***Bài 1: Led nhấp nháy***

# 1.Mô tả:

Hệ thống được thiết kế để thực hiện Bật/ Tắt đèn Led, Đèn được nối vào Arduino ở cổng số 12, khoảng thời gian là 300ms.

# 2.Thiết bị :

# 3.Sơ đồ thiết kế *:*

# ♣4. Mã lệnh:

|  |
| --- |
| **Code đèn led nhấp nháy** |
| |  | | --- | | void setup() { | |  |  | |  | pinMode (13, OUTPUT); | |  | } | |  |  | |  | void loop() { | |  |  | |  | digitalWrite (13, HIGH); // bật đèn | |  | delay(500); | |  | digitalWrite (13, LOW); // tắt đèn | |  | delay(500); | |  | } | |

# Bài 2 : Đèn sáng khi nhấn phím.

# 1.Mô tả:

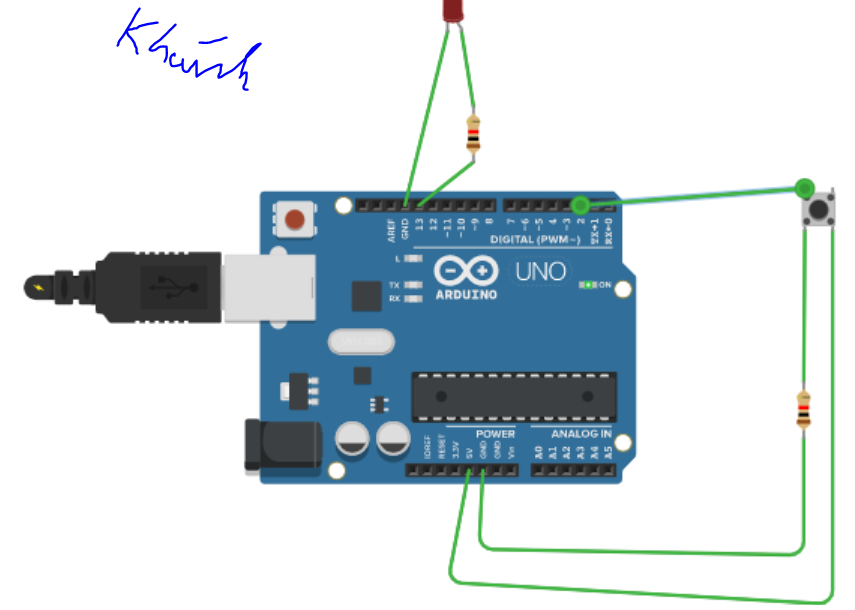
Hệ thống thực hiện Bật Led thông qua nút nhấn, Đèn được nối vào Arduino ở cổng số 13. đầu ra Nút bấm được nối vào pin 2. Đèn sang khi nút nhấn và ngược lại

# 2.Thiết bị:

* SIMULINO UNO
* BUTTON
* LED-RED

# *3.*Sơ đồ thiết kế *:*

***Hình 2.1 Thiết kế trên Proteus***



**Hình 2.2 Thiết kế trên TINKERCAD**

# Mã lệnh chính

|  |
| --- |
| int x = 0; // Để lưu giá trị đọc được từ cổng 2  void **setup**() {  pinMode(2, INPUT);  pinMode(13, OUTPUT);  }  void **loop**() {  x = digitalRead(2); //Đọc cổng 2, lưu trữ vào biến x  if (x == HIGH){ //Kiểm tra xem nút có đang bấm không  digitalWrite(13,HIGH); // Bật led  } else {  digitalWrite(13,LOW); //Tắt led  }  delay(100); //Đợi trong 100 mili giây  } |

# Bài 3. Bật đèn Led với cảm biến nhiệt độ TMP36

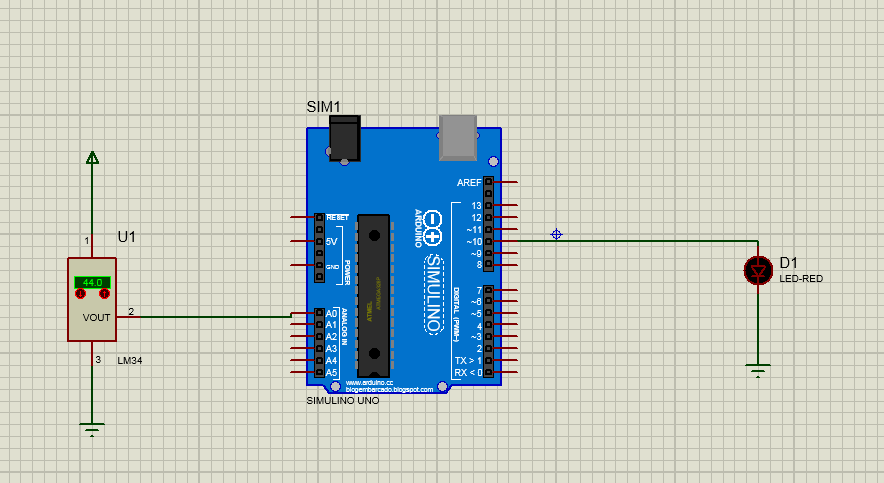
1. Mô tả

Bài này thực hiện bằng cách cho cảm biến nhiệt độ TMP36 lớn hơn 37 độ thì đèn led sẽ sáng lên.

1. Đặc điểm linh kiện

* Arduino uno
* Đèn led
* TMP36

1. Sơ đồ thiết kế



1. Mã lệnh chính

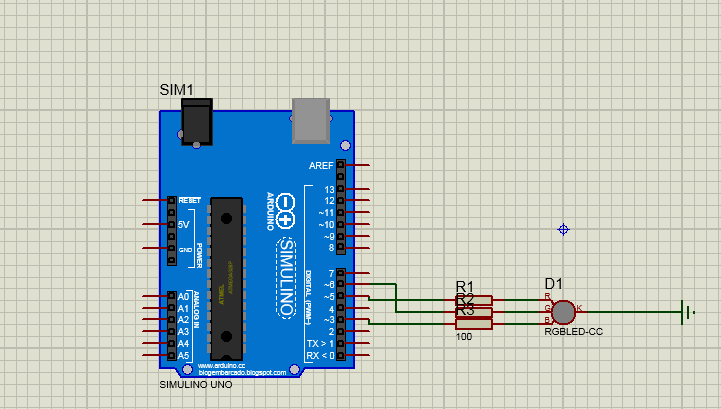
|  |
| --- |
| *void setup()*  *{*  *Serial.begin(96000);*  *}*  *void loop()*  *{*  *//Doc gia tri cam bien*  *int giatri = analogRead(A0);*  *int nhietDo = map(giatri,20,358,-40,125);*  *//Serial.print(nhietDo);*  *if(nhietDo>37) digitalWrite(10,HIGH);*  *else digitalWrite(10,LOW);*    *delay(1000);*  *}* |

# Bài 4. Chạy đèn Led RGB

1. Mô tả

Bài này thực hiện việc thay đổi màu trong liên tục trong 1s của đèn Led RGB. Được nối vào cổng 9, 10 và 11.

1. Sơ đồ thiết kế



Hình 4. Chạy đèn Led RGB

1. Đặc điểm linh kiện

* Arduino uno
* Led RGB
* Mã lệnh chính

4.Mã lệnh chính

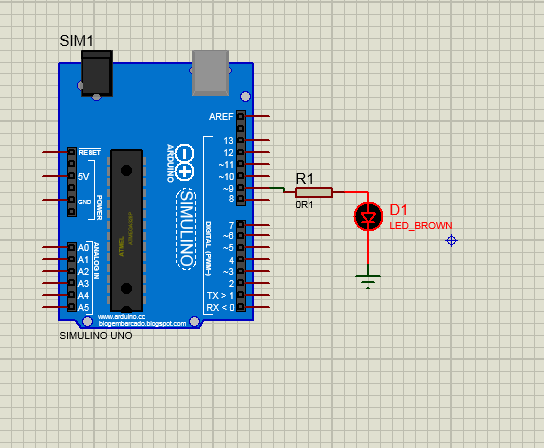
|  |
| --- |
| *const int redPin = 11;*  *const int greenPin = 10;*  *const int bluePin = 9;*  *void setup() {*  *pinMode(redPin, OUTPUT);*  *pinMode(greenPin, OUTPUT);*  *pinMode(bluePin, OUTPUT);*  *}*  *void loop()*  *{*  *color(255, 0, 0); // turn the RGB LED red*  *delay(1000);*  *color(0,255, 0); // turn the RGB LED green*  *delay(1000);*  *color(0, 0, 255); // turn the RGB LED blue*  *delay(1000);*  *// Example blended colors:*  *color(255,0,252); // turn the RGB LED red*  *delay(1000);*  *color(237,109,0); // turn the RGB LED orange*  *delay(1000);*  *color(255,215,0); // turn the RGB LED yellow*  *delay(1000);*  *color(34,139,34); // turn the RGB LED green*  *delay(1000);*  *color(0,112,255); // turn the RGB LED blue*  *delay(1000);*  *color(0,46,90); // turn the RGB LED indigo*  *delay(1000);*  *color(128,0,128); // turn the RGB LED purple*  *delay(1000);*  *}*  *void color (unsigned char red, unsigned char green, unsigned char blue)*  *{*  *analogWrite(redPin, red);*  *analogWrite(greenPin, green);*  *analogWrite(bluePin, blue);*  *}* |

# Bài 5. LED SÁNG DẦN

1. Mô tả

Bài này thực hiện bằng cách sử dụng các chân PWM, xuất các mức điện áp đầu ra thay đổi từ 0-5V, để làm cho led sáng dần.

# 2. Sơ đồ thiết kế



Hình 5 Led sáng dần

# 3. Đặc điểm linh kiện

- Arduino UNO

- Led

- Điện trở

# 4. Mã lệnh chính

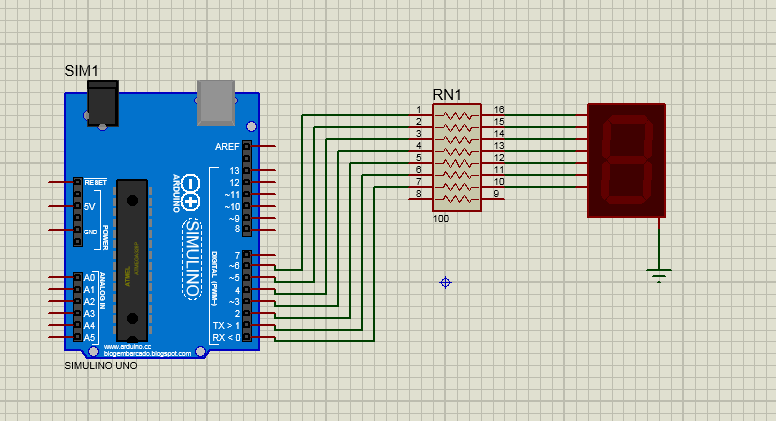
|  |
| --- |
| *// C++ code*  *//*  *int brightness = 0;*  *void setup()*  *{*  *pinMode(9, OUTPUT);*  *}*  *void loop()*  *{*  *for( brightness =0;brightness<=255;brightness +=5)*  *{*  *analogWrite(9,brightness);*  *delay(30);*  *}*  *for( brightness =0;brightness<=255;brightness -=5)*  *{*  *analogWrite(9,brightness);*  *delay(30);*  *}*  } |

# Bài 6: LED 7 chân

1. Mô tả

Bài này thực hiện bằng việc chạy con Led 7 segement đi từ số 0, mỗi 1 giây tăng 1 đơn vị cho đến số 9 sẽ trở về 0.

1. Sơ đồ thiết kế



1. Đặc điểm linh kiện

1 con led 7 chân

1 điện trở 74HC4511

Arduino Uno

1. Mã lệnh chính

|  |
| --- |
| *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 KHONG(){*  *digitalWrite(a,HIGH); digitalWrite(b,HIGH);*  *digitalWrite(c,HIGH); digitalWrite(d,HIGH);*  *digitalWrite(e,HIGH); digitalWrite(f,HIGH);*  *digitalWrite(g,LOW);*    *}*  *void MOT(){*  *digitalWrite(a,LOW); digitalWrite(b,HIGH);*  *digitalWrite(c,HIGH); digitalWrite(d,LOW);*  *digitalWrite(e,LOW); digitalWrite(f,LOW);*  *digitalWrite(g,LOW);*    *}*  *void HAI(){*  *digitalWrite(a,HIGH); digitalWrite(b,HIGH);*  *digitalWrite(c,LOW); digitalWrite(d,HIGH);*  *digitalWrite(e,HIGH); digitalWrite(f,LOW);*  *digitalWrite(g,HIGH);*  *}*  *void BA(){*  *digitalWrite(a,HIGH); digitalWrite(b,HIGH);*  *digitalWrite(c,HIGH); digitalWrite(d,HIGH);*  *digitalWrite(e,LOW); digitalWrite(f,LOW);*  *digitalWrite(g,HIGH);}*  *void BON(){*  *digitalWrite(a,LOW); digitalWrite(b,HIGH);*  *digitalWrite(c,HIGH); digitalWrite(d,LOW);*  *digitalWrite(e,LOW); digitalWrite(f,HIGH);*  *digitalWrite(g,HIGH);}*  *void NAM(){*  *digitalWrite(a,HIGH); digitalWrite(b,LOW);*  *digitalWrite(c,HIGH); digitalWrite(d,HIGH);*  *digitalWrite(e,LOW); digitalWrite(f,HIGH);*  *digitalWrite(g,HIGH);}*  *void SAU(){*  *digitalWrite(a,HIGH); digitalWrite(b,LOW);*  *digitalWrite(c,HIGH); digitalWrite(d,HIGH);*  *digitalWrite(e,HIGH); digitalWrite(f,HIGH);*  *digitalWrite(g,HIGH);}*  *void BAY(){*  *digitalWrite(a,HIGH); digitalWrite(b,HIGH);*  *digitalWrite(c,HIGH); digitalWrite(d,LOW);*  *digitalWrite(e,LOW); digitalWrite(f,LOW);*  *digitalWrite(g,LOW);*  *}*  *void TAM(){*  *digitalWrite(a,HIGH); digitalWrite(b,HIGH);*  *digitalWrite(c,HIGH); digitalWrite(d,HIGH);*  *digitalWrite(e,HIGH); digitalWrite(f,HIGH);*  *digitalWrite(g,HIGH);*  *}*  *void CHIN(){*  *digitalWrite(a,HIGH); digitalWrite(b,HIGH);*  *digitalWrite(c,HIGH); digitalWrite(d,HIGH);*  *digitalWrite(e,LOW); digitalWrite(f,HIGH);*  *digitalWrite(g,HIGH);*  *}*  *void loop() {*  *KHONG(); delay(1000);*  *MOT(); delay(1000);*  *HAI(); delay(1000);*  *BA(); delay(1000);*  *BON(); delay(1000);*  *NAM(); delay(1000);*  *SAU(); delay(1000);*  *BAY(); delay(1000);*  *TAM(); delay(1000);*  *CHIN(); delay(1000);*  *}* |

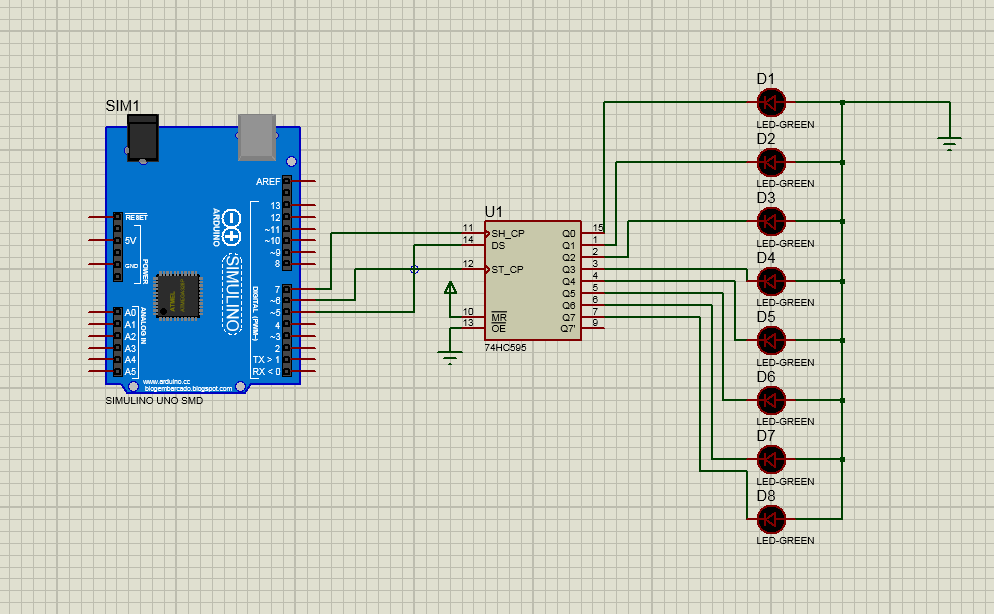
|  |
| --- |
| **Code Đèn sáng khi nhấn phím** |
| Int x =0;   |  |  | | --- | --- | |  | void setup() | |  | { | |  | pinMode(2, INPUT); | |  | pinMode(13, OUTPUT); | |  | } | |  |  | |  | void loop() | |  | { | |  | // Đọc cổng 2, cất vào biến x | |  | x = digitalRead(2); | |  | // Kiểm tra xem nút có đang nhấn hay không | |  | if (x == HIGH) { | |  | // Bật led | |  | digitalWrite(13, HIGH); | |  | } else { | |  | // Tắt led | |  | digitalWrite(13, LOW); | |  | } | |  | delay(1000); // Wait for 1000 millisecond(s) | |  | } | |

# Bài 7: 8 led nhấp nháy

1. Mô tả

Bài này thực hiện bằng cách cho 8 đèn led vào 4 chân là 9,10,11,12. Đầu tiên đèn led sẽ sáng lần lượt từng cặp từ trái sang phải. Sau đó sáng 4 bóng đèn chẳn cuối cùng nhấp nháy cả 8 led 3 lần.

1. Sơ đồ thiết kế



1. Đặc điểm linh kiện

Arduino uno

8 Led red

8 điện trở

1. Mã lệnh chính

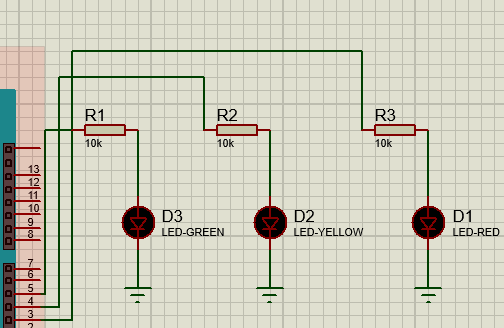
|  |
| --- |
| *void setup(){*  *pinMode(12, OUTPUT);*  *pinMode(11, OUTPUT);*  *pinMode(10, OUTPUT);*  *pinMode(9, OUTPUT);*    *pinMode(13, OUTPUT);*  *pinMode(7, OUTPUT);*  *}*  *void piscar(){*  *digitalWrite(12, HIGH);*  *digitalWrite(11, LOW);*  *digitalWrite(10, LOW);*  *digitalWrite(9, LOW);*  *delay(500);*  *digitalWrite(12, LOW);*  *digitalWrite(11, HIGH);*  *digitalWrite(10, LOW);*  *digitalWrite(9, LOW);*  *delay(500);*  *digitalWrite(12, LOW);*  *digitalWrite(11, LOW);*  *digitalWrite(10, HIGH);*  *digitalWrite(9, LOW);*  *delay(500);*  *digitalWrite(12, LOW);*  *digitalWrite(11, LOW);*  *digitalWrite(10, LOW);*  *digitalWrite(9, HIGH);*  *delay(500);*  *digitalWrite(12, LOW);*  *digitalWrite(11, LOW);*  *digitalWrite(10, LOW);*  *digitalWrite(9, LOW);*  *delay(500);*  *}*  *void SangChan(){*  *digitalWrite(12, LOW);*  *digitalWrite(11, HIGH);*  *digitalWrite(10, LOW);*  *digitalWrite(9, HIGH);*  *delay(500);*  *digitalWrite(12, LOW);*  *digitalWrite(11, LOW);*  *digitalWrite(10, LOW);*  *digitalWrite(9, LOW);*  *delay(500);*  *}*  *void ChopNhay(){*  *digitalWrite(12, HIGH);*  *digitalWrite(11, HIGH);*  *digitalWrite(10, HIGH);*  *digitalWrite(9, HIGH);*  *delay(500);*  *digitalWrite(12, LOW);*  *digitalWrite(11, LOW);*  *digitalWrite(10, LOW);*  *digitalWrite(9, LOW);*  *delay(500);*  *digitalWrite(12, HIGH);*  *digitalWrite(11, HIGH);*  *digitalWrite(10, HIGH);*  *digitalWrite(9, HIGH);*  *delay(500);*  *digitalWrite(12, LOW);*  *digitalWrite(11, LOW);*  *digitalWrite(10, LOW);*  *digitalWrite(9, LOW);*  *delay(500);*  *digitalWrite(12, HIGH);*  *digitalWrite(11, HIGH);*  *digitalWrite(10, HIGH);*  *digitalWrite(9, HIGH);*  *delay(500);*  *digitalWrite(12, LOW);*  *digitalWrite(11, LOW);*  *digitalWrite(10, LOW);*  *digitalWrite(9, LOW);*  *delay(500);*  *}*  *void loop(){*    *piscar();*  *delay(1000);*  *SangChan();*  *delay(1000);*  *ChopNhay();*  *delay(1000);*  *}* |

# Bài 8: Đèn giao thông

1. Mô tả

Bài này thực hiện bằng cách nối 3 chiếc đèn led màu xanh, vàng, đỏ lần lượt vào cổng D5,D4,D3. 30 giây đầu đèn led màu xanh sẽ sáng, 10 giây sau đèn vàng sáng và đèn đỏ sẽ sáng 25 giây.

1. Sơ đồ thiết kế



1. Đặc điểm linh kiện

Arduino uno

Led xanh, vàng, đỏ

3 điện trở

1. Mã lệnh chính

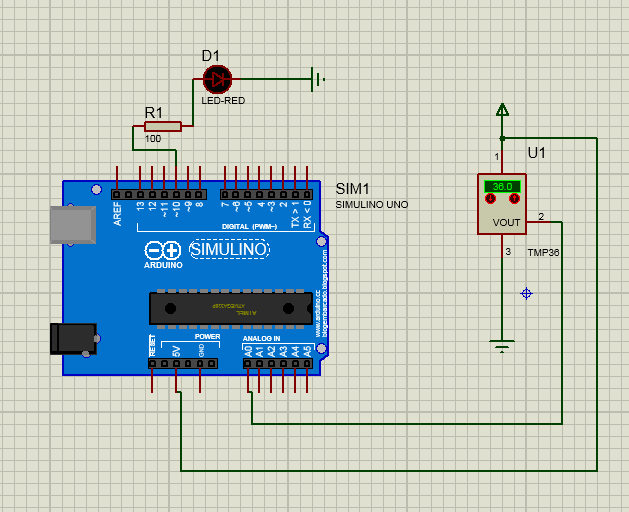
|  |
| --- |
| *// C++ code*  *//*  *int dem=0;*  *void setup()*  *{*  *pinMode(5, OUTPUT);*  *pinMode(4, OUTPUT);*  *pinMode(3, OUTPUT);*  *}*  *void loop()*  *{*  *dem += 1;*  *delay(1000);*  *if(dem == 65){*  *dem=0;*  *}*  *if(dem<=30){*  *digitalWrite(5, HIGH);*  *digitalWrite(4, LOW);*  *digitalWrite(3, LOW);*  *}*  *if(dem>30&&dem<40){*  *digitalWrite(5, LOW);*  *digitalWrite(4, HIGH);*  *digitalWrite(3, LOW);*  *}*  *if(dem>=40){*  *digitalWrite(5, LOW);*  *digitalWrite(4, LOW);*  *digitalWrite(3, HIGH);*    *}*  *}* |

# Bài 9: Cảm biến nhiệt độ và độ ẩm DTH11

1. Mô tả

Bài này thực hiện bằng cách nối cảm biến nhiệt độ và độ ẩm DTH11 vào cổng D4. Nối bóng đèn vàng vào cổng 11 và đèn đỏ vào cổng 13. Nếu nhiệt độ lớn hơn 30 độ C thì bóng đèn đỏ sáng lên. Nếu độ ẩm thấp hơn 40 đèn vàng sẽ sáng.

1. Sơ đồ thiết kế



1. Đặc điểm linh kiện

* Arduino uno
* Led vàng và led đỏ
* DTH11(Là cảm biến nhiệt độ, độ ẩm 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. Mã lệnh chính

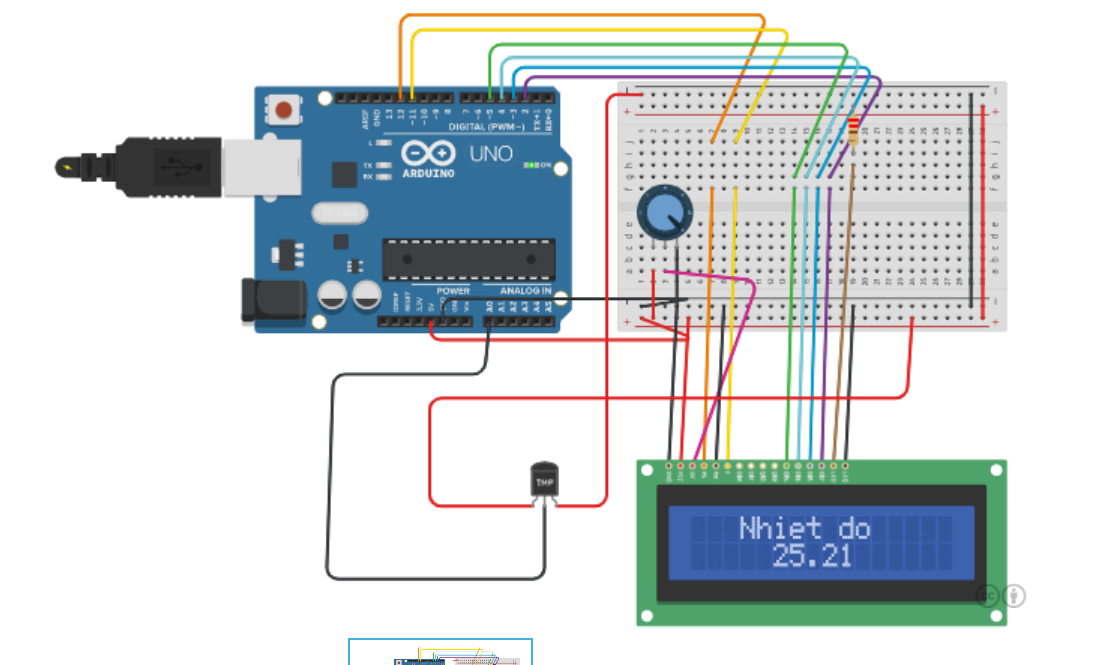
|  |
| --- |
| *#include "DHT.h" //khai bao thu vien cho DTH*  *const int DHTPIN = 4; // chan vao cua cam bien DHT11*  *const int DHTTYPE = DHT11;*  *DHT dht(DHTPIN, DHTTYPE);*  *void setup() {*  *dht.begin();*  *}*  *void loop() {*  *int h = dht.readHumidity(); // do am*  *float t = dht.readTemperature(); // nhiet do*  *if(t >=30){*  *digitalWrite(13,HIGH);*  *}*  *else digitalWrite(13,LOW);*    *if(h <= 40){*  *digitalWrite(11,HIGH);*  *}*  *else digitalWrite(11,LOW);*  *}* |

# Bài 10 đề chẵn: Hiển thị cảm biến ánh sáng sang LCD

1. Mô tả

Bài này thực hiện bằng cách sử dụng LCD đọc và hiển lên độ sáng của quang trở. Nếu giá trị cảm nhận được nhỏ hơn 50% thì đèn led sẽ sáng lên và ngược lại.

1. Sơ đồ thiết kế



1. Đặc điểm linh kiện

Arduino kết hợp LCD ( Sẽ giúp tiết kiện thời gian hơn khi nối LCD với Arduino)

Quang trở

Led đỏ

Điện trở

1. Mã lệnh chính

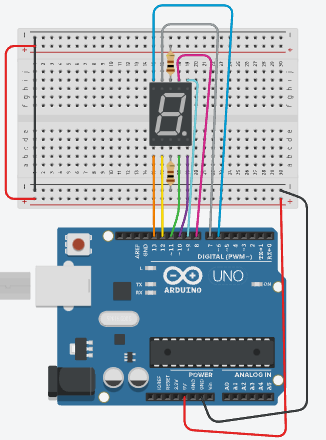
|  |
| --- |
| *#include <LiquidCrystal.h>*  *LiquidCrystal lcd(12, 11, 5, 4, 3, 2);*  *int LED = 8;*  *int LIGHT = 0;*  *int val =0;*  *void setup() {*    *lcd.begin(16, 2);*  *lcd.setCursor(4,0);*  *pinMode (LED, OUTPUT);*  *Serial.begin(9600);*    *lcd.print("DO SANG");*  *}*  *void loop() {*    *lcd.setCursor(0, 1);*  *val = analogRead(LIGHT);*  *Serial.println(val);*  *if (val < 680)*  *{*  *digitalWrite (8, LOW);*  *}*  *else*  *{*  *digitalWrite (8, HIGH);*  *}*  *delay(10);*    *lcd.print(val);*  *delay(1000);*  *}* |

# Bài 10-2A: Hiện các số đếm từ 0 đến 9

1. Mô tả

Bài này thực hiện bằng việc chạy con Led 7 segement đi từ số 0, mỗi 1 giây tăng 1 đơn vị cho đến số 9 sẽ trở về 0.

1. Sơ đồ thiết kế



1. Đặc điểm linh kiện

* 1 con led 7 chân
* 1 điện trở
* Arduino Uno

1. Mã lệnh chính

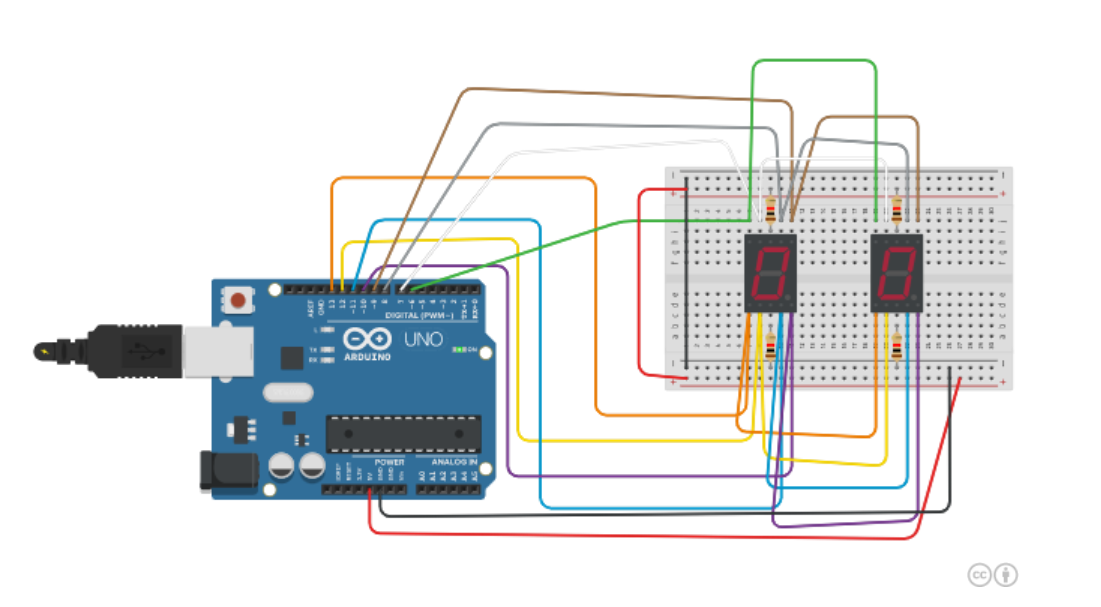
|  |
| --- |
| *int E = 13;*  *int D = 12;*  *int C = 11;*  *int DP = 10;*  *int B = 9;*  *int A = 8;*  *int F = 7;*  *int G = 6;*  *void zero()*  *{*  *digitalWrite(E,LOW);*  *digitalWrite(D,LOW);*  *digitalWrite(C,LOW);*  *digitalWrite(DP,HIGH);*  *digitalWrite(B,LOW);*  *digitalWrite(A,LOW);*  *digitalWrite(F,LOW);*  *digitalWrite(G,HIGH);*  *}*  *void one()*  *{*  *digitalWrite(E,HIGH);*  *digitalWrite(D,HIGH);*  *digitalWrite(C,LOW);*  *digitalWrite(DP,HIGH);*  *digitalWrite(B,LOW);*  *digitalWrite(A,HIGH);*  *digitalWrite(F,HIGH);*  *digitalWrite(G,HIGH);*  *}*  *void two()*  *{*  *digitalWrite(E,LOW);*  *digitalWrite(D,LOW);*  *digitalWrite(C,HIGH);*  *digitalWrite(DP,HIGH);*  *digitalWrite(B,LOW);*  *digitalWrite(A,LOW);*  *digitalWrite(F,HIGH);*  *digitalWrite(G,LOW);*  *}*  *void three()*  *{*  *digitalWrite(E,HIGH);*  *digitalWrite(D,LOW);*  *digitalWrite(C,LOW);*  *digitalWrite(DP,HIGH);*  *digitalWrite(B,LOW);*  *digitalWrite(A,LOW);*  *digitalWrite(F,HIGH);*  *digitalWrite(G,LOW);*  *}*  *void four()*  *{*  *digitalWrite(E,HIGH);*  *digitalWrite(D,HIGH);*  *digitalWrite(C,LOW);*  *digitalWrite(DP,HIGH);*  *digitalWrite(B,LOW);*  *digitalWrite(A,HIGH);*  *digitalWrite(F,LOW);*  *digitalWrite(G,LOW);*  *}*  *void five()*  *{*  *digitalWrite(E,HIGH);*  *digitalWrite(D,LOW);*  *digitalWrite(C,LOW);*  *digitalWrite(DP,HIGH);*  *digitalWrite(B,HIGH);*  *digitalWrite(A,LOW);*  *digitalWrite(F,LOW);*  *digitalWrite(G,LOW);*  *}*  *void six()*  *{*  *digitalWrite(E,LOW);*  *digitalWrite(D,LOW);*  *digitalWrite(C,LOW);*  *digitalWrite(DP,HIGH);*  *digitalWrite(B,HIGH);*  *digitalWrite(A,LOW);*  *digitalWrite(F,LOW);*  *digitalWrite(G,LOW);*  *}*  *void seven()*  *{*  *digitalWrite(E,HIGH);*  *digitalWrite(D,HIGH);*  *digitalWrite(C,LOW);*  *digitalWrite(DP,HIGH);*  *digitalWrite(B,LOW);*  *digitalWrite(A,LOW);*  *digitalWrite(F,HIGH);*  *digitalWrite(G,HIGH);*  *}*  *void eight()*  *{*  *digitalWrite(E,LOW);*  *digitalWrite(D,LOW);*  *digitalWrite(C,LOW);*  *digitalWrite(DP,HIGH);*  *digitalWrite(B,LOW);*  *digitalWrite(A,LOW);*  *digitalWrite(F,LOW);*  *digitalWrite(G,LOW);*  *}*  *void nine()*  *{*  *digitalWrite(E,HIGH);*  *digitalWrite(D,LOW);*  *digitalWrite(C,LOW);*  *digitalWrite(DP,HIGH);*  *digitalWrite(B,LOW);*  *digitalWrite(A,LOW);*  *digitalWrite(F,LOW);*  *digitalWrite(G,LOW);*  *}*  *void setup()*  *{*  *pinMode(E, OUTPUT);*  *pinMode(D, OUTPUT);*  *pinMode(C, OUTPUT);*  *pinMode(DP, OUTPUT);*  *pinMode(B, OUTPUT);*  *pinMode(A, OUTPUT);*  *pinMode(F, OUTPUT);*  *pinMode(G, OUTPUT);*  *}*  *void loop()*  *{ zero();*  *delay(1000);*    *one();*  *delay(1000);*    *two();*  *delay(1000);*    *three();*  *delay(1000);*    *four();*  *delay(1000);*    *five();*  *delay(1000);*    *six();*  *delay(1000);*    *seven();*  *delay(1000);*    *eight();*  *delay(1000);*      *nine();*  *delay(1000);*  *}* |

# Bài 10-2B: Hiện các số đếm từ 00 đến 99

1. Mô tả

Bài này thực hiện bằng cách cho 2 đèn led 7 đoạn chạy liên tục từ số 00 đến số 99 mỗi 0,02 giây sẽ nhảy 1 số.

1. Sơ đồ thiết kế



1. Đặc điêm linh kiện

* Arduino uno
* 2 led 7 đoạn
* 7 điện trở

1. Mã lệnh chính

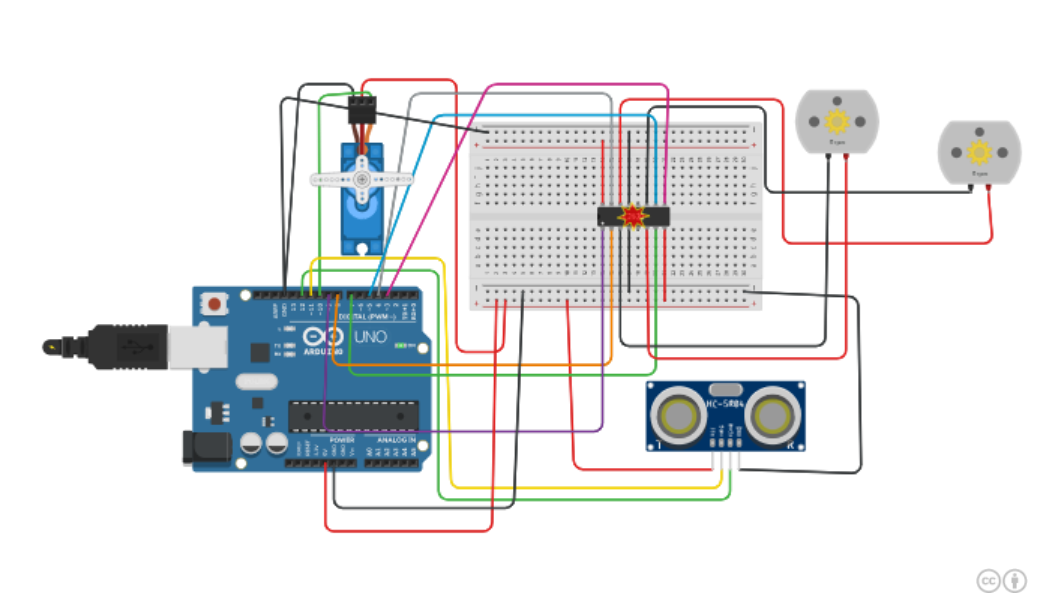
|  |
| --- |
| *int a = 2;*  *int b = 3;*  *int c = 4;*  *int d = 5;*  *int e = 6;*  *int f = 7;*  *int g = 8;*  *int dp=12;*  *int digit1 = 10;*  *int digit2 = 9;*    *char Num[10] = {*  *0b1111110, // 0*  *0b0110000, // 1*  *0b1101101, // 2*  *0b1111001, // 3*  *0b0110011, // 4*  *0b1011011, // 5*  *0b1011111, // 6*  *0b1110000, // 7*  *0b1111111, // 8*  *0b1111011 // 9*  *};*    *void number(int n)*  *{*  *digitalWrite(a, n & 0b1000000);*  *digitalWrite(b, n & 0b0100000);*  *digitalWrite(c, n & 0b0010000);*  *digitalWrite(d, n & 0b0001000);*  *digitalWrite(e, n & 0b0000100);*  *digitalWrite(f, n & 0b0000010);*  *digitalWrite(g, n & 0b0000001);*  *}*    *void setup()*  *{*  *pinMode(a, OUTPUT);*  *pinMode(b, OUTPUT);*  *pinMode(c, OUTPUT);*  *pinMode(d, OUTPUT);*  *pinMode(e, OUTPUT);*  *pinMode(f, OUTPUT);*  *pinMode(g, OUTPUT);*  *pinMode(dp, OUTPUT);*  *pinMode(digit1, OUTPUT);*  *pinMode(digit2, OUTPUT);*    *digitalWrite(digit1, 0);*  *digitalWrite(digit2, 1);*  *}*  *long interval ;*  *int count = 0;*  *void loop()*  *{*    *int d1 = count / 10;*  *int d2 = count % 10;*    *if(millis() - interval > 1000)*  *{*  *count++;*  *interval = millis();*  *}*    *digitalWrite(digit2, 1);*  *digitalWrite(dp,HIGH );*  *digitalWrite(digit1, 1);*  *number(Num[d1]);*  *digitalWrite(digit1, 0);*  *delay(10); // 1*    *digitalWrite(digit2, 1);*  *digitalWrite(digit1, 1);*    *number(Num[d2]);*  *digitalWrite(digit2, 0);*  *delay(10);*    *}* |

# Bài 11: Xe tự hành

1. Mô tả

Bài này thực hiện bằng cách cho 1 động cơ điều khiển bánh trước và 2 DC motor điều khiển bánh sau. Gắn cho 1 thiết bị cảm biến siêu âm HC-SR04 dùng để xác định vật cản trước mặt với khoảng cách bao nhiêu nhiêu. Nếu tốc độ xe chậm, dừng xe và lùi xe ngược lại trong khoảng 10 giây. Nếu tốc độ xe nhanh đánh tay lái sang trái 20 độ và giảm tốc độ Z+10 đơn vị, rồi dừng hẳn sau 10 giây.

1. Sơ đồ thiết kế

3.Đặc điểm linh kiện

* Arduino uno kết hợp với servo(bộ điều khiển chuyển hướng)
* 2 động cơ DC motor
* L293D (là một phần board mở rộng cho các board arduino, dùng để điều khiển các loại động cơ DC, động cơ bước và động cơ servo.Breadboard Small
* Cảm biến siêu âm HC-SR04 (là cảm biến dùng để xác định khoảng cách trong phạm vi nhỏ bằng cách phát sóng siêu âm. Cảm biến với độ chính xác chính xác khá cao)

# 4.Mã lệnh chính

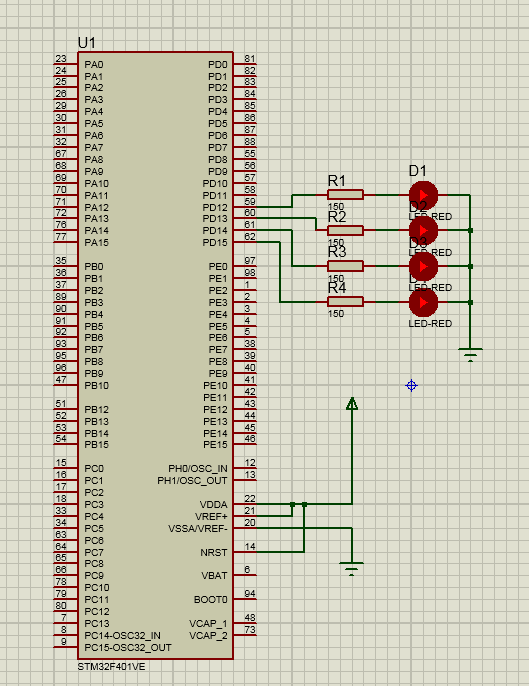
|  |
| --- |
| *// Motor A*  *int enA = 9;*  *int in1 = 8;*  *int in2 = 7;*  *// Motor B*  *int enB = 3;*  *int in3 = 5;*  *int in4 = 4;*  *//*  *const int trig = 11; //chân trig của HC-SR04*  *const int echo = 12; //chân echo của HC-SR04*  *#include <Servo.h>*  *Servo servo\_10;*  *void setup() {*    *pinMode(enA, OUTPUT);*  *pinMode(enB, OUTPUT);*  *pinMode(in1, OUTPUT);*  *pinMode(in2, OUTPUT);*  *pinMode(in3, OUTPUT);*  *pinMode(in4, OUTPUT);*  *pinMode(trig,OUTPUT); //chân trig sẽ phát tín hiệu*  *pinMode(echo,INPUT);*    *servo\_10.attach(10, 500, 2500);*  *// Turn off motors - Initial state*  *digitalWrite(in1, LOW);*  *digitalWrite(in2, LOW);*  *digitalWrite(in3, LOW);*  *digitalWrite(in4, LOW);*  *Serial.begin(96000);*  *}*  *// Chuyển hướng động cơ*  *void directionControl() {*  *// Đặt động cơ ở tốc độ tối đa 255*  *analogWrite(enA, 255);*  *analogWrite(enB, 255);*  *// Bật động cơ A & B*  *digitalWrite(in1, HIGH);*  *digitalWrite(in2, LOW);*  *digitalWrite(in3, HIGH);*  *digitalWrite(in4, LOW);*  *delay(2000);*    *}*  *void cham() {*  *servo\_10.write(90);*  *analogWrite(enA, 180);*  *analogWrite(enB, 180);*  *// Bật động cơ A & B*  *digitalWrite(in1, HIGH);*  *digitalWrite(in2, LOW);*  *digitalWrite(in3, HIGH);*  *digitalWrite(in4, LOW);*  *//dừng xe*  *digitalWrite(in1, LOW);*  *digitalWrite(in2, LOW);*  *digitalWrite(in3, LOW);*  *digitalWrite(in4, LOW);*  *//lùi xe*  *digitalWrite(in1, LOW);*  *digitalWrite(in2, HIGH);*  *digitalWrite(in3, LOW);*  *digitalWrite(in4, HIGH);*  *delay(10000);*  *}*  *void nhanh() {*  *analogWrite(enA, 255);*  *analogWrite(enB, 255);*  *// Bật động cơ A & B*  *digitalWrite(in1, HIGH);*  *digitalWrite(in2, LOW);*  *digitalWrite(in3, HIGH);*  *digitalWrite(in4, LOW);*  *//quay trái 20*  *servo\_10.write(110);*  *//giảm tốc độ*  *for (int i = 255; i >= 145; --i) {*  *analogWrite(enA, i);*  *analogWrite(enB, i);*  *delay(20);*  *}*  *//dừng xe*  *digitalWrite(in1, LOW);*  *digitalWrite(in2, LOW);*  *digitalWrite(in3, LOW);*  *digitalWrite(in4, LOW);*  *delay(5000);*  *}*  *void loop() {*  *unsigned long duration; //biến đo thời gian*  *int distance; //biến lưu khoảng cách*  *directionControl();*  */\* Phát xung từ chân trig \*/*  *digitalWrite(trig,0); //tắt chân trig*  *delayMicroseconds(2); //chờ 2ms*  *digitalWrite(trig,1); //phát xung từ chân trig*  *delayMicroseconds(5); //xung có độ dài 5 microSeconds*  *digitalWrite(trig,0); //tắt chân trig*    */\* Tính toán thời gian \*/*  *duration = pulseIn(echo,HIGH); // Đo độ rộng xung HIGH ở chân echo.*  *distance = int(duration/2\*0.0340); // Tính khoảng cách đến vật.*  *Serial.println(distance);*  *delay(1000);*  *if(distance < 20000){*  *//cham();*  *nhanh();*  *}*  *}* |

# Bài 12. Nháy Led -STM32

# 1.Mô tả

Đây là bài thực hành lập trình trên board mạch STM32, trong bài này sẽ làm cho 4 đèn Led nhấp nháy liên tục với thời gian delay là 1 giây. Bốn đèn sẽ lần lượt được đấu vào chân PA12, PA13, PA14, PA15 của STM32.

# 2.Sơ đồ thiết kế



# 3.Đặc điểm linh kiện

* Led: 4 LED (màu đỏ)
* Điện trở: 4 điện trở (150 Ω)
* Mạch STM32F401VE

# 4.Mã lệnh chính

|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  int **main**(void){  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();    **while** (1) {  HAL\_GPIO\_TogglePin(GPIOD,GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15);  HAL\_Delay(1000);  }  } |

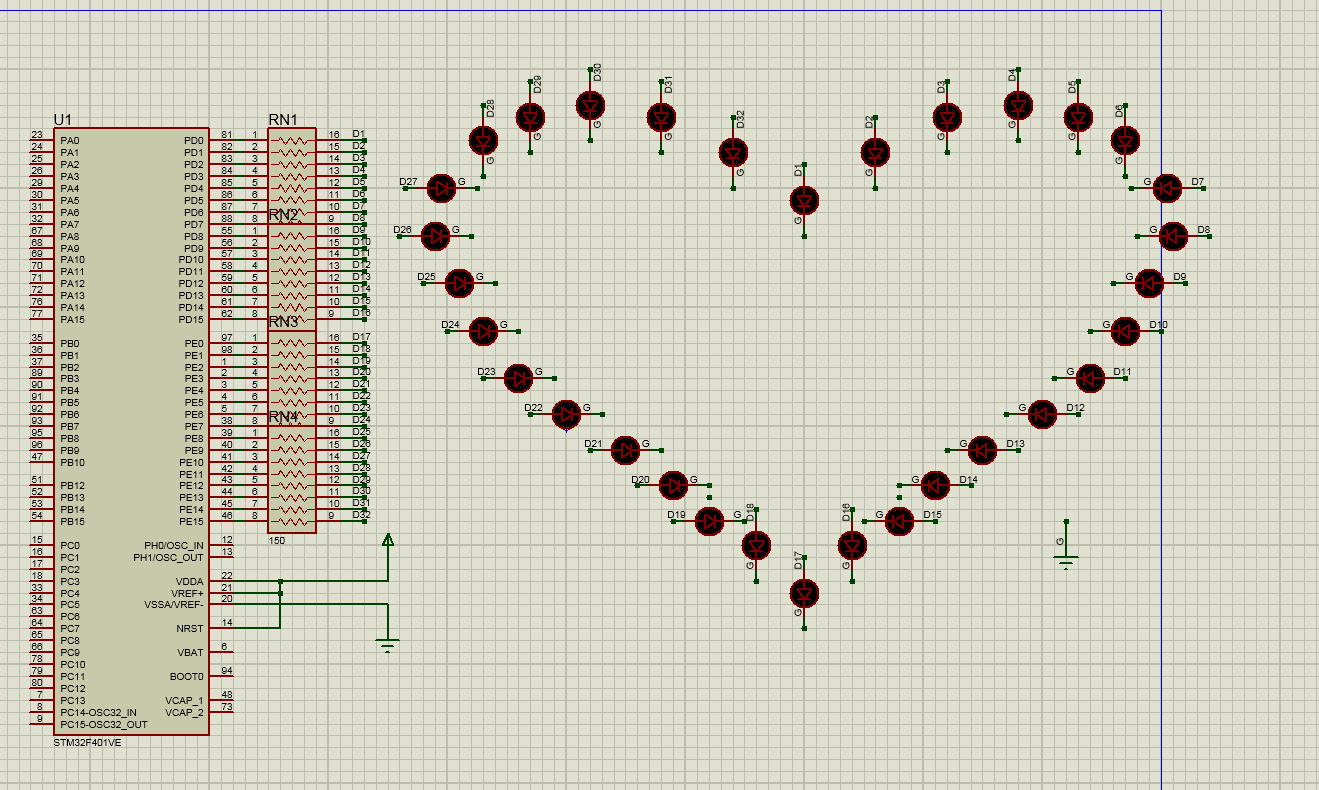
# Bài 13. Nháy Led Trái Tim - STM32

# 1.Mô tả

Đây là bài thực hành lập trình trên board mạch STM32, trong bài này sẽ làm cho 32 đèn Led được xếp thành hình trái sáng với nhiều hiệu ứng khác nhau, như: bật hết đèn, tắt hết đèn, sáng dần theo chiều đồng hồ, tắt dần theo chiều đồng hồ, và sáng từ trong trung tâm đi ra. 32 đèn led sẽ được đấu vào chân PD0 -> PE15 của board mạch STM32.

# 2.Sơ đồ thiết kế

.



# 3.Đặc điểm linh kiện

* Led: 32 LED (màu đỏ)
* Điện trở: 4 bảng mạch điện trở (32 điện trở) (150 Ω)
* Mạch STM32F401VE

# 4.Mã lệnh chính

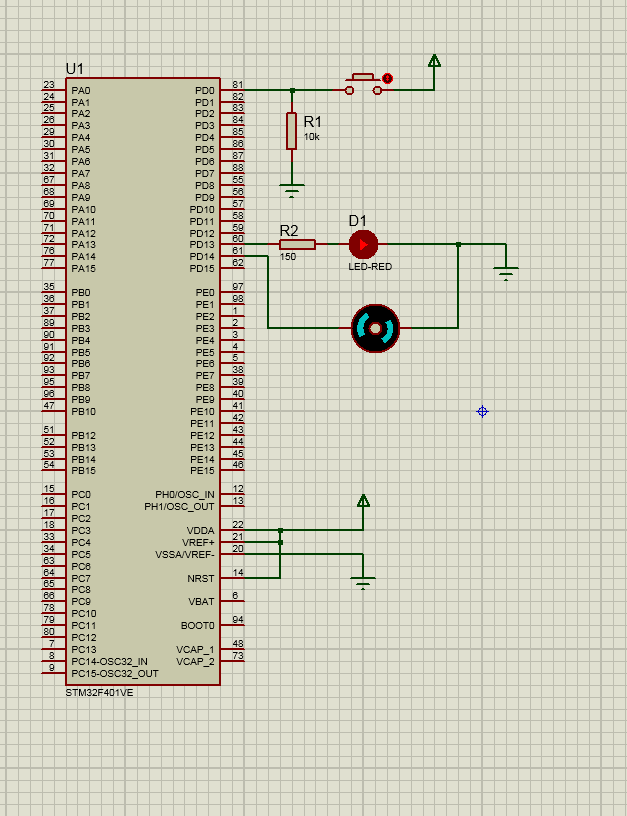
|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  void **batHetDen**();  void **tatHetDen**();  void **sangTheoKimDongHo**();  void **tatTheoKimDongHo**();  void **sangTuTrongRaNgoai**();  int **main**(void) {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();    **while** (1) {  batHetDen();  HAL\_Delay(500);  sangTheoKimDongHo();  HAL\_Delay(500);  tatHetDen();  HAL\_Delay(500);  tatTheoKimDongHo();  HAL\_Delay(500);  sangTuTrongRaNgoai();  HAL\_Delay(500);  tatHetDen();  HAL\_Delay(500);  }  }  void **batHetDen**(){  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0,GPIO\_PIN\_SET); // BAT DEN D1  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_1,GPIO\_PIN\_SET); // BAT DEN D2  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_2,GPIO\_PIN\_SET); // BAT DEN D3  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_3,GPIO\_PIN\_SET); // BAT DEN D4  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_4,GPIO\_PIN\_SET); // BAT DEN D5  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_5,GPIO\_PIN\_SET); // BAT DEN D6  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_6,GPIO\_PIN\_SET); // BAT DEN D7  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_7,GPIO\_PIN\_SET); // BAT DEN D8  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_8,GPIO\_PIN\_SET); // BAT DEN D9  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_9,GPIO\_PIN\_SET); // BAT DEN D10  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_10,GPIO\_PIN\_SET); // BAT DEN D11  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_11,GPIO\_PIN\_SET); // BAT DEN D12  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_12,GPIO\_PIN\_SET); // BAT DEN D13  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_13,GPIO\_PIN\_SET); // BAT DEN D14  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_14,GPIO\_PIN\_SET); // BAT DEN D15  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_15,GPIO\_PIN\_SET); // BAT DEN D16    HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_0,GPIO\_PIN\_SET); // BAT DEN D17  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_1,GPIO\_PIN\_SET); // BAT DEN D18  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_2,GPIO\_PIN\_SET); // BAT DEN D19  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_3,GPIO\_PIN\_SET); // BAT DEN D20  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_4,GPIO\_PIN\_SET); // BAT DEN D21  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_5,GPIO\_PIN\_SET); // BAT DEN D22  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_6,GPIO\_PIN\_SET); // BAT DEN D23  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_7,GPIO\_PIN\_SET); // BAT DEN D24  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_8,GPIO\_PIN\_SET); // BAT DEN D25  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_9,GPIO\_PIN\_SET); // BAT DEN D26  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_10,GPIO\_PIN\_SET); // BAT DEN D27  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_11,GPIO\_PIN\_SET); // BAT DEN D28  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_12,GPIO\_PIN\_SET); // BAT DEN D29  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_13,GPIO\_PIN\_SET); // BAT DEN D30  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_14,GPIO\_PIN\_SET); // BAT DEN D31  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_15,GPIO\_PIN\_SET); // BAT DEN D32  }  void **tatHetDen**() {  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0,GPIO\_PIN\_RESET); // TAT DEN D1  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_1,GPIO\_PIN\_RESET); // TAT DEN D2  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_2,GPIO\_PIN\_RESET); // TAT DEN D3  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_3,GPIO\_PIN\_RESET); // TAT DEN D4  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_4,GPIO\_PIN\_RESET); // TAT DEN D5  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_5,GPIO\_PIN\_RESET); // TAT DEN D6  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_6,GPIO\_PIN\_RESET); // TAT DEN D7  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_7,GPIO\_PIN\_RESET); // TAT DEN D8  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_8,GPIO\_PIN\_RESET); // TAT DEN D9  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_9,GPIO\_PIN\_RESET); // TAT DEN D10  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_10,GPIO\_PIN\_RESET); // TAT DEN D11  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_11,GPIO\_PIN\_RESET); // TAT DEN D12  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_12,GPIO\_PIN\_RESET); // TAT DEN D13  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_13,GPIO\_PIN\_RESET); // TAT DEN D14  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_14,GPIO\_PIN\_RESET); // TAT DEN D15  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_15,GPIO\_PIN\_RESET); // TAT DEN D16    HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_0,GPIO\_PIN\_RESET); // TAT DEN D17  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_1,GPIO\_PIN\_RESET); // TAT DEN D18  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_2,GPIO\_PIN\_RESET); // TAT DEN D19  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_3,GPIO\_PIN\_RESET); // TAT DEN D20  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_4,GPIO\_PIN\_RESET); // TAT DEN D21  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_5,GPIO\_PIN\_RESET); // TAT DEN D22  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_6,GPIO\_PIN\_RESET); // TAT DEN D23  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_7,GPIO\_PIN\_RESET); // TAT DEN D24  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_8,GPIO\_PIN\_RESET); // TAT DEN D25  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_9,GPIO\_PIN\_RESET); // TAT DEN D26  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_10,GPIO\_PIN\_RESET); // TAT DEN D27  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_11,GPIO\_PIN\_RESET); // TAT DEN D28  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_12,GPIO\_PIN\_RESET); // TAT DEN D29  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_13,GPIO\_PIN\_RESET); // TAT DEN D30  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_14,GPIO\_PIN\_RESET); // TAT DEN D31  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_15,GPIO\_PIN\_RESET); // TAT DEN D32  }  void **sangTheoKimDongHo**(){  **tatHetDen**();  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0,GPIO\_PIN\_SET); // BAT DEN D1  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_1,GPIO\_PIN\_SET); // BAT DEN D2  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_2,GPIO\_PIN\_SET); // BAT DEN D3  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_3,GPIO\_PIN\_SET); // BAT DEN D4  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_4,GPIO\_PIN\_SET); // BAT DEN D5  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_5,GPIO\_PIN\_SET); // BAT DEN D6  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_6,GPIO\_PIN\_SET); // BAT DEN D7  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_7,GPIO\_PIN\_SET); // BAT DEN D8  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_8,GPIO\_PIN\_SET); // BAT DEN D9  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_9,GPIO\_PIN\_SET); // BAT DEN D10  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_10,GPIO\_PIN\_SET); // BAT DEN D11  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_11,GPIO\_PIN\_SET); // BAT DEN D12  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_12,GPIO\_PIN\_SET); // BAT DEN D13  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_13,GPIO\_PIN\_SET); // BAT DEN D14  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_14,GPIO\_PIN\_SET); // BAT DEN D15  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_15,GPIO\_PIN\_SET); // BAT DEN D16  HAL\_Delay(500);    HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_0,GPIO\_PIN\_SET); // BAT DEN D17  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_1,GPIO\_PIN\_SET); // BAT DEN D18  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_2,GPIO\_PIN\_SET); // BAT DEN D19  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_3,GPIO\_PIN\_SET); // BAT DEN D20  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_4,GPIO\_PIN\_SET); // BAT DEN D21  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_5,GPIO\_PIN\_SET); // BAT DEN D22  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_6,GPIO\_PIN\_SET); // BAT DEN D23  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_7,GPIO\_PIN\_SET); // BAT DEN D24  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_8,GPIO\_PIN\_SET); // BAT DEN D25  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_9,GPIO\_PIN\_SET); // BAT DEN D26  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_10,GPIO\_PIN\_SET); // BAT DEN D27  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_11,GPIO\_PIN\_SET); // BAT DEN D28  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_12,GPIO\_PIN\_SET); // BAT DEN D29  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_13,GPIO\_PIN\_SET); // BAT DEN D30  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_14,GPIO\_PIN\_SET); // BAT DEN D31  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_15,GPIO\_PIN\_SET); // BAT DEN D32  HAL\_Delay(500);  }  void **tatTheoKimDongHo**(){  **batHetDen**();  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0,GPIO\_PIN\_RESET); // TAT DEN D1  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_1,GPIO\_PIN\_RESET); // TAT DEN D2  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_2,GPIO\_PIN\_RESET); // TAT DEN D3  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_3,GPIO\_PIN\_RESET); // TAT DEN D4  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_4,GPIO\_PIN\_RESET); // TAT DEN D5  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_5,GPIO\_PIN\_RESET); // TAT DEN D6  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_6,GPIO\_PIN\_RESET); // TAT DEN D7  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_7,GPIO\_PIN\_RESET); // TAT DEN D8  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_8,GPIO\_PIN\_RESET); // TAT DEN D9  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_9,GPIO\_PIN\_RESET); // TAT DEN D10  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_10,GPIO\_PIN\_RESET); // TAT DEN D11  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_11,GPIO\_PIN\_RESET); // TAT DEN D12  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_12,GPIO\_PIN\_RESET); // TAT DEN D13  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_13,GPIO\_PIN\_RESET); // TAT DEN D14  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_14,GPIO\_PIN\_RESET); // TAT DEN D15  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_15,GPIO\_PIN\_RESET); // TAT DEN D16  HAL\_Delay(500);    HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_0,GPIO\_PIN\_RESET); // TAT DEN D17  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_1,GPIO\_PIN\_RESET); // TAT DEN D18  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_2,GPIO\_PIN\_RESET); // TAT DEN D19  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_3,GPIO\_PIN\_RESET); // TAT DEN D20  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_4,GPIO\_PIN\_RESET); // TAT DEN D21  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_5,GPIO\_PIN\_RESET); // TAT DEN D22  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_6,GPIO\_PIN\_RESET); // TAT DEN D23  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_7,GPIO\_PIN\_RESET); // TAT DEN D24  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_8,GPIO\_PIN\_RESET); // TAT DEN D25  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_9,GPIO\_PIN\_RESET); // TAT DEN D26  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_10,GPIO\_PIN\_RESET); // TAT DEN D27  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_11,GPIO\_PIN\_RESET); // TAT DEN D28  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_12,GPIO\_PIN\_RESET); // TAT DEN D29  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_13,GPIO\_PIN\_RESET); // TAT DEN D30  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_14,GPIO\_PIN\_RESET); // TAT DEN D31  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_15,GPIO\_PIN\_RESET); // TAT DEN D32  HAL\_Delay(500);  } |

# Bài 14. Nút Bấm - STM32

# 1.Mô tả

Đây là bài thực hành lập trình trên board mạch STM32, trong bài này sẽ điều khiển đèn Led và Motor hoạt động khi nhấn nút bấm, và ngược lại sẽ ngừng hoạt động Led và Motor khi thả nút bấm ra. Nút bấm được đấu vào chân PD0 của STM32, Led được đấu vào chân PD13, và Motor được đấu vào chân PD14 của board mạch STM32.

# 2.Sơ đồ thiết kế



# 3.Đặc điểm linh kiện

* Led: 1 LED (màu đỏ)
* Điện trở: 2 điện trở (10k Ω ,150 Ω)
* Motor: 1 động cơ 1 chiều
* Nút bấm
* Mạch STM32F401VE

# 4.Mã lệnh chính

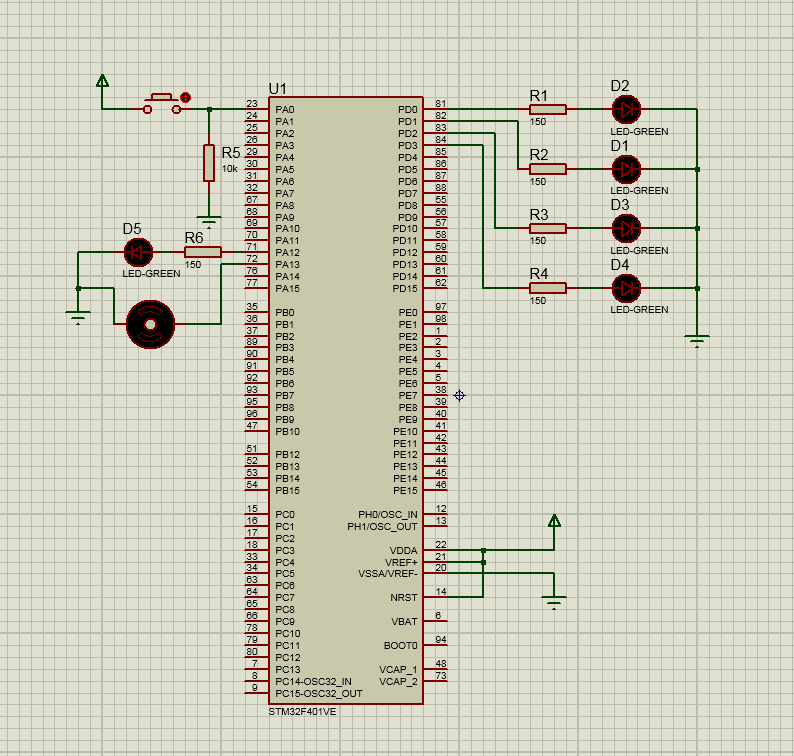
|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  int **main**(void) {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();    **while** (1) {  GPIO\_PinState pin0State = HAL\_GPIO\_ReadPin(GPIOD, GPIO\_PIN\_0);  **if** (pin0State == GPIO\_PIN\_SET)  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13|GPIO\_PIN\_14, GPIO\_PIN\_SET);    **else**  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13|GPIO\_PIN\_14, GPIO\_PIN\_RESET)  }  } |

# Bài 15. Lập Trình Ngắt - STM32

# 1.Mô tả

Đây là bài thực hành lập trình trên board mạch STM32, trong bài này sẽ thực hiện lập trình ngắt. Tiến trình chính sẽ liên tục nhấp nháy 4 Led. Còn trong tiến trình phụ, khi nhấn và thả nút bấm thì sẽ làm cho 1 Led và Motor hoạt động, và nhấn thả nút thêm lần nữa thì sẽ làm Led và Motor đó ngừng hoạt động. 4 Led của tiến trình chính được đấu vào chân PD0 - PD3 của STM32, ở tiến trình phụnút bấm được đấu vào chân PA0, Led đấu vào chân PA12, và Motor được đấu vào chân PA13 của board mạch STM32.

# 2.Sơ đồ thiết kế



# 3.Đặc điểm linh kiện

* Led: 5 LED (màu xanh)
* Điện trở: 6 điện trở (10k Ω ,5 điện trở 150 Ω)
* Motor: 1 động cơ 1 chiều
* Nút bấm
* Mạch STM32F401VE

# 4.Mã lệnh chính

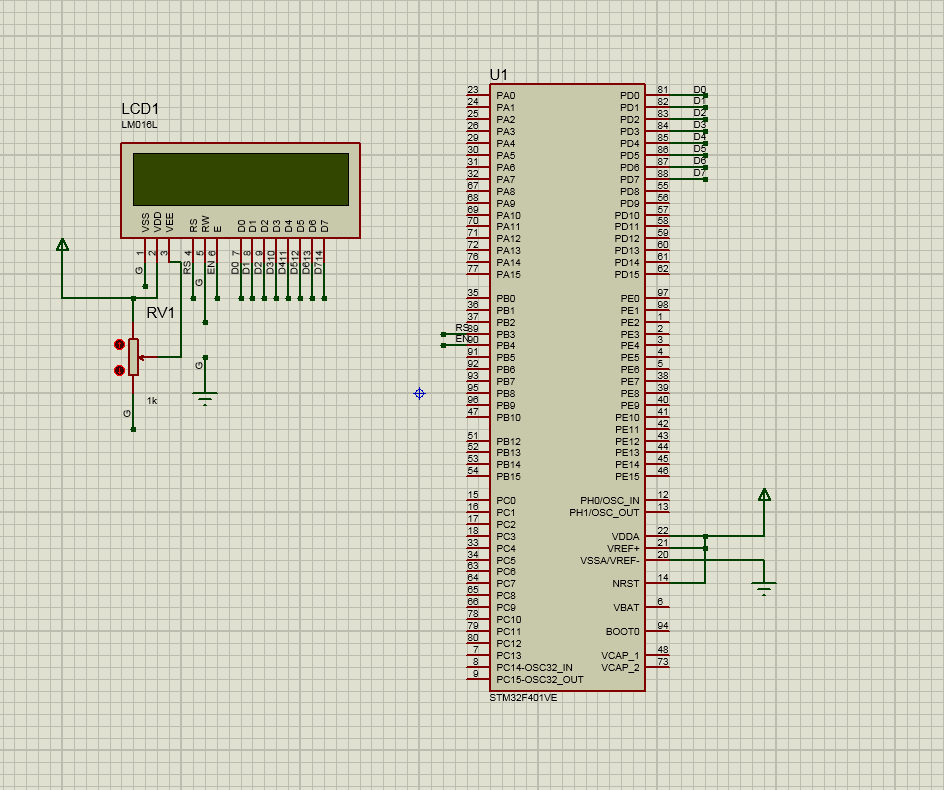
|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  int **main**(void) {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();    **while** (1) {  HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_0);  HAL\_Delay(500);  HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_1);  HAL\_Delay(500);  HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_2);  HAL\_Delay(500);  HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_3);  HAL\_Delay(500);  }  }  void **HAL\_GPIO\_EXTI\_Callback**(uint16\_t GPIO\_Pin) {  **if** (GPIO\_Pin == GPIO\_PIN\_0) {  HAL\_GPIO\_TogglePin(GPIOA, GPIO\_PIN\_12|GPIO\_PIN\_13);  }  } |

# Bài 16. Màn hình LCD - STM32

# 1.Mô tả

Đây là bài thực hành lập trình trên board mạch STM32, trong bài sẽ lập trình để màn hình LCD hiển thị mà không cần dùng thư viện bên ngoài. Chân D0 - D7 của LCD được đấu vào chân PD0 - PD7 của board mạch STM32, chân RS và E của LCD được đấu vào chânPB3 và PB4 của board mạch STM32, Chân RW của LCD được đấu vào đất để mặc định luôn luôn ở chế độ ghi.

# 2.Sơ đồ thiết kế



# 3.Đặc điểm linh kiện

* Màn hình LCD:
* VSS: GND
* VDD: nối nguồn
* VEE: độ tương phản
* RS: lựa chọn thanh ghi (0: chọn thanh ghi lệnh; 1: chọn thanh ghi dữ liệu)
* RW: chọn thanh ghi đọc/viết dữ liệu (0: thanh ghi viết; 1: thanh ghi đọc)
* E: Enable
* D0 – D7: chân truyền dư liệu (8 bit: DB0DB7)
* A: cực dương led nền (0V – 5V)
* K: cực âm led nền (0v)
* Biến trở
* Mạch STM32F401VE

# 4.Mã lệnh chính

|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  // Ham gui data/cmd ra 8 chan cua LCD  void **send8BitLCD**(char D) {  // Dem D and so hoc voi 2^i de biet bit thu i =0 hay khac ko  // tim gia tri cac bit  int b0,b1,b2, b3,b4, b5, b6, b7;  if ((D & 1) == 0) b0=0; // 1== 2^0  else b0 =1;  if ((D & 2) == 0) b1=0; // 2== 2^1  else b1 =1;  if ((D & 4) == 0) b2=0; // 4== 2^2  else b2 =1;  if ((D & 8) == 0) b3=0; // 8== 2^3  else b3 =1;  if ((D & 16) == 0) b4=0; // 16== 2^4  else b4 =1;  if ((D & 32) == 0) b5=0; // 32== 2^5  else b5 =1;  if ((D & 64) == 0) b6=0; // 64== 2^6  else b6 =1;  if ((D & 128) == 0) b7=0; // 128== 2^7  else b7 =1;  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0, b0);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_1, b1);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_2, b2);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_3, b3);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_4, b4);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_5, b5);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_6, b6);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_7, b7);  }  // Ham gui lenh  void **sendCMD2LCD**(char cmd) {  //B1. Done  //B2. Dat chan RS =0, de noi rang cmd là lenh  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_3, GPIO\_PIN\_RESET);  //B3. Gui 8 bit CMD vao 8 pin  send8BitLCD(cmd);  //B4. Enable cho cmd-->lcd  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_4, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_4, GPIO\_PIN\_SET);  //  HAL\_Delay(1);  }  // Ham gui ky tu hien thi  void **sendChar2LCD**(char Char) {  //B1. Done  //B2. Dat chan RS =1,  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_3, GPIO\_PIN\_SET);  //B3. Gui 8 bit CMD vao 8 pin  send8BitLCD(Char);  //B4. Enable cho cmd-->lcd  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_4, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_4, GPIO\_PIN\_SET);  //  HAL\_Delay(1);  }  // Chuyen so thanh ky tu  int **itoa**(int value,char \*ptr) {  int count=0,temp;  if(ptr==NULL)  return 0;  if(value==0) {  \*ptr='0';  return 1;  }  if(value<0) {  value\*=(-1);  \*ptr++='-';  count++;  }  for(temp=value;temp>0;temp/=10,ptr++)  \*ptr='\0';  for(temp=value;temp>0;temp/=10) {  \*--ptr=temp%10+'0';  count++;  }  return count;  }  void sendString2LCD(char \*str) {  for (int i=0; str[i] != '\0'; i++) {  sendChar2LCD(str[i]);  }  }  int **main**(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();    // Xoa noi dung tren LCD  sendCMD2LCD(0x01);  // Bat hien thi man hinh, tat con tro  sendCMD2LCD(0x0C);  // Test thu chuoi Hello  sendString2LCD("Hello K61 NTU");  // Xuong dong 2  sendCMD2LCD(0x38); // CHE DO 2 DONG  sendCMD2LCD(0xC0); // TRO XUONG DONG 2  sendString2LCD("PHAN PHUC ");  char str[10];  itoa(333,str);  sendString2LCD(str);  HAL\_Delay(1000);    while (1)  {  }  } |