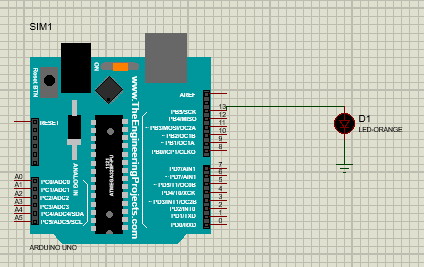
* **Nháy led**

1. Mô tả

Bài này, thực hiện việc tự động nháy led sau khoản thời gian 1 giây. Led được đầu vào cổng 13 của board.

1. Sơ đồ thiết kế



Hình . Led nhấp nháy

1. Đặc điểm linh kiện

* Led
* Arduino uno

1. Mã lệnh chính

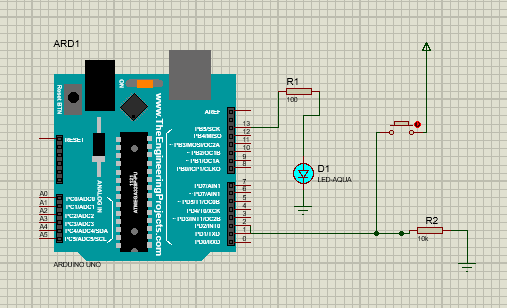
|  |
| --- |
| *void setup()*  *{*  *pinMode(13, OUTPUT);*  *}*  *void loop()*  *{*  *digitalWrite(13, HIGH);*  *delay(1000); // Wait for 1000 millisecond(s)*  *digitalWrite(13, LOW);*  *delay(1000); // Wait for 1000 millisecond(s)*  *}* |

* **Bật tắt led**

1. Mô tả

Bài này thực hiện bằng cách bấm giữ nút thì đèn led sẽ sáng lên, sau khi thả tay ra khỏi nút thì đèn led cũng tắt theo.

1. Sơ đồ thiết kế



Hình . Bật tắt led

1. Đặc điểm linh kiện

* Led
* Nút bấm
* Arduino uno
* Điện trở R1(100 ohm)
* Điện trở R2 (10000 ohm)

1. Mã lệnh chính

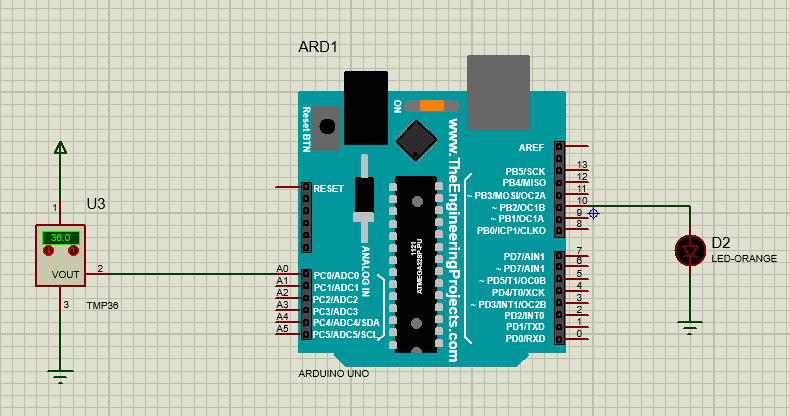
|  |
| --- |
| *int x=0;*  *void setup()*  *{*  *pinMode(2, INPUT);*  *pinMode(13, OUTPUT);*  *}*  *void loop()*  *{*  *x = digitalRead(2);*  *if(x==HIGH){*  *digitalWrite(13,HIGH);*  *}*  *else{*  *digitalWrite(13,LOW);*  *}*  *delay(1000);*  *}* |

* **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. Sơ đồ thiết kế



Hình . . Bật đèn Led với cảm biến nhiệt độ TMP36

1. Đặc điểm linh kiện

* Arduino uno
* Đèn led
* TMP36

1. Mã lệnh chính

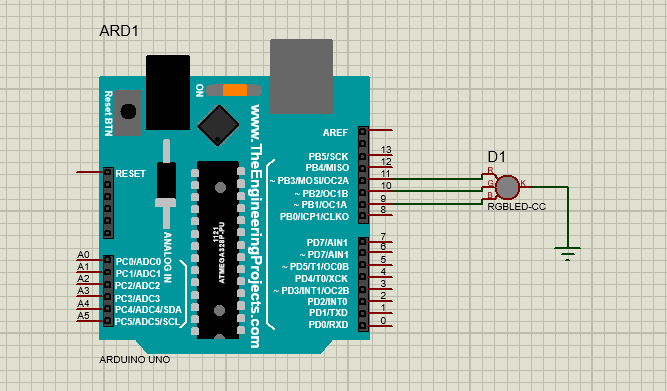
|  |
| --- |
| *void setup()*  *{*  *Serial.begin(96000);*  *}*  *void loop()*  *{*  *//Doc gia tri cam bien*  *int giatri = analogRead(A0);*  *int nhietDo = map(giatri,20,358,-40,125);*  *//Serial.print(nhietDo);*  *if(nhietDo>37) digitalWrite(10,HIGH);*  *else digitalWrite(10,LOW);*    *delay(1000);*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

* **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 . Chạy đèn Led RGB

1. Đặc điểm linh kiện

* Arduino uno
* Led RGB

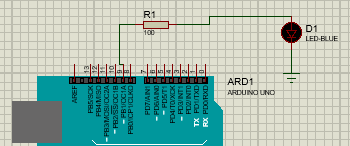
1. Mã lệnh chính

|  |
| --- |
| *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);*  *}* |

* **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 Led sáng dần

3. Đặc điểm linh kiện

- Arduino UNO

- Led

- Điện trở

4. Mã lệnh chính

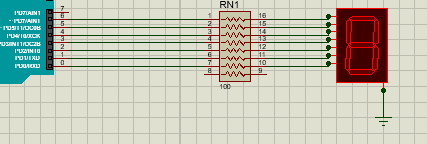
|  |
| --- |
| *// C++ code*  *//*  *int brightness = 0;*  *void setup()*  *{*  *pinMode(9, OUTPUT);*  *}*  *void loop()*  *{*  *for( brightness =0;brightness<=255;brightness +=5)*  *{*  *analogWrite(9,brightness);*  *delay(30);*  *}*  *for( brightness =0;brightness<=255;brightness -=5)*  *{*  *analogWrite(9,brightness);*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

* **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

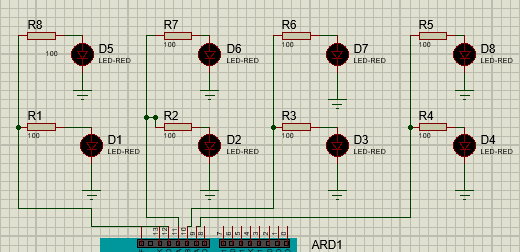
|  |
| --- |
| *int a=6, b=5, c=4,d=3,e=2, f=1,g=0;*  *void setup() {*  *pinMode(a,OUTPUT); pinMode(b,OUTPUT); pinMode(c,OUTPUT); pinMode(d,OUTPUT);*  *pinMode(e,OUTPUT); pinMode(f,OUTPUT); pinMode(g,OUTPUT);*  *}*  *void KHONG(){*  *digitalWrite(a,HIGH); digitalWrite(b,HIGH);*  *digitalWrite(c,HIGH);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

* **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

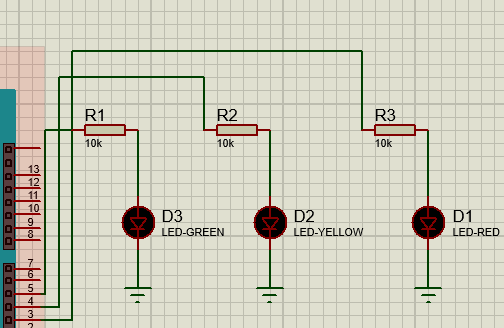
|  |
| --- |
| *void setup(){*  *pinMode(12, OUTPUT);*  *pinMode(11, OUTPUT);*  *pinMode(10, OUTPUT);*  *pinMode(9, OUTPUT);*    *pinMode(13, OUTPUT);*  *pinMode(7, OUTPUT);*  *}*  *void piscar(){*  *digitalWrite(12, HIGH);*  *digitalWrite(11, LOW);*  *digitalWrite(10, LOW);*  *digitalWrite(9, LOW);*  *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);*  *}* |

* **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

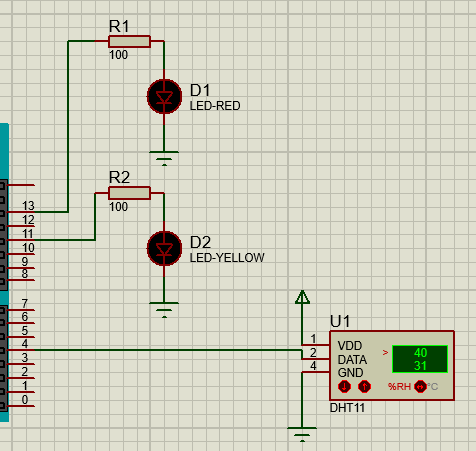
|  |
| --- |
| *// C++ code*  *//*  *int dem=0;*  *void setup()*  *{*  *pinMode(5, OUTPUT);*  *pinMode(4, OUTPUT);*  *pinMode(3, OUTPUT);*  *}*  *void loop()*  *{*  *dem += 1;*  *delay(1000);*  *if(dem == 65){*  *dem=0;*  *}*  *if(dem<=30){*  *digitalWrite(5, HIGH);*  *digitalWrite(4, LOW);*  *digitalWrite(3,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

* **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

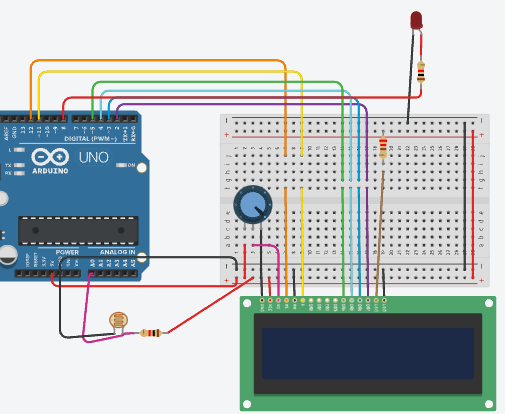
|  |
| --- |
| *#include "DHT.h" //khai bao thu vien cho DTH*  *const int DHTPIN = 4; // chan vao cua cam bien DHT11*  *const int DHTTYPE = DHT11;*  *DHT dht(DHTPIN, DHTTYPE);*  *void setup() {*  *dht.begin();*  *}*  *void loop() {*  *int h = dht.readHumidity(); // do am*  *float t = dht.readTemperature();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

* **Bài 1 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

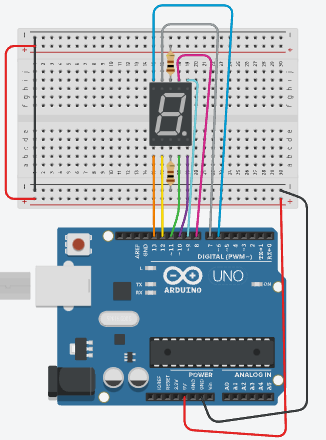
|  |
| --- |
| *#include <LiquidCrystal.h>*  *LiquidCrystal lcd(12, 11, 5, 4, 3, 2);*  *int LED = 8;*  *int LIGHT = 0;*  *int val =0;*  *void setup() {*    *lcd.begin(16, 2);*  *lcd.setCursor(4,0);*  *pinMode (LED, OUTPUT);*  *Serial.begin(9600);*    *lcd.print("DO SANG");*  *}*  *void loop() {*    *lcd.setCursor(0, 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 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

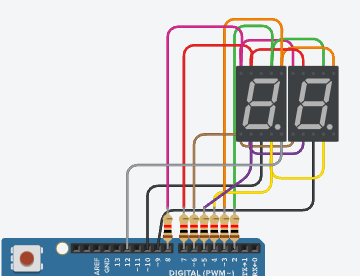
|  |
| --- |
| *int E = 13;*  *int D = 12;*  *int C = 11;*  *int DP = 10;*  *int B = 9;*  *int A = 8;*  *int F = 7;*  *int G = 6;*  *void zero()*  *{*  *digitalWrite(E,LOW);*  *digitalWrite(D,LOW);*  *digitalWrite(C,LOW);*  *digitalWrite(DP,HIGH);*  *digitalWrite(B,LOW);*  *digitalWrite(A,LOW);*  *digitalWrite(F,LOW);*  *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 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

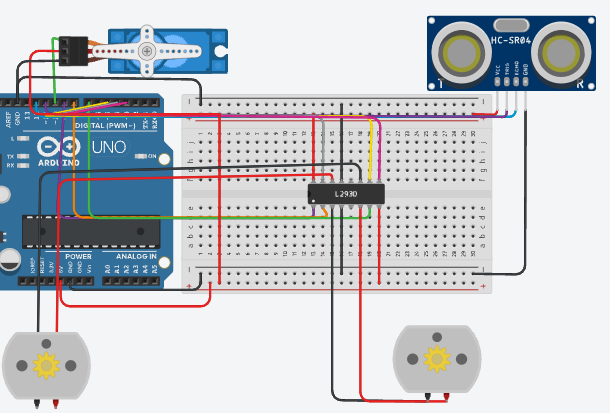
1. Mã lệnh chính

|  |
| --- |
| *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 3: 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

1. 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)

1. Mã lệnh chính

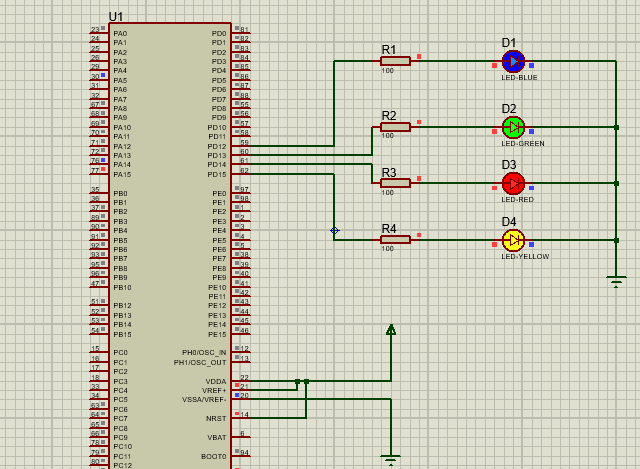
|  |
| --- |
| *// Motor A*  *int enA = 9;*  *int in1 = 8;*  *int in2 = 7;*  *// Motor B*  *int enB = 3;*  *int in3 = 5;*  *int in4 = 4;*  *//*  *const int trig = 11; //chân trig của HC-SR04*  *const int echo = 12; //chân echo của HC-SR04*  *#include <Servo.h>*  *Servo servo\_10;*  *void setup() {*    *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();*  *}*  *}* |

* **STM nháy led**

1. Mô tả

Bài này thực hiện bằng cách cho 4 đèn led vào 4 chân PD12,13,14,15 của mạch STM32F401VE. Led sẽ lần lượt nháy trong 1 giây.

1. Sơ đồ thiết kế



1. Đặc điểm linh kiện

* 4 đèn led
* 1 mạch STM32F401VE

1. Mã lệnh chính

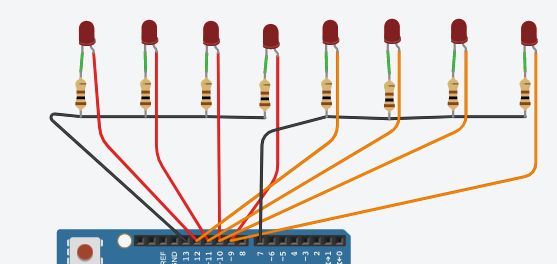
|  |
| --- |
| *#include "main.h"*  *void SystemClock\_Config(void);*  *static void MX\_GPIO\_Init(void);*  *int main(void)*  *{*  *HAL\_Init();*  *SystemClock\_Config();*  *MX\_GPIO\_Init();*    *while (1)*  *{*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_12);*  *HAL\_Delay(1000);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_13);*  *HAL\_Delay(1000);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_14);*  *HAL\_Delay(1000);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_15);*  *HAL\_Delay(1000);*    *}*    *}*  *void SystemClock\_Config(void)*  *{*  *RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};*  *RCC\_ClkInitTypeDef RCC\_ClkInitStruct = {0};*    *\_\_HAL\_RCC\_PWR\_CLK\_ENABLE();*  *\_\_HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR\_VOLTAGE\_SCALE2);*    *RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSI;*  *RCC\_OscInitStruct.HSIState = RCC\_HSI\_ON;*  *RCC\_OscInitStruct.HSICalibrationValue = RCC\_HSICALIBRATION\_DEFAULT;*  *RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_ON;*  *RCC\_OscInitStruct.PLL.PLLSource = RCC\_PLLSOURCE\_HSI;*  *RCC\_OscInitStruct.PLL.PLLM = 8;*  *RCC\_OscInitStruct.PLL.PLLN = 64;*  *RCC\_OscInitStruct.PLL.PLLP = RCC\_PLLP\_DIV2;*  *RCC\_OscInitStruct.PLL.PLLQ = 4;*  *if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)*  *{*  *Error\_Handler();*  *}*    *RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK*  *|RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;*  *RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_PLLCLK;*  *RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV4;*  *RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV2;*  *RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1;*  *if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_0) != HAL\_OK)*  *{*  *Error\_Handler();*  *}*  *}*  *static void MX\_GPIO\_Init(void)*  *{*  *GPIO\_InitTypeDef GPIO\_InitStruct = {0};*    *\_\_HAL\_RCC\_GPIOH\_CLK\_ENABLE();*  *\_\_HAL\_RCC\_GPIOD\_CLK\_ENABLE();*  *\_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE();*    *HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15, GPIO\_PIN\_RESET);*    *GPIO\_InitStruct.Pin = GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15;*  *GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*  *GPIO\_InitStruct.Pull = GPIO\_NOPULL;*  *GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;*  *HAL\_GPIO\_Init(GPIOD, &GPIO\_InitStruct);*  *}*  *void Error\_Handler(void)*  *{*    *\_\_disable\_irq();*  *while (1)*  *{*  *}*    *}*  *#ifdef USE\_FULL\_ASSERT*  *void assert\_failed(uint8\_t \*file, uint32\_t line)*  *{*    *}* |

* **8 Led**

1. Mô tả

Bài này thực hiện bằng cách gắn 8 chiếc đèn led vào 4 chân 9,10,11 và 12. Đầu tiên đèn led sẽ nháy đôi đi từ trái sang phải, sau đó sáng đèn led ở vị trí chẳn, cuối cùng nháy cả 8 đèn 3 lần.

1. Sơ đồ thiết kế



1. Đặc điểm linh kiện

* Arduino uno
* 8 led
* 8 Điện trở

1. Mã lệnh chính

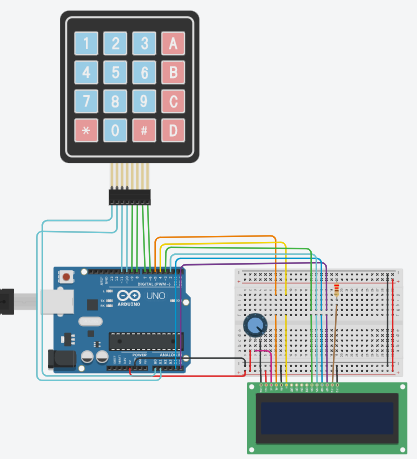
|  |
| --- |
| *void setup(){*  *pinMode(12, OUTPUT);*  *pinMode(11, OUTPUT);*  *pinMode(10, OUTPUT);*  *pinMode(9, OUTPUT);*    *pinMode(13, OUTPUT);*  *pinMode(7, OUTPUT);*  *}*  *void piscar(){*  *digitalWrite(12, HIGH);*  *digitalWrite(11, LOW);*  *digitalWrite(10, LOW);*  *digitalWrite(9, LOW);*  *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(2000);*  *SangChan();*  *delay(2000);*  *ChopNhay();*  *delay(2000);*  *}* |

* **Keypad kết hợp với LCD**

1. Mô tả

Bài này thực hiện bằng cách nối keypad 4x4 vào Adruino kết hợp với LCD. Khi Keypad nhấn vào phím gì thì LCD sẽ hiện ra chữ hoặc số đó.

1. Sơ đồ thiết kế



1. Đặc điểm linh kiện

* Arduino kết hợp với LCD ( này là do tinkercad thiết kế sẳn, giúp hiển thị chữ hoặc câu nói muốn in ra)
* Keypad 4x4 (là một thiết bị nhập chứa các nút bấm cho phép người dùng nhập các chữ số, chữ cái hoặc ký tự điều khiển)

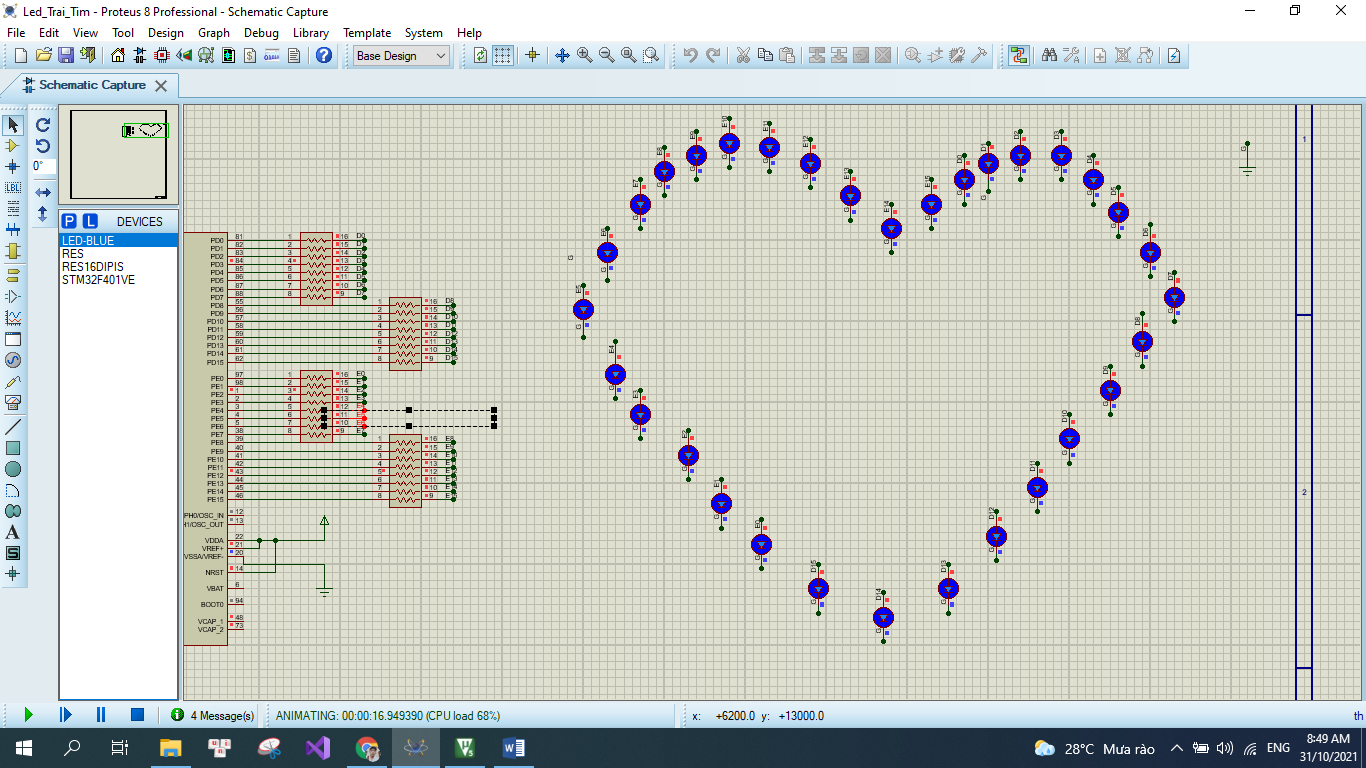
1. Mã lệnh chính

|  |
| --- |
| *#include <Keypad.h>*  *#include <LiquidCrystal.h>*  *LiquidCrystal lcd(5, 4, 3, 2, A4, A5);*  *const byte ROWS = 4;*  *const byte COLS = 4;*  *char keys[ROWS][COLS] = {*  *{'1','2','3','A'},*  *{'4','5','6','B'},*  *{'7','8','9','C'},*  *{'\*','0','#','D'}*  *};*  *byte rowPins[ROWS] = {A0, A1, 11, 10};*  *byte colPins[COLS] = {9, 8, 7, 6};*  *int LCDRow = 0;*  *Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );*  *void setup(){*  *Serial.begin(9600);*  *lcd.begin(16, 2);*  *lcd.setCursor(LCDRow, 0);*  *}*    *void loop(){*  *char key = keypad.getKey();*    *if (key){*  *Serial.println(key);*  *lcd.print(key);*  *lcd.setCursor (++LCDRow, 0);*  *}*  *}* |

* **Led hình trái tim**

1. Mô tả

Bài này thực kiện bằng cách xếp 30 đèn led thành hình trái tim kết nối với mạch STM32F401VE. Đầu tiên những chiếc đèn led sẽ sáng theo cặp từ trên xuống dưới tiếp đó sẽ tắt dần theo chiều kim đồng hồ rồi sáng theo cặp từ dưới lên, sau cùng là chớp nháy 3 lần và lặp lại vô thời hạn

1. Sơ đồ thiết kế
2. Đặc điểm linh kiện

* STM32F401VE
* Điện trở 8 chân RES16DIPIS
* 30 đèn led

1. Mã lệnh chính

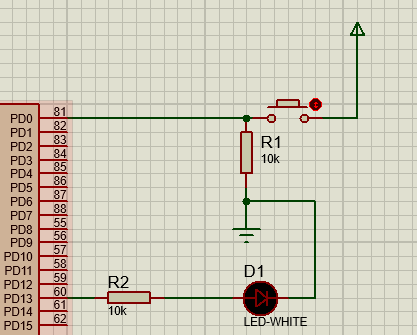
|  |
| --- |
| *#include "main.h"*  *void SystemClock\_Config(void);*  *static void MX\_GPIO\_Init(void);*  *void lednhapnhay(){*    *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_0);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_1);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_2);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_3);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_4);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_5);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_6);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_7);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_8);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_9);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_10);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_11);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_12);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_13);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_14);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_15);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_0);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_1);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_2);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_3);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_4);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_5);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_6);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_7);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_8);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_9);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_10);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_11);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_12);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_13);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_14);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_15);*  *HAL\_Delay(100);*  *}*  *void ledsangchieuduong(){*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_14);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_15);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_0);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_1);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_2);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_3);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_4);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_5);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_6);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_7);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_8);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_9);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_10);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_11);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_12);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_13);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_14);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_15);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_0);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_1);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_2);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_3);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_4);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_5);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_6);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_7);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_8);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_9);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_10);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_11);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_12);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_13);*  *HAL\_Delay(100);*  *}*  *void ledsanghaibentrenxuong(){*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_14);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_15);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_13);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_0);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_12);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_1);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_11);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_2);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_10);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_3);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_9);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_4);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_8);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_5);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_7);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_6);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_6);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_7);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_5);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_8);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_4);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_9);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_3);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_10);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_2);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_11);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_1);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_12);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_0);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_15);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_13);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_14);*  *HAL\_Delay(100);*    *}*  *void ledsanghaibenduoilen(){*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_14);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_15);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_13);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_12);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_0);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_11);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_1);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_10);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_2);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_9);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_3);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_8);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_4);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_7);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_5);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_6);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_6);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_5);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_7);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_4);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_8);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_3);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_9);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_2);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_10);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_1);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_11);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOD, GPIO\_PIN\_0);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_12);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_15);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_13);*  *HAL\_Delay(100);*  *HAL\_GPIO\_TogglePin( GPIOE, GPIO\_PIN\_14);*  *HAL\_Delay(100);*  *}*  *int main(void)*  *{*  *HAL\_Init();*  *SystemClock\_Config();*  *MX\_GPIO\_Init();*  *while (1)*  *{*    *ledsanghaibentrenxuong();*  *HAL\_Delay(100);*  *ledsangchieuduong();*  *HAL\_Delay(100);*  *ledsanghaibenduoilen();*  *HAL\_Delay(100);*  *lednhapnhay();*  *HAL\_Delay(100);*  *lednhapnhay();*  *HAL\_Delay(100);*  *lednhapnhay();*  *HAL\_Delay(100);*    *}*    *}*  *void SystemClock\_Config(void)*  *{*  *RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};*  *RCC\_ClkInitTypeDef RCC\_ClkInitStruct = {0};*    *\_\_HAL\_RCC\_PWR\_CLK\_ENABLE();*  *\_\_HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR\_VOLTAGE\_SCALE2);*      *RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSI;*  *RCC\_OscInitStruct.HSIState = RCC\_HSI\_ON;*  *RCC\_OscInitStruct.HSICalibrationValue = RCC\_HSICALIBRATION\_DEFAULT;*  *RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE;*  *if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)*  *{*  *Error\_Handler();*  *}*  */\*\* Initializes the CPU, AHB and APB buses clocks*  *\*/*  *RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK*  *|RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;*  *RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_HSI;*  *RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1;*  *RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV1;*  *RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1;*  *if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_0) != HAL\_OK)*  *{*  *Error\_Handler();*  *}*  *}*  *static void MX\_GPIO\_Init(void)*  *{*  *GPIO\_InitTypeDef GPIO\_InitStruct = {0};*    *\_\_HAL\_RCC\_GPIOE\_CLK\_ENABLE();*  *\_\_HAL\_RCC\_GPIOH\_CLK\_ENABLE();*  *\_\_HAL\_RCC\_GPIOD\_CLK\_ENABLE();*  *\_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE();*  *HAL\_GPIO\_WritePin(GPIOE, GPIO\_PIN\_2|GPIO\_PIN\_3|GPIO\_PIN\_4|GPIO\_PIN\_5*  *|GPIO\_PIN\_6|GPIO\_PIN\_7|GPIO\_PIN\_8|GPIO\_PIN\_9*  *|GPIO\_PIN\_10|GPIO\_PIN\_11|GPIO\_PIN\_12|GPIO\_PIN\_13*  *|GPIO\_PIN\_14|GPIO\_PIN\_15|GPIO\_PIN\_0|GPIO\_PIN\_1, GPIO\_PIN\_RESET);*    *HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_8|GPIO\_PIN\_9|GPIO\_PIN\_10|GPIO\_PIN\_11*  *|GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15*  *|GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3*  *|GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7, GPIO\_PIN\_RESET);*  *GPIO\_InitStruct.Pin = GPIO\_PIN\_2|GPIO\_PIN\_3|GPIO\_PIN\_4|GPIO\_PIN\_5*  *|GPIO\_PIN\_6|GPIO\_PIN\_7|GPIO\_PIN\_8|GPIO\_PIN\_9*  *|GPIO\_PIN\_10|GPIO\_PIN\_11|GPIO\_PIN\_12|GPIO\_PIN\_13*  *|GPIO\_PIN\_14|GPIO\_PIN\_15|GPIO\_PIN\_0|GPIO\_PIN\_1;*  *GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*  *GPIO\_InitStruct.Pull = GPIO\_NOPULL;*  *GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;*  *HAL\_GPIO\_Init(GPIOE, &GPIO\_InitStruct);*  *GPIO\_InitStruct.Pin = GPIO\_PIN\_8|GPIO\_PIN\_9|GPIO\_PIN\_10|GPIO\_PIN\_11*  *|GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15*  *|GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3*  *|GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7;*  *GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*  *GPIO\_InitStruct.Pull = GPIO\_NOPULL;*  *GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;*  *HAL\_GPIO\_Init(GPIOD, &GPIO\_InitStruct);*  *}*  *void Error\_Handler(void)*  *{*    *\_\_disable\_irq();*  *while (1)*  *{*  *}*    *}*  *void assert\_failed(uint8\_t \*file, uint32\_t line)*  *{*    *}* |

* **Nút bấm sáng led STM32**

1. Mô tả

Bài này thực hiện bằng cách gán nút bấm vào chân PD0 và đèn led vào chân PD13. Khi bấm vào nút thì đèn led sẽ sáng, khi thả ra đèn led sẽ tắt đi.

1. Sơ đồ thiết kế



1. Đặc điểm linh kiện

* STM32F401VE
* Nút bấm
* Led red
* 2 điện trở

1. Mã lệnh chính

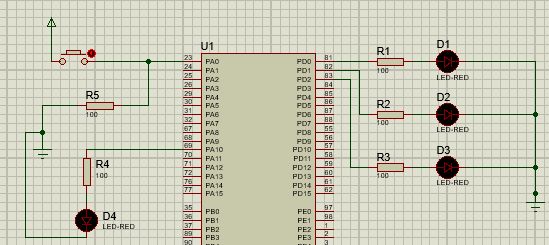
|  |
| --- |
| *#include "main.h"*  *void SystemClock\_Config(void);*  *static void MX\_GPIO\_Init(void);*  *int main(void)*  *{*    *HAL\_Init();*  *SystemClock\_Config();*  *MX\_GPIO\_Init();*  *while (1)*  *{*  *GPIO\_PinState pin0State = HAL\_GPIO\_ReadPin(GPIOD, GPIO\_PIN\_0);*  *if(pin0State == GPIO\_PIN\_SET)*  *{*  *HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);*  *}*    *else*  *{*  *HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);*  *}*  *}*  */\* USER CODE END 3 \*/*  *}*  *void SystemClock\_Config(void)*  *{*  *RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};*  *RCC\_ClkInitTypeDef RCC\_ClkInitStruct = {0};*  */\*\* Configure the main internal regulator output voltage*  *\*/*  *\_\_HAL\_RCC\_PWR\_CLK\_ENABLE();*  *\_\_HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR\_VOLTAGE\_SCALE2);*  *RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSI;*  *RCC\_OscInitStruct.HSIState = RCC\_HSI\_ON;*  *RCC\_OscInitStruct.HSICalibrationValue = RCC\_HSICALIBRATION\_DEFAULT;*  *RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE;*  *if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)*  *{*  *Error\_Handler();*  *}*  */\*\* Initializes the CPU, AHB and APB buses clocks*  *\*/*  *RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK*  *|RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;*  *RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_HSI;*  *RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV2;*  *RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV1;*  *RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1;*  *if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_0) != HAL\_OK)*  *{*  *Error\_Handler();*  *}*  *}*  *static void MX\_GPIO\_Init(void)*  *{*  *GPIO\_InitTypeDef GPIO\_InitStruct = {0};*  */\* GPIO Ports Clock Enable \*/*  *\_\_HAL\_RCC\_GPIOH\_CLK\_ENABLE();*  *\_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE();*  *\_\_HAL\_RCC\_GPIOD\_CLK\_ENABLE();*  */\*Configure GPIO pin Output Level \*/*  *HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13|GPIO\_PIN\_0, GPIO\_PIN\_RESET);*  */\*Configure GPIO pin : PA0 \*/*  *GPIO\_InitStruct.Pin = GPIO\_PIN\_0;*  *GPIO\_InitStruct.Mode = GPIO\_MODE\_INPUT;*  *GPIO\_InitStruct.Pull = GPIO\_NOPULL;*  *HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);*  */\*Configure GPIO pins : PD13 PD0 \*/*  *GPIO\_InitStruct.Pin = GPIO\_PIN\_13|GPIO\_PIN\_0;*  *GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*  *GPIO\_InitStruct.Pull = GPIO\_NOPULL;*  *GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;*  *HAL\_GPIO\_Init(GPIOD, &GPIO\_InitStruct);*  *}*  *void Error\_Handler(void)*  *{*    *\_\_disable\_irq();*  *while (1)*  *{*  *}*    *}*  *#ifdef USE\_FULL\_ASSERT*  *void assert\_failed(uint8\_t \*file, uint32\_t line)*  *{*  *}*  *#endif /\* USE\_FULL\_ASSERT \*/* |

* **Nháy led và nút bấm trên thanh STM32**

1. Mô tả

Bài này thực hiện bằng cách cho lần lượt 3 chiếc đèn led gắn vào chân PD0,PD1,PD2, gắn nút bấm vào chân PA0 và đèn led vào chân PA10. Các đèn led ở các chân PD cứ mỗi 1 giây sẽ chớp tắt một lần. Đèn led ở chân PA thì cần bấm nút để mở và bấm nút để tắt nó đi.

1. Sơ đồ thiết kế



1. Đặc điểm linh kiện

* STM32F401VE
* 4 đèn Led
* 5 điện trở
* 1 nút bấm

1. 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|GPIO\_PIN\_1|GPIO\_PIN\_2);*  *HAL\_Delay(1000);*    *}*  *}*  *void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)*  *{*  *if(GPIO\_Pin == GPIO\_PIN\_0)*  *{*  *HAL\_GPIO\_TogglePin(GPIOA, GPIO\_PIN\_10);*  *}*  *}*  */\*\**  *\* @brief System Clock Configuration*  *\* @retval None*  *\*/*  *void SystemClock\_Config(void)*  *{*  *RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};*  *RCC\_ClkInitTypeDef RCC\_ClkInitStruct = {0};*  */\*\* Configure the main internal regulator output voltage*  *\*/*  *\_\_HAL\_RCC\_PWR\_CLK\_ENABLE();*  *\_\_HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR\_VOLTAGE\_SCALE2);*  */\*\* Initializes the RCC Oscillators according to the specified parameters*  *\* in the RCC\_OscInitTypeDef structure.*  *\*/*  *RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSI;*  *RCC\_OscInitStruct.HSIState = RCC\_HSI\_ON;*  *RCC\_OscInitStruct.HSICalibrationValue = RCC\_HSICALIBRATION\_DEFAULT;*  *RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE;*  *if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)*  *{*  *Error\_Handler();*  *}*  */\*\* Initializes the CPU, AHB and APB buses clocks*  *\*/*  *RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK*  *|RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;*  *RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_HSI;*  *RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1;*  *RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV1;*  *RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1;*  *if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_0) != HAL\_OK)*  *{*  *Error\_Handler();*  *}*  *}*  */\*\**  *\* @brief GPIO Initialization Function*  *\* @param None*  *\* @retval None*  *\*/*  *static void MX\_GPIO\_Init(void)*  *{*  *GPIO\_InitTypeDef GPIO\_InitStruct = {0};*  */\* GPIO Ports Clock Enable \*/*  *\_\_HAL\_RCC\_GPIOH\_CLK\_ENABLE();*  *\_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE();*  *\_\_HAL\_RCC\_GPIOD\_CLK\_ENABLE();*  */\*Configure GPIO pin Output Level \*/*  *HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_10, GPIO\_PIN\_RESET);*  */\*Configure GPIO pin Output Level \*/*  *HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, GPIO\_PIN\_RESET);*  */\*Configure GPIO pin : PA0 \*/*  *GPIO\_InitStruct.Pin = GPIO\_PIN\_0;*  *GPIO\_InitStruct.Mode = GPIO\_MODE\_IT\_RISING;*  *GPIO\_InitStruct.Pull = GPIO\_NOPULL;*  *HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);*  */\*Configure GPIO pin : PA10 \*/*  *GPIO\_InitStruct.Pin = GPIO\_PIN\_10;*  *GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*  *GPIO\_InitStruct.Pull = GPIO\_NOPULL;*  *GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;*  *HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);*  */\*Configure GPIO pins : PD0 PD1 PD2 \*/*  *GPIO\_InitStruct.Pin = GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2;*  *GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*  *GPIO\_InitStruct.Pull = GPIO\_NOPULL;*  *GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;*  *HAL\_GPIO\_Init(GPIOD, &GPIO\_InitStruct);*  */\* EXTI interrupt init\*/*  *HAL\_NVIC\_SetPriority(EXTI0\_IRQn, 0, 0);*  *HAL\_NVIC\_EnableIRQ(EXTI0\_IRQn);*  *}*  */\* USER CODE BEGIN 4 \*/*  */\* USER CODE END 4 \*/*  */\*\**  *\* @brief This function is executed in case of error occurrence.*  *\* @retval None*  *\*/*  *void Error\_Handler(void)*  *{*  */\* USER CODE BEGIN Error\_Handler\_Debug \*/*  */\* User can add his own implementation to report the HAL error return state \*/*  *\_\_disable\_irq();*  *while (1)*  *{*  *}*  */\* USER CODE END Error\_Handler\_Debug \*/*  *}*  *#ifdef USE\_FULL\_ASSERT*  */\*\**  *\* @brief Reports the name of the source file and the source line number*  *\* where the assert\_param error has occurred.*  *\* @param file: pointer to the source file name*  *\* @param line: assert\_param error line source number*  *\* @retval None*  *\*/*  *void assert\_failed(uint8\_t \*file, uint32\_t line)*  *{*  */\* USER CODE BEGIN 6 \*/*  */\* User can add his own implementation to report the file name and line number,*  *ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) \*/*  */\* USER CODE END 6 \*/*  *}*  *#endif /\* USE\_FULL\_ASSERT \*/*  */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* (C) COPYRIGHT STMicroelectronics \*\*\*\*\*END OF FILE\*\*\*\*/* |