**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\*\*\*\*\*\*\*\*\*\*\*\*🙢 🕮 🙠\*\*\*\*\*\*\*\*\*\*\*\***



**HỌC PHẦN: LẬP TRÌNH NHÚNG**

**GIẢNG VIÊN : MAI CƯỜNG THỌ**

MSSV: 61131788

Lớp: 61.CNTT1

BÀI BÁO CÁO THỰC HÀNH LẬP TRÌNH ARDUINO

sinh viên: phạm minh hoàng

October 3, 2021

**MỤC LỤC**

[Bài 1. Nháy Led 1](#_Toc89932170)

[Bài 2. Đèn sáng khi bấm nút 2](#_Toc89932171)

[Bài 3. làm việc với cảm biến nhiệt độ 3](#_Toc89932172)

[Bài 4. Led sáng dần 4](#_Toc89932173)

[Bài 5. Thực hành với Led RGB 5](#_Toc89932174)

[Bài 6. Điều khiển độ sáng của Led qua chiết áp 6](#_Toc89932175)

[Bài 7. Led 7 đoạn (7 segment) 7](#_Toc89932176)

[Bài 8. Sáng 8 Led theo 1 trình tự, sử dụng IC 74C595 8](#_Toc89932177)

[Bài 9. Ma trận led 9](#_Toc89932178)

[Bài 10. Động cơ DC 12](#_Toc89932179)

[Bài 11. Sáng led lần lượt – STM32 14](#_Toc89932180)

[Bài 12. Lập trình Ngắt ngoài – STM32 15](#_Toc89932181)

[Bài 13. Led Trái Tim – STM32 16](#_Toc89932182)

[Bài 14. Nút bấm – STM32 20](#_Toc89932183)

[Bài 15. Đọc tín hiệu tương tự số từ chiết áp – STM32 22](#_Toc89932184)

[Bài 16. LCD – STM32 23](#_Toc89932185)

[Bài 1. Kiểm tra giữa kỳ 25](#_Toc89932186)

[Bài 2A. Kiểm tra giữa kỳ 27](#_Toc89932187)

[Bài 2B. Kiểm tra giữa kỳ 28](#_Toc89932188)

[Bài 3. Kiểm tra giữa kỳ 31](#_Toc89932189)

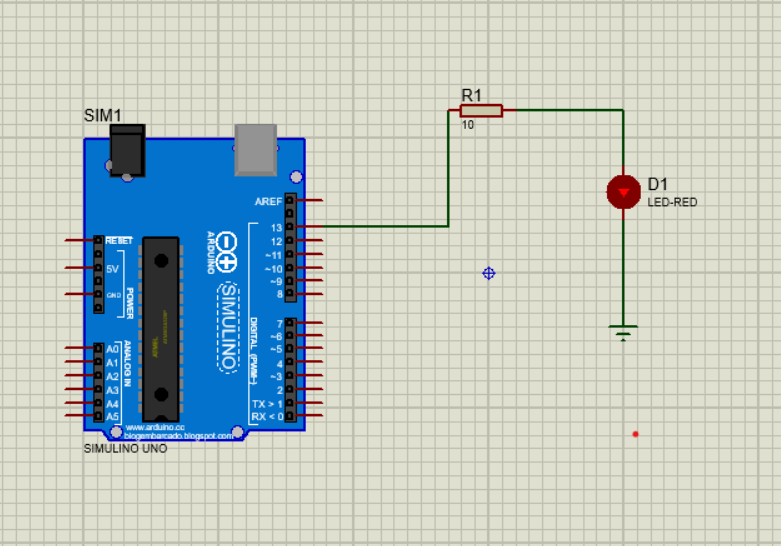
[Cảm biến vật cản hồng ngoại, điều khiển servo 34](#_Toc89932190)

## Bài 1. Nháy Led

### Mô tả:

Bài này, hệ thống được thiết kế để thực hiện việc tự động bật/tắt đèn led sau khoảng thời gian delay 1 giây. Đèn led được đấu vào cổng số 13 thông qua điện trở của Board mạch arduino.

### Sơ đồ thiết kế:



Hình . Sơ đồ kết nối của hệ thống

### Đặc điểm linh kiện:

* + 1 mạch arduino uno
  + 1 đèn Led:
  + Điện trở : R1(100 Ω)

### Mã lệnh chính:

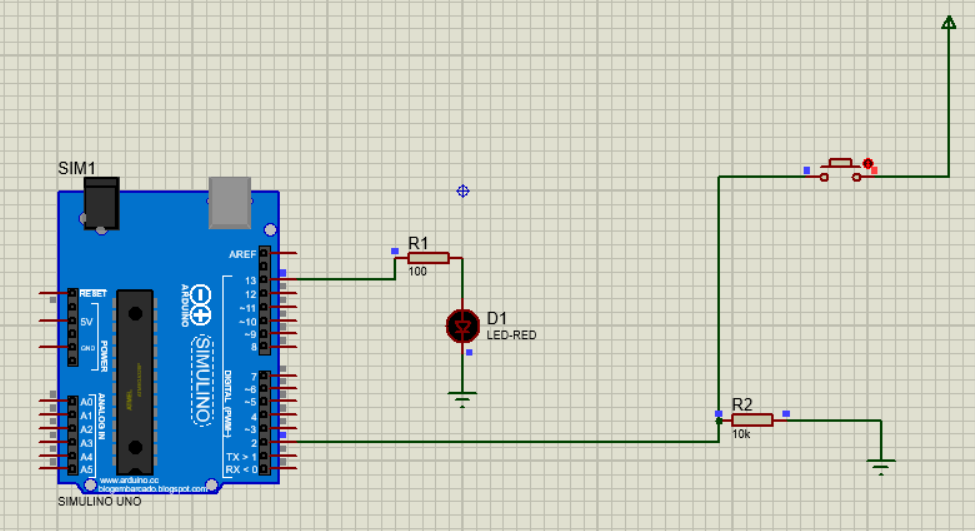
|  |
| --- |
| void setup() {  pinMode(10, OUTPUT);  }  void loop() {  digitalWrite(10, HIGH); //bật led  delay(1000); //để led sáng 1s  digitalWrite(10, LOW); //tắt led  delay(1000  } |

## Bài 2. Đèn sáng khi bấm nút

### Mô tả

Hệ thống thực hiện bật Led thông qua nút bấm, đèn được nối vào Arduino ở cổng số 13. Đầu ra nút bấm được nối vào pin 2. Đèn sáng khi nút bấm và ngược lại.

### Sơ đồ thiết kế:



Hình . Sơ đồ thiết kế

### Đặc điểm linh kiện:

* 1 Đèn led.
* 2 Điện trở: R1(100 Ω), R2 (10k Ω).
* 1 nút bấm
* 1 mạch arduino uno

### Mã lệnh chính:

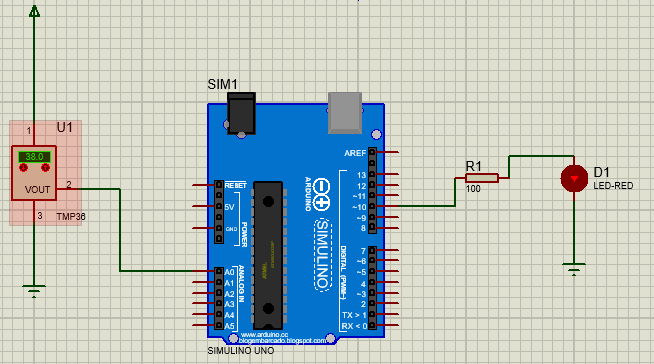
|  |
| --- |
| int x = 0; board  void setup() {  pinMode(2, INPUT);  pinMode(13, OUTPUT);  }  void loop() {  x = digitalRead(2);  if(x == HIGH){  digitalWrite(13, HIGH);  }else {  digitalWrite(13, LOW);  }  delay(1000);  } |

## Bài 3. làm việc với cảm biến nhiệt độ

### Mô tả:

Đọc nhiệt độ môi trường với cảm biến TMP36, đèn LED sáng khi nhiệt độ quá 37 độ. Led (pin 10), TMP (pin A0).

### Sơ đồ thiết kế:



**Hình 3. Sơ đồ cảm biến nhiệt độ**

### Đặc điểm linh kiện:

* + 1 Đèn Led.
  + 1 Điện trở: R1(100 Ω)
  + 1 Cảm biến nhiệt độ TMP 36: cảm biến nhiệt độ môi trường, cho ra tín hiệu analog, nối vào chân analog của board mạch, đọc tín hiệu bằng hàm analogRead.
  + 1 mạch arduino uno.

### Mã lệnh chính:

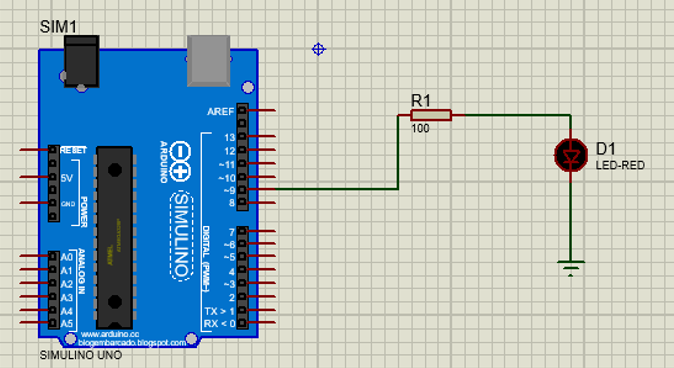
|  |
| --- |
| void setup() {  pinMode(10,OUTPUT);  }  void loop() {  int x=analogRead(A0);  int t=map(x,20,358,-40,125);  if(t>37) digitalWrite(10,HIGH);  else digitalWrite(10,LOW);  delay(1000);  } |

## Bài 4. Led sáng dần

### Mô tả:

Sử dụng các chân ~PWM, xuất các mức điện áp đầu ra thay đổi từ 0-5V , để làm cho đèn LED sáng dần.

### Sơ đồ thiết kế:

****

**Hình 4. Led sáng dần**

### Đặc điểm linh kiện:

* Đèn Led:
* Điện trở: R1(100 Ω)
* 1 mạch arduino uno

### Mã lệnh chính:

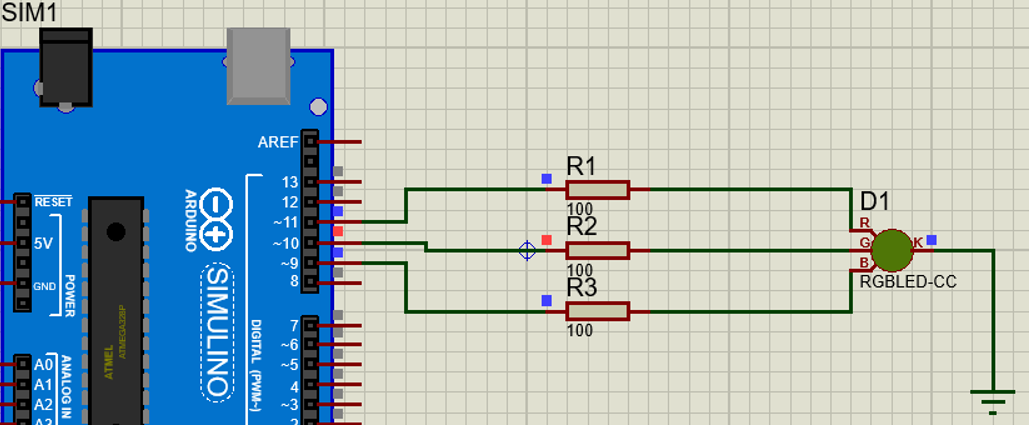
|  |
| --- |
| int brightness = 0;  void setup()  {  pinMode(9, OUTPUT);  }  void loop()  {  for(brightness = 0; brightness <= 255; brightness+=5){  analogWrite(9,brightness);  delay(30);  }  for(brightness = 255; brightness >= 0; brightness-=5){  analogWrite(9,brightness);  delay(30);  }  } |

## Bài 5. Thực hành với Led RGB

### Mô tả

Hệ thống được thiết kế để thực hiện việc đổi màu đèn led sau khoảng thời gian là 1 giây, hiển thị tổ hợp màu red-green-blue với cường độ của từng chân màu sẽ được thay đổi ngẫu nhiên. Từ đó tạo nên việc chuyển màu liên tục.

### Sơ đồ thiết kế

****

**Hình 5. Led RGB**

### Đặc điểm linh kiện:

* 1 mạch arduino uno.
* 3 điện trở: R1(100Ω), R2(100Ω), R3(100Ω).
* 1 Led RGB common cathode.

### Mã lệnh chính:

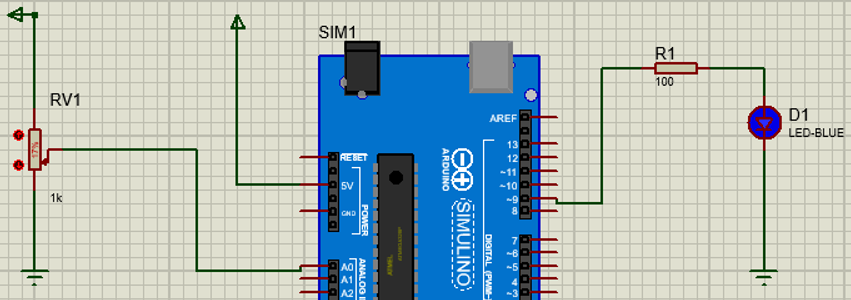
|  |
| --- |
| int redPin = 9;  int bluePin = 10;  int greenPin = 11;  void setup()  {  pinMode(redPin, OUTPUT);  pinMode(bluePin, OUTPUT);  pinMode(greenPin, OUTPUT);  }  void loop()  {  colorRGB(random(0,255),random(0,255),random(0,255));  //R:0-255 G:0-255 B:0-255  delay(1000);  }  void colorRGB(int red, int green, int blue){  analogWrite(redPin,constrain(red,0,255));  analogWrite(greenPin,constrain(green,0,255));  analogWrite(bluePin,constrain(blue,0,255));  } |

## Bài 6. Điều khiển độ sáng của Led qua chiết áp

### Mô tả:

Hệ thống được thiết kế để cho phép điều khiển độ sáng của Led (chân 9) thông qua 1 biến trở gắn ở chân A0.

### Sơ đồ thiết kế



**Hình 6. Điều khiển bằng chiết áp**

### Đặc điểm linh kiện:

* 1 mạch arduino uno
* 1 chiết áp
* 1 điện trở: R1(100Ω)
* 1 đèn Led

### Mã lệnh chính:

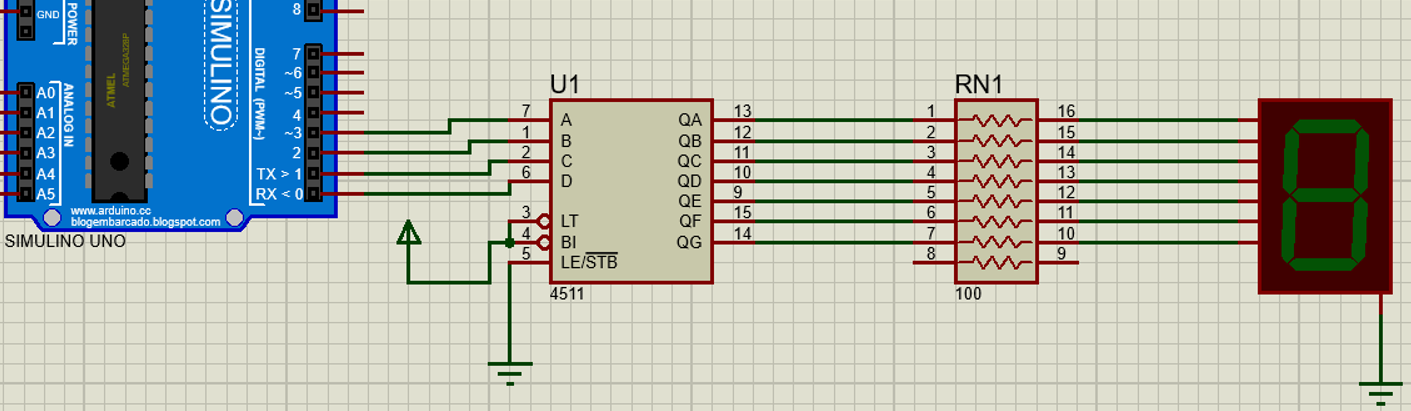
|  |
| --- |
| int x =0;  void setup()  {  Serial.begin(960000);  pinMode(A0, INPUT);  pinMode(9, OUTPUT);  }  void loop()  {  x = analogRead(A0);  Serial.print(x);  int brightness = map(x,0,1023,0,255);  digitalWrite(9,brightness);  delay(1000);  } |

## Bài 7. Led 7 đoạn (7 segment)

### Mô tả:

Hệ thống được thiết kế để thực hiện điều khiển đèn led 7 đoạn sáng bằng lập trình thông qua tín hiệu đầu vào BCD được đấu vào các chân 0,1,2,3.

### Sơ đồ thiết kế



**Hình 7. Led 7 đoạn**

### Đặc điểm linh kiện:

* 1 mạch arduino uno.
* 1 IC 4511: là 1 IC giải mã, làm nhiệm vụ giải mã từ mã nhị phân (binary) sang mã của led 7 vạch để xuất ra led, gồm 16 chân trong đó: chân 3(lamp test) dùng để kiểm tra đèn nếu có giá trị là 0 thì đầu ra sẽ là mức logic 1, chân 4(blanking) tương tự như chân 3 nhưng tác dụng ngược lại (nối nguồn), chân 5 (latch enable) dùng để điều khiển cho phép IC hoạt động bình thường nếu bằng 0 (nối đất), không bình thường nếu bằng 1, chân 1,2,6,7 là chân đưa dữ liệu vào, dữ liệu sẽ được mã hóa và xuất ra ở các chân 9, 10, 11, 12, 13, 14, 15 dưới dạng 7 vạch.
* 1 điện trở 16 chân với R = 100Ω.
* 1 Led 7 đoạn common cathode.

### Mã lệnh chính:

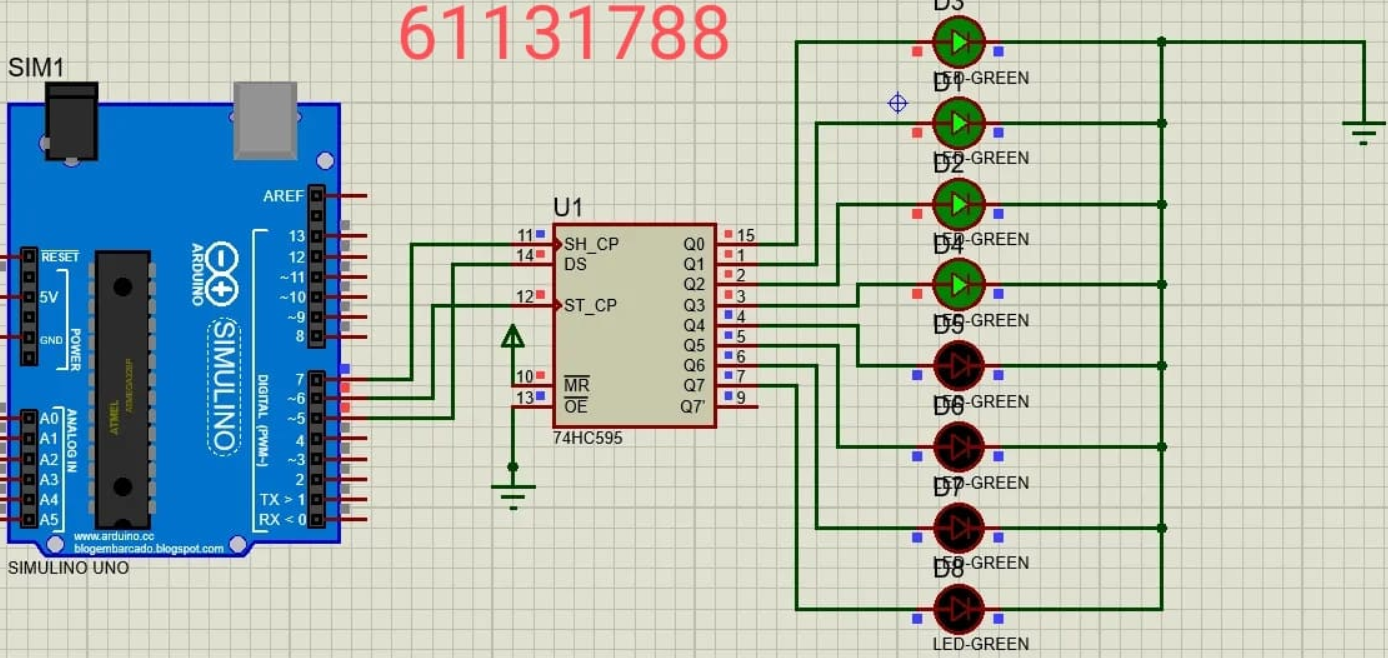
|  |
| --- |
| int a=3, b=2,c=1,d=0;  void setup()  {  pinMode(a, OUTPUT);pinMode(b, OUTPUT);  pinMode(c, OUTPUT); pinMode(d, OUTPUT);  }  void loop()  {  khong();delay(500);  mot();delay(500);  hai();  }  void khong(){  digitalWrite(a, LOW); digitalWrite(b, LOW);  digitalWrite(c, LOW); digitalWrite(d,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

## Bài 8. Sáng 8 Led theo 1 trình tự, sử dụng IC 74C595

### Mô tả:

Bài thực hành lập trình arduino điều khiển 8 Led sáng lần lượt từ trên xuống và tắt từ dưới lên có sử dụng IC 74HC595. IC được kết nối với board mạch arduino thông qua các chân: Chân 11(clock) của IC đấu với chân 7 của Arduino, chân 14(input) của IC đấu với chân 6 của Arduino,và chân 12(latch) của IC đấu với chân 5 của Arduino. 8 Led được đấu vào IC lần lượt ở các chân từ 15 đến 7(Q0-Q7) của IC.

### Sơ đồ thiết kế:



**Hình 8. Led sáng tuần tự**

### Đặc điểm linh kiện:

* 1 board mạch arduino uno.
* 1 IC 74HC595: là một IC thông dụng với khả năng dịch bit và ghi nhớ tạm thời 8-bit. Với khả năng đó, IC 74HC595 có khả năng mở rộng số lượng chân vốn đã ít của Board mạch Arduino.

+ chân 10 MR (master reset): Reset tất cả các chân đầu ra ở mức thấp. Khi cấp nguồn cho chân này sẽ giữ chân ở mức cao để hoạt động bình thường.

+ chân 13 OE (output enable): dùng để tắt đầu ra. Khi nối chân này với đất sẽ giữ chân ở mức thấp để hoạt động bình thường.

+ chân 11 SH\_CP (clock): Đây là chân clock mà tín hiệu clock (1 xung) phải được cung cấp từ vi điều khiển hoặc vi xử lý.

+ chân 12 ST\_CP (latch): Chân Latch dùng để cập nhật dữ liệu vào các chân đầu ra. Nó kích hoạt mức cao.

+chân 14 DS (data serial) : Đây là chân mà dữ liệu được gửi đến, dựa trên đó 8 đầu ra được điều khiển.

* 8 Led Green.

### Mã lệnh chính:

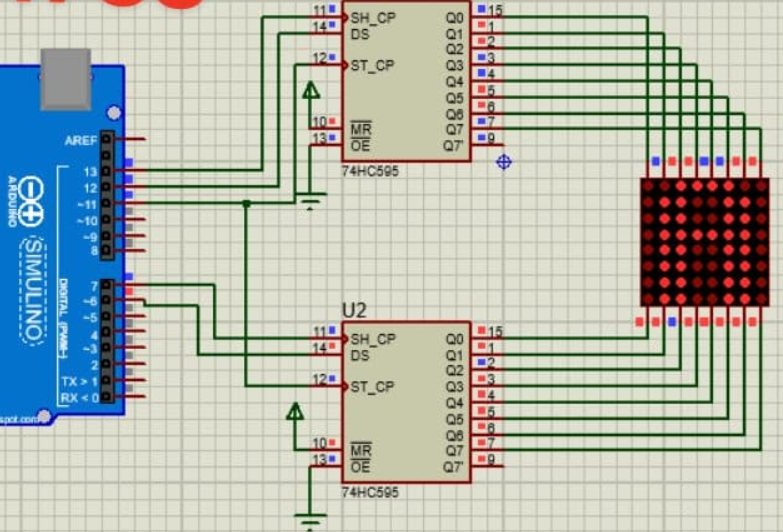
|  |
| --- |
| int \_data=5,\_latch=6,\_clock=7;  byte ledStatus;  void setup() {  pinMode(\_latch, OUTPUT);  pinMode(\_clock, OUTPUT);  pinMode(\_data, OUTPUT);  }  void loop() {  //Sáng tuần tự  ledStatus = 0;//mặc định là không có đèn nào sáng hết (0 = 0b00000000)  for (int i = 0; i < 8; i++) {  ledStatus = (ledStatus << 1) | 1;//Đẩy toàn bộ các bit qua trái 1 bit và cộng bit có giá trị là 1 ở bit 0  digitalWrite(\_latch, LOW);  //ShiftOut ra IC  shiftOut(\_data, \_clock, MSBFIRST, ledStatus);  digitalWrite(\_latch, HIGH);//các đèn LED sẽ sáng với trạng thái vừa được cập nhập  delay(500); // Dừng chương trình khoảng 500 mili giây để thấy các hiệu ứng của đèn LED  }    //Tắt tuần tự  for (int i = 0;i<8;i++) {  ledStatus <<= 1; //Đẩy tất cả các bit qua bên trái 1 bit  digitalWrite(\_latch, LOW);  shiftOut(\_data, \_clock, MSBFIRST, ledStatus);  digitalWrite(\_latch, HIGH);  delay(500);  }  } |

## Bài 9. Ma trận led

### Mô tả:

Bài thực hành lập trình điều khiển ma trận led 8x8 sáng hình chữ A, thông qua 2 IC 74HC595, 1 dùng để điều khiển cột, 1 dùng để điều khiển hàng, 2 IC được đấu trực tiếp vào arduino thông qua các cổng 13(SH\_CP IC1), 12(DS IC1), 7(SH\_CP IC2), 6(DS IC2) và đều dùng cổng 11 để nối vs chân 12(ST\_CP) của 2 IC, các chân từ Q0-Q7 của IC 1 lần lượt được đấu vào ma trận led để điều khiển cột. Tương tự, các chân từ Q0-Q7 của IC 2 lần lượt được đấu vào ma trận led để điều khiển hàng.

### Sơ đồ thiết kế:



Hình 9. Ma trận Led

### Đặc điểm linh kiện:

* 1 board mạch arduino uno.
* 2 IC 74HC595: là một IC thông dụng với khả năng dịch bit và ghi nhớ tạm thời 8-bit. Với khả năng đó, IC 74HC595 có khả năng mở rộng số lượng chân vốn đã ít của Board mạch Arduino.

+ chân 10 MR (master reset): Reset tất cả các chân đầu ra ở mức thấp. Khi cấp nguồn cho chân này sẽ giữ chân ở mức cao để hoạt động bình thường.

+ chân 13 OE (output enable): dùng để tắt đầu ra. Khi nối chân này với đất sẽ giữ chân ở mức thấp để hoạt động bình thường.

+ chân 11 SH\_CP (clock): Đây là chân clock mà tín hiệu clock (1 xung) phải được cung cấp từ vi điều khiển hoặc vi xử lý.

+ chân 12 ST\_CP (latch): Chân Latch dùng để cập nhật dữ liệu vào các chân đầu ra. Nó kích hoạt mức cao.

+chân 14 DS (data serial) : Đây là chân mà dữ liệu được gửi đến, dựa trên đó 8 đầu ra được điều khiển.

* 1 Led ma trận.

### Mã lệnh chính:

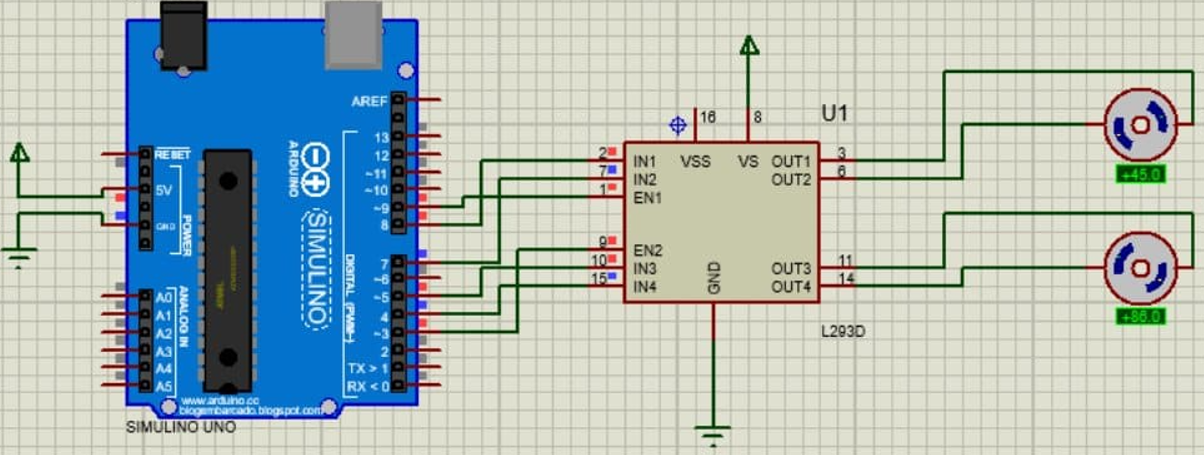
|  |
| --- |
| int lock\_hang = 13;  int lock\_cot = 7;  int latch = 11;  int data\_hang = 12;  int data\_cot = 6;  byte row[8]={  B00111100,  B01100110,  B01100110,  B01111110,  B01100110,  B01100110,  B01100110,  B01100110  };  byte hex[8]={  B11111110,  B11111101,  B11111011,  B11110111,  B11101111,  B11011111,  B10111111,  B01111111  };  void setup()  {  pinMode(lock\_hang, OUTPUT);  pinMode(lock\_cot, OUTPUT);  pinMode(latch, OUTPUT);  pinMode(data\_hang, OUTPUT);  pinMode(data\_cot, OUTPUT);  }  void loop()  {  for(int i=0;i<8;i++){  digitalWrite(latch, LOW);  shiftOut(data\_hang,lock\_hang,MSBFIRST,row[i]);  shiftOut(data\_cot,lock\_cot,MSBFIRST,hex[i]);  digitalWrite(latch, HIGH);  }  } |

## Bài 10. Động cơ DC

### Mô tả:

Bài thực hành lập trình arduino điều khiển 2 Motor DC hoạt động, thông qua IC L293D, được đấu trực tiếp vào arduino thông qua các chân: chân 3 của arduino nối với chân 9 của IC (Enable 3,4),chân 9 của arduino nối với chân 1 của IC (Enable 1,2), chân 4 của arduino nối với chân 15 của IC(input4), chân 5 của arduino nối với chân 10 của IC(input3), chân 9 của arduino nối với chân 1 của IC(input1), chân 7 của arduino nối với chân 7 của IC(input2).

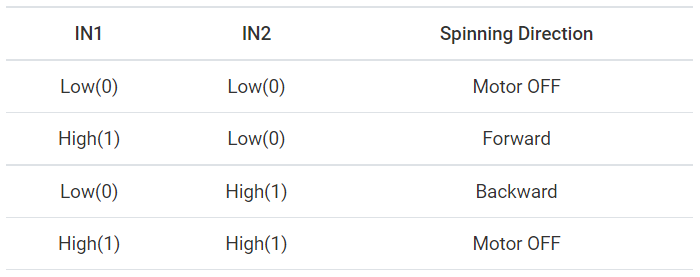
### Sơ đồ thiết kế:



Hình 10. Động cơ DC

### Đặc điểm linh kiện:

* 1 board mạch arduino uno.
* 2 motor DC.
* 1 IC L293D: là IC cầu H điều khiển động cơ bằng cách thay đổi cực tính của điện áp đầu vào của nó.Cầu H chứa 4 công tắc, với động cơ ở trung tâm từ đó khi đóng đồng thời 2 công tắc cụ thể sẽ đảo ngược cực tính của điện áp đặt vào động cơ . Điều này làm thay đổi hướng quay của động cơ. IC L293D gồm 4 kênh điều khiển 2 động cơ DC.
* IC có 2 chân đầu vào cấp nguồn là VCC1 (16) và VCC2 (8). VCC1 dùng để điều khiển mạch logic bên tay phải là 5V. VCC2 tương tự nhưng điều khiển mạch logic bên trái. Cả 2 đều được đấu chung vào chân đất của IC.
* IC có 4 chân đầu ra để điều khiển 2 động cơ tại các chân OUT1 (3), OUT2 (6), OUT3 (11), OUT4 (14) . Mỗi kênh có thể cung cấp cho động cơ 600 mA, tuy nhiên vẫn phụ thuộc vào hệ thống.
* đối với mỗi kênh của IC đều có 2 loại chân điều khiển cho phép thay đổi tốc độ và hướng quay của động cơ. Các chân này điều khiển động cơ quay về phía trước, hay lùi về phía sau bằng cách điều khiển đóng mở cụ thể các công tắc bên trong IC, các chân IN1 (2), IN2 (7) điều khiển động cơ A, IN3 (10) IN4 (15) điều khiển động cơ B. Chiều quay của động cơ có thể điều khiển bằng cách áp dụng logic HIGH hoặc LOW cho các chân này như sau:



* IC còn có thêm 2 chân để bật, tắt và điều khiển tốc độ là ENA (1), ENB (9). Nếu áp dụng logic HIGH sẽ làm cho động cơ quay, ngược lại LOW làm cho động cơ dừng. Tuy nhiên, có thể điều khiển tốc độ của động cơ bằng cách đấu vào các chân PWM.

### Mã lệnh chính:

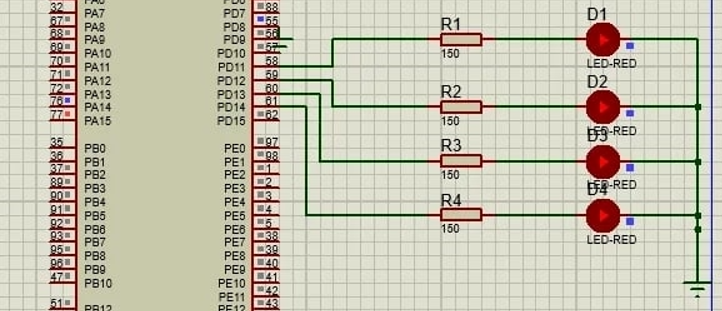
|  |
| --- |
| int enA = 9, in1 = 8, in2 = 7; // Motor A connections  int enB = 3,int in3 = 5, in4 = 4; // Motor B connections  void setup() {  pinMode(enA, OUTPUT);pinMode(enB, OUTPUT); pinMode(in1, OUTPUT);  pinMode(in2, OUTPUT); pinMode(in3, OUTPUT); pinMode(in4, OUTPUT);    // khởi tạo  digitalWrite(in1, LOW); digitalWrite(in2, LOW);  digitalWrite(in3, LOW); digitalWrite(in4, LOW);  }  void loop() {  directionControl();delay(1000); speedControl();  }  void directionControl() {  analogWrite(enA, 255);  analogWrite(enB, 255); // xuất chân enable cho IC hoạt động  // Turn on motor A & B  digitalWrite(in1, HIGH); digitalWrite(in2, LOW); digitalWrite(in3, HIGH);  digitalWrite(in4, LOW); delay(1000);    // change motor directions  digitalWrite(in1, LOW); digitalWrite(in2, HIGH); digitalWrite(in3, LOW);  digitalWrite(in4, HIGH); delay(1000);    // Turn off motors  digitalWrite(in1, LOW); digitalWrite(in2, LOW);  digitalWrite(in3, LOW); digitalWrite(in4, LOW);  }  // This function lets you control speed of the motors  void speedControl() {  // Turn on motors  digitalWrite(in1, LOW); digitalWrite(in2, HIGH);  digitalWrite(in3, LOW); digitalWrite(in4, HIGH);  // Accelerate from zero to maximum speed  for (int i = 0; i < 256; i++) {  analogWrite(enA, i); analogWrite(enB, i); delay(20);  }  // Decelerate from maximum speed to zero  for (int i = 255; i >= 0; --i) {  analogWrite(enA, i); analogWrite(enB, i); delay(20);  }  // Now turn off motors  digitalWrite(in1, LOW); digitalWrite(in2, LOW);  digitalWrite(in3, LOW);digitalWrite(in4, LOW);  } |

## Bài 11. Sáng led lần lượt – STM32

### Mô tả:

Bài thực hành lập trình sử dụng vi điều khiển STM32 (ARM), thực hiện làm cho 4 đèn sáng lần lượt với độ trễ 0,5 giây. Các đèn được đấu lần lượt vào các chân PD12, PD13, PD14, PD15.

### Sơ đồ thiết kế:



Hình 11. STM32-Sáng led lần lượt

### Đặc điểm linh kiện:

* 1 vi điều khiển STM32F401VE.
* 4 đèn led.
* 4 điện trở R=150 Ω

### Mã lệnh chính:

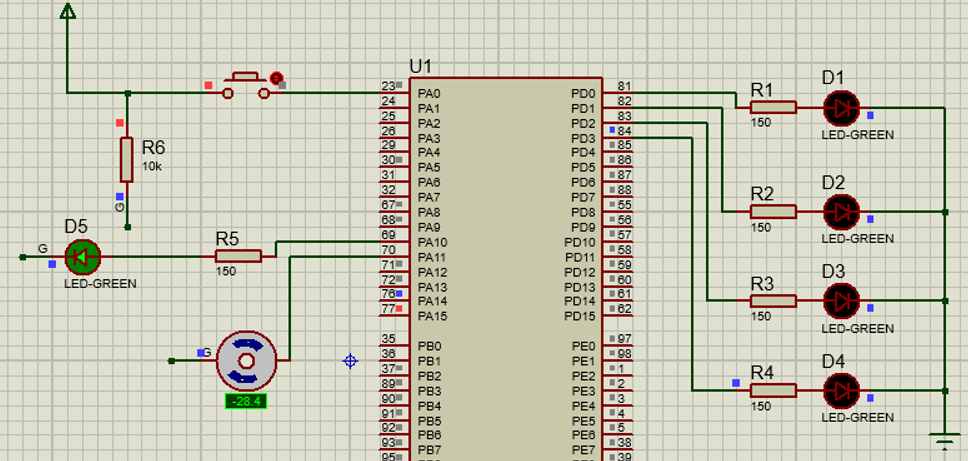
|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  int main(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  while (1)  {  HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15);  HAL\_Delay(1000);  } 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

## Bài 12. Lập trình Ngắt ngoài – STM32

### Mô tả:

Bài thực hành lập trình trên board mạch STM32, thực hiện lập trình ngắt ngoài. Chương trình chính sẽ liên tục nhấp nháy 4 Led. Trong khi đó, khi nhấn 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 phụ: nút bấm được đấu vào chân PA0, Led đấu vào chân PA12, và Motor được đấu vào chân PA13 của board mạch STM32.

### Sơ đồ thiết kế:



Hình 12. STM32-ngắt ngoài

* Đặc điểm linh kiện:

1 vi điều khiển STM32F401VE.

5 đèn led.

5 điện trở R = 150 Ω.

1 động cơ Motor DC.

1 nút bấm.

### Mã lệnh chính:

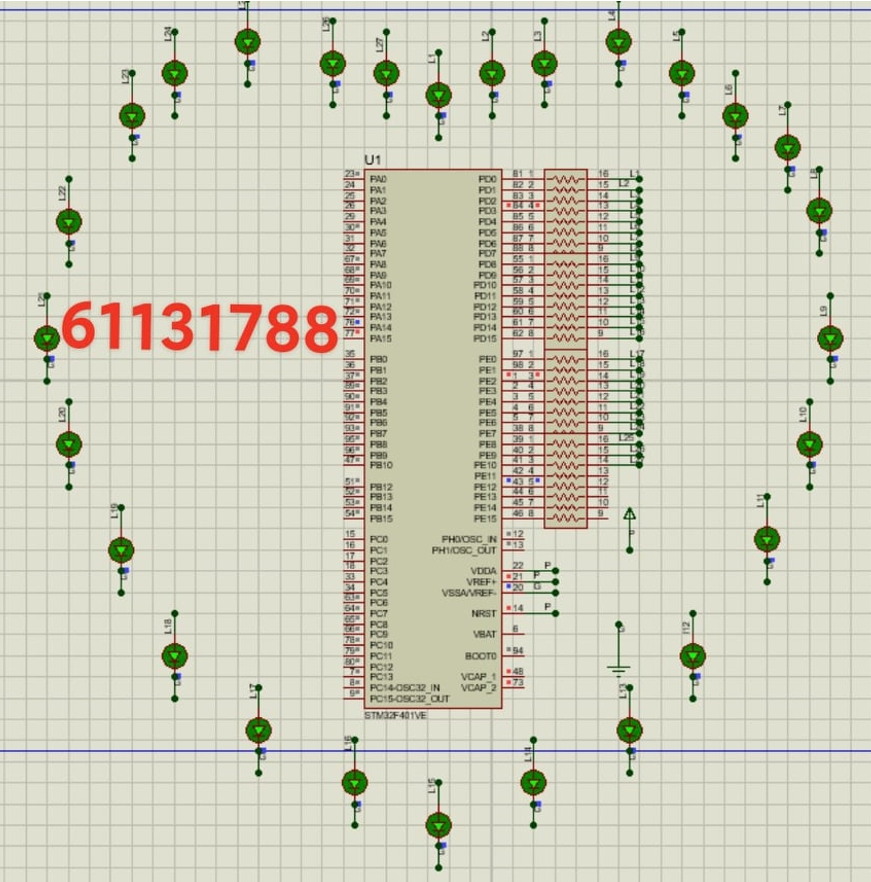
|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  int main(void)  {    HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  while (1)  {  HAL\_GPIO\_TogglePin(GPIOD,GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3);  HAL\_Delay(1000);  }  /\* USER CODE END 3 \*/  }  void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin) {  if(GPIO\_Pin == GPIO\_PIN\_0){  HAL\_GPIO\_TogglePin(GPIOA,GPIO\_PIN\_10|GPIO\_PIN\_11);  }  } |

## Bài 13. Led Trái Tim – STM32

### Mô tả:

Bài thực hành lập trình sử dụng vi điều khiển STM32 (ARM), thực hiện làm cho 25 đèn Led sáng theo hình trái tim, với nhiều trình tự sáng khác nhau như: bật hết led, tắt hết led, sáng tuần tự và tắt tuần tự. Các chân từ PD0 đến PE10 lần lượt được đấu vào hệ thống led dưới hình thức wireless bằng cách sử dụng tính năng wire label mode (tạo nhãn cho các chân) giúp tiết kiệm dây và hạn chế bất tiện khi nối dây.

### Sơ đồ thiết kế:



Hình 13. Led trái tim

### Đặc điểm linh kiện:

1 vi điều khiển STM32F401VE.

25 Led Green.

2 điện trở 16 chân với R = 150Ω

### Mã lệnh chính:

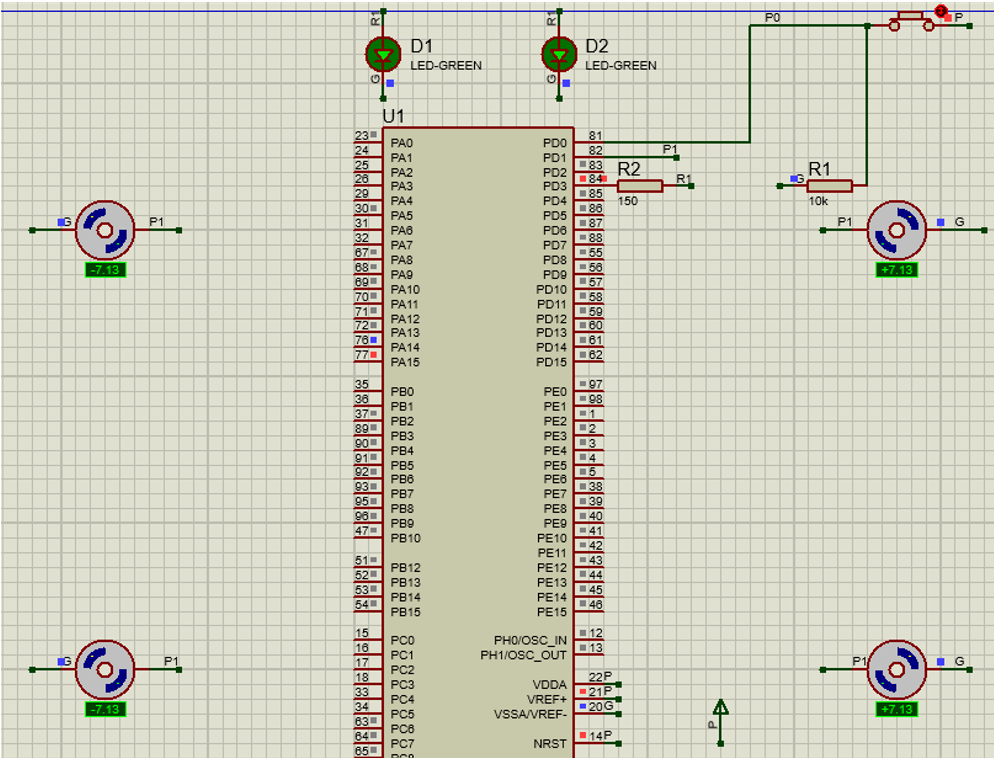
|  |
| --- |
| int main(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  while (1)  {  Light\_All(); // sang het led  HAL\_Delay(1000);  Dark\_all();//tat het led  HAL\_Delay(1000);  light\_in\_turn(); //sang lan luot  HAL\_Delay(1000);  dark\_in\_turn();//tat lan luot  }  }  void Light\_All(){  HAL\_GPIO\_WritePin( GPIOD,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\_8|GPIO\_PIN\_9|GPIO\_PIN\_10|GPIO\_PIN\_11|GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15,GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin( GPIOE,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\_8|GPIO\_PIN\_9|GPIO\_PIN\_10,GPIO\_PIN\_SET);  }  void Dark\_all(){  HAL\_GPIO\_WritePin( GPIOD,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\_8|GPIO\_PIN\_9|GPIO\_PIN\_10|GPIO\_PIN\_11|GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15,GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin( GPIOE,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\_8|GPIO\_PIN\_9|GPIO\_PIN\_10,GPIO\_PIN\_RESET);  }  void light\_in\_turn(){  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_1,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_2,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_3,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_4,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_5,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_6,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_7,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_8,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_9,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_10,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_11,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_12,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_13,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_14,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_15,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_0,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_1,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_2,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_3,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_4,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_5,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_6,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_7,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_8,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_9,GPIO\_PIN\_SET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_10,GPIO\_PIN\_SET);  HAL\_Delay(500);  }  void dark\_in\_turn() {  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_1,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_2,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_3,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_4,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_5,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_6,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_7,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_8,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_9,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_10,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_11,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_12,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_13,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_14,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_15,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_0,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_1,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_2,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_3,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_4,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_5,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_6,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_7,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_8,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_9,GPIO\_PIN\_RESET);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(GPIOE,GPIO\_PIN\_10,GPIO\_PIN\_RESET);  HAL\_Delay(500);  } |

## Bài 14. Nút bấm – STM32

### Mô tả:

Bài thực hành lập trình sử dụng vi điều khiển STM32(ARM), thực hiện điều khiển đèn 2 Led và 4 Motor được thiết kế tựa như bố cục của 1 ô tô với 4 bánh xe là 4 motor, 2 led là 2 đèn pha, hoạt động khi nhấn nút bấm, và ngược lại sẽ ngừng hoạt động khi thả nút bấm ra. Nút bấm được đấu vào chân PD0 của STM32, Led được đấu vào chân PD3, và Motor được đấu vào chân PD1 của board mạch STM32.

### Sơ đồ thiết kế:



Hình 14. STM32 – nút bấm

### Đặc điểm linh kiện:

* + 1 vi điều khiển STM32F401VE.
  + 4 Motor DC.
  + 2 Led green.
  + 2 điện trở với R1 = 150Ω, R2 = 10kΩ
  + 1 nút bấm.

### Mã lệnh chính:

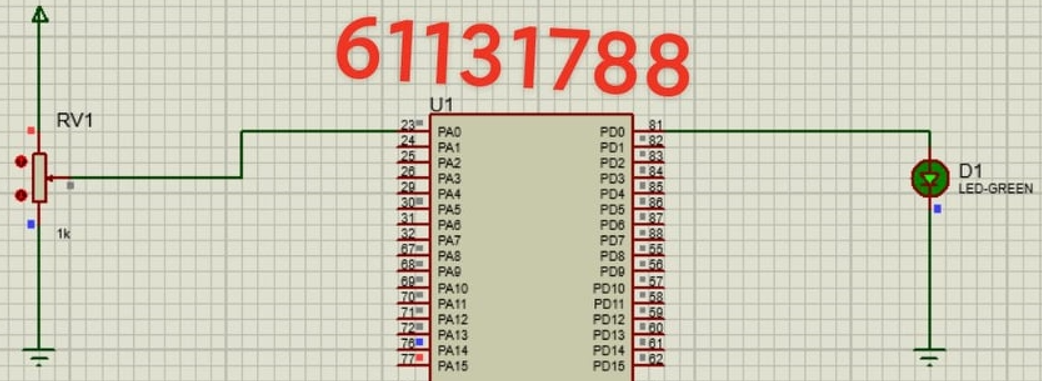
|  |
| --- |
| int main(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  while (1)  {  GPIO\_PinState pin0State = HAL\_GPIO\_ReadPin(GPIOD,GPIO\_PIN\_0);  if(pin0State == GPIO\_PIN\_SET)  {  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_1,GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_3,GPIO\_PIN\_SET);  }  else  {  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_1,GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_3,GPIO\_PIN\_RESET);  }  }  } |

## Bài 15. Đọc tín hiệu tương tự số từ chiết áp – STM32

### Mô tả:

Bài thực hành lập trình sử dụng vi điều khiển STM32(ARM), thực hiện điều khiển Led thông qua chiết áp khi điều chỉnh giá trị của chiết áp lớn hơn 25 thì đèn sáng ngược lại thì tắt, tín hiệu được đọc vào ở chân PA0, và xuất ra trực tiếp vào đèn Led ở chân PD0.

### Sơ đồ thiết kế:



Hình 15. STM32 - ADC

### Đặc điểm linh kiện:

* 1 Vi điều khiển STM32F401VE.
* 1 Chiết áp.
* 1 Led green.

### Mã lệnh chính:

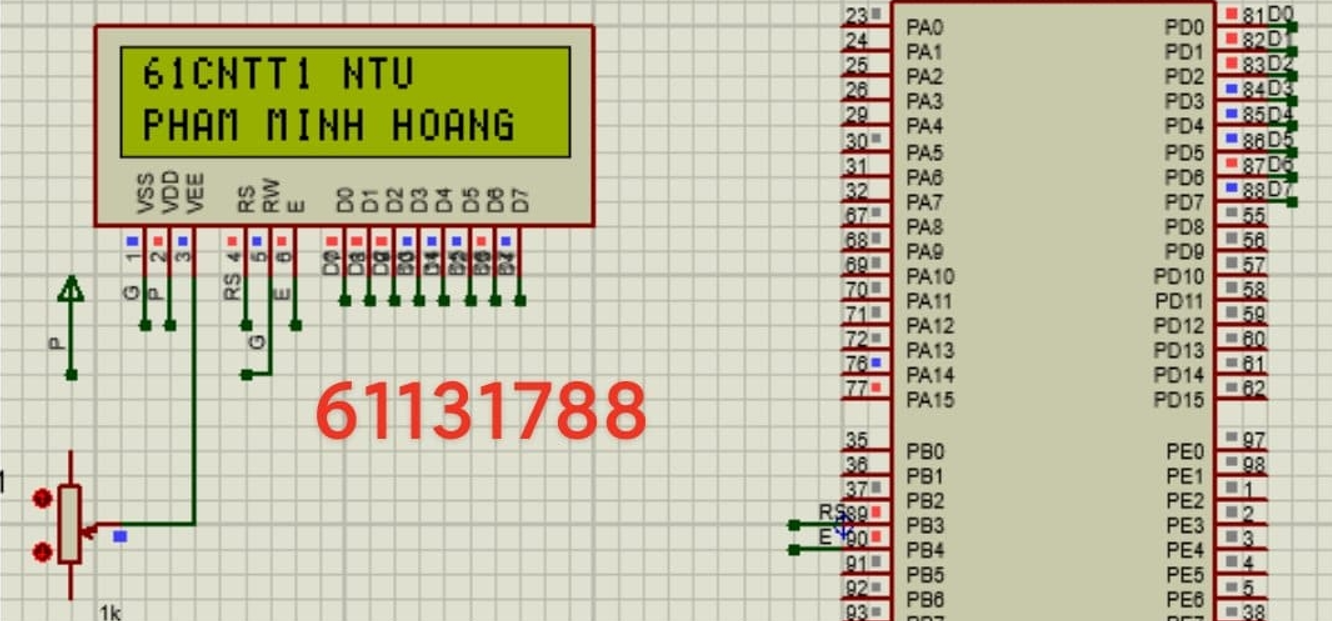
|  |
| --- |
| #include "main.h"  ADC\_HandleTypeDef hadc1;  DMA\_HandleTypeDef hdma\_adc1;  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  static void MX\_DMA\_Init(void);  static void MX\_ADC1\_Init(void);  int main(void)  { HAL\_Init();  MX\_GPIO\_Init();  MX\_DMA\_Init();  MX\_ADC1\_Init();  uint32\_t data;  while (1)  {  /\* USER CODE END WHILE \*/  HAL\_ADC\_Start\_DMA(&hadc1,(uint32\_t\*)data,1);  HAL\_Delay(1000);  HAL\_ADC\_Stop\_DMA(&hadc1);  if(data > 4000){  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0,GPIO\_PIN\_SET);  }  else {  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0,GPIO\_PIN\_RESET);  }  }  } |

## Bài 16. LCD – STM32

### Mô tả:

Bài thực hành lập trình sử dụng vi điều khiển STM32(ARM), thực hiện điều khiển hiển thị thông điệp lên màn hình LCD. Từ chân D0-D7 của LCD để nhận tín hiệu vào được nối trực tiếp với vi điều khiển lần lượt từ PD0-PD7, chân RS và EN tương tự được nối vào cổng PB3, PB4 để nhận tín hiệu kích hoạt, Chân RW (chọn thanh ghi) khi nối với đất thì mặc định mức logic ở chân này là 0 thì sẽ luôn ở chế độ ghi.

### Sơ đồ thiết kế:



Hình 16. STM32-LCD

### Đặc điểm linh kiện:

* 1 Vi điều khiển STM32F401VE.
* 1 Chiết áp.
* 1 màn hình LCD: gồm 8 chân (D0 – D7) : để nhận tín hiệu từ vi điều khiển. Chân VSS được dùng để nối đất, còn chân VDD được dùng để cấp nguồn cho LCD hoạt động, Chân VEE để điều khiển độ tương phản. Chân RS chọn thanh ghi lệnh, hay dữ liệu cho LCD, chân RW sẽ cho biết đọc hay ghi dữ liệu vào LCD, còn lại chân E (enable) khi được kích hoạt thì sẽ cho phép 8 bit được đưa vào LCD từ đó, xử lý thông tin hiện lên màn hình LCD.

### Mã lệnh chính:

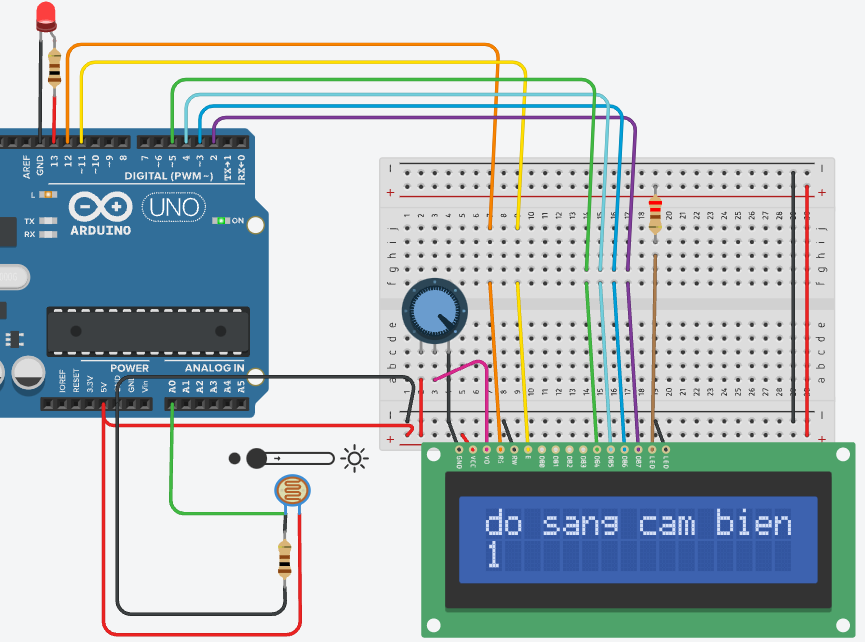
|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  void send8BitLCD(char D) {  // Dem D and so hoc voi 2^i de biet bit thu i =0 hay khac ko  // tim gia tri cac bit  int b0,b1,b2, b3,b4, b5, b6, b7;  if ((D & 1) == 0) b0=0; // 1== 2^0  else b0 =1;  if ((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);  }  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);  }  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);  }  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("61CNTT1 NTU");  // Xuong dong 2  sendCMD2LCD(0x38); // CHE DO 2 DONG  sendCMD2LCD(0xC0); // TRO XUONG DONG 2  sendString2LCD("PHAM MINH HOANG");  HAL\_Delay(1000);  while (1)  {  }  } |

## Bài 1. Kiểm tra giữa kỳ

### Mô tả:

bài thực hành lập trình arduino thực hiện đọc giá trị độ sáng của cảm biến và hiện ra LCD sau mỗi giây. Nếu giá trị nhỏ hơn 50% thang đo thì bật đèn điện sáng và ngược lại.

### Sơ đồ thiết kế:

****

Hình 1. Điều khiển LCD

### Đặc điểm linh kiện:

* Mạch arduino.
* 1 chiết áp.
* 1 màn hình LCD 16 x 2 hiển thị kết quả.
* 1 quang điện trở đo độ sáng.
* 1 đèn led.
* 3 điện trở: R1 (100 Ω), R2 (220 Ω), R3 (220Ω)

### Mã lệnh chính:

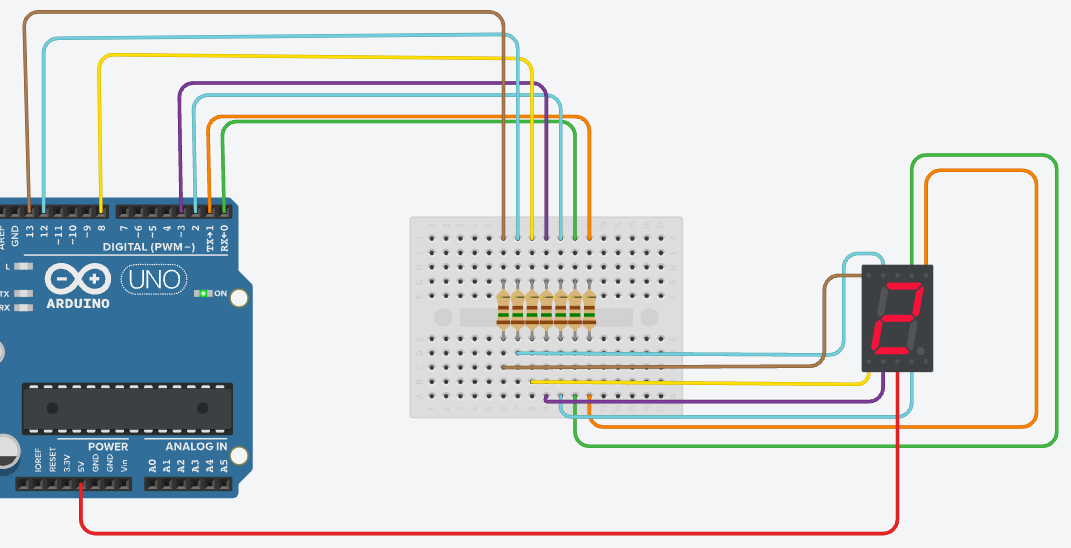
|  |
| --- |
| #include <LiquidCrystal.h>  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  int led = 13;  void setup() {  pinMode(A0,INPUT);  pinMode(led,OUTPUT);  lcd.begin(16, 2);  lcd.print("do sang cam bien: ");  }  void loop() {:  lcd.setCursor(0, 1);  int x = analogRead(A0);  lcd.print(x);  delay(1000);  if (x < 84) { digitalWrite(led,HIGH); }  else digitalWrite(led,LOW);  } |

## Bài 2A. Kiểm tra giữa kỳ

### Mô tả:

Xây dựng hệ thống nhúng với led 7 đoạn thực hiện đếm tuần tự từ 0 đến 9.

### Sơ đồ thiết kế:



Hình 2a. đếm số từ 0 - 9

### Đặc điểm linh kiện:

* 1 board mạch arduino uno.
* 1 bảng mạch.
* 1 đèn led 7 đoạn (chung cực dương)
* 7 điện trở: R1, R2, R3, R4, R5, R6, R7, R8 (150 Ω)

### Mã lệnh chính:

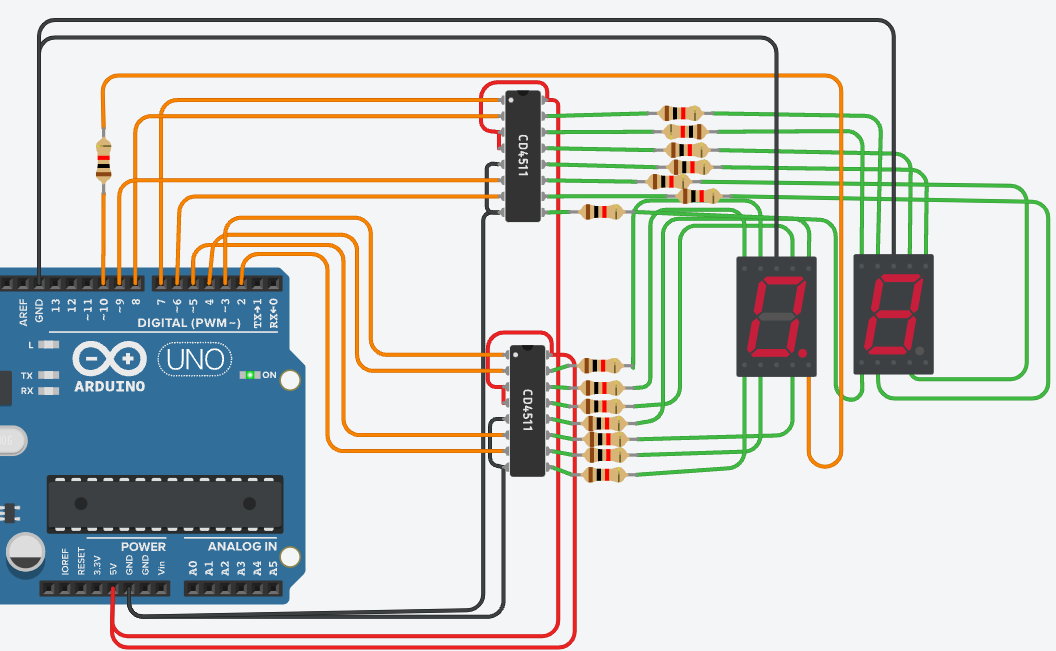
|  |
| --- |
| int a=0, b=1,c=2,d=3,e=8,f=13,g=12;  void setup()  {  pinMode(a, OUTPUT);pinMode(b, OUTPUT);pinMode(c, OUTPUT);  pinMode(d, OUTPUT);pinMode(e, OUTPUT);pinMode(f, OUTPUT);  pinMode(g, OUTPUT);  }  void loop()  {  khong();delay(500); mot(); (500); hai();delay(500); ba();delay(500); bon();delay(500);  nam();delay(500); sau();delay(500); bay();delay(500); tam();delay(500); chin();  }  void khong(){  digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, LOW);  digitalWrite(e, LOW); digitalWrite(f, HIGH); digitalWrite(g, LOW); delay(1000);  }  void mot(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH); digitalWrite(d, HIGH);  digitalWrite(e, LOW); digitalWrite(f, HIGH); digitalWrite(g, LOW); delay(1000);  }  void hai(){  digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, HIGH); digitalWrite(d, LOW);  digitalWrite(e, LOW); digitalWrite(f, LOW); digitalWrite(g, HIGH); delay(1000);  }  void ba(){  digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, LOW);  digitalWrite(e, HIGH); digitalWrite(f, LOW); digitalWrite(g, HIGH); delay(1000);  }  void bon(){  digitalWrite(a, HIGH); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, HIGH);  digitalWrite(e, HIGH); digitalWrite(f, LOW); digitalWrite(g, LOW); delay(1000);  }  void nam(){  digitalWrite(a, LOW); digitalWrite(b, HIGH); digitalWrite(c, LOW); digitalWrite(d, LOW);  digitalWrite(e, HIGH); digitalWrite(f, LOW); digitalWrite(g, LOW); delay(1000);  }  void sau(){  digitalWrite(a, LOW); digitalWrite(b, HIGH); digitalWrite(c, LOW); digitalWrite(d, LOW);  digitalWrite(e, LOW); digitalWrite(f, LOW); digitalWrite(g, LOW); (1000);  }  void bay(){  digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, HIGH);  digitalWrite(e, HIGH); digitalWrite(f, HIGH); digitalWrite(g, HIGH); delay(1000);  }  void tam(){  digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, LOW);  digitalWrite(e, LOW); digitalWrite(f, LOW); digitalWrite(g, LOW); delay(1000);  }  void chin(){  digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, LOW);  digitalWrite(e, HIGH); digitalWrite(f, LOW); digitalWrite(g, LOW); delay(1000);  } |

## Bài 2B. Kiểm tra giữa kỳ

### Mô tả :

Xây dựng hệ thống nhúng với led 7 đoạn thực hiện đếm tuần tự từ 0.0 đến 9.9

### Sơ đồ thiết kế:



Hình 2b. đếm tuần tự từ 0.0 – 9.9

### Đặc điểm linh kiện:

* + - 1 board mạch arduino uno.
    - 2 IC CD4511: là 1 IC giải mã, làm nhiệm vụ giải mã từ mã nhị phân (binary) sang mã của led 7 vạch để xuất ra led, gồm 16 chân trong đó: chân 3(lamp test) dùng để kiểm tra đèn nếu có giá trị là 0 thì đầu ra sẽ là mức logic 1, chân 4(blanking) tương tự như chân 3 nhưng tác dụng ngược lại (nối nguồn), chân 5 (latch enable) dùng để điều khiển cho phép IC hoạt động bình thường nếu bằng 0 (nối đất), không bình thường nếu bằng 1, chân 1,2,6,7 là chân đưa dữ liệu vào, dữ liệu sẽ được mã hóa và xuất ra ở các chân 9, 10, 11, 12, 13, 14, 15 dưới dạng 7 vạch.
    - 2 đèn led 7 đoạn (chung cực âm).
    - 15 điện trở với R = 220 Ω.

### Mã lệnh chính:

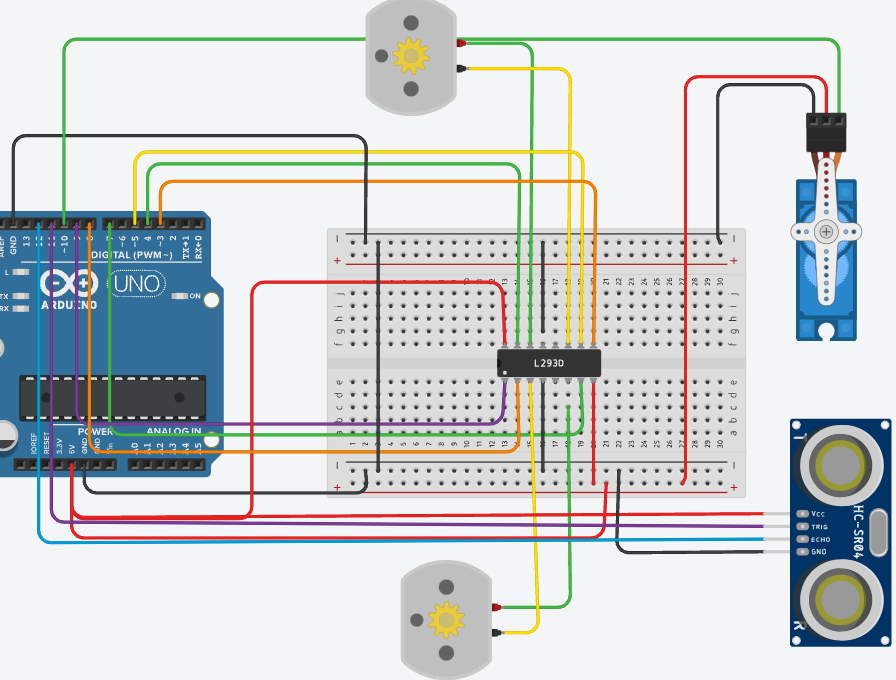
|  |
| --- |
| int a1=2, a2=3,a3=4,a4=5,b1=6, b2=7,b3=8,b4=9;  void setup()  {  pinMode(a1, OUTPUT); pinMode(a2, OUTPUT); pinMode(a3, OUTPUT); pinMode(a4,OUTPUT);  pinMode(b1, OUTPUT); pinMode(b2, OUTPUT); pinMode(b3, OUTPUT); pinMode(b4, OUTPUT);  pinMode(10,OUTPUT);  }  void loop()  {  digitalWrite(10, HIGH);  for(int i = 0;i <= 9; i++){  Tkhong();lap();Tmot();lap();Thai();lap();Tba();lap();Tbon();lap();Tnam();lap();Tsau();lap();Tbay();  lap();Ttam();lap();Tchin();lap();  }  }  void Tkhong(){  digitalWrite(a1, LOW); digitalWrite(a2, LOW); digitalWrite(a3, LOW); digitalWrite(a4, LOW);  }  void Tmot(){  digitalWrite(a1, HIGH); digitalWrite(a2, LOW); digitalWrite(a3, LOW); digitalWrite(a4, LOW);  }  void Thai(){  digitalWrite(a1, LOW); digitalWrite(a2, HIGH); digitalWrite(a3, LOW); digitalWrite(a4, LOW);  }  void Tba(){  digitalWrite(a1, HIGH); digitalWrite(a2, HIGH); digitalWrite(a3, LOW); digitalWrite(a4, LOW);  }  void Tbon(){  digitalWrite(a1, LOW); digitalWrite(a2, LOW); digitalWrite(a3, HIGH); digitalWrite(a4, LOW);  }  void Tnam(){  digitalWrite(a1, HIGH); digitalWrite(a2, LOW); digitalWrite(a3, HIGH); digitalWrite(a4, LOW);  }  void Tsau(){  digitalWrite(a1, LOW); digitalWrite(a2, HIGH); digitalWrite(a3, HIGH); digitalWrite(a4, LOW);  }  void Tbay(){  digitalWrite(a1, HIGH); digitalWrite(a2, HIGH); digitalWrite(a3, HIGH); digitalWrite(a4, LOW);  }  void Ttam(){  digitalWrite(a1, LOW); digitalWrite(a2, LOW); digitalWrite(a3, LOW); digitalWrite(a4, HIGH);  }  void Tchin(){  digitalWrite(a1, HIGH); digitalWrite(a2, LOW); digitalWrite(a3, LOW); digitalWrite(a4, HIGH);  }  void Pkhong() {  digitalWrite(b1, LOW); digitalWrite(b2, LOW); digitalWrite(b3, LOW); digitalWrite(b4, LOW);  }  void Pmot(){  digitalWrite(b1, HIGH); digitalWrite(b2, LOW); digitalWrite(b3, LOW); digitalWrite(b4, LOW);  }  void Phai(){  digitalWrite(b1, LOW); digitalWrite(b2, HIGH); digitalWrite(b3, LOW); digitalWrite(b4, LOW);  }  void Pba(){  digitalWrite(b1, HIGH); digitalWrite(b2, HIGH); digitalWrite(b3, LOW); digitalWrite(b4, LOW);  }  void Pbon(){  digitalWrite(b1, LOW); digitalWrite(b2, LOW); digitalWrite(b3, HIGH); digitalWrite(b4, LOW);  }  void Pnam(){  digitalWrite(b1, HIGH); digitalWrite(b2, LOW); digitalWrite(b3, HIGH); digitalWrite(b4, LOW);  }  void Psau(){  digitalWrite(b1, LOW); digitalWrite(b2, HIGH); digitalWrite(b3, HIGH); digitalWrite(b4, LOW);  }  void Pbay(){  digitalWrite(b1, HIGH); digitalWrite(b2, HIGH); digitalWrite(b3, HIGH); digitalWrite(b4, LOW);  }  void Ptam(){  digitalWrite(b1, LOW); digitalWrite(b2, LOW); digitalWrite(b3, LOW); digitalWrite(b4, HIGH);  }  void Pchin(){  digitalWrite(b1, HIGH);digitalWrite(b2, LOW); digitalWrite(b3, LOW);digitalWrite(b4, HIGH);  }  void lap() {  Pkhong();delay(500);Pmot();delay(500);Phai();delay(500); Pba();delay(500);Pbon();delay(500);  Pnam();delay(500); Psau();delay(500);Pbay();delay(500);Ptam();delay(500); Pchin();delay(500);  } |

## Bài 3. Kiểm tra giữa kỳ

### Mô tả:

Xây dựng hệ thống nhúng thực hiện điều khiển xe tự hành gồm 3 bánh, trong đó động cơ điều khiển hướng ở bánh trước, 2 động cơ điều khiển 2 bánh sau, phía trước được gắn 1 cảm biến vật cản sử dụng sóng âm HC-SR04 liên tục phát hiện vật cản phía trước.

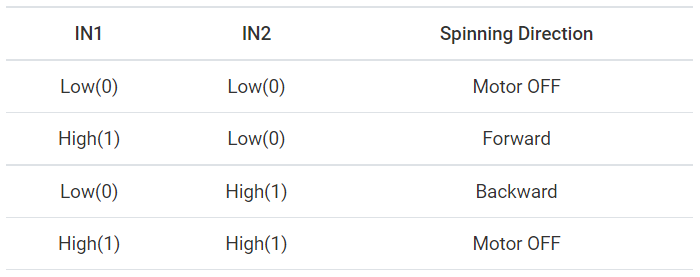
### Sơ đồ thiết kế:



Hình . Hệ thống xe tự hành

### Đặc điểm linh kiện:

* 1 board mạch arduino uno.
* 1 bảng mạch.
* 2 động cơ DC.
* 1 động cơ điều khiển hướng Micro Servo: khi kết nối chân điều khiển của servo motor với cổng số 10 của arduino để nhận tín hiệu điều khiển hướng.
* 1 cảm biến khoảng cách HC-SR04: cảm biến khoảng cách
* 1 IC L293D: là IC cầu H điều khiển động cơ bằng cách thay đổi cực tính của điện áp đầu vào của nó.Cầu H chứa 4 công tắc, với động cơ ở trung tâm từ đó khi đóng đồng thời 2 công tắc cụ thể sẽ đảo ngược cực tính của điện áp đặt vào động cơ . Điều này làm thay đổi hướng quay của động cơ. IC L293D gồm 4 kênh điều khiển 2 động cơ DC.
* IC có 2 chân đầu vào cấp nguồn là VCC1 (16) và VCC2 (8). VCC1 dùng để điều khiển mạch logic bên tay phải là 5V. VCC2 tương tự nhưng điều khiển mạch logic bên trái. Cả 2 đều được đấu chung vào chân đất của IC.
* IC có 4 chân đầu ra để điều khiển 2 động cơ tại các chân OUT1 (3), OUT2 (6), OUT3 (11), OUT4 (14) . Mỗi kênh có thể cung cấp cho động cơ 600 mA, tuy nhiên vẫn phụ thuộc vào hệ thống.
* đối với mỗi kênh của IC đều có 2 loại chân điều khiển cho phép thay đổi tốc độ và hướng quay của động cơ. Các chân này điều khiển động cơ quay về phía trước, hay lùi về phía sau bằng cách điều khiển đóng mở cụ thể các công tắc bên trong IC, các chân IN1 (2), IN2 (7) điều khiển động cơ A, IN3 (10) IN4 (15) điều khiển động cơ B. Chiều quay của động cơ có thể điều khiển bằng cách áp dụng logic HIGH hoặc LOW cho các chân này như sau:



* IC còn có thêm 2 chân để bật, tắt và điều khiển tốc độ là ENA (1), ENB (9). Nếu áp dụng logic HIGH sẽ làm cho động cơ quay, ngược lại LOW làm cho động cơ dừng. Tuy nhiên, có thể điều khiển tốc độ của động cơ bằng cách đấu vào các chân PWM.

### Mã lệnh chính:

|  |
| --- |
| #include <Servo.h>  // Motor A connections  int enA = 9, in1 = 8, in2 = 7;  // Motor B connections  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  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); //chân echo sẽ thu tín hiệu  servo\_10.attach(10, 500, 2500);  }  void loop() {  unsigned long duration; //biến đo thời gian  int distance; //biến lưu khoảng cách  directionControl(); //cho động cơ hoạt động  /\* 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)\*340); // Tính khoảng cách đến vật.  Serial.println(distance);  delay(1000);  if(distance > 20000){  //cham();  nhanh();  }  }  void directionControl() {  analogWrite(enA, 255); //khởi động motor  analogWrite(enB, 255);  // Turn on motor A & B  digitalWrite(in1, HIGH);  digitalWrite(in2, LOW);  digitalWrite(in3, HIGH);  digitalWrite(in4, LOW);  delay(2000);  }  void cham() {  analogWrite(enA, 180);analogWrite(enB, 180);  //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() {  servo\_10.write(20);//quay trái 20  //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);  } |

Bài Làm Thêm:

## Cảm biến vật cản hồng ngoại, điều khiển servo

### Mô tả:

bài thực hành lập trình arduino mô phỏng điều khiển 4 servo quay khi nhận được tín hiệu của cảm biến hồng ngoại. Nếu có vật cản thì servo quay 90 độ, đồng thời tắt led, Ngược lại, nếu gặp vật cản thì trở về trạng thái ban đầu và bật led. Các chân 6,7,8,9 được nối với 4 servo để điều khiển, chân 2 của arduino được dùng để nhận tín hiệu cảm biến.

### Đặc điểm linh kiện:

* 1 mạch arduino uno.
* 4 servo.
* 1 điện trở R = 150 Ω.
* 1 đèn led.
* 1 Cảm biến hồng ngoại.
* 1 logicstate (thanh trạng thái thay thế cho vật cản, 1 – có vật cản, 0-không có vật cản)

### Sơ đồ thiết kế:

### Mã lệnh chính:

|  |
| --- |
| #include <Servo.h> //khai báo thư viện servo  Servo myservo1, myservo2, myservo3, myservo4 ; //tạo 4 đối tượng servo  #define sensor 2  #define led 10  void setup()  {  pinMode(sensor, INPUT);  pinMode(led, OUTPUT);  myservo4.attach(9);//thiết lập các chân kết nối vs motor  myservo3.attach(8);  myservo2.attach(7);  myservo1.attach(6);  }  void loop()  { //  if (digitalRead(sensor) == 1)  {  myservo4.write(0); //nếu gặp vật cản thì quay về trạng thái ban đầu  myservo3.write(0);  myservo2.write(0);  myservo1.write(0);  digitalWrite(led, HIGH);// và bật led  }  else  {  myservo4.write(90); // k phat hien thì sẽ quay 1 góc 90  myservo3.write(90);  myservo2.write(90);  myservo1.write(90);  digitalWrite(led, LOW);//và tắt led  }  } |