
* 2 NÚT BẤM
* Để có thể điều khiển hoàn toàn động cơ DC tức là chúng ta phải điều khiển tốc độ và chiều quay của nó. Để thực hiện diều này ta có thể kết hợp hai kỹ thuật sau. **Chế độ rộng xung (PWM)** để kiểm soát tốc độ và **Mạnh cầu H (H-Bridge)** để điều khiển hướng quay.
* PWM - Để kiểm soát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

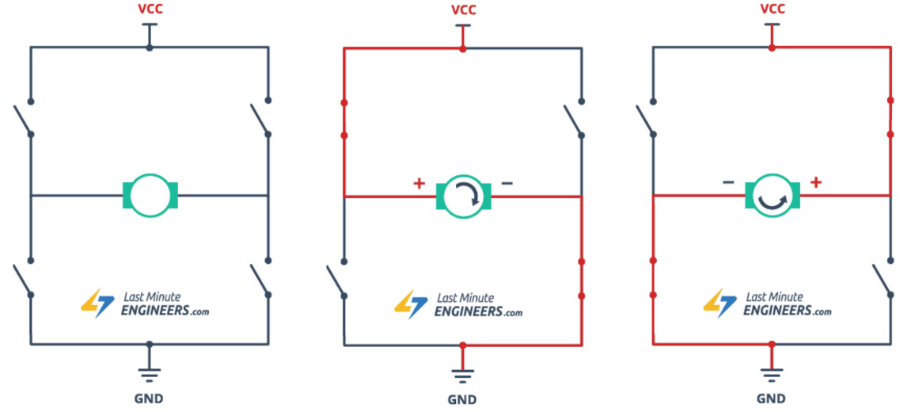
* 

Hình 36. Kỹ thuật điều chế độ rộng xung (PWM)

* H-Bridge - Để điều khiển hướng quay

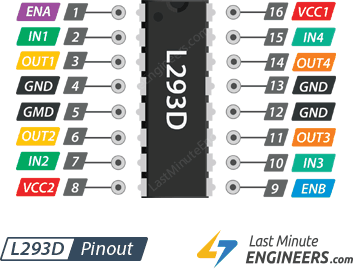
Chiều quay của động cơ DC có thể được điều khiển bằng cách thay đổi cực tính của điện áp đầu vào. Một kỹ thuật phổ biến để làm điều này là sử dụng mạch cầu H. Một mạch cầu H gồm có bốn công tắc với động cơ ở trung tâm tạo thành một sắp xếp giống như hình chữ H.

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 37. Hình minh họa H-Bridge hoạt động

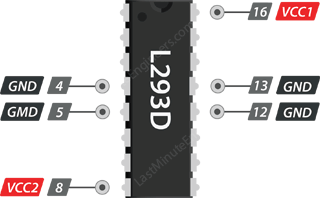
* IC điều khiển động cơ L293D



Hình 38. 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.

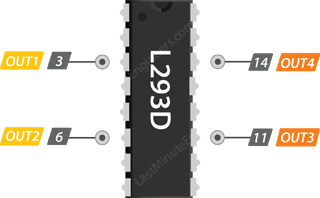
* Nguồn IC L293D



Hình 39. 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



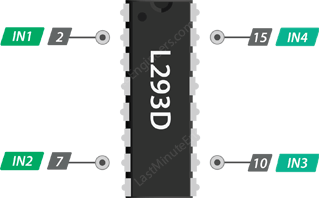
Hình 40. Các chân kết nối đầu ra của L293D

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.

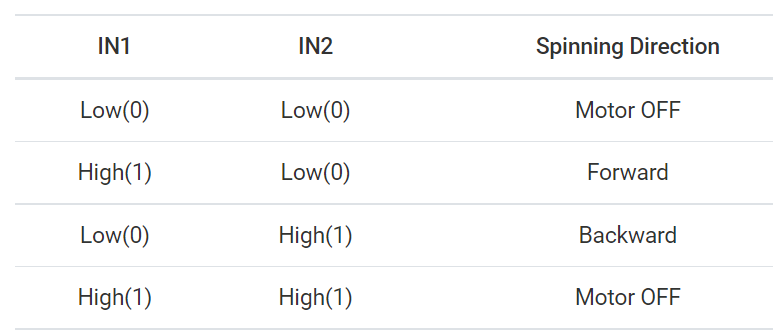
+ Direction Control Pins



Hình 41. Các chân điều khiển hướng của L293D

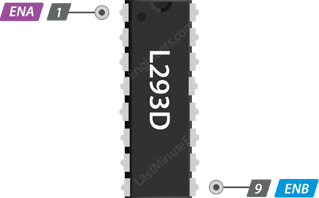
Sử dụng các chân điều khiển hướng, chúng ta có thể điều khiển động cơ quay tiến hoặc lùi. Các chân này điều khiển các công tắc của mạch H-Bridge nằm bên trong IC L293D. IC có hai chân điều khiển hướng cho mỗi phía. Các chân **IN1**, **IN2** điều khiển hướng quay của động cơ A và **IN3**, **IN4** điều 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 42. Biểu đồ minh họa cách thực hiện

+ Speed Control Pins



Hình 43. 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.

## Code chương trình

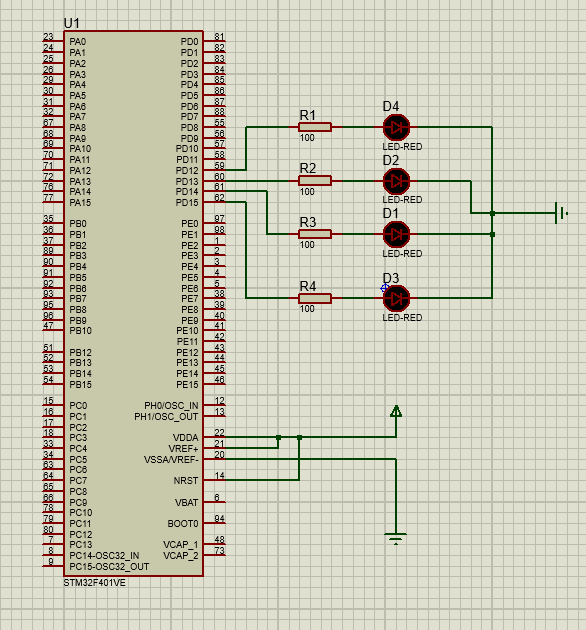
|  |
| --- |
| ***// dong co 1***  *int tocdo = 9;*  *int inp1 = 11;*  *int inp2 = 10;*  ***// dong co 2***  *int tocdo2 = 6;*  *int inp3 = 5;*  *int inp4 = 4;*  *void* ***setup****()*  *{*  *pinMode(tocdo, OUTPUT);*  *pinMode(inp1, OUTPUT);*  *pinMode(inp2, OUTPUT);*  *pinMode(tocdo2, OUTPUT);*  *pinMode(inp3, OUTPUT);*  *pinMode(inp4, OUTPUT);*  *digitalWrite(2, HIGH);*  *attachInterrupt(0, tatdongco, CHANGE);*  *digitalWrite(3, HIGH);*  *attachInterrupt(1,dieukientocdogiam, CHANGE);*    ***//tat dong co***  *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 19. Nháy 4 đèn Led thông qua STM32

## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều kiển 4 đèn Led tự động bật/tắt sau 1 giây thông qua 4 chân PD12, PD13, PD14 và PD15 của STM32F401VE.

## Sơ đồ thiết kế



Sơ đồ mạch trên proteus

## Đặc điểm của linh kiện

* 4 điện trở: 100 Ω
* 4 đèn Led
* 1 STM32F401VE

## Code chương trình

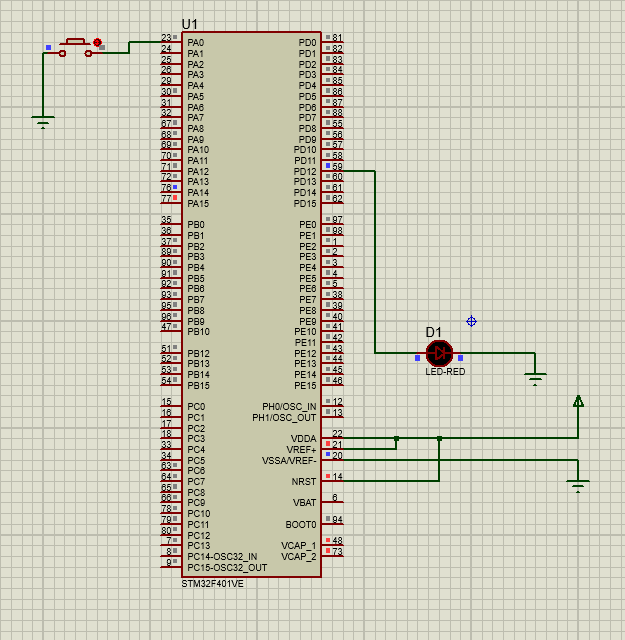
|  |
| --- |
| int **main**(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  **while** (1)  { **//Hàm đảo trạng thái các chân 12,13,14,15**  HAL\_GPIO\_TogglePin(GPIOD,GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15);  HAL\_Delay(1000); **//delay 1s**  }  *}* |

# Bài 20. Ngắt ngoài trên STM32

## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều kiển ngắt ngoài bật/tắt đèn Led trên STM32F401VE.Trên MCU này, nhà sản xuất đã kết nối sẵn cho chúng ta chân PA0 với User button nên ta sử dụng chân này để kết nối với nút bấm. Khi ta nhấn một nút nhấn thì phát sinh sự kiện ngắt ngoài gửi vào vi điều khiển, khi đó vi điều khiện triệu gọi một chương trình con phục vụ ngắt để thực hiện bật/tắt led ở PA12.

## Sơ đồ thiết kế



Sơ đồ trên proteus

## Đặc điểm của linh kiện

* 1 đèn Led
* 1 Nút bấm
* 1 STM32F401VE

## Code chương trình

Trong hàm void **HAL\_GPIO\_EXTI\_Callback (uint16\_t GPIO\_Pin)** ta sẽ viết như sau

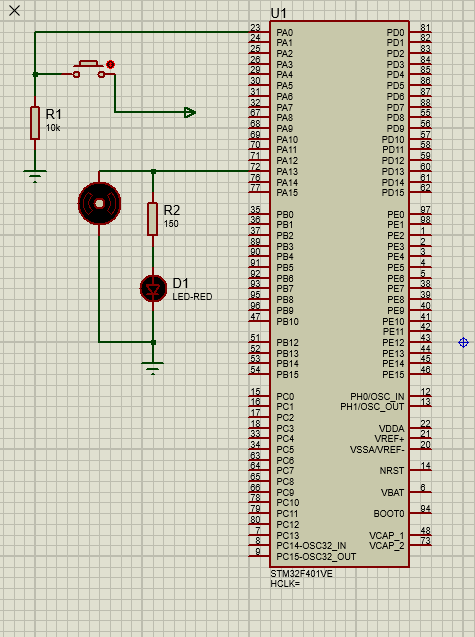
|  |
| --- |
| **/\*Câu lệnh if (GPIO\_Pin == GPIO\_PIN\_0) sẽ giúp kiểm tra, phân luồng, phát hiện ngắt có đúng đang sinh ra có phải ở chân 0 hay không.\*/**  void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)  {  if(GPIO\_Pin == GPIO\_PIN\_0)  { HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_12); }  } |

# Bài 21. Điều khiển đèn Led/ DC Motor bằng nút bấm thông qua STM32

## Mô tả

Hệ thống được thiết kế lập trình điều khiển đèn Led /hoặc DC motor thông qua một nút bấm, nếu nút bấm được bấm, thì sáng Led/hoặc quay động cơ, ngược lạ thì tắt Led/hoặc dừng động cơ.

## Sơ đồ thiết kế



Sơ đồ trên proteus

## Đặc điểm của linh kiện

* 2 điện trở: 100 Ω
* 1 đèn Led
* 1 DC Motor
* 1 STM32F401VE

## Code chương trình

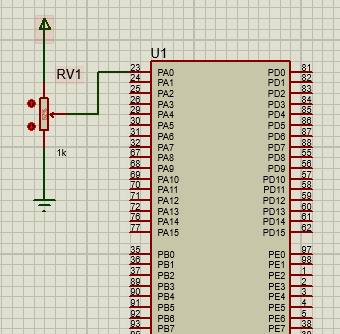
|  |
| --- |
| int **main**(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  **while** (1)  {  **//Đọc tín hiệu chân PA0**  GPIO\_PinState pin0State = HAL\_GPIO\_ReadPin(GPIOA, GPIO\_PIN\_0);  if(pin0State == GPIO\_PIN\_SET) **//nut vua bam**  {  **//Sáng đèn và bật động cơ**  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_13, GPIO\_PIN\_SET);  }else  {  **//Tắt đèn và tắt động cơ**  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_13, GPIO\_PIN\_RESET);  }  }  *}* |

# Bài 22. Chức năng ADC trên vi điều khiển STM32F4

## Mô tả

ADC (Analog-to-Digital Converter) bộ chuyển đổi tín hiệu tương tự - số là thuật ngữ nói đến sự chuyển đổi một tín hiệu tương tự thành tín hiệu số để dùng trong các hệ thống số (Digital Systems) hay vi điều khiển.

## Sơ đồ thiết kế



Sơ đồ trên proteus

## Đặc điểm của linh kiện

* 1 Nút bấm
* 1 STM32F401VE

## Code chương trình

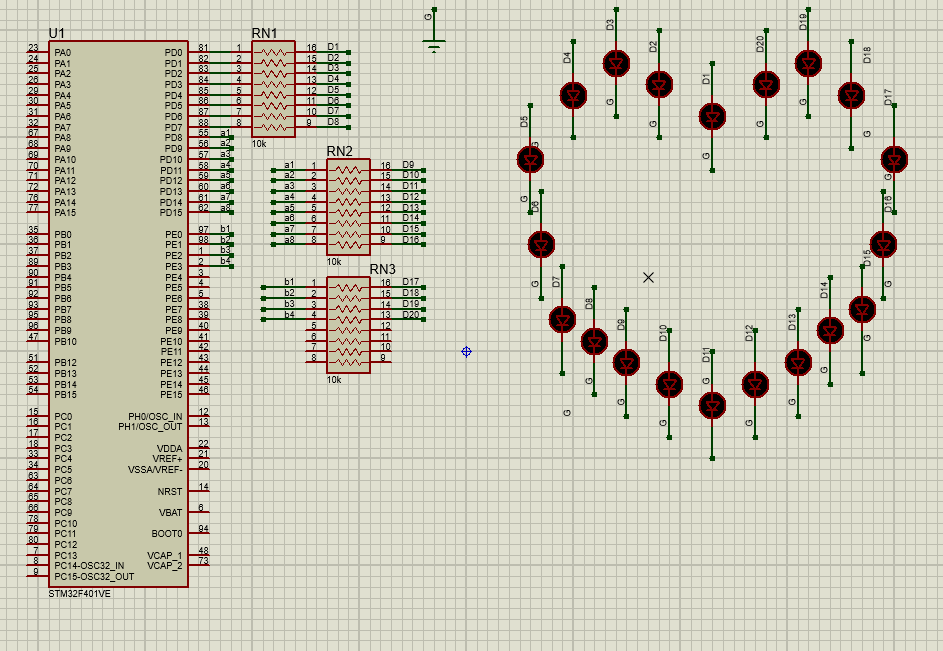
|  |
| --- |
| void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)  {  if(GPIO\_Pin == GPIO\_PIN\_0)  { HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_12); }  } |

# Bài 23. Sáng Led hình trái tim bằng STM32

## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều kiển bật/tắt 20 đèn Led được sắp xếp thành hình trái tim, các đèn Led được kết nối vào STM32 thông qua 3 qua điện trở 16 cổng.

## Sơ đồ thiết kế



Sơ đồ trên proteus

## Đặc điểm của linh kiện

* 20 đèn Led
* 3 bộ điện trở 16 cổng
* 1 STM32F401VE

## Code chương trình

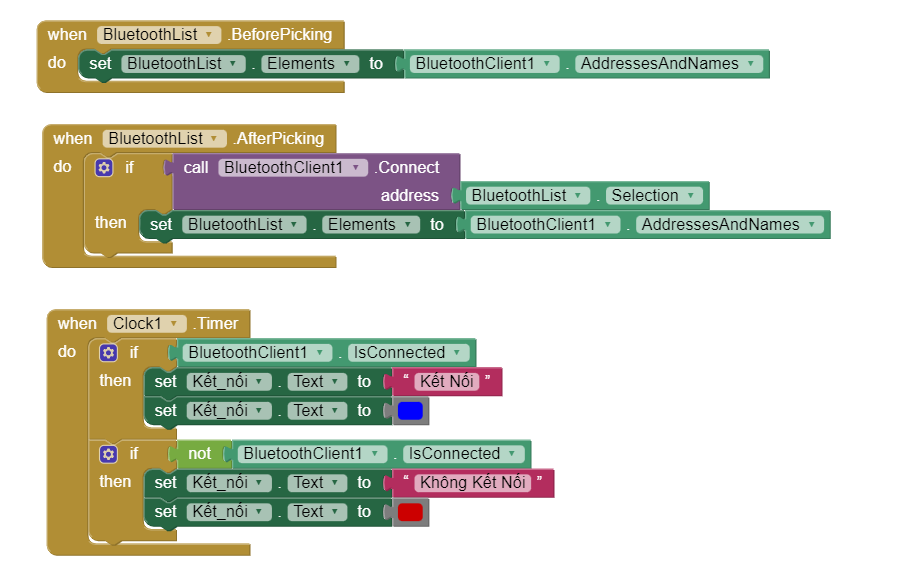
|  |
| --- |
| **//Xây dựng các hàm tắt đèn, sáng đèn, sáng lần lượt**  void SangHetDen(); **//Bật hết tất cả các đèn**  void TatHetDen(); **//Tắt hết tất cả các đèn**  void SangLanLuot(); **//Bật sáng từng đèn, giữ nguyên không tắt**  void SangLanLuot2(); **//Nháy từng đèn**  int **main**(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  **while** (1)  { **//Chương tình chính**  SangHetDen();  HAL\_Delay(1000);  TatHetDen();  HAL\_Delay(1000);  SangLanLuot();  HAL\_Delay(1000);  SangLanLuot2();  }  }  void **SangHetDen**(){  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);  **//Sáng đèn ở các cổng PE**  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);  }  void **TatHetDen**(){  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_RESET);  **// Tắt đèn ở các cổng PE**  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_RESET);  }  void **SangLanLuot**(){  TatHetDen();  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);  HAL\_Delay(100);  }  void **SangLanLuot2**(){  TatHetDen();  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_RESET);    HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_RESET);  *}* |

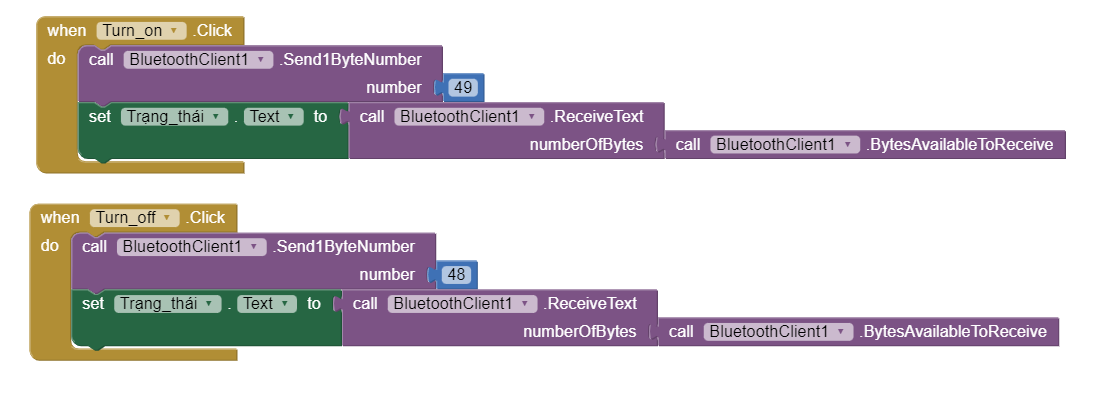
# Bài 24. Điều khiển đèn led đơn giản qua Bluetooth bằng appinventor

## Giao diện

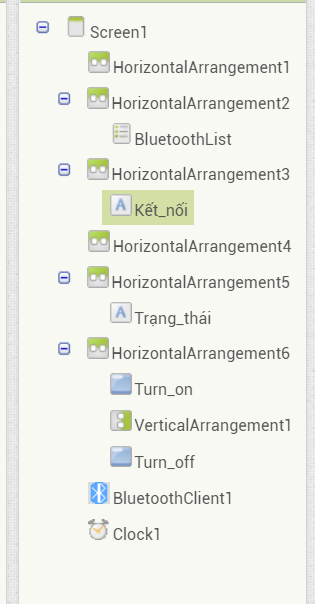


## Blocks





## Components

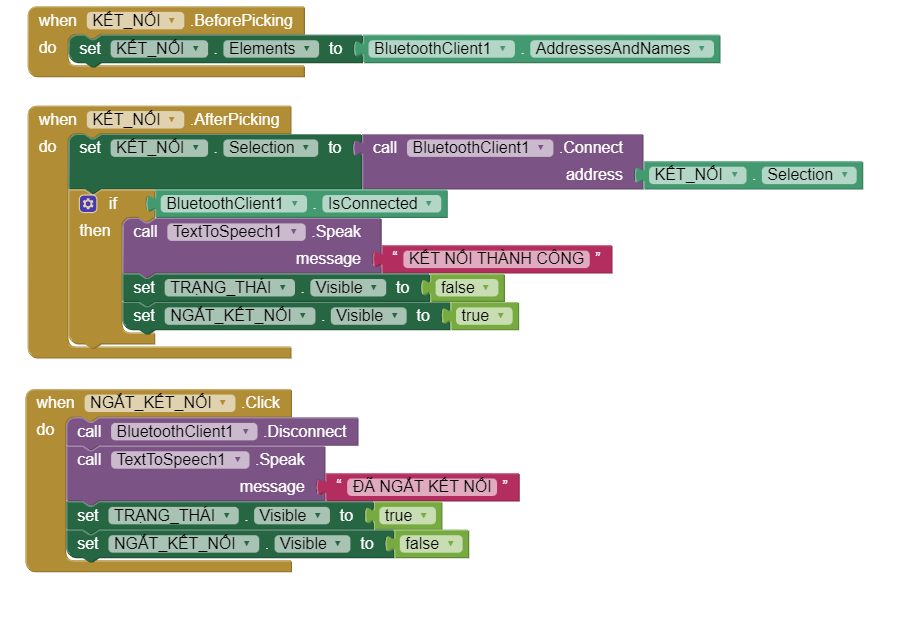
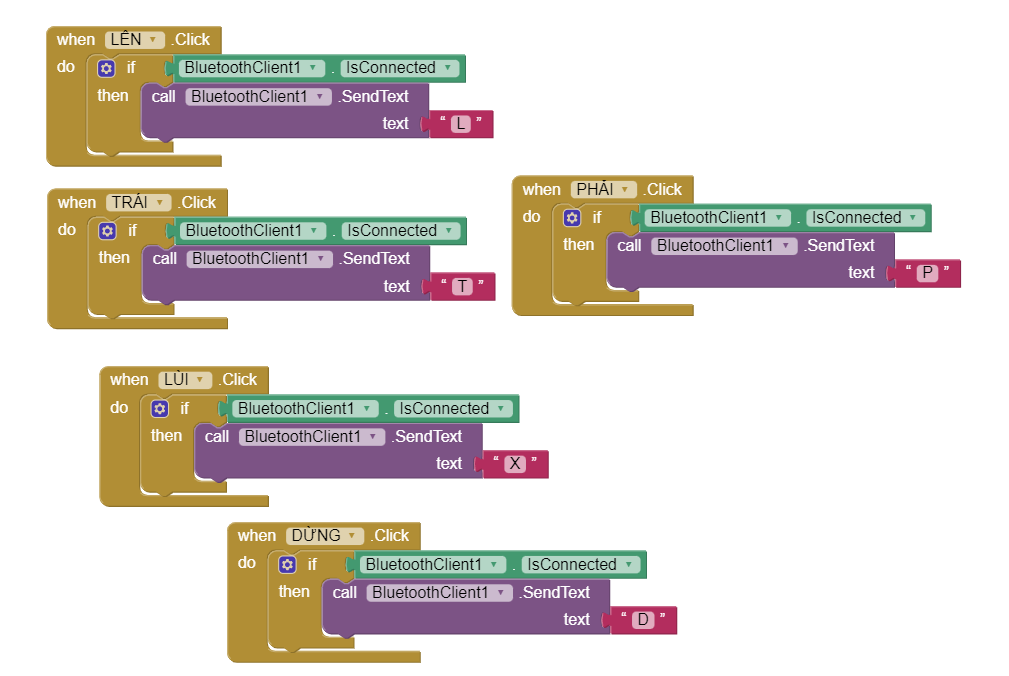


# Bài 25. Điều khiển xe qua Bluetooth bằng appinventor

## Giao diện



## Blocks

## Components

